android

Rooting for Fun and Profit

A study of PHA exploitation in Android

Shiv Mel
smel@google.com
Android Security

Google

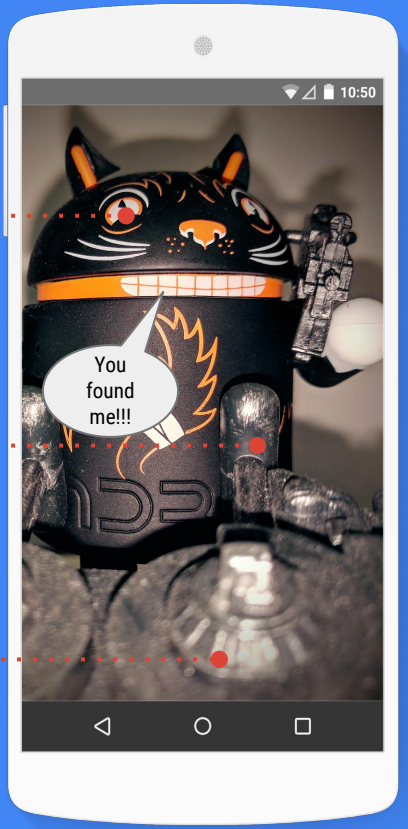
Agenda

What are we looking at?

Background
- Glossary of terms
- PHA trends

How are exploits used?
- Going from vulnerability to exploit
- Targets components

From Fun to Profit
- Rooting for Fun
- Rooting for Profit



Scope of this study - Before we begin

In Scope

- Exploits used among PHAs & Rooting Apps
- **Classes of vulnerabilities** rather than individual vulnerabilities
- Covers data from 2016

Out of Scope

- Device specific exploitation
- MitM/Network based exploitation
- App(s) specific/introduced vulnerabilities and their exploitation
- Security testing tools

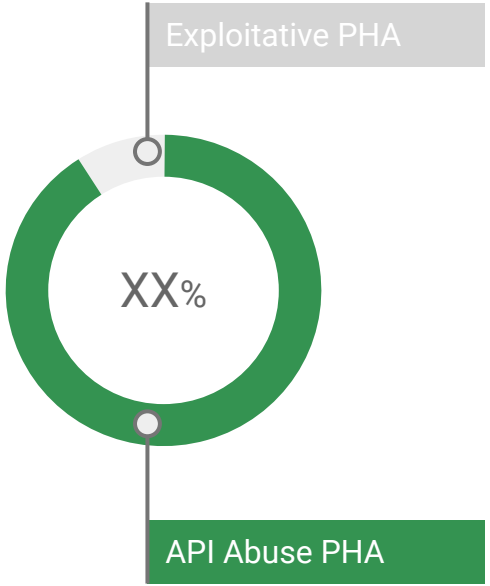
Background

Terms

	Abuse	Vulnerabilities	Exploits
Definition	Improper use of Platform APIs for often nefarious and malicious motivations. API Abuse stays within the Security model.	Weakness which allows an attacker to reduce a system's information assurance. Doesn't indicate exploitation	Code designed to take advantage of a flaw in a computer system. Takes a vulnerability and makes it dangerous
Example	SmsManager.sendTextMessage()	CVE-2014-3153	TowelRoot

PHAs that Abuse APIs

- PHAs that stay within the Security model
- These PHAs do not include exploit(s)
- Platform has flexible APIs for various functionality
- About XX% of PHA abuse APIs
- Social Engineering is typical distribution mechanisms



Vulnerability in Android

- A lot of attention on Android vulnerabilities
- Actual exploitation paints a completely different picture
- It takes work to convert a vulnerability into a useful exploit

Successful Stagefright, Certifigate Exploits

Plus, Android users who install apps outside of Google Play are 10 times

New Towelroot Exploit Hits for Easy Rooting of (Some) Android Phones

BY DAVID MURPHY JUNE 22, 2014 03:17PM EST 4 COMMENTS

Sorry, Motorola and HTC fans; the Towelroot exploit works for a number of people, just not for you.

PINGPONG ROOT

Sources (From Top): DarkReading, PCMag, XDA developers

Vulnerability + Lots of work = Exploit

Preparation includes tailoring the exploit to work on a device

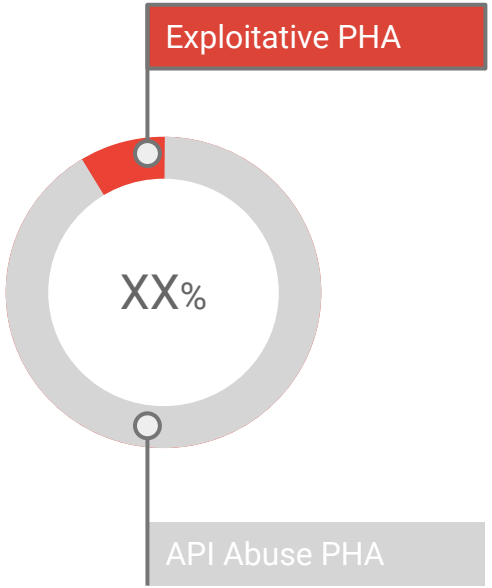
Delivering exploit into a vulnerability results in privilege escalation or provides access

Execute the exploit and maintain access and/or privilege

Each of these steps can be made complicated by security measures like ASLR, SELinux, Google Security Services etc.

PHAs that exploit vulnerabilities

- Two kinds of exploitative PHAs
  - User disclosed Rooting tools
    - Eg: KingRoot, TowelRoot App
  - Malicious rooting
    - Eg: Hacking Team's RCS
- About XX% of PHAs (includes rooting tools) exploit vulnerability/vulnerabilities
  - Most of them stick to local privilege rooting exploits

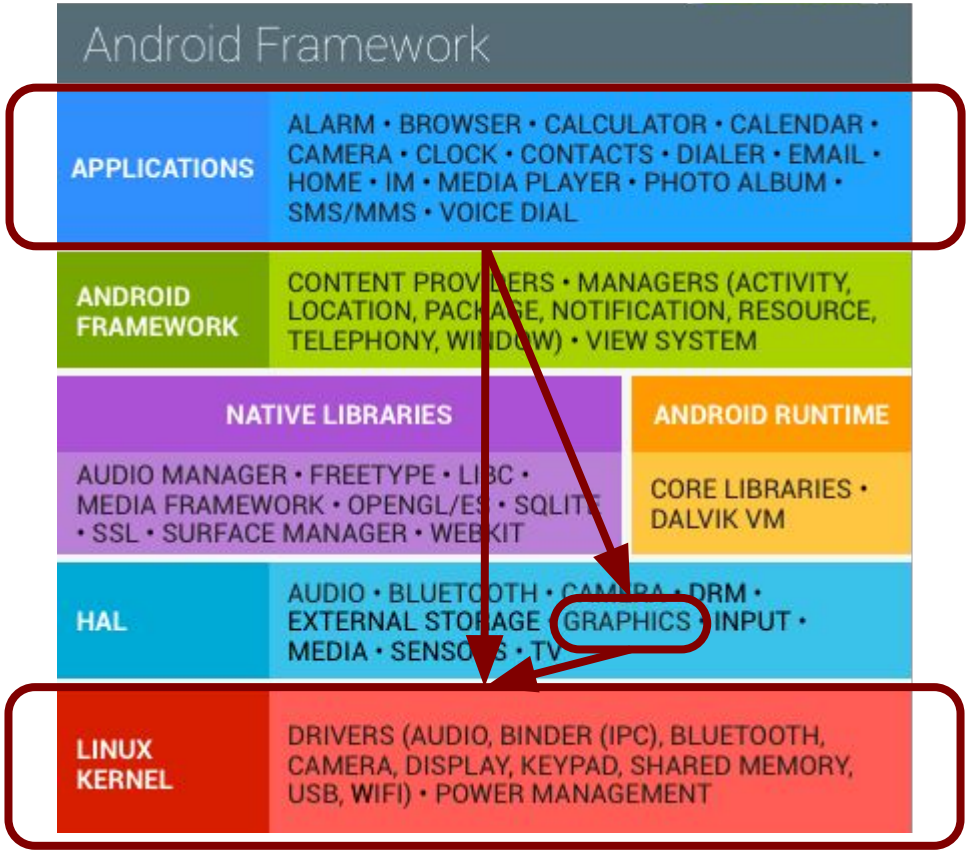


So reverse them Exploits!

What are they targeting?

Exploitative PHAs use some form of local privilege Kernel/Driver vulnerability that targets the kernel

- XX% of exploitations try multiple exploits collected into a rooting kits (eg. KingRoot)
- XX% used CVE 2014-3153 (Futex bug in the Linux Kernel)
- X% appears to target Qualcomm Drivers
- X% appears to target GPU
- Remaining are old exploits like exploid, psneuter etc.



Source: https://source.android.com/security/images/android_software_stack.png

Let’s look at rooting

Preparation is usually in the form of discerning kernel addresses

Deliver the exploit into a vulnerable codepath that allows code execution in kernel

Persistence involves dropping/customizing SELinux policies, dropping implants (modified binaries, system apps) on system partition

Skew towards Rooting kits?

Vibrant rooting community that develops user disclosed rooting tools

- End to end - from preparation, code injection to persistence
- Community ensure compatibility and deal with device quirks
- Well tested by the community hence ensures level of reliability
- Often times easy to reverse and/or easy to incorporate

PHAs authors don’t incur exploit development costs as they leverage the rooting community

What about other vulnerabilities?

Other classes of vulnerabilities are harder to exploit

- Prevented by SELinux policies
- Platform features like ASLR
- Application sandbox is potentially inadequate
- Lack of end to end exploit chain
- Fragmentation leads to frustration

This increases exploit development cost excluding the time required to tailor for target device.

Rooting for Fun!!

Kingroot

Popular community based rooting tool (user disclosed)

Collects device metrics before serving exploits

Exploits are specific to target device, however post exploit persistence is same/similar across devices

As of May, claims *103,790 supported Android models**

Support both App based and Windows sideloading based exploitation of rooting vulnerabilities

*Source: https://kingroot.net/

What’s iovec?

Upstream Linux kernel vulnerability in pipe_{read, write} and pipe_iov_*

This was patched upstream during a refactor. But patches for old stable kernel releases committed a year later

Improper handling of iov->iov_len lead to memory corruption

Vulnerable code reachable via multiple syscalls, example:

- Readv
- Writev
- Recvmmsg
- Sendmmsg

We identified exploitation of this vulnerability in Kingroot

Rooting for Profit!!

Porn Clicker

Family of hostile downloaders initially abused APIs

Monetizes through fraudulent AdClicks

Recently pivoted to a two stage exploitation

- Uses CVE-2012-2871 to initially attack WebView (patched in 2012)
- Immediately followed with CVE-2014-3153 (TowelRoot - patched in 2014)

Where did they get exploit chain from?

Webview

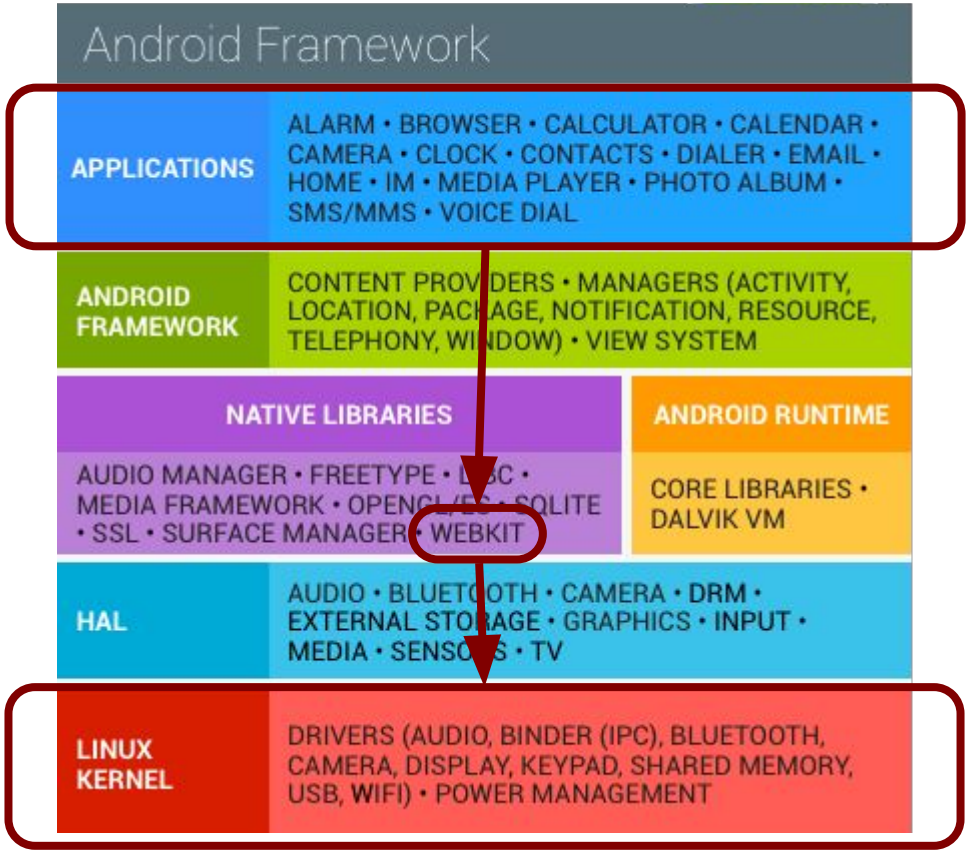
CVE-2012-2871 aka libxml in WebView vulnerability

WebView is a delivery mechanism

- Not as popular for delivery since it chains multiple other webview exploits
- In isolation doesn't provide a lot of privilege
- .X % of all **exploitative** PHAs

Kernel/Driver vulnerability for additional privilege escalation

However, this exploit chain was ….



Google

Source: https://source.android.com/security/images/android_software_stack.png

… Leaked

Hacking team hack leaked this exploit in Jul 6, 2015

PHA exploitation of same CVEs exploit chain detected in Jan 2016

PHAs authors leveraged leaked exploit

Roughly 6 months to widespread PHA use

GhostPush

Malvertising campaign uses rooting exploits

Monetizes app installations

Exploitation is tailored specific to the target device

- Gather device metrics like manufacturers and serve exploits that are most likely to succeed

Unlike Porn Clicker, exploits delivered through downloaded JAR

Why is GhostPush painful?

Rooting exploit used to drop su binary

Pushes questionable apps as system apps

Implants (System apps and su) are persistent through boot and factory reset

Requires an OTA **with a patch** to kill this campaign

Timeline from Fun to Profit is shortening

CVE-2015-1805 was made public

Kingroot ported to Android and noticed exploitation in Jan 2016

Vulnerability patched and made available in March 2016, however unpatched devices remain vulnerable

GhostPush PHA campaign detected using the same exploit in May, 2016

Also used by VideoCharm/Matrix family discovered by BitDefender

Roughly 45 days from Fun to Profit

Source: http://www.hotforsecurity.com/blog/android-malware-promises-porn-but-roots-device-and-installs-other-malware-13900.html

Conclusion

To conclude

Exploits

- Most exploitation involves rooting and exploits have good value for time
- Most exploitation targets previously patched vulnerabilities

PHAs

- Passionate Rooting Community writes end to end rooting exploit chains that gets used by malicious Authors
- Mean time from Fun to Profit decreasing

Patching

- Patching helps, squashes ability to elevate privileges
- It would squash these classes of PHAs

THANK YOU

And it’s time for Questions