

Breaking into IoT From Analysis to Fuzzing

Roland Sako; Security Researcher

WINSLOW.IO

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Create the main window  
root = tk.Tk()  
root.title("Simple Keyboard")
```

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```



- **Roland Sako; Geneva; Switzerland;**

Network & Security Lecturer @[University of Applied Science Geneva]

Security Research, training development @[Winsl0w.io]

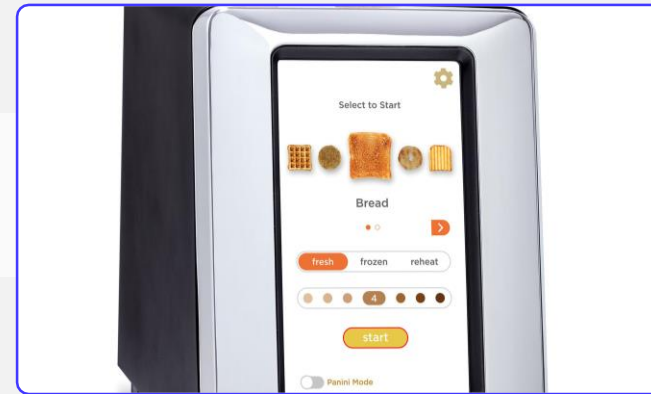
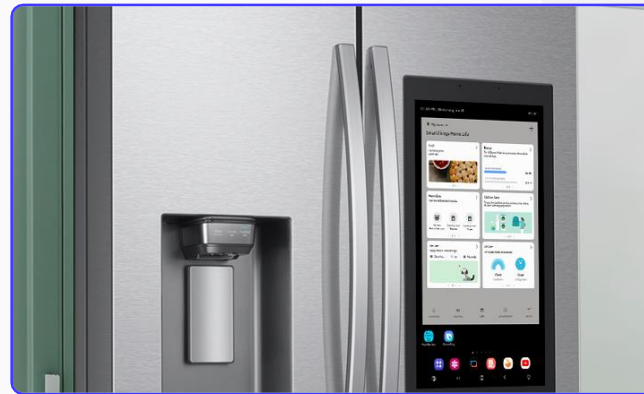
Vulnerability Research @[Kaspersky ICS CERT]

Internet of Things?

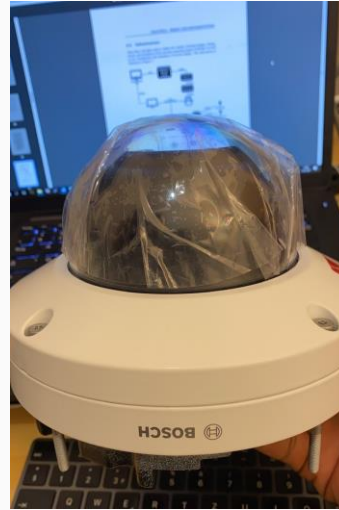
"Smart" devices that talk to each other (too much)

Gadgets that collect data, even when you don't want to

Everyday objects + Internet



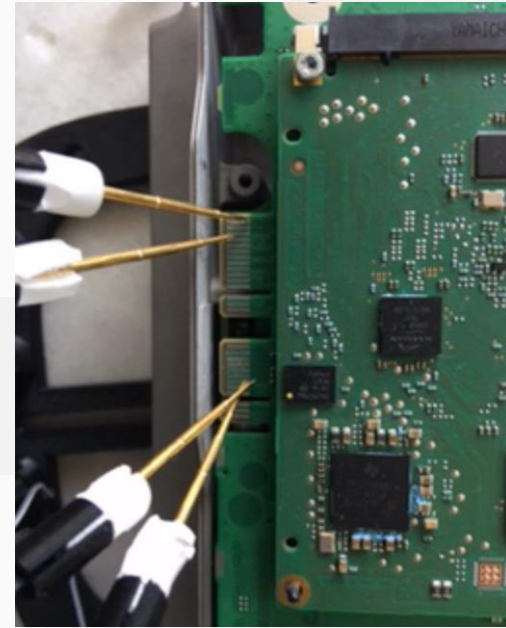
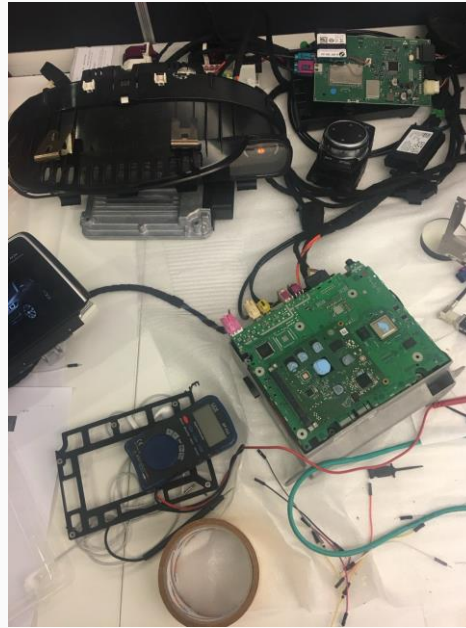
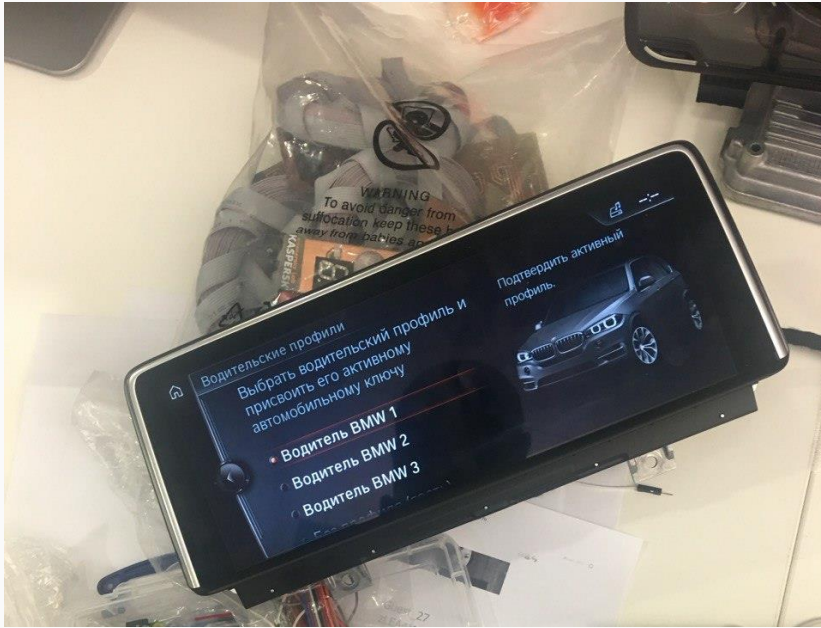
Internet of Things?



Internet of Things?



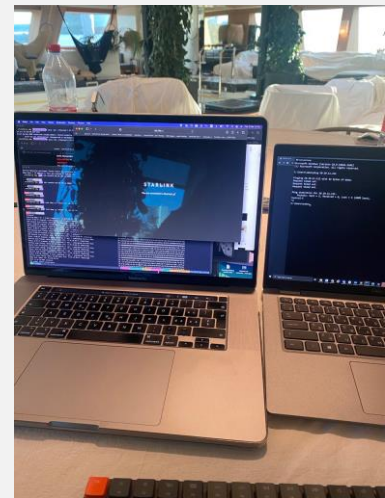
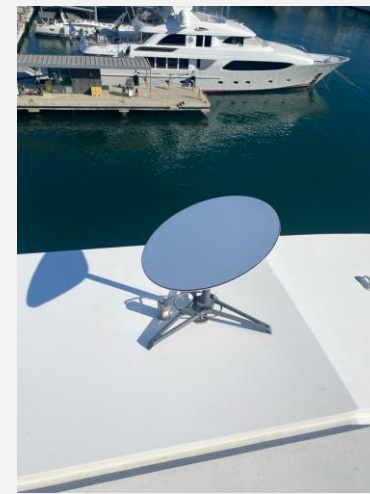
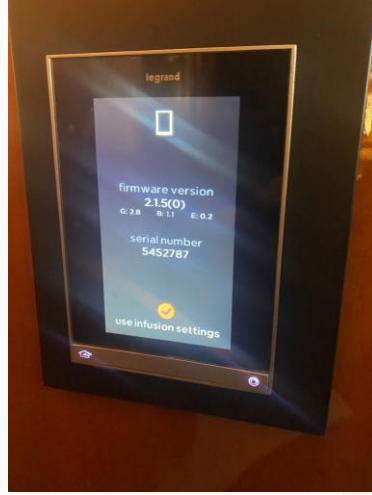
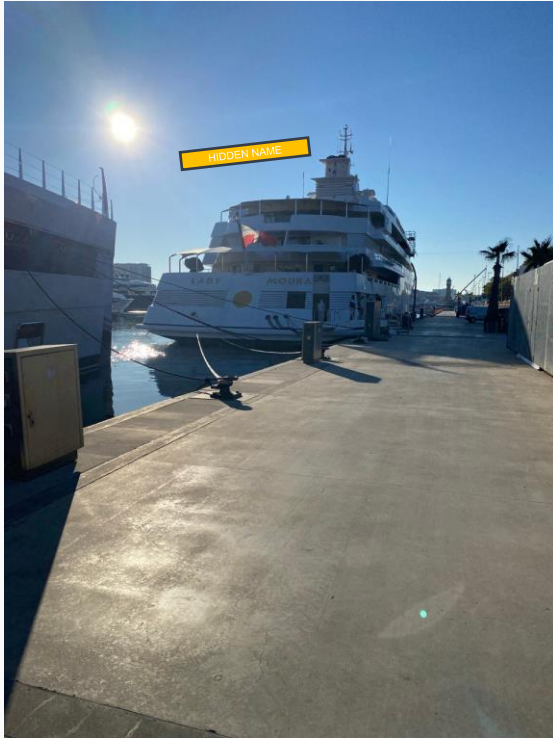
Internet of Things?



Internet of Things?



Internet of Things?



Common Issues

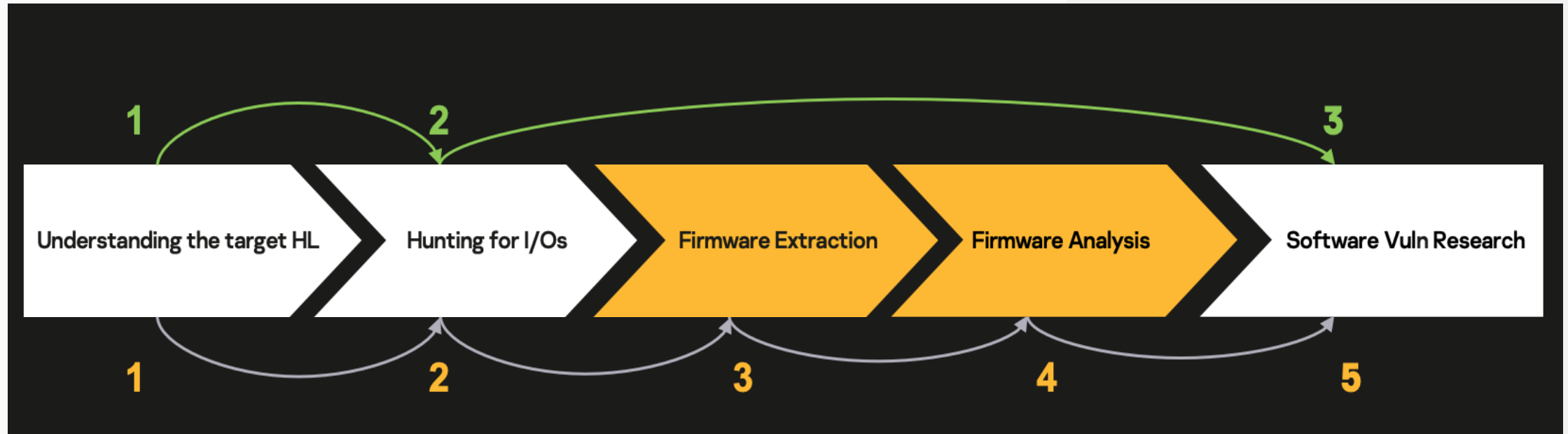
OWASP IoT TOP 10



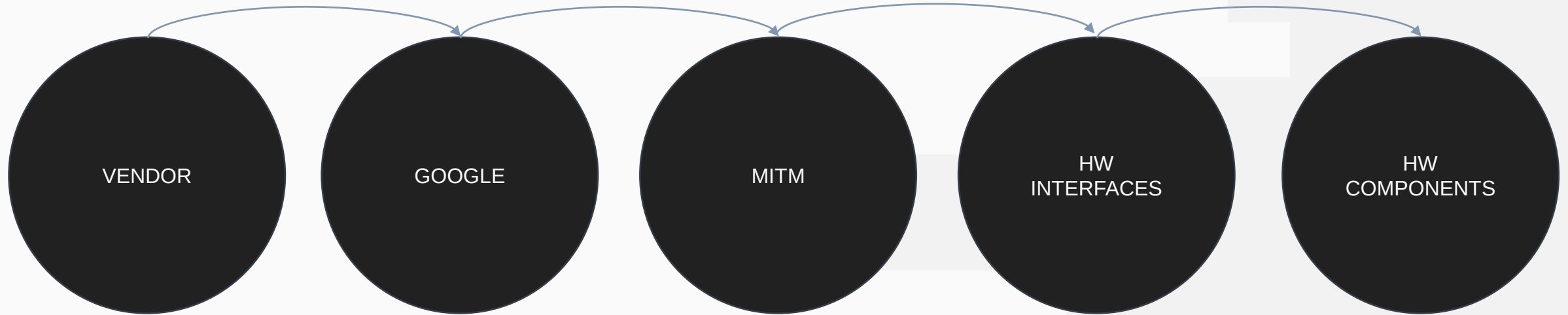
- | | |
|---|---------------------------------------|
| 1 Weak, Guessable, or Hardcoded Passwords | 6 Lack of a Proper Privacy Protection |
| 2 Insecure Network Services | 7 Insecure Data Transfer and Storage |
| 3 Insecure Ecosystem Interfaces | 8 Absence of Device Management |
| 4 Lack of Secure Update Mechanism | 9 Insecure Default Settings |
| 5 Using Insecure or Outdated Components | 10 Lack of Physical Hardening |

<https://www.wattlecorp.com/owasp-iot-top-10/>

Embedded Vuln Research



Getting the firmware





WORKING SMART HOME BUSINESS PROVIDERS OUR SECURITY TAPO

Download Center

For firmware, drivers, user guide, utility or any other download resources, please select the product from the search engine or the tab list.

Please note: The products of TP-Link partly contain software code developed by third parties and are licensed under the GNU General Public Licence ("GPL"), Version 1/Version 2/Version 3 or GNU Lesser General Public License. Please refer to the respective software condition to following the GPL licence terms.

You can review, print and download the respective GPL licence terms [here](#). You receive the GPL licence terms for the software used in TP-Link products for direct download and further information, including a list of the software code under [GPL Code Center](#).

The respective programs are distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the respective GNU General Public License for more details.

Type:

Model Number:

[? How to find the model number of your TP-Link device](#)

tp-link NETWORKING SMART HOME BUSINESS PROVIDERS OUR SECURITY TAPO STORE

Download for TL-WR841N V14.80

Please choose hardware version:

> How to find the hardware version on a TP-Link device

IMPORTANT: Model and hardware version availability varies by region. Please refer to your TP-Link regional website to determine product availability.
Vx.0=Vx.6/Vx.8 (eg V1.0=V1.6/V1.8)
Vx.x0=Vx.x6/Vx.x8 (eg V1.20=V1.26/V1.28)
Vx.30=Vx.32 (eg V3.30=V3.32)

Manual
[TL-WR841N\(US\)_V14.8_User Guide](#)

Firmware

A firmware update can resolve issues that the previous firmware version may have and improve its current performance.

To Upgrade

IMPORTANT: To prevent upgrade failures, please read the following before proceeding with the upgrade process.

- Please upgrade firmware from the local TP-Link official website of the purchase location for your TP-Link device, otherwise it will be against the warranty. Please click [here](#) to change site if necessary.
- Please verify the hardware version of your device for the firmware version. Wrong firmware upgrade may damage your device and void the warranty.
- [How to find the hardware version on a TP-Link device](#)
- **Do NOT turn off the power during the upgrade process, as it may cause permanent damage to the product.**

[More](#)

TL-WR841N(US)_V14.8_241230

Published Date: 2025-01-21 Language: English File Size: 4.72 MB

With a bit of luck



TL-WR841N firmwar|

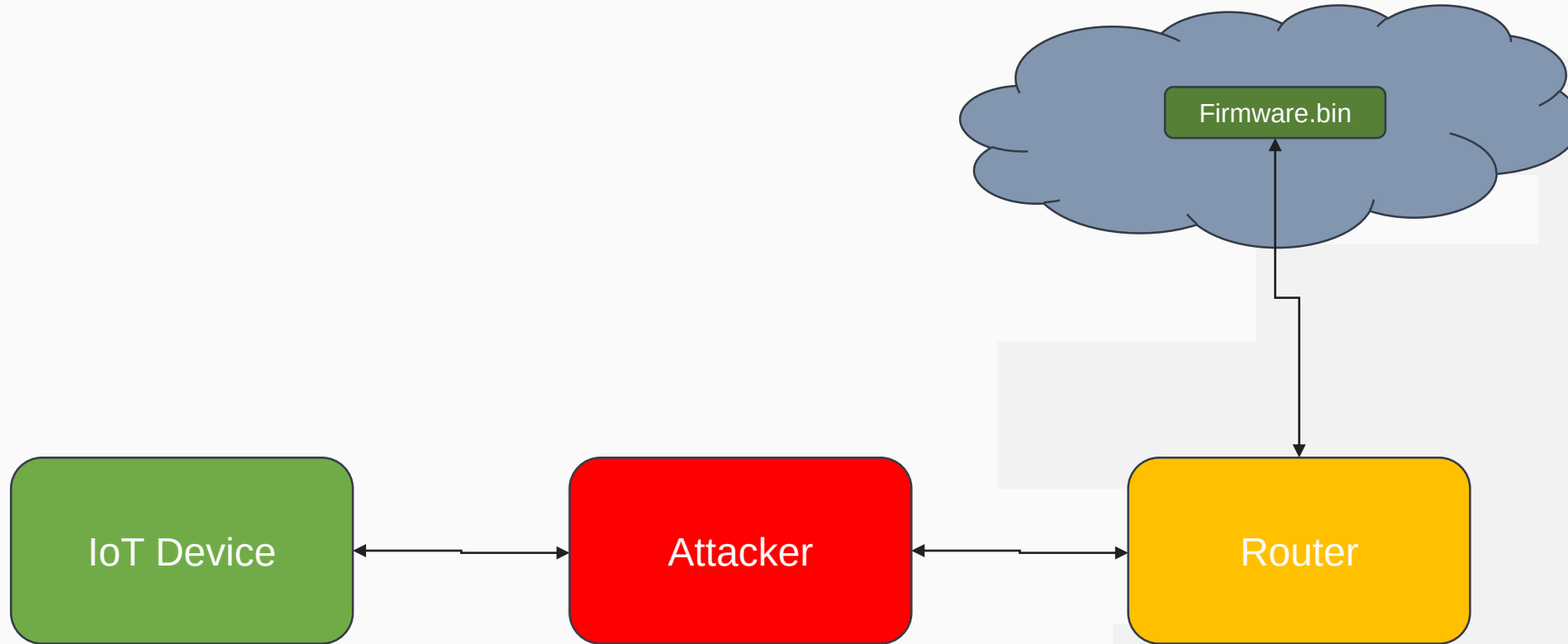


tl-wr841n firmware
tl-wr841n firmware **update**
tl-wr841n firmware **dd-wrt**
tl-wr841n firmware **download**
tl-wr841n firmware **v9**
tl-wr841n firmware **v8**
tl-wr841n firmware **v7**
tl-wr841n firmware **v10**

OTA Update



OTA MitM Attack



No.	Time	Source	Destination	Protocol	Length	Info
188	981.797493863	192.168.12.203	52.86.88.211	TLSv1.2	123	Application Data
189	981.954210600	52.86.88.211	192.168.12.203	TLSv1.2	123	Application Data
190	982.127956423	192.168.12.203	52.86.88.211	TCP	54	56169 → 8883 [ACK] Seq=9176 Ack=7059 Win=1834 Len=0
191	987.093249754	Intel_84:ce:8c	Intelliroids_82:aa:...	ARP	42	Who has 192.168.12.203? Tell 192.168.12.1
192	987.113423942	Intelliroids_82:aa:...	Intel_84:ce:8c	ARP	42	192.168.12.203 is at d4:ad:fc:82:aa:07
193	988.549207315	192.168.12.203	192.168.12.1	DNS	73	Standard query 0x59f5 A app.govee.com
194	988.549287808	192.168.12.203	192.168.12.1	DNS	73	Standard query 0x59f5 A app.govee.com
195	988.713262409	192.168.12.1	192.168.12.203	DNS	169	Standard query response 0x59f5 A app.govee.com A 3.231.106.7 A 34.203.99.187 A 34.207.10.56 A 52.203.102.153 A 3.218.204.125 A
196	988.751935743	192.168.12.203	3.231.106.7	TCP	58	56170 → 80 [SYN] Seq=0 Win=2920 Len=0 MSS=1460
197	988.917741706	3.231.106.7	192.168.12.203	TCP	58	80 → 56170 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1452
198	988.956408293	192.168.12.203	3.231.106.7	TCP	54	56170 → 80 [ACK] Seq=1 Ack=1 Win=2920 Len=0
199	988.957849560	192.168.12.203	3.231.106.7	TCP	234	56170 → 80 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=180 [TCP segment of a reassembled PDU]
200	989.122515921	3.231.106.7	192.168.12.203	TCP	54	80 → 56170 [ACK] Seq=1 Ack=181 Win=27872 Len=0
201	989.160985243	192.168.12.203	3.231.106.7	HTTP/JSON	212	POST /device/rest/devices/v1/wifiCheckVersion HTTP/1.1 , JSON (application/json)
202	989.327169508	3.231.106.7	192.168.12.203	TCP	54	80 → 56170 [ACK] Seq=1 Ack=339 Win=28944 Len=0
203	989.327178230	3.231.106.7	192.168.12.203	HTTP/JSON	796	HTTP/1.1 200 OK , JSON (application/json)
204	989.391654733	192.168.12.203	3.231.106.7	TCP	54	56170 → 80 [ACK] Seq=339 Ack=743 Win=2178 Len=0
205	989.396179569	192.168.12.203	3.231.106.7	TCP	54	56170 → 80 [FIN, ACK] Seq=339 Ack=743 Win=2178 Len=0
206	989.398966331	192.168.12.203	3.231.106.7	TCP	58	56171 → 80 [SYN] Seq=0 Win=2920 Len=0 MSS=1460
207	989.531876817	3.231.106.7	192.168.12.203	TCP	58	80 → 56171 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1452
208	989.531889313	3.231.106.7	192.168.12.203	TCP	54	80 → 56170 [FIN, ACK] Seq=743 Ack=348 Win=28944 Len=0
209	989.570189218	192.168.12.203	3.231.106.7	TCP	54	56171 → 80 [ACK] Seq=1 Ack=1 Win=2920 Len=0
210	989.571259630	192.168.12.203	3.231.106.7	TCP	54	56170 → 80 [ACK] Seq=348 Ack=744 Win=2177 Len=0
211	989.572726894	192.168.12.203	3.231.106.7	TCP	208	56171 → 80 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=146 [TCP segment of a reassembled PDU]
212	989.738898203	3.231.106.7	192.168.12.203	TCP	54	80 → 56171 [ACK] Seq=1 Ack=147 Win=27872 Len=0
213	989.775496954	192.168.12.203	3.231.106.7	HTTP/JSON	216	POST /bff-device/v1/config/gateway/group HTTP/1.1 , JSON (application/json)
214	989.942968518	3.231.106.7	192.168.12.203	TCP	54	80 → 56171 [ACK] Seq=1 Ack=389 Win=28944 Len=0
215	989.954486974	3.231.106.7	192.168.12.203	HTTP/JSON	466	HTTP/1.1 200 , JSON (application/json)
216	989.954496968	3.231.106.7	192.168.12.203	TCP	54	80 → 56171 [FIN, ACK] Seq=413 Ack=389 Win=28944 Len=0
217	989.985990268	192.168.12.203	3.231.106.7	TCP	54	56171 → 80 [ACK] Seq=389 Ack=414 Win=2507 Len=0
218	989.986955897	192.168.12.203	3.231.106.7	TCP	54	56171 → 80 [FIN, ACK] Seq=389 Ack=414 Win=2507 Len=0
219	989.148923752	3.231.106.7	192.168.12.203	TCP	54	80 → 56171 [ACK] Seq=414 Ack=318 Win=28944 Len=0
220	989.885329615	192.168.12.203	192.168.12.1	DNS	76	Standard query

Going after the HW

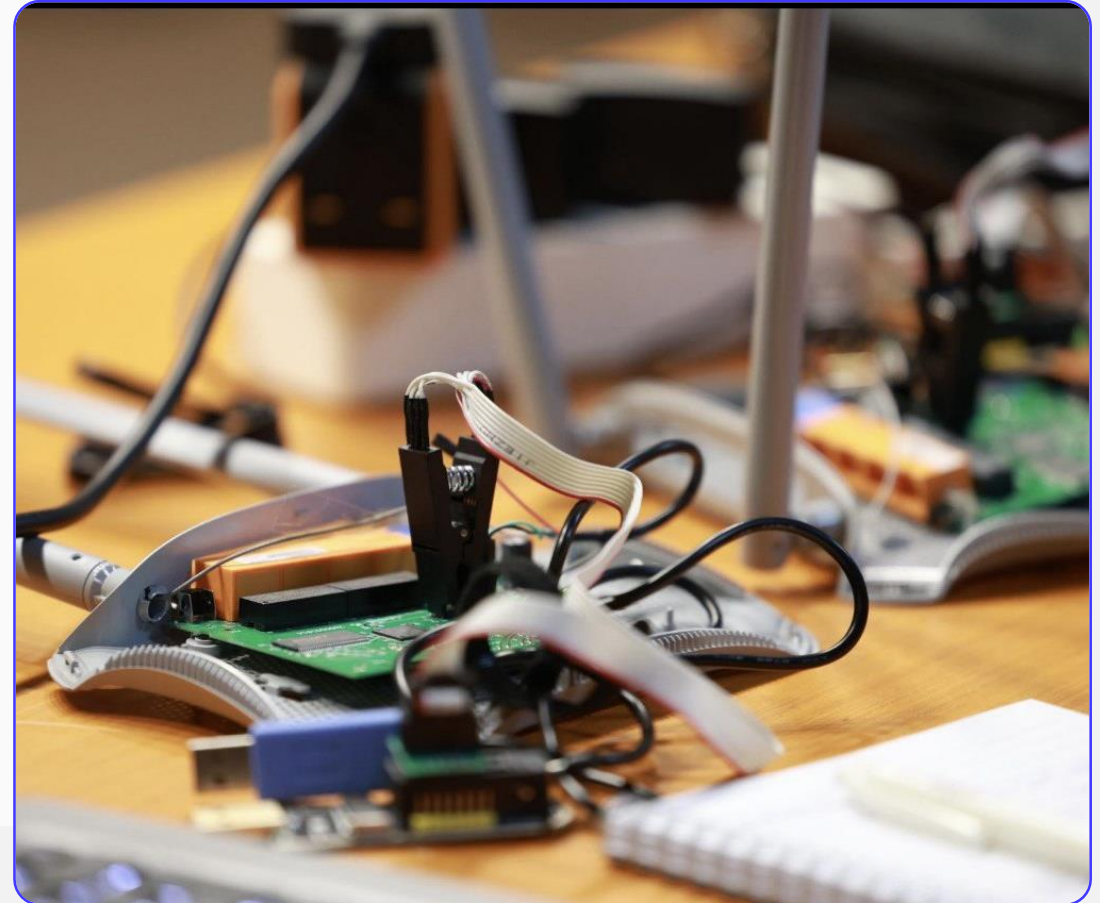
- Getting the firmware the [hard] way 😊

Open up the device

Identify interesting areas

Hook up

Extract



Going after the HW

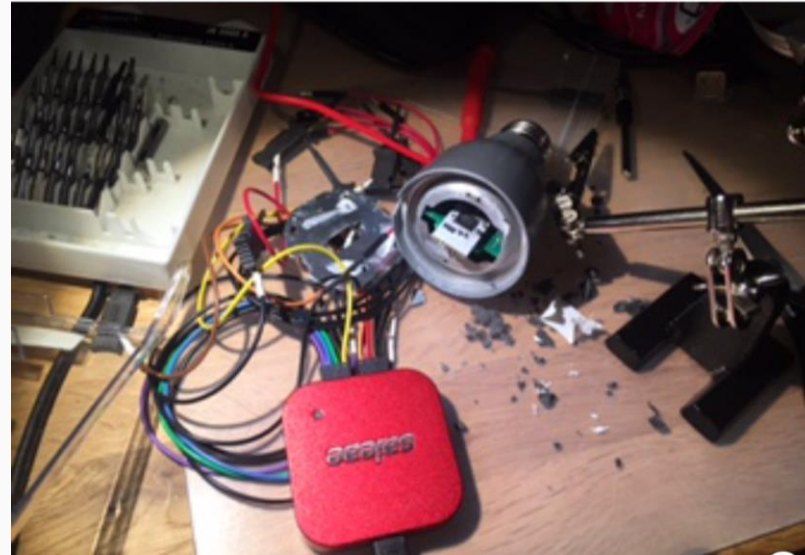
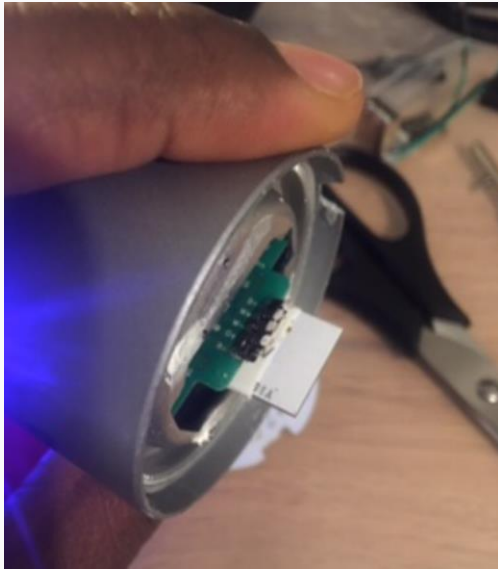
- Getting the firmware the [hard] way 😊



"I'll return it next time in the office"

Going after the HW

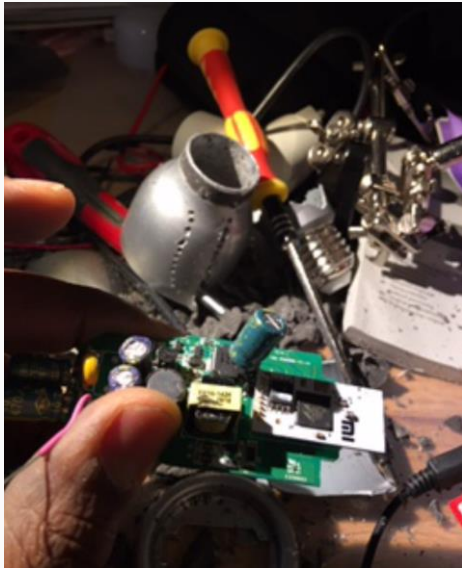
- Getting the firmware the [hard] way 😊



"I'll return it next time in the office"

Going after the HW

- Getting the firmware the [hard] way 😊



"I'll return it next time in the office"

Going after the HW

- Getting the firmware the [hard] way 😊

Open up the device

Identify interesting areas

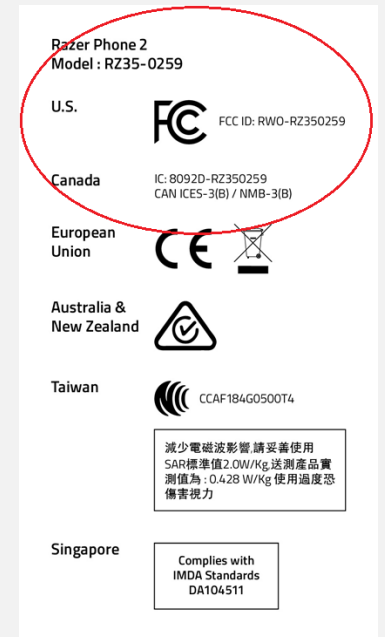
Hook up

Extract



Going after the HW

- Getting the firmware the [hard] way ☺



Going after the HW

- Getting the firmware the [hard] way 😊

LED BULB (Color) YLDP03YL
120V 60Hz 140mA
600lm 9W YYWW
1700K~6500K
FCC ID:2ABEU-YLDP03YL
MAC:XXXXXXXXXXXX
Made in China

CAUTION-SUITABLE FOR DAMP
LOCATIONS NOT FOR USE IN
TOTALLY ENCLOSED LUMINAIRES
DO NOT USE WITH DIMMERS

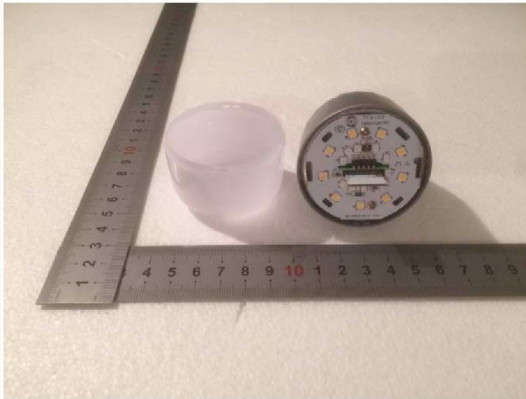


FCC ID.io Blog Search		
Exhibits		
All		
Document	Type	Submitted Available
Users Manual	Users Manual Adobe Acrobat PDF (912 kB)	2016-12-08 2016-12-08
Setup Photos	Test Setup Photos Adobe Acrobat PDF (103 kB)	2016-12-08 2016-12-08
Test Report	Test Report Adobe Acrobat PDF (1114 kB)	2016-12-08 2016-12-08
RF Exposure	RF Exposure Info Adobe Acrobat PDF (112 kB)	2016-12-08 2016-12-08
Int Photos	Internal Photos Adobe Acrobat PDF (647 kB)	2016-12-08 2016-12-08
FCC ID Label	ID Label/Location Info Adobe Acrobat PDF (308 kB)	2016-12-08 2016-12-08
Ext Photos	External Photos Adobe Acrobat PDF (1323 kB)	2016-12-08 2016-12-08
Confidentiality Letter	Cover Letter(s) Adobe Acrobat PDF (20 kB)	2016-12-08 2016-12-08
Letter POA	Cover Letter(s) Adobe Acrobat PDF (15 kB)	2016-12-08 2016-12-08
Schematics	Schematics Adobe Acrobat PDF (395 kB)	2016-12-08

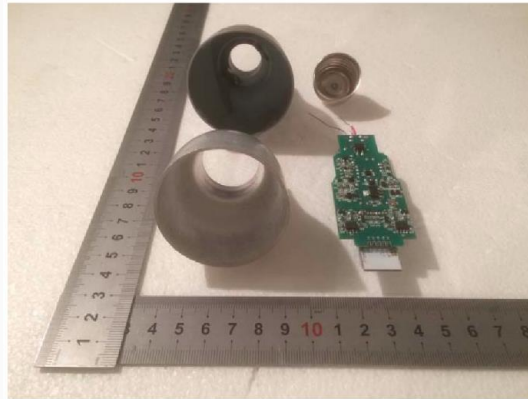
Going after the HW

- Getting the firmware the [hard] way 😊

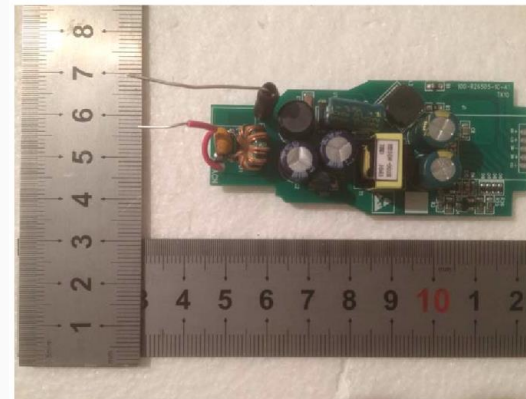
EUT – Shielding off View-1



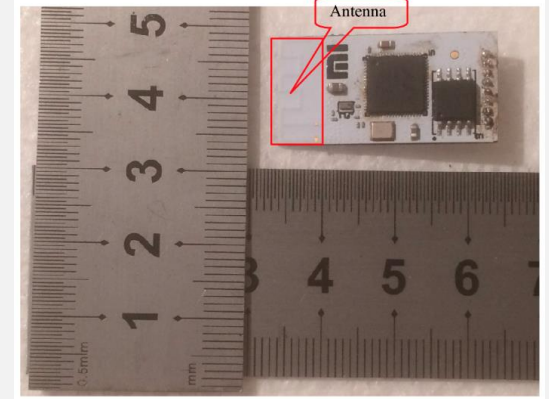
EUT – Shielding off View-4



EUT – Power Board Top View

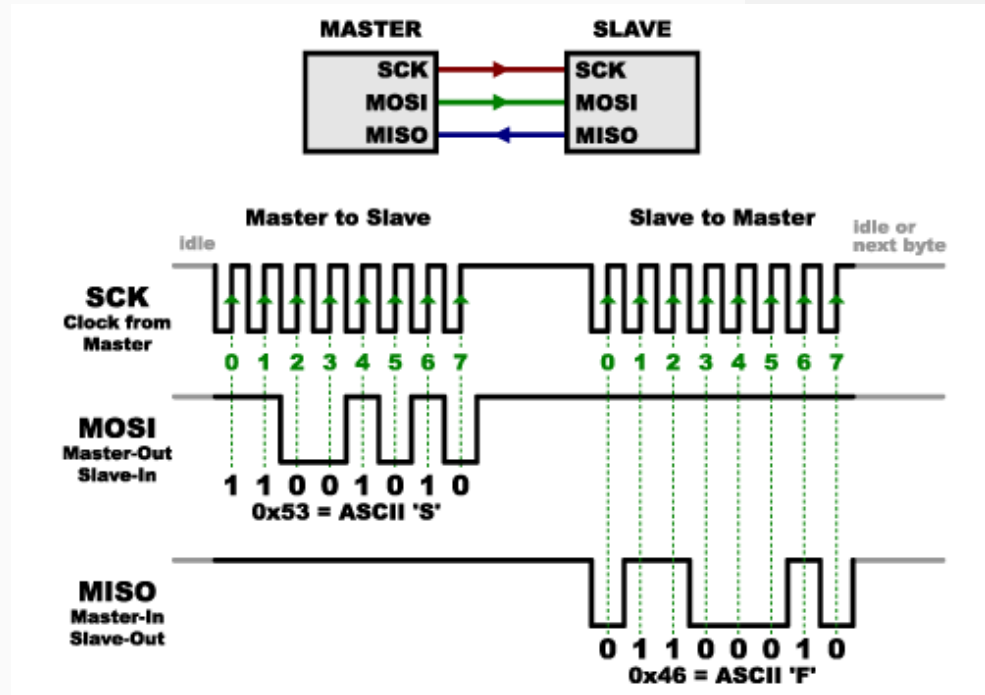
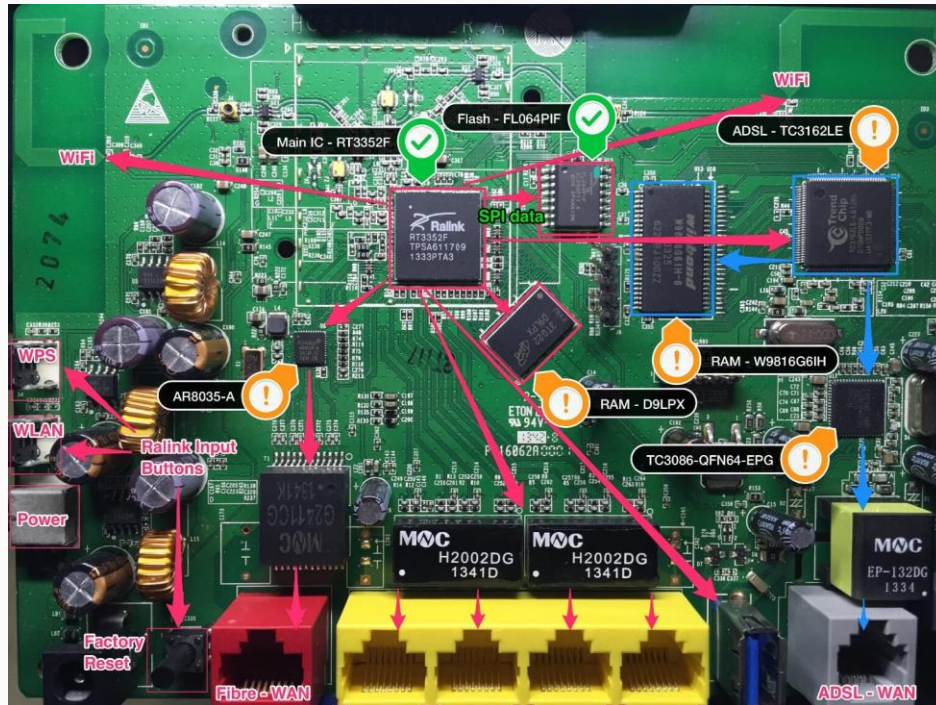


EUT – WIFI Module Top View



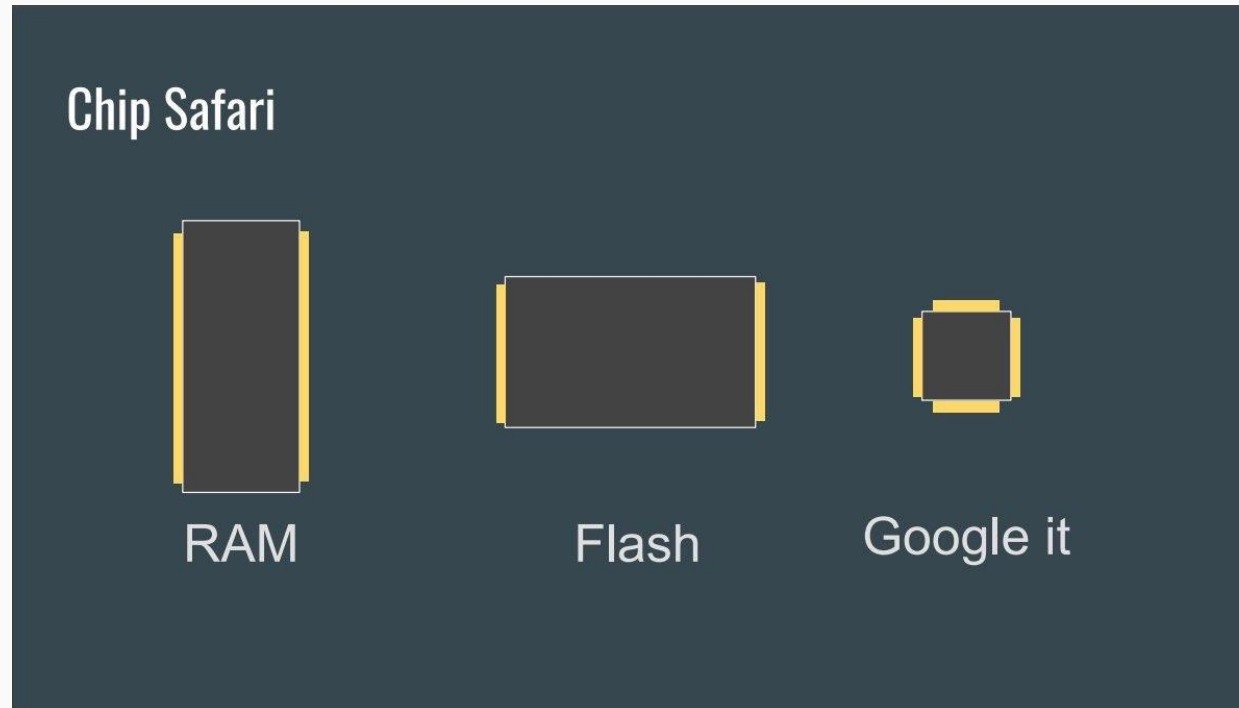
Going after the HW

- Getting the firmware the [hard] way 😊



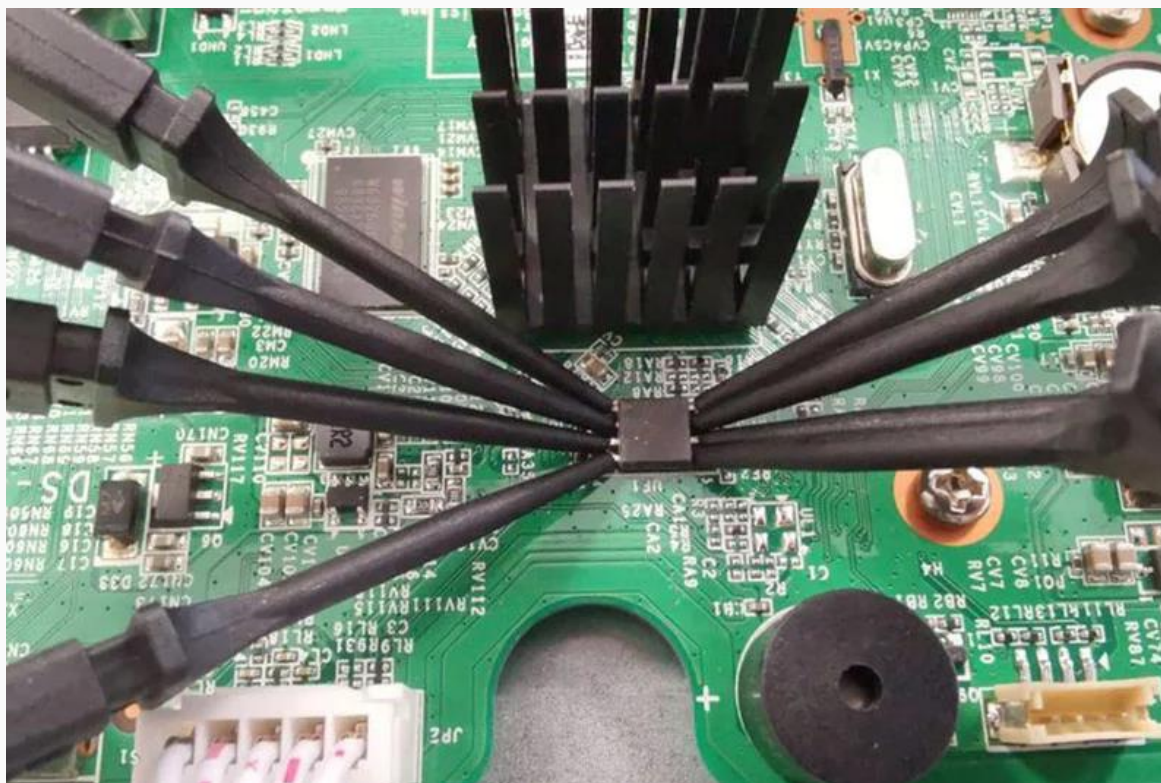
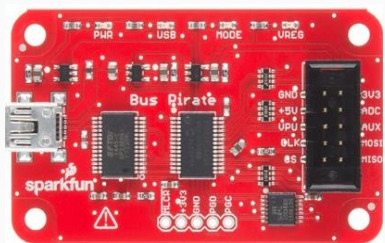
Going after the HW

- Getting the firmware the [hard] way 😊



Going after the HW

- Getting the firmware the [hard] way ☺



```
pi@icbm:~$ sudo flashrom -p linux_spi:dev=/dev/spidev0.0,spispeed=8000 -c S25FL128P.....0 -r S25FL128P-dump.bin
```

flashrom p1.0-65-ga9a03cc-dirty on Linux 4.9.35-v7+ (armv7l)
flashrom is free software, get the source code at <https://flashrom.org>

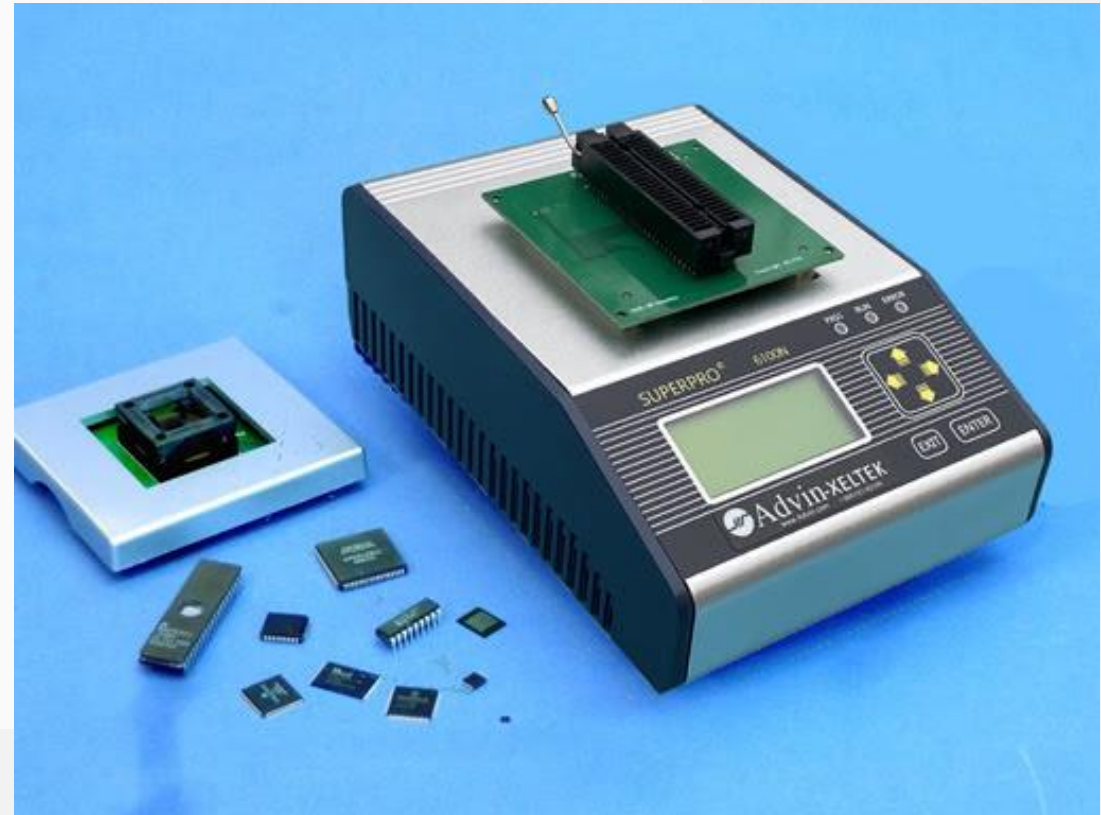
Using clock_gettime for delay loops (clk_id: 1, resolution: 1ns).
Found Spansion flash chip "S25FL128P.....0" (16384 kB, SPI) on linux_spi.
==

This flash part has status UNTESTED for operations: PROBE READ ERASE WRITE
The test status of this chip may have been updated in the latest development version of flashrom. If you are running the latest development version, please email a report to flashrom@flashrom.org if any of the above operations work correctly for you with this flash chip. Please include the flashrom log file for all operations you tested (see the man page for details), and mention which mainboard or programmer you tested in the subject line.
Thanks for your help!
Reading flash... done.

This flash part has status UNTESTED for operations: PROBE READ ERASE WRITE
The test status of this chip may have been updated in the latest development version of flashrom.

Going after the HW

- Getting the firmware the [hard] way ☺



Firmware Analysis

Bootloader

Kernel

File System

=> Linux

```
672 buildroot 0x2A0 uImage header, header size: 64 bytes, header CRC: 0xA5801A8,  
nt: 0x80000000, data CRC: 0xBD205199, OS: Linux, CPU: MIPS, ima  
736 0x2E0 LZMA compressed data, properties:  
1508000 0x1702A0 Squashfs filesystem, little endian
```

```
lab@ubuntu:~/Desktop/firmware/miwifi_rlc_firmware_b9d56.2.7.11.bin.extracted/squashfs-roots$ ls  
bin data dev etc extdisks lib mnt opt overlay proc readonly rom root sbin sys usr var www  
total 712  
drwxr-xr-x 2 lab lab 4096 Sep 26 03:35 .  
drwxr-xr-x 21 lab lab 4096 Oct 26 2015 ..  
-rwxr-xr-x 1 lab lab 6583 Oct 25 2015 ac  
-rwxr-xr-x 1 lab lab 9927 Oct 25 2015 acl  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 ash -> busybox  
-rwxr-xr-x 1 lab lab 8439 Oct 26 2015 ated  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 base64 -> busybox  
-rwxr-xr-x 1 lab lab 465983 Oct 25 2015 busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 cat -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 chgrp -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 chmod -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 chown -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 cp -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 date -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 dd -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 df -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 dmesg -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 dnsdomainname -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 echo -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 egrep -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 false -> busybox  
lrwxrwxrwx 1 lab lab 7 Oct 26 2015 fgrep -> busybox  
-rwxr-xr-x 1 lab lab 279 Oct 25 2015 flash_check.sh
```

Firmware Analysis

Bootloader

Kernel / App

Additional
Files

=> RTOS

DECIMAL	HEXADECIMAL	DESCRIPTION
4548	0x11C4	Unix path: /home/lab/esp/esp-idf/components/es
9388	0x24AC	Unix path: /home/lab/esp/esp-idf/components/es
9948	0x26DC	Unix path: /home/lab/esp/esp-idf/components/es
10160	0x27B0	Unix path: /home/lab/esp/esp-idf/components/es
10412	0x28AC	Unix path: /home/lab/esp/esp-idf/components/es

```
Lab@ubuntu:~/Desktop/rtos$ r2 scan.bin
-- I did it for the punz.
[0x00000000]> 1
blockz 0x0
block 0x100
fd 3
file scan.bin
format any
lowr false
mode r-x
size 0x85fc0
humansz 535.9K

[0x00000000]> px
.offset 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x00000000 e909 0210 0412 0840 ee00 0000 0000 0000 .....0.....
0x00000010 0000 0000 0000 0001 2000 403f f015 0100 .....07....
0x00000020 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x00000030 0000 0000 1414 1407 0000 0000 0000 0000 .....
0x00000040 0000 0000 4800 0000 4800 0000 4800 0000 ....H...H...
0x00000050 4800 0000 4800 0000 4800 0000 4800 0000 ....H...H...
0x00000060 0100 0000 4b41 5350 4552 534b 5000 0000 ...KASPERSKY...
0x00000070 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x00000080 0000 0000 4b41 5350 4552 534b 505f 5041 ...KASPERSKY_PA
0x00000090 5353 0000 0000 0000 0000 0000 0000 0000 SS.....
0x000000a0 0000 0000 0000 0000 0000 0000 0000 .....
0x000000b0 0000 0000 0000 0000 0000 0000 0000 .....
0x000000c0 0000 0000 8233 0d40 c530 0d40 ca30 0d40 ....3.0.0.0.0
0x000000d0 2131 0d40 5333 0d40 5133 0d40 fc32 0d40 !1.053.003.0.2.0
0x000000e0 2033 0d40 ac34 0d40 5531 0d40 9634 0d40 3.0.4.0U1.0.4.0
0x000000f0 0134 0d40 ac34 0d40 ac34 0d40 ac34 0d40 .4.0.4.0.4.0.4.0
[0x00000000]>
```

Firmware Analysis

The screenshot shows the GitHub repository for 'binwalk'. At the top, it indicates the repository is 'Public' with 360 watches, 1.6k forks, and 12.1k stars. Below this, there's a navigation bar with 'master' branch selected, 1 branch, and 12 tags. A search bar and 'Add file' / 'Code' buttons are also present. The main content area is divided into two columns. The left column lists files and their commit history, including .github/workflows, dependencies, fuzzing, images, scripts, src, tests, .dockerignore, .gitattributes, .gitignore, CARGO_README.md, Cargo.lock, Cargo.toml, Dockerfile, LICENSE, and README.md. The right column contains an 'About' section, 'Releases' (showing Binwalk v3.1.0 as the latest release), 'Packages' (no packages published), 'Contributors' (82 contributors), and 'Languages' (Rust 99.7%, Other 0.3%).

File	Description	Commit Date
.github/workflows	feat(gh-workflow-quality): add lint check	4 months ago
dependencies	Replaced external DTB extractor with an internal one	3 months ago
fuzzing	Code formatting fixes	3 weeks ago
images	Added carve.png for wiki entry	3 months ago
scripts	Added binwalk-ui script	4 months ago
src	Added support for partial ZIP files	3 weeks ago
tests	Added ZIP tests	3 weeks ago
.dockerignore	feat: add dockerfile	5 months ago
.gitattributes	Added initial integration testing	4 months ago
.gitignore	Included Cargo.lock	4 months ago
CARGO_README.md	Minor cargo readme change	4 months ago
Cargo.lock	Updated to use delink git repo	2 months ago
Cargo.toml	Updated to use delink git repo	2 months ago
Dockerfile	Fixed docker issues	4 months ago
LICENSE	binwalkv3	5 months ago
README.md	Update README.md	4 months ago

Unpacking the firmware

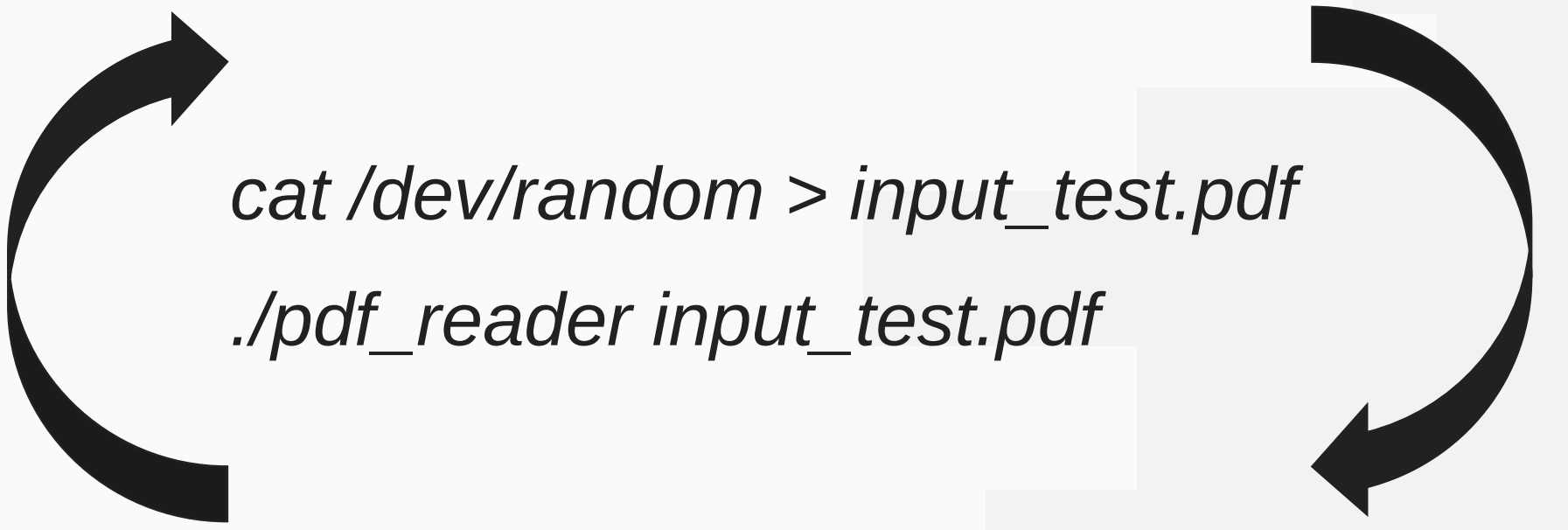
```
# binwalk -Mre <firmware.bin>
```

Quick search for potentially interesting binaries/scripts

```
# grep -rE 'system\(|popen\(|exec\(|ve\|vp\|vpe\)\(|sprintf\(' /bin /sbin /usr/bin /usr/sbin /opt
```

*An **automated** method for feeding a target with **input** to identify those that lead to **unexpected** conditions.*

Fuzzing 101



```
cat /dev/random > input_test.pdf  
./pdf_reader input_test.pdf
```

did it crash?

Fuzzing 101

TL;DR;

Triggers Buffer Overflow if the input file starts with *PHDAYCAIRO*.

```
// Read first 10 bytes of file into buffer
if (fread(buffer, 1, 10, fp) < 10) {
    perror("Error reading file");
    return 1;
}

// Check for "PHDAYCAIRO" in the first 10 bytes of the file
if (strncmp(buffer, "PHDAYCAIRO", 10) == 0) {
    //trigger a dummy buffer overflow
    strncpy(buffer + BUFFER_SIZE, "CRASHME", 10);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define BUFFER_SIZE 100

int main(int argc, char *argv[])
{
    char *buffer;
    char *filename;
    FILE *fp;
    int i;

    if (argc < 2) {
        printf("Usage: %s <filename>\n", argv[0]);
        return 1;
    }
    filename = argv[1];

    buffer = malloc(BUFFER_SIZE);
    if (buffer == NULL) {
        perror("malloc failed");
        return 1;
    }

    fp = fopen(filename, "r");
    if (fp == NULL) {
        perror("Error opening file");
        return 1;
    }

    // Read first 10 bytes of file into buffer
    if (fread(buffer, 1, 10, fp) < 10) {
        perror("Error reading file");
        return 1;
    }

    // Check for "PHDAYCAIRO" in the first 10 bytes of the file
    if (strncmp(buffer, "PHDAYCAIRO", 10) == 0) {
        //trigger a dummy buffer overflow
        strncpy(buffer + BUFFER_SIZE, "CRASHME", 10);
    }

    fclose(fp);
    free(buffer);

    return 0;
}
```

Fuzzing 101

```
// Read first 10 bytes of file into buffer
if (fread(buffer, 1, 10, fp) < 10) {
    perror("Error reading file");
    return 1;
}

// Check for "PHDAYCAIRO" in the first 10 bytes of the file
if (strncmp(buffer, "PHDAYCAIRO", 10) == 0) {
    //trigger a dummy buffer overflow
    strncpy(buffer + BUFFER_SIZE, "CRASHME", 10);
}
```

```
> cat crashme.txt
PHDAYCAIROAAAAAAAAAAAAAAAAAAAA
> ./hackme crashme.txt
hackme(79078,0x204c28840) malloc: nano zone abandoned due to inability to reserve vm space.
=====
==79078==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60b0000000a4 at pc 0x000100e9a4a4 bp 0x00016f5df290 sp 0x00016f5dea40
WRITE of size 10 at 0x60b0000000a4 thread T0
#0 0x100e9a4a0 in strncpy+0x538 (libclang_rt.asan_osx_dynamic.dylib:arm64e+0x4e4a0)
#1 0x100823c2c in main+0x1d8 (hackme:arm64+0x100003c2c)
#2 0x19af38270 (<unknown module>)

0x60b0000000a4 is located 0 bytes after 100-byte region [0x60b000000040,0x60b0000000a4)
allocated by thread T0 here:
#0 0x100ea0c04 in malloc+0x94 (libclang_rt.asan_osx_dynamic.dylib:arm64e+0x54c04)
#1 0x100823b38 in main+0xe4 (hackme:arm64+0x100003b38)
#2 0x19af38270 (<unknown module>)

SUMMARY: AddressSanitizer: heap-buffer-overflow (libclang_rt.asan_osx_dynamic.dylib:arm64e+0x4e4a0) in strncpy+0x538
Shadow bytes around the buggy address:
0x60afffffffe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x60afffffff80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x60afffffff00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x60afffffff80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x60b000000000: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
=>0x60b000000080: 00 00 00 00[04]fa fa fa fa fa fa fa fa fa fa fa fa
0x60b000000100: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x60b000000180: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x60b000000200: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x60b000000280: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x60b000000300: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00 01 02 03 04 05 06 07
Partially addressable: 08 09 0a 0b 0c 0d 0e 0f
Heap left redzone: fa
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Intra object redzone: ac
ASan internal: bb
Left alloca redzone: ca
Right alloca redzone: cb
==79078==ABORTING
[1] 79078 abort ./hackme crashme.txt
```

How ![to fuzz]

```
#!/bin/bash
```

```
i=0
```

```
while [[ 0 ]];
```

```
do
```

```
    echo "run #${i}.."
```

```
    head /dev/random > corpus/fuzz-${i}
```

```
    ./hackme corpus/fuzz-${i}
```

```
    [[ $? != 0 ]] && exit 1
```

```
    i=$((i + 1))
```

```
done
```

```
/tmp
> head -c 20 /dev/random | hexdump -C
00000000 42 95 77 7c ff d3 67 47 a1 44 0d 23 bb b9 ce fc |B.w|..gG.D.#...|
00000010 d0 78 48 b0                                     |.xH.|
00000014

/tmp
> head -c 20 /dev/random | hexdump -C
00000000 d6 38 62 23 3b 18 5f ff 40 e3 22 c8 73 27 a7 59 |.8b#;._.@".'s'.Y|
00000010 88 63 67 74                                     |.cgt|
00000014

/tmp
> head -c 20 /dev/random | hexdump -C
00000000 e0 9a 3a e8 28 49 ea f1 22 27 62 b2 da f5 f2 5a |....(I.."'b....Z|
00000010 d4 ab 9d 3a                                     |...:|
00000014

/tmp
> head -c 20 /dev/random | hexdump -C
00000000 f3 54 31 11 26 44 ae 00 33 6c f3 08 c3 4e e1 39 |.T1.&D..3l...N.9|
00000010 cb 53 c4 d1                                     |.S..|
00000014

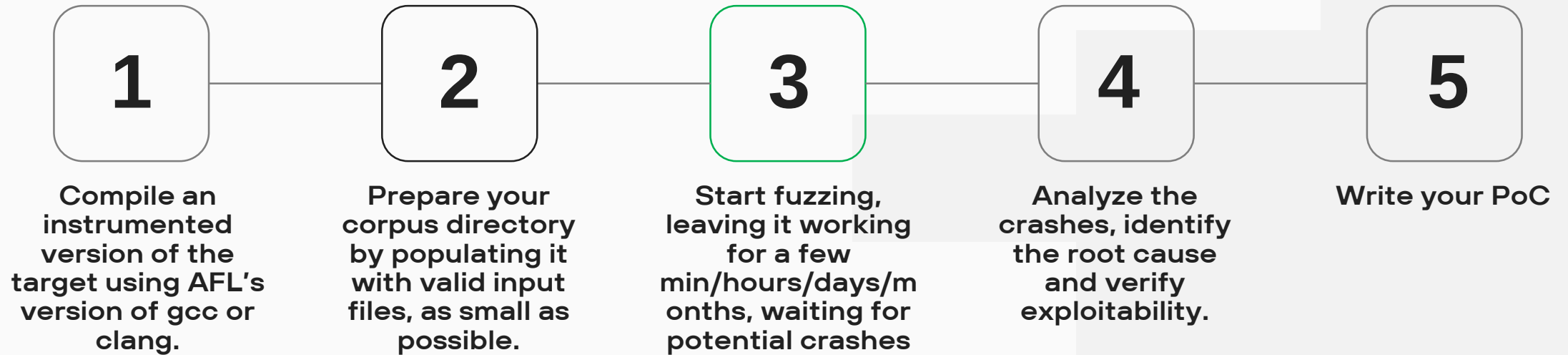
/tmp
> head -c 20 /dev/random | hexdump -C
00000000 e2 89 53 7e d9 d4 dc e0 fc 5e 9c 7f 67 4d 65 e2 |...S~.....^..gMe.|
00000010 f5 5b b8 fb                                     |.[..|
00000014

/tmp
> head -c 20 /dev/random | hexdump -C
00000000 ae 83 11 a0 6d 42 9a 98 78 a0 71 28 81 22 b8 f1 |....mB..x.q(,"..|
00000010 4d 15 35 25                                     |M.5%|
00000014
```

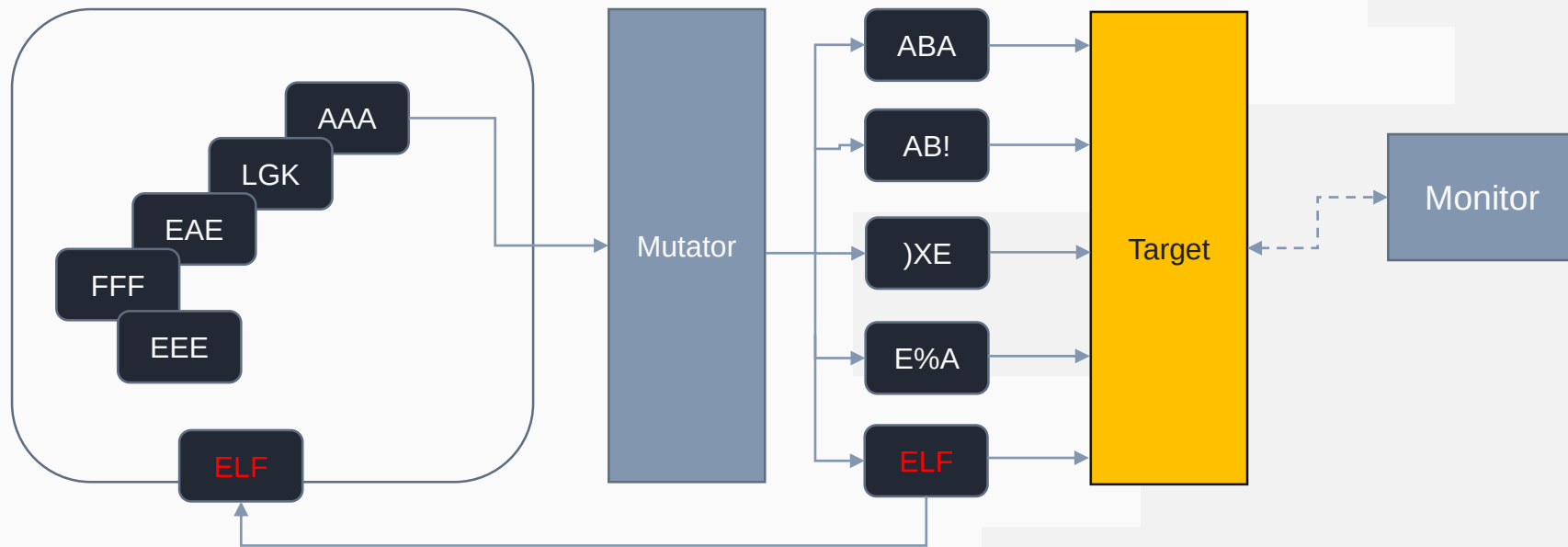
[illegible]

Fuzzing 101

AFL Steps



AFL The Great



AFL The Great

```
1 import sys
2
3 if len(sys.argv) != 2:
4     print("Usage: python3 elf-check.py filename")
5     sys.exit(1)
6
7 try:
8     with open(sys.argv[1], "rb") as f:
9         magic = f.read(4)
10        if magic[0] == 0x7f and magic[1] == 0x45 and magic[2] == 0x4c and magic[3] == 0x46:
11            print(f"{sys.argv[1]} is an ELF file.")
12        else:
13            print(f"{sys.argv[1]} is not an ELF file.")
14 except IOError as e:
15     print(f"Error opening file: {e}")
16     sys.exit(1)
17
18
```

```
1 import sys
2
3 if len(sys.argv) != 2:
4     print("Usage: python3 elf-check.py filename")
5     sys.exit(1)
6
7 try:
8     with open(sys.argv[1], "rb") as f:
9         magic = f.read(4)
10        if magic[0] == 0x7f:
11            print("Branch 1 taken.")
12            if magic[1] == 0x45:
13                print("Branch 2 taken.")
14                if magic[2] == 0x4c:
15                    print("Branch 3 taken.")
16                    if magic[3] == 0x46:
17                        print(f"{sys.argv[1]} is an ELF file.")
18                    else:
19                        print("Branch 4 not taken.")
20                        print(f"{sys.argv[1]} is not an ELF file.")
21                else:
22                    print("Branch 3 not taken.")
23                    print(f"{sys.argv[1]} is not an ELF file.")
24            else:
25                print("Branch 2 not taken.")
26                print(f"{sys.argv[1]} is not an ELF file.")
27        else:
28            print("Branch 1 not taken.")
29            print(f"{sys.argv[1]} is not an ELF file.")
30 except IOError as e:
31     print(f"Error opening file: {e}")
32     sys.exit(1)
33
```

AFL The Great

```
00000000 48 65 6c 6c 6f 2c 20 57 6f 72 6c 64 21 0a |Hello, World!.|
0000000e
```

```
00000000 7f 45 79 46 0a |.EyF.|
00000005
```

```
00000000 7f 45 4c 78 0a |.ELx.|
00000005
```

```
00000000 7f 45 4c 46 |.ELF|
00000004
```

```
Branch 1 not taken.
hello.txt is not an ELF file.
```

```
Branch 1 taken.
Branch 2 taken.
Branch 3 not taken.
myelf2 is not an ELF file.
```

```
Branch 1 taken.
Branch 2 taken.
Branch 3 taken.
Branch 4 not taken.
myelf1 is not an ELF file.
```

```
Branch 1 taken.
Branch 2 taken.
Branch 3 taken.
myelf is an ELF file.
```

AFL The Great

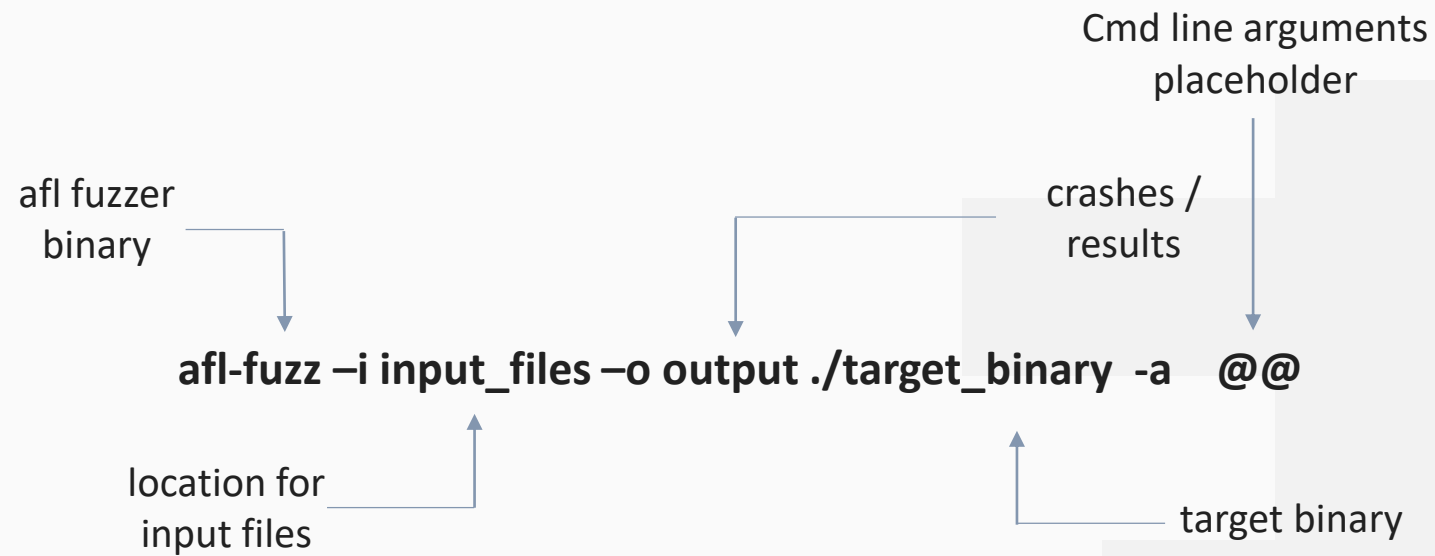
```
afl-gcc fuzzme.c
```

```
afl-g++ fuzzme.cpp
```

```
CC=afl-gcc CXX=afl-g++ ./configure  
make
```

```
cmake -DCMAKE_CXX_COMPILER=afl-g++ -DCMAKE_C_COMPILER=afl-gcc  
make
```

AFL The Great



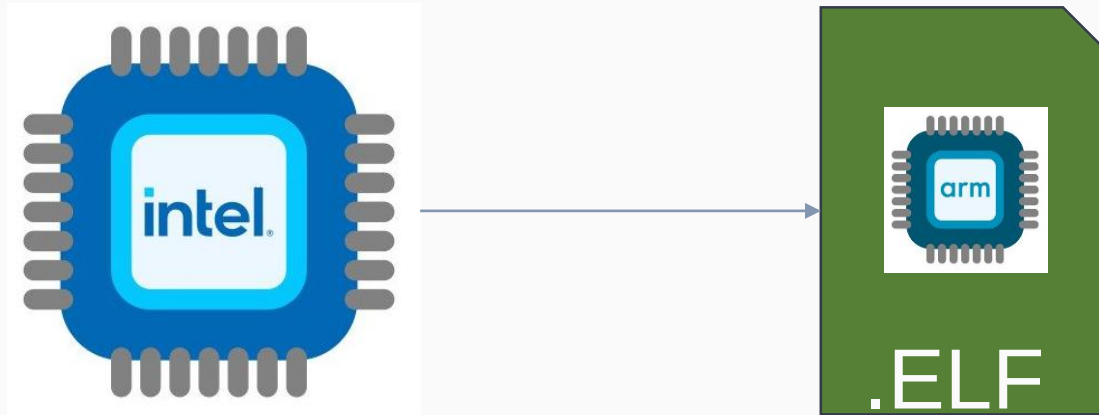
AFL The Great

american fuzzy lop 2.52b (slave2)

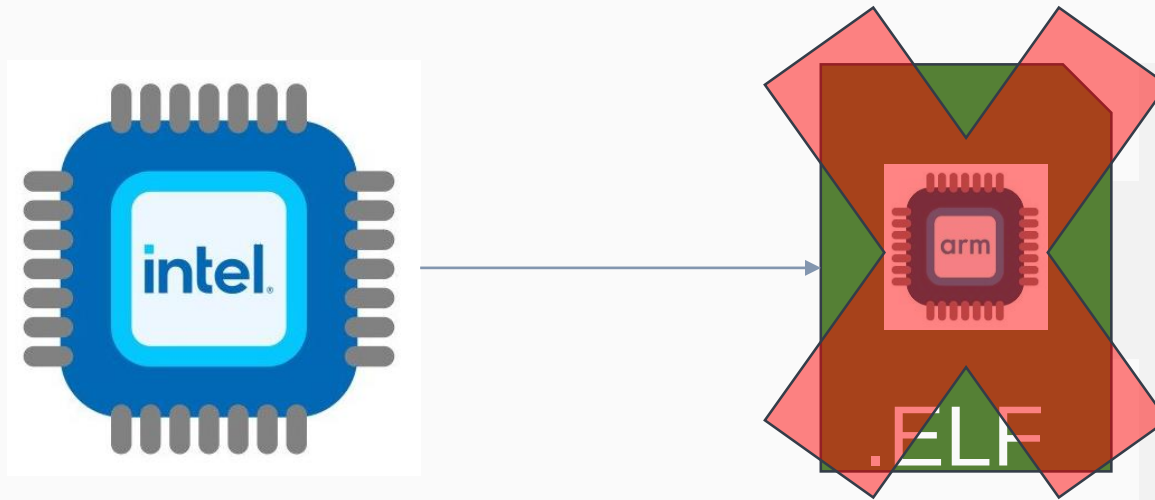
process timing		overall results	
run time	: 0 days, 0 hrs, 12 min, 23 sec	cycles done	: 0
last new path	: 0 days, 0 hrs, 0 min, 2 sec	total paths	: 4315
last uniq crash	: 0 days, 0 hrs, 0 min, 21 sec	uniq crashes	: 19
last uniq hang	: 0 days, 0 hrs, 0 min, 0 sec	uniq hangs	: 226
cycle progress		map coverage	
now processing	: 2747 (63.66%)	map density	: 2.31% / 8.91%
paths timed out	: 0 (0.00%)	count coverage	: 4.18 bits/tuple
stage progress		findings in depth	
now trying	: splice 11	favored paths	: 493 (11.43%)
stage execs	: 113/512 (22.07%)	new edges on	: 1025 (23.75%)
total execs	: 900k	total crashes	: 20 (19 unique)
exec speed	: 32.00/sec (slow!)	total tmouts	: 616 (226 unique)
fuzzing strategy yields		path geometry	
bit flips	: n/a, n/a, n/a	levels	: 4
byte flips	: n/a, n/a, n/a	pending	: 4066
arithmetics	: n/a, n/a, n/a	pend fav	: 410
known ints	: n/a, n/a, n/a	own finds	: 2204
dictionary	: n/a, n/a, n/a	imported	: 2110
havoc	: 1057/273k, 1162/182k	stability	: 100.00%
trim	: 0.19%/400k, n/a		

[cpu002:139%]

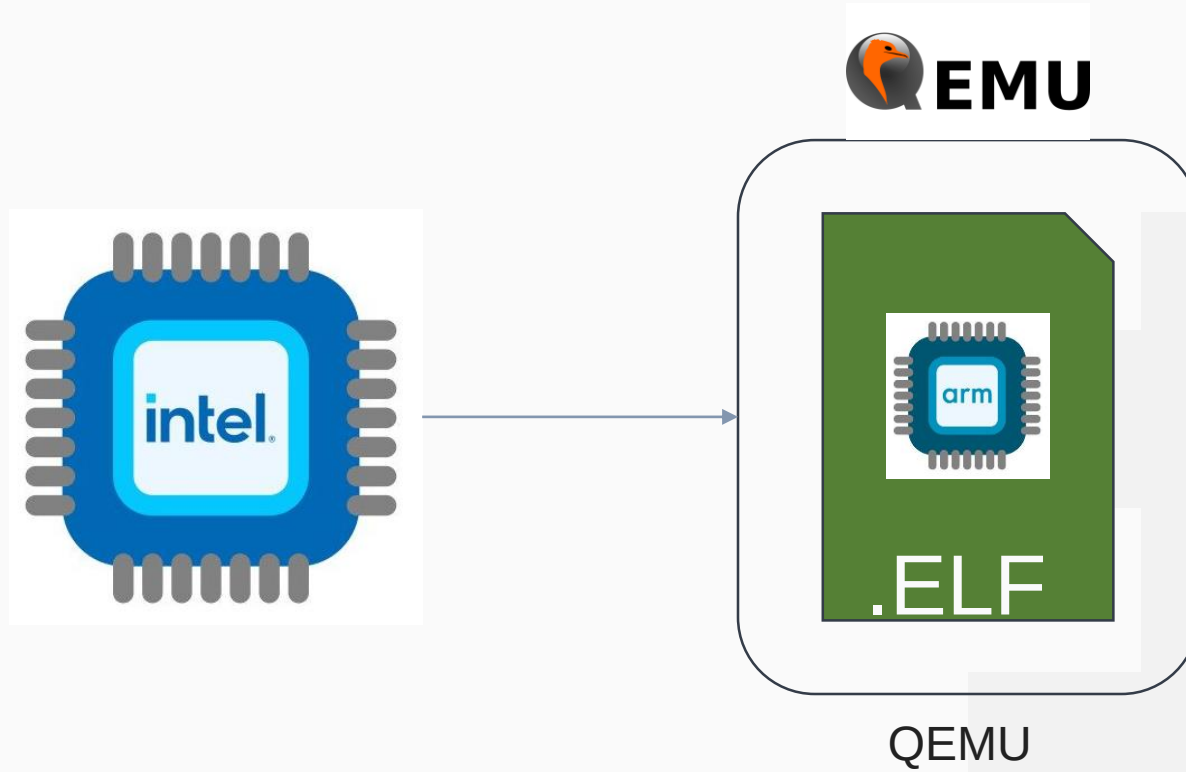
How ~[to fuzz]



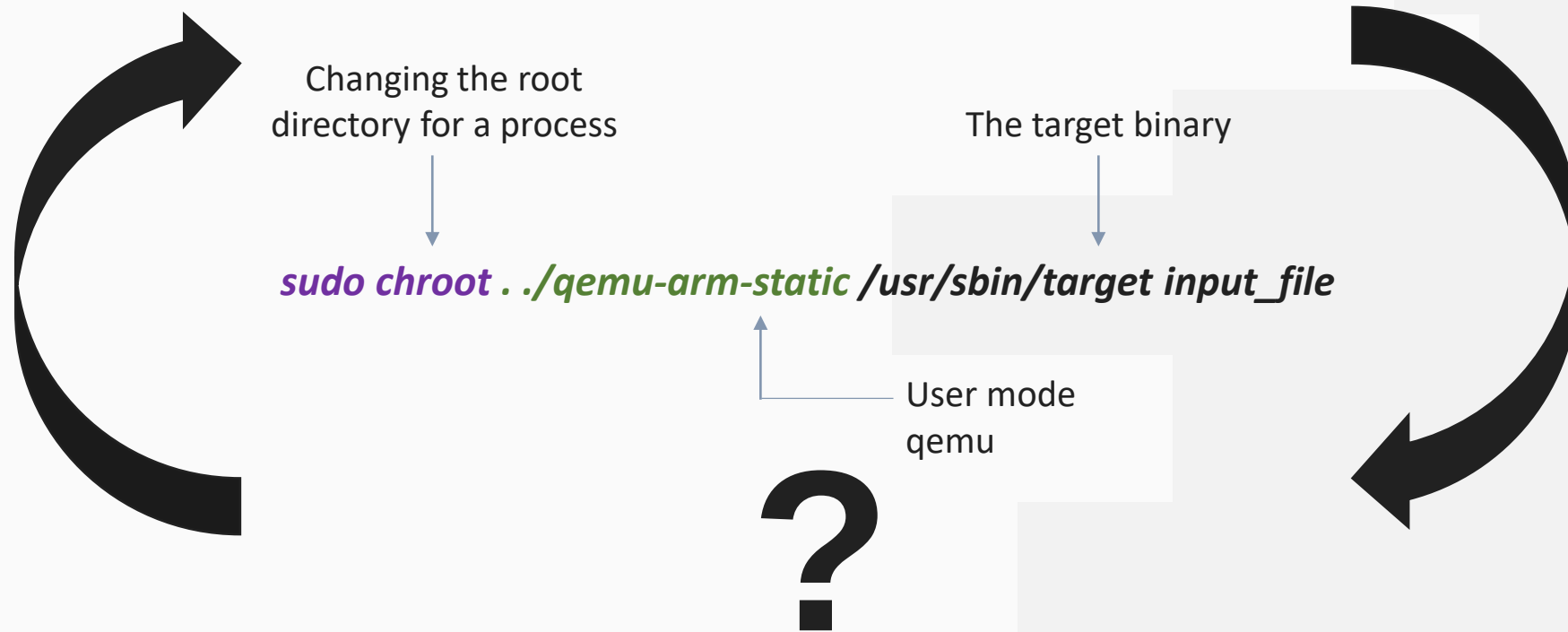
How ~[to fuzz]



How ~[to fuzz]



AFL The Great



AFL The Great

Telling AFL where to
find the libs



```
QEMU_LD_PREFIX=. afl-fuzz -Q -i in -out -- ./usr/sbin/target @@
```



QEMU mode

**ALRIGHT, BUT..
WHAT IF...**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int __attribute__((noinline)) complex_parsing(char *input){
    printf("[+] I've parsed the input <%s>\n", input);
    return 0;
}

int logging(char *input){
    FILE *logfile;
    logfile = fopen("logfile.txt", "a+");
    fprintf(logfile, "[INFO] - %s\n", input);
    return 0;
}

int main(int argc, char **argv){
    if (argc < 2){
        printf("No specified arguments\n");
        logging("No argument was provided");
        exit(1);
    }

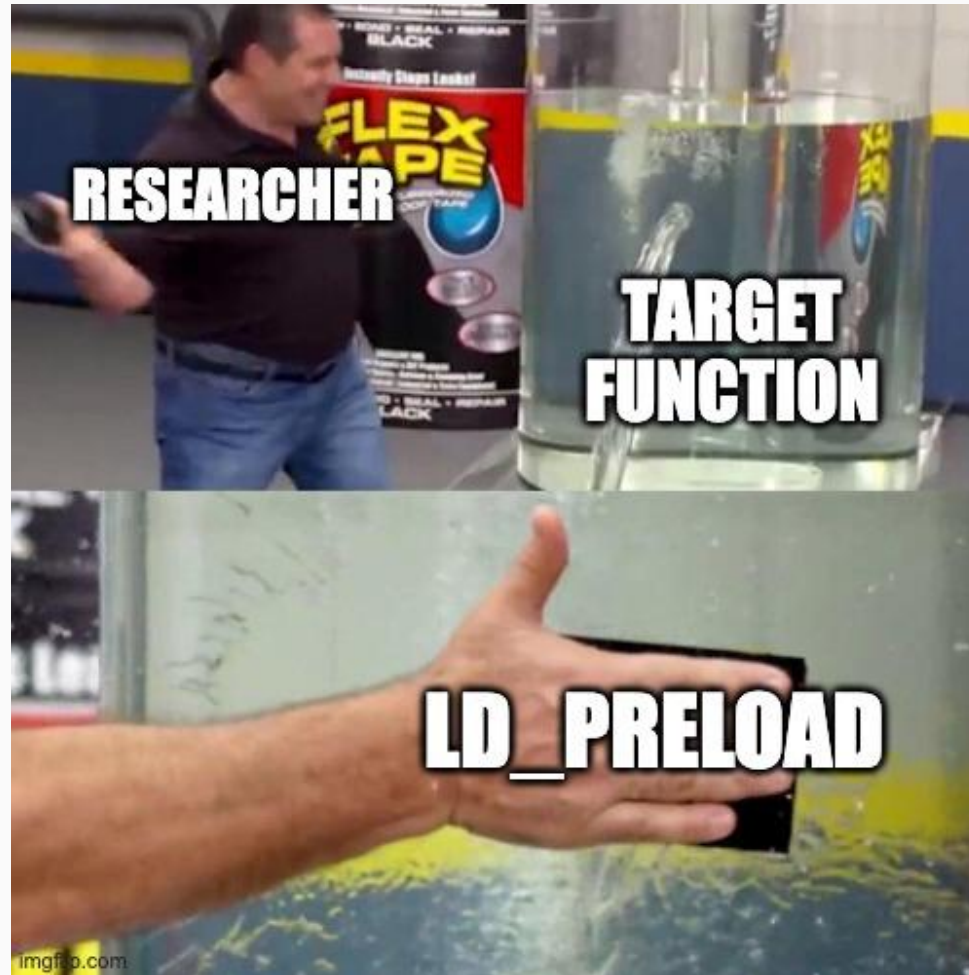
    size_t input_size = strlen(argv[1]);
    char *input = (char*)malloc(input_size + 1);
    if (input == NULL){
        printf("Couldnt allocate on heap\n");
        exit(1);
    }

    memcpy(input, argv[1], input_size);
    input[input_size] = '\0';

    printf("The input is %s\n", input);
    complex_parsing(input);
    logging(input);

    free(input);
    return 0;
}
```

AFL The Great



AFL The Great

```
#include <stdio.h>
#include <dlfcn.h>
#include <string.h>

static int (*original_main)(int, char **, char **);
typedef int(*complex_parsing_t)(char*);
char fuzz[100];

int over_main(int argc, char **argv, char **envp){
    printf("[-] executing main..\n");
    complex_parsing_t complex_parsing = (complex_parsing_t)<PUT_THE_ADDRESS_OF_THE_PARSING_FUNCTION_HERE>;
    sprintf(fuzz, "%s-%d", argv[1], i);
    complex_parsing(fuzz);
    return 0;//retval;
}

int __libc_start_main(int (*main)(int, char **, char **), int argc, char **argv, int (*init)(int, char**, char **), void
(*fini)(void), void (*rtld_fini)(void), void *stack_end){
    original_main = main;
    typeof(&__libc_start_main) original_libc_main = dlsym(RTLD_NEXT, "__libc_start_main");
    return original_libc_main(over_main, argc, argv, init, fini, rtld_fini, stack_end);
}
```

gcc -fPIC -shared -o bypass.so bypass.c -ldl -D_GNU_SOURCE

LD_PRELOAD=./bypass.so ./target input_string

AFL The Great

Telling AFL where to
find the libs



Equivalent to
LD_PRELOAD, but in
the context of *AFL*



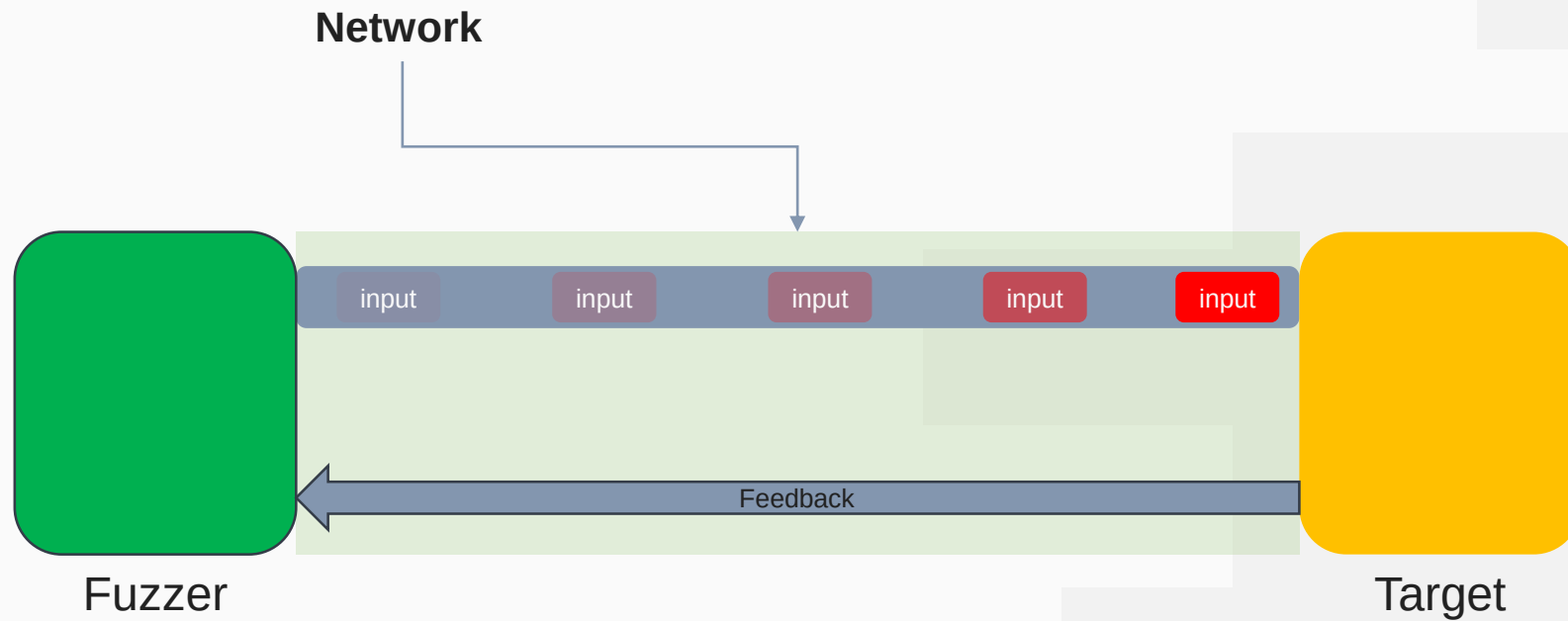
QEMU_LD_PREFIX=. AFL_PRELOAD=./bypass.so afl-fuzz -i in -o out -Q ./target



Our custom library

**RIGHT.. BUT..
WHAT IF (AGAIN)...**

AFL The Great



AFL The Great

 preeny Public

Stop ignoring 47 Fork 173 Star 1.6k

master 1 Branch 1 Tag Add file <> Code

 zardus relicense as bsd

2be4739 · last month 114 Commits

src	relicense as bsd	last month
test	added missing header, compilation was failing on wsl and ...	4 years ago
.gitignore	Add CMake (#46)	7 years ago
CMakeLists.txt	Add nowrite	4 years ago
LICENSE	adding BSD license	9 years ago
Makefile	Closes 52: only remove directories during cleanup	7 years ago
README.md	Make the section link clickable on GitHub	9 months ago
cmake-build-multiarch.sh	Add CMake (#46)	7 years ago
cmake-build.sh	Add CMake (#46)	7 years ago

README BSD-2-Clause license

About

Some helpful preload libraries for pwnning stuff.

- Readme
- BSD-2-Clause license
- Activity
- 1.6k stars
- 47 watching
- 173 forks

Report repository

Releases 1

 Initial tagged release. Latest
on Aug 18, 2016

Packages

No packages published

AFL The Great

Telling AFL where to
find the libs



Equivalent to
LD_PRELOAD, but in
the context of *AFL*



QEMU_LD_PREFIX=. AFL_PRELOAD=./libdesock_dup.so afl-fuzz -i in -o out -Q ./target

Our custom library



LET'S **SCALE** :))



```
american fuzzy lop ++4.02c [default] (...hfs-root/usr/sbin/jsonparse) [fast]
```

process timing		overall results	
run time : 0 days, 0 hrs, 0 min, 34 sec		cycles done : 0	
last new find : 0 days, 0 hrs, 0 min, 32 sec		corpus count : 4	
last saved crash : 0 days, 0 hrs, 0 min, 34 sec		saved crashes : 2	
last saved hang : none seen yet		saved hangs : 0	
cycle progress		map coverage	
now processing : 3.1 (75.0%)		map density : 0.07% / 0.07%	
runs timed out : 0 (0.00%)		count coverage : 1.35 bits/tuple	
stage progress		findings in depth	
now trying : splice 4		favored items : 3 (75.00%)	
stage execs : 18/73 (24.66%)		new edges on : 2 (50.00%)	
total execs : 5723		total crashes : 5487 (2 saved)	
exec speed : 166.5/sec		total tmouts : 0 (0 saved)	
fuzzing strategy yields		item geometry	
bit flips : disabled (default, enable with -D)		levels : 2	
byte flips : disabled (default, enable with -D)		pending : 0	
arithmetics : disabled (default, enable with -D)		pend fav : 0	
known ints : disabled (default, enable with -D)		own finds : 3	
dictionary : n/a		imported : 0	
havoc/splice : 5/3988, 0/1659		stability : 100.00%	
py/custom/rq : unused, unused, unused, unused			
trim/eff : 25.71%/25, disabled		[cpu000: 25%]	

AFL The Great

```
# starting the main fuzzer01 with -M
$ QEMU_LD_PREFIX=./squashfs-root/ afl-fuzz -M fuzzer01 -Q -i ./squashfs-root/in/ -o ./squashfs-root/sync_dir/ -- ./squashfs-root/usr/sbin/jsonparse @@

# adding a second secondary fuzzer02 with -S
$ QEMU_LD_PREFIX=./squashfs-root/ afl-fuzz -S fuzzer02 -Q -i ./squashfs-root/in/ -o ./squashfs-root/sync_dir/ -- ./squashfs-root/usr/sbin/jsonparse @@

# adding a third secondary fuzzer03 with -S
$ QEMU_LD_PREFIX=./squashfs-root/ afl-fuzz -S fuzzer03 -Q -i ./squashfs-root/in/ -o ./squashfs-root/sync_dir/ -- ./squashfs-root/usr/sbin/jsonparse @@

# invoking afl-whatsup command with simplified vue, to update every 2 seconds
$ watch -n 2 afl-whatsup -s ./squashfs-root/sync_dir/

... 
```

AFL The Great

```
american fuzzy lop ++4.02c {fuzzer01} (...fs-root/usr/sbin/jsonparse) [fast]
├─ process timing
│   run time : 0 days, 0 hrs, 10 min, 29 sec
│   last new find : 0 days, 0 hrs, 0 min, 24 sec
│   last saved crash : 0 days, 0 hrs, 10 min, 28 sec
│   last saved hang : none seen yet
├─ cycle progress
│   now processing : 18.0 (90.0%)
│   runs timed out : 0 (0.00%)
├─ stage progress
│   now trying : splice 10
│   stage execs : 124/128 (96.88%)
│   total execs : 92.2k
│   exec speed : 83.18/sec (slow!)
├─ fuzzing strategy yields
│   bit flips : disabled (default, enable with -D)
│   byte flips : disabled (default, enable with -D)
│   arithmetics : disabled (default, enable with -D)
│   known ints : disabled (default, enable with -D)
│   dictionary : n/a
│   havoc/splice : 5/37.5k, 7/54.3k
│   py/custom/rq : unused, unused, unused, unused
│   trim/eff : disabled, disabled
├─ map coverage
│   map density : 0.08% / 0.09%
│   count coverage : 2.53 bits/tuple
├─ findings in depth
│   favored items : 5 (25.00%)
│   new edges on : 5 (25.00%)
│   total crashes : 66.7k (2 saved)
│   total tmouts : 19 (0 saved)
├─ item geometry
│   levels : 3
│   pending : 3
│   pend fav : 1
│   own finds : 10
│   imported : 9
│   stability : 100.00%
└─ overall results
    cycles done : 20
    corpus count : 20
    saved crashes : 2
    saved hangs : 0
[cpu000: 25%]
```

```
american fuzzy lop ++4.02c {fuzzer02} (...fs-root/usr/sbin/jsonparse) [fast]
├─ process timing
│   run time : 0 days, 0 hrs, 7 min, 26 sec
│   last new find : 0 days, 0 hrs, 0 min, 0 sec
│   last saved crash : 0 days, 0 hrs, 7 min, 26 sec
│   last saved hang : none seen yet
├─ cycle progress
│   now processing : 3.25 (15.0%)
│   runs timed out : 0 (0.00%)
├─ stage progress
│   now trying : havoc
│   stage execs : 9/145 (6.21%)
│   total execs : 56.3k
│   exec speed : 77.75/sec (slow!)
├─ fuzzing strategy yields
│   bit flips : disabled (default, enable with -D)
│   byte flips : disabled (default, enable with -D)
│   arithmetics : disabled (default, enable with -D)
│   known ints : disabled (default, enable with -D)
│   dictionary : n/a
│   havoc/splice : 3/21.9k, 4/34.2k
│   py/custom/rq : unused, unused, unused, unused
│   trim/eff : disabled, disabled
├─ map coverage
│   map density : 0.06% / 0.09%
│   count coverage : 2.53 bits/tuple
├─ findings in depth
│   favored items : 5 (25.00%)
│   new edges on : 5 (25.00%)
│   total crashes : 38.2k (2 saved)
│   total tmouts : 24 (0 saved)
├─ item geometry
│   levels : 2
│   pending : 1
│   pend fav : 0
│   own finds : 5
│   imported : 14
│   stability : 100.00%
└─ overall results
    cycles done : 25
    corpus count : 20
    saved crashes : 2
    saved hangs : 0
[cpu001: 25%]
```

```
american fuzzy lop ++4.02c {fuzzer03} (...fs-root/usr/sbin/jsonparse) [fast]
├─ process timing
│   run time : 0 days, 0 hrs, 0 min, 29 sec
│   last new find : 0 days, 0 hrs, 0 min, 24 sec
│   last saved crash : 0 days, 0 hrs, 0 min, 29 sec
│   last saved hang : none seen yet
├─ cycle progress
│   now processing : 18.1 (94.7%)
│   runs timed out : 0 (0.00%)
├─ stage progress
│   now trying : splice 6
│   stage execs : 18/55 (32.73%)
│   total execs : 3596
│   exec speed : 79.98/sec (slow!)
├─ fuzzing strategy yields
│   bit flips : disabled (default, enable with -D)
│   byte flips : disabled (default, enable with -D)
│   arithmetics : disabled (default, enable with -D)
│   known ints : disabled (default, enable with -D)
│   dictionary : n/a
│   havoc/splice : 3/1778, 0/1475
│   py/custom/rq : unused, unused, unused, unused
│   trim/eff : 87.88%/131, disabled
├─ map coverage
│   map density : 0.08% / 0.09%
│   count coverage : 2.46 bits/tuple
├─ findings in depth
│   favored items : 4 (21.05%)
│   new edges on : 5 (26.32%)
│   total crashes : 3146 (2 saved)
│   total tmouts : 6 (0 saved)
├─ item geometry
│   levels : 2
│   pending : 13
│   pend fav : 0
│   own finds : 1
│   imported : 17
│   stability : 100.00%
└─ overall results
    cycles done : 0
    corpus count : 19
    saved crashes : 2
    saved hangs : 0
[cpu002: 25%]
```

Every 2.0s: afl-whatsup -s squashfs-root/sync_dir/

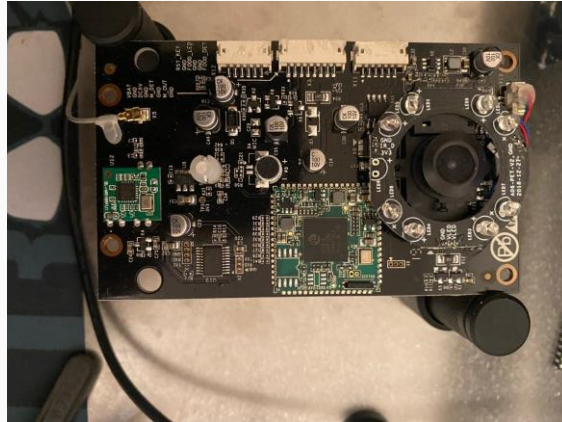
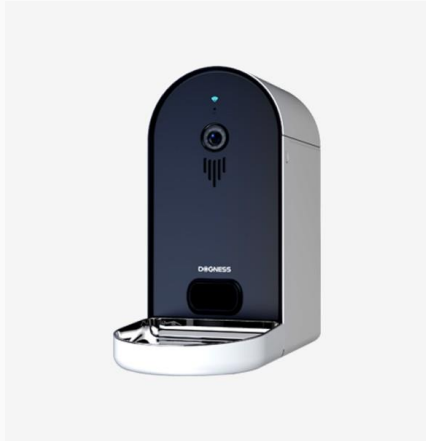
/usr/local/bin/afl-whatsup status check tool for afl-fuzz by Michal Zalewski

Summary stats

=====

```
Fuzzers alive : 3
Dead or remote : 1 (excluded from stats)
Total run time : 17 minutes, 1 seconds
Total execs : 141 thousands
Cumulative speed : 273 execs/sec
Average speed : 91 execs/sec
Pending items : 2 faves, 4 total
Pending per fuzzer : 0 faves, 1 total (on average)
Crashes saved : 4
Cycles without finds : 0/0/0/0
Time without finds : 27 seconds
```

A straightforward example



<https://securelist.com/smart-pet-feeder-vulnerabilities/110028/>

A straightforward example

```
main 650, packet_length:5
main 685, packet_length:5
read(124) on read packet time:7!
Packet Header: 0x30...
recv topic:voice/shootreq/1 msg:{
  "cmd": "shootreq",
  "feedweight": "1",
  "audioindex": "0",
  "rand": "0"
}
device-->publish:{"cmd":"shootresp","devid":"1","result":"0", "rand":"0" }
keep alive send ret:1
main 685, packet_length:124
read(2) on read packet time:0!
Packet Header: 0xd0...
recv server keep alive !!
```

```
{
*((_BYTE *)mqtt_topic + mqtt_parse_pub_topic((int)packet_buffer, mqtt_topic, v40)) = 0;
*((_BYTE *)mqtt_message + mqtt_parse_publish_msg(packet_buffer, mqtt_message)) = 0;
printf("recv topic:%s msg:%s\n", (const char *)mqtt_topic, (const char *)mqtt_message);
json_object = cJSON_Parse((int)mqtt_message);
json_object2 = json_object;
if ( !json_object )
{
puts("parser error!!");
continue;
}
json_rand = *(_DWORD *) (cJSON_GetObjectItem(json_object, "rand") + 16);
json_cmd = *(const char **) (cJSON_GetObjectItem(json_object2, "cmd") + 16);
if ( !strcmp(json_cmd, "shootreq") )
{
json_feedweight = *(const char **) (cJSON_GetObjectItem(json_object2, "feedweight") + 16);
json_autoindex = *(const char **) (cJSON_GetObjectItem(json_object2, "audioindex") + 16);
strcpy(json_formatstring, "{ \"cmd\": \"shootresp\", \"devid\": \"%s\", \"result\": \"0\", \"rand\": \"%s\" }");
memset(&json_formatstring[60], 0, 0x44u);
memset(full_json_cmd, 0, 0x80u);
sprintf((char *)full_json_cmd, json_formatstring, &[8], json_rand);
printf("device-->publish:%s\n", (const char *)full_json_cmd);
memset(cmd_to_send, 0, 0x18u);
cmd_to_send[2] = 10;
cmd_to_send[0] = 1;
cmd_to_send[1] = 1;
cmd_to_send[2] = atoi(json_feedweight);
cmd_to_send[3] = atoi(json_autoindex);
pShareMemWrite(shared_mem_id0, 0, cmd_to_send, 24);
v48 = v82;
full_json_cmd2 = (char *)full_json_cmd;

v23 = mqtt_publish((int)&broker, v48, full_json_cmd2, 0);
}
```

<https://securelist.com/smart-pet-feeder-vulnerabilities/110028/>

A straightforward example

```
arm-buildroot-linux-uclibcgnueabi-gcc -Wall -fPIC -shared -o hook.so  
hook.c -ldl -D_GNU_SOURCE
```

```
LD_PRELOAD=./hook.so QEMU_LD_PREFIX=. afl-fuzz -t 2000 -Q -i in -o out  
-- ./usr/sbin/mqtt_alex @@
```

<https://securelist.com/smart-pet-feeder-vulnerabilities/110028/>

A straightforward example

```
Packet Header: 0xd0...  
receive server keep alive !!  
main 685, packet_length:2  
read(102) on read packet time:4!  
Packet Header: 0x30...  
receive topic:voice/shootreq/10 msg:{  
    "cmd": "shootreq",  
    "feedweight": "1",  
    "rand": "0"  
}  
Segmentation fault
```

<https://securelist.com/smart-pet-feeder-vulnerabilities/110028/>

Conclusion

Cool stuffs we haven't mentioned

libAFL

Snapshot fuzzing

Structure fuzzing

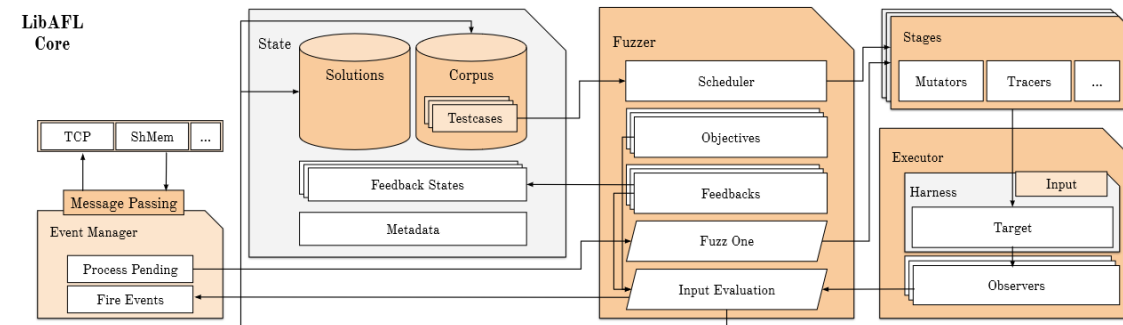


Figure 1: LibAFL Core architecture. Links are a representation of a non-comprehensive picture of the interactions.

<https://github.com/AFLplusplus/LibAFL>

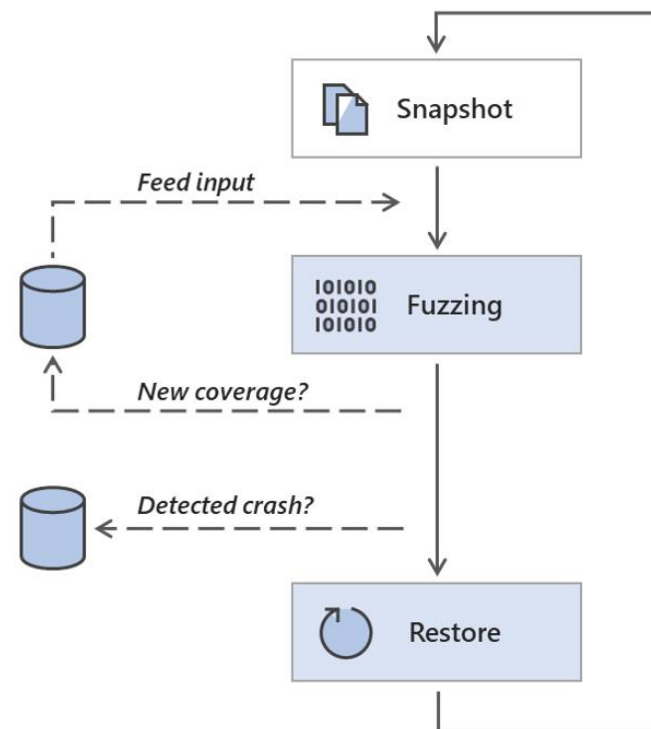
Conclusion

Cool stuffs we haven't mentioned

libAFL

Snapshot fuzzing

Structure fuzzing



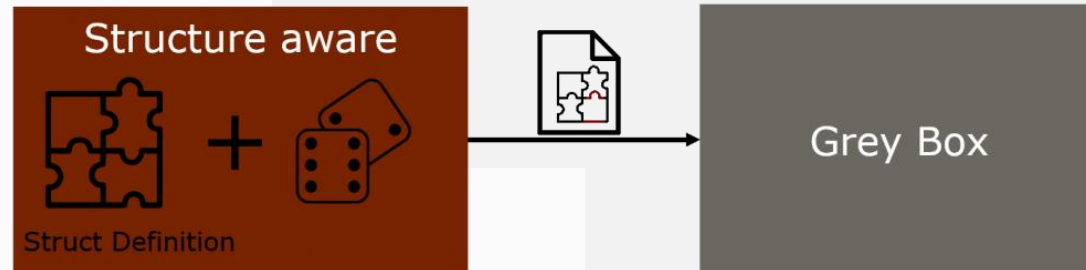
Conclusion

Cool stuffs we haven't mentioned

libAFL

Snapshot fuzzing

Structure fuzzing



Contacts

Roland Sako

Security Researcher; Teacher

✉ roland.sako@gmail.com

in linkedin.com/in/rolandsako/

✂ @rosako

www.winsl0w.io

trainings@winsl0w.io



A few code samples

 github.com/winsl0w-io/PHTCairo2025