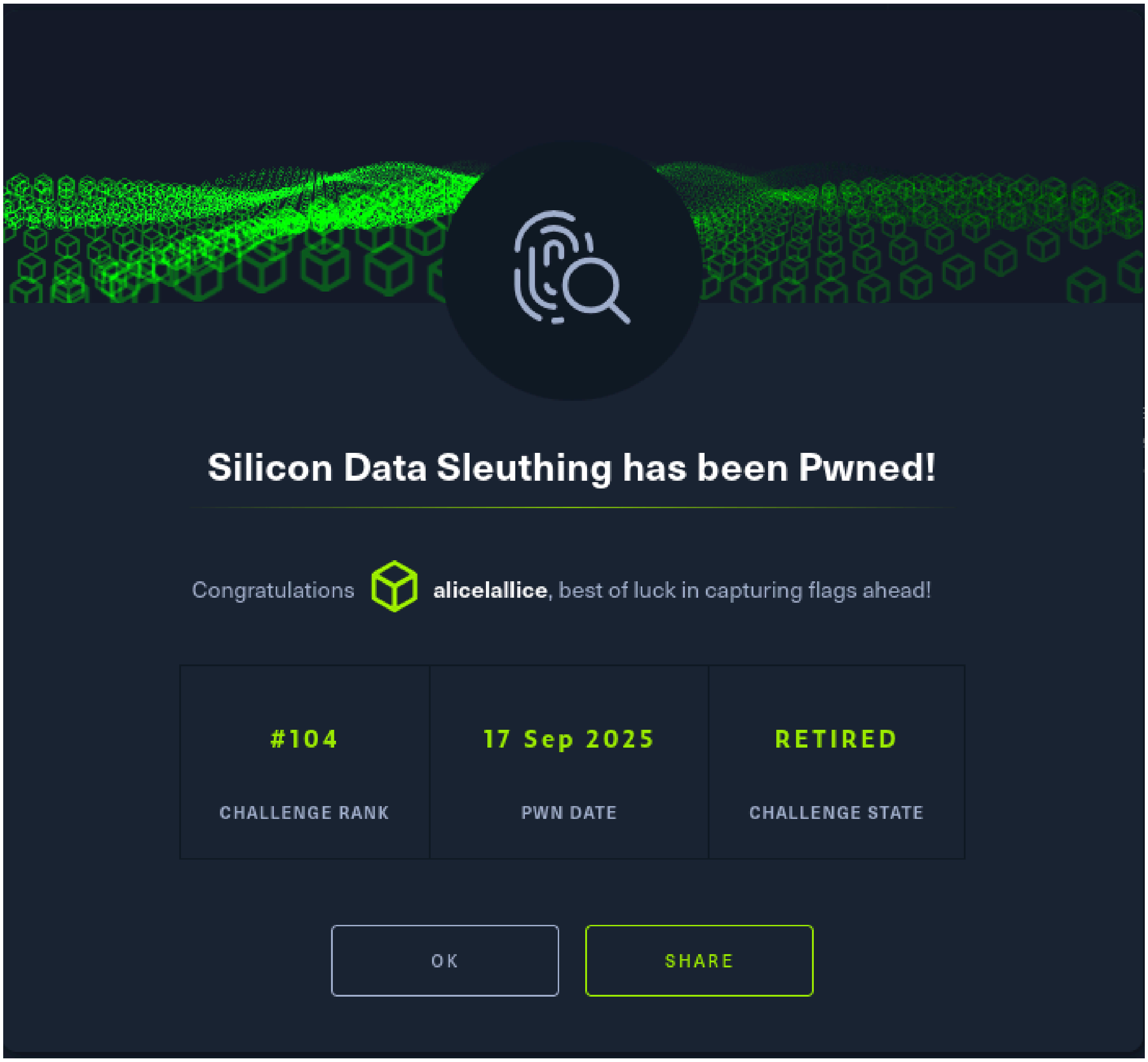
Silicon Data Sleuthing

▨ Types	forensic
▨ CTF	HTB



Silicon Data Sleuthing has been Pwned!

Congratulations alicelallice, best of luck in capturing flags ahead!

#104	17 Sep 2025	RETIRED
CHALLENGE RANK	PWN DATE	CHALLENGE STATE

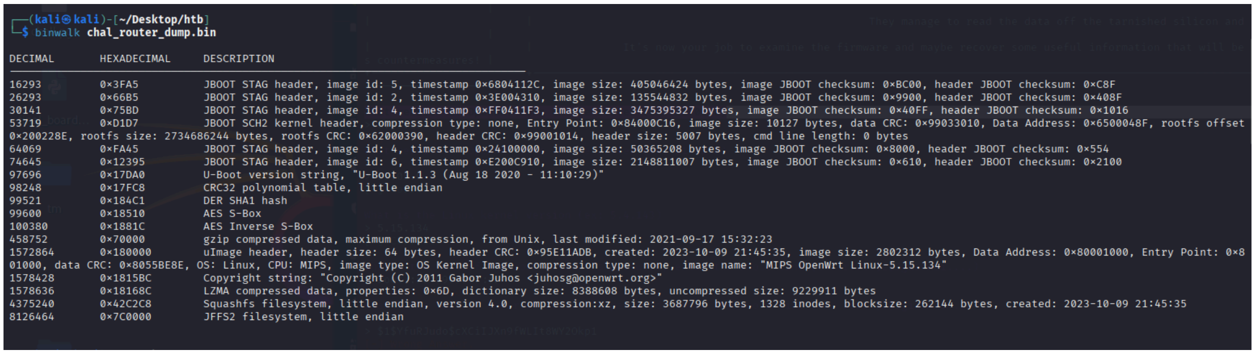
OpenWrt Router Firmware Forensics — Lab Writeup

Goal: Extract useful configuration and secrets from a router firmware image (chal_router_dump.bin) and document commands, findings, and screenshot suggestions so someone else can reproduce the work.We were given a raw router firmware image containing multiple partitions (uImage kernel, SquashFS rootfs, and a JFFS2 overlay). The tasks were to discover the OpenWrt version, Linux kernel, root password hash, PPPoE credentials, Wi-Fi SSID and password, and WAN→LAN DNAT ports.

> *What version of OpenWRT runs on the router (ex: 21.02.0)*

Identify Embedded Filesystems with `binwalk`

```
binwalk chal_router_dump.bin
```



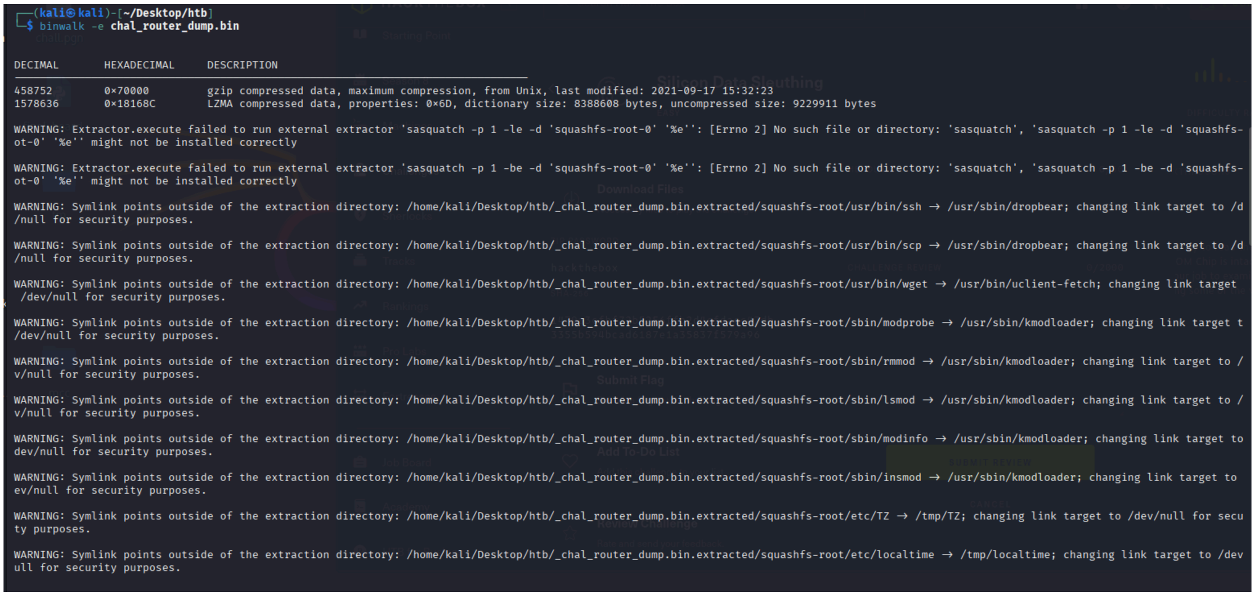
```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ binwalk chal_router_dump.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
16293         0x3FA5          JBOOT STAG header, image id: 5, timestamp 0x6804112C, image size: 405046424 bytes, image JBOOT checksum: 0xBC00, header JBOOT checksum: 0xC8F
26293         0x66B5          JBOOT STAG header, image id: 2, timestamp 0x3E004310, image size: 135544832 bytes, image JBOOT checksum: 0x9900, header JBOOT checksum: 0x408F
30141         0x75BD          JBOOT STAG header, image id: 4, timestamp 0xFF0411F3, image size: 3475395327 bytes, image JBOOT checksum: 0x40FF, header JBOOT checksum: 0x1016
53719         0xD1D7          JBOOT SCH2 kernel header, compression type: none, Entry Point: 0x84000C16, image size: 10127 bytes, data CRC: 0x99033010, Data Address: 0x6500048F, rootfs offset
0x200228E, rootfs size: 2734686244 bytes, rootfs CRC: 0x62000390, header CRC: 0x99001014, header size: 5007 bytes, cmd line length: 0 bytes
64069         0xFA45          JBOOT STAG header, image id: 4, timestamp 0x24100000, image size: 50365208 bytes, image JBOOT checksum: 0x8000, header JBOOT checksum: 0x554
74645         0x12395         JBOOT STAG header, image id: 6, timestamp 0xE200C910, image size: 2148811007 bytes, image JBOOT checksum: 0x610, header JBOOT checksum: 0x2100
97696         0x17DA0         U-Boot version string, "U-Boot 1.1.3 (Aug 18 2020 - 11:10:29)"
98248         0x17FC8         CRC32 polynomial table, little endian
99521         0x184C1         DER SHA1 hash
99600         0x18510         AES S-Box
100380        0x1881C         AES Inverse S-Box
458752        0x70000         gzip compressed data, maximum compression, from Unix, last modified: 2021-09-17 15:32:23
1572864       0x180000        uImage header, header size: 64 bytes, header CRC: 0x95E11ADB, created: 2023-10-09 21:45:35, image size: 2802312 bytes, Data Address: 0x80001000, Entry Point: 0x8
01000, data CRC: 0x8055BE8E, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: none, image name: "MIPS OpenWrt Linux-5.15.134"
1578428       0x1815BC        Copyright string: "Copyright (C) 2011 Gabor Juhos <juhosg@openwrt.org>"
1578636       0x18168C        LZMA compressed data, properties: 0x6D, dictionary size: 8388608 bytes, uncompressed size: 9229911 bytes
4375240       0x42C2C8        Squashfs filesystem, little endian, version 4.0, compression:xz, size: 3687796 bytes, 1328 inodes, blocksize: 262144 bytes, created: 2023-10-09 21:45:35
8126464       0x7C0000        JFFS2 filesystem, little endian
```

This scanned the binary for known signatures and revealed:

- Multiple **JBOOT headers** (custom bootloader format)
- A **U-Boot version string** (`U-Boot 1.1.3`)
- A **uImage kernel** (`OpenWrt Linux-5.15.134`)
- A **SquashFS filesystem** (offset: `0x42C2C8`)
- A **JFFS2 filesystem** (offset: `0x7C0000`)

Extract Filesystems with `binwalk -e`

```
binwalk -e chal_router_dump.bin
```



```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ binwalk -e chal_router_dump.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
458752        0×70000         gzip compressed data, maximum compression, from Unix, last modified: 2021-09-17 15:32:23
1578636       0×18168C        LZMA compressed data, properties: 0×6D, dictionary size: 8388608 bytes, uncompressed size: 9229911 bytes

WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -le -d 'squashfs-root-0' '%e'': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -le -d 'squashfs-
ot-0' '%e'' might not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -be -d 'squashfs-root-0' '%e'': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -be -d 'squashfs-
ot-0' '%e'' might not be installed correctly

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/usr/bin/ssh → /usr/sbin/dropbear; changing link target to /d
/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/usr/bin/scp → /usr/sbin/dropbear; changing link target to /d
/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/usr/bin/wget → /usr/bin/uclient-fetch; changing link target
 /dev/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/sbin/modprobe → /usr/sbin/kmodloader; changing link target t
/dev/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/sbin/rmmod → /usr/sbin/kmodloader; changing link target to /
v/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/sbin/lsmod → /usr/sbin/kmodloader; changing link target to /
v/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/sbin/modinfo → /usr/sbin/kmodloader; changing link target to
dev/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/sbin/insmod → /usr/sbin/kmodloader; changing link target to
ev/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/etc/TZ → /tmp/TZ; changing link target to /dev/null for secu
ty purposes.

WARNING: Symlink points outside of the extraction directory: /home/kali/Desktop/htb/_chal_router_dump.bin.extracted/squashfs-root/etc/localtime → /tmp/localtime; changing link target to /dev
ull for security purposes.
```

This attempted to extract embedded filesystems automatically. You hit a few snags:

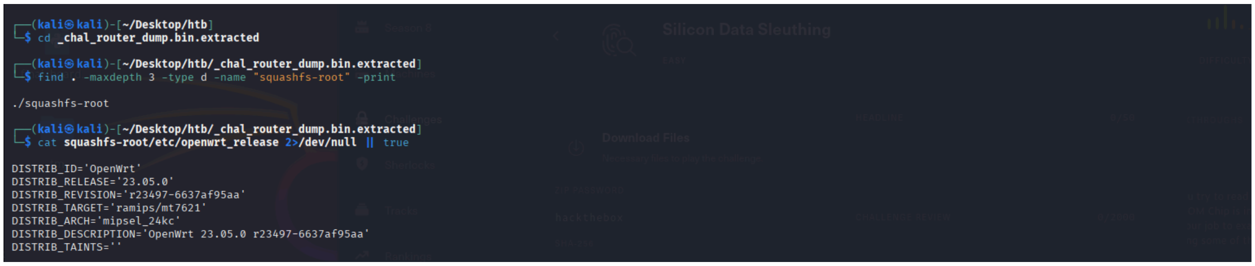
- **Missing** `sasquatch`: This tool is needed to extract SquashFS with non-standard compression (e.g., xz).
- **Missing** `jefferson`: Needed for JFFS2 extraction.
- **Symlink warnings**: Binwalk redirected unsafe symlinks to `/dev/null` for security.

Despite the warnings, binwalk successfully extracted:

- `squashfs-root`: the main root filesystem
- `jffs2-root`: persistent storage (though extraction failed due to missing `jefferson`)

Inspect Extracted Filesystem

```
cat squashfs-root/etc/openwrt_release 2>/dev/null || true
```



```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ cd _chal_router_dump.bin.extracted

┌──(kali㉿kali)-[~/Desktop/htb/_chal_router_dump.bin.extracted]
└─$ find . -maxdepth 3 -type d -name "squashfs-root" -print

./squashfs-root

┌──(kali㉿kali)-[~/Desktop/htb/_chal_router_dump.bin.extracted]
└─$ cat squashfs-root/etc/openwrt_release 2>/dev/null || true

DISTRIB_ID='OpenWrt'
DISTRIB_RELEASE='23.05.0'
DISTRIB_REVISION='r23497-6637af95aa'
DISTRIB_TARGET='ramips/mt7621'
DISTRIB_ARCH='mipsel_24kc'
DISTRIB_DESCRIPTION='OpenWrt 23.05.0 r23497-6637af95aa'
DISTRIB_TAINTS=''
```

This confirmed the firmware is:

- **OpenWrt 23.05.0**
- Target: `ramips/mt7621` (MIPS-based SoC)
- Architecture: `mipsel_24kc`

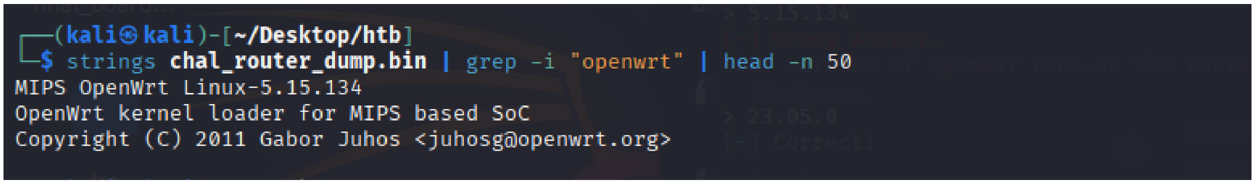
Answer: **23.05.0**

> *What is the Linux kernel version (ex: 5.4.143)*

Run these first — often they immediately show the version.

1. `strings` + `grep` (very quick)

```
strings chal_router_dump.bin | grep -i "openwrt" | head -n 50
```



```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ strings chal_router_dump.bin | grep -i "openwrt" | head -n 50
MIPS OpenWrt Linux-5.15.134
OpenWrt kernel loader for MIPS based SoC
Copyright (C) 2011 Gabor Juhos <juhosg@openwrt.org>
```

Answer: **5.15.134**

> *What's the hash of the root account's password, enter the whole line (ex: root:$2$JgiaOAai....)*

Prereqs

- Kali (or similar) with `binwalk`, `squashfs-tools` and Python3.
- `jefferson` for JFFS2 extraction (install inside a venv as shown).

Minimal, exact steps

1. Create and activate a Python venv and install `jefferson` (one-time):

```
python3 -m venv ~/jefferson-env
source ~/jefferson-env/bin/activate
pip install jefferson
```

(You only need to run the venv steps once; afterward just `source ~/jefferson-env/bin/activate`.)

1. Identify partitions with `binwalk` (confirm where SquashFS / JFFS2 live):

```
binwalk chal_router_dump.bin
```

Look for lines indicating `Squashfs filesystem` and `JFFS2 filesystem`. Note their offsets (binwalk prints them).

1. Carve out the JFFS2 partition (use the offset from binwalk; example offset was `0x7C0000` = `8126464`):

```
dd if=chal_router_dump.bin of=fs.jffs2 bs=1 skip=8126464 status=progress
file fs.jffs2
```

`file` confirms it's a JFFS2 image.

1. Extract JFFS2 with `jefferson` (this writes a `jffs2-root/` directory):

```
jefferson fs.jffs2
ls -la jffs2-root
```

You should see `upper/` and `work/` entries; often `upper/sysupgrade.tgz` will be present.

1. List the tarball contents (if present) to see what files are inside the overlay:

```
tar -tzf jffs2-root/upper/sysupgrade.tgz | sed -n '1,200p
```

Look for `etc/shadow`, `etc/passwd`, `etc/config/*`, etc.

1. Extract `/etc/shadow` from the sysupgrade tarball (to a temp directory) and check it:

```
mkdir -p /tmp/jffs2_extract
tar -xzf jffs2-root/upper/sysupgrade.tgz -C /tmp/jffs2_extract ./etc/shadow 2
>/dev/null || true
sed -n '1,200p' /tmp/jffs2_extract/etc/shadow 2>/dev/null || true
```

If it prints nothing, the shadow might exist elsewhere in the dump—search the entire `jffs2-root` tree next.

1. Grep the entire extracted JFFS2 dump for any `root:` lines (fast and reliable):

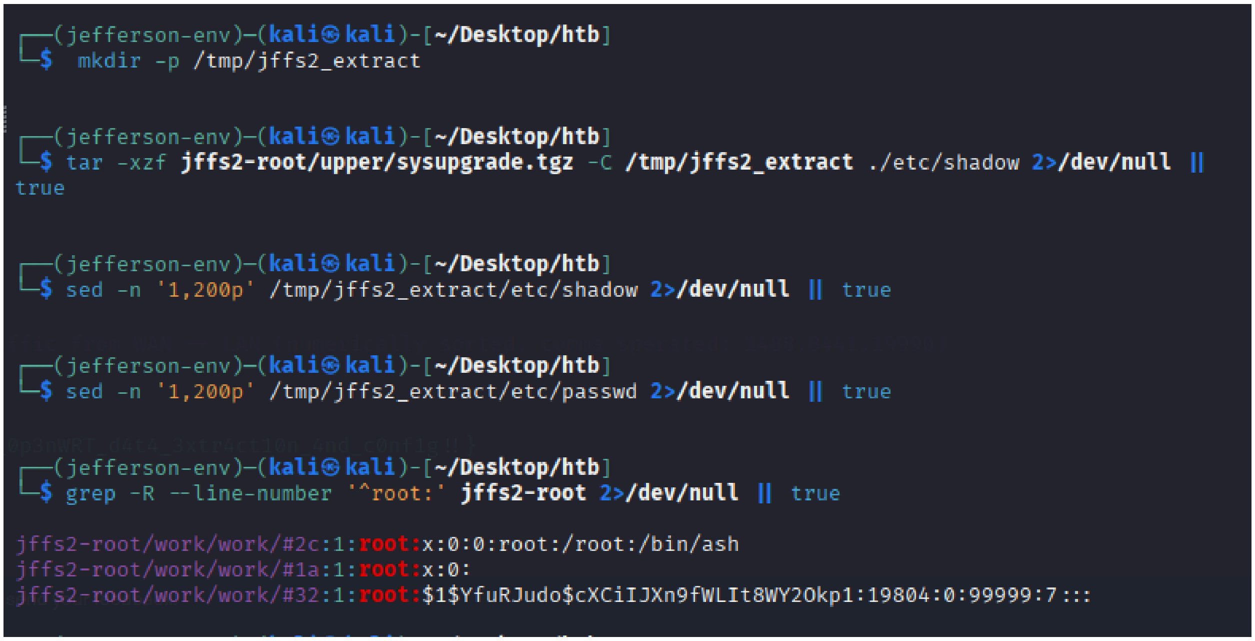
```
grep -R --line-number '^root:' jffs2-root 2>/dev/null || true
```

This finds *all* files containing a `root:` line across `upper/` and `work/`. In this firmware the relevant entry was inside a `work/work/#32` file produced by `jefferson`.

1. Narrow the search to hash-like patterns (common hash prefixes `$1$`, `$6$`, `$2y$`, etc.):

```
grep -R --line-number -E '^root:[^:]*\$[126y]\$|^root:[^:]*\$2[aby]\$' jffs2-roo
t 2>/dev/null || true
```

This helps ignore simple `root:x` or empty-root entries and shows the file containing the actual hash.



```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ mkdir -p /tmp/jffs2_extract

┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ tar -xzf jffs2-root/upper/sysupgrade.tgz -C /tmp/jffs2_extract ./etc/shadow 2>/dev/null ||
true

┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ sed -n '1,200p' /tmp/jffs2_extract/etc/shadow 2>/dev/null || true

┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ sed -n '1,200p' /tmp/jffs2_extract/etc/passwd 2>/dev/null || true

┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ grep -R --line-number '^root:' jffs2-root 2>/dev/null || true

jffs2-root/work/work/#2c:1:root:x:0:0:root:/root:/bin/ash
jffs2-root/work/work/#1a:1:root:x:0:
jffs2-root/work/work/#32:1:root:$1$YfuRJudo$cXCiIJXn9fWLIt8WY2Okp1:19804:0:99999:7:::
```

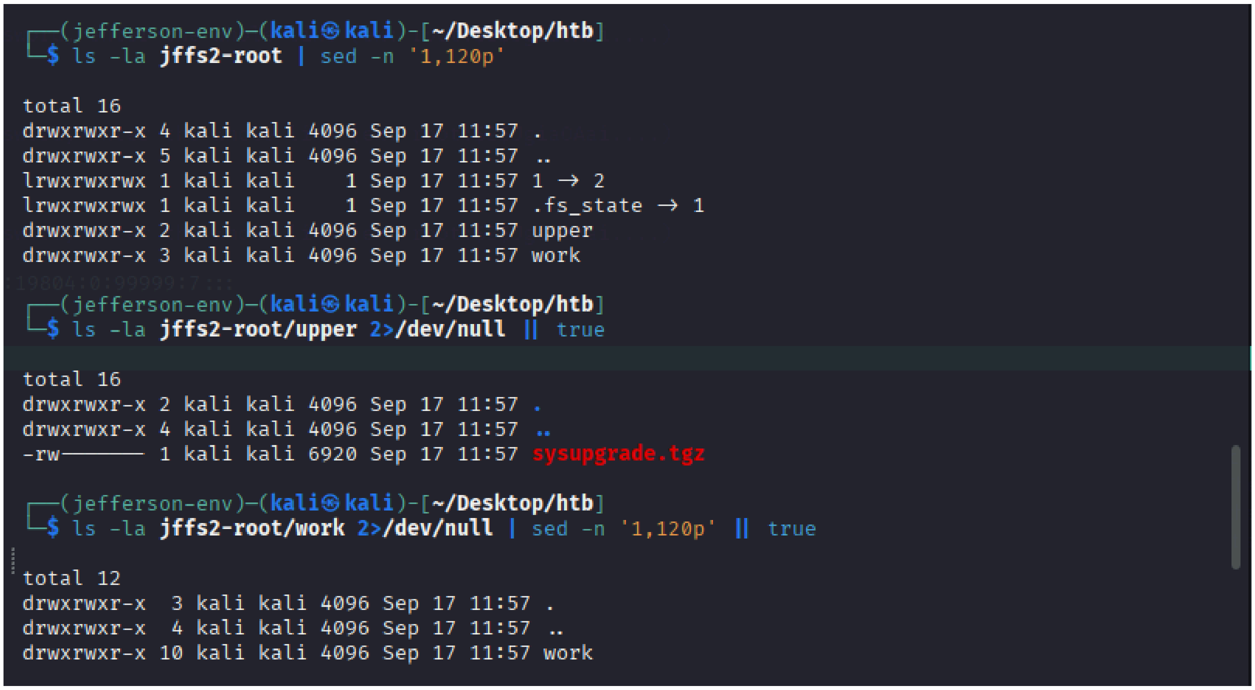
Answer: **root:$1$YfuRJudo$cXCiIJXn9fWLIt8WY2Okp1:19804:0:99999:7:::**

> *What is the PPPoE username*

Quick checklist (confirm extraction)

```
# show the important dirs
ls -la squashfs-root     # read-only factory files
ls -la jffs2-root        # overlay (upper/ and work/)
ls -la jffs2-root/upper  # often contains sysupgrade.tgz
```

Why: PPP and Wi-Fi overrides are often in the overlay (`jffs2-root/upper/sysupgrade.tgz` or inside `work/`

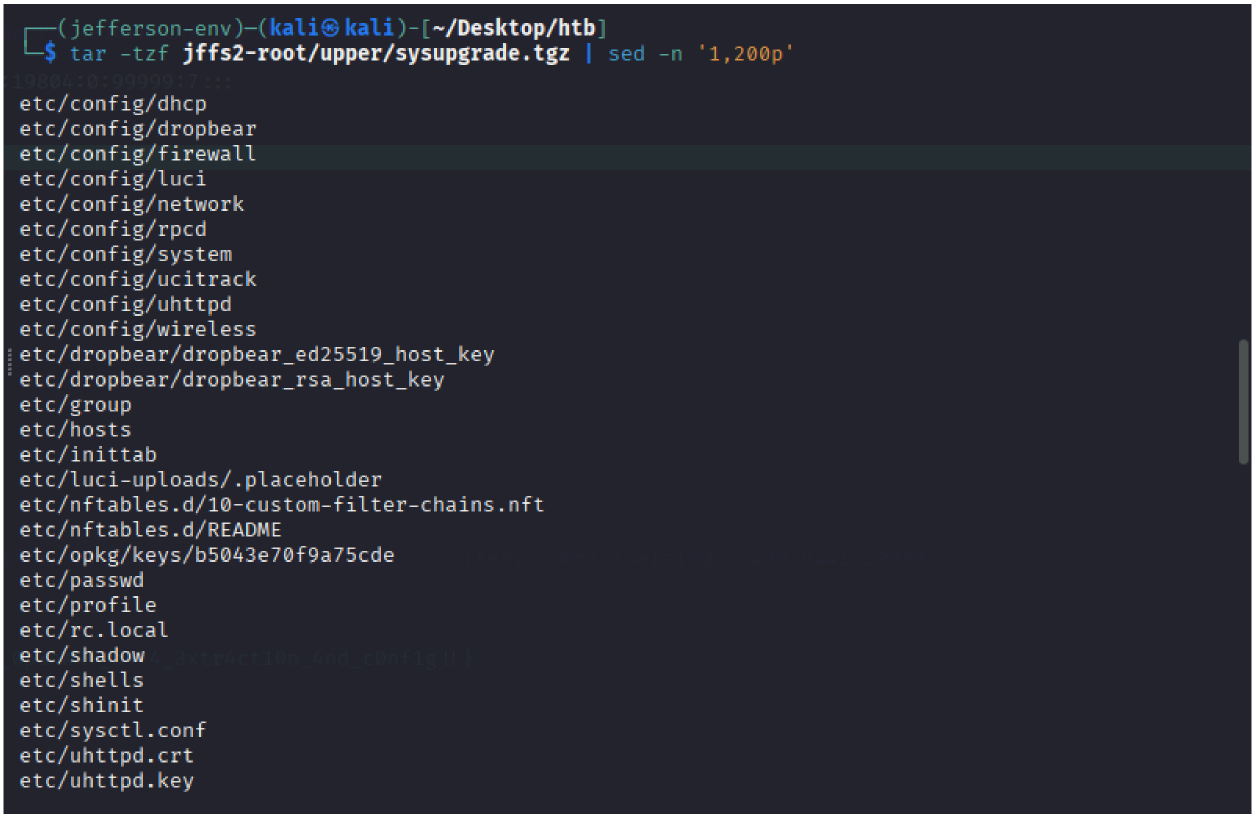


```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ ls -la jffs2-root | sed -n '1,120p'

total 16
drwxrwxr-x 4 kali kali 4096 Sep 17 11:57 .
drwxrwxr-x 5 kali kali 4096 Sep 17 11:57 ..
lrwxrwxrwx 1 kali kali    1 Sep 17 11:57 1 → 2
lrwxrwxrwx 1 kali kali    1 Sep 17 11:57 .fs_state → 1
drwxrwxr-x 2 kali kali 4096 Sep 17 11:57 upper
drwxrwxr-x 3 kali kali 4096 Sep 17 11:57 work

┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ ls -la jffs2-root/upper 2>/dev/null || true

total 16
drwxrwxr-x 2 kali kali 4096 Sep 17 11:57 .
drwxrwxr-x 4 kali kali 4096 Sep 17 11:57 ..
-rw─────── 1 kali kali 6920 Sep 17 11:57 sysupgrade.tgz

┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ ls -la jffs2-root/work 2>/dev/null | sed -n '1,120p' || true

total 12
drwxrwxr-x  3 kali kali 4096 Sep 17 11:57 .
drwxrwxr-x  4 kali kali 4096 Sep 17 11:57 ..
drwxrwxr-x 10 kali kali 4096 Sep 17 11:57 work
```

Inspect overlay tarball (common place)

```
# list contents of sysupgrade.tgz (if present)
tar -tzf jffs2-root/upper/sysupgrade.tgz | sed -n '1,200p'
```

Look for: `etc/config/network`, `etc/config/wireless`, `etc/ppp/chap-secrets` or `etc/shadow`.



```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ tar -tzf jffs2-root/upper/sysupgrade.tgz | sed -n '1,200p'
etc/config/dhcp
etc/config/dropbear
etc/config/firewall
etc/config/luci
etc/config/network
etc/config/rpcd
etc/config/system
etc/config/ucitrack
etc/config/uhttpd
etc/config/wireless
etc/dropbear/dropbear_ed25519_host_key
etc/dropbear/dropbear_rsa_host_key
etc/group
etc/hosts
etc/inittab
etc/luci-uploads/.placeholder
etc/nftables.d/10-custom-filter-chains.nft
etc/nftables.d/README
etc/opkg/keys/b5043e70f9a75cde
etc/passwd
etc/profile
etc/rc.local
etc/shadow
etc/shells
etc/shinit
etc/sysctl.conf
etc/uhttpd.crt
etc/uhttpd.key
```

Extract the relevant files to a temp directory

```
mkdir -p /tmp/jffs2_extract
tar -xzf jffs2-root/upper/sysupgrade.tgz -C /tmp/jffs2_extract \
  ./etc/config/network ./etc/config/wireless ./etc/ppp/chap-secrets ./etc/ppp/pap-secrets 2>/dev/null || true
```

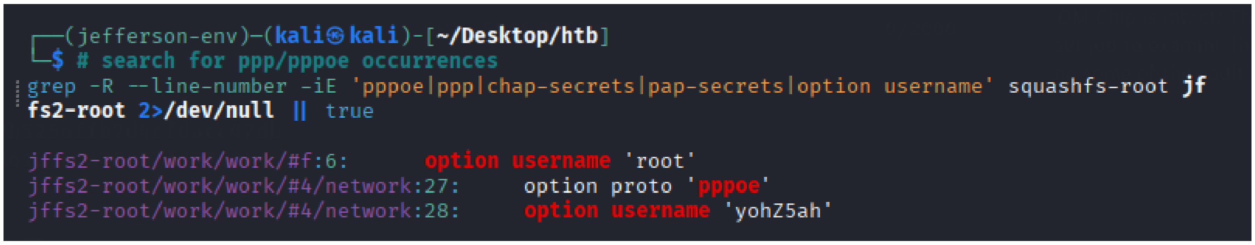
Why: Extract only the files we need to inspect safely.

PPPoE username & password

Common places:

- `/etc/config/network` (OpenWrt style `option username` / `option password`)
- `/etc/ppp/chap-secrets` or `pap-secrets`
- If `option username '...'` found → the username string.

```
grep -R --line-number -iE 'pppoe|ppp|chap-secrets|pap-secrets|option username' squashfs-root jffs2-root 2>/dev/null || true
```



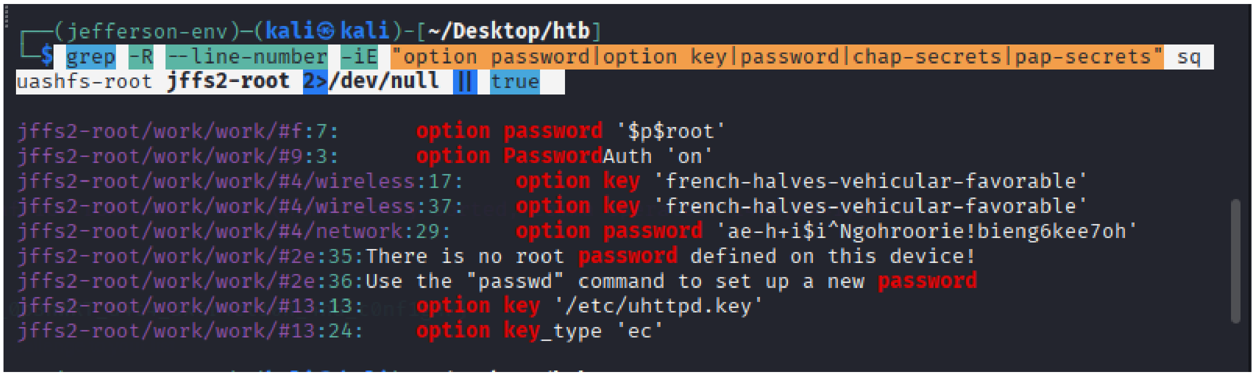
```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ # search for ppp/pppoe occurrences
grep -R --line-number -iE 'pppoe|ppp|chap-secrets|pap-secrets|option username' squashfs-root jf
fs2-root 2>/dev/null || true

jffs2-root/work/work/#f:6:      option username 'root'
jffs2-root/work/work/#4/network:27:     option proto 'pppoe'
jffs2-root/work/work/#4/network:28:     option username 'yohZ5ah'
```

Answer: **yohZ5ah**

> *What is the PPPoE password*

If `option password '...'` found → submit the password string.

```
 grep -R --line-number -iE "option password|option key|password|chap-secrets|pap-secrets" squashfs-root jffs2-root 2>/dev/null || true
```



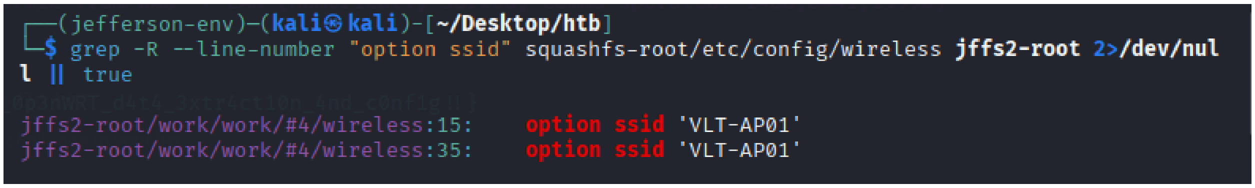
```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ grep -R --line-number -iE "option password|option key|password|chap-secrets|pap-secrets" sq
uashfs-root jffs2-root 2>/dev/null || true

jffs2-root/work/work/#f:7:      option password '$p$root'
jffs2-root/work/work/#9:3:      option PasswordAuth 'on'
jffs2-root/work/work/#4/wireless:17:    option key 'french-halves-vehicular-favorable'
jffs2-root/work/work/#4/wireless:37:    option key 'french-halves-vehicular-favorable'
jffs2-root/work/work/#4/network:29:     option password 'ae-h+i$i^Ngohroorie!bieng6kee7oh'
jffs2-root/work/work/#2e:35:There is no root password defined on this device!
jffs2-root/work/work/#2e:36:Use the "passwd" command to set up a new password
jffs2-root/work/work/#13:13:    option key '/etc/uhttpd.key'
jffs2-root/work/work/#13:24:    option key_type 'ec'
```

Answer: **ae-h+i$i^Ngohroorie!bieng6kee7oh**

> *What is the WiFi SSID*

Primary file: `/etc/config/wireless` (or inside the overlay tarball)

```
grep -R --line-number "option ssid" squashfs-root/etc/config/wireless jffs2-root 2>/dev/null || true
```

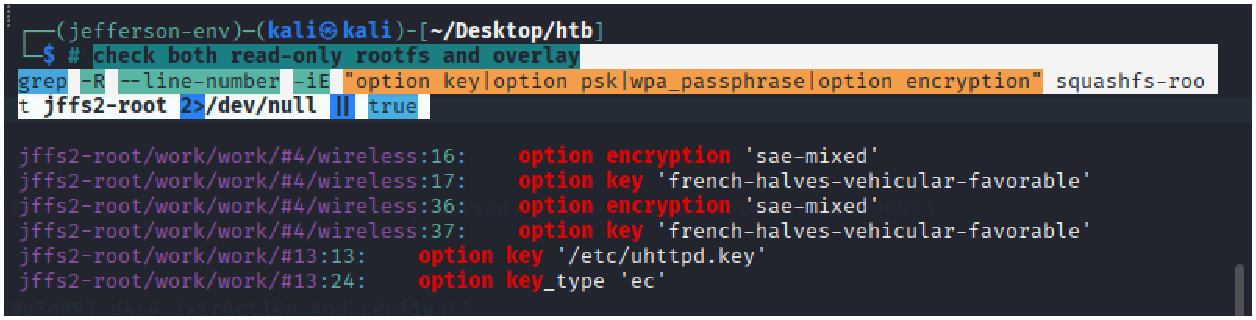


```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ grep -R --line-number "option ssid" squashfs-root/etc/config/wireless jffs2-root 2>/dev/nul
l || true
jffs2-root/work/work/#4/wireless:15:    option ssid 'VLT-AP01'
jffs2-root/work/work/#4/wireless:35:    option ssid 'VLT-AP01'
```

- SSID string found after `option ssid` (example: `VLT-AP01`)

> *What is the WiFi Password*

check both read-only rootfs and overlay `grep -R --line-number -iE "option key|option psk|wpa_passphrase|option encryption" squashfs-root jffs2-root 2>/dev/null || true`



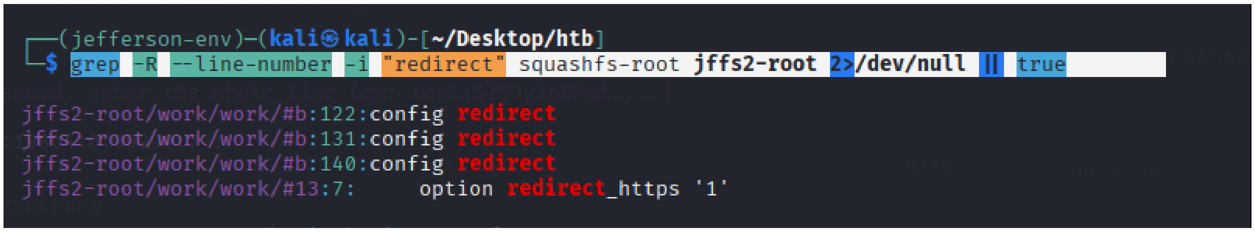
```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ # check both read-only rootfs and overlay
grep -R --line-number -iE "option key|option psk|wpa_passphrase|option encryption" squashfs-roo
t jffs2-root 2>/dev/null || true

jffs2-root/work/work/#4/wireless:16:    option encryption 'sae-mixed'
jffs2-root/work/work/#4/wireless:17:    option key 'french-halves-vehicular-favorable'
jffs2-root/work/work/#4/wireless:36:    option encryption 'sae-mixed'
jffs2-root/work/work/#4/wireless:37:    option key 'french-halves-vehicular-favorable'
jffs2-root/work/work/#13:13:    option key '/etc/uhttpd.key'
jffs2-root/work/work/#13:24:    option key_type 'ec'
```

- Wi-Fi password from `option key` (example: `french-halves-vehicular-favorable`)

> *What are the 3 WAN ports that redirect traffic from WAN → LAN (numerically sorted, comma sperated: 1488,8441,19990)*

Search firewall config or any redirect blocks in the overlay:

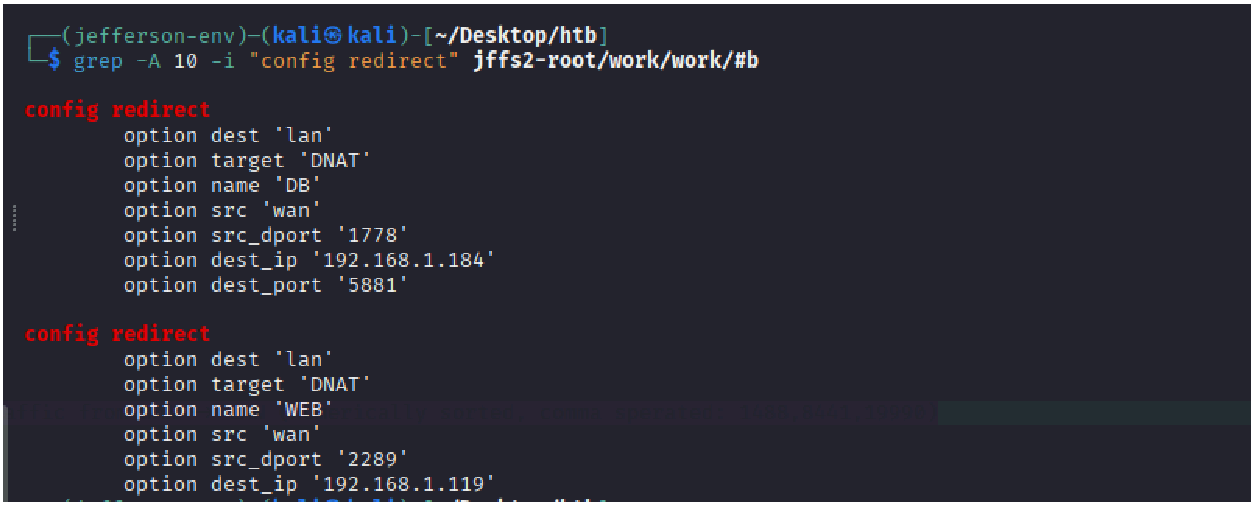
```
grep -R --line-number -i "redirect" squashfs-root jffs2-root 2>/dev/null || true
```



```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ grep -R --line-number -i "redirect" squashfs-root jffs2-root 2>/dev/null || true

jffs2-root/work/work/#b:122:config redirect
jffs2-root/work/work/#b:131:config redirect
jffs2-root/work/work/#b:140:config redirect
jffs2-root/work/work/#13:7:     option redirect_https '1'
```

Each `config redirect` block will contain `option src 'wan'` and `option src_dport 'NNNN'` — the `src_dport` values are the WAN ports being redirected to LAN.

```
grep -A 10 -i "config redirect" jffs2-root/work/work/#b
```



```
┌──(jefferson-env)─(kali㉿kali)-[~/Desktop/htb]
└─$ grep -A 10 -i "config redirect" jffs2-root/work/work/#b

config redirect
        option dest 'lan'
        option target 'DNAT'
        option name 'DB'
        option src 'wan'
        option src_dport '1778'
        option dest_ip '192.168.1.184'
        option dest_port '5881'

config redirect
        option dest 'lan'
        option target 'DNAT'
        option name 'WEB'
        option src 'wan'
        option src_dport '2289'
        option dest_ip '192.168.1.119'
```

- `DB` → `src_dport '1778'`
- `WEB` → `src_dport '2289'`
- `NAS` → `src_dport '8088'`

These are the **WAN-side ports** that accept incoming traffic and redirect it internally to LAN destinations.

Answer: **1778,2289,8088**

by submitting that last questions we will get our flag

Tools

- `binwalk`, `strings`, `dd`, `unsquashfs` / `squashfs-tools`, `jefferson`, `tar`, `grep`, `sed`, `awk`.

What I did (high level)

1. `binwalk` + `strings` to locate embedded images (uImage, SquashFS, JFFS2).
2. Extracted SquashFS (`unsquashfs` / `binwalk -e`) and read `/etc` for OpenWrt info.
3. Carved JFFS2 (offset from binwalk), extracted it with `jefferson`.
4. Listed and inspected `jffs2-root/upper/sysupgrade.tgz` (tarball) and `jffs2-root/work/*` fragments.
5. Grepped extracted files for `root:`, `option username`, `option password`, `option ssid`, `option key`, and `config redirect` to get exact values.

Exact files checked

- `squashfs-root/etc/openwrt_release`, `/etc/banner` (OpenWrt version)
- `uImage` header / `strings` (kernel version)
- `jffs2-root/upper/sysupgrade.tgz` → `etc/shadow`, `etc/config/network`, `etc/config/wireless`
- `jffs2-root/work/...` (jefferson-produced fragments)
- `jffs2-root` firewall fragments for `config redirect` blocks

Key commands (one-liners)

- `binwalk chal_router_dump.bin`
- `dd if=chal_router_dump.bin of=fs.jffs2 bs=1 skip=<offset>`
- `jefferson fs.jffs2`
- `tar -tzf jffs2-root/upper/sysupgrade.tgz`
- `tar -xzf jffs2-root/upper/sysupgrade.tgz -C /tmp/extract ./etc/shadow ./etc/config/network ./etc/config/wireless`
- `grep -R --line-number '^root:' jffs2-root`
- `grep -R --line-number -iE 'option ssid|option key|option username|option password|config redirect' jffs2-root /tmp/extract squashfs-root`

Results (answers you submitted)

- **OpenWrt version:** (found in `/etc/openwrt_release`) — *you discovered it earlier*
- **Kernel version:** `5.15.134`

- **Root `/etc/shadow` line:** `root:$1$YfuRJudo$cXCilJXn9fWLIt8WY2Okp1:19804:0:99999:7:::`
- **PPPoE username:** `yohZ5ah`
- **PPPoE password:** `ae-h+i$i^Ngohroorie!bieng6kee7oh`
- **Wi-Fi SSID:** `VLT-AP01`
- **Wi-Fi Password:** `french-halves-vehicular-favorable`
- **WAN→LAN redirect ports (sorted):** `1778,2289,8088`
- **Flag obtained:** `HTB{Y0u'v3_m4st3r3d_0p3nWRT_d4t4_3xtr4ct10n_4nd_c0nf1g!!}`