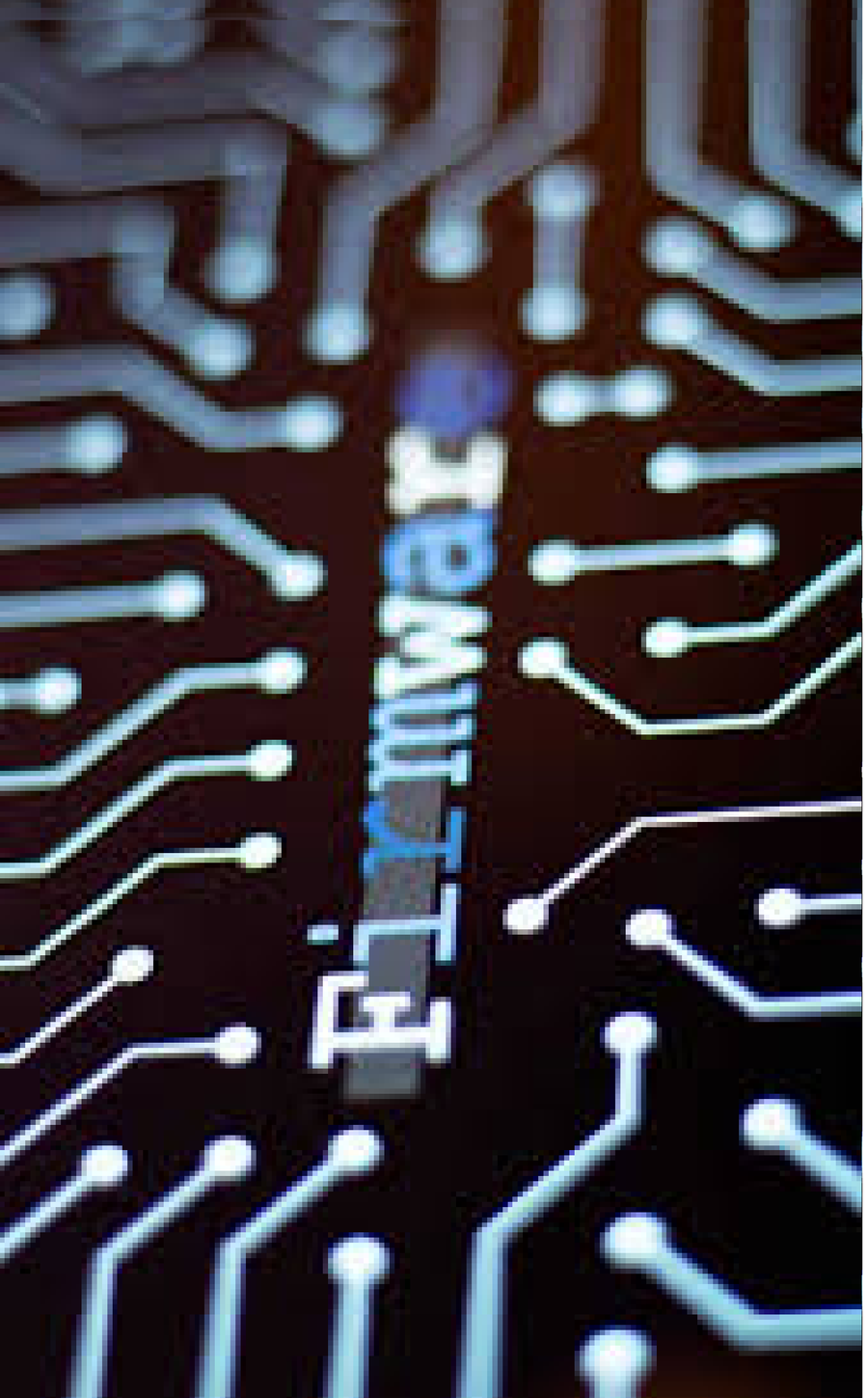
PROJECT PRESENTATION ON FIRMWARE EXTRACTION

PRESENTED BY SHRUTI KONDE



Objective

The main objective of firmware extraction is to read and study the software stored inside a device’s memory. This helps to understand how the device works, find security issues, fix errors, add new features, or recover data. It is mainly used for reverse engineering, security testing, and repairing devices.

Operating System

Ubuntu Linux: A powerful and open-source operating system commonly used for firmware extraction and analysis due to its flexibility, command-line tools, and compatibility with various hardware interfaces.

Tools Used:

Flashrom: A command-line utility used to read, write, erase, and verify firmware from flash memory chips.

Binwalk:

A firmware analysis tool used to extract, inspect, and reverse-engineer binary files.

Hardware Used:

Flash Memory Chip: Programmer/Reader (e.g., CH341A):

A hardware interface used to connect the flash chip to the Linux system for reading or writing firmware data.

Soldering Tools (if needed): Used to access the memory chip directly when no debugging port is available.

Linux Commands

1.Check Connected Devices (Optional):
lsusb
Lists connected USB devices like programmers or readers.

2.Install Required Tools:
s**udo apt install flashrom binwalk**
Installs the firmware extraction and analysis tools.

3.Read/Dump Firmware using Flashrom:
sudo flashrom -p ch341a_spi -r backup.bin
Reads the firmware from the flash memory chip and saves it as backup.bin.

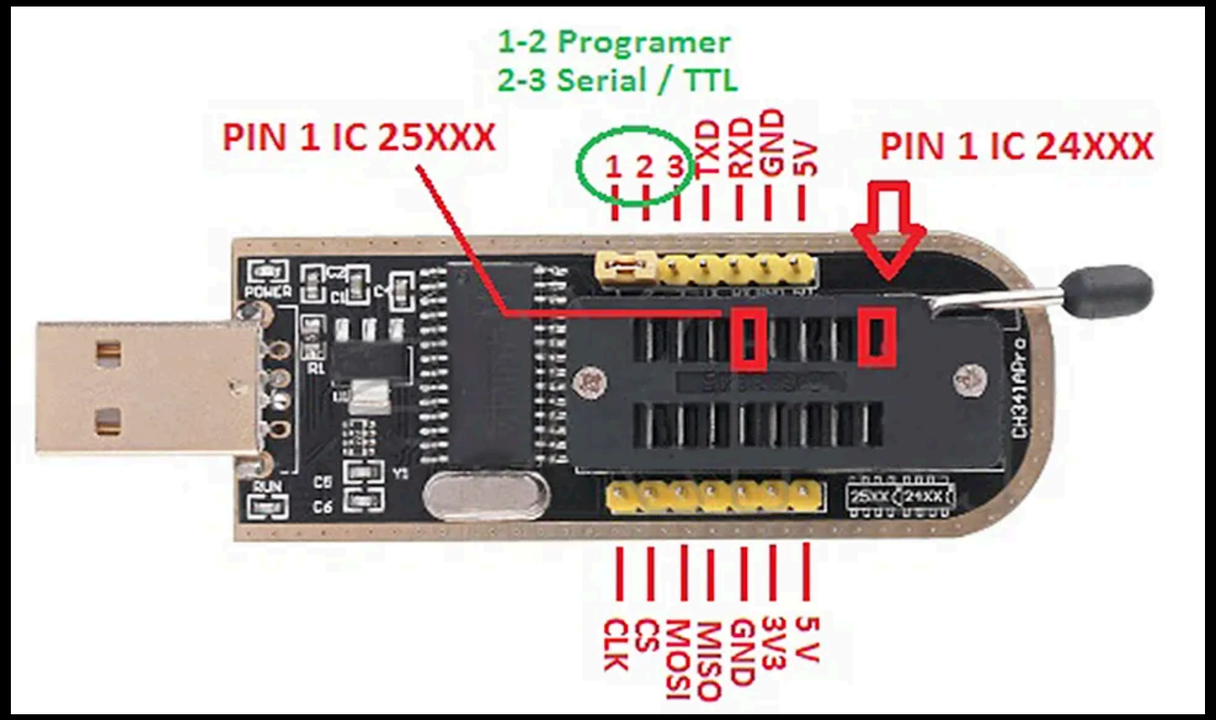
4.Verify the Dumped Firmware:
s**udo flashrom -p ch341a_spi -v backup.bin**
Verifies that the extracted firmware is correct.

5.Analyze Firmware with Binwalk:
binwalk backup.bin
Scans the firmware file to identify file systems, compressed data, or code sections.

6.Extract Contents from Firmware:
binwalk -e backup.bin
Extracts embedded files and directories from the firmware image.

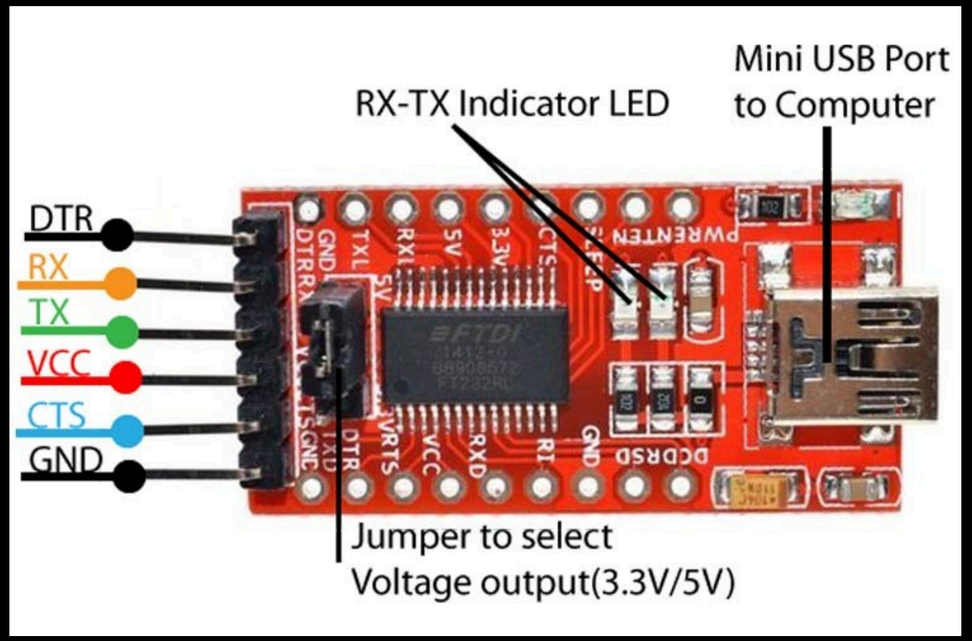
7.View Contents of Extracted Files:
ls _backup.bin.extracted/
Displays the files and directories extracted from the firmware.

Devices used



CH341 Programmer

The CH341 is a low-cost USB-to-serial and USB-to-SPI/I²C adapter commonly used for reading, writing, and programming flash memory chips during firmware extraction. It allows your Linux system to communicate directly with the chip, making it easy to dump firmware, update it, or recover devices. Due to its affordability, wide support, and compatibility with tools like Flashrom, it is one of the most popular choices for firmware analysis and reverse engineering.



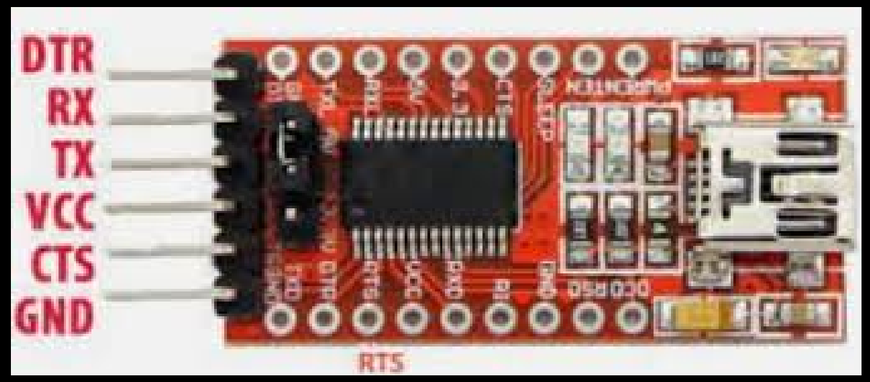
UART

Universal
Asynhronous
Reciever
Transmiter

This is a USB to TTL (Serial) Converter Module, commonly based on the FTDI FT232RL chip. It is used to establish a serial communication link between a computer and microcontrollers or other devices.

Key Features:

Mini USB Port: Connects the module to the computer.
RX & TX Pins: Used for data transmission and reception.
DTR/CTS Pins: Control signals for communication flow.
Voltage Jumper: Allows selection of 3.3V or 5V output.
Indicator LEDs: Show data transmission activity.



USB Programmer

Name: HW-417 V1.2 CH341A USB Programmer

Purpose: A low-cost USB programmer used for reading, writing, and erasing SPI flash memory and EEPROM chips — commonly used for BIOS flashing, firmware extraction, and recovery.

Key Features:

Supports 24-series and 25-series EEPROM/SPI flash chips (e.g., 24Cxx, 25Qxx).

Works with SOIC8 test clips and adapter boards.

Compatible with tools like flashrom on Linux and Windows GUI programmers.

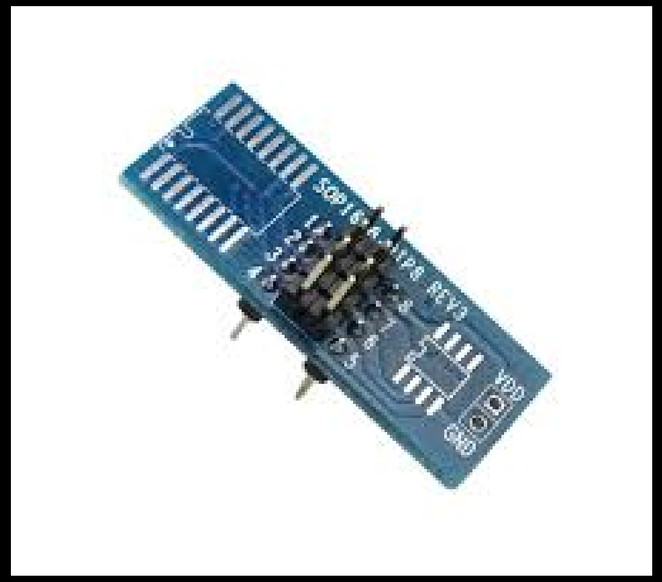
Connects directly to the USB port for power and data transfer.

Use Case:

Extract or update router/PC motherboard firmware.

Repair corrupted BIOS/UEFI chips.

Backup or clone SPI flash chips.



Name: SOP8/SOIC8 to DIP8 Adapter Board
Purpose: It is used to convert a surface-mount 8-pin (or 16-pin) SPI flash chip (like a BIOS or EEPROM) to a DIP format so it can be easily connected to a programmer or breadboard.

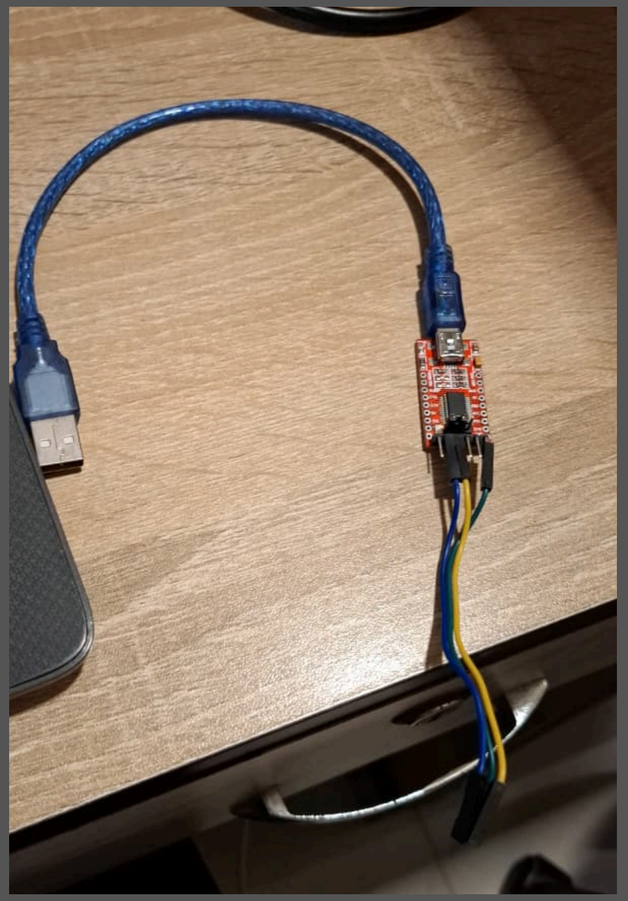
Use Case: Commonly used in firmware extraction, BIOS flashing, or reading/writing SPI flash memory chips.

Name: SOIC8/SOP8 Test Clip (also called SPI Flash Clip or SOIC8 Programming Clip).

Purpose: This spring-loaded clip connects directly to an 8-pin SPI flash chip in-circuit (without desoldering) so you can read, write, or extract firmware using tools like CH341A programmer.

Use Case: Widely used in hardware hacking, BIOS recovery, and firmware dumping/modification.

Interfacing with a Microcontroller via USB



- Components: A laptop connected to a USB to TTL serial converter module (e.g., based on a CH340 or FTDI chip).
- Function: The converter acts as a bridge, allowing the computer's USB interface to communicate with a device that uses TTL-level serial communication (TX/RX pins).
- Use Case: Essential for uploading code to microcontrollers like Arduino or ESP8266, and for viewing debug messages on a serial monitor.

ROUTERS

Routers

A router is a networking device that connects multiple networks and directs data between them.

It acts as a gateway between your local network (LAN) and the internet.

Routers ensure that data packets reach the correct destination quickly and securely.

🔧 Key Functions:

📍 Routes data between devices and networks

🔐 Provides security with firewall and filtering

🌐 Assigns IP addresses to devices (DHCP)

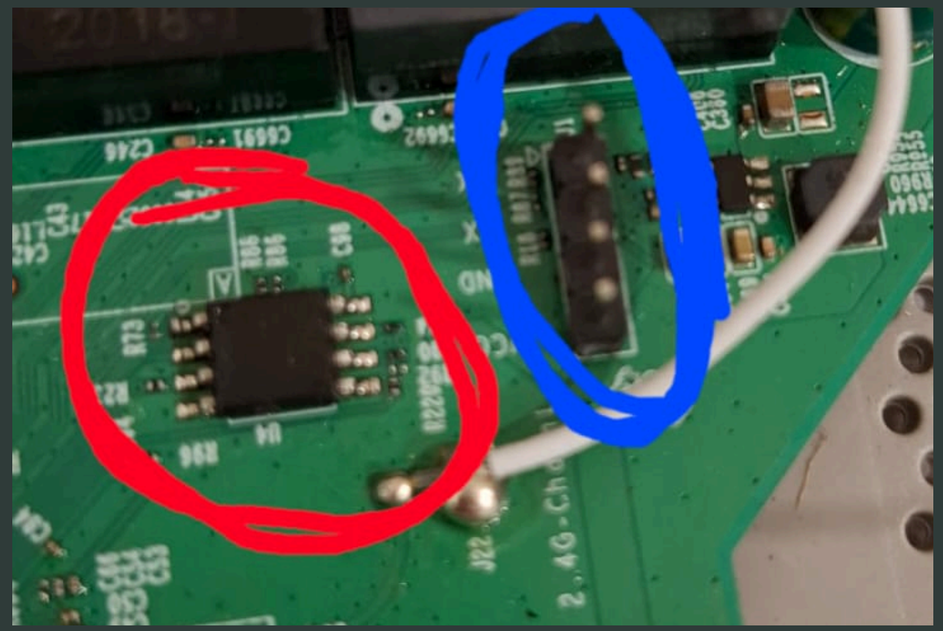
📡 Enables Wi-Fi connectivity (in wireless routers)

🛠️ Common Uses:

Home and office internet connection

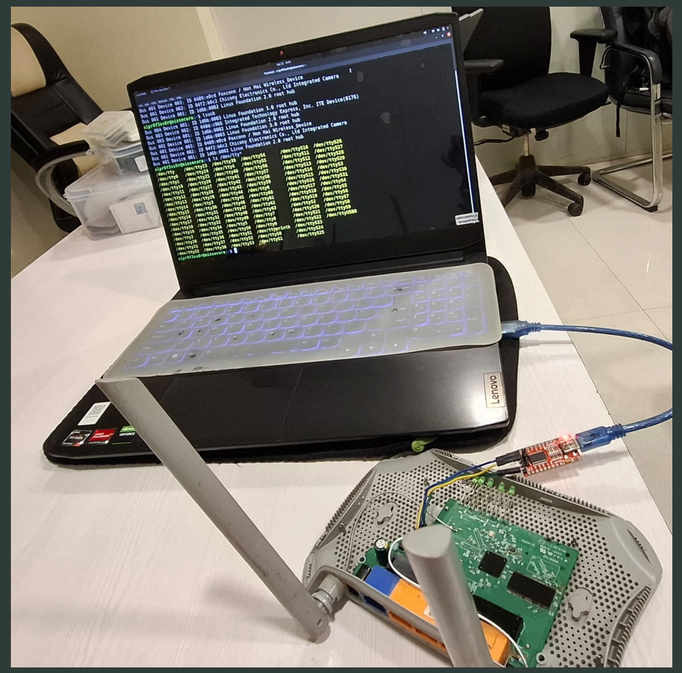
Sharing internet with multiple devices

Secure communication between networks



The red highlighted part surrounds an integrated circuit (IC) with 8 pins, which typically suggests it could be a small memory chip (such as EEPROM or Flash), a voltage regulator, or an operational amplifier. Its actual function can be verified by reading the part number printed on the chip, though this is not clearly visible in the picture. In network devices, 8-pin ICs in that location are often used for roles such as signal amplification, voltage regulation, or non-volatile memory storage.

The blue highlighted part focuses on a group of pins and possibly a pin header used for interfacing or programming. This is often a UART, JTAG, or serial interface header used for debugging, programming firmware, or monitoring the device during development or repair. Such headers are common in network devices for maintenance and flashing new firmware.



Accessing a Router's Serial Console

This setup is used to gain low-level command-line access to the internal operating system of a Wi-Fi router.

- What's Happening?

Disassembled Router: The router's case is open to expose the main circuit board (PCB).

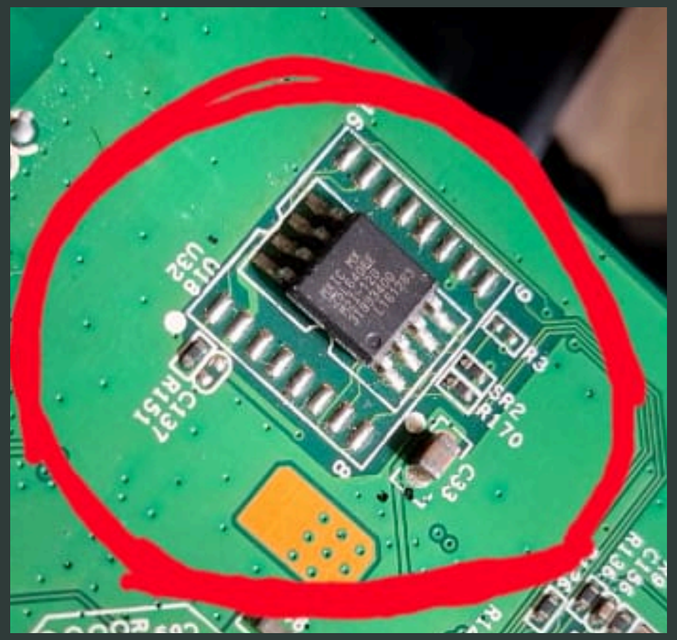
- Serial Connection: A USB to TTL serial adapter (the small red board) is connected between the laptop's USB port and a specific set of pins on the router's board.
- Laptop Interface: The laptop is running a Linux operating system, with a terminal window open.

Why Do This?

- This direct hardware connection allows a user to bypass the standard web interface and interact directly with the router's bootloader and operating system.
- De-bricking: Recovering a router that has become unresponsive ("bricked") after a failed firmware update.
- Debugging: Viewing detailed boot-up messages to diagnose hardware or software issues.
- Security Research: Exploring the device's software to find and analyze potential vulnerabilities.

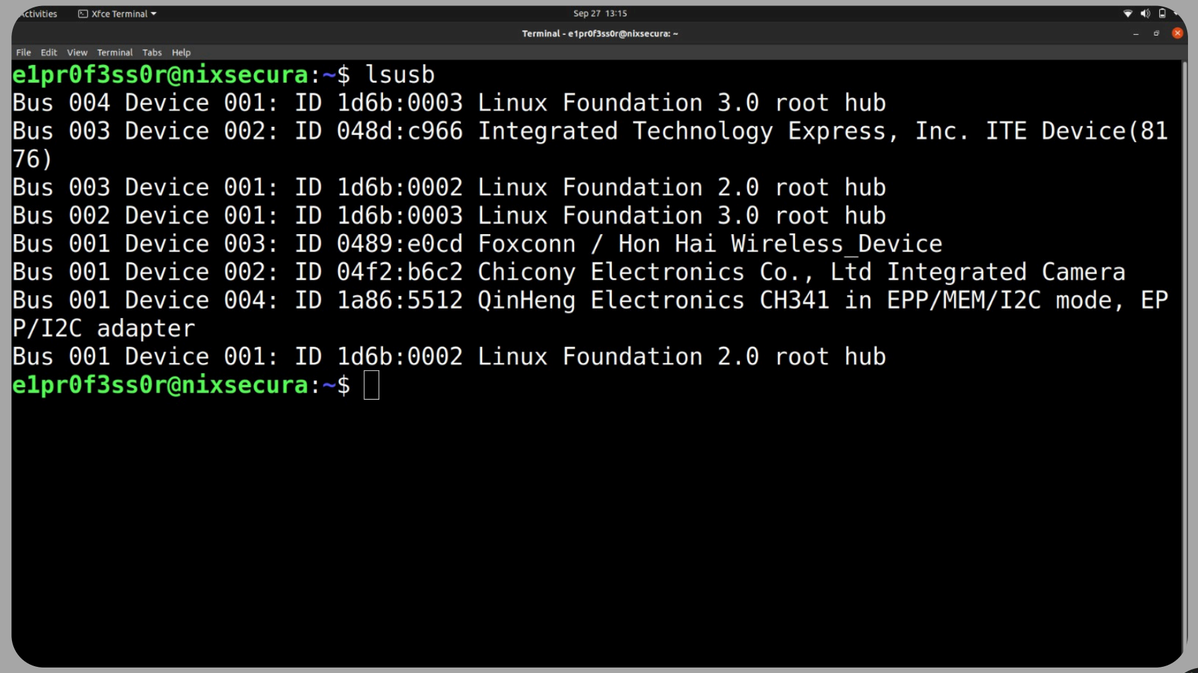
D-Link Routers DIR 816

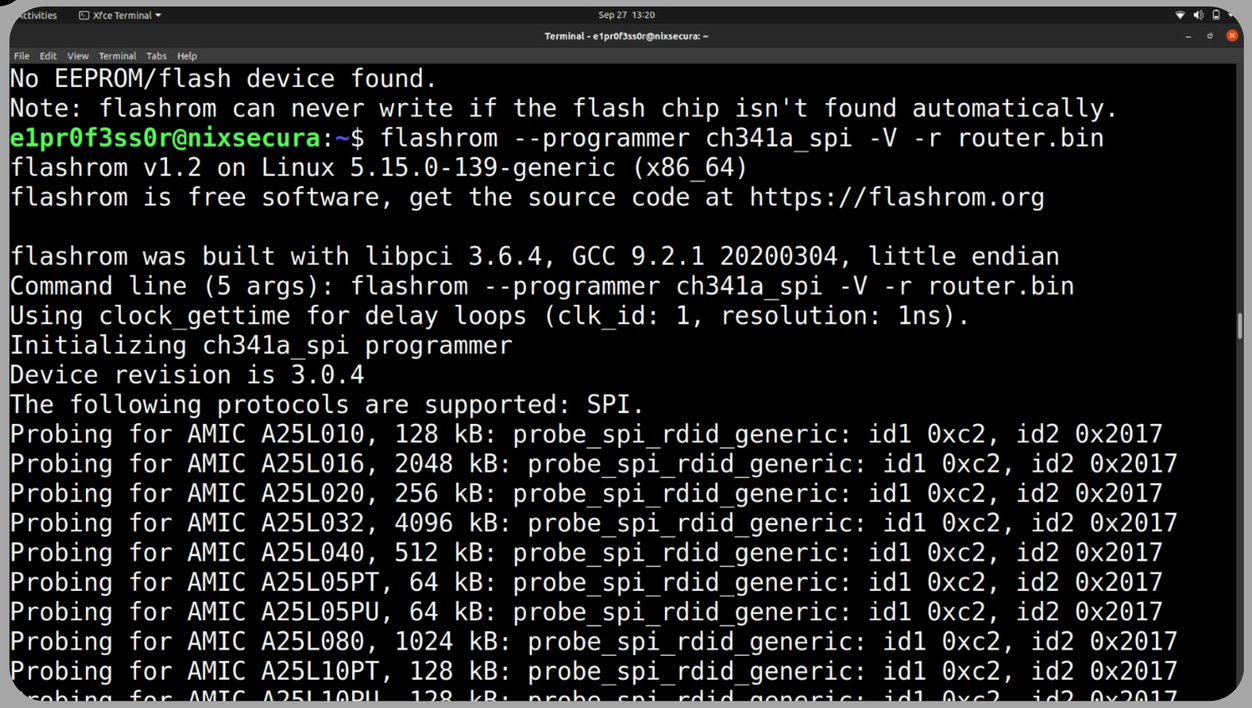
D-Link DIR-816 (AC750 Dual-Band Router) is a budget-friendly Wi-Fi router for homes and small offices. It supports dual-band wireless (2.4 GHz up to 300 Mbps + 5 GHz up to 433 Mbps), has 1 WAN and 4 LAN ports (100 Mbps), and includes a USB 2.0 port for storage or modem use. It features 3 external antennas, WPA/WPA2 security, guest Wi-Fi, and can work as a router, repeater, or access point. Compact and easy to set up, but limited to Fast Ethernet speeds and basic performance.



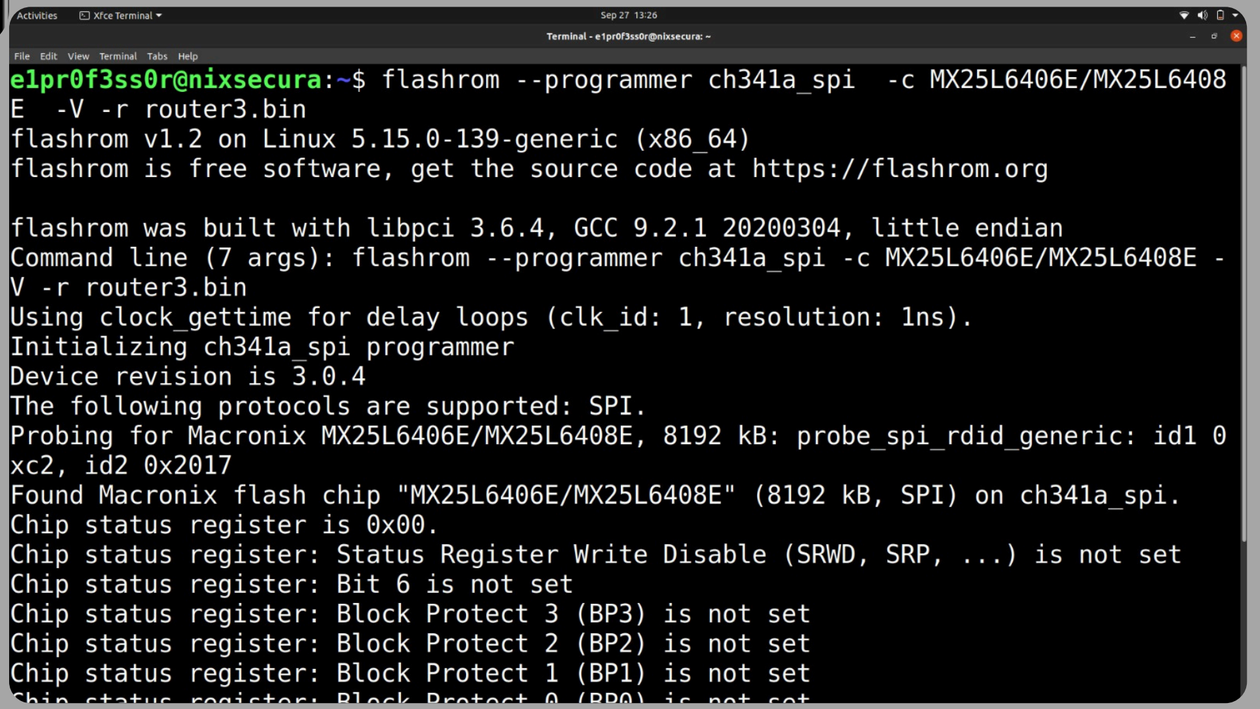
Flashmemory of DIR 816

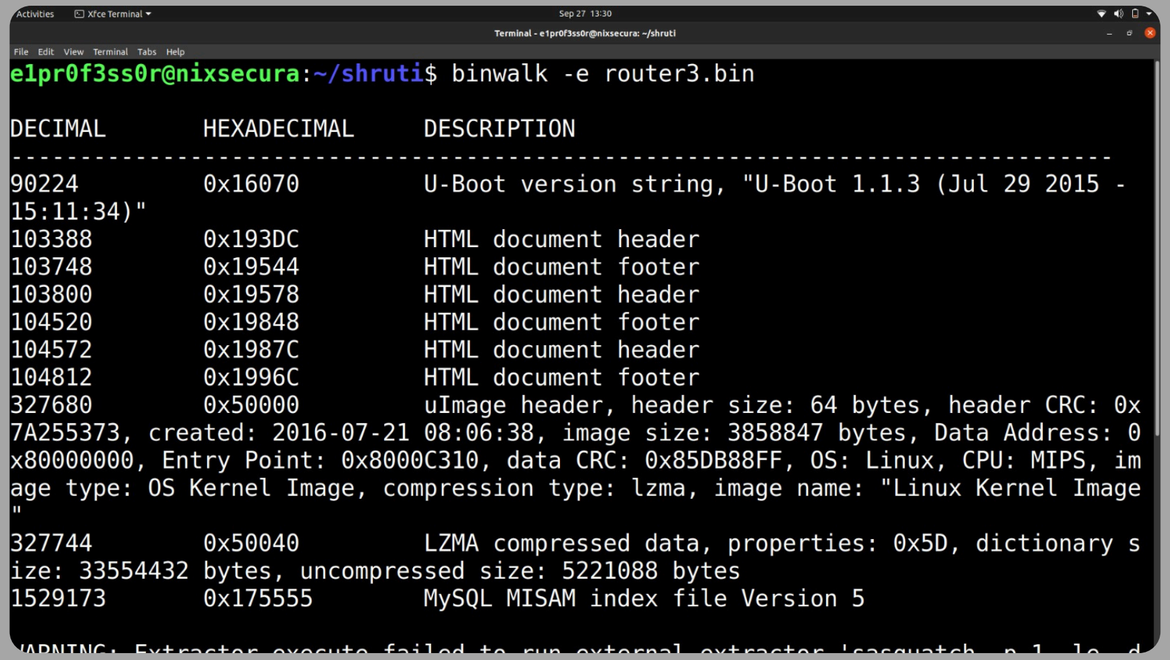
The highlighted part in the image is an 8-pin integrated circuit (IC) on the PCB, likely serving as a small controller, memory chip, or voltage regulator. These types of ICs are often used for EEPROM, op-amps, or interface roles, and their exact function depends on the part number printed on the chip and the circuit context in the device. Typical uses include storing configuration data, regulating power, or amplifying signals in electronic circuits.

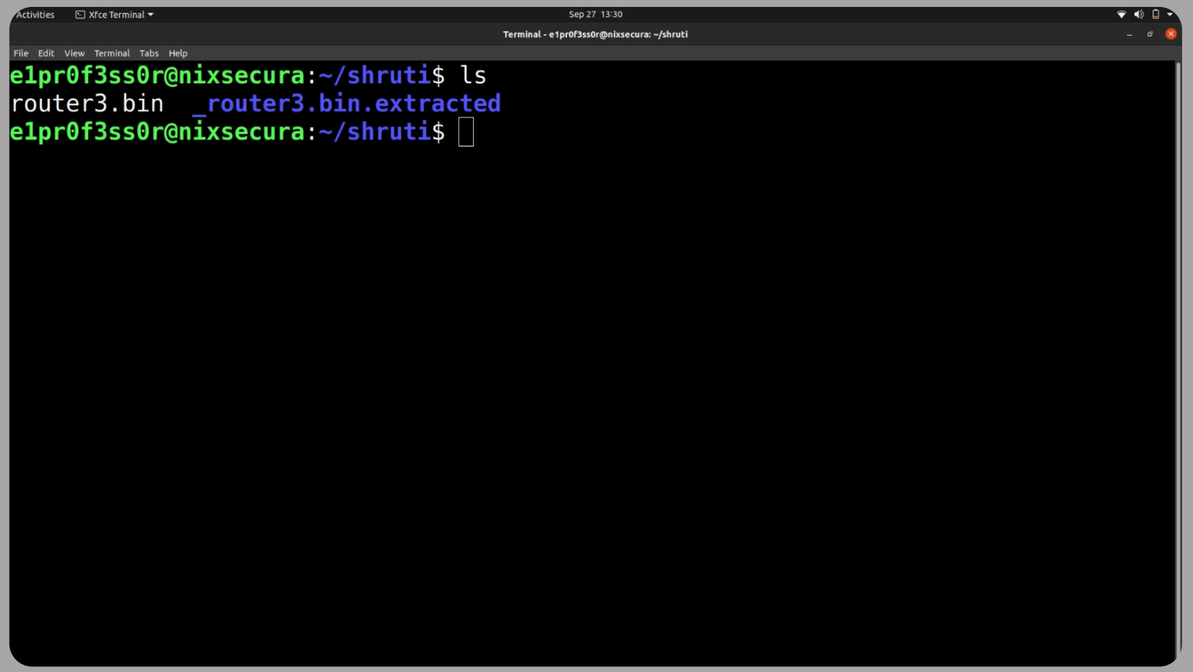

Activities
Xfce Terminal
Sep 27 13:15
Terminal - e1pr0f3ss0r@nixsecura: ~
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 048d:c966 Integrated Technology Express, Inc. ITE Device(8176)
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0489:e0cd Foxconn / Hon Hai Wireless_Device
Bus 001 Device 002: ID 04f2:b6c2 Chicony Electronics Co., Ltd Integrated Camera
Bus 001 Device 004: ID 1a86:5512 QinHeng Electronics CH341 in EPP/MEM/I2C mode, EPP/I2C adapter
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
e1pr0f3ss0r@nixsecura:~$

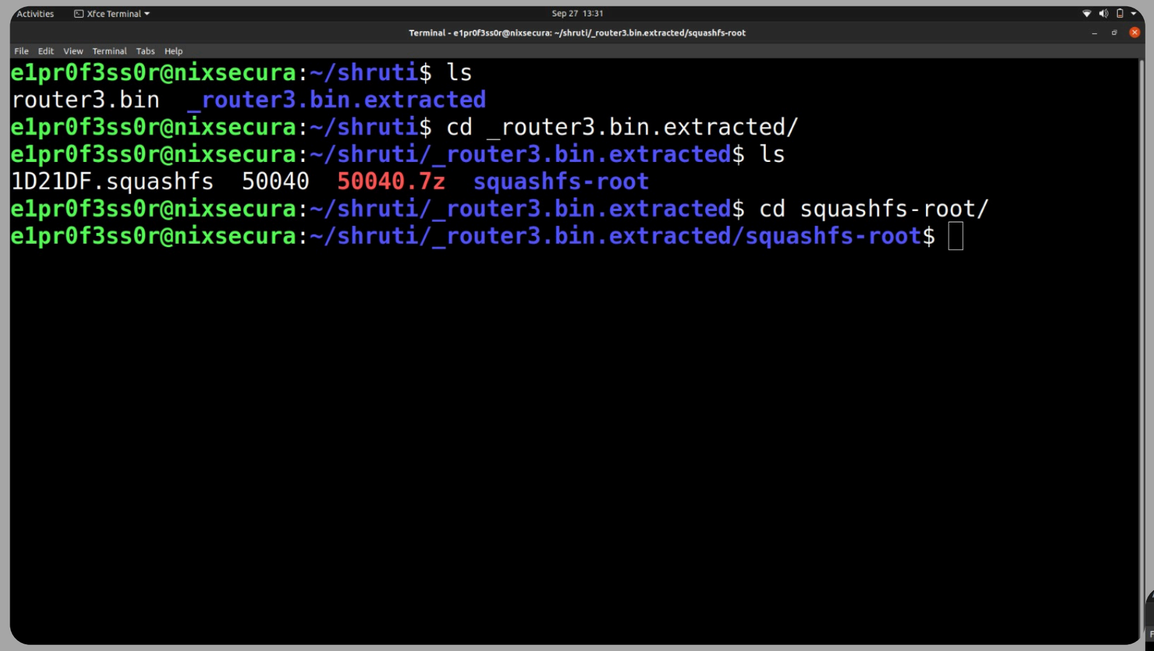

Activities
Xfce Terminal
Sep 27 13:20
Terminal - e1pr0f3ss0r@nixsecura: ~
File Edit View Terminal Tabs Help
No EEPROM/flash device found.
Note: flashrom can never write if the flash chip isn't found automatically.
e1pr0f3ss0r@nixsecura:~$ flashrom --programmer ch341a_spi -V -r router.bin
flashrom v1.2 on Linux 5.15.0-139-generic (x86_64)
flashrom is free software, get the source code at https://flashrom.org
flashrom was built with libpci 3.6.4, GCC 9.2.1 20200304, little endian
Command line (5 args): flashrom --programmer ch341a_spi -V -r router.bin
Using clock_gettime for delay loops (clk_id: 1, resolution: 1ns).
Initializing ch341a_spi programmer
Device revision is 3.0.4
The following protocols are supported: SPI.
Probing for AMIC A25L010, 128 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L016, 2048 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L020, 256 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L032, 4096 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L040, 512 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L05PT, 64 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L05PU, 64 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L080, 1024 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Probing for AMIC A25L10PT, 128 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017

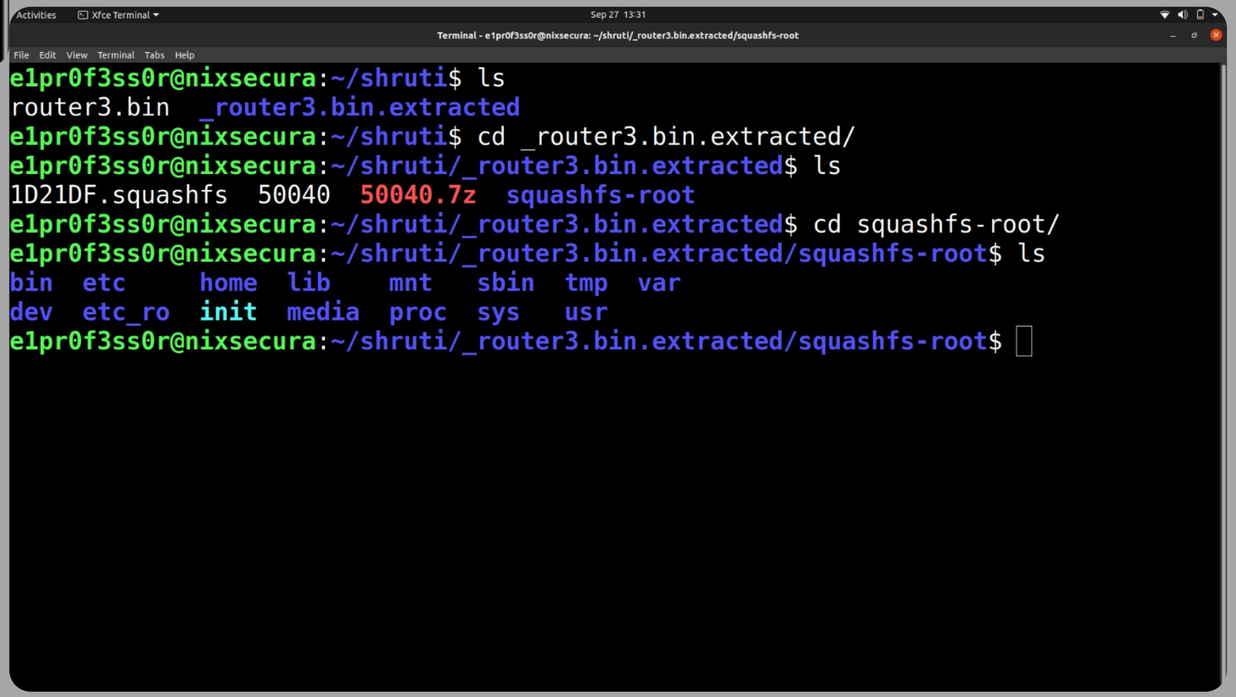

Activities
Xfce Terminal
Sep 27 13:22
Terminal - e1pr0f3ss0r@nixsecura: ~
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~$ flashrom -L | grep -i MX25L6406E
Macronix MX25L6406E/ PREW 8192 SPI
e1pr0f3ss0r@nixsecura:~$

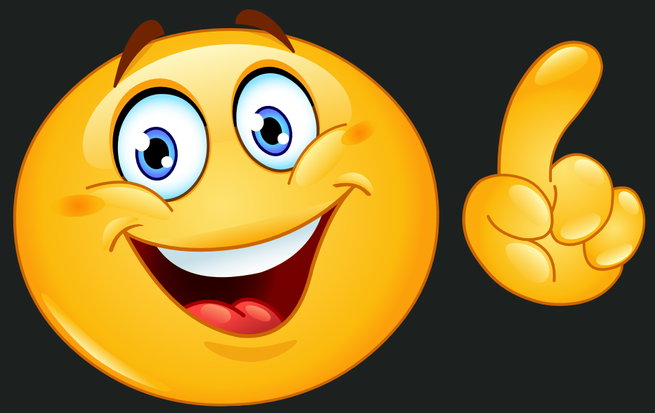

Activities
Xfce Terminal
Sep 27 13:26
Terminal - e1pr0f3ss0r@nixsecura: ~
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~$ flashrom --programmer ch341a_spi -c MX25L6406E/MX25L6408E -V -r router3.bin
flashrom v1.2 on Linux 5.15.0-139-generic (x86_64)
flashrom is free software, get the source code at https://flashrom.org
flashrom was built with libpci 3.6.4, GCC 9.2.1 20200304, little endian
Command line (7 args): flashrom --programmer ch341a_spi -c MX25L6406E/MX25L6408E -V -r router3.bin
Using clock_gettime for delay loops (clk_id: 1, resolution: 1ns).
Initializing ch341a_spi programmer
Device revision is 3.0.4
The following protocols are supported: SPI.
Probing for Macronix MX25L6406E/MX25L6408E, 8192 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2017
Found Macronix flash chip "MX25L6406E/MX25L6408E" (8192 kB, SPI) on ch341a_spi.
Chip status register is 0x00.
Chip status register: Status Register Write Disable (SRWD, SRP, ...) is not set
Chip status register: Bit 6 is not set
Chip status register: Block Protect 3 (BP3) is not set
Chip status register: Block Protect 2 (BP2) is not set
Chip status register: Block Protect 1 (BP1) is not set

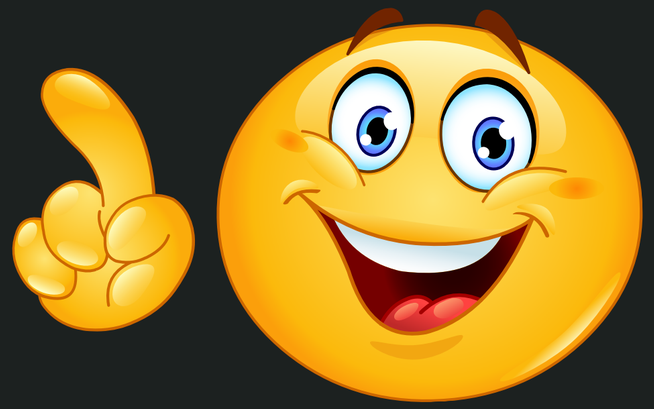

Activities
Xfce Terminal
Sep 27 13:30
Terminal - e1pr0f3ss0r@nixsecura: ~/shruti
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~/shruti$ binwalk -e router3.bin
DECIMAL HEXADECIMAL DESCRIPTION
90224 0x16070 U-Boot version string, "U-Boot 1.1.3 (Jul 29 2015 - 15:11:34)"
103388 0x193DC HTML document header
103748 0x19544 HTML document footer
103800 0x19578 HTML document header
104520 0x19848 HTML document footer
104572 0x1987C HTML document header
104812 0x1996C HTML document footer
327680 0x50000 uImage header, header size: 64 bytes, header CRC: 0x7A255373, created: 2016-07-21 08:06:38, image size: 3858847 bytes, Data Address: 0x80000000, Entry Point: 0x8000C310, data CRC: 0x85DB88FF, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name: "Linux Kernel Image"
327744 0x50040 LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes, uncompressed size: 5221088 bytes
1529173 0x175555 MySQL MISAM index file Version 5


Activities
Xfce Terminal
Sep 27 13:30
Terminal - e1pr0f3ss0r@nixsecura: ~/shruti
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~/shruti$ ls
router3.bin _router3.bin.extracted
e1pr0f3ss0r@nixsecura:~/shruti$


Activities
Xfce Terminal
Sep 27 13:31
Terminal - e1pr0f3ss0r@nixsecura: ~/shruti/_router3.bin.extracted/squashfs-root
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~/shruti$ ls
router3.bin _router3.bin.extracted
e1pr0f3ss0r@nixsecura:~/shruti$ cd _router3.bin.extracted/
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted$ ls
1D21DF.squashfs 50040 50040.7z squashfs-root
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted$ cd squashfs-root/
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted/squashfs-root$


Activities
Xfce Terminal
Sep 27 13:31
Terminal - e1pr0f3ss0r@nixsecura: ~/shruti/_router3.bin.extracted/squashfs-root
File Edit View Terminal Tabs Help
e1pr0f3ss0r@nixsecura:~/shruti$ ls
router3.bin _router3.bin.extracted
e1pr0f3ss0r@nixsecura:~/shruti$ cd _router3.bin.extracted/
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted$ ls
1D21DF.squashfs 50040 50040.7z squashfs-root
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted$ cd squashfs-root/
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted/squashfs-root$ ls
bin etc home lib mnt sbin tmp var
dev etc_ro init media proc sys usr
e1pr0f3ss0r@nixsecura:~/shruti/_router3.bin.extracted/squashfs-root$





Thank You !!!

Here is the feedback form kindly fill it and share the feedback related to the session.