Hacking IOT devices using SPI flash

HiTB 2024 – Hardware Village

Security
Research
Labs

Agenda

1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

What is SPI flash memory?

What is it?

SPI flash memory, also known as flash memory, has become widely used in the embedded industry and is commonly used for storage and data transfer in portable devices.

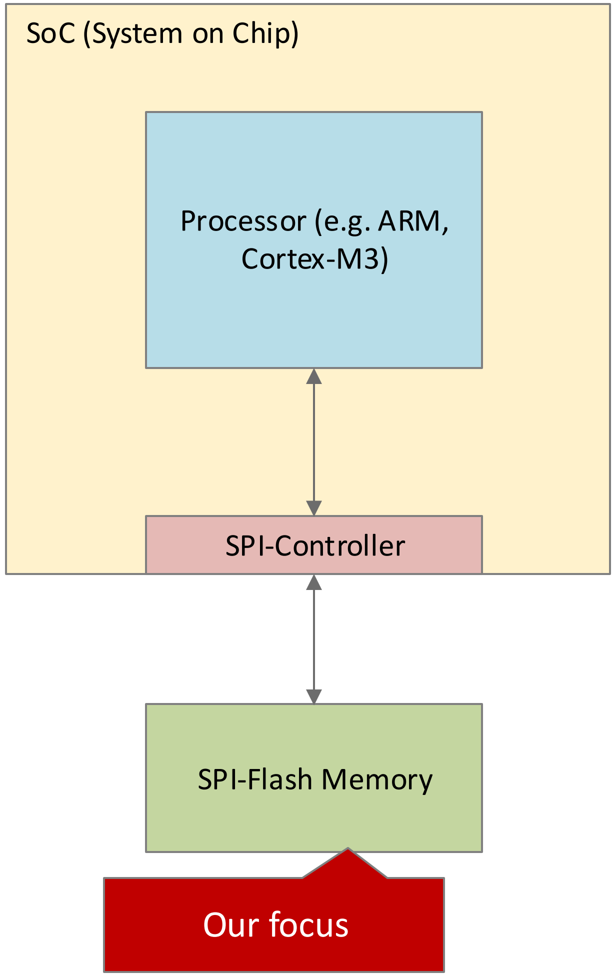
Where's it found?

Common devices include phones, tablets, and media players, as well as industrial devices such as security systems and medical products.

How is it used in IOT devices?

The flash memory is non-volatile, meaning that it retains its stored data when the device is powered down.

Typically, a SOC will contain a first-stage bootloader which will invoke a second-stage bootloader (U-boot) stored in the flash memory. Additional filesystems are also stored in flash memory, which normally contain vendor binaries and configuration scripts for device operation.



Agenda

1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

Identifying flash memory

Physical characteristics

Serial Flash Memory is available in many packages. One of the most common packages is SOP8 (see right).

Vendor identification

Look for the logo, common vendors in the IOT space are GigaDevice, Winbond, Puya.

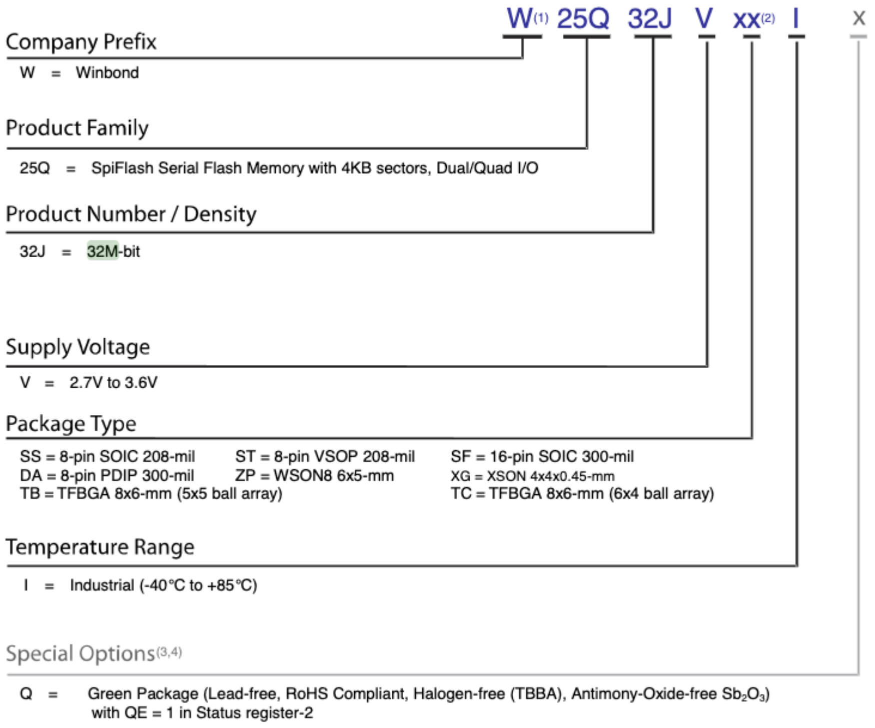
Part numbers

Vendor names/logos as well as part numbers are typically printed on the top-side.

Datasheets

Using Open Source Intelligence (OSINT) aka Google, we can lookup the part number of the packages to locate their data sheet.

Top tip: Check out alldatasheet.com



W[1] 25Q 32J V xx[2] I X

Company Prefix

W = Winbond

Product Family

25Q = SpiFlash Serial Flash Memory with 4KB sectors, Dual/Quad I/O

Product Number / Density

32J = 32M-bit

Supply Voltage

V = 2.7V to 3.6V

Package Type

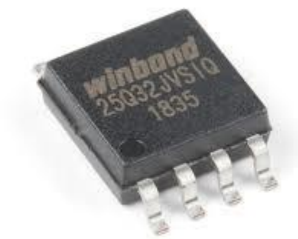
SS = 8-pin SOIC 208-mil
DA = 8-pin PDIP 300-mil
TB = TFBGA 8x6-mm (5x5 ball array)
ST = 8-pin VSOP 208-mil
ZP = WSON8 6x5-mm
SF = 16-pin SOIC 300-mil
XG = XSON 4x4x0.45-mm
TC = TFBGA 8x6-mm (6x4 ball array)

Temperature Range

I = Industrial (-40°C to +85°C)

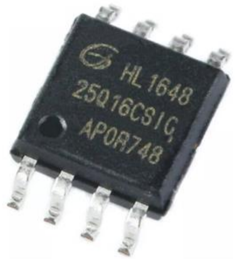
Special Options[3,4]

Q = Green Package (Lead-free, RoHS Compliant, Halogen-free (TBBA), Antimony-Oxide-free Sb_2O_3) with QE = 1 in Status register-2



25Q32JVSIQ

family size (m-bit)

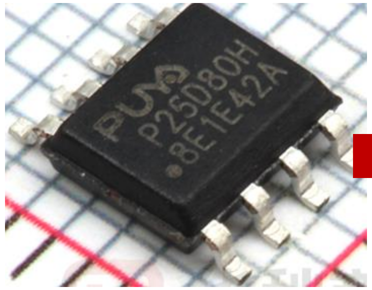


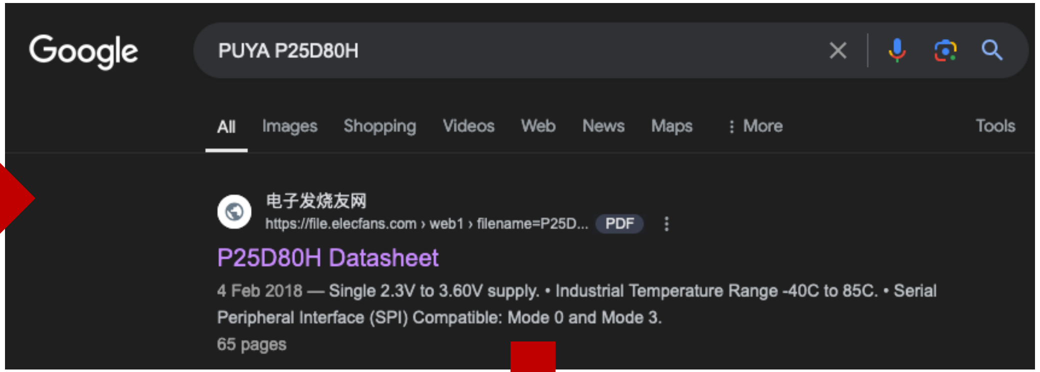
25Q16CSIG

family size (m-bit)

Identifying flash memory – Flash memory datasheets

Flash memory specifications can *often* be found with a quick Google **search for the product part number printed on to the surface of the chip**



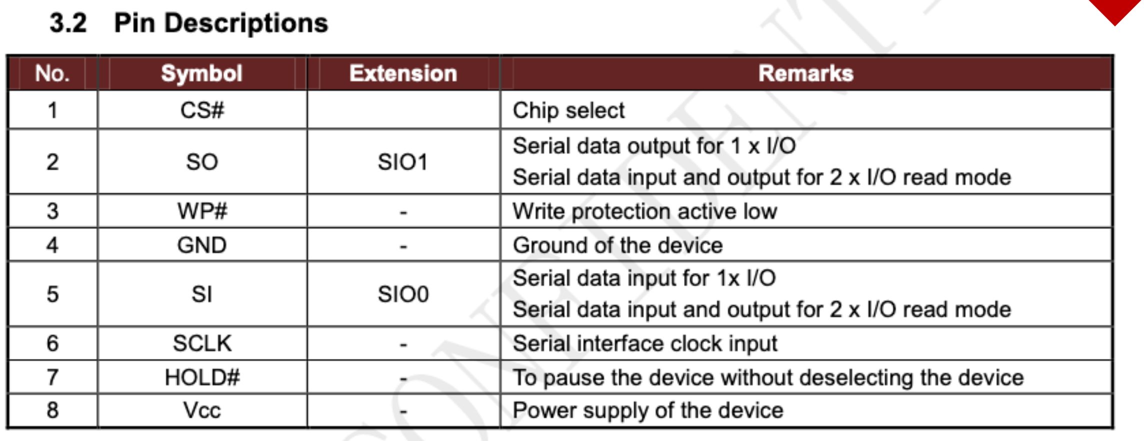


电子发烧友网
https://file.elecfans.com › web1 › filename=P25D...

P25D80H Datasheet

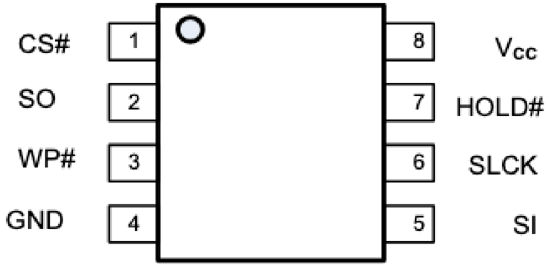
4 Feb 2018 — Single 2.3V to 3.60V supply. • Industrial Temperature Range -40C to 85C. • Serial Peripheral Interface (SPI) Compatible: Mode 0 and Mode 3.

65 pages



3.2 Pin Descriptions

No.	Symbol	Extension	Remarks
1	CS#		Chip select
2	SO	SIO1	Serial data output for 1 x I/O Serial data input and output for 2 x I/O read mode
3	WP#	-	Write protection active low
4	GND	-	Ground of the device
5	SI	SIO0	Serial data input for 1x I/O Serial data input and output for 2 x I/O read mode
6	SCLK	-	Serial interface clock input
7	HOLD#	-	To pause the device without deselecting the device
8	Vcc	-	Power supply of the device



8-PIN SOP (150mil/208mil) and TSSOP

Source:
https://file.elecfans.com/web1/M00/9E/D4/pIYBAF06m9-AAQD1ABreaXbSNRY789.pdf

Agenda

1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

Target Selection

Target Selection

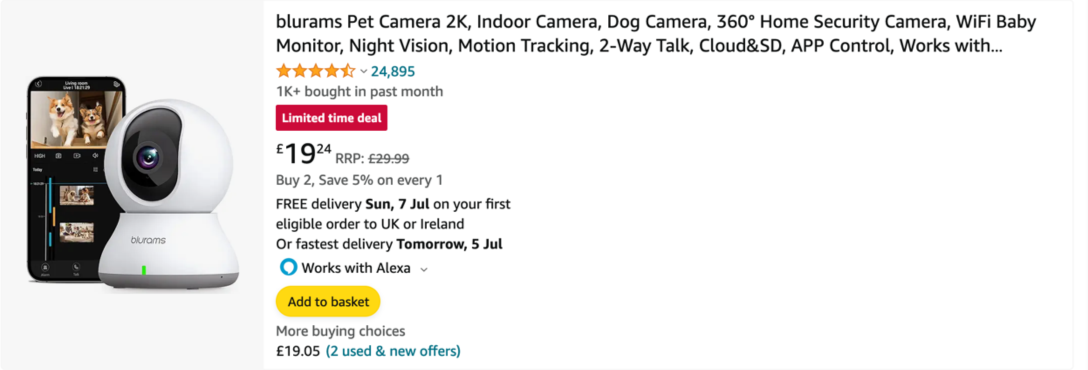
IPcams are a good place to start when getting into IOT hacking as they are very affordable and offer a large attack surface

Common Attack Surface

- UART serial
- SPI flash memory
- Network traffic between IPcam and cloud
- Network traffic between Mobile app and cloud
- Network traffic between IPcam and mobile app
- Mobile app decompilation

Before you buy

Check for FCC submissions. Vendors selling wireless capable products in the the US are required to register their products with FCC. These normally include submitting **user manuals, technical specifications and teardown images of the device internals**. These are very helpful creating a threat model to determine the attack surface of the device and whether it is a suitable candidate for security research.



blurams Pet Camera 2K, Indoor Camera, Dog Camera, 360° Home Security Camera, WiFi Baby Monitor, Night Vision, Motion Tracking, 2-Way Talk, Cloud&SD, APP Control, Works with...

★★★★☆ ⌄ 24,895

1K+ bought in past month

Limited time deal

£19.24 RRP: ~~£29.99~~

Buy 2, Save 5% on every 1

FREE delivery **Sun, 7 Jul** on your first eligible order to UK or Ireland

Or fastest delivery **Tomorrow, 5 Jul**

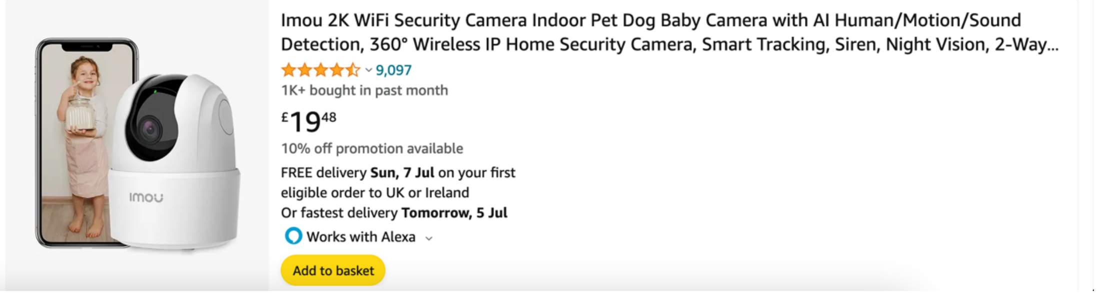
○ Works with Alexa ⌄

Add to basket

More buying choices

£19.05 (2 used & new offers)

FCC documents: https://fccid.io/2ASAQ-A31



Imou 2K WiFi Security Camera Indoor Pet Dog Baby Camera with AI Human/Motion/Sound Detection, 360° Wireless IP Home Security Camera, Smart Tracking, Siren, Night Vision, 2-Way...

★★★★☆ ⌄ 9,097

1K+ bought in past month

£19.48

10% off promotion available

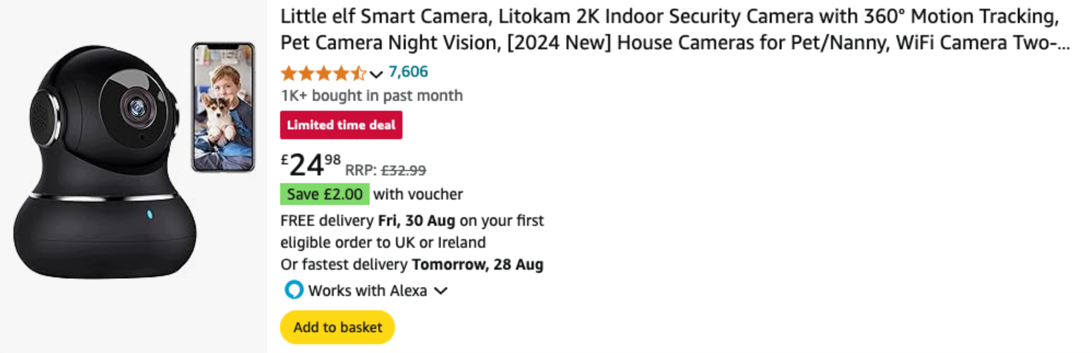
FREE delivery **Sun, 7 Jul** on your first eligible order to UK or Ireland

Or fastest delivery **Tomorrow, 5 Jul**

○ Works with Alexa ⌄

Add to basket

FCC documents: https://fcc.report/FCC-ID/2AVYF-IPC-TAX2C/



Little elf Smart Camera, Litokam 2K Indoor Security Camera with 360° Motion Tracking, Pet Camera Night Vision, [2024 New] House Cameras for Pet/Nanny, WiFi Camera Two-...

★★★★☆ ⌄ 7,606

1K+ bought in past month

Limited time deal

£24.98 RRP: ~~£32.99~~

Save £2.00 with voucher

FREE delivery **Fri, 30 Aug** on your first eligible order to UK or Ireland

Or fastest delivery **Tomorrow, 28 Aug**

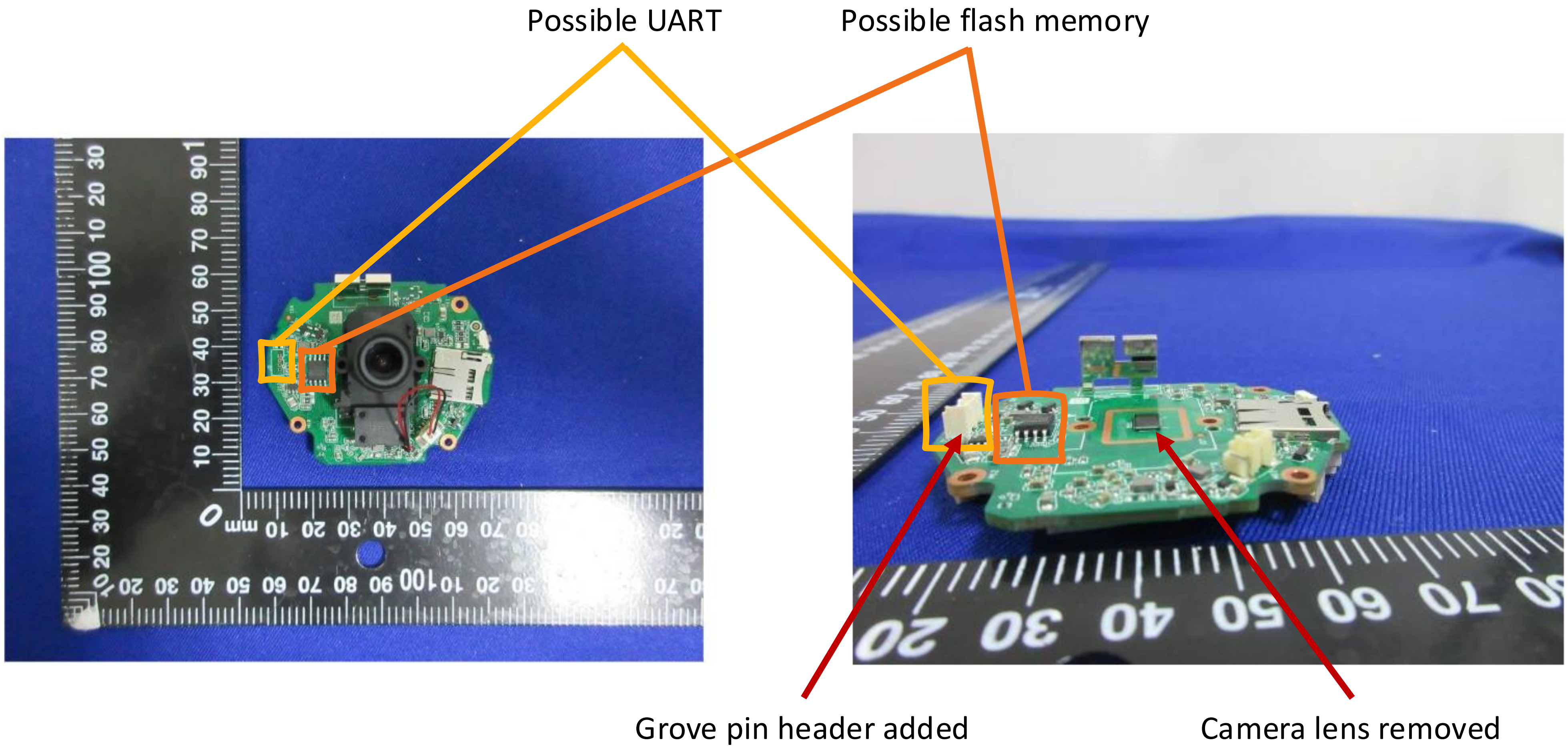
○ Works with Alexa ⌄

Add to basket

FCC documents: https://fccid.io/2AK47LF-P1

Target Selection – FCC documents

From FCC submitted documents it is possible to gain insights as to the possible attack surfaces present on the target device ahead of purchase



Possible UART

Possible flash memory

Grove pin header added

Camera lens removed

Source: https://fcc.report/FCC-ID/2AVYF-IPC-TAX2C/5016580

Agenda

1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

Dumping flash memory – tooling requirements

SPI flash can be read/written from devices with an SPI interface, these range in price and availability. If you are on a budget, you can use existing hardware you have lying around or purchase specialist hardware to increase reliability.

Development Boards

Arduino

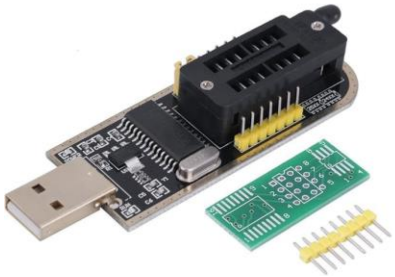
Raspberry Pi

Multi-tools

Bus Pirate

Flipper Zero

Dedicated Programmers



CH341A Programmer

Xgecu T56

Dumping flash memory – connecting to SOP8 chip

SOP8 Test clip / PCB probes

Pros

Flash memory can be extracted without removing the chip from the board

Cons

Reading (worse writing) via the chip can be unreliable as the chip reader may power up the SOC via the VCC rail

SOP8 to DIP8 socket

Pros

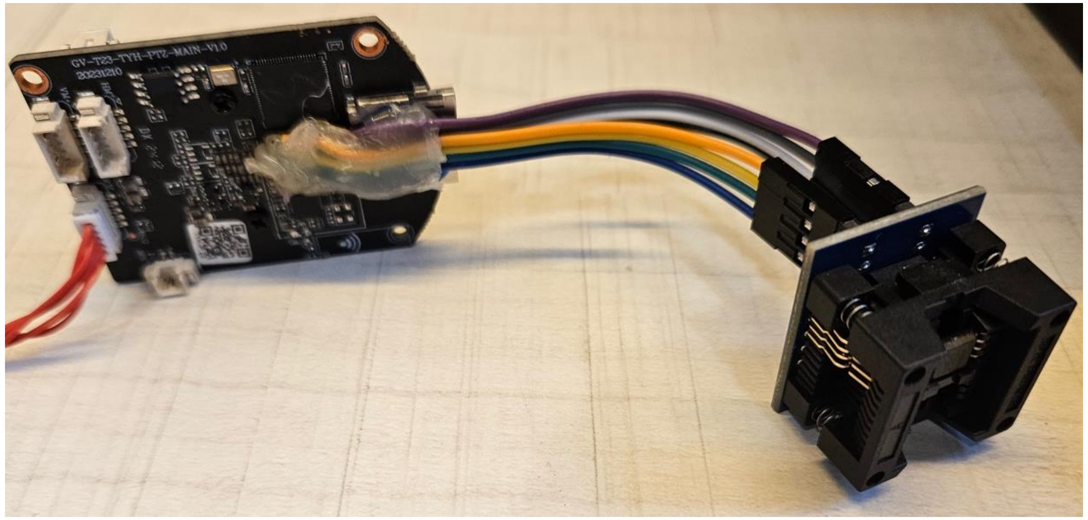
Flash memory can be read/written with high success rate

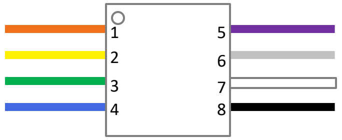
Cons

Chip needs to be removed from board with hot air/soldering iron

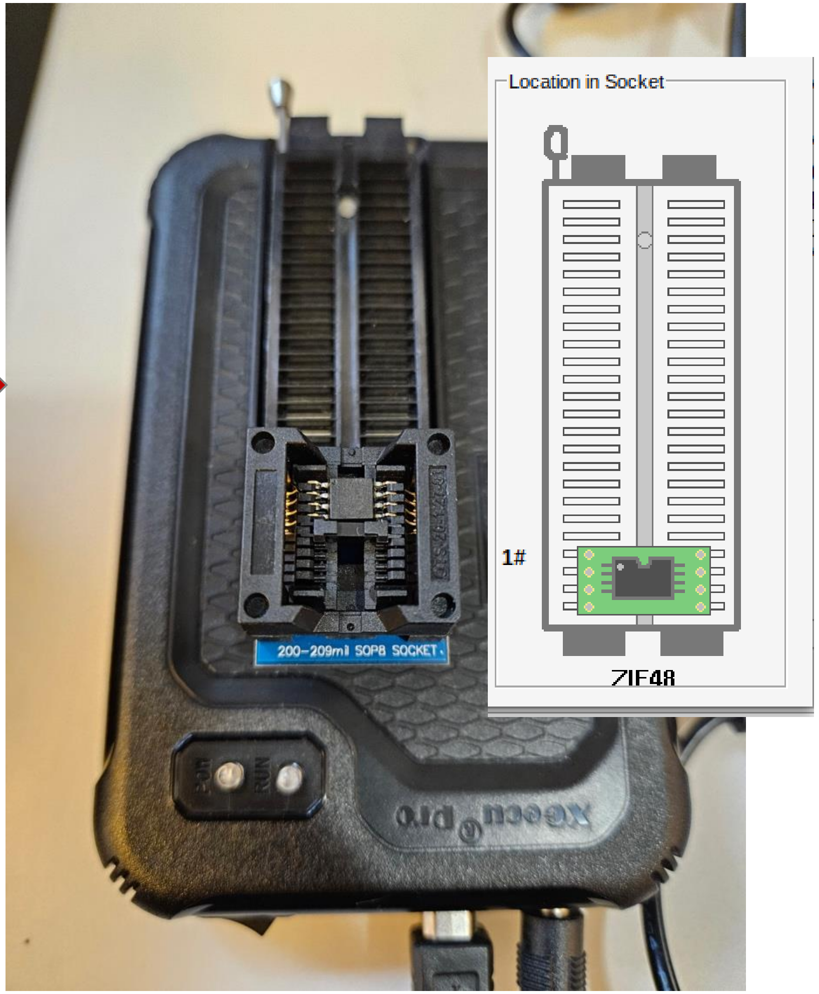
Dumping flash memory – attaching the chip to the programmer device

To remove the requirement of desoldering the chip, a wiring harness been soldered to the pads and connect to a SOP8 > DIP8 socket





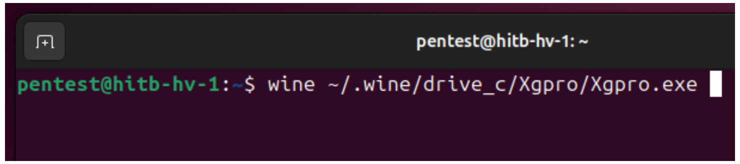
Disconnect DIP8 socket from wiring harness and place in the Xgeco programmer



Dumping flash memory – Detecting the SOP8 chip in Xgpro

Launch Xgpro software

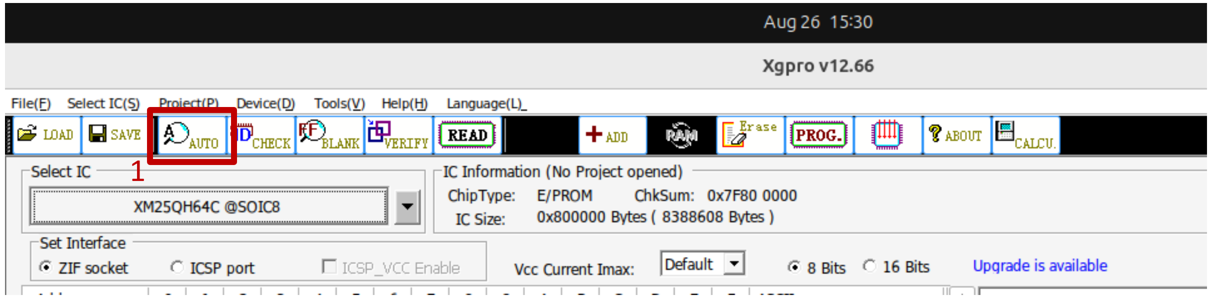
Using wine, we can run the Windows binary on Linux



```
pentest@hitb-hv-1: ~
pentest@hitb-hv-1:~$ wine ~/.wine/drive_c/Xgpro/Xgpro.exe
```

Detect the flash memory chip

1. Click “Auto” on the top menu
2. Click “Detect” in the Auto Search window
3. Select the highlighted Model
4. Click the “Select” button



Select IC: XM25QH64C @SOIC8

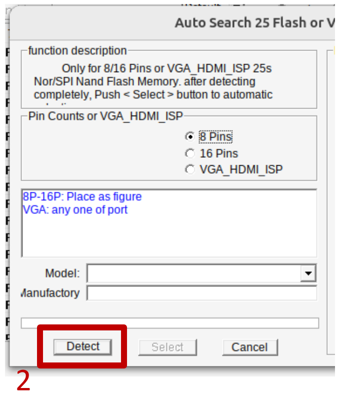
IC Information (No Project opened)
ChipType: E/PROM ChkSum: 0x7F80 0000
IC Size: 0x800000 Bytes (8388608 Bytes)

Set Interface: ● ZIF socket ○ ICSP port □ ICSP_VCC Enable

Vcc Current Imax: Default

● 8 Bits ○ 16 Bits

Upgrade is available

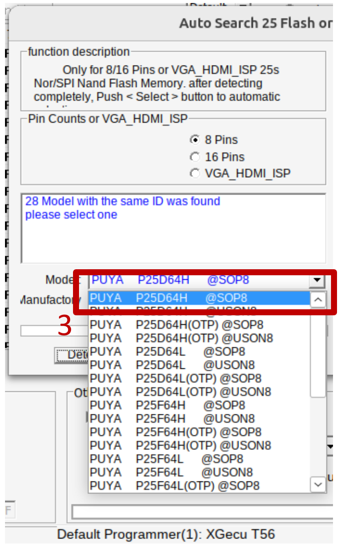


function description: Only for 8/16 Pins or VGA_HDMI_ISP 25s Nor/SPI Nand Flash Memory. after detecting completely, Push < Select > button to automatic

Pin Counts or VGA_HDMI_ISP: ● 8 Pins ○ 16 Pins ○ VGA_HDMI_ISP

8P-16P: Place as figure
VGA: any one of port

Model:
Manufactory



function description: Only for 8/16 Pins or VGA_HDMI_ISP 25s Nor/SPI Nand Flash Memory. after detecting completely, Push < Select > button to automatic

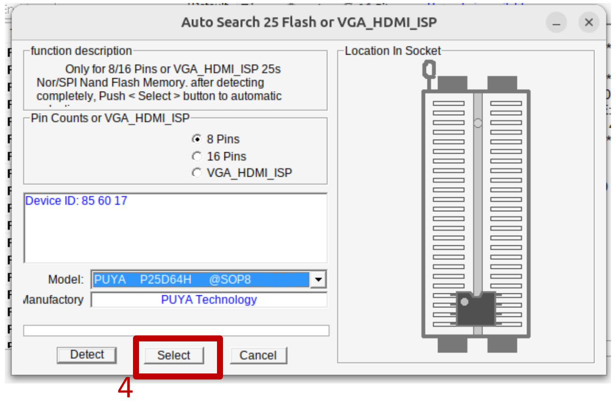
Pin Counts or VGA_HDMI_ISP: ● 8 Pins ○ 16 Pins ○ VGA_HDMI_ISP

28 Model with the same ID was found
please select one

Model: PUYA P25D64H @SOP8

```
PUYA   P25D64H     @SOP8
PUYA   P25D64H     @USON8
PUYA   P25D64H(OTP) @SOP8
PUYA   P25D64H(OTP) @USON8
PUYA   P25D64L     @SOP8
PUYA   P25D64L     @USON8
PUYA   P25D64L(OTP) @SOP8
PUYA   P25D64L(OTP) @USON8
PUYA   P25F64H     @SOP8
PUYA   P25F64H     @USON8
PUYA   P25F64H(OTP) @SOP8
PUYA   P25F64H(OTP) @USON8
PUYA   P25F64L     @SOP8
PUYA   P25F64L     @USON8
PUYA   P25F64L(OTP) @SOP8
```

Default Programmer(1): XGecu T56



function description: Only for 8/16 Pins or VGA_HDMI_ISP 25s Nor/SPI Nand Flash Memory. after detecting completely, Push < Select > button to automatic

Pin Counts or VGA_HDMI_ISP: ● 8 Pins ○ 16 Pins ○ VGA_HDMI_ISP

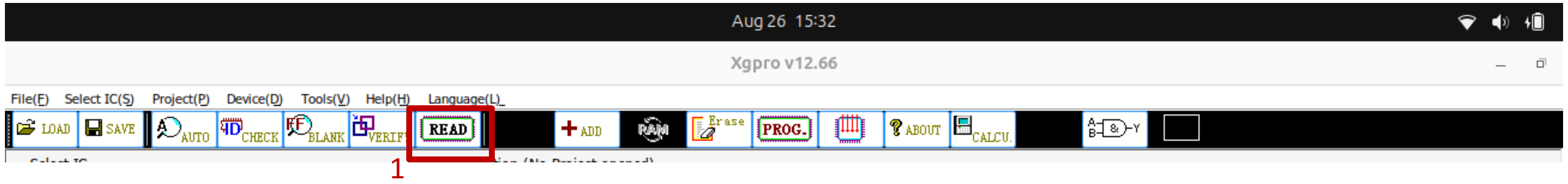
Device ID: 85 60 17

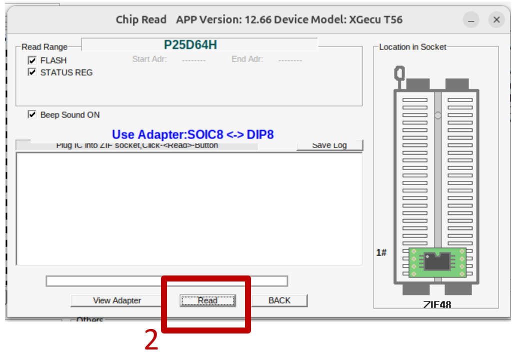
Model: PUYA P25D64H @SOP8
Manufactory: PUYA Technology

Dumping flash memory – Reading & saving the contents of the SOP8 flash chip

Read the flash memory

1. Click “READ” on the top menu
2. Click “Read” in the Chip Read window
3. Click “BACK” once read is complete

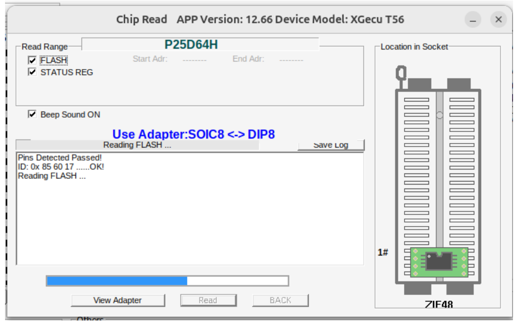




☑ FLASH
☑ STATUS REG
☑ Beep Sound ON

Use Adapter:SOIC8 <-> DIP8

Plug IC into ZIF socket,Click-<Read>-Button

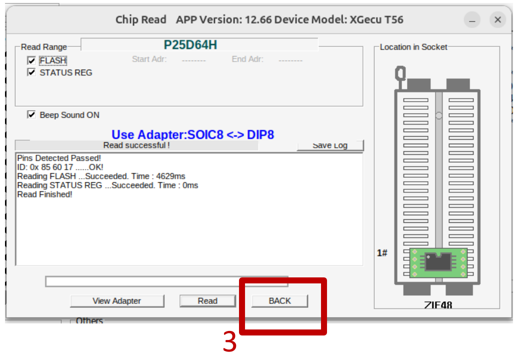


☑ FLASH
☑ STATUS REG
☑ Beep Sound ON

Use Adapter:SOIC8 <-> DIP8

Reading FLASH ...

```
Pins Detected Passed!
ID: 0x 85 60 17 ......OK!
Reading FLASH ...
```



☑ FLASH
☑ STATUS REG
☑ Beep Sound ON

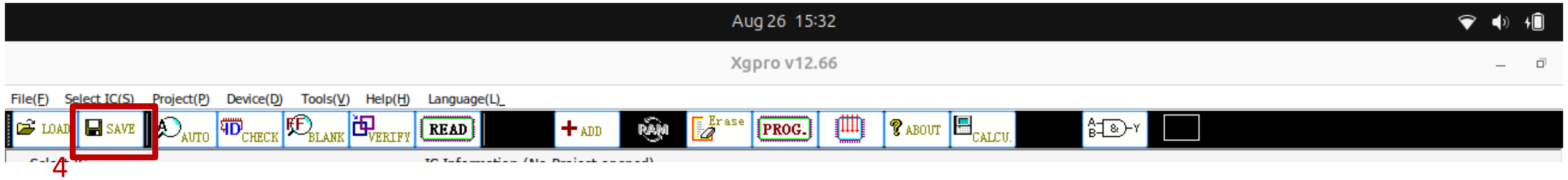
Use Adapter:SOIC8 <-> DIP8

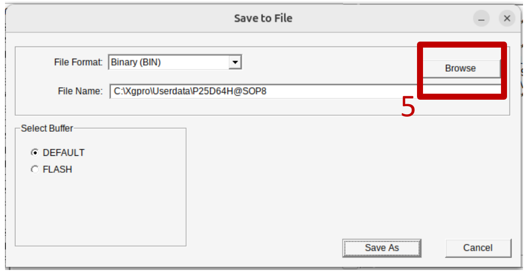
Read successful !

```
Pins Detected Passed!
ID: 0x 85 60 17 ......OK!
Reading FLASH ...Succeeded. Time : 4629ms
Reading STATUS REG ...Succeeded. Time : 0ms
Read Finished!
```

Read the flash memory

4. Click “SAVE” on the top menu
5. Click “Browse” in the Save to file window
6. Locate your preferred location and click “Save”
7. Click “Save As”
8. Click “OK”



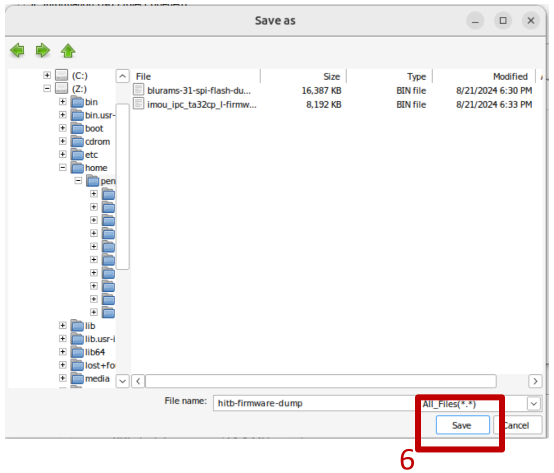


File Format: Binary (BIN)

File Name: C:\Xgpro\Userdata\P25D64H@SOP8

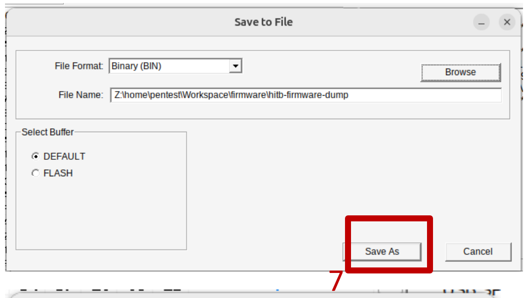
Select Buffer

● DEFAULT
○ FLASH



File	Size	Type	Modified
blurams-31-spi-flash-du...	16,387 KB	BIN file	8/21/2024 6:30 PM
imou_ipc_ta32cp_l-firmw...	8,192 KB	BIN file	8/21/2024 6:33 PM

File name: hitb-firmware-dump All_Files(*.*)

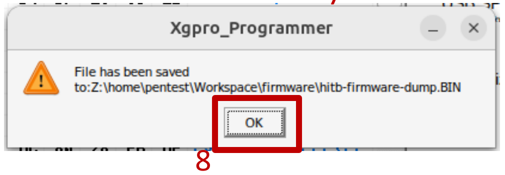


File Format: Binary (BIN)

File Name: Z:\home\pentest\Workspace\firmware\hitb-firmware-dump

Select Buffer

● DEFAULT
○ FLASH



File has been saved to:Z:\home\pentest\Workspace\firmware\hitb-firmware-dump.BIN

Agenda

1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

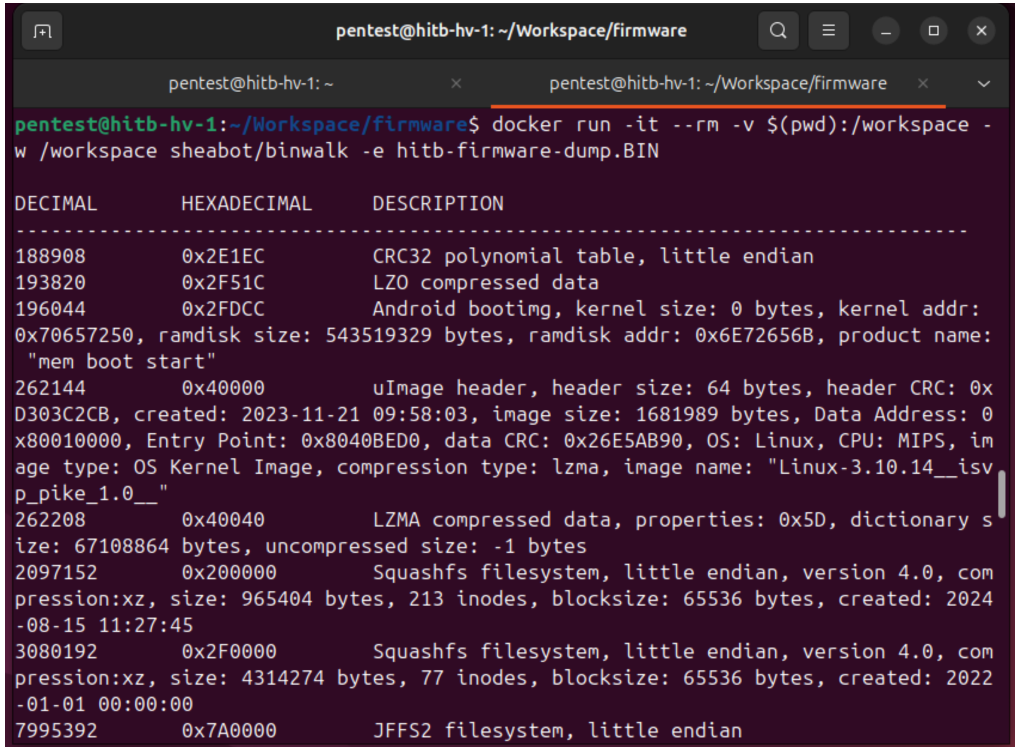
Modifying filesystems – extracting flash contents and modifying files

Extract contents

Using the binwalk[1] firmware analysis tool we can extract the contents of the flash memory dump.

binwalk attempts to calve out different areas of the binary file into filesystem sections.

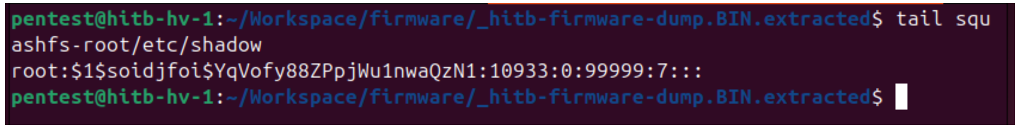
Our focus should be on SquashFS and Jefferson filesystems to try and locate interesting files


pentest@hitb-hv-1: ~/Workspace/firmware
pentest@hitb-hv-1: ~
pentest@hitb-hv-1: ~/Workspace/firmware

```
pentest@hitb-hv-1:~/Workspace/firmware$ docker run -it --rm -v $(pwd):/workspace -
w /workspace sheabot/binwalk -e hitb-firmware-dump.BIN

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
188908        0x2E1EC         CRC32 polynomial table, little endian
193820        0x2F51C         LZO compressed data
196044        0x2FDCC         Android bootimg, kernel size: 0 bytes, kernel addr:
0x70657250, ramdisk size: 543519329 bytes, ramdisk addr: 0x6E72656B, product name:
 "mem boot start"
262144        0x40000         uImage header, header size: 64 bytes, header CRC: 0x
D303C2CB, created: 2023-11-21 09:58:03, image size: 1681989 bytes, Data Address: 0
x80010000, Entry Point: 0x8040BED0, data CRC: 0x26E5AB90, OS: Linux, CPU: MIPS, im
age type: OS Kernel Image, compression type: lzma, image name: "Linux-3.10.14__isv
p_pike_1.0__"
262208        0x40040         LZMA compressed data, properties: 0x5D, dictionary s
ize: 67108864 bytes, uncompressed size: -1 bytes
2097152       0x200000        Squashfs filesystem, little endian, version 4.0, com
pression:xz, size: 965404 bytes, 213 inodes, blocksize: 65536 bytes, created: 2024
-08-15 11:27:45
3080192       0x2F0000        Squashfs filesystem, little endian, version 4.0, com
pression:xz, size: 4314274 bytes, 77 inodes, blocksize: 65536 bytes, created: 2022
-01-01 00:00:00
7995392       0x7A0000        JFFS2 filesystem, little endian
```

Modify root password

The shadow file on *nix systems contains local user password hashes. We can change the hash to something we know

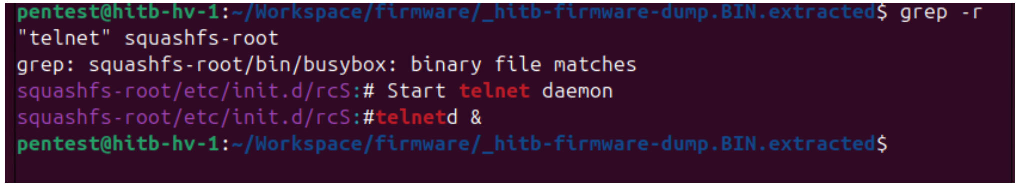


```
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted$ tail squ
ashfs-root/etc/shadow
root:$1$soidjfoi$YqVofy88ZPpjWu1nwaQzN1:10933:0:99999:7:::
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted$
```

```
$ openssl passwd -1 -salt [salt] [password]
```

Enable Telnet Access

The telnet daemon is installed but commented in the init.d/rcS script. By removing the # we enable the service



```
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted$ grep -r
"telnet" squashfs-root
grep: squashfs-root/bin/busybox: binary file matches
squashfs-root/etc/init.d/rcS:# Start telnet daemon
squashfs-root/etc/init.d/rcS:#telnetd &
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted$
```

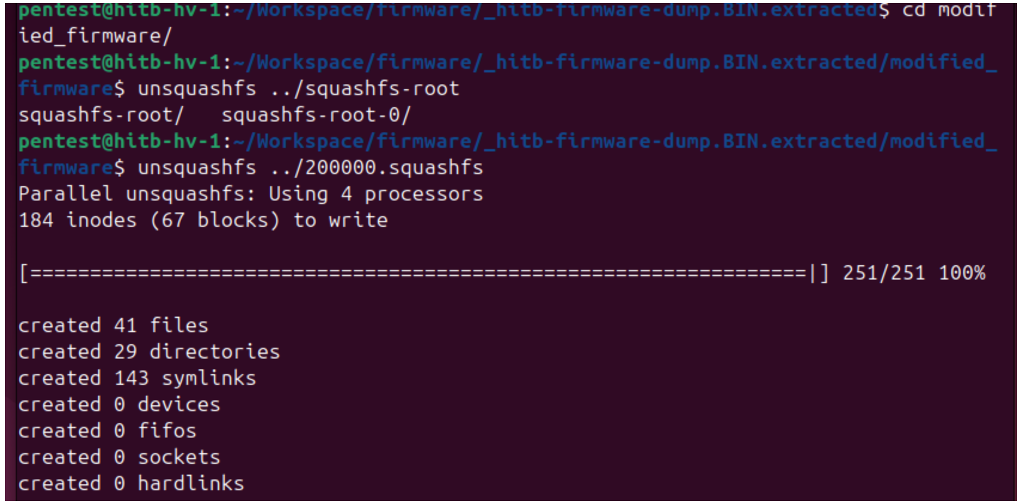
Security Research Labs

[1] https://github.com/ReFirmLabs/binwalk

Modifying filesystems – extracting and creating Squashfs filesystems to inject into flash binary

Extracting the Squashfs filesystem

The binwalk tool previously extracted the Squashfs filesystems using sasquatch. Let's extract it again using a stand alone binary (unsquash) which we have full control of.

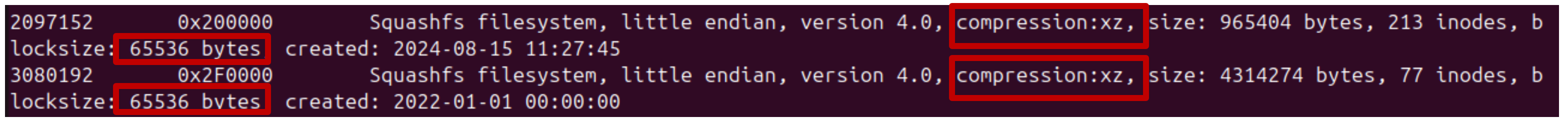


```
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted$ cd modif
ied_firmware/
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted/modified_
firmware$ unsquashfs ../squashfs-root
squashfs-root/   squashfs-root-0/
pentest@hitb-hv-1:~/Workspace/firmware/_hitb-firmware-dump.BIN.extracted/modified_
firmware$ unsquashfs ../200000.squashfs
Parallel unsquashfs: Using 4 processors
184 inodes (67 blocks) to write

[=================================================================|] 251/251 100%

created 41 files
created 29 directories
created 143 symlinks
created 0 devices
created 0 fifos
created 0 sockets
created 0 hardlinks
```

```
$ unsquash ../200000.squashfs
```

Filesystem parameters

Blocksize and compression type can be obtained from the previously ran binwalk command.

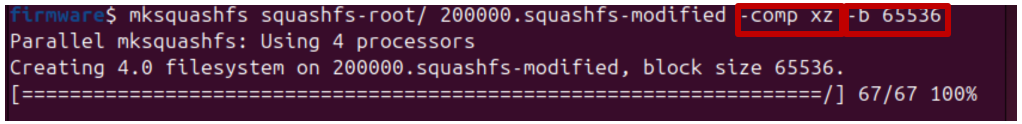


```
2097152       0x200000       Squashfs filesystem, little endian, version 4.0, compression:xz, size: 965404 bytes, 213 inodes, b
locksize: 65536 bytes  created: 2024-08-15 11:27:45
3080192       0x2F0000       Squashfs filesystem, little endian, version 4.0, compression:xz, size: 4314274 bytes, 77 inodes, b
locksize: 65536 bytes  created: 2022-01-01 00:00:00
```

```
$ binwalk ./firmware
```

Create filesystem

Using the above parameters, we can create a new Squashfs filesystem which closely matches the original.

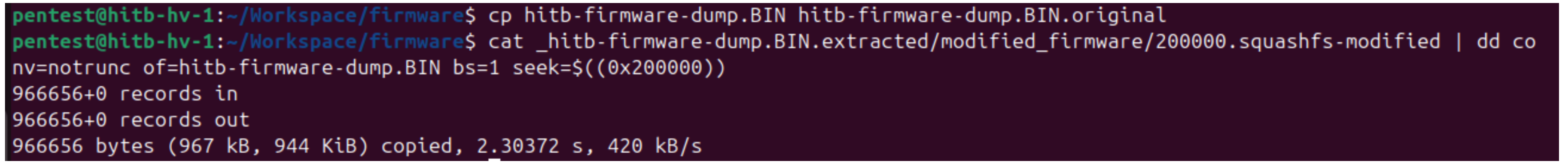


```
firmware$ mksquashfs squashfs-root/ 200000.squashfs-modified -comp xz -b 65536
Parallel mksquashfs: Using 4 processors
Creating 4.0 filesystem on 200000.squashfs-modified, block size 65536.
[=================================================================/] 67/67 100%
```

```
$ mksquashfs squashfs-root/ 200000-modified.squashfs -comp xz -b 65536
```

Injecting into flash dump

The modified squashfs partition can be injected into the original flash dump using cat and dd.



```
pentest@hitb-hv-1:~/Workspace/firmware$ cp hitb-firmware-dump.BIN hitb-firmware-dump.BIN.original
pentest@hitb-hv-1:~/Workspace/firmware$ cat _hitb-firmware-dump.BIN.extracted/modified_firmware/200000.squashfs-modified | dd co
nv=notrunc of=hitb-firmware-dump.BIN bs=1 seek=$((0x200000))
966656+0 records in
966656+0 records out
966656 bytes (967 kB, 944 KiB) copied, 2.30372 s, 420 kB/s
```

```
$ cat mnt/modified.squashfs | dd conv=notrunc
of=hitb-firmware-dump.bin bs=1 seek=$((0x200000))
```

Security Research Labs

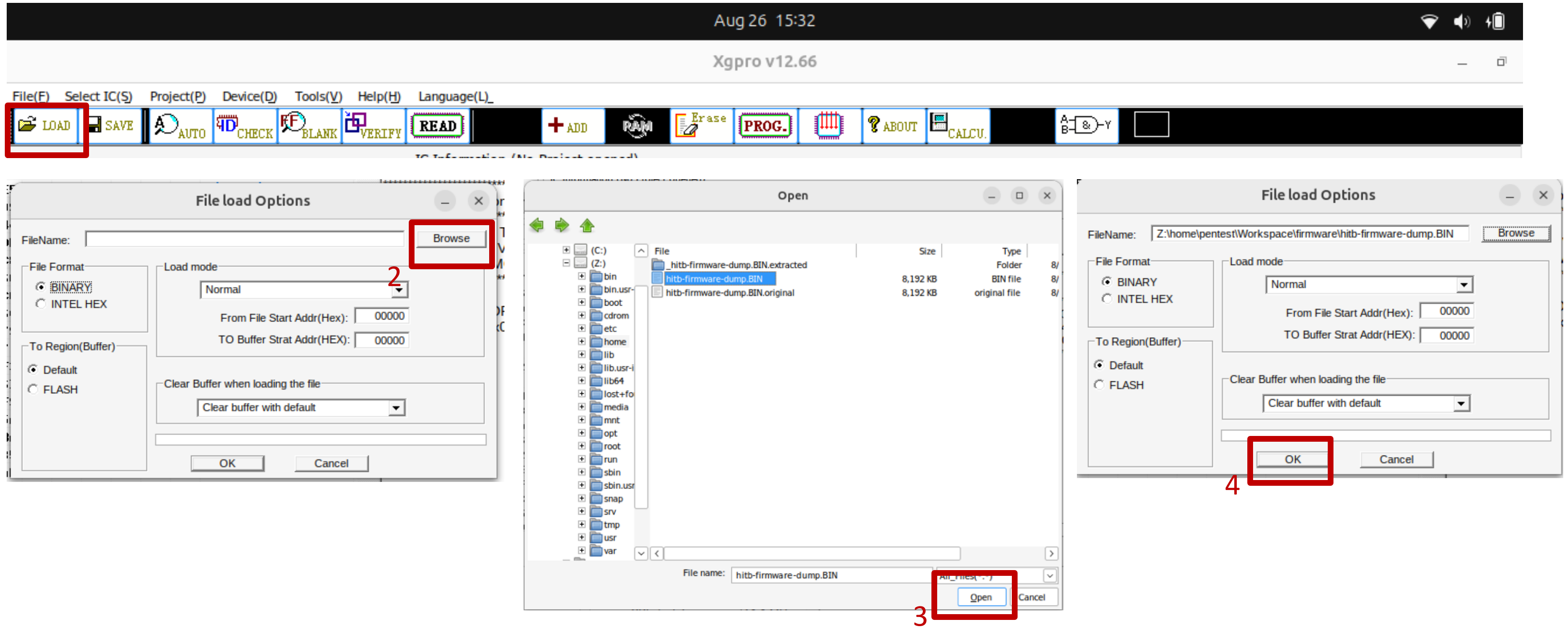
Agenda

1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

Writing filesystem back to chip

Load the flash dump

1. Click “LOAD” on the top menu
2. Click “Browse” in the File load window
3. Locate the dump.bin and click “Open”
4. Click “OK”

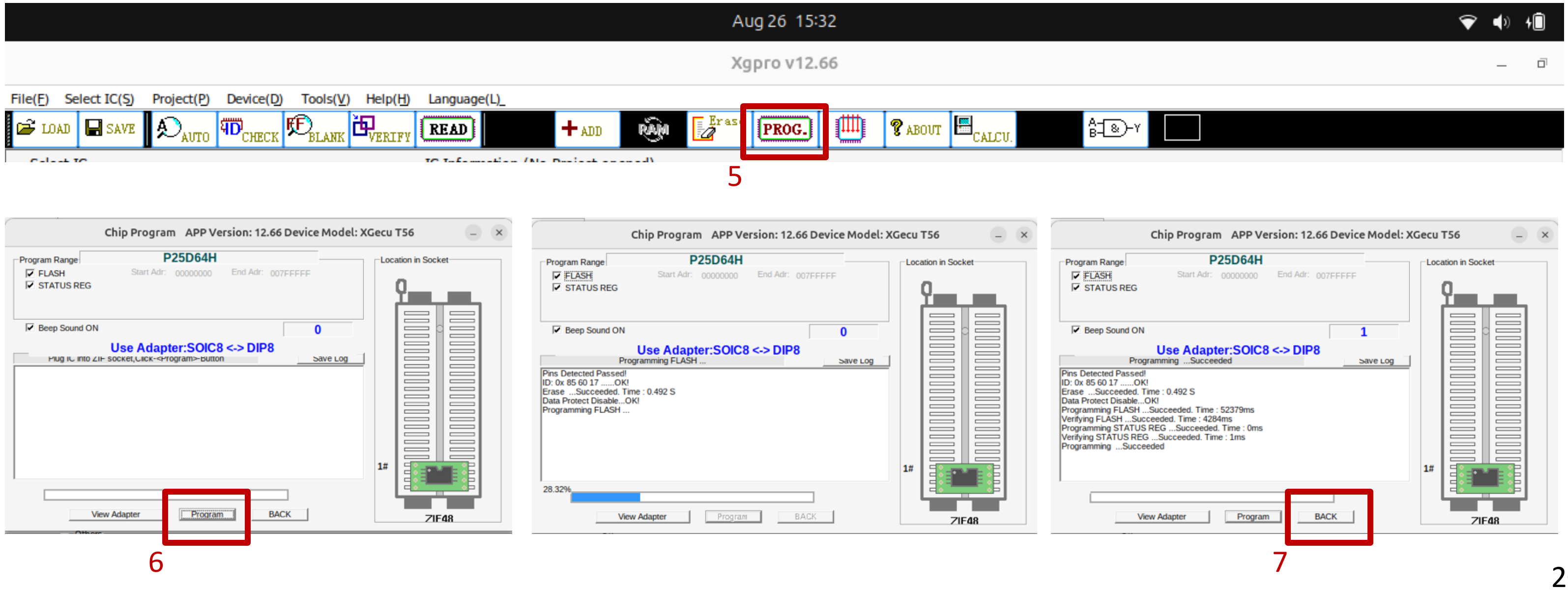

Aug 26 15:32
Xgpro v12.66
File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)
LOAD SAVE AUTO CHECK BLANK VERIFY READ ADD RAM Erase PROG. ABOUT CALCU.
1
File load Options
Browse
2
Open
3
Open Cancel
File load Options
OK Cancel
4

File load Options: FileName: Z:\home\pentest\Workspace\firmware\hitb-firmware-dump.BIN; File Format: ● BINARY ○ INTEL HEX; Load mode: Normal; From File Start Addr(Hex): 00000; TO Buffer Strat Addr(HEX): 00000; To Region(Buffer): ● Default ○ FLASH; Clear Buffer when loading the file: Clear buffer with default

Open: hitb-firmware-dump.BIN.extracted (Folder); hitb-firmware-dump.BIN (8,192 KB, BIN file); hitb-firmware-dump.BIN.original (8,192 KB, original file); File name: hitb-firmware-dump.BIN

Write the dump to the chip

5. Click “PROG” on the top menu
6. Click “Program” in the Chip Program window
7. Click “BACK” once complete


Aug 26 15:32
Xgpro v12.66
File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)
LOAD SAVE AUTO CHECK BLANK VERIFY READ ADD RAM Erase PROG. ABOUT CALCU.
5
Chip Program APP Version: 12.66 Device Model: XGecu T56
P25D64H
Program Range
FLASH
STATUS REG
Location in Socket
ZIF48
1#
Save Log
View Adapter
Program
BACK
6
7

Program Range: ☑ FLASH ☑ STATUS REG; ☑ Beep Sound ON

Use Adapter:SOIC8 <-> DIP8

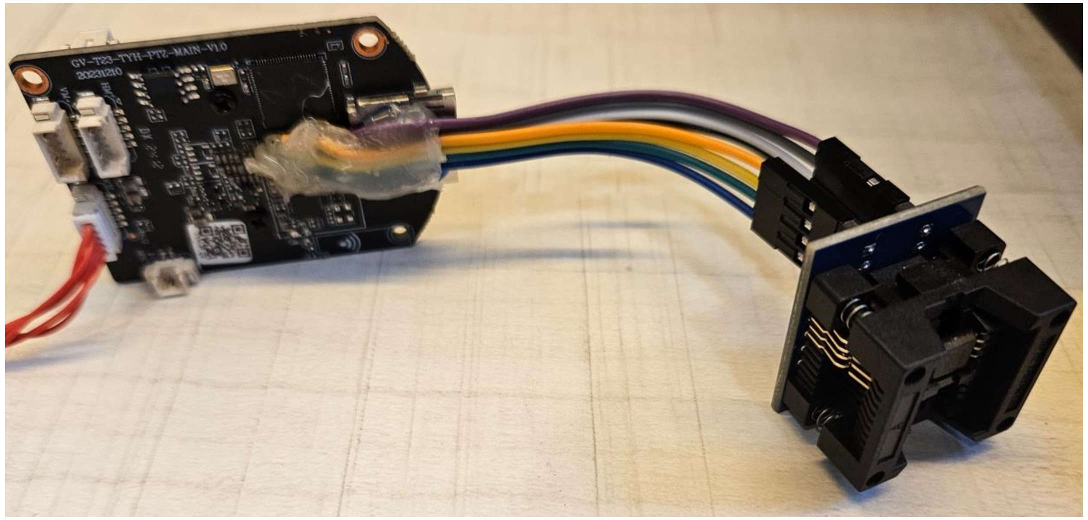
```
Programming FLASH ...
Pins Detected Passed!
ID: 0x 85 60 17 .......OK!
Erase  ...Succeeded. Time : 0.492 S
Data Protect Disable...OK!
Programming FLASH ...
28.32%
```

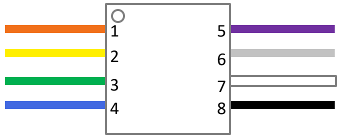
```
Programming  ...Succeeded
Pins Detected Passed!
ID: 0x 85 60 17 .......OK!
Erase  ...Succeeded. Time : 0.492 S
Data Protect Disable...OK!
Programming FLASH ...Succeeded. Time : 52379ms
Verifying FLASH ...Succeeded. Time : 4284ms
Programming STATUS REG ...Succeeded. Time : 0ms
Verifying STATUS REG ...Succeeded. Time : 1ms
Programming  ...Succeeded
```

Agenda

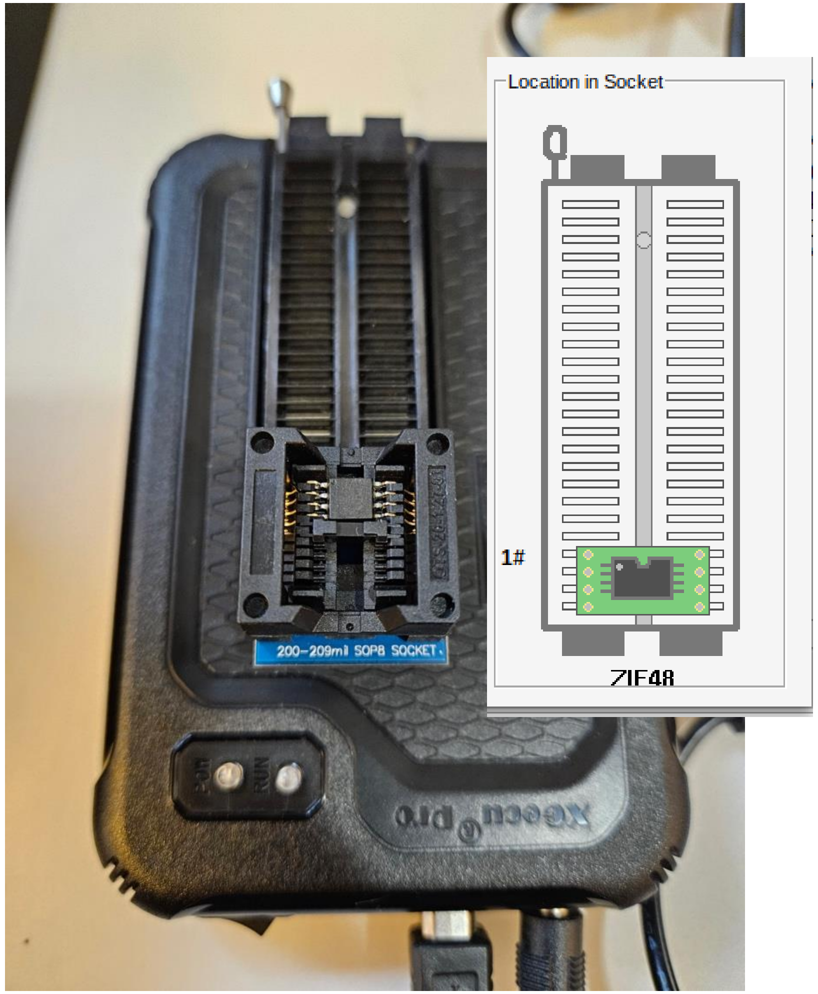
1. What is SPI flash memory?
2. Identifying flash memory
3. Target selection
4. Dumping flash memory
5. Modifying filesystems
6. Writing filesystem back to chip
7. Getting a root shell

Getting root shell – re-attaching the chip IPcam





Remove DIP8 socket from the Xgeco programmer and connect back to the wiring harness



Getting root shell – telnet if camera no longer broken :/

Root shell here ;-)