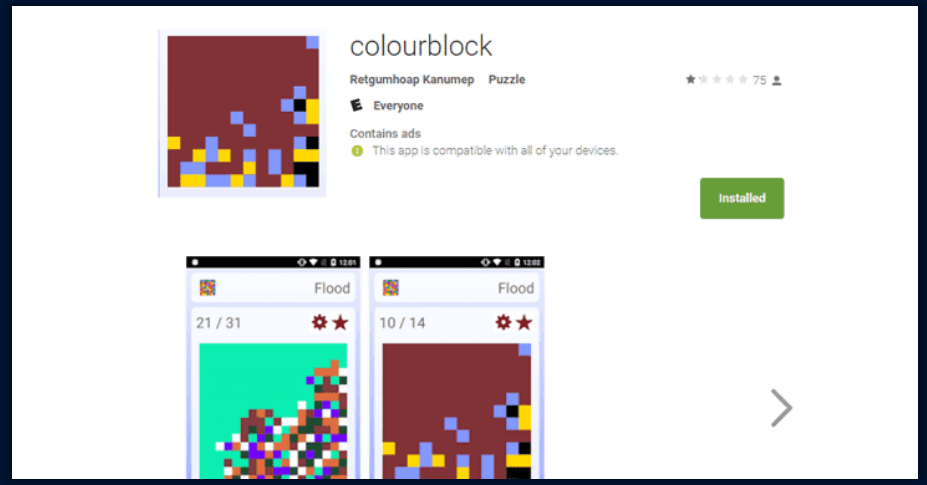
科恩实验室
KEEN SECURITY LAB

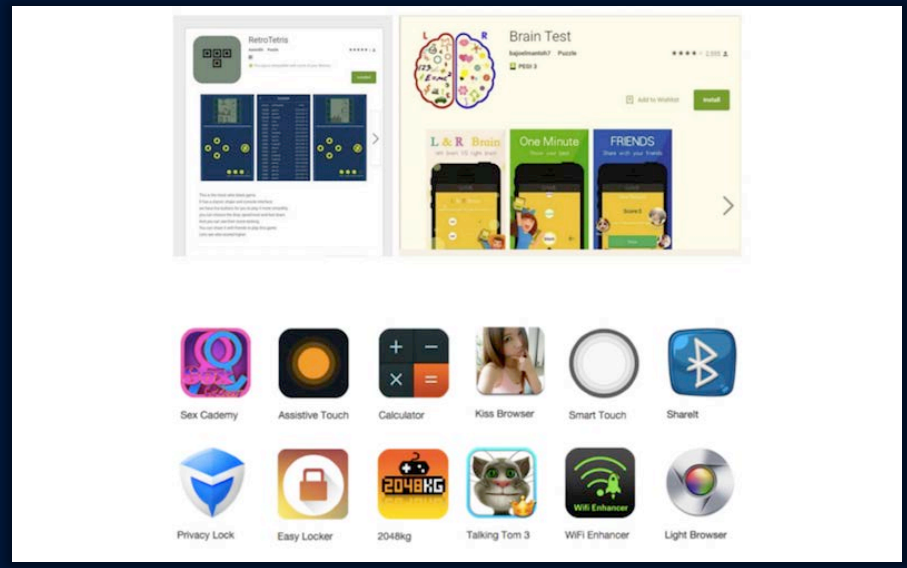
Securing Android Eco-System

11.Oct.2018

Beyond Root

- Root for fun vs. Root for profit
- There is no fun in malware





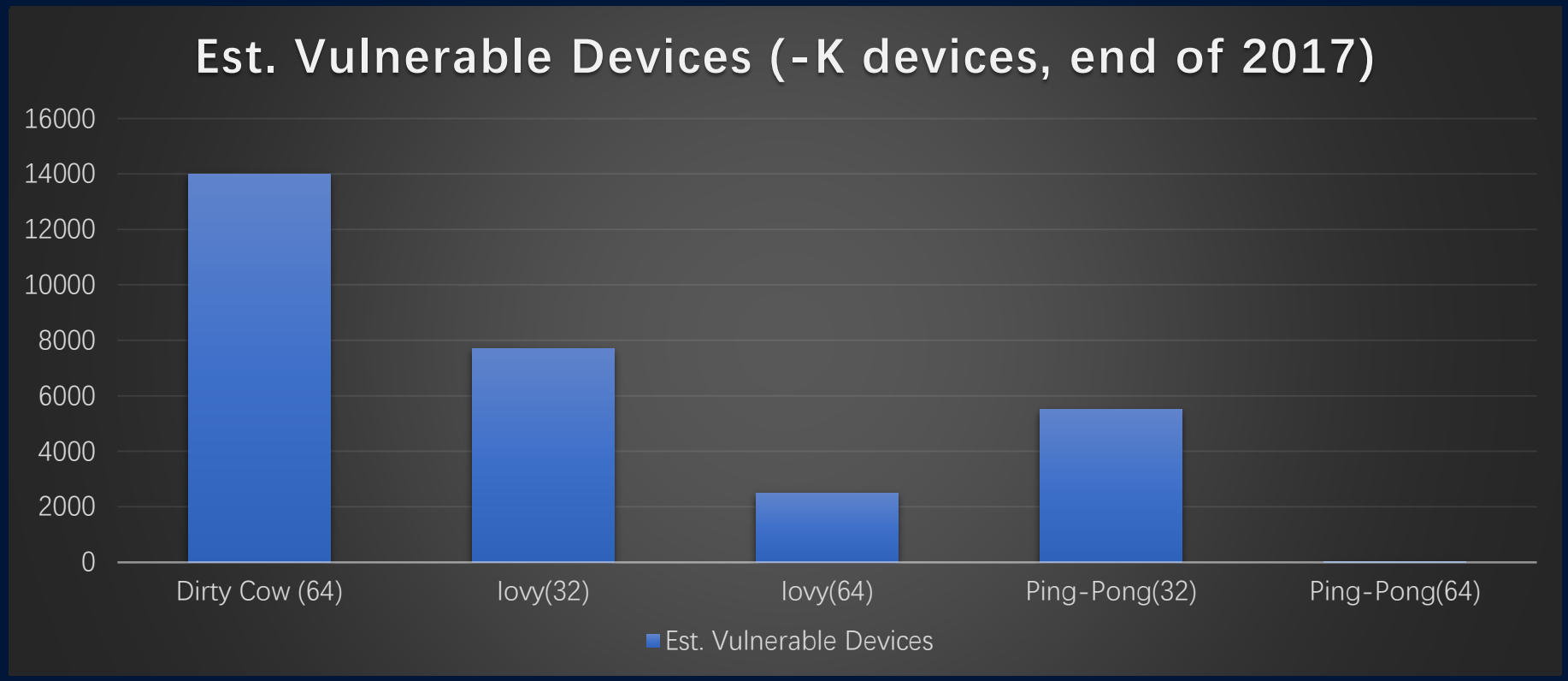
Source: Softpedia News

Malware Evolving

- …Profit driven malware “community”
- Early 2016 – stone-age exploits
- Late 2016 – early 2016 exploits

Source: https://www.youtube.com/watch?v=pB7xOnBybgU

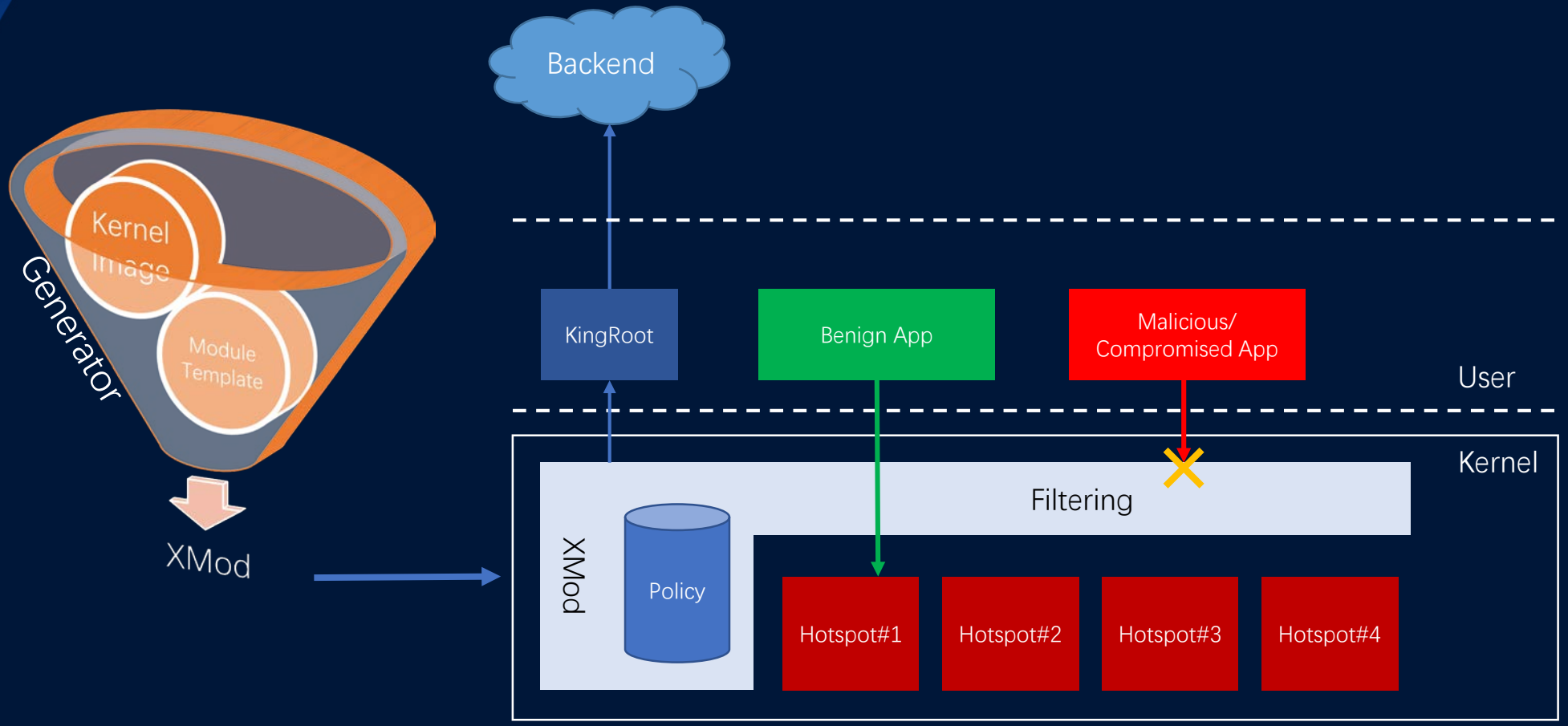
Root-able == Vulnerable



Root-able == Repairable

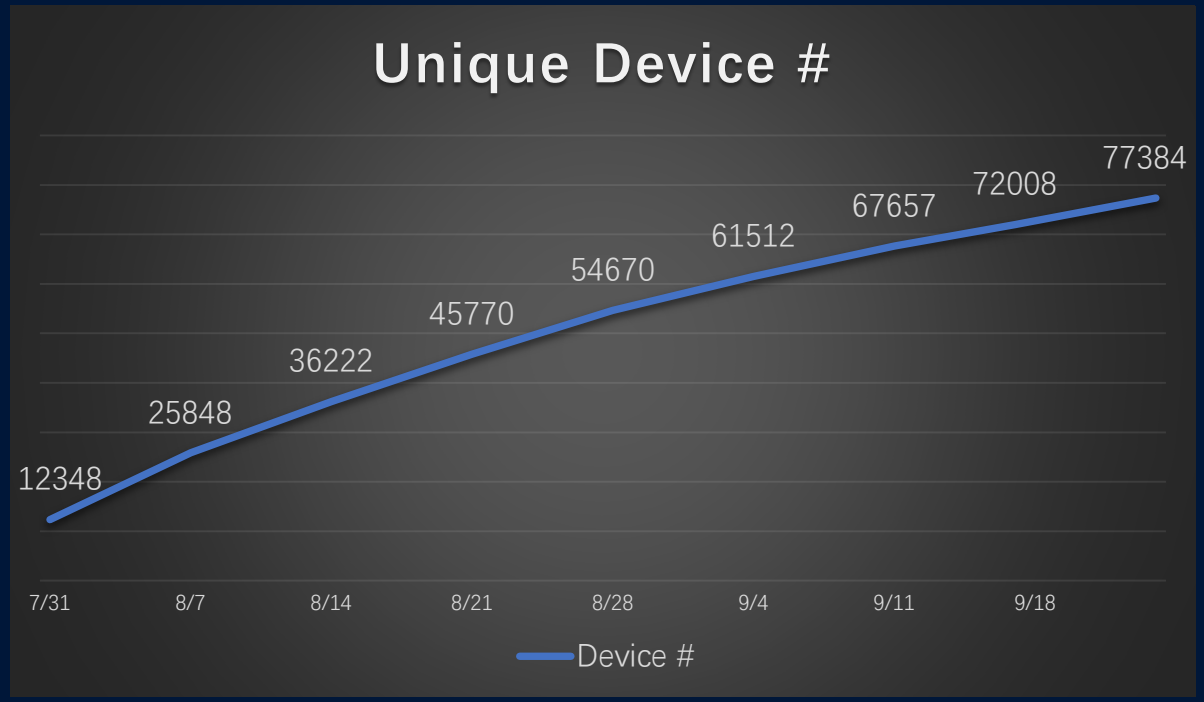
- Step 1: Exploit to own the kernel
- Step 2: … so that we can plug the loophole
- Introducing XMod
  - “root killer”
  - Internal tool for our own Android devices
  - Prototype developed after Ping-Pong root
- Tuned to fit the eco-system
  - Kernel is not REALLY open-source
  - So we have to deal with binaries

How it Works


Backend
Kernel Image
Module Template
Generator
XMod
KingRoot
Benign App
Malicious/ Compromised App
User
Kernel
Filtering
XMod
Policy
Hotspot#1
Hotspot#2
Hotspot#3
Hotspot#4

Observations

- Dozens of unique malware samples per day
- New devices join “victim club” every day



Android Eco-System

- Phone is not the only Android device facing the threat
- Those have no display…
  - Drones
  - Smart home
- …, and those have displays too big
  - Smart TV
  - Vehicles
- Connected cars
  - Big and complex
  - Cool 8-)

Android on Cars

- Most commonly used in IVI
  - “Big tablet”?
  - Opens a door to vehicle’s internal networks
- Just like phones: Varies of hardware platforms
  - Patching Support
- Unlike phones: Manufacturer
  - In house developed
  - …, or supplied by Tier 1
- Has (or will have) an app market
  - Open eco-system


ECU
Telemetry (Tbox)
4G
IVI
Ports
ECU
Gateway
Kernel
Passenger
ECU


ECU
Telemetry (Tbox)
4G
IVI
Ports
Step 1: Luring into a malicious wifi
ECU
Gateway
Kernel
Passenger
ECU


ECU
Telemetry (Tbox)
4G
IVI
Step 2: Pwn an application
Ports
ECU
Gateway
Kernel
Passenger
ECU

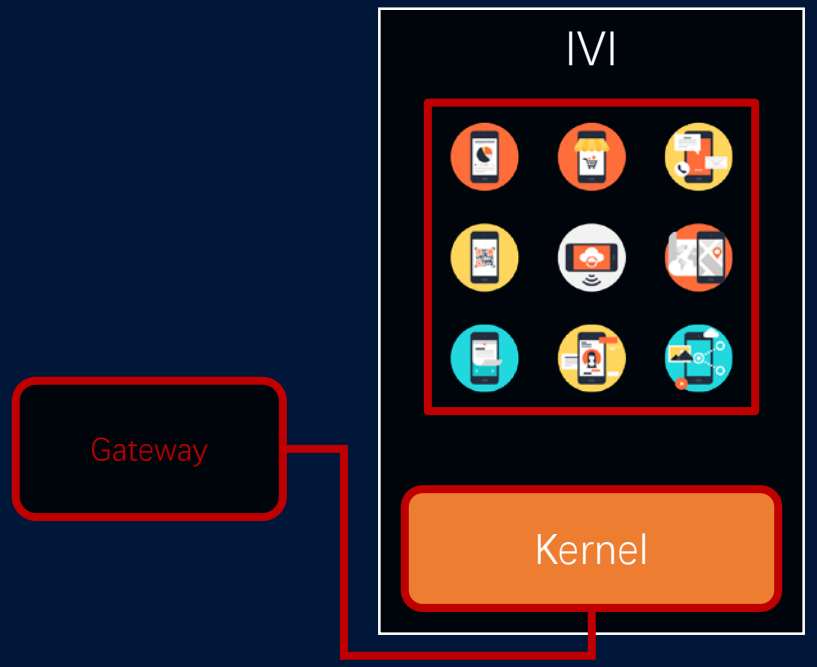

ECU
Telemetry (Tbox)
4G
IVI
Ports
ECU
Gateway
Passenger
Kernel
ECU
Step 3: Pwn the kernel
and get full control


ECU
Telemetry (Tbox)
4G
IVI
Ports
ECU
Step 4: Hack the gateway
Gateway
Kernel
Passenger
ECU


ECU
Telemetry (Tbox)
4G
Step 5: Send CAN packets
and have fun!
IVI
Ports
ECU
Gateway
Kernel
Passenger
ECU

Who is the Culprit?

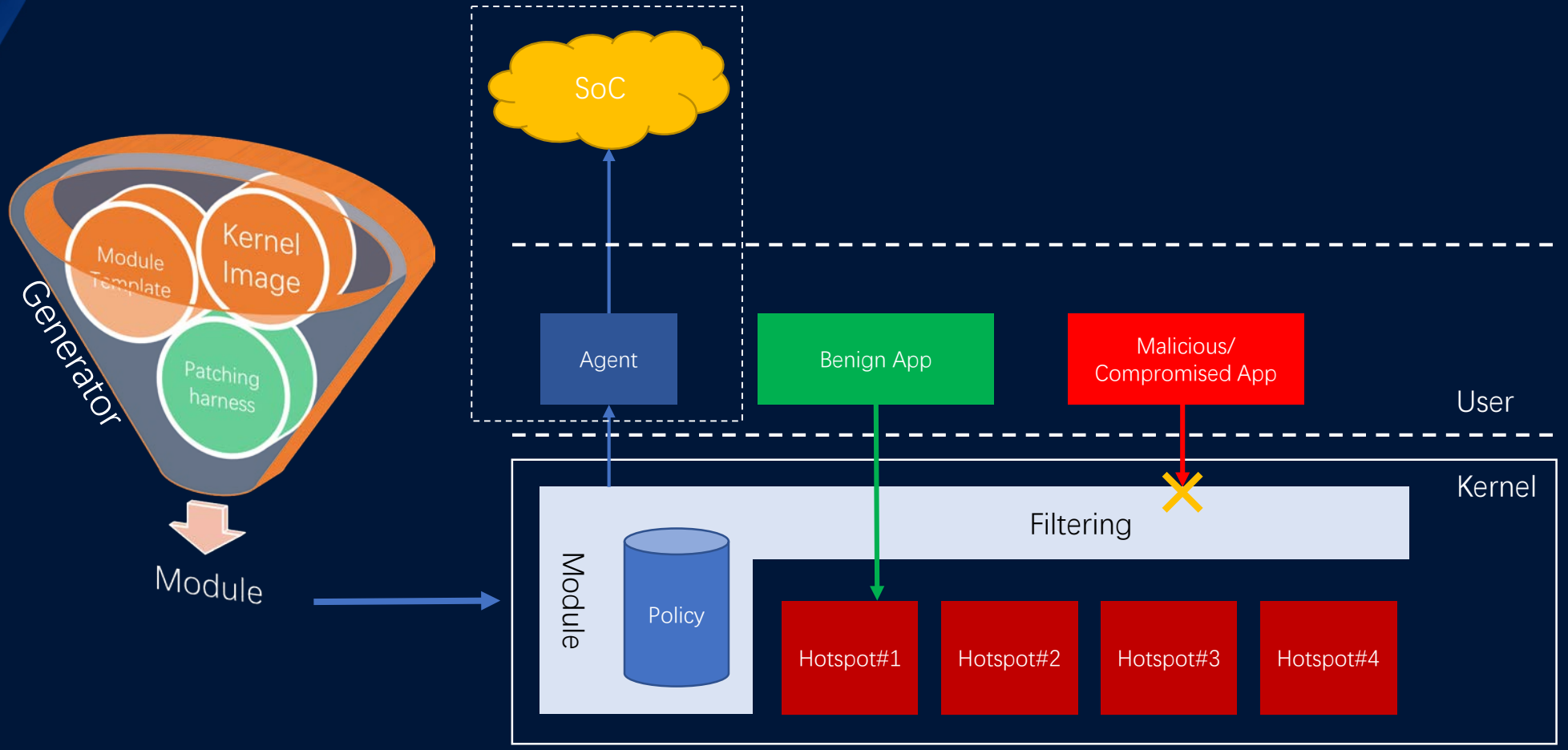
- Application
  - Vulnerable
  - Malicious
- Vulnerable Kernel
- Gateway
  - When privilege is gained…
  - … or through ODBII
- ECU
  - Supposed be protected by gateway



Kernel Patching…

- Patching cycle: monthly ==> NEVER
- Support lifecycle: 5 to 10 years

From Phones to Vehicles



Hardening

- For those whom has full control over it
- Most actively developed part of Linux
  - A business by itself
- Eliminating the consequence rather than source
- Typical hardening “hotspots”
  - Stack and heap
  - Ref-count
  - Control flow

What's the Problem

- Kernel versions
- What's missing on a typical 3.18 kernel
  - KASLR
  - __ro_after_init
  - THREAD_INFO_IN_TASK
  - Probably buggy -fstack-protector
  - Randomized free-list
  - STATIC_USERMODEHELPER
  - Removal of set_fs
  - …
- Also missing all HW capabilities after armv8.0

With source access

- Fully utilizing HW capabilities
  - E.g, PXN on armv7
- Backporting critical security enhancements
  - Aka, those we find annoying in our exploits
- Implementing new hardening features
  - Partial CFI (anti JOP/ROP)
  - Read-only dir-map/trampoline-map
  - SW pointer authentication
  - …

A Trivial SW Ptr Auth

Goal: Context-aware pointer signature

```
[   16.879643] arch_process_rcu_cb_ptr: ptr 0xffffff9095bec0fc -> 0xa2dc5d4c37626220
[   16.879724] arch_process_rcu_cb_ptr: ptr 0xffffff9095bec0fc -> 0xf5740ae460ca3588
[   16.879873] arch_process_rcu_cb_ptr: ptr 0xffffff9095bec0fc -> 0x97da684a02645726
[   16.879984] arch_process_rcu_cb_ptr: ptr 0xffffff9095bec0fc -> 0xec4913d979f72cb5
```

- Ch#1: How to generate the key?
- Ch#2: Where to store the key?
- Ch#3: What to protect?

A Trivial SW Ptr Auth

- Ch#1: How to generate the key?
- kASLR capable kernels
  - EFI_RNG_PROTOCOL
  - Use the bits kASLR is not using
- Non-kASLR?
  - Pass through register from EL2
  - Time value hash?
  - Vendor specific RNG

```
author      Runmin Wang <runminw@codeaurora.org>
committer   Runmin Wang <runminw@codeaurora.org>
commit      8ecc3f600f9f42ab0a1babf52e5f76f486dccbdb (patch)
tree        f6fedc86d8e4b1e1c774ab7965fec3378976e445 /QcomModulePkg/Libra
parent      ce0331c7966d5f9ba39bb4c6a5e66cccce92af0c (diff)

QcomModulePkg: Add support for kaslr

Add support to generate a 64 bits random number for Kaslr(Kernel
address space layout randomization) as the kaslr-seed.

Change-Id: Ib63722c86a5fccc3bef8a561f33e12c8849475bd
```

```
33          node = fdt_path_offset(fdt, "/chosen");
34          if (node < 0)
35                  return 0;
36
37          prop = fdt_getprop_w(fdt, node, "kaslr-seed", &len);
```

```
132         /* use the top 16 bits to randomize the linear region */
133         memstart_offset_seed = seed >> 48;
134
135         #ifdef CONFIG_RCU_CALLBACK_GUARD
136         rcu_cb_key = seed & 0xffffffff;
137         #endif
```

A Trivial SW Ptr Auth

- Ch#2: Where to store the key?
- In memory?
  - Not info-leak proof
  - Leaked key == failure
- In register?
  - GCC option: -ffixed-reg
  - Avoid using particular register for key storage
  - Potential leakage on stack?
  - Or any better idea?

A Trivial SW Ptr Auth

- Ch#3: What to protect?
- Easy-to-catch usages
- Metadata
  - Function pointers
    - Usually well-organized usage
    - E.g, RCU callback pointers
  - Data pointers
    - Linked lists
    - Free-list
    - …

A Trivial SW Ptr Auth

- Towards practical
- Better coverage (compiler?)
  - “vtable” pointers
- Better crypto algorithms
- Module binary compatibility?

Vehicle SOC

- Kernel is important, but not the only one
  - USB devices
  - SMS, Wi-Fi, Bluetooth
  - OTA packages
- Awareness is just as important as intervention
- A **security** operation center to…
  - Monitor
  - Assess
  - Defend
- Defeat sophisticated attacks
  - Big data
  - Cloud-side policies

KEEN SOC

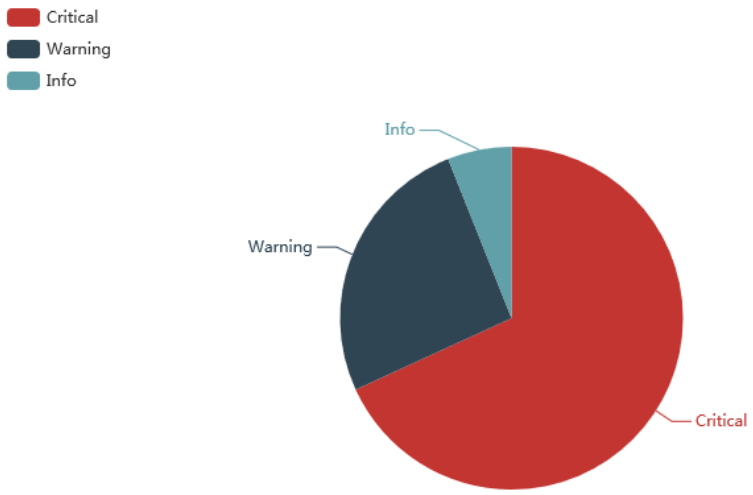
English

- Dashboard
- Vehicle Asset
- Vehicle Status
- Vulns Exploit
- USB Monitor
- WiFi Monitor
- Sandbox Monitor
- Alert Settings
- System Settings

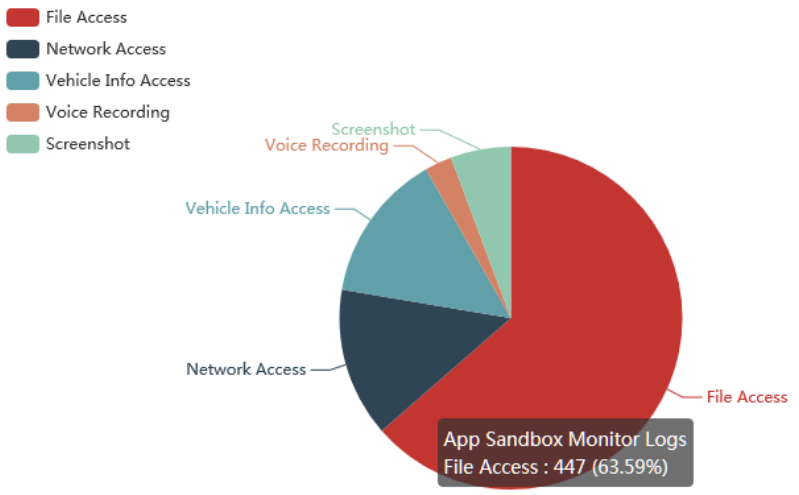
Vehicle Asset	Vehicle Status	Vulns Exploit	USB Monitor	WiFi Monitor	Sandbox Moni
0 Recent	0 Recent	0 Recent	0 Recent	0 Recent	
119 All	4449 All	4720 All	121 All	112 All	
Detail	Detail	Detail	Detail	Detail	Detail

Statistics

Vulns Exploit Logs



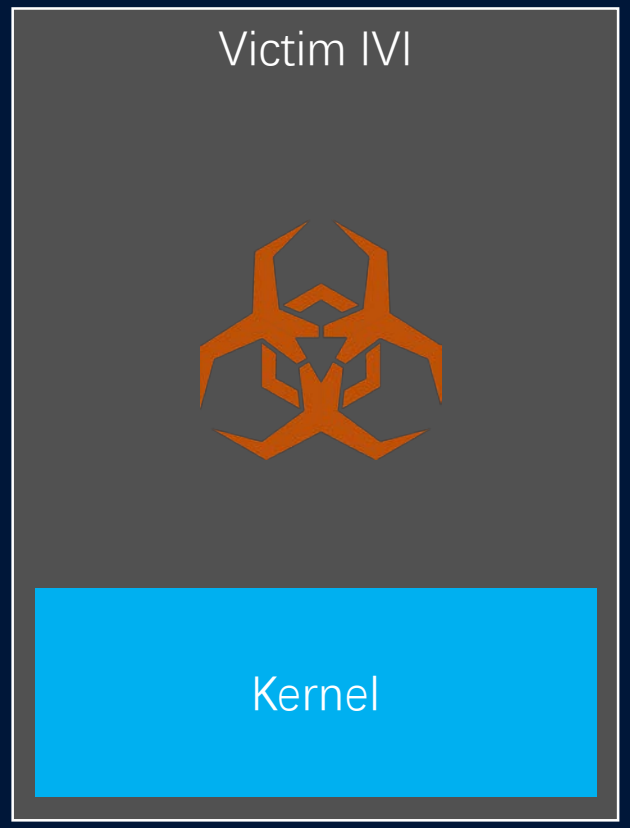
App Sandbox Monitor Logs



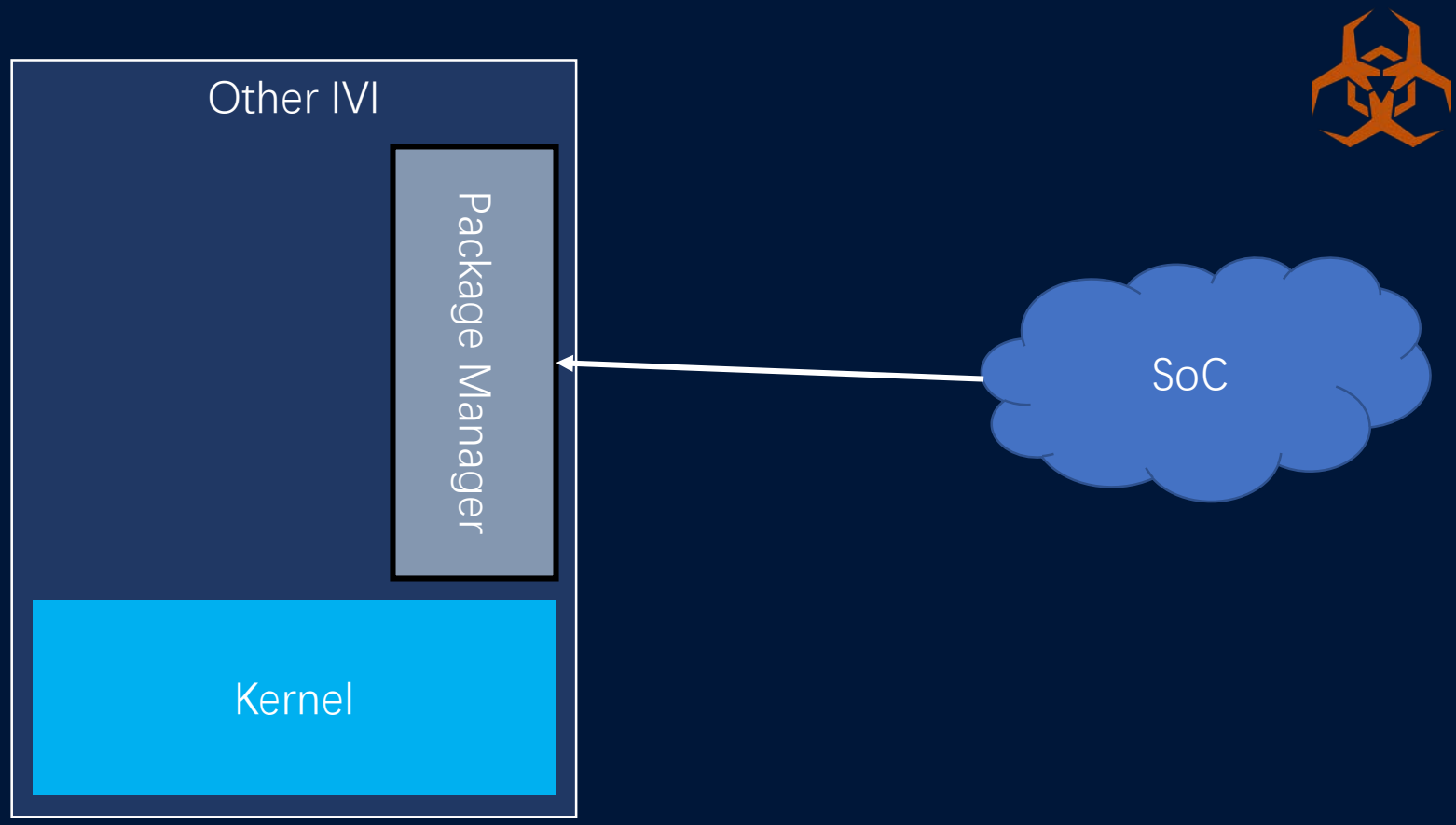
WiFi Monitor Logs

USB Monitor Logs

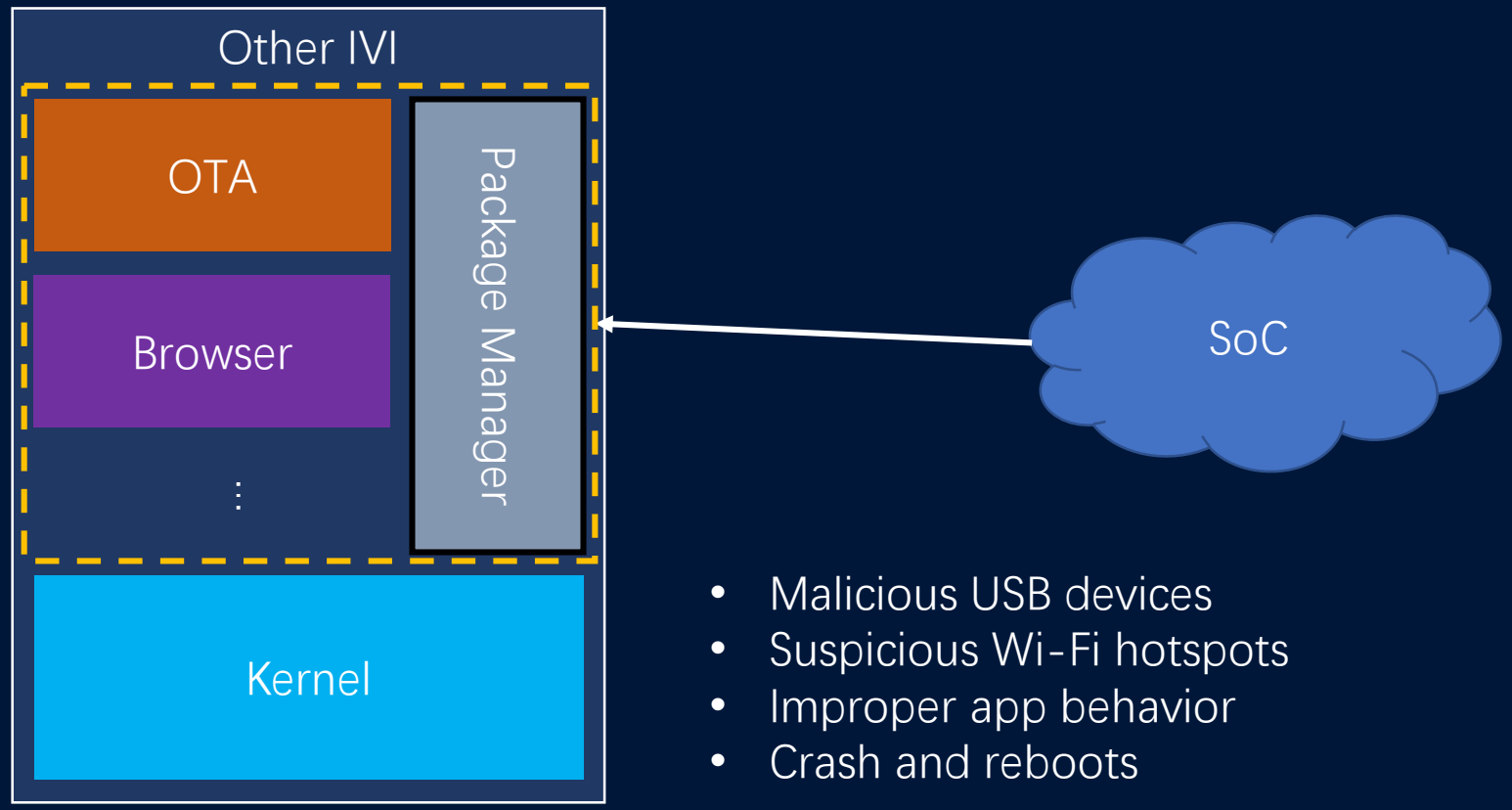
Malware Scenario



Malware Scenario



Malware Scenario



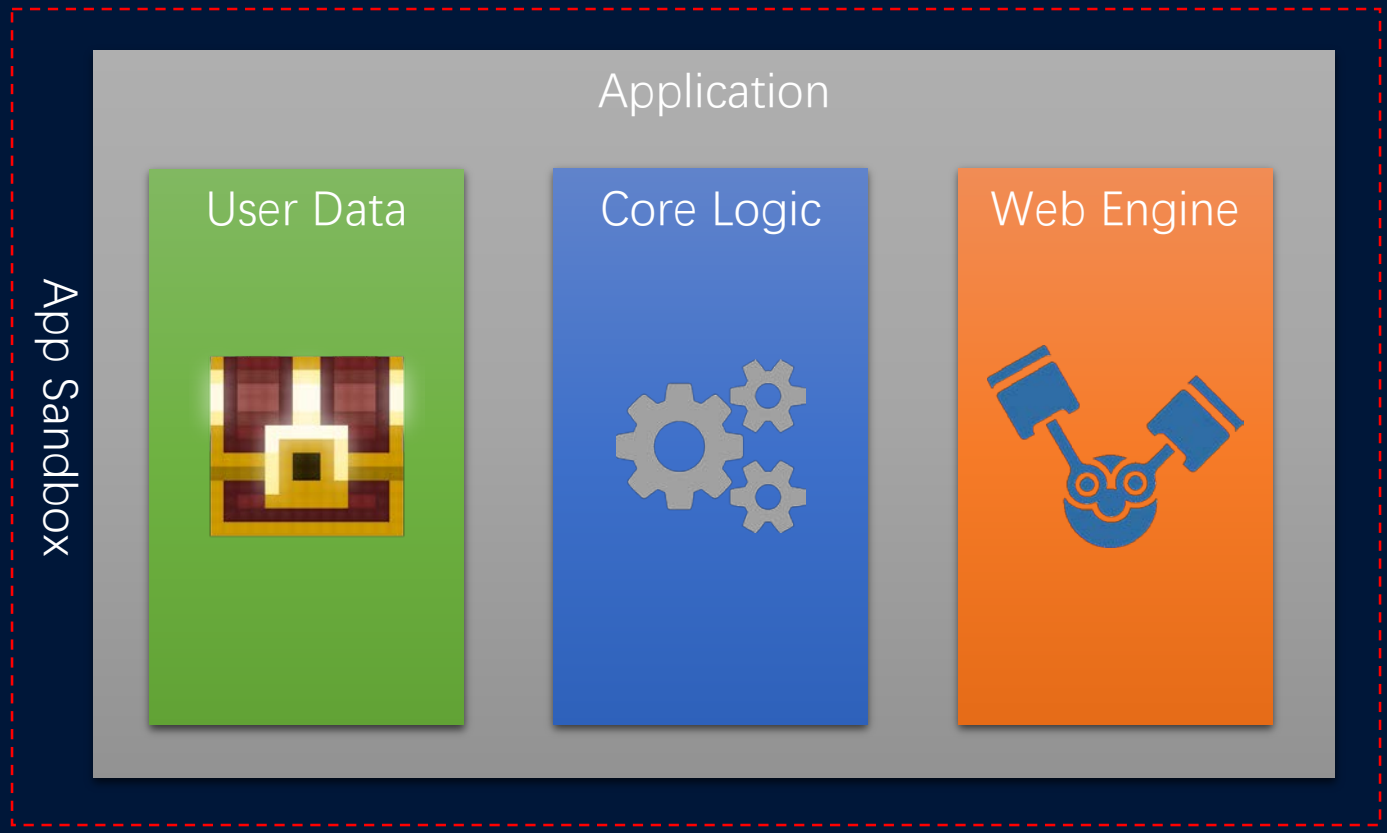
- Malicious USB devices
- Suspicious Wi-Fi hotspots
- Improper app behavior
- Crash and reboots
- ...

Apps - Troublemaker

- Popular victim of remote attack vectors
  - And proxy for further attack
- Most “open” part of the eco-system
  - Quality control
  - Security
  - Sustainability
- Strategies
  - Prevention
  - Damage Control

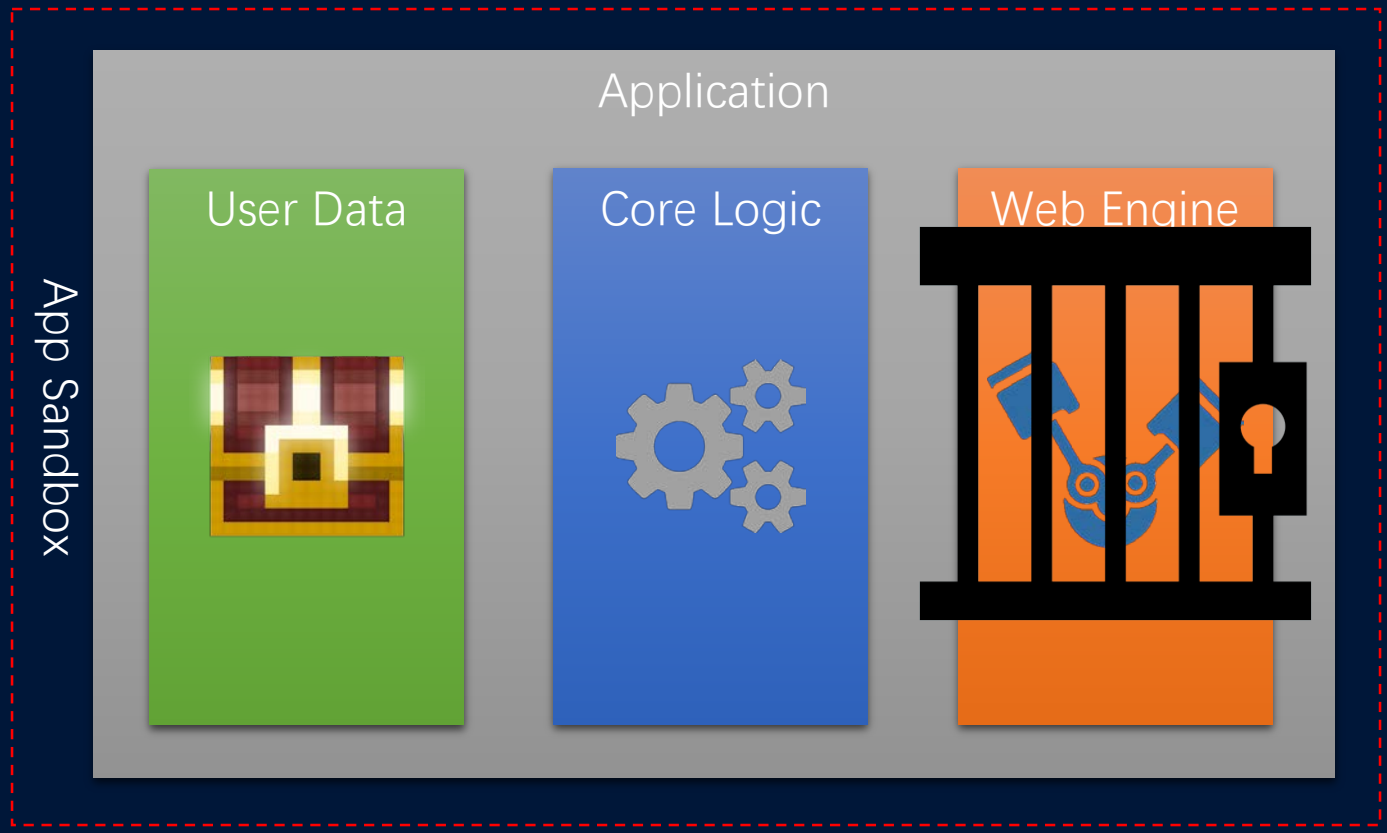
Damage Control

- How user data got comprised

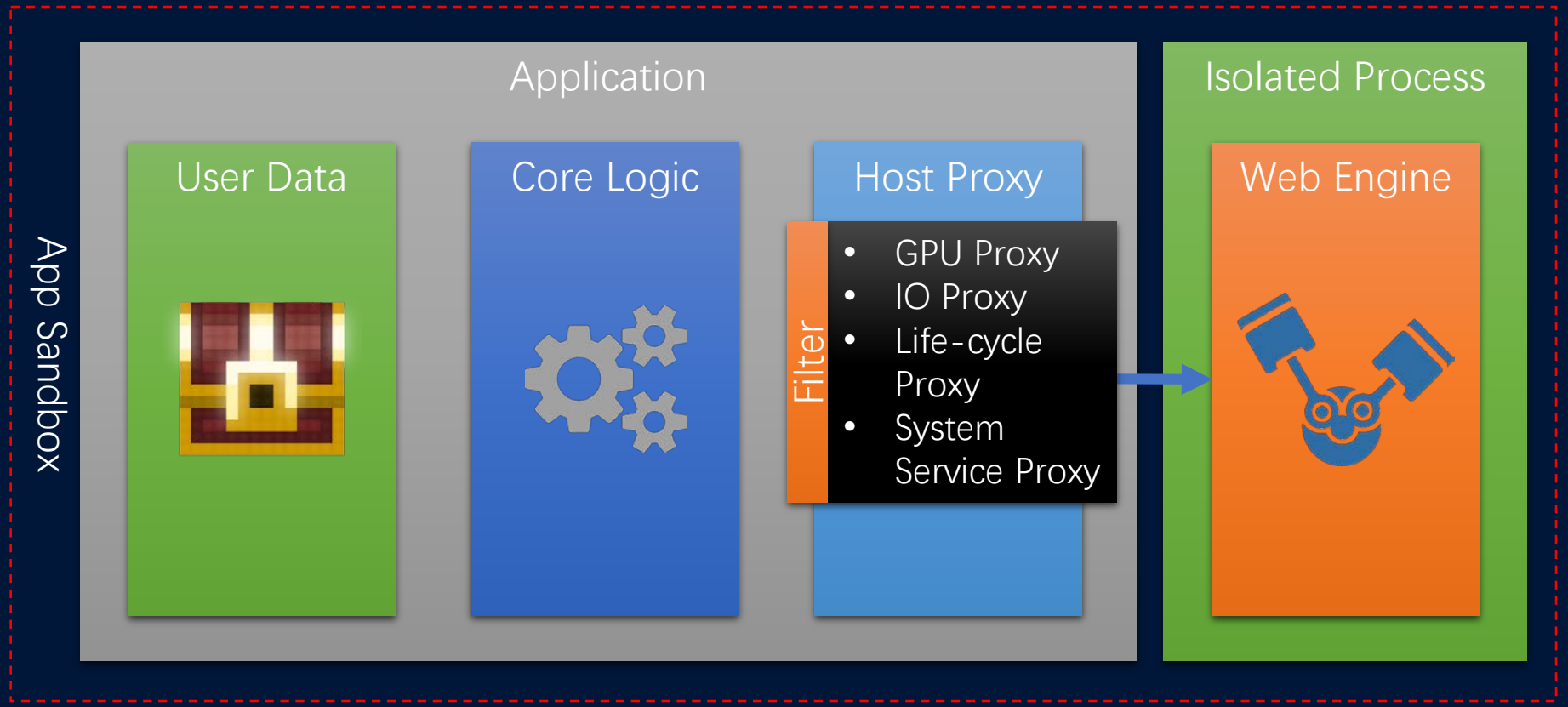


Damage Control

- A “jail” for the troublemakers



Damage Control



Prevention

- We scan the apk file
  - As all the other solutions
- Current challenges
  - Coverage: non-trivial bugs
  - Accuracy: high FP rate
- Long report ≈ useless report
  - Manual work vs. automation
- Goal: Pin-point accuracy
  - While maintaining automation

ApkPecker

- To achieve an accurate report
  - Optimize control flow, data flow and taint analysis
- Control flow
  - Dead code elimination
  - Keep irrelevant code from actual analysis
- Data flow
  - Contextual information on data
  - Eliminate false positive
- Taint checking
  - Complete code path from attack surface to vulnerability
  - Speed up manual work

Controllable WebView Callback Parameters (2)

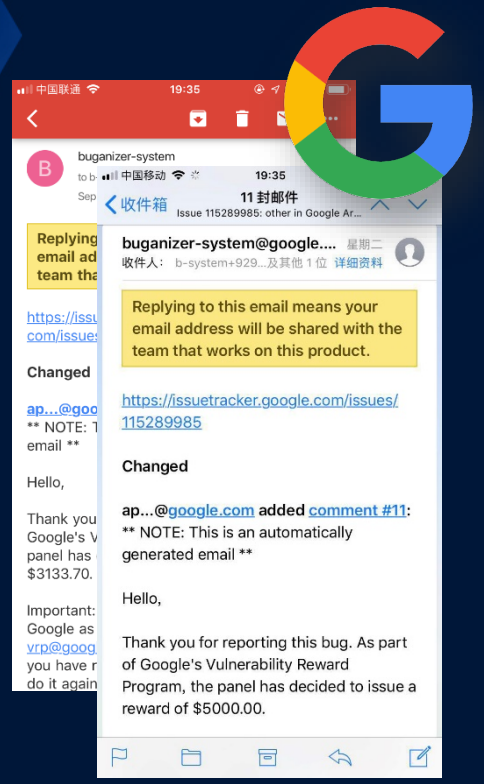
Description : The attacker can perform sensitive operations if the developers doesn't verify the parameters of interfaces provided by WebViewClient.

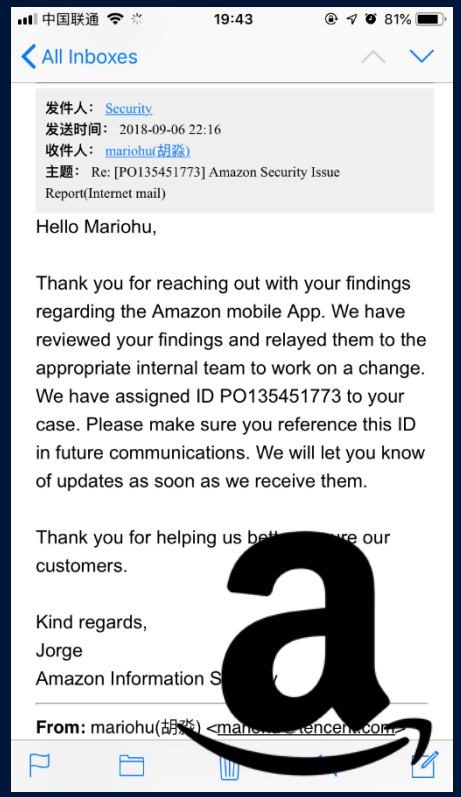
Suggestion : Check the parameters of interfaces provided by WebViewClient strictly.

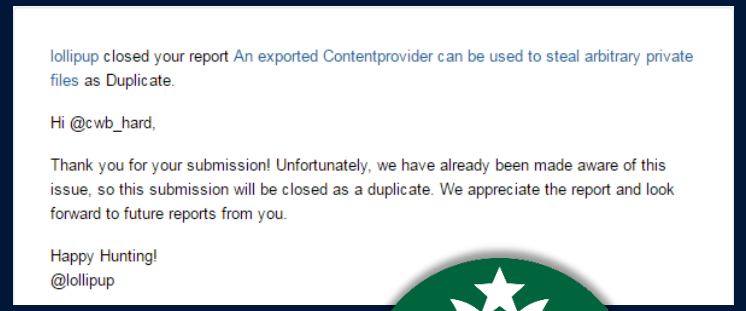
Code Path :

```
<com.tencent.keen.benchmark.Communication.WebViewActivity$1: boolean shouldOverrideUrlLoading(android.webkit.WebView,java.lang.String)>
    -- $r2 := @parameter1: java.lang.String
    -- $r3 = staticinvoke <android.content.Intent: android.content.Intent parseUri(java.lang.String,int)>($r2, 1)

-----------------------------------------------------------------------------------------------------------------------------------------------

<com.tencent.keen.benchmark.Communication.WebViewActivity$1: boolean
shouldOverrideUrlLoading(android.webkit.WebView,android.webkit.WebResourceRequest)>
    -- $r2 := @parameter1: android.webkit.WebResourceRequest
    -- $r3 = interfaceinvoke $r2.<android.webkit.WebResourceRequest: android.net.Uri getUrl()>()    [ FLOW: $r2 -> $r3 ]
    -- $r4 = virtualinvoke $r3.<android.net.Uri: java.lang.String toString()>()    [ FLOW: $r3 -> $r4 ]
    -- $r5 = staticinvoke <android.content.Intent: android.content.Intent parseUri(java.lang.String,int)>($r4, 1)
```

A Pilot Run


buganizer-system
11 封邮件
buganizer-system@google....
Replying to this email means your email address will be shared with the team that works on this product.
https://issuetracker.google.com/issues/115289985
Changed
ap...@google.com added comment #11:
** NOTE: This is an automatically generated email **
Hello,
Thank you for reporting this bug. As part of Google's Vulnerability Reward Program, the panel has decided to issue a reward of $5000.00.


All Inboxes
发件人： Security
发送时间： 2018-09-06 22:16
收件人： mariohu(胡淼)
主题： Re: [PO135451773] Amazon Security Issue Report(Internet mail)
Hello Mariohu,
Thank you for reaching out with your findings regarding the Amazon mobile App. We have reviewed your findings and relayed them to the appropriate internal team to work on a change. We have assigned ID PO135451773 to your case. Please make sure you reference this ID in future communications. We will let you know of updates as soon as we receive them.
Thank you for helping us be
re our customers.
Kind regards,
Jorge
Amazon Information S
From: mariohu(胡淼)


lollipup closed your report An exported Contentprovider can be used to steal arbitrary private files as Duplicate.
Hi @cwb_hard,
Thank you for your submission! Unfortunately, we have already been made aware of this issue, so this submission will be closed as a duplicate. We appreciate the report and look forward to future reports from you.
Happy Hunting!
@lollipup

Beyond Android

- More than a “big tablet”
  - T-Box
  - Gateway
  - ECUs
- And don’t forget the bootloader(s) :)
- Far more systems than IVI’s Android
- Chance to be part of the attack chain
- In other context we usually call these embedded gadgets
  - “IoT”

Challenges

- Unpacking automation
  - Premise to further analysis
  - Countless of file formats
  - Inconsistent/incomplete firmware files
- Vulnerability detecting
  - Known bug detection
  - Typical bugs in proprietary components
- Scaling
  - Routinely scan on release/OTA, or even CI
  - Private or even public online service

Unpacking: YAFFS

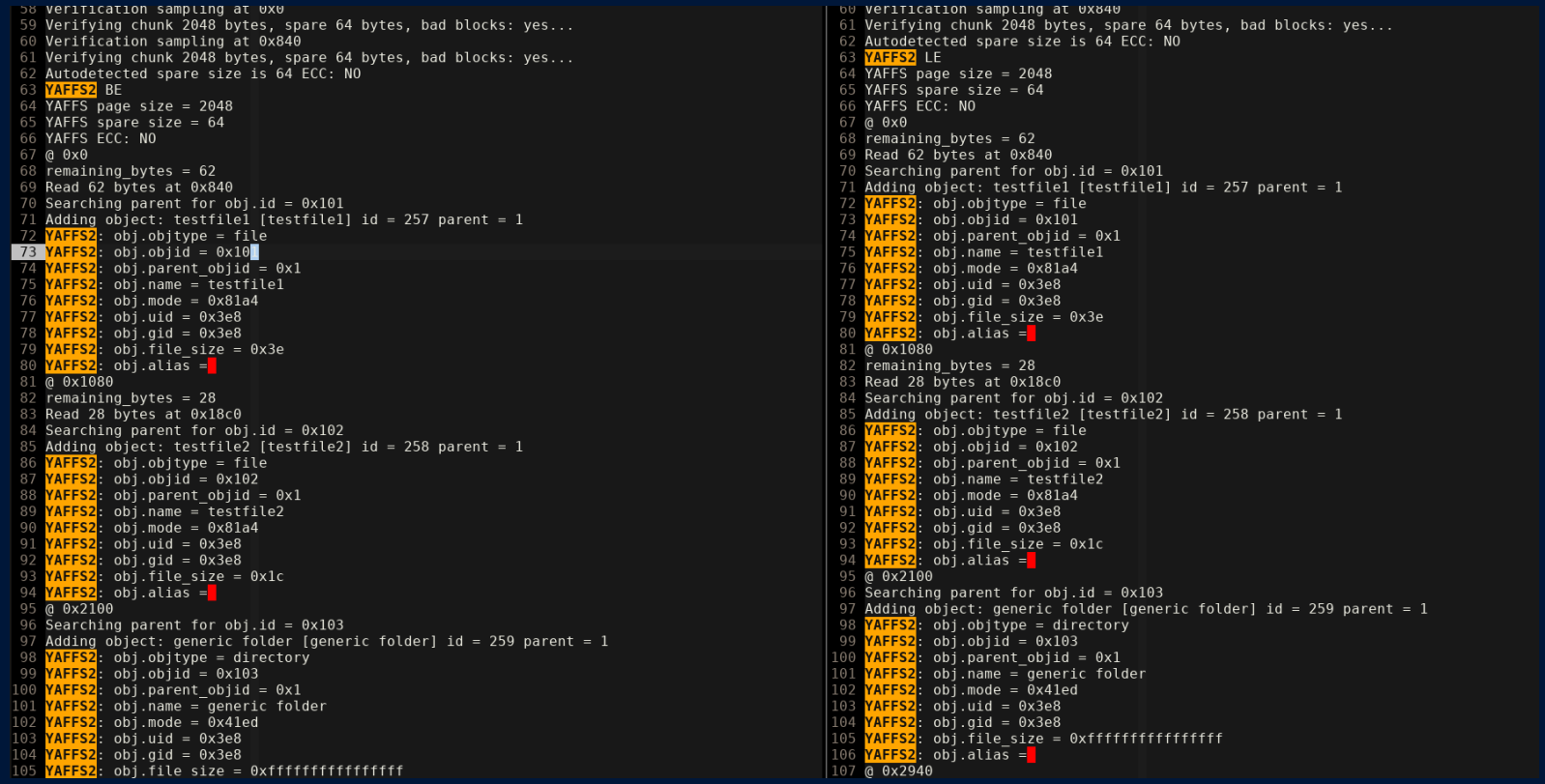
- YAFFS：Non-standard and many parameters
  - Involves manual determination

```
[akochkov@ceph-osd1 unyaffs]$ ./unyaffs -v ../iotsec/unpacker/tests/yaffs2/yaffs2_le.img
Header check OK, chunk size = 2K, spare size = 64, bad block info.
-rw-r--r--       62 2017-01-13 17:18 testfile1
-rw-r--r--       28 2017-01-13 17:18 testfile2
drwxr-xr-x        0 2017-01-13 17:18 generic folder
-rw-r--r--       20 2017-01-13 17:18 generic folder/test file 3_.txt
[akochkov@ceph-osd1 unyaffs]$ ./unyaffs -v ../iotsec/unpacker/tests/yaffs2/yaffs2_be.img
Can't determine flash layout, perhaps not a yaffs2 image
[akochkov@ceph-osd1 unyaffs]$
```

```
[akochkov@ceph-osd1 unpacker]$ binwalk tests/yaffs2/yaffs2_
yaffs2_be.img  yaffs2_le.img
[akochkov@ceph-osd1 unpacker]$ binwalk tests/yaffs2/yaffs2_be.img

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------

[akochkov@ceph-osd1 unpacker]$ binwalk tests/yaffs2/yaffs2_le.img

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------

[akochkov@ceph-osd1 unpacker]$
```

Smart Unpacking

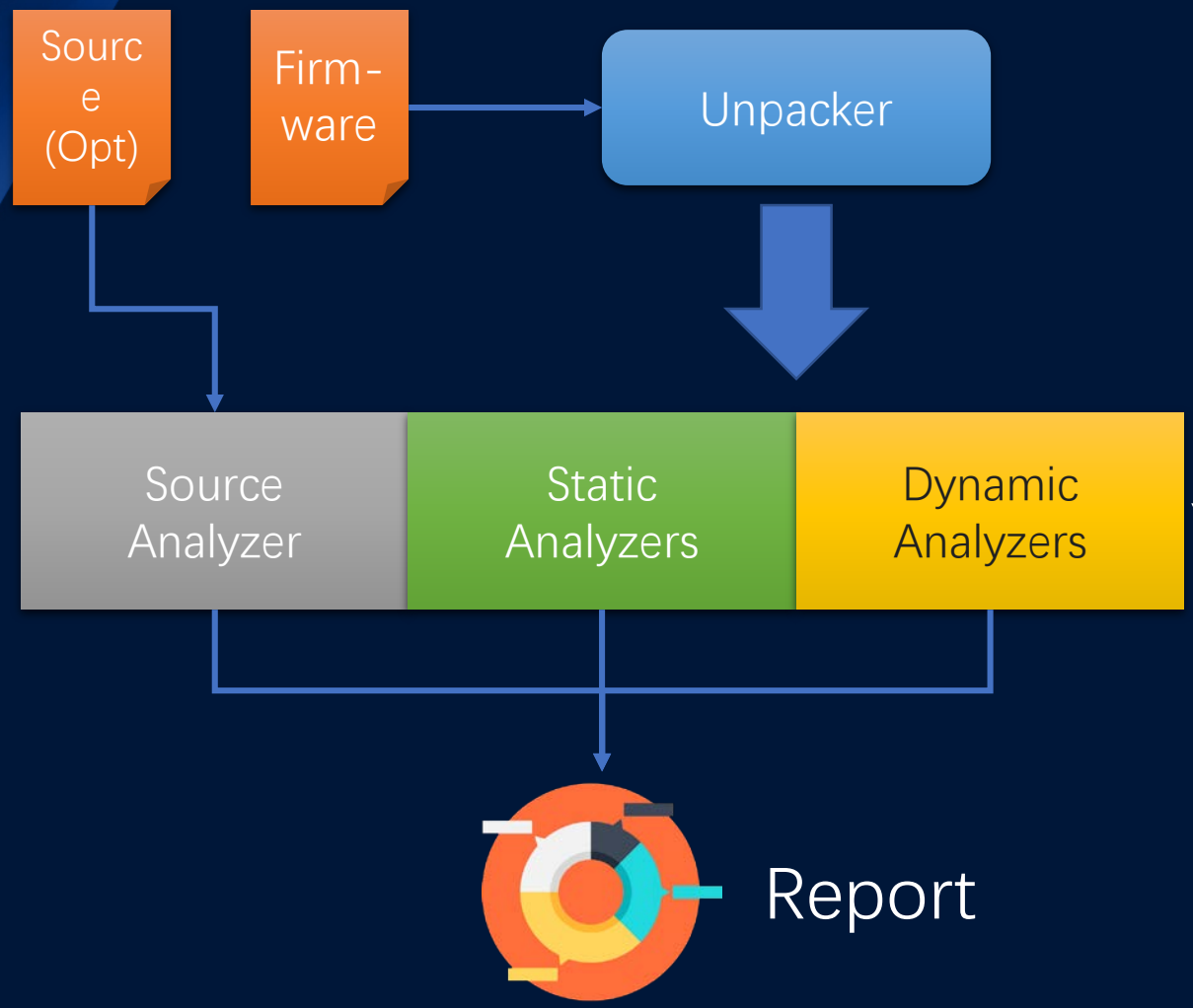
- Extensive use of OCAML for smart and faster code

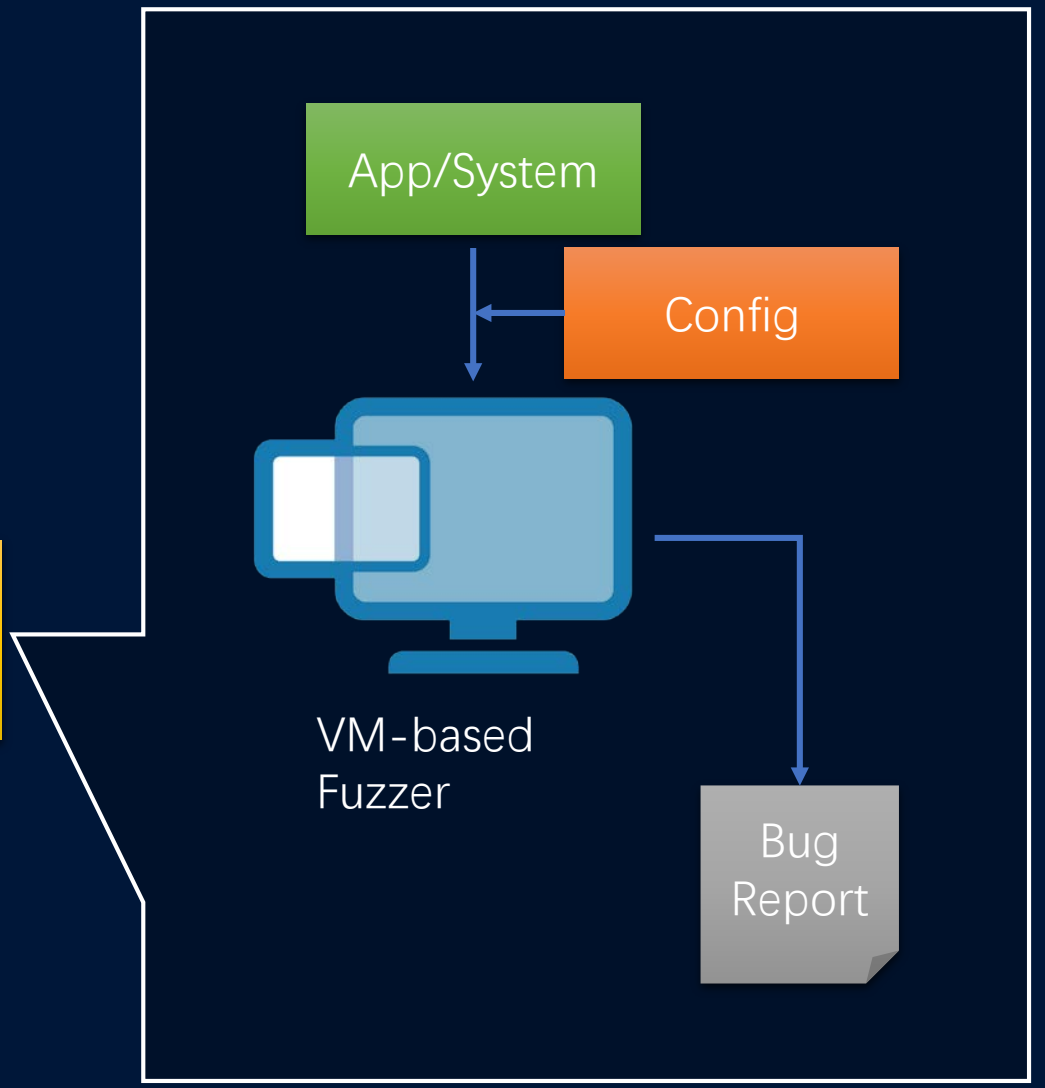


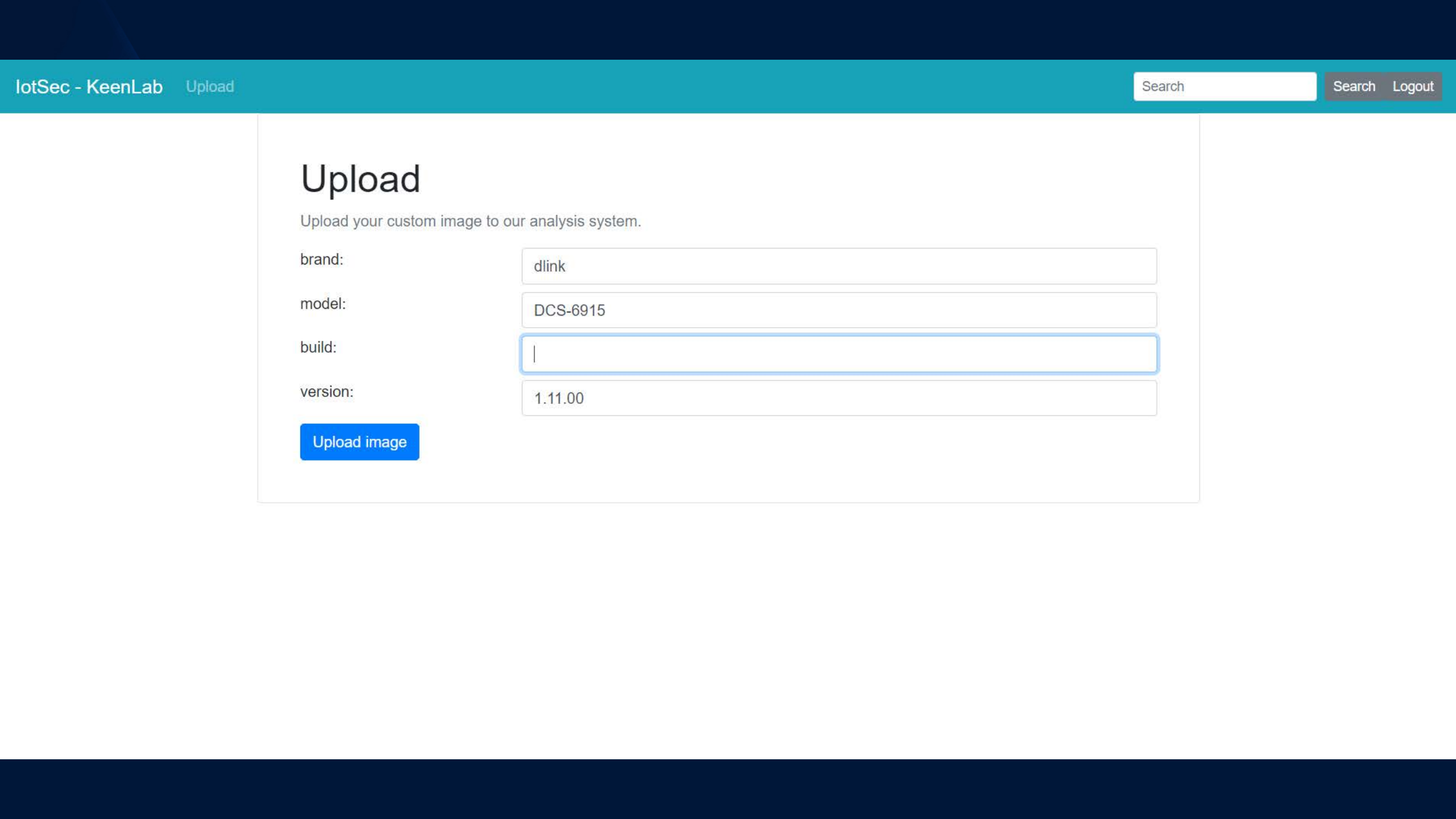
Unpacker Efficiency

FS/Image format	Binwalk	Our unpacker
YAFFS1/YAFFS2	Partial	Supported
Microtik	Partial	Supported
Tenvis	Partial	Supported
UPG images	Not supported	Supported
Android sparse image	Not supported	Supported
Dlink HAD images	Not supported	Supported

Analyzers


Sourc
e
(Opt)
Firm-
ware
Unpacker
Source
Analyzer
Static
Analyzers
Dynamic
Analyzers
Report


App/System
Config
VM-based
Fuzzer
Bug
Report


IotSec - KeenLab
Upload
Search
Search
Logout
Upload
Upload your custom image to our analysis system.
brand:
dlink
model:
DCS-6915
build:
version:
1.11.00
Upload image

IotSec - KeenLab Upload

Search Search Logout

Filter Type to Search Clear Sort -- none -- Asc

Sort direction Asc Per page 5

Brand	Model	Version	Status	Actions
[redacted]		1.10	finish	Info modal Show Details
[redacted]		1.0	finish	Info modal Hide Details

Basic CVEs Details

Optimization-proof and cross architecture detection

CVE	CVSS	Description
CVE-2017-7187	7.2	The sg_ioctl function in drivers/scsi/sg.c in the Linux kernel through 4.10.4 allows local users to cause a denial of service (stack-based buffer overflow) or possibly have unspecified other impact via a large command size in an SG_NEXT_CMD_LEN ioctl call, leading to out-of-bounds write access in the sg_write function.
CVE-2017-16939	7.2	The XFRM dump policy implementation in net/xfrm/xfrm_user.c in the Linux kernel before 4.13.11 allows local users to gain privileges or cause a denial of service (use-after-free) via a crafted SO_RCVBUF setsockopt system call in conjunction with XFRM_MSG_GETPOLICY Netlink messages.
CVE-2017-7618	7.8	crypto/ahash.c in the Linux kernel through 4.10.9 allows attackers to cause a denial of service (API operation calling its own callback, and infinite recursion) by triggering EBUSY on a full queue.
CVE-2016-2847	4.9	fs/pipe.c in the Linux kernel before 4.5 does not limit the amount of unread data in pipes, which allows local users to cause a denial of service (memory consumption) by creating many pipes with non-default sizes.
CVE-2017-7533	6.9	Race condition in the fsnotify implementation in the Linux kernel through 4.12.4 allows local users to gain privileges or cause a denial of service (memory corruption) via a crafted application that leverages simultaneous execution of the inotify_handle_event and vfs_rename functions.
CVE-2014-8172	4.9	The filesystem implementation in the Linux kernel before 3.13 performs certain operations on lists of files with an inappropriate locking approach, which allows local users to cause a denial of service (soft lockup or system crash) via unspecified use of Asynchronous I/O (AIO) operations.
CVE-2015-8839	1.9	Multiple race conditions in the ext4 filesystem implementation in the Linux kernel before 4.5 allow local users to cause a denial of service (disk corruption) by writing to a page that is associated with a different user's file after unsynchronized hole punching and page-fault handling.

IotSec - KeenLab Upload

Search Search Logout

Filter Type to Search Clear Sort -- none -- Asc

Sort direction Asc Per page 5

Brand	Model	Version	Status	Actions
[redacted]		1.10	finish	Info modal Show Details
[redacted]		1.0	finish	Info modal Show Details
[redacted]		v5.0.23	finish	Info modal Show Details
[redacted]		4.16	finish	Info modal Show Details
[redacted]			finish	Info modal Hide Details

Basic CVEs Details Crypto Keys

Description

Static key detection by pattern as well as back tracked from crypto functions detected

system/app/[redacted]: -----BEGIN EC PRIVATE KEY----- [redacted] hmTxZ3/mZI3vyk7p3U3wBf+WIop6hDhkFzJhmLcqoAoGCCqGSM49
AwEHoUQDQgAEV+WusXPf06y7k7iB/xKu7uZTrM5VU/Y0Dswu42MlC9+Y[redacted]o+HcKPkxw== -----END EC PRIVATE KEY

system/app/[redacted]: -----BEGIN EC PRIVATE KEY----- [redacted]QEEIPEqEyB2AnCoPL/9U/YDHvdqXYbIogTywwyp6/UfDw6noAoGCCqGSM49
AwEHoUQDQgAEN8xW2XYJHlpyPsdZLf8gbu58+QaRdNCtFLX3aCJZYpJO5QDYIxH/ 6i/SNF1dFr2KiMJrdw1VzYoqDvoByLTt/w== -----END EC PRIVATE KEY

Conclusion

- Android eco-system is much more than phones
- Connected cars are one of the most complex devices
  - A “hub” in the whole system
- Takes more than system hardening
- Keen Lab solutions
  - ApkPecker
  - IoTScanner
- Availability
  - Now internally
  - Est. end of year publicly

THANKS !