Why This Talk Exits and Thanks RD

This Talk Is Part of 2nd Nov, Fuzzing Talk
About NGUYEN Anh Quynh

- Nanyang Technological University, Singapore
- PhD in Computer Science
- Operating System, Virtual Machine, Binary analysis, etc
- Usenix, ACM, IEEE, LNCS, etc
- Blackhat USA/EU/Asia, DEFCON, Recon, HackInTheBox, Syscan, etc
- Capstone disassembler: http://capstone-engine.org
- Unicorn emulator: http://unicorn-engine.org
- Keystone assembler: http://keystone-engine.org
About KaiJern

The Shepherd Lab
Stay in the office 24/7 by hoping making the world a better place

- IoT Research
- Blockchain Research
- Fun Security Research

Badges Maker
Founder of hackersbadge.com, RE && CTF fan

- Reversing Binary
- Reversing IoT Devices
- Part Time CTF player

HITB Security Conference
Hack in the box, Netherland and Singapore. Soon to be Beijing and Dubai

- 2006 till end of time
- Core Crew
- Review Board

- 2005, HITB CTF, Malaysia, First Place /w 20+ Intl. Team
- 2010, Hack In The Box, Malaysia, Speaker
- 2012, Codegate, Korean, Speaker
- 2015, VXRL, Hong Kong, Speaker
- 2015, HITCON Pre Qual, Taiwan, Top 10 /w 4K+ Intl. Team
- 2016, Codegate PreQual, Korean, Top 5 /w 3K+ Intl. Team
- 2016, Qcon, Beijing, Speaker
- 2016, Kcon, Beijing, Speaker
- 2016, Intl. Antivirus Conference, Tianjin, Speaker

- 2017, Kcon, Beijing, Trainer
- 2017, DC852, Hong Kong, Speaker
- 2018, KCON, Beijing, Trainer
- 2018, DC010, Beijing, Speaker
- 2018, Brucon, Brussel, Speaker
- 2018, H2HC, San Paolo, Brazil
- 2018, HITB, Beijing/Dubai, Speaker
- 2018, beVX, Hong Kong, Speaker
- MacOS SMC, Buffer Overflow, suid
- GDB, PE File Parser Buffer Overflow
- Metasploit Module, Snort Back Orffice
- Linux ASLR bypass, Return to EDX
Your Very First IoT Device

Some Said Wi-Fi Router
Why Hacking IoT

Remembering,
smashing the stack for fun and profit
Why IoT Research Is Important

**Firmware Emulation**
- Without built-in shell access for user interaction
- Without development facilities required for building new tools
  - Compiler
  - Debugger
  - Analysis tools

**Skorpio DBI**
- Binary only - without source code
  - Existing guided fuzzers rely on source code available
    - Source code is needed for branch instrumentation to feedback fuzzing progress
    - Emulation such as QEMU mode support in AFL is slow & limited in capability
    - Same issue for other tools based on Dynamic Binary Instrumentation

**Guided Fuzzer for Embedded**
- Most fuzzers are built for X86 only
  - Embedded systems based on Arm, Arm64, Mips, PPC
- Existing DBIs are poor for non-X86 CPU
  - Pin: Intel only
  - DynamoRio: experimental support for Arm
Back to School Edition: DEFINATION of IoT
Definition of IoT – From The Book

Any Online-able THINGS
The Real Definition of IoT

Human Operated + Online-able Item + AI Capability

* Data Mining(maybe) Business *
Attack Surface
Attacker Perspective

- Vendor Data Center Security
- Communication Protocol

- Server OS Security
- Application Security

Again, Why?
Is To Discover The Truth

- Data Transmission Hijack
- Sniffing

- Household Security
- Device Password
Everything is small, Including SECURITY

- System on Chip
- A chip with all the PCI-e slot and card in it
- Pinout to different parts
- WiFi, Lan, Bluetooth and etc
- Low power device
- Strip Down Power Usage
- Strip Down Size
- Strip Down Processing
- Strip Down SECURITY
Understanding The Board
Requirement: Software

Skill @ GNU Command Set
Let's Get Started
### Device Limited Bug

#### In The Beginning:

We Need Firmware

<table>
<thead>
<tr>
<th>CVE ID</th>
<th>CVSS Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2012-5582</td>
<td>7.5</td>
<td>Exec Code Overflow Bypass 2017-03-26 2017-07-17</td>
</tr>
<tr>
<td>CVE-2012-5384</td>
<td>7.5</td>
<td>Exec Code Overflow Bypass 2017-03-26 2017-07-17</td>
</tr>
<tr>
<td>CVE-2012-5393</td>
<td>7.5</td>
<td>Exec Code Overflow Bypass 2017-03-26 2017-07-17</td>
</tr>
</tbody>
</table>

Netgear: Security Vulnerabilities

CVE ID: CVE-2012-5582

**Description:** Exec Code Overflow Bypass

**CVE ID:** CVE-2012-5582

**Score:** 7.5

**Date:** 2017-03-26 2017-07-17

**Details:**

- **Exploit:** None
- **Access Level:** Remote
- **Complexity:** Low
- **Authentication:** Not required
- **Conf.:** Partial
- **Integ.:** Partial
- **Avail.:** Partial

Cross-site request forgery (CSRF) vulnerability in Netgear DGND2000 routers with firmware 10.0.0.20 through 10.0.5.0 allows remote attackers to hijack the authentication of users for requests that perform full lookups via the host_name parameter to doxilksup.cgi. Note: This issue can be combined with CVE-2017-6334 to execute arbitrary code remotely.

- **CVE ID:** CVE-2017-6334
- **Score:** 7.5
- **Date:** 2017-03-09 2017-06-30
- **Details:**
  - **Exploit:** None
  - **Access Level:** Remote
  - **Complexity:** Low
  - **Authentication:** Not required
  - **Conf.:** Partial
  - **Integ.:** Partial
  - **Avail.:** Partial

If all _model_ = one_firmware

In The Beginning: We Need Firmware
Getting Firmware
Firmware and Hardware

Extract From APK, Traffic Sniffing or Just Download
1. Download
2. Patch with Backdoor
3. Flash
4. pwned

If we need more?
1. RCE
2. Study the firmware
We learn from the hard way (aka story time)
The Easy Way
Complete Kit to Success

MIPS
Interchangeable Base Board

ARM

AARCH64

If There are only 3 platform, Download, Flash, Reverse and pwn !!!
If *ARM/AARCH64* Why Not Raspberry PI
LIBC Compatibility

MIPS
Not Supported by Raspberry PI

ARM

AARCH64

Raspberry PI Is not *reverser* Friendly
So, QEMU is a MUST
Assembly Instruction Compatibility

ARM

AARCH64
Current Work Around
QEMU-Static is good for binary execution without additional software or hardware interaction.
Current Primitive Firmware Emulation

Leaving squashfs and going into a unknown world

Its not easy after 2016
Why Firmware Emulation
More Resources = More Power

**Processor**
Normally 1-2 Core

**RAM**
Normally 256MB/512MB

**FLASH**
Normally 8MB/16MB/32MB/256MB

Most Important, we got apt-get
Objectives
Only One Process with Interaction

Hunt for the one that spawn listener port

most of the devices comes with one big binary
Booting Up
Distro and Kernel Mix and Match

**script to boot arm**

```bash
#!/bin/bash
sudo tunctl -d tap0
sudo screen -d /opt/qemu/bin/qemu-system-arm -n 2048 -M virt -cpu cortex-a15 -smp cpus=4,mmap=4,maxcpus=4 -kernel boot.stretch.arenf.vt/vmlinuz-4.9.0-6-ammp-lpae -initrd boot.stretch.arenf.vt/vmlinuz-4.9.0-6-ammp-lpae -append "console=ttyS0,115200n8,8 noresume quietboot initrd=initrd.4.9.0-6-ammp-lpae.d" -drive file=debian-stretch.arenf.vt/boot/stretch-arenf-vt/ramdisk,if=none,format=qcow2,io=none,cache=none,media=cdrom,LOAD=FORCE-boot=linux root=LABEL=initrd疳=LABEL=debian-stretch.arenf.vt boot=live net=nic -net tap,ifname=tap0,script=no,downscript=no -net nic -net tap,ifname=te0,script=no,downscript=no -net graphic
sudo sysct1 -w net.ipv4.ip_forward=1
```

**script to boot mips**

```bash
#!/bin/bash
sudo screen -d /opt/qemu/bin/qemu-system-mipsel -n 512 -M malta -kernel boot.stretch.mipsel/vmlinuz-4.9.0-4-kc-malta -initrd boot.stretch.mipsel/initrd.4.9.0-4-kc-malta/boot=linux root=LABEL=malta0 net=nic -net tap,ifname=tap0,script=no,downscript=no -net nic -net tap,ifname=te0,script=no,downscript=no -net graphic
sudo tunctl -d tap0
```

**argument:** running new or old distro + kernel
chroot
chroot is easy (still hardware dependent), but we will have issue with tools

Running without chroot
Classic Case: File Not Found
The File Missing Trick

We Missed You

```bash
chdir("/")
execve("/bin/bash", ["/bin/bash", "-i"], 0xffffca14f650 /* 18 vars */) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/aarch64-linux-gnu/charset.alias", O_RDONLY|O_NOFOLLOW) = -1 ENOENT (No such file or directory)
write(2, "chroot: ", 8chroot: ) = 8
write(2, "failed to run command '/bin/bash'", 33failed to run command '/bin/bash') = 33
write(2, ": No such file or directory", 27: No such file or directory) = 27
write(2, "\n", 1 ) = 1
close(1) = 0
close(2) = 0
exit_group(127) = ?
```

We found you

```
root@rpi3:/opt/
~/lib64# file ./bin/bash
./bin/bash: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-aarch64.so.1 for GNU/Linux 3.14.0, BuildID[sha1]=22e2854c58b1814825b95c9a1036c658a371f5b0, stripped
```
The missing .SO and binary Issue
Out from chroot, we need feeding

Feeding all the required so and binary with “ln –s”
Out from chroot, we need feeding

bash-3.2# /usr/bin/appmainprog
<appmain>******************************************************************************
<appmain>child process id is 3931
<appmain>Appcliation Init Begin
<appmain>Audio Mas process Init
[Aud][PPC] AudioPPControl constructor
[Aud][PPC] AudioPPControl getInstance
[Aud][PPC] AudioPPControl freeInstance
[Aud][PPC] AudioPPControl destructor
[Aud][PPC]AudioPPControl deinit begin.
[Aud][PPC][ppcStructUnalloc] ppc_destroy_info begin.
Segmentation fault
bash-3.2#

“segfault” without clear error. strace come to rescue
The Secretive NVRAM
Dark side of NVRAM

Relationship between main binary is so intimate, but in actual fact, it's just a hit and run.
Dark Side of NVRAM

Relationship between main binary is so intimate, but in actual fact, it's just a hit and run.

Dark Side of the main process, we ignore and can't move to the next step.
A fake NVRAM

IF interactor is the medium, can we fake it?

ask for nvram info

reply with nvram info
A fake NVRAM

IF interactor is the medium, can we fake it?

ask for nvram info

reply with nvram info

interactor

Custom Interactor
The bridge trick

The switch looking device
Wireless Device
Faking wpa_supplicant

making eth0 looks like wlan0 works too
Everything Things Else Fail
BL, BNE, BEQ and friends

Original BIN

Patched BIN

Argument: To Patch or To Fulfill Firmware Needs
DEMO *bug disclosed in geekpwn 2018, shanghai*
Web Cam Buffer Overflow

Pre Authentication Bug

Buffer Overflow

Address Overwritten

Debug is almost Impossible *watchdog*

Emulation comes into play
IoT with UDP Access

Web Cam with Motor
Command Execution Injection

Chinese based WiFi Router
Questions

Multi ARCH Firmware Emulation