Firmware Extraction

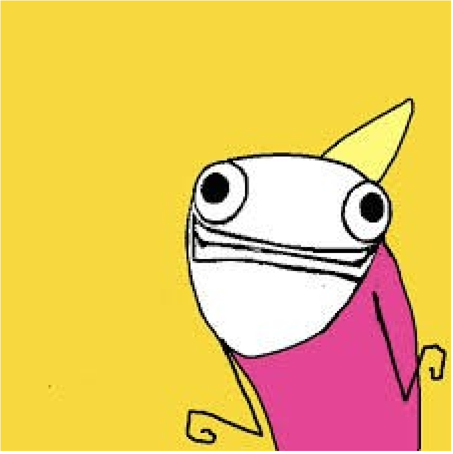
Hack.lu 2019

Pauline Bourmeau

« Snarf it »

Motivations

- Curiosity !
- Learning challenge
- Get root \o/

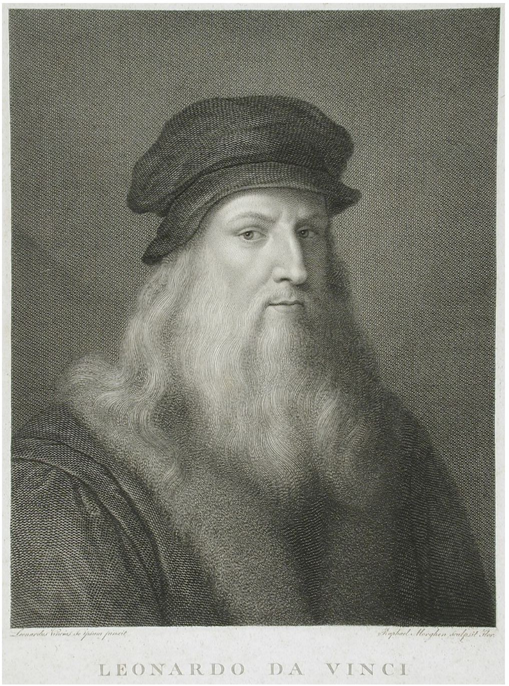


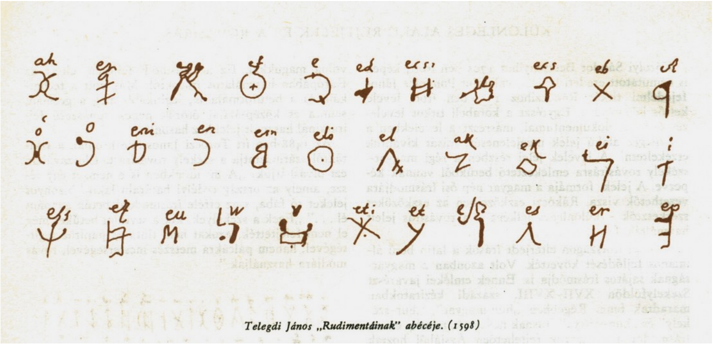
Hyperbole and a half –
Annie Brosh

- « what’s inside the box », getting the ropes of linux systems
- Challenge myself
- Teach friends

Who am I ?

- IT background
- Linguistics
- Use to be a teacher
- Passionate about human thinking and history




Telegdi János „Rudimentáinak" abécéje. (1598)

Sharing with you

- Mistakes
- Questions
- Notions

- Introduction level
  - Start digging
  - Get a hacking project
  - Fun
  - Discovery, new places...

Can I do it ?

- Intuitive
- Requires no knowledge in electronics to start
- Problems about « embedded » system
- Step by Step workshop, with choices

You’ll know how to

1. Examine the hardware, find a serial port
2. Test the pins, connect the adapter
3. Set up of a minicom working environment
4. Extract the firmware
5. Uncompress the firmware for analysis

Open the « box »

- Physical access to the router, why is it cool ?
- Open it and see what's inside – care and tools
- Gather information about the hardware - eyes and click
- GOAL → get a root shell and extract the firmware

Targets

- GliNet Mango router
- Netgear D300 router

?

?

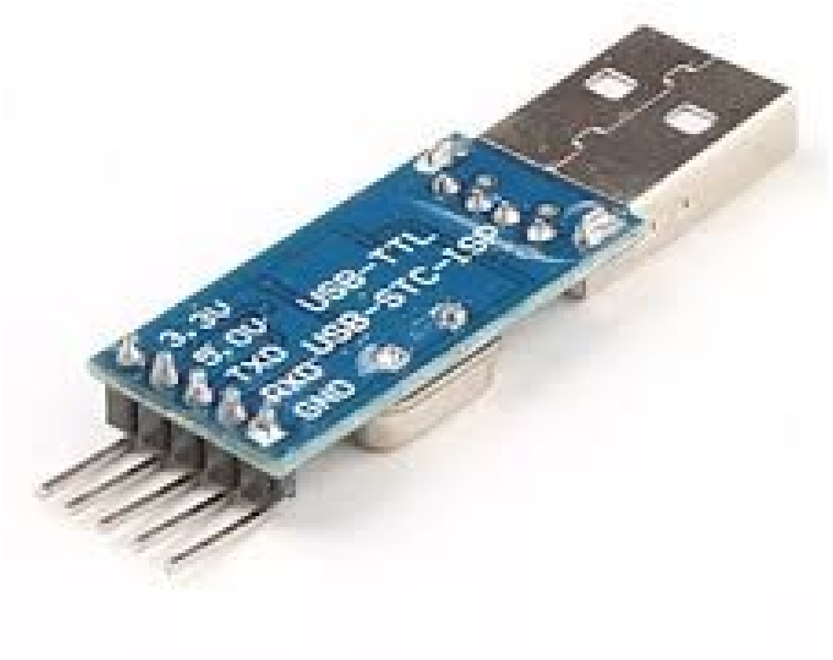
- Both recent and cheap
- GliNet comes with USB port \o/
- Mango is good for custom VPN

Why uart ?

- It’s easy and cheap, you wont break anything
- Root console
- Access to : Boot, filesystem, execute binaries…

UART-USB (TTL) adapter

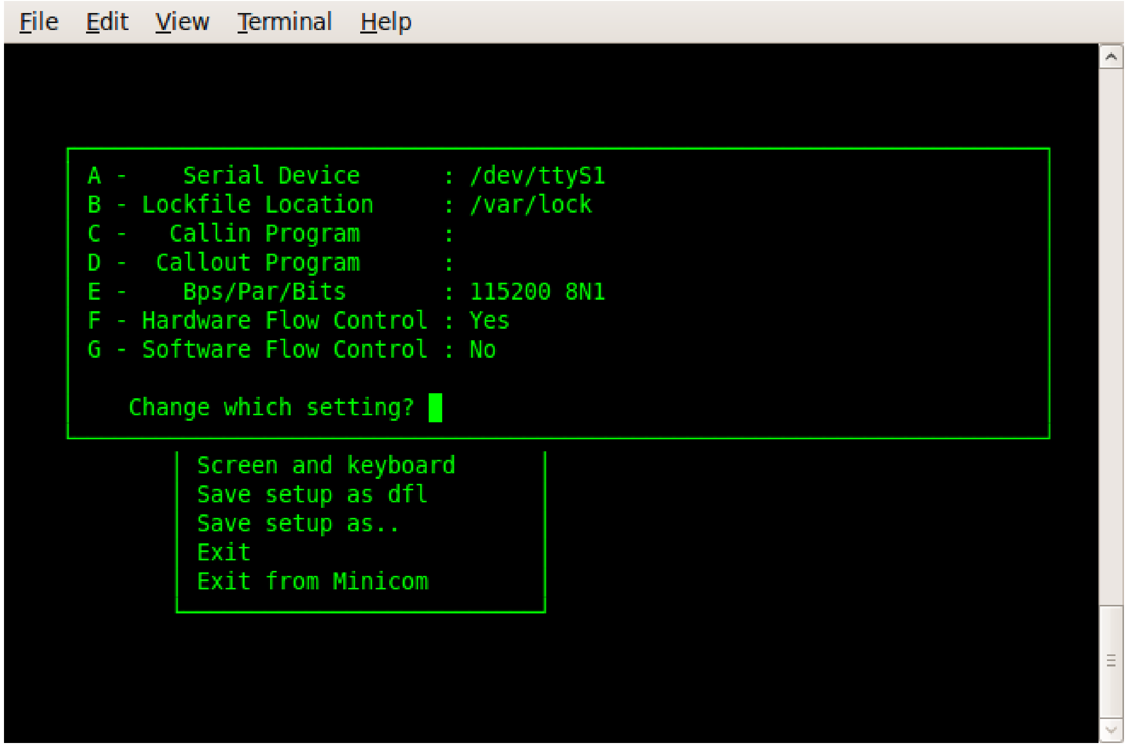
(Universal asynchronous receiver-transmitter)



Expl : Cost around 2 euros on eBay

Minicom

- Setting up a (remote) serial console
- Connect to embed linux (like) systems
- Menu and options
- Runs in terminal



```
A -    Serial Device      : /dev/ttyS1
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Bps/Par/Bits       : 115200 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No

   Change which setting?
```

```
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom
```

Image youtube.com

Netgear router

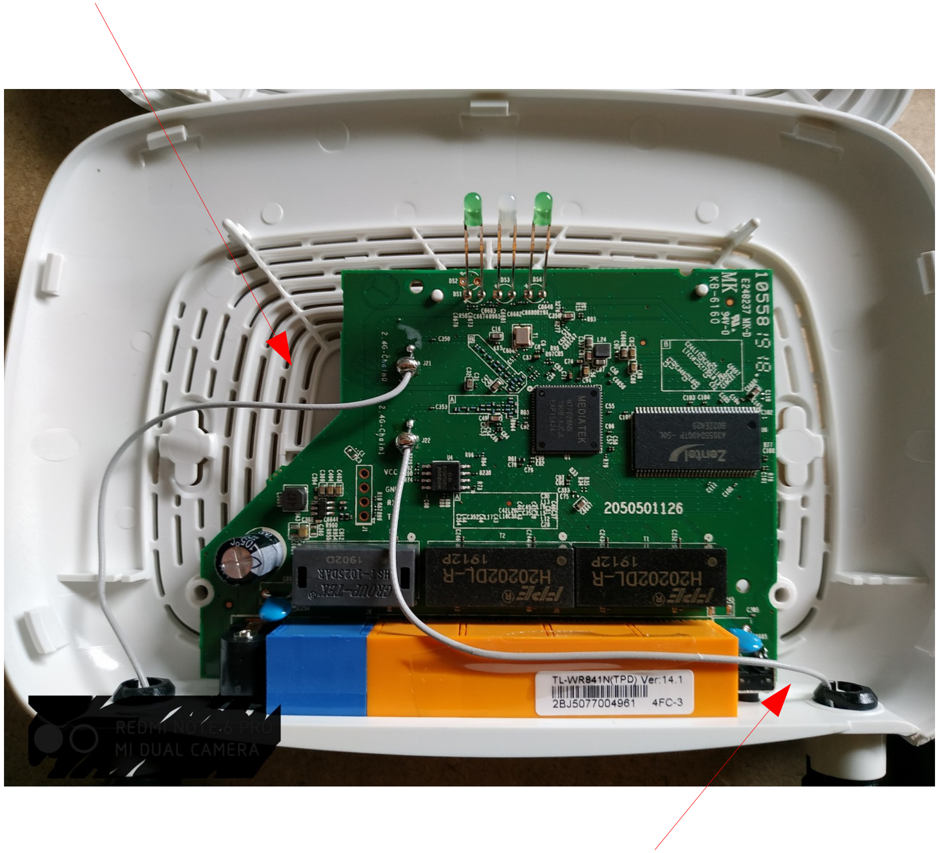
- No usb
- Open-WRT as firmware
- Simple home router

Inspect the device

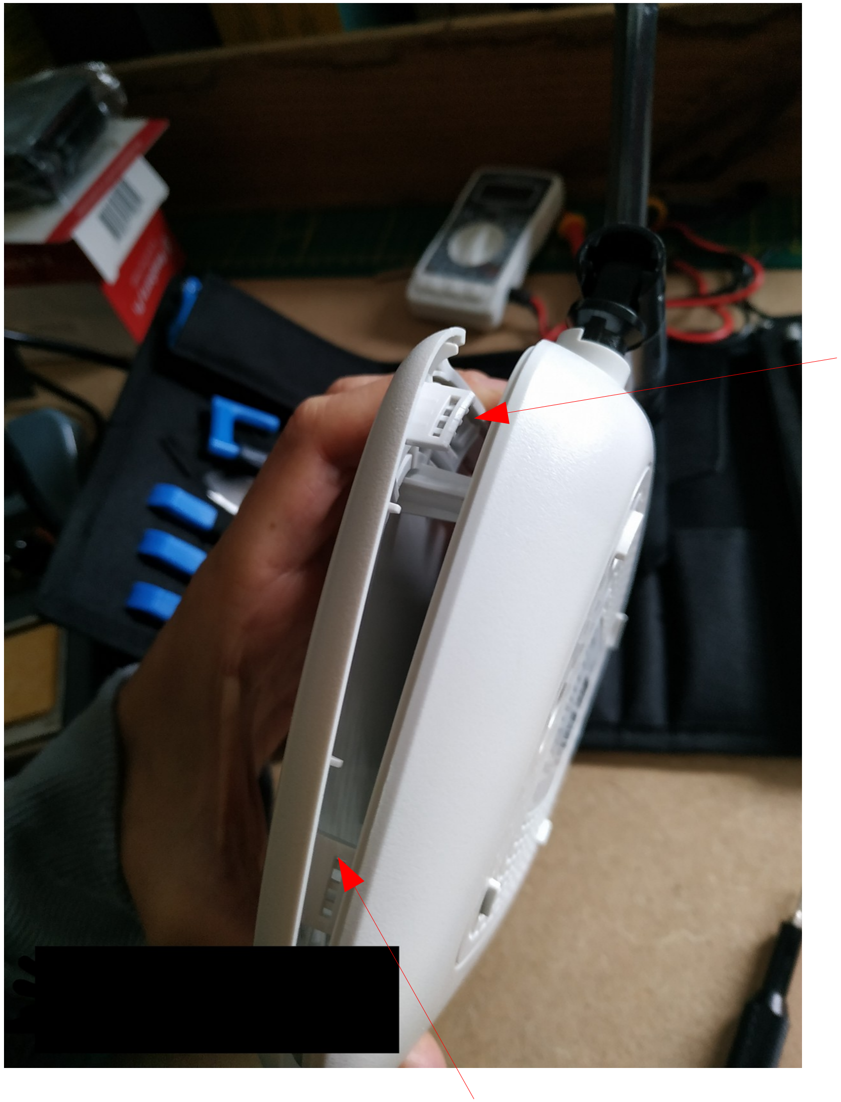
- Open without break, careful to wires of antennas, components…
- Is a serial port accessible ?
- What pins are needed ?

- I see the pins, test with multimeter now (to confirm)

Opening the box 1/3



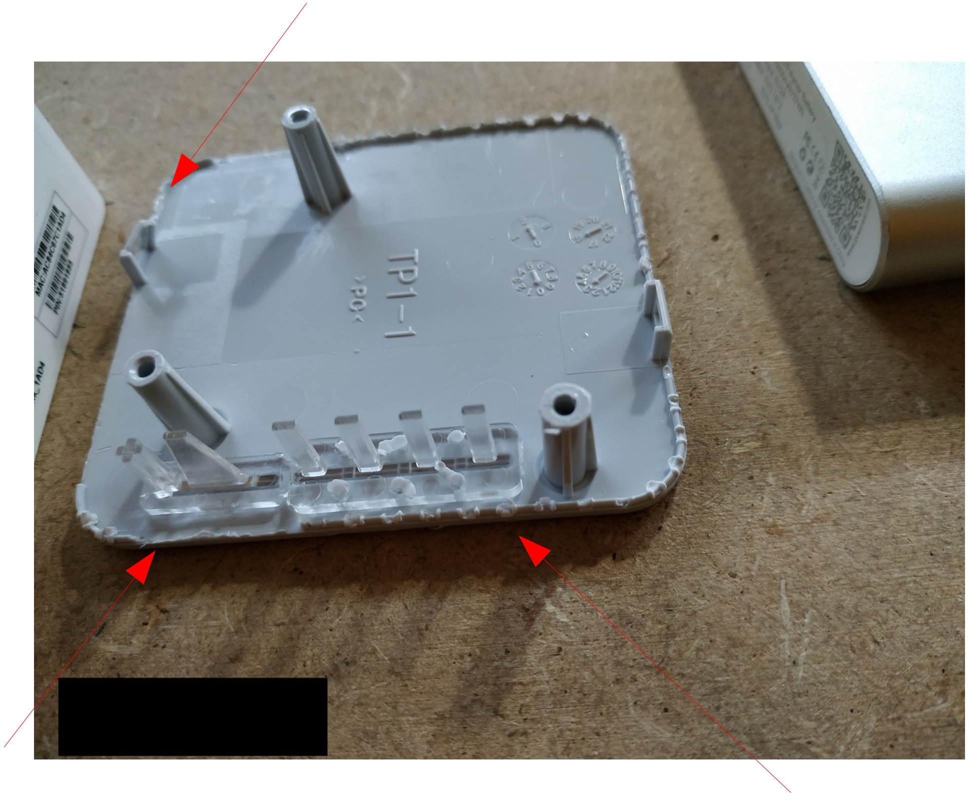
[redacted]



[redacted]

Opening the box 2/2

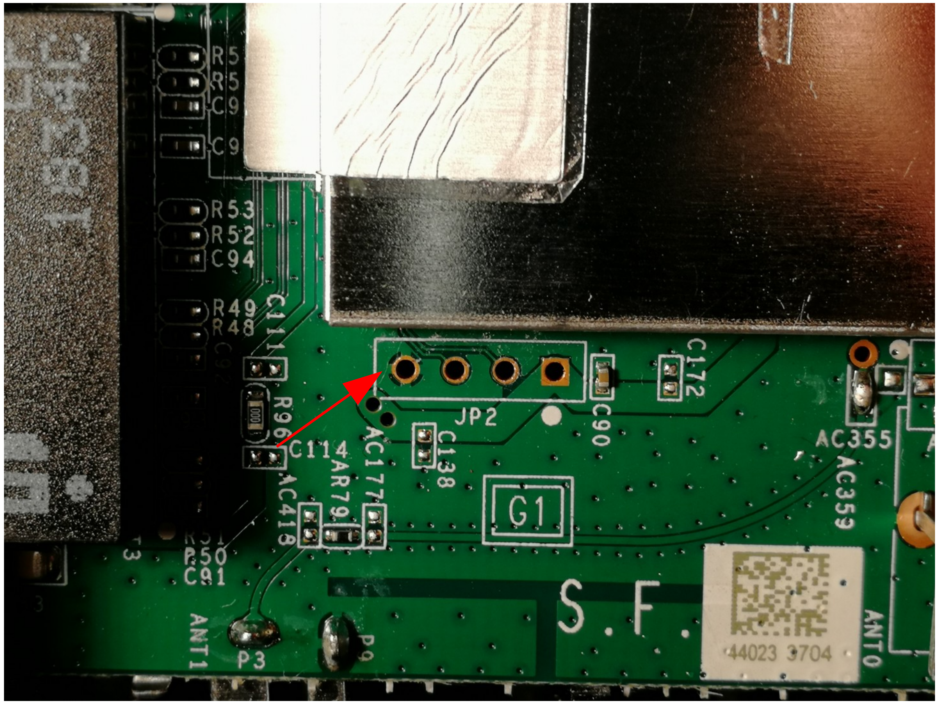
pain

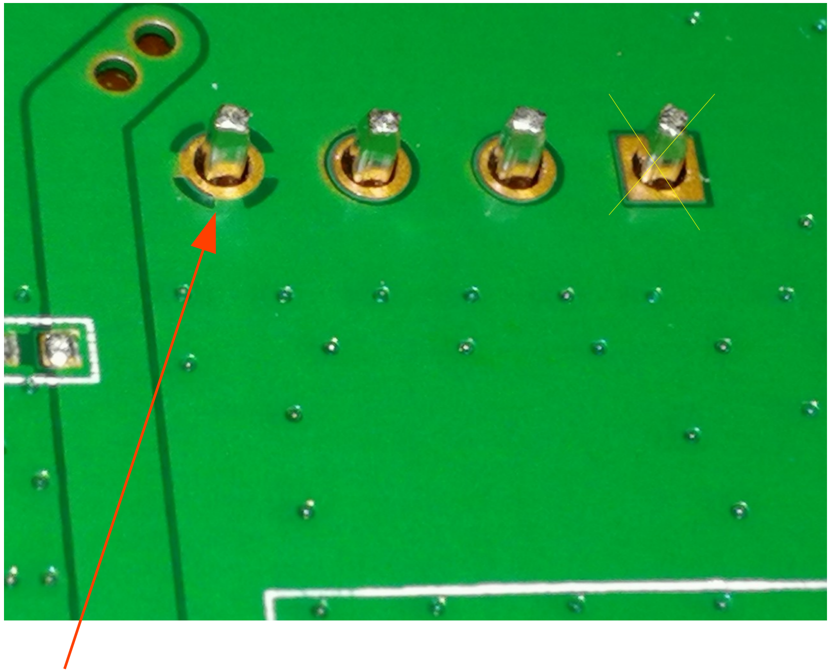


[redacted]

Find, identify, test, solder

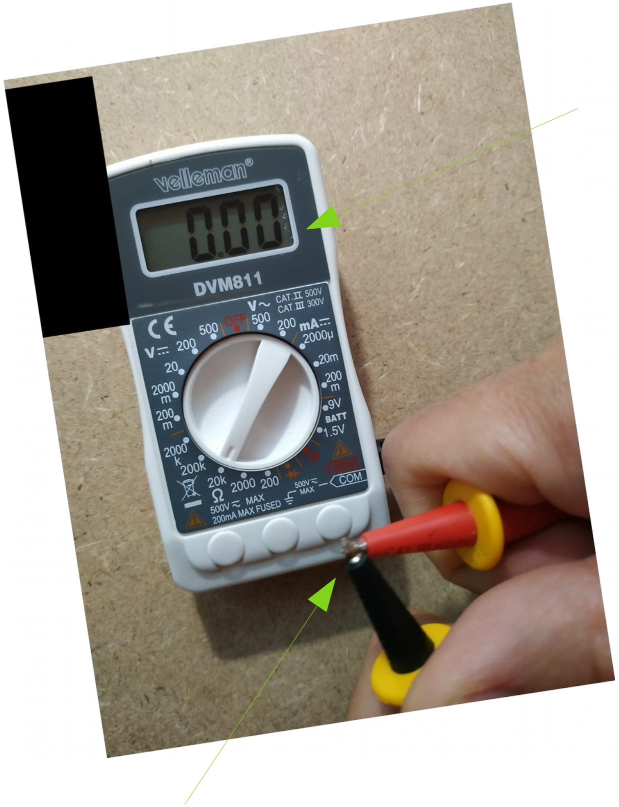
- Ground
- RX
- TX



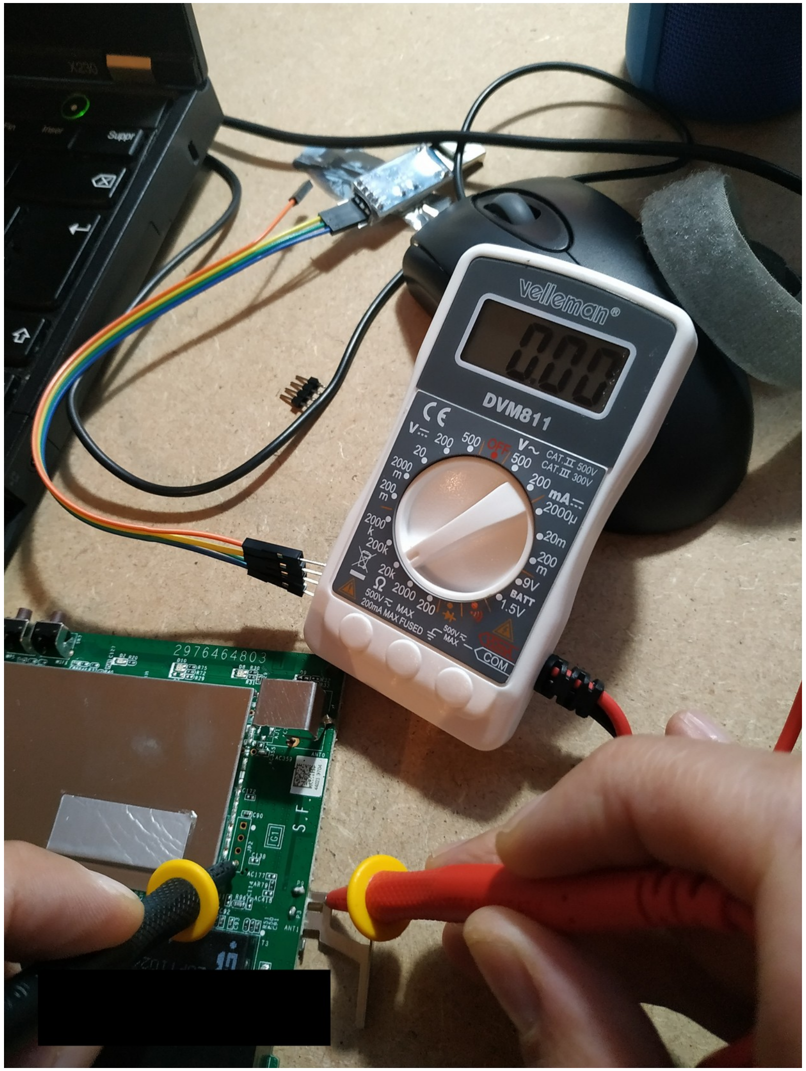


Find, identify, test, solder

- Continuity test



[redacted]

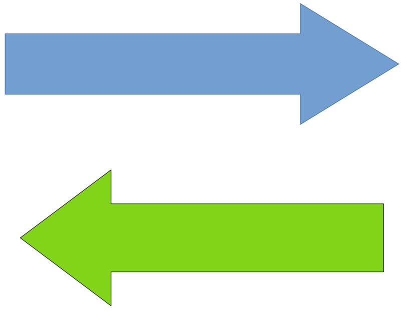


[redacted]

Serial communication interface

Hardware level

1 bit at a time, device to computer, here for debug purpose



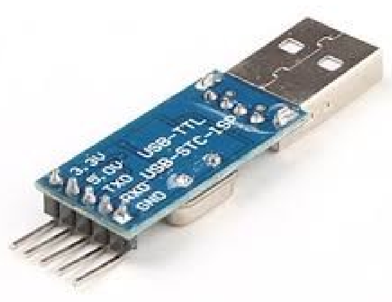
Transmit is TX, or TX0, TX1...

Receive is RX, RX0, RX1...

Or something else ! :)

Use TTL – as TTL Serial communication (transistor to transistor logic)

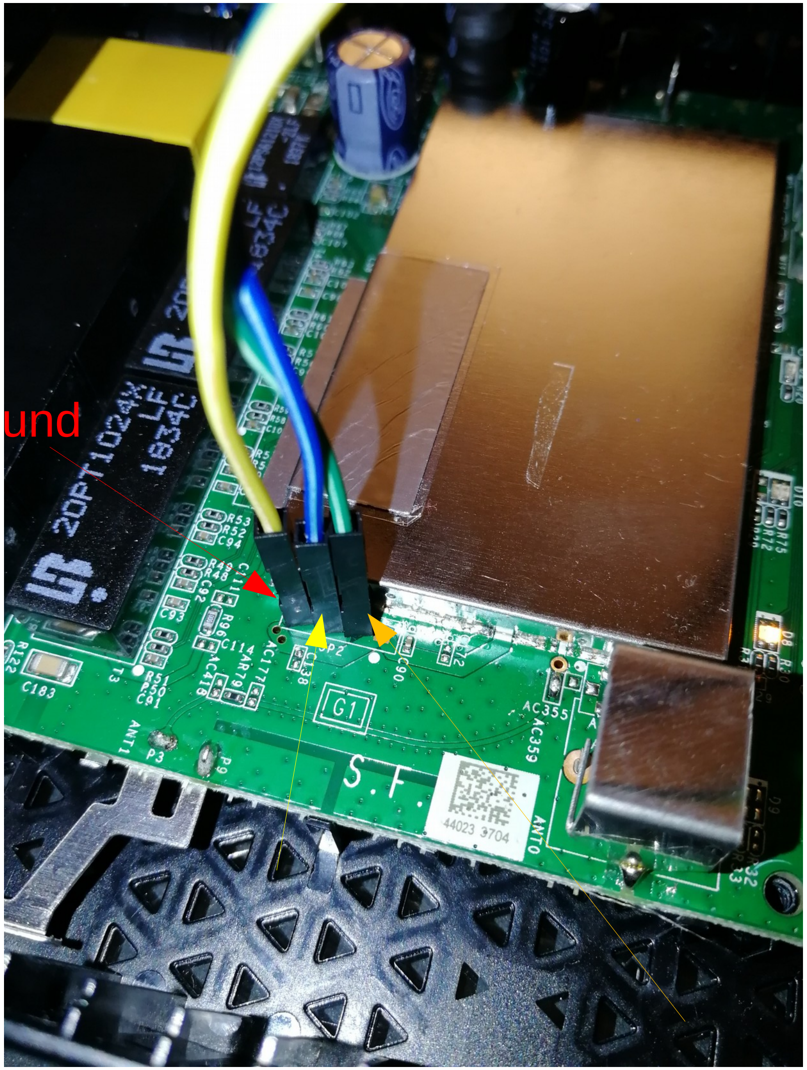
Need an Adapter :

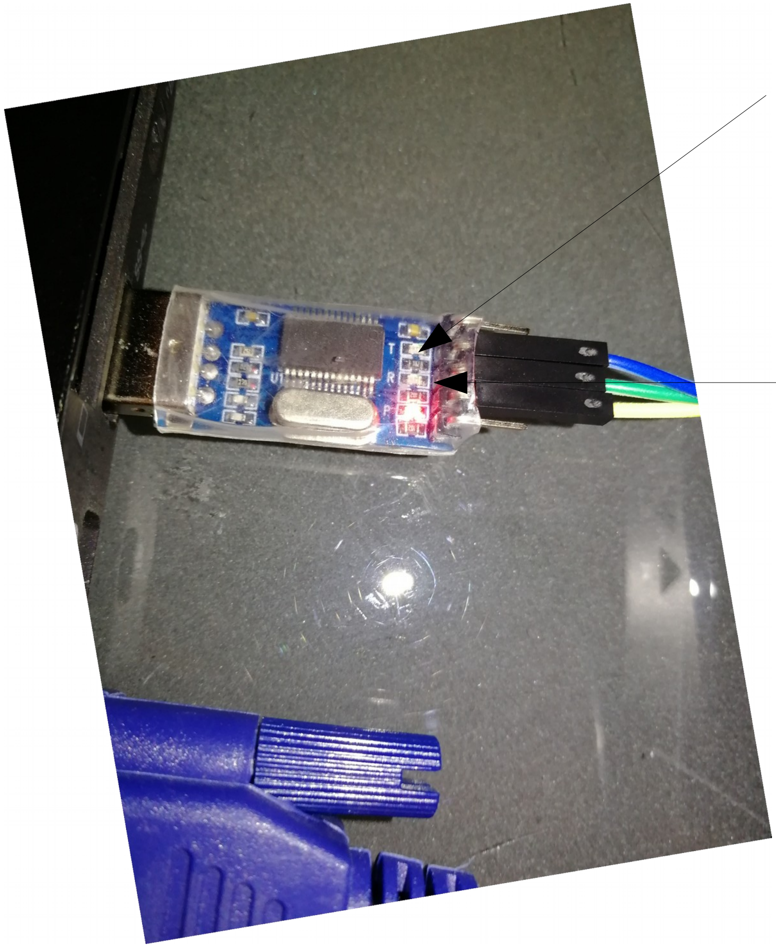


RX into TXD and TX into RXD

connecting

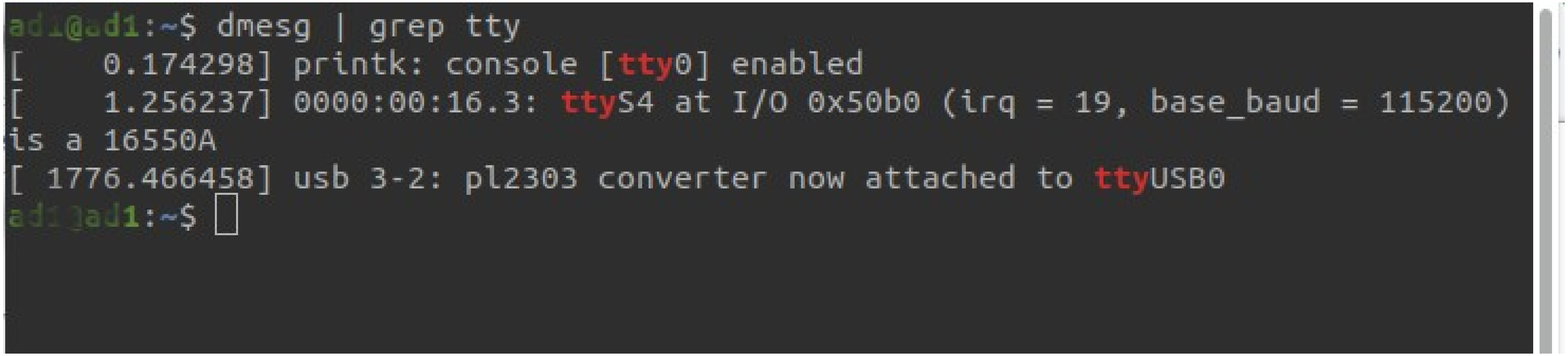
ground





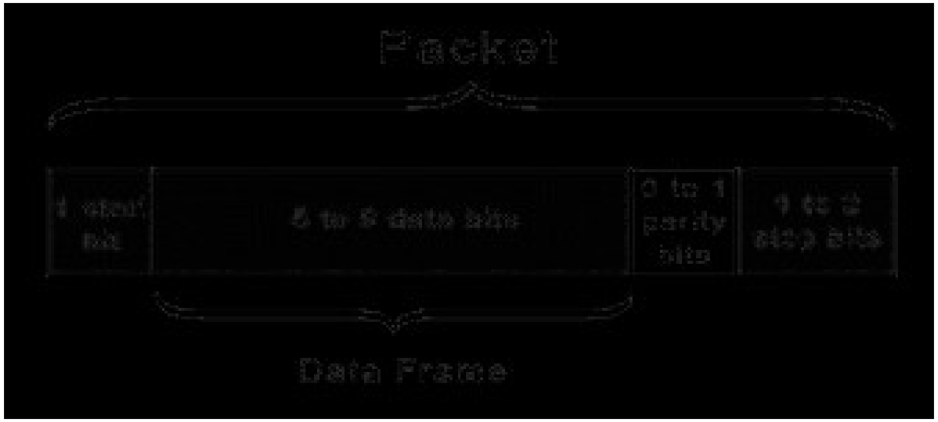
Is serial well connected ?

- Simple run dmesg command
- $ dmesg | grep tty
- Ls -l */dev/*tty*



```
ad1@ad1:~$ dmesg | grep tty
[    0.174298] printk: console [tty0] enabled
[    1.256237] 0000:00:16.3: ttyS4 at I/O 0x50b0 (irq = 19, base_baud = 115200)
is a 16550A
[ 1776.466458] usb 3-2: pl2303 converter now attached to ttyUSB0
ad1@ad1:~$ 
```

UART



http://www.circuitbasics.com/basics-uart-communication/

- Universal Asynchronous Receiver Transmiter
- Transfert data over the data bus
- For minicom configuration :
  - Bits of data
  - Parity bits
  - Stop bits
  - Baudrate

Transmission parameters

- Baudrates :
  - 38 400 baud
  - 57 600 baud
  - 115 200 baud

  Tranmission parameters are set over :
  - minicom [option]

- How fast the data is send over serial
- Test for most common
- Python script for this also :

  https://github.com/devttys0/baudrate

Victim1

- sudo minicom -b 115200 -D /dev/ttyUSB0
- Booting up, initialize
- Press Enter

troubleshooting :

- Nothing on the console ? Is the wiring ok ?
- Nothing happen when press Enter ?
  - Check Minicom options (Control+A and O)

root@WNR2000v5:/#

Explore : what is there ?

- pwd
- cd
- ls -l
- mount
- ps

- cat */proc/*cmdline
  - Where is rootfs ?
- Cat *proc/*version

Take a look at mtdblocks :

- Cat */proc/partitions*
  - *Ls /dev/mtdblock**

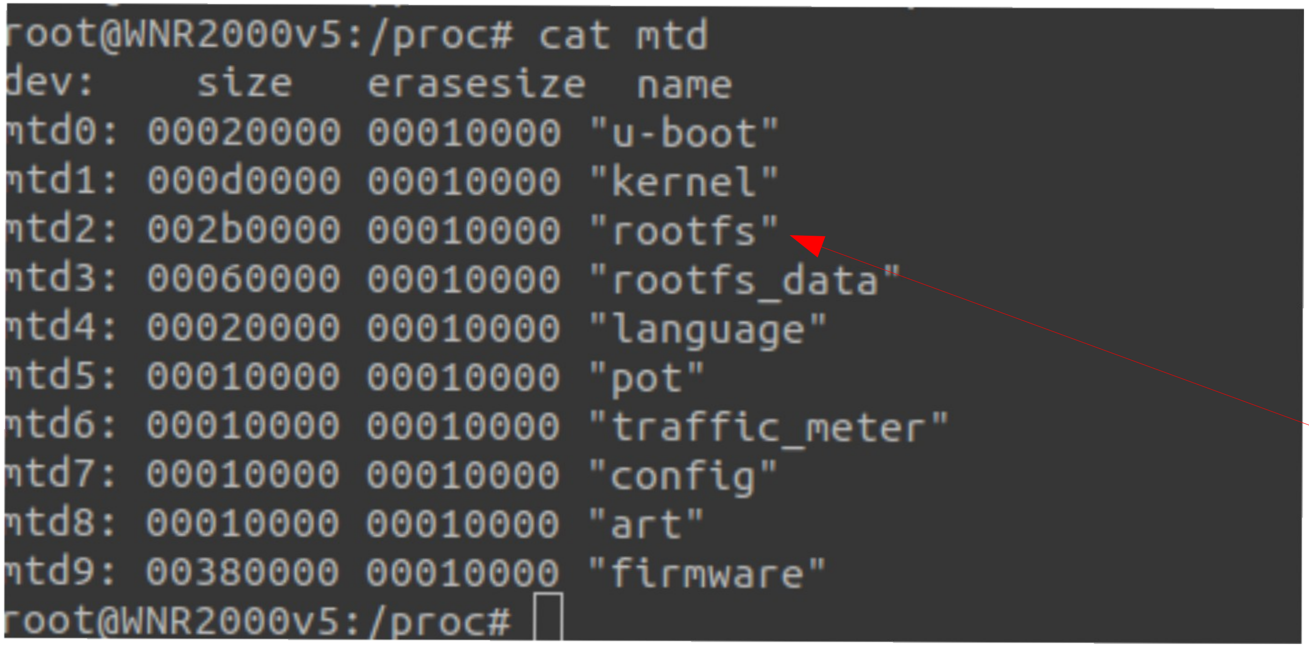
Flash memory

Mtdblock : Memory Technology Device subsystem for Linux
« emulate » block devices over MTD

Each block is « mounted »
/dev/mtdblock0

Searching for mtdblocks

- What are the names of mtdblocks we found ?
  - Cat */proc/*mtd
- What mtdblock do we want ?
- Remember where to find it ?



```
root@WNR2000v5:/proc# cat mtd
dev:    size   erasesize  name
mtd0: 00020000 00010000 "u-boot"
mtd1: 000d0000 00010000 "kernel"
mtd2: 002b0000 00010000 "rootfs"
mtd3: 00060000 00010000 "rootfs_data"
mtd4: 00020000 00010000 "language"
mtd5: 00010000 00010000 "pot"
mtd6: 00010000 00010000 "traffic_meter"
mtd7: 00010000 00010000 "config"
mtd8: 00010000 00010000 "art"
mtd9: 00380000 00010000 "firmware"
root@WNR2000v5:/proc# 
```

grabbing mtdblocks

How to extract mtdblocks?

How do extract

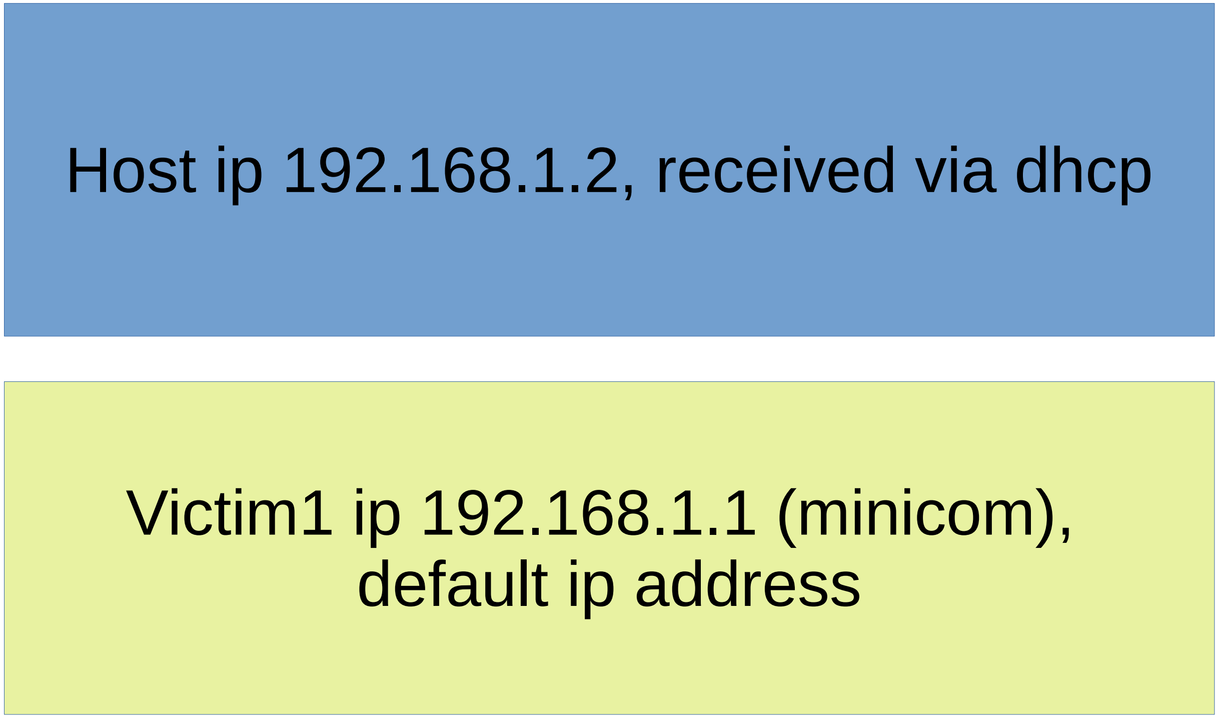
- Via ~~USB~~
- Via the Network (wifi or Ethernet)

- Searching for binaries to run on the router : anything useful ?
- dd, nc are all I need
- No nc or netcat binary !

Well...

An old schooler

- TFTP
- Send to Victim1 a netcat binary



Host ip 192.168.1.2, received via dhcp

Victim1 ip 192.168.1.1 (minicom), default ip address

On host

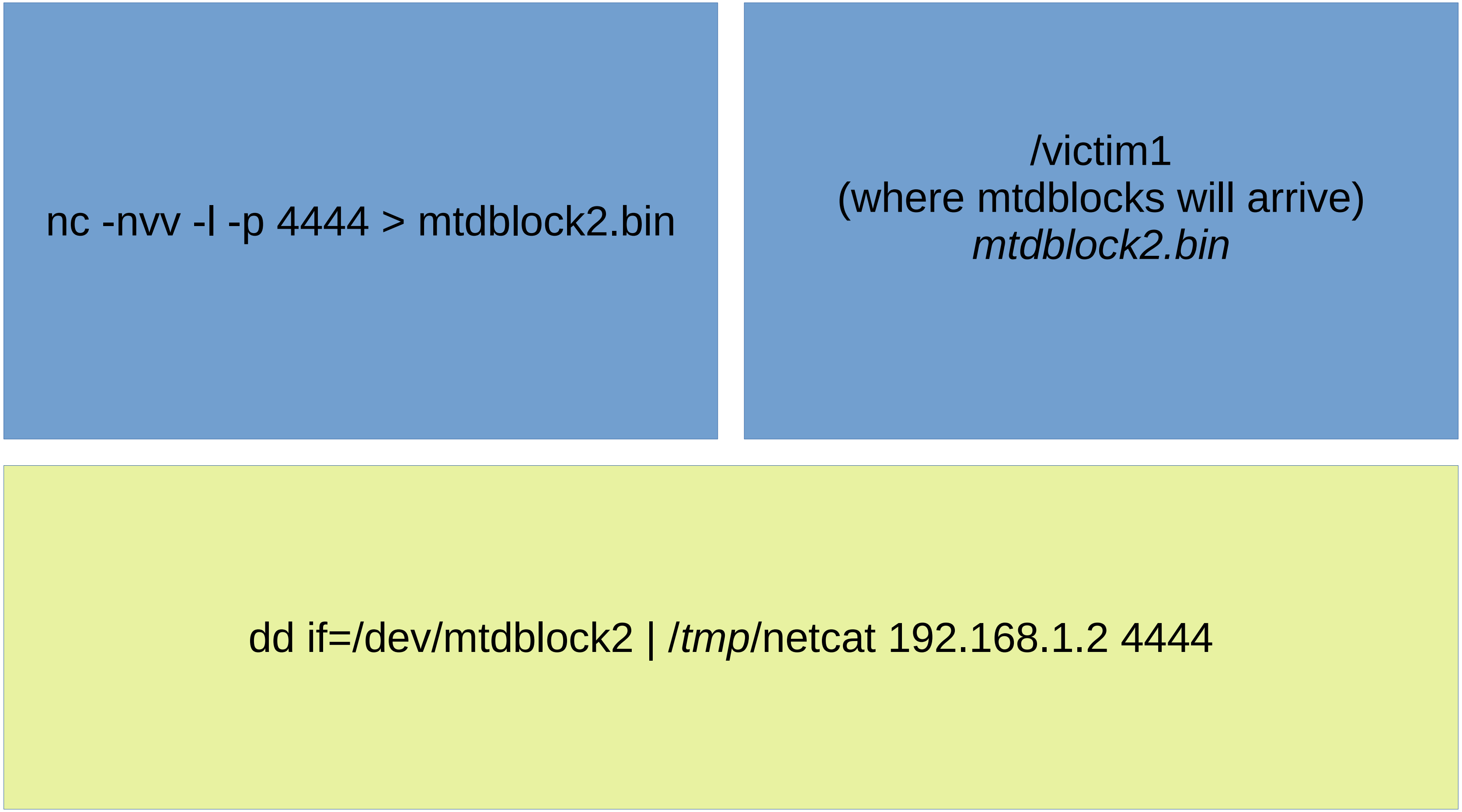
- On the target directory you want, copy the binaries you’ll need :
  - Statically linked netcat binary (MIPS)
  - a TFTP Server (x86 statically linked binary also)

- Chmod +x tftpserver
- Run the server on port 6969
  - sudo ./tftpserver . 6969

On target

- Connect to the target
- Go to /tmp directory
- Get the netcat binary
  - Tftp -g -r netcat 192.168.1.2:6969
  - Ls -la
  - Is there ?
    - Yes, chmod +x netcat

Transferring mtdblocks over UART

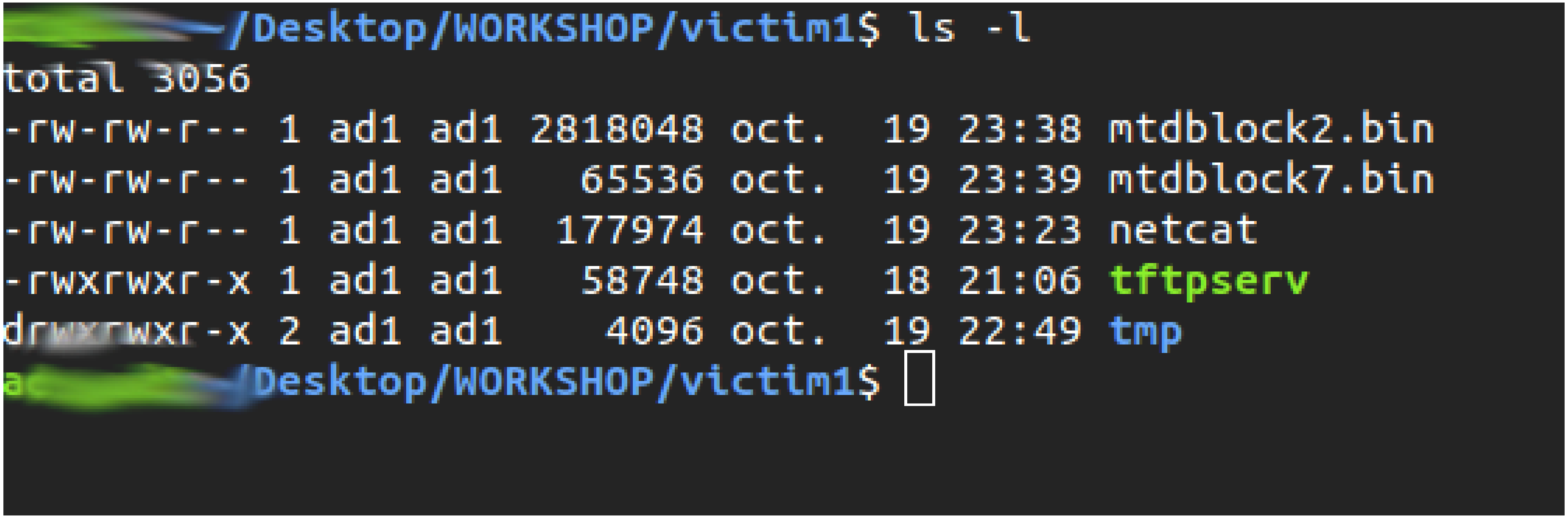


```
nc -nvv -l -p 4444 > mtdblock2.bin
```

/victim1
(where mtdblocks will arrive)
mtdblock2.bin

```
dd if=/dev/mtdblock2 | /tmp/netcat 192.168.1.2 4444
```

Did it work ?



```
[redacted]/Desktop/WORKSHOP/victim1$ ls -l
total 3056
-rw-rw-r-- 1 ad1 ad1 2818048 oct.  19 23:38 mtdblock2.bin
-rw-rw-r-- 1 ad1 ad1   65536 oct.  19 23:39 mtdblock7.bin
-rw-rw-r-- 1 ad1 ad1  177974 oct.  19 23:23 netcat
-rwxrwxr-x 1 ad1 ad1   58748 oct.  18 21:06 tftpserv
drwxrwxr-x 2 ad1 ad1    4096 oct.  19 22:49 tmp
[redacted]/Desktop/WORKSHOP/victim1$
```

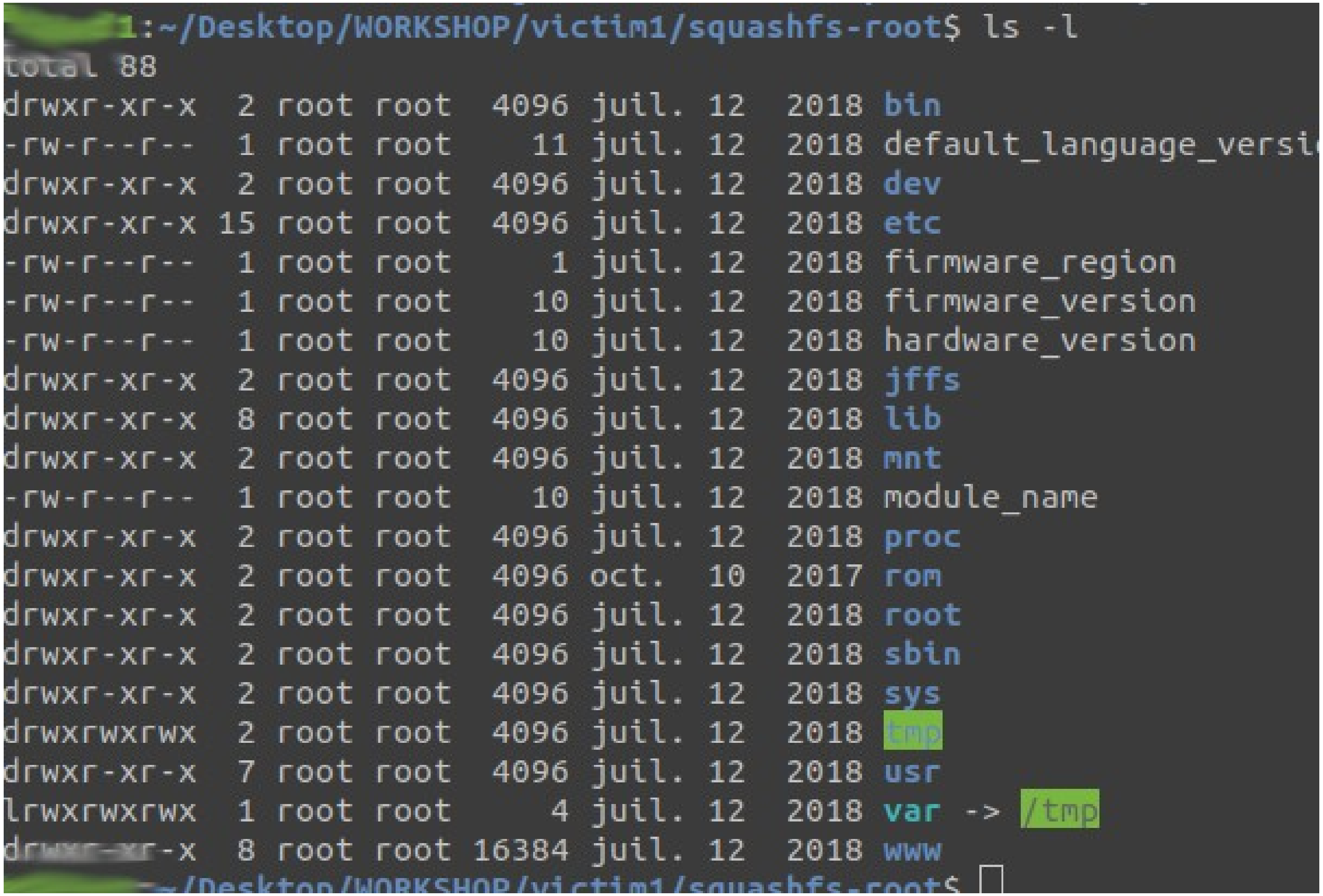
Now analyse

Uncompress the filesystem

- File mtdblock2.bin
- Strings mtdblock7.bin
- Root unsquashfs mtdblock2.bin
  - Quick install of unsquashfs-tools with apt

- Ls
  - New folder : /squashfs-root !

And « voila ! »



```
[redacted]:~/Desktop/WORKSHOP/victim1/squashfs-root$ ls -l
total 88
drwxr-xr-x  2 root root  4096 juil. 12  2018 bin
-rw-r--r--  1 root root    11 juil. 12  2018 default_language_versi
drwxr-xr-x  2 root root  4096 juil. 12  2018 dev
drwxr-xr-x 15 root root  4096 juil. 12  2018 etc
-rw-r--r--  1 root root     1 juil. 12  2018 firmware_region
-rw-r--r--  1 root root    10 juil. 12  2018 firmware_version
-rw-r--r--  1 root root    10 juil. 12  2018 hardware_version
drwxr-xr-x  2 root root  4096 juil. 12  2018 jffs
drwxr-xr-x  8 root root  4096 juil. 12  2018 lib
drwxr-xr-x  2 root root  4096 juil. 12  2018 mnt
-rw-r--r--  1 root root    10 juil. 12  2018 module_name
drwxr-xr-x  2 root root  4096 juil. 12  2018 proc
drwxr-xr-x  2 root root  4096 oct.  10  2017 rom
drwxr-xr-x  2 root root  4096 juil. 12  2018 root
drwxr-xr-x  2 root root  4096 juil. 12  2018 sbin
drwxr-xr-x  2 root root  4096 juil. 12  2018 sys
drwxrwxrwx  2 root root  4096 juil. 12  2018 tmp
drwxr-xr-x  7 root root  4096 juil. 12  2018 usr
lrwxrwxrwx  1 root root     4 juil. 12  2018 var -> /tmp
[redacted]-x  8 root root 16384 juil. 12  2018 www
[redacted]~/Desktop/WORKSHOP/victim1/squashfs-root$
```

- Questions
- Try on the mango routers now!

<u>Further developpments</u> : Try Cutecom and other GUI for programs like minicom, explore minicom options, explore memory mapping, firmware emulation...

Thanks

Be curious

Break things !

Thanks to my friend @therealsaumil

@ko97551819

Thank you !