



# **Proceedings of The 5<sup>th</sup> Australian Digital Forensics Conference**

**3<sup>rd</sup> December 2007**

**Edith Cowan University  
Mount Lawley Campus**

**Published By**

**School of Computer and Information Science  
Edith Cowan University  
Perth, Western Australia**

**Edited by**

**Dr. Craig Valli and Dr. Andrew Woodward  
School of Computer and Information Science  
Edith Cowan University  
Perth, Western Australia**

**Copyright 2007, All Rights Reserved**

**ISBN 0-7298-0646-4**

## Table of Contents

1. Anti-Forensics and the Digital Investigator.....	1
2. An approach in identifying and tracing back spoofed IP packets to their sources .....	8
3. The effectiveness of investigative tools for Secure Digital (SD) Memory Card forensics.....	22
4. An overview and examination of digital PDA devices under forensics toolkits .....	34
5. Profiling Through a Digital Mobile Device.....	52
6. Forensic Analysis Avoidance Techniques of Malware .....	59
7. ID Theft: A Computer Forensics' Investigation Framework.....	67
8. Extracting Inter-arrival Time Based Behaviour from Honeypot Traffic using Cliques .....	79
9. Multi-Step Scenario Matching Based on Unification .....	88
10. Steganalysis in Computer Forensics .....	98
11. Managing Digital Forensic Knowledge An Applied Approach .....	109
12. ADSL Router Forensics Part 1: An introduction to a new source of electronic evidence.....	119
13. An examination of the Asus WL-HDD 2.5 as a Nepenthes malware collector.....	128
14. A Proof-of-Concept Project for Utilizing U3 Technology in Incident Response.....	136
15. Introduction to Mobile Phone Flasher Devices and Considerations for their Use in Mobile Phone Forensics.....	143
16. Pocket SDV with SDGuardian: A Secure & Forensically Safe Portable Execution Environment .....	154
17. Can SDV Technology be Utilised in a Smartphone to Prevent Forensic Analysis?.....	164
18. A forensically tested tool for identification of notebook computers to aid recovery: LIARS phase I proof of concept.....	179
19. Mood 300 IPTV decoder forensics.....	185
20. A Methodology for the Forensic Acquisition of the TomTom One Satellite Navigation System – A Research in Progress.....	195
21. BLOGS: ANTI-FORENSICS and COUNTER ANTI-FORENSICS.....	199
22. An Overview of ADSL Homed Nepenthes Honeypots In Western Australia.....	204
23. Tracing USB Device artefacts on Windows XP operating system for forensic purpose.....	210
24. Oops they did it again: The 2007 Australian study of remnant data contained on 2 <sup>nd</sup> hand hard disks	219

## **Conference Foreword**

This year has seen the conference grow in size and magnitude yet again. There are several definite strands of established research and interest within the subject of computer based forensics these include disk sanitisation, honeypots and discovery techniques. We are seeing emergent papers in the areas of mobile device forensics including PDAs, mobile phones and GPS devices. The papers authors are drawn from a cross section of the forensics community from practitioners to academics.

All papers were subject to a double blind peer review process and of the 42 papers submitted only 24 were accepted for final publication.

Conferences such as these are simply not possible without willing volunteers who follow through with the commitment they have initially made and I would like to take this opportunity to thank the conference committee for their tireless efforts in this regard. These efforts have included but not been limited to the reviewing and editing of the conference papers, helping with the planning, organisation and execution of the conferences.

To our sponsors also a vote of thanks for both the financial and moral support provided to the conference. Finally, to the administrative and technical staff of the School of Computer and Information Science for their contributions to the running of the conference.

Dr Craig Valli  
Conference Chair

### **Conference Organising Committee**

Dr Craig Valli	Conference Chair & Co – Editor	Edith Cowan University
Dr Andrew Woodward	Conference Editor	Edith Cowan University
Chris Bolan	Committee Member	Edith Cowan University
Dr Trish Williams	Committee Member	Edith Cowan University
Professor Bill Hutchinson	Committee Member	Edith Cowan University
Lisa McCormack	Committee Member	Edith Cowan University
Rebecca Treloar-Cook	Committee Member	Edith Cowan University

### **Sponsors**

Secure Systems

Cengage

Research Network for Secure Australia

Best Paper Award

Best Presentation Award

## Anti-Forensics and the Digital Investigator

Gary C. Kessler  
Champlain College  
Burlington, VT, USA  
[gary.kessler@champlain.edu](mailto:gary.kessler@champlain.edu)

Edith Cowan University  
Mount Lawley, WA, Australia

### Abstract

*Viewed generically, anti-forensics (AF) is that set of tactics and measures taken by someone who wants to thwart the digital investigation process. This paper describes some of the many AF tools and methods, under the broad classifications of data hiding, artefact wiping, trail obfuscation, and attacks on the forensics tools themselves. The concept of AF is neither new nor solely intended to be used by the criminal class; it also has legitimate use by those who wish to protect their privacy. This paper also introduces the concept of time-sensitive anti-forensics, noting that AF procedures might be employed for the sole purpose of delaying rather than totally preventing the discovery of digital information.*

### Keywords

Anti-forensics, data hiding, artefact wiping, trail obfuscation, attacks on computer forensics tools, privacy

## INTRODUCING ANTI-FORENSICS

The term *anti-forensics* (AF) has recently entered into the vernacular of digital investigators. Although conceptually not new, it is instructive to observe that there is no clear industry definition (Harris, 2006). Rogers (2006), a practicing digital forensics educator and investigator, defines AF as "attempts to negatively affect the existence, amount, and/or quality of evidence from a crime scene, or make the examination of evidence difficult or impossible to conduct." Liu and Brown (2006), practicing creators of AF methods and tools, offer a slightly darker definition: "application of the scientific method to digital media in order to invalidate factual information for judicial review."

The term *forensics* is significant and quite specific -- whatever AF is pertains to the scientific analysis of evidence for court. Anti-forensics, then, is that set of tools, methods, and processes that hinder such analysis.

It is difficult to think of any *legitimate* uses of AF processes and tools. Indeed, most -- if not all -- digital investigators find AF to be the bane of their existence and only used by someone who has something to hide.

But AF might also be employed by the person who just wants to be left alone. An early text about computer forensics devoted significant time to the examination process as well as ways to thwart that process, all in the name of privacy (Caloyannides, 2001). One can argue that it is a fine line between protecting one's privacy and preventing a court-sanctioned search, but that line has existed for centuries -- only with digital devices does a letter that the writer burned well in the past continue to hang around on a hard drive or backup tape. And there are those that will argue that AF techniques can protect a Good Person from a Bad Government. Of course, those same tools can block a Good Government from investigating a Bad Person and that, of course, is the rub.

Laws, traditions, mores, and culture affect a society's view of privacy. Many nations purport to value people's privacy on the one hand but also recognize that the "right to privacy" -- should that right exist in a particular jurisdiction -- is not absolute.

## CATEGORIES OF ANTI-FORENSICS METHODS

Much of the recent discussion in articles, conferences, and blogs seems to suggest that AF tools and methods have appeared suddenly (Harris, 2006; Liu & Brown, 2006; Rogers, 2006). While this is certainly true in some cases, it is also important to note that many forms of AF -- although not created to hinder the detection of evidence, per se -- have been around for quite some time.

Rogers (2006) suggests that there are four basic categories of AF: data hiding, artefact wiping, trail obfuscation, and attacks against the computer forensics process or tools. This section will provide examples of tools and processes that fit in these categories.

### **Data Hiding**

Data hiding can be accomplished in a variety of ways. Hidden writing (i.e., *steganography*) has been around for over two thousand years. Digital steganography tools have been available since at least the mid-1990s and stego software is available for almost every computer operating system. *Any* form of digital information can be stored inside many types of carrier files, including image, audio, video, and executable files (StegoArchive.com, 2005).

Low-technology stego methods can also be employed to hinder computer forensics. As an example, a person can hide a picture, table, or text block under an image in a PowerPoint or Impress graphical presentation. Alternatively, a white text block over a white background can store a hidden message. Morse code messages can be embedded in a picture. Null ciphers form messages by selecting a pre-determined pattern of letters from a sequence of words. Many other forms of low-tech stego can be employed that no automated tool can detect (Kessler, 2004).

Covert channels in data communications protocols allow hidden communication over public and private networks. The Transmission Control Protocol/Internet Protocol (TCP/IP) suite, for example, has several weaknesses that can be exploited to allow covert communications. The concept of covert channels in networking protocols dates back at least 30 years (Ahsan, 2002; Rowland, 1996).

There are numerous ways to hide data, at least from cursory searches. Data can be hidden in the slack and unallocated spaces on computer hard drives, as well as the metadata of many types of files. The Master Boot Record (MBR), allocated but unused device driver registers and tables, protected system areas, and hidden (and possibly encrypted) partitions on hard drives are other places to secret information, as well as the Basic Input/Output System (BIOS) chip itself (Budimir & Slay, 2007). Homographic file names (where non-Latin characters that appear similar to Latin characters are used in the name), use of very long path names (greater than 256 characters), and use of hidden characters (e.g., 0xFF) in file names can all be used to hide data from the operating system. Data can also be hidden in closed sessions on compact discs, in another user's disk space on a poorly secured server, or out in the open using public shares. All of this data can still be found by forensics tools and the astute examiner, but they are harder to find and harder to explain to the non-technical audience.

### **Artefact Wiping**

Artefact wiping tools have been available for many years. Wiping programs such as BC Wipe, Eraser, and PGP Wipe destroy data files using multiple overwrites that makes any retrievable impractical, if not impossible.

Automated artefact wiping software is available, ostensibly, to allow a user to recover storage space -- and protect one's privacy -- by removing unneeded temporary files that clutter up the hard drive. But software such as Evidence Eliminator, Secure Clean, and Window Washer remove browser history and cache files, delete certain operating system files, and wipe slack and unallocated space. Many of these programs come preconfigured to eliminate the detritus of various operating systems and common utilities (e.g., Windows, MS Office, Internet Explorer, AOL Instant Messenger, and Firefox), and have additional plug-ins for a large number of other common applications (e.g., Eudora, Picasa, RealAudio, and WinZip). Many guides are available that directly address which files should be cleaned out so as to make analysis difficult or to render forensics largely moot (Caloyannides, 2001; Geiger & Cranor, 2006).

The best of these tools do not, of course, merely delete offensive files -- which would be the benign approach -- but wipe them. Deleting unnecessary files is sufficient to recover disk space; wiping the files certainly does suggest that someone is interested in more than simple space recovery.

Artefact wiping tools make analysis more difficult for forensics examiners but the fact is that they are not perfect. Most of the programs leave identifiable traces of the wiping and many are not as complete as they advertise to be, often leaving behind remnants of the very things they are supposed to delete (Geiger & Cranor, 2006).

### **Trail Obfuscation**

Trail obfuscation has been an issue since the 1970s with logon spoofing followed in the 1980s with IP and Medium Access Control (MAC) address spoofing. Denial-of-service (DoS) and distributed DoS attacks -- widespread since 1996 -- depend upon successful IP address spoofing and have made network intrusions more difficult to investigate. Indeed, defences to DoS/DDoS attacks have been more about prevention, response, and

recovery rather than detection of the attacker, although some methods of IP traceback have been devised in order to track packets on the network back to their source (Levine & Kessler, 2002). So-called *onion routing*, however, can make network traffic analysis nearly impossible (Forte, 2002).

There are a variety of ways to confuse e-mail investigations. E-mail anonymizers ostensibly provide privacy services, preventing an investigator from determining the source of a sent mail message. Planting false headers, open Simple Mail Transfer Protocol (SMTP) proxies, and anonymous Secure Shell (SSH) tunnel servers are among the other mechanisms that can add complexity to tracking back the origins of e-mail. Web anonymizers hide a Web site user's identity and anonymity toolsets such as Tor can effectively bring an Internet-based investigation to a halt (Akin, 2003; EFF, 2007).

Trail obfuscation can also be accomplished by wiping and/or altering server log files and/or system event files, or altering the dates of various files (e.g., using `touch`). A Bad Guy hacker who can break into a server very likely has the knowledge to hide his/her tracks and/or leave false clues by modifying these files.

### Attacks Against Computer Forensics Tools

Direct attacks on the computer forensics process are the newest type of AF and potentially the most threatening. Palmer (2001) describes six phases in the process of digital forensics; all are open to attack:

1. *Identification* refers to the method by which an investigator learns that there is some incident to investigate. This phase can be undermined by obscuring the incident, or hiding the nexus between the digital device and the event under investigation.
2. *Preservation* describes the steps by which the integrity of the evidence is maintained. This phase can be undermined by interrupting the evidentiary chain or calling into doubt the integrity of the evidence itself.
3. *Collection* is the process by which data from the evidence medium is acquired. This step can be undermined by limiting the completeness of the data being collected or calling into question the hardware, software, policies, and procedures by which evidence is gathered.
4. *Examination* addresses how the evidence data is viewed. This part of the process can be undermined by showing that the tools themselves are inadequate, incomplete, or otherwise not scientifically valid.
5. *Analysis* is the means by which an investigator draws conclusions from the evidence. This phase relies on the tools, investigative prowess of the examiner, and the rest of the evidence that was found. If a case hinges solely on digital evidence, the interpretation of the evidence is the part most open to attack.
6. *Presentation* refers to the methods by which the results of the digital investigation are presented to the court, jury, or other fact-finders. If the evidence is otherwise solid, anti-forensics tools and methods will be used to attack the reliability and thoroughness of the reports -- or the examiner.

Courts throughout the world have long had to deal with scientific evidence and have had to establish rules for what is acceptable and unacceptable in this realm. In the U.S., the guiding principle in federal courts and many state courts is patterned after the seminal case of *Daubert v. Merrell Dow Pharmaceuticals* (Supreme Court of the United States, 1993). According to *Daubert*, a judge can determine the admissibility of scientific evidence based upon four factors:

- *Testing*: Can -- and has -- the procedure been tested?
- *Error Rate*: Is there a known error rate of the procedure?
- *Publication*: Has the procedure been published and subject to peer review?
- *Acceptance*: Is the procedure generally accepted in the relevant scientific community?

Anti-forensics procedures, then, can attack the reliability of digital evidence; if the reliability of the evidence can be successfully called into question, it becomes worthless in a court of law. In fact, Van Buskirk and Liu (2006) argue that forensics software seems to have been granted a presumption of reliability by the courts that may be undeserved -- and actually in conflict with *Daubert*.

There have already been successful attacks on many of the major computer forensics tools, including EnCase, FTK, iLook, SleuthKit, and WinHex. Programs that fiddle with FAT directories, NTFS master file tables, and ext inodes have been around for years, as well as programs that write to file slack, alter file signatures, and flip bits in order to evade hashset detection (Rogers, 2006).

An example of the tension between the AF community and software providers is exemplified by a report from the 2007 U.S. Black Hat conference. In this instance, a group presented the results of applying several exploitation techniques to a number of commercial and open-source computer forensics applications (Guidance Software, 2007; Palmer, Newsham, Stamos, & Ridder, 2007). They concluded that:

- Forensics software vendors do not design their products to function in a hostile environment; i.e., they do not generally create software that could acquire evidence from machines that have been configured to withstand or falsify evidence during acquisition by known forensic tools.
- Forensics software developers do not create products that are properly protected against such flaws as stack overflows, improper management of memory pages, and unsafe exception handling leakage.
- Forensics software users do not apply sufficiently strong criteria to the evaluation of the products that they purchase. In fact, most computer forensics labs purchase the software that "everyone else" is using and do not perform independent tests of reliability, thoroughness, and veracity, particularly as new versions of the software get released.

If the aim of anti-forensics is to render moot digital evidence, then calling into question the effectiveness of the very tools that we use to find, analyse, examine, and report on this evidence will have a chilling effect on the entire digital investigation community. In the final analysis, the computer examiner may find all of the evidence that is present and interpret it correctly -- but if it is not believed in court, then the entire exercise is meaningless.

## **ADDITIONAL ASPECTS OF ANTI-FORENSICS**

### **The Metasploit Project**

The Metasploit Project is an open-source collaborative with a stated goal of providing information to the penetration testing, intrusion detection system (IDS) signature development, and information system exploit research communities (Metasploit LLC, 2007b). The Metasploit Anti-Forensics Project (Metasploit LLC, 2007a), in particular, has the stated goal of investigating the shortcomings in computer forensics tools, improving the digital forensics process, and validating forensics tools and processes. One output of this project has been the Metasploit Anti-Forensic Investigation Arsenal (MAFIA), a suite of programs that includes:

- *Sam Juicer* -- A program that acquires the hashes from the NT Security Access Manager (SAM) file without touching the hard drive
- *Slacker* -- A program to hide files within the slack space of NTFS files
- *Transmogrify* -- "Defeats" EnCase's file signature detection capabilities by allowing a user to mask and unmask files as any file type
- *Timestomp* -- A program that alters all four NT File System (NTFS) file times: modified, access, creation, and file entry update (so-called MACE times)

These tools represent a combination of a demonstration of capability as much as practical ways in which a user can confuse digital forensics examinations; the software authors acknowledge that the software does not necessarily take the AF process to the n-th degree. Timestomp, for example, modifies only the MACE times stored in a file's \$STANDARD\_INFORMATION attribute and those not in the \$FILE\_NAME attribute, thus, leaving some indicator of suspicious activity. It is only a minor modification to the program, however, to make them more thorough (Liu & Brown, 2006).

While the MAFIA tools are successfully proving their point, they can also be used for other purposes -- such as hiding a guilty party's incriminating evidence or placing incriminating evidence on the drive of an innocent party.

### **Cryptography**

Cryptography, in some ways, is the ultimate anti-forensics tool. And, of course, it is not new -- crypto tools have made, and will continue to make, digital investigations difficult or impossible. Cryptography is perhaps the most troublesome AF tool because it is easy for the general user to employ and, once turned on, required minimal maintenance or attention on the part of the user. That said, many applications use weak encryption which can easily be defeated (e.g., Wired Equivalent Privacy [WEP]) or users may not manage their keys well.

Crypto protection comes in a variety of flavours. Many applications, such as Adobe Acrobat, MS Office, and WinZIP, provide mechanisms so that users can password protect and/or encrypt individual files. Pretty Good Privacy (PGP) has been encrypting e-mails since the early-1990s. Encrypting file systems and whole disk



encryption (e.g., PGP Desktop, SafeBoot, Vista with BitLocker, and Windows EFS) will continue to thwart -- or, at least, resist -- computer forensics examinations. Encrypted Internet-based communication (e.g., Secure Sockets Layer, Transaction Layer Security, virtual private networks, and IEEE 802.11 secure wireless networks) can make analysis of the contents of network traffic nearly impossible (Casey, 2004).

### The User

One would assume that there is a cadre of users who will employ every tool in the arsenal to make computer examinations difficult. In general, there is a linear relationship between the difficulty in using an AF tool and how much the user really has to hide. As it is today,

- Not every user is going to install AF tools
- Not every user who installs AF tools will use them consistently, thus leaving behind usable information
- Not every user who uses AF tool will use them correctly, which will leave usable information
- Not all AF tools work as perfectly as advertised, thus leaving remnants and traces

### TIME-SENSITIVE ANTI-FORENSICS

Another purpose for anti-forensics may be to "protect" certain data until it is moot. Rather than prevent forensics analysis from occurring, it may be sufficient to bog the examination process down until the data loses its evidentiary -- or intelligence -- value.

This is conceptually similar to the information security concept of *time-based security* (Schwartau, 1999). That model suggests that an information security system does not need to keep an attacker out forever but only long enough for the attack to be detected and for a response to be mounted. Expressed as a formula:

$$PS_t > D_t + R_t$$

where  $PS_t$  = the length of time that the security system can protect against an attack;  $D_t$  = the length of time to detect an attack; and  $R_t$  = the length of time to respond to an attack.

The user of anti-forensics methods wants to keep the digital investigator at bay for some period of time (possibly forever). As a formula, this might be expressed:

$$PAF_t > I_t + AQ_t + E_t + AN_t$$

where  $PAF_t$  = the length of time that the anti-forensics method can protect data against discovery;  $I_t$  = the length of time to identify potentially useful data;  $AQ_t$  = the length of time to acquire the data;  $E_t$  = the length of time to examine the data; and  $AN_t$  = the length of time to analyse the data.

In the security model, detection and response to an attack is usually automated and requires a relatively short period of time (i.e., seconds, minutes, or hours). Conversely, identification, acquisition, examination, and analysis of digital evidence require human intelligence and, generally, a relatively long period of time (i.e., days or weeks) even in the best of circumstances. In some cases the use of AF methods only adds a little to an otherwise lengthy time delay but in other cases can frustrate an active, time-sensitive investigation.

It is worth considering the notion that some class of users actually *wants* digital examinations to take place but to be slowed down, wasting the time, money, and personnel resources of the investigating agency. This class of user does not want to prevent the exam, per se; if so, they might just use 256-bit whole disk encryption and be done with it. Instead, they want the exams to take place, but they want them to take a long time. In some sense, they are trying to confuse *their* adversary by inundating them with information but keeping it at arm's length.

### CONCLUSION

This paper has attempted to broaden the scope of those methods and processes that could be fairly described as *anti-forensics*. It has identified many -- certainly not all -- AF processes and further work is necessary to categorize these methods based upon their ease (and, therefore, likelihood) of use, efficacy, degree of difficulty to detect and/or overcome, and other characteristics.

Every attempt to make the digital forensics process harder has been met with changes in the process to address the new challenges. Encryption taught investigators to think twice before the pulling the plug on a computer; live imaging of both hard drives and RAM is becoming much more common in the field today. Steganography

taught investigators to not take everything at face value. Malware taught investigators that the scene of the computer exam might be hostile. AF software that attacks our tools may or may not result in better tools but will certainly cause us to change the process so as to rely less on fully automated exams; more human intelligence -- and time -- will be needed for investigations and this will also demand more detailed knowledge of file systems.

It is important to note that while many of the AF methods might make information derived from an investigation useless as evidence in court, they may not diminish the intelligence value of the information; *reasonable doubt* in the mind of a jury does not translate into non-actionable information for an intelligence gatherer. Other AF methods, of course, do secure data from the digital investigation process although it is unclear where the crossover of the value of the information retrieved and the "cost" of the resources expended to retrieve the data occurs.

Anti-forensics tools and methods will continue to provide difficulties and challenges to the digital investigation and e-discovery communities. As AF developers continue to produce tools, however, it becomes incumbent upon academia and industry to coordinate and fund anti-AF research and development. This may well be the next New Thing in digital investigations.

## **ACKNOWLEDGEMENTS**

The author is partially supported by Grant No. 2006-DD-BX-0282 awarded by the Bureau of Justice Assistance. The Bureau of Justice Assistance is a component of the Office of Justice Programs, which also includes the Bureau of Justice Statistics, the National Institute of Justice, the Office of Juvenile Justice and Delinquency Prevention, and the Office for Victims of Crime. Points of view or opinions in this document are those of the author and do not represent the official position or policies of the United State Department of Justice.

## **ABOUT THE AUTHOR**

Gary C. Kessler, Ed.S., CCE, CISSP is director of the Champlain College Center for Digital Investigation (C3DI) and an associate professor and director of the Computer & Digital Forensics program at Champlain College in Burlington, Vermont, and an adjunct associate professor at Edith Cowan University in Mount Lawley, Western Australia. He is a member of the High Technology Crime Investigation Association (HTCIA) and International Society of Forensic Computer Examiners (ISFCE). Kessler is also a technical adviser to the Vermont Internet Crimes Against Children (ICAC) and Internet Crimes Task Forces, a member of the editorial board of the *Journal of Digital Forensics, Security and Law (JDFSL)*, an associate editor of the *Journal of Digital Forensic Practice (JDFP)*, and a principal in GKS Digital Services, LLC.

## **REFERENCES**

- Ahsan, K. (2002). *Covert Channel Analysis and Data Hiding in TCP/IP*. Thesis presented to the Edwards S. Rogers Sr. Graduate Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario. Retrieved September 11, 2007, from <http://gray-world.net/papers/ahsan02.pdf>
- Akin, T. (2003). WebMail Forensics. BlackHat Briefings. Retrieved September 11, 2007, from <http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-akin.pdf>
- Budimir, N., & Slay, J. (2007). Identifying Non-Volatile Data Storage Areas: Unique Notebook Identification Information as Digital Evidence. *Journal of Digital Forensics, Security and Law*, 2(1), 75-91.
- Caloyannides, M.A. (2001). *Computer Forensics and Privacy*. Boston: Artech House.
- Casey, E. (2004). *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet* (2nd ed.). London: Elsevier Academic Press.
- Electronic Frontier Foundation (EFF). (2007, September 18). Tor Web page. Retrieved September 18, 2007, from <http://tor.eff.org/index.html.en>
- Forte, D. (2002, August). Analyzing the Difficulties in Backtracking the Onion Router's Traffic. *Proceedings of the 2002 Digital Forensics Research Workshop*. Retrieved September 11, 2007, from [https://www.dfrws.org/2002/papers/Papers/Dario\\_Forte.pdf](https://www.dfrws.org/2002/papers/Papers/Dario_Forte.pdf)
- Geiger, M., & Cranor, L.F. (2006, September/October). Scrubbing Stubborn Data: An Evaluation of Counter-Forensic Privacy Tools. *IEEE Security & Privacy*, 4(5), 16-25.
- Guidance Software. (2007, July 26). Guidance Software Response to iSEC Report. Retrieved September 11, 2007, from <http://www.securityfocus.com/archive/1/474727>

- Harris, R. (2006). Arriving at an Anti-Forensics Consensus: Examining How to Define and Control the Anti-Forensics Problem. *Proceedings of the 2006 Digital Forensics Research Workshop. Digital Investigation*, 3(S), S44-S49. Retrieved September 11, 2007, from <http://dfrws.org/2006/proceedings/6-Harris.pdf>
- Kessler, G.C. (2004, July). An Overview of Steganography for the Computer Forensics Examiner. *Forensics Science Communication*, 6(3). Retrieved September 11, 2007, from [http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004\\_03\\_research01.htm](http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004_03_research01.htm)
- Levine, D.E., & Kessler, G.C. (2002). Denial of Service Attacks. In M. Kabay & S. Bosworth (Eds.), *Computer Security Handbook*, 4th ed. New York: John Wiley & Sons.
- Liu, V., & Brown, F. (2006, April 3). Bleeding-Edge Anti-Forensics. Presentation at InfoSec World 2006. Retrieved September 11, 2007, from [stachliu.com/files/InfoSecWorld\\_2006-K2-Bleeding\\_Edge\\_AntiForensics.ppt](http://stachliu.com/files/InfoSecWorld_2006-K2-Bleeding_Edge_AntiForensics.ppt)
- Metasploit LLC. (2007a). Metasploit Anti-forensics home page. Retrieved September 11, 2007, from <http://www.metasploit.com/projects/antiforensics/>
- Metasploit LLC. (2007b). Metasploit Project home page. Retrieved September 11, 2007, from <http://www.metasploit.com/>
- Palmer, C., Newsham, T., Stamos, A., & Ridder, C. (2007, August 1). Breaking Forensics Software: Weaknesses in Critical Evidence Collection. Abstract of presentation at Black Hat USA 2007. Retrieved September 11, 2007, from <http://www.blackhat.com/html/bh-usa-07/bh-usa-07-speakers.html#Palmer>
- Palmer, G. (2001, November 6). *A Road Map for Digital Forensics Research*. Digital Forensic Research Workshop (DFRWS) Technical Report (DTR) T001-01 Final. Retrieved September 11, 2007, from <http://www.dfrws.org/2001/dfrws-rm-final.pdf>
- Rogers, M. (2006, March 22). Panel session at CERIAS 2006 Information Security Symposium. Retrieved September 11, 2007, from <http://www.cerias.purdue.edu/symposium/2006/materials/pdfs/antiforensics.pdf>
- Rowland, C.H. (1997, May 5). Covert Channels in the TCP/IP Protocol Suite. *First Monday*, 2(5). Retrieved September 11, 2007, from [http://www.firstmonday.org/issues/issue2\\_5/rowland/](http://www.firstmonday.org/issues/issue2_5/rowland/)
- Schwartau, W. (1999). *Time Based Security*. Seminole, FL: Interpact Press.
- StegoArchive.com. (2005). Stego Archive Web site. Retrieved September 11, 2007, from <http://www.stegoarchive.com>
- Supreme Court of the United States. (1993). *Daubert v. Merrell Dow Pharmaceuticals* (92-102), 509 U.S. 579. Retrieved September 11, 2007, from <http://supct.law.cornell.edu/supct/html/92-102.ZS.html>
- Van Buskirk, E., & Liu, V.T. (2006, March). Digital Evidence: Challenging the Presumption of Reliability. *Journal of Digital Forensic Practice*, 1(1), 19-26.

## **COPYRIGHT**

Gary C. Kessler ©2007. The author assigns SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.

## **An approach in identifying and tracing back spoofed IP packets to their sources**

Krishnun Sansurooah  
School of Computer and Information Science (SCIS)  
Edith Cowan University Perth, Western Australia.  
Email: ksansuro@student.ecu.edu.au

### **Abstract**

*With internet expanding in every aspect of businesses infrastructure, it becomes more and more important to make these businesses infrastructures safe and secure to the numerous attacks perpetrated on them conspicuously when it comes to denial of service (DoS) attacks. A Dos attack can be summarized as an effort carried out by either a person or a group of individual to suppress a particular outline service.*

*This can hence be achieved by using and manipulating packets which are sent out using the IP protocol included into the IP address of the sending party. However, one of the major drawbacks is that the IP protocol is not able to verify the accuracy of the address and has got no method to validate the authenticity of the sender's packet. Knowing how this works, an attacker can hence fabricate any source address to gain unauthorized access to critical information. In the event that attackers can manipulate this lacking for numerous targeted attacks, it would be wise and safe to determine whether the network traffic has got spoofed packets and how to traceback. IP traceback has been quite active specially with the DOS attacks therefore this paper will be focusing on the different types of attacks involving spoofed packets and also numerous methods that can help in identifying whether packet have spoofed source addresses based on both active and passive host based methods and on the router-based methods.*

### **INTRODUCTION**

Referring to RFC 1791, (1981) packets that are sent out using the IP protocol include the IP address of the sender. After receiving this packet, the recipient directs replies to the sender using the original source address. Nonetheless, the correctness of this address is not verified by the IP protocol that unfortunately has no way of validating the packet's source if not only been based on the sender's IP address. Therefore this involves that an attacker can at any pointing time fake the source address and act as a legitimate party to gain access to unauthorized details or information. Most of the time sending spoofed packet is carried out to legitimately access unauthorized information.

Spoofing of network traffic can certainly occur at many different layers. One of the layers that could be affected by is the network layer which is responsible in dealing with MAC spoofing or at a non IP transport layer such as IPX, NetBEUI or even at an application layer in the instance of Email spoofing.

Even through tough access control technologies such as firewalls which are mainly used in protecting the network, they are helpless when it comes to specific attacks ranging from SYN-flood, TCP connection spoofing, Smurf and many more. Subsequently with these attacks increasing, more and more companies are now turning towards deploying Intrusion Detection System (IDS) onto their network. However, it does detect the network attacks and hence display the alerts but unfortunately does not identify the attacker's source. This is quite enigmatic especially when it comes to DOS attacks because the attacker normally can remain masked due to the fact that these attacks, the attacker does not have to interact with the targeted host as s/he does not need to receive any packet as s/he is initialing the attack. The focus of this report is to illustrate an overview of the numerous ways that can be used to determine whether the received IP packets on the network has been spoofed and if so, how to trace them back to their originators or their source addresses. This process is most common known as IP traceback. The main concept behind the IP traceback is to determine the exact IP address of the source from which the attacks are being launched. Despite that this can normally be gathered by locating the IP address field from the IP packet, the attacker can unfortunately easily manipulate and changed these details, thus masking its original and true identity.

However, the concept of IP traceback is not well defined as to what it normally should be performing. Its purpose is mainly to identify the true IP address and the source of its attacker, in other words, the ability of identifying the source of a particular IP packet, its destination and an approximate time of reception of the packet. IP traceback can hence be summarized as belonging to two different methods: proactive and reactive.

### **Proactive Tracking Methodology**

This method would involve detecting and tracing attacks when packets are in transit. If packet tracing is needed, the victim can therefore refer to this information to identify the attacking source. However the proactive methods can be further split into two different proactive methods namely marking and packet messaging respectfully and described below

#### **Packet Marking**

This would involve packets that contain information about each and every router that they go through as they (IP packets) has through the network. Therefore, this means that the receiver of the designated packet can make use of the information held by the router to trace back packet's route to its originator. It is imperative that routers can imprint and sign packets without interrupting the normal packet processing.

#### **Message Marking**

In this particular approach, the different routers, through which the packets travel across, generate and broadcast messages with call the information about the forwarding nodes that a particular packet transit across.

### **Reactive Tracking Methodology**

The reactive tracing method operates differently to the proactive one. In this approach, the tracing will only commence when an attack has been perpetrated and following its detection. However, the numerous trials in developing a practical traceback algorithm and packet matching techniques have tried to resolve these dilemmas. Among those analyzed approaches are hop-by-hop tracing, IPSec authentication and monitoring traffic patterns matching.

The focus of this paper will be to identify the various types of attacks involved with spoofed packets and also how to analyze the tracking back of suspicious packets to their originator(s). Whilst, accessing the routers logs about all the data packets that have been passed through and even to other nodes. However, the methodology used in this particular approach will be reactive and using the hop-by-hop tracing with a probabilistic packet marking scheme.

## **IP TRACEBACK**

During the past decade, a lot of attention has been focused on the security of Internet infrastructure in place as being part of transmission, reception and storage of paramount importance within the increasing e-commerce applications. Yet, with high-profile Distributed Denial-of Service (DDOS) attacks, numerous ways have been elaborated to identify the source of these attacks and one methodological approach is using IP traceback.

Normally, IP traceback is not restricted to only DOS or DDOS attacks but with the ability to forge the IP address of those packets make the identification of the originator even harder and in such routine approaches of locating the system (attacker) with a given IP address (e.g. Traceroute) is no longer feasible due to the fact that the packet has already been spoofed. Belenky, A. & Ansari, N. (2003) implies that more advanced approaches of locating the source of attacking packets are needed. They also mentioned that identify the source of the offending packet would not necessarily means identifying the attack originator as these packets may be a host linked in a chain a reflector, a zombie or a device that by the attacker at an earlier stage. However, they did mention that IP traceback approaches are not meant to impede or cease those attacks but they are used to identify the source(s) of the initial incriminating packets during and after the attack.

## **IP TRACEBACK CLASSIFICATION**

### **End-host IP Traceback**

#### **Probabilistic Packet marking (PPM)**

This approach is structured around the routers that imprint the packets that flow through them through either their address or part of their address. This is normally carried out in a randomly process. This PPM technique also known as hop-by-hop tracing introduced by Savage, S. et al. (2001) originally and hence been improved by Song, D.X. & Derrig, A. (2001) in its coding and security was primarily targeted at both DOS and DDOS.

Figure 1 below gives an overview of how the PPM works where attacker would launch an attack towards victim A and as shown in Figure 1, assuming that the packet travel path is R1, R2, R4 and R12, each router enforcing PPM recognize the packet flow and prior to routing them to their destination, it probabilistically imprint them with its partial address – i.e. the router's partial address into to IP address header. Therefore when the victim acknowledges reception of enough packets, it can then remodel the addresses of all PPM-enabled packets and hence reconstruct the attack path. Obviously to reconstruct the whole attack path, a large number of packets would be required as well as the reconstruction of the data frames.

We do note that PPM can deal with the modification of packet which are directed to the victim. But, when it comes to packet generation transformation by a reflector, traceback will only be obviously, the traffic will become fragmented and will be corrupted without having whatsoever impact on the traceback. This means that when fragmentation takes place, usually the ID field is the field that is being marked and if only a single fraction of the source is marked, the reassembly process will not be possible at the destination. Even that this might pose a problem, traceback would still be in a position of retracing the path due to the fact that the marking would have taken place before the reassembly process. This is resolved by opting for a much reduced option in the marking of the packets but that need to be understood is that on doing so, this will definitely increases the number of packets needed for reconstructing the path.

Another issue with PPM is when using tunneling technology, it must be ensured that the markings are not removed prior to the headers are removed.

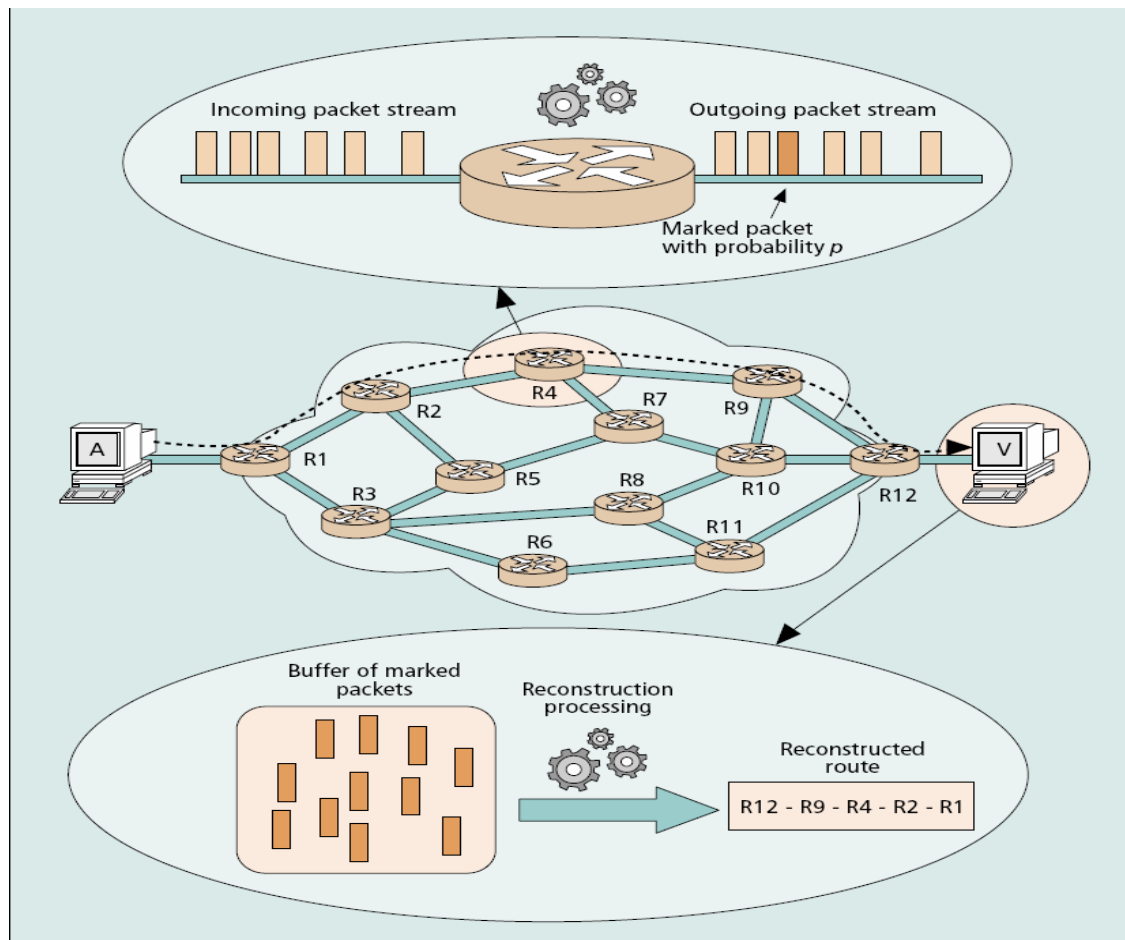


Figure 1 Probabilistic packet marking (Belenky, A. & Ansari, N. (2003))

## ICMP Traceback

With the ICMP traceback, the concept of tracking down the full pathway of the intrusion is completely different from PPM. In figure 2 there is an illustration of how an ICMP traceback schema operates. According to Belenky, A. & Ansari, N. (2003) every router on the network is set up in such a way that they have the ability to pick any packet at random (one in every 20,000 packets recommended) and hence produce an ICMP traceback message which would be targeted to the corresponding destination of the selected packet. The ICMP traceback message would normally consist of the following information:

- i) The previous hop,
- ii) The next hop,
- iii) A timestamp of the packet

However, since there are numerous bytes of the traced packet which are possible, which are duplicated in the payload of the traceback message. Therefore the Time-To-Live (TTL) field is extended to 255 in order to be used in identifying the attack pathway with the ICMP traceback; the routers sitting on the network pathway will produce a completely new packet with an ICMP traceback message. The entire opposite of how PPM would handle this situation. The traceback information was entirely in-band. If we go by the assumption that the victim is under a DOS attack it would definitely means that the volume of packets going through would be huge hence afterward capturing all the addresses of the different routers on the attack pathway that generate the traceback message.

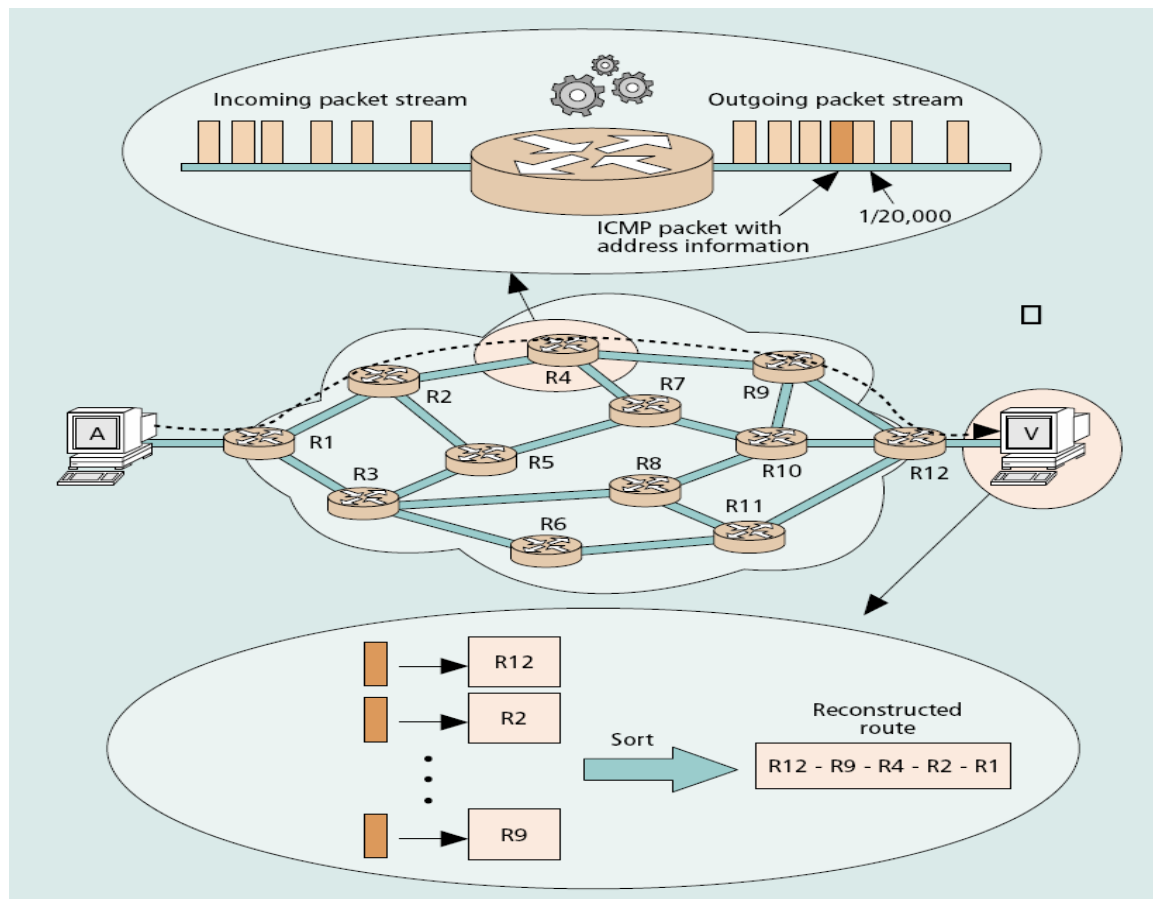


Figure 2 shows the process of the ICMP Traceback (Belenky, A. & Ansari, N. (2003))

Wu, F.S et al. (2001) did mention that while this methodological approach is quite conducive and probably secure, the probability of getting a meaningful traceback message is very minimal if a major DDOS attack is cascaded onto the victim mainly if proper attention been meticulously invested in minimizing the chances of detecting the traceback messages.

However, this can be resolved by associating a particular score to every traceback message created. If there is any allocation, the value will be affected therefore, to deploy and implement the traceback and its attack path reconstruction based on the ICMP traceback will involve a change in the organization's routing tables of the routers sitting onto the network. We need to keep in mind that prior to start tracing back the attack, all the software of the routers would need to undergo upgrades from their respective vendors and then only the ICMP traceback must be activated at the Internet Service Provider (ISP) interaction. Furthermore, with ICMP traceback, routers can be set up individually thus allowing good scalability. The number of packets required for reconstructing the attack pathway is planned and based on thousands since the chances of producing an ICMP traceback message is  $1/2000$  and for partial stratagem to be effective, the victim must be conscious about the network topology and the routing of the network. We also have to keep in mind that the reconstruction of data frames would consist of thousand entries thus leading to require enormous memory to process those entries. If in case that a router that is responsible for the marking of the packets happen to be corrupted, it can hence be reprogrammed to produce incorrect traceback messages thus giving out false reconstruction attacks pathways. Based upon Wu, F.S. et al. (2001) described in his report that handling major DDOS attacks with ICMP Traceback was possible but would not be true if a large number of reflectors were to be used. Therefore, the capability very similar to Probabilistic Packet Marking thus leading to the conclusion that transformation involving reflector prove to be more difficult thus confining the limit of the traceback to the reflector.

### Packet Logging IP Traceback

According to Snoeren, A.C et al, (2002), this scheme is more commonly known as Source Path Isolation Engine (SPIE). With packet logging also referred as hash-based traceback, every single router on the network keep



some data or information of every single that have passed through that particular router so that later, it can hence determine if that packet have already been through. The mechanism through which that hash-based IP traceback operate is that routers on the network are called data generation agent (DGAs) and since that network is symmetrically divided into zones. In each and every zone, SPIE collection and reduction agent (SCARs) are linked to all DGAs thus allow a communication link to be established as depicted in figure 3. Also the SPIE traceback Manager (STM) is a central unit that has a link to IDS of the victims.

So basically when packets flows across the network, partial information of packets get captured and stored in the DGAs. This partial information consist of the IP header and the initial 8 bytes of the payload of each packet is hashed and then recorded in bloom filter which reaching 70% of its capacity is archived for later analysis and a new one is used. However, these bloom filters are used during the capturing stage and to the time taken to use of these bloom filter is known as a time period. Having said so, DGAs can capture any transformations that occurs an influence on the field. Normally the type of transformation and the information required to reconstruct the attack pathway are recorded in a Transform Lookup Table (TLT) which each bloom filter has it's won TLT for it time period.

Therefore when the STM is notified of an attack from the victim's IDS, appropriate request are transmitted to SCAR which in turn look up for recorded partial information and transformation tables from DGAs from the allocated time period. After analyzing and comparing the tables, SCAR will be able to trace which router in the zone forwarded that packet. It is then the responsibility of the scar to retrace the router through which the packet has been going through and finally send a report to the STM. With this schematic hash-based IP Traceback approach, it can easily handle massive DDOS attacks. It is quite normal that a bigger amount of memory is required by the SCAR and the STM which is dedicated to the traceback processes. Given that this scheme is extremely difficult to bypass, is can therefore handle more or less any packet transformation.

### **Specialized Routing IP Traceback**

With this approach, the introduction of Tracking Router (TR) onto the network checks all the traffic that flows through the network and this is achieved by routing all the packets through the TR. This is then realized by creating a Generic Route Encapsulation (GRE) from every interface of the router to the TR. This architecture is illustrated in Figure 3 with the TR in the centre and with all of the edge routers on the network connected to with GRE tunnels using a star-like logical network is known as an overlay network.

However a single TR will not be capable of handling this load of packet from the entire network thus having several TRs which can still be logically implemented as a single TR that will be using signature-based intrusion detection. With this approach, if an attack is sensed, this means that the source of the attack can be traced back because it is only one hop away according to Belenky, A. & Ansari, N. (2003).

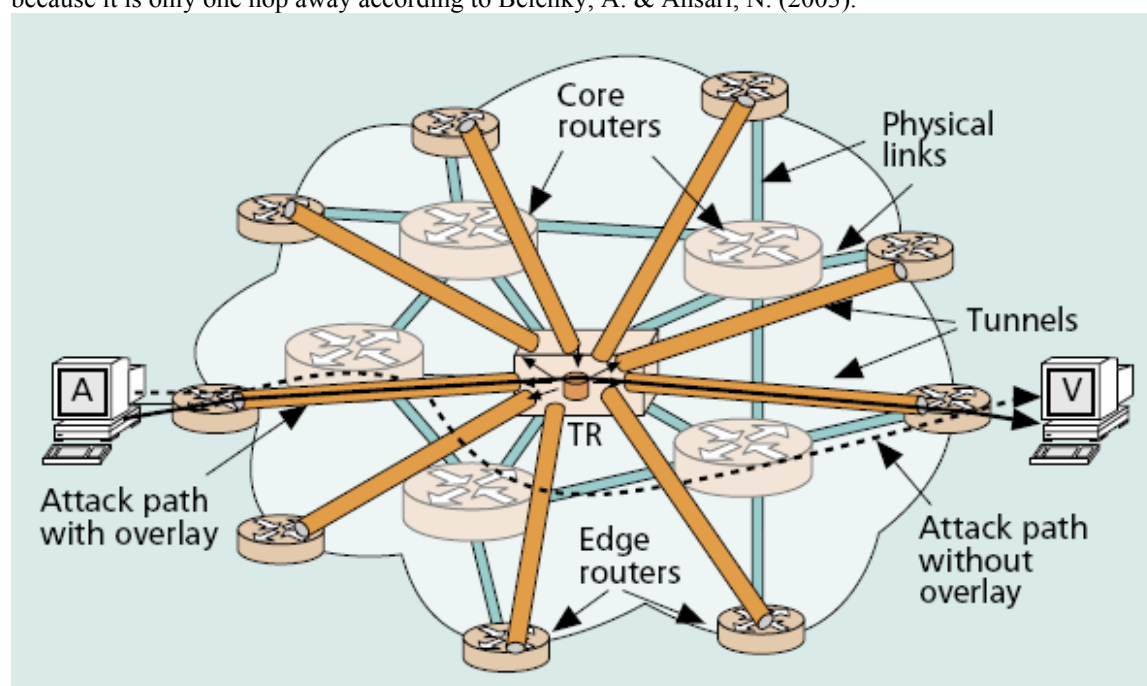


Figure 3. depicts the overlay network (Belenky, A. & Ansari, N. 2003)

This method make use of the most common features that are available on routers of today consequently the involvement of ISP is massive as it will have carry out a traceback and identify the source of the attack on its own.

Nevertheless this approach does have an extreme condition; it will only function within a single domain, therefore for the overlay network to be effective over ISPs, it would be wise to connect all the TRs into a single system. Of course with the method, a single packet is necessary to traceback an attack provided that the attack has been identified and reported. This then happen as soon as the IDS sitting on the TR identified the attack, it would trace it to its endpoint of the GRE tunnel. Now, if the edge router has numerous interfaces then it would be impossible to know exactly from which interfaces the attack was launched. However with this approach has a give-and-take collaboration between overhead and protection. But if the tunnels are mounted with IPSec, consequently the overhead bandwidth will increase and so will the level of protection. Moreover, this approach would be suitable to manage extensive DDOS attacks where tracing back the source of any particular packet ever to the edge of the network. Also handling packet transformation will not be a problem with this method.

### IPSec Traceback with IPSec

This method is normally introduced as forming part of an intrusion detection structure specially designed for network based mask known as **DECIDUOUS** Chang H.Y et al. (1999). Given that this particular framework is far beyond the scope of this report, the method of operation in detecting the source address of any attack is of great significance. Therefore this approach relies on the fact that the complete network structure is of understanding and control to the system. This denotes that if at any pointing time there is an IPSec security involvement between an inconsistent router and the victim. However, if the identified packets are picked up by the security associations, therefore the initial attack is further than this router but if the opposite happens, if the identified packets are not detected, it will denote that the origin of the attack lied between this router and its victim. Thus allowing us to possibly identify a single or a group of routers from where the attack was launched.

Following the explanation, in figure 4 below when an attack is sensed, an IPSec security association between Router 4 (R4) and victim denoted by the letter 'V'. In fact if 'A' being an attacker, his or her attack packets will have to flow through the network and the tunnel thus requesting them to be authenticating before hopping through the tunnel. Then the tunnel from Router 1 (R1) to the victim is created. It is also noted that from Router 4 (R4) to the victim point 'V' there will be 2 tunnels that will encapsulate data traffic from the attacker. (This is not actually represented onto the figure 4)

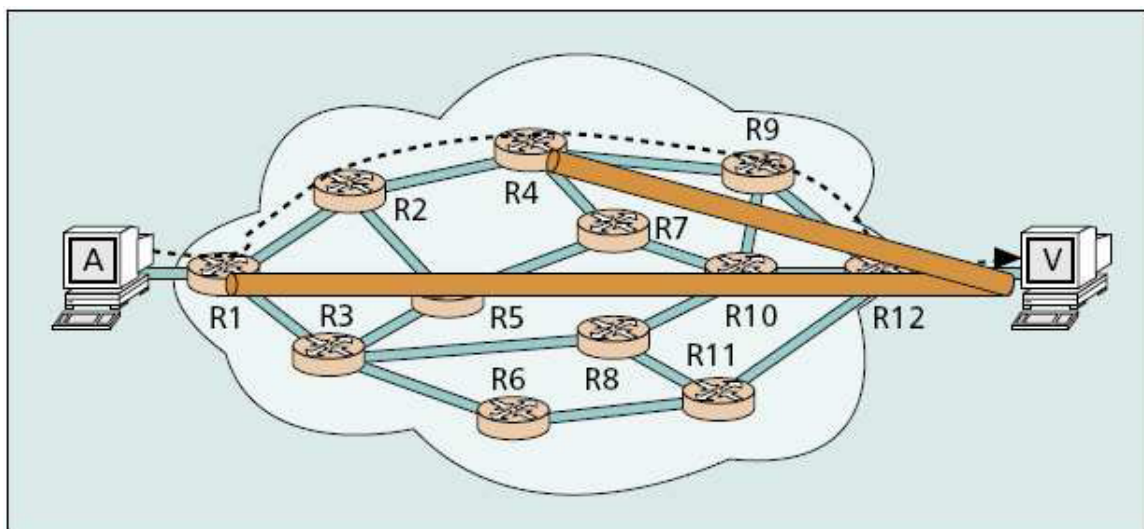


Figure 4 illustrating the IPSec Traceback approach (Belenky, A. & Ansari, N. 2003)

The second tunnel will perform its encapsulation over the first tunnel. Moreover, using two security associations to authenticate the traffic flow, it is obvious that if the attack was initiated behind RA but if the attack were to be authenticated by the only first tunnel then it would clearly identify the attacker would lies between Router 1 (R1) and Router 4 (R4) it would be on Router 2 (R2).

The system, however, will undergo numerous possible iterations in order to consider all the viable pathways before determining with which routers the victim should lean on for the IPSec security associations given that the source address is unknown. With this approach, the only interaction would come from the ISP which will have to communicate its network topology to its entire client in order to create the IPSec tunnels to the routers. Yet we to understand that in case 'shared key' authentication is used, every systems need to be aware of any changes on any routers on the network. We also assume that the authentication process will include digital certificate to be used at security associations. It is also to mention that with this approach the number of packets to perform traceback is low compared to the previously discussed methods.

This approach is very secure and even packet transformation is not an issue, being even capable of tracing back major DDOS attacks by tracing the path individually but we have to understand that when DDOS attacks are being performed, DDOS can themselves be targeted since ISPs have to remain open for clients to create IPSec tunnels thus making it unsuitable to manage complex DDOS attacks.

## **SPOOFED PACKET ATTACKS**

Packet spoofing can actually be part of different and various attacks type. So having the knowledge of how they operate would definitely reveal itself to be fruitful as then we know how they behave. One of the major aspect in whichever packet spoofing attack types, it does not have to receive packet replies from the targeted source. Therefore this part of this report will elaborate on the different types of attacks and discept their security associations.

### **SMURF Attack**

According to Computer Emergency Response Team (1998), SMURF attack can be defined as an invasion on the network that floods that network with a huge amount of requests that unfortunately disrupt the normal traffic. This is hence carried out by sending spoofed ICMP echo requests (ping) packet to a subnet broadcast. Therefore when a broadcast address is picked up by all the nodes on the subnet, this will then drive each active host to send an echo reply to the source where here in this attack. The source address is directed to the address of the target. This then amplifies causing a huge amount of packet to be generated and directed towards the target. This would definitely congest the network thus causing a major degeneration of the service on its network. Again Smurf attack is much more concerned with the multiplication of the packets and address spoofing to overflow the targeted network. Moreover, packet return is of no importance to the attacker as it is not desired. For successfully accomplish this type of attack, the attacker should be able to grab the broadcast address and then create to ICMP echo requests which unfortunately are broadly available.

### **SYN-Flood Attack**

SYN-Flood attacks the most classic denial-of-service (DOS) attack where as mentioned earlier return packets are irrelevant. With SYN-Flood attacks, the attacker has to continually send a huge number of TCP SYN packets to the target which the host in return will acknowledge by sending an acknowledgement packet to the pretended sender. The sending host will wait for an acknowledgement reply which will never be sent out by the attacker forcing the host to engage in an undetermined wait thus causing the buffer on the target host to be tied up. Eventually when the entire buffer is used, no further network connection would be possible. Obviously the connection will finish by timing-out where the kernel will release the buffer and allow for new connections. Because of the huge amount of packet sent earlier in the SYN-Flood by the attacker is more likely to use the buffer again rather than a normal packet from a genuine connection. This SYN-Flood attack is not interested with the return packets but for the attack to be successful, the attacker will have to spoof source addresses from host that are non-existent or inactive.

### **Bounce Scanning**

Using scanning in general resides a difficulty that the attacker must be able to view the replies and hence make it quite complex to use spoofed addresses. One of the easiest ways of achieving this type of attack is to spoof the address of a different computer on your network subdivision and hence listen to the network traffic for echo to the spoofed address. Following a report published by security focus (2001), a brilliant option would be to make use of spoofed packets and then to unwillingly listen to the targeted replies.

This particular type of attack normally exploit the IP header known as the "identification number" field. This number is usually incremented by one each time that a packet is sent over. Therefore the bounce attack makes use of that to send spoofed SYN packets to the targeted host through a part. However, Wu, S. et al (1998) mentioned that of the port is closed, a reply is generated back with a reset. Where on action is undertaken when

it is received by the spoofed host. After all in the case that the port is opened, the target still replies back to the spoofed source through an acknowledgement. However given that the spoofed host is not responsible for launching the SYN-flood attack, it therefore transmit a reset to the target whilst still incrementing the IP id number.

For these particular attacks to be successful there are 3 main factors that need to be considered:

- 1) Scrutinize the spoofed host requesting to locate its actual id number.
- 2) Direct the spoofed scan packet to the targeted source.
- 3) Reconfirm the id number of the spoofed host

Once these 3 main areas have been achieved by the attacker, s/he can easily determine whether the targeted host's port was either open or closed denoted by the id member incrementing by one would return port was closed and if it were to be incremented by 2 then this would mean that the port was opened.

However, the attacker has to make sure that other packets are not directed to the spoofed host during the scanning, this is therefore achieved by either selecting a host to spoof with some or no network activity at all and such example would be a printer onto the network. On the other hand, if the spoofed host does not show any increase in the id numbers by at least one, the attacker can therefore make use of a multitude of queries to each port and therefore deduce its mode by monitoring the changes occurring with the id numbers.

### **TCP Spoofing**

What makes TCP connection spoofing attack quite unique is that it is a combination and coordination of more than one attack. This would mean creating a DOS attack on one hand and on the other hand spoofing packets of the attack target. SO, basically to conduct a DOS attack on a trusted host could be anything that would prevent the trusted host to send out reset packet to the host in the instance of a SYN-Flood attack. The other form of attack would require the device to be transmitting spoofed packet to the target while impersonating the source host. Moreover, due to the DOS being launched prior to sending the spoofed packet to the target, the trusted host unfortunately cannot reply to the packets being received from the target. In addition the attacker can forcefully lead the target to believe that the sent packets are from a trusted source thus later allowing the attacker to use the target to act as a trusted host at a pointing time.

However, this type of attack is quite complex and hence does require some knowledge as TCP does require reply packets to carry the sequence number of the former packet if the attacker cannot examine the packet, therefore she/he would have to guess the sequence number which could be made very hard to guess but is still feasible to achieve.

### **Zombie Control attack**

McNevin, T. (2005), a DDOS has two primary goals, firstly to flood the victim's server and secondly to consume the most of the bandwidth of the victim. Nonetheless DDOS attacks normally consist of attacker(s), several intermediary computers, and finally many "zombies". Zombies are when an attacker has been able to infiltrate other computers through their weaknesses. Once the attacker has gained control of the machine, she/he can install tools, or programs that will allow that attacker the ability to communicate back and with other zombies. McNevin, T. (2005) also pointed out that it is very probable that an attacker might go through the process of recruiting numerous zombies over an indefinite period of time and when the attacker has built up a huge network of zombies begin flooding packets towards its victim(s). In other words when the attacker decides to launch his or her attack, the initiator(s) will hence convey the message to the intermediary computers which will trigger the zombies to start flooding insignificant data in the diversion of the victim. However, this data traffic flow is not always ludicrous as the attacker may mask their traffic to appear like genuine and appropriate traffic to overcome any filtering defenses in detecting packet attacks.

Above are some of the ways through which spoofed packets could be used for different DOS or DDOS attacks. It is somehow useful to know if the packets have been spoofed or not as it will definitely help in mitigating attacks, or even help to traceback the true attack source.

## **SPOOFED PACKETS DETECTION METHODS**

Spoofed packet detection methods can be categorized as of those depending upon router sustenance, passive host based, active host-based methods and finally upon administrative methods which is one of the most frequently used methodology. This implies that when an attack occurs, the responsible personnel at the attacked location

will liaise contact with the authorized personnel at the supposedly attack site and ask for an acknowledgement which is totally delicate. Therefore, the need of having computerized ways and means of detecting whether IP packets have been spoofed. This section of this report will have a closer look into the different methods and approaches in detecting spoofed packets.

## **ROUTING APPROACHES AND METHODS**

Routers are devices that can identify IP addresses and through which network interface to they originally come from and can also point out packets that should not have been received by an interface. This therefore means that a router will definitely be able to identify if addresses are either for the internal or external network. Nonetheless, if the router is addressed with IP packets with external IP address on an internal interface and vice versa, therefore it may come to the conclusion that the packet source has probably been faked.

Recently with DOS attacks including spoofed attack packets, methods to counter measure these treats have been developed and are put in place filtering methods are being implemented most commonly known as Ingress Filtering, which filter inbound packets thus protecting from outside attacks. Similarly for filtering outbound traffic known as Egress Filtering which prevent internal machine to be compromised from spoofed attacks. Yet, if all the routers would have that sort of filtering in place, both Ingress and/or Egress then it would push an attacker to corrupt that router. However, internal routers with an adequate understanding of the inside and outside can spot fake packets but with the implementation of certain network topologies, unnecessary pathways make its confusing, thus making use of host based methods to detect the spoofed packet at the router's level.

Templeton J.S & Levitt K.E. (2000) mentioned that IANA does control a certain number of IP addresses for special purposes. Table 1 does illustrate those special IP addresses and what they used for. Most firewalls will then compares that table with the packets they're handling. Nevertheless this method dose poses a constraint and can only be used only when IP packets are being thrown through. Yet an attacker cold still fake packets if on the same Ethernet subnet as both the IP and the MAC address would be spoofed. Therefore the need for other approaches is needed.

## **NON – ROUTING APPROACHES AND METHODS**

### **ACTIVE DETECTION METHODS**

With the active detection methods inquiries are performed to identify the originating source of the packet (reactive) or influence protocol specific guideline for the sender to take action upon (proactive). These different approaches do have an enormous advantage over the routing approaches and methods discussed earlier as there is no involvement of ISPs and prove to be impressive and operative even through the attacker may be sitting on the same subnet as the targeted host.

The active methods normally will depend on an acknowledgement from the claimed source but only if the spoofed host is operational that in can be influenced meaning that only when linked to the network gathering and handling packets. It is to mention that if a host is fully firewalled thus not replying to quests is then deduced to be inactive. However, these hosts are frequently manipulated as source addresses in spoofed packets. Therefore when hosts do not reply to any requests, passive methods is then used for acknowledgement and validation.

### **Time-To-Live (TTL) Approach**

Since IP packets are normally routed across the network, their time-to-live (TTL) is subject to decrease. Therefore, the IP packet header can be controlled to make sure that IP packets are not being routed interminably when their host cannot be located in a fix number of hops according to Stevens W.R (1994). This technique is also used by some networked devices to halt IP packets from being transmitted outside a host's subnet. Moreover, Templeton J.S & Levitt K.E. (2000) mentioned that TTL is a useful approach and method in detecting spoofed packet but are based upon the following assumptions.

1. The number of hops will be the same then a packet is sent between two hosts provided the same routing path is taken
2. Packet transmitted near in time to each other will use the same trace route to its destination
3. Routes rarely changes
4. Whenever routes are altered, there is no compelling modification in the number of hops.

Having described the above assumptions, these mentioned approaches might outcome in false alerts if these assumptions are not respected. Hence repeatedly checking the packets should not breach those assumptions.

### **Direct TTL Requests:**

With direct TTL, the mechanism of operation is quite simply by sending out a packet to a host that will have induce an acknowledgement, we can verify whether the TTL from the acknowledgement is identical to the one that was transmitted as it they are from the same protocol, they usually will have the same TTL thus considering the fact that different protocols would definitely use different basic TTL. However when different protocols are used then it is imperative that we deduce the true hop count. In the instance of TCP/UDP, 64 and 128 are the most commonly used whereas with ICMP, the initial value used are 128 and 255 respectively the number of hops can be calculated by subtracting the monitored TTL from the assumed value, we can then deduce the number of hops.

After all if an attacker does happen to have the same number of hops, this approach will return a false negative, but if the attacker was to be aware fo the exact number of hops between the faked host and target, it fake the TTL field

### **OS Fingerprinting**

OS fingerprinting can be classified into 2 different categories, active fingerprinting which is associated with the direct probing of computer, and passive fingerprinting which refers to initially monitor the traffic and hence forth analyzing it to the various standardized benchmark for various operating system. On the other hand, we can create a controlled passive fingerprint as we monitor the traffic from a host and afterwards examine this against an active OS fingerprinting. Thus ascertaining if both OS are likely to be the same but if that is not the case, we then deduce that the packets are faked.

### **TCP Flow Control**

In general a TCP header does have a send window size (SWS) which is the upper bound on the number outstanding frames that the sender can transmit (not the ACKed) or the maximum number or amount of data that the sender can receive. So, if even those SWS are to be set to zero, then the sender should stop transmitting data. But it is very important that the sender respond to the flow control by acknowledging the ACK-packets. Other wise the sender should terminate once the very first window size is over flowed else we could infer that the packets are spoofed. To make sure that this is not to occur, the sender can transmit a primary window size that is quite small thus if this margin is exceeded, the conclusion would be that the packets have been faked.

However, TCP packet spoofing is quite hard to achieve as the correct sequence number to a lot of TCP packets is required and most of the TCP connection don't get pass the initial acknowledgement packet (ACK-PK) Therefore the best way for this to be effective is in the handshake process. The TCP handshake normally requests that the host transmitting initiate that first SYN wait for the SYN-ACK before transmitting the first ACK packet. Modifying the wonder size of the ACK-SYN to zero, would indicate whether the sender is accepting and reacting to the packets. Moreover, if the sender just return an ACK-PK certain that true originator is not addressing to our packets and hence a spoofed packet

### **TCP Approach and Methods**

When it comes to detect spoofed TCP packets, a number of approaches and methods on top of the IP packet methods described later. The role of the TCP is to maintain reliable packet transmission which implies that both sender and receiver should be communicating. Thus allowing us to uncover the faked packet by masquerading the fact that the sender spoofed data packet will not be responsive to any packet from the receiver. Two different approaches combining ACK-PK will be used

1. Request the sender to delay sending packets
2. Request the sender to recovery a packet

### **IP Identification Number**

As mentioned earlier in the Bounce Scanning, the sending host normally increase the identification number (IP) in the IP header through each packet send. Given that IP identification number can easily be altered, thus making all the alterations calculable. Compared with TTL, IP ID can be used to detect spoofed IP packets despite the attacker sitting onto the same subnet as the target.

In a very simple way when we sent inquiring packets to the so said “claimed” source we expect to receive a reply, therefore the IDs should be very close to the previously received one from the host. Yet the packets should be slightly higher than the IDs’ in the controversial case. Therefore, we can infer that the packets sent out where not from the claimed source. It is also totally true that, if the host is bounded with the so called “source” is considerably active; the ID’s will change speedily.

With this approach, it is not unfamiliar to come across certain system that unfortunately changes the initiating IDs using a more complex methodology rather than incrementing by a constant number. To be in accordance with the RFC 791, Postel, J., (1981) mentioned that for fragmented packet assembly to be possible, only the ID numbers have to be in a successive order. Therefore this will favor more complex ID numbers. These can hence be overcome in 2 different ways. Firstly we could use a separate counter for every packet and secondly use a pseudo-random value which will have for aim to limit the actual IP data stream from interfering with each other. In those causes where more complex ID number is being used, using this particular approach might become ambiguous.

### **Retransmission of packets**

In TCP, the use of sequence number is vital as it helps in determining which packets have already been acknowledged. Therefore, an ACK-PK normally informs the receiver of all the packets that has been send out together with the sequence number of the successfully received packet. When the packet is received the ACK-PK number is compared to the minimum and the maximum values and if less or greater than the required value, the packet is dropped thus allowing the connection to be resynchronized by sending out a packet with the minimum ACK-PK number. Moreover these replies can still be exploited to examine faked packets by probing a packet which has been spoofed from the internal host having an ACK-PK number higher than the required minimum value. We hence force a resynchronization ACK from the host being forged where if an RST reply is received, we can therefore deduced that the connection has been tampered with.

However, a major concern that arouse with this approach according to Joncheray, L (2001), it will conduce to an ACK-Storm since both ends-ie sender and receiver will struggle for resynchronization. Yet, their approach is better carried out on a firewall where the fake reply could be seized thus prevent an ACK-Storm as the interval host will not see the reply.

### **PASSIVE APPROACH**

With the passive approach, we can hence say that monitored data will have a predicted value as we can learn what values are to be expected and hence separate the packets that don’t fit the expected norms. Since TTL values are dependable upon the hosting OS, the network topology, the packets protocols are fairly, static; therefore TTL can be opted as a core support for passive detection. Unfortunately this cannot be applied to IP identification Numbers which have a predominantly distinct association with packet and consequently eliminate the choice of using a passive approach.

#### **Passive TTL approach**

As previously discussed with TTL values are a very good way of identifying the different hops that lie between both the sender and the receiver’s and i.e. the source and the destination. Therefore, on performing a monitoring observation over a lap of time, we can learn the TTL values of specific IP addresses source and who deduce what would be their expected values at the host’s side. Most of times, if not nine out of ten times, we will come across packets that the host has never come across. Therefore, by comparing the IP addresses which are generally the same for the number of hops away.

Nonetheless, to be able to construct a better model for the detection of spoofed IP packets, both passive and reactive approaches should be used in conjunction where the reactive methods would be used for when certain packets seems dubious.

Since passive TTL approaches are very firm and reliable especially when it come network routing attacks which normally happened when packets destined for a host are routed to a different host impersonating the first host. Unfortunately, this is not entirely classified as packet spoofing due to the fact that these packets are still emanating from a valid IP address of the sender. In addition using passive TTL approach will definitely act as a routing detector

### **RELATED WORKINGS:**

So far, there have not been a lot of work relations about detecting spoofed IP packets. Most of the works that have been published were addressing different spoofing attacks. The most common one ARP spoofing and according to Whalen, S (2001) associated with sending packet through Ethernet MAC of a different host than the IP address thus confusing local hosts of the local network to channel the packets to the wrong interface onto the network. In accordance to Aura, T. and Nikander, D. (1997), some firewalls make use of SYN-Cookies as an approach to reduce the effect of SYN-Flood type Dos. SO basically a SYN-cookie is a crypto-graphical ICP sequence number which is related to time, port number, and source IP address. The process is quite simple, if a SYN packet is received, instead of going around opening a buffer for the connection, the server will send a SYN-ACK-PK with the SYN\_Cookie thus creating a stateless handshake. However, when an ACK-DK is acquired from an inactive socket, the returned sequence value is verified and compared if it is a valid SYN packet which was sent out from the host. Provided that the sequence number is valid, then only a buffer is allowed and hence begins the connection else the packet is dropped. This method is somehow used to mitigate SYN-flood attacks but not detect spoofed packets.

## **CONCLUSION:**

Tracing IP packets have got a few limitations, such as tracing beyond corporate firewalls. To be able to effectively traceback the IP addresses, we need to be able to reach the host from where the attack where initiated and hence the difficulty of tracing back these IP packets through corporate firewalls and when trying to get the last IP address would show the firewall address.

However, detecting spoofed IP packets goes well beyond simple detection. Since faked packets are among the most common attacks, therefore identifying them at an earlier stage and hence preventing them in occurring will be of a major help to improve the network security. Though this paper we have shown a large range of techniques available in detecting spoofed packets. These various techniques and approaches can either be used on their own or could be combined to enhance detection effectiveness due to their ease of implementation. Yet we do understand that all the discussed methods above are not entirely complete as an attacker can still transmit spoofed packets which remain undetected. We also need to keep in mind that there is no such system which is fully –ie 100% reliable. There approaches are not the entire solution but they greatly contribute to increase the detection of spoofed IP packets.

## **REFERENCES**

- Belenky, A & Ansari, N. (2003). *On IP traceback*. Retrieved September 7, 2007, from <http://ieeexplore.ieee.org/iel5/35/27341/01215651.pdf>
- Bellovin, M.S. (2000). *ICMP Traceback Messages*. Retrieved September 3, 2007, from <http://www.research.att.com/smb/papers/draftbellovin-itrace-00.txt>.
- CERT (1998). *Smurf IP denial-of-service attacks*. Retrieved September 10, 2007, from <http://www.cert.org/advisories/CA-1998-01.html>
- CERT (1999). *Spoofed/Forged Email*. Retrieved September 7, 2007, from [http://www.cert.org/tech\\_tips/email\\_spoofing.html](http://www.cert.org/tech_tips/email_spoofing.html)
- Chang, H., Narayan, R., Wu, S., et al.(1999). *DECIDUOUS: decentralized source identification for network-based intrusions*. Proc. of the Sixth IFIP/IEEE International Symposium on Integrated Network Management.
- Chang, R.K. (2002). *Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial*. IEEE Communication Magazine.
- Chang, H., Wu, S., and Jou, Y. (2001). *Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks*. ACM Transaction on Information and System Security (TISSEC).
- Droms, R. (1997). *RFC 2131: Dynamic Host Configuration Protocol*. Retrieved August 28, 2007, from <http://www.ietf.org/rfc/rfc2131>
- Joncheray, L. (1995). *A Simple Active Attack against TCP*. Retrieved August 15, 2007, from



[www.insecure.org/stf/iphijack.txt](http://www.insecure.org/stf/iphijack.txt)

Lau, F., Rubin, S.H., Smith, M.H., and Trajkovic, L. (2000). *Distributed denial of service attacks*. Proc.

2000 IEEE Int. Conf. on Systems, Man, and Cybernetics, Nashville, TN, pp. 2275-2280, October 2000.

Postel, J. (1981). *RFC 791: DARPA Internet Program Protocol Specification*. Retrieved August 23, 2007, from <http://www.ietf.org/rfc/rfc791>

Postel, J. (1981). RFC793: Transmission Control Protocol. Retrieved August 27, 2007 from <http://www.ietf.org/rfc/rfc793.txt>

Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2000). Practical Network Support for IP Traceback. Retrieved July 26, 2007, from <http://www.cs.washington.edu/homes/savage/traceback.html>

Stevens, R. (1994). *TCP/IP Illustrated. Volume I – The Protocols*. Addison-Wesley. 1st edition.

Templeton, S., and Levitt, K. (2000). A Requires/Provides Model for Computer Attacks. Retrieved September 17, 2007, from <http://seclab.cs.ucdavis.edu/papers/DetectingSpoofed-DISCEX.pdf>

Staniford, S., Hoagland, J., and McAlerney, J. (n.d.). Practical Automated Detection of Stealthy Portscans. Retrieved August 26, 2007, from <http://www.silicondefense.com/pptntext/Spice-JCS>

Snoeren, C.A. et al. (2002). Single-Packet IP Traceback. Retrieved August 31, 2007, from <http://delivery.acm.org/10.1145/620000/611410/01134298.pdf?key1=611410&key2=2006031911&coll=GUIDE&dl=GUIDE&CFID=37231185&CFTOKEN=12343255>

Song, D.X., and Perrig, A. (2001). Advanced and Authenticated Marking Schemes for IP Traceback. Proceedings of INFOCOM.

Stone, R. (2000). An IP Overlay Network for Tracking DoS Floods,” Proceedings of the 9th USENIX Sec. Symposium.

Whalen, S. (2001). An Introduction to ARP Spoofing. Retrieved August 11, 2007, from [http://packetstorm.securify.com/papers/protocols/intro\\_to\\_arp\\_spoofing.pdf](http://packetstorm.securify.com/papers/protocols/intro_to_arp_spoofing.pdf)

Wu, S., Chang, H., et al. (1999). Design and Implementation of a Scalable Intrusion Detection System for the OSPF Routing Protocol. *Journal of Computer Networks and ISDN Systems*.

Zalewski, M. (2001). Strange Attractors and TCP/IP Sequence Number Analysis. Retrieved September 21, 2007, from <http://razor.bindview.com/publish/papers/tcpseq.html>

## **COPYRIGHT**

Krishnun Sansurooah ©2007. The author/s assigns SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.

## The effectiveness of investigative tools for Secure Digital (SD) Memory Card forensics

Haitham Al-Hajri  
Edith Cowan University  
halhajri@student.ecu.edu.au

Patricia A H Williams  
Edith Cowan University  
trish.williams@ecu.edu.au

### Abstract

*There are many freeware based tools that can be downloaded from the World Wide Web. This paper reports the retrieval results of using these tools on digital images which have been deleted from Secure Digital (SD) cards. Since SD cards and USB flash drives are considered solid state technology, the tools selected are specifically for solid state drives. This research helps classify the selection of the most effective freeware tools that could be used to recover lost or deleted images. Further, it includes some of the issues that would face forensic examiners undertaking such investigations. The tools were tested using the Windows environment and did not require any specific forensics background in order to retrieve the images. The testing procedures included retrieval time and the state of the deleted image, viewable or damaged (corrupt). A review of tool functionality is given together with the most effective tools useful in retrieving images from deleted Secure Digital cards.*

### Keywords

Forensics, retrieval, secure digital (SD), solid state disks, removal media, freeware based tools.

## INTRODUCTION

Digital storage devices have developed rapidly over the past five years which includes storage size, physical size and shape. In computer security it is accepted that technology can be used for illegal purposes when its fall into criminal hands even if the technology was designed for a purely different purpose. One popular type of storage device is the Secure Digital (SD) Card, which is part of the 'solid state' family of technologies. Therefore a number of tools have been developed to recover data from solid state disks. Most of the currently available tools are commercially developed and therefore are expensive. The purpose of the paper is to test the ability of freeware tools to recover deleted digital images from SD cards. Some of the tools have the ability to retrieve other file formats and these are noted in the results. The tools selected for testing are all freeware based tools that are comparable in functionality to the commercial tools. This paper investigates the effectiveness of some of these freeware tools obtainable from the World Wide Web.

## TESTING SCENARIO

### Experiment procedure

The experiment was carried on an 256 MB Secure Digital card that had been used to store 101 photos along with other document such as word, PDF and other file formats. The procedure for the investigation was as follows:

- The testing tools were installed in a stable, clean windows environment;
- The SD card was hashed before and after using the tools for analyses of the card;

### SD Card Integrity

In order to ensure the integrity of the SD card, an initial analysis of the card was undertaken to ensure that the card did not contain any files. A format utility from Windows XP was utilized to format the new SD card. After verifying that the card was clean, a number of files such as MSWord and PDF were saved to it together with 101 digital images. Before beginning the analysis with each tool, an MD5 hash was generated from the card to compare it with the original hash to ensure that running the tools had not affected the state of the SD card. The hash value of each image was not taken because the aim of the experiment was to analyse the ability of the tools

to recover the digital photos in a viewable format, not test the images themselves. Maintaining the integrity of the SD card whilst examining the effect of the tools was a priority to provide an equitable testing environment for each tool.

### **Cleaning the SD card**

The SD card was formatted using the simple deletion function in Windows. No secure deletion tool was used on the SD card so that the capability of the tools to recover standard deleted images could be tested.

### **Testing Schema**

All tools were tested on the same Secure Digital (SD) card, on a Windows environment machine using a Windows Picture and Fax Viewer (a standard tool in Windows). The targeted data was deleted photos regardless of the different file formats in which they were saved. The goal was to recover photographs in a viewable state. The selected tools used are user friendly and able to be operated without the need of manual or background knowledge on digital forensics. The experiments tested the ability of the tools to recover photos and assessed whether they were viewable or corrupt, in addition it assessed how long it took to recover the images.

The following section presents a background of each tool, software details, functional overview and the test results.

### **Testing Environment**

1. SD card 256 MB
2. SD Card 128 MB for backup and testing purposes.
3. Two card reader via USB port
4. Stop watch (recording the time run)
5. Windows environment machine (computer).

## **TOOL BACKGROUND**

This section gives a background on each tool selected. It includes a brief description of the software, an example of the interface and the tool functionality.

### **MjM Free Photo Recovery Software**

This software claims that it has the ability of recovering images from memory cards that have been deleted or formatted. The tool automatically locates the memory card once its been plugged in the card reader and is then ready to scan. It displays what is contained on the card as thumbnail images. It can view photos in full size or recover them all. An example of the type of testing that MjM Data Recovery Ltd has previously conducted on the tool is:

During our review, we first deleted all images from the card via Windows - the program found and recovered all of them. We then formatted the card in the camera and restarted the search - and again it found them all (recovery results after formatting may vary depending on the formatting method used by the camera). Works with Compact Flash, Smart Media, Memory Sticks and other media storage cards. (MjM,2007)

#### **Software Details**

<b>Publisher</b>	MjM Data Recovery Ltd
<b>File Size</b>	3053 kb
<b>Version &amp; Last Updated</b>	0.12 beta - Dec 29, 2006
<b>License</b>	Freeware
<b>Windows</b>	98/ME/2000/XP
<b>Site</b>	<a href="http://www.snapfiles.com/get/mjmphotorecovery.htm">http://www.snapfiles.com/get/mjmphotorecovery.htm</a>

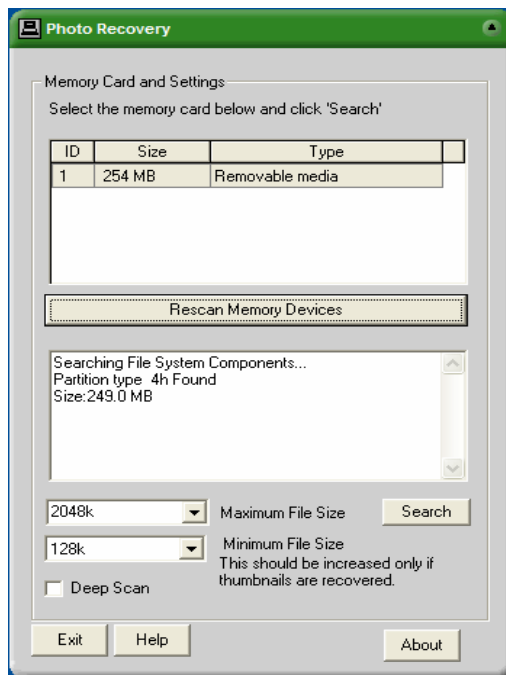


Figure 1. MjM (single card)

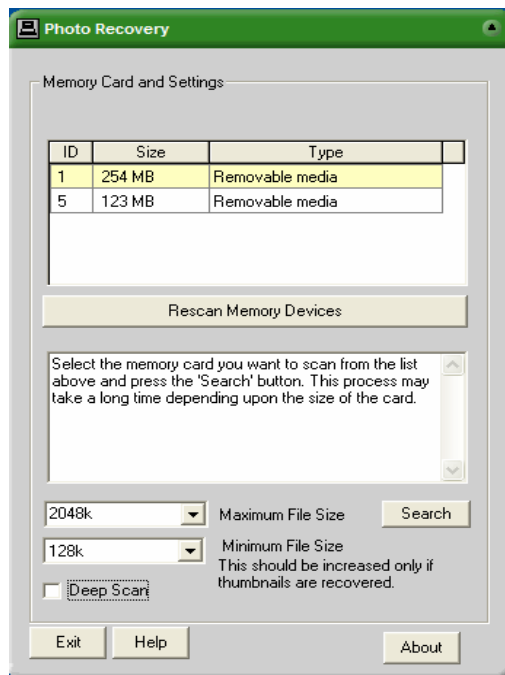


Figure 2. MjM (dual cards)

### Tool Functionality

The interface is simple as shown in Figure 1. It contains a number of sections in each terminal window and in the test displayed memory card details as shown in Figure 1. The top screen shot indicates that it recognized the presence of removal media SD card 256 MB, but it is just shown as 254 MB. Between the two windows there was memory rescanning function to scan for added memory cards if applicable. In this case the second card reader had been added, an SD card 128 MB to test the ability of the tool to handle added memory cards. The tool recognized the added memory card and displayed this as 123 MB removable media, as shown in Figure 2. The second window displayed a message to inform the user to select the memory device to be tested as shown in Figure 2.

The tool uses a drop down menu to select the size of the file with the maximum and minimum file size by default maximum (2048k) and minimum (128k). The tool supports a deep scan functionality where the user ticks the box will display a message context saying "deep can only needs to be used if you photos do not show using default settings ,using a normal scan or if the memory file system is corrupt" as shown in Figure 3.

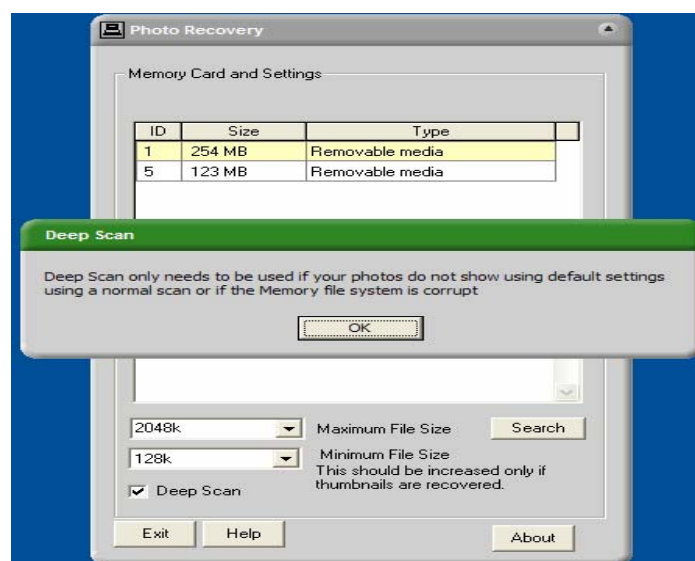


Figure 3. MjM (Deep scan option)

### PC Inspector Smart Recovery

PC Inspector Smart Recovery is an inspection based program to look at removable media such as secure digital and multimedia cards. This tool been advertised to work with any removable media from digital cameras and recover any files that have been deleted (GmbH, 2007).

#### Software Details

<b>Publisher</b>	CONVAR DEUTSCHLAND GmbH
<b>File Size</b>	Size 6233 kb
<b>Version &amp; Last Updated</b>	3.0 - Jun 27, 2004
<b>License</b>	Freeware
<b>Windows</b>	98/ME/2000/XP/vista
<b>Site:</b>	<a href="http://www.snapfiles.com/get/smartrecovery.htm">http://www.snapfiles.com/get/smartrecovery.htm</a>

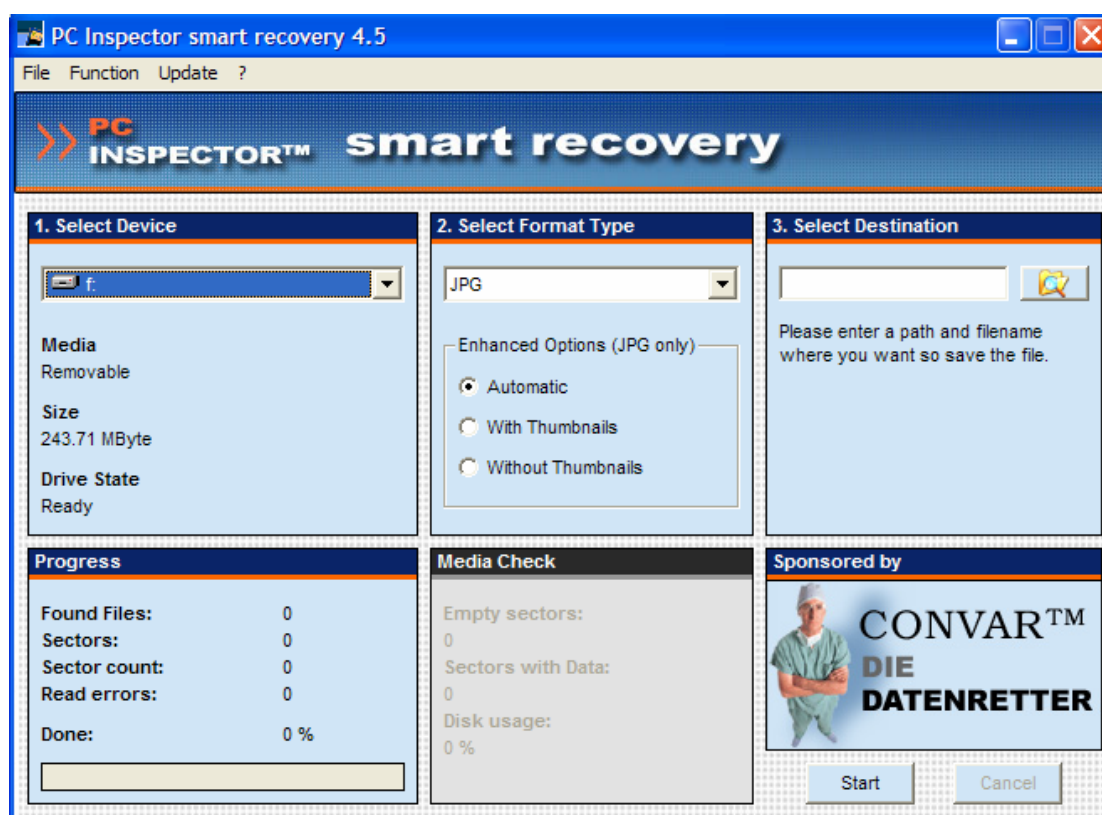


Figure 4. PC Inspector (main page)

#### Tool functionality

The interface of the tool is simple. It consists of two drop down menus and a browser bar to select the destination of the retrieved images. The first drop down menu is it to select the removable drive intended for inspection and retrieval. The device selection details the device such as the size of media drive and if it is fixed or removable. Whilst a second drop down menu offers selection of the format of the files intend to be retrieved as shown in Figure 4. Since this paper will look in to the effectiveness of the tool in retrieving images from an SD card, the tool was set to recover JPG photo format. In selecting the JPG format, an extra service becomes available, called enhanced options to display with or without thumbnail images. The tool supports different image format however for the purpose of this investigation the tool recovered JPG files only.

### Art Plus Digital Photo Recovery

Art Plus digital photo recovery tool claims that it can recover images from formatted or corrupted memory cards. The software has the ability to retrieve images from different memory card types. Moreover it has the ability to retrieve number of other media formats. The tool claims that it has the ability to "Read corrupted cards

(even if they're not recognized by Windows)" in addition the current Version 2.3 may include unspecified updates, enhancements, or bug fixes (Art Plus, 2007).

#### Software Details

<b>Publisher</b>	Art Plus Marketing & Publishing
<b>File Size</b>	9 912 kB
<b>Version &amp; Last Updated</b>	2.3 - unknown
<b>License</b>	Freeware
<b>Windows</b>	98/ME/2000/XP

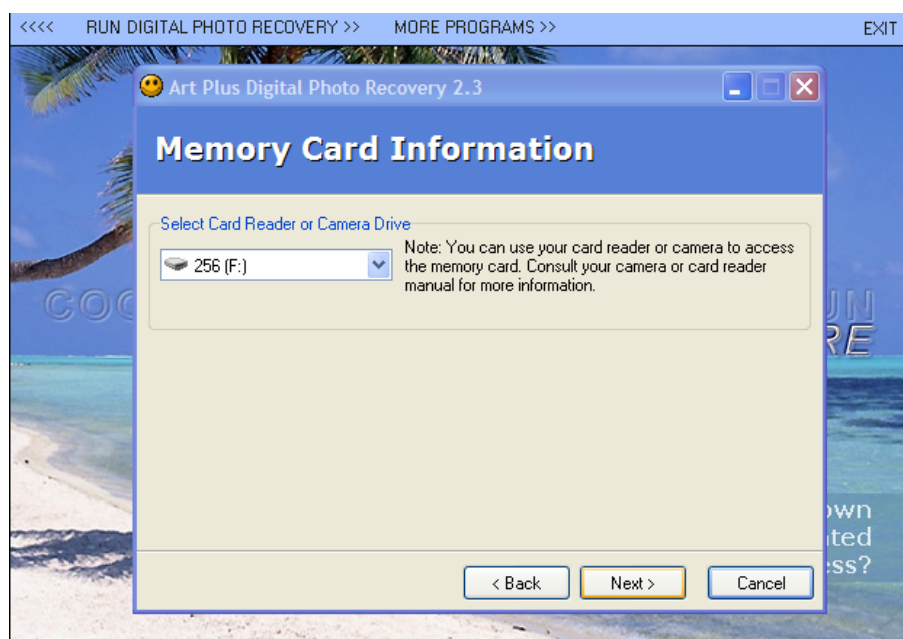


Figure 5. Art Plus (main page)

#### Tool Functionality

The software is self executable with no need for user specified installation. It contains a drop down menu to choose the removable drive to perform the retrieval on as shown in Figure 5. The subsequent window asks the user to select the target folder where the retrieved files will be saved. The last window displays the retrieved files and has the option to view the targeted folder or to process another removable media card as shown in Figure 6.

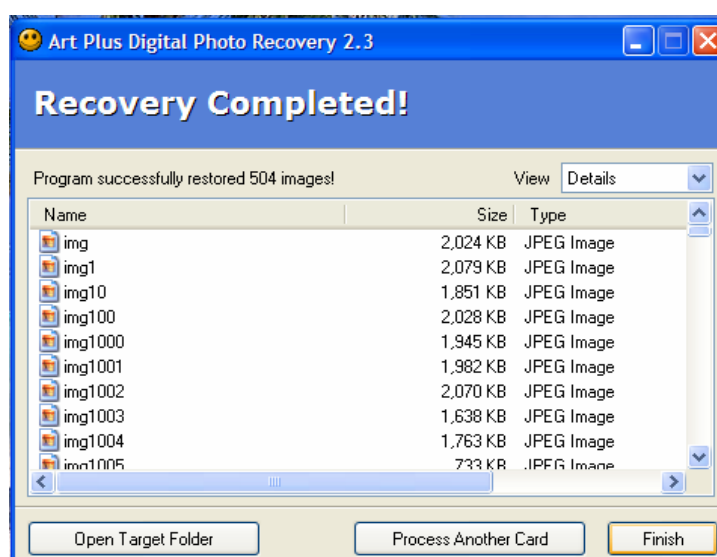


Figure 6. Art Plus (retrieved files)

### Free Undelete 2.0

Free undelete is a recovery program that has been developed to retrieve files that has been deleted on an NTFS FAT32 or FAT16 file system. (Recoveronix Ltd, 2007)

#### Software Details

<b>Publisher</b>	Recoveronix Ltd
<b>File Size</b>	705 kB
<b>Version &amp; Last updated</b>	2.0 – Unknown
<b>License</b>	Freeware
<b>Windows</b>	98/ME/2000/XP

#### Tool Functionality

The tool interface is simple with two main display windows. The left hand side window displays the removable media drives, where the user can select the memory card to scan. A second window displays the files that have been retrieved. The bottom of the tool window contains a search filter box where user can specify what type of files to view and the destination folder for the retrieved files as shown in Figure 7.



Figure 7. FreeUndelete (main page)

### Recuva – File Recovery

Recuva is recovery tool that works in the Windows environment. It recovers files that has been deleted from camera memory cards even if they have been emptied from the recycle bin. The software claims that it can retrieve files that have been affected and deleted by bugs or viruses. (Recuva, 2007)

#### Software Details

<b>Publisher</b>	recuva -file recovery
<b>File Size</b>	282 kB
<b>Version &amp; Last Updated</b>	v1.06.132- 1st October 2007
<b>License</b>	Freeware
<b>Windows</b>	98/ME/2000/XP

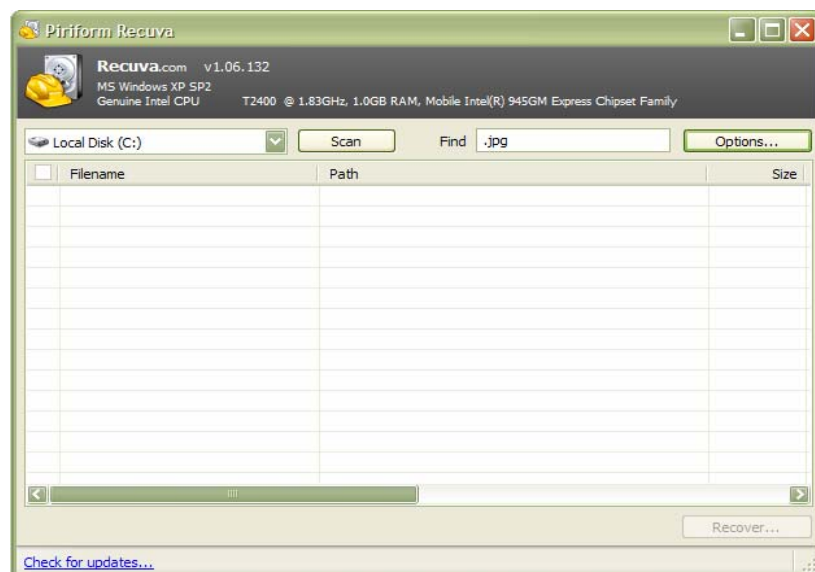


Figure 8. Recuva (main interface)

#### Tool functionality

The interface is easy to follow. It has a drop down menu to select the memory card to scan, as shown in Figure 8. The option button contains settings that help the user to customize the view. When the scan is complete the recover button is displayed in the bottom right hand corner to select the folder where the recovered files will be saved as shown in Figure 9. In the banner of the tool the details of the machine specification are displayed along with operating system type.

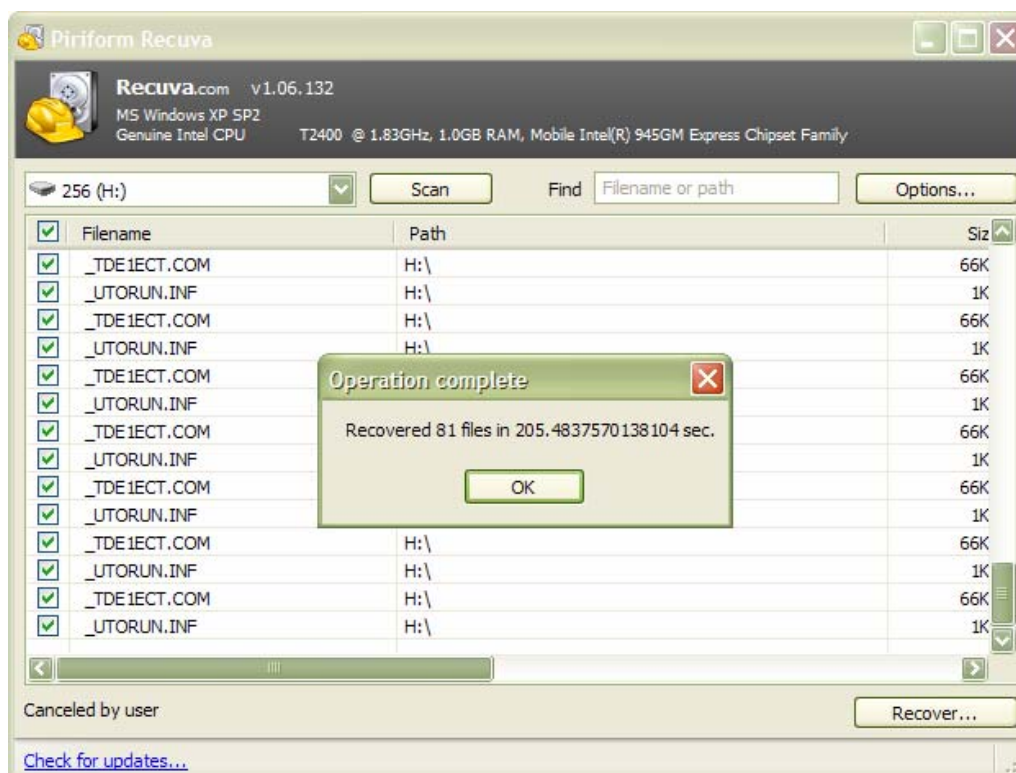


Figure 9. Recuva (scanned files)



## Soft Perfect File Recovery

The tool is designed to retrieve accidentally deleted files from different storage devices. It supports CF, SD, MMC and flash drives, in addition to other storage formats ranging from FAT12 to FAT32 along with NTFS. This software is self executable and needs no user installation (Softreaserch, 2006).

### Software Details

Publisher	SoftPerfect Research
File Size	248 kB
Version & Last Updated	1.1 - March 14, 2007
License	Freeware
Windows	98/ME/2000/XP/vista

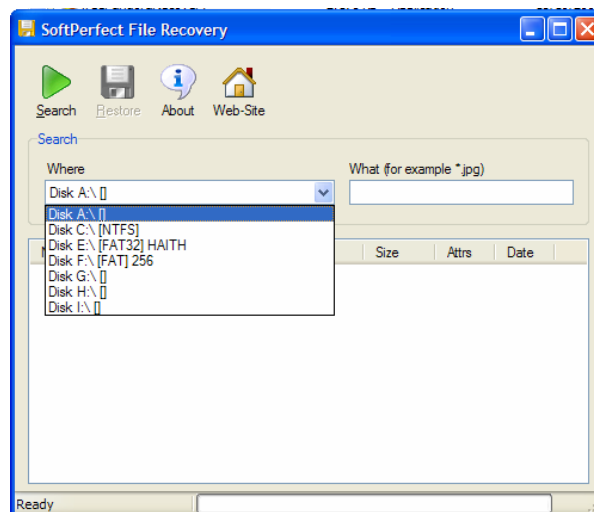


Figure 10. SoftPerfect (main page)

### Tool Functionality

The tool is self executable and simple to use. The tool has a drop down menu to select the removable media for scanning and file retrieval, together with a search facility to specify what type of file to retrieve. The main window displays the findings from the memory card as shown in Figure 10. Once the files are displayed, the user can select the files to be restored and click on the restore button to select the destination of the retrieved files as shown in Figure 11.

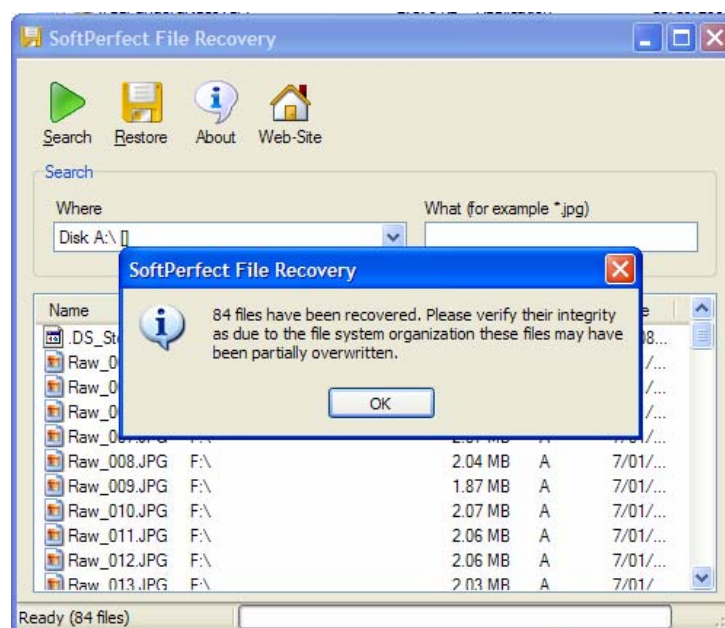


Figure 11. SoftPerfect (retrieved files)

## Undelete Plus

Undelete plus is another recovery tool advertised as fast and effective retrieval of data which has been deleted. It is effective for files that have been removed from the recycle bin and after a system restart. It also supports removable media such as flash drives and secure digital memory cards. (FDRLab, 2007)

### Software Details

<b>Publisher</b>	FDRLab Data Recovery Centre
<b>File Size</b>	1.06 MB
<b>Version &amp; Last Updated</b>	3.0 - August,21 2007
<b>License</b>	Freeware
<b>Windows</b>	98/ME/2000/XP/vista
<b>Site</b>	<a href="http://www.undelete-plus.com/">http://www.undelete-plus.com/</a>

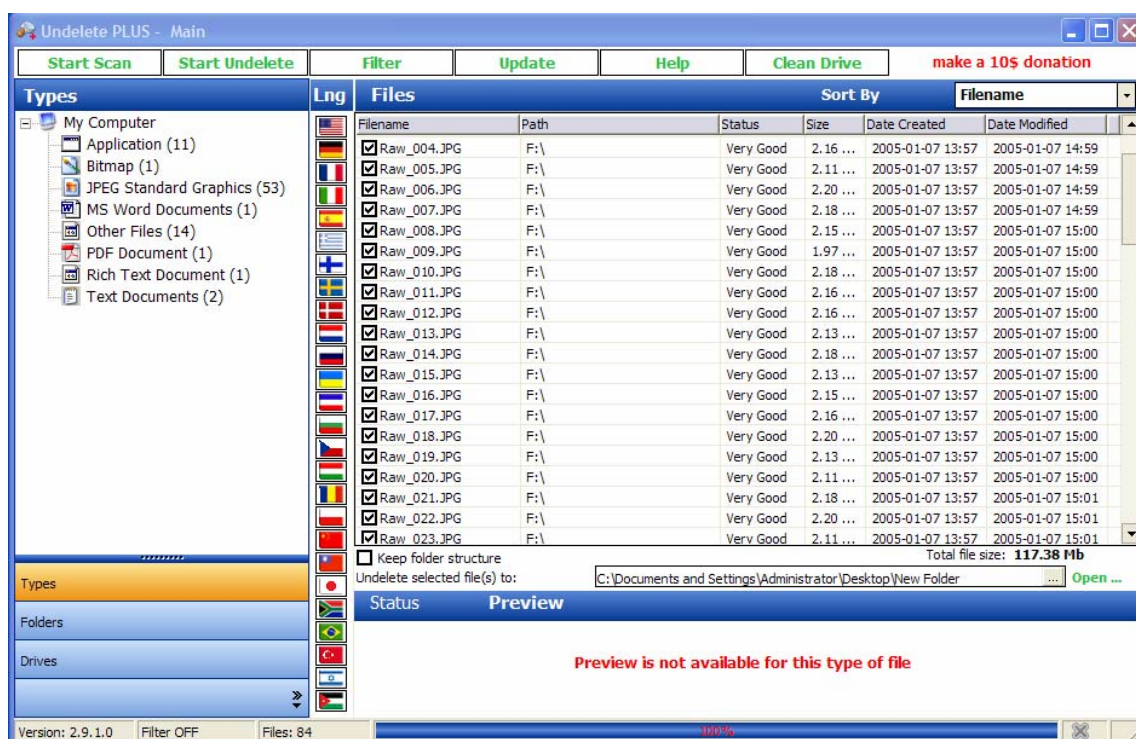


Figure 12. UndeletePlus (retrieved files)

### Tool Functionality

The tool has a lot of functions. The tool displays all storage media available for the user can select the drive to recover form. Once selected a list of file extensions is displayed, followed by the number of files that have been retrieved for each extension. The tool supports multiple languages and each country flag displayed can demonstrate the language spoken in that country as shown in Figure 12. This tool supports filtering so the user can customize the search and retrieval options. It also supports cleaning the drive so the drive will be cleaned for future use.

## RESULTS

Table 1. Results of test scans

Software	Time to Scan/ Retrieve	Number of retrieved files	Number of damaged files (Non Viewable )	Supported formats	Issues
MjM Data Recovery Ltd	39 min 36 sec	132	4	JPG	The scale of the minimum file size is 64k also takes very long time to recover the files
Pc inspector smart recovery	23 min 2 sec	135 files, no folders	4	Multimedia only such as photos, sounds and videos	The tool will retrieve all the files at once , it does not support selecting option to retrieve selected files
Art Plus Marketing & Publishing	21 min 22 sec	504 files	372	JPG	It retrieved 8kb file identified as jpg but cant be viewed out of 504 files 133 viewable file could be retrieved
Free undelete 2.0	17 min 34 sec	21 files and 5 folders	All	JPG and number of unknown files	The images retrieved where not viewable
Recuva –file Recovery	3 min 21 sec	81	All	unknown	The tool did not successfully retrieve the viewable images also a miss match of the displayed retrieved files and actual files in the saved folder ,in this case the tool displayed that it has retrieved 81 files, but the recovered folder shows 64 files
Soft Perfect Research	1 min 4 sec	84 files	All	JPG, DOC, PDF and some unknown formats	The images where not viewable
Undelete plus	1 min 4 sec	57 files	All	JPG, DOC, PDF,TXT and some unknown formats	The images where corrupt and did not open the word nor the PDF file.

## DISCUSSION

### MjM Data Recovery Ltd

The tool recovered the photos in viewable format. Four images were partially corrupt and the rest where viewable. The tool took the longest time of all the tools tested to recover the images but it was complete. The only drawback was the scale of the images at 64k with some thumbnails smaller than 10k and usually they do not get corrupt because they are easy to recover.

#### PC Inspector –Smart recovery

This tool was also reasonably successful and took 23 minutes to scan and recover the files. The total number of files found was 135 files and only 4 files were corrupt but could still be viewed as thumbnails. This tool only worked with multimedia file format, which makes it a very good tool to recover images from solid state drives.

#### Art Plus

The tool took 21 minutes to recover the files, however it did 504 files. 372 of these files were only 8 kb and could not be viewed with the remaining 132 files viewable. This matched the files that were recovered by the MJM tool. However this tool took half as much time to retrieve more files.

#### FreeUndelete 2.0

The tool took some time to scan the files. It retrieved 21 files and 5 folders, but these were not viewable. Other file formats were recovered, however the folders contained different file format than the photos.

#### Recuva

This tool was fast, but unfortunately it did not retrieve as many files as the other programs. In addition, the files were not viewable. It was able to retrieve some other files however these were not identified as it was out of scope of this paper. The display of the retrieved images did not match the actual retrieved files on the saved folder.

#### Soft Perfect

Very fast tool which only took one minute to scan and retrieve 84 files. Unfortunately the files were not viewable. It did retrieve other file formats such as word and PDF.

#### Undelete Plus

This was another fast tool which only took one minute to scan and retrieve files. It recovered 57 files, but the photos were not viewable. Other formats were recovered such as PDF, TXT and other unknown formats. These files were not examined.

## CONCLUSION

This paper tested eight freeware tools, obtained from the internet, to retrieve deleted photos in a viewable format. The results show that the faster tools did not retrieve the photos in a viewable format. However, the faster tools did retrieve some evidence of photos had been present on the SD card and that subsequent forensic examination of these files would be warranted. As such these tools would be useful as an initial rapid analysis of the removable media which would then need further analysis using other tools to recover the viewable images from the memory device. As it has been shown not all of the tools work in the same approach, neither do they recover the identical amount of files. Out of eight tools, three of the tools have successfully retrieved images in a viewable format. The tools have retrieved photos that can not be viewed on the Windows platform, however they may be viewable on other operating systems or photos viewer programs. Future research will look at using a known secure file deletion tool on the card and then rerunning the tests to see how effective they are under secure deletion conditions.

## REFERENCES

- ArtPlus. (2007). *Free Art Plus Digital Photo Recovery*. Retrieved 15 November 2007, from <http://www.artplus.hr/adapps/eng/dpr.htm>
- FDRLab. (2007). *Undelete Plus. Free file recovery software. Retrieve accidentally deleted files*. Retrieved 15 November 2007, from <http://www.undelete-plus.com/>
- GmbH. (2007). *PC Inspector Smart Recovery download and review - recover digital camera files from SnapFiles*. Retrieved 15 November 2007, from <http://www.snapfiles.com/get/smartrecovery.html>
- MjM. (2007). *MjM Free Photo Recovery Software download and review - recover images from media cards from SnapFiles*. Retrieved 15 November 2007, from <http://www.snapfiles.com/get/mjmphotorecovery.html>
- RecoveronixLtd. (2007). *Undelete Plus. Free file recovery software. Retrieve accidentally deleted files*. Retrieved 15 November 2007, from <http://www.undelete-plus.com/>
- Recuva. (2007). *Recuva - Undelete, Unerase, File Recovery - Home*. Retrieved 15 November 2007, from <http://www.recuva.com/>

Softreaserch. (2007). *Restore accidentally deleted files from FAT and NTFS volumes*. Retrieved 15 November 2007, from <http://www.softperfect.com/products/filerecovery/>

## **COPYRIGHT**

[Haitham Al-Hajri & Patricia A H Williams] ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **An overview and examination of digital PDA devices under forensics toolkits**

Krishnun Sansurooah  
School of Computer and Information Science (SCIS)  
Edith Cowan University Perth, Western Australia.  
Email: ksansuro@student.ecu.edu.au

### **Abstract**

*Personal Digital Assistants most commonly known as PDAs are becoming more and more fashionable and affordable in the working environment. With the advent and rapidly increasing technology these handheld devices are now targeted by a lot of person with criminal intentions. But unfortunately crime does not choose its technology and nowadays those ultra light handhelds are getting more and more involved in crimes. This therefore become an onerous task for the forensics examiners who needs the proper forensics tools to investigate the information held on these devices. The purpose of this report will browse the current forensics toolkits available and analyze some targeted PDAs.*

### **Keywords**

PDA, Forensics Analysis, Encase, PDA Seizure, Image Acquisition, PDA Memory

### **INTRODUCTION**

Today's technology is advancing rapidly and when it comes to handheld devices it's even growing quicker especially in their capabilities and in their use. With this increasing technology, it is not a surprise to come across those devices be either PDAs or smart phones which can contain as much processing power as would have held a normal desktop couple of years ago. With those amazing handheld devices, their storage capacities are phenomenon and keep increasing even though these digital devices are getting ultra light in weight.

Being concerned by this evolution, it is therefore necessary that nowadays, the analysis of those handheld devices be combined with the existing digital forensic procedures and methodologies already in place to keep up with the technology. However, most PDAs that are on the market follow similar basic designs, but obviously differs in their operating system (OS), and their hardware components which in turn, unfortunately does not facilitate the forensic data acquisition on the handheld devices without modifying their actual or current state. Having therefore mentioned that this process is not quite easy to performed, the data acquisition can still be performed on the PDAs through some of the currently existing forensic software for that type of acquisition.

To narrow the focus of this research paper, the digital handheld devices looked after would be Palm devices running the Palm OS. This paper will also take into consideration of the different forensic tools available for the acquisition on Palm OS and will not emphasizes on data acquisition on WinCE or Windows mobile phone or Microsoft Pocket.

### **BACKGROUND**

According to Kruse & Heiser (2002) the preservation, identification, acquisition, documentation, interpretation and reporting of computer or digital data is defined as digital forensics. The field of digital forensics has long emphasizes on the seizing and recovering of evidence from a personal computer. But nowadays, criminal are running in parallel with the technology and are hence using the latest up to date devices to achieve to their need in committing illegal activities. With this evolution, the life of forensic experts have become more complicated and forensically acquire these handheld digital devices be either smart phones, pagers, and PDAs have unfortunately become an onerous task.

According to Kruse and Heiser (2002) there are three stages in the acquisition of data for the basic methodology which are as described below:

1. The acquisition of evidence should be performed without modifying or corrupting the original source

2. The acquired evidence needs to be authenticated when raised to the original evidence.
3. The acquired evidence should be analyzed without any alteration to it.

Following Kruse and Heiser (2002) the first stage entails the procedures that need to be observed and fully recorded and documented in the early phases of forensic analysis. These procedures would be

- a) Gathering and collection of evidence
- b) The correct handling of evidence
- c) Maintaining the chain of custody
- d) Identifying the evidence collected
- e) The methods of transporting the evidence
- f) And finally, how the evidence is stored or presented

The second stage described by Kruse and Heiser (2002) in the basic methodology demands that the acquired evidence is verified against the original source. This is a very crucial step in the task of a forensic expert or analysis as this will determine whether the piece of evidence is review from a legally acceptable and presentable process with all the necessary documents to support this finding, especially if these findings are to be pursued to court of law. In a report on digital forensics, McKemmisk (1999) reported that there are four rules to be observed when acquiring evidence for a criminal investigation to be pursued in a court of law.

Those rules to be observed are:

- i) Minimize the handling of original data source.
- ii) Account for any changes in the original data
- iii) Use are follows the rules of evidence, especially when it comes to use of software tools for the acquisition of the evidence

To achieve this composition of acquired evidence versus the original source, the best and most current and reliable way would be archived by electronically fingerprinting the evidence and time stamping both calculated by hashes with cryptographic hashing algorithms such as MD5 Sum check or SHA1 Check. This method is quite reliable in ensuring that the digital evidence in ensuring that the digital evidence has been imaged properly and hence allowing and maintaining the chain of evidence due to the high volatility of the digital evidences.

Using the hashing algorithms allow the digital evidences to be presented in a court of law on the basic that when the incriminated digital device is initially acquired and stored, it can at a later stage be crossed verified to the original source to show and prove that no alteration has occurred during the acquisition of evidence thus keeping the original source intact.

Finally, Kruse and Heiser (2002) elaborate on the analysis of the acquired evidence without any alteration to it by ensuring that the original evidence source has not been found or altered. This process is normally conducted on the imaged copy which is an exact bit wise copy of the original source. This analysis normally starts with examining the files and then a further analysis of physical image or search for either deleted or hidden files. To conduct this process, there are some forensic tools that can be used in the instance of Encase V4, Autopsy, Hexadecimal editors which are toolkits available for refining search though the ASCII and hexadecimal deleted files.

## **DIGITAL HANDHELD DEVICES**

According to Canals (2004), the market of digital handheld portable devices has known a considerable growth and keeps growing in the working environment and also for personal use these digital music or mp3 players, smart phones and without excluding the most common personal digital assistants (PDAs). With the growing technology these PDAs have widely evolved and nowadays are equipped with in-built memory with a minimum capacity of 128 MB and some even more whereas Apple Computer (2004) has announced its digital music player with a capacity higher than 40 GB.

With all these digital devices, the PDAs have been designed to overcome the physical constraints set by either personal computer (PCS) or even laptops. Some of the major advantages that PDAs offer compared to PC or laptops are illustrated below:

- i) The are compact and ultra light thus allowing mobility to the uses;

- ii) They store user data on volatile memory, the Random Access Memory (RAM) and Read Only Memory (ROM) for the operating system (OS)
- iii) They also suspend processor when powered off, to avoid consuming time when rebooting.
- iv) They comprise with organizing functionality in the instance of emails, calendars and memos.
- v) They also offer the ability to synchronize data with a personal computer.

Having therefore enumerated those major differences of the PDAs, it is therefore very difficult and very challenging to soundly forensic those digital devices without the proper and specialized forensic toolkits and also the proper procedures due to the PDAs architecture. In the PDA family there are at present 3 main OS which shares the market. Those are Palm OS, Microsoft Pocket PC and finally portable Linux-based OS which regardless of their brand or family, those digital devices all support some basic functionalities such as contact, email, task management and calendar known as Personal Information Management (PIM) applications. And since we are turning to the new age of technology evolution, the PDAs market share is tending to split into only 2 categories now with are the most 2 dominant ones Palm OS and Microsoft Pocket PC. Another ability of the Palm nowadays is that it has the ability to communicate through wireless medium, surf on the web and even provide editing facilities for electronic document. While those PDAs allow a high level of mobility to their users, they also add up another special aspect to their reputation when it comes to storage of data on the PDAs by introducing the use of removable media such as external media cards with enormous capacities ranging from 128 MB to 4 GB thus making the PDAs more desirable for the users or the criminals.

## **REMOVABLE MEDIA**

Forensic analysis of these removable media is quite similar in the process of analyzing hard drive. These media can therefore be removed and then inserted in a card reader, then an exact image is performed and then forensically examined as unlike the Random Access Memory (RAM) on a device, the removable media is non-volatile and therefore requires no source of prove to retain its data.

Even though removable media are part of the PDAs, the analysis of such media will not be covered in this report but a brief overview of these removable media are described below even small in size, they can however hide enormous amount of data if not gigabytes of data in relation to an investigation.

- ***Compact Flash Cards (CF)***

Compact Flash memory is a solid-state disk card with a 50-pin connector, consisting of two parallel rows of 25 pins on one side of it. They are designed for PCMCIA-ATA; it normally has a 16-bit data bus, and is used more as a hard drive than as the RAM. The flash memory technology is a non-volatile storage media solution that retains its information once power is suppressed from the card. Compact Flash cards are about the size of a matchbook (length-36mm, width-42.8 mm, thickness-3.3 mm for Type I and 5mm for Type II) and consume a minimum amount of power.

- ***Multi-Media Cards (MMC)***

A Multi-Media Card (MMC) is also a solid-state disk card but with a lower number of pins (7-pin connector). It has a 1-bit data bus and same as the compact card, it is hence designed with flash technology, a non-volatile storage solution that retains information once power is removed from the card. The cards contain no moving parts and provide greater protection of data than conventional magnetic disk drives. Those Multi-Media Cards are about the size of a postage stamp but do have in the same family a reduced size Multi-Media cards (RS-MMC) which is half the size of the standard MMC card. Even though they were designed to fit mobile phones, it can nevertheless be used with PDAs.

- ***Hitachi Microdrive***

Hitachi Microdrive digital media is a rotating mass storage device with high-capacity, contained in a Compact Flash Type II having a 16-bit data bus. A micro glass disk is opted as the storage media, which is obviously more fragile than solid-state memory and which do require energy to rotate. As in for the flash memory cards, the 6GB Microdrive storage card is preloaded with a FAT32 file system required to authorize storage over 2GB. In doing so, more space can be easily accessed according to Hitachi Global Storage Technologies (2004).

- ***Secure Digital (SD) Card***

SD Card Association (2004) mentioned that the Secure Digital (SD) memory cards can be compared to the solid-state design of MMC cards. However, the SD card slots often can accommodate MMC cards as well with their 9-pin connector and 4-bit data bus; it can therefore allow a quicker transfer rate. SD cards do offer an erasing in erasure-prevention option so that data cannot be deleted accidentally. Another option that is offered



by the SD card is the security controls for content protection (in other words Content Protection Rights Management). MiniSD cards are also available and do run on the same principle but in a more compact with the same hardware bus and same interface as in SD cards. It does offer the same prevention as SD cards but in a smaller dimension depending on their capacity.

▪ **Memory Stick:**

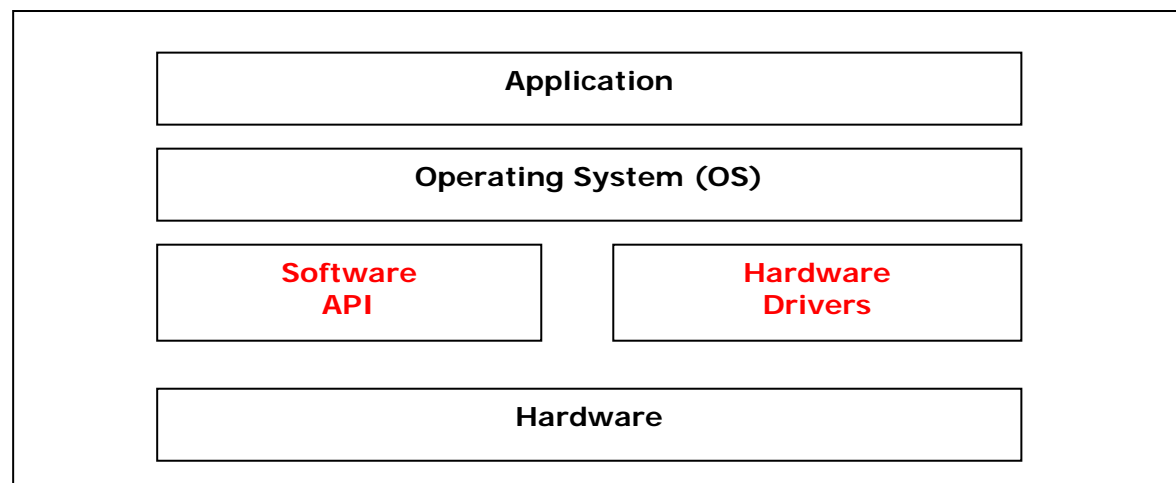
Following Memorystick.com Business Center (2004), memory sticks are also solid-state memory in a smaller size. It has a 10-pin connector and a 1-bit data bus. Same as SD cards, it also has an erasure-prevention switch in build in it to stop the card's content to be erased unintentionally. It therefore offers higher capacity in storage media and quicker transfer rates than standard memory sticks.

## **PDA HARDWARE AND SOFTWARE**

As mentioned earlier, PDA support a set of core Personal Information Management (PIM) capabilities and most of the PDA allow communicating wirelessly through networks with validation and authentication. Therefore data stored on a PDA can be synchronized with either a laptop or desktop PC and would hence facilitate using a synchronization protocol. These protocols can be used to transfer all kinds of data be either text, audio, jpeg images and archive file format

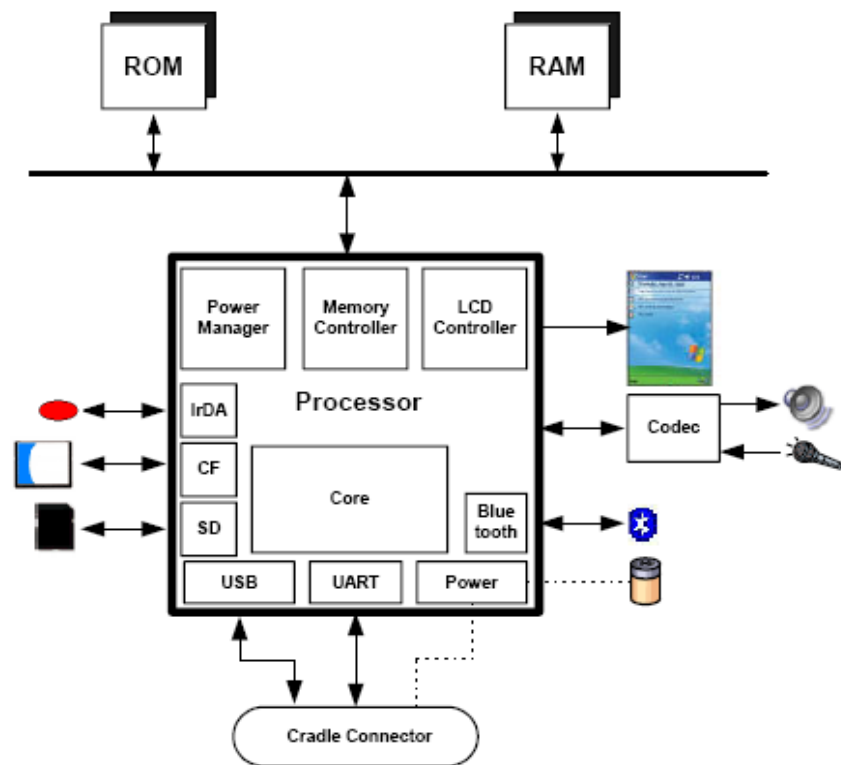
## **PALM OS ARCHITECTURE**

According to Grand & Mudge (2001), the Palm OS are built-in applications which are stored into the Read Only Memory (ROM) while both the user and application data rest into the Random Access Memory (RAM). In a report , Tanker B (2004) stated that add-on utilities are also used frequently to back up PIM data onto ROM. In an article published from the Palm OS Programmer's Companion (2004) that Palm OS split the total available RAM into 2 logical areas which are dynamic RAM area and storage RAM area. The Dynamic RAM area is complied into a single heap and therefore used the working areas for temporally allocations, independents of the RAM mounted on a typical desktop system. However, the rest of the RAM is hence designed as storage RAM area to be used by when holding non-volatile user data. In Palm OS the memory storage in compiled into records which in turn are held in database – here the equivalence of files. The different layers of the Palm OS architecture comprise of Application, Operating System, Software API & hardware drivers and Hardware. Figure 1 below online the different layers level and their relationship in between when communication is effected



*Figure1. Demonstrate the different layers of the Palm OS architecture.*

With technology increasing at the tremendous rate, the latest PDA comes with so many advantages bundled to it which makes its very likely to be possessed by everyone which have a 'busy life' due to its considerable capacity of memory, its powerful micro processors with wireless communication devices embedded on such as wireless, infrared, and Bluetooth technology. A generic hardware diagram of the system level microprocessor is illustrated in Figure 7 below.



*Figure 2. Illustrates the generic hardware diagram of the system level microprocessor*

## **PDA SOFTWARE**

Together with the purchase of a Palm OS, there is some software that comes along with the handheld digital devices which therefore ease the user to synchronize its Palm OS with its computer system at a later stage.

- **PALM DESKTOP**

This software which is normally delivered on the purchase of a Palm enable the user to organize and manage the data which they have stored on their PDA earlier and it therefore helps the user to trace back what editor took place where and when. In other words, it looks after date, address and any memo entry at that time thus providing the user with a more convenient way of making his/her entry into the Palm.

Together with the Palm Desktop, it also allows to install HotSync Operations. HotSyn has been developed by Palm which enables the user to synchronize the data between a personal computer and the digital handheld device. It therefore, gives the user the capability of transferring data between the digital device and the personal computer, which can be also used on a back in case that data came to be lost with the draining of the battery.

- **PALM DEBUGGER**

The Palm Debugger used in Palm commands is usually carries out in low-level system debugging. This debugger is attached on all Palm devices and into the Palm OS Emulator

According to Fernandes (2003), there are two different modes that Palm PDAs can enter which are “Debug Mode” and “Console Mode” The Palm Debugger which is included on all Palm handhelds provides a low-level system debugging for Palm application. This would therefore describe Fernandes (2003) first mode

## PDA FILE SYSTEM

In Palm OS technology, the use of Hot file system differs from the traditional system. According to the Palm memory architecture give a detailed illustration of the Palm OS memory structure and analyzer the essential building blocks of the Palm memory Figure 3 outlines the pattern view of the RAM showing the layer of the dynamic RAM area and the storage RAM area.

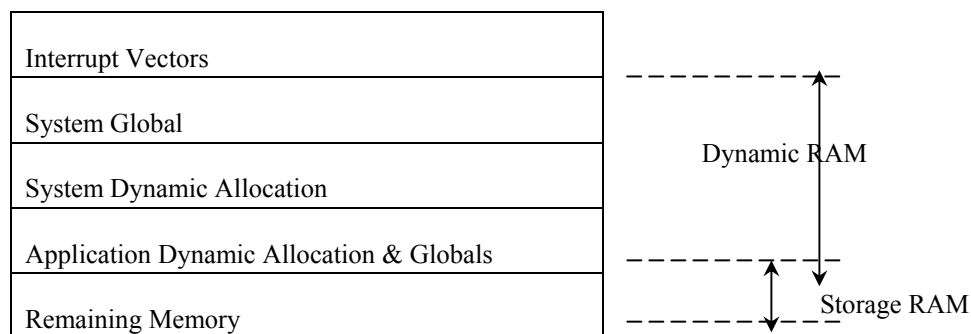


Figure 3. outlines the pattern view of the Dynamic RAM and the Storage RAM.

Dynamic RAM can be compared to the RAM sitting onto a typical personal computer system but in Palm OS the size of the dynamic RAM area would depends on the version of the OS and on the total available memory onto the device and this would keep changing continuously during the usage of the handheld device whereas the remaining RAM is used as storage RAM area similar to using a disk drive

## FORENSICS AND PDA

With the increase of those powerful digital handheld devices, the methodologies and procedures in place for the analysis of digital forensics is being re-examined, re-considered and re-executed to adapt to the new age of digital handheld devices such as PDAs, portable digital music devices and mobile phones. Having to reconsider the methodological approach to these new handheld devices, the two most crucial parts in soundly forensically examining those devices are the acquisition stage and the authentication stage as in any basic computer forensic analysis. However, in the case of Palms this task would be most delicate and important as it should be delicately and correctly carried out to the maximize accuracy on the Palm which in fact rejoin what have been mentioned earlier that most PDAs depends on transitional storage.

A crucial aspect of the PDA vis-à-vis the acquisition and analysis in the use of their memory – i.e. both the RAM and the ROM – when it comes to the storage of data on the PDA and their OS as RAM storage is volatile, the PDA is powered by a battery that allows the memory to be kept alive and hence conduct the operation needed such as storing of data on the PDA. Yet, carrying forensic analysis or acquisition on this device would be very risky as such operation would definitely required draining the battery power hence causing all data in the RAM to be lost similarly as on a PC when is switched off, which discards the data on the RAM. Therefore much care and consideration should be given in the acquisition of PDAs which are quite delicate handheld devices in comparison to personal computers.

## FORENSIC TOOLKITS:

When it comes to forensic acquisition and analysis of PDA, the variety and number of toolkits are very limited compared with PCs or workstations. The specialized toolkits are very limited and are restricted to the most popular PDA handheld devices based on either Pocket PC or Palm OS.

In the section of this paper, a more in-depth analysis of the forensics acquisition would be emphasis on Palm OS. As in for Linux based devices forensic acquisition can be achieved by using the ‘dd’ utility similarly as in Linux desktop and hence later be injected into a forensic tool such as Encase or Autopsy. Considering that Palm OS have been around longer than its competitors, the pool of forensic tools available for such practice is much wilder compared with the other handheld devices. Also consideration should be given that the forensic examiner has full access to the handheld devices. Having examined some of the forensics toolkits available, a table has been designed to tabulate the different facilities that these tools often.

TOOLS	PALM OS	POCKET PC	LINUX BASED
Encase	ACQ / EXA / REP	N/A	EXA/REP
PDA Seizure	ACQ / EXA / REP	ACQ / EXA / REP	N/A
Pilot-Link	ACQ	N/A	N/A
POSE	EXA/REP	N/A	N/A
pdd	ACQ	N/A	N/A
dd	N/A	N/A	ACQ

### Legend

ACQ – Acquired	EXA – Examined	REP – Reported	N/A – Not Available
----------------	----------------	----------------	---------------------

Table 1: PDA Forensic Toolkits.

Actually acquisition of data from a device is carried into 2 different ways:

1. Physical acquisition – in this particular type of acquisition, an exact copy bit-by-bit is collected of the entire physical storage which can be either a RAM chip or a Disk drive.
2. Logical acquisition – which implies an exact copy bit-by-bit of the logical storage such as file and directories, involved residing on a logical store which could be several disk drives. The actual difference however lays in between the memory as separate by a process through the OS facilities, known as logical views against the memory as interpreted by the processor and other related hardware components known as physical view.

Moreover, physical acquisition is much more preferred compared to logical acquisition due to its numerous advantages. With physical acquisition, deleted files and other small remaining pieces of data are looked into closely which could be missed out while carrying a logical acquisition. Another aspect of physical acquisition which is once more preferred to logical acquisition is that physical device images can easily be imported in a different forensic toolkit for examination and reporting. Yet, logical acquisition has an advantage in providing a more natural and readable organization of the data acquired. Having looked at both types of acquisitions, it is recommended that both methods are practiced when undergoing forensic acquisition and analysis of Palm OS.

This is also the concern of using non-forensic tools for the acquisition process which normally focuses on the logical acquisition using an available protocol for device synchronization thus allowing a flow of communication to the handheld device as opposed to a debugging protocol that can be dedicated in acquiring a memory image. With those non-forensic tools which support a two-way communication, the flow of information issues will be raised up when such non-forensic tools are unconcerned and negligent with modifications happening leading to avoid checking hashes of acquired images of the PDA before and after the acquisition stage henceforth losing the integrity of the images acquired which will definitely be questionable at time of court. Such example would be when a tool unwillingly modify time stamp on the PDA device such as time and date of last synchronization.

According to Casey (2000), some tools of Palm OS devices require that the device to be placed in 'console delay mode', maintaining the option of a soft reset after collection. This soft reset triggers huge compilation and therefore deleted records may be overwritten. This being a double edge sword allow non-forensic tools to retrieve the evidence relevant to the information or data gathered or might cause evidential information or data to be overwritten or altered as mentioned earlier in the time stamping leading to the loss of unrecoverable pieces of data or information evidence.

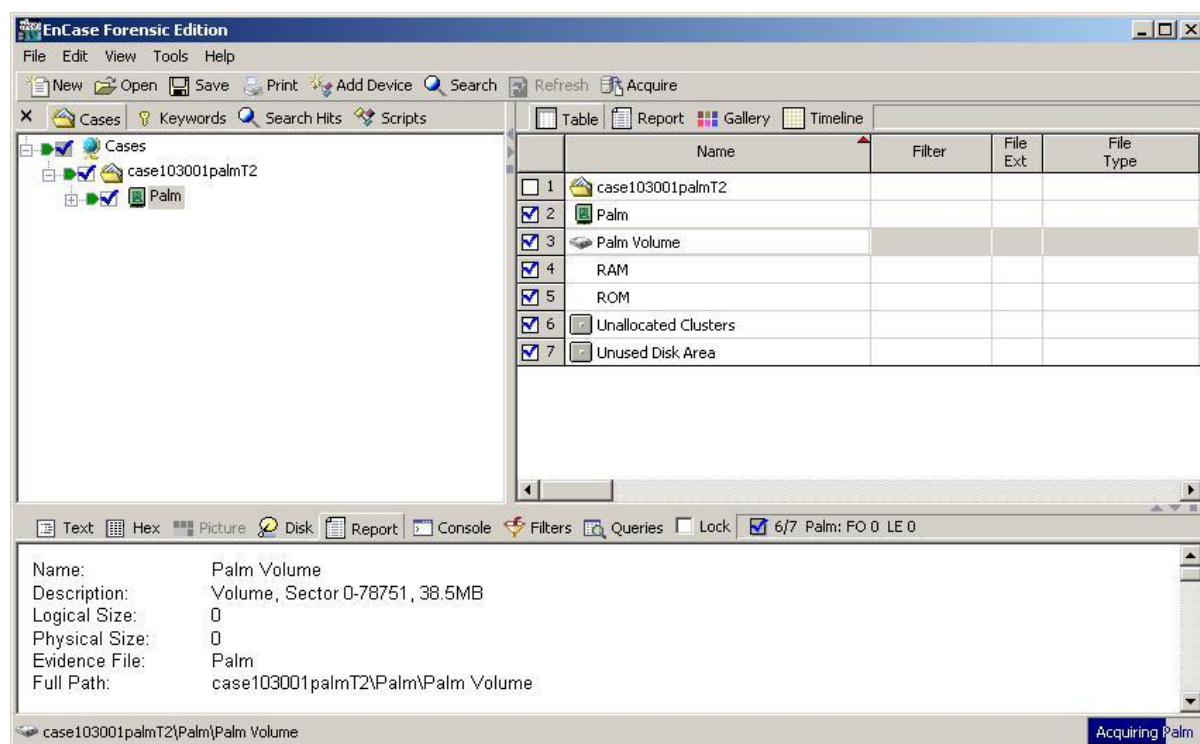
### ENCASE FORENSIC TOOLKIT (Version 4)

Encase being a commercial forensic software toolkit claims to be one of the fully integrated software that allow forensic examiners to acquire suspected PDAs, search and analyze those media with the feature to generate reports in a simple environment. It also automatically generates finger prints, i.e. generation of hash values for both individual and group files and data capture hence, providing the examiner with immediate verification of any suspected acquired evidence. Even though Encase is more widely used for the examination of PC, it includes the support for Palm OS handhelds. The latest Encase software (Version 4) does support the Palms family

including Palm IIx, Palm IIIxe, Palm V, PalmVx and Palm m series. According to Guidance Software (2004), Encase also support internal macro programming language, multiple cases and multiple acquisitions. The choice of using Encase for forensic analysis of Palm OS devices resides is the strength that it can generate on entire physical bit-stream image of the Palm OS device which continuously compared the Cyclic Redundancy Checksum (CRC) blocks. This bit-stream image is being identified as an Encase evidence file which can therefore be mounted as a “virtual drive” permitting the forensic examiner to search and analyze the content of the device using either physical or logical perspective.

Moreover, Encase have the option to save files, folders or partial sections of a file for later reference known as bookmarks. These bookmarks are recorded in the case files with each having a unique bookmark file and can therefore be accessed and consulted at a later stage.

Encase software is relatively important in the writing of this paper due to it’s ability to acquire Palm OS devices which can be accessed and analyzed like any other raw image imported into the forensic toolkit. It is one of the most complete and fully integrated software for an examiner to search and analyze a Palm based PDA but unfortunately Encase does not support analysis of other PDA in the name of Windows Pocket PC or Linux based PDAs. However, the examiner must not limit him to using only Encase as forensics toolkit for the acquisition and the examination of PDAs as there are also other forensic tools available on the markets which are described further in this paper.



*Figure 4 shows an acquisition phase by Encase.*

## **PDA SEIZURE TOOLKITS**

Another interesting toolkit which has been developed by Paraben is known as Paraben’s PDA Seizure. The latest version of PDA Seizure 3.03 allows the forensic investigator to acquire data, conduct an analysis of the acquired data and finally create a report. Contrary to Encase, PDA Seizure has been developed and adapted for either Palm OS devices or Microsoft Pocket PC and can therefore acquire both digital devices.

One amongst the features of the Paraben PDA Seizure is that it can create a forensic image of the handhelds and allow the investigator to conduct searches on the data acquired earlier, and later to execute a report generation of its findings.

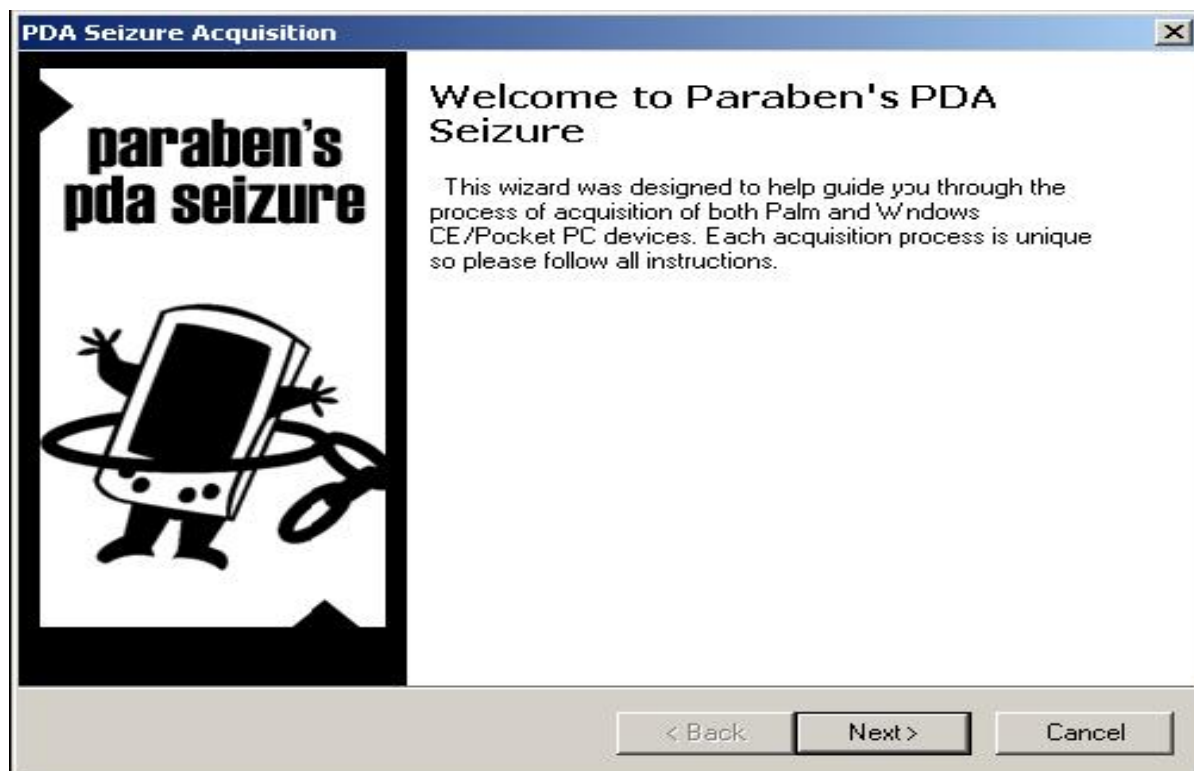


Figure 5 shows the acquisition wizard of the PDA Seizure

Another advantage of the PDA Seizure software is that it allows the investigator to analyze or search the registry of those devices thus choosing the type of acquisition is either physical or logical images as mentioned earlier.

PalmSource Inc.(2002) mentioned that PDA Seizure is preferred to Encase as it not only acquires images of the RAM and/or ROM, but it can also download the entire individual database off the Palms which can therefore be injected though Palm OS Emulators.

To acquire images in Palm OS, the PDA must first be entered into the '*debug mode*', most commonly referred to as '*console mode*' and all active HotSync applications should be exited. This is achieved by entering as Short keystroke combination by drawing a cursive 'L' character and taps the stylus twice to number "2" illustrated in the figure 6 below.



Figure 6 shows how to type in the cursive 'L' followed by the Dot and the number '2' to put the Palm in 'Console Mode'

This '*console mode*', once enable will listens on the RS232 serial port and the Universal Serial Bus (USB) port for communication from the host. Once the image of the Palm OS memory is acquired, the investigator is requested to select the HotSync button on the PDA but his time to acquire the logical data which need to be performed separately even though it has been acquired through physical acquisition. However, the forensic investigator has to bear in mind that during the "*Console Mode*" stage, the power consumption will significantly increased and has to ensure that the Palm has sufficient battery power before engaging into the acquiring process.



File Path	File Name	Type	Create Date	Modify Date	Attr...	Size	Status	MD5 Hash
	Registry					221,324	Registry	8C3B2C3C3D6
	MemImage					93,266,672	MemoryIm	676B2351D31
{Storage Card\}	ignore_niy_docs		2003/07/03 01:23:48	2003/07/03 01:23:48	HA	0	Acquired	
{Storage Card\}	f1.png	.png	2003/07/03 01:24:06	2003/07/03 01:18:34	A	4,545	Acquired	591755C36AC
{IPAQ File Store\}	BioSwipe.cpl	.cpl		2003/07/02 04:35:50	A	2,212	Acquired	91684FCCA3A
{IPAQ File Store\}	ignore_niy_docs		2003/06/18 07:03:19	2003/06/18 07:03:19	HA	0	Acquired	
{IPAQ File Store\Compaq\Nevo\UserData\}	3D81.dat	.dat	2003/06/19 01:37:36	2003/06/19 01:37:36	A	1	Acquired	93B885ADFE0
{IPAQ File Store\Compaq\Nevo\UserData\}	C05B.dat	.dat	2003/06/19 01:37:37	2003/06/19 01:37:37	A	1	Acquired	93B885ADFE0
{IPAQ File Store\Compaq\Nevo\UserData\}	673B.dat	.dat	2003/06/19 01:37:37	2003/06/19 01:37:37	A	1	Acquired	93B885ADFE0
{IPAQ File Store\Compaq\Nevo\UserData\}	4F2C.dat	.dat	2003/06/19 01:37:37	2003/06/19 01:37:37	A	1	Acquired	93B885ADFE0
{IPAQ File Store\Compaq\Nevo\UserData\}	9E4A.dat	.dat	2003/06/19 01:37:37	2003/06/19 01:37:37	A	1	Acquired	93B885ADFE0
{IPAQ File Store\Compaq\Nevo\UserData\}	Rooms1.dat	.dat	2003/06/19 01:37:37	2003/06/19 01:37:38	A	540	Acquired	E5C79F3715E
{IPAQ File Store\Compaq\Nevo\UserData\}	Users1.dat	.dat	2003/06/19 01:37:38	2003/06/19 01:37:38	A	72	Acquired	F55948DCC7
	mdmlog10.txt	.txt	2003/07/03 03:24:03	2003/07/03 03:24:03	A	54	Acquired	0CA8F822045
	GCounterFile.mmf	.mmf	2003/07/03 01:24:40	2003/07/03 01:24:40	HA	10,500	Acquired	98E85D1AF95
	CMMMapP		2002/06/27 21:00:01	2002/06/27 21:00:01	HA	56	Acquired	BFEAC405E80
	CMMMapG		2002/06/27 21:00:01	2002/06/27 21:00:01	HA	60	Acquired	619E024E905
{Program Files\PHM Tools\}	regedit.exe	.exe	2002/11/11 14:58:20	2002/11/11 14:58:20	A	68,608	Acquired	6ED834835F8
{Program Files\}	IPAQ Image Viewer.	.lnk	2002/06/27 12:59:50	2002/06/27 12:59:50	A	24	Acquired	33D88F9142A
{Program Files\Windows Media Player\}	Welcome To Window.	.wma	2002/06/27 12:59:50	2002/06/27 12:59:50	A	24	Acquired	818AA698890
{Program Files\Windows Media Player\}	default.skn	.skn	2002/06/27 12:59:50	2002/06/27 12:59:50	A	28	Acquired	6ED00F8218A
{My Documents\}	f3.png	.png	2003/07/03 01:18:34	2003/07/03 01:18:34	A	4,685	Acquired	78810A8FAC1
{My Documents\}	f1.png	.png	2003/07/03 01:18:34	2003/07/03 01:18:34	A	4,545	Acquired	591755C36AC
{My Documents\}	Recording1.wav	.wav	2003/06/18 07:03:18	2003/06/18 07:03:18	A	2,868	Acquired	C3C0F42E1F8
{My Documents\Business\}	IX.psw	.psw	2003/06/18 07:05:21	2003/06/18 07:05:21	A	8,880	Acquired	DB748D373DE
{My Documents\Templates\}	Vehicle Mileage Log.	.pxt	2002/06/27 12:59:50	2002/06/27 12:59:50	HRA	7,498	Acquired	9C91B8EFB:3

Figure 7 illustrate an acquisition of a PDA through PDA Seizure

## PILOT – LINK

Open source software developed and designed by Linux community to provide a communication bridge between Linux host and Palm OS digital devices which is known as pilot-link. This software is thus comparable with other platforms such as Mac OS and Windows. Pilot-Link uses the HoySync protocol for acquisition which is of interest to the forensic investigator. Normally there are 2 types of programs which are of interest to the investigators which are pi-getram and pi-getrum which basically extract the content stored on either the RAM or the ROM of any device. Another interest piece of software is the pilot-xfer, which install program and create backup and allow restoration of databases hence provide a means of logically acquiring the content of a device. Once acquired, the retrieved contents can therefore be manually searched through or analyzed using either Encase or POSE or a HEX Editor. Pilot-Link does not support any generation of hashing algorithms of the acquired information which unfortunately to be have generated by a different utility to match them for comparison.

## PALM OS EMULATOR (POSE)

Palm OS Emulator (POSE) is software that was developed by the Palm Corporation that run on Desktop computers under different operating system and would give the exact replica of how a Palm OS device would behave once the appropriate ROM has been loaded. However, the POSE software can also be categorized into the forensic toolkit for the investigators.

Once the image has been acquired using either pilot-link or another acquisition tool and loaded into the emulator, the investigator will have the possibility to work with on image of the seized handheld in its original source without any tampering occurring thus allowing the investigator to an application same as would the original user do.

Although POSE was software that was originally developed to run, test and debug Palm OS applications without having to physically download them onto the device, it can also be configured to map Palm OS serial port to one of the available port of a PC. This emulator also reveal as a useful tool for capturing screen shots of evidences found on incriminated or seized devices.

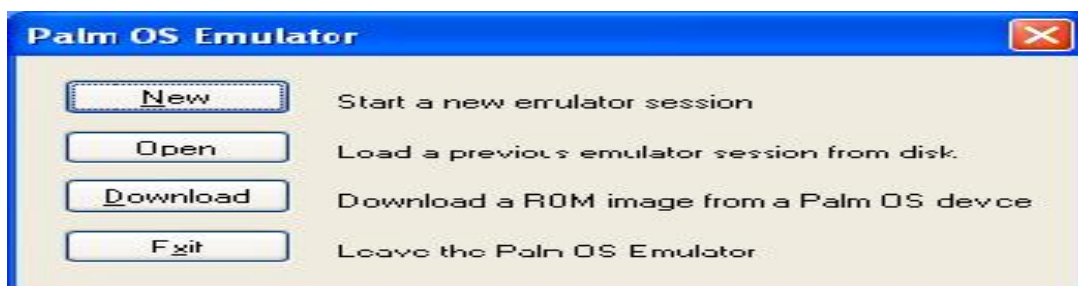
Then the acquired images are transferred onto a PC which can have different file formats also known as Palm File Format (PFF) which matches to one of the three types described below by Howlett (2001)

- a) **Palm Database (PDB):** A record which is used to store application or user data.
- b) **Palm resource (PRC):** Palm application file that is uncompressed and can be installed directly from your PC to your Palm during synchronization.
- c) **Palm Query Application (PQA):** A Palm file containing www content for use with Palm OS wireless devices.

While the Palm OS Emulator (POSE) is installed on the PC, PDA Seizure should also be installed. POSE is therefore used to search for data in association with the PDA device within a desktop environment. The use of the emulator hence allows the examiner to have a closer look to the data that are not held by the internal viewers of the PDA Seizure.

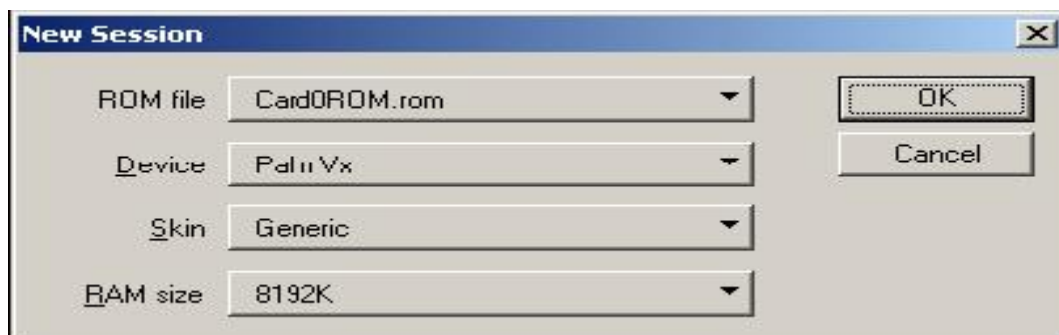
A brief outline of the steps that should be performed while installing POSE with PDA Seizure.

- 1) Install the POSE emulator during the installation of PDA Seizure
- 2) Perform acquisition from the PDAs.
- 3) Select tools from the PDA Seizure Menu Bar and then Export All Files
- 4) Exporting All files will lead to the creation of 2 subfolders.
  - a) Card0-RAM and
  - b) Card0-ROM
- 5) Examiners should use the ROM acquired instead of downloading a ROM images, because of the ROM upgrades.
- 6) Start POSE
  - a) Select Tools and then Palm Emulator
  - b) Select New and then start a new emulator session



*Figure 8 shows how to start the Palm OS Emulator.*

- 7) Select the ROM file
- 8) Choose Other
- 9) Select the ROM image that was saved to the Card0-ROM folder.



*Figure 9 shows how to select the ROM or device*



Having selected the different ROM or device, the POSE will start the session and to be able to view specific files in the POSE you just have to drag and drop the different files of nature PRC, PDB, PQA onto the POSE emulator. Figure 9 shows an illustration of how the POSE emulator would look like after having loaded the imported acquired ROM or RAM of the handheld. POSE is a very useful way to perform virtual displays and confining these screen shots of relevant information as demonstrated below.



*Figure 10 shows how the POSE emulator would look like after having loaded the imported acquired ROM or RAM.*

### **PALMDD (pdd):**

Designed by Grand Idea Studio; Palmdd or pdd can be used for imaging all devices running Palm OS. It is a window-based command line tool that allows the investigator to carry out the physical acquisition of data from Palm OS handhelds. The process carried out during the acquisition stage, is a bit-for-bit image of the memory's device is collected. Since pdd is a command line driven application, some features such as graphic libraries, generation of report, search facilities are not included. However, when the information has been acquired, two files are created, one containing the device-specific information such as RAM and ROM size, processor type, OS version and the second one, with the bit-for-bit image of the PDA. The created files in pdd can then be exported to a different forensic toolkit in the instance of Encase or Autopsy or simply by using a hexadecimal editor according to Carrier (2002).

### **DUPLICATE DISK 'dd':**

The duplication disk 'dd' tools is similar to pdd and which is so far most familiar to forensic investigators as being one of the original Unix utilities which has been around for decades. To use the 'dd' command, the PDA needs to be connected directly to a PC and executing the 'dd' command and hence dumping the content in another location.

But this operation should be carefully carried out as 'dd' may destroy part of the file system thus overwriting the stored data. Images created with 'dd' can be imparted into different forensic toolkit or can be mounted onto a Linux machine for analysis. Same as pdd, 'dd' does not support the creation of hash values which need to be carried out by another utility.

## **ANALYSIS OF THE FORENSIC TOOLS**

In order to measure the scopes of the forensic toolkits mentioned earlier in this paper, the approach of a very basic methodology was applied. The first step would be gathering a set of handheld digital devices which would

be used for forensic examination. Also a set of recommended tasks would be expected to be performed on the collected devices and finally after acquiring the images of the devices with the different available toolkits, we will try to find out if the outcomes of any activities carried out could lead to the recovery of any remnant information.

Having carried out these predefined set of tasks, a table would be generated to illustrate whether the forensic tools have met the required tasks or miss out the stated tasks or whether it has exceeded the expected outcomes or fell below.

Basically, each case will define a set of activities ranging from the data acquisition to file formats and device settings. Even though that these scenarios were not supposed to be comprehensive, they however, try to condense the situations that are most frequently encountered while conducting analysis or examination on a PDA such as data hiding, data purging, etc. This is therefore illustrated below in table 2.

No	Area	Actions Taken	Expectations
1.	<b>PIM Applications</b> – To find out whether the forensic tool can trace out deleted information related to the PIM applications such as memos, calendar, emails, and contacts and to do list.	Creation of some PIM files on the Palms, delete some entries made and then acquire the contents of the Palms, and display the information.	Expecting that all PIM – related information on the Palm OS can be found and recorded if it has not previously been deleted. Also that part of information or data be recovered and recorded.
No	Area	Actions Taken	Expectations
2.	<b>Web/ Email Application</b> – To find out if the tool can trace out any web site or email message information obtain via wireless network or synchronized from a PC.	Allow the Palm to visit some websites and make use of email, acquire the contents, try deleting some email on purpose, try to locate display the URL used or visited headers of email messages.	Expecting that data about the most recent visited websites, web mail and recent email activities can be traced out and reported and partial email data can be traced out and recorded.
3.	<b>Graphic File</b> – To find out if the tool can trace out and compile the different graphic files that were viewed on the device.	Enter several types of graphic images, acquire the content, locating and displaying the images.	All files having the common file extensions such as .jpg, .gif, .bmp, .tif and other can be traced back, reported and comprehensively located and displayed.
4.	<b>Compressed File</b> – Try to find out if the root can locate and acquire text, images and other compressed archive files such as .zip formats and less known archive such as .tar .gz tgz .rgv or self extracted exe files.	Load the PDA with several types of files acquired the contents of the PDA, try to locate and display the filename and file contents.	Text, images and other information acquired on the compressed files format be found and reported.
5.	<b>Device Content</b> – To find out if the tool used was successful in acquiring the content of the device.	Install the forensic toolkit on a PC, try to connect with device and start acquisition, and verify gathered data.	Hoping that information sitting on the PDA has been acquired successfully.
6.	<b>Erased File</b> – Find out the tool can trace and recover deleted files from PDA which involves 2 types: Attempt recovery before and after	It involves in the creation of one or more files, delete the files, acquire the device and try to	All deleted files can be traced out, recovered and reported.

	synchronization with a PC.	locate the deleted file.	
7.	<b>Memory Cards</b> – Try to find out if the forensic tool can acquire individual files stored on removable memory cards which are inserted into the device and if the deleted files can be retraced and recovered.	Put a memory card with a file system in the memory slot of the PDA, delete some files on the memory card, acquire the device, trace out the selected content of the file, including deleted files should be reported.	All input files that have been injected at very start can be traced out, recovered and reported.
8.	<b>Password protection</b> – Find out if the tool can retrieve the users' password to acquire the device.	Enable the password on the palm OS, use any utility to crack the password when acquire the Palm OS device. If the password cannot be obtained try to crack the password.	That the device could still be acquired even the password is turned on.

Table 2 illustrates the different scenario possibilities that may be faced by the examiner when in real life.

Having already defined what would be the expected results, the mentioned scenarios are hence applied to the different Palm OS devices under examination to find out to which extent the forensic tools meets the expected list.

Table 3 below illustrates the different forensic toolkits used against the targeted Palm OS devices and their release version and what we note in that most of the devices used in this experiment come with the operating system already pre-installed by their manufacturer.

Devices	Encase	PDA Seizure	dd	Version
Palm III	✓	✓	N/A	3.0
PDA Palm V	✓	✓	N/A	3.0
Dd Palm Vx	✓	✓	N/A	3.3
Tungsten C	✓	✓	N/A	5.2.1

Table 3. Different forensics toolkits available to analyze the Palm.

### Legend

N/A	Not Available
-----	---------------

Having already identified the forensic toolkits and the targeted devices, under examination, we tabulate the result of each case and hence compared to the predefined expectation illustrated in table 2 to see whether the forensic tool met the expected result.

To achieve this, a scoreboard needs to be defined to determine the different level reached by the forensic tool.

The following entries describe the different actions. "Exceed" in where the forensic tool has overcome the expected result. "Expected" in when the software has met the predefined expectations. "N/F" found would indicate that the toolkit has not been able to meet any expected results. "Poor" would be classified as not having met the expectation and finally "Not Available" would define scenarios that were not subject to the device.

The results are illustrated in Table 4 to the exception of the exception on Tungsten C handheld devices which involve the removable media which was related to deleted file recovery.

## OUTCOMES OF ENCASE TOOLKIT

The cases were carried out using a Windows XP Pro machine together with the targeted Palm devices.

Scenarios/ Cases	Palm III	Palm V	Palm Vx	Tungsten C
1. PIM Application	Exp	Exp	Exp	Exp
2. Web/Email Application	N/A	N/A	N/A	Exp
3. Graphic File	Exp	Exp	Exp	Exp
4. Compressed File	N/A	N/A	N/A	Exp
5. Device Content Acquisition	N/F	Exp	Exp	Exp
6. Erased File	Poor(2)	Poor(2)	Poor(2)	Poor(2)
7. Memory Card	N/A	N/A	N/A	Poor(3)
8. Password Protection	N/A	N/A	N/A	N/A

Table 4 shows the targeted Palms under Encase forensics toolkit.

### Legend

N/F – Not Found	Exp – Expected	Poor – Poor	N/A – Not Available
-----------------	----------------	-------------	---------------------

- 1- Encase being unsuccessful, another method was chosen to perform the acquisition with pdd.
- 2- Only some files were recovered, not entirely.

#### ▪ Detailed analysis of the different Palms III/V/Vx using Encase (v4)

1. **PIM Application:** All the action PIM data was found and reported and all deleted PIM data was also recommend.
2. **Web/Email applications:** Not Applicable
3. **Graphic File:** All graphic files even the .jpg were found and reported. Note that .jpg were connected when the data was synched and displayed in the graphic library
4. **Compressed File:** Not Applicable as when the .zip files were transferred to the Palm, the files were automatically uncompressed before being uploaded
5. **Device Content Acquisition:** Encase could not acquire the Palm device thus having to use pdd for successfully acquiring the Palm which took approximately 25 minutes. The analysis was then imported to Encase as a raw image
6. **Erased File:** Most of the erased files were recovered and reported. Some partial data of graphic files were found but could not be displayed in the graphic library. We note that if HotSync was carried out after the file deletion took place but prior to acquisition stage, the files would be lost for good.
7. **Memory Card:** Not Applicable
8. **Password Protection:** Not Applicable as the Encase software does not include password creating feature.

#### ▪ Detailed analysis of the Palms Tungsten using Encase (v4)

1. **PIM Application:** All the active PIM data was found and reported and most deleted PIM data was recovered expected for calendar and contact information.
2. **Web/Email Application:** Tungsten C has got 802.11b inbuilt capabilities thus allowing sending and receiving email and browse the internet. All visited websites were traced back and reported with all their data such as .tif, text, etc. All emails send out or received were found and reported.
3. **Compressed File:** Not Applicable.
4. **Erased File:** Most of the erased files were recovered and reported. Some partial data of graphic files were found but could not be displayed in the graphic library. We note that if HotSync was carried out after the file deletion took place but prior to acquisition stage, the files would be lost for good.
5. **Device content Acquisition:** Device was successfully acquired within 30 minutes.
6. **Memory Card –** As SD card were used but nothing could be found because the media is a removable one.

7. **Password Protection:** Not Applicable as the Encase software does not include password creating feature.

## OUTCOMES OF PDA SEIZURE TOOLKIT

With PDA Seizure, the targeted Palm devices need to be put in console mode and all active HotSync applications must be exited before beginning the physical acquisition. The Paraben PDA Seizure used is the version 3.03 and were opted to acquire file and memory. If the handheld is password protected, the decode password option should be chosen before any acquisition is carried out and during acquisition stage the investigator is requested to press the HotSync button on the cradle to begin logical acquisition using the HotSync protocol.

Quick Install protocol was used for the transferring of files to the Palm. Given that Palm OS supports only the jpeg format; therefore all other formats when transferred was connected to the .jpg format.

- **Detailed analysis of the different Palms III/V/Vx using PDA Seizure (v3.03)**
  1. **PIM Application:** All active PIM data was found and reported. The deleted PIM data was partly recovered compared to deleted item in the calendar and TO DO list were not recovered whereas deleted data from contact List and Memo list was traced back.
  2. **Web/Email Application:** Not applicable
  3. **Graphic Files:** All graphics were found but graphic type were connected to .jpg format
  4. **Compressed File:** Not applicable
  5. **Device Content Acquisition:** It took 30 minutes and were successfully acquired
  6. **Erased File:** Most of the erased files were recovered and reported. Some partial data of graphic files were found but could not be displayed in the graphic library. We note that if HotSync was carried out after the file deletion took place but prior to acquisition stage, the files would be lost for good.
  7. **Memory Card:** An SD card was used but nothing could be found because the media is a removable one
  8. **Password protection:** Not Applicable as the Encase software does not include password creating feature

Scenarios/ Cases	Palm III	Palm V	Palm Vx	Tungsten C
1. PIM Application	Exp	Exp	Exp	Exp
2. Web/Email Application	N/A	N/A	N/A	Exp
3. Graphic File	Exp	Exp	Exp	Exp
4. Compressed File	N/A	N/A	N/A	N/A
5. Device Content Acquisition	Exp	Exp	Exp	Exp
6. Erased File	Poor1	Poor1	Poor1	Poor1
7. Memory Card	N/A	N/A	N/A	N/F2
8. Password Protection	Exp	Exp	Exp	N/A

Table 5 shows the targeted Palms under PDA Seizure toolkit

### Legend

N/F – Not Found	Exp – Expected	Poor – Poor	N/A – Not Available
-----------------	----------------	-------------	---------------------

- 1 – Some data was recovered but not the entire content  
 2 – Memory Card was not discovered and its content not acquired

### Tungsten C

**Password Protection** – Not Applicable because the OS version is higher than 4.0 hence PDA Seizure could not crack the password.

## CONCLUSION

The outcome produced in this paper reveals that to perform forensic acquisition of PDA devices is not an easy task and with technology advancing rapidly this practice becomes more and more difficult for forensic investigators specially when it comes to acquire images off such devices which are used to store run-time information which sometimes can be updated and altered on a frequently time basics. PDA forensic is a growing area in mobile forensic. However, the forensic toolkits used in this paper behaved as expected. Therefore it is crucial that the forensic investigator knows the limits of the used tool and also when to and how to turn to other forensic tools as means of examination in order to maintain a chain of custody. The outcomes of this experiment definitely emphasizes on proper documentation especially when dealing with mobile handheld devices such as Palm. The investigator should by all means ensure that every actions are fully documented even the slightest change of time or environment to prove that no temping of data has occurred.

## REFERENCES

- Ayers, R. & Jansen W. (2004). PDA Forensic Tools: An Overview and Analysis. Retrieved May 3, 2007 from NIST: <http://csrc.nist.gov/publications/nistir/nistir-7100-PDAForensics.pdf>
- Canalys. (2004). Global Mobile Device Market Shows Tremendous Growth. Retrieved May 25, 2007, from <http://www.canalys.com/pr/2004/r2004081.htm>
- Carrier, B. (2005). File System Forensic Analysis. Addison Wesley Professional.
- Cheong, K.,W. & Wong, L., W. (2005) Forensic Image Analysis of Familiar-based iPAQ. Retrieved May 12, 2007, from <http://www.forensicfocus.com/downloads/familiar-ipaq-forensic-analysis.pdf>
- CompactFlash Association. (2004). CompactFlash Association Homepage. Retrieved April 22, 2007, from <http://www.compactflash.org/>
- Fernandes, L. (2003). Palm OS dot short-cuts. Retrieved May 18, 2007, from <http://www.ee.rverson.ca/~elf/visor/dot-shortcuts.html>
- Frichot, C (2004). An Analysis of the Integrity of Palm Images Acquired with PDD. Retrieved May 11, 2007, from [http://scissec.scis.ecu.edu.au/publications/2004\\_FRICHOT\\_ACNIFC\\_Analysis\\_of\\_the\\_Integrity\\_of\\_Palm\\_Images\\_Acquired\\_with\\_PDD.pdf](http://scissec.scis.ecu.edu.au/publications/2004_FRICHOT_ACNIFC_Analysis_of_the_Integrity_of_Palm_Images_Acquired_with_PDD.pdf)
- Grant, J. (2002). Pdd: Memory Imaging and Forensic Analysis of Palm OS Devices. Retrieved May 26, 2007 from [http://www.grandideastudio.com/files/security/mobile/pdd\\_palm\\_forensics.pdf](http://www.grandideastudio.com/files/security/mobile/pdd_palm_forensics.pdf)
- Guidance Software. (2004). Encase Forensic Edition: The Standard in Computer Forensics. Retrieved May 30, 2007, from <http://www.encase.com/products/downloads/efe-datasheet.pdf>
- Hillerman, G. (2003). Palm OS File Format Specification. Retrieved May 17, 2007 from <http://www.palmos.com/dev/support/docs/fileformats/FileFormatsTOC.html>
- Hitachi Global Storage Technologies. (2004). Hitachi Global Storage Technologies Product Information. Retrieved April 11, 2007, from <http://www.hgst.com/portal/site/en/menuitem.a994b57654279b5daa67bca4bac4f0a0/>
- Howlett, A., (2201) Palm OS Programmer's FAQ. Retrieved May 31, 2007 from [http://palmtops.about.com/od/pdaglossary/g/PRC\\_File.htm](http://palmtops.about.com/od/pdaglossary/g/PRC_File.htm)
- InformIT (2005). PDA Forensics. Retrieved May 18, 2007, from <http://www.informit.com/guides/printerfriendly.asp?g=security&seqNum=104>
- Jansen, W., & Ayers, R. (2004). Guidelines on PDA Forensics. Retrieved May 19, 2007 from <http://www.csrc.nist.gov/publications/nistpubs/800-72/sp800-72.pdf>
- Kruse II, W. G., & Heiser, J. G. (2002). Computer Forensics: Incident Response Essentials. Boston: Addison- Wesley.

- McKemmish, R. (1999). What is Forensic Computing? Canberra, Australia: Australian Institute of Criminology.
- Memorystick.com Business Center. (2004). Memory Stick Media Capacity. Retrieved April 12, 2007, from <http://www.memorystick.com/en/ms/variety1.html>
- Multimediacard Association. (2004). MMCA: Home Page. Retrieved April 12, 2007 from <http://www.mmca.org/>
- PalmSource (2004). Palm OS® Programmer's Companion Volume I. Retrieved April 27, 2007, from <http://www.palmos.com/dev/support/docs/palmos/Memory.html>
- PalmSource Inc. (2002). Palm OS Emulator (Version 3.5). Retrieved April 22, 2007, from [http://www.access-company.com/developers/documents/docs/emulator/Emulator\\_Front.html](http://www.access-company.com/developers/documents/docs/emulator/Emulator_Front.html)
- PalmSource Inc. (2004). Memory | Palm OS Programmer's Companion. Retrieved May 25, 2007, from <http://www.palmos.com/dev/support/docs/palmos/Memory.html>
- PalmSource Inc. (2004b). PalmSource | Palm OS. Retrieved May 29, 2007, from <http://www.palmsource.com/palmos/>
- Paraben Corporation. (2005). PDA Seizure. Pleasant Grove, UT.
- PaulEggleton (2005). FamiliarFaq. Retrieved May 17, 2007, from <http://handhelds.org/moin/moin.cgi/FamiliarFaq#head-87a7fac0185ca2be60ecd1a946827adef5921208>
- SD Card Association. (2004). Concept of SD memory Card. Retrieved April 12, 2007, from [http://www.sdcard.org/sd\\_memorycard/index.html](http://www.sdcard.org/sd_memorycard/index.html)

## **COPYRIGHT**

Krishnun Sansurooah ©2006. The author/s assigns SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.



## Profiling Through a Digital Mobile Device

Lee Fueng, Yap<sup>1</sup> Andy, Jones<sup>2</sup>

<sup>1</sup>British Telecommunications plc.  
Asian Research Centre, Kuala Lumpur, Malaysia

<sup>2</sup>British Telecommunications plc.  
Security Research Centre, Ipswich, United Kingdom;  
Adjunct, Edith Cowan University, Perth, Australia

<sup>1</sup>leefueng.yap@bt.com, <sup>2</sup>andrew28@bt.com

### Abstract

*Mobile digital devices have evolved from telecommunications device to a lifestyle product over the years. These devices are used in all aspects of our daily life. As a result, more information is stored within the devices. Unprotected mobile digital device is susceptible to privacy invasion attack when the device falls into the wrong hand of any unscrupulous third party. The main objective of this paper is to provide an implication analysis on the possible risks of information leakage through digital mobile devices, in the case when users forget to, or choose never to apply any security protection to their devices*

### Keywords

Profiling, mobile devices, security protections

### INTRODUCTION

In this modern era, digital mobile devices such as cellular phones and personal digital assistances (PDA) play a crucial role in all aspects, personal and business, of our daily life. The widespread usage of these devices in today's society is mainly due to the advancement of a range of technologies that enable the creation of an enhanced user experience. This includes faster and cheaper communication access technologies and cheaper and more powerful digital mobile devices in terms of processing speeds and storage capacities. Over the years, mobile digital devices have transformed from the most fundamental voice-base communication gadget into a multi functional device that incorporates a camera, multimedia player, personal organiser, file storage system, text editor and web browser functionalities. With these capabilities, increasing volumes of data are being stored and exchanged between the mobile digital devices. Hence, mobile devices have become a gold-mine for the forensic investigator in serious crimes investigations (Williams 2007) because of the possibility that useful evidence can be obtained or recovered from these devices.

Form Factor	2004	2005	2006	2007	2008	2009	2010
Clamshell	109,539.6	205,741.3	301,108.8	357,477.9	390,069.9	429,105.9	470,929.5
Candy bar	563,720.0	597,148.6	652,560.9	686,317.7	713,339.2	770,832.4	826,272.2
Slider	857.3	12,524.9	29,994.7	50,164.5	69,681.5	90,310.6	110,251.8
Other	-	1,116.5	4,544.8	8,638.6	13,365.6	19,254.6	23,328.9
Total	674,116.9	816,531.3	988,209.2	1,102,598.7	1,186,456.2	1,309,503.5	1,430,782.4

Source: Gartner Dataquest (March 2007)

Figure 1: Forecast of Sales of Mobile Devices to End Users, by Form Factor, Worldwide, 2004-2010 (Units of 1000)

According to the recent mobile device sales forecast by Gartner (2007), it is estimated that by year 2010, the total number of mobile device sold worldwide will reach 1.4 billion. As more people will be using the digital mobile device for multimedia communications, and e-commerce application, more critical information will be stored in the digital mobile device. Consequently more information can be obtained and recovered from the devices if sufficient security features have not been installed or implemented in these mobile devices.

This paper provide an implication analysis on how the information recovered from a totally unprotected second hand mobile device could be use to profile the owner of the mobile device and his other close contacts. Section 1 of this paper briefly discusses the digital mobile devices usage models evolution and the projected future of



digital mobile devices adoption. Section 2 discusses the general public awareness towards the importance of protecting the information stored in their digital mobile devices and briefly goes through the existing technologies which can be used for information protection. Section 3 introduces the potential security implications towards the corporate and individual when no security protection discussed in section 2 is being implemented in the digital mobile devices. One real example is used to illustrate the type of information that could be collected, analysed and inferred from an unprotected digital mobile device. Finally, section 4 concludes the paper with a discussion on the creation of security awareness among digital mobile devices users in order to assist them in the protection of both their personal information and their corporate business secrets.

## SECURITY AND MOBILE DIGITAL DEVICE

By default, information stored in most of the digital mobile devices is not protected against any privacy infringement action. This indicates that anybody who is able to get hold of the unprotected mobile device will be able to retrieve virtually every bit of information that resides within the digital mobile device. The information captured includes, but is not limited to, Short Message Service (SMS) and email history, business and private contacts information, calendar information, call histories and stored data such as images and files. The information may also include the owner's private data, social behaviour and social contacts. (Yap, Jones 2006) If used appropriately, the information could be enough to profile and finally track down the owner.

The most fundamental way to protect Global System for Mobile communications (GSM) based digital mobile device from information leakage in the event of the loss of one's mobile phone is through the setting of both the Subscriber Identity Module Personal Identification Number (SIM Pin) and device based power-on password or most frequently known as the security code (GSM Security 2006). These two security passwords aim to provide different level of protection to the mobile device users. The main motivation of setting the SIM Pin is to protect the SIM card from illegal access while the security code plays an important role in refraining unauthorized third party to use the mobile device even when a new SIM card is inserted into the mobile device's SIM slot. SIM pin and security code are commonly made up of four digits that are configurable by the owner of the mobile device.

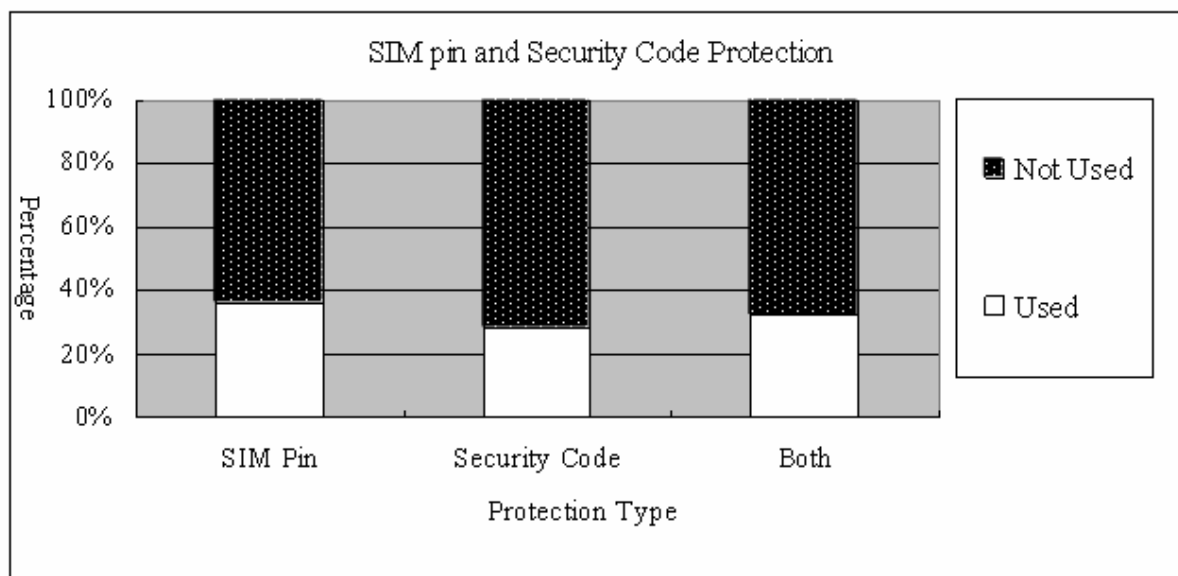


Figure 2: SIM Pin and Security Code Protection Survey

A survey on the habit of setting SIM Pin and security code on the mobile phones has been recently conducted on a sample of 25 Malaysians to gauge their attitudes towards applying non-physical security protection to their mobile phones (BT Research On-Going). The sample consists of both female and male working adults in the age range of 25 to 40. Figure 2 shows the results of the survey. Less than 35% of the people being surveyed apply both the SIM Pin and security code protection. From the responses obtained, the main reasons for not applying any security protection to their phones were a lack of awareness on how critical and useful the information stored within their mobile phones could be if their phones fall into the hands of any unscrupulous third party. The second reason is mainly due to ease of used consideration that deterred most of them from applying these security protections. A third reason being that there are too many passwords to remember nowadays such as personal emails passwords, bank card's pin numbers and company's login credentials. They did not want to further increase the existing passwords list they managed. 20% of the respondents were totally not aware of the existence of the security code protection features within their mobile phone. All of the respondents had heard of

SIM Pin protection features but only thirty six percent of them were using the protection. One of the most significant implications of not applying any security protection to the mobile devices was the compromise of privacy. A possible secondary implication would be potential of financial loss if the SIM in the mobile devices was used for other online transaction authentications such as online banking and e-commerce applications.

Other more complicated security protection mechanisms do exist for individuals or corporate users who are opting for implementing additional security protection to the information stored within their mobile devices. These mechanisms can be broadly classified into two categories namely the preventive and the restorative security mechanism. The preventive mechanism involving the protection of data stored within the mobile device from any aspect of third party illegal invasions while the restorative mechanism focusing on the destruction of data stored within the mobile devices when these devices are reported lost or stolen.

Example of preventive mechanism including data encryption support and authentication services which can be implemented using both software and hardware. Data encryption provides users with the ability to safeguard sensitive personal and corporate data stored in the mobile devices or memory cards. For mobile devices that support Microsoft Windows Mobile 5.0, Symbian, and the Palm Platform, encryption features are being bundled with the operating system. Third party encryption software solutions are also available for users that require slightly better encryption features than those offered by the mobile handheld operating software. The most popular encryption algorithm supported by these third party encryption solutions is the 128-bit/256-bit Advanced Encryption Standard (AES) algorithm (Pointsec 2006) (TrustDigital 2007) (Bluefire Security Technologies 2007). For the most comprehensive encryption protection, users should opt for the encryption capability offered by the Mobile Trusted Module which is a secured hardware base encryption solution. (Trusted Computing Group 2007) On the other hand, restorative mechanism usually relies on software based solution where a copy of automated remote wipe-off software is installed in the mobile device. The software can be triggered automatically through communication networks such as GSM or Wireless Fidelity (Wi-Fi) once the devices are reported lost. This software can not be formatted or removed by any third party and thus guarantee its protection viability.

Preventive mechanisms have several advantages over the restorative mechanisms described above. First, in preventive mechanisms, data has been encrypted and hence user needs to be authenticated to access the data stored in the mobile devices making any attempt to steal information by an unauthorized user deem technical challenging. On the other hand, restorative mechanisms do not mandate the implementation of encryption and authentication to the data stored in the mobile devices. Remote wiping can only be conducted when users report that their mobile device is lost and at the same time the mobile device is connected to the network. The software is then being triggered to erase all the data in that mobile device. Hence, in order to provide a more holistic protection against privacy infringements both preventive and restorative mechanism should be deployed.

Most of the additional security protection measures such as remote wipe-off usually do not come bundled together with standard mobile devices packages. A nominal service charge is usually required for individual that chooses these additional security protections. Nevertheless, the benefits of protecting your data can worth much more than the monthly or annual subscription fees required by the additional data privacy protection. An individual who does not plan to subscribe to any paid security protection service should at least use the data encryption features bundled with mobile operating system besides activating the SIM pin and security code protection that comes free with any GSM based mobile device.

## **RISKS AND IMPLICATION OF LACK OF SECURITY PROTECTION**

The risks and implications of not applying any security protection mechanism described in section 2 are tremendous. For the purpose of security risk investigation and analysis, a second-hand and totally unprotected RIM Blackberry device, obtained from the United Kingdom was used. Essentially the information obtained from the Blackberry device includes both business and private information. With help from the Internet, such as the owner's employer web page, web based interactive maps and public directory servers, the identities of the owner and his contacts were tracked down.

The collected and analysed information was grouped into two categories for ease of explanation, namely business related information and personal information. However, some of the information collected from the Blackberry does not fit distinctly into either of these categories as it appears that it could belong to either category. This type of information is discussed at a separate sub section.

### **Business Information**

Figure 3 summarised the categories and the information occurrence frequency of all business related information obtained from the Blackberry. Corporate email addresses were mainly captured from the inbox folder of the Blackberry while some were found in the address book. A total of distinct 90 corporate email addresses were

recovered from the Blackberry excluding the count of customer and private email addresses. Nevertheless, only approximately 30 emails addresses had frequent interaction with the owner. There are a total of 249 address book entries found on the Blackberry, but less than 2% of the total numbers of corporate phone numbers captured were obtained from the address book. The address book of the Blackberry contained additional corporate customer contact information and some private contact information.

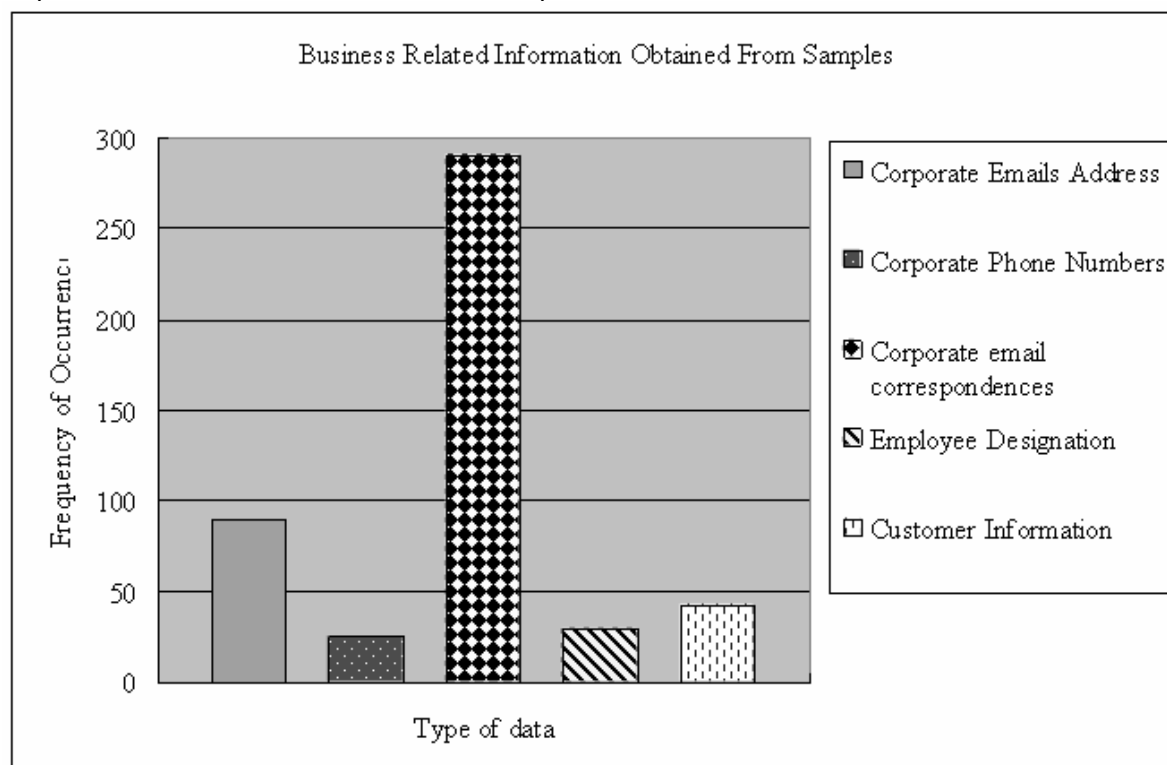


Figure 3: Business related information obtain from Blackberry

A total of 301 undeleted emails were found in the Blackberry. Out of these, 291 emails contained business related information. The email correspondences were identified as one of the best sources that could be used for effective profiling on the owner and his daily activities. From this information, it was found that not only superficial information such as the corporation to which the owner belonged, his roles and responsibilities in the corporation, the locations of the corporation offices, the core business of the corporation and its structure, it was also possible to recover business sensitive information. This information included corporate meeting minutes, sales forecast reports, product pricing strategies, correspondence with customers, competitor's information, products roadmaps and office politics information. From the corporate perspective, this information could be as valuable as the products and branding reputation of the corporation. This information, if used by an unscrupulous individual, could cause huge damage to the corporation in terms of financial loss and reputation damage.

From the email correspondence, a total of 29 employee designations have been identified. This information is useful for any outsider who intended to map out the organization chart of the corporate. Figure 4 illustrated the interpolated organisation chart for that corporation based on the information obtained from the Blackberry. The red box indicates the position of the owner of the Blackberry in the organisation. He is the continental managing director of a multinational organization. Directly under him, there are five main departments lead by individual department directors. The owner of the Blackberry is also in charge of the Group Strategy & Business Development team and the regional office managers. Under the main departments, various personnel have also been identified. The corporation name and individual names in the interpolated organization chart has been purposely omitted to protect the privacy of the individuals involved.

As shown in Figure 3, customers' information is one of the significant elements of information gathered from the Blackberry. A total of 42 customer entries have been identified from the address book, email correspondences history, memo and task folders. Of these 42 customers, less than 15 are identified to have regular correspondences with the owner and his team members. The customers contact details include information such as their core business, phone numbers, email addressees and the relevant contact person with

the Blackberry owner's sales team. This type of information would be valuable to competitors who are fighting for the same pool of customers.

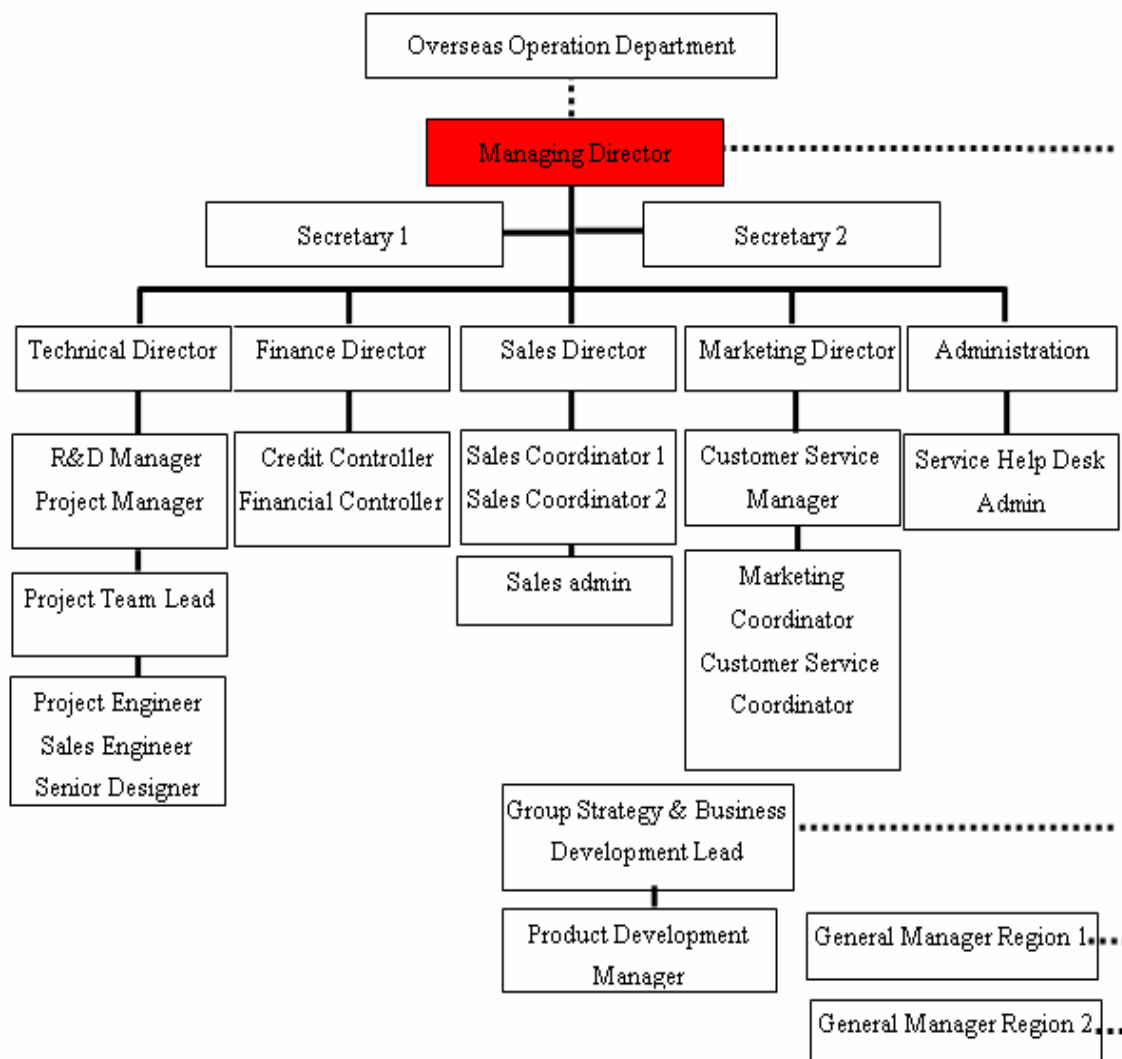


Figure 4: Interpolated Organisation Chart

## Personal Information

Figure 5 illustrates the private information of the owner captured from his Blackberry. His home address information is stored insecurely in the Blackberry. By referring to the Internet based interactive map, the distances of his home and various frequently visited places such as office, laundry, florists, country club, and friends' houses are easily identified. By using the Google Map, it was also possible to interpolate the routes that the owner took to reach these places.

The owner also stored his car dealer and car registration information on his Blackberry. By referring to the car dealer information, we can possibly gauge the brand of car that he is driving and hence deducing his social class. The bank sort code obtained from the Blackberry was used to discover the bank where the owner does his financial transaction. Nevertheless, neither credit card information nor online bank login information was recovered from the Blackberry.

The owner's family information such as the family tree structure, the family members names, occupations, home addresses were all found in the Blackberry. From the email correspondences and the recovered deleted emails, it was found that the owner has two sons. One is working in the car industry and the other is still attending high school in a town near to where the owner stays. His oldest son shares his interest in football and it is common for them to spend time watching football matches together. The younger boy's mother lives in a town that is about 30 miles away from the owner's home. No further information was found with regards to the relationship between the owner and the mother(s) of his sons.

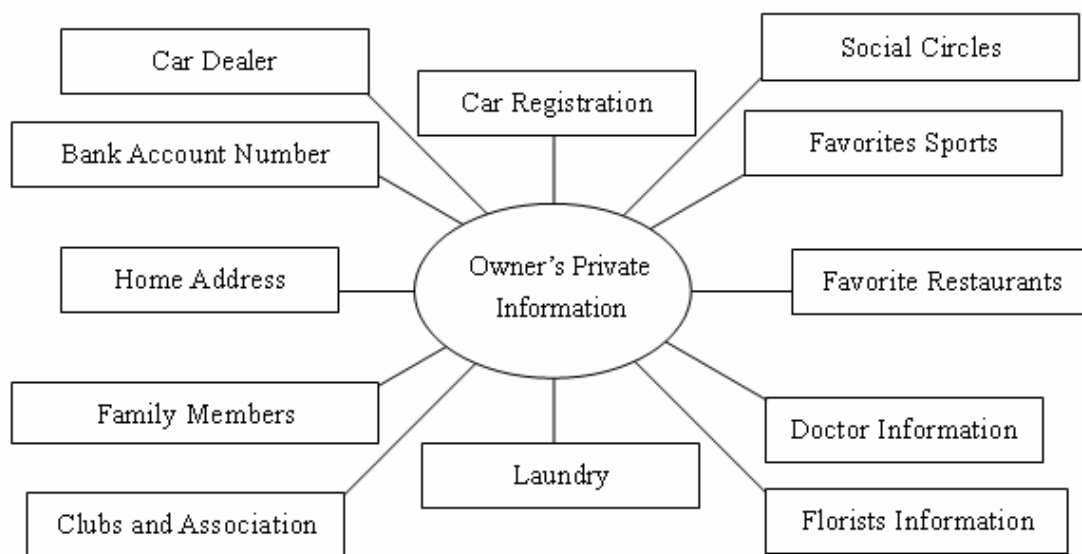


Figure 5: Owner's private information

Besides footballs, the owner is also an avid fan in golf; he owns a golf club membership and kept golf equipment shops contacts information in his Blackberry. However, no information was found on his golf playing skills or his handicap. It is probable that his role within the organisation as the managing director requires him to play this sport in order to socialize with business partners and customers. From the address book of the Blackberry, it was also possible to identify the doctor and local surgery which is approximately 3 miles from the owner's house. From the list of restaurants stored on the Blackberry, an indication of the tastes and dietary preference of the owner can be inferred. Almost 70% of the restaurants found on his Blackberry are Chinese or Italian restaurants. In general, these restaurants are dispersed into three main areas which are roughly 6 miles, 12 miles and 25 miles respectively from his home.

The social circle pattern of the owner can also be inferred from the email correspondences history and address book information. These people home addresses, phone numbers and interaction with the owner are all stored in the Blackberry. It was observed from the deleted email correspondence that was recovered that the owner has a very close relationship with one of his colleagues; they used to message each other when the owner was travelling aboard for business. There is no intend implication that any romantic relationship existed between these two people except for a close bond of friendship.

### Interrelated Information

There was some information that was recovered from the Blackberry that could not easily be defined as either totally business related or personal. The information that fell into this grey area was that of the hotel and airlines choices of the owner. From the Blackberry, it was possible to recover a list of hotels and airlines contacts for which there are three possible explanations. First, these options are his corporate preferred hotels and airlines selection. Second, these options are his personal preferences. Finally, these options are a combination of his personal preferences and the option given by his organisation. Nevertheless, we can safely infer that there are high probabilities that the owner flies with one of the identified airlines or stays in one of the hotels when he was aboard either on business or holiday.

## CONCLUSION

With the introduction of mobile devices into the corporate network, the structure of traditional perimeter protection defensive measures implemented in the corporate networks has changed. Mobile devices including laptops, PDAs, Blackberries, smart mobile phones and I-pods which, if they are not securely protected, can be a good target for any individual who are keen in gathering information from a particular corporation or individual. At the time of the writing, security products and services available for PDAs, Blackberries, smart mobile phones and I-pods are limited when compared with the services offers for laptop computers. Furthermore, the awareness of the need for protection of the information contained in small scale mobile devices is also poor. Awareness should be created in the corporate environment in order to protect both the business and individual privacy.

## **REFERENCE**

- Bluefire Security Technologies, (2007) Bluefire Mobile Security® Enterprise Edition, URL [http://www.bluefiresecurity.com/\\_assets/pdf/Bluefire\\_Products\\_Mobile-Security\\_Ent-Ed.pdf](http://www.bluefiresecurity.com/_assets/pdf/Bluefire_Products_Mobile-Security_Ent-Ed.pdf) Accessed 15 September 2007
- BT Research (On Going) BT Research into Residual Data On Mobile Devices, To be published on Jan 2008
- Chris Williams., (2007) Mobile Forensics Turns Up Heat On Suspects, URL [http://www.theregister.co.uk/2007/02/11/mobile\\_forensics\\_guidance/](http://www.theregister.co.uk/2007/02/11/mobile_forensics_guidance/) Accessed 25 August 2007
- GSM Security, (2006) GSM Security FAQ, URL <http://www.gsm-security.net/gsm-security-faq.shtml> Accessed 8 September 2007
- Lee Fueng, Yap Andy, Jones, (2007) Deleted Mobile Device's Evidences Recovery: A Review, International Conference Media & Information Warfare
- Pointsec, (2006) Pointsec® for Smartphone, Pointec URL, [http://www.filtermax.hu/data/files/pointsec/Pointsec4Smartphone\\_Eng.pdf](http://www.filtermax.hu/data/files/pointsec/Pointsec4Smartphone_Eng.pdf) Accessed on 25 September 2007
- TCG, (2007) TCG Mobile Trusted Module Specification, Specification version 1.0 Revision 1, Trusted Computing Group
- TrustDigital (2007) Making Smartphone Security & Management Easy Smartphone Security Version 7, URL, [http://www.trustedigital.com/downloads/SmartphoneSecurityv7\\_20507.pdf](http://www.trustedigital.com/downloads/SmartphoneSecurityv7_20507.pdf) Accessed on 25 September 2007
- Tuong Huy Nguyen, Annette Zimmermann, (2007) Forecast: Mobile Devices by Form Factor Worldwide 2004-2010, Gartner

## **COPYRIGHT**

Lee Fueng Yap, Andy Jones ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **Forensic Analysis Avoidance Techniques of Malware**

Murray Brand

School of Computer and Information Science, Edith Cowan University,  
Bradford Street, Mt Lawley, Western Australia 6050  
mbrand0@student.ecu.edu.au

### **Abstract**

*Anti-forensic techniques are increasingly being used by malware writers to avoid detection and analysis of their malicious code. Penalties for writing malware could include termination of employment, fines or even, imprisonment. Malware writers are motivated not to get caught and are actively using subversive techniques to avoid forensic analysis. Techniques employed include obfuscation, anti-disassembly, encrypted and compressed data, data destruction and anti-debugging. Automated detection and classification work is progressing in this field. This includes analysing statistical structures such as assembly instructions, system calls, system dependence graphs and classification through machine learning.*

### **Keywords**

Forensics, Anti-Forensics, Reverse Engineering, Analysis Avoidance, Malware Classification Analysis.

### **INTRODUCTION**

Forensic analysis of software that is determined to be malicious could result in the termination and/or prosecution of the author and/or the user of the code who knows its true malicious purpose. This could include logic bombs, viruses, worms, backdoors and trojans. There is no doubt that forensic analysis of software takes time and money. It is to the advantage of the malware author, or deliberate user, that the investigation takes longer than management is prepared to spend on the investigation. Conceivably, the malware author or user would like the malicious component to go completely undetected, and avoid the risk of prosecution.

Scenarios could include programmers writing salami attack style code to assist embezzlement through altering billing algorithms to redirect funds to their own accounts. Programmers could include a backdoor into software so that they can perform some malicious action at a time of their own choosing. Covert channel methods of communication could be implemented into code so that information can be passed out from an organization undetected. The scenarios are boundless.

Malware can incorporate various techniques to not only avoid forensic detection, but can also avoid forensic analysis. Grugq (n.d.) refers to this as “anti-forensics” and suggests that there are 3 fundamental ways of achieving this. Namely:

- Data Destruction
- Data Hiding
- Data Contraception

Essentially data destruction means deleting file residue such as inodes, directory entries and data blocks. It also includes deleting file system activity though inode time stamps. Data hiding means putting data where it is not expected to be. This can include storing data in blocks that have been marked as bad. Data contraception means making sure data is never stored on the disk. Grugq implies that it is better to never have created the data in the first place than having to destroy it.

Forensic analysis avoidance techniques can include, but are not limited to:

- Exploiting flaws in analysis tools.
- Attempting system destruction when being analysed.
- Generation of false disassembly listings.
- Subversion of disassembler heuristics.

This paper analyses these techniques to show how they can be recognised so that forensic analysis can be performed. This treatment is far from exhaustive, and serves merely as an introduction to the techniques employed to avoid forensic analysis of malicious software. The background section of this paper shows how reverse engineering techniques are typically used for the forensic analysis of malware binaries. This analysis can be either static or active. Static analysis is used to analyse the binary without executing it, whilst active analysis requires the execution of the binary and studying its behaviour. This paper then moves into how such reverse engineering analysis techniques can be avoided by the malware writer. The final section of this paper then looks at automated malware classification techniques that are currently being developed. This is because commercial virus scanners will be very unlikely to detect the presence of customized, or purpose built malware.

For the purpose of this paper, legal issues surrounding the reverse engineering of software is ignored. It is assumed that the software that is being examined is the intellectual property of the organization that will have authorised this activity, or that the investigative team is authorised to conduct the investigation in some other way.

## **BACKGROUND**

### **Evidence Elimination**

Penalties for hacking and writing malware are increasing. Awareness of hacking is rising within the community as well as forensic analysis techniques and methodologies. It is in the interest of hackers and malware writers to work against forensic investigators and eliminate forensic evidence. They have the motivation, the opportunity and now the methods to defeat forensic analysis. Kotadia (2006) reported in an online article that a speaker at the IT Security in Government Conference held in Canberra in July 2006 claimed that 65% of new malware “uses some type of stealth or anti-forensic technology in an attempt to remain undetected before, during and after an attack”.

### **Static Analysis**

Various techniques used for static analysis include “information gathering, disassembly, symbol table regeneration, decompilation techniques, and methodologies for determining the order of decompiling subroutines” (The Honeynet Project, 2004, p.452). These techniques are discussed in the following paragraphs.

#### **Information Gathering**

Information can be gathered from a binary using various tools. The “file” command can provide the file format and identify the target platform. Use of the “ldd” command can give information on dynamic link libraries required for the code to run. Embedded strings can be found in the binaries by using the “strings” command. The “hexdump” command with appropriate switches can do this too, as well as listing the name and version of the compiler used. Symbols can be listed by using the “nm” command. The language used to write the program can be determined by various characteristics found in the binary. This can include the way the strings in the binary are terminated, how data arrays are stored, how subroutines are called and the order of storing parameters on the stack.

#### **Disassembly**

The process of disassembly is used to create a more readable version of the binary. Typical programs in the Linux environment include ndisasm and objdump. Peikari and Chuvakin (2004, p.51) point out that objdump provides a sequential disassembly, and “no attempt is made to reconstruct the flow of the target”. It also cannot cope with binaries that have had their section headers removed, or that are invalid. This can be done by the tool “sstrip” which is discussed later. The Honeynet project (2004, p. 456) also point out that compiled programs usually put the data in the data segment, and the code in the code segment, but it is possible that data can be treated like code and code may be treated as data. Heuristics are used by disassemblers to separate code from data, and more professional disassemblers such as IDA Pro can trace program flow by using signature based heuristics.



## Symbol Table Regeneration

The “nm” command can be used in the linux environment to list symbols in object files. This is useful because the addresses of functions in programs can then be located and then analysed. More than likely, the malicious software writer will have used the “strip” command to remove all symbols from the object file. Programs can be either statically or dynamically linked. If dynamically linked, external library calls can be identified, however, if statically linked, all external routines are combined into the binary when compiled. This makes it impossible to distinguish external routines from linked libraries from the routines written by the programmer. The Honeynet Project (2004, p.458) say that “An ideal solution would be to recreate a symbol table containing the mapping of library names to addresses so that only the user code needs to be analysed”. They go on to say that a database of signatures could be created from subroutines from all known libraries. Then, to recreate the symbol table and insert it back into the program, the signatures could be matched against the code being examined. Then, external routines can be factored out, leaving only the code written by the malware writer to be examined.

## Decompilation Techniques

Decompilation is the action of transforming the results of a disassembly back into a high level language. The process works basically by understanding how particular compilers work in the first place to generate the object code, with particular consideration of how optimisations can take place. After disassembly, the entry points to all functions can be determined. Usually this is done by searching the disassembly for calls to absolute addresses. Subroutines can also be identified by common header and footer signatures. Analysis of each subroutine can then begin. This can consist of recognising flow control structures such as loops, conditional evaluation and branches. Parameters passed to and from these routines can also be determined, as well as other statements and assignments.

## Active Analysis

Active analysis techniques listed by the Honeynet Project (2004, p. 464) include sandboxing the analysis environment, blackbox analysis and tracing. These techniques are discussed below.

### Sandboxing the Analysis Environment

The Honeynet Project (2004, p.464) emphasises that “It would be foolish to execute the program on a production system or one connected to a live network”. They go on to recommend that the ideal place to perform active analysis would be on the honeynet on which it was captured from. It would be expected that local and remote logging would be in place already, and network packet capture is available. The analyst should have super user privileges so that kernel modules can be loaded and unloaded, system logs can be examined, file system images saved and restored and routing tables modified. They also recommend a wide variety of debuggers are available. This is because if different debuggers give different results, it would indicate that the malware is most likely targeting specific debuggers with avoidance techniques.

Other types of sandboxes include hard and soft virtual machines. A typical soft virtual machine is vmware which allows the investigator to take snapshots of the system state at any time and restore that particular state. This makes it very easy to repeat an analysis from a known point. Another advantage is that vmware allows the establishment of virtual networks, that can be confined from the host. This enables the investigator to observe the network behaviour of malware in a sandboxed environment.

### Blackbox Analysis

Blackbox testing in software engineering means testing software without knowing how the internals work. The focus is on state, by varying inputs and recording outputs. If a program is in an endless loop, it could indicate that the process is a daemon. Possibly the process is waiting for a network event such as a connection. It should also be possible to determine which files have been accessed by the process by examining file system states and access time stamps. It should also be possible to capture network packets using tools such as ethereal. It should also be possible to determine if any child processes were forked.

## Tracing

Tracing provides a more internal view than does blackbox analysis by tracing system calls, library calls and internal calls. Linux provides the “strace” program that is used. “strace works by intercepting and recording the system calls used by a process and the signals it receives. For each system call, the name, arguments and return value are printed to standard error (stderr) or to a file specified” Frye (2005). This is really useful, because it is possible to see the names and paths of files accessed and in which mode, system calls, shared libraries it needs, as well as many other parameters. It is possible too to attach to a running process

## DETECTION AND ANALYSIS AVOIDANCE

### Obfuscation

A paper by Christoderescu and Jha (2003) show how effectively obfuscation techniques such as dead-code insertion, code transposition and instruction substitution can defeat commercial virus scanners. One of the simple tests was to insert nop codes into the Chernobyl virus. They show that the Norton Anti-Virus software did not detect this obfuscated virus.

Two common techniques used by malware writers to avoid detection through obfuscation are polymorphism and metamorphism (Christoderescu, Jha, Seisha, Song, Bryant, n.d., p.1 ). A virus can use polymorphism to avoid detection by decrypting its encrypted malicious payload when it is being executed. “A polymorphic virus obfuscates its decryption loop using several transformations, such as nop-insertion, code transposition (changing the order of instructions and placing jump instructions to maintain the original semantics), and register reassignment (permuting the register allocation)” (Christoderescu et.al., n.d., p.1.). A metamorphic virus changes the code a number of different ways when it replicates. This can include code transposition, changing conditional jumps, register reassignment and substitution of equivalent instruction sequences.

These techniques need not remain in the domain of self replicating viruses. These techniques can be used in any malicious software to hide its presence through obfuscation. Most worms and viruses are variations of an original virus or worm and signatures are soon determined once they have been detected in the wild. There is no doubt that these techniques can be used by the hacker to create a single version instance of malware customised to perform some malicious act within the company in which they are employed as a software or network engineer. They may use these techniques to deliberately obfuscate its function. The 2003 paper by Christoderescu et.al. points out that most malware detectors are easily defeated by obfuscation because they use pattern matching and that they are not resilient to slight variations. They go on to say that pattern matching ignores the semantics of instructions.

By being able to run a semantics focused parser over suspicious binaries under forensic examination, it could be possible to determine which binaries have a malicious intent. It could help the forensics investigator to filter out binaries of interest faster. The sections that follow discuss techniques used to avoid detection and analysis. Understanding the basic mechanics of these techniques will not only help the forensic analyst detect these techniques but should also assist in developing automated malware classification programs.

### Anti-disassembly

By knowing that disassemblers use signature based heuristics, malware can be written to trick a particular disassembler into producing an incorrect disassembly for the binary.

Library calls can be made to be hard to identify by using static linking. The disassembler has to be able to match library function signatures.

A paper written by Linn and Debray (n.d.) describes algorithms that can be used to improve resistance of programs to static disassembly through obfuscation. Their focus is on providing security to protect intellectual property, stop piracy, and identification of breaches through making the disassembling process difficult. The same principles can be used by the malware writer equally well to hide their own intentions. Linn et.al.’s goal is to make the cost of reconstructing the high level structure too high, which is exactly the motivation of the malware writer as well. They say “In order to thwart a disassembler, we have to somehow confuse, as much as possible, its notion of where the instruction boundaries in a program lie” (Linn et.al., n.d., p.3). Methods discussed in their paper include introducing disassembly errors that “repair themselves” and injection of “junk

bytes” into the instruction stream. These techniques are reliant upon the type of instruction set of the processor, and on the type of sweep of the disassembler. That is whether the disassembler performs a linear sweep or a recursive traversal. A weakness of a linear sweep algorithm is that it is subject to errors if data is embedded in the instruction stream. Such a disassembler is aware of disassembly errors only if an invalid opcode is found. It does not allow for the control flow behaviour of the code. A recursive traversal algorithm does take into account the control flow behaviour of the code. However a “weakness is that its key assumption, that we can precisely identify the set of control flow successors of each control transfer operation in the program, may not always hold in the case of indirect jumps” (Linn et.al. n.d., p.3). Another technique they use for inserting junk bytes is what they refer to as jump table spoofing. They do this to mislead recursive traversal disassembly. Fundamentally, the idea is to make the code addresses in the jump table to addresses in the text segment, which do not correspond to instructions and hence create disassembly errors.

## **Encrypted Data**

### **Packers**

Miras and Steele (2005, p.20) explain that packers compress executable programs in the objective of making the task of reverse engineering as difficult as possible. A packed program contains a stub that is used for decompressing the compressed data. The executable is decompressed during loading. Popular packers used in the Windows environment that use PE format files include PECompact, UPX, ASPack and Armadillo. Packers use various techniques to subvert forensic analysis. These include:

- Built in anti-debugging.
- Insertion of junk code.
- Abuse of exception coding.
- Trick disassemblers by jumping into longer instructions.

Packer detection methods include examining signature and heuristics. An effective signature based method is to compare the program entry point against a database. Heuristical methods include seeing how the byte distribution (entropy) is changed by the packers as well as checking import tables.

### **Burneye**

Burneye “is an executable and linking format (ELF) encryption tool that limits forensics tools’ reverse-engineering capabilities by protecting the binary program. Burneye users can manipulate executable code so only an attacker with a password can run the program” (Saita, 2003). A loadable kernel module called burndump can be used to remove burneye from binaries. Another tool that detects the use of burneye is called bindview. Burncrack is a cracker that works with the John the Ripper program that can crack and unwrap burneye protected binaries without having to run them. This would be a very useful tool when required to perform a forensic analysis of the malware.

## **Data Destruction**

The purpose behind data destruction is to leave nothing useful for a forensics investigator, effectively removing all trace of evidence. The Defiler’s Toolkit is a set of programs whose purpose is to prevent forensic analysis, specifically targeting the ext2fs filesystem, commonly found on linux systems. Necrofile is one of the programs on the Defiler’s Toolkit for this purpose. Ordinarily, when a file is deleted, the inode and directory entries, known as the metadata are left untouched. A forensic investigator will look at the metadata to see if the supposedly erased data can be recovered. Necrofile can remove this metadata making it extremely difficult for the investigator to recover files. Klismafile is another program in the toolkit that removes directory entries of filenames that have been deleted. Through the use of these programs, forensic evidence can be removed. It would not be inconceivable for malicious code to perform these two actions autonomously if it detected that forensic analysis was being performed.

## **Data Hiding**

The purpose of data hiding is to hide evidence from the forensic investigator, and is only successful if the investigator does not know where to look for the evidence.

In the past, knowing that tools such as The Coroner’s Toolkit (TCT) did not look at bad blocks on a disk drive that was using the Second Extended File System (ext2fs), an attacker could use the bad blocks inode to include good blocks, and hide data there. Ordinarily, the bad blocks inode only points to bad blocks, and these blocks

will not be used for files. It is advisable to make sure that TCT's more recent version (TASK) is used and that bad blocks on a disk are also investigated (Skoudis, 2003). There is no doubt that this is a bit dated, but the point should be clear that flaws can be found in the forensics tools, and most likely will continue to be found as tools are improved and developed.

### **Data Contraception**

Grugq (n.d.) says that the two core principles of data contraception are to prevent data from being written to disk, operating purely in memory and to use common tools rather than custom tools. The idea is to limit the value of any evidence that does touch the disk. Rootkits can operate in memory and "use ptrace() to attach to an existing process and inject code into it's address space. Additionally, injecting kernel modules directly into the kernel is also a well known technique" (Grugq, n.d.).

Grugq recommends using common utilities such as rexec, which remotely executes a command on a remote host. This allows the malware or hacker to never have to write anything to disk.

### **Antidebugging**

A common debugger used in the linux environment is gdb. If a program has been compiled with the option to produce debugging information, it could be possible to hide malicious intent, such as a logic bomb. The following code, ptrace.c demonstrates this.

```
#include <sys/ptrace.h>
#include <stdio.h>
int main()
{
    if (ptrace(PTRACE_TRACEME, 0, 1, 0) < 0) {
        printf("Debugging detected, goodbye!\n");
        return 1;
    }
    printf("Malicious purpose here.");
    return 0;
}
```

The program is compiled with the following command:

```
gcc -o ptrace ptrace.c -g
```

When run at the command line, the program will print the following line.

Malicious purpose here.

When run inside gdb, it will print

Debugging detected, goodbye!

Hopefully, this sort of thing will be picked up in a peer review, or some other audit activity, but could be missed by an inexperienced auditor, or an auditor not focussing on security. The code could even be in a library that is not in version control and is going to be linked in, in a nightly build. It is possible to hide the approach even more by replacing the call to ptrace with an int 80 system call with inline assembly (Peikari et.al., 2004, p.69).

The ptrace trick can be overcome by setting breakpoints at the start and the end of the subroutine that calls ptrace (The Honeynet Project, 2004, p.469). Then when running in the debugger, the return value from the call to ptrace is changed from failure to success.

### **Compression Bombs**

Compression bombs "are files that have been repeatedly compressed, typically dozens or hundreds of times" (NIST, 2006, p.45). They can cause forensic tools used for examination to fail, or use up so many resources that the computer may hang or crash. It is possible that they may contain a malicious payload. NIST (2006, p45)

suggest it could be very difficult to detect a compression bomb before it is uncompressed. However it is very important to be working on the forensic image, so that the system can be restored if required. It is also suggested by NIST (2006, p45) that a virus scanner may detect the bomb. But this is reliant upon signatures being kept up to date, and also that the bomb has been in the wild for a while, and that anti virus companies have analysed it before. More of a concern, would be a one off, customised compression bomb that would not be recognised by its signature and could be effected once malicious software determines that an investigation is taking place.

## **AUTOMATED MALWARE DETECTION AND CLASSIFICATION**

Malware detection and analysis by an investigator can be a labor intensive process using static and active techniques. Due to time constraints and the abilities of the investigator, there is a possibility that critical forensic evidence could be overlooked. To this end, automated malware detection and classification tools are being developed.

A presentation by Bilar (2005) shows how malware can be classified by analyzing statistical structures. Three perspectives are examined by Bilar, including Assembly Instructions, Win 32 API Calls and System Dependence Graphs. Examination of Assembly Instructions is primarily a static analysis technique where the frequency distribution of opcodes are developed from the disassembly of the binary. Bilar shows that this technique can be useful to provide a quick identification. Just looking at the most frequent opcodes is a weak predictor. Looking at fourteen of the most infrequently used opcodes such as int and nop it may be possible to classify malware. Bilar suggests that root kits make heavy use of software interrupts, and viruses make use of nop for padding sleds. Additional work being carried out in this area includes investigating equivalent opcode substitution effects, and association effects between compilers and types of opcodes.

Tracking Win 32 API Calls is an active analysis technique that observes the API calls that a program under investigation makes. These calls are recorded and a count vector is saved into a database. These vectors are then compared to known malware vectors in the database, and it is determined if the vectors are related. Bilar (p.25) claims that this vector classification is successful in classification of malware into a family. The Win 32 API call fingerprint is shown by Bilar (p.27) to be robust, even though various packers were used.

System Dependence Graphs is a newly developing static analysis technique described by Bilar (p.31) that represents control, call and data dependencies of a program through graph modeling. Then graph structures can be used as fingerprints, which assist in the process of identification, classification and prediction of behaviour.

Lee and Mody “propose an automated classification method based on runtime behavioral data and machine learning” (2006, p.3). Essentially the run time behaviour of a file is represented by a sequence of events which is stored in a canonical format in a database. Machine learning is used to recognize patterns and similarities which are then used to classify new objects.

## **CONCLUSION:**

This paper has shown how forensic analysis can be avoided using techniques used by the writers of malicious software. It has been reported that malware is increasingly using these techniques to avoid forensic analysis, and that the majority of new malware is incorporating these techniques. A number of avoidance techniques were discussed including obfuscation, anti-disassembly, encrypted and compressed data, data destruction and anti-debugging. This list of techniques is far from exhaustive. These techniques need not remain in the domain of mass distributed malware such as worms and viruses. They could be developed primarily for a one off, malicious mission of performing some malicious act within a company, collection of commercially sensitive data or other equally criminal activities.

Automated detection and classification work is progressing in this field. This includes analysing statistical structures such as assembly instructions, system calls, system dependence graphs and classification through machine learning.

## **REFERENCES:**

Bilar, D., (2005). *Statistical Structures: Fingerprinting Malware for Classification and Analysis*. Retrieved September 2, 2006 From [www.blackhat.com/presentations/bh-usa-06/BH-US-06-Bilar.pdf](http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Bilar.pdf)

- Christoderescu, M., Jha, S., Seisha, S.A., Song, D., Bryant, R.E., (date unknown). *Semantics-Aware Malware Detection*. Retrieved August 30, 2006 From <http://www.cs.cmu.edu/~bryant/pubdir/oakland05.pdf#search=%22malware%20%2BObfuscation%22>
- Christoderescu, M., Jha, S., (2003). *Static Analysis of Executables to Detect Malicious Patterns*. Retrieved September 2, 2006 From [www.cs.cornell.edu/courses/cs711/2005fa/papers/cj-usenix03.pdf](http://www.cs.cornell.edu/courses/cs711/2005fa/papers/cj-usenix03.pdf)
- Frye, M., (August 2005). *Debugging codes with strace*. Retrieved August 26 2006 from [www.redhat.com/magazine/010aug05/features/strace/](http://www.redhat.com/magazine/010aug05/features/strace/)
- Grugq, (date unknown). *The Art of Defiling, Defeating Forensic Analysis on Unix File Systems*. Retrieved August 26, 2006 from <http://opensores.thebunker.net/pub/mirrors/blackhat/presentations/bh-asia-03/bh-asia-03-grugq/bh-asia-03-grugq.pdf>
- Kotadia, M., (28 July 2006). *Beware 'suicidal' malware, says CyberTrust*. Retrieved August 27, 2006 from <http://software.silicon.com/malware/0,3800003100,39160966,00.htm>
- Linn, C., Debray, S., (date unknown). *Obfuscation of Executable Code to Improve Resistance to Static Disassembly*. Retrieved August 30, 2006 From [www.cs.arizona.edu/solar/papers/CCS2003.pdf](http://www.cs.arizona.edu/solar/papers/CCS2003.pdf)
- Miras, L., Steele, K., (September 2005). *Static Malware Detection*. Retrieved September 2, 2006 From <http://www.toorcon.org/2005/slides/lmirasksteele-staticmalwaredetection.pdf#search=%22malware%20%2Bpacker%22>
- NIST, (August 2006). *Guide to Integrating Forensic Techniques into Incident Response*. Retrieved September 1, 2006 From <http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf>
- Phrack Inc, (date unknown). *Defeating Forensic Analysis on Unix*. Retrieved August 27 2006 from <http://www.theparticle.com/files/txt/hacking/phrack/p59-0x06.txt>
- Saita, A., (May 2003). *Antiforensics: The Looming Arms Race*. Retrieved August 26, 2006 from <http://infosecuritymag.techtarget.com/2003/may/antiforensics.shtml>
- Skoudis, E., (June 6 2003). *Breaking News – The Latest Computer Attacks and Defences*. Retrieved August 27 2006 from [www.counterhack.net/UFL.ppt](http://www.counterhack.net/UFL.ppt)
- The HoneyNet Project, (2004). *Know Your Enemy Learning About Security Threats*. Addison-Wesley, Boston

## **COPYRIGHT**

[Murray Brand] ©2007. The author/s assign SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors

## ID Theft: A Computer Forensics' Investigation Framework

Olga Angelopoulou  
University of Glamorgan  
oangelop@glam.ac.uk

### Abstract

*The exposure of online identities grows rapidly nowadays and so does the threat of having even more impersonated identities. Internet users provide their private information on multiple web-based agents for a number of reasons, online shopping, memberships, social networking, and many others. However, the number of ID Theft victims grows as well, resulting to the growth of the number of incidents that require computer forensics investigation in order to resolve this type of crime. For this reason, it appears of value to provide a systematic approach for the computer forensics investigators aiming to resolve such type of computer based ID Theft incidents. The issues that demand individual examinations of this type of crime are discussed and the plan of an ID Theft computer forensics investigation framework is presented.*

### Keywords

ID theft, incident investigation, digital evidence, computer forensics, computer crime, computer forensic investigator

### INTRODUCTION

According to the Credit Industry Fraud Avoidance System (CIFAS) (2007), the UK's Fraud Prevention Service, in 2006 alone 80000 cases of ID Theft were recorded, comparing to 9000 cases in 1999. It appears as the wide use and the anonymity occurring on the Internet has influenced a number of people proceeding to Internet related, non-legitimate actions.

This is a global problem. ID Theft is considered as a standalone crime since 1998 in the United States as defined in the Identity Theft and Assumption Deterrence Act (1998) and belongs to federal crimes, where the establishment of the Offence is made as follows:

*"knowingly transfers or uses, without lawful authority, a means of identification of another person with the intent to commit, or to aid or abet, any unlawful activity that constitutes a violation of federal law, or that constitutes a felony under any applicable state or local law."*

In the H. R. 2622, Fair and Accurate Credit Transactions Act of 2003, the American Identity Theft legislation provides the state approach of combating ID Theft and protecting the consumers. Based on the U.S. Federal Trade Commission's (FTC) report for National and State Trends in Fraud and Identity Theft 2004, of the 635,173 complaints received, 246,570 were ID Theft reports. The most common form of reported ID Theft was Credit Card fraud, followed by phone or utilities fraud, bank and employment fraud. It is very important to note that only 30% of victims notified a police department. It can therefore be assumed that the majority of people are either not aware that they could have contacted law enforcement agencies or prefer not to make their ID theft incident known.

In plain words, the intention and plan of the person who decides to steal someone's identity, the ID Thief, is to collect the more personal details the possible for the person he's interested in, attempt to use this information for the largest personal gain of his and finally continue his life under someone else's name. It seems that the popular saying 'there is no perfect crime' is not taken seriously from some individuals. However, the world's history has proved that no matter the precautions and strategies followed to combat a crime regardless its nature, the fraudsters will discover a way to conduct it. Under no circumstances should people give up the effort of eliminating a type of crime, nevertheless there should be also invented radical and innovative ways of discovering and uncovering evidence of those already taken place.

Based on the above unfolds the rationale of this paper presenting a piece of under development work. The first aim is to provide some insight of the basic terms, ID Theft and Digital Evidence and their sequence in order to lead to the successful computer forensics investigation. The issues concerning the importance of such an

individual approach of ID Theft incidents towards Computer Crime are discussed; the design of the proposed ID Theft Investigation Framework is presented and defended by an example.

## ID THEFT AND THE DIGITAL EVIDENCE

The fraudulent use of another's personal details has become an increasingly significant concern. One out of ten people in Britain was a victim of online fraud during 2006 revealed a survey, corresponding to 3.5 million British internet users (unknown, 2007). Attacks on financial institutions have risen from 39% in 2003 to 83% for 2004 (McKenna, 2004).

The Home Office (2006) defines Identity Theft as:

*“Criminals can find out your personal details and use them to open bank accounts and get credit cards, loans, state benefits and documents such as passports and driving licenses in your name.”*

Identity theft (ID Theft) can be perpetrated in a number of ways. Discarded documents containing personal details can provide a rich source of personal information. Simple forms of deception can also be used to extract the information from the victim an example would be an attacker poses as a legitimate government official or business person collecting personal data door to door. Other methods include the so called ‘brute force’ techniques such as the stealing of wallets and purses containing identification and credit and bank cards or the removal of personal documents during a burglary. In particular stolen mail, where the perpetrator may have access to bank and credit card statements, pre-approved credit offers, checks and tax information, can be used to gather information for an ID Theft. This may be followed up by social engineering. The perpetrator contacts the person who has lost his card claiming that they found it, asks for personal details and then uses this information fraudulently (Dwan, 2004).

However, personal identity is increasingly being stored and used in a range of digital forms. This can leave individuals exposed to possible threats as a result. Examples include; Phishing e-mails, web spoofing and numerous other techniques. This emerging and developing trend in crime can result in complex investigations that involve information technology, both as a medium for analysis and as evidence at the same time. Fraudsters are obtaining more sophisticated technological ways and manage to conceal their crimes.

The following table summarises all different methods by which ID Theft is performed, separated in offline and online:

Offline Techniques	Online Techniques
Stolen wallets or bags	Phishing
Stolen mail	Pharming
Deceased people	Web-Spoofing
Dumpster diving	Social Engineering
Burglars	Card Cloning
Shoulder surfing	Storage Devices and Media
Social Engineering	Biometrics
	Malicious Software
	Key-loggers
	CCTV Cameras
	Data Retrieval

*Table 1: Summary of offline and online ID Theft Techniques*

Digital evidence is any kind of digitally processed information that is stored in any sort of digital media. The data strengthens or neglects the assumption of an electronic crime in the terms of the investigation process. It can be therefore presented as supportive proof in a court of law. (Carrier B., 2006)

In the late 20<sup>th</sup> century Dr.Edmund Locard, director of Lyons Institute of Forensic Medicine, defined an important theorem for the foundation of the forensic science that is widely known as the *Locard Exchange Principle*:

*“Any action of an individual, and obviously, the violent action constituting a crime, cannot occur without leaving a mark. What is admirable is the variety of these marks. Sometimes they will be*



*prints, sometimes simple traces, and sometimes stains*” (Chisum W., J., and Turvey B.E. (2006) from Locard, 1934).

The theorem has been transformed and misinterpreted during the years aiming to cover the science needs (Chisum, Tervey, 2006). The simplest form that can be found in literature is “*with contact between two items, there will be an exchange*” (Thornton, 1997). Casey (2003) has noted that this fact holds true in the digital world as a digital exchange between two devices results in an exchange of information. For example a request to view a web page from a client may be logged on the server and the web page, if downloaded, may then reside temporarily on the client.

As stated from Marcella and Greenfield (2002), computer forensics demands accurate evidence and results of the investigation. For this reason, the use of state of the art equipment and methods should be used in order to reassure it. The world of Computer Forensics is dealing with a number of situations from industrial espionage to damage assessment and holds back to the beginning of the 1980s. Nevertheless, the last few years have made it widely known to the public and demand even more expertise. It is by nature a science that requires detail by all means and there is where the handling of the digital evidence should be based.

## **THE “SOLITARY” OF ID THEFT TOWARDS COMPUTER CRIME INCIDENTS**

Initially, it is worth mentioning some issues concerning computer crime in general. Those types of crime where a computer or any other electronic device is involved in order to perform the crime or as the target of it are considered as computer crimes (Postnote Computer Crime, 2006). The criminals become more and more sophisticated nowadays and attempt to use technology by any means in order to avoid detection and perform the crime in greater detail and excess deception. A simple glance on news articles enhances the anxiety to information security people and the need to eliminate the problem. However, this could only happen on a virtual world as the use of computers and online transactions becomes only wider, giving the fraudsters’ the chance to increase their ways of attacking systems.

Computer crime involves different types of offences as hacking, copyright, child pornography, fraud, viruses, and harassment. They can be categorised in different ways, according to the methods used in order to prevent them. Icove et al. (1995) in Computer Crime classify them with this approach, grouping them in:

- Physical security breaches
- Personnel security breaches
- Communications and data security breaches
- Operations security breaches

Each security breach involves several fraudster actions that lead to computer crime. Hence, computer crime as a general matter can be treated based on the facts and the incidents that surround it. This basically requires treating computer crimes independently, in order to achieve a more analytical and in depth examination of a case. The investigation process time will be accelerated as the investigator will be able to follow specific steps once the type of the crime is revealed and he will be able to track on a certain process.

Concerning the current research, the digital investigation of computer based ID Theft is a computer crime that requires the expertise of a computer forensic investigator in order to be resolved. The digital evidence that comes into sight after the analysis of a related to crime computer misuse is of critical value as it should be efficient to accuse someone with a crime or not. Therefore, the manipulation of the evidential data should be treated sensibly and with sensitivity.

As described in a number of existing published sources, ID Theft is besides considered as a major threat for individuals and corporations and consists of multiple types of crime. It involves multiple ways of achieving it, either by the aid of technology or not. This is the major difference from other types of computer crime and the way the digital evidence should be treated.

In view of this influence, for the function of science, it could be considered that all technology aided evidential elements will be represented with the term ‘online’, whereas all non-technology aided will be called ‘offline’. Consequently, the investigator has to take into consideration the volume of the offline sources that influence the outcome of the investigation, as a number of offline techniques could have been used to commit the crime. A characteristic example of this issue could be the offence of hacking. In such a case, the actual evidence will be hidden inside the suspect’s computer, as the hacker’s only weapon is that; and the assigned to the incident investigator will have to trace all evidential data from there. At the same time, in a computer-based ID Theft incident the investigator’s findings depend on the fraudster’s computer, in addition other sources, such as a card cloning machine and forged documents could enhance the evidence. However, this is not the purpose of the

computer forensics investigator, but still differentiates this type of crime from any other computer related and raises the need of treating this type of computer crime in an individual manner.

Based on FTC's Identity Theft Data Clearinghouse (2007), ID Theft was established as the top complaint category in consumer fraud with 36 percent. Therefore, in order to support the need of treating a computer crime as an independent entity, an example of an ID Theft case for financial purposes can be considered, where the investigator can first focus on credit history, transactions made on the victim's name, applications for bank accounts, loans and credit cards. This evidence trail is to be recovered in the form of data, logs etc. formats through various systems within one or even multiple financial organisations. As a result, the investigation is complicated and time-consuming. With identity-related ID Theft cases, the investigator will need to consider not only the financial evidence but the personal information gained, subsequent actions triggered by a hijacked identity etc.

The difficulty the investigators need to face when dealing with an ID Theft incident and what really makes this type of crime individually-treated is that they actually have to face two investigative categories; victim or perpetrator. This is where all starts, as the need to distinguish and separate the investigation process is going to differentiate such a detailed process from others. A victim's machine should provide such evidential data that will be able to prove the fraud against the computer user, while on the perpetrator's machine the evidential data should be treated in such a way that will reveal the deception's proof. One might argue that the existing computer forensic investigation frameworks can cover this argument. However, a generic guideline cannot reach to a far detailed phase of the investigative process as it aims to cover all different types of computer related crimes. In respect to the existing computer forensic frameworks and based on the substantial increment of ID Theft the need to aid the computer forensic investigation of this type of crime leads to the point that the investigation process needs to be focused on a different perspective each time as different sort of evidence is required.

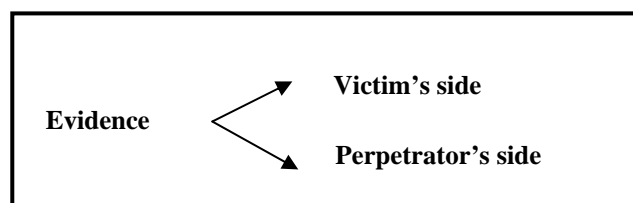


Figure 1: The different aspects of evidence concerning ID Theft incidents

People who work at this field need to be able to use constructive methods in order to facilitate their actual aim that is to provide evidential data after a computer forensic investigation. The threat of becoming an ID Theft victim becomes even greater day by day for everyone, especially those who use the Internet by any mean, make transactions, socialize, interact, anything that someone could take part on through it. Simply because the use of Internet and the public dependency will only grow, there should be invented and developed efficient ways that could cope with this *rapidly spreading threat*.

The following sections are going to support the above arguments on a practical approach, describing the theoretical procedure of designing and implementing such a computer forensics investigation framework.

## DESIGN DESCRIPTION

There is the need at this stage to describe 'why' and 'how' the foundation of this work is set. For this reason, the following paragraphs are going to set the principles of this work.

The fact is that in order to accomplish a comprehensive and structured investigation about a computer forensic case, the steps followed should be of extreme diligence. The procedure that is followed should give an answer to the question of what information might be stolen and how this information could be stolen. The major aim is to collect the data that give evidence and can prove a possible attack. Only after a detailed and constructed approach the data analysis can return and verify the only premise that might appear in the beginning of the research; that the investigator has to deal with an ID Theft case.

Fundamentally, a framework is considered as **tool to aid in planning, monitoring and evaluation of research projects** (Carrier, 2006). The general investigation scientific method presented by Carrier and Spafford at the DFRWS 2006 though, structures a checklist of high-level phases on a theoretical foundation in order to propose and describe a procedure in a digital investigation research field. The phases have been applied on existing frameworks and demonstrate an accurate approach to the specific area of research. These include:

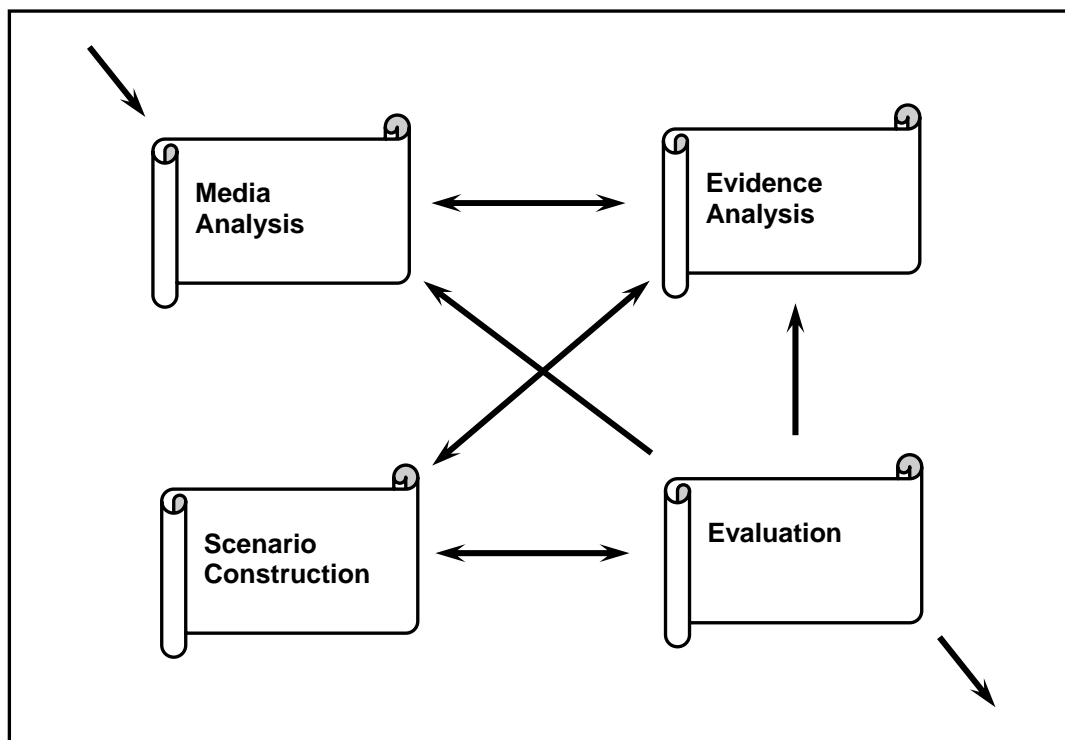
- Observation, where the researcher needs to observe the field in order to create a clear picture about the processes and the activities that take place on the investigation.
- Hypothesis Formulation, where the researcher focuses on the results of the observation phase and is able to categorise the techniques that will aid the analysis of the findings.
- Prediction, this phase will support the Hypothesis Formulation as the results of this part will prove whether or not the Hypothesis is formed on a constructed basis and will lead the researcher to the last phase.
- Testing and Searching, where the tests and experiments that take place on a generally approved manner will probably result to new predictions and evaluate the Hypothesis the researcher has set.

The procedure described should give answers to questions for the existence of an **x** file to a **y** event and should be responded accurately only after a successful analysis of the data. The investigator should be able to have access and examine each one of them, every file and any event that influences the file's behaviour.

### **ID THEFT INVESTIGATION FRAMEWORK DESIGN**

For the ID Theft Investigation Framework the research of the existing literature has revealed that the most suitable approach would be first to identify and define the phases that will lead the researcher to the appropriate procedures, based on the model described above and consequently the implementation of a conceptual framework that will aid the forensic investigator. The idea is based on the concept of handing over the procedure on a fundamental basis. This way the process will correspond to any possible procedure during the investigation. The presented process is adjusted to the needs that an ID Theft incident investigation requires as it is going to be analysed on the following chapter. The phases at the first level of the process follow a generic pattern. However, in the advance of the process the model will reveal the second level phases that contain the processes of the framework and the third level phases, including the activities that take place. Every Phase is influenced by the inputs and the outputs (I/Os) that design it.

Therefore, the study should be distinguished in four phases, where every phase represents a major procedure throughout the ID Theft investigation lifecycle. The impact on every phase is featured from all influential actions taken place on each phase. This states the first level of the framework's formulation.



*Graph 1: First Level Investigation Process Phases*

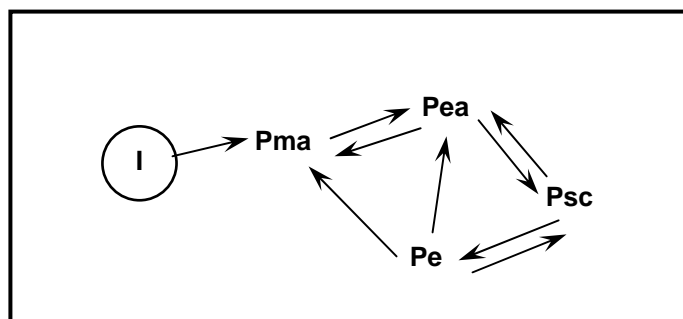
According to the procedure that is proposed to guide the researcher in order to formulate the investigation process, the above graph outlines the first level of it. There is always one start and one end point concerning the

investigation. The forensic analyst needs to begin under a concrete method and finish so. Hence, the Media Analysis (Observation) is the phase that appears of extreme importance in order to provide the data that will prolong to the Evidence Analysis (Hypothesis Formulation). The findings of the disk analysis will move forwards to the analysis and the decision whether this evidence can stand as accurate to the Scenario Construction (Prediction), where it is going to validate or not the Evidence Analysis. However, the analyst should always be able to return from the Evidence Analysis to the Media Analysis for any further data that might appear of value during the examination of the media, as well as he should always be able to revisit the Evidence Analysis at any stage in the Scenario Construction for any information that could emerge and indicate further investigation. Then, the Scenario Construction will need to be evaluated in order to prove its validation that is going to direct to the end of the process. However, the Evaluation phase requires the possibility to recall any of the previous phases of the investigation process in order to prove the objectivity of the research outcome. The following table sets the Variable Names at this Phase of the process in order to be used during this level of the procedure.

First Level Investigation Process Variable Names	
Description	Variable
Incident Investigation	I
Media Analysis	Pma
Evidence Analysis	Pea
Scenario Construction	Psc
Evaluation	Pe

Table 2: First Level Investigation Process Variable Names

A graphical representation demonstrates the process that is proposed to be followed from the ID Theft Investigation Framework concerning an Incident.

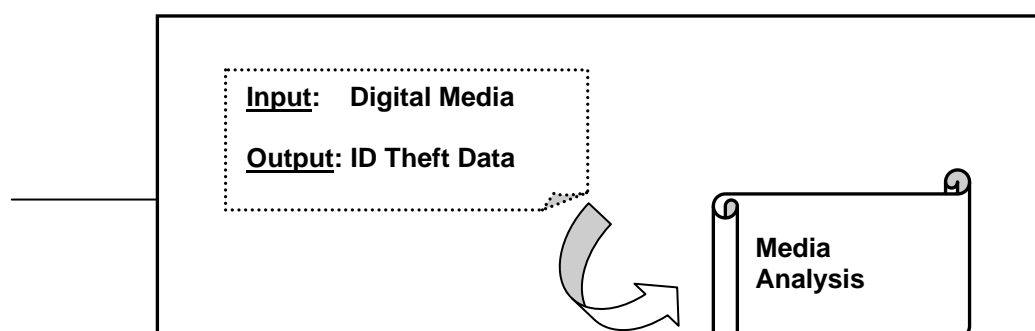


Graph 2: Incident Investigation Process Lifecycle

Every phase at this level of the investigation process needs to respond to an input and an output (I/O) practice. This provides the necessity of defining the processes and the activities that will be set for the further analysis of this research. The I/Os support the general process in the terms of continuity during the investigation's lifecycle. Below, there is a graphical representation of every phase in correspondence with the First Level I/Os that manipulate it. Every phase of the process requires as an input the output of the preceding phase for supporting the coherence of the research outcome.

### Media Analysis

The Pma requires as an input any type of Digital storage Media that could give as an output possible ID Theft data in order for the further investigation to take place. At this point the term Digital Media is going to represent any type of computer storage device.



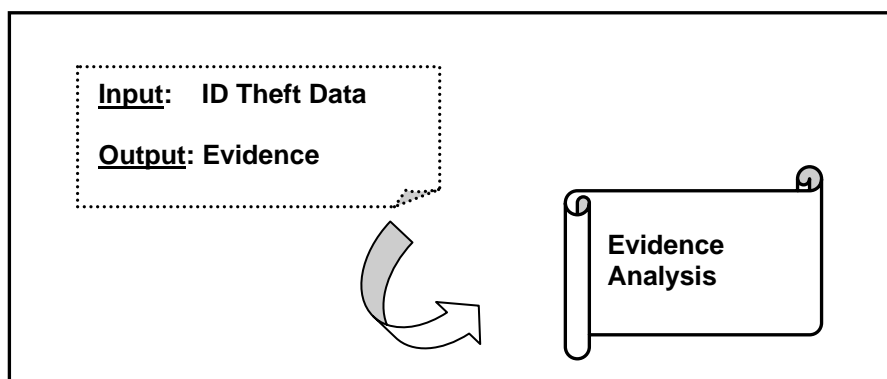


Possible Digital storage media	
Computer / Laptop / Server Hard Disk	PCMCIA cards
External Hard Disk	Memory Cards
Mobile Phone / SIM Card	USB Memory Stick
Raid Hard Disks	PDA
Tape Back-ups	Floppy Disk
CD / DVD	

Table 3: Possible Digital storage Media

### Evidence Analysis

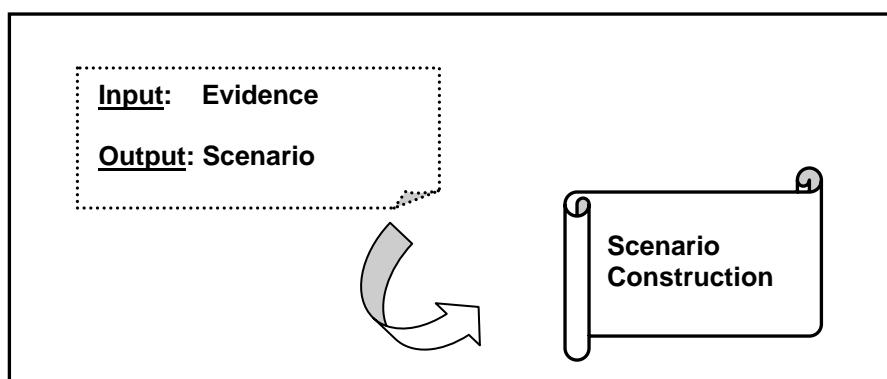
The Pea takes as input the possible ID Theft data provided by the Pma and will try to convert it to Evidence. At any time during an investigation further data may significantly come into view and the analyst will return to the previous phase for the analysis of the Digital Media.



Graph 4: Phase 2 - Evidence Analysis I/Os

### Scenario Construction

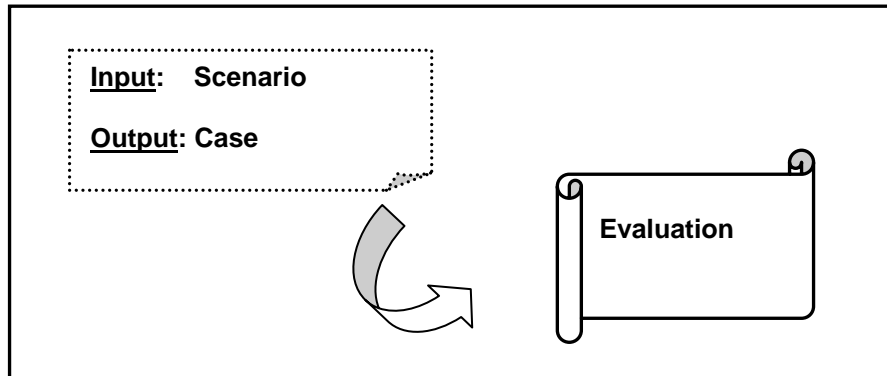
The input of the Psc should be anything else but the Evidence from the Pea aiming to produce a Scenario as an output. The scenario is the 'story' that is created by the investigator based on the findings of the media examination. Only when the required information is gathered, the analyst is able to present a coherent and efficient chronicle of the evidence. The reconstructed story of what has taken place to the original media. At this Phase it may be required for the analyst to search for further details on the two previous phases.



Graph 5: Phase 3 - Scenario Construction I/O

## Evaluation

The Pe uses the Scenario from the Psc as an input in order to present the Case that is also the required output of the whole investigation. At this stage the Media Analysis will be either proved or disproved. However, the analyst needs to be able to revisit all previous Phases for the validation of the research outcome.



Graph 6: Phase 4 - Evaluation I/O

On a theoretical basis the Investigation Process obtains the following instruction format that is going to be analysed in detail on the analysis chapter. Each phase, the processes (I/Os) and the activities of the first level of the investigation framework are list numbered in order to create a logical continuation. By progressing the activities at this level of the procedure appear, where the separation of the framework in two parts comes into sight, the victim's and the perpetrator's that are going to lead to the second level of the Investigation Process.

## First Level Investigative Process

### Phase 1. Media Analysis

- Process 1.1. Digital Media
  - Activity 1.1.1. Source Identification
  - Activity 1.1.2. Digital Media collection
  - Activity 1.1.3. Copy/ image the source
- Process 1.2. ID Theft Data
  - Activity 1.2.1. Evidential data identification
    - a. Victim
    - b. Fraudster
  - Activity 1.2.2. Target identification
  - Activity 1.2.3. Threat agent identification / intention

### Phase 2. Evidence Analysis

- Process 2.1. ID Theft Data
  - Activity 2.1.1. Data Analysis
  - Activity 2.1.2. Target Analysis
    - a. Victim
    - b. Fraudster
  - Activity 2.1.3. Threat Agent Analysis
- Process 2.2. Evidence
  - Activity 2.2.1. Evidence Collection
  - Activity 2.2.2. Evidence Classification

### Phase 3. Scenario Construction

- Process 3.1. Evidence
  - Activity 3.1.1. Structure of evidential data
  - Activity 3.1.2. Structure threat agent's profile
    - a. Victim
    - b. Fraudster
  - Activity 3.1.3. Structure analysed digital evidence
- Process 3.2. Scenario
  - Activity 3.2.1. Scenario Outline
  - Activity 3.2.2. Scenario Preparation Documentation

#### **Phase 4. Evaluation**

Process 4.1.	Scenario
Activity 4.1.1.	Scenario Testing / Evaluation
Activity 4.1.2.	Scenario Clarification
Process 4.2.	Case
Activity 4.2.1.	Case Construction
Activity 4.2.2.	Case Clarification
Activity 4.2.3.	Case Evaluation
Activity 4.2.4.	Evidential Case Representation

### **EXAMPLE FICTIONAL SCENARIO**

Assuming there is a computer hard disk delivered to a computer forensics lab from a major company suspecting an employee for computer misuse, but without any further details. Consequently, this is going to be the first attempt to apply the above described first level framework, in an incident that the analyst is not provided with any further information.

In such a case, and for Phase 1 of the investigation process, the investigator receives the input of the Process 1.1 that needs to collect (Activity 1.1.1), identify (Activity 1.1.2) and image (Activity 1.1.3) the hard disk. The output process of this Phase is the 1.2, where for the needs of this example is ID Theft Data or even data that could stand as ID Theft evidence and this is what the investigator is challenged to discover. The Activity 1.2.1 reveals the first element, whether the evidence belongs to a victim or a fraudster. From this point the investigation process on a lower level framework should progress under a bipolar perspective according to the data provided from 1.2.1. However, for this example it can easily give the first glance charging the employee as a fraudster committing ID Theft from his work computer, including every instance of it, inside the company or towards outside targets. His machine could also state him as a victim of ID Theft, meaning that the company has probably got information leak to the outside. There is when Activity 1.2.2 identifies the target, that could be a vulnerable system, or information published on the public domain, as well as the Activity 1.2.3 where the threat agent and his intentions can be identified.

Phase 2, aims to analyse the evidence from the original media on Process 2.1, the investigator analyses the ID Theft data under three Activities 2.1.1, 2.1.2 and 2.1.3, data, target and threat agent accordingly. The target analysis activity is divided to victim and fraudster, as the inputs for each category are different and guide the investigation towards different perspective. In case the employee has been a victim, the investigator will be able to analyse the target from this side, if the employee was the fraudster perhaps the investigator will be able to collect more details about the target. Therefore, the Process 2.2 provides the evidence with the Activities 2.2.1 that collects the evidence and 2.2.2, where the evidence is classified.

Phase 3 constructs the scenario of the incident. At this point, Process 3.1 is the evidence extracted from the investigation, where Activity 3.1.1 structures the evidential data, while 3.1.2 structure the threat agent's profile that is divided to the victim's and the fraudster's side, as there is going to be different sort of data gathered to give the investigator the information required to construct the attacker's profile. Activity 3.1.3 structures the analysed digital evidence that refers to the incident as a whole. In such a manner, on Process 3.2 the activities that follow are 3.2.1, the scenario outline and 3.2.2, the documentation preparation. At this point the investigator has a clear aspect about his suspect. He could tell with structured evidence whether the suspected employee has committed ID Theft or has only been another victim.

However, he owes to continue with Phase 4, the evaluation of the scenario. So, on Process 4.1 that is the scenario, Activity 4.1.1 is followed, evaluates or not the scenario assumption and 4.2.2 clarifies the scenario. Process 4.2 leaves the investigator with a case where he needs to construct it (Activity 4.2.1), clarify it (Activity 4.2.2) and evaluate it (Activity 4.2.3). The last Activity 4.2.4 for the investigator is the evidential case representation, the computer forensic report, where all evidence will be described and could also stand in a court of law, charging the fraudster of the case that could be either the victim or not.

### **CONCLUSION**

When someone who can describe his relation with computers and the Internet as professional or even as advanced user, reaches to the point that feels insecure and suspicious with the interaction with them, then it obviously appears that the situation requires some attention. The statistics prove that the ID Theft is a type of old fashioned crime that transformed into a Cybercrime because of the intense 'investment' of online sources. There may be more 'bad' people in the world than good ones and the spur of committing the perfect crime that will never be revealed still runs in some people's minds.



This situation leads to the improvement of tools and popularity of studying and research in computer forensics the last few years. It is the type of science that corresponds to the needs of digital investigations. Therefore for the conditions where ID Theft is combined with computer usage, computer forensics is the type of science that will be requested to provide the evidence. In such a perspective, there should be an effort to provide the computer forensic analysts with more detailed and concrete procedures that will help them accomplish their target.

For this reason, with respect and based on the computer forensics frameworks aiming to aid digital investigations, there is an approach of investigating ID Theft incidents with an independent investigation methodology. The ID Theft investigation framework distinguishes the examination in the victim's and the fraudster's side and the first level of this investigation process analysis was hence presented. This type of investigation method aims to provide results on a more focused basis regarding an ID Theft incident. Future work includes a more detailed approach to the findings of the investigation process based on the evidence left behind on a fraudster's digital storage media and the victim's accordingly. An experimental assessment on fictional cases by analysing reliable, residual data from hard disk drives will validate the research; and that will be accomplished in two parts, where the researcher behaves as the perpetrator in a closed network attack in the laboratory, and where the researcher uses the evidence that is left behind (from the first experiment) and acts as a forensic examiner, analysing the hypothetical victims' hard disk drives.

## REFERENCES

- Beebe N.L., Clark J. G, 2005, A hierarchical, objectives-based framework for the digital investigations process, Digital Investigation, Volume 2, Issue 2, June 2005, Pages 147-167
- Carrier B.D., Spafford E.H., 2006, Categories of digital investigation analysis techniques based on the computer history model, Digital Investigation Volume 3, Supplement 1, September 2006, Pages 121-130  
The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)
- Carrier B., 2006, Basic Digital Forensic Investigation Concepts, Internet, Available from: [http://www.digital-evidence.org/di\\_basics.html](http://www.digital-evidence.org/di_basics.html), [Last Accessed: 12/03/2007]
- Casey, E., April 2003, Determining Intent — Opportunistic vs Targeted Attacks, Computer Fraud & Security, Volume 2003, Issue 4, pp. 8-11
- Chisum W.J., Turvey B., 2006, Crime Reconstruction, Academic Press Inc., U.S., ISBN-10: 0123693756, Chapter 1: A History of Crime Reconstruction, p.23
- CIFAS, 2007, Is Identity Theft Serious?, Internet, Available from: [http://www.cifas.org.uk/default.asp?edit\\_id=556-56](http://www.cifas.org.uk/default.asp?edit_id=556-56), [Last Accessed: 14/09/2007]
- Dwan B., 2004, Identity Theft, Computer Fraud and Security, Volume 2004, Issue 4, Page 14-17
- Federal Trade Commission, February 2005, National and State Trends in Fraud & Identity Theft, January – December 2004., pdf, downloaded from: <http://www.ftc.gov/opa/2005/02/top102005.htm> [Last Accessed: 11/09/2007]
- Federal Trade Commission, February 2007, Consumer Fraud and Identity Theft Complaint Data, January – December 2006, .pdf, downloaded from: <http://www.consumer.gov/sentinel/pubs/Top10Fraud2006.pdf> [Last Accessed: 14/09/2007]
- Home Office Identity Fraud Steering Committee, 2006, <http://www.identity-theft.org.uk>, [Last Accessed: 02/04/2007]
- H. R. 2622, Fair and Accurate Credit Transactions Act of 2003, H. R. 2622, Fair and Accurate Credit Transactions Act of 2003, 2003, Internet, Available from: <http://financialservices.house.gov/media/pdf/108hr2622ai.pdf> [Last Accessed: 11/09/2007]
- Icove D. et al., 1995, Computer Crime, A Crimefighter's Handbook, O'Reilly, ISBN: 1-56592-086-4
- ID Theft and Assumption Deterrence Act, 1998, available from: <http://www.ftc.gov/os/statutes/itada/itadact.htm>, United States Code, § 003 (7), 30/10/1998 [Last Accessed: 11/09/2007]
- Marcella A. J., Greenfield R., 2002, Cyber Forensics: A Field Manual for Collecting, Examining and Preserving Evidence of Computer Crimes, CRC Press LLC, ISBN: 0-8493-0955-7
- McKenna B., 2004, Cyber attacks on banks double from 2003, Computer Fraud and Security, Volume 2004, Issue 6, Page 3

Postnote Computer Crime, October 2006, Parliamentary Office of Science and Technology, Number 271,  
Available from: <http://www.parliament.uk/documents/upload/postpn271.pdf> [Last Accessed: 25/04/2007]

Thornton, J.I., 1997, "The General Assumptions And Rationale Of Forensic Identification", in David L. Faigman, David H. Kaye, Michael J. Saks, & Joseph Sanders, *Modern Scientific Evidence: The Law And Science Of Expert Testimony*, vol. 2, St. Paul: West Publishing Co.

unknown, 26/03/2007, One in ten Britons victim to online fraud, Available from:  
[http://www.inthenews.co.uk/news/news/finance/one-in-ten-britons-victim-online-fraud-\\$1071167.htm](http://www.inthenews.co.uk/news/news/finance/one-in-ten-britons-victim-online-fraud-$1071167.htm)  
[Last Accessed: 03/09/2007]

## **COPYRIGHT**

Olga Angelopoulou ©2007. The author assigns Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## Extracting Inter-arrival Time Based Behaviour from Honeypot Traffic using Cliques

Saleh Almotairi<sup>1</sup>, Andrew Clark<sup>1</sup>, Marc Dacier<sup>2</sup>, Corrado Leita<sup>2</sup>,  
George Mohay<sup>1</sup>, Van Hau Pham<sup>2</sup>, Olivier Thonnard<sup>2</sup>, Jacob Zimmermann<sup>1</sup>  
<sup>1</sup>Queensland University of Technology, GPO Box 2434, Brisbane 4001, Australia  
<sup>2</sup>Institut Eurécom BP 193, F06904 Sophia Antipolis cedex, France  
{s.almotairi, a.clark, g.mohay, j.zimmerm}@isi.qut.edu.au  
{marc.dacier, corrado.leita, van-hau.pham}@eurecom.fr olivier.thonnard@rma.ac.be

### Abstract

*The Leurre.com project is a worldwide network of honeypot environments that collect traces of malicious Internet traffic every day. Clustering techniques have been utilized to categorize and classify honeypot activities based on several traffic features. While such clusters of traffic provide useful information about different activities that are happening in the Internet, a new correlation approach is needed to automate the discovery of refined types of activities that share common features. This paper proposes the use of packet inter-arrival time (IAT) as a main feature in grouping clusters that exhibit commonalities in their IAT distributions. Our approach utilizes the cliquing algorithm for the automatic discovery of cliques of clusters. We demonstrate the usefulness of our methodology by providing several examples of IAT cliques and a discussion of the types of activity they represent. We also give some insight into the causes of these activities. In addition, we address the limitation of our approach, through the manual extraction of what we term supercliques, and discuss ideas for further improvement.*

### Keywords

Honeypots, Internet traffic analysis, clustering, inter-arrival times

## INTRODUCTION

The work described in this paper builds upon previous work by Pouget et al. (2006) in the use of data obtained in the *Leurre.com* environment for detecting anomalous Internet traffic. This distributed low interaction honeypot environment currently consists of 50 platforms located in 30 different countries. In the previous work we have shown that the analysis of the inter arrival times (IATs) between packets collected in this environment could provide a valuable contribution to network forensics.

In that work, we used the notion of *attack clusters* proposed in earlier work by some of the same authors and we introduced the notion of cliques of clusters as an automated knowledge discovery method. A clique is a group of clusters that share common characteristics related to one or maybe a few attack processes. This paper revisits the approach to identifying IAT-based cliques in an attempt to achieve the automatic derivation of cliques of clusters with common IAT characteristics to better identify the repeated use of commonly used tools, and also to identify spurts of activity by such commonly used tools. We present a systematic approach for building new cliques and provide an extensive validation of the approach over large datasets. Last but not least, by means of *classes of cliques*, we show the usefulness of the approach as a result of the new knowledge they help to derive about the attack traces collected.

The structure of the paper is as follows. First we define our old notion of clusters as well as our new definition of the cliques and the process to build them. We then validate the approach thanks to experimental results carried out over data obtained during a period of 3 months (March to May 2007) in the *Leurre.com* environment. Finally we provide conclusions and discussion about the results of the paper.

## CLUSTERS AND CLIQUES

In this section we describe the techniques used to classify the traffic collected by the various honeypots. We first introduce the basic concepts of the *Leurre.com* traffic analysis. We then recall the *Leurre.com* clustering algorithm, and then present the novel cliquing algorithm introduced in this paper.

### Terminology

A *platform* is one of the many *Leurre.com* honeypot sites. Each platform contains three *virtual hosts*, with distinct IP addresses, impersonating three different OS behaviours taking advantage of the *honeyd* software. The

honeypots are configured to obtain a minimal level of interaction, replying to ICMP echo requests and establishing TCP connections on their open ports.

All activity observed by the honeypots is attributed to a *source*, meant to uniquely identify an attacker taking into consideration dynamic addressing. Two activities generated by a given IP address and separated by a period of more than 25 hours are attributed to two different sources.

A *large session* is a collection of packets exchanged between one source and one platform, while a *tiny session* groups the packets exchanged between one source and one virtual host. A large session is thus composed of up to three tiny sessions, ordered according to the virtual hosts' IP addresses.

A *port sequence* is the sequence of (TCP or UDP) ports targeted on a virtual host, within a tiny session.

A packet *inter-arrival time* or *IAT*, is the time difference (in seconds) between the arrival of two consecutive packets at a virtual host (i.e., within a tiny session).

### Clustering Algorithm

The first step of the clustering algorithm consists in grouping large sessions into *bags*. This grouping aims at differentiating between various classes of activity taking into consideration a set of preliminary discriminators, namely the number of targeted virtual hosts and the *unsorted* list of port sequences hitting them.

In order to further refine the bags, a set of continuous parameters is taken into consideration for each large session, namely: its duration, the total number of packets, the average IAT, and the number of packets per tiny session. These parameters can assume any value in the range  $[0, \infty]$ , but some ranges of their values may be used to define bag subclasses. This is done through a peak picking algorithm that identifies ranges of values considered discriminating for the bag refinement. Large sessions belonging to a bag and sharing the same matching intervals are grouped together in a *cluster*.

A very last refinement step is the *payload validation*. The algorithm considers the concatenation of all the payloads sent by the attacker within a large session ordered according to the arrival time. If it identifies within a cluster multiple groups of large sessions sharing similar payloads, it further refines the cluster according to these groups.

### Cliquing Algorithm

Due to the large quantity of data we collect, we need to rely on an automated methodology that is able to extract relevant information about the attack processes. Our correlative analysis relies on concepts from graph and matrix theory. In this context, a *clique* (also called a complete graph) is an induced subgraph of an (un)directed graph in which the vertices are fully connected. In our case, each node represents a cluster, while an edge between a pair of nodes represents a similarity measure between two clusters. The main focus of this work is on computing similarities between IAT distributions, but our methodology can be applied to any type of vector or time series.

Determining the largest clique in a graph is often called the *maximal clique problem* and it is a classical graph-theoretical, NP-complete problem (Bron and Kerbosch, 1973). Although numerous exact algorithms (Kumlander 2004a, 2004b, Bomze et al. 1999) and approximate methods (Bomze et al. 2000, Pavan and Pelillo 2003) have been proposed to solve this problem, we address the computational complexity of the clique problem by applying our own heuristics to generate sets of cliques very efficiently. While our technique is relatively straightforward, it possesses two significant features. Firstly, our technique is able to deliver very coherent results with respect to the analysed similarities. Secondly, regarding the computational speed, our technique outperforms other algorithms by several orders of magnitude. For example, we applied the approximate method proposed by Pavan and Pelillo (2003) which consists of iteratively extracting *dominant sets* of maximally similar nodes from a similarity matrix. On our dataset, the total computation was very expensive (several hours) whereas our custom cliquing algorithm only takes a few minutes to generate the same cliques of clusters with the same dataset.

On the other hand, our heuristic imposes a constraint on the similarity measure, namely that it has to be *transitive*. With this restriction, it is sufficient to compute the correlation between one specific node and all other nodes in order to find a maximal clique of similar nodes. We achieve this transitive property by carefully setting a global threshold on the measurement of similarities between clusters (see next section).

Here are the different steps of our cliquing algorithm:

We define a quantitative representation for the feature to correlate (in this work: the IAT distribution within clusters).

We choose a well-suited similarity measure for this characteristic.

Consider the list of all clusters. While this list is not empty:

We consider the next cluster in the list and we take the corresponding characteristic vector;

We compute the similarities with all other remaining vectors;

If there are other similar clusters (with respect to the defined threshold), we put all of them in a new clique. We remove those clusters from the list and start the next iteration.

If there is no other similar cluster, we remove the current cluster from the list, store it in a separate group, and start the next iteration.

Clearly, this algorithm takes advantage of the already created cliques to progressively decrease the search space; so in the average case the algorithmic complexity will be less than  $O(n^2)$ , and we could expect typically a complexity order of  $O(n \log(n))$ . The exact complexity analysis of our algorithm is out of the scope of this paper.

#### Cluster Correlation using Packet Inter-arrival Times

The first step in our methodology is to construct the cluster characteristics. We represent the IAT distributions of the clusters with a vector in which every element corresponds to the IAT frequency of a pre-defined bin (range of time values). We end up with an IAT vector of 152 bins where the first bin groups IATs falling in the interval 0-3 seconds, and the last bin corresponds to IATs of 25 hours or more.

To circumvent the limitations of our previous work, we now rely on a similarity measure that is based on a recent technique called *symbolic aggregate approximation* (SAX) (Lin et al. 2003). SAX aims at reducing a complex time series to a symbolic approximation without losing too much quality with respect to the “shape” of the signals. It is a *piecewise aggregation approximation* (PAA) technique which tends to approximate time series by segmenting them into intervals of equal size and summarizing each of these intervals by its mean value.

SAX uses predetermined breakpoints during the quantization, chosen so as to maximize the energy of the quantized representation of the time series, which are then interpreted as a string of symbols taken from a finite alphabet. Figure 1 gives an example of a time series converted to a SAX representation which has been mapped to the string “eefeffecbbabaab”. A SAX representation of a time series  $T$  of length  $N$  can be denoted by  $W_T(N, w, \alpha)$ , where:  $N$  is the number of elements in  $T$ ;  $w$  is the number of elements in the SAX representation of  $T$  (i.e. the length of  $W_T$ ); and  $\alpha$  is the alphabet size (number of quantization levels). The ratio  $r = N/w$  is called the compression ratio. A value of  $r = 10$  means that 10 elements of  $T$  are mapped to a single symbol in  $W_T$ .

One of the strong advantages of SAX resides in the fact that this technique allows a distance measure that lower bounds the original distance measure (e.g. the Euclidean distance, see Lin et al. (2003) for the proof). SAX defines a MINDIST function that returns the minimum distance between the original time series of two words. Let  $T_1$  and  $T_2$  be two time series of same length  $N$ , then the minimum distance given by SAX can be calculated as follows:

The *dist()* function returns the inter-symbol distance and can be implemented using a table lookup for better computational efficiency (see Lin et al. (2003) for more details).

*Figure 1. Example of SAX representation of a time series  $W_T(256, 16, 6)$ .*

In order to achieve a transitive similarity function, we set a global threshold on the distance computed with SAX. Only if the similarity measure exceeds 99% of the maximal theoretical value, do we assume that the two vectors are completely similar. This experimental heuristic gives fair results. A drawback of this approach, as for every method which relies upon a threshold, is that a good preliminary tuning is needed to fit it to the data.

SAX can also typically compress the time series, but we chose here not to compress the IAT vectors because we already defined packet IAT bins which regroup all values falling in those respective intervals.

## Experimental Results

We now describe our analysis of the IAT-based cliques obtained using the above approach when applied to the *Leurre.com* dataset. We consider a dataset covering three months of traffic (March – May 2007) collected from the *Leurre.com* environment.

For the sake of conciseness, in the analysis presented in this paper, we only consider clusters which have at least one bin, after the 21<sup>st</sup> bin (the 22<sup>nd</sup> bin corresponds to around five minutes), with a count of more than 10. This

means that we ignore clusters which do not have more than 10 occurrences of at least one IAT value greater than five minutes.

After this filtering, we obtained 1475 vectors representing the IAT frequency distributions of the corresponding clusters. The clique algorithm described above was then applied to these vectors, yielding 111 IAT-based cliques comprising 875 clusters. The remaining 600 clusters did not fall into any clique.

Each clique contains a group of clusters which, based upon their IAT distribution (and the parameters of the cliquing algorithm) are similar. Prior to describing our detailed analysis of the cliques obtained we present three types of cliques that we expected would, *inter alia*, be represented in the results:

Type I: Cliques which contain clusters of large sessions targeting the same port sequences. The difference between the various clusters contained within such a clique lies in the number of packets sent to the targeted ports. These cliques are mostly symptomatic of classes of attacks where the attacker repeatedly tries a given attack, a varying number of times.

Type II: Cliques composed of clusters of large sessions targeting different port sequences but exhibiting the same IAT profile. These cliques are symptomatic of tools that send packets to their target according to a very specific timing and that have been used in several distinct campaigns targeting different ports.

Type III: Cliques which contain clusters grouped together based upon the presence of long IATs (longer than 25 hours), representing sources which are observed on one platform, then, within 25 hours, are detected on another platform, before again returning to the original platform. Such behaviour would be indicative of a source which is scanning large numbers of devices across the internet, in a predictable manner, resulting in them repeatedly returning to the same platform.

We also found many similarities across the different cliques that were generated. We identified a number of so-called *supercliques* as a result which suggests that the IAT-based analysis we have focused on in this paper is good at automatically identifying very specific types of activity within a very large dataset. Our analysis of these supercliques is presented below.

### Type I Cliques

Type I cliques are expected to contain clusters which are very similar with respect to most traffic features, including port sequence, with one exception being that the large and tiny sessions within the clusters contain varying durations (both in terms of time, and the number of packets sent by the source). The variation in the duration of the sessions will account for such traffic being arranged in different clusters. Two particular cliques that are seen to fall clearly into the Type I category are Clique 7 and Clique 49 (summarised in Table 1).

Clique 7 is composed of 8 clusters, 9 large sessions and a total of 821 packets. These clusters are mainly contained in one bag (corresponding to the very common port sequence of TCP/135). In this clique, there are 5 platforms targeted by 6 distinct IP addresses originating from 4 different countries (China, Germany, Japan, and France). The peak IAT bin is bin 32 with IAT values in the range 554-583 seconds, and the average duration is 70491 seconds with a minimum duration of 4657 seconds, and a maximum of 236350 seconds.

All three virtual hosts on each of the targeted platforms were hit with the same number of packets, with the average number of packets per session equal to 35. Also, several IP addresses were found to occur in multiple clusters within the clique. While these sources were grouped in different clusters due to their varying durations, there were strong similarities in terms of the IAT characteristics of the sessions, resulting in these clusters being grouped in the same clique.

Clique	Clusters	Large Sessions	Packets	Bags	Platforms Targeted	Source IPs	Countries	Targeted port sequence	Peak IATs (bin)	Min, average, max durations in secs	No of target virtual hosts per platform
7	8	9	821	1	5	6	4	TCP/135	554-583 (32)	4657, 70491, 236350	3

49	11	285	3274	1	37	248	46	TCP/22	2703-3597 (49)	1035, 9922, 137528	3
----	----	-----	------	---	----	-----	----	--------	----------------	--------------------	---

Table 1: Type I Cliques

Clique 49 contains 11 clusters, 285 large sessions, and 3274 packets, and the targeted port sequence is TCP/22. There are 248 distinct IP addresses which attacked 37 different platforms. The sources of the IPs are widely spread among 46 different countries. Despite the widespread location of the sources of the traffic in this clique, there are a number of similarities in the observed behaviour. Firstly, large sessions in this clique always targeted all three virtual hosts on each platform, and the number of packets sent to each virtual host was similar in each case (one packet for the Windows hosts and an average of 10 packets for the UNIX host). The average duration of attacks is 9922 seconds with minimum and maximum durations in the range of 1035 to 137528 seconds. The majority of the clusters in this clique belong to the same bag. The IAT sequences of these clusters are similar with all IATs in the session being short except one which belongs to bin 49 (2703-3597 seconds).

Cliques 7 and 49 were typical examples of Type I cliques where attack traffic ends up in different clusters due to the variations in either the duration of the attack or the number of packets sent. In each case the duration and number of packets varied significantly between the sessions, while the IAT behaviour remained consistent. Also, a number of IP address were shared between clusters within each clique, with over 50 % of the clusters sharing IP addresses or class C networks.

The identification of cliques of Type I addresses a weakness of the original clustering algorithm which was, by design, unable to group together activities that clearly were related to each other and should have, therefore, be analysed together.

#### Type II Cliques

Type II cliques are those which contain a large variety of targeted port sequences, yet each cluster exhibits similar IAT characteristics. We hypothesise that clusters belonging to this type of clique correspond to the same attack tool using the same strategy to probe a variety of ports (such as a worm which targets multiple vulnerable services, or some other type of systematic scanner targeting a number of different ports). Two cliques which exhibit this type of behaviour are Cliques 92 and 69 (see Table 2).

Clique	Clusters	Large Sessions	Packets	Bags	Platforms Targeted	Source IPs	Countries	Targeted Port Sequences	Peak IATs (bin)	Min, average, max duration in secs	No of target virtual hosts per platform
92	40	502	4234	7	1	502	25	(all TCP) 6769 7690 12293 18462 29188 64697 64783	933-1797 (46); 1803-2702 (48);	953, 9278, 53941	1
69	64	1336	17097	8	2	1300	37	(all TCP) 4662 6769 7690	933-1797 (46)	133, 44163, 225224	1

								12293			
								29188			
								38009			
								64697			
								64783			

Table 2: Type II Cliques

Clique 92 consists of 40 clusters, 502 large sessions and 4234 packets in total. While a variety of ports are targeted by these clusters, traffic within each cluster only targets a single port. The TCP ports targeted within this clique are: 6769, 7690, 12293, 18462, 29188, 64697, and 64783. This clique is a result of 502 distinct source IP addresses originating from 25 different countries, and targeting only a single platform. Additionally, only one virtual host was targeted on this platform. The average number of packets per large session was 16 (minimum 3 and maximum 103), and the average duration was 9278 seconds. Clusters in this clique belong to 7 different bags (corresponding to the 7 different ports targeted). Clique 92 contains peak IAT bins of 46 (933-1797 seconds) and 48 (1803-2702 seconds) where the IAT sequences are repeated patterns of short and long IATs. A possible explanation for the traffic which constitutes this clique is that it corresponds to the same tool being used to scan for the existence of services which use a strange port (such as peer-to-peer related services) – where the scan uses a regular (long) delay between retransmissions.

Clique 69 is similar to Clique 92 in that it also contains a variety of clusters where each cluster contains traffic targeting a single, unusual port. This clique contains 64 clusters, 1336 large sessions and 17097 packets. It is a result of 1300 distinct attacking IP addresses, that originate from 37 different countries and target 2 platforms (all but one target the same platform as that targeted by the traffic in Clique 92). The targeted TCP ports are: 4662, 6769, 7690, 12293, 29188, 38009, 64697, and 64783. Clusters in this clique belong to 8 different bags, and in each case only one virtual host was targeted per platform. The durations of attacks range from 133 to 225224 seconds with an average of 44163 seconds. The number of packets sent in each large session is in the range 2 to 135 with an average of 25 packets. The IAT sequences are repeated patterns of short, short, and long IATs with a peak IAT bin of 46 (933-1797 seconds).

The traffic in Cliques 92 and 69 represent a large number of distinct sources from a variety of counties targeting a variety of ports, predominantly (with one cluster being the exception), targeting the same platform in China. These cliques represent very interesting activity which is difficult to characterise in further detail due to the lack of interactivity of the honeypots on these ports. The significance of the ports being targeted is unclear, but might be easier to determine if packet payloads were available. The fact that all of these sources exhibit a very distinct fingerprint in terms of their IAT characteristics makes the activity all the more unusual.

The identification of cliques of Type II enables us to highlight, in a systematic way, the existence of tools with a specific IAT profile that are reused to launch different attack campaigns against various targets. Without such analysis, the link that does exist between the IPs belonging to different clusters in a given clique would have remained hidden.

### Type III Cliques

Based upon our observation of the *Leurre.com* data over a long period of time, we found that there are a number of large sessions which continue for an extended duration (sometimes many weeks). Of these there are a number which target multiple platforms within a 25 hour period, where the intervening time before returning to the same platform is more than 25 hours. These very long IATs are placed into bin 152 during the cliquing process. A number of cliques that resulted from the cliquing algorithm were characterised by these long IATs, and here we investigate two of them in detail – Cliques 31 and 66 (see Table 3).

Clique 31 is a large clique of 150 clusters, 3456 large sessions, and a total of 21422 packets. The port sequence for Clique 31 is the single port UDP/1434 (MS SQL). In Clique 31, there are 277 distinct IP addresses originating from 22 different countries which target 39 different platforms. Characteristics of clusters in this clique include: a varying number of hosts targeted, with the average number of packets sent per host equal to 12 (minimum 2 and maximum 85) and an average duration equal to 1142131 seconds. Clusters in this clique belong to 7 different bags and have IAT values that exceed 25 hours (IAT peak bin 152). These sessions are indicative of a very slow scanner which is seen on multiple platforms, returning to the same platform only after an extended delay of more than 25 hours.



Clique	Clusters	Large Sessions	Packets	Bags	Platforms Targeted	Source IPs	Countries	Ports targeted	Peak IATs in secs (bin)	Min, average, max durations in secs	No of target virtual hosts per platform
31	150	3456	21422	7	39	277	22	UDP/1434	>25 hours (152)	132, 1142131, 7509849	varies
66	3	13	171	3	12	9	2	UDP/1026; UDP/1027	Very large (152)	1, 381408, 915002	3

Table 3: Type III Cliques

Clique 66 contains 3 clusters, 13 large sessions and 171 packets. These sessions are characterised by sending multiple packets, alternating between UDP ports 1026 and 1027 repeatedly. In Clique 66, 12 platforms were targeted by 9 distinct IP addresses originating from 2 different countries. All clusters within this clique contain sessions which target all three virtual hosts on the target platforms, with only a small number of packets sent per session (on average 4, with a minimum of 3, and a maximum of 6). The average session duration is 381408 seconds. Clusters in this clique belong to 3 different bags and, again, the IAT durations are very large.

Cliques 31 and 66 represent examples of activities where a source IP is scanning the globe, targeting different honeypot platforms in less than 25 hours. UDP port 1434 is used by the MS SQL Monitor service and is the target of several worms, such as W32.SQLExpWorm and Slammer. It is likely that traffic targeting this port is result of worms that scan for vulnerable servers. UDP ports 1026 and 1027 are common targets for Windows Messenger spammers, who have been repeatedly targeting these ports since June 2003<sup>1</sup>.

### Supercliques

We observed that the across all of the obtained cliques, only a relatively small number of peak IAT bin values were represented. Indeed, from the point of view of the peak bin values, we found that a limited number of combinations existed. This suggests that the cliques we obtained possess a high level of uniformity in terms of the activities that they represent. Based upon the small set of common peak bins, and the dominant port sequences targeted within those cliques, we manually grouped the cliques together into 6 *supercliques*, which are summarised in Table 4.

Superclique	Cliques	Clusters	Large Sessions	Distinct IPs	Peak Bins	Port Sequence
1	7	166	3505	277	152	1434U
2	5	12	22	12	152	1026U1027U ...
3	6	29	288	247	46, 48, 49	135T
4	4	21	541	429	46, 48, 49	22T
5	23	183	6313	6188	46, 48, 49	unusual TCP ports

6	23	74	164	152	31, 32	135T
---	----	----	-----	-----	--------	------

Table 4: Supercliques and their representative properties.

As can be seen from the table, the supercliques account for just over half of the cliques generated. The cliques not represented within the supercliques were not considered in the remaining analysis.

Representative examples of each of the first five supercliques have been presented in the previous three sections. The Type I Cliques 7 and 49 are examples of Supercliques 3 and 4, respectively. Superclique 6 contains Type I cliques which target port TCP/135, similar to Superclique 3, with the difference being that the dominant IAT for cliques from Superclique 6 are in bins 31 and 32, rather than 46, 48, and 49 (for Superclique 3). Cliques 92 and 69 (Type II) are examples of cliques from Superclique 5. The Type III Clique 31 is an example of a clique that belongs to Superclique 1; while Type III Clique 66 is an example of a clique from Superclique 2.

## DISCUSSION AND CONCLUSIONS

We have generated automatically a number of cliques that represent a variety of interesting activities which target the *Leurre.com* environments. We have shown that more than half of the cliques can be easily characterized as one of the three major types identified above. Indeed, in accordance with the supercliques that we manually identified, there are six major classes of activity that the cliquing algorithm has extracted for the time period that we examined (the supercliques). The strong similarities within the supercliques highlight the usefulness of the cliquing algorithm for identifying very particular kinds of traffic observed by the honeypots. Further fine-tuning of the cliquing algorithm may allow these (super)cliques to be automatically generated.

The automatic identification of cliques of the different types outlined in this paper represents a significant contribution to both addressing weaknesses in the original clustering algorithm, as well as highlighting, in a systematic way, the existence of tools with a specific IAT profile. While our analysis has focused only on the *cleanest* cliques (i.e., the ones that represent consistent behaviour in terms of the characteristics we investigated, such as port sequence, number of virtual hosts targeted, and the targeted platforms), there are many other cliques that contain potentially interesting activities that should be further investigated in the future.

Due to the low interaction nature of the honeypots used by the *Leurre.com* project the majority of the activities observed will relate to different types of scanning (or backscatter), such as that from automatically propagating malware, or scanners which may be cataloguing the existence of various servers around the world, for example. While it is difficult to reach accurate conclusions about the exact nature of the tools which generate the packets collected, we have shown that the cliquing approach adds useful detail to the existing clusters by automatically extracting significant classes of activity from an extremely large dataset.

## REFERENCES

- Bomze, I.M., Budinich, M., Pardalos, P. M., and Pelillo, M. (1999) The maximum clique problem, *Handbook of Combinatorial Optimization*, vol. 4. Kluwer Academic Publishers, Boston, MA.
- Bomze, I.M., Pelillo, M., and Stix, V. (2000) Approximating the Maximum Weight Clique Using Replicator Dynamics, *IEEE-NN Journal*, vol. 11, no. 6.
- Bron, C. and Kerbosch, J. (1973) Algorithm 457: finding all cliques of an undirected graph, *Comm. ACM Press*, vol. 16, no. 9, pp. 575-577, New-York, USA.
- Kumlander, D. (2004a) A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search, in proceedings of *Fourth International Conference on Engineering Computational Technology*, Civil-Comp Press, p. 137-138 (an extended abstract - the full paper is published on a CD-ROM).
- Kumlander, D. (2004b) An exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring, *Modelling, Computation and Optimization in Information Systems and Management Sciences* (edited by Le Thi Hoai An & Pham Dinh Tao), Hermes Science Publishing, pp. 202-208.
- Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003) A symbolic representation of time series, with implications for streaming algorithms, in proceedings of *Eighth ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, California, USA.
- Pavan, M. and Pelillo, M. (2003) A new graph-theoretic approach to clustering and segmentation, in proceedings of *IEEE Conference on Computer Vision and Pattern Recognition*.

- Pouget, F., Dacier, M., and Pham, V.H. (2004) Towards a better Understanding of Internet Threats to Enhance Survivability, in proceedings of *International Infrastructure Survivability Workshop (IISW'04)*, Lisbonne, Portugal.
- Pouget, F. (2005) Distributed system of Honeypot Sensors: Discrimination and Correlative Analysis of Attack Processes, PhD Thesis from the Ecole Nationale Supérieure des Télécommunications, available through the Eurécom Institute library
- Pouget, F., Dacier, M., Zimmermann, J., Clark, A., and Mohay, G. (2006) Internet Attack Knowledge Discovery via Clusters and Cliques of Attack Traces, *Journal of Information Assurance and Security*, vol. 1, pp. 21-32.
- Spitzner, L. (2003) The HoneyNet Project: Trapping the Hackers, *IEEE Security and Privacy*, 1, p. 15.
- Zimmermann, J., Clark, A., Mohay, G., Pouget, F., and Dacier, M. (2005) The Use of Packet Inter-Arrival Times for Investigating Unsolicited Internet Traffic, in proceedings of *Systematic Approaches to Digital Forensic Engineering (SADFE)*, Taipei, Taiwan.

## **COPYRIGHT**

Saleh Almotairi, Andrew Clark, Marc Dacier, Corrado Leita, George Mohay, Van Hau Pham, Olivier Thonnard, Jacob Zimmermann ©2007. The authors assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## Multi-Step Scenario Matching Based on Unification

Sorot Panichprecha, Jacob Zimmermann, George Mohay, Andrew Clark  
Information Security Institute  
Queensland University of Technology  
{s.panichprecha, j.zimmerm}@isi.qut.edu.au, {g.mohay, a.clark}@qut.edu.au

### Abstract

*This paper presents an approach to multi-step scenario specification and matching, which aims to address some of the issues and problems inherent in to scenario specification and event correlation found in most previous work. Our approach builds upon the unification algorithm which we have adapted to provide a seamless, integrated mechanism and framework to handle event matching, filtering, and correlation. Scenario specifications using our framework need to contain only a definition of the misuse activity to be matched. This characteristic differentiates our work from most of the previous work which generally requires scenario specifications also to include additional information regarding how to detect the misuse activity. In this paper we present a prototype implementation which demonstrates the effectiveness of the unification-based approach and our scenario specification framework. Also, we evaluate the practical usability of the approach.*

### Keywords

Computer Forensics, Event Correlation, Multi-step Scenario, Signature Matching, Event Representation

## INTRODUCTION

This paper presents our approach for detecting multi-step misuse scenarios using a unification algorithm as a signature matching mechanism. The unification algorithm has been mainly used in logic programming and in the artificial intelligence field due to its ability to confirm or refute solutions to problems using inference on a set of given rules in order to find an expression that matches the fact base. The unification algorithm, or some form of it, has been used by some misuse-based intrusion detection systems (Olivian and Goubault-Larrecq 2005, Mounji 1997). Rules in the unification algorithm, when applied to misuse-based intrusion detection systems, are considered to be descriptions of misuse activities. We have extended and refined previous approaches to such use of the unification algorithm and have adapted it to the special requirements of scenario detection in order to provide increased benefits of extensibility and flexibility. Previous work using built-in unification (as in Prolog based intrusion detection systems) has required scenario writers in addition to specify state information and context information. In addition, we incorporate an event correlation framework which utilises abstraction of events derived from heterogeneous sources, e.g., system audit data, application audit data, and captured network traffic.

Multi-step scenario matching is a complex process. There are several issues regarding the detection of multi-step scenarios as follows. The detection, in most cases, requires dealing with events from multiple sources e.g., system audit data, application logs, and captured network traffic. Therefore, multi-step scenario specification and detection is problematic due to the fact that event data from different sources may use different semantics and syntax. The specification of scenarios in most of the previous work (Kumar 1995, Lindqvist and Porras 1999, Yang et al. 2000, Meier et al. 2005, Illgun et al. 1995) requires an intimate understanding of the underlying matching mechanisms and how to match a given scenario. It is thus easily prone to human errors. In our work, we aim at minimising the scenario writers' focus on the how of the matching mechanism providing them with a framework which requires them only to express what to detect, as suggested by (Roger and Goubault-Larrecq 2001).

To address the lack of a uniform event representation, we employ an event abstraction model as proposed in (Panichprecha et al. 2006). The event abstraction model provides a uniform representation of events from multiple sources, e.g., system audit data, application logs, and network traffic. By incorporating the tools provided by the model, our work can access events across multiple platforms and sources. □

We have developed a prototype of the unification-based multi-step scenario matching. The prototype is intended to be a tool which facilitates log analysis. We have also developed a Python-based scenario specification framework for specifying multi-step scenarios. A number of misuse scenarios have been used to demonstrate the framework and to evaluate the unification-based scenario matching framework.

## RELATED WORK

There has been a considerable body of research into multi-step scenario detection focusing on intrusion detection (Lindqvist and Porras 1999, Mounji 1997, Ilgun et al. 1995, Kumar 1995), alert correlation (Morin et al. 2002, Carey et al. 2003), computer forensics (Abbott et al. 2006), and vulnerability assessment (Ou et al. 2005). Existing work in these fields share a common characteristic, in particular, they aim to compare two expressions (e.g., event data with scenario specifications or host information with threat information) and return the results of the comparison.

There are two multi-step scenario matching techniques of relevance to this work: the use of rule-based expert systems (Lindqvist and Porras 1999, Mounji 1997, Roger and Goubault-Larrecq 2001) and state transition models (Ilgun et al. 1995, Kumar 1995, Cuppens and Ortalo 2000, Meier et al. 2005). We now provide brief details of these techniques, their advantages, and limitations.

Rule-based expert system techniques define mechanisms to compare rules (scenarios or signatures) against audit records. Examples of such systems are EMERALD (Lindqvist and Porras 1999), ASAX (Mounji 1997), and ORCHIDS (Olivain and Goubault-Larrecq 2005). EMERALD uses a forward chaining rule-based expert system where the system establishes a chain of rules which links facts (audit records) to goals (signatures). Similarly, ASAX uses a rule-based expert system, however, it specifies signatures as pairs of conditions and actions. When a condition is met, the corresponding action is triggered, which can either be activating another set of chain condition pairs or reporting an alert. ORCHIDS is an intrusion detection system based on the technique proposed in (Roger and Goubault-Larrecq 2001) whose idea is derived from ASAX. The detection is performed by comparing event streams against application-specific temporal logic expressions. The advantages of the rule-based expert system approach and its variants are the simplicity and straightforwardness of the signature matching mechanism. However, this technique generally suffers from the problem of being inefficient and if the set of rules is large, which is common to intrusion detection systems, this technique will not perform well.

Similar techniques have been employed in vulnerability assessment and attack graph generation fields, i.e., adopting a logic programming approach to detect vulnerabilities on computer hosts. MulVAL (Multihost, multistage Vulnerability Analysis) proposed by Ou et. al. (Ou et al. 2005) is an example of a vulnerability assessment tool. It uses a unification-based model for the analysis of a system's exposure to various threats. The MulVAL system comprises a Datalog implementation<sup>1</sup>, a library of predefined predicates that model common threat effects (such as the ability to execute code, to transmit data over the network, to modify access control or users' privileges, etc.), a vulnerability scanner that generates a base of Datalog clauses which list the vulnerabilities present on the analysed host(s), and a base of clauses which describe the effects of known vulnerabilities. The MulVAL system utilises the Datalog built-in unification algorithm to carry out various types of analysis, namely: threat assessment, security policy assessment, and speculative analysis. One advantage of the MulVAL system is its ability to represent threats that result from a combination of vulnerabilities. This is a direct consequence of the use of Prolog-style unification: each vulnerability effect is represented as a Datalog clause, which can in turn be reused as a condition for another clause. Our present work is motivated by this benefit, however its approach and purpose are different. Our goal is the detection of actual occurrences of multi-step attack scenarios in analysed logs. Instead of relying on Prolog-style variables and clauses only, we use a higher-level data model, i.e., the event abstraction model proposed in (Panichprecha et al. 2006), which provides a uniform and generic representation of events on the system and network being monitored. We also represent attack scenarios as clauses, however we introduce specific features designed to take full advantage of the underlying framework's expressive power. We have implemented a dedicated unification engine, adapted to the framework and scenario model. Finally, for prototyping purposes, we define a simple concrete syntax for attack scenario specification, based on the Python programming language (Python Software Foundation 2007).□

*State transition techniques* model attacks against a system in terms of system states and state transitions. An occurrence of an attack is identified by reaching the terminal state of a signature. Examples of such systems are IDIOT (Kumar 1995), EDL (Meier et al. 2005), LAMBDA (Cuppens and Ortalo 2000), and STAT (Ilgun et al. 1995). IDIOT and EDL use a Petri-net based modelling approach to signature specification and matching. The state transition techniques have been proved to perform well in real-time detection. However, they suffer from the complexity of the state and transition instantiation mechanism.□

*Scenario and signature specification languages* have been typically designed to match their underlying matching techniques. In this paper, we are interested in the languages which allow the expression of multi-step misuse activities (Lindqvist and Porras 1999, Mounji 1997, Eckmann et al. 2000, Cuppens and Ortalo 2000, Michel and Mé 2001, Meier et al. 2002, 2005, Lin et al. 1998, Ning et al. 2002, Pouzol and Ducassé 2001). To name a few, P-BEST (Production-Based Expert System Toolset) (Lindqvist and Porras 1999) and STATL (State

---

<sup>1</sup> Data log is a subset of the Prolog programming language.

Transition Analysis Technique Language) (Eckmann et al. 2000) are widely recognised and are good examples of systems which implement the rule-based expert system techniques and the state transition techniques respectively. □

P-BEST is used in several rule-based expert system namely EMERALD (Porras and Neumann 1997) and several other systems (Sebring et al. 1988, Lunt et al. 1989, Anderson et al. 1995). The P-BEST language provides syntax for expressing inference rules and responses to derived facts. It provides operators and data structures for modelling low-level operating system and network activities. Also, the language provides an interface to the C programming language. Hence, it allows scenario developers to incorporate functions written in the C programming language.

STATL is used in the STAT framework (Eckmann et al. 2000). The STATL provides syntax for modelling system states and activities which affect the states. The modelling of activities is expressed in terms of STATL's internal structures which are provided by STAT's providers and extensions. The providers and extensions provide a single-level event abstraction where audit data and network traffic are converted into STATL's built-in structure, which is in fact a struct data type in the C++ programming language. □

In summaries, there are two main limitations with existing scenario specification languages. Firstly, most languages require scenario developers to specify scenario details at a low-level, e.g., system calls and network protocol-level details. Secondly, many of these languages expose their internal scenario matching mechanisms to scenario developers. For instance, the STATL requires scenario developers to specify transition types which need an intimate understanding about the transitions. Thus, the scenarios are often complex and easily prone to human error.

The multi-step scenario matching framework proposed in this work is rule-based and employs a unification algorithm. In fact, existing systems implicitly use some form of unification. EMERALD and ASAX use a forward-chaining (bottom-up) unification approach to detect misuse activities which consider event data to be facts and conclude new facts from the event data and assertion rules but do so by employing a built-in or implicit unification engine which is correspondingly inflexible and non-extensible.

Our work differs from previous systems which make use of the unification algorithm in several points. Firstly, we use the unification algorithm explicitly and thus derive the benefits of flexibility and extensibility which we demonstrate later in this paper. We have implemented the unification algorithm and adapted it to our needs. The advantage of implementing the unification algorithm is the flexibility to extend and optimise the matching engine thus addressing one of the limitations of previous work on scenario specification and matching. Secondly, we utilise the event abstraction model proposed in (Panichprecha et al. 2006) which allows our misuse scenario specification to specify scenarios using event abstraction rather than low-level events as in the previous work. Finally, our misuse-scenario specification approach does not require scenario developers to compile scenarios into binary executable format unlike EMERALD, ASAX, and ORCHIDS. Scenarios in our framework can be used by the engine right away which reduces the complexity and makes the system more human operator friendly.

## **OUR APPROACH: UNIFICATION-BASED SCENARIO MATCHING**

The unification algorithm was first proposed by Robinson (Robinson 1965) as a mechanism for comparing two expressions and substituting variables in one expression with variables or sub-expressions from the other so that the two expressions can be tested for syntactic equivalence (Baader and Snyder 2001). This section discusses our approach to scenario matching using the unification algorithm. We first describe how events are represented in our framework, followed by a description of the operators that can be applied to scenario specifications. We then present our overall architecture and the Python-based scenario specification framework, and describe how composite scenarios can be implemented using the framework.

### **Event Representations**

The event representation proposed in the event abstraction model (Panichprecha et al. 2006) provides a range of event representations from operating system activities, to network activities, and to application events. The event representation is built based on an object-oriented approach which comprises two object hierarchies: the Sensor Event Tree (SET) and the Abstract Event Tree (AET). The SET provides an abstraction of sensor event types derived from heterogeneous sources, e.g., audit data, application data, and network traffic. The entries from the SET are then used by the AET to represent abstract system and network activities occurring on the system or network being monitored.

In order to use this event abstraction model with the unification algorithm, we need to define additional rules to the unification algorithm. The rules are as follows:

6. Free variables can represent either objects or values of object attributes.□
7. If a variable is constrained by the class of the objects it can represent, it can be instantiated with any object which belongs to that class, or one of its subclasses.□
8. A free variable that represents the value of an attribute can be instantiated with either an atomic value, e.g., string, integer, or date, or with an object when appropriate. In the latter case, it can also possibly be constrained by the class of the object.

We use the modified unification algorithm with these rules to match attack signatures against events.

### The Operators

We have defined the following operators to enable scenario writers to specify patterns of scenarios in several aspects, i.e., term equivalence, string patterns, and chronological order of events:

- “==” defines equality of two terms based on their type. For example, if the two terms are strings, the operator means the two strings must be an exact match. If one of the two terms is a variable and the other is a class, the operator produces a class constraint on the variable;
- “!=” is a shorthand notation of the complement of ==. Note that this operator does not support event objects, due to the fact that the complement of an event object refers to all other types of event objects which can cause the number of unifiers to grow exponentially. As far as we concerned, there is no scenario that requires the application of event object complement;
- *containsPattern* defines string pattern matching using regular expression;
- *sizeGreaterThan* and *sizeLessThan* compare the lengths of two strings;
- *before* and *after* compare the timestamps of two events. Optionally a time threshold (*timeout*) can be specified.

The last two operators (before and after) are significant for multi-step scenario specifications since they allow scenario writers to specify chronological relationships between two events, which is common to multi-step scenarios. However, these two operators rely on the assumption that the timestamps are derived from a trustworthy time source and clocks of the two event sources are synchronised.

### Architecture

The architecture of our system is depicted in Figure 1. The left-hand side of the figure shows components of the event abstraction model framework proposed in (Panichprecha et al. 2006). The framework comprises a set of sensor event generators, a SET persistent object store, a set of abstract event generators, and an AET persistent object store. When the framework is executed, the sensor event generators read data from heterogeneous sources and generate corresponding sensor event objects which are recorded in the SET persistent object store. Then, the abstract event generators read sensor event objects from the SET persistent object store and generate abstract event objects which are recorded in the AET persistent object store. The right-hand side of the figure shows our unification-based scenario matching system proposed in this paper. The prototype comprises three components: a set of scenarios, the unification-based scenario matching engine, and a reporting module. When our system is executed, the unification-based scenario matching engine reads the scenarios written in our Python-based scenario specification framework and unifies them with event objects from the AET persistent object store. If the unification succeeds, the unifiers are sent to the reporting module which, at this stage, prints out all values stored in the event objects.

### Python-based Scenario Specification Framework

As a proof-of-concept, we have defined a Python-based scenario specification framework and have incorporated the event representations and the operators as discussed above into the framework. It is not intended that our framework introduce novel features or define new syntactic constructs. Rather, it is intended to demonstrate the expressiveness of unification for scenario matching. We avoid introducing new syntax and “yet another scenario specification language”, but instead use the popular Python programming language (Python Software Foundation 2007) as a foundation for scenario expressions.

We have developed a set of Python APIs which provide the means to write multi-step scenarios (i.e., expressions) for our unification algorithm. Since they are built on the Python programming language, the APIs have full access to the Python language's features and other APIs if needed. In addition, our framework incorporates the APIs provided by the event abstraction framework which enable access to uniform event representations regardless of platforms or applications (Panichprecha et al. 2006).

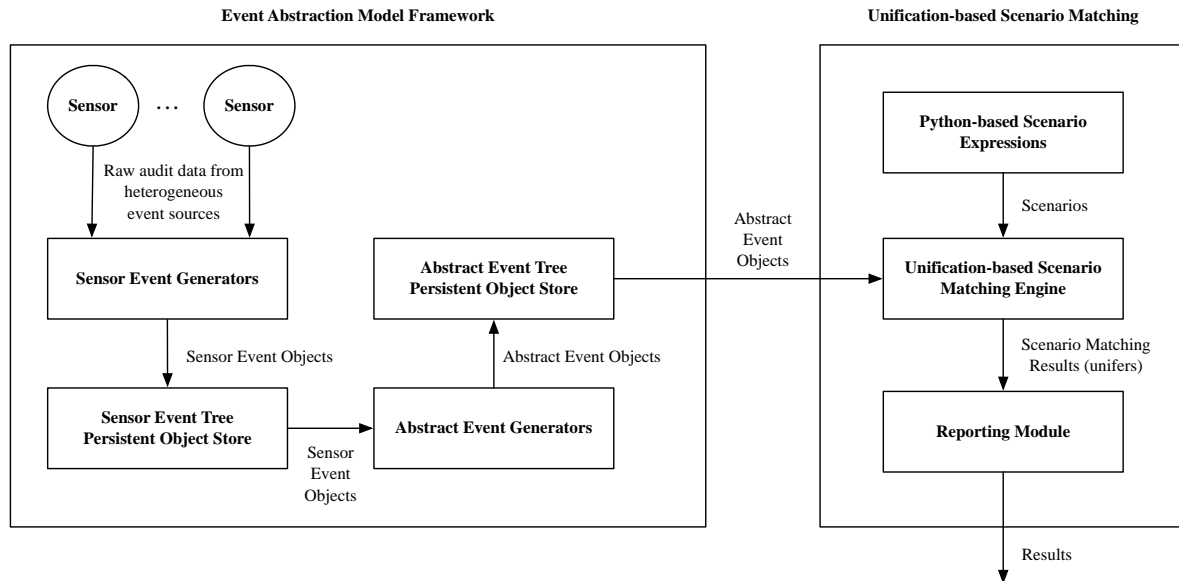


Figure 1: The Prototype Unification-based Multi-Step Scenario Matching

### Composite Scenarios

In our Python-based scenario specification framework, a scenario can incorporate other scenarios (i.e., a composite scenario) and use them to describe attack steps in a multi-step scenario. Composite scenarios are used for two reasons: to represent abstract events and to facilitate signature development. Firstly, for abstract event representations, a scenario can represent a set of concrete events as one abstract event which can then be included by referencing it in other higher order scenarios. Secondly, for facilitating scenario development and management, the scenario composition provides re-usability of existing scenarios. This feature helps decrease time for scenario development and eases scenario management.

### THE PROTOTYPE

We have developed a proof-of-concept implementation of our unification-based multi-step scenario matching framework. The scenario matching framework has been implemented in the Python programming language due to a number of benefits of the language. For instance, the Python language's built-in data structures (dictionaries and lists) facilitate the development of the unification algorithm. Also, some implementations of the Python language allow a Python program to incorporate APIs from other programming languages, e.g., the Jython interpreter allows a Python program to incorporate Java libraries (Bock et al. 2007) and the IronPython project which enables a Python program to call Microsoft .Net libraries (Microsoft Corporation 2007). The Jython interpreter provides two major benefits. Firstly, it allows a Python program to run on any platform that supports the Java Virtual Machine. Secondly, it allows our unification-based scenario matching framework to incorporate the APIs provided by the event abstraction framework (Panichprecha et al. 2006). Therefore, we have chosen the Jython interpreter for implementing our prototype. We have implemented the Python-based scenario specification framework which provides the necessary components for developing scenarios, i.e., variables, operators, and access to abstract events.

Note that it is possible to use the Prolog language to implement the scenario matching framework, since the matching mechanism used in the Prolog language is a unification algorithm. However, it will be difficult and time consuming to develop a set of Prolog programs to interface with the event abstraction framework. Also, we have little-to-no control of the unification algorithm in Prolog. In particular, we need to use the persistent object stores (object-oriented database) provided by the event abstraction model framework but the Prolog interpreter uses its own internal database, to which we have no direct access. All in all, the Python programming language allows quick development of the unification matching framework and more importantly can connect to the AET persistent object store.



## CASE STUDIES AND EVALUATION

This section demonstrates the proof-of-concept implementation of our system. The prototype has been run in an experimental environment, which comprises four machines, i.e., a victim machine (running as a PXE server<sup>2</sup> and a mail server), a PXE-enabled client machine, a Microsoft Windows 2000 machine, and an adversary machine. All machines, except the machine running Microsoft Windows 2000, use the Linux operating system. All machines are connected to the same physical network and allocated in the same subnet.

Due to space limitations, in this paper, we demonstrate two multi-step attacks. All attacks related to the signatures are run in the experimental network, alongside random harmless traffic such as SSH sessions, HTTP traffic, etc. Related logs and network traffic were collected and parsed using SET and AET generators. The unification algorithm was then applied to the generated AET database to identify the attacks post-hoc. Although, these attacks are tested in a controlled environment, they are real attacks and they can be run in real environments.

During the course of the experiment, audit data was collected from corresponding systems and applications i.e., Apache web server logs, Unix system logs, Unix system call logs, and Microsoft Windows security logs. Also, network traffic from the experimental network was captured with tcpdump. All audit data and network traffic data was converted to event objects using the parsers and generators provided by the event abstraction model framework. From the collected data, the generators produce 24,981 sensor event objects and 25,325 abstract event objects. The objects are stored in the SET and AET persistent object store respectively. The abstract event objects from the AET persistent object store are considered to be facts and are used by our unification-based scenario matching framework.

### Masqueraded Preboot Execution Environment Server Scenario

Due to the lack of a host authentication mechanism, the PXE operation is prone to at least two attacks: denial of service attacks and an adversary host masquerading as a PXE server. In the first attack, an adversary launches a number of successful DHCP handshakes, where a newly generated MAC address is used for each request, until the DHCP server's table of allocated IP addresses is full. This attack causes the PXE server to enter a state where it cannot provide IP addresses (denial of service). In the second attack, the adversary can run DHCP and TFTP services on his/her host which serves as a PXE server. By combining these two attacks, the adversary can create a masqueraded PXE server and use it to serve malicious operating system images (for example, images containing backdoors or spyware) to clients. The scenario comprises steps listed below:

1. DHCP no lease event: The error message reported by a DHCP server when the pre-allocated block of IP addresses is exhausted;
2. DHCP offer: The network event which indicates that a DHCP server is offering an IP address to a client. If the IP address of the DHCP server is not the same as the address in step 1, it signifies an anomalous event;
3. TFTP session: This event identifies TFTP communications between a TFTP server and a TFTP client. In this scenario, if the IP address of the TFTP server is not the same as in Step 1, it indicates that a PXE client downloads a bootstrap file namely "boot.msg" from the adversary machine. The download leads to retrieving an operating system image which may be preloaded with backdoors.

The scenario definition to detect this attack is shown in Figure 2. The scenario comprises four methods. Note that this is purely a code readability choice and it illustrates the seamless use of all Python language features in the scenario specification. The details of the four methods are described as follows:

- `detect`: This is the main method of the scenario. This method is executed by our unification-based scenario matching engine;
- `dhcp_no_lease`: This method detects the DHCP no lease event. Two variables `dhcpnolease` and `realDHCPserver` are instantiated with the `DhcpNoLeases` events and the address of DHCP server respectively;
- `dhcp_offer`: This method detects a potential masqueraded DHCP server on line 14 which specifies that the address of the DHCP server is different from `realDHCPserver`. On line 13, we demonstrate the application of our after operator which specify a 2 second timeout between the DHCP offer and DHCP no leases events;

---

<sup>2</sup> PXE (Preboot Execution Environment) is a hybrid of DHCP and TFTP (Intel Corporation 1999).

- `tftp_session`: This method detects a TFTP download session which corresponds to a client downloading an operating system image from the masqueraded PXE server.

```

1  class pxeattackScenario(Scenario):
2      def detect(self):
3          self.dhcp_no_lease()
4          self.dhcp_offer()
5          self.tftp_session()
6
7      def dhcp_no_lease(self):
8          Variable('dhcpnolease') == AET(DhcpNoLeases)
9          Variable('realDHCPserver') == Variable('dhcpnolease').dhcpServerIPAddress
10
11     def dhcp_offer(self):
12         Variable('dhcponoffer') == AET(DhcpOffer)
13         Variable('dhcponoffer').eventTime == after(dhcpnolease.eventTime, 2000)
14         Variable('dhcponoffer').serverIPAddress != Variable('realDHCPserver')
15         Variable('fakeServer') == Variable('dhcponoffer').serverIPAddress
16
17     def tftp_session(self):
18         Variable('tftpssession') == AET(TFTPSession)
19         Variable('tftpssession').eventTime == after(Variable('dhcponoffer').eventTime, 2000)
20         Variable('tftpssession').destinationAddress == Variable('fakeServer')
21         Variable('tftprequest') == AET(TFTPReadRequest)
22         Variable('tftprequest').destinationAddress == Variable('fakeServer')
23         Variable('tftprequest').fileName == containsPattern("boot.msg")

```

Figure 2: Masqueraded Preboot Execution Environment Server Scenario

This scenario demonstrates an application of our Python-based scenario specification framework. The scenario also shows how to specify a sequence of events and timeout using the *after* operator. Corresponding events and their attribute values are stored in six variables. The attack scenario was successfully detected. However, two sets of events have been generated, while there is only one instance of the attack. This is caused by the fact that there are two instances of DHCP no leases events recorded by syslog with the same timestamp due to the well known 1 second resolution of syslog. This problem can be solved by either implementing a simple 'tidy-up' in the AET generators which detects exact duplicate objects or adding an expression which checks for duplicates.

### Sendmail Executing a Shell Scenario

Several versions of Sendmail are vulnerable to buffer overflow attacks. In this example, we demonstrate a scenario specification which detects a buffer overflow attack in Sendmail version 8.11.6 (SecurityFocus and Zalewski 2003). The attack exploits vulnerability in the *prescan* function, where it fails to check the size of e-mail addresses in SMTP headers. We have analysed the exploit code, *sormail.c* from (SecurityFocus and Zalewski 2003), and found that the code exploits the vulnerability and executes a shell with the privilege of the user who runs Sendmail which, in most cases, is run with system administrator privileges.

The signature for this attack is shown in Figure 3. It comprises two scenarios: the scenario which detects that the Sendmail process executes a shell (Figure 3a) and the scenario which detects that Sendmail is executed (Figure 3b). When executed, Scenario 3a, on line 3, invokes Scenario 3b, where the variable `execSendmail` is instantiated with an Execute event. Note that the Execute event is an abstract event which represents program execution independent from the platform. Also, the `sendmailPID` is instantiated with the process ID of *sendmail*. Then, the expression on line 4 of Scenario 3a calls the `start_root_shell` method which instantiates the variable `startingRootShell`. Also, the `start_root_shell` method specifies two constraints on the variable where one of them specifies that the process ID of the *sendmail* must be equal to the variable `processID` which is instantiated in Scenario 3b.

This scenario demonstrates the ability of our Python-based scenario specification framework to invoke a scenario from another scenario (scenario composition). The composition is possible because the unifiers are implemented as a global Python variable. Thus, a variable instantiated in one scenario is accessible from all other scenarios. In this scenario, the variable `sendmailPID` is instantiated in Scenario 3b. Thus, the value is accessible by Scenario 3a. The attack was successfully detected with no false alarms. Since abstract events are used in the scenario, this scenario can detect other attacks which have similar behaviour, i.e., the sendmail process executing a shell.

```

1  class sendmailExecutingShell(Scenario):
2      def detect(self):
3          executingSendmail()
4          self.start_root_shell()
5
6      def start_root_shell(self):
7          Variable('startingRootShell') == AET(ProcessOperationEvent)
8          Variable('startingRootShell').processID == Variable('sendmailPID')
9          Variable('startingRootShell').processName == containsPattern("sh")

```

(a) Sendmail Executing a Shell

```

1  class executingSendmail (Scenario):
2      def detect(self):
3          Variable('execSendmail') == AET(Execute)
4          Variable('execSendmail').newProcessName == containsPattern("sendmail")
5          Variable('sendmailPID') == Variable('execSendmail').processID

```

(b) Executing Sendmail

Figure 3: Sendmail Executing a Shell Scenario

## CONCLUSION AND FUTURE WORK

We have developed a proof-of-concept unification-based multi-step scenario matching framework. The framework uses abstract events to address the heterogeneity of event sources and aims to address the complexity of scenario specification by using a unification algorithm and a Python-based scenario specification framework. The unification algorithm and our scenario specification framework enable scenario writers to specify descriptions in terms of *what* to detect rather than *how* to detect it. By employing the unification algorithm, scenarios are easier to read, write, and understand. In our Python-based scenario specification framework, we have implemented operators and data types which are sufficient for specifying multi-step scenarios as well as single-step scenarios.

Our proof-of-concept framework has been tested with several scenarios which involve events from multiple sources. Due to space limitations, only two scenarios have been demonstrated in this paper. We have demonstrated our Python-based scenario specification framework in specifying multi-step attacks and constructing composite scenarios. The system has successfully detected all instances of the misuse activities in those scenarios.

Our future work plans are to focus on addressing time-related issues. Time related issues are very important for correlating events from heterogeneous sources and multi-step scenario specifications. In the current stage, we correlate events from heterogeneous sources based on the assumption that clock synchronisation of all the event sources is properly implemented. However, this is not always the case. Although clock synchronisation mechanisms are widely available, they are often not properly implemented. In order to address these issues, we are planning to look into employing *time tolerance*. The principle of time tolerance is to use a time range instead of a single point of time. By applying time tolerance to an event, the timestamp is converted into a time range. In addition to time tolerance, our future work will incorporate a standard reporting format, i.e., Intrusion Detection Message Exchange Format (IDMEF) (Debar et al. 2007), into the reporting module.

## REFERENCES

- Jonathon Abbott, Jim Bell, Andrew Clark, Olivier De Vel, and George Mohay. (2006) Automated Recognition of Event Scenarios for Digital Forensics. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing*, Dijon, France.
- Debra Anderson, Thane Frivold, and Alfonso Valdes. (1995) Next-generation Intrusion Detection Expert System (NIDES): A summary. Technical Report SRI-CSL-95-07, SRI International.
- F. Baader and W. Snyder. (2001) Unification theory. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, pages 447–533. Elsevier Science Publishers.
- Finn Bock, Barry Warsaw, Jim Hugunin, and The Jython Development Team. The jython project., Accessed April 2007.

- Nathan Carey, George Mohay, and Andrew Clark. (2003) Attack Signature Matching and Discovery in Systems Employing Heterogeneous IDS. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003)*, pages 245–254, Las Vegas, Nevada.
- Frédéric Cuppens and Rodolphe Ortalo. (2000) LAMBDA: A Language to Model a Database for Detection of Attacks. In *Proceedings of Recent Advances in Intrusion Detection, 3rd International Symposium, RAID 2000*, volume 1907 of *Lecture Notes in Computer Science*, pages 197–216, Toulouse, France, Springer. ISBN 3-540-41085-6.
- H. Debar, D. Curry, and B. Feinstein. (2007) The Intrusion Detection Message Exchange Format (IDMEF). Request for Comments (RFC): 4765.
- S.T. Eckmann, G. Vigna, and R.A. Kemmerer. (2000) STATL: An Attack Language for State-based Intrusion Detection. In *Proceedings of the ACM Workshop on Intrusion Detection Systems*, Athens, Greece.
- K. Ilgun, R.A. Kemmerer, and P.A. Porras. (1995) State Transition Analysis: A rule-based intrusion detection system. *IEEE Transactions on Software Engineering*, 21(3):181–199.
- Intel Corporation. (1999) Preboot execution environment (PXE) specification version 2.1, September 1999.
- Sandeep Kumar (1995). *Classification and Detection of Computer Intrusions*. PhD thesis, Purdue University.
- Jia-Ling Lin, X. Sean Wang, and Sushil Jajodia. (1998) Abstraction-based Misuse Detection: High-level Specifications and Adaptable Strategies. In *The Eleventh Computer Security Foundations Workshop*, pages 190–201, Rockport, MA.
- Ulf Lindqvist and Phillip A Porras. (1999) Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST). In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 146–161, Oakland, California. IEEE Computer Society Press, Los Alamitos, California.
- T. F. Lunt, R. Jagannathan, R. Lee, A. Whitehurst, and S. Listgarten. (1989) Knowledge-based Intrusion Detection. In *Proceedings of the Annual AI Systems in Government Conference*, pages 102–107, Washington, D.C., IEEE Computer Society Press.
- Michael Meier, Niels Bischof, and Thomas Holz. (2002) SHEDEL-A Simple Hierarchical Event Description Language for Specifying Attack Signatures. In *Proceedings of the Security in the Information Society: Visions and Perspectives, IFIP TC11 17th International Conference on Information Security (SEC2002)*, pages 559–572.
- Michael Meier, Sebastian Schmerl, and Hartmut Koenig. (2005) Improving the efficiency of misuse detection. In *Proceedings of the Second Conference on Detection of Intrusion and Malware and Vulnerability Assessment (DIMVA2005)*. Springer Verlag.
- Cédric Michel and Ludovic Mé. (2001) ADELE: An Attack Description Language for Knowledge-based Intrusion Detection. In *Proceedings of the 16th International Conference on Information Security (IFIP/SEC 2001)*, pages 353–365.
- Microsoft Corporation (2007) IronPython, URL <http://www.ironpython.com>.
- Benjamin Morin, Ludovic Mé, Hervé Debar, and Mireille Ducassé. (2002) M2D2: A Formal Data Model for IDS Alert Correlation. In *Proceedings of Recent Advances in Intrusion Detection, 5<sup>th</sup> International Symposium, RAID 2002*, volume 2516 of *Lecture Notes in Computer Science*, pages 115–137, Zurich, Switzerland, Springer. ISBN 3-540-00020-8.
- Abdelaziz Mounji. (1997) *Languages and Tools for Rule-Based Distributed Intrusion Detection*. PhD thesis, University of Namur, Belgium.
- Peng Ning, Sushil Jajodia, and X. Sean Wang. (2002) Design and Implementation of a Decentralized Prototype System for Detecting Distributed Attacks. *Computer Communications, Special Issue on Intrusion Detection Systems*, 25(15): 1374–1391.
- Julien Olivain and Jean Goubault-Larrecq. (2005) The ORCHIDS Intrusion Detection Tool. In Kousha Etessami and Sriram Rajamani, editors, *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, volume 3576 of *Lecture Notes in Computer Science*, pages 286–290, Edinburgh, Scotland, UK, Springer.
- Xinming Ou, Sudhakar Govindavajhala, and Andrew W. Appel. (2005) MulVAL: A Logic-based Network Security Analyzer. In *Proceedings the 14th USENIX Security Symposium*, Baltimore, Maryland.

- Sorot Panichprecha, Jacob Zimmermann, George Mohay, and Andrew Clark. (2006) An Event Abstraction Model for Signature-based Intrusion Detection Systems. In *Proceedings of the 1st Information Security and Computer Forensics (ISCF-2006)*, pages 151–162, Chennai, India. Allied Publishers Pvt. Ltd.
- Phillip A. Porras and Peter G. Neumann. (1997) EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proceedings of the 10th National Information Systems Security Conference*, pages 353–365, Baltimore, Maryland. National Institute of Standards and Technology/National Computer Security Center.
- Jean-Philippe Pouzol and Mireille Ducassé. (2001) From Declarative Signatures to Misuse IDS. In *Proceedings of Recent Advances in Intrusion Detection, 4th International Symposium, RAID 2001*, volume 2212 of *Lecture Notes in Computer Science*, pages 1–21, Davis, CA, USA. Springer. ISBN 3-540-42702-3.
- Python Software Foundation. (2007) Python programming language, URL <http://www.python.org>, Accessed April 2007.
- J. A. Robinson. (1965) A Machine-oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machinery*, 12(1):23–41. ISSN 0004-5411.
- Muriel Roger and Jean Goubault-Larrecq. (2001) Log Auditing Through Model Checking. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW'01)*, pages 220–236, Cape Breton, Nova Scotia, Canada. IEEE Computer Society Press.
- M. M. Sebring, E. Shellhouse, (1988) M. E. Hanna, and R. A. Whitehurst. Expert Systems in Intrusion Detection: A case study. In *Proceedings of the 11th national Computer Security Conference*, pages 74–81. National Institute of Standards and Technology/National computer Security Center.
- SecurityFocus and Michal Zalewski. (2003) Sendmail address prescan memory corruption vulnerability, Bugtraq id: 7230. URL <http://www.securityfocus.com/bid/7230>.
- Jiahai Yang, Peng Ning, X. Sean Wang, and Sushil Jajodia. (2000) CARDS: A Distributed System for Detecting Coordinated Attacks. In *Proceedings of IFIP TC11 the Sixteenth Annual Working Conference on Information Security*, pages 171–180.

## **COPYRIGHT**

[Sorot Panichprecha, Jacob Zimmermann, George Mohay, Andrew Clark] ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## Steganalysis in Computer Forensics

Ahmed Ibrahim  
School of Computer and Information Science  
Edith Cowan University  
aibrahi0@student.ecu.edu.au

### Abstract

*Steganography deals with secrecy and covert communication and today the techniques for countering this in the context of computer forensics has somewhat fallen behind. This paper will discuss on how steganography is used for information hiding and its implications on computer forensics. While this paper is not about recovering hidden information, tools that are used for both steganography and steganalysis is evaluated and identifies the shortcomings that the forensic analysts would face. In doing so this paper urges on what the stakeholders in the field of computer forensics needs to do to keep ahead of criminals who are using such techniques to their advantage and obscure their criminal activities.*

### Keywords

Steganalysis, Steganography, Information Hiding, LSB, Stegdetect, Steghide, Outguess, Chi-Square, Digital Invisible Ink Toolkit

## INTRODUCTION

One of the first widely used method for secure communication was steganography, also referred to as secret writing. A variety of techniques such as the application of invisible ink and masking the secret text inside an inconspicuous text existed during the early days (Pieprzyk, Hardjono & Seberry, 2003). It even dates back to ancient Greeks who practised the art of hiding messages by tattooing onto the shaved heads of messengers. Today, steganography has taken new meaning and is referred to the science of hiding messages in different electronic media such as graphic, audio, and video files (Schneier, 2000).

Besides the reasons for covert communication to maintain secrecy, steganography is also used to protect intellectual property rights using watermarking techniques that embed a digital fingerprint in the media (Silman, 2001). While each of these aforementioned purposes of steganography has its own applications, this paper will be concerned with the former.

Historically, much attention has been given on cryptography to ensure the secrecy of confidential information whether it is for storage or communication, however in recent times, different motivations have led people to pursue even more guaranteed approaches. Krenn (2004) reports that terrorist organisations use steganography to send secret messages using websites and newsgroups according to claims by the United States government. Although there is no substantial evidence supporting these claims, one would wonder why such an approach maybe attractive and realistic for such organisations. Other sources such as Kelley (2001a, 2001b), and McCullagh (2001) have also made similar claims. According to Schneier (2000), the privacy offered by steganography is far beyond that provided by encryption. This is because the goal of steganography is to hide the secret while encryption simply makes the secret unreadable.

New steganographic techniques are being developed and information hiding is becoming more advanced based on the motives of its use (Krenn, 2004). Besides the hype of terrorists using steganography, very recently there has been a case of corporate espionage reported by Phadnis (2007), where confidential information was leaked to a rival firm using steganographic tools that hid the information in music and picture files. Although the perpetrator was caught in this case, it does give an idea of the wide landscape in which steganography can be applied in.

This paper will focus on steganography of graphic or image files. It will describe some technical aspects of steganography used by different tools that are specific to certain types of image files. Following that, steganalysis techniques used to detect the presence of hidden information from the forensic analyst's point of view will be discussed. Finally, the limitations in steganalysis will be presented along with the evaluation of some steganalysis tools.

## **TECHNICAL PERSPECTIVE OF STEGANOGRAPHY**

Steganography may be implemented using a variety of techniques and methods and a steganographic tool may employ any such method or a combination or even variations of such methods. These methods may range from the use of Least Significant Bit (LSB), manipulation of image and compression algorithms, and modifications of image properties such as its luminance (Johnson & Jajodia, 1998).

The use of LSB is the most commonly used technique for image steganography. Such tools are also referred to as image domain tools that manipulate the LSB using bitwise methods (Krenn, 2004). Since this is more like using noise to hide information in the LSB, small variations in the LSB are unnoticeable to the human eye (Wayner, 2002). However, one major limitation in the use of LSB is the amount of usable space to hide the secret message, thus a suitable cover image is essential. If the cover image does not satisfy the capacity requirements to hide the data, the steganographic image would appear to be suspicious (Krenn, 2004).

Furthermore, the success is also dependent on a reliable compression algorithm to ensure that the hidden message is not lost after the transformation (Krenn, 2004). According to Silman (2001), the most commonly used compression algorithms are Windows Bitmap (BMP), Graphic Interchange Format (GIF), and Joint Photographic Experts Group (JPEG). When LSB method is used, lossless compression algorithms such as BMP and GIF are preferable. This is because lossy compression algorithms such as JPEG are mainly used to save on storage space due to the fact that the compression gets rid of unwanted noise, limiting the amount of space that can be used for steganography (Wayner, 2002).

For lossy algorithms like JPEG, usually when the image is 24bits or grayscale, a more robust approach is to use masking/filtering where the luminance of parts of the image are modified. A more complex way to hide information, particularly in JPEG files is to use Discrete Cosine Transformations (DCT). DCT is also used by the JPEG compression algorithm, and the resulting steganographic image does not have any detectable visible changes as the technique makes use of the frequency domain of the image (Krenn, 2004).

## **STEGANALYSIS TECHNIQUES**

Steganalysis is mostly about the discovery (Silman, 2001) and simply identifying the existence of a hidden message (Wayner, 2002).

Some literature such as Silman (2001) also refer to steganalysis as the destruction of the hidden information. And it can be done even if there is no knowledge of the existence of the hidden information (Krenn, 2004). However, forensics is about finding information and not destroying it. But it may indeed be reasonable to do so in other contexts. This is mainly because recovering hidden information can become rather complex and sometimes impossible without knowing which tool or technique was used for the steganography (Johnson & Jajodia, 1998). Even if the steganographic tool was somehow discovered, extracting the hidden information can prove to be rather daunting as most algorithms employ cryptographic techniques to scramble the secret message when it is embedded (Wayner, 2002). Although it use to be possible in classical steganographic systems where the security lies in the secrecy of the encoding scheme (Provos & Honeyman, 2002), modern systems have adopted Kerchoff's principle of cryptography, and the security depends on the secret key that is used to encode and not the encoding scheme (Provos & Honeyman, 2002; Krenn, 2004).

Therefore the challenge of recovering the hidden information remains for the forensic investigator, but the first step would be to identify suspicious articles with hidden information. According to Silman (2001) steganalysis attacks depend on the information that is available to the steganalyst such as:

- when only the steganographic object is available
- when the steganographic algorithm is known and the steganographic object is available
- when the steganographic object and the original cover object is available
- when both the steganographic and the cover object is available and the steganographic algorithm is known

If both the steganographic object and the cover object is available, checking and comparing file attributes such as size, file format, last modified timestamps, and colour palette can give some clues whether some information has been hidden (Krenn, 2004). However in forensic investigations, the most likely situation the investigator

will be in is when only the steganographic object is available, and that is assuming if an object can be classified as a steganographic object in the first place. In such a situation it would not be possible to make comparisons with attributes of the original cover image and the steganographic image such as size as mentioned by (Silman, 2001).

Steganographic systems generally leave detectable traces (Provos & Honeyman, 2002). This is because often the process alters media properties and introduces degradations or abnormal characteristics that can be used as steganographic signatures for detection (Johnson & Jajodia, 1998). Although these cannot be detected by the human eye due to careful application of steganography, the signatures left can be electronically discovered (Silman, 2001). These signatures can also be used to identify the steganographic tools and techniques (Johnson & Jajodia, 1998), thereby aiding the investigator in the retrieval of the hidden information.

A commonly used steganographic technique is to make use of the LSB data, because the LSB data mostly appears random to a human observer although it contains hidden patterns (Wayner, 2002). Statistical analysis of the LSB data is a widely used method for detecting these patterns (Krenn, 2004). One of the most common pattern is a correlation between the High-Order Bits and the LSB which is often introduced by the hardware, such as the camera, used to generate the original data (Wayner, 2002). This attack is mostly successful because most of the steganographic algorithms operate under the assumption that the LSB is random, however statistical analysis can detect changes made to the LSB especially in the case of encrypted messages as it is more random and has higher entropy (Krenn, 2004).

Comparisons and analysis of numerous original and steganographic images can reveal anomalies and patterns can be classified. These can be detected due to factors such as unusual sorting of colour palettes, relationship between colours in colour indexes, and in exaggerated noise (Johnson & Jajodia, 1998).

## **LIMITATIONS IN STEGANALYSIS**

Although there are some techniques that can detect steganography there are major problems that steganalysts face. Even if there are noticeable distortions and noise, predictable patterns cannot always be detected. Some steganographic techniques are particularly difficult to detect without the original image (Johnson & Jajodia, 1998). And in most cases, it is highly unlikely that a forensic investigator will be conveniently presented with the steganographic and original image.

To avoid detection, some steganographic technique spread the information and the diffusion makes it harder and less suspicious for detection. Some steganographic tools even use Random Number Generators (RNG) to make the LSB choosing process more random and to distribute the distortions throughout the file (Wayner, 2002)

According to Wayner (2002) another approach in defending against statistical attacks is not to saturate the cover image by packing in too much data, thereby leaving most of the LSB untouched hence making it highly indistinguishable from an untouched pure file.

Even until today, most steganalysis techniques are based on visual attacks and methods beyond this are being explored. Unfortunately a general steganalysis technique has not been devised (Johnson & Jajodia, 1998). While visual attacks are more prominent, JPEG images, which is one of the most commonly distributed type of image format, the steganographic modifications take place in the frequency domain. This means that this type of steganography is not susceptible to visual attacks unlike in image formats such as GIF images where the modifications happen in the spatial domain (Provos & Honeyman, 2002).

In order to verify the claims about terrorists using the Internet to distribute secrets using steganography, Niels Provos created a cluster that scans images from newsgroups to detect steganographic content (Krenn, 2004). In Provos' and Honeyman's (2002) work, upon investigating two million images from particular sources in the Internet, they were unable to find a genuine message and suggest the following explanations:

- steganography is not significantly used on the Internet
- the sources of the images analysed are not used for steganographic communication
- the steganographic systems detectable by the study are not being used
- strong passwords, not susceptible to dictionary attacks have been used by all steganographic systems users.

For reasons that no hidden messages were discovered, it raises the question of the practicality of such detection systems (Krenn, 2004).



## EVALUATION OF STEGANALYSIS TOOLS

In order to evaluate the steganalysis tools, it is essential that the whole process is forensically sound to ensure the validity of the findings. Therefore, the following are the steps that will be followed throughout the process:

1. obtain the steganographic and steganalysis tools
2. verify the tools (to ensure the tools is doing what it claims)
3. obtain cover images, and generate MD5 hashes
4. apply steganalysis on cover images, and generate MD5 hashes
5. generate steganographic images, and generate MD5 hashes
6. apply steganalysis on the steganographic image, and generate MD5 hashes

In each of the steps where the cover images or the steganographic images are involved, MD5 hashes have been used to verify whether the image has changed in any sense.

### Obtaining the tools

To keep the evaluation as realistic as possible, and the circumstances applicable to a wide range of users, the steganographic tools have been chosen based on how easy it is to obtain it, and the type of images it deals with. The tools that will be used are Steghide (Steghide Website, 2003), Outguess (Provos, 2004), and Digital Invisible Ink Toolkit (DIIT) (Hempstalk, 2005). All these tools are freely available, including their source codes, and can be downloaded by anyone from the Internet. Which means that anyone with programming experience can even make changes to the steganographic algorithms used. They can also be used on both Windows and Linux platforms. Therefore, this covers a wide range of users, who can make use of these tools. The following table (table 1) gives their version information and the output steganographic image format.

Table 4: Steganographic Tools

Tool	Version	Output Image Format	Platform	Source
Steghide	0.5.1-8	JPEG, BMP	Windows & Linux	(Steghide Website, 2003)
Outguess	1:0.2-6	JPEG, PNM	Windows & Linux	(Provos, 2004)
Digital Invisible Ink Toolkit (DIIT)	1.5	PNG, BMP	Windows, Linux & Mac OS	(Hempstalk, 2005)

Steghide can be used to hide data in JPEG and BMP image formats. It uses a graph-theoretic approach to perform the steganography (Steghide Website, 2003). More detailed descriptions of the tool can be found in the documentations available on Steghide Website (2003). Outguess performs steganography by inserting the hidden information into the redundant bits of the cover image. Its steganographic technique is able to protect the steganographic JPEG image from statistical attacks based on frequency counts (Provos, 2004). DIIT is a steganographic tool that allows to use four highly customisable algorithms to perform the steganography. The algorithms are BlindHide, HideSeek, FilterFirst, and BattleSteg (Hempstalk, 2005).

Similar to the steganographic tools, the choice of steganalysis tools were made based on its availability as a free software, and also its approach. It is important to note that there are other more expensive commercial steganalysis tools, such as StegAnalyzerSS (Steganography Analyzer Signature Scanner) by Backbone Security (2007). According to Backbone Security (2007), StegAnalyzerSS scans for unique hexadecimal byte patterns or known signature patterns in order to detect the steganography. But for this evaluation, two freely available tools were chosen. The first one is Stegdetect developed by Niels Provos and it can detect jsteg, jphide, invisible secrets, outguess 01.3b, F5 (header analysis), appendiX and camouflage steganographic schemes (Provos, 2004). This tool will be used on JPEG images generated by Steghide and Outguess. The next steganalysis tool was developed by Guillermito to implement the chi-square analysis to perform a statistical attack and detect any hidden messages (Guillermito, 2004). It will be used on the BMP images generated by Steghide and DIIT.

Table 5: Steganalysis Tools

Tool	Version	Steganalysis Attack
Stegdetect	0.6	JPEG image by Steghide and Outguess
Chi-Square	0.1	BMP image by Steghide and DIIT

The machine used to do the evaluation is based on Debian Linux. Therefore all the mentioned steganographic tools and steganalysis tools were obtained using the Debian repositories available on the Internet except for the DIIT and Chi-Square. DIIT is provided as a JAR package from the website and Chi-Square is provided in EXE form compiled to run on Windows platform. However it was possible to use Wine to run the EXE file in the Linux environment.



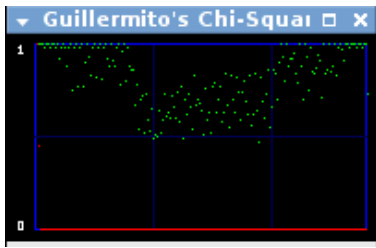

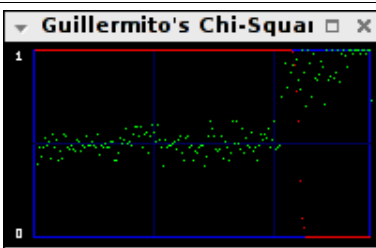
### Verifying the tools

In order to verify that the tools are doing what they claim they can do, it has to be verified using test data. The steganographic tools test, verifies that the tools are able to hide a secret message in a cover image, and is able to retrieve the exact message using that tool, which would confirm the steganographic process.

In order to verify the steganographic tools, a sample text file was first created. The MD5 of the files was generated and recorded for later reference. All three steganographic tools were used to hide this text file in image files, and these tools were used to retrieve the hidden text file. The retrieved text files from each tool were used to generate their MD5 hashes, and was compared with the first MD5 hash that was generated. The results showed that they are all identical and was able to retrieve the exact data, that was hidden in the first place.

The steganalysis tools test, verifies whether it can detect the presence of a hidden message, using test steganographic images. The test steganographic images to test Stegdetect was obtained from the Stegdetect 0.6 source package available from the (Provos, 2004), and for the Chi-Square test, the images were obtained from (Guillermi, 2004). The source of each test steganographic image is the same source as the actual steganographic tool, therefore this guarantees the authenticity of the test images.

Table 6: Steganalysis Tools Verification Tests



#	Image	MD5 hash	Output
1	 (testimg.jpg found in <a href="http://www.outguess.org/stegdetect-0.6.tar.gz">http://www.outguess.org/stegdetect-0.6.tar.gz</a> )	<b>before:</b> 01a77444369f4de7c7e3aea597f30324  <b>after:</b> 01a77444369f4de7c7e3aea597f30324	\$ stegdetect testimg.jpg testimg.jpg : jphide(***)
2	 ( <a href="http://www.guillermi2.net/stegano/tools/googlemondrian.bmp">http://www.guillermi2.net/stegano/tools/googlemondrian.bmp</a> )	<b>before:</b> c4d2fc1028910ba53841ddaeb435d05e  <b>after:</b> c4d2fc1028910ba53841ddaeb435d05e	
3	 ( <a href="http://www.guillermi2.net/stegano/tools/googlemondrian_02k.bmp">http://www.guillermi2.net/stegano/tools/googlemondrian_02k.bmp</a> )	<b>before:</b> b2b15002f0b23b741c84c0bb0fdf53f7  <b>after:</b> b2b15002f0b23b741c84c0bb0fdf53f7	

In the above table (table 3), the first entry shows the Stegdetect test on the image testing.jpg, which was obtained from the stegdetect-0.6.tar.gz source package and the test shows that jphide was used to embed a secret in it. The second and third test shows the Chi-Square test. Since the output is in the form of a graph, the test was carried out on a plain image with nothing embedded in it, which is the second entry. The third entry is the same image with 2KB of data embedded in it. According to Guillermito (2004), the red line (bottom line in entry 2 output) is the result of the chi-square test and if it is close to 1, then it shows that there is a high probability of an embedded message. If the green curve (collection of dots) is close to 0.5, then again it shows that there is a random message embedded. And lastly, the vertical blue lines indicate intervals of 1KB. As it can be seen, In entry 2, the red line is on 0 (zero) and the green curve is spread out. In entry 3, the red line is on 1 until the 2KB interval and the green curve is also close to 0.5 until the 2KB interval. These tests verify that both the steganalysis tools are working.

### Obtain cover image

The cover image was first acquired using a Digital Camera. The image was originally in JPEG format in 680x480 resolution. Since a BMP image was also required for the evaluation, a second image in BMP format was generated using the same JPEG image using Gimp. Once both the cover images have been obtained, the MD5 hashes for both the images were created. As it can be seen in the following table (table 4), they are different, because they use different compression algorithms.

Table 7: Cover Images

flower.jpg	flower.bmp
	
<b>MD5:</b> f34cc0ae3fb2a1c9be2faa674a2812d0	<b>MD5:</b> de24e73fd06702f577495af16eea7ddb

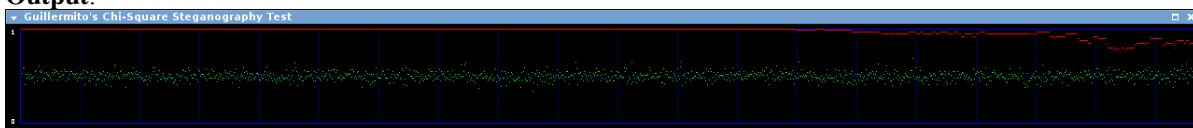
### Steganalysis of cover image

Steganalysis is done on the cover images, to ensure that there are no hidden messages embedded in the first place. Even though, in this particular case, there is knowledge that there are no messages hidden, it is a necessary step to make the process forensically sound.

Table 8: Stegdetect Test on flower.jpg

<b>MD5 after test:</b> f34cc0ae3fb2a1c9be2faa674a2812d0
<b>Output:</b> \$ stegdetect flower.jpg flower.jpg : negative

Table 9: Chi-Square Test on flower.bmp


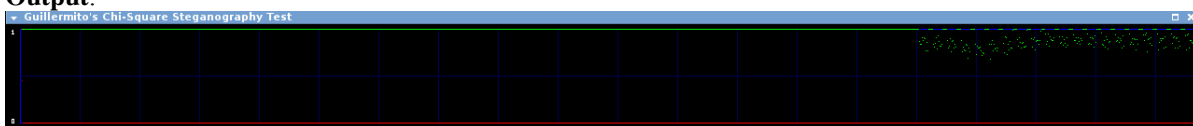
<b>MD5 after test:</b> de24e73fd06702f577495af16eea7ddb
<b>Output:</b> 

As seen in table 5, the Stegdetect test is negative for any embedded messages. However the Chi-Square test output shows all the characteristics of a graph that would show high probability of random embedded message. This is definitely a false positive, therefore it would be pointless to continue with this BMP image. The output might have been because the original BMP image contains too much random data. Another BMP image is needed and to avoid the same situation, the BMP image was created from scratch using Gimp.

As it can be seen in table 7, the new BMP image constructed from scratch satisfies the test as a clean image with no embedded messages. Although it maybe unlikely that an actual user would go through this process of creating an image from scratch in order to hide a message, this has been done for the sake of this evaluation to proceed by using a valid cover image, with respect to the Chi-Square test.

Once the steganalysis was carried out on the cover images, MD5 hashes were generated for each image and compared with the original hashes. The comparisons reveal that the steganalysis process did not alter the image as the hashes match.

Table 10: New BMP cover image

 <p>newbmp.bmp</p>	
<b>MD5 before test:</b> 9b190be2345100aebad2493e0d915522	
<b>Output:</b> 	
<b>MD5 after test:</b> 9b190be2345100aebad2493e0d915522	

### Generate steganographic image

Each image was embedded with and without passwords for the encryption of the hidden message. The hidden message in the following cases are text files of different sizes. Different sizes have been used due to the different input requirements of the steganographic tools. Once the steganographic image was created, MD5 hashes of each image reveal that they have indeed been altered by the steganographic process.

```
$ steghide embed -cf flower.jpg -ef msg_small.txt -sf
steghide_np_flower.jpg
Enter passphrase:
Re-Enter passphrase:
embedding "msg2.txt" in "flower.jpg"... done
writing stego file "steghide_np_flower.jpg"... done
```

Figure 4: Steghide Process for JPEG images

The above figure (figure 1) shows the process of creating a steganographic image “steghide\_np\_flower.jpg” by using Steghide, using the cover image “flower.jpg”, and secret message in “msg\_small.txt” file. No passphrases were entered. Another steganographic image was created by using a passphrase with the above process, to generate “steghide\_wp\_flower.jpg”. The following figure (figure 2) shows the Steghide process for generating the BMP file. Similar to the previous method, “steghide\_np\_newbmp.bmp” was generate without a passphrase, where as “steghide\_wp\_newbmp.bmp” was generated using a passphrase.

```
$ steghide embed -cf newbmp.bmp -ef msg_big.txt -sf
outguess_np_newbmp.bmp
Enter passphrase:
Re-Enter passphrase:
embedding "msg_big.txt" in "newbmp.bmp"... done
writing stego file "steghide_np_newbmp.bmp"... done
```

Figure 5: Steghide Process for BMP images

The following (figure 3) is the process of generating the steganographic image using Outguess. The same cycle was followed by generating with and without passphrases as “outguess\_wp\_flower.jpg” and “outguess\_np\_flower.jpg” respectively.

```

Reading flower.jpg....
JPEG compression quality set to 75
Extracting usable bits: 36976 bits
Correctable message size: 12962 bits, 35.06%
Encoded 'msg_smallest.txt': 7912 bits, 989 bytes
Finding best embedding...
  0: 3916(49.3%)[49.5%], bias 4169(1.06), saved: 5,
total: 10.59%
  5: 3876(48.8%)[49.0%], bias 4125(1.06), saved: 10,
total: 10.48%
 30: 3882(48.9%)[49.1%], bias 4113(1.06), saved: 9,
total: 10.50%
 47: 3898(49.1%)[49.3%], bias 4083(1.05), saved: 7,
total: 10.54%
 56: 3901(49.1%)[49.3%], bias 4048(1.04), saved: 6,
total: 10.55%
 99: 3883(48.9%)[49.1%], bias 3981(1.03), saved: 9,
total: 10.50%
99, 7864: Embedding data: 7912 in 36976
Bits embedded: 7944, changed: 3883(48.9%)[49.1%], bias: 3981,
tot: 36927, skip: 28983
Foiling statistics: corrections: 1788, failed: 3, offset:
92.138055 +- 203.759058
Total bits changed: 7864 (change 3883 + bias 3981)
Storing bitmap into data...
Writing outguess_np_flower.jpg....

```

Figure 6: Outguess process

The next tool that is used is DIIT. The following is a screenshot (figure 4) of the GUI of the tool. As it can be seen, it also provides an option to enter a password for encryption. Like the previous processes, in this process also, steganographic images were created with and without passwords as “diit\_wp\_newbmp.bmp” and “diit\_np\_newbmp.bmp” respectively.

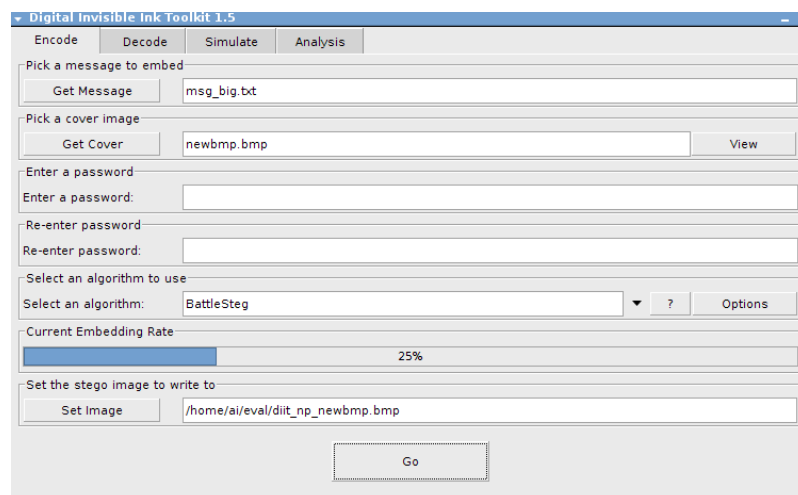


Figure 7: DIIT Screenshot

After each process, the MD5 hash was generated for original and the steganographic images. The following figure (figure 5) shows the MD5 hashes for all the images involved the Steghide, Outguess, and DIIT steganographic processes.

f34cc0ae3fb2a1c9be2faa674a2812d0 flower.jpg

4c2a9fb3860b299460a4be912a806437	steghide_np_flower.jpg
cdac07608cdf45f1e62ab96086dc362e	steghide_wp_flower.jpg
e7cd6d440badb0404db9e02f1c2dd9c6	outguess_np_flower.jpg
bbd68076246b513669e94180ee02ee5b	outguess_wp_flower.jpg
9b190be2345100aebad2493e0d915522	newbmp.bmp
54b03c5c48e374f697abb2809c4d3222	steghide_np_newbmp.bmp
a463186d7cbc0bfc1f1af13f2117c016	steghide_wp_newbmp.bmp
01f4acd389266a01a07acba0153108a6	diit_np_newbmp.bmp
fc914686e06b46e5672a1fdaea72c235	diit_wp_newbmp.bmp

Figure 8: MD5 hashes

As it can be seen, the original images (flower.jpg and newbmp.bmp) were not altered during the steganographic process and each of the resulting steganographic image is different from each other.

### Steganalysis of steganographic images

The steganalysis is carried out using Stegdetect and Chi-Square. The following (figure 6) shows the result of Stegdetect on all the JPEG images. Once Stegdetect was carried out, MD5 hashes were generated for each JPEG image, but revealed that there were no changes. The result of the Stegdetect shows that, it was unable to detect the steganography of Steghide and Outguess.

```
outguess_np_flower.jpg : negative
outguess_wp_flower.jpg : negative
steghide_np_flower.jpg : negative
steghide_wp_flower.jpg : negative
```

Figure 9: Stegdetect Results

The following (figure 7-10) are the results of Chi-Square analysis on the BMP images. As it can be seen, the results cannot confirm the presence of a hidden message.

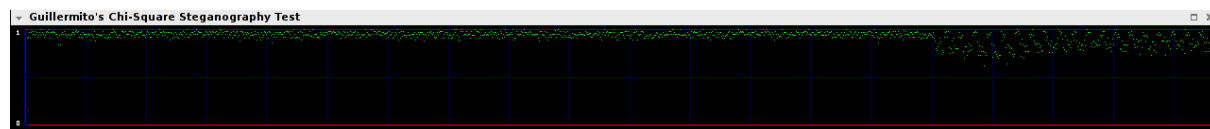


Figure 10: Chi-Square Result for Steghide (no passphrase)

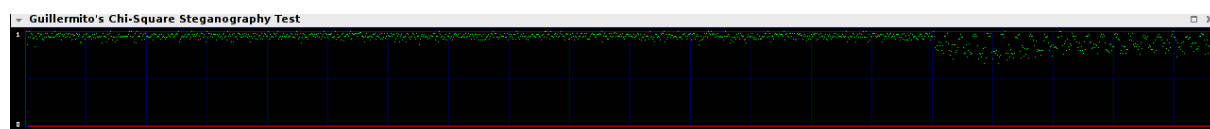


Figure 11: Chi-Square Result for Steghide (with passphrase)

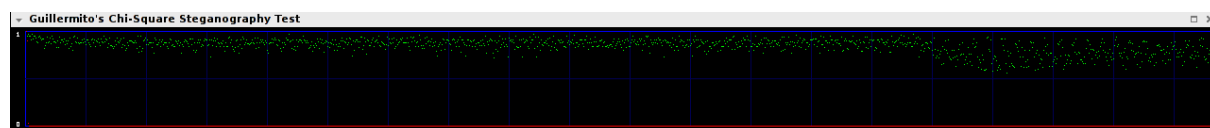


Figure 12: Chi-Square Result for DIIT (no passphrase)

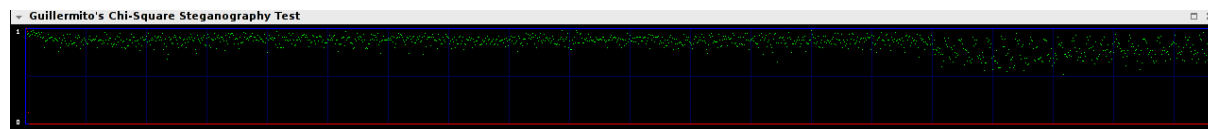


Figure 13: Chi-Square Result for DIIT (with passphrase)

None of the tools were able to positively identify the existence of hidden information. The messages were embedded in plain and encrypted form. The MD5 hashes show that the resulting images are different when the encryption is used by providing the passphrase during the steganographic process. However, the steganalysis was unable to recognise the existence of the hidden messages.

## CONCLUSION

From the information that has been presented in this paper, it would be difficult to come to a firm conclusion regarding the state of steganalysis tools. Since it is not an extensive research with large amounts of data sets, it would be arguable if such a conclusion is made. However, it can be said that steganalysis is not as straight forward or convenient as steganography. This translates to a great deal of advantage for those who hide secrets using steganography. And a huge disadvantage for the forensic analysts, who has the challenge of detecting and retrieving the hidden messages without destroying it.

Furthermore, it is also apparent that steganalysis fails when such tools are applied to detect steganographic techniques it wasn't designed to detect. It has also been observed that, false positives are also possible when generic techniques are used to detect factors such as randomness of LSB. Perhaps with more data and research, these tools can be enhanced to be more effective and accurate.

As steganographic tools are easily available in different varieties for anyone who intend to keep or communicate secrets, and with the emerging signs of its use in different arenas, forensic analysts face new challenges in their investigations. Criminals would indeed exploit every opportunity available to ensure the success of their plans. This could involve mass distribution of terror plans over the Internet or even more covert means of transmitting and storing illegal content on a portable storage devices.

Whatever the case maybe, it cannot be denied that there is a need to be concerned about the current state of forensic knowledge and tools available in this particular area of computer science. Perhaps it is because of a lack of interest among academics and other stakeholders due to less encouraging results of current research. Or it maybe because there is an unrealistic expectation of a magic pill to this problem. Nevertheless, as Johnson and Jajodia (1998) has mentioned, developments in steganalysis techniques will be extremely useful for law enforcement authorities in computer forensics, and an urgently needed development.

## REFERENCES

- Backbone Security (2007). *SARC – steganography analysis and research center*. Retrieved October 7, 2007, from <http://www.sarc-wv.com/stegalyzers.aspx>
- Guillermi (2004). *Steganography: a few tools to discover hidden data*. Retrieved September 29, 2007, from <http://www.guillermi2.net/stegano/tools/index.html>
- Hempstalk, K. (2005). *Digital Invisible Ink Toolkit*. Retrieved September 28, 2007, from <http://diit.sourceforge.net>
- Johnson, N. F., & Jajodia, S. (1998). Steganalysis: the investigation of hidden information. *Proceedings of the 1998 IEEE Information Technology Conference*. (pp. 113-116). Syracuse, New York, USA.
- Kelley, J. (2001a). *Terrorist instructions hidden online*. Retrieved September 14, 2007, from <http://www.usatoday.com/tech/news/2001-02-05-binladen-side.htm>
- Kelley, J. (2001b). *Terror groups hide behind web encryption*. Retrieved September 14, 2007, from <http://www.usatoday.com/tech/news/2001-02-05-binladen.htm>
- Krenn, R. (2004). *Steganography and steganalysis*. Retrieved September 8, 2007, from <http://www.krenn.nl/univ/cry/steg/article.pdf>
- McCullagh, D. (2001). *Secret messages come in .wavs*. Retrieved September 14, 2007, from <http://www.wired.com/politics/law/news/2001/02/41861>

- Phadnis, S. P. (2007). *Data leak: cyber sherlocks outwit hackers*. Retrieved October 13, 2007, from [http://economictimes.indiatimes.com/Infotech/Data\\_leak\\_Cyber\\_sherlocks\\_outwit\\_hackers/articleshow/2451089.cms](http://economictimes.indiatimes.com/Infotech/Data_leak_Cyber_sherlocks_outwit_hackers/articleshow/2451089.cms)
- Pieprzyk, J., Hardjono, T., & Seberry, J. (2003). *Fundamentals of computer security*. Berlin: Springer.
- Provos, N., & Honeyman, P. (2002). *Detecting steganographic content on the internet*. Retrieved September 2, 2007, from <http://www.citi.umich.edu/u/provos/papers/detecting.pdf>
- Provos, N. (2004). *OutGuess - universal steganography*. Retrieved September 30, 2007, from <http://www.outguess.org>
- Schneier, B. (2000). *Secrets & lies: digital security in a networked world*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Silman, J. (2001). *Steganography and steganalysis: an overview*. Retrieved September, 8, 2007, from [http://www.sans.org/reading\\_room/whitepapers/steganography/553.php](http://www.sans.org/reading_room/whitepapers/steganography/553.php)
- Steghide Website (2003). *Steghide*. Retrieved September 28, 2007, from <http://steghide.sourceforge.net>
- Wayner, P. (2002). *Disappearing cryptography information hiding: steganography & watermarking* (2nd ed.). Amsterdam: Morgan Kaufmann Publishers.

## **COPYRIGHT**

Ahmed Ibrahim ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.



## Managing Digital Forensic Knowledge An Applied Approach

David P. Biros and Mark Weiser, Oklahoma State University and Edith Cowan University  
david.biros@okstate.edu, weiser@okstate.edu

John Whitfield, Air Force Institute of Technology  
John.Whitfield@afit.edu

### Abstract

*The science of digital forensics is continually changing as technological advances are made and new digital devices are developed. This environment forces analysts to regularly extend their skills with training and frequent research to develop new and admissible techniques. Unfortunately, the same and similar methods are re-discovered by other analysts who are unaware of earlier peer efforts. The situation is aggravated by a nearly universal backlog in qualified digital forensics facilities. This leaves little time for communication between analysts even within a single agency.*

*To address these issues and facilitate an increase in efficiency across all law enforcement agencies, we apply the lessons of knowledge management to digital forensics and extend them with special characteristics required by the law enforcement profession. The result is the development of the National Repository of Digital Forensic Intelligence. This system has been implemented in the largest accredited digital forensics lab in the world and is currently being extended to many other local, state, and federal agencies to increase effectiveness and efficiency among analysts.*

### INTRODUCTION

Rarely does a day pass that we are not made aware of a significant computer security breach that potentially puts our private information, finances, or even personal security at risk. The anonymity and freedom that the public demands in digital interactions is the same anonymity and freedom exploited by those who wish to do us harm. No well-connected computer is perfectly secure, so there is a balancing act between ease of use for legitimate transactions and security against illegitimate actions. Because technical counter-measures and training are insufficient to offset breaches, existing and new laws have been applied to the digital world to allow legal pursuit of those who seek to violate our digital worlds.

Digital devices may be the object of a crime, or an instrument to commit a criminal act. More often than not, however, digital evidence is being brought to bear in crimes that are not computer-based in any way. Even beat cops are being trained to ensure that digital evidence is preserved and seized in a manner acceptable to the courts. The increased awareness of the value of this evidence has resulted in a greater demand for forensic analysts and a need for them to work more efficiently and effectively. The bulk of leading-edge digital forensic knowledge is held in the minds of the analysts. The combination of a growing backlog and requirement for continuous innovation to keep up, however, has left little time for collaboration between examiners. Managing their knowledge in a way that limits redundant creation and allows sharing and efficient use among law enforcement agencies is the only way that these critical techniques can be properly leveraged.

This paper frames digital forensics as a knowledge management issue and applies some special characteristics of law enforcement. It then describes and updates progress on an ongoing collaboration between Oklahoma State University's Center for Telecommunications and Network Security (CTANS) and the United States' Defense Cyber Crime Center (DC3) to develop the National Repository of Digital Forensic Intelligence (NRDFI). This system, now in beta testing with multiple agencies, has the potential to rapidly and centrally make available forensic discoveries throughout the DOD and law enforcement, without exposing those techniques to those who could exploit them for criminal activity. The goal is to provide a conduit for sensitive and relevant information interchange in a manner tailored to the needs of forensic analysts and law enforcement.

## **APPLICATION OF KNOWLEDGE MANAGEMENT**

Digital Forensics is defined (Biros and Weiser, 2006) as “Scientific knowledge and methods applied to the identification, collection, preservation, examination, and analysis of information stored or transmitted in binary form in a manner acceptable for application in legal matters.” Meaning that all facets of identification, collection, preservation, examination, and analysis, must be verifiable and repeatable, and the results generally accepted by the digital forensic community. The rapidly changing nature of digital technology makes “general acceptance” difficult to attain. Reusing discoveries from other law enforcement agencies that have been successfully presented and accepted in a court is critical to gaining legal admissibility of the techniques.

Although law enforcement has some special characteristics, the sharing of knowledge between experts is important in many organizations. Knowledge and the ability to marshal and deploy knowledge across an organization are key factors for an organization’s competitive advantage (Vizcaino, Soto, Portillo, & Piattini, 2007; Vouros, 2003; Teece, 1998; Tsai & Ghoshal, 1999). In order for organizations to remain competitive, knowledge management systems (KMSs) have been designed to manage an organization’s knowledge (Vizcaino et al., 2007). In light of this, knowledge management systems are becoming ubiquitous in today’s corporations (Davenport & Prusak, 1998). KMSs are tools that affect the management of knowledge and are manifested in a variety of implementations including document repositories, expertise databases, discussion lists, and context-specific retrieval systems incorporating collaborative filtering technologies (Hahn & Subramani, 2000). The main objective of a KMS is to support the creation, transfer, and application of knowledge in organizations (Bock, Zmud, Kim, & Lee, 2005; Kahkanhalli, Tan, & Wei, 2005). Alavi and Leidner (2001) defined a KMS as an information technology based system developed to support and enhance the processes of knowledge creation, storage/retrieval, transfer, and application.

KMS encompass a variety of technology-based initiatives such as the creation of databases of experts and expertise profiling and the hardwiring of social networks to aid access to resources of non-colocated individuals (Davenport & Prusak, 1998; Pickering & King, 1995). The primary focus of many of the KMS efforts has been on developing new applications of information technology such as data warehousing and document repositories linked to search engines to support the digital capture, storage, retrieval and distribution of an organization’s explicitly documented knowledge (Hahn & Subramani, 2000). Today’s KMSs store vast amounts of information and serve a variety of issues such as the creation and acquisition of knowledge in organizations, the storage and retrieval of available knowledge, and the sharing of knowledge among individuals and organizations, while they address the needs of an individual to interpret and reason about collective knowledge (Tiwana, 2000; Fahei, Srivastava, & Smith, 2001; Shin, Holden, & Schmidt, 2001).

## **KNOWLEDGE MANAGEMENT SYSTEM ACCEPTABILITY**

Recent literature in the information systems field extols the virtue of knowledge management systems as the next state-of-the-art innovation pertinent to business practitioners (Adams & Lamont, 2003). For example, researchers such as Davenport & Prusak (1988), Johnson (1988), Zack (1999), and Alvai & Leidner (2001) emphasize the criticality associated with corporations developing organizational-wide KMSs to create and maintain competitive advantages in increasingly dynamic business environments (Adams & Lamont, 2003).

A number of organizations have implemented KMSs only to find that employees do not use them (Hansen & Von, 2001). Issues such as motivating employees to share knowledge (Wasko & Faraj, 2005), creating positive attitudes around knowledge sharing (Bock, Zmud, Kim, & Lee, 2005), and trust (McEvily, Perronne, & Zaheer, 2003) continue to be addressed in research and in practice. As with any other information system implementation, the success of these systems inevitably begins with the individual; individual acceptance and usage are critical (Money & Turner, 2004). With continuing business resource investments, understanding and creating conditions under which information systems will be accepted and used in human organizations remains a high priority within the research community (Vankatesh & Davis, 2000). However, understanding why individuals accept or reject systems has proven to be one of the most challenging issues in information systems research (Doll, Hendrickson, & Xiandong, 1998).

User acceptance of information systems and usage are unquestionably crucial factors in the ultimate determination of information systems success, because information systems that are not used are of little value (Mathieson, Peacock, & Chin, 2001). Similarly, creating knowledge management systems likely to be accepted by target users is critical to harnessing a new system’s potential (Lin, Hu, Chen, & Schroeder, 2004). For present purposes, user acceptance is defined as the demonstrable willingness within a user to employ information technology for the tasks it is designed to support (Dillon & Morris, 1996). User participation in system design is seen as a key factor to achieving acceptance (Mathieson, 1991). Many believe that systems developed with user participation will better match user requirements and capabilities than systems designed

solely by information system professionals (Mathieson, 1991). In addition, Ambrosio (2000) asserts that the most common error in implementing systems is failing to coordinate efforts between information technology and human resources. In his literature review of information system failure factors, Malhotra (2004) noted that systems should ensure that adaptation and innovation of business performance outcomes occurs in alignment with changing dynamics of the business environment. Armed with the knowledge of why people resist using information systems, researchers can develop better methods for designing technology, for evaluating systems, and for predicting how users will respond to new technology (Gould, Boies, & Lewis, 1991). As organizations become more dependent on information systems and their use spreads across society, the concern for developing information systems that will be used becomes even more important.

Although KMSs have become a popular focus in many firms, many KMS initiatives fail to achieve their goals. There have even been major failures documented within the law-enforcement domain. Eggen and Witte (2006) describe the FBI-contracted development of a network system (Virtual Case File) for tracking criminal cases. After spending \$170 million, the FBI still had an archaic computer system and had to restart development. "The collapse of the attempt to remake the FBI's filing system stemmed from the new system being incomplete, inadequate, and so poorly designed that it would be essentially unusable under real world conditions" (pg. A01). In addition, the system could not properly sort data and lacked common features, such as bookmarking that would help agents navigate through million of files. As a result, the FBI found the system so incomplete and unusable that they discarded the system altogether.

## **KNOWLEDGE MANAGEMENT IN LAW ENFORCEMENT**

One context in which we find evidence of the need for effective and widely accepted knowledge management systems is in the discipline of digital forensics in law enforcement agencies. Law enforcement agencies possess a large but unstructured community memory with respect to digital forensics because there is not an explicit mechanism for disseminating the experiences of every digital forensic technician and investigator (Harrison, Aucsmith, Heuston, Mocas, Morrissey, & Russelle, 2002). The explosive growth in the digital information maintained in the management systems of law enforcement agencies and the spiraling need for cross-agency access to that information have made utilizing such information both increasingly urgent and increasingly difficult (Hu, Lin, & Chen, 2005).

Incompatible content and information formats often create barriers to data access and utilization that make knowledge management a complex and daunting process (Jones & Jordan, 1998). For example, information and knowledge are captured within law enforcement agencies in various forms ranging from computer records to documented institutional orders to the personal experience of digital forensic officers (Luen & Al-Hawamdeh, 2001). The crux of the issue for law enforcement is how to surface such knowledge and bring it to bear on the problems faced by digital forensic examiners in a timely and effective manner.

Digital forensic investigators also need timely access to relevant and accurate knowledge presented in an integrated and easily analyzed manner. According to Hauck and Chen (1999), the ideal knowledge management system for law enforcement agencies should be able to provide information about problems that have not been identified previously, and thus be able to give innovative and creative support for new investigations. In the case of digital forensics, the data may be available but not in a form that makes them useful for higher level processing (Hauck, 1999). For example, digital forensic investigators devise tactics, techniques, and practices that are difficult to search and analyze. Often, only experienced and knowledgeable investigators are able to use such organizational resources effectively.

There are a number of available systems that currently serve as information management or intelligence analysis tools for law enforcement (Chen, Schroeder, Hauck, Ridgeway, Atabakhsh, Gupta, Boarman, Rasmussne, & Clements, 2002). Each of these systems has its own drawbacks and implements only a certain aspect of storing and disseminating knowledge for law enforcement. Harrison et al. (2002) proposed a prototype web-based repository (Lessons Learned Repository for Computer Forensics) for sharing information, but the effort was not widely accepted (Biros, Weiser, & Mosier, 2006) by a significant portion of the law enforcement community in a manner that allows previous discoveries to be applied to future cases.

## **NEED FOR A NATIONAL REPOSITORY OF DIGITAL FORENSIC INTELLIGENCE**

The National Repository of Digital Forensic Intelligence (NRDFI) was designed to address the knowledge management issues across many law enforcement and intelligence agencies through an integrated system that allows investigators to access and share information with other agencies. The NRDFI, a digital forensic knowledge repository development project between Oklahoma State University's Center for

Telecommunications and Network Security and the Defense Cyber Crimes Center, is a mechanism that provides flexible information sharing between law enforcement agencies. The NRDFI aims to reduce the time required to analyze evidence and advance the investigation of current cases by capturing and correlating digital forensic intelligence related information in social and organizational contexts.

Many issues and obstacles must be addressed to ensure the successful deployment of the NRDFI in the digital forensics community. There is a great sense of ownership by law enforcement agencies and individual investigators which impacts trusts and willingness to share information and creates a kind of competition between the groups (Biros et al., 2006). There are often technical and bureaucratic barriers between various law enforcement systems. The inability to integrate and access the vast number of law enforcement management systems and the inability to share information with other systems prevents an agency from receiving timely information from other data sources ultimately decreasing the efficiency of crime prevention and investigations (Hauck, 1999).

Law enforcement professionals, and more specifically digital forensic investigators, like computer network security experts tend to rely more on personal social networks or ego-centric networks rather than more formal repositories of information thus impeding information sharing in this domain (Jarvenpaa & Majchrzak, 2005). Security and confidentiality of an investigation are additional confounds to open sharing, because inappropriate controls could lead to severe consequences. Examiners are trained to search for proven techniques which provide immediate benefit for the time invested. Because there is no immediate gain in providing information for others and a very real fear that current and future criminals may improve their own skills with this knowledge, agencies are not motivated to share.

The NRDFI project was implemented to address some of the major issues described above and mimic the way digital forensic experts work. The NRDFI is designed to allow geographically diverse law enforcement agencies to share digital forensic information that will hopefully aid every agency in successfully prosecuting their case (Biros et al., 2006). In its full implementation, the NRDFI has the potential to provide exceptional gains in efficiency for forensic examiners and investigators by providing a better conduit to share relevant information between agencies and a structure through which cases can be cross referenced to have the most impact on any current investigation (Biros et al., 2006).

## **NRDFI OVERVIEW**

Details of the early NRDFI and its underlying design can be found in (Biros, et al, 2006). Based on feedback from over a year of active use at DC3, interface and features have been redesigned for deployment in multiple beta agencies within the Department of Defense and law enforcement. This paper provides a brief description of the overall design, as well as new features that have been implemented to address many issues raised in earlier research.

The essence of the repository is to capture and share the best practices of examiners with those who would otherwise need to discover or develop the same or similar techniques. The types of documentation and information contained therein are widely varied. A common search mechanism across all information is critical for finding earlier discoveries that can be brought to bear on current cases. The social networks that are heavily relied upon in law enforcement have driven the underlying structure of the NRDFI.

Each agency has its own repository in which it can maintain its own information in a very flexible structure that can be adapted to best match that organization's methods. The repository supports virtually any type of binary or text files and we continue to add parsing mechanisms for document types that will allow full-text searching of submitted items. We recognize the wealth of information that is publicly available on the Web, so we also support any URL-addressable item as a resource and will parse that for searching as well.

Resources can be grouped into panes that make sense in the agency. A subset of DC3's repository is shown in Figure 1, with groupings that align to their internal agencies and other commonly used items within the group. Panes or individual documents can be shared with any agency or group that the administrator has allowed. Individual users can also customize their own screen by moving or hiding panes that are available to them.



Figure 1: Agency Repository Interface

## COMMUNICATION SUPPORT

A need exists for a common secure communication mechanism across law enforcement agencies. Among other methods, agencies encrypt and e-mail a document over standard e-mail systems and then call the recipient with the password. The NRDFI provides two mechanisms to support online discussion. The first is a threaded discussion forum that can be created by any user. It can be attached to any resource or pane, or be created as a stand-alone resource. As a resource, it can also be shared with other cooperating agencies and appropriately vetted users in the same manner as any other document.

There is also a secure communication mechanism that serves as a secure messaging system. It has significant additional flexibility over the function of the rest of the NRDFI. Any user with access to this feature can communicate through the system with any other user on any repository, regardless of whether or not that person's agency is allowed to receive other resources. Communications are for groups of users, rather than be limited to two participants, and the membership is controlled by the person who initially creates the communication. Unlike all other sharing capabilities, the secure communication feature can extend to users who are not vetted in any repository. There is a separate secure messaging server into which any user can invite any person with HTTPS access and a basic e-mail account. Once that user creates an account, they are admitted into one or more communications to which they have been invited.

Secure messaging also has two different structures. The default mechanism simply displays interactions in reverse chronological order, allowing participants to view the entire history of the communication. If a

document is being revised collaboratively, the second mechanism allows a wiki-like structure, where a common document is edited and marked up by all invited users in an asynchronous approach.

## **ACCESS RESTRICTIONS**

Users who have access to the information within that agency are vetted by a local administrator who can also assign a certain classification sub-level. The following levels are offered as a default for both users and documents. They are strictly hierarchical and additional levels can be added in an agency for refinements when that is found to be necessary.

Unclassified – Law Enforcement Sensitive: Are particularly sensitive, especially in pending cases and could be damaging to current or future cases if the information were made available beyond the law enforcement community

Unclassified – For Official Use Only: Should not be made available outside the agency, unless there is a specific reason to do so

Unclassified – General: Available to everyone who has an account on that repository or another repository that has read access

Any user granted access at a certain level has access to documents at that level and below. Additionally, the classified hierarchy can be used on a network that is certified for that type of information. If an agency chooses to insert additional levels between these, they too are strictly hierarchical.

## **INTER-AGENCY SHARING**

Inter-agency sharing is made possible through a mesh structure in which every repository administrator selects peer repositories with which they choose to cooperate (thus mimicking the ego-centric networks of digital forensic specialists). By default, there is a core repository to which everyone has read access and administered posting access. Like any repository in the system, however, an administrator may choose to not allow his local users to access information from the core.

Grouping of outside repositories is also possible. For instance, a Department of Defense group may be established. When a user wishes to share a document to all DOD agencies that have been allowed by the administrator, he can simply select that grouping, rather than individually selecting each agency. That also facilitates future DOD agency repositories' immediate access to all these shared documents.

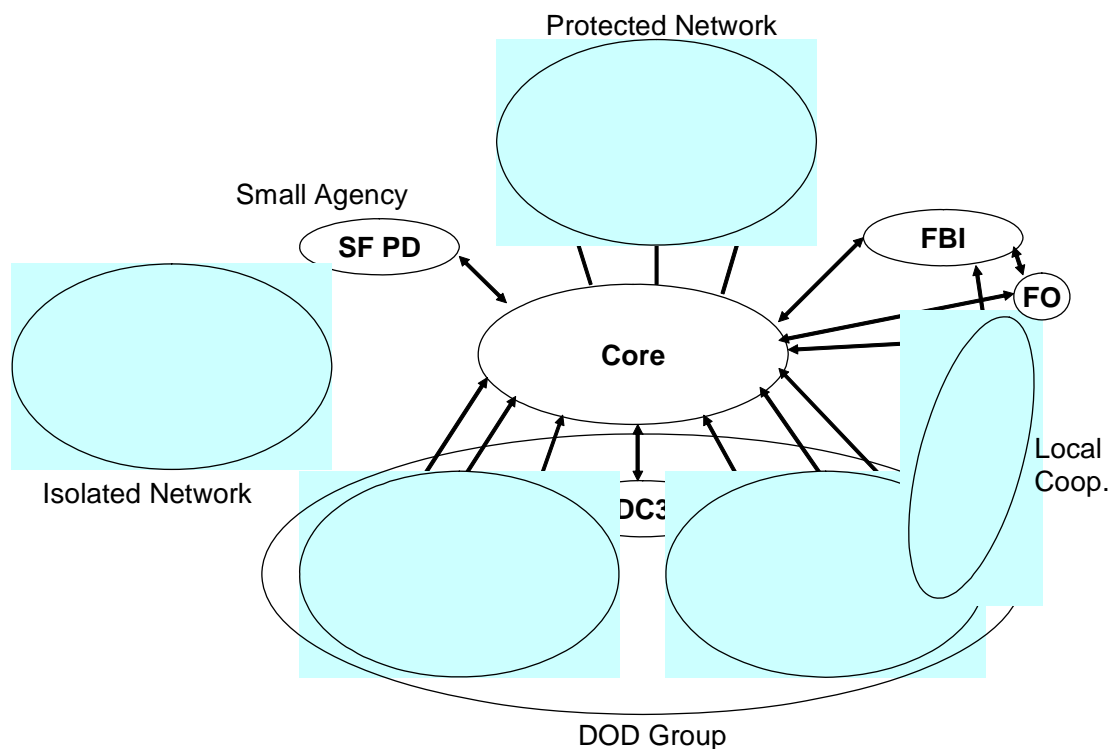


Figure 2: Flexible Mesh of Repositories

Figure 2 shows several different examples of how this may be implemented. The diagram is not intended to reflect current or planned cooperative relationships between specific agencies that might participate in the repository. It is provided purely as a notional illustration:

DC3 has a repository for storing information that they want to make available to their investigative agencies, but not outside the DOD, although the Naval Criminal Investigative Service (NCIS), the Air Force Office of Special Investigation (AFOSI), and their field offices can directly use and contribute to the core repository as well, or retain data only within their agency without elevating it even to the level of DOD.

The FBI offices have a similar structure, but one of the field offices may cooperate extensively with one of the NCIS or AFOSI field offices in the same city and liberally share new discoveries with each other. This creates a new “neighborhood” that is labeled local coop in the figure.

Small agencies may have a single repository for their lessons learned, but they share with the core repository. In the extreme, there may be no local storage at all, but a web interface directly into the core repository. A small sheriff’s office with a forensic capability can leverage the lessons learned in many other participating agencies with little investment.

Some data is very sensitive. In the figure, the NSA is shown with a neighborhood among its own central node and field offices, but only as a consumer of data from the central repository. This will not benefit other agencies; however, some organization’s requirements will prohibit sharing information.

Some agencies will choose to be entirely isolated. They can neither benefit from the central repository nor enhance it, because of a logical and/or physical separation. The underlying system design, however, allows them to share among their own neighborhood, while retaining complete control of hardware, software, and data.

Because a DOD grouping has been established with all relevant agencies, when a document is shared that is intended only for people vetted in those repositories, the user need only select that group, rather than all the repositories shown in the bottom oval. This does not override sharing relationships within the oval, so the intersection of DOD agencies that have sharing from the source agency will see the document.

## **FUTURE RESEARCH**

We believe that many systems fail even when they achieve the TAM goals of Usability and Ease of Use. Through our beta test and a series of data collection methods, we hope to demonstrate that developing an effective NRDFI requires development in accordance with the way digital forensic examiners work. First, we will collect data on the nature of digital forensic work from a large body of specialists at a national conference in early 2008. Second, we will use structured interviews to better understand the needs of the examiners through the better test. When complete, we will then use the data to refine the NRDFI to better meet the needs of the examiners.

## **CONCLUSION**

The increase in digital forensic cases far outpaces the growth in numbers of forensic examiners. General requirements for legal admissibility, however, are strict and unchanging. With constant modifications to the technologies that are examined, mechanisms to share new knowledge are critical in keeping up. An information repository that allows geographic and bureaucratic agencies responsible for the analysis to communicate and share new discoveries may be the only way to efficiently and effectively process these cases.

The collaborative effort between Oklahoma State University and the Defense Cyber Crime Center seeks to fill this need. The NRDFI prototype implementation has provided a great deal of feedback about how to overcome impediments that have been recognized in prior research. Technical constraints and relationships between repositories, as well as limitations on document access attempt to support the social networking that is currently used between agencies and personnel. The wealth of information that will be available in a widely-adopted system on this platform will be invaluable to assist investigators who are working with unfamiliar cases.

Although we have addressed many of the recognized impediments, there are still social issues that will prevent some individuals and agencies from sharing. By augmenting the system with private communication mechanisms, we hope to address the needs of even the most conservative examiner. New examiners, however, should be able to come up to speed and serve their constituents much more quickly with the wealth of knowledge that would immediately be made available through their agency's and cooperating agencies' repositories. Both the experienced and novice agent, however, will better be able to leverage the knowledge of others for successful legal outcomes.

## **REFERENCES**

- Adams, G. L., & Lamont, B. T. (2003). Knowledge management systems and developing sustainable competitive advantage. *Journal of Knowledge Management*, 7(2), 142-154.
- Ambrosio, J. (2000). Knowledge management mistakes, *ComputerWorld*, Retrieved August 2, 2007, from [www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=46693&pageNumber=2](http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=46693&pageNumber=2).
- Bock, G., Zmud, R. W., Kim, Y., & Lee, J. (2005). Behavioral intention formation in knowledge sharing: Examining the roles of extrinsic motivators, social-psychological forces, and organizational climate. *MIS Quarterly*, 29(1), 87-111.
- Biros, D., Weiser, M., & Mosier, G. (2006). Development of a national repository of digital forensic intelligence, *Journal of Digital Forensics, Security, and Law*, 1(2), 5-17.
- Chen, H., Schroeder, J., Hauck, R. V., Ridgeway, L., Atabakhsh, H., Gupta, H., Boarman, C., Rasmussne, K., & Clements, A. W. (2002). COPLINK connect: Information and knowledge management for law enforcement. *Decision Support Systems*, 34(3), 271-285.
- Davenport, T. H., & Prusak L. (1998). *Working Knowledge: How organizations manage what they know*. Boston: Harvard Business School Press.
- Dillon, A., & Morris, M. (1996). User acceptance of information technology: Theories and models, *Annual Review of Information Science and Technology*, 31, 3-32.
- Doll, W., Hendrickson, A., & Xiandong, D. (1998). Using davis' perceived usefulness and ease of use instruments for decision making: A confirmatory and multi-group invariance analysis. *Decision Sciences*, 29(4), 839-869.
- Eggen, D., & Witte, G. "The FBI's Upgrade That Wasn't," *The Washington Post*, 18 August 2006, sec. A:01.



- Gould, J. D., Boies, S. J., & Lewis, C. (1991). Making useable, useful, productivity-enhancing computer applications. *Communications of the ACM*, 34(1), 74-85.
- Hahn, J., & Subramani, M. R. (2000). A framework of knowledge manage systems: Issues and challenges for theory and practice. *Proceedings from the twenty first international conference of Information Systems*, Australia.
- Hansen, M. T., & Von, O. B. (2001). Introducing T-shaped manger: Knowledge management's next generation. *Harvard Business Review*, 79(3), 106-116.
- Harrison, W., Aucsmith, D., Heuston, G., Mocas, S., Morrissey, M., & Russelle, S., (2002). A lessons learned repository for computer forensics, *International Journal of Digital Evidence*, 1(3), Retrieved August 6, 2007, from <http://www.utica.edu/academic/institutes/ecii/publications/articles/A049D6C7-93E9-51F2-A468BF90038985DB.pdf>.
- Hauck, R. (1999). COPLINK: Exploring usability of a multimedia database application for law enforcement. Report prepared for a National Institute of Justice site visit, <http://ai.eller.arizona.edu/COPLINK/publications/nij.pdf>.
- Hauck, R. V., & Chen, H. (1999). COPLINK: A case of intelligent analysis and knowledge management. *Proceedings of the International Conference of Information Systems*, 15-28, Charlotte NC, ICIS.
- Hu, P. J., Lin, C., & Chen, H. (2005). User acceptance of intelligence and security informatics technology: A study of COPLINK. *Journal of the American Society for Information Science and Technology*, 56(3), 235-244.
- Jarvenpaa, S. L., & Majchrzak, A. (2005). Developing individuals' transactive memories of their ego-centric networks to mitigate risks of knowledge sharing: The case of professionals protecting cybersecurity. *Proceedings from the Twenty Sixth the International Conference of Information Systems*, Australia.
- Jones, P., & Jordan, J. (1998). Knowledge orientations and team effectiveness. *International Journal of Technology Management*, 16(1-3), 152-161.
- Lin, C., Hu, P. J., Chen, H., & Schroeder, J. (2004). Technology implementation management in law enforcement: COPLINK system usability and user acceptance evaluations. *Social Science Computer Review*, 22(1), 24-36.
- Luen, T. W., & Al-Hawamdeh, S. (2001). Knowledge management in the public sector: Principles and practices in police work. *Journal of Information Science*, 27(5), 311-318.
- Malhotra, Y. (2004). Why knowledge management systems fail? Enablers and constraints of knowledge management in human enterprises. Retrieved August 7, 2007, from [www.brint.org/WhyKMSFail.htm](http://www.brint.org/WhyKMSFail.htm).
- Mathieson, K. (1991). Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research*, 2(3), 173-191.
- Mathieson, K., Peacock, E., & Chin, W. (2001). Extending the technology acceptance model: The influence of perceived user resources. *The Database for Advances in Information Systems*, 32(3), 86-112.
- McEvily, B., Perronne, V., & Zaheer, A. (2003). Trust as an organizing principle. *Organization Science*, 14(1), 93-103.
- Money, W., & Turner, A. (2004). Application of the technology model to a knowledge management system. *Proceedings of the 37th Hawaii International Conference on Systems Science*, Hawaii.
- Tiwana, A. (2000). *The Knowledge Management Toolkit: Practical Techniques for Building a Knowledge Management System*. New Jersey: Prentice Hall.
- Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2), 186-204.
- Vizcaino, A., Soto, J. P., Portillo, J., & Piattini, M. (2007). A multi-agent model to develop knowledge management systems. *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, HICSS07.
- Wasko, M., & Faraj, S. (2005). Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. *MIS Quarterly*, 29(1), 35-57.

## **COPYRIGHT**

Biros, Weiser, Whitfield ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **ADSL Router Forensics Part 1: An introduction to a new source of electronic evidence**

Patryk Szewczyk  
School of Computer and Information Science  
Edith Cowan University  
p.szewczyk@ecu.edu.au

### **Abstract**

*Currently there appears to be a lack of research in the area of developing tools, testing methodologies, and creating standards for ADSL router forensics. The paper examines a wide range of literature and introduces the concept of ADSL router forensics as a new and potential field of research for digital forensics investigators. It begins by examining why there is a need for router forensics by detailing some of the more common threats which consumers may experience while online. An outline will be provided discussing the feasibility, limitations and potential risks of router forensics. The paper will then examine one possible avenue for undertaking router forensics and how this applies to the Linksys WRT54g and finally portrays where the research will continue to hereafter.*

### **Keywords**

ADSL router forensics, digital forensics, embedded systems, JTAG, Linksys wrt54g

## **INTRODUCTION**

The demand for Asymmetric Digital Subscriber Line (ADSL) devices has increased considerably as consumers are now offered fast and inexpensive methods to connect to the Internet. The plug and play nature of ADSL routers permits consumers to bypass the tedious configuration process and hence connect to the Internet within minutes. History has shown that the number of published exploits and threats for a particular device or software is generally proportional to the number of individual's utilising that system. In this instance as broadband technology is becoming the predominant method for Internet connectivity the number of published exploits on the more common range of Small office Home office (SoHo) routers is also on a parallel rise. Hence, in order to combat and pursue criminals whom endeavour to maliciously destroy, alter and degrade a consumer's online experience, techniques and standards must be developed to ensure a thorough forensic investigation of the networking device. As there are numerous threats to ADSL routers the number of standards, techniques and frameworks which could be used to forensically investigate a router is non-existent. The paper hereafter will introduce and discuss the necessity for sound ADSL router forensic principles.

## **THREATS TO ADSL ROUTERS**

ADSL routers control the traffic flow between the Internet and the internal hosts on a SoHo network. As broadband technology becomes faster and more reliable consumers may opt to use their ADSL router to share an Internet connection in a home or office, utilise a personal or business internal web server, enjoy the benefits of Voice over Internet Protocol (VoIP) and the convenience of file sharing. In some instances consumers are unaware of the potential security risks and recommended security approaches for ADSL routers (Szewczyk 2006). However, the threats presented hereafter show that a simple and effortless method may disrupt or terminate an entire network connection to the rest of the world (Chang 2002).

### **Denial of service attack**

A Denial of Service (DoS) attack targets routers in a 'reflector' or 'direct' mode and may halt the connection between the router and the Internet Service Provider (ISP). Most recently ADSL routers are predominately becoming victim to 'reflector' based attacks. The attacker transmits numerous synchronisation requests to router 'A' with a preset static source IP address of the soon to be compromised, router 'B'. Router 'A' responds to these numerous requests and sends the numerous acknowledgement packets to router 'B'. Eventually two routers have had their resources consumed as the SYN and ACK packets are constantly transmitted. Alternatively 'direct' attacks also consume the router's resources by flooding it with numerous Transmission Control Protocol (TCP), Internet Control Message Protocol (ICMP), or User Datagram Packets (UDP) sent directly from the attacker. In this instance, the source IP address is spoofed each time hence leaving half open

connections on the targeted router. Due to the resource limitations of the router the device may halt and hence require a power-cycle reboot (Chang 2002).

### **Access control**

Routers are pre-configured to prevent remote management from an IP address outside of the internal network. Research has shown (Stamm et al. 2006) that this security technique can be compromised when an unsuspecting individual accesses a webpage and permits a malicious Java Applet to load. Once the malware is loaded onto a workstation, it begins to ping the other hosts on the subnet in an attempt to discover the location of the gateway/router. After a number of attempts the malware should discover the gateway by detecting which IP address is hosting a web configuration management system. Specific images used in the web configuration management system are unique to each router and hence the make and model of the router may be detected by the type of images retrieved from the default gateway. Once the router type is known, the malware may manipulate configuration settings, alters DNS addresses, enables port forwarding and remote management to a specific IP address, and disables Network Address Translation (NAT) (Stamm et al. 2006).

### **Software robots**

Network address translation (NAT) permits a number of internal workstations to access the Internet simultaneously using a router allocated, private IP address. Essentially NAT provides a level of protection to the internal hosts by not using a public IP address and is enabled by default on many router brands. However, once NAT is purposefully, maliciously or accidentally disabled, the internal hosts may become victim to the control of a BotMaster controlling a collection of BotNets. A BotNet is a collection of hosts under the control of a BotMaster who controls the infectors and infected hosts to carry out various malicious tasks (Rajab et al. 2006a; Rajab et al. 2006b; Ramachandran & Feamster 2006).

A BotNet is instantiated by a BotMaster whom initiates a transmission of a shell code from an already infected host to a newly targeted host. The newly targeted host may become infected through a malware binary download (e.g. through an email attachment). The malware begins to download through the trivial file transfer protocol (TFTP), which once complete, automatically initiates the configuration scripts, allowing the BotMaster to control the victimised workstation. The infected host may then send mass amounts of spam email, install and collect data from key loggers, and launch distributed denial of service (DDoS) attacks. Routers are similar in computational power to the computer of the mid 1990's. Hence, BotMasters are able to abuse the processor cycle and memory resources of both a workstation and a router to undertake a range of tasks including, cracking complex password or encryption schemes.

### **DNS hacking**

The Domain Name Server (DNS) is a dynamic address in most instances is allocated ISP to the router. However, the DNS may be easily changed manually by a non-technical individual through the router's web management interface. A compromised ADSL router, configured with a malicious DNS may have all of its traffic forwarded to a bogus DNS server (Heron 2007). When the unsuspecting user requests to access their bank, email or Microsoft update website they are instead redirected to what looks like the authentic website they would usually visit. The illegitimate website may then be used to capture usernames and passwords entered by the end-user when attempting authentication. It may also be used to send malicious updates to end-user or redirect them to illegal websites.

### **Voice over Internet Protocol vulnerabilities**

Voice over Internet Protocol (VoIP) uses the existing network infrastructure in-conjunction with ADSL routers to conduct voice communication between nodes. VoIP communication can only be permitted by opening specific ports to allow the transmission of packets between the external and internal VoIP hosts (Whitworth 2006). Opening ports on a router increases the number of holes an attacker may use to exploit a system. Furthermore, most VoIP devices are computers with scaled down web-servers allowing them to be queried and exploited like any web-server if not properly configured (Bradbury 2007).

If using NAT on a SoHo network this instantiates a problem when attempting to encrypt VoIP communication using IPSec (Walsh & Kuhn 2005). The problem lies within the header of the packet, which is encrypted by the router on all outbound traffic. This is problematic for the routing device as it is unable to read the destination IP addresses of the sender and receiver. Hence, various publications (Tucker 2004; Walberg 2007) propose a feasible yet dangerous approach that end-users simply disable NAT and hence use public IP addresses for all of the internally networked devices. This approach would permit all internal workstation to be accessible to potential offenders over the Internet.

### **Embedded System Vulnerabilities**

An ADSL router is a scaled down computer system matching the computational power of the desktop workstations of the mid 1990's. Collectively an ADSL router is an embedded system comprising of a processor, memory and embedded software. A single programming error may cause numerous faults to consumers. Acre (2003) discusses that the research department at Core Security Technologies discovered a stack and heap buffer overflow error within many embedded systems. More specifically this error targets routers and may allow an attacker to bypass all authentication techniques thus acquiring effortless access to the router.

Tsow (2006) argues that the reason embedded systems and more specifically embedded software are being targeted within routers is that virus and malware detectors are unable to scan embedded software. More specifically the firmware can be compiled with a static malicious DNS server address. The newly compiled malicious firmware may be uploaded to the router prior to sale. Unless a forensic examination is made, scanning the router's operating system for malicious code or processor activity is not available. Numerous router firmware images are open-source, permitting an attacker to experiment with the software to discover flaws and weaknesses prior to launching an attack (Tsow 2006). Unlike a computer with a new operating system upgrade from Windows XP to Windows Vista, a highly skilled hacker changing the firmware on a router without authorisation should see no difference to the consumer (Tsow 2006). The router is ideally intended to allow trouble free Internet access for the non-technical user.

## **ADSL ROUTER FORENSICS FEASIBILITY**

### **Digital forensics**

Reith et al. (2002) agree that online offenders believe there is still a degree of anonymity when using technology to commit electronic crimes. However, computer forensics which once dealt solely with 'persistent data' (Nolan et al. 2005) now takes on a new stream of evidence acquisition from volatile memory. Forensics investigators have identified that volatile evidence may remain on a computer system a long time after the crime was committed. Hence, because certain electronic devices do not have persistent storage medium does not mean they cannot be forensically investigated. Desktop workstations are no longer the sole means of forensic interest to investigators with offenders breaching laws on cell phones, PDAs and importantly network routers. Researchers have begun developing methods to collect evidence from devices with volatile memory such as Jansen and Ayers (2004) in *Guidelines on PDA Forensics* and also Jansen and Ayers (2006) in *Guidelines on Cell Phone Forensics*. However, researchers have yet to pursue the development of guidelines, frameworks, models and practices on acquiring volatile memory evidence from network routers.

Computer systems with storage mediums such as hard drives and volatile memory may be forensically analysed using pre-tested models, frameworks and techniques. Reith et al. (2002) outline various forensic protocols and argue that these practices are not standardised. Each entity whether it is the Federal Bureau of Investigation (FBI) or US Department of Justice alters these protocols dependant on their needs and requirements. Alternatively these entities may also develop new frameworks depending on the device, operating system and resources available to the forensic investigator although do not release this to the public realm. The numerous digital forensic frameworks and models available (Carrier & Spafford 2003; Department of Justice 2001; Ó'Ciardhuáin 2004; Palmer 2001) provide a set of principles and techniques for acquiring, preserving, analysing and presenting digital evidence acquired from numerous digital devices. Unfortunately neither of these acknowledges ADSL routers as a potential source of evidence. The Department of Justice (2001) does however recognise network routers as a potential source of evidence for forensic investigations— specifically the configuration files.

### **Volatile memory forensics**

Although numerous digital devices have had their volatile memory investigated for potential evidence, viruses and malware are yet to be extracted by forensic investigators. Malicious software such as the Code Red and SQL Slammer work reside solely in volatile memory (Carrier & Grand 2004). Hence, investigating the hard drive may not have recovered evidence of a worm. From a computer forensics perspective, investigating the memory contents may recover the current running processes, unencrypted data such as passwords and current user activity (Carrier & Grand 2004). One such tool which is capable of forensically examining the contents of volatile memory includes WinHex (WinHex 2007). This tool permits investigators to acquire a memory dump which may easily recover unencrypted passwords which have not yet been overwritten in volatile memory

(Casey 2004). The tool is specifically designed for desktop workstations although it does show that an extraction of evidence from volatile memory is feasible.

A proof of concept device 'Tribble' has been developed that allows an investigator to perform a forensic analysis on a live workstation (Carrier & Grand 2004). The concept was a Peripheral Component Interconnect (PCI) hardware based card which is installed in a desktop workstation prior to its usage. If an attack on the target system was undertaken and an analysis was required, Tribble could be used to forensically capture evidence from the volatile memory modules (Carrier & Grand 2004). However, this product requires that it be installed prior to any attack or investigation being carried out. Furthermore, in the context of ADSL routers such a device would require each individual or vendor to attach an additional chipset to their device prior to its sale and usage which may not be feasible for the customer or the vendor from an economical sense.

Typical computer forensic tools may examine a cloned hard disk within an isolated laboratory. However, data of a volatile nature cannot be removed from the location of interest as shutting down the device would erase the non-persistent data. According to Nelson et al. (2006) network forensics would typically follow the systematic process of:

- Closing the network ports or processes that allowed the intruder to carry out the attack.
- Acquire the drive which had been compromised.
- Make an exact replica of the drive with a bit-stream image.
- Verify the duplicate image to the original image.

However, using the procedures detailed is not feasible as closing ports, and making a replica would alter the volatile data and hence erase potential evidence. For this reason, there is a need for sound volatile memory forensic methods which can extract evidence at the scene of the crime within a timely manner, without interfering or overwriting existing evidence (Petroni et al. 2006). Thus, the proposed systematic principles are not ideal techniques for router forensics as each step may potentially alter or erase the evidence within memory.

### **Difficulties of volatile memory modules**

The router's various memory modules are the key to extracting data of forensic interest (Brown 2006). The only persistent component of an ADSL router is the non-volatile Random Access Memory (NVRAM) or flash memory and may be forensically examined without altering key evidence to determine if the firmware or operating system has been altered. NVRAM on all ADSL routers should contain; power on boot procedures and configuration files for the particular network. The Dynamic Random Access Memory (DRAM) or Static Random Access Memory (SRAM) contains; current running processes, routing tables, simplified network logs, network and connectivity statistics (Brown 2006). Any device which can not be switched off is classified in computer forensics as a 'live' device. Investigating a 'live' device follows the scientific principle that any action used to observe the evidence may in essence alter it. Alternatively, shutting down the device may erase the data of forensic interest located in the volatile memory modules. Nikkel (2005) suggests the following assumptions about router forensics whilst attempting to manifest techniques, methods and tools:

- The router where the evidence is stored will not be in full custody of the investigator when carrying out the investigation.
- A forensic image of the router's volatile memory may not be feasible to initiate.
- The evidence on the router is dynamic and may be erased or altered before the forensic examination may take place.
- Verification of the data collected may prove difficult or impossible if the router is switched off hence erasing the volatile memory.

Hence, when forensically examining volatile memory in a device such as a router Petroni et al. (2006) state that in order to decipher the memory status at any point the investigator must thoroughly analyse two main criteria. Firstly, identify how the operating system would generally interact with the memory (i.e. what should be in a specific memory cell in a normal state) and secondly, how the memory cells would have been rearranged or altered during and after an incident occurs (i.e. what the memory cells should contain after the incident occurred).

### **Industry router forensics**

Although Cisco routers encompass the same volatile memory as SoHo routers they have had some publicity on methods to forensically investigate the device after a breach has occurred. Livingston (2005) details what

evidence (if any) should be extracted from volatile memory initially if an incident occurs. Although she only details the commands on a Cisco router using the 'show' command, these commands may be replicated on common ADSL routers through the Linux telnet or SSH interface. Ideally a connection would be instantiated using HyperTerminal or the equivalent to allow logging of the session. The follow points detail a list of items of evidence ranked in importance from highest to lowest in conducting a forensic examination (Livingston 2005):

- Router system clock information.
- Firmware type and version.
- IP address of authenticated users.
- Configuration files located in NVRAM – boot sequence, default libraries.
- Routing table information.

To date there is only one router forensic tool available which is constantly undergoing research and testing. All of the evidence above is automatically collected by the evidence collection tool namely the "Cisco Router Evidence Extraction Disk" (CREED) created by researcher Thomas Akin (2002). The tool is a small 1.7MB program which resides on a bootable floppy disk, requiring minimal interaction from the investigator to complete the acquisition procedure. The investigator places the floppy disk into the computer, connects a serial cable from the acquisition computer to the router, and once booted the investigator types 'acquire' in the console and the automated process acquires the evidence (Akin 2002). The tool does however require the serial port on the router as part of the connection process which unfortunately is not standard on ADSL routers. Furthermore, on many of popular ADSL routers, consumers are only presented with numerous Ethernet ports and a Registered Jack (RJ11) telephone line port.

### Investigative procedures

Conducting an investigation on an ADSL router is feasible and in many instances is able to acquire evidence which may be of use to an investigator. Table 11 below details a list of commands which may be issued on a Linux workstation. However, since ADSL routers operate under a Linux architecture these commands may also be issued on an ADSL router once a telnet or SSH connection is made (Burdach 2004). However, although an investigation may be of use to an extent, executing each of the commands may in fact overwrite a potential important piece of data in volatile memory.

Table 11 Investigation commands for an ADSL router

Priority	Description	Command
1	Current date and time	date -u
2	Cache tables	arp -an route -n
3	Open TCP/UDP Connections	/proc/net/tcp /proc/net/udp
4	Image of physical memory	/proc/kcore
5	Loaded kernel modules	/proc/modules insmod -f /proc/ksmys
6	Active processes	lsdf -n -P -l
7	Useful extra information <ul style="list-style-type: none"> <li>• Version</li> <li>• Host name</li> <li>• Domain name</li> <li>• Hardware info</li> <li>• Swap partitions</li> <li>• Local partitions</li> <li>• Mounted file systems</li> <li>• Uptime</li> </ul>	/proc/version /proc/sys/kernel/name /proc/sys/kernel/domainname /proc/cpuinfo /proc/swaps /proc/partitions /proc/self/mounts /proc/uptimes

## ACQUIRING EVIDENCE USING JTAG

The first principle in undertaking a thorough forensic investigation of any digital device is to ensure that the data of forensic interest remains unchanged throughout the entire process. As detailed previously the number of external ports on an ADSL router is quite limiting in their functionality. A prospective method to acquire evidence from an ADSL router is to utilise the Joint Test Action Group (JTAG) boundary-scan. The JTAG port is generally utilised by manufacturers for testing circuit boards before they are released to the public for sale purposes (Breeuwsma 2006). However, the same principles which are applied to testing circuit boards can also be used to forensically acquire data and/or an image from a specific embedded system. The following section will briefly outline the potential benefits and methods by which the JTAG boundary-scan may be used.

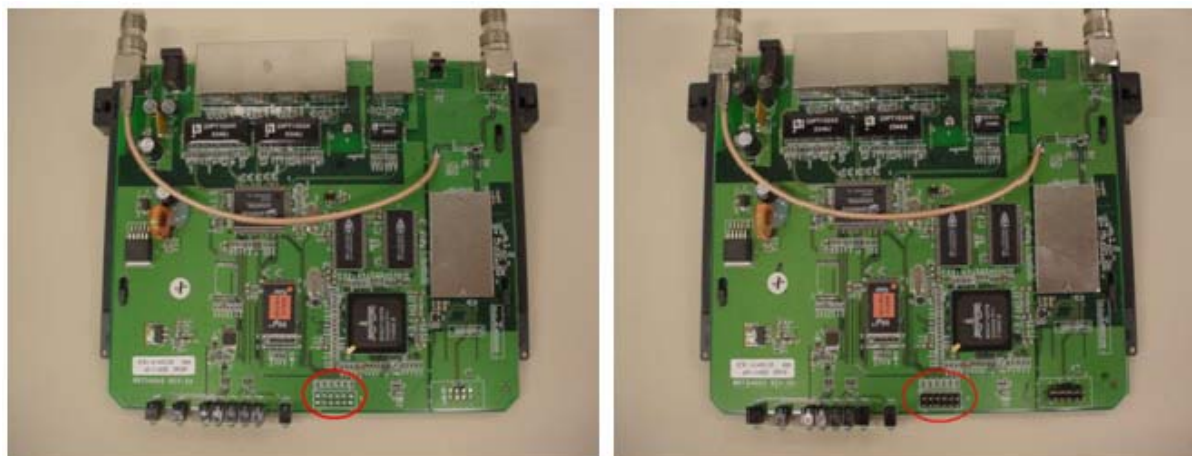
### Potential benefits

Many forensic applications load data into the memory which is then executed by the processor. In the instance of acquiring evidence from a hard disk on a desktop workstation, a boot disk could be utilised where direct write access to the disk is restricted. Only volatile memory and the processor are utilised and data is read from the hard disk and is transmitted over a network (using Net Cat) creating a replicated forensic image. Whilst, data could be retrieved in ADSL routers utilising a direct Ethernet connection, this would in essence load data into memory, potentially leaving a memory footprint which contravenes with the first principle of undertaking a forensic investigation. Secondly, had the user set any passwords, attempting to retrieve or bypass these may also prove difficult (Breeuwsma 2006) and evidence may be lost before access is granted. Utilising the JTAG port permits the investigator to communicate directly with the memory modules acquiring evidence in a sound forensic manner.

In numerous instances ADSL routers halt when a software error is executed or when all available system memory is exhausted. Consumers, whom face these dilemmas and contact their Internet Service Provider, are instructed to power cycle their router, which clears memory. Furthermore, the power cycle will also cause the operating system to reboot and thus resolving any programming error within the software. Unfortunately power cycling a router will clear memory and hence remove all potential evidence. The investigator may not be aware of how long the router has been operating for and hence a sudden exhaustion of memory by loading a forensic tool may again cause the router to come to a halt. Utilising a JTAG port allows the investigator to communicate directly with the processor and hence bypassing the need for memory to be allocated to specific forensic tools for execution (Breeuwsma 2006).

### Using JTAG for communication

The JTAG interface is already publicly used amongst the hacker community for imaging firmware onto various ADSL routers but more specifically the Linksys WRT54g. By default the Linksys WRT54g does not have a user accessible JTAG interface as with many of the publicly available ADSL routers (**Error! Reference source not found.**). However, resoldering a new 12 pin header onto the board permits the user to connect a pre-purchased or a custom made un-buffered JTAG cable as outlined by the developers of the embedded system operating system 'OpenWRT' (OpenWRT 2007).



*Figure 14 A Linksys WRT54g with and without a JTAG interface header*



The JTAG interface cable may either connect to the serial or parallel port of a workstation. Once a connection to the onboard Linksys WRT54g JTAG interface is established, a utility such as the “Hairy Dairy Maid Debricking” utility may be utilised. This tool may be run to debug and program the onboard memory modules. Whilst, the developer of the firmware recovery utility chooses to remain anonymous for legal purposes, the software is not full proof and may damage the router beyond repair. Does utility does permit a user to flash, erase or backup the current operating system stored on the Linksys WRT54g router (HairyDairyMaid 2007). Whilst the utility does permit an end-user to communicate directly with the processor and hence execute commands it does so in a non-forensic manner. Utilising the same principles of direct processor communications, research will be undertaken into developing and testing a tool which may essentially trick the processor into retrieving the contents from all memory modules and presenting this information in a way which is feasible to understand for a forensic investigator.

## **CONCLUSION:**

The current rate of research on ADSL router forensics is almost non-existent. Research, tools, and methodologies on computer hard disk forensics have exhausted itself over the past few years, and it appears that the era of cell phone forensics is beginning to gain interest. However, many individuals may be underestimating the potential crimes associated with the use of ADSL routers. Until there is a wide interest amongst the forensic computing community, the number and severity of these crimes may only escalate. A forensic acquisition solution for ADSL routers may also apply to a wide range of consumer electronics in the future.

The aim of the paper was to increase knowledge on the yet to become mainstream area of ADSL router forensics. Whilst routers may already be used to commit electronic crimes as detailed in this paper, as technology progresses the rate and publicity of these crimes should also increase. As the cost of manufacturing consumer electronics decreases, the next few years should see ADSL routers with increased memory and processing power for enhanced firewall rule sets, improved VoIP capability and furthermore permit an increase in the number of simultaneous connections.

## **REFERENCES:**

- Acre, I. (2003). The Rise of the Gadgets. *IEEE Journal*, 1(5), 78-81.
- Akin, T. (2002). CREED (Cisco Router Evidence Extraction Disk), URL <http://web.archive.org/web/20040214172413/http://cybercrime.kennesaw.edu/creed/> Accessed 10 April, 2007
- Breeuwsma, I. M. F. (2006). Forensic imaging of embedded systems using JTAG (boundary-scan). *Digital Investigation*, 3(1), 32-42.
- Brown, C. L. T. (2006). *Computer Evidence Collection and Preservation*. Hingham, MA: Charles River Media.
- Burdach, M. (2004). Forensic Analysis of a Live Linux System, Pt. 1, URL <http://www.securityfocus.com/infocus/1769> Accessed 20 April, 2007
- Carrier, B., & Spafford, E. H. (2003). Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence*, 2(2), 1-20.
- Carrier, B. D., & Grand, J. (2004). A hardware-based memory acquisition procedure for digital investigations. *Digital Investigation*, 1(1), 50-60.
- Casey, E. (2004). Tool review-WinHex. *Digital Investigation*, 1(2), 114-128.
- Chang, R. (2002). Defending against flooding-based distributed denial-of-service attacks: a tutorial. *IEEE Communications Magazine*, 40(10), 42-51.
- Department of Justice. (2001). *Electronic Crime Scene Investigation - A Guide for First Responders*, URL <http://www.ncjrs.gov/pdffiles1/nij/187736.pdf> Accessed 2 April, 2007
- HairyDairyMaid. (2007). WRT54G EJTAG DeBrick Guide, URL [http://www.ranvik.net/prosjekter-privat/jtag\\_for\\_wrt54g\\_og\\_wrt54gs/HairyDairyMaid\\_WRT54G\\_v22.pdf](http://www.ranvik.net/prosjekter-privat/jtag_for_wrt54g_og_wrt54gs/HairyDairyMaid_WRT54G_v22.pdf) Accessed 1 September, 2007

- Jansen, W., & Ayers, R. (2004). Guidelines on PDA Forensics, URL <http://csrc.nist.gov/publications/nistpubs/800-72/sp800-72.pdf> Accessed 5 April, 2007
- Jansen, W., & Ayers, R. (2006). Guidelines on Cell Phone Forensics, URL [http://www.cert.org/archive/pdf/FRGCF\\_v1.3.pdf](http://www.cert.org/archive/pdf/FRGCF_v1.3.pdf) Accessed 12 April, 2007
- Livingston, O. (2005). Effective Data Investigation on Cisco Routers, URL <http://www.securitydocs.com/library/3474> Accessed 23 March, 2007
- Nikkel, B. J. (2005). Generalizing sources of live network evidence. *Digital Investigation*, 2(3), 193-200.
- Nolan, R., O'Sullivan, C., Branson, J., & Waits, C. (2005). First Responders Guide to Computer Forensics, URL [http://www.cert.org/archive/pdf/FRGCF\\_v1.3.pdf](http://www.cert.org/archive/pdf/FRGCF_v1.3.pdf) Accessed 22 March, 2007
- Ó'Ciardhuáin, S. (2004). An Extended Model of Cybercrime Investigations. *International Journal of Digital Evidence*, 3(1), 1-22.
- OpenWRT. (2007). JTAG Cables, URL [http://wiki.openwrt.org/OpenWrtDocs/Customizing/Hardware/JTAG\\_Cable](http://wiki.openwrt.org/OpenWrtDocs/Customizing/Hardware/JTAG_Cable) Accessed 10 September, 2007
- Palmer, G. (2001). A Road Map for Digital Forensic Research. Paper presented at the First Digital Forensic Research Workshop, Utica, New York.
- Petroni, N. L., Waltersb, A., Fräsera, T., & Arbaugh, W. A. (2006). FATKit: A framework for the extraction and analysis of digital forensic data from volatile system memory. *Digital Investigation*, 3(4), 197-210.
- Rajab, M. A., Monroe, F., & Terzis, A. (2006a). On the Impact of Dynamic Addressing on Malware Propagation. Paper presented at the Proceedings of the 4th ACM workshop on Recurring malware WORM '06, George Mason University, Fairfax.
- Rajab, M. A., Zarfoss, J., Monroe, F., & Terzis, A. (2006b). Security and privacy: A multifaceted approach to understanding the botnet phenomenon. Paper presented at the Proceedings of the 6th ACM SIGCOMM on Internet measurement IMC '06, Rio de Janeiro, Brazil.
- Ramachandran, A., & Feamster, N. (2006). Understanding the Network Level Behavior of Spammers. Paper presented at the Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '06, Pisa, Italy.
- Reith, M., Carr, C., & Gunsch, G. (2002). An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, 1(3), 1-12.
- Stamm, S., Ramzan, Z., & Jakobsson, M. (2006). Drive-By Pharming, URL [http://www.symantec.com/avcenter/reference/Driveby\\_Pharming.pdf](http://www.symantec.com/avcenter/reference/Driveby_Pharming.pdf) Accessed 3 April, 2007
- Szewczyk, P. (2006). Individuals Perceptions of Wireless Security in the Home Environment. Paper presented at the 4th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia.
- Tsow, A. (2006). Phishing with Consumer Electronics: Malicious Home Routers. Paper presented at the 15th International World Wide Web Conference, Edinburgh, Scotland.
- Tucker, G. S. (2004). Voice Over Internet Protocol (VoIP) and Security, URL [http://www.giac.org/practical/GSEC/Greg\\_Tucker\\_GSEC.pdf](http://www.giac.org/practical/GSEC/Greg_Tucker_GSEC.pdf) Accessed 12 March, 2007
- Walberg, S. (2007). How to configure SIP and NAT. *Linux Journal*, 2007(155).
- Walsh, T. J., & Kuhn, D. R. (2005). Challenges in securing voice over IP. *IEEE Security & Privacy Magazine*, 3(3), 44-49.
- Whitworth, M. (2006). VoIP – a call for better protection. *Network Security*, 2006(4), 11-12.
- WinHex. (2007). WinHex: Computer Forensics & Data Recovery Software, Hex Editor & Disk Editor, URL <http://www.winhex.com/winhex/> Accessed 13 August, 2007

## **COPYRIGHT**

[Patryk Szewczyk] ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on

the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **An examination of the Asus WL-HDD 2.5 as a Nepenthes malware collector**

Patryk Szewczyk

School of Computer and Information Science

Edith Cowan University

p.szewczyk@ecu.edu.au

### **Abstract**

*The Linksys WRT54g has been used as a host for network forensics tools for instance Snort for a long period of time. Whilst large corporations are already utilising network forensic tools, this paper demonstrates that it is quite feasible for a non-security specialist to track and capture malicious network traffic. This paper introduces the Asus Wireless Hard disk as a replacement for the popular Linksys WRT54g. Firstly, the Linksys router will be introduced detailing some of the research that was undertaken on the device over the years amongst the security community. It then briefly discusses malicious software and the impact this may have for a home user. The paper then outlines the trivial steps in setting up Nepenthes 0.1.7 (a malware collector) for the Asus WL-HDD 2.5 according to the Nepenthes and tests the feasibility of running the malware collector on the selected device. The paper then concludes on discussing the limitations of the device when attempting to execute Nepenthes.*

### **Keywords**

ADSL routers, Nepenthes, OpenWRT, malware, network forensics

### **INTRODUCTION**

Trends in router technology advancements are permitting consumers to use their device as a gateway for Internet connectivity, a resource sharing point and a wireless access point. As routing devices become increasingly powerful with superior processors and significant increases in memory, developers are opting to utilise routers for a variety of applications. One router which received immense publicity was the Linksys WRT54g due to its highly manipulative firmware and basic configurable nature. Literature demonstrating the flexibility and ease of use of the Linksys WRT54g for both researchers and hobbyists (Asadoorian & Pesce 2007) is still being publicly released years after the initial product release.

Hackers and security enthusiasts may customise the Linksys WRT54g firmware (Al-Zarouni 2005) dependant on the place and purpose of use. Numerous pre-compiled firmware images are available specifically for the Linksys WRT54g and many other embedded system architectures. Innes (2005) discussed the application of some of the publicly available, pre-compiled software packages for the Linksys WRT54g operating on the OpenWRT firmware. The paper demonstrated how a Small office Home office (SoHo) router may be transformed to undertake various 802.11 wireless monitoring, intrusion detection and network forensic tasks. However, as time progresses the device which once permitted a wide range of software to be on the Linksys WRT54g is slowly becoming obsolete. One significant aspect in which the Linksys WRT54g is now insufficient is in the memory and storage availability which prevents it from being used for various network analysis and forensic activities.

Linksys released numerous versions of the Linksys WRT54g (versions 1 through to 8) with processors ranging from MIPS 125 MHz through to 240 MHz (OpenWRT 2006). However, as the processor performance increased on the high end routers the availability of flash and random access memory (RAM) decreased. Hence system performance was balanced across all the Linksys WRT54g routers. Whilst the device specifications are sound for the router's requirements any intensive third party software will halt the device and require a manual power cycle. In contrast certain network analysis and forensic software does not only consume excessive resources but also requires a medium on which to store data it collects. One specific setup which is not feasible, is operating a Snort intrusion detection system (Snort 2007) coupled with a database server to log events. As the Linksys router has minimal non-volatile storage availability, the only feasible option is to utilise a remote database server to which the Snort intrusion detection system may connect and store log files.

One of the ways to bypass the resource requirements of a router is to utilise an Asus Wireless Hard Disk (WL-HDD). Competing with the Linksys WRT54g, the Asus WL-HDD encompasses a *Broadcom 4710* – 125 MHz processor coupled with four megabytes of flash and sixteen megabytes of RAM. The device mimics the specifications of the Linksys WRT54g up to and including revision four. The device utilises a removable

external dipole antenna, a 10/100 Ethernet connection and a user attachable hard disk connector for a notebook forty gigabyte hard drive (Johnson 2005). By default the Asus WL-HDD utilises a proprietary pre-imaged firmware stored on the four megabyte flash memory. The default operating system allows end-users to: enable MAC address filtering, Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) encryption for wireless access. Further attributes include the potential to share and access resources amongst multi users using the Samba server package and the File Transfer Protocol (FTP). The device also has the added benefit of being able to copy the contents of USB devices on the fly without the need for a workstation.

## **MALWARE FOR FORENSIC ANALYSIS**

Traditional forensic investigations have focused purely on recovering data from persistent storage mediums. However, the number of online crimes is increasing with distributed denial of service attacks and the propagation of malicious software. Thus the need to identify the source and design methods of prevention is becoming increasingly prevalent. Throughout the early 1990's forensic investigators focused predominately in attempting to identify the 'author' of viruses, Trojan horses and worms through a process of "software forensics" (Spafford & Weeber 1993). As time has progressed the same needs exist, although forensic investigators are not only attempting to identify the 'authors' or source of malicious software but are also attempting to bypass 'anti-forensic' techniques (Forte & Power 2007).

### **Malware**

The amount of malware propagating on the web is on a steady rise, and malicious software developers are constantly advancing methods by which this concealed malware may spread and infect hosts. This is evident with the continual release of patches and updates for anti-virus and spyware applications. On a recent study initiated by Google it was found that 10 percent of the web pages indexed contained some form of malware (Anonymous 2007). In most instances the malware was stored on third party servers and hence not on the actual server hosting the web site. Whenever a user would attempt to access an infected web site they would be instructed to install an applet without any significant indication of its malicious nature. In the study Google admitted that consumers would find it difficult to protect themselves and may succumb to the associated threats of a malicious workstation.

Malware trends are steering away from traditional viruses and worms which may cause the specific workstation to become unstable and thus are moving towards botnets. Botnets are a collection of hosts under the control of a master who would generally utilise the hosts to carry out malicious tasks including the cracking of cryptographic ciphers or the undertaking of distributed denial of service attacks (Schultz 2006). Certain organisations are utilising malware for unwanted advertising targeted towards a specific host. Alternatively consumers are being victimised by malware which redirects the web browser to malicious clones of their bank's web site (Pemble 2005). As unsuspecting individual's input personal information as per usual, these details are being logged and stored for various malicious financial acts.

According to Secure Computing (2007) Trojans and targeted Spyware accounted for 78 percent of malware activity on the Internet for the month of August. Of that almost 97 percent was targeted at Microsoft Windows based workstations. Avoiding malware for consumers may prove difficult with as many as 11,906 new web sites discovered in August by Secure Computing (2007) containing malware destructive to Microsoft Windows based workstation. Combating malware may also prove potentially difficult for consumers with the need of being informed of the latest Internet scams, installing appropriate operating systems updates and patching anti-virus signatures for their workstation.

### **Malware Forensics**

Nikkel (2006) details the use of a "portable network forensics evidence collector" built on a desktop processor utilising 128 megabytes of memory and an operating system for a desktop workstation. However, the evidence collector virtually mimics a desktop host utilising a honey pot as an intrusion detection system. As this paper demonstrates, a similar system may be developed using commercially sold pre-built hardware managed by an open source operating system in turn reducing the overall cost of producing a malware collector. Utilising a malware collector host for forensic purposes is vital as the "examination of more than 14,000 unique and valid binaries showed that current anti-virus engines have some limitations and fail to detect all malware propagating in the wild" (Baecher et al. 2006, p.183).

In order to ensure a thorough forensic capture and analysis of malware the executable binaries must be preserved in a manner which adheres to the forensic principles. One method by which forensic investigators may then attempt to analyse malicious software is by capturing the binary prior to infection of a system and then apply forensic techniques to identify the source, malicious intent, anti-forensic techniques and targeted system

(Gray et al. 1997). By applying forensic techniques, investigators and researchers may understand and discover new trend patterns but most importantly develop new rule sets for intrusion detection systems (Baecher et al. 2006). In turn this research will improve the detection rate and hence allow forensic investigators to uncover new anomaly intrusion patterns in the future.

### **Nepenthes**

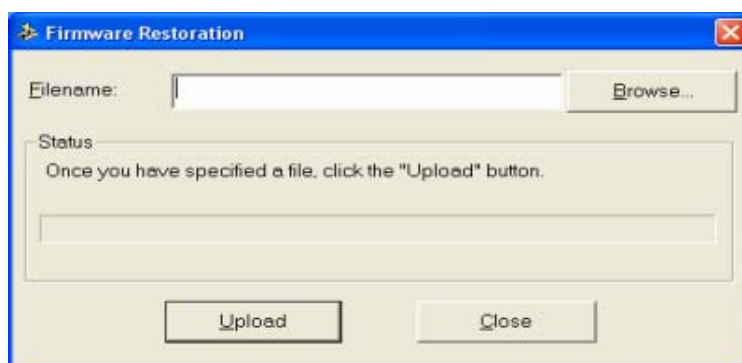
An approach to capture, document and analyse malicious software in a forensic manner is through the use of a honey pot with a pre-configured up to date rule set. Nepenthes is a low interaction honey pot based on the principles of honeypd. Nepenthes operates in a passive mode emulating well documented Microsoft Windows vulnerabilities (Nepenthes 2007b). Utilising a method of deception the attacker believes they are exploiting a vulnerable machine. Nepenthes emulates and responds to the attacks as would any other vulnerable workstation. Hence the attacker believes the exploit has been successfully executed. The malicious software binary is stored safely without infecting any other connected hosts on the same subnet as the Nepenthes server. Nepenthes may operate on many workstations with lower-end hardware and minimal user interaction. The only requirement is an ADSL connection with an available Ethernet port on the router. Rule sets are then configured on the router which forwards all traffic on pre-defined ports to the malware sensor for monitoring and analysis of threats.

Developers of Nepenthes argue that the software is open source and may be recompiled to suit the environmental needs (Nepenthes 2007b).. Network administrators may monitor network activity and determine which exploits may have disrupted certain hosts on the network. Furthermore, the malware collected is free and the binaries can be reverse engineered and analysed to identify country of origin, developers and trends in malware development. As the number of malware activity circulating on the Internet is on a steady rise, Nepenthes makes it simple to counter the threat by collecting, analysing and developing methods to prevent a compromise in future attempts.

### **CONFIGURING AN ASUS WL-HDD**

By default the Asus WL-HDD is shipped with proprietary firmware which is of no use to developers whom wish to execute third party software. Hence, the first task of installing Nepenthes onto the device is to remove the existing firmware and image an operating system which is able to execute Nepenthes successfully. The chosen firmware was OpenWRT. This operating system is highly customisable permitting end-user's to install and remove packages as desired whilst making full use of any additional features the device may incorporate including; wireless and USB interfaces (OpenWRT 2007). The developers of OpenWRT have released precompiled firmware images for the most common embedded system architectures. The selected firmware was "WhiteRussian 0.9 stable" for the Broadcom 2.4 chipset as per the hardware on Asus WL-HDD.

The Asus WL-HDD is shipped with a Firmware Restoration utility (Figure 15) used mainly for restoring the original firmware in the event of an operating system failure or upgrade. However, an alternative firmware may also be imaged utilising the same utility. Whilst there are other approaches to imaging the device including the Trivial File Transfer Protocol (TFTP), the firmware restoration utility is the simplest.



*Figure 15 Asus Firmware Restoration Utility*

Once the device has been successfully imaged, users may use the Secure Shell (SSH) protocol to access the device. Upon connection users are presented with a Linux console interface. The commands utilised to interact

with the device are a replica of those used on a Linux workstation and may be identified by pressing the <Tab> key. By default the device does not have sufficient storage space to house the complete Nepenthes installation with all dependent files. As can be seen the dynamic partition has only two megabytes of non-volatile storage. However, the entire Nepenthes installation requires at least six megabytes, thus additional storage is attached to the device.

```
root@OpenWrt:~# df
```

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
dev/root/	1024	1024	0	100%	/rom
none	7152	40	7112	1%	/tmp
/dev/mtdblock/4	2240	616	1624	28%	/jffs
/jffs	1024	1024	0	100%	/

The Asus WL-HDD whilst marketed as a router is in fact a Network Attached Storage (NAS) device. Hence, utilising its main feature, a forty gigabyte pre-partitioned notebook hard disk was attached as per the Asus documentation. Whilst any hard disk size may be attached to the device, the maximum partition size must be no more than forty gigabytes. The default installation of the firmware does not allow the user to mount any external storage due to missing packages. Thus two packages are installed and automatically configured by the OpenWRT package management system (IPKG). The user has the option of either installing an IDE or USB package dependant on the storage device, in conjunction with EXT2 or EXT3 file system package again dependant on the partition type.

```
root@OpenWrt:~# ipkg install kmod-ide
```

```
Downloading http://downloads.openwrt.org/whiterussian/packages/kmod-ide_2.4.30-brcm-5_mipsel.ipk
```

```
Installing kmod-ide (2.4.30-brcm-5) to root...
```

```
Configuring kmod-ide
```

```
Successfully terminated.
```

```
root@OpenWrt:~# ipkg install kmod-ext3
```

```
Downloading http://downloads.openwrt.org/whiterussian/packages/kmod-ext3_2.4.30-brcm-5_mipsel.ipk
```

```
Installing kmod-ide (2.4.30-brcm-5) to root...
```

```
Configuring kmod-ext3
```

```
Successfully terminated.
```

Once the packages are successfully installed the modules file must be edited to include the newly installed packages as per the instruction through OpenWRT developers. The modules file is located in the /etc directory of the file system.

```
root@OpenWrt:~# vi /etc/modules
```

```
ide-core
```

```
pdc202xx_old
```

```
ide-detect
```

```
ide-disk
```

```
wl
```

```
jbd
```

ext3

Once the module file has been edited and successfully saved a reboot of the device is instantiated. Upon reboot, the module file is loaded with the newly added attributes permitting the external hard disk to be recognised as a storage medium in the partitions table. In this instance the forty gigabyte hard disk has three partitions including a two gigabyte swap partition which is used further.

```
root@OpenWrt:~# cat /proc/partitions
```

major	minor	#blocks	name
3	0	39070080	ide/host0/bus0/target0/lun0/disc
3	1	18595206	ide/host0/bus0/target0/lun0/part1
3	2	10554705	ide/host0/bus0/target0/lun0/part2
3	3	2891700	ide/host0/bus0/target0/lun0/part3

Mounting a specific partition on the disk will now enable the user to interact and save specific files on the disk. As shown below OpenWRT has a slightly different path for locating physical disks and this is the only way that a hard disk may be mounted.

```
root@OpenWrt:~# mount /dev/discs/disc0/part1 /mnt/disk1 cd /mnt/disk1
```

```
root@OpenWrt:~# cd /mnt/disk1
```

```
root@OpenWrt:/mnt/disk1#
```

```
root@OpenWrt:/mnt/disk1# ls -al
```

drwxr-xr-x	4	root	root	4096	Jan	1	00:49	.
drwxr-xr-x	1	root	root	0	Jan	1	2000	..
drwxr-xr-x	5	root	root	4096	Jan	1	00:05	exp

The files required for the Nepenthes installation exceed the amount of non-volatile memory that the Asus device includes onboard by default. Hence, the default installation path for package management system was altered. Rather than installing the packages into the root directory located in non-volatile memory to a specific directory located on physical disk.

```
root@OpenWrt:~# vi /etc/ipkg.conf
```

```
src whiterussian http://downloads.openwrt.org/whiterussian/packages
```

```
src non-free http://downloads.openwrt.org/whiterussian/packages/non-free
```

```
dest root /
```

```
dest ram /tmp
```

```
dest mnt /mnt/disk1/exp/usr/lib
```

### **Installing Nepenthes**

Although Nepenthes is currently released under version 0.2.2, a precompiled version of 0.1.7 was utilised to determine if the system would successfully handle a stable version. Furthermore, Nepenthes has a number of dependencies which have been precompiled and are available as a public download specifically for version 0.1.7 (Nepenthes, 2007a).



```
root@OpenWrt:/mnt/disk1/exp# ipkg install -d mnt nepenthes_0.1.7-0_mipsel.ipk
Installing nepenthes (0.1.7-0) to mnt...
Configuring nepenthes
Successfully terminated.
root@OpenWrt:/mnt/disk1/exp# cd /opt/nepenthes/bin
root@OpenWrt:/mnt/disk1/exp/opt/nepenthes/bin# ./nepenthes
./nepenthes: can't load library 'libadns.so.1'
```

As demonstrated above, installing and attempting to execute the Nepenthes binary will lead to an error with missing library files. By default Nepenthes searches for files within (specific directory). However, due to the limited non-volatile memory resources on the device, the installation of the libraries was transferred to the disk.

```
ipkg install -d mnt adns_1.2-0_mipsel.ipk
...
ipkg install -d mnt curl_7.15-3_mipsel.ipk
...
ipkg install -d mnt file_4.17-_mipsel.ipk
...
```

The remaining library files *libgcc\_s.so.1* and *libstdc++.so.6* were manually transferred to the library location now stored on the disk. In order to ensure Nepenthes may execute successfully, symbolic links were created to the library files located on the disk. It is important to note that the symbolic links are not static by default upon reboot or shutdown of the device hence a script was created which would automatically setup all symbolic links.

```
root@OpenWrt:/mnt/disk1# cat nepenthes-script
ln -s /mnt/disk1/exp/usr/lib/libadns.so.1 /usr/lib/libadns.so.1
ln -s /mnt/disk1/exp/usr/lib/libmagic.so.1 /usr/lib/libmagic.so.1
ln -s /mnt/disk1/exp/usr/lib/libpcrc.so.0 /usr/lib/libpcrc.so.0
ln -s /mnt/disk1/exp/usr/lib/libcurl.so.3 /usr/lib/libcurl.so.3
ln -s /mnt/disk1/exp/usr/lib/libgcc_s.so.1 /usr/lib/libgcc_s.so.1
ln -s /mnt/disk1/exp/usr/lib/libstdc++.so.6 /usr/lib/libstdc++.so.6
root@OpenWrt:/mnt/disk1# ./nepenthes-script
```

After tweaking and configuring the Nepenthes configuration files, the program was finally ready for execution. The execution process was extracted from the Nepenthes documentation provided by the Nepenthes developers (Nepenthes 2007b).

```
root@OpenWrt:/mnt/disk1/exp/opt/nepenthes/bin#
./nepenthes -w /opt/nepenthes -c /opt/nepenthes/etc/nepenthes/nepenthes.conf --version
Nepenthes Version 0.1.7
Compiled on Linux/MIPS at May 23 2006 18:07:36 with g++ 3.4.4 (OpenWrt-1.0)
Started on root running Linux/mips
```

### **Limitations of Nepenthes 0.1.7**

Whilst Nepenthes is capable of operating on embedded devices, the efficiency and stability is questionable. Whilst it is quite noticeable that Nepenthes is monitoring and collecting malware, after a period of approximately thirty minutes the Asus WL-HDD was no longer responding. Hence, a reboot was performed, and certain processes were killed in the hope of ensuring Nepenthes had sufficient resources to operate successfully. After a number of successive attempts the halting times of the device were random and it was concluded that too many malicious binaries were attempting to infect the system.

The OpenWRT documentation stipulates that it is possible to instantiate a swap partition for those devices which are lacking resources to execute processes successfully. After formatting a two gigabyte swap partition on the hard disk, the necessary packages were obtained to allow the operating system to manage swap storage. Further testing of Nepenthes with swap space made no significant difference to the performance of the process. It was hypothesised that the device continuously halted with swap space enabled, as the amount of volatile memory would be consumed faster than the processor was able to execute and write data to the swap.

### **Nepenthes Features**

Nepenthes 0.1.7 features two main methods of analysing binaries once a malicious act has been detected; submit the file to a specific destination on local storage, or submitting the file to the Norman Sandbox (Norman 2007; Riden 2006). The Norman Sandbox will collect, analyse and store the binary and transmit results of the analysis to the email provided in the Nepenthes configuration file. The malware will also be stored locally and hashed to a specific folder again dependant on the configuration file. Alternatively Nepenthes may be configured to only store and log all malware locally without transmitting the binary for analysis to a third party host.

## **CONCLUSION**

The constant rise of malware spreading throughout the Internet is gradually becoming more difficult for consumers and anti-virus vendors to control. An analysis of captured binaries by malware honey pots shows that some anti-virus products are unable to identify the executables as malicious. Whilst organisations and research institutions are able to allocate resources towards forensically analysing malware, this paper demonstrates that a security conscious enthusiast is able to contribute to this research utilising simple and inexpensive hardware. By establishing a malware collector the binaries may then be transmitted to the central Nepenthes server for further analysis.

This paper depicts how an Asus WL-HDD can be turned into a powerful device for hosting various security applications. The onboard specifications for the Asus and Linksys product are identical however the Asus device does permit a user to attach a physical hard disk for additional storage space. Although the paper focused on a malware collector for a SoHo environment, future work could test the feasibility of executing software for example Kismet and logging intrusions natively which if executed on the Linksys WRT54g would require a second host to store log files. Furthermore, increasing the memory capacity of the device could prove useful in testing the effects this has on software which consumes more than the available memory. Since Nepenthes is becoming increasingly utilised amongst the research and security community it may also prove feasible to test the effects of Nepenthes on router's with high performance processors and increased memory capacity.

## **REFERENCES**

- Al-Zarouni, M. (2005). Taxonomy of WRT54G(S) Hardware and Custom Firmware. Paper presented at the Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australian.
- Anonymous. (2007). Google scans Web pages for malware – finds one in 10 infected. *Computer Fraud & Security*, 2007(5), 20.
- Asadoorian, P., & Pesce, L. (2007). *Linksys WRT54G Ultimate Hacking*: Syngress Publishing.
- Baecher, P., Koetter, M., Holz, T., & Dornseif, M. (2006). The Nepenthes Platform: An Efficient Approach to Collect Malware, URL <http://honeyblog.org/junkyard/paper/collecting-malware-final.pdf> Accessed 2 September, 2007

- Forte, D., & Power, R. (2007). A tour through the realm of anti-forensics. *Computer Fraud & Security*, 2007(6), 18-20.
- Gray, A., Sallis, P., & MacDonell, S. (1997). Software Forensics: Extending Authorship Analysis Techniques to Computer Programs. Paper presented at the 3rd Biannual Conference of the International Association of Forensic Linguists, Durham NC, USA.
- Innes, S. (2005). Turning A Linksys Wrt54g, Into More Than Just A Wireless Router. Paper presented at the 3rd Australian Computer, Network & Information Forensics Conference, School of Computer and Information Science, Edith Cowan University, Perth, Western Australia.
- Johnson, C. (2005). ASUS WL-HDD 2.5" - NAS and Wireless AP, URL [http://www.tweaktown.com/reviews/787/2/asus\\_wl\\_hdd\\_page\\_2\\_specifications/index.html](http://www.tweaktown.com/reviews/787/2/asus_wl_hdd_page_2_specifications/index.html) Accessed 2 September, 2007
- Nepenthes. (2007a). Nepenthes Development, URL <http://nepenthes.mwcollect.org/~nepenthesdev/openwrt/> Accessed 11 September, 2007
- Nepenthes. (2007b). Nepenthes finest collection, URL <http://nepenthes.mwcollect.org/documentation/readme> Accessed 14 September, 2007
- Nikkel, B. J. (2006). A portable network forensic evidence collector. *Digital Investigation*, 3(3), 127-135.
- Norman. (2007). Norman Sandbox, URL <http://www.norman.com/microsites/nsic/> Accessed 30 September, 2007
- OpenWRT. (2006). Table of Hardware – OpenWRT, URL <http://wiki.openwrt.org/TableOfHardware?action=show&redirect=toh>, Accessed 9 September, 2007
- OpenWRT. (2007). What is OpenWRT?, URL <http://openwrt.org/> Accessed 26 September, 2007
- Pemble, M. (2005). Evolutionary trends in bank customer-targeted malware. *Network Security*, 2005(10), 4-7.
- Riden, J. (2006). Using Nepenthes Honeypots to Detect Common Malware, URL <http://www.securityfocus.com/infocus/1880> Accessed 2 October, 2007
- Schultz, E. E. (2006). Where have the worms and viruses gone?—new trends in malware. *Computer Fraud & Security*, 2006(7), 4-8.
- Secure Computing. (2007). Secure Computing's Trends in Email, Web, and Malware Threats, URL <http://www.securecomputing.com/index.cfm?skey=1739> Accessed 5 October, 2007
- Snort. (2007). Snort - the de facto standard for intrusion detection/prevention, URL <http://www.snort.org/> Accessed 26 August, 2007
- Spafford, E. H., & Weeber, S. A. (1993). Software forensics: Can we track code to its authors? *Computers & Security* 12(6), 585-595.

## **COPYRIGHT**

[Patryk Szewczyk] ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the evidence is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **A Proof-of-Concept Project for Utilizing U3 Technology in Incident Response**

Marwan Al-Zarouni  
Haitham Al-Hajri  
School of Computer and Information Science  
Edith Cowan University  
usb@marwan.com  
Haitham@MySecured.com

### **Abstract**

*This paper discusses the importance of live forensics and the use of an automated USB based smart data gathering technology to be used in incident response. The paper introduces the technology and its application in incidence response as well as highlight how it works. It also explains the tools that it uses to gather the live data from target systems. The paper also highlights some of the advantages and disadvantages of the technology as well as its limitations. The paper concludes with mentioning the importance of testing the tool and ways it can be developed and taken further.*

### **Keywords**

U3, Live Forensics, Incident Response, Computer Forensics, Network Forensics, Forensics Tools, Automation.

## **INTRODUCTION**

Computer forensics is traditionally looked at as “dead” forensics. This means that the target system analysed is unplugged from the power source. Although dead forensics has less of a chance to modify data on the disk of the target system, it most defiantly loses “live” volatile data from the target system forever.

Live forensics which is more commonly known as “Incident Response” is an important aspect of digital forensics especially when it comes to cases involving network forensics or hacking which are incidents where live volatile data is of upmost importance. The aim of live forensics is to collect volatile data before unplugging the target system and to collect and preserve memory, process, and network information that would be otherwise lost with traditional forensics.

As the name implies, incidence response requires timely access to the target system and live data collection from it directly. This is considered by many as an intrusive way to collect data from the target system because it involves modifying and affecting the target system by using it to perform the live acquisition. Therefore, extensive expertise and knowledge when it comes to the target operating system (OS) must be possessed by live forensic investigators at the scene of the crime. This is because actions taken by them can and will alter the content on the target system. Therefore, the goal is to minimize impacts to the integrity of the system as much as possible while capturing volatile data from the system.

It is not always possible to have a live forensics expert at every incident and therefore there is a need for a basic tool which can be used by anyone with minimal training to allow for live capture of volatile data from target systems with the least interaction with the system as possible.

## **USB BASED INCIDENT RESPONSE**

The purpose of this paper is to introduce a basic proof-of-concept USB-based tool that collects volatile data from target systems automatically and with little interference from the user of the device. This USB device can be used for live forensic data collection in remote areas or in areas where live forensic investigators cannot reach in person or in a reasonable amount of time or if there was a need to satisfy urgent investigation requirements.

### **Police Forces Operating in Remote Areas**

The device targets police forces that operate in a wide geographical area and have limited staff when it comes to live digital forensics. This includes police forces such as Western Australian Police Force (WA Police) which operates the world’s largest single police jurisdiction, an area covering 2.5 million square kilometres with a structure comprising three regions, 14 districts and 163 police stations (WAPolice 2007). The USB tool addresses areas that live CDs cannot address. It collects live volatile data and network and traffic related data that will be or may be lost in an event of a re-boot which is required by live CDs.

## INTRODUCTION TO U3 TECHNOLOGY

U3 technology was developed to allow users to take their data and portable applications with them to any Windows XP based computer and to launch them automatically once the device is inserted in a USB port. Applications written for U3 smart drives were meant to be easy to install and provided users with portability without violating any copyright laws or end-user licences.

### U3 Features

One of the most important features of U3 technology is the seamless launch of applications within the USB drive. This is done by fooling the OS into thinking that the USB drive is in fact two pieces of hardware rather than one. The operating system in this case thinks that the U3 device is a CDROM device and a USB storage device both inserted into the USB port at the same time.

The U3 in fact has two parts within it. The first part is a small part portion of the flash memory drive space containing an ISO image. The second large part is the remainder of the flash memory space which is used to store the actual applications and user data. An ISO image is basically a file with an extension of (.iso). This specifically means that it is compliant with the ISO 9660 standard which defines a file system for CDROM media. Generally speaking though, the term refers to any optical disk image. An ISO image contains both data and metadata such as boot code, structures, and attributes of files. In U3 devices though, the ISO containing portion of the U3 device appears to the computer as an actual CDROM disk within a CDROM drive which is contains an autorun.inf file which is used to automatically launch the U3 application. Figure 1 shows how Windows XP sees the U3 device.



Figure 1: The two devices under device manager and as Removable Disk (F:) and CD Drive (E:)

## U3'S APPLICATION IN LIVE FORENSICS

As highlighted above, there are two features of U3 smart USB technology that distinguish it from normal USB drives. These features are:

- The ability to Auto-run once inserted into the USB port of a Windows XP system.
- Having a read-only ISO based portion

These features can be utilized in the forensic tool as follows:

- The auto-run feature can be used to execute a batch file that in turn runs the forensic tools contained in the read-only part to the USB drive (the CD ROM portion) and store the output from the tools into the re-writable portion of the USB drive.
- Having a read-only the (CD ROM) portion means that the tools can be stored in a tamper proof area which cannot be modified by the target OS thus insuring the forensic integrity of the tools.

## THE AIM OF THE PROJECT

There are several aims and different stages of implementation for this project. However, this paper will focus on the first phase of the project which is a basic proof of concept device that contains the following:

- A Read-only virtual CD ROM portion of the U3 USB device which contains a basic set of too tools
- A Flash disk portion of the USB which is re-writable and will be used to store the collected evidence

The tool also features verification of the collected data via the use of hashing algorithms. The primary goal of the project is to demonstrate the capabilities of U3 technology the application of such technology to the area of live digital forensics.

## **CREATING AND UPLOADING THE ISO IMAGE**

The U3 forensic tool works by auto executing a batch file which in turn executes other command line utilities in a consecutive order. These command line utilities and the batch file itself are placed in the CDROM portion of the U3 drive. This is done by first creating an ISO file containing the batch file and the utilities. MagicISO is the program that can be used for this purpose (MagicISO 2007). The ISO image is then renamed to *cruzer-autorun.iso* and placed in the same folder as the SanDisk's update utility "*LPInstaller.exe*". The update utility is then executed and what it simply does is replace the ISO image on the U3 drive with the one it finds in its folder.



*Figure 2: Contents of installation folder:*

*The update utility "*LPInstaller.exe*" and  
the *cruzer-autorun.iso* file.*

## **PHASE 1 FOCUS AND LIMITATIONS**

The first phase of the project will focus on one operating system (OS) namely Windows XP. The reason for that is that the CDROM auto-run feature is enabled by default in the Windows XP which makes it an ideal candidate for the U3 based forensic tool. Moreover, Microsoft tools can be obtained directly from Microsoft and can be used to capture volatile data from its operating system therefore ensuring that the source of the tools is trustable.

Because the auto-run feature can be disabled in Windows XP, it can be a limitation of the automation of the U3 tool. This can be easily overcome by manually double-clicking on the CDROM icon of the CDROM portion of the U3 device once the device is inserted in the USB port.

## **PHASE 1 TOOL SET**

Phase 1 of the project focuses on using a small selection of utilities to collect some basic volatile information from the target system. The tools used are divided in steps based on the degree of volatility of data collected. Most volatile data is collected first. The steps are as follows:

- Pre Data Collection Step
  1. Collection of Memory Related Data
  2. Collection of Network Related Data
  3. Collection of System Related Data
  4. Collection of Log Files

The following section of the paper explains each of the above steps in detail. It discusses the data collected and displayed in the evidence data file. The U3 tool uses a batch file to execute each of the commands in the order mentioned from step 1 to step 4.

## **PRE DATA COLLECTION STEP**

Before starting the actual collection of data, the batch file first determines the target drive letter where the information from the tools must be saved. This is because drive letter has to be the one associated with the re-writable portion of the U3 USB device. This is simply done by the process of basically finding the drive with a pre-determined folder in its root. This has to be a unique folder name, for example, a folder by the name of

“data\_dump121123787238” or any other unique name. This path is then set by the batch file for the tools to dump their contents in. Then the batch file executes the data collection steps.

## STEP ONE: COLLECTION OF MEMORY RELATED DATA

In this step, the tool will start by collection the memory related data such as, the loaded programs in the memory. The tool used to perform this operation is *mem.exe*, it is an external command line tool, which can be downloaded from Microsoft’s web site, and this tool has a number of extended operations (Microsoft 2007d). In addition to the ability to display the loaded programs in memory, it also can show the status of the programs along with the overall memory usage of the computer. Finally the last extension that will be used in this script is the option classify, where this extension will summarize, the overall memory utilization, along with the available free memory blocks.

## STEP TWO: COLLECTION OF NETWORK RELATED DATA

This step is dedicated to the collection of network related data. In this step, a number of tools are used to collect network related volatile data:

### Display Network Configurations

Using the tool *ipconfig.exe* displays the settings of the network along with other relevant information such as host MAC address, current network host, current IP address used by the host, the gateway IP address and additional network related data. Ipconfig is a standard Windows tool that is available in most of windows platforms but it can also be downloaded from Microsoft’s web site (Microsoft 2007e).

### Display Network Statistics

*Netstat.exe* is another tool from Microsoft, this tool displays the TCP/IP protocol connections on the host machine, in addition to the state of each of those connections, whether they are in lessening, established or in a time wait mood (Microsoft 2007b).

### Display Network Settings

This is achieved by using the *net.exe* command which is an external command line tool from Microsoft (MSDN 2007a). This tool enables the forensic examiner to view the host network, and the network settings. This tool has many extensions with different data outcomes for each extension. Using the command *net start* displays the started windows services. This information can aid the forensic examiner in defining what the starting services on the host machine are. The second extension is the *net session* command which displays all the connected sessions to the host computer. The third command is the *net share* which displays the shared files on the network. This information can alert the examiner to the possibility of exploring different network shred folders. The forth extension of the net command is the *net group* which displays the network domain groups members, which the host machine is a member of. The fifth extension of the net command used in the U3 Forensic USB script is the *net use* which displays the remotely opened folders on the host machine. The sixth command extension for the net tool is the *net user* which displays a list of the user accounts within the host machine. The seventh extension command of the net tool is the *net accounts* which displays the accounts policy of the host machine. Finally the last extension of the net command is the *net view* which displays a list of the computers within the current domain. This option enables the forensic examiner to identify what computers are connected in this domain.

### Display Routing Table

The command *route.exe* is a command line tool that is available in most of windows platforms and also can be downloaded from Microsoft’s website (MSDN 2007b). This tool displays the information regarding the routing table including IP address, default gateway, and network mask. This information produces better understanding of the network configuration and therefore it will help the forensic examiner in his investigation.

### Display Address Resolution Protocol (ARP) Table

Address resolution protocol, is a wide topic, especially when it comes to the digital forensics from the security point of view. Due to the fact that ARP is involved in many of the security issues. Address resolution protocol could be used in many forums of attacks, such as man in the middle, ARP poisoning, and ARP password sniffing (Bryner 2006). *arp.exe* is a command line based tool, from Microsoft, its available on most of windows platforms and could be downloaded from Microsoft web site. This tool displays the ARP table with the current

connections add IP address. The obtained information from the ARP table will provide the forensic examiner a better understanding of the network state.

### **STEP 3: COLLECTION OF SYSTEM RELATED DATA**

This step is devoted towards the collection of relevant information regarding the host machine. The information gathered can be used as a verification of the host machine, in addition to other data that help the forensic examiner to identify any malicious activities on the host machine. The tools used in the step are as follows:

#### **Display System Information**

*Systeminfo.exe* is a command line tool, developed by Microsoft. This tool is available in Windows XP and can also be downloaded from the Microsoft website. This tool illustrates to the forensic examiner all the system information from the security level, configuration of the host machine operating system to the hardware system such as network cards, hard disks and other hardware devices (Microsoft 2007c).

#### **Display Current User**

A small utility program from Microsoft is used for this purpose and it is called *whoami.exe* which can be used to determine the current logged in user to the host machine (Microsoft 2007j).

#### **Display MAC address**

This tool, *getmac.exe* is a command based tool, that display the media access control of the host machine, this tool is ideal for quick retrieval of the host MAC address (Microsoft 2007j).

#### **Display Log Events**

The *psloggedon.exe* is command line tool is a part of PStools which are hosted under sysinternals.com, and is available to download from the Windows website. This tool can identify who have logged into the machine locally (Microsoft 2007f).

#### **Display a List of Open Ports**

*fport.exe* is a freeware tool developed by FoundStone. It is a command line tool that displaces all TCP/IP and UDP connections and relates it all to their host application in addition to displaying the open ports on the host system (Foundstone 2007).

#### **Display a List of the Current Running Process**

*pslist.exe* is a command line tool that displays a list of the current running process in the system, this allows the forensic examiner to verify which process is running, in addition it aids the discovery of malicious processes in the system. Moreover this tool has an extension command to display the process in a tree format which makes it easy to the forensic examiner navigate through the process, this tool is hosted under Microsoft's SysInternals website (SysInternals 2007).

#### **Show Process Services**

*psservice.exe* is a command line tool that views the process services of the host machine, also it shows the configuration of the system and any running stopped process (Microsoft 2007g).

#### **Schedule Process Using AT**

*At* is a scheduling tool that enables the user to preset a task to be carried on in a specific time, this tool displays and schedules process on a host machine. Using this tool will assist the forensic examiner to determining any scheduled processes, or any malicious process on the target machine. This tool is available to download on the Microsoft's website (Microsoft 2007a).

#### **Collect Server Uptime History**

Uptime is a command line tool developed by Microsoft which enables the forensic examiner to collect the uptime statistics of the host machine (Microsoft 2007i).



## STEP 4: COLLECTION OF LOG FILES

Selected tools have been added to the USB toolset to collect log files from the target system. A list of the tools and a brief description of them follows:

### Event Log

*psloglist.exe* is a tool that can view and save a list of the local and remote log events that accrued on the host system. If used without a switch it retrieves more information about the host system. If the tool is used with an extension such as system *psloglist-s system*, it shows the system event log. *psloglist-s application* displays all the log events for the applications on the host system. This in turn allows the forensic examiner to trace the events of those applications. Finally the command *psloglist-s security* allows the forensic examiner to view the security event log. This tool is a command line based tool that can be downloaded from Microsoft (Microsoft 2007h).

### Internet Explorer History File

The *iehv.exe* tool allows the forensic examiner to view a list of the URL address that has been visited on the host machine, this tool can be used as GUI or in a command line. This tool has been developed by NirSoft. The tool is useful when the forensic investigator intends to get a copy of the URL address list from the host machine (NirSoft 2007a).

### Display USB connections list

*USBDeview.exe* is a tool to list the USB devices that are currently connected and any USB that have been used on the machine before, the tool records sufficient information regarding the USB devices that been connected in the host machine (NirSoft 2007b).

This concludes the data collection steps and the tool pops up a dialog box to the user instructing them to remove the USB drive from the target system.

## ADVANTAGES OF THE U3 DRIVE FORENSIC TOOL

One of the main advantages of this tool is that it requires no expertise from the officer at the crime scene. It also requires little action by the officer at the crime scene which means fewer things can go wrong in the investigation. Also, having both read only and read and write portions means that the evidence-containing U3 drive can be shipped to the forensic experts for evaluation. Another advantage of the U3 forensic device is that the re-writable portion of the USB drive can be forensically wiped and sanitized again for re-use in other live forensic cases.

## DISADVANTAGES OF THE U3 DRIVE FORENSIC TOOL

One of the disadvantages of U3 drive forensic tool is cost when compared to the low cost of Live CD based solutions. A 4 Giga Byte U3 enabled USB stick costs between: 50-100 Australia Dollars which can be considered high when compared to Live CDs which cost under 1 Australian dollar. Capacity limitation is another issue and limitation that has to be taken into consideration when selecting tools to be placed into the ISO image and the amount of data collected from the target machine. Another limitation that has to be considered by the users of the U3 USB tool is the Limit of re-writes on Flash media and its limitations.

## PHASE 2 AND BEYOND

Phase 2 of this project will be based on testing the U3 USB forensics tool using established testing standards, to determine its effects on the forensic integrity of the of the target system. Further testing of the U3 forensics tool in regards to compatibility with anti-virus and anti-hacking software should also be tested in the second phase. Even though the tools are saved in the CDROM portion of the USB drive and cannot be deleted by the programs on the target OS, they can still be stopped from execution on the target system which can hinder the investigation.

Further development of the tool can see the introduction of a new set of tools with more data gathering capabilities and further standardized testing at each phase of the project.

## CONCLUSION AND FURTHER WORK

U3 technology has great potential when applied in field of live forensics as demonstrated in this paper. Further development of the project could see the formation of a USB-based framework for live data acquisition. Such a framework can then become the basis on which other forensic tools can be modified and adapted to work with U3 technology. This could then be used to develop truly portable forensic applications fully utilizing U3 technology.

## REFERENCES

- Bryner, J. (2006) Address Resolution Protocol Spoofing and Man-in-the-Middle Attacks, URL [https://www2.sans.org/reading\\_room/whitepapers/threats/474.php?portal=6e96e71bd547149c4b10c3c7a5ce70f0](https://www2.sans.org/reading_room/whitepapers/threats/474.php?portal=6e96e71bd547149c4b10c3c7a5ce70f0), Accessed 2 November 2007
- Foundstone (2007) Foundstone Network Security Free Tools, URL <http://www.foundstone.com/us/resources-free-tools.asp>, Accessed 2 November 2007
- MagicISO (2007) MagicISO, URL <http://www.magiciso.com/>, Accessed 13 November 2007
- Microsoft (2007a) HOW TO: Use the At.exe Command to Schedule a Backup in Windows NT, URL <http://support.microsoft.com/kb/313289>, Accessed 2 November 2007
- Microsoft (2007b) Microsoft Windows XP - Netstat, URL <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/netstat.mspix?mfr=true>, Accessed 2 November 2007
- Microsoft (2007c) Microsoft Windows XP - Systeminfo, URL <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/systeminfo.mspix?mfr=true>, Accessed 2 November 2007
- Microsoft (2007d) MS-DOS 5.0 Internal and External Commands, URL <http://support.microsoft.com/kb/71986>, Accessed 12 November 2007
- Microsoft (2007e) Options for Ipconfig.exe in Windows 2000, URL <http://support.microsoft.com/kb/223413> Accessed 2 November 2007
- Microsoft (2007f) PsLoggedOn v1.33, URL <http://www.microsoft.com/technet/sysinternals/Security/PsLoggedOn.mspix>, Accessed 2 November 2007
- Microsoft (2007g) PsService v2.21, URL <http://www.microsoft.com/technet/sysinternals/utilities/psservice.mspix>, Accessed 2 November 2007
- Microsoft (2007h) PsTools v2.43, URL <http://www.microsoft.com/technet/sysinternals/SystemInformation/PsTools.mspix>, Accessed 2 Nov 2007
- Microsoft (2007i) Uptime.exe Tool Allows You to Estimate Server Availability with Windows NT 4.0 SP4 or Higher, URL <http://support.microsoft.com/kb/232243>, Accessed 2 November 2007
- Microsoft (2007j) Windows 2000 Resource Kit Tools for administrative tasks, URL <http://support.microsoft.com/kb/927229>, Accessed 2 November 2007
- MSDN (2007a) Net.exe Utility, URL <http://msdn2.microsoft.com/en-us/library/Aa939914.aspx>, Accessed 2 November 2007
- MSDN (2007b) TCP/IP Utilities, URL <http://msdn2.microsoft.com/en-us/library/Aa940250.aspx>, Accessed 2 November 2007
- NirSoft (2007a) IE HistoryView: Freeware Internet Explorer History Viewer, URL <http://www.nirsoft.net/utls/iehv.html>, Accessed 2 November 2007
- NirSoft (2007b) USBDeview - View all installed/connected USB devices on your system, URL [http://www.nirsoft.net/utls/usb\\_devices\\_view.html](http://www.nirsoft.net/utls/usb_devices_view.html), Accessed 2 November 2007
- SysInternals (2007) PsTools: Please READ Before POSTING, URL [http://forum.sysinternals.com/printer\\_friendly\\_posts.asp?TID=3748](http://forum.sysinternals.com/printer_friendly_posts.asp?TID=3748), Accessed 2 November 2007
- WAPolice (2007) Western Australia Police - About Us, URL <http://www.police.wa.gov.au/ABOUTUS/tabid/893/Default.aspx>, Accessed 12 October 2007

## **COPYRIGHT**

Marwan Al-Zarouni, Haitham Al-Hajri ©2007. The authors assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **Introduction to Mobile Phone Flasher Devices and Considerations for their Use in Mobile Phone Forensics**

Marwan Al-Zarouni  
School of Computer and Information Science  
Edith Cowan University  
forensics@marwan.com

### **Abstract**

*The paper gives an overview of mobile phone flasher devices and their use for servicing mobile phones, their illegitimate uses and their use in mobile phone forensics. It discusses the different varieties of flasher devices and the differences between them. It also discusses the shortcomings of conventional mobile forensics software and highlights the need for the use of flasher devices in mobile forensics to compensate for the shortcomings. The paper then discusses the issues with the use of flasher devices in mobile forensics and precautions and considerations of their use. The paper goes further to suggest means of testing the flasher devices and suggest some tools that can be used to analyse raw data gathered from mobile phones that have been subjected to flasher devices.*

### **Keywords**

Mobile Forensics, Cell Phone Forensics, Flasher Box, Hex Dumping, UFS-3 Tornado.

## **INTRODUCTION**

The need to address issues with mobile phone forensics is ever important. The number of mobile phone users nowadays surpasses 2.5 billion people across 218 countries and territories (Smith and Pringle 2007). Mobile phone abuse and problems caused by the use of camera devices within mobile phones are also increasing (Tarica 2007). Yet, conventional mobile phone forensic solutions do not seem to keep up with advances in mobile phone technologies. Furthermore, the development cost for supporting less popular mobile phones by such forensic solutions contributes to driving the prices of such forensic solutions higher (Espiner 2007). This is in addition to expensive updates and yearly subscriptions or service agreements that are sometimes needed to get support for the latest mobile phone devices.

New types of devices called "flasher boxes", also known as "flashers", are relatively cheap and are now becoming significant additions to mobile forensic investigators' arsenal of forensic tools. These devices are being used by forensic investigators in Europe and the United States of America to acquire forensic images directly from mobile phone devices (Breeuwsma et al. 2007, Purdue 2007).

## **ABOUT FLASHERS AND THEIR MOBILE SERVICE USES**

Flasher boxes are also known as flashers or clips and they are mobile phone service devices used by mobile phone service providers and shops. They are mainly used to recover user data from dead or faulty mobile phones that otherwise will not provide access to data stored on their internal memory. They can also be used to update or replace software that is stored in the mobile phone's Read Only Memory (ROM). This software is commonly referred to as "firmware" and is usually pre-installed on phones by either the manufacturer of the phone such as Nokia and Sony-Ericsson or phone service providers such as Three Mobile or Telstra.

Flashers are also used to add language support and set regional settings for mobile phones. Changing regional settings can enable a user that bought a mobile phone device from Australia with Telstra-based firmware for example and did not have Arabic language support by default in the firmware to re-flash it with an Arabic-supported firmware supplied by Nokia in the Middle East. Therefore, he or she will have a mobile phone that

now supports the Arabic language and will therefore be able to send and receive Arabic Short Message Service (SMS) messages.

Other uses for flasher boxes include removing or changing carrier settings and unlocking SIM restrictions or carrier based locks or call restrictions. Even though Subscriber Identity Module (SIM) unlocking is legal in some countries such as Australia, it can be illegal in some other countries.

## IMEI AND THE ILLEGAL USE OF FLASHERS

International Mobile Equipment Identity (IMEI) is a unique 15 digit international serial number used to identify a mobile phone handset to a mobile phone network. This number can be used to identify illegal mobile phone handsets. Each time a mobile phone is switched on or a call is made on it, the network provider checks the IMEI number of the handset, then it cross references it with a blacklist register such as the Central Equipment Identity Register (CIER) used in the United Kingdom. If it is on the blacklist then the network will either refuse to send a signal to the phone or will supply a signal but will not allow any outgoing or incoming calls (UnlockMe 2007).

Flashers can be illegally used to change the IMEI number of some mobile phone devices. This in effect enables criminals to illegally re-enable stolen or lost mobile phones that won't be otherwise usable on a certain mobile phone network.

Figure 1 below is a screen shot of the flasher software for UFS3 by SarasSoft that shows the option to change (rebuild) the IMEI number of the mobile device under the Aux features box within the DCTL group of devices options for the Nokia mobile phone brand flashing. It is worth noting that for Nokia, only DCT3 and DCTL group of devices allow for IMEI modification. Newer Nokia mobile phone devices embed the IMEI number in a non-re-writable chip and therefore are not subject to IMEI rebuilding.

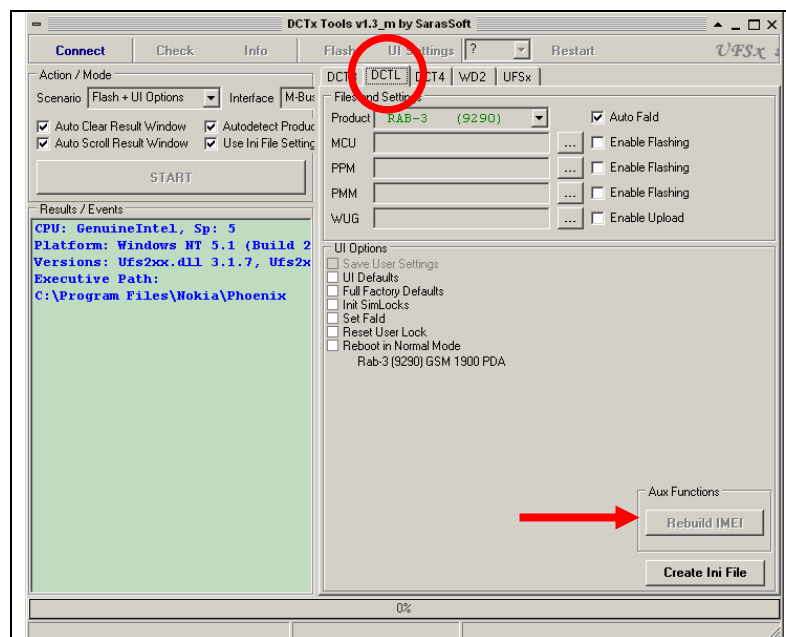


Figure1: Rebuild IMEI option for DCTL range of Nokia mobile phones

## FLASHER BOX COMPONENTS AND VARIETIES

Flashers are a combination of software, hardware and drivers. There are many varieties of flasher boxes covering a wide variety of mobile phones. Therefore, choosing the correct box for a type of mobile phone device or phone model or mobile phone manufacturer can be a daunting task. There are two main categories of flasher boxes:

- Branded Boxes. The features of which include:
  - They are more expensive than their proprietary counterparts.
  - They have well known names and model numbers.
  - They have unique serial numbers.

- Some boxes need activation. Software, updates and support is provided for these boxes. The level of support varies depending on manufacturer of box.
- They are widely used by service technicians.
- They are sold by recognized suppliers and an "approved supplier list" is often found on the manufacturer's website.
- Easier to get support for them in forums and on other support websites.
- Some boxes come with a large amount of cables and can cover both GSM and CDMA phones.
- They do not usually require an external power supply to function. They rely on the USB interface as a power source.
- Unbranded (Proprietary) Boxes:
  - Much cheaper than branded boxes
  - Sometimes match the original flasher boxes in components and functionality.
  - Sometimes combine the functionality and phone support of more than one branded flasher box.
  - Sometimes support the addition of a smartcard from branded flasher boxes.
  - Do not usually come with any software and/or drivers and put the onus on the buyer to come up with the software from other Internet sources.
  - Some boxes come with phone flashing/servicing cables while others do not.
  - Some require an external power supply that is not usually provided with the purchase (IPMart 2007).



*Figure 2: I-Pmart 2 In 1 Flasher Box With Smart Card Holder (IPMart 2007)*

It is worth mentioning that none of the flasher boxes, branded or unbranded, are supported or indorsed by the manufacturers of mobile phones such as Nokia, Sony-Ericsson and others. The top selling branded boxes for the Nokia brand of mobile phone devices include:

- Universal box (UniversalBox 2007).
- JAF box (Odeon 2007).
- MT-Box for Nokia. There is a separate MT-Box for Sony-Ericsson. Even though both boxes are exactly the same and come with a 10 uses trial for the opposite brand (MT-Box 2007).
- UFS 3 tornado: The original flasher box and most widely recommended and used (UFSxSupport 2007).



*Figure 3: UFS 3 Tornado Flasher Box*

Widely used flasher boxes with support for multiple brands of mobile phones include:

- Smart Clip: Motorola, Sendo and others (Smart-Clip 2007).
- GTS Box: Nokia, Motorola, Samsung, Sharp, LG, Sony Ericsson and Siemens (GTS 2007).
- Vygis: LG, Sharp, Sanyo, NEC, BenQ, Alcatel, and Toshiba (Vygis 2007).

There are paid service sites and free phone repair communities that provide the following:

- Video tutorials on setup and use of boxes (FoneFunShop 2007).
- Constantly updated raw ROM images and language packs to flash mobile phone memory with.
- Service manuals and updates for software to cover a wide variety of mobile phones and flasher boxes.

USB flasher dongles that can be used for mobile phone servicing often offer less functionality than USB flasher boxes but may offer other added services such as:

- Remote unlocking and de-branding of phones.
- Credit points that can be used to do things such as IMEI change or unlocking of devices from a service provider.

An example of a product that needs pre-paid credit to unlock and de-brand mobile phones is the JAF device for Windows Mobile Phones (GSMServer 2007). It should be noted however that the JAF device will not work with all phone models running Windows Mobile software. While it supports some phones made by the Taiwanese HTC manufacturer, the do not support devices made by Palm which run Windows Mobile software.



*Figure 5: JAF WM software and USB Dongle (PolPhone 2006)*

## **ISSUES WITH COMMAND BASED FORENSICS SOFTWARE TOOLS**

There are a wide range of software applications and mobile forensic toolkits that claim to acquire data from mobile phones in a forensically sound manner without altering any content in the mobile phone's memory. Such claims however cannot be verified. The basic flaw in these forensic software tools is in the way they gain access to data in the phone's memory. They use command and response protocols that provide indirect access to memory (McCarthy 2005).

Command and response protocols such as AT Commands (AT is short for attention) are commands that were originally developed to control modems to do things like dial, hang up, switch modes, and other modem commands. These commands are utilized by current command based forensic software to communicate with the mobile phone and query it about certain data held in the phone's memory. This means that the forensic software does not have direct access or low level access to data within the phone's memory and in effect treats every mobile phone as a black box. This also means that the software is dependant on the phone's operating system based command to retrieve data in the phone's memory. This could also mean that by querying the operating system, the device could be creating changes to the memory of the device. Because of this dependency on the operating system, such forensic toolkits cannot recover data from dead or faulty mobile phones.

Another flaw with these forensic software applications is that they cannot recover deleted data. This is because they access data at a high level or logical level which means that when a file is deleted, the pointer to that file within the operating system is erased which means that the file is no longer accessible by the operating system or visible to the phone's software. In addition, some mobile phone devices do not respond to AT commands making acquiring them with command based tools impossible (Purdue 2007).

Some command based mobile forensics software were not originally developed for forensic purposes and therefore they could unexpectedly write to the mobile phone device's memory (Horenbeeck 2007). Some forensic software suits such as MOBILedit Forensic 2.2 sometimes require the investigator to install additional software on the target mobile device (MOBILedit 2007). This is in direct violation of the principles of electronic evidence as published by the United Kingdom's Association of Chief Police Officers (ACPO) Good Practice Guide for Computer based Electronic Evidence (ACPO 2003). The guide states the following:

"No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may subsequently be relied upon in court."

It is also in violation of the Guidelines for Best Practice in the Forensic Examination of Digital Technology published by the European Network of Forensic Science Institutes (ENFSI) which states (ENFSI 2006):

"Upon seizing digital evidence, actions taken should not change that evidence." and "Wherever possible no actions taken during the seizing of any evidential material should cause that material to be changed and this is of particular importance when dealing with digital evidence which could be seen as prone to accidental 'tampering'. Where actions have been taken that change the data, this should be fully documented."

Therefore, new ways to gain direct access to data held on mobile phones without resorting to the operating system software or hardware command and response protocols must be utilized in mobile phone forensics. Flasher boxes can provide this direct low level access and therefore they can be considered as a future pathway on the quest for a more optimal acquisition of mobile phones.

## **FLASHER BOXES AND MOBILE PHONE FORENSICS**

The forensic use of flashers is already being taught to future digital forensic examiners in Purdue's College of Technology in the United States of America (Purdue 2007). It is also being used by European investigators in mobile forensic cases (Purdue 2007, Gratzer and Naccache 2007).

Flasher boxes offer access to the phone memory unmatched by command based methods. They also do not require the investigator to install any software on the target mobile phone and therefore do not disrupt the evidence in that way. This in turn means that they follow rules of evidence more closely than command based forensic software tools. But because they are not usually documented, there are no easy methods of determining if they do actually preserve evidence in the phones memory and there is no guarantee that the flashers will work in a consistent manner (Gratzer and Naccache 2007).

Moreover, these devices not approved or tested by the mobile phones manufacturers to work properly on their mobile phone headsets. Furthermore, they are not forensically proven nor tested for forensic soundness.



Because of that, investigators should be careful when attempting to use such devices in mobile phone forensics cases.

Flasher software and hardware were designed for mobile phone servicing needs which means that they are capable of writing to the memory of the phone as well as reading from it. By design, the flasher software does not offer write blocking as with made-for-purpose forensic software. So, the flasher software could be writing to the phone while reading data from it, it effect altering evidence on the phone.

One of the limitations of flasher reading capabilities is dependant on the mobile phone device and/or range of mobile phone devices and the design of the software itself. With some mobile phone devices, full access to memory is blocked and only partial access is possible through the use of flasher software. Some flasher software skip some spare areas in the memory space and do not perform a full copy of the devices memory (Breeuwsma et al. 2007).

Moreover, flasher software present the user with both the memory reading and writing buttons on the same screen which can lead to accidental pressing or the wrong button on the flasher software which could lead to the total loss of evidence from the phone's memory. Figure 6 below shows some of the dangerous buttons that should be avoided by forensic investigators:

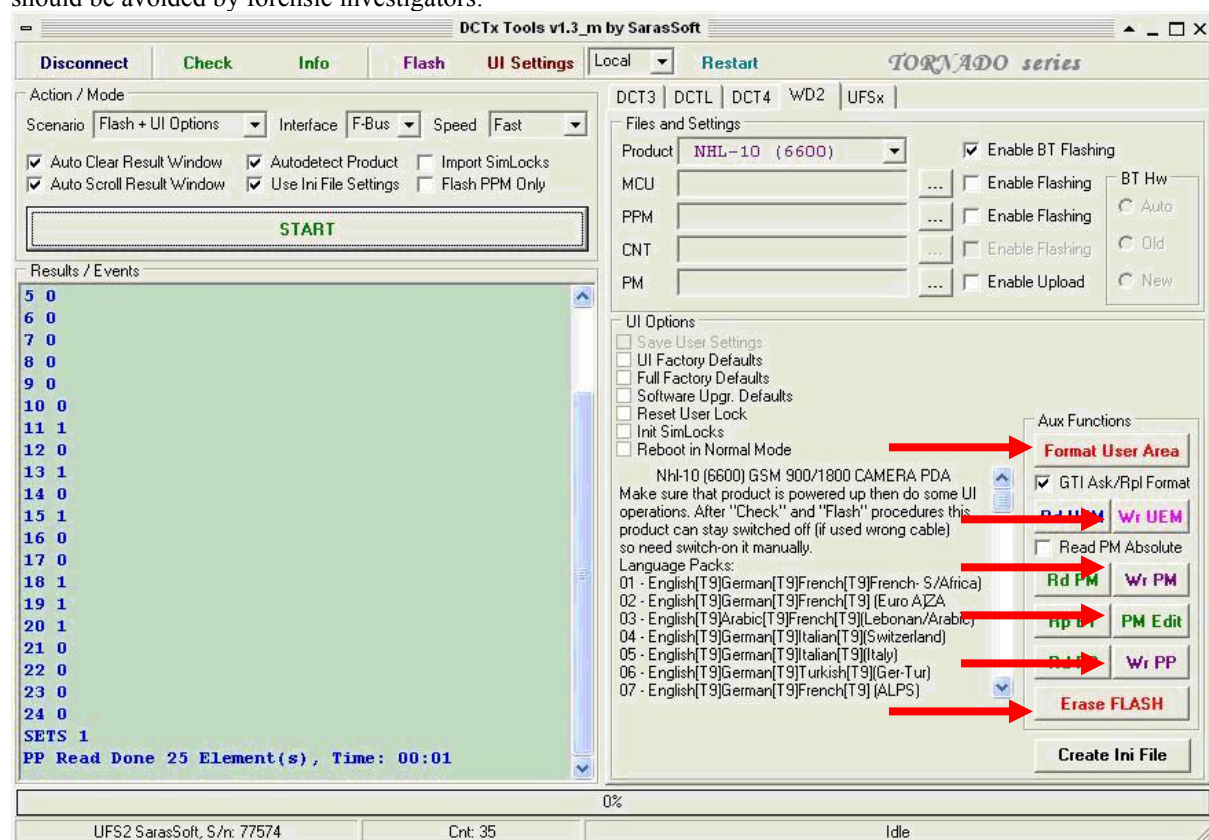


Figure 6: Some of the buttons that should be avoided.

Pressing the wrong write button could also damage the phone's memory in a way that could render the device useless turning the device into a "brick" (Harrington 2007).

## FLASHER CABLES AND INTERFACES

The flasher box typically connects to the mobile device via a special cable made for that phone model. One side of the cable is the RJ-45 standard Ethernet networking cable interface. The other side usually contains a number of pins that contact the mobile phone's service ports through the Joint Test Action Group (JTAG) connection or the Mbus/Fbus connections (Harrington 2007). Figure 7 below shows a Nokia 6600 cable for the UFS3 Tornado Box.





*Figure 7: Connectors on the UFS3 cable for Nokia 6600*

### **Software Installation Precautions**

The appropriate software for each type of flasher box is usually made available through the official support site for the flasher box manufacturer. A username and password are given to each customer once they purchase a flasher box. Each flasher box has a unique serial number that is displayed in the software's dialog box after it's installed.

Choosing the right driver for the type of mobile device can be confusing at times. This is because the support sites usually update the drivers frequently. Sometimes an older version of a USB driver and software bundle will run perfectly with some mobile phone models while a newer USB driver and software bundle will not work with the same device. Information about the best version of driver for each type of device or device range can be found in phone service forums as well as the support site itself.

USB drivers for the flasher box hardware in addition to the phone servicing software should always be installed BEFORE connecting the USB cable to the flasher box. If a certain version of software does not work properly with a mobile phone model or phone range then both the flasher servicing software and the USB drivers associated with it should be completely uninstalled. After restarting the machine after the un-installation the investigator can try another USB driver and software bundle until the appropriate driver and software combination is found. The following section of the paper describes some further considerations when using flasher boxes.

### **CONSIDERATIONS WHEN USING FLASHER BOXES**

Some phones are accessible through service ports located on the bottom of the phone as with some Nokia models such as the 3220 shown below:

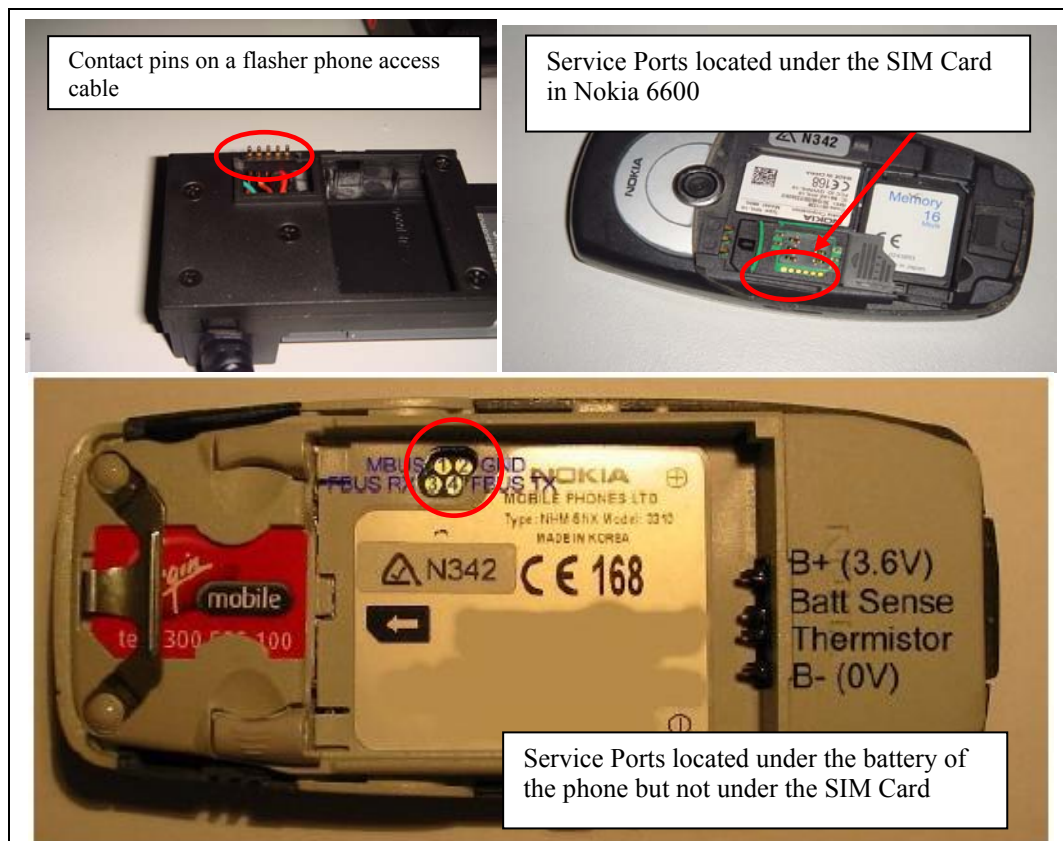


*Figure 8: Nokia 3220 Fbus connections (Harrington 2007).*

Some phones such as the Nokia N95 require an external 9V battery to be connected to the cable to power the phone while operating it with the flasher box. The investigators must always make sure that the battery is fully charged to insure consistent operation and results.

One of the biggest concerns when it comes to acquisitions through the use of flashers is the loss of volatile data. This is because, in some cases, the phone needs to be turned off and the battery for the phone needs to be removed to allow for access to the phone's service ports which are pin contact points on the back of the phone that enable the acquisition of the mobile phone device. These points can be located under the battery of the phone, underneath the SIM card or just below the phone itself without the need to remove the battery of the phone. The location of the service ports is highly dependent on the model of the mobile phone.

Investigators should be careful when they deal with mobile phones with service ports under the SIM card. This is because when SIM cards are removed, some phones tend to lose information associated with them and this information might not be recoverable again. The pictures below show a connection cable with contact pins, a mobile phone with the pin contact points under the battery but not under the SIM card, and another mobile phone where the contact points are located beneath the SIM card (Nokia 6600).



*Figure 9: Contact pins that the cable from the flasher device connects to can be either under the SIM card or not depending on the device model(EmbedTronics 2005).*

On the other hand, if a phone to be investigated has no SIM card inserted in its SIM card slot, it is recommended that a flasher box is used before any other command based tools. This is because if another SIM card is inserted in the phone, or if the phone is powered up normally without a SIM card inserted, it might lose important information about the SIM card previously inserted into it.

Some mobile phones require a SIM card to be inserted into them before allowing access to the phone, this means that command based software will not be able to acquire the phone without a SIM card present. Therefore, through testing of flasher boxes with each phone model is essential before using them for the forensic acquisition of mobile phones. Scenarios such as the ones described above, with and without SIM cards with AT commands first then flashers and vice versa should also be tested. Additional in depth testing considerations and suggestions are listed hereafter.

## **TESTING AND VERIFYING FLASHER ACQUISITIONS**

One of the ways to verify the functionality of flasher boxes is to disassemble the flasher's code and track its behaviour with a logical analyser to understand its effect on the handset. This is not always easy to do and sometimes not possible at all and depends on the competence of the investigators and their knowledge in the practical use of logical analysers (Gratzer and Naccache 2007).

Another way to verify the use of the flasher device is to test it with a large number of mobile phone devices of the same model investigated in a particular case. One study into the use of flashers in mobile forensics suggests that some of these devices be used to develop an experimental protocol or acquisition procedure (Breeuwsma et al. 2007). The protocol is then fine tuned and made more stable and the procedures modified until they produce desired results. The device investigated is then examined using the tested procedure.

Another study takes this further and suggests that the finalized protocol should not be applied to the investigated device after testing the protocols or procedures but rather it should be tested on another set of mobile phones and the occurrences of the following six possible outcomes are then calculated: {information extracted, information not extracted} X {device unaltered, device altered, device destroyed}. This is then carefully documented and all the results are presented to the investigating judge to make a decision on whether to allow the use of flashers in the investigation (Gratzer and Naccache 2007).

## **PHYSICAL IMAGE ANALYSIS TOOLS**

There are many tools that have surfaced in the last couple of years that address the need for the analysis of physical memory dumps from mobile phone devices. The tools range from easy to use tools to tools that require extensive forensics and hex editing and decoding expertise. The following is a rundown some of the tools and their features.

- FTS Hex: The first forensic tool that was developed for the purpose of low level examination of hex dumps from mobile phone memory. It is very basic and mainly sold to law enforcement officers (Knijff 2007, FTS 2007).
- BK forensics' Cell Phone Analyzer: The tool is a simple to use Windows based program that can analyse physical dumps from the following phone manufacturer devices: Sony-Ericsson, Nokia, Blackberry, Motorola and Samsung. The tool does not give the investigator great flexibility to examine the raw data in the dumped image but rather attempts to decode and display phone records, SMS data, pictures and other forms of data to the examiner. An evaluation copy is available to investigators for evaluation purposes from the developer's website (BKForensics 2007).
- Pandora's Box: A new tool developed by Mike Harrington. It recently passed beta testing and is now available in a full retail version. This tool is a very affordable alternative to BK Forensics' Cell Phone Analyser and offers the investigator with more control over the hex decoding process. It can retrieve data such as power down time and date on Series 30 Nokia phones (MFC 2007).
- Neutrino: A mobile phone acquisition device by Guidance Software to be used with Encase version 6. Extracted mobile device data is stored in an EnCase® Logical Evidence File (LEF) format and can be examined via EnCase v6 only (GuidanceSoftware 2007).

Conventional hex editors, decoder software and file comparison tools can also be used to examine the physical dump image and provide the investigator with more flexibility in examining the hex dump but require good knowledge in hex editing, some decoding skills and an eye for recognizing patterns and oddities.

## **CASE HISTORIES**

The first case involves a witness who declared that he recorded a confession with the video camera in his mobile phone. The XRY forensic toolkit was used in his mobile phone to try to recover this piece of evidence but it did not find any videos files on the mobile phone device. Copying raw data from the phone memory did result in the recovery of a plethora of information about videos recorded in the memory and meta-data related to the videos in addition to the recovery of some thumbnails of some videos. The examination also resulted in the discovery of fragments of files in .3gp format that are used for video as well. No evidence was found though to back up the witness's claims (Breeuwsma et al. 2007).

The second case involves the discovery of two yet to be detonated improvised explosive device which used a mobile phone as detonator. The examination of the physical image of the non-volatile memory from the mobile device resulted in the recovery of the history of the of three to four International Mobile Subscriber Identity

(IMSI) of the SIM cards used within those two devices and helped in linking the suspects to the mobile phones and to each other (Knijff 2007). This evidence would have not been recoverable by using command based mobile phone forensic toolkits that rely on logical acquisition of mobile phone devices.

## **CONCLUSION**

As with all digital forensic investigations, it is essential to recover potential evidence from a device in a well documented manner and in a scientifically reliable manner with affecting the data on the device as little as possible. All examinations must be conducted within the law of the country or state in which the investigation is taking place. The usage of flasher boxes requires a high degree of knowledge and competency from the investigator and a great deal of preparation, carefulness and a large amount of research and testing before the examination of a mobile device. Therefore, ample time should be allowed for the examination of such devices.

The future of mobile phones seems to be heading towards convergence with other digital devices such as the MP3 player, GPS navigational devices, laptop computers, camcorders, Personal Digital Assistants (PDA) and digital cameras. This means that more data will be held on such devices and the need for direct access to this data held in flash memory in a forensically sound manner will dramatically increase because of the complexity and the sheer amount of data stored on mobile devices. Therefore, it is very important for law enforcement personnel to familiarize themselves with the use of flasher devices and other means of acquisition and analysis of data held on flash memory.

## **REFERENCES**

- ACPO (2003) Good Practice Guide for Computer Based Electronic Evidence, URL [http://www.acpo.police.uk/asp/policies/Data/gpg\\_computer\\_based\\_evidence\\_v3.pdf](http://www.acpo.police.uk/asp/policies/Data/gpg_computer_based_evidence_v3.pdf), Accessed 1 October 2007
- BKForensics (2007) Cell Phone Analyzer, URL <http://www.bkforensics.com/CPA.html>, Accessed 8 October 2007
- Breeuwsma, M., Jongh, M. d., Klaver, C., Knijff, R. v. d. & Roeloffs, M. (2007) Forensic Data Recovery from Flash Memory. Small Scale Digital Device Forensics Journal, 1.
- EmbedTronics (2005) Welcome to Embedtronics, URL <http://www.embedtronics.com/>, Accessed 6 October 2007
- ENFSI (2006) Guidelines for Best Practice in the Forensic Examination of Digital Technology v, URL [http://www.enfsi.org/ewg/fitwg/documents/ENFSI\\_Forensic\\_IT\\_Best\\_Practice\\_GUIDE\\_5.0.pdf](http://www.enfsi.org/ewg/fitwg/documents/ENFSI_Forensic_IT_Best_Practice_GUIDE_5.0.pdf), Accessed 1 October 2007
- Espiner, T. (2007) Mobile phone forensics 'hole' reported, URL <http://news.zdnet.co.uk/hardware/0,1000000091,39277347,00.htm?r=6>, Accessed 29 September 2007
- FoneFunShop (2007) JAF WM : Flashing Windows Mobile Smart Phones Video Tutorials, URL <http://www.fonefunshop.co.uk/helpzone/jafwm.htm>, Accessed 29 September 2007
- FTS (2007) Forensic Telecommunication Services Ltd, URL <http://www.forensicts.co.uk/phone-forensics.asp>, Accessed 8 October 2007
- Gratzer, V. & Naccache, D. (2007) Cryptography, Law Enforcement, and Mobile Communications, URL [http://info.computer.org/portal/site/security/menuitem.6f7b2414551cb84651286b108bcd45f3/index.jsp?&pName=security\\_level1\\_article&TheCat=1001&path=security/2006/v4n6&file=crypto.xml](http://info.computer.org/portal/site/security/menuitem.6f7b2414551cb84651286b108bcd45f3/index.jsp?&pName=security_level1_article&TheCat=1001&path=security/2006/v4n6&file=crypto.xml), Accessed 27 September 2007
- GSMServer (2007) JAF WM - Windows Mobile based phones solution. Debranding, software update, language change functions implemented, URL <http://gsmserver.com/software/JAF-WM.php>, Accessed 29 September 2007
- GTS (2007) GTS-Box.org, URL <http://www.gts-box.org/>, Accessed 5 October 2007
- GuidanceSoftware (2007) Neutrino, URL <http://www.guidancesoftware.com/products/neutrino.aspx>, Accessed 8 October 2007
- Harrington, M. (2007) Hex Dumping Primer Part 1, URL [http://www.mobileforensicscentral.com/mfc/include/Hex\\_Primer\\_Pt\\_1.pdf](http://www.mobileforensicscentral.com/mfc/include/Hex_Primer_Pt_1.pdf), Accessed 6 October 2007
- Horenbeeck, M. V. (2007) Key constraints in forensic mobile device acquisition, URL <http://www.daemon.be/maarten/mobforensics.html>, Accessed 2 October 2007

- IPMart (2007) I-Pmart 2 In 1 Box With Smart Card Holder, URL <http://www.ipmart.com/main/product/IPmart,2,In,1,Box,With,Smart,Card,Holder,,Packaged,with,187,pcs,,GSM,,CDMA,Cables,16927.php?cat=10&prod=16927&prod=16927>, Accessed 29 September 2007
- Knijff, R. v. d. (2007) Ten Good Reasons Why You Should Shift Focus to Small Scale Digital Device Forensics, URL [http://www.dfrws.org/2007/proceedings/vanderknijff\\_pres.pdf](http://www.dfrws.org/2007/proceedings/vanderknijff_pres.pdf), Accessed 6 September 2007
- McCarthy, P. (2005) Forensic Analysis of Mobile Phones, URL [http://esm.cis.unisa.edu.au/new\\_esml/resources/publications/forensic%20analysis%20of%20mobile%20phones.pdf](http://esm.cis.unisa.edu.au/new_esml/resources/publications/forensic%20analysis%20of%20mobile%20phones.pdf), Accessed 31 September 2007
- MFC (2007) Mobile Forensics Central, URL <http://www.mobileforensicscentral.com/mfc/products/pandora.asp?pg=d&prid=346>, Accessed 8 October 2007
- MOBILedit (2007) MOBILedit! Forensic Application Overview, URL <http://www.mobiledit.com/forensic/default.asp>, Accessed 1 September 2007
- MT-Box (2007) MT-Box Nokia, URL <http://www.mt-box.org/products.php>, Accessed 29 September 2007
- Odeon (2007) JAF by Odeon, URL <http://www.odeon.cn/>, Accessed 29 September 2007
- PolPhone (2006) JAF WM, URL [http://polphone.pl/img/resources/1150455234JAF\\_WM.jpg](http://polphone.pl/img/resources/1150455234JAF_WM.jpg), Accessed 6 October 2007
- Purdue (2007) Expert: 'Flasher' technology digs deeper for digital evidence, URL <http://www.physorg.com/news95611284.html>, Accessed 27 September 2007
- Smart-Clip (2007) Smart-Clip is a professional device for unlocking cell phones. Overview, URL <http://www.smart-clip.com/>, Accessed 5 October 2007
- Smith, M. & Pringle, D. (2007) Global Mobile Communication is 20 years old, URL <http://www.gsmworld.com/index.shtml>, Accessed 28 September 2007
- Tarica, E. (2007) Trouble in Cyberia, URL <http://www.theage.com.au/news/education-news/trouble-in-cyberia/2007/10/19/1192301048347.html?page=fullpage#contentSwap3>, Accessed 28 September 2007
- UFSxSupport (2007) UFS 3 Universal Flasher Software, URL <http://www.ufsxsupport.com/>, Accessed 29 September 2007
- UniversalBox (2007) Universal Box, URL <http://www.universalbox.com/>, Accessed 29 September 2007
- UnlockMe (2007) Is your Mobile phone barred or blacklisted?, URL <http://www.unlockme.co.uk/blacklist.html>, Accessed 29 September 2007
- Vygis (2007) VygisToolbox - Oficial website, URL <http://www.vygistoolbox.com/>, Accessed 5 September 2007

## **COPYRIGHT**

Marwan Al-Zarouni ©2007. The author assigns Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The author also grants a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the author.

## **Pocket SDV with SDGuardian: A Secure & Forensically Safe Portable Execution Environment**

Peter Hannay<sup>3</sup>

Peter James<sup>4</sup>

Secure Systems Limited

Osborne Park

Western Australia.

pjames@securesystems.com.au

### **Abstract**

*Storage of sensitive and/or business critical data on portable USB attachable mass storage devices is a common practice. The ability to transport large volumes of data from the standard place of work and then access and process the data on an available PC at a different location provides both convenience and flexibility. However, use of such USB attachable mass storage devices presents two major security risks; the risk of loss of the portable storage device during transport and the risk of data remnants residing on a PC after accessing the data from the USB storage device. The latter risk is due to the way Windows and third party applications store temporary information on the host PC's hard disk. Even if every effort is made to delete temporary information it may be possible to recover this information by using forensic data recovery techniques such as header analysis and magnetic force microscopy.*

*The Pocket SDV with SDGuardian provides an elegant solution to the aforementioned security risks. The Pocket SDV is a commercially available USB attachable secure hard disk drive. Features of the Pocket SDV include hardware based encryption, strong authentication, differentiated access rights and cryptographically separate partitioning capabilities. Only a user with the correct authentication credentials can gain access to data stored on the Pocket SDV, thus providing assurance if the Pocket SDV is lost. SDGuardian is a proof of concept toolkit that minimises the remnants left on a PC if it is used to process data stored on a Pocket SDV. Forensic examination of the PC, following processing of data held on a Pocket SDV with SDGuardian, should not reveal any remnants of protected data. In this paper an overview of the Pocket SDV is given and its functionality is enumerated. The motivation for SDGuardian is outlined before discussing the design, capabilities and limitations of the Pocket SDV with SDGuardian.*

### **KEYWORDS**

Secure Portable Storage, Forensically Safe Portable Execution Environment, Digital Forensics.

### **INTRODUCTION**

The Pocket SDV is a secure portable USB attachable mass storage device. The Pocket SDV enforces correct user authentication before data on the integral hard disk drive (HDD) may be accessed. Once the user has been correctly authenticated, the SDV allows access to the partitions (drives/volumes) on the Pocket SDV integral HDD. The Pocket SDV provides cryptographically enforced access to data contained on the integral HDD according to a previously configured data access profile for each user. The Pocket SDV operates independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from it. If the Pocket SDV is lost or stolen its owner can be assured that no one can gain access to the data due to strong authentication, nor use digital forensic tools to gain access to the data due to strong encryption.

When a Pocket SDV is connected to a PC (with its own internal HDD, operating system and applications) and sensitive data is accessed (from the Portable SDV) then temporary copies of the data may be saved on the PC's internal HDD by the operating system and/or applications. For instance, data accessed using Microsoft Word from a file stored on the Pocket SDV may leave temporary files inside temporary folders on the PC's internal HDD. As a result these folders may contain sensitive/private data of which the user may not necessarily be

---

<sup>3</sup> Peter Hannay is completing his Honours degree at the School of Computer & Information Science at Edith Cowan University. The research detailed in this paper was performed for Secure Systems Ltd while Peter was performing a Western Australian Government Science & Innovation Scholarship in 2007.

<sup>4</sup> Peter James is registered on a Professional Doctorate programme at the School of Computer & Information Science at Edith Cowan University. Peter is the CEO of Secure Systems Ltd.

aware. If the PC is used by other users it may be possible for those users to find data remnants (temporary copies of files created during of processing of sensitive data) may remain on the PC's internal HDD after the user has detached the Pocket SDV. Also, if the PC does not use encryption technology to encrypt everything written to its internal HDD then it may be possible for digital forensic tools to find sensitive data if the HDD were to be obtained by an inappropriate source.

SDGuardian (Sensitive Data Guardian) is a proof of concept toolkit aimed at addressing the issue of accessing sensitive data from a USB mass storage device like the Pocket SDV in an untrusted environment, e.g. a PC, not owned by the user, is used to process sensitive data held on a Pocket SDV, then subsequently other people use the PC and are able to find remnants of sensitive data left in temporary files. A variety of technologies are employed in order to address the aforementioned issue. SDGuardian may be commercialised, depending upon market demand, and used with the Pocket SDV (or other portable products offered by Secure Systems).

## **AN OVERVIEW OF THE POCKET SDV: DESIGN, METHODS OF USE & CONCEPT OF OPERATION**

### **Overview of Design**

The Pocket SDV is one of a range of SDV products; the product range also includes the Laptop SDV, the SDV Duo and SDV Plus. The primary objective of the Pocket SDV is to provide strong security for data at rest<sup>5</sup>. The Pocket SDV is a cryptographic hardware device (James et al 2004) that asserts total control over its integral HDD at start-up and enforces correct user authentication before data on the Pocket SDV is accessible.

The encryption processes utilised by the Pocket SDV are implemented in the hardware. The hardware implementation of cryptographic functions avoids many of the inherent insecurities of a software-based approach, for example the hardware based approach ensures that keys are not present within the PC RAM; in addition the hardware implementation results in security enforcement that is transparent to the user and not dependant on the resources of the host PC.

Once successful authentication has been achieved the Pocket SDV allows access to data based on pre-defined access rights. The implementation of the Pocket SDV's security mechanisms in hardware coupled with independence from the PC's operating system ensures that successful direct attacks and/or exploitation of operating system vulnerabilities are minimised. Figure 1 provides a pictorial image of the Pocket SDB.



*Figure 1: Image of Pocket SDV<sup>6</sup>*

The Pocket SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the integral HDD. The Pocket SDV operates independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the integral HDD; ensuring the data stored on the hard disk drive is cryptographically secured at rest. A conceptual model of a Laptop SDV topology is given in Figure 2 below.

---

<sup>5</sup> Data at rest is a term that is used to refer to all data in computer storage while excluding data that is traversing a network or temporarily residing in computer memory.

<sup>6</sup> Image of Pocket SDV made available by Secure Systems Limited.



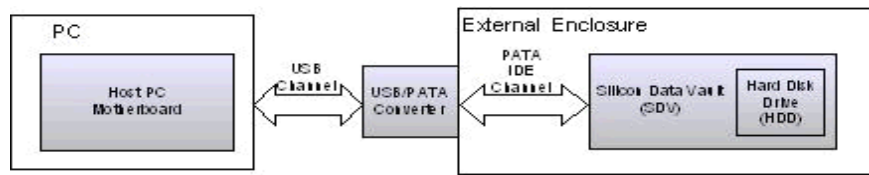


Figure 2 – Conceptual Model of Pocket SDV Topology

There are two modes of authentication supported by the Portable SDV; pre-boot and post-boot authentication. When authenticating using the pre-boot method the host PC will boot off the attached Portable SDV and the Authentication Application (AA) will be launched from the Portable SDV's on-board flash memory. Once successful authentication has been performed the operating system on the PC's internal HDD is loaded. Authentication via the post-boot method requires that the Portable Authentication Application (PAA) is installed on the host PC. When the Pocket SDV is attached the user will be prompted for authentication details by PAA.

The authentication credentials of a Portable SDV user are tied to a specific set of access rights for each partition on the Portable SDV. These rights can be no access, read only or read/write. These user profiles could be used by different individuals or by the same individual, e.g. one user profile could be used for work and one for home. The key functionality and attributes of the Pocket SDV can be summarised as:

- *Pre-boot authentication:* The Pocket SDV achieves a high level of portability by performing authentication before the operating system has loaded. The only requirement is that the host PC provides the capability to allow a USB device to be the first boot device. Pre-boot authentication ensures no hostile software or operating system vulnerabilities can be exploited to obtain the Pocket SDV's authentication credentials.
- *Post-boot authentication:* A Pocket SDV can be authenticated to a PC running an operating system using the Portable Authentication Application (PAA).
- *Full disk encryption:* All data on the Pocket SDV is encrypted. With no data in plain text the opportunities to gain a 'starting point' to break the encryption are eliminated.
- *Sector level encryption:* Encrypting at the lowest level of formatted storage reduces the possibility that pattern matching can be performed to break the encryption.
- *Control of data channel:* Physically positioning the SDV technology between the PC USB controller and Pocket SDV integral HDD ensures all writes are encrypted. Also access control to parts of the HDD can be enforced.
- *Totally independent of PC Operating System:* The Pocket SDV behaves like a standard USB mass storage device and has no dependencies upon the PC operating system to which it is attached.
- *Security functionality implemented in hardware:* Implementing the SDV technology in an Integrated Circuit is recognised as a superior trusted platform; exploiting and attacking hardware is extremely difficult.
- *Multiple Partitions:* Up to 15 partitions can be defined for a Pocket SDV with each partition cryptographically separated from the other partitions by its own cryptographic key.
- *Differentiated Access Rights & User Profiles:* The Portable SDV allows user profiles (roles) to be defined with different authentication credentials and access rights allowing different parts of the Pocket SDV integral HDD to be accessed according to the selected user profile.
- *Audit Log:* Security related events are written to an audit log only accessible by the Pocket SDV administrator role. This log can be used for forensic purposes.

## Methods of Use

The rich functionality of the Pocket SDV allows the device to be configured and used in a number of ways; three configurations are summarised below:

- *Highly Portable Secure Storage Device:* The ability to authenticate via pre-boot authentication results in a highly portable device that can be accessed via any PC capable of booting a USB device. The PAA provides the convenience of accessing data on the Pocket SDV on a fully booted system. Whilst



the Pocket SDV provides Defence7 level security for data at rest, it like all other USB mass storage devices cannot prevent data remnants remaining on a host PC's internal HDD.

- *Highly Portable Secure Storage Device with SDGuardian:* As this paper will show, the SDGuardian toolkit enables a user to attach a Pocket SDV to an untrusted or semi-trusted PC with the assurance that sensitive data remnants be minimised upon completion of data processing.
- *Highly Portable Secure Storage Device with USB Bootable Operating System:* A forensically safe alternative to using the tools and techniques of the SDGuardian is to use a Pocket SDV with a USB bootable operating system, e.g. a version of Linux or Windows PE. For a specific application the bootable operating system approach provides an ideal solution. A disadvantage of this approach is that USB bootable operating systems do not contain the functionality and look and feel of Microsoft Windows XP or Vista, resulting in a disincentive for many users. Current research at Secure Systems includes the development of a business toolkit as part of a bootable operating system with a Pocket SDV.

A stepwise summary of the Pocket SDV pre and post boot authentication is given below to enable a concept of operation to be acquired.

#### **Concept of Operation: Pre-boot Authentication**

The PC must be configured to boot from a USB device at power up with a Pocket SDV attached; operation then proceeds as follows:

- The PC loads a Master Boot Record from the Pocket SDV, which in turn loads an Authentication Application (AA) stored in the Pocket SDV flash memory. N.B. While the AA is running, the user has no access to the Pocket SDV's integral HDD.
- The user is prompted to authenticate.
- The AA passes the entered authentication credentials to the Pocket SDV for authentication. Should the authentication process fail, the AA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts the PC must be powered down and restarted to continue the user pre-boot authentication process.
- Once the user has successfully authenticated, the Pocket SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. Information in the user profile is used by the Pocket SDV to ensure data on its integral HDD is accessed according to the access rights defined for the user.
- The user is then prompted to select one of the following:
  - Boot an operating system from the PC internal HDD.
  - Boot an operating system held on the Pocket PC.
  - Select to authenticate another SDV.
- If the user selects to boot an operating system from the PC's internal HDD the Pocket SDV loads a Master Boot Record for the operating system on the PC's internal HDD.
- The boot process continues and loads the operating system from the PC's HDD.
- The Pocket SDV continues to operate independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the Pocket SDV integral HDD until either the Pocket SDV is detached from the PC USB port or the computer is shut down.

#### **Concept of Operation: Post-boot Authentication**

The PC must have an operating system fully booted and the PAA installed and its underlaying Windows service running; operation then proceeds as follows:

- When the Pocket SDV is attached to a USB port it is detected and a pop up authentication window presented to the user. N.B. The PAA can also be invoked to authenticate a Pocket SDV previously attached. While the PAA is running, the user has no access to the Pocket SDV's integral HDD.

---

<sup>7</sup> The SDV product range has successfully passed rigorous Australian, USA and International cryptographic and security evaluation standards.

- The user enters the authentication credentials and the PAA passes the entered authentication credentials to the Pocket SDV for authentication. Should the authentication process fail, the PAA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts the PAA must be restarted.
- Once the user has successfully authenticated, the Pocket SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. Information in the user profile is used by the Pocket SDV to ensure data on its integral HDD is accessed according to the access rights defined for the user.
- If another portable SDV is detected the user is given the opportunity to authenticate the device.
- The Pocket SDV continues to operate independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the Pocket SDV integral HDD until either the Pocket SDV is detached from the PC USB port or the computer is shut down.

The Pocket SDV makes use of proven encryption standards and strong authentication to provide strong hardware based security for data at rest. The 'set and forget' nature of the device with all encryption handled in hardware results in a secure solution that is transparent to the end user.

## **SDGUARDIAN DEVELOPMENT**

### **Motivation for Development**

The objective of the research was to ensure no data remnants remain on the internal HDD of a PC used to process sensitive data retrieved from a Pocket SDV. Initial investigations considered how a utility (known as SDCleaner) could remove data remnants, which had been written to the PC's internal HDD following the completion of data processing, i.e. a reactive approach was considered.

Research into file system structures identified that with a traditional file system (such as FAT32, NTFS and ext3) there is an area of the file system that provides a table of contents (the TOC). The TOC provides a list of all files located on the file system and where they are located logically (logical locations can be file paths such as C:\Program Files\). In addition to this logical file structure the TOC provides a list of physical locations for each logical item, these physical locations can then be used to read and write data. The key issue for any SDCleaner utility is that when files are deleted the reference in the TOC is simply removed, with the physical data remaining elsewhere on the file system.

Another issue arises when a user launches a program that is used to access sensitive data on the Pocket SDV, this data would then be copied to a temporary location on the host PC's internal HDD. Upon exiting the application it is possible that this temporary data would then be deleted in an insecure fashion (i.e. physical data remains, logical construct removed). This series of events leads to a situation in which an SDCleaner utility would have no means of locating the physical data and is therefore unable to erase said data.

It was therefore decided to adopt a proactive approach and prevent the creation of data remnants on a PC's internal HDD as a result of retrieving and processing sensitive data from a Pocket SDV. The research project was redefined as the SDGuardian. The SDGuardian would act primarily as a preventative measure, with the aim of avoiding the situation where any sensitive information reached the host PC's internal HDD in the first place. Additional measures would also be employed to securely delete sensitive data in situations where prevention is not possible.

### **Implementation of the SDGuardian Toolkit - Design, Capabilities and Limitations**

*Implementation Scenarios:* The research considered a number of different scenarios where a Pocket SDV could be attached to a PC:

- *Scenario 1:* Data processing is to be performed on a "semi-trusted" PC where access to all installed applications is allowed and operation in Windows Administrator mode is permitted. Performance is also a requirement in this scenario, i.e. data processing needs to be performed at close to standard PC processing time.
- *Scenario 2:* Data processing is to be performed on an "untrusted" PC where the installed applications cannot be trusted. However, operation in Windows Administrator mode is permitted.
- *Scenario 3:* As per Scenario 2 but operation must be performed in Windows user mode, i.e. non administrator privileges are available.

To satisfy the requirements of the three scenarios SDGuardian was developed using Junction Points, Secure Deletion and Virtualisation to provide a toolkit for the proactive prevention of sensitive data from a Pocket SDV remaining on a PC's internal HDD. Table 1 shows the tools used to satisfy each scenario.

Scenario	Tool/Technique
1	<i>Junction points</i> are used to prevent specific temporary files from being written to the host PC's HDD. In addition to junction points, secure deletion is used to securely erase the Windows page file after use.
2	<i>Virtualisation technology running in privileged Administrator mode</i> is used to provide a virtualised environment where work can be performed without requiring the use of the PC's installed applications.
3	<i>Virtualisation technology running in non-privileged user mode</i> is used to provide a virtualised environment where work can be performed without requiring the use of the PC's installed applications.

Table 1: Tools/Techniques used in SDGuardian to Meet the Requirements for Each Scenario

#### Junction Points - File System Manipulation

The NTFS file system supports 'junction points' which are similar to symlinks under 'unix like' operating systems. These junction points allow for an empty folder on the file system to be mapped to a different physical location on the disk. The result of this is that two or more logical folders can reference the same physical data. Junction points can reference folders on a different volume or physical storage device.

An initial investigation into the currently available tools for the creation and manipulation of junction points was performed. The details of these tools can be found in the table below.

Software name	Description	License type	Source availability
Junction(Russinovich, 2006)	A command line utility that allows for the manipulation of NTFS junction points	Proprietary	Available
Junction Link Magic (Rekenwonder, 2006)	A GUI based utility that allows for the manipulation of NTFS junction points	Proprietary	Not available

Table 2: A comparison of existing junction point manipulation software

Both Junction and Junction Link Magic were used to evaluate the premise that junction points could satisfy the requirement to remap operating system and application specific directories.

There are a number of shortcomings associated with the use of NTFS junction points. The first of these is that when performing a delete operation on a junction point removal of the associated data from the disk occurs even if that physical data is referenced logically elsewhere in the file system. The window GUI however does not reflect this shortcoming and this deletion is unlikely to be noticed until the user next attempts to access this data. It is due to these factors that care was taken to ensure that junction points were not deleted with the standard tools provided by Windows; instead a custom utility was developed for this purpose.

SDGuardian used junction points to remap common temporary directories onto a partition of the Pocket SDV. This mapping was performed prior to the user accessing sensitive data located on the Pocket SDV. The result of the junction point would be that specific temporary data would never be written to the disk of the host PC, radically reducing the complexities associated with the standard methods of secure erasure.

SDGuardian removes these junction points and recreates the empty temporary directories after the user has finished working with any sensitive material. The temporary data stored on the Pocket SDV is then erased.

As a minimum junction points are used to ensure the security of the Windows temporary directories and the temporary internet directories present on a Windows system.

#### Secure Deletion

Secure deletion tools allow for the forensically sound erasure of data from a hard disk or other storage device, this is achieved by overwriting the data in question several times with different sets of data. Typically the data being written will be all zeros, all ones or the output of a pseudo random number generator (Gutmann, 1996). This is often accomplished by using the Windows disk defragmentation API, this API allows a logical file

location to be resolved into a physical location (MSDN, 2007). Once the physical location of the data is known it is possible to overwrite this data as needed.

An initial investigation into some of the most common tools for secure erasure was performed. The details of these tools can be found in the table below.

Software name	Description	License type	Source availability
Sdelete (Rusinovich, 1999)	Command line secure erasure utility	Proprietary	Available
Eraser (Tolvanen, 1997)	Graphical secure erasure utility	GPL	Available

**Table 3: A comparison of existing secure deletion software**

Both Sdelete and Eraser were used to determine the best strategy to adopt for the implementation of a secure deletion solution in the SDGuardian toolkit.

#### Virtualisation - Application Sandboxing

Application sandboxing attempts to isolate running processes from performing modifications to the host system on which they are being executed. The type of isolation depends heavily on each sandboxing application's specific implementation. There are two main types of application sandboxes, the first attempts to create multiple isolated environments on a system, while the second attempts to limit or prevent specific processes from making changes to the host environment.

There are two main approaches to implementing sandboxing, the first of these is the use of a full virtualised environment. This environment has its own operating system that runs on top of the native operating system. The second approach to implementation involves the use of kernel hooks to isolate a specific application or set of applications from accessing specific system resources (Gibson, 2006).

The use of application sandboxing utilities was investigated. The aim of such utilities is to create an environment which runs on the Pocket SDV to prevent any data from being written to the internal HDD of the host PC, the virtualisation software directs all writes to a specific partition on the Pocket SDV. Unfortunately virtualisation in itself cannot be used as a complete solution due to the nature of Windows virtual memory.

Windows makes use of a 'page file' which acts as virtual memory when adequate physical memory is unavailable. This 'page file' is located on the host PC's internal HDD (Mallery, 2006). The issue arises when a user accesses data stored on the Pocket SDV, the sandbox application can prevent all user level hard disk writes, however the Windows memory management system operates at a kernel level and as such it is not possible to prevent Windows from storing sensitive data in the Windows page file.

The use of a full virtualised environment such as those provided by VMWare or Qemu would allow a user to create a complete operating system environment that would run on top of the host machine's operating system. The advantage of this is that the majority of file system writes would be contained within the virtual machines disk image file, this image would be stored on the Pocket SDV. An advantage of this implementation is that the user would have the ability to install software such as a word document viewer within the disk image, thus negating the need for this software to be present on the host system.

A range of application sandboxing/virtualisation solutions can be found in the table below.

Software name	Description	License type	Source availability
Sandboxie (Tzur, 2006)	An application sandbox utility capable of redirecting file system writes to a specified location.	Proprietary	Not available
Vmware (Vmware, 2007)	A full virtualization application capable of emulating a host computer.	Proprietary	Not available
Parallels (Parallels, 2006)	A full virtualization application capable of emulating a host computer.	Proprietary	Not available
Mojopac (MojoPac, 2006)	A sandboxing utility capable of creating an isolated environment on a host computer. MojoPac is intended to be installed on a portable storage device.	Proprietary	Not available
Qemu (Bellard, 2006)	A full virtualization application capable of	GPL	Available

	emulating a host computer completely in software, as such administrator rights are not needed on the host computer.		
--	---	--	--

Table 4: A comparison of existing sandboxing / virtualisation software

The full range of application sandboxing/virtualisation software specified in Table 4 were tested. Qemu was selected for SDGuardian due to its ability to execute in both Windows Administrator and User modes.

### Implementation Life Cycle

The proposed solution went through a phase of requirements specification and development over the period of several weeks. The SDGuardian was written primarily in the C# programming language. The following features were implemented:

- Junction Points
- Secure Deletion
- Virtualisation

A series of screenshots of the SDGuardian application itself are provided below:

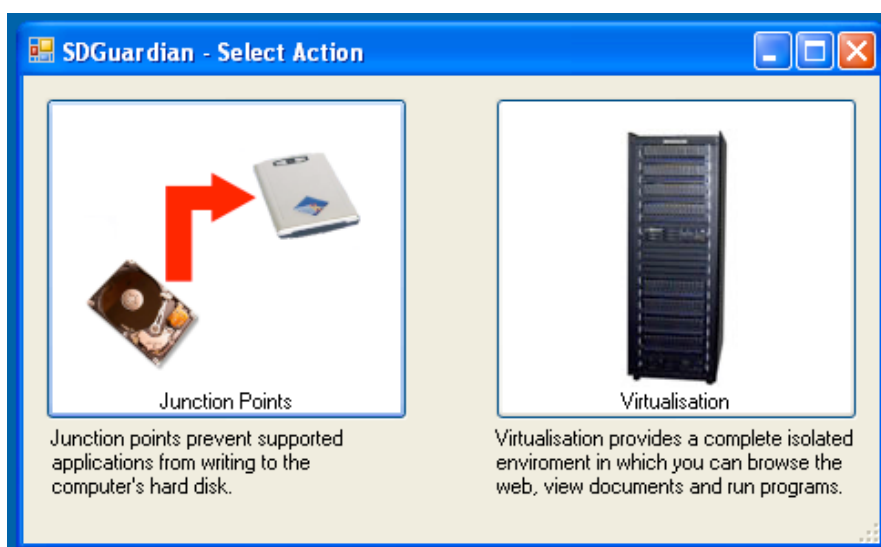


Figure 3: Main screen

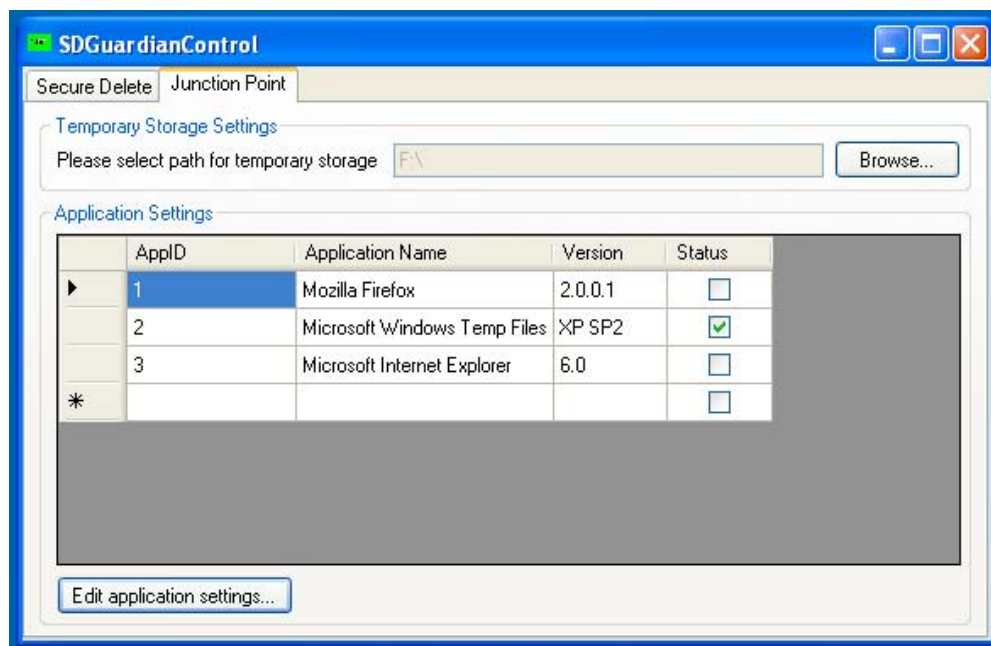


Figure 4: Junction points options dialog

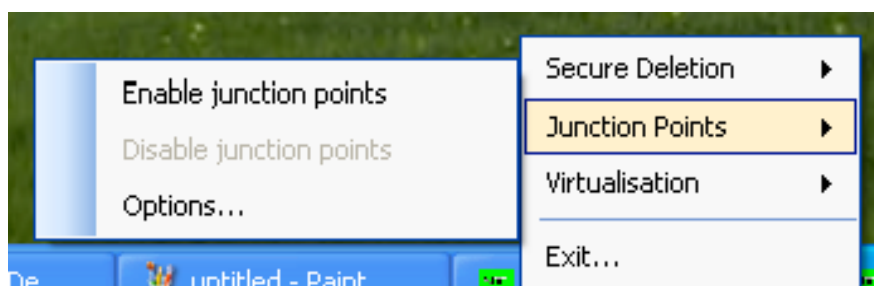


Figure 5: Junction points menu

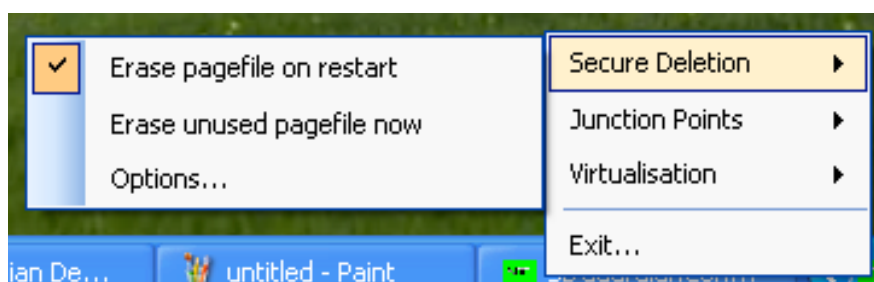


Figure 6: Secure deletion menu

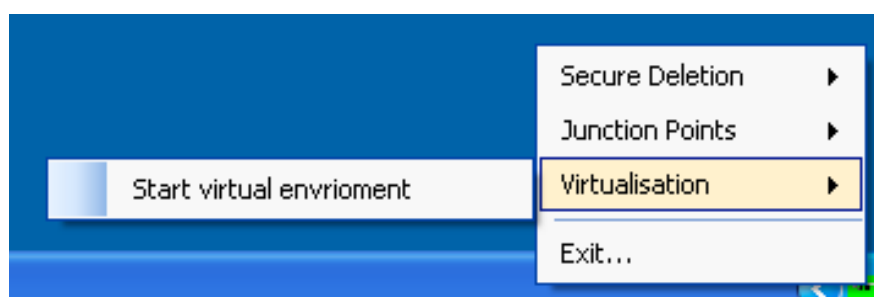


Figure 7: Virtualisation menu

## **CONCLUSION & FUTURE DEVELOPMENT**

The research and development of the SDGuardian was successfully achieved. A proof of concept implementing junction points, secure deletion and virtualisation was developed which meets the original goals of the project. Plans have been made for the continued development of the SDGuardian. These plans include improvements to the virtualisation system employed, additional focus on portability and an in depth forensic evaluation of the software.

## **REFERENCES**

- Bellard, F. (2006). "QEmu." Retrieved January 11, 2007, from <http://fabrice.bellard.free.fr/qemu/about.html>.
- MojoPac. (2006).
- James, P. & Wynne, M 2004, Securing Data at Rest, 2nd Australian Information Security Conference, Edith Cowan University, Perth November 2004.
- Gibson, S. (2006, Oct 26, 2006). "Security Now - Transcript of Episode #63." Retrieved 11 January, 2007, from <http://www.grc.com/sn/SN-063.htm>.
- Gutmann, P. (1996). Secure Deletion of Data from Magnetic and Solid-State Memory. Sixth USENIX Security Symposium, San Jose, California.
- Mallery, J. R. (2001, December 6, 2006). "Secure File Deletion: Fact or Fiction?" Retrieved January 11, 2007, from [http://www.cybercrimelaw.org/documents/secure\\_delete.pdf](http://www.cybercrimelaw.org/documents/secure_delete.pdf).
- MojoPac. (2006). "What is MojoPac?" Retrieved January 11, 2007, from <http://www.mojopac.com/portal/content/what/>.
- MSDN. (2007). Defragmenting Files. Retrieved 8th of January, 2007, from <http://msdn2.microsoft.com/en-us/library/aa363911.aspx>
- Parallels. (2007). "Parallels Workstation." Retrieved January 11, 2007, from <http://www.parallels.com/en/products/workstation/>.
- Rekenwonder Software. (2007). Junction Link Magic, Rekenwonder Software.
- Russinovich, M. (1999). SDelete - Secure Delete, Systems Internals.
- Russinovich, M. (2006). Junction, Systems Internals.
- Tolvanen, S. (1997). Eraser, Heidi Computers Limited.
- Tzur, R. (2006, 14 December 2006). "Sandboxie." Retrieved January 11, 2007, from <http://www.sandboxie.com/>.
- VMware. (2007). "VMware: Virtualization, Virtual Machine & Virtual Server Consolidation." Retrieved January 11, 2007, from <http://www.vmware.com/>.

## **COPYRIGHT**

Secure Systems Ltd ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## Can SDV Technology be Utilised in a Smartphone to Prevent Forensic Analysis?

Peter James<sup>8</sup>  
Secure Systems  
pjames@securesystems.com.au

### Abstract

*Eliminating the opportunities to successfully acquire data from mobile devices is a critical security objective for certain organisations. In particular, Government agencies require assurance that classified data is secured against hostile forensic analysis. The Secure Systems Silicon Data Vault (SDV) is a hardware based data encryption and access control device that has been accredited by the Australian Government to secure classified information held on laptops and portable hard disk drives; hardware is recognised as a superior trusted platform to implement security mechanisms. The SDV's 128bit Advanced Encryption Standard (AES) cryptography, sophisticated key management & access controls and total disk encryption makes the SDV an extremely difficult device from which to acquire data and perform forensic analysis.*

*With the increasing functionality and storage capabilities of Smartphones strong security mechanisms are required by organisations that may hold sensitive data on these devices. Software based security applications exist for Smartphones that provide good security and severely impact the acquisition of data suitable for forensic analysis. If strong hardware based security can be integrated into a Smartphone, forensic analysis could be further constrained. This paper considers the feasibility of implementing the SDV technology into a Palm Treo. An overview of the SDV is given and six security design principles are enumerated. Implementation of the six design principles ensure the SDV provides strong security. The Treo architecture is reviewed and the concept of operation enumerated. The challenges with respect to implementing a Smartphone SDV that is conformant with the security design principles are discussed. Possible Smartphone SDV conceptual designs are presented. The concept of operation, implementation issues and conformance of each conceptual design to the SDV security design principles are discussed.*

### Keywords

Smartphone security, Silicon Data Vault, pre-boot authentication, encryption, PalmOS 5, Treo 650, mobile forensics.

### Introduction

The Secure Systems Silicon Data Vault (SDV) (Armstrong et al 2004, SDVTech 2006) is an award winning (iAward 2006, SoAITI 2005) hardware based data protection solution for mobile applications. The SDV provides protection for data at rest when the data is stored on Integrated Drive Electronics (IDE) Parallel Advanced Technology Attachment (PATA) and IDE Serial ATA (SATA) hard disk drives (HDD). The SDV technology has been implemented into a range of laptop and portable HDDs to provide amongst the strongest commercially available protection for data at rest. The SDV product range has been accredited by the Australian Government to protect classified information. A number of Secure Systems customers have asked if SDV technology could be implemented into Smartphones to provide strong security.

A Smartphone is essentially the merging of mobile phone and Personal Digital Assistant (PDA) technology into the one fully featured product. Typically, a Smartphone provides more features and functions than a standard mobile, for example Smartphones usually have a qwerty keyboard and a push email capability. Smartphones started to emerge in the late 1990s and have now become a key business communication tool for managers, executives and mobile workers. Smartphones use sophisticated operating systems to provide memory management, device control, application management & scheduling and data storage. There are five operating systems that dominate the Smartphone market; Symbian, Windows Mobile, Linux, Blackberry and PalmOS. There is little or no compatibility between the five operating systems and therefore consolidation is likely in the future.

---

<sup>8</sup> Peter James is registered on a Professional Doctorate programme at the School of Computer & Information Science at Edith Cowan University. Peter is the CEO of Secure Systems Ltd.



Palm Inc (Palm 2007), traditionally a vendor of PDAs, produces a range of Smartphones branded the Treo range. Early models of the Treo range came with the PalmOS operating system; however more recent models now support the Windows Mobile operating system as an alternative to PalmOS. The particular Treo model considered in this paper is the Treo 650. The Treo 650 supports only a basic password protection mechanism as standard security. Numerous software security applications exist to provide stronger protection of data stored on the Treo; good examples include Pointsec Mobile (Pointsec 2007) and Teallock (Teallock 2007), both applications provide stronger authentication based access controls and encryption of data stored in internal and external flash memory.

With no standard Smartphone hardware architecture or dominant Smartphone operating system, designing a Smartphone SDV is a challenging proposition; the existing SDV design was able to rely upon established PC and HDD technology standards. Integrating SDV technology directly into a Smartphone's circuitry is considered infeasible (due to the close alliance required with a Smartphone manufacturer like Palm Inc) and it has therefore been assumed that a Smartphone SDV would be an attachable device using an industry standard interface. A high level review performed by Secure Systems (Geddes 2004) on the possible integration of SDV technology into PDAs proposed using the Secure Digital (SD) card interface on a PDA to connect/insert a device containing SDV functionality. A number of Smartphones including the Palm Treo range have an SD card slot. This paper builds upon the idea of using the SD card interface by proposing a conceptual design for a Smartphone SDV device using the Secure Digital Input Output (SDIO) card.

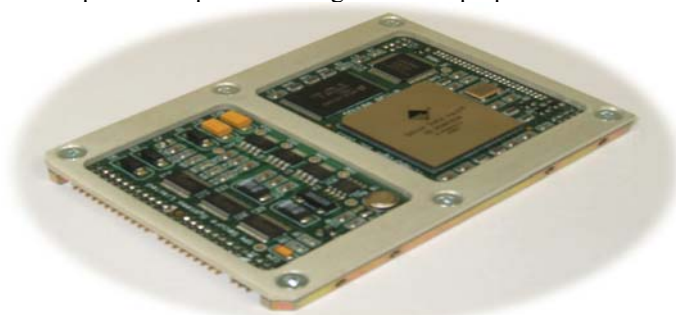
An SDIO card (SDIO 2007) has the same mechanical, electrical, power and signalling attributes of an SD card; an SDIO device can be inserted into an SD card slot and if the host device supports SDIO devices the SDIO device can be operated. Devices that support SDIO cards usually provide the single slot for both SD cards and SDIO cards. The SDIO card provides high-speed data I/O with low power consumption for mobile electronic devices. An SDIO device is able to interrupt the host (e.g. a Smartphone) when the SDIO device is inserted into an SD/SDIO card slot. While an SD card is a storage device, an SDIO card allows hardware accessories to be developed; examples include Wi-Fi and Bluetooth adapters, GPS receivers, TV tuners, cameras, RFID readers and fingerprint readers. The SDIO standard provides a suitable interface to enable an external SDV device to be attached to a Smartphone. The Palm Treo range supports SD cards and SDIO cards/devices.

## **AN OVERVIEW OF THE SDV (LAPTOP SDV)**

The Laptop SDV is the core SDV unit that all other SDV models utilise; it also provides the most appropriate model to use for analysis in this paper. Only the attributes and features of the Laptop SDV necessary to support the discussion on the feasibility of a Smartphone SDV are presented.

### **Overview of Design**

The Laptop SDV (SDVTech 2006) is an alternative secure HDD for a laptop PC; it has the same form factor as a laptop 2.5" HDD. The Laptop SDV replaces the HDD in a laptop; it is connected to the host motherboards IDE controller. Figure 1 below presents a pictorial image of the Laptop SDV.



*Figure 1 – Picture of Laptop SDV*

The implementation of security mechanisms in hardware coupled with total independence of security mechanisms from the laptop's operating system ensures that successful direct attacks and/or exploitation of operating system vulnerabilities are extremely difficult. The primary objective of the SDV is to provide strong security for data at rest<sup>9</sup>. The SDV is a cryptographic hardware device (James et al 2004) that asserts total control over a HDD at system start-up and enforces correct user authentication before data on the HDD is

---

<sup>9</sup> Data at rest is a term that is used to refer to all data in computer storage while excluding data that is traversing a network or temporarily residing in computer memory.

accessible. Once successful authentication has been achieved the SDV allows the laptop's operating system to be loaded. The SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the HDD. The SDV operates independently of the host computer's resources, providing real time encryption and decryption of all data transferred to and from the integral HDD; ensuring the data stored on the hard disk drive is cryptographically secured at rest, even if the SDV is physically removed from the laptop. A conceptual model of a Laptop SDV topology is given in Figure 2 below.

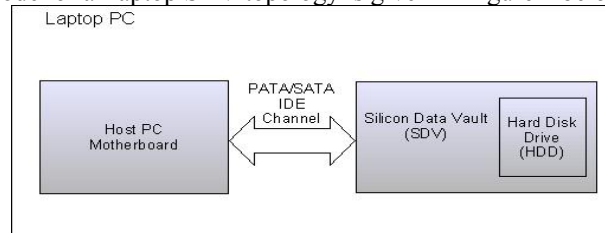


Figure 2 – Conceptual Model of Laptop SDV Topology

### Concept of Operation

At system power-up, a Laptop without an SDV installed will identify the storage devices available and load a Master Boot Record (MBR) from the main boot device; usually the primary HDD. The boot device in turn loads the operating system present on the storage device. While the operating system is running, the user typically has unrestricted access to all sections of the storage media. Conversely a laptop with an SDV inside operates as follows:

- At system power-up the laptop loads the Master Boot Record from the SDV. This in turn loads an Authentication Application (AA) stored in the SDV. While the AA is running, the user has no access to the SDV's integral HDD.
- The user is prompted to authenticate.
- The AA passes the information entered by the user to the SDV for authentication processing. Should the authentication process fail, the AA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts, the computer must be powered down and restarted to continue the user pre-boot authentication process.
- Once the user has successfully authenticated, the SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. This information is used by the SDV to ensure protected hard disk data is accessed according to the profile for each user. The system continues the boot process and loads the OS from the SDV hard disk drive.
- The SDV continues to operate independently of the host computer's resources, providing real time encryption and decryption of all data transferred to and from the SDV integral hard disk storage device until the computer is shut down.

### SDV Security Design Principles

To be considered a valid implementation of SDV technology any Smartphone SDV design would need to encompass the design characteristics that deliver strong security and hence reduce the ability to acquire data. Conformance to the following SDV security design principles will ensure opportunities to use forensic analysis techniques on acquired data are significantly reduced:

4. *Pre-boot authentication*: Performing authentication before the operating system has loaded ensures no hostile software or operating system vulnerabilities can be exploited to obtain authentication credentials.
5. *Full disk encryption*: With no data in plain text the opportunities to gain a 'starting point' to break the encryption are eliminated.
6. *Sector level encryption*: Encrypting at the lowest level of formatted storage reduces the possibility that pattern matching can be performed to break the encryption.
7. *Control of data channel*: Physically positioning the SDV between the PC motherboard and HDD ensures all writes are encrypted. Also access control to parts of the HDD can be enforced.
8. *Totally independent of PC Operating System*: The SDV behaves like a standard HDD and resides beneath the operating system so no attacks or vulnerabilities can be exploited.

- These six SDV security design principles will be used in this paper as criteria to assess if the proposed Smartphone SDV conceptual design can provide the same level of security as the security provided by the Laptop and Portable SDVs.

A Palm product, and the Treo 650 in particular, was selected as the host for a (proposed) Smartphone SDV design due primarily to the information available, from both Palm Inc and the Internet. As with any (closed) proprietary product range, Palm does not publish extensive technical information. However, sufficient information was able to be sourced from a combination of Palm developer documentation (PalmDev Guide 2007) and developer & hacker web sites (Treo Web Sites 2007) that have appeared over the past few years dedicated to the Treo range.

## Overview of the Treo 650 Storage and Memory Management

The Treo 650 does not have an internal HDD. Prior to the 650, the Treo stored all application and data in volatile memory with the PalmOS operating system loaded from masked Read Only Memory (ROM); as a consequence power had to be supplied to the Treo all the time, if power was lost all the data and applications were lost. The Treo 650 has both non-volatile memory for storage of PalmOS, applications and data, and volatile memory for execution of PalmOS and applications. Two other storage devices are available on the Treo 650:

- The Treo 650 has 32MB of non-volatile NAND flash memory (sometimes referred to as a DiskOnAChip) which is structured into two partitions. The first partition contains a boot loader and the compressed PalmOS operating system, known as the ‘ROM’ or ‘compressed ROM’, and occupies approximately 9MB. The second partition is available storage space for applications and data. The second partition is approximately 23MB and is structured into a 512 byte sector file system - the PalmOS Non-Volatile File System (NVFS). Figure 4 presents a memory map of the non-volatile memory.

---

Page 167

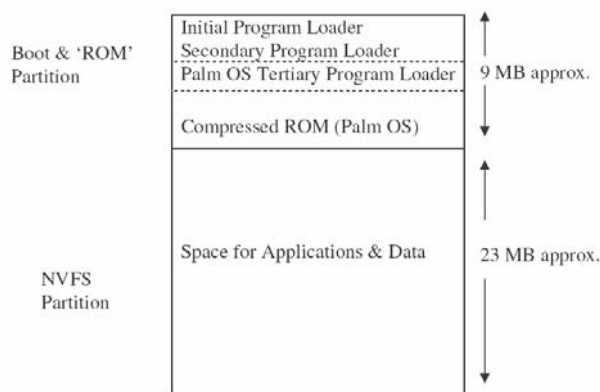


Figure 4 – Non-volatile NAND Flash Memory Map

The Treo 650 has 32MB of volatile SDRAM which is structured into three parts. Approximately 16MB of SDRAM is allocated to the executing PalmOS image, known as the decompressed ROM. A further 5MB is allocated for the PalmOS and application dynamic heap and temporary space. The remaining memory is used for the executing applications and data, known as the DBCache. The PalmOS image is protected from corruption from other executing applications by setting the area of SDRAM to Read-Only. Figure 5 presents a memory map of the volatile SDRAM memory.

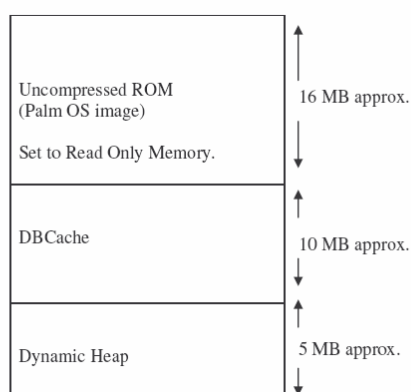


Figure 5 – Volatile SDRAM Memory Map

The Treo 650 will be automatically placed into sleep mode after a defined period to conserve power. Upon receiving a sleep notification PalmOS writes any changes to the applications/data partition. This does not, however, mean that the contents of the SDRAM are removed or PalmOS is stopped. Only a loss of power (depleted battery) or a soft or warm reset causes the SDRAM to be cleaned and a fresh reload of the PalmOS image (from the compressed ROM partition on the non-volatile NAND memory) to occur. A hard reset results in clearing of both the SDRAM and the NVFS partition of the non-volatile NAND memory.

An SD/SDIO card memory has any file system on the card mounted before data can be accessed. The Treo 650 officially supports SD/SDIO cards with up to 2GB of memory. Once the SD/SDIO file system is mounted, applications (and data) on the SD/SDIO card can be loaded into the DBCache and executed<sup>11</sup>. The SIM card memory is accessible and available for storage via certain applications (e.g. SIMBook). SIM memory can vary in size; typically the size of a SIM card's memory is 64KB. It is assumed that an application reading or writing to the SIM card memory would process the data in the Treo's SDRAM.

Protecting data on any SIM card memory has been deemed beyond the scope of this paper.

### Overview of PalmOS – File Systems, DBCache Management & SDIO Slot Management

The Treo 650 comes loaded with PalmOS version 5.4.8; this is a sophisticated operating system providing comprehensive memory, device and file system management in addition to graphical input and output. An

<sup>11</sup> No documentation could be identified to confirm that an application on an SD card is loaded in to the DBCache to execute, but logically it would appear the viable approach as SD memory is block readable/writeable NAND memory where execute in place is not possible

overview is given of the PalmOS file systems, SDRAM management and SDIO slot management capabilities, as these capabilities are relevant to supporting a Smartphone SDV design.

PalmOS 5.4.8 supports two file systems; NVFS for managing information stored in the non-volatile NVFS partition and Virtual File System (VFS) for managing information stored on SD/SDIO cards. SIM card memory is managed by applications that directly read and write to it and is not considered in this paper.

*NVFS:* PalmOS formats the NVFS partition into 512 byte sectors. When an application is invoked it is loaded into the DBCache in the SDRAM together with any data to be processed. Depending upon the application, as data is updated it is written back to the NVFS partition. Also certain PalmOS events (e.g. Treo going into sleep mode) will cause the DBCache to update the NVFS partition to ensure data is not lost. To ensure all available memory is utilised and avoid fragmentation in the NVFS partition, PalmOS will look for available space in NVFS sectors and allocate data to a sector from more than one DBCache record (essentially PalmOS terminology for a file) or downloaded application.

*VFS:* VFS is a unified interface that allows PalmOS to access different file systems on different media types, e.g. VFS allows a FAT 12 or FAT 16 file system on an SD card to be accessed using the same method/procedure call. There appears to be no relationship between VFS and NVFS. It is assumed that an application and data held on an SD/SDIO card is loaded into the SDRAM and that PalmOS performs updates to the SD/SDIO card as required in a similar way in which records are written from DBCache to the NVFS partition.

*DBCache Management:* As the DBCache is only 10MB the PalmOS cache manager has to manage this section of SDRAM efficiently to ensure an application can execute when invoked. Therefore PalmOS will write data back to its source location (NVFS partition or SD/SDIO card) upon an applications instruction or when space is required (typically once the DBCache exceeds 9MB). When application's start, stop, or use memory, fragmentation can occur so the cache manager continuously moves data into contiguous blocks to maximise available SDRAM.

*SDIO Slot Management:* PalmOS has a set of libraries to enable an application or PalmOS to control and read/write to an SDIO card. The PalmOS Expansion Manager detects insertion and removal of the SDIO card and mounts/unmounts any file systems. VFS manager provides the unified file system management. Both the expansion and VFS managers interface to the SDIO card through the SDIO Slot driver which manages power, interrupts, notification of events and essentially all other functionality specified in the SDIO Card Specification (SDIO 2007). Figure 6 presents a conceptual model of the interactions between the libraries and an application.

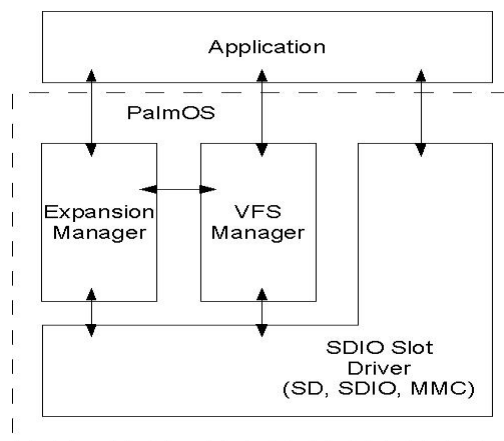


Figure 6 – Model of PalmOS libraries required to support SDIO card

An SDIO card/device can exist as a single function device or as device and storage combination. An SDIO device has its data and executable code located in the SDIO Code Storage area (CSA). The CSA is accessible as a mounted file system through the VFS manager. Once mounted, code in the CSA can be downloaded and 'autorun' in the Treo.

### Concept of Operation

Upon power being supplied for the first time or subsequent to a hard reset the following occurs:

10. The NVFS partition and SDRAM will be empty.

11. The Initial Program Loader (IPL) executes from the non-volatile memory. Whilst the IPL is located in the non-volatile block addressable NAND flash memory, a very small part of the memory allows the IPL to execute in place. The IPL performs some initialisation of the Treo 650 processor and hardware then the IPL loads the Secondary Program Loader (SPL) from non-volatile memory into the Treo's SDRAM.
12. Once loaded control is passed to the SPL, which initialises Treo devices (e.g. LCD and keyboard) and loads the Tertiary Program Loader (TPL) from non-volatile memory into SDRAM, passing control to the TPL once loaded.
13. The TPL decompresses the compressed PalmOS image held in the non-volatile memory and loads the decompressed PalmOS image into SDRAM passing control to it once loaded.
14. The NVFS partition will be available to install applications and data.
15. Any PalmOS function, or application loaded into the NVFS partition or on an SD/SDIO card, will be available for selection and execution.
16. When an application is selected, PalmOS loads the application (and its respective data) from its source location (either the NVFS or an SD/SDIO card) into the SDRAM DBCache and executes it.
17. PalmOS remains active until either power is lost or until a soft reset or system reset occurs.

Upon a soft reset/system reset the following will occur:

18. The NVFS partition will remain unchanged, but the SDRAM will be cleared.
19. Events 2 to 4 above are performed.
20. Events 6 to 8 above apply.

Upon entering and resuming from sleep mode:

21. No clearing of NVFS partition or SDRAM occurs.
22. In sleep mode certain devices are switched off (e.g. LCD screen) to reduce power consumption.

Upon SDIO card/device insertion:

23. Power is supplied to the device and it is initialised.
24. The CSA is mounted, if the device is a combo device the file system on the flash memory is also mounted.
25. Code in the CSA is downloaded and executed.

## CHALLENGES IN ACHIEVING THE SDV DESIGN PRINCIPLES FOR A SMARTPHONE SDV

The overview of the memory capabilities & management, file systems and SDIO card management has highlighted that the Treo 650 with PalmOS 5.4.8 works differently to a laptop PC and its respective HDD. Designing a Smartphone SDV that meets the six security design principles will therefore be difficult and need to consider the following:

*Treo & PalmOS are Closed Technologies:* Whilst Palm and Access Co Ltd (a co-developer of PalmOS 5) do publish good documentation and APIs for PalmOS 5 (which is more significantly informative than documentation available from other proprietary Smartphone operating system vendors e.g. Microsoft and Symbian) detailed descriptions of PalmOS internals appear only to be available to strategic partners. No information appears to be published on the Treo hardware design. Lack of comprehensive hardware and operating system documentation presents a considerable challenge to implementing *the six security design principles* for a Smartphone SDV.

*Different Modus Operandi:* When a laptop is to be used it is turned on and the operating system is booted, work is performed and when finished the laptop operating system is shutdown. A Treo 650, however, is effectively always on; there is an on/off mode but this mode puts the Treo 650 to sleep to conserve power. Provided the battery has sufficient charge and a reset is not performed, the PalmOS image and executing applications (and data) remain active in the SDRAM even when the Treo is 'sleeping'. This different mode of operation (between Smartphone and PC) will make *the pre-boot authentication design principle* difficult to achieve.

*Different Storage Technologies:* A PC's HDD is separated from the PC motherboard and accessed through the IDE bus, hence the SDV is able to be located on the IDE bus between the PC and HDD. Whilst the internal bus

structure of the Treo 650 is not known<sup>12</sup> it is highly likely that the NAND Flash and the SDRAM are closely coupled (i.e. physically connected circuitry). Interposing SDV technology (as it is currently conceived) to control the data channel, between the two memories via an SDIO card would be impossible. Therefore, it follows that fully encrypting the non-volatile NAND Flash (Disk On A Chip) memory it not possible as the boot start point could not be moved to an SDIO device. As a result performing *full disk encryption and controlling the data channel*, as per the SDV design, would not be possible for internal Treo storage.

*NVFS Partition is not Fully at Rest:* An important difference between the Treo 650 and a PC is that data in the NVFS partition (the equivalent of an internal HDD in a Treo) can never be considered to be at rest. As outlined above, PalmOS optimises storage by moving data and filling partially filled sectors in the NVFS partition. This method of storage optimisation may potentially make *sector level encryption* difficult to achieve, e.g. if a sector is encrypted by a Smartphone SDV (assuming it is possible to implement some form of internal sector level encryption beneath PalmOS) following a write request to the NVFS partition and then subsequently the PalmOS NVFS manager performs storage optimisation and changes the contents of the sector, then when the sector is re-read it will not decrypt correctly due to the changed contents of the sector.

*PalmOS & Storage Are Highly Integrated:* PalmOS provides a rich set of functionality to manage memory, file systems and devices in a compact and efficient package. Developing a Smartphone SDV that is totally independent of the operating system and implementing security functionality in hardware would require a large amount of functionality to be built to emulate some of the capabilities of expansion card manager, SDIO slot manager, VFS manager and NVFS manager.

## POSSIBLE SMARTPHONE SDV DESIGN OPTIONS

### Packaging a Smartphone SDV as an SDIO Device

Implementing a Smartphone SDV as an SDIO card/device provides a logical way of retrofitting SDV technology into a Treo 650. It is envisaged that a Smartphone SDV would be packaged into a “block” on the end of an SDIO card which protrudes out of the top of Treo 650 SDIO slot. Figure 7 presents a possible example of how a Smartphone SDV may be packaged.

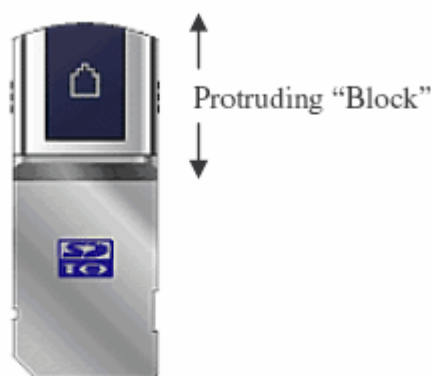


Figure 7 – Possible SDIO Smartphone SDV Packaging<sup>13</sup>

On a Treo 650 the SDIO slot is located on the top of the phone (see Figure 8). It is envisaged that the SDIO Smartphone SDV protruding “block” would be approximately the same height and width as the Treo 650 external aerial (see figure 3 for frontal image of Treo 650 with external aerial). The size of the protruding “block” would, however, vary depending upon the amount of functionality and supporting circuitry required.

---

<sup>12</sup> No detailed documentation could be located on hardware design and schematics of the Treo 650.

<sup>13</sup> Image obtained from SD Worldwide web site May 2007





*Figure 8 – Top down view of SD/SDIO slot on Treo 650<sup>14</sup>*

### **Qualifications to Designs**

The proposed SDV Smartphone design options are conceptual; no qualification has been performed to confirm the:

- Treo 650 can supply sufficient power to the SDIO packaged Smartphone SDV circuitry.
- Required Integrated Circuits (ICs) and supporting circuitry can be packaged into an acceptable size SDIO “block”.
- Cost to build. Neither the development nor manufacturing costs have been estimated to qualify if any of the options are commercially feasible.
- Market demand. No detailed market research has been performed to ascertain if a viable market exists for a Smartphone SDV. A few existing customers indicating interest would not be sufficient to commence development.
- Host Smartphones. The Treo 650 with PalmOS was selected for this research because it is a tried and tested product with good documentation available. However, if a Smartphone SDV was to proceed it would need to be a product that could work with the broadest range of Smartphones and operating systems.

### **Infeasible Functionality**

A number of challenges have been identified with respect to designing a Smartphone SDV that is conformant with the SDV security design principles. Developing functionality for a Smartphone SDV for a Treo 650 with PalmOS 5 would appear to be infeasible for the following areas:

- Hardware based encryption of the NVFS partition
- Sector level encryption of the NVFS partition
- Control of the data between SDRAM and the NVFS partition
- Full disk encryption of the internal “Disk on Chip” non volatile NAND Flash storage.

### **Option 1 – A Full ‘SDV like’ Implementation**

This conceptual design is the most conformant to the six SDV security design principles. It would also be the most difficult to implement – it may, after further investigation, prove infeasible to implement. In this option the proposed core functionality will include:

- Pre-boot authentication.
- Access to data on external flash only possible after successful authentication.
- Hardware based encryption of the external flash memory.
- Sector level encryption of the external flash memory.
- Software based encryption of NVFS partition.

Pre-boot authentication would be achieved by replacing the standard SPL with a ‘secure SPL’ that interfaces with the inserted (SDIO) Smartphone SDV to download an authentication application. Upon successful authentication the SPL loads the standard TPL and the standard PalmOS boot process resumes. Access to data on the external flash memory and the CSA is blocked until successful authentication.

---

<sup>14</sup> Image obtained from Palm Inc web site May 2007



Hardware based, sector level encryption of the external flash memory would be performed on the fly by the crypto capabilities of the Smartphone SDV and would be separate and transparent to the Treo and PalmOS. Encryption key generation will be based on authentication credentials.

Software encryption of the NVFS partition would be achieved by downloading an application from the inserted Smartphone SDV (SDIO) CSA.

It is proposed a Smartphone SDV would mimic the SDV hardware architecture. Figure 9 presents a model of the SDV hardware architecture.

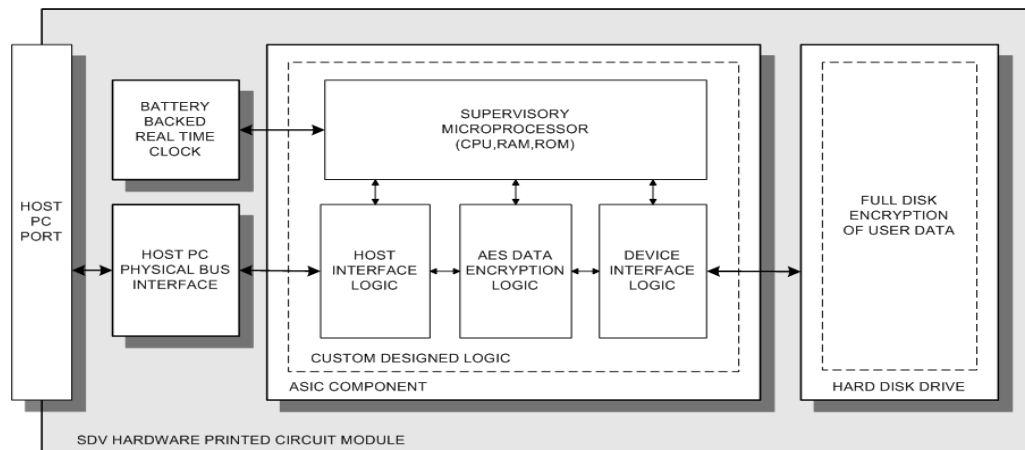


Figure 9 – Model of Key Components and Interfaces in SDV

In a Smartphone SDV the:

- Host PC physical bus interface will be an SDIO slot physical interface.
- Host interface logic will be the SDIO interface logic.
- Device interface logic will be the interface logic to flash memory.

The logic components could be packaged into a single application specific IC or a number of ICs each implementing one or more of the specialist functions.

The SDIO interface logic will work in both a pre-boot and post-boot mode. In pre-boot mode the 'secure SPL' will need to communicate with the Smartphone SDV through SDIO logic to enable the authentication application to be downloaded. In post-boot mode the SDIO logic interface will operate as standard SDIO card. PalmOS will identify the SDIO device and mount the CSA and file system on the flash memory. When the CSA is mounted the SDIO capability to automatically download an application in the CSA will be used to load an NVFS encryption application; it is envisaged that this application will operate in a similar manner to existing software encryption applications (Teallock 2007) that are available, i.e. particular applications and data held in the NVFS partition are selected for encryption with actual encryption taking place once the Treo goes into sleep mode, with decryption occurring once the Treo is woken up.

Data will be written/read to/from the external SDIO flash memory using the VFS manager but as each sector is written/read to/from memory the Smartphone SDV will encrypt/decrypt each sector on the fly unbeknown to the Treo. The hardware and software crypto systems will adopt different key generation and management strategies to ensure that if the weaker software encryption is broken the stronger hardware encryption is not immediately vulnerable.

The downloaded encryption application will include an authentication function that will be activated when the Treo goes into sleep mode. This authentication function will communicate with Smartphone SDV to perform authentication. Only successfully authentication will allow the Treo to exit sleep mode.

Concept of operation

As the NVFS software based encryption will be weaker than the hardware based external flash memory security it would be expected that a user of a Treo will move as many applications and as much data as possible to the external flash memory in the Smartphone SDV.

Insert SDIO Smartphone SDV and immediately perform a soft reset - the following set of events will occur:

26. The IPL loads the Smartphone SDV 'secure SPL'.

27. If the 'secure SPL' does not detect a correctly inserted Smartphone SDV (N.B. for occasions when a soft reset is performed without Smartphone SDV being inserted) the secure SPL behaves like a normal SPL, otherwise the 'secure SPL' will supply power to the Smartphone SDV and load an authentication application from the Smartphone SDV, passing control to the authentication application.
28. The authentication application requests the authentication credentials from the user and passes them to the Smartphone SDV for authentication. If correct authentication occurs the TPL loads and control passes to the TPL; upon correct authentication the Smartphone will have correctly generated the encryption keys for both hardware and software based crypto systems.
29. The TPL decompresses and loads the PalmOS image into SDRAM and passes control to PalmOS
30. PalmOS will detect the Smartphone SDV and mount both the CSA and external flash memory file system. The NVFS encryption application will be downloaded from the CSA and commence execution.
31. Whenever data is written to the external flash memory it will be encrypted, likewise for selected NVFS based applications the respective data will be encrypted when written from SDRAM.

#### Possible Implementation Issues

Theoretically this design option can be implemented. A lot of information is available (Treo Web Sites 2007) on how "customised ROMs" (customised boot loader and PalmOS) and Linux implementations have been installed into a Treo 650, therefore changing the boot loader to include a 'secure SPL' is entirely feasible. However, the following implementation questions arise:

Is performing a soft/system reset user friendly? On a Treo a soft reset requires the battery to be removed and then re-inserted, whilst a system reset requires the reset button positioned under the battery cover to be pushed while pressing the up arrow on the keyboard. Neither reset option is particularly elegant to perform.

Can a concise 'secure SPL' be developed that can detect, power and communicate with an SDIO device? It has been shown that SDIO device management requires comprehensive PalmOS libraries, implementing the necessary software to enable communication with an SDIO device and downloading an authentication application will be challenging.

Can a concise authentication application be developed with the drivers required to accept input from the keyboard and display output on the LCD? As authentication is performed pre-boot none of the PalmOS input/output drivers will be available.

Will performance of external flash based applications be acceptable? As 'SDV like' strong security can only be provided on external memory all, data and applications requiring protection should be located to the external flash memory. Loading from flash is noticeably slower than loading from the NVFS partition. Coupled with 'on-the-fly' encryption, performance may become a barrier to use.

Can the Smartphone SDV be removed while the software encryption application is resident in PalmOS SDRAM without corrupting the NVFS partition? Either the software encryption application will need to detect if the Smartphone SDV has been removed and then perform an orderly closure, or the encryption application is developed so that it can remain a resident application, independent of the Smartphone SDV, to provide on-going protection for applications and data held in the NVFS partition.

Will the Flash Translation Layer (FTL) prevent sector level encryption? The FTL allows NAND flash to be addressed as logical 512 byte sectors and ensures flash 'bad blocks' and 'worn out' blocks are not used. Figure 10 shows how FTL is positioned in the flash memory addressing scheme. The FTL manages the flash while providing a simple logical sector interface to the host system. It is possible that the FTL changes the location of data (FTL discussion 2007) as part of FTL management, i.e. as blocks become bad or worn data is moved; such movement of data may cause major problems for sector level encryption.

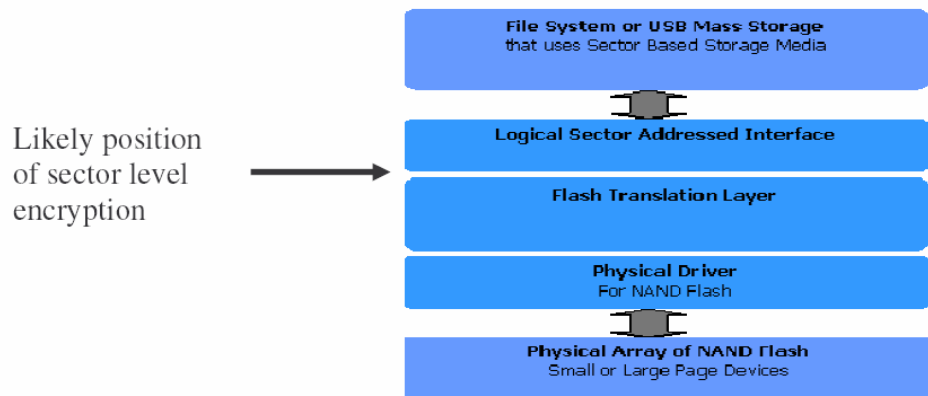


Figure 10 – Position of FTL in Flashing Memory Addressing Scheme

#### Conformance to SDV Design Principles

- *Pre-boot authentication*: Theoretically met.
- *Full disk encryption*: Partially met, external flash memory will be fully encrypted but internal flash will not.
- *Sector level encryption*: Partially met, external flash memory will use sector level encryption but the NVFS partition will use file encryption.
- *Control of data channel*: Partially, SDV technology will be positioned between the Smartphone and external flash memory. Not possible for internal memory.
- *Totally independent of PC Operating System*: Partially, external based flash memory security will be independent of the operating system. However, the NVFS encryption application would utilise PalmOS capabilities.
- *Security functionality implemented in hardware*: Partially, the external flash memory encryption will be implemented in hardware; software encryption will encrypt data in internal memory.

#### Option 2 - Secure Authentication and Software Encryption

In this design option the proposed functionality will include:

- Pre-boot authentication.
- Access to data on external flash only possible after successful authentication.
- Software based encryption of external flash memory located in Smartphone SDV.
- Software based encryption of NVFS partition.

Pre-boot authentication is implemented as described in option1 with access to data on the external flash memory and the CSA blocked until successful authentication.

Software encryption of the external flash memory and the NVFS partition would be achieved by downloading an application from the inserted Smartphone SDV (SDIO) CSA.

A simpler hardware architecture is required consisting of:

- SDIO interface logic.
- A simple (secure) microcontroller to process authentication credentials and perform key generation and management.

The PalmOS SDIO management capabilities will write encrypted data to the external flash memory via the encryption application running on the Treo. No complex encryption hardware is required.

The rationale for developing this option is to provide a secure separate storage device protected by strong pre-boot authentication. Whilst this option will not be as secure as option 1, it will be less complex to develop.

#### Concept of Operation

Insert SDIO Smartphone SDV and immediately perform a soft reset - the following set of events will occur:

32. Events 1 to 4 in option 1 are performed.
33. PalmOS will detect the Smartphone SDV and mount both the CSA and external flash memory file system. The encryption application for both the internal (NVFS partition) and external flash memory will be downloaded from the CSA and commence execution.
34. Whenever data is written to the external flash memory it will be encrypted, likewise for selected NVFS based applications the respective data will be encrypted when written from SDRAM.

#### Possible Implementation Issues

With the exception of pre-boot authentication, this option will be considerably less complex to implement. The option 1 useability and pre-boot authentication implementation issues exist, and due to software encryption of the external flash memory performance is like to be worse than option 1.

To avoid potentially corrupting both the internal and external flash memory either the software encryption application will need to detect if the Smartphone SDV has been removed and then perform an orderly closure, or the encryption application is developed so that it can remain a resident application, independent of the Smartphone SDV, to provide on-going protection for applications and data held in the NVFS partition.

#### Conformance to SDV Design Principles

- *Pre-boot authentication*: Theoretically met.
- *Full disk encryption*: Partially met, external flash memory would be fully encrypted, albeit using software encryption.
- *Sector level encryption*: No.
- *Control of data channel*: No.
- *Totally independent of PC Operating System*: Partially, pre-boot authentication will be performed before the operating system is loaded.
- *Security functionality implemented in hardware*: No.

#### Option 3 – Secure External Storage

This design option is the least conformant to the SDV security design principles. It will be a simple SDIO device providing:

- Post-boot authentication.
- Access to data on external flash only possible after successful authentication.
- Software based encryption of external flash memory located in Smartphone SDV.
- Software based encryption of NVFS partition.

No soft/system reset will be required as the Smartphone SDV will be inserted into a booted Treo. The Smartphone will operate like a standard SDIO device, i.e. upon insertion into the SDIO slot the Smartphone SDV will be powered and notify PalmOS of its existence, the CSA in the Smartphone SDV will be mounted and the encryption application downloaded. In this option the Smartphone SDV relies upon the PalmOS SDIO management libraries.

This option offers comparatively little advantage over currently available software encryption applications and an SD card. The major difference is that access to the Smartphone SDV external flash memory is blocked until authentication is complete.

#### Concept of Operation

Insert Smartphone SDV into the SDIO slot of a full powered and running Treo 650 – the following events will occur:

35. Power is supplied to the Smartphone SDV and it is initialised.
36. The Smartphone SDV CSA is mounted together with the file system on the Smartphone SDV flash memory.
37. An authentication application is downloaded from the Smartphone SDV CSA.

38. The user will be prompted to enter authentication credentials.
39. If authentication is successful, the software encryption application in the CSA is downloaded and executed. No access to the external flash memory will be allowed until successful authentication.
40. Whenever data is written to the external flash memory it will be encrypted, likewise for selected NVFS based applications the respective data will be encrypted when written from SDRAM.

#### Possible Implementation Issues

There should be relatively few implementation issues. Standard SDIO hardware can be used, no specialist ICs or microcontroller will be required. The implementation issues with respect to the software encryption application identified in option 2 apply to this option.

#### Conformance to SDV Design Principles

- *Pre-boot authentication*: No.
- *Full disk encryption*: Partially, external flash memory would be fully encrypted, albeit using software encryption.
- *Sector level encryption*: No.
- *Control of data channel*: No.
- *Totally independent of PC Operating System*: No.
- *Security functionality implemented in hardware*: No.

## CONCLUSION

A comprehensive review of the hardware and software architecture of a sophisticated Smartphone has been performed to identify if SDV technology can be integrated into a Smartphone to make it more secure and restrict the opportunity for acquire data and perform forensic analysis. Three conceptual design options have been presented and assessed against SDV security design principles with varying degrees of compliance.

So, can SDV technology be utilised in a Smartphone to prevent forensic analysis? There is no clear yes or no answer. It has been shown not all of the SDV security features, as currently conceived, can be integrated into a Smartphone, e.g. control of the data channel and sector level encryption for internal storage. However, some SDV functions can be integrated into a Smartphone SDV that would strengthen security and virtually eliminate the opportunity to acquire meaningful data for forensic analysis.

If the Smartphone SDV is captured in an authenticated state (whilst in a Treo) then the opportunity exists to acquire sensitive data. If however, sensitive data and applications are held in the Smartphone SDV external flash memory and the Smartphone SDV is removed from the SDIO slot when it is not in use, acquiring sensitive data can be prevented.

Future work is planned to both consider other options for a Smartphone SDV and develop a proof of concept Smartphone SDV based on the approach proposed in this paper.

## REFERENCES

- Armstrong A, Wynne M, O'Shea A 2004, Who has the keys to the vault? Protecting secrets on Laptops, IEEE Information Assurance Workshop 2004.
- FTL discussion 2007, Mobile Forensics class discussion, School of Computer and Information Sciences, Edith Cowan University, May 2007.
- Geddes 2004, Mike Geddes, PDA Security, Internal Discussion Paper, Secure Systems Limited, 2004.
- iAward 2006, Australian Information Industries Association, iAward Competition Security Category, URL <http://www.aiia.com.au/i-cms.isp?page=1346>
- James P, Wynne M 2004, Securing Data at Rest, 2nd Australian Information Security Conference, Edith Cowan University, Perth November 2004.
- Palm 2007, Palm Inc URL <http://www.palm.com>
- PalmDev Guide 2007, Palm® Developer Guide, Palm OS Platform Software and Hardware Rev. F April 30, 2007

Pointsec 2007, Pointsec Mobile Technologies Inc, URL <http://www.pointsec.com>

SDIO 2007, SD Specifications Part E1, SDIO Simplified Specification, Version 2.0, 8/2/07, Technical Committee, SD Card Association.

SDVTech 2006, SDV Technical Overview, SSL-TD 0098, Version 1.4, 14/7/06

SoAITI 2005, Secrets of Australian IT Innovation Competition Security Category, URL [http://www.dcita.gov.au/\\_\\_\\_data/assets/pdf\\_file/68179/2005\\_Secrets\\_of\\_IT\\_Inovation\\_competition\\_winners.pdf](http://www.dcita.gov.au/___data/assets/pdf_file/68179/2005_Secrets_of_IT_Inovation_competition_winners.pdf)

Teallock 2007, Teallock User Manual, Version 7.2, TealPoint Inc.

Treo 650 2007, Product description and specification of Palm Treo 650, URL <http://www.palm.com/au/products/smartphones/treo650/>, accessed May 2007.

Treo Web Sites 2007, URLs (accessed May 2007)

<http://www.grack.com/blog/articles/2006/02/27/treo-650-memory-management>

[http://www.shadowmite.com/wiki/index.php/The\\_Treo\\_650\\_Bootloader](http://www.shadowmite.com/wiki/index.php/The_Treo_650_Bootloader)

<http://www.grack.com/blog/articles/2006/02/07/the-lowdown-on-dbcache-and-rom-size>

<http://mytreo.net/archives/2005/07/living-with-nvs-on-your-treo-650.html>

<http://mytreo.net/treofaq/Treo650FileManagement>

<http://doc.trolltech.com/qtopia4.2/greenphone-integration-guide.html>

<http://hazelware.luggle.com/archive.html?2005.2>

## **COPYRIGHT**

Secure Systems Ltd. ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **A forensically tested tool for identification of notebook computers to aid recovery: LIARS phase I proof of concept**

Peter Hannay, Andrew Woodward and Nic Cope  
School of Computer and Information Science  
Edith Cowan University  
Perth, Western Australia  
peter@peterhannay.com  
a.woodward@ecu.edu.au  
ncope@student.ecu.edu.au

### **Abstract**

*The LIARS tool was designed to enable identification, and potentially the return, to the rightful owner of stolen laptop or notebook computers. Many laptops are discovered by Police, but time constraints prevent recovered devices from being identified. This project has produced a proof of concept tool which can be used by virtually any police officer, or other investigator, which does not alter the hard drive in any fashion. The tool uses a modified version of the chntpw software, and is based on a forensically tested live Linux CD. The tool examines registry hives for known location of keys which may provide information about the owner of the laptop. This paper outlines the successful first phase of the project and looks at future directions.*

### **Keywords**

Forensic tools, software validation, linux, chntpw

### **INTRODUCTION**

Loss of corporate data is an on-going and increasing problem. There are both reports of laptops being stolen in the media where they concern government agencies or large corporations losing important data, as well as statistical reports as to the loss of both IP and hardware.

It seems that every other week an online IT news website reports of data going missing due to the theft of a laptop or notebook computer. In July 2007, a laptop belonging to an employee of the web security firm Verisign went missing (Leyden 2007). The data was not encrypted and it contained employee records including names, addresses and social security numbers. In April of the same year two laptops containing employee data for the Chicago public school system were stolen (Cullen 2007). These are just the high profile cases where the company has made them public. It does not include those that may have been kept quiet, and the smaller cases with a lower public profile.

There are also the statistics which attest to the fact that laptop theft is an on-going problem. Costs of laptop theft have two components: the hardware and the claimed value of the data on the stolen device. The most recent Australian report on computer crime, produced by AUSCERT, the 2006 Australian computer crime and Security survey, indicated that of those organisations surveyed, 58% reported theft of a laptop (AUSCERT 2006). This was up from 53% the previous year, but the year before that was also 58% (AUSCERT 2006). They also reported that 69% of organisations suffered financial loss as a result of laptop theft. No information was provided in terms of what proportion of the loss was due to the device itself, or any information contained on it. However, a value of 2.267 million dollars was attributed to laptop theft alone (AUSCERT 2006). Considering that there were only 126 respondents, when extrapolated to the whole of the Australian financial sector, this represents a significant loss. These figures also represent just organisations: they do not take into account theft of laptops from individuals.

Whilst a number of stolen devices are found, if their legitimate owners are not able to be found, then effectively, the device at least in the eyes of insurance companies has not been recovered. This creates two issues for those who have had devices stolen. Both issues are two-fold. Firstly, if the device is not returned to the original owner then they have effectively lost all of the information on it. The other aspect to this issue is that if the original owner cannot be identified, then the hard drive will be erased, and the information contained on it is again lost. Secondly, there is the financial cost of having to replace the laptop once stolen, or in the event that it is insured, the insurance company incurs the cost of having to replace the device. In addition there is the hidden cost of increased premiums in the case of the insured.

A large number of laptops are found by police in the course of their duties, but unless the original owner can be found then the device is not considered to have been recovered. The sheer volume of devices and prevalence of more serious computer related crimes mean that the time required to properly identify the original owner of a laptop just does not exist. As a result, the official recovery rate of laptops remains low.

These facts and issues demonstrate the need for a basic tool which can be used by anyone with minimal training to allow for identification of the recovered laptops. This will allow for several important outcomes. The first, and most important for the legitimate owner, is the recovery of their laptop. Both the device itself and information contained in it will be important to them. To the police, identification of the original owners will allow for an increase in the so-called clear up rate, an aspect which is of importance to them. Thirdly, the insurance companies, who are ultimately responsible for paying for the replacement cost of the laptop, are also an interested party.

This paper outlines the tool that has been produced as a proof of concept for phase I of the LAIRS project. Namely, a tool which examines hard drives of laptop computers, which have not been formatted, and which are running Windows XP.

## **THE PROOF OF CONCEPT – LAIRS PHASE I**

There are three main components to the LAIRS system: the underlying Live CD, the tool itself, and testing. All three are equally as important, but the Live CD needs to be established firstly, as the tool requires it to run. Testing will be very important, because although the information may not be used in a legal proceeding, there is the chance that it will, and therefore it needs to meet established forensic standards.

### **Assumptions**

There is one main assumption or decision that has been made for this project and it relates to the operating system. This project will work from the assumption that the majority of laptops are running Windows XP as their primary operating system. This is a reasonably safe assumption, as Windows XP was released in 2001, and although Windows Vista is due for release shortly, it is not yet available other than in beta form. This assumption will cause some issues later on, particularly with Windows Vista being made available shortly. However, as with Windows XP, it is likely that it will be some time before Windows Vista achieves widespread adoption. It is estimated by the Western Australian police force that it took windows XP approximately 2 years to achieve majority use (D. Taylor, personal communication 18<sup>th</sup> June 2007).

### **The Application - chntpw**

The application to be developed for use will need to be able to extract information from the Windows registry, as user and owner data for Windows XP systems is stored in the registry hives.

One of the advantages of using a linux distribution is that it is covered by the GPL, meaning that source code of all components is available. It also means that the software is available without any financial commitment. Although there are programs such as Registry Viewer available, this is proprietary software, which requires both licensing fees and the use of a dongle for its operation (AccessData 2006). Also, it will only run in a Windows environment, making it unsuitable for this project.

This project will use the chntpw utility, a program designed to change administrator passwords on Windows computers (Hagen 2004). Although designed for locating passwords in the SAM (security account manager) registry hive and resetting them, this program also has registry viewing and editing functionality. It is this aspect of the chntpw utility which is desirable, as much information relating to legitimate owner and any organisational registration information can be found in the registry. This software is also open source under the GPL and LGPL licenses, which provides several advantages. The first advantage is that the code can be examined to make sure that there are no unexpected features or pieces of code which may affect the host system in an unwanted manner. The second is that as we have the source code, it can be altered or changed to suit our purposes, saving a lot of development time. The third is that in this case there is no financial cost, as long as the original author is acknowledged.

### **Location of Information on the hard drive / Hives examined**

The LAIRS tool interrogates the following registry hives:

- SAM (HKEY\_LOCAL\_MACHINE\SAM)
- Security (HKEY\_LOCAL\_MACHINE\Security)



- Software (HKEY\_LOCAL\_MACHINE\Software)
- System (HKEY\_LOCAL\_MACHINE\System)
- NTUSER.dat (HKEY\_USERS\DEFAULT)

It should be noted that the tool is capable of interrogating other registry hives if required. The currently used hives are simply a matter of configuration.

When a Windows XP operating system is installed, the user is prompted to enter both their name, and that of the company, if applicable. This may or may not happen where the laptop is bought from a local supplier, but it is highly likely that this will be done where a corporate or standard operating environment (SOE) image is used. A search of an ECU laptop with an SOE image reveals many registry keys where the word “ECU” is found. In addition, there are numerous other applications which also store user information in the registry hives. Information is also stored about other applications, such as messenger clients and email programs (AccessData 2005). Table 1 contains a list of common user information and its location in the registry. Some of this information is stored in the NTUSER.DAT file. In addition to the information contained in the table, there are other locations and applications which store information about the user or registered organisation / owner of the laptop.

There are registry keys created by Office 2003, Office XP, Office 2000, Outlook Express and Outlook. In addition, virtually any software package that installs itself correctly will also create registry information which will contain the user’s details. For example, the Adobe family of products contains this information. This is very useful, as virtually every computer has a copy of the freely available Adobe Reader in order to read PDF files (Adobe 2007).

*Table 1: Information about the registered user, company, and other software variables and their respective location in the registry hive.*

Identifier	Key	Value
Office XP Company name	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall\< GUID>	RegCompany
Windows XP User name	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion	RegisteredOwner
Windows XP Company name	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion	RegisteredOrganization
Windows XP User name	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\<GUID>\Products\<ID>\InstallProperties	RegOwner
Windows XP Company name	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\<GUID>\Products\<ID>\InstallProperties	RegCompany

## Technical Implementation

A number of components are utilised as part of the LIARS project, the first of these is the underlying Linux operating system that forms the base of the live environment, from which LIARS runs (Figure 1). This live environment has been provided by the Simple Image Preview Live Environment (SIMPLE) project. This environment ensures that all data is accessed in manner in which the forensic integrity of the host system is not compromised. The LIARS tool itself exists on top of this live environment.

The LIARS tool is currently comprised of a database, a file system analysis script and a modified version of the chntpw utility. The database stores information relating to the registry hives of interest and the registry keys to be examined, this database is utilised by the file system analysis script and the modified chntpw utility.

The file system analysis script retrieves information relating to the registry hives, primarily expected file names and mime types. This information is then used to locate registry hives of interest; the location and hive types of the located hives are then stored in the database for later use.

The database is then read by the modified chntpw utility which cross-references the previously stored hive locations with a list of registry keys of interest. The modified chntpw utility then reads the value of each of these keys from the applicable registry hives and displays them to the user (Figure 2). At this stage the operation of the LIARS tool has completed and the system can be powered off at the user’s discretion.

## **Preliminary investigation into LIARS vs Vista**

The LIARS proof of concept tool has been tested on a Vista machine, and while it ran without problem, it did not return any registry values. This is likely due to the values currently being used by LIARS being specific to Windows XP. These values are likely to be differently named which would mean that they are not present at all on a Windows Vista PC. Available technical data on the Windows Vista registry indicates that the registry structure and location is the same for that of Windows XP (Microsoft 2007). Investigation of the Windows Vista registry for keys relating to the location of registered owner and registered company returned several values (Table1). This makes it likely that the LAIRS tool can easily be reconfigured to work on Windows Vista as well as Windows XP.

Table 2: Registry keys found in the Windows Vista registry relating to registered company and owner

<b>Identifier</b>	<b>Key</b>	<b>Value</b>
Windows Vista	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\5820D59FFDE05A2418084F7929EC5388\InstallationProperties	RegCompany RegOwner
Windows Vista	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\7B6E62F3B230B4042903A325C7F63EB6\InstallationProperties	RegCompany RegOwner
Windows Vista	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\B1F8F46A682D28A4689BE77F64CCD443\InstallationProperties	RegCompany RegOwner

## **LIMITATIONS**

There are a number of limitations that effect the current implementation of the LIARS project, it should be noted that the majority of these will be addressed as part of the future development of the project. In its current state the LIARS software is unable to extract data from hard disks that have been formatted or where the contained data is otherwise deleted. As a current limitation of the libraries used by LIARS it is not yet possible to recover information from systems that are running and operating system other than Microsoft Windows XP. Finally there have been instances in which the underlying Linux live environment provided by the SIMPLE project has failed to operate on some hardware.

## **FUTURE WORK**

Whilst development of this proof of concept tool has been successful, there is still more work to be done. This includes, dealing with formatted hard drives, other operating systems (Windows Vista) and the inclusion of more registry keys. Lastly, and most importantly, is to recode the tool so as to not use any of the chntpw source code.

### **Examining formatted drives**

The second phase will be to expand upon the basic module, and look at deleted data. In the event that a laptop has been formatted, no data will be available to an investigator at a topical level. At this point, it will be necessary to examine the drive using a forensic analysis tool in order to attempt to recover information. This functionality will be added to the basic application, but user intervention, and thus training, will be kept to a minimum. As with Phase I, this phase will also be built upon the forensically validated Live CD, with the application also undergoing thorough testing.

### **Windows Vista**

At the time of writing, numerous organisations have published statistics as to the low rate of uptake of Microsoft's latest desktop operating system, Windows Vista (Whipp 2007; Larsen 2007; Orion 2007). Whilst there may be doubts as to the accuracy of these surveys, they all indicate a low rate of usage, meaning that the majority of laptops are likely to be running Windows XP. A report on a survey conducted by the Sunbelt company reported that for all users, Windows XP accounted for approximately 83% of operating systems present, with Vista accounting for only 9.3% for home users and a very low 0.03% in business machines (Orion 2007). However low the uptake of Vista may be, at some point it will become an issue if the LIARS tool is not able to extract data from its registry. As part of ongoing development the project will be expanded to include

support for Windows Vista and other operating systems as is deemed appropriate by the developers. This will allow for LIARS to remain useful as adoption of this new operating system increases.

Increase the number of registry keys examined

In addition to the other changes listed, the database of included registry entries will be enhanced to include information that is made available by selected third party applications. In particular those programs that are used by almost everyone would be likely candidates. For example, there would not be many users that do not have Adobe Acrobat Reader installed on their computers. The portable document format (PDF) is in widespread use, and Acrobat Reader is an essential utility if you wish to view these files. A preliminary examination by the authors has found that this product has several registry entries which would be of use to the project. Others may include third party web browsers, such as Mozillas Firefox (Mozilla 2007).

Re-write the code

In the event that the tool is to gain a commercial profile, it is important that there be no intellectual property or other licensing issues in relation to the use of the chntpw code. Whilst this is an open source tool, it would still be preferable that there be no issues of ownership. This will entail some additional work, but the authors now have a greater understanding of how the registry is structured, and how to extract information from it.

## **CONCLUSION**

The first phase of LIARS is now complete, with a proof of concept tool able to interrogate the registry hives of aim of the LIARS project is to develop a tool that can be used by a police officer, or other investigator, with little knowledge of computers. The tool will be produced and subsequently tested to determine its forensic validity with an appropriate framework. The Live CD which will be used as the base for the examination tool is currently undergoing testing. It is hoped that when the application has been successfully developed, that it will be used by the WA Police force in the field, and that its use will result in a significant increase in return of laptops to the official owners.

Future phases will add additional functionality to the LIARS system, with the ultimate aim to the ability to examine deleted sectors of the hard drive. Future research will look at examining Windows installations for other sources of identifying information, in addition to those being used now. Searching of email for user details is a potential avenue which will be explored. With the slow, but increasing, uptake of Windows Vista, it will also be necessary to examine the tools functionality in relation to its ability to locate the same information.

## **REFERENCES**

- AccessData (2005). Registry quick find chart, URL [http://www.accessdata.com/media/en\\_us/print/papers/wp.Registry\\_Quick\\_Find\\_Chart.en\\_us.pdf](http://www.accessdata.com/media/en_us/print/papers/wp.Registry_Quick_Find_Chart.en_us.pdf) accessed 18 September 2007
- AccessData (2006). AccessData Registry Viewer, URL <http://www.accessdata.com/products/rv/> accessed 18 September 2007
- Adobe (2007). Adobe Reader, URL <http://www.adobe.com/products/acrobat/readmain.html> accessed 19 October 2007
- AUSCERT (2006). 2006 Australian computer crime and Security survey, URL <http://www.auscert.org.au/images/ACCSS2006.pdf> accessed 10 October 2007
- Cullen, D. (2007). Laptop thefts expose 40,000 Chicago teachers, URL [http://www.theregister.co.uk/2007/04/09/chicago-teachers\\_security\\_breach/](http://www.theregister.co.uk/2007/04/09/chicago-teachers_security_breach/) accessed 16th October 2007
- Hagen, P.N. (2004) The Offline NT Password & Registry Editor, URL <http://home.eunet.no/pnordahl/ntpasswd/> accessed 19 October 2007
- Helix (2006). The Helix Live CD page, URL <http://www.e-fense.com/helix/> accessed 20 October 2007
- Knoppix (2006). Knoppix, URL <http://www.knoppix.org/> accessed 15 October 2007
- Larsen, E. (2007). Vista uptake slow as companies shy away, URL <http://www.itweek.co.uk/personal-computer-world/news/2185624/vista-uptake-slow-research> accessed 20th October 2007
- Leyden, J. (2007). VeriSign worker exits after laptop security breach, URL [http://www.theregister.co.uk/2007/08/06/verisign\\_laptop\\_theft/](http://www.theregister.co.uk/2007/08/06/verisign_laptop_theft/) accessed 16th October 2007

- Microsoft (2007). Windows registry information for advanced users, URL <http://support.microsoft.com/kb/256986/> accessed 20th October 2007
- Mozilla (2007). Firefox web browser, URL <http://en.www.mozilla.com/en/firefox/> accessed 22nd October 2007
- Orion, E. (2007). Vista uptake is barely more than Windows 98 share: Less than 1 per cent in businesses, URL <http://www.theinquirer.net/gb/inquirer/news/2007/10/08/vista-uptake-barely-windows> accessed October 20th 2007
- Whipp, M. (2007). Vista uptake is slow - Net Applications, URL <http://www.pcpro.co.uk/news/103914/vista-uptake-is-slow-net-applications.html> accessed 20th October 2007

## **COPYRIGHT**

Peter Hannay, Andrew Woodward and Nic Cope ©2007. The author/s assign SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.

## Mood 300 IPTV decoder forensics

An Hilven  
School of Computer and Information Science  
Edith Cowan University  
ahilven@student.ecu.edu.au

### Abstract

*Since June 2005, viewers in Belgium can get access digital TV or IPTV available via ADSL through Belgacom, the largest telecommunications provider in the country. The decoders used to enjoy these services are the Mood 300 series from Tilgin (formerly i3 Micro Technology). As of the Mood 337, the decoders contain a hard disk to enable the viewer to record and pause TV programs. Although it is publicly known that the Mood's hard disk is used to save recorded and paused TV programs, it was still unknown if it contains any data that could be of interest during a forensic investigation. Interesting data ranges from which TV programs were watched, over discovery of unauthorized data storage, to criminal profiling and alibi verification. This paper will research the possibilities, especially with regards to which TV programs were watched and alternate data storage, as criminal profiling and alibi verification is not merely a task the forensic investigator can do alone.*

*Just like game consoles that use a hard disk, the Mood 337 can easily be disassembled and attached to a PC for forensic analysis. The reason why analysis of this system is necessary is simply because it contains a hard disk. Anyone with a screwdriver can remove, replace or modify it not only for experimenting purposes but also for illegitimate uses. Analysis shows that most of the 80 Gb of disk space on the disk is not even in use, and can easily have data being written on it without interfering with the system's primary function of providing IPTV services. It was also found that the Mood runs on a Linux base system with a 2.4 kernel, using XML file for the configuration of IPTV functions and services. Analysis reveals that even the (billable) 'pause' function is nothing more but a 'yes' or 'no' flag in an XML file. Other files that would be expected on a Linux system, such as /etc/fstab or /etc/passwd, were not found, while these might have been proven useful in this analysis. Further examination of the hard disk indicates the use of certificates for protection against piracy. However, it was proven to be a trivial task to simply copy recorded data to a PC and play it with a media player.*

*The most important discovery of this research is that correctness of time and date appears to be of lesser value for the creators and/or distributors of the Mood 337. Throughout the system, various different time stamps and time zones were used, and more importantly time and date were changed several times. Even though two NTP servers are configured for time synchronisation, neither one of them seems to be correct. In order for data recovered from this hard disk to be acceptable before a court of law, fixing the time and date should be one of the highest priority changes that are needed.*

### Keywords

Belgacom, IPTV, Mood 300, forensics

## INTRODUCTION

It has a hard disk...

It has a network connection...

It does not have any "warranty void if removed" stickers on the sides...

These three facts together make the Mood 337 decoder, used for IPTV in Belgian homes, motivate a curious forensic investigator-to-be to disassemble it and see if any forensically interesting information can be dissected from its hard drive.

What started out as mere curiosity about the contents of the little black box grew into actual forensic analysis of the Mood 337. In the end it was clear that analysis of its hard disk was not just fun, but can also prove helpful during real digital investigations. Probably the most obvious reason for performing forensic analysis is to discover if it was used as a hidden data storage device. The hard disk of a Mood 337 is not likely to be the most obvious place for law enforcement to search for evidence of illegitimate use and illegal data storage. Furthermore, the fact whether or not non-IPTV-related data is found can indicate that the suspect is technically

savvy. Another reason to forensically analyse the decoder is that it could be used in criminal profiling. Because traces of watched TV programs can be recovered, comparisons can be made between them. A decoder containing mostly traces of horror movies probably has an owner with a completely different personality than one containing nothing but Disney cartoons. The last reason for Mood 337 forensics is that the traces of watched TV programs can help identify when the owner was watching TV, and might help in verifying alibis. A suspect could claim he watched soccer all evening yesterday, and may be able to answer questions such as “which team won” and “who made the goals”, but he may as well have heard that on the radio later on. The Mood 337’s hard disk may well contain traces of last night’s soccer match. However, for this data to actually stand before a court of law an effort should be made by the designers and/or distributors of the device to ensure correct timestamps are used throughout the system.

Forensic analysis on alternate data storage devices is nothing new. It was already done before by for example, Schroader and Cohen in their recently published book “Alternate data storage forensics”, and by Burke and Craiger (2006) in their paper about forensics on Microsoft’s Xbox gaming console. In their paper, Burke and Craiger explain that the Xbox can quite easily be modified to support multiple operating systems, and thereby making it possible to store non-game-related data on the device. The same is true for data storage on the Mood 337.

Although it was not tested during this research to run additional operating systems, the Mood’s hard disk holds plenty of disk space that can be experimented with, even if only for data storage. It has been tested, however, by a user (Dreamweaver, 2006) of an unofficial Belgacom ADSL forum, that the IPTV signal can be captured by a regular PC and played, posted that he was able to use GeeXbox Linux to get the IPTV signal on his PC, being able to swap channels and watch TV in high quality video using just Mplayer. Additionally, Dreamweaver created an image he created of a working system and made it publicly available (Dreamweaver, 2006). The fact that the IPTV signal can be replayed on a regular PC proves that no specific hardware or operating system is required, and thus it should be possible as well to run another or additional operating systems on the Mood 337.

Burke and Craiger (2006) were also able to set up a network connection to the Xbox system and perform their analysis via a simple SSH connection. For the analysis of the Mood 337, however, a different approach was used, namely physically removing the hard disk from the casing and attaching it to a PC. As it is not known beforehand whether or not a network connection would change data on the Mood’s hard disk, this risk was not taken. The reason for this is that the system used in this analysis was not a test device, but is in fact still used by the owner. Siglio, a user on the Userbase.be forums, was in fact able to set up a network connection to the Mood and could set up a CGI script that allowed him to make changes to the system directly from the Mood’s internal web server he was connected to (Siglio, 2007).

In an article on Linuxdevices.com (2002), i3 GM for Streaming Products Chris Chalkitis revealed that the Mood’s kernel is merely a 2.4.x kernel retrieved from kernel.org and then modified for IPTV use. Other publicly available Linux tools were used as well, and because most of these are GPL licensed, i3 decided to make their changes to the source code publicly available as well on their FTP server. These sources were not immediately made public, yet after requests from an end user (Siglio, 2006) they were made available in June 2006 on <ftp://ftp.opensource.tilgin.com/MOOD/>.

It may be clear that quite some research was already done on the Mood decoder, but most of this research was merely either out of curiosity or experimentation with the IPTV signal. It appears that no (publicly available) research has yet been performed from a forensic angle. This paper will analyse the contents of the Mood’s internal hard disk, and will discuss the findings that came forth from it..

## **HARDWARE INFORMATION**

All research is performed on is the Mood 337 V2 BE from Tilgin AB, manufactured in August 2006. Detailed information on all components used in this device can be found in its product sheet (Tilgin, 2006). This paper will focus on the Mood 337’s hard disk, which in this case is of the type WD Caviar (WD800BB-55JKC0) from Western Digital with manufacturing date 8 May 2006. This is an IDE hard disk running at 7200 rpm. Although analysis of other sources of data such as the NOR and NAND flash chips may reveal interesting information, it is outside of the scope of this paper and saved for possible future work.

## **FORENSIC PROCESS**

### **Preparation**

The hard disk was removed from the Mood 337 hardware and attached as a regular IDE disk to a PC. The ‘cable select’ jumper setting was left in place during this process. The power connector was left attached to the original

Mood 337 casing instead of hooking it up to the PCs power supply, because it was uncertain how the Mood 337 fed power to the hard disk.

### Acquisition

The acquisition of the hard disk was done from a Debian Etch system, using *dd* to create a bitwise copy of each partition. A total of 3 partitions were found, namely hda1, hda2 and hda3.

### Analysis

To retrieve some initial general information about the partitions, each of the images was loaded into *FTK* (under Windows XP Professional) and *Autopsy* (under Helix). Both came up with the same information, confirming the outcome is likely to be correct.

NAME	SIZE	FORMATTING
<b>hda1</b>	1 Gb	raw, unformatted
<b>hda2</b>	1 Gb	ext3
<b>hda3</b>	72 Gb	ext2

Fig. 1 – Size and formatting of the Mood 337 hard disk

On each partition, a large amount of free space was found. This was 100% for hda1, 80% for hda2, and 91% for hda3:

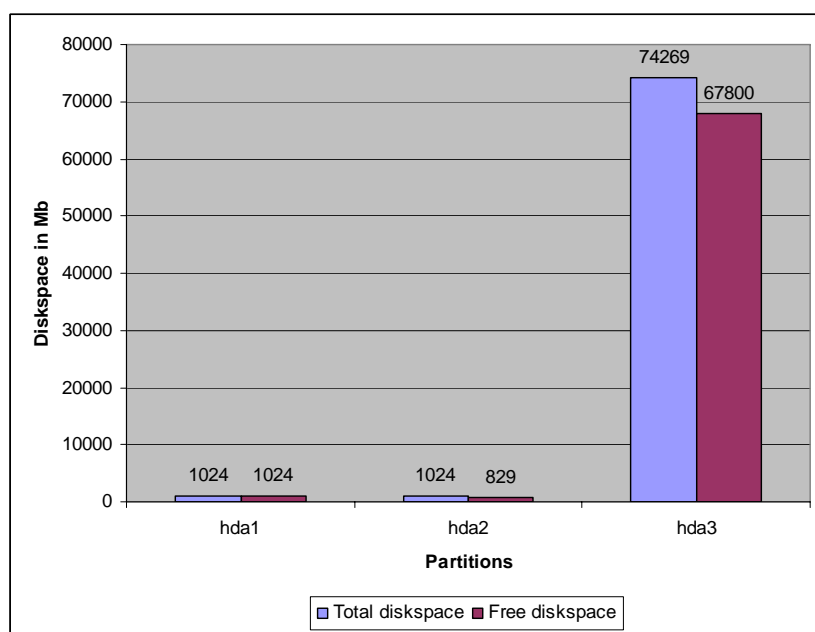


Fig. 2 – Amount of free disk space compared to the total partition size for a Mood 337 decoder

More detailed analysis was done with a three applications. For general analysis *FTK* and *Autopsy* were used. However, because these tools are not sufficient for more advanced file carving, *Scalpel* was used for this purpose instead.

For each partition, first the directory structure is analysed and interesting files are listed and discussed. Next, manual analysis of the free space areas is done. And to conclude, file carving is done on the free space, and the outcome is discussed.

## PARTITION 1 (HDA1)

### Directory structure

*FTK* and *Autopsy* found that this is a raw, unformatted, partition containing nothing more than free space. Manually analysing this partition in hexadecimal format reveals that it is indeed completely clean, and does not appear to have ever been written to at all.

### File carving

To ensure nothing was overlooked, the image was sent through *Scalpel*, with a configuration file edited to search for all known file types. The result confirmed the initial findings, as no files were found.

## PARTITION 2 (HDA2)

This section describes the findings of examination of each of the directories (see below) and their contents with both *FTK* and *Autopsy*. For each of the directories listed, a brief explanation is also given as to whether or not and in which cases its contents can be of evidentiary value during a forensic investigation.

### Directory structure

This partition is formatted with the ext3 file system, and its directory structure looks quite similar to a normal Linux system, especially when looking at the structure of the */root* directory. Besides the directories normally expected on a Linux system, two directories appear to be the odd ones out, being */persist* and */localexec/conf*.

```
VOLUME ROOT
|- /localexec
    |- /conf
    |- /root
        |- /dev
        |- /etc
        |- /lib
        |- /media
        |- /sbin
        |- /usr
        |- /var
        |- /www
|- /lost+found
|- /persist
```

Fig. 3 – High-level directory structure of the *hda2* partition on a *Mood 337* system

#### */localexec/conf*

This directory holds an XML file that seems to be the configuration of where the firmware (*netimage-myrioi-3.7.2-bel-137.tar.gz*) was copied or downloaded from, and that it is/was encrypted with AES.

Note that the statement regarding copy or download locations is merely speculation, based on the fact that the path in the “from” line does not exist on the hard disk, and such it might be an (incomplete) web or network location.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<addonlist count="1">
  <version>
    <type>Mood300</type>
  </version>
  <addon-key type="AES">59EC1A5B172E119D558CA3B93FD62E4B</addon-key>
  <file>
    <revision>myrioi-3.7.2-bel-137</revision>
    <from>myrioi-3.7.2-bel-137/netimage/netimage-myrioi-3.7.2-bel-
      137.tar.gz</from>
```



```
<to>netimage-myrioi-3.7.2-bel-137</to>
<size>15899140</size>
<crc>30873</crc>
<md5>83bf45d9d2b0d5a7d149dab7245e5346</md5>
<moddate>2007-06-23 00:56:26</moddate>
</file>
</addonlist>
```

Fig. 4 – Myrioi firmware information for the Mood 337

The other three files in this directory appear to be related to the XML file. The first file, *netimage-myrioi-3.7.2-bel-137.tar.gz.md5* contains the same MD5 sum as found in the XML file. The second file, *netimage-myrioi-3.7.2-bel-137.tar.gz.dirlist*, contains a listing of directories likely to be included in the firmware package or the locations it should be extracted to. The last file, *netimage-myrioi-3.7.2-bel-137.tar.gz.nodelist*, contains a list of symbolic links likely to be created during installation of the firmware.

The “moddate” line appears to contain the date and time when the firmware was installed or updated. The last firmware update was indeed done around 23 June 2007, and the same firmware version is visible in the decoder’s menu (Belgacom, n.d.). However, the fact that other timestamps and time zones used in the system are inconsistent may raise doubts to the reliability of this information. An example of this inconsistency is that the MAC times of these files are all 1 January 2000, instead of the date of the firmware update.

/localexec/root/dev

This directory does not contain any devices. It merely holds two symbolic links, one for the internal web browser (*myriohandler.fifo*) and one for a mouse driver (*gpmdata.fifo*). The symbolic links reference to files in this same directory, yet these files do not exist.

/localexec/root/etc

In contrast with a regular Linux system, this directory does not contain the files that are usually of interest during investigations, such as *fstab*, *mtab*, *passwd*, etcetera. However, it does contain other interesting data such as files listing the hardware and software version.

```
VER_HARDWARE=i3-mood-HD
BASEMODEL=I3MICRO-MOOD
DISKTYPE=COMPACTFLASH
CD=NOCD
NETCARD=NATSEMI
```

Fig. 5 –Mood 337 hardware information as discovered in configuration files

```
VER_SOFTWARE=3.7.2-137
CLIENT_VARIANT=bel
BUILD_STAMP="build@build-vm on Fri Jun 22 15:51:45 PDT 2007"
```

Fig. 6 –Mood 337 software information as discovered in configuration files

This confirms, as stated earlier, that time and dates are used inconsistently on this system. For example the previously discussed XML file used CET, while the software information file uses PDT. Note that in June there was a time difference of 7 hours between CET and PDT, and thus the timestamps do not even match when converting the time zones.

Further interesting information is found in the *rc2.d* subdirectory, as this will show which processes are started at system boot time. Although expected processes such as *syslogd* and *klogd* are missing, DHCP configuration and swap file creation do initiate at system boot. Various other processes are started; all of which seem to be not (or less) used on regular Linux systems. These processes are: *zapper*, *dvr*, *ipmd*, *moodplayer*, *movie*, *recorders*, *hwversion*, *mpersist*, *savemoodconf*, *loadkeys*, *myriohandle*, *setkeycodes*, *myriodispd*, *startapp*, *security\_engine*.

In the *X11* subdirectory, an *X Windows* configuration file is found. Even though it was found already that time and date seem to be an issue on this system, it does appear that *X Windows* is synchronized with a time server (although it does not seem to be configured in an officially supported manner).

```
# TODO - FIXME - temp hacks - BEGIN
# force sync with server time
DOMAIN=nat.myrio.net
. /etc/dhpcp/dhpcpd-eth0.info
if [ "$DOMAIN" = "nat.myrio.net" ]
then
    rdate time.nat.myrio.net
```

```
fi
# TODO - FIXME - temp hacks - END
```

Fig. 7 – Temporary hacks for the Mood's time synchronisation in X11

One last piece of interesting information found in this directory is a configuration file, *movie.conf*, which seems to be used to configure where media streams should be sent to. However, this location, */media/hdd/PVR*, does not exist on the hard disk.

*/localexec/root/lib*

This directory only contains two symbolic links, one to *../usr/lib/dspimage.out* and one to *../usr/lib/dspimage.ver*. Analysis of *dspimage.ver* shows that it contains nothing more than two numbers (4 and 34), which is likely a version number of some kind. Examination of *dspimage.out* indicates that this could be a library used to decode media streams, as it contains many references to media such as mpeg3, mp3, ac3, aacdec, mpeg2, as well as various uses of the words 'dec' and 'decode'.

*/localexec/root/media*

A subdirectory named *persist* is the only data found in this directory. It holds files nearly identical to the ones in */persist*, with the only difference that the MAC times are off by 2 minutes. The contents of these files will be discussed in the */persist* section below.

*/localexec/root/sbin*

Just one file, named *zapper*, resides in this directory. *Zapper* was sent through strings in order to get an idea what this file is used for. The results indicate that it has something to do with IGMP, as it contains strings related to this subject. Some further investigation (Juniper Networks, 2007) to the relationship between IGMP and IPTV teaches that IGMP is used as the method for changing TV channels in IPTV environments.

*/localexec/root/usr*

This directory mainly contains two types of files: Java APIs and kernel modules.

The JAVA APIs reveal a little more information with regards to which software runs on the decoder. The most eye-catching ones are: *Bouncycastle* (used for crypto), *Apache Crimson* (used for XML parsing), *Apache Ant* (a build tool), *Apache Jakarta ORO* (used for regex processing), *Apache Log4j* (used for logging), *Myrio Escape* (a web browser)

Besides minimal kernel modules needed for the system to actually run, a few other interesting modules were found in this directory. For example, there are modules for IGMP processing, MTS file processing (MTS files will be discussed later). One kernel module, *cas\_verimatrix.so*, is the first indication discovered that this system was set up with security in mind, as this module seems to be responsible for the handling of SSL, TLS, RSA keys, etcetera.

*/localexec/root/var*

Yet again, this is one more directory which's contents are nothing like what would be found on a regular Linux system. It was expected that logging would be found here, but instead two symbolic links were found, one to the *../media/persist* directory and one to the *../media/persist/myrio/persist* directory.

*/localexec/root/www*

This directory contains various GIF files that do not appear to be used anywhere. Furthermore there are symbolic links named *zapper.cgi* and *playtv.cgi*, both linked to a non-existing file *cgi.cgi*. A live search with *FTK* for any references to *.cgi* files did not result in any possible related files.

*/lost+found*

This directory is empty.

*/persist*

The */persist* directory holds all files and information used to configure the decoder, such as the locale, cache size, buffer size, video standard, logging settings and enabling or disabling the 'pause' option. It is remarkable, however, that although Apache Log4j is configured in *mclient\_log4j.properties*, the file all logging should be written to (*var/data/log/mclient.log*) does not exist.

```
#### Second appender writes to a file
log4j.appender.R=org.apache.log4j.RollingFileAppender
```

```
log4j.appender.R.File=/var/data/log/mclient.log
# Control the maximum log file size
log4j.appender.R.MaxFileSize=2000KB
# Archive log files (one backup file here)
log4j.appender.R.MaxBackupIndex=2
```

*Fig. 8 – Snippet of Apache Log4j configuration on the Mood 337*

In this same directory a root certificate was found as well, revealing that Verimatrix is used as the Certificate Authority. More research teaches that this is a company that incorporates security features in Pay TV systems with the goal of enhancing revenue and verifying the legitimacy of streaming content to protect against piracy. Verimatrix does this through a PKI infrastructure and the use of X.509 digital certificates.

Further the directory contains a list of all hardware related error message the end-user might see, such as hard disk maintenance start and completion, and warnings that the temperature is too high.

On a side note; the option to pause live programs (mentioned above) is only enabled if paid for on a monthly basis. Testing and experimenting with editing configuration files may be interesting for future work.

#### Swap file

Analysis of the swap files in the root of the decoder revealed a few things that could not be deduced from configuration files. Something that may be very important during an investigation is that a trace of the device's IP address can be found. Because the swap file is overwritten every time the system restarts, it is likely that only the last IP address can be found here (if at all).

```
Lease of 10.131.27.19 obtained, lease time 604800
```

*Fig. 9 – The Mood's last IP address and lease time was found in the swap file*

Another interesting piece of information is which NTP server is used and how often it is polled. Strange enough, this is a different server than the one that was found in a configuration file earlier.

```
NTPSERVER="ntp.nat.myrio.net"
NTP_INTERVAL="172800"
```

*Fig. 10 – Another NTP server configuration extracted from the swap file*

The swap file also revealed what the management IP addresses are that are allowed to make a network connection to the decoder. This might prove very valuable during an investigation (if the investigator's PC can be configured with one of these addresses), and interesting for future research.

```
CONTROL_IP="10.48.18.122,195.238.8.137,195.238.8.78,81.245.3.187"
```

*Fig. 11 – IP addresses the Mood 337 can be managed from*

Besides the above findings, no information was discovered in the swap file that could not have been found by analysing configuration files.

#### Free Space

First, the boot record was analysed. Even though a boot record exists, it is empty and looks like it has never been written too.

Next, *FTK* was used to extract all free disk space of this partition in the form of 33 files with an average size of 25 Mb. Each of these files was analysed manually through *FTK*'s capability of viewing files in hexadecimal form, in order to get an idea whether or not any forensically interesting information could be found.

At the beginning of the free space, mostly gibberish was found. However, once in a while text appears indicating start and completion of hard disk maintenance and temperature warnings.

At around three quarters of the free space, contents of configuration and Java class files were visible, but did not contain any information that could not already be found in the configuration files and class files themselves.

At the very end, contents of the online TV guide were found from mid-December 2004 and January 1971. This indicates another issue with the usage of time and date on this system. For example, a description (in Dutch) of

the final episode of the TV series 'Heroes' was found. This episode was broadcasted in Belgium on 4 June 2007. However, the time stamp indicates that it was aired 17 December 2004.

```
1103287700|0|Heroes|5|2|110|FI|C.Serie|0|us|1|De
ontknoping nadert... De helden zijn in New York voor de
belangrijkste dag van hun leven. Zij zijn uitverkoren om de
wereld te redden van een naderende ondergang. Hun missie is
om te voorkomen dat Mendez' afbeelding van de ontploffende
man realiteit wordt. Kunnen de helden de vernietiging van New
York en de wereld voorkomen? Kan de ontploffende man
tegenhouden worden?|6|Hayden Panettiere|Masi Oka|Ali
Larter|||false|false|false|false|false|false|false|false|
```

Fig. 12 – Heroes goes back in time

Another description was recognized, this time of 'Big Cat Diary'. This was aired on 14 October 2007, while the time stamp indicates this was 18 December 1970. It may be worth to note that the firmware upgrade which was discussed earlier falls between the two 'real' dates. It appears very likely that during this upgrade the time and date was changed or reset.

### File Carving

To see if any further interesting data could be found, all free space files extracted by *FTK* were fed to *Scalpel* with a configuration file edited to search for all known file types (identical to the configuration used for analysis of hda1)

This resulted in various GIF and JPG files. The GIF files contained all kinds of logos of for example i3 and Espial, but also MGM and Disney. Looking through the user interface on TV, these logos do not seem to be used anywhere. The JPG files are images for movies as they would appear in the online movie catalogue for rental movies. However, this only contains relatively old movies, none of which are currently available in the online catalogue. It is assumed that either these movies were available a long time ago, or they were left behind during the initial installation of the system.

### PARTITION 3 (HDA3)

This partition uses the ext2 file system. It does not contain much data, other than various files that appear to belong to three groups if grouped by MAC time (23 June 2007, 5 October 2007 and 19 October 2007). Each group consists of four files with identical filenames but different extensions. These extensions are .idx, .info, .time and .mts.

The .idx files do not appear to contain any information, except for repeated sequences of same hexadecimal values over and over again.

```
00 00 00 00 ec 84 00 00 00 00 00 00 04 f6 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 bc 00 bc 00 00 00 00 00 ec 84 00 00
00 00 00 00 04 f6 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 bc 00 bc 00
```

Fig. 13 – Example of repeating sequence in an .idx file found on the Mood 337's hda3 partition

The .info files contain XML data. Although information such as title and description are not available, these files indicate when a recording was started and stopped.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tv><channel id="-1">
<display-name></display-name>
<url></url>
<icon src="" />
```

```
</channel>
<programme start="20071005201000 -0000" channel="">
<title>title not set</title>
<desc></desc>
<desc lang="">STOPPED</desc>
<length units="seconds">3</length>
<task>RECORDING</task>
</programme>
</tv>
```

Fig. 14 – Example of start and stop times in an .info file found on the Mood 337's hda3 partition

No interesting information was found in the .time files. It is however notable that these files always appear to be 1.92 Mb in size.

The .mts files are the largest files. The ones found here were 4 Mb, 1.8 Gb and 3.7 Gb in size respectively. Due to these file sizes; it is thought that these files might contain the actual recorded TV stream. To test this, the largest file was exported from *FTK* and played with *Windows Mediaplayer*. *Windows Mediaplayer* did appear to recognize the file as a media file, but threw an error indicating the required codec was not available. More research regarding the .mts file extension revealed that this type of files can be played with the *VLC media player*. This application was downloaded, and playback of the .mts file showed the movie 'King Arthur', recorded from TV on 23 June 2007, in excellent mpeg4 quality. Also note that the other files, with extensions .idx, .info and .time were not required during playback of the movie.

Due to the fact that up to 74 Gb of recorded data can be stored on this partition, it may be possible to profile the owner of the decoder, because it is possible to know which TV programs he or she likes.

### Free space

Again first, the boot record was analysed. As with hda2 a boot record exists, but it is empty and looks like it has never been written too.

Similar to hda2, *FTK* was used to extract all free disk space of this partition in the form of 2712 files with an average size of 25 Mb. Although a tedious work, each of these files was analysed manually in the same manner the free space files on hda2 were analysed (i.e. through hexadecimal viewing).

The only data that was found was in free space files 1 through 69. After free space file 70, only zeroes exist. However, the files that do contain data do not hold any interesting information such as configuration files or logging. To analyse this further, file carving is needed.

### File Carving

Similarly to processing hda1 and hda2, all free space files of hda3 extracted by *FTK* were fed to *Scalpel* with a configuration file edited to search for all known file types (again identical to the configuration used for analysis of hda1 and hda2).

*Scalpel* carved hundreds of small MPEG files. When playing these files consecutively in the order they were carved, fragments of TV programs that had been watched could be seen. In some cases, several of these MPEG files together formed a complete TV program, while for other TV programs only 1 or 2 minutes were found. It appears that the more recently the TV program was watched, the more fragments of it can still be found in free space. The oldest recognizable fragments were from a program broadcasted over 2 months ago. With a little help from the TV station (which can easily be identified by its logo in one of the corners), the exact date and time a specific fragment was broadcasted can be determined. With this information, it would be possible to prove that someone was watching a specific program on a certain date and time (or at least that the TV was on during that timeframe). This might help in verifying alibis and the like.

## WRITING TO DISK

Because the decoder used for this research is in fact still in use, no major testing was done with writing data to the disk, or editing configuration files. For future work, a decoder should be used that is no longer active, so that this can be experimented with as well. It would be interesting to see what happens when for example the 'pause' setting is changed, or what the logging would contain if it were properly enabled. And especially whether or not the decoder would continue to work as normal, as well as seeing if the configuration files are overwritten at a next reboot.

What was tested, however, is that it possible to copy files to hda3 without having any impact on the working of the device. It was also possible to delete the files, again without impacting the device's working. Therefore the hard disk of a Mood 337 makes an excellent choice for storing data that is intended to be kept secret.

## CONCLUSION

Due to inconsistent use of time, date and time zones, using evidence retrieved from a Mood 337 set top box used for Belgacom customers would be likely to be rejected before a court of law. However, fragments of TV programs and help from TV stations may result in information of higher evidentiary value, as they could help in believing or rejecting alibis. Furthermore, with an analysis of the recorded TV programs it might even be possible to profile the owner of the decoder.

Forensic analysis of these decoders may also prove valuable in cases of copyright infringement, because it was found that it is really easy to just copy recorded programs and watch them on PC. A next step of editing the file to cut out commercial breaks, and burning it to a DVD is not a large one to take.

Because a simple, unprotected, hard drive is used for saving configurations and recorded data, two possible issues arise. First, one may be able to edit configurations and making paid services free (which was not tested in this paper, and is saved for future work). And second, any other data can be written to this hard disk. So if someone really wants to hide data, but is afraid that it will be revealed when his or her PC is forensically examined, why not write it to a hard disk that has less chance to be spotted by law enforcement?

## REFERENCES

- Belgacom (n.d.). How can I verify the firmware version of my decoder. Retrieved on October 8, 2007, from [http://selfcare.belgacom.net/index.html?l=private:search&a=default&r=613671&p\\_faqid=9028&p\\_create\\_d=1140432120&p\\_sid=ZS6tGoPi&p\\_lva=&p\\_sp=cF9zcmNoPTEmcF9zb3J0X2J5PSZwX2dyaWRzb3J0PSZwX3Jvd19jbnQ9MSZwX3Byb2RzPTU2LDYwJnBfY2F0cz0mcF9wdj0yLjYwJnBfY3Y9JnBfcGFnZT0xJnBfc2VhcmNoX3RleHQ9ZmlybXdhcmU\\*&p\\_li=&p\\_topview=1](http://selfcare.belgacom.net/index.html?l=private:search&a=default&r=613671&p_faqid=9028&p_create_d=1140432120&p_sid=ZS6tGoPi&p_lva=&p_sp=cF9zcmNoPTEmcF9zb3J0X2J5PSZwX2dyaWRzb3J0PSZwX3Jvd19jbnQ9MSZwX3Byb2RzPTU2LDYwJnBfY2F0cz0mcF9wdj0yLjYwJnBfY3Y9JnBfcGFnZT0xJnBfc2VhcmNoX3RleHQ9ZmlybXdhcmU*&p_li=&p_topview=1) (Dutch)
- Burke, P. K., Craiger, Ph. (2006). Xbox Forensics, *Journal of Digital Forensic Practice*, 1:4, 275 – 282, Retrieved on November 23, 2007, from <http://dx.doi.org/10.1080/15567280701417991>
- Dreamweaver (2006). BGTV on GeeXboX. Retrieved on November 26, 2007, from <http://forum.adsl-bc.org/viewtopic.php?t=35357> (French)
- Juniper Networks (October 2007). Introduction to IGMP for IPTV networks. Retrieved on October 22, 2007, from [http://www.juniper.net/solutions/literature/white\\_papers/200188.pdf](http://www.juniper.net/solutions/literature/white_papers/200188.pdf)
- Lehrbaum, R. (July 2002). Device profile: i3 micro Mood Box. Retrieved on November 26, 2007, from <http://linuxdevices.com/articles/AT9483972214.html>
- Schroader, A., Cohen, T. (November 2007). *Alternate data storage forensics*. Syngress. ISBN 1-59749-163-2.
- Siglio (2006). Does Belgacom/Tilgin violate GPL? Retrieved on November 26, 2007, from <http://forum.adsl-bc.org/viewtopic.php?t=30063> (Dutch/French)
- Siglio (2007). Belgacom Mood hacking. Retrieved on November 26, 2007, from <http://www.userbase.be/forum/viewtopic.php?t=13104> (Dutch)
- Tilgin (2006). Mood 300 Series. Retrieved on September 18, 2007, from [http://www.tilgin.com/Documents/Product%20sheets/Mood%20300\\_PAL\\_ProductSheet.pdf](http://www.tilgin.com/Documents/Product%20sheets/Mood%20300_PAL_ProductSheet.pdf)

## COPYRIGHT

An Hilven ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **A Methodology for the Forensic Acquisition of the TomTom One Satellite Navigation System – A Research in Progress**

Peter Hannay  
Edith Cowan University  
phannay@student.ecu.edu.au

### **Abstract**

*The use of Satellite Navigation Systems (SNS) has become increasingly common in recent years. The wide scale adoption of this technology has the potential to provide a valuable resource in forensic investigations. The potential of this resource is based on the ability to retrieve historical location data from the device in question while maintaining forensic integrity. This paper presents a methodology to acquire forensic images of the TomTom One/ satellite navigation unit. This methodology aims to be comprehensive and straightforward, while maintaining forensic integrity of the original evidence. However, in consideration of the aforementioned methodology it should be noted that the defined method may not extract all potential evidence and the viability of collected evidence is dependent on future research into the analysis of said evidence. In order to address this consideration, research into this area is currently ongoing.*

### **Keywords**

Global Positioning System, GPS, NAVSTAR, forensic methodology, digital forensics, GPS forensics, satellite navigation system, satnav, satnav forensics.

### **INTRODUCTION**

The NAVSTAR Global Positioning System (GPS) was declared fully operational in 1995. At this time however the civilian GPS signal was artificially degraded in order to limit the threat that it posed if used by those who opposed the United States. The non-degraded signal known as ‘M-CODE’ was reserved for military use only. On May 2<sup>nd</sup> 2000 this artificial degradation, known as ‘selective availability’ was disabled and the fully functional signal became available to civilians worldwide (Braunschvig, Garwin, & Marwell, 2003). With the full GPS signal available to the civilian population commercial applications of the GPS network began to increase rapidly. This increase would eventually lead to the wide scale availability of Satellite Navigation Systems (SNS) (Theiss, Yen, & Ku, 2005).

Automotive satellite navigation systems such as the TomTom One<sup>1</sup> (TomTom, 2007), aim to provide navigational assistance to its’ users. Often the user will provide a destination point then based on this the device will provide a map and verbal turn-by-turn directions to the specified destination. Such devices are becoming more common and are decreasing in price. It should also be noted that many new cars now come with SNS as standard.

The ability to acquire forensic images from satellite navigation devices is becoming increasingly relevant with the aforementioned increase in availability of these devices. Satellite Navigation units have the potential to provide valuable historical locational data to investigators.

In the application of forensic procedure to satellite navigation systems and indeed any digital evidence as a whole there are a number of issues that must be understood. The primary issue faced by digital forensics is the intangibility of the evidence being collected. As the evidence only exists in digital form the method of acquisition heavily depends on the nature of the storage media on which the target information is located. An example is that data stored in volatile memory can often be erased if power to the device is lost (Noblett, Pollitt, & Presley, 2000). In addition to this the contents of volatile often changes constantly as data is re-arranged. In such cases special methods may needed to forensically preserve evidence that is located in volatile memory.

Digital forensics procedure is focused on preserving the integrity of the original evidence and allowing this integrity to be verified at a later stage (HB171, 2003, pp. 17-18). This verification is normally performed by the use of hashing algorithms and careful documentation.

In order for the evidence to be useful a copy must be acquired, this copy can then be used as part of an investigation without the possibility of compromising the original in the process. In order for this copy to be useful it must be what is known as a 1:1 or bit stream copy of the original (ACPO, pp. 20-21). A bit stream copy is a complete duplicate of the original data, instead of copying the files or other logical structures of the original

device the raw data that comprises these structures is read piece by piece and copied to a specified location or device. This method allows for an analysis to be performed on data that has been deleted or otherwise exists in unallocated space.

## **ACQUISITION SOURCE**

This paper focuses on the forensic acquisition of the digital evidence located on the SD card required for the operation of the device. The aforementioned SD card must be inserted into the device at all times in order for the device to function as its core operating system resides on the card. Initial research suggests the data on the SD card is comprised of at least the following:

- x86 boot sector
- Mapping data
- Operating system files
- Configuration files
- Swap space

The SD card has been chosen as the source of information to be acquired for a number of reasons. Firstly it is easily accessible in a non-invasive manner. It is also possible to acquire the SD card with a minimum of equipment and experience. In addition to this it is possible to acquire an image of the SD card in a covert fashion, in many cases it is not possible to determine that the device has been tampered with.

A number of alternate sources for potential evidence exist, however access to these would likely require some access to the internals of the device. For example internal flash chips and the GPS receiver chip itself could serve as a potential source of information. The acquisition of these components will not be covered in this document, as further research is required into the viability of these. However it is recommended that the satellite navigation unit itself is stored in an EM shielded area and connected to an appropriate power source, the device however should not be turned on. In the event that further research leads to the discovery of new evidence sources this may assist in ensuring that volatile memory is not erased and that forensic information is not overwritten as a result of GPS signals being received.

## **METHODOLOGY**

As the media to be acquired is a standard SD card the procedure for acquiring a forensic image of this media involves attaching the device to a system in read only mode and acquiring a bit stream copy of the SD card. As with any forensic procedure the media should be hashed before and after acquisition, the resulting copy of the data should also be hashed in order to verify its integrity. The methodology and technical explanation of hash computation is however out of scope of this document, as such these procedures will not form part of the outlined methodology.

It should be noted that powering the satellite navigation unit on whilst the SD card is inserted will result in data being written to the SD card and the hash changing. In this case the position of the write protect tab on the SD card is irrelevant as the TomTom One<sup>1</sup> does not discriminate if writing should be permitted based on the tab's position. Instead the SD card is treated as writable regardless of the tab's position.

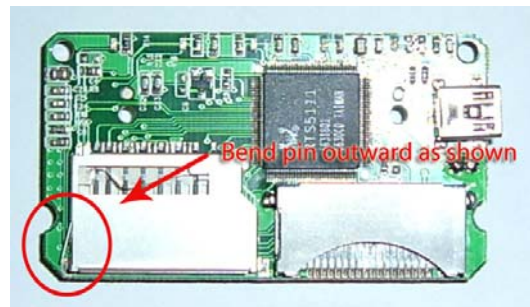
### **Equipment**

In order to perform the acquisition of the SD card it is necessary to have a number of items.

Write blocking SD card reader

Initial examination of commercially available SD card readers has shown that it is possible to modify these devices so that they will not perform write operations. This modification can be performed as shown in the diagram below.





*Figure 16. SD Card Reader with Read Only Modification*

A movable tab on the side of SD cards is commonly used to set the media to operate in 'read only' mode. This tab is similar to that on 3¼ inch floppy drives, in that the accessing device detects the tab's position rather than the read only logic existing on the media itself. As such the aforementioned modification works by manually bending the pin that detects the position of this tab so that the device will always detect the tab as being in the 'read only' position. The result of this is that all media will be treated as read only, regardless of the position of the tab.

#### Forensic Workstation

The forensic workstation is typically a standard PC with USB capabilities and a storage device with adequate free space to store the acquired image. In this case it is assumed that the workstation in question is running a Linux operating system with access to a terminal or other standard Linux command line interface (CLI).

#### 'dd' Software

The dd software is capable of performing low-level data operations such as performing a bit stream copy of the data to be acquired.

#### Process

1. Attach write blocked USB SD card reader
2. Insert a non-critical SD card for testing purposes
3. Perform a hash of the SD card
4. Ensure the file system (if any) present on the SD card has not been mounted
5. Attempt to write to the SD card
6. Perform an additional hash of the SD card
7. Ensure that the hash matches the original
8. Remove the SD card
9. Insert the SD card to be acquired
10. Perform a hash of the SD card
11. Ensure the file system (if any) present on the SD card has not been mounted
12. Acquire a copy of the SD card using dd
13. Perform a hash of the SD card
14. Perform a hash of the acquired file
15. Ensure that the hashes match the original
16. Remove the SD card from the reader

#### LIMITATIONS OF THE RESEARCH

A number of limitations are inherent in the methods outlined in this paper. These limitations are primarily due to the focus on a single satellite navigation unit. It is due to this focus that the methods outlined here may have

limited use in the field, as research has yet to be performed into the application of the aforementioned methods with other satellite navigation units. Additionally there is a limitation to the data which is acquired through the means outlined within this paper, for example location data may be present in the flash memory of the device's internal GPS module. This particular source of information is not acquired through this method.

An additional factor limiting the usefulness of data acquired through these means is the lack of an established method to extract meaningful data from the acquired images. Currently the author is pursuing further research into this area with the aim of evaluating the feasibility of analysing and extracting historical locational data from the acquired images. Indeed it is possible that the acquired images may have limited use, as the extent of data present in these images is currently unknown. It should however be noted that preliminary research into this issue suggests that at least some historical location data can be gained from these images.

## **CURRENTLY ONGOING RESEARCH**

Research is currently being conducted in order to determine the significance of evidence located on areas of the device other than the SD card. In addition to this an analysis of the contents of the SD cards utilised by the TomTom One<sup>1</sup> device. Furthermore a number of other satellite navigation units are currently being examined in order to determine the forensic value of data contained within and viable methods for acquiring that data.

## **CONCLUSION**

In conclusion satellite navigation is a field of increasing importance to law enforcement and other investigative agencies. The methodologies outlined within this paper should allow someone with adequate forensic training to acquire an image from the SD card of the TomTom One<sup>1</sup> satellite navigation system. It should however be noted that the forensic value of this image is dependant on future research into the significance of the data stored within the image, however initial research suggests that historical location data is present. Significant evidence may exist in other parts of the TomTom One satellite navigation system, however this has yet to be determined. Research is currently ongoing in this area.

## **REFERENCES**

- ACPO. Good Practice Guide for Computer based Electronic Evidence. 3.0. Retrieved 16 Oct, 2007, from [http://www.acpo.police.uk/asp/policies/Data/gpg\\_computer\\_based\\_evidence\\_v3.pdf](http://www.acpo.police.uk/asp/policies/Data/gpg_computer_based_evidence_v3.pdf)
- Braunschvig, D., Garwin, R. L., & Marwell, J. C. (2003). Space Diplomacy. *Foreign Affairs*, 82(4), 156.
- HB171. (2003). *HB171: Guidelines for the management of IT evidence : handbook*. Sydney: Standards Australia.
- Noblett, M. G., Pollitt, M. M., & Presley, L. A. (2000). Recovering and Examining Computer Forensic Evidence. *Forensic Science Communications*, 2(4).
- Theiss, A. K., Yen, D. C., & Ku, C.-Y. (2005). Global Positioning Systems: an analysis of applications, current development and future implementations. *Computer Standards & Interfaces*, 27(2), 89-100.
- TomTom. (2007). TomTom, portable GPS car navigation systems - TomTom One<sup>1</sup> Australia. Retrieved 31st October, 2007, from <http://www.tomtom.com/products/product.php?ID=399&Category=0&Lid=8>

## **COPYRIGHT**

Peter Hannay ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## BLOGS: ANTI-FORENSICS and COUNTER ANTI-FORENSICS

Glenn S. Dardick  
dardickgs@longwood.edu  
Longwood University

Claire R. La Roche  
larochecr@longwood.edu  
Longwood University

Mary A. Flanigan  
flaniganma@longwood.edu  
Longwood University

### Abstract

*Blogging gives an ordinary person the ability to have a conversation with a wide audience and has become one of the fastest growing uses of the Web. However, dozens of employee-bloggers have been terminated for exercising what they consider to be their First Amendment right to free speech and would-be consumer advocates face potential liability for voicing their opinions. To avoid identification and prevent retribution, bloggers have sought to maintain anonymity by taking advantage of various tools and procedures - anti-forensics. Unfortunately some anonymous bloggers also post content that is in violation of one or more laws. Some blogging content might be viewed as harassing others - an area known as cyber-bullying. Law enforcement and network forensics specialists are developing procedures called Counter Anti-forensics that show some promise to identify those who violate the law. However, these techniques must be used with caution so as not to violate the rights of others.*

### Keywords

digital forensics, anti-forensics, counter anti-forensics, blogs, stylometrics

### INTRODUCTION

In 2006, Time magazine's Person of the Year was "You". Blogging was in part, responsible for that choice. Time's explanation was that the Web was being revolutionized and used as "a tool for bringing together the small contributions of millions of people and making them matter" (Grossman 2006). Recent surveys conducted by the Pew Internet & American Life Project indicated that approximately 39% of adults in the U.S. read blogs and 8% of Americans participate in this form of personal publishing (Lenhart and Fox 2006).

Time saw the Web as "an opportunity to build a new kind of international understanding, not politician to politician, great man to great man, but citizen to citizen, person to person. It's a chance for people to look at a computer screen and really, genuinely wonder who's looking back at them." (Grossman 2006) As it turns out, many of those who blogged now know who has been looking back at them and perhaps wished their contributions could have been made anonymously, and kept so.

### RETRIBUTION

While many blogs are frequently posted without consideration of the appropriateness of the contents, others are posted fully cognizant that the content may be inappropriate or offensive. (Barnes 2007, Howell 2007, Williard 2007) There appears to be a tendency to reveal information or express thoughts in a blog that one would be reluctant to say in person or in a traditional print medium. This is particularly true if the blogger believes that s/he is blogging anonymously.

According to a 2007 Proofpoint survey, approximately 1 in 10 companies have fired an employee for blogging or message board postings (Proofpoint). Matthew Brown, a former Starbucks' employee in Toronto, was fired for mentioning in one posting that his supervisor did not let him go home when he was sick. (Koulouras 2004). Another blogger, Heather Armstrong, the original "dooce" (terminated for blogging) employee found out regarding blogs and employers that "They specifically will find it and read it, and all hell will break loose." (Witt 2004) Armstrong was fired from dooce.com in 2002 when her employer found the contents of her blog to be offensive.

Blogging is by definition a public activity and as such there should be no reasonable expectation of privacy. In fact, many tools such as Google's Blog Search and Really Simple Syndication (RSS) feeds make it very easy to find and read specific blog content. Although bloggers have a First Amendment right to express their opinions, they are not protected from the consequences of such expressions.

## **ANONYMITY AND ANTI-FORENSICS**

### **Blogging and Anonymity**

To avoid identification and possible retribution, some bloggers will attempt to remain anonymous. However, to maintain anonymity on the Internet, bloggers must hide not only "who" they are, but "where" they are, and "what" equipment they are using. This is all information that may readily be obtained by accessing the blog site.

A blogger's identity can be determined easily through payment and/or registration records. A blogger's identity is at risk of being exposed when the blogger uses their credit card or real name in paying for, or registering a website or e-mail address. Bloggers can prevent this by obtaining an e-mail address that provides anonymity and paying with a Virtual debit/credit card that cannot be traced back to the purchaser.

A blogger might be located through the IP address that was assigned to the computer used at the time the blogger accesses the site via the Internet. To keep their IP address anonymous a blogger can go through an intermediary on the Internet referred to as a proxy. Proxies may be used to access other proxies referenced by proxy systems such as Tor (Tor 2007). The Tor proxy system randomly selects a chain of proxies from an inventory of available proxies provided voluntarily by users of the Tor system. The system supplies proxies from multiple countries.

Information specific to the blogger, such as the operating system and browser being used, is passed through the Internet via "Headers" when access to the blog is made via the Internet. Additional information may be gathered from the blogger's system if scripts are allowed to run within the blogger's browser. To avoid passing information that might identify details of the blogger's system, "Headers" from the blogger's system, can be dynamically altered via proxies, or transcoders, capable of filtering and modifying such information. Potential threats from scripts can also be blocked via the use of filters from such proxies. Filters are capable of changing or completely removing code which can compromise a system's and/or blogger's identity. Filtering capability is provided by proxy products such as Privoxy (Privoxy 2007) that are readily available on the Internet (Privoxy 2007, Tor Download 2007, Torbutton 2007, and Vidalia 2007). Transcoding has previously been used to selectively optimize bandwidth by converting graphics files (Han 1998). Transcoding can also be used to replace HTML statements to allow web content to be displayed more appropriately on a wider variety of devices including hand-held devices. While transcoding can be server-based it can also be client-based or proxy-based using tools such as Proxomitron (Lemmon 2007). Originally, Scott Lemmon's Proxomitron was meant as a way to dynamically modify HTML to remove advertisements and pop-ups. Proxomitron eventually grew to become a general purpose proxy-based transcoder that could be a client-based or a server-based proxy. Much of the software is readily available as downloads from the Internet along with ample advice, recommendations and support from organizations and individual websites (Electronic Frontier Foundation 2005, Morris 2005, Zuckerman 2007).

### **Harmful Blogging: Juror Misconduct, Cyberbullying and Cyberstalking**

Unfortunately, sometimes blogging can run counter to the law and/or cause harm. In some cases, the blogger may be ignorant of the illegality of their actions and may or may not try to conceal their identity. Blogging has also become an avenue for both Cyberbullying (Williard 2007) and Cyberstalking (Barnes). Several cases have occurred recently where blogging has been used to inflict harm, some resulting in death (Taylor 2007). In other cases, jurors have ignored instructions from the court and have not only discussed the case prior to deliberations, but have actually written about the cases in blogs (Howell 2007). Perhaps even more insidious is when someone, such as a witness, tries to influence a jury and public opinion through information published on the Internet. In such cases, the witness would attempt to remain anonymous. Only if it could be proven that it was, in fact, a witness or other person directly involved in the proceedings, would there possibly be consequences resulting in a mistrial. It is possible that further evidence may be uncovered indicating the motives as to why a witness would be trying to influence a jury above and beyond what their testimony would provide.

Recently, President Bush signed the "Violence against Women and Department of Justice Reauthorization Act of 2005" (H.R. 3402). In effect, the bill modified 47 U.S.C. §223 to read "Whoever...utilizes any device or software that can be used to originate telecommunications or other types of communications that are transmitted, in whole or in part, by the Internet... without disclosing his identity and with intent to annoy, abuse, threaten, or

harass any person...who receives the communications...shall be fined under Title 18 or imprisoned not more than two years, or both.” (McCullagh 2006)

## **COUNTER ANTI-FORENSICS**

Counter Anti-Forensics may be used to determine the identities of bloggers who attempt to remain anonymous. Such techniques may be as basic as determining if the blogger has hidden all of the identifying tracks. For instance, a blogger may have acquired a false e-mail address and applied a fictitious name; however they might not have hidden their IP address (Zuckerman 2005). If a blogger successfully takes all of the necessary steps to hide their whereabouts, their name, and all identifying information about their equipment, what is left to identify the blogger is the content. There are methods that may tie the content of the blog to an individual, but the reliability may not be sufficient in a court of law. It may, however, be sufficient to establish probable cause and to acquire a warrant to search for additional information on the suspect's equipment. Such methods utilize stylometrics for determining author attribution.

### **Stylometrics**

Stylometrics is defined as “techniques used for the quantitative examination of textual styles, often used as a method for authorship attribution studies.” (AHDS 2007). The research into stylometrics has resulted in its application within forensics examinations. One such method of author attribution, QSUM or CUSUM, was developed by Jill M. Farrington (Farrington 1996). It creates, in effect, a cyber “fingerprint” for an author. Cyber “fingerprints” have been researched and referenced by Li and Chen (2006) as “Writeprints”. The work is based in part on earlier research into author attribution of e-mails (de Vel, Anderson, Corney and Mohay 2001). Cyber fingerprints can be classified into four categories: lexical, syntactic, structural, and content-specific (Abbasi and Chen 2005).

Lexical attributes include characteristics such as total number of words, words per sentence, word length distribution, vocabulary, total number of characters, characters per sentence, characters per word, and the usage frequency of individual letters. Syntax attributes refer to the patterns used to form sentences such as punctuation. Structural attributes refer to the text's organization and layout as well as font, hyperlink, and embedded image characteristics. Content-specific attributes are words that may be important or have special relevance within a domain (Abbasi and Chen 2005). Unfortunately, the ability to disguise authorship of electronic communications through imitation, and techniques such as cut and paste, is potentially high (De Vel 2001). However, stylometric evidence has been admitted in court, passing both the Daubert and Frye criteria (Chaski 2005).

In one recent case, *Connecticut v. Julie Amero*, a detective who was the investigator in the case was active in communicating his “case” to the public (Bass 2007). There were similarities in those communications and several blogs that were posted anonymously while the defendant was awaiting sentencing of up to 40 years. Because of the sensitivity of the case at that moment, the blogs were not further analysed, and the verdict was in fact thrown out for unrelated reasons. Had the verdict not been thrown out at sentencing and a tie-in between the blogs and the detective established, the legal ramifications might have been very enlightening,

## **CONCLUSION**

As the use of anti-forensics methods increases, the application of counter anti-forensics methods will increase as well, going beyond traditional Digital Forensics methods and incorporating stylometrics-based methods to assist in determining authorship. In fact, such methods have been deemed sufficient in a court of law to determine authorship. Results of digital forensics methods might also result in the ability to acquire warrants and enable law enforcement personnel to retrieve additional evidence in an investigation.

A significant amount of research has been done in the areas of stylometrics and author attribution. Much of the research is applied in determining whether certain material is likely, or not, to be attributable to a specific author. While much of the early research focused on literary works, these methods are now being applied to blogging posts and other electronic communications such as e-mail. Much of the research uses a defined set of authors to determine attribution. This research results in attempting to show attribution of a specific document(s) to a specific author from a defined set. More research is necessary to determine how stylometrics should be used within the digital forensics process models and the level of certainty required to show probable cause, reasonable suspicion and/or obtain warrants.

There is a need for a closer tie between stylometrics and investigations employing digital forensics. Many of the digital forensics process models start with a specific suspect and evidence potentially related to that suspect. Thus, the role has been one of confirmation rather than identification of a suspect. The process models need to

look at piercing the shield of anonymity in order to reasonably identify potential suspects and discover evidence. The question for the future is what role can the digital-forensics process play in light of anonymity? Can it be effective, not simply after the filing of charges or the issuing of a warrant, but prior to the identification of a specific suspect?

## REFERENCES

- Abbasi, A. and Chen, H. (2005). Applying Authorship Analysis to Extremist-Group Web Forum Messages. *IEEE Intelligent Systems* 20(5):67-75
- AHDS (2007). "Stylometrics".  
<http://ahds.ac.uk/ictguides/methods/method.jsp;jsessionid=4F089923B357BA5BFA77FEF1C1374391?methodId=64>
- Barnes, S and Biros, D. (2007) "An Exploratory Analysis of Computer Mediated Communications on Cyberstalking Severity". *Journal of Digital Forensics, Security and Law*. 2(3):7-27
- Bass, S. (2007). "Detective Speaks Out in Teacher Porn Case".  
<http://blogs.pcworld.com/tipsandtweaks/archives/003745.html>
- Chaski, Carole E. (2005), "Who's At The Keyboard? Authorship Attribution in Digital Evidence Investigations". *International Journal of Digital Evidence*, Spring 2005, 4(1)
- de Vel, O., Anderson, A., Corney, M., and Mohay, G. 2001. Mining e-mail content for author identification forensics. *SIGMOD Rec.* 30, 4 (Dec. 2001), 55-64. DOI= <http://doi.acm.org/10.1145/604264.604272>
- Electronic Frontier Foundation (2005) <http://tor.eff.org/eff/tor-legal-faq.html.en> (accessed September 5, 2007).
- Farrington, J. (1996). "QSUM The Cumulative Sum (cusum) Technique for Authorship Analysis & Attribution". <http://members.aol.com/qsums/>
- Firefox <http://www.mozilla.com/en-US/firefox/> (accessed September 5, 2007).
- Grossman, Lev (2006), "Time's Person of the Year: You", *TIME*  
<http://www.time.com/time/magazine/article/0,9171,1569514,00.html>
- Han, R. et al., Dynamic Adaptation In an Image Transcoding Proxy For Mobile Web Browsing in *IEEE Personal Communications*, Dec 1998, 8-17.
- Howell, D. (2007). "Blogging jury duty". <http://blogs.zdnet.com/Howell/?p=104> (accessed November 16, 2007)
- H.R. 3402. President Signs H.R. 3402, the "Violence Against Women and Department of Justice Reauthorization Act of 2005". <http://www.whitehouse.gov/news/releases/2006/01/print/20060105-3.html>
- Koulouras, Jason (2004) "Employee fired by Starbucks over Blog", *Blogcritics magazine*,  
<http://blogcritics.org/archives/2004/09/04/141004.php> (accessed July 29, 2007).
- Lemmon, S. (2007) "Proxomitron". <http://www.proxomitron.info/> (accessed November 16, 2007)
- Lenhart, Amanda, Fox, Susannah (2006) "Bloggers A portrait of the Internet's new storytellers", *Pew Internet & American Life Project*, July 19, 2006.
- Li, J., Zheng, R., and Chen, H. 2006. From fingerprint to writeprint. *Commun. ACM* 49, 4 (Apr. 2006), 76-82. DOI= <http://doi.acm.org/10.1145/1121949.1121951>
- McCullagh, D. (2006). "FAQ: The new 'annoy' law explained" . [http://www.news.com/FAQ-The-new-annoy-law-explained/2100-1028\\_3-6025396.html](http://www.news.com/FAQ-The-new-annoy-law-explained/2100-1028_3-6025396.html) (accessed November 16, 2007)
- Morris, Sofia (2005), "An Anonymous Blogger Tells All",  
<http://journalism.nyu.edu/pubzone/notablog/story/anonymous/> (accessed September 5, 2007).
- Privoxy <http://www.privoxy.org/> (accessed September 5, 2007).
- Proofpoint, Inc. (2007) "Outbound email and Content Security in Today's Enterprise" [www.proofpoint.com](http://www.proofpoint.com)
- Taylor, B. (2007). "Mom: Web Hoax Led Girl to Kill Herself"  
<http://ap.google.com/article/ALeqM5gg5xCtQtLBF6vJqWXStItGEOsJfwD8SV6U680> (accessed 11/16/2007).
- Tor <http://tor.eff.org/overview.html.en> (accessed September 5, 2007).

Tor Download <http://tor.eff.org/download.html.en> (accessed September 5, 2007).

Torbutton <http://freehaven.net/~squires/torbutton/> (accessed September 5, 2007).

Vidalia <http://vidalia-project.net/index.php> (accessed September 5, 2007).

Williard, N. (2007) "Educator's Guide to Cyberbullying and Cyberthreats",  
<http://cyberbully.org/cyberbully/docs/cbcteducator.pdf> (accessed November 16, 2007)

Witt, April (2004), "Blog Interrupted", Washington Post, August 15, 2004, p. W12.

Zuckerman, Ethan (2005) Global Voices <http://www.globalvoicesonline.org/?p=125> (accessed September 5, 2007).

## **COPYRIGHT**

Glenn S. Dardick, Claire R. La Roche and Mary A. Flanigan ©2007. The authors assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **An Overview of ADSL Homed Nepenthes Honeypots In Western Australia**

Craig Valli and Aaron Wooten  
School of Computer and Information Science  
Edith Cowan University  
c.valli@ecu.edu.au  
awooten@student.ecu.edu.au

### **Abstract**

*This paper outlines initial analysis from research in progress into ADSL homed Nepenthes honeypots. One of the Nepenthes honeypots prime objective in this research was the collection of malware for analysis and dissection. A further objective is the analysis of risks that are circulating within ISP networks in Western Australian. What differentiates Nepenthes from many traditional honeypot designs it that is has been engineered from a distributed network philosophy. The program allows distribution of results across a network of sensors and subsequent aggregation of malware statistics readily within a large network environment.*

**Keywords:** *honeypot, Nepenthes, malware*

### **INTRODUCTION**

Nepenthes (Wicherski, 2007) is a malware collection program that emulates vulnerability in Microsoft Windows operating systems. The Nepenthes system typically runs on a Linux or UNIX based system providing honeypot capabilities. Atypically honeypots are focused on tracking interactions between the attacker and the victim machine, Nepenthes however, is focused on the collection of malware and consequently has a significant change in *modus operandi*. Nepenthes does have significant logging capability and can be used alongside established honeypots such as honeyd to track interactions, however, one of its primary purposes is the collection of malware for analysis. It works by emulating known vulnerabilities such as MS03-26 (Microsoft, 2003) such that it will receive a malicious payload should it be available from the attacking entity. The emulation of the target vulnerability itself it should be noted is not a complete replication of malware signature. The emulation is sufficiently convincing to deceive the attacking entity into believing they have successfully produced a compromise in the host and are therefore confident enough to transmit the malcode to the Nepenthes server.

What further differentiated Nepenthes from traditional honeypot designs it that is has been engineered from a distributed network philosophy. Initial modules are created in the program to allow distribution of results across a network of sensors. This allows for the aggregation of malware statistics readily within a large network environment. Furthermore, Nepenthes has significant SQL logging capabilities allowing logging to popular SQL servers such as MySQL and PostgreSQL database systems for later aggregation and analysis with other tools one such implementation is Surfnet.

Surfnet (ids.surfnet.nl, 2007) is a series of programs used to aggregate and analyse the data that is trapped within a Nepenthes honeypot architecture. Surfnet is a combination of open source tools to produce a fully distributed sensor network and logging system. The use of VPN technologies (OpenVPN) allows for the use of VPN tunnels to transmit data from the sensors to the central logging computer. The model for which is illustrated in Figure 1



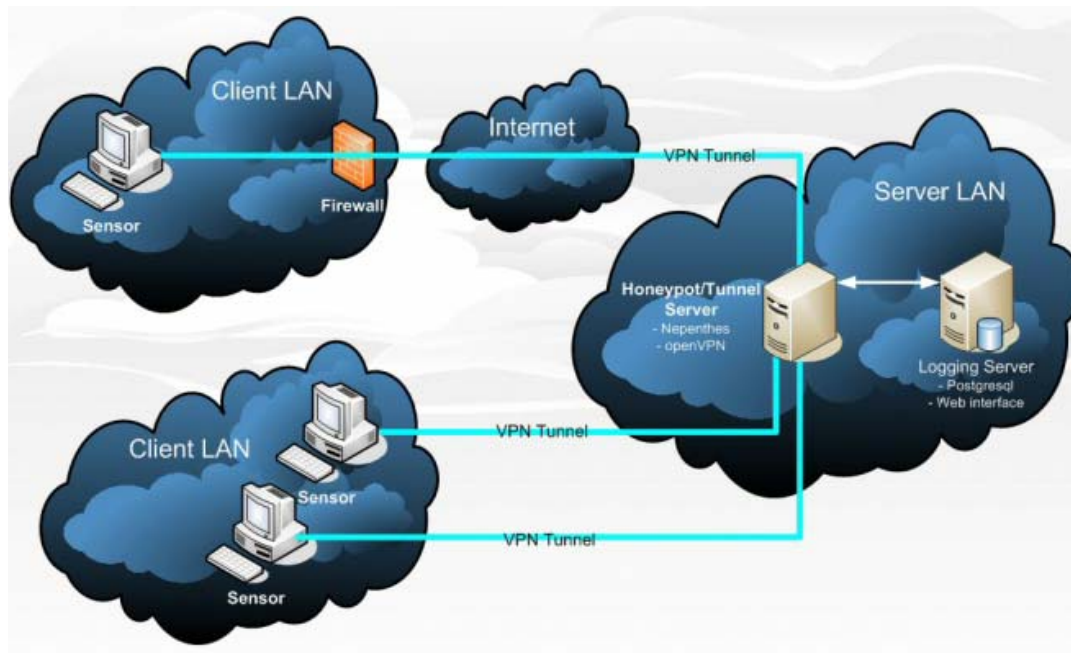


Figure 1: Surfnet Design (*ids.surfnet.nl*, 2007)

The Surfnet model as can be seen clearly from Figure 1 allows for the easy separation of the honeypot from the logging. Other honeypot systems make it difficult or often impossible to place logging and monitoring readily on another system other than the one that is running the physical honeypot.

The sensors in this design can be a CD-ROM or USB based implementation of the system this allows for very cheap resilient sensors. The sensors are designed to be updated across the wire via the use of the Subversion revision control system to update configurations and binaries. The logging server uses a variety of technologies and techniques to readily analyse the data that is drawn in from the honeypot activity. This activity is illustrated in Figure 2

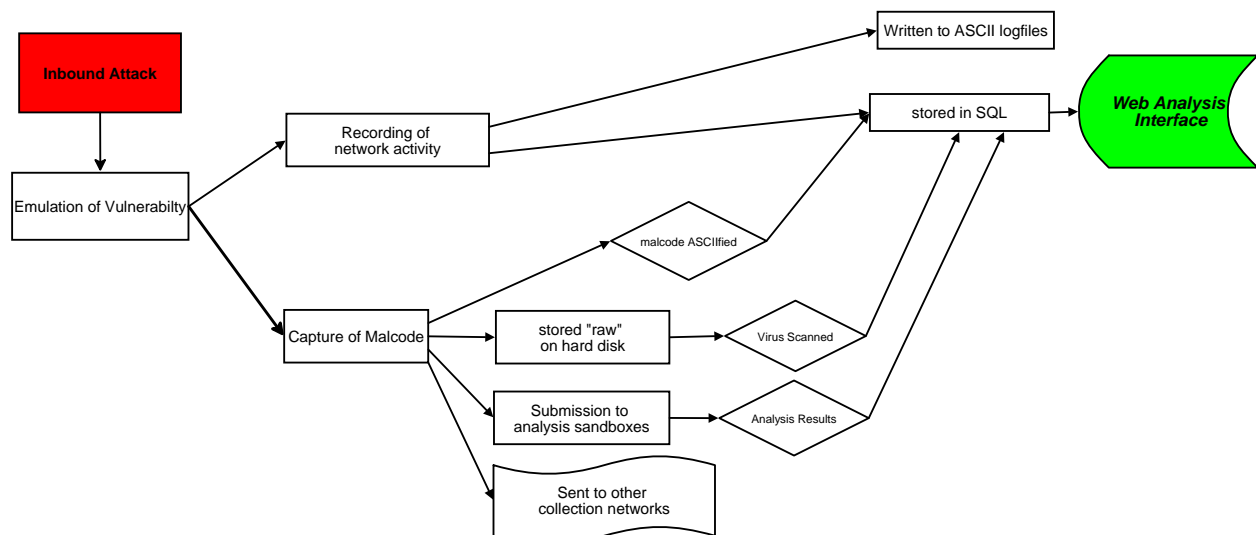


Figure 2: Nepenthes system

The use of Google maps and geographic locational services to produce a worldwide map of attackers is just one of the readily available analysis tools in the Surfnet system. The web interface also has the ability to search by each of the various attributes of the captured data set for example by attacking IP number, network sensor or malware file.

## WHERE IS THE VALUE IN THIS FORM OF HONEYPOT?

There are numerous honeypots that are designed to map the interaction between attacker and victim machine. There is sufficient evidence produced that these systems are in of themselves an invaluable tool for tracking malicious activity on a network. One of the attendant problems with these types of systems is the ongoing maintenance and analysis required for them to be effective. These systems typically presume that there is some human factor in the interaction or attack. However, this paper argues that this type of attack is becoming somewhat of an arcane way to attack and ultimately compromise systems and that there is a shift in the motivation for compromise of machines.

The increasing use of commoditised and fully automated attack tools such as *nmap*, *nessus* and others is making the attack of systems a relatively trivial task. Furthermore, the increasing customisation and simplification of malware production through the use of tools such as the MetaSploit framework ([www.metasploit.com](http://www.metasploit.com), 2007) is presenting significant challenges for the detection and remediation of malicious code. These frameworks will generate exploit that have various and often unique signatures many of which evade detection by heuristic analysers.

Research into the interactions may allow the production of attack signatures which by their very nature will allow the alerting of systems that an attack is occurring but often ultimately may not stop or prevent the compromise. New or unknown (zero day) compromises of systems are often unavoidable and can take considerable time to mitigate. This is due to a variety of issues which could be the nature of the attack being based in a protocol such as the 802.11b exploit of WEP. The fact that the software vendor may not have a fix for the particular compromise let alone be aware of its existence is an increasingly common phenomenon.

The stereotyped notion of a hacker's profile being a solitary attacker bent on destruction or simple system penetration is dated as it is ignorant. The modern hacker or *üebercracker* (Venema & Farmer, 1993) is a reality and is now network enabled and understands programming, exploit and automata of same. There is mounting evidence to suggest that the *modus operandi* of attackers is changing from a merely curiosity or personal fame perspective to one increasingly motivated by personal gain or profit through the concept of covert ownership of third-party computers. The concept of "oWning" a computer whereby a system is compromised for later purposeful use is at odds with the conventional wisdom on hackers motivations which is the defacement or destruction of a system for fame. It moves the malicious intentions from a predatory one to parasitic one based on a profit motive. Profit here does not necessarily mean an exchange of fungible goods but the availability of botnets for use in denial of service on a fee basis on the Internet is just one example of this growing trend of "ownership" with a profit imperative rather than a destructive one. There have been documented cases where these botnets have been used to create denials of service on victims with subsequent demand for the money to desist from the denial of service. In the case of online casinos or other businesses that require and rely on service availability as a revenue stream, payment of these demands is a cheaper option then continued denial of service. In addition now organised crime syndicates and enterprising hackers now produce commoditised and customised malware. These off the shelf malware can be purchased on-line and have full support infrastructures rivalling or exceeding service levels given by legitimate vendors to make your malware dollar go further. Hence lessening the need for expert or specialist IT knowledge to compromise or "oWn" a network of computers or enterprise.

The major and reducible threat is the "ownership" of the victim machine by a malicious entity as a result of executing malicious code on the victim machine that allows control of the computing resource by a third-party. The reduction in threat is achievable by disallowing the execution of the code itself or by disallowing the external communication to the third party which allows for control of the system on which the new malcode has been executed. One way of preventing this is the reduction in threat achieved through the collection and dissection of the offending malcode. The dissection of the malcode can allow mitigation of the threat by disallowing the execution of the code or achievement of the objective which is typically open of a covert channel for control or compromise of valuable data. As a simplistic minimum it can provide a signature for intrusion detection systems to deny download of the malware.

## OUTLINE OF NEPENTHES COLLECTION NETWORK

The current network is a network of 5 – 8 computers deployed within the geographical locale of Perth, Western Australia. The sensors are located on various ISP networks all are commodity based ADSL connections from

ADSL 512K through to ADSL2 24Mbit connections. These are further classified as being National or Tier 1 these ISPs are major national infrastructure backbone providers, Local Tier 2 – major state based backbone providers and Local Tier 3 who are state based providers who use Tier 2 or Tier 1 for backbone access.

<b>Tier 1</b>	3
<b>Tier 2</b>	4
<b>Tier 3</b>	2

When a piece of malware was successfully downloaded by the Nepenthes engine it was subjected to a range of internal and external tests. Internally the malware was run against a database of updated virus engines this update and subsequent scan of malware is spawned on a daily basis. Externally on arrival all new malware was submitted automatically to the Norman Sandbox and Anubis testing system for decode and detection of exploit. The results for each piece of downloaded malware was then downloaded and stored in the database. Furthermore, all pieces of malware downloaded are submitted to the alliance.mwcollect.org for processing from a global perspective. There is logging of all this activity to standard text log files, SQL and capture of the various outputs to disk.

## PRELIMINARY RESULTS

Table 1 outlines the detected connections to the honeypot network. It should be noted that these are connections that the complete network believes to have occurred.

Detected connections	Statistics
Possible malicious attack	70,359
Malicious attack	4,814
Malware offered	4,678
Malware downloaded	949

**Table 1 – Detected Connections**

The network detected 70359 attacks of which 4814 (6.8%) it believed were malicious attacks. Of these 97.2 % (4678) offered malware to the Nepenthes systems. Of the offered malware only 20.2% (949) resulted in successful download of malware to the system. Of overall possible malicious attacks 70359 these 949 malware downloads only represents 1.35% of all connections made.

Vendor A	601	63.3%
Vendor B	56	5.9%
Vendor C	646	68.1%
Vendor D	70	7.4%

*Table 2 - Virus Scanners unidentified malware*

Once downloaded the 949 were subjected to analysis by 4 mainstream virus detectors in use. At the last check Vendor B appears to have the best recognition rate of malware with only 56 tagged as Suspicious by the scanners. Vendors A and C do not identify over 60% of the downloaded malware by name or type which is of great concern. At this stage there has been no correlation to these samples with actual reverse engineering and identification of the malware which is underway as a separate research project.

	Address	Total
1	Australia1	1775
2	Australia2	623
3	Japan	412
4	USA 1	387
5	USA 2	364
6	Taiwan	364
7	USA 3	364

8	Spain	364
9	China	364
10	Australia 3	364

Table 3 - Top Attackers by Country

Table 3 indicates the Top 10 attacking IPs by number of connections. It should be noted that Australia1 and Australia2 were on the same ISP networks as 2 of the Nepenthes sensors. Australia3 was from an Eastern States ISP for which all intents and purposes is geographically and topically remote from Australia1 and Australia2.

#	Protocol	Total
1	link	1900
2	creceive	1598
3	ftp	1277
4	tftp	475
5	blink	338

Table 4 - Top 5 download protocols of all sensors

The top 5 protocols as in Table 4 hold no surprises with respect to protocols used to transport malware.

## IMPLICATIONS OF RESULTS

The initial results clearly indicate that there are significant and growing problems with the spread of network borne malware. Firstly, some infrastructure providers appear to be blissfully unaware or dilettante in providing a secure as possible network for their customers and the wider Internet community. Seven of the nine ISPs appear to provide a filtered environment for end users while the remaining two ISPs appear content to allow attacks to occur within their networks. These internally focussed attacks are sustained, persistent and prevalent evinced by the top two attacking IPs from Table 3. These ISPs are leaving the way open for possible litigation by allowing these attacks to occur and seemingly providing no mitigation of the malicious behaviour.

The age of some of the exploit code that is still in circulation would point to the fact that many of these older exploits still compromise hosts. MS03-26 is a relatively old exploit for which there is vendor based patching available that stops the exploit. Of further interest is that some of the vulnerable Microsoft platforms for this specific attack are nearing end of life support with Microsoft. Yet these types of attack were high in occurrence in the overall statistics. This trend points to organisations or individuals deploying computers with little or no patching and few protections such as firewalls.

Another trend is the speed of released exploit code spread across the networks. There is strong evidence that the release of exploit is often accompanied by a surge in the production of variants or clones of particular malware. What is further exacerbating this problem is the incorporation of code into releases of exploit frameworks such as Metasploit that likewise saw a resultant surge in the number of cloned malware. These surges have significant and profound impacts on signature based systems ability to detect and mitigate threats.

The Surfnets framework allowed for automated scanning of malware against a range of virus analysers and malware detection. Persistent zero day malware was found as a result of virus scanners inability to detect or even quarantine the new malware. The researchers have one piece of malware that remained undetected for a period of 8 months during the research by some popular virus detection engines. There were other pieces of malware that also proved resistant to detection for long windows of time some for as long as 3 months. The statistics in Table 2 see two of the virus detectors leave over 60% of the malware tagged as suspicious being unable or unwilling to make a determination of malware type. This has significant implications for effective mitigation of threats if one was reliant on these virus detection engines for protection. To digress this form of reliance on this form of protection is akin to Rumsfeldian logic "There are known knowns; there are things we know we know. We also know there are known unknowns"(news.bbc.co.uk, 2007)

The framework allowed submission of malware to analysis boxes such as Anubis or Norman Sandbox and macro virus submission sites such as virustotal.com. These malware dissection systems utilise sandboxing and other automation to dissect and analyse suspicious malware submitted to the sites. During the conduct of the current research various malware sites and blogs provided details on how to build malware that avoids detection by the Anubis system.

## CONCLUSION

Honeypot purists may see this as an approach defeatist and one that does not track the elite or discover new exploit. In the limited analysis we have conducted this type of honeypot does successfully trap zero day exploit as was demonstrated in the data collected in this research. Hackers and corporate criminals are now targeting and using the Internet as a revenue stream. Comprise of network connected computers to create botnets for sale is now almost a mundane occurrence. We need to adapt our modus operandii to mitigating the threat posed by the uebercracker and their legion of automaton attackers utilising their commoditised exploit codes or attack tools.

Nepenthes style honeypots used in a widely distributed mode provide a mechanism to gather significant credible attack intelligence and perform environmental scanning beyond traditional research based honeypots. These types of honeypots accept that compromise is not a rarity but an eventuality in the ever changing exploit space. These systems in fact welcome compromise by mimicry of successful exploit and capture of malware for dissection in an attempt to stem the tide of malware.

## REFERENCES

- ids.surfnet.nl. (2007). SURFids [SURFids]. Retrieved 13 November 2007, from <http://ids.surfnet.nl/wiki/doku.php>
- Microsoft. (2003). Microsoft Security Bulletin MS03-026 - Buffer Overrun In RPC Interface Could Allow Code Execution (823980). Retrieved 6th Feb, 2006, from <http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx>
- news.bbc.co.uk. (2007). BBC NEWS | Americas | Rum remark wins Rumsfeld an award. Retrieved 13 November 2007, from <http://news.bbc.co.uk/2/hi/americas/3254852.stm>
- Venema, W., & Farmer, D. (1993). Improving the Security of your site by breaking into it.
- Wicherski, G. (2007). Nepenthes. Retrieved 13 November 2007, from <http://Nepenthes.mwcollect.org/>
- www.metasploit.com. (2007). The Metasploit Project. Retrieved 13 November 2007, from <http://www.metasploit.com/>

## COPYRIGHT

Craig Valli and Aaron Wooten ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **Tracing USB Device artefacts on Windows XP operating system for forensic purpose**

Victor Chileshe Luo  
School of Computing and Information Science  
Edith Cowan University  
vluo@student.ecu.edu.au  
cvluo@yahoo.com

### **Abstract**

*On Windows systems several identifiers are created when a USB device is plugged into a universal serial bus. Some of these artefacts or identifiers are unique to the device and consistent across different Windows platforms as well as other operating systems such as Linux. Another key factor that makes these identifiers forensically important is the fact that they are traceable even after the system has been shut down. Hence they can be used in forensic investigations to identify specific devices that have been connected to the system in question.*

### **Keywords**

USB device identifier, forensic, artefacts, registry key, log file, Windows XP, Operating system

## **INTRODUCTION**

Demand for USB devices such as memory sticks has increased enormously in recent years. In some ways this increase has resulted in more powerful, faster and bigger capacity USB devices. Furthermore USB devices have become more popular in workplaces, education institutions etc. Many employees use them to store company information such as e-mails, corporate documents, third party sensitive data, company directories and business calendars, while Students use them to store assignments, lecture notes and other personal files. USB storage devices can also be used in contrary to the organisation policies. Their size and nature of use sometimes make them suitable to carry out malicious activities. The ability to hold gigabytes of data has certainly introduced considerable security risks, particularly in corporate environments. In addition to providing a means to move data to and from a system, USB storage devices may also be used to introduce malicious code into an otherwise protected system (Gorge, 2005).

However, the popularity or capacity of these devices is not this paper's main focus, but the ability to be able to trace the trails of these tiny devices for accountability. In this paper will discuss how USB storage devices can possibly leave identifiers imbedded within them by manufacturers on Windows XP system.

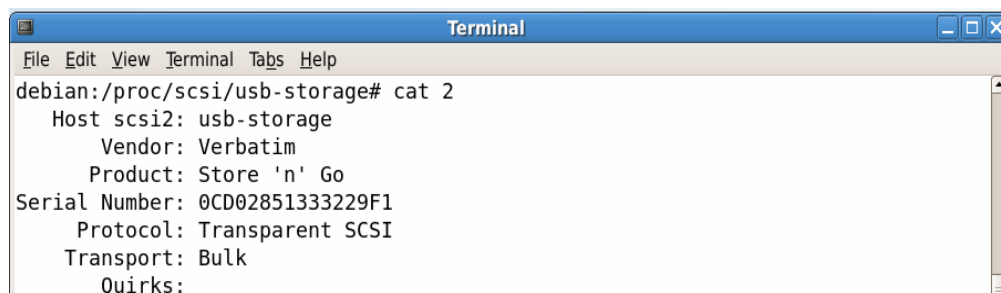
## **USB ARTIFACTS**

All USB devices have manufacturer's information embedded in them. It is this information that Windows XP operating system uses to build a unique profile that is used to uniquely identify these devices. When these tiny storage devices are attached to a USB port on the system running Windows XP, in-built drivers collect information (manufacturer specifications) from the device and then use that information to create a profile of identifiers. These identifiers end up in different locations on the system and tend to be persistent after shut down (Gorge, 2005). This ability to preserve information about devices reduces reinstallations every time the device is attached to the system. It also increases Windows ability to create profiles of smaller devices such as those devices from same manufacturer.

### **Proof of consistency**

On Linux systems these identifiers are more clear, specific and consistent. Addition information such as manufacturer's name and device description is also clearly identified.

As proof of concept, a Verbatim thumb drive was attached to Linux system (Debian) on two different occasions. The first attachment was an attempt to allow the system to collect relevant information about the device. The second attachment was done at least two weeks after the thumb drive was first attached to the system. The idea of attaching the USB thumb drive a second time was to capture USB information in memory using "cat" command as shown in figure 1 and to ensure the information belonged to the currently attached USB thumb drive.

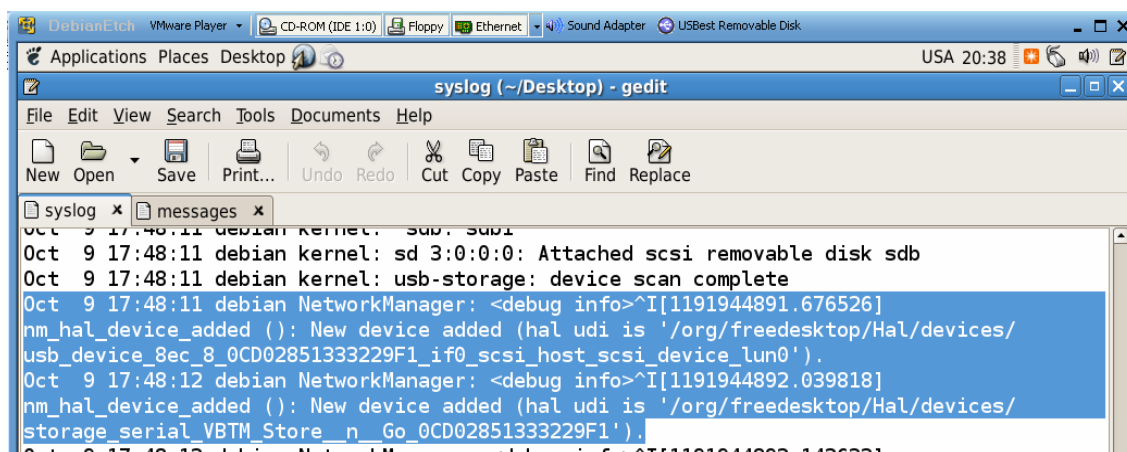


```

Terminal
File Edit View Terminal Tabs Help
debian:/proc/scsi/usb-storage# cat 2
Host scsi2: usb-storage
Vendor: Verbatim
Product: Store 'n' Go
Serial Number: 0CD02851333229F1
Protocol: Transparent SCSI
Transport: Bulk
Quirks:
    
```

Figure 1. Cached USB identifiers on Linux system

The information collected was then used to locate and compare similar information from log files such as `masseges.log` and `syslog.log`. By comparing information in figure 1 and 2, information such as serial number, abbreviation of manufacturer name (VBTM for Verbatim) and product name (Store\_n\_Go) was successfully found dating back to two weeks. This information was not only well preserved, but also matched the information collected from Windows XP system on the same thumb drive. The outlined discovery is a clear indication that some form of profile is created and preserved every time a new device is attached to the system.



```

syslog (~/.Desktop) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
syslog x messages x
Oct 9 17:48:11 debian kernel: sd 3:0:0:0: Attached scsi removable disk sdb
Oct 9 17:48:11 debian kernel: usb-storage: device scan complete
Oct 9 17:48:11 debian NetworkManager: <debug info>^I[1191944891.676526]
nm_hal_device_added (): New device added (hal udi is '/org/freedesktop/Hal/devices/
usb_device_8ec_8_0CD02851333229F1_if0_scsi_host_scsi_device_lun0').
Oct 9 17:48:12 debian NetworkManager: <debug info>^I[1191944892.039818]
nm_hal_device_added (): New device added (hal udi is '/org/freedesktop/Hal/devices/
storage_serial_VBTM_Store_n_Go_0CD02851333229F1').
Oct 9 17:48:12 debian NetworkManager: <debug info>^I[1191944892.1426221]
    
```

Figure 2. syslog file on Linux system showing the logged USB identifiers

## WINDOWS XP APPLICATION

### Windows USB identifiers

Windows XP operating system uses USB hub drivers to detect newly installed or attached USB device. When a device is attached to a port, the Windows operating system finds the appropriate driver to read and collects descriptors from it. Then the operating system uses the descriptors to build a unique profile for the device. Information collected is then used by the operating system to find the appropriate driver for the device. To achieve this, the operating system attempts to find device ID in `usbstor.inf` for those explicitly supported devices. If the USB hub driver enumerates one of these devices, the system will automatically load the USB storage port driver (Microsoft, 2007).

The device IDs for USB mass storage devices listed in `usbstor.inf` take the usual form for USB device IDs composed using information in the USB device's device descriptor. On Windows XP, a complete device unique identifier takes the following format: `USB\VID_v(4)&PID_d(4)&REV_r(4)`. According to Microsoft cooperation, `v(4)` is the 4-digit vendor code that the USB committee assigns to the vendor, `d(4)` is the 4-digit product code that the vendor assigns to the device, and `r(4)` is the revision code (Microsoft, 2007). This can be illustrated using the device instance ID from the figure 3: `USB\VID_08EC&PID_0008\0CD02851333229f1`, where 08EC is the vendor code, 0008 is the product code and 0CD0 is the revision code. All the three descriptors form a unique ID called Device Instance ID.



Figure 3. USB Device Instant ID as shown in device manager

According to Carvey and Altheide Windows also queries the device descriptor for class code (bDeviceClass field), subclass code (bDeviceSubClass field) and protocol code (bDeviceProtocol field) in order to develop a list of compatible Device identifiers (Carvey & Altheide, 2005). The general descriptors Windows uses to generate a profile for a device is shown in figure 4.

Offset	Field	Size	Value	Description
0	bLength	Byte	12h	Size of this descriptor in bytes
1	bDescriptorType	Byte	01h	DEVICE descriptor type
2	bcdUSB	Word	???h	USB specification release number in binary-coded decimal (i.e. 2.10 = 210h). this field identifies the release of the USB specification with which the device and its descriptors are compliant
4	bDeviceClass	Byte	00h	Class is specified in interface descriptor by USB working group
5	bDeviceSubClass	Byte	00h	Subclass is specified in interface descriptor by USB working group
6	bDeviceProtocol	Byte	00h	Protocol is specified in interface descriptor by USB working group
7	bMaxPacketSize0	Byte	??h	Maximum packet size for endpoint zero. (only 8, 16, 32, or 64 are valid (08h, 10h, 20h, 40h))
8	idVendor	Word	???h	Vendor identifier (assigned by the USB-IF)
10	idProduct	Word	???h	Product identifier (assigned by the manufacturer)
12	bcdDevice	Word	???h	Device release number in binary-coded decimal
14	iManufacturer	Byte	??h	Index of string descriptor describing the manufacturer
15	iProduct	Byte	??h	Index of string descriptor describing this product
16	iSerialNumber	Byte	??h	Index of string descriptor describing the device's serial number
17	bNumberConfigurations	Byte	??h	Number of possible configurations

Figure 4. A profile of identifiers Windows uses to uniquely identify a device (USB, 1999).

### Registry as a USB log file

Anyone looking into Windows registry for forensic purpose must understand that Windows registry is a repository of all information about all aspects of the computer, which includes the hardware, operating system, applications and users. In general, the investigator must be clear of what to look for and where to look for it. In terms of the USB, Windows registry stores information that ensures proper USB devices drivers are loaded, services required by applications are made available, proper application is loaded to open a file when you double click on the icon in the explorer, and that an application window appears in the proper place on your screen when you first launch it (Mee, Tryfonas, & Sutherland, 2006).



USB connections history in the registry is maintained under the following key:

HKEY\_LOCAL\_MACHINE\System\ControlSet00x\Enum\USBSTOR

The ControlSet in use by the system depends upon the data associated with the following registry value:

HKEY\_LOCAL\_MACHINE\System\Select\Current (Carvey, 2005).

Every USB device currently and previously connected to system has the device instance identifier listed under USBSTOR key as shown in figure 5.

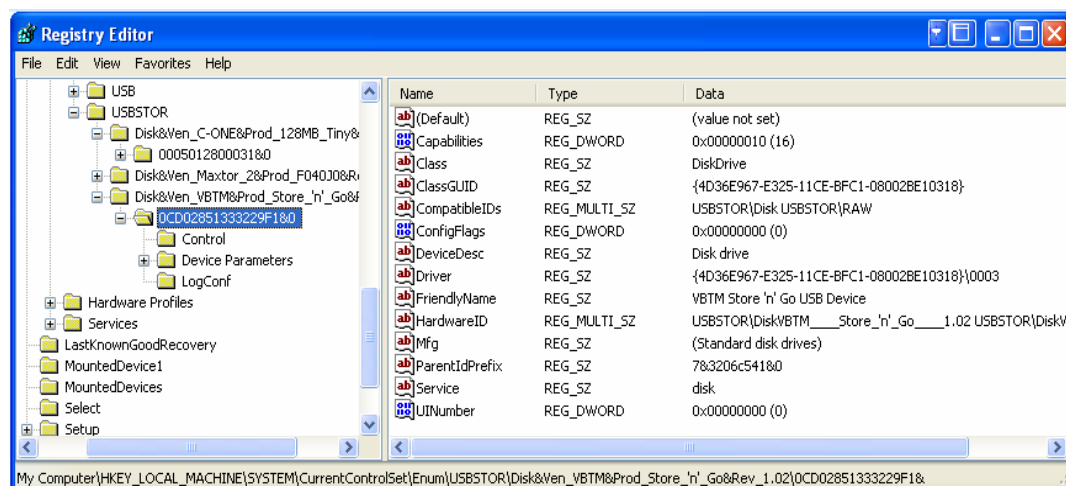


Figure 5. view USB unique ID entry under USBSTOR entry key

The highlighted entry in figure 5 is a unique device identifier, and also a unique serial number for that particular device assigned by the manufacturer. From the findings explained earlier in the paper, this number remains consistent across platforms.

According to Carvey, not all thumb drives will have serial numbers registered in the registry. Some thumb drives are manufactured without serial numbers. If the second character of the unique instance ID is a '&', then the ID was generated by the system (Carvey, 2005).

Another important registry entry is HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion. This key contains specific information about the location of plug and play device .inf files. The information to locate the .inf file is defined in DevicePath value which holds REG\_EXPAND\_SZ data types. REG\_EXPAND\_SZ is expandable, capable of holding multiple paths for the DevicePath. (Carvey & Altheide, 2005).

DevicePath registry key list of paths is used by plug and play manager to match the device identifiers with driver ranking the lowest on a scale of 0 to 0xFFFF. Once the driver is identified and loaded, the plug and play (PnP) uses the driver to retrieve any descriptors from the device and attempts to match them with explicitly supported device identifiers in the usbstor.inf. If the match is found, the usbstor.sys driver is installed and creates a new physical device object for each of the device's logical units. The newly formed physical device object has the following format: USBSTOR\v(8)p(16)r(4). To the PnP manager the PDO format is interpreted as v(8) for 8-character vendor identifier, p(16) for 16-character product identifier, and r(4) for 4-character revision level value (Microsoft, 2007).

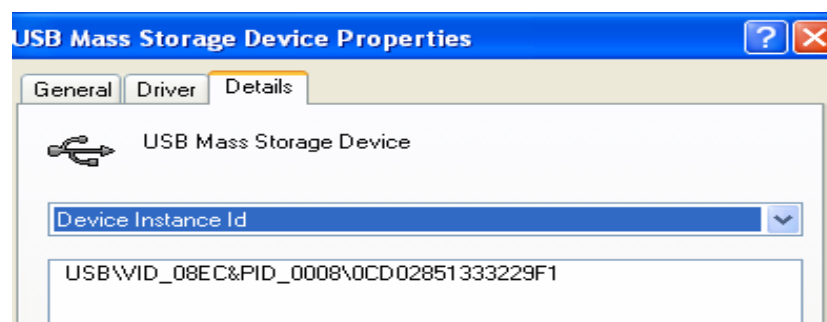


figure 6. View device manufacturer serial number via Device Manager

When PDO of a USB storage device is viewed under device manager, additional 12 characters may be appended to the end of device ID. This is the serial number of the device and the index to this serial number is found in iSerialNumber, which is a value contained in device descriptor. If the value for iSerialNumber is 0x00, then the device was not assigned serial number by its manufacturer. This 12 character number is unique and persistent across platforms, but the inclusion of this unique identifier in the device is optional as per USB specification (Carvey & Altheide, 2005).

Devices that do not have serial numbers are assigned a 12 character sequence number. This number contains an "&" character and the final value corresponds to the USB port to which the device is connected. The 12 character sequence generated by PnP manager, hence changes when the device is plugged to a different system.

In addition to these device identifiers, usbstor.inf contains compatible class identifiers for each USB based device. These devices can be CD-ROM devices, removable media devices or generic SCSI media devices. During installation these devices can be classified under any of the following classes and subclasses:

USB\CLASS\_08&SUBCLASS\_02&PROT\_50

USB\CLASS\_08&SUBCLASS\_05&PROT\_50

USB\CLASS\_08&SUBCLASS\_06&PROT\_50

All devices are firstly classified as mass storage devices (class 08h), then matched with appropriate subclass where subclass 02h is matched with SFF-8020i ATAPI CD-ROM devices, while subclass 05h is matched with SFF-8070i ATAPI removable media and subclass 06h is matched generic SCSI media. Protocol 50h simply means the devices attached are bulky-only transport protocol. According to the results from the investigation carried out earlier, the data retrieved from the USB storage device descriptor must match the USB\CLASS\_08&SUBCLASS\_06&PROT\_50 for the system to load usbstor.sys (Microsoft, 2007).

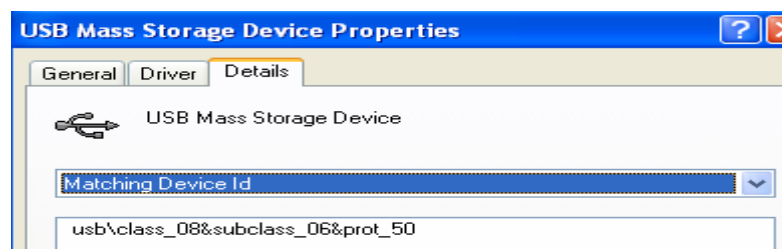


Figure 7. shows a class match for USB storage device

An example of these class and subclass identifiers can be viewed from device manager. While a USB storage device is connected to USB port, open the device manager, under the Universal Serial Bus Controller, right-click on USB Mass Storage Device and choose properties from the drop-down menu, then choose the Details tab, and select "Matching Device ID" from the drop-down menu and the corresponding value will appear below as shown in figure 7.

When compatible USB storage devices are connected to the Windows system, their artefacts are visible in Windows registry and log files. Under HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Enum\USB registry key, evidence of subkeys representing device IDs of similar format can be easily identified. More subkeys representing instance IDs follow under each subkeys identifying devices that have been connected to the system. Another important registry key for more analysis is:

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Enum\USBStor

USBStor key is similar to the device ID subkeys beneath the USB key, but values under USBStor are in human readable format while values under USB key are in hexadecimal format. As compared to the amount of subkeys under USB key, which is generally for all USB-connected devices, USBStor has fewer subkeys and specifically for USB mass storage devices (Carvey & Altheide, 2005). Beneath this key are several instance ID subkeys, representing each devices that have been connected to the system as shown figure 5.

Associating the timeline of the USB connections with user activities involving USB storage devices is important during registry analysis. When an entry is created in the registry, each keys found under that entry has a value associated with it called "LastWrite" time. This value represents the last time the registry key was modified. During forensic investigation of a USB storage device, the LastWrite times of the keys can be used to determine the timeline with respect to user activities involving USB storage devices (Carvey & Altheide, 2005).

Another interesting entry in the registry is HKEY\_LOCAL\_MACHINE\SYSTEM\MountDevices\. This particular key provides information about the drive letters association with the devices. The value in

ParentidPrefix which is found under MountDevices key can be used to exactly determine or map to the MountedDevices Registry in order to identify the drive letter to which the device was mounted. Beneath the MountedDevices registry key are several values in binary or REG\_BINARY data types as shown in figure 8.

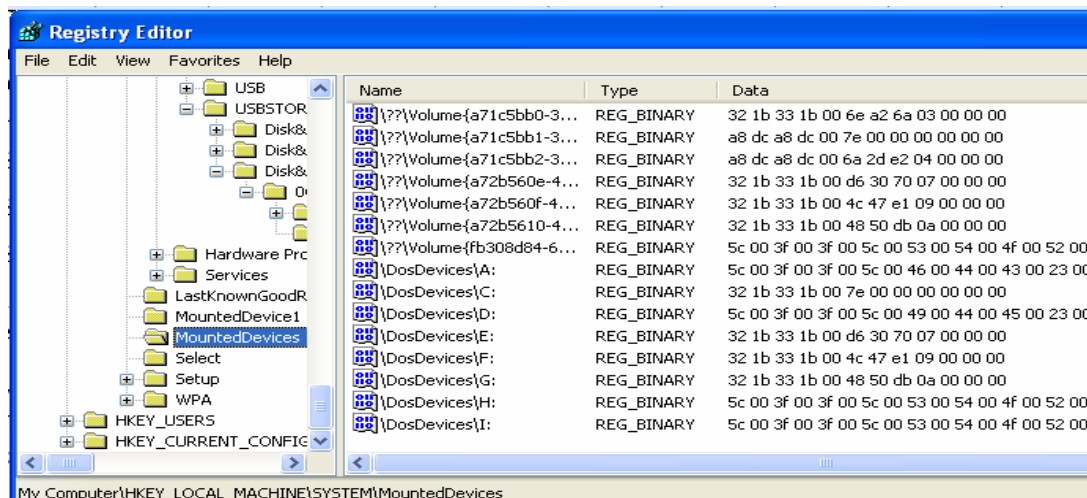


Figure 8 MountedDevices registry keys showing drive letters and unique binary

However, some of the values start with \DosDevices\ followed by drive letter e.g. \DosDevices\H. To find out, Right click on one of them and choose modify. In the “Edit Binary Value” dialog on right-most column, appears characters like this:

\??\STORAGE#RemovableMedia#7&e3d6b7b&0&RM&{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

The 7&e3d6b7b&0&RM portion of the right-most columns is the ParentidPrefix for the device. Using this ParentidPrefix we can determine the last time the device was connected to the system. To do so navigate to the following registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\control\DeviceClasses

Clicking on key identical to 53f56307-b6bf-11d0-94f2-00a0c91efb8b taken from the right-most column of the “Edit Binary Value” dialog box , reveals information about several USB devices that have been attached to the system before as shown in figure 9.

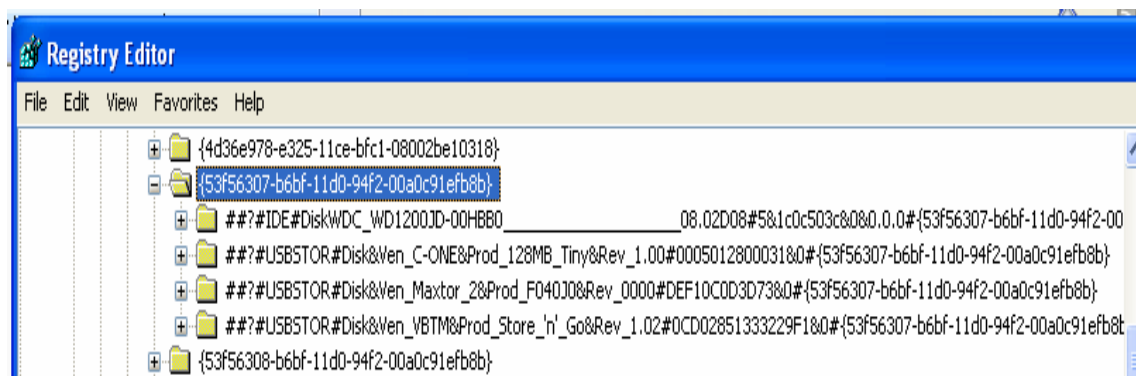


Figure 9 shows devices under DeviceClass registry key

Looking at the last subkey for the highlighted registry key in figure 9, clearly shows the unique instance identifier (OCD02851333229F1) for a USB storage device with product ID “Store\_n\_Go and manufacturer ID VBTM which is an abbreviation for Verbatim. The portion after unique instance ID (product serial number) is the ParentidPrefix value for the device (Forensic-Wiki, 2006).

To determine the LastWrite time for a specific USB device, open the registry (Click Start, Run and type Regedit.exe), navigate to the USB device key, from the file menu, click “Export”, in the “Save As” type drop-down menu, select “Text Files (\*.txt), then type the file name and press “Enter”. Open the text file using Notepad, and look at the last write time value as shown in figure 10 (Winhelponline, 2007).

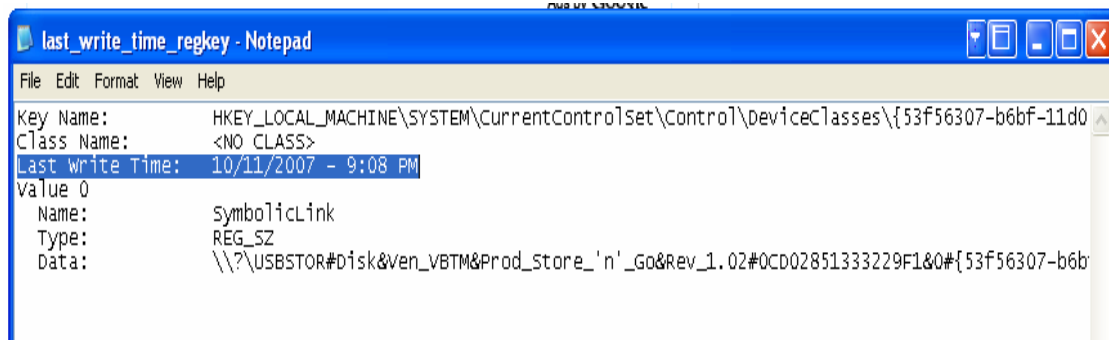


Figure 10 showing the last write time exported to text file from registry

## Windows Log Files

Windows log files can help in reinforcing the information collected from the registry. The log file of interest is setupapi.log which is found in %SYSTEMROOT% (C:\WINDOWS on the standard Windows XP install). Every installation of hardware drivers on the system is recorded in this file (Carvey & Altheide, 2005). After installing Store\_n\_Go USB storage device the setupapi.log recorded the following activities:

```
#I306 DICS_START: Device has been started.
[2007/09/30 12:27:03 1496.3 Driver Install]
#-019 Searching for hardware ID(s): usb\vid_08ec&pid_0008&rev_0100,usb\vid_08ec&pid_0008
#-018 Searching for compatible ID(s):
usb\class_08&subclass_06&prot_50,usb\class_08&subclass_06,usb\class_08
#-198 Command line processed: C:\WINDOWS\system32\services.exe
#I022 Found "USB\Class_08&SubClass_06&Prot_50" in C:\WINDOWS\inf\usbstor.inf; Device:
"USB Mass Storage Device"; Driver: "USB Mass Storage Device"; Provider: "Microsoft"; Mfg:
"Compatible USB storage device"; Section name: "USBSTOR_BULK".
#I023 Actual install section: [USBSTOR_BULK.NT]. Rank: 0x00002000. Effective driver date:
07/01/2001.
#-166 Device install function: DIF_SELECTBESTCOMPATDRV.
#I063 Selected driver installs from section [USBSTOR_BULK] in "c:\Windows\inf\usbstor.inf".
#I320 Class GUID of device remains: {36FC9E60-C465-11CF-8056-444553540000}.
#I060 Set selected driver.
#I058 Selected best compatible driver.
#-166 Device install function: DIF_INSTALLDEVICEFILES.
#I124 Doing copy-only install of "USB\VID_08EC&PID_0008\0CD02851333229F1".
#-166 Device install function: DIF_REGISTER_COINSTALLERS.
#I056 Coinstallers registered.
#-166 Device install function: DIF_INSTALLINTERFACES.
#-011 Installing section [USBSTOR_BULK.NT.Interfaces] from "c:\Windows\inf\usbstor.inf".
#I054 Interfaces installed.
#-166 Device install function: DIF_INSTALLDEVICE.
#I123 Doing full install of "USB\VID_08EC&PID_0008\0CD02851333229F1".
#I121 Device install of "USB\VID_08EC&PID_0008\0CD02851333229F1" finished successfully.
```

On line number I306, the setupapi.log file recorded the time and date the device driver installation began, while on very last line shows that the device was successfully installed. By comparing the installation date from line I306 of the setupapi.log file and the LastWrite time in the registry, it is possible to determine when the device

was first connected to the system and for how long the activities might have been repeated. On line I022, the setupapi.log file recorded more vital information, which is the USB\Class\_08&SubClass\_06&Prot\_50. Subclass 06h in Windows XP system is a predefined driver for generic SCSI media; in this case the USB storage successfully installed and identified with device instance ID or serial number 0CD02851333229F1 on line I121.

## CONCLUSION

The unique identification numbers imbedded in some devices by manufacturer are returned as iserialNumber values on Windows XP system. These unique identifications should be noted to be persistent across identified platforms. The finding raises some interesting issues, for example, an administrator could gather information of good known authorised devices that have been attached to the system. From gathered information, an administrator can determine if any unauthorised USB based storage device has been installed on the restricted machine.

Investigation techniques discussed in this paper cannot only help solve USB storage related cases such information stealing, but can strongly help law enforcers have an idea of how other crimes unrelated to one discussed were committed. In explicitly material investigations, forensic investigators could equip law enforcers with information from setupapi log file showing potential devices used when committing such horrific crimes. The type of drivers installed and identifiers associated with the drivers could help identify specific devices once attached to the system in question. The following setupapi log file shows an artefact depicting a digital camera installation:

[2007/10/11 18:27:16 1488.3 Driver Install]

#-019 Searching for hardware ID(s): usb\vid\_040a&pid\_05bd&rev\_0100,usb\vid\_040a&pid\_05bd

#-018                      Searching                      for                      compatible                      ID(s):  
usb\class\_06&subclass\_01&prot\_01,usb\class\_06&subclass\_01,usb\class\_06

#-198 Command line processed: C:\WINDOWS\system32\services.exe

#I022 Found "USB\VID\_040A&PID\_05bd" in C:\WINDOWS\inf\oem18.inf; Device: "KODAK Digital Camera"; Driver: "KODAK Digital Camera"; Provider: "Eastman Kodak"; Mfg: "Kodak"; Section name: "UsbScan.Camera".

#I023 Actual install section: [UsbScan.Camera]. Rank: 0x00000001. Effective driver date: 06/14/2002.

#I393 Modified INF cache "C:\WINDOWS\inf\INFCACHE.1".

#I022 Found "USB\Class\_06&SubClass\_01&Prot\_01" in C:\WINDOWS\inf\ptpusb.inf; Device: "Digital Still Camera"; Driver: "Digital Still Camera"; Provider: "Microsoft"; Mfg: "Generic"; Section name: "PTP".

#I023 Actual install section: [PTP]. Rank: 0x00002000. Effective driver date: 07/01/2001.

#-166 Device install function: DIF\_SELECTBESTCOMPATDRV.

#I063 Selected driver installs from section [UsbScan.Camera] in "c:\Windows\inf\oem18.inf".

#I320 Class GUID of device remains: {36FC9E60-C465-11CF-8056-444553540000}.

#I060 Set selected driver.

#I058 Selected best compatible driver.

#-166 Device install function: DIF\_INSTALLDEVICEFILES.

#I124 Doing copy-only install of "USB\VID\_040A&PID\_05BD\C713\_0C0390345".

#-166 Device install function: DIF\_REGISTER\_COINSTALLERS.

#I056 Coinstallers registered.

From the log file, forensic investigators could use line #-019 to determine the type device being installed at that time and the time the installation started by referring to line above it. Line #I022 could help in depicting specific device installed including manufacturer name; in this case KODAK camera was clearly recorded with detailed information attached to it. Forensic investigators could identify specify device by using its unique ID as shown in line #I124.

To law enforcers this evidence could help answer their many questions such as whether the system was used as a storage media for criminal data or perhaps the device at the centre of an investigation might have been used to commit crime.

## REFERENCES

- Carvey, H. (2005). The Windows Registry as a forensic resource Retrieved 9 October, 2007, from [http://0-www.sciencedirect.com.library.ecu.edu.au/science?\\_ob=ArticleURL&\\_udi=B7CW4-4GX1J3B-1&\\_user=1385697&\\_coverDate=09%2F30%2F2005&\\_rdoc=1&\\_fmt=&\\_orig=search&\\_sort=d&view=c&\\_acct=C000052520&\\_version=1&\\_urlVersion=0&\\_userid=1385697&md5=f4f6c35575ded24887ccff6cdad1bc5c](http://0-www.sciencedirect.com.library.ecu.edu.au/science?_ob=ArticleURL&_udi=B7CW4-4GX1J3B-1&_user=1385697&_coverDate=09%2F30%2F2005&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000052520&_version=1&_urlVersion=0&_userid=1385697&md5=f4f6c35575ded24887ccff6cdad1bc5c)
- Carvey, H., & Altheide, C. (2005). Tracking USB storage: Analysis of Windows artifacts generated by USB storage devices. Retrieved 2 October, 2007, from [http://www.sciencedirect.com/science?\\_ob=ArticleURL&\\_udi=B7CW4-4G82Y3M-1&\\_user=10&\\_coverDate=06%2F30%2F2005&\\_rdoc=1&\\_fmt=&\\_orig=search&\\_sort=d&view=c&\\_acct=C000050221&\\_version=1&\\_urlVersion=0&\\_userid=10&md5=14db0715620630bcf24ee0ced035f073](http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B7CW4-4G82Y3M-1&_user=10&_coverDate=06%2F30%2F2005&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=14db0715620630bcf24ee0ced035f073)
- Forensic-Wiki. (2006). USB History Viewing. Retrieved 15 October, 2007, from [http://www.forensicswiki.org/wiki/USB\\_History\\_Viewing](http://www.forensicswiki.org/wiki/USB_History_Viewing)
- Gorge, M. (2005). USB & other portable storage device usage. Retrieved 9 October, 2007, from [http://www.sciencedirect.com/science?\\_ob=ArticleURL&\\_udi=B6VNT-4GY9043-8&\\_user=10&\\_coverDate=08%2F31%2F2005&\\_rdoc=1&\\_fmt=&\\_orig=search&\\_sort=d&view=c&\\_acct=C000050221&\\_version=1&\\_urlVersion=0&\\_userid=10&md5=57444a1440590bffc1945e26c93eee02](http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6VNT-4GY9043-8&_user=10&_coverDate=08%2F31%2F2005&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=57444a1440590bffc1945e26c93eee02)
- Mee, V., Tryfonas, T., & Sutherland, L. (2006). The Windows Registry as a forensic artefact: Illustrating evidence collection for Internet usage Retrieved 10 October, 2007, from [http://0-www.sciencedirect.com.library.ecu.edu.au/science?\\_ob=ArticleURL&\\_udi=B7CW4-4M0S394-1&\\_user=1385697&\\_coverDate=09%2F30%2F2006&\\_rdoc=1&\\_fmt=&\\_orig=search&\\_sort=d&view=c&\\_acct=C000052520&\\_version=1&\\_urlVersion=0&\\_userid=1385697&md5=e5322a5cb4f4119534e0a0273159db63](http://0-www.sciencedirect.com.library.ecu.edu.au/science?_ob=ArticleURL&_udi=B7CW4-4M0S394-1&_user=1385697&_coverDate=09%2F30%2F2006&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000052520&_version=1&_urlVersion=0&_userid=1385697&md5=e5322a5cb4f4119534e0a0273159db63)
- Microsoft. (2007). Identifiers Generated by USBSTOR.SYS. Retrieved 10 october, 2007, from <http://msdn2.microsoft.com/en-us/library/ms791086.aspx>
- USB. (1999). Universal Serial Bus Mass Storage Class Bulk-Only Transport. Retrieved 9 October, 2007, from [http://www.usb.org/developers/devclass\\_docs/usbmassbulk\\_10.pdf](http://www.usb.org/developers/devclass_docs/usbmassbulk_10.pdf)
- Winhelponline. (2007). Determining the "Last Write Time" of a registry key? Retrieved 15 October, 2007, from <http://www.winhelponline.com/articles/12/1/>

## COPYRIGHT

Victor Chileshe Luo ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.

## **Oops they did it again: The 2007 Australian study of remnant data contained on 2<sup>nd</sup> hand hard disks**

Dr. Craig Valli<sup>1</sup>

Dr. Andrew Woodward<sup>2</sup>

School of Computer and Information Science  
Edith Cowan University, Perth, Western Australia

c.valli@ecu.edu.au<sup>1</sup>

a.woodward@ecu.edu.au<sup>2</sup>

### **ABSTRACT**

*The 2007 study used a biased selection process where the primary focus was the purchase of high-speed SCSI drives and drive packs, in addition 2.5 inch laptop drives were targeted. Conventional IDE based hard drives were also examined in the study. A total of 84 drives were examined this year, 23 yielded data that represented significant and in some cases profound exposure of data. Encouragingly more hard disks were erased in this study than in previous studies. However, there is still a significant gap in erasure procedures in organisations, which is particularly concerning given that the drives were from large corporations.*

### **Keywords**

Hard disks, forensics, erasure, enterprise drives

### **INTRODUCTION**

The Australian study this year used a biased selection process where the primary focus was the purchase of high-speed SCSI drives and drive packs, with 2.5 inch laptop drives also targeted. This was done with the intent of building a profile of large corporate and government sources and essentially this was achieved as an outcome. In addition, USB memory sticks were also targeted this year although there were only three able to be purchased and analysed in the study. The sources for the hard disk drives were a mixture of national on-line auctions and also traditional face to face auctions in the metropolitan area of Perth, Western Australia.

The research of the disks was again undertaken using only tools that achieved the same effect that it was considered would be available to anyone who had obtained such disks. Should the hard disk not boot in suitable hardware, tools were used which carried out the same functions as the Windows Unformat and Undelete commands. In these cases a hex editor was also used to view any information that existed in the unallocated portion of the disk. The Helix CD was utilised in the analysis, and in particular the Foremost file carver (Foremost 2007) and Autopsy the forensic browser (Sleuthkit 2006) were used in the analysis phase of disks that had been formatted.

As with previous years (Valli, 2004; Valli & Jones, 2005; Jones 2006), the first objective of the research was to determine whether there was any information on the disk that was readily recoverable with the tools identified above. The second stage of the research was to look for specific elements of information that would allow for the identification of the organisation or individual that had used the disk. Further, and, if possible, information such as the usernames, email addresses or documents, spreadsheets and data types of interest were examined. The purpose of this phase of the research was to determine the proportion of the disks that could be traced to an organisation or an individual and the level of exposure that data recovered would represent.

### **OVERALL TRENDS UNCOVERED IN THE RESEARCH**

Overall in the study once again a lack of pornographic material was uncovered given the hyperbole and claims made by some parties with respect to the use of the Internet to access this material. There was one standout case of a 40GB drive that contained over 8000 images of hardcore pornography. It had no other material on the drive at all, and would appear to have been used as a slave drive for storage of such material. Unlike the 2006 study, there were no detected cases of child pornography in any of the analysed hard disks from Australia.



A high percentage of the examined hard disks that yielded data contained significant personal and corporate exposure of confidential and commercial information. This year, however, did see an increase in the amount of competently erased hard disks. This could be a reflection of the fact that many of the hard disks were from organisations with the ability and resources to erase the hard disks. Potentially also with many of the disks coming from servers, there may be a recognised need to erase these before disposal.

## **SIGNIFICANT CASES OF INTEREST**

Case15-16AU these were unformatted 9 GB Sun SCSI hard disks that contained information from a major international merchant bank and stockbroking firm. The disk of Case16 had a last access date on the hard disk of July of 2006. The hard disk contained computer account details, VPN connection details and other password files. In the hard disks also contained sensitive network information that could be used to perform reconnaissance or attack these networks. One of the hard disks also contained information that indicated it mounted large disk arrays. What was of even greater alarm was the fact that these hard disks readily booted when installed in appropriate Sun server hardware. The partitioning configurations of the hard disks concerned indicated that they were primary operating system drives in servers from which they were extracted.

Case17-AU This was a 1 GB USB storage stick this was found to contain digital family photos, as well as tourist pictures of Sydney Harbour. In addition to these amateur shots were high-quality stock photos of a range of commercial cleaning products. The high-quality stock photos were small in file size and overall size it could be reasonably deduced they were most probably produced by a web designer.

Case18-AU was a 256 MB USB storage stick that contained only a collection of children's stories and obvious class assignments.

Case23-AU A formatted laptop hard drive from a senior academic in Information Systems school within an Eastern States based Australian university. The hard drive itself not only had the expected student results and communications but also as a result of the academic's position in the school contained confidential minutes of departmental meetings, strategic planning and course development initiatives being undertaken. In addition, the material included confidential information on staff and covered for example a sensitive ongoing industrial relations issue with one of the staff members.

Case24-AU Another 20 GB laptop hard drive appeared initially to be blank because the first areas of the drive appeared erased and in fact it was out to the 8 GB limit, the remainder of the hard drive was not. It would be reasonable to assume that the utility being used to erase the hard disk did not go beyond 8 GB limit either because it was a trial version of software or was faulty with respect to the 8 Gbyte limit on hard drive size. The hard drive still yielded confidential documents and personal information from the remainder of the hard drive.

Case25-AU This case was a unformatted 36GB SCSI hard drive from a Queensland Real Estate agent. The hard disk contained a wide range of confidential documents that would relate to real estate based transactions these included the were not limited to bank account details, credit card details, eviction notices, rent arrears notices, property sale and transfer details with the appropriate signatories to effect same. There was also a limited amount of hardcore pornographic pictures contained on the hard disk. The index.dat file upon review indicated frequent surfing of hardcore pornographic sites in addition to standard business-related activity.

Case28-AU This was a formatted hard drive from a medical provider. It contained personal medical records, including addresses, next of kin, phone numbers and other data that would be contained in a medical record. In addition, there were letters to patients, credit card and banking details of the patients as well contained on the hard disk drive. This case represented significant exposure of patients personal and financial details.

Case29-AU This was unformatted hard disk containing a 60GB NTFS partition from a religiously devout household due to the profile of web sites visited by various members of the family. There was no data of commercial nature found on the hard disk. However it, would appear that a young adult male in the house shaves their pubic regions and takes digital pictures of their erect penis for distribution via the Internet. This was substantiated by family photos found on the hard disk where the young adult male was featured clothed.

Case34-AU this was a 18 GB hard drive recovered from an accounting firm. In addition to a large number of accounting files, there were only thumbnails of pornography and other non-business related browsing present. There were personal digital photographs (wedding), some personal documents and business based form letters were also recovered. There was a large number of password protected PDF files on the hard drive no attempt was made to break the cryptography on these files due to time constraints. However, there are tools that can be used to readily extract these PDF files (REF)

Case54-AU This hard drive was a 18GB SCSI mechanism that contained Adobe Acrobat PDF files with names and addresses, vehicle types, registrations, vehicle identification numbers and policy expiry dates for a motor



vehicle insurance company. There were also large ZIP archive files with large amounts of information on many policy holders, and these also contained customer invoices as well expired policy notices. This is a significant exposure on several levels not only could these details be used to commit fraud but also allow a car thief to readily target selected high value cars for theft. The vehicle identification numbers and other details could also be useful for individuals who steal and re-birth high value or high performance cars which is an increasing trend of criminality.

Case 64 -AU This was a RAID SCSI hard disk pack, the devices were clearly marked as coming from a government based superannuation provider. This had significant exposure of confidential and sensitive information members details and statements in addition to letters of correspondence about members details. This case combined with the previous case has been the worst and most significant exposure of confidential data uncovered by the authors in 3 years of research.

Case70 This case was a pack of unformatted U160 73G SCSI drives purchased on eBay. The drives were less than 1 month old and contained data indicating that they were from a large multi-national mining company. The disks contained a large amount of corporate documents and would be problematic should they have fallen into the wrong hands for either profit through advantage or by fraudulent means. They disclosed operational procedures, purchasing details and other commercially sensitive information. It should be noted that these drives were advertised as coming from a large corporate customer in the sales promotion on eBay.

## **DISCUSSION OF RESULTS**

Once again this study has uncovered significant and alarming levels of exposure from the incorrect and dilettante disregard for defunct data. This year the hard disks were targeted in such a way that it would be highly probable that they were from a corporate entity or significant business. By profile compared to previous studies (Valli, 2004; Valli & Jones, 2005; Jones 2006) conducted in Australia the extent and level of exposure this year has been the highest. A significant number of hard drives contained large volumes of sensitive data that could be used for any number of illegal activities including identity theft, fraud and theft. As an exemplar, the case of the insurance company policies that were recovered the details contained in these would allow for targeted direct theft of high value cars and re-birthing of same. Re-birthing involves the use of substitute Vehicle Identification Numbers to hide the identity of stolen cars and these details were in the policies uncovered. What is of alarm here is that reconnaissance etc that conventional criminals would have to undertake is no longer needed they simply had to extract the records and search for the required car and then drive to the registered address.

It is of considerable concern that so many enterprise level hard disks still contained recoverable data. Of greater concern is that some of this data was essential to the companies operations, and if made public could lead to significant civil and even criminal ramifications for the company. It seems unlikely that the organisations concerned would allow the disposal of such drives without adequate erasure having first been performed. This does beg the question of how is this occurring. The most likely explanation is that individuals within these organisations are using spare or decommissioned server drives as an alternate revenue stream for themselves. A number of the descriptions accompanying these drives on auction sites referred to the items as being surplus or never used and now unwanted. It could be speculated that the surplus or unused spare description is being used to allay any fears of buyers that the drives have been unlawfully obtained. It could also be safely deduced that organisations are failing to conduct adequate risk analysis of the issue of remnant data on secondary memory devices. If not, why are these drives appearing in auctions?

It is clear that public and private organisations across a range of industry sectors are failing to discharge their responsibility to protect customer's details and sensitive data adequately. There is no official mandated law or statute that requires organisations in Australia to erase secondary memory devices such as hard disk drives. Evidence such as that uncovered in this and previous studies surely should now see this being considered by governments. Failing a legislated approach to the problem the creation for government based organisations of a centralised clearance service for the destruction/safe erasure of secondary storage media.

It could also be safely deduced that organisations are failing to conduct adequate risk analysis of the issue of remnant data on secondary memory devices. It could not be similarly reasonably argued that the problem of remnant data is not a new or unknown phenomenon (Anonymous, 2003; de Paula, 2004; Duvall, 2003; Garfinkel & Shelat, 2003; Jones, 2006; Rohan, 2002; Spring, 2003; Valli, 2004; Valli & Jones, 2005; Valli & Patak, 2005).

The risk versus return equation is simply not making sense for any modern organisation or individual. Auctioneers and sellers of these hard disks are also unwittingly providing potential criminals with targeted options for purchase with advertising that clearly indicates that the devices are from financial institutions, superannuation boards, insurance companies to name a few. One has to ask what the provenance of these devices has to do with their suitability or value for use as storage mechanisms.

## CONCLUSION

This year has again uncovered significant exposure of private, sensitive or fungible data on inadequately erased secondary memory devices. Organisations spend millions on protecting IT assets annually with firewalls, virus protection, intrusion prevention systems and other security silver bullets. We argue that these expenditures are largely symbolic and almost supercilious when hard disks are disposed of without adequate protections.

This study and others have found equally as serious exposures of data by the Small Office Home Office (SOHO) user. Likewise the SOHO user will typically purchase a virus package, use a firewall and have maybe a spyware detector but rarely is any mention ever made about safe disposal of secondary storage. The SOHO user is somewhat at the mercy of manufacturers of operating systems and hardware to enable secure erasure but one has to ask how hard is to have a “decommission” process or button that securely erases all secondary media on a computer that is about to be disposed of?

Education is one alternative but again government and policy makers have to be brought to task here. In Australia there have been massive media campaigns on the evils of drugs, or the perils of sharing unclean needles but no argument about for the need for this type of education. One has to ask where are the advertisements for reducing your risks about sharing unclean hard drives when in 2007 the world trade in cybercrime now exceeds US\$ 105 billion (www.itnews.com.au, 2007)?

Finally, the most expensive hard disk purchased in this study was a 2.5” 40 GB laptop disk that cost \$60. To juxtapose the commercial proposition, do government organisations and company boards see it a legitimate business practise to sell all corporate secrets, customer personal details or commercial sensitive information for \$1.50 per gigabyte or less from their shop front? Yet this is exactly what is occurring today somewhere at an IT disposal sale or online auction in Australia.

## REFERENCES

- Anonymous. (2003). Computer castoffs. *American Bankers Association. ABA Banking Journal*, 95(4), 22.
- de Paula, M. (2004). One Man's Trash Is... Dumpster-diving for disk drives raises eyebrows. *USBanker*, 114(6), 12.
- Duvall, M. (2003). Memory Loss ; How a missing \$100 pocket-sized drive spooked 825,000 customers of canadian companies. *Baseline*, 1(16), 65.
- Foremost (2007) Foremost, retrieved 19<sup>th</sup> October 2007 from <http://foremost.sourceforge.net/>
- Garfinkel, S. L., & Shelat, A. (2003). Remembrance of Data Passed: A Study of Disk Sanitization Practise. *IEEE Security and Privacy*, 1(1).
- Jones, A., Valli, C., Sutherland, I. and Thomas, P. (2006). The 2006 Analysis of Information Remaining on Disks Offered for Sale on the Second Hand Market. *Journal of Digital Forensics, Security and Law*, 1(3), 23-36.
- Rohan, R. (2002). The ghost of information past. *Black Enterprise*, 33(1), 47.
- Sleuthkit (2006). Autopsy overview, retrieved 19<sup>th</sup> October 2007 from <http://www.sleuthkit.org/autopsy/>
- Spring, T. (2003, May 2003). Hard drives exposed. *PC World*, 21, 22.
- Valli, C. (2004). *Throwing the Enterprise out with the Hard Disk*. Paper presented at the 2nd Australian Computer, Information and Network Forensics Conference, Fremantle, Western Australia.
- Valli, C., & Jones, A. (2005). *A UK and Australian Study of Hard Disk Disposal*. Paper presented at the 3rd Australian Computer, Information and Network Forensics Conference, Edith Cowan University, Perth, Western Australia.
- Valli, C., & Patak, P. (2005). *An investigation into the efficiency of forensic erasure tools for hard disk mechanisms*. Paper presented at the 3rd Australian Computer, Information and Network Forensics Conference, Edith Cowan University, Perth, Western Australia.
- www.itnews.com.au. (2007). Cyber-threats outpace security measures, says McAfee CEO - Security - www.itnews.com.au. Retrieved 12 November 2007, from <http://www.itnews.com.au/News/61497,cyberthreats-outpace-security-measures-says-mcafee-ceo.aspx>

## **COPYRIGHT**

Craig Valli & Andrew Woodward ©2007. The author/s assign SCISSEC & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to SCISSEC & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors