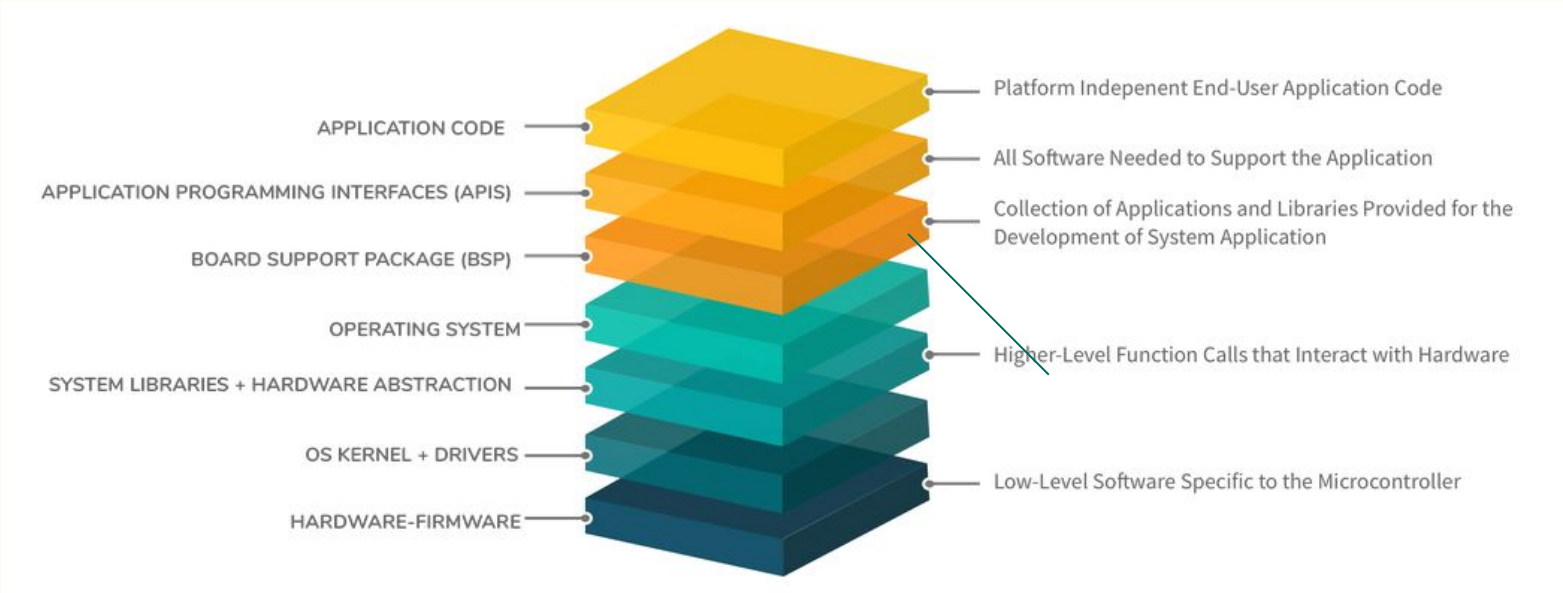
Finding CVEs through Firmware Emulation

—

Firmware

- Firmware is a type of software that is embedded in electronic devices and provides low-level control for the hardware.
- It serves as a bridge between the hardware and higher-level software, enabling the device to perform specific functions.
- Unlike traditional software, firmware is typically stored in non-volatile memory and remains persistent even when the device is powered off.

Software Stack



Challenges of Firmware Testing on Bare Metal

- **Cost and Availability:** Physical hardware can be expensive, especially when considering the need for multiple devices for thorough testing. Additionally, obtaining and maintaining a diverse range of hardware configurations can be a financial challenge.
- **Limited Resources:** Physical hardware testing often requires dedicated resources, such as space, power, and cooling which can be impractical.
- **Time-Consuming:** Deploying and configuring physical hardware for testing can be time-consuming for each test iteration.
- **Accessibility:** Physical hardware may not be easily accessible, especially if it is located in remote or secure environments. This can impede the ability to perform tests and gather data promptly.
- **Scalability:** Testing at scale becomes challenging.
- **Reproducibility:** Achieving consistent and reproducible test results on physical hardware can be difficult due to variations in hardware components, environmental conditions, and other factors.

Pros of Firmware Emulation for Security Analysis

- **Reproducibility:** Emulation provides a consistent and reproducible testing environment. Analysts can recreate specific scenarios easily, aiding in the identification and verification of vulnerabilities and exploits.
- **Diverse Hardware and Architectures:** Security researchers often encounter a variety of hardware architectures in embedded systems. Firmware emulation allows the testing of security measures across different architectures without the need for specialized physical devices for each target.
- **Automation and Scaling:** Emulation tools often provide scripting and automation capabilities, allowing analysts to script complex test scenarios and automate repetitive tasks. This facilitates efficient scaling for systematic security analysis, especially when dealing with large datasets, multiple firmware versions or fuzzing.

Security-oriented Frameworks for Emulation

- **QEMU:** QEMU, in combination with the GDB allows for dynamic analysis, debugging, and reverse engineering of firmware for various architectures. Many firmware emulation frameworks are based on QEMU.
- **FIRMADYNE:** FIRMADYNE is an open-source framework designed for emulating and analyzing embedded firmware. It leverages QEMU to emulate different architectures commonly found in embedded devices.
- **FIRMAE:** An emulation framework similar to Firmadyne.
- **FRANKENSTEIN**: FRANKENSTEIN provides a virtual environment to fuzz wireless firmwares. Firmwares can be hooked during runtime to extract their current state. Then, they can be re-executed in a virtual environment for fuzzing. Firmware images are reassembled to be executed with QEMU.

Proposed Analysis Model

- Select three state-of-art emulation frameworks for analysis
  - Frankenstein
  - FirmAE
  - Firmadyne
- Select a IoT pentesting tools that includes exploits for IoT devices (mainly routers/bluetooth for our study)
  - Routersploit
- Select images based on the exploits available in RouterSploit for various vendors like ASUS, Netgear, D-Link.
- Emulate the firmware images on both the frameworks.

Comparison

- Using the analysis method described, we will compare both analysis framework based on:
  - Ease of emulation (no crashes, boot-loops etc).
  - Vendor images supported.
  - Architecture supported (MIPS, ARM).
  - Are the CVEs reproducible on firmware binaries?

FRANKENSTEIN

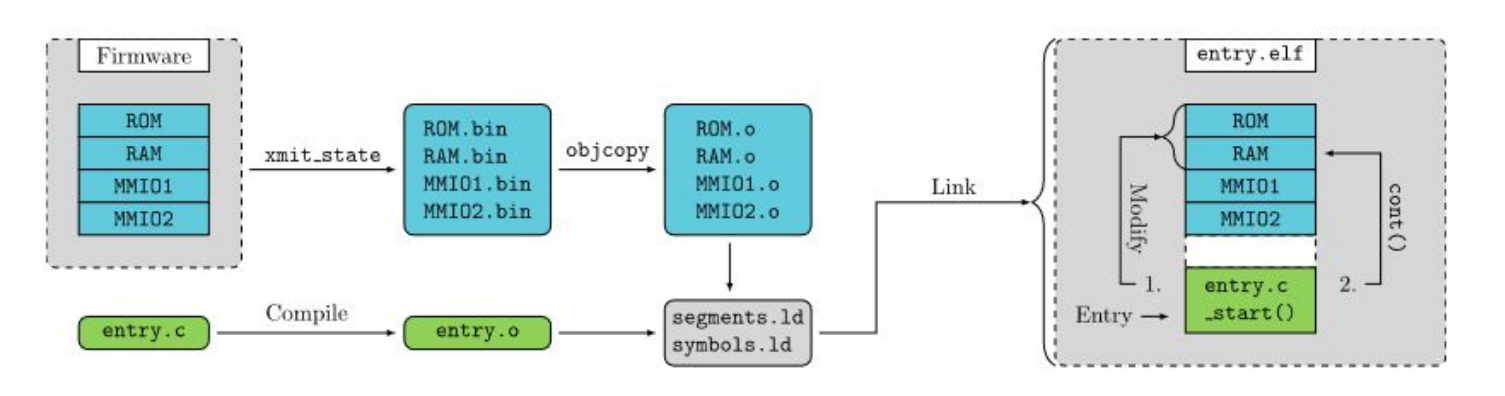
- Dynamic analysis framework for firmware which leverages QEMU emulation.
- Target: Bluetooth Stack on Cypress and Broadcom firmware.
- Method: Fuzzing.
- Targeted Bugs: RCEs (Remote Code Execution).
- Why?
  - Generic over-the-air fuzzing suffers from
    - constrained speed,
    - limited repeatability, and
    - restricted ability to debug

Methodology

- Create a physical device snapshot and then emulates it in QEMU to fuzz the full stack.
- Over-the-air data is provided by a virtual modem.
- Emulated firmware implements thread and task switches to fuzz multiple handlers.
- Attaches to a real Linux host.

Methodology

- Basically, use C hooks within the firmware to attack to the real Linux stack.
- For example, BlueZ stack on Linux for Bluetooth.
- Thus, Frankenstein triggers realistic full-stack behavior.

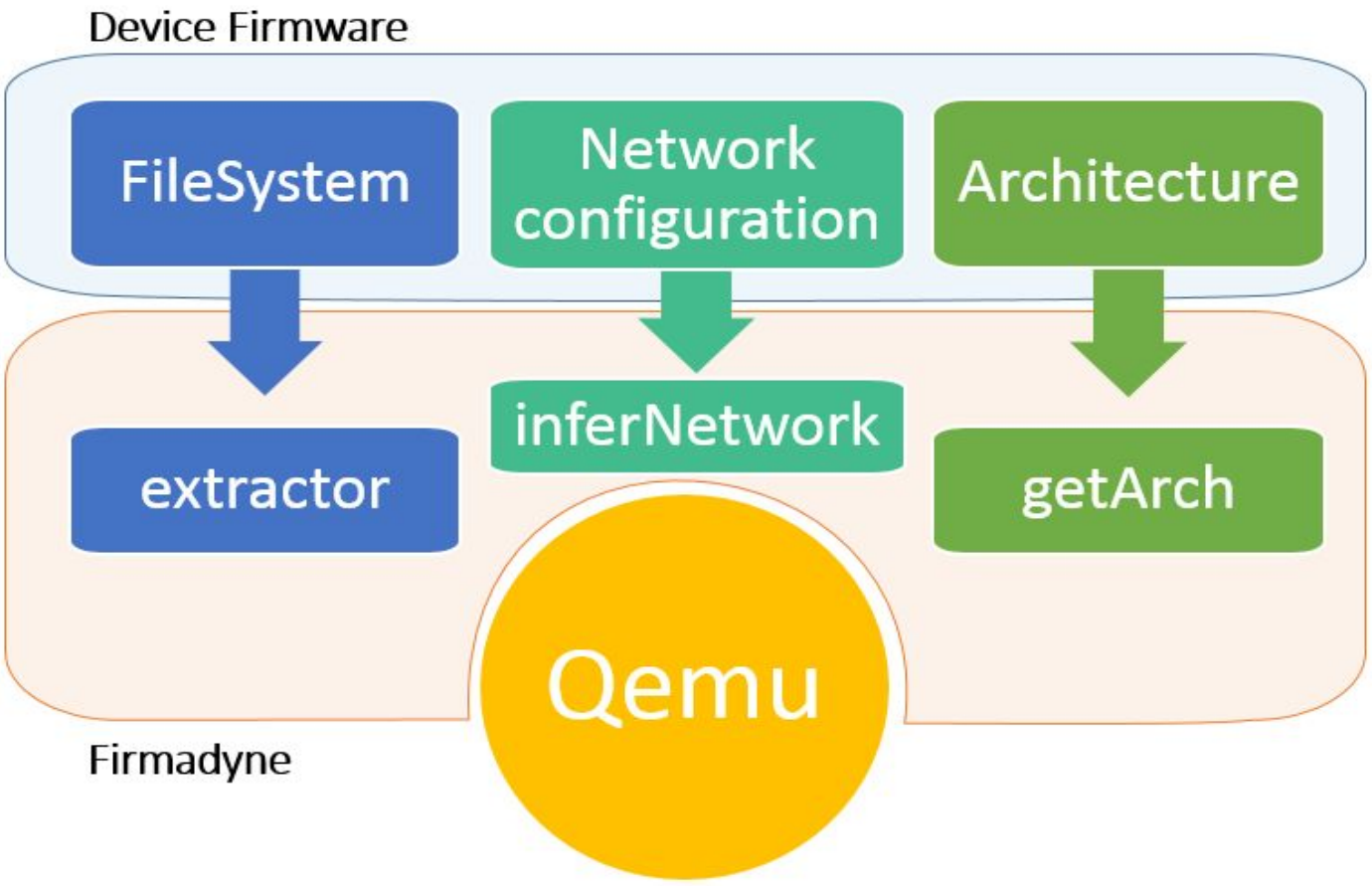


Implementation Issues with Frankenstein

- While we were able to set up Firmadyne with some difficulty (and got our pull request for a setup script accepted to the official repo), we found reproduction of CVEs through Frankenstein difficult.
- Compiling a Firmware snapshot to QEMU-runnable ELF file was harder than it seemed.
- Even with pre-compiled images, attaching it a Linux host seemed impossible as there were a lot of bugs to fix.
- Our inexperience with firmware emulation in general was a major roadblock, so we moved on to different frameworks.

Firmadyne

- FIRMADYNE is an automated and scalable system for performing emulation and dynamic analysis of Linux-based embedded firmware.
- Why is it useful?
  - Vulnerability Check: Firmadyne comes with a script that tests for the presence of 60 known vulnerabilities using exploits from Metasploit. In addition, it also checks for 14 previously-unknown vulnerabilities.
  - Using these exploits from Metasploit, and the 14 previously-unknown vulnerabilities, the researchers showed that 846 out of 1,971 (43%) firmware images were vulnerable to at least one exploit, which they estimated to affect 89+ different products.


Device Firmware
FileSystem
Network configuration
Architecture
extractor
inferNetwork
getArch
Qemu
Firmadyne

FirmAE

- FirmAE is an open-source framework designed for emulating and analyzing firmware of embedded devices.
- What makes it different?
  - By applying simple heuristics, emulation failure cases can be resolved even if they originate from different root causes
  - These arbitration techniques bypass the failure cases.
  - Thus, this approach can emulate numerous firmware images that previous approaches failed to emulate, and effectively aid in finding real vulnerabilities.

Analysis Set

- Selected FW images based on vendors(10 each)
- With most of the images, the behaviour we noticed loads of crash, FW image stuck in boot loop
- This caused a lot of issues doing analysis. The results shows only emulation success (not CVE reproducibility)
- Images were selected based on support in RouterSploit

Vendor	Netgear	ASUS	D-Link	Belkin
Firmadyne	10	10	10	10
FirmAE	3	1	2	2

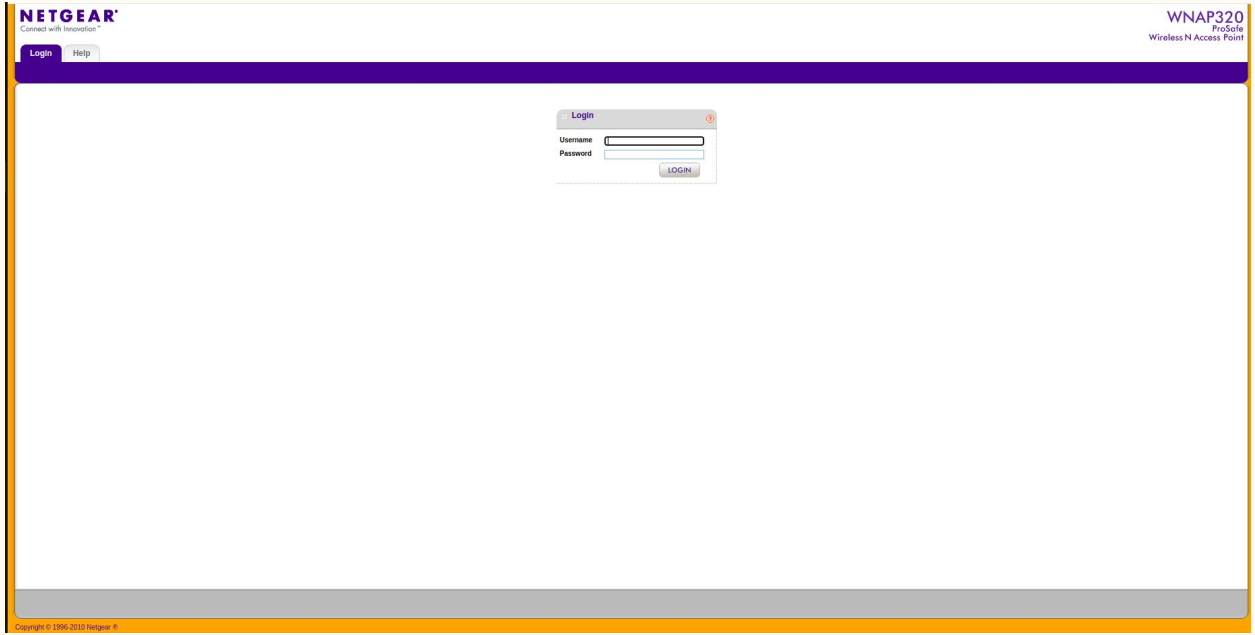
Firmadyne: Emulation

Access to shell!

```
[   22.444000]
[   22.444000] Cpu 0
[   22.444000] $ 0   : 00000000 1000a400 00000004 00000000
[   22.444000] $ 4   : 00000004 00419f18 00000000 00000001
[   22.444000] $ 8   : 2b487004 004470b8 00000031 fffffff0
[   22.444000] $12   : 8f085eb0 00000234 06ca3695 2b43b578
[   22.444000] $16   : 7f9ec2d0 7f9ec160 7f9f6bc4 ffffffff
[   22.444000] $20   : 7f9ec224 00401834 00000001 004019f0
[   22.444000] $24   : 00000002 2b45e7d0
[   22.444000] $28   : 00435880 7f9ebbb8 7f9ebbb8 00416804
[   22.444000] Hi    : 00000005
[   22.444000] Lo    : 19999999
[   22.444000] epc   : 2b45e7d0 0x2b45e7d0
[   22.444000]     Not tainted
[   22.444000] ra    : 00416804 0x416804
[   22.444000] Status: 0000a413    USER EXL IE
[   22.444000] Cause : 10800008
[   22.448000] BadVA : 00000004
[   22.448000] PrId  : 00019300 (MIPS 24Kc)
Segmentation fault
[hostapd_tr]

Starting Translator...      [nmbd_tr]

Starting Translator...      sh: cannot create /proc/sys/net/bridge/bridge-http-redirect-flush-mac: nonexistent directory
sh: cannot create /proc/sys/net/bridge/bridge-http-redirect-enabled: nonexistent directory
[http_redirect_tr]

Starting Translator...      [dhcp]

Starting Translator...      kill: cannot kill pid 601: No such process
[ntp]

Starting Translator...      [timezone]

Starting Translator...      kill: cannot kill pid 614: No such process
[sc_radio]
Error in opening the device.
: No such device

System initilization is ..  [DONE...]


Welcome to SDK.

Have a lot of fun...

netgear123456 login: [   26.704000] brtrunk: port 1(eth0) entering forwarding state
[   31.888000] brtrunk: no IPv6 routers present
[   32.152000] eth0: no IPv6 routers present
admin
Password:
netgear123456#
```

Firmadyne: Emulation

Access to web server!



FirmAE: Emulation

```
$: command not found
gaurav@gaurav-HP-Spectre-x360-Convertible-13-aw0xxx:~/IoTSec/FirmAE$ sudo ./run.sh -a D-Link DIR-868L_fw_revB_2-05b02_eu_multi_20161117.zip
[*] DIR-868L_fw_revB_2-05b02_eu_multi_20161117.zip emulation start!!!
[*] extract done!!!
[*] get architecture done!!!
mke2fs 1.46.5 (30-Dec-2021)
e2fsck 1.46.5 (30-Dec-2021)
[*] infer network start!!!

[IID] 3
[MODE] analyze
[+] Network reachable on 192.168.0.1!
[+] Web service on 192.168.0.1
[*] Waiting web service...
Creating TAP device tap3_0...
Set 'tap3_0' persistent and owned by uid 0
Initializing VLAN...
Bringing up TAP device...
Starting emulation of firmware... 192.168.0.1 true true 19.389573966 45.067761987
[+] start pentest!
[*] FirmAE web server initializer
```

Routersploit 101

- Full arsenal of exploits and default creds hack for various vendor images

```
rsf > use
creds     encoders  exploits  generic   payloads  scanners
rsf > use
```

- Support IoT devices apart from routers like IP camera.

```
rsf > use exploits/
exploits/cameras/  exploits/misc/     exploits/routers/
rsf > use exploits/
```

- Routersploit contains exploits as per each target device.

```
rsf (Belkin Auth Bypass) > show devices

Target devices:
   0 - Belkin Play Max (F7D4401)
   1 - Belkin F5D8633
   2 - Belkin N900 (F9K1104)
   3 - Belkin N300 (F7D7301)
   4 - Belkin AC1200

rsf (Belkin Auth Bypass) >
```

```
rsf (Netgear N300 Auth Bypass) > show devices

Target devices:
   0 - Netgear N300
   1 - Netgear JNR1010v2
   2 - Netgear JNR3000
   3 - Netgear JWNR2000v5
   4 - Netgear JWNR2010v5
   5 - Netgear R3250
   6 - Netgear WNR2020
   7 - Netgear WNR614
   8 - Netgear WNR618

rsf (Netgear N300 Auth Bypass) >
```

Weird-loops

FW keep on resetting.

```
[112 watchdog:btn_check +485] button RESET pressed
[  484.384000] firmadyne: ioctl: 0xc0084701
[  484.384000] firmadyne: ioctl: 0xc0084705
[  484.384000] firmadyne: ioctl: 0xc0084703
[  484.484000] firmadyne: ioctl: 0xc0084704
[112 watchdog:btn_check +485] button RESET pressed
[  484.484000] firmadyne: ioctl: 0xc0084701
[  484.484000] firmadyne: ioctl: 0xc0084705
[  484.484000] firmadyne: ioctl: 0xc0084703
[  484.584000] firmadyne: ioctl: 0xc0084704
[112 watchdog:btn_check +485] button RESET pressed
[  484.584000] firmadyne: ioctl: 0xc0084701
[  484.584000] firmadyne: ioctl: 0xc0084705
[  484.584000] firmadyne: ioctl: 0xc0084703
[  484.684000] firmadyne: ioctl: 0xc0084704
[112 watchdog:btn_check +485] button RESET pressed
[  484.684000] firmadyne: ioctl: 0xc0084701
[  484.684000] firmadyne: ioctl: 0xc0084705
[  484.684000] firmadyne: ioctl: 0xc0084703
[  484.784000] firmadyne: ioctl: 0xc0084704
[112 watchdog:btn_check +485] button RESET pressed
[  484.784000] firmadyne: ioctl: 0xc0084701
[  484.784000] firmadyne: ioctl: 0xc0084705
[  484.784000] firmadyne: ioctl: 0xc0084703
[  484.884000] firmadyne: ioctl: 0xc0084704
[112 watchdog:btn_check +485] button RESET pressed
[  484.884000] firmadyne: ioctl: 0xc0084701
[  484.884000] firmadyne: ioctl: 0xc0084705
[  484.884000] firmadyne: ioctl: 0xc0084703
[  484.984000] firmadyne: ioctl: 0xc0084704
[112 watchdog:btn_check +485] button RESET pressed
```

Kernel crashes and exceptions

```
gaurav@gaurav-HP-Spectre-x360-Converti...   gaurav@gaurav-HP-Spectre-x360-Converti...   gaurav@gaurav-HP-Spec
SERVD: start service [IP6T.WAN-4]
aaaaaaaa
wlfia0: WLC_GET_VAR(chanspec): No such device
wlfig0: WLC_GET_VAR(chanspec): No such device
wlfia0: WLC_GET_VAR(authe_sta_list): No such device
wlfig0: WLC_GET_VAR(authe_sta_list): No such device
[   45.337198] hostapd: unhandled page fault (11) at 0x50572068, code 0x005
[   45.337396] pgd = ce754000
[   45.337536] [50572068] *pgd=00000000
[   45.337727] CPU: 0 PID: 19118 Comm: hostapd Tainted: G        W       4.1.17+ #10
[   45.337835] Hardware name: Generic DT based system
[   45.337904] task: ce62fc00 ti: ce776000 task.ti: ce776000
[   45.338135] PC is at 0x2b54c
[   45.338215] LR is at 0x2bbac
[   45.338289] pc : [<0002b54c>]    lr : [<0002bbac>]    psr: 20000010
[   45.338289] sp : bed94d28  ip : b6f90cb4  fp : 00000000
[   45.338468] r10: 00243de0  r9 : 00000001  r8 : 00000002
[   45.338564] r7 : 74732041  r6 : 50572067  r5 : 0004872c  r4 : 00000000
[   45.338675] r3 : 0002bba0  r2 : 50572067  r1 : 00000000  r0 : 0004872c
[   45.338834] Flags: nzCv  IRQs on  FIQs on  Mode USER_32  ISA ARM  Segment user
[   45.338970] Control: 10c5387d  Table: 4e754059  DAC: 00000015
[   45.339081] CPU: 0 PID: 19118 Comm: hostapd Tainted: G        W       4.1.17+ #10
[   45.339192] Hardware name: Generic DT based system
[   45.339305] [<c001c8dc>] (unwind_backtrace) from [<c0019c70>] (show_stack+0x10/0x14)
[   45.339435] [<c0019c70>] (show_stack) from [<c001e7a4>] (__do_user_fault+0x74/0xbc)
[   45.339574] [<c001e7a4>] (__do_user_fault) from [<c0417384>] (do_page_fault+0x27c/0x2c0)
[   45.339706] [<c0417384>] (do_page_fault) from [<c0009214>] (do_DataAbort+0x34/0xb4)
[   45.339830] [<c0009214>] (do_DataAbort) from [<c0416d9c>] (__dabt_usr+0x3c/0x40)
[   45.340122] Exception stack(0xce777fb0 to 0xce777ff8)
[   45.340280] 7fa0:                                     0004872c 00000000 50572067 0002bba0
[   45.340416] 7fc0: 00000000 0004872c 50572067 74732041 00000002 00000001 00243de0 00000000
[   45.340553] 7fe0: b6f90cb4 bed94d28 0002bbac 0002b54c 20000010 ffffffff
[   45.346179] potentially unexpected fatal signal 11.
[   45.346302] CPU: 0 PID: 19118 Comm: hostapd Tainted: G        W       4.1.17+ #10
[   45.346430] Hardware name: Generic DT based system
[   45.346490] task: ce62fc00 ti: ce776000 task.ti: ce776000
[   45.346580] PC is at 0x2b54c
[   45.346613] LR is at 0x2bbac
[   45.346649] pc : [<0002b54c>]    lr : [<0002bbac>]    psr: 20000010
[   45.346649] sp : bed94d28  ip : b6f90cb4  fp : 00000000
[   45.346775] r10: 00243de0  r9 : 00000001  r8 : 00000002
[   45.346890] r7 : 74732041  r6 : 50572067  r5 : 0004872c  r4 : 00000000
[   45.346989] r3 : 0002bba0  r2 : 50572067  r1 : 00000000  r0 : 0004872c
[   45.347107] Flags: nzCv  IRQs on  FIQs on  Mode USER_32  ISA ARM  Segment user
[   45.347353] Control: 10c5387d  Table: 4e754059  DAC: 00000015
[   45.347430] CPU: 0 PID: 19118 Comm: hostapd Tainted: G        W       4.1.17+ #10
[   45.347506] Hardware name: Generic DT based system
[   45.347593] [<c001c8dc>] (unwind_backtrace) from [<c0019c70>] (show_stack+0x10/0x14)
[   45.347692] [<c0019c70>] (show_stack) from [<c002e6cc>] (get_signal+0x41c/0x47c)
[   45.347786] [<c002e6cc>] (get_signal) from [<c00194a8>] (do_signal+0x8c/0x35c)
[   45.347900] [<c00194a8>] (do_signal) from [<c00198d8>] (do_work_pending+0x54/0xac)
[   45.348040] [<c00198d8>] (do_work_pending) from [<c0016c8c>] (work_pending+0xc/0x20)
SERVD: stop service [IP6T.WAN-5]
SERVD: service [IP6T.WAN-5] is already stopped.
```

Stuck on Emulation

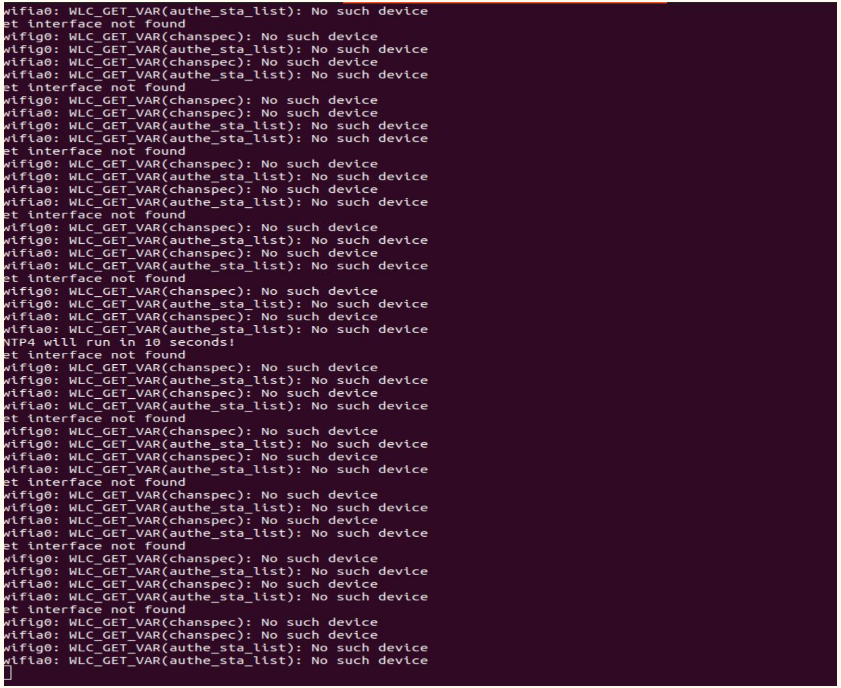
Especially in FirmAE, we observed the no access to shell or webserver.

Cannot get past the emulation stage.

```
gaurav@gaurav-HP-Spectre-x360-Convertible-13-aw0xxx:~/IoTSec/FirmAE$ sudo ./run.sh -b Netgear ../WNAP-FW.zip
[*] ../WNAP-FW.zip emulation start!!!
[*] extract done!!!
[*] get architecture done!!!
mke2fs 1.46.5 (30-Dec-2021)
e2fsck 1.46.5 (30-Dec-2021)
[*] infer network start!!!
ln: failed to create symbolic link '/home/gaurav/IoTSec/FirmAE/scratch/1/run_debug.sh': File exists
ln: failed to create symbolic link '/home/gaurav/IoTSec/FirmAE/scratch/1/run_analyze.sh': File exists
ln: failed to create symbolic link '/home/gaurav/IoTSec/FirmAE/scratch/1/run_boot.sh': File exists

[IID] 1
[MODE] boot
[+] Network reachable on 192.168.0.100!
[+] Web service on 192.168.0.100
[+] Connect with gdb-multiarch -q ./binaries/vmlinux.mipseb.4 -ex='target remote:1234'
Creating TAP device tap1_0...
Set 'tap1_0' persistent and owned by uid 0
Bringing up TAP device...
Starting emulation of firmware...
```

No access to services

- Even though the FW was emulated, frameworks failed to bring up web services, ssh services etc.
- This made testing the FW images hard as most of the exploits were based on accessing the web server, ssh, telnet etc.



```
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
NTP4 will run in 10 seconds!
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
et interface not found
wifig0: WLC_GET_VAR(chanspec): No such device
wifia0: WLC_GET_VAR(chanspec): No such device
wifig0: WLC_GET_VAR(authe_sta_list): No such device
wifia0: WLC_GET_VAR(authe_sta_list): No such device
```

Results

- Although FirmAE is relatively new, we felt more ease of emulation with Firmadyne.
- Community has also developed custom frameworks on top of firmadyne given its ease of use.
    - https://github.com/attify/firmware-analysis-toolkit (FAT)
- With both of them, we were not able to reproduce the CVEs for images using Routersploit.

```
rsf (Asus RT-N16 Password Disclosure) > run
[*] Running module exploits/routers/asus/rt_n16_password_disclosure...
[-] Connection error: http://192.168.1.1:8080/error_page.htm
```

- On top of this, emulation was itself a challenging task.
- Most of our time was spent rather selecting which FW images were emulated.
    - Even after that, some services like ssh, web were not working at all.