Hacking IOT devices using UART

HiTB 2024 – Hardware Village

Security
Research
Labs

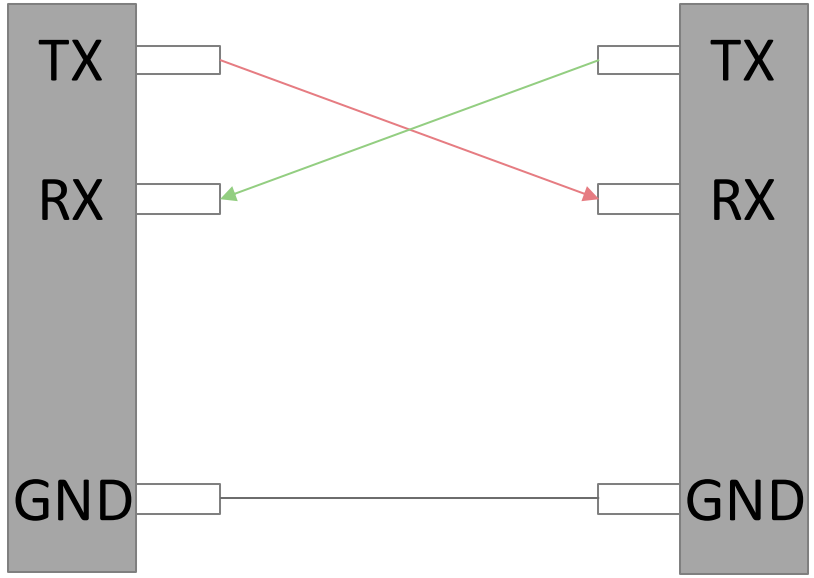
Agenda

1. **What is UART?**
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. Interactive root console

What is UART?

- A **universal asynchronous receiver-transmitter (UART)** is an asynchronous system, that rely on pre-configured baud rates to synchronize the data transmission, with the data being sent one bit at a time through a pair of wires: a transmit (TX) line and a receive (RX) line.

- UART is commonly used for **short-range serial communication** between microcontrollers, embedded systems, and various peripheral devices, such as displays, keypads, computers, wireless modules, and industrial equipment.



Objective of gaining UART access

UART is one of the most common interfaces available in many devices.
By accessing UART, we target to achieve the followings:

Interactive console

```
[   37.284000] set port 22 on
[   37.624000] fuse init (API version 7.12)
[   37.804000] usbcore: registered new interface driver usbfs
[   37.812000] usbcore: registered new interface driver hub
[   37.824000] usbcore: registered new device driver usb
[   37.848000] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[   37.856000] Port Status 1c000004
[   37.860000] ath-ehci ath-ehci.0: ATH EHCI
[   37.864000] ath-ehci ath-ehci.0: new USB bus registered, assigned bus number 1
[   37.872000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.880000] ehci_reset Port Status 1c000000
[   37.884000] ath-ehci ath-ehci.0: irq 3, io mem 0x1b000000
[   37.892000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.896000] ehci_reset Port Status 1c000000
[   37.916000] ath-ehci ath-ehci.0: USB 2.0 started, EHCI 1.00
[   37.920000] usb usb1: configuration #1 chosen from 1 choice
[   37.924000] hub 1-0:1.0: USB hub found
[   37.928000] hub 1-0:1.0: 1 port detected
[   37.936000] Port Status 1c000000
[   37.936000] ath-ehci1 ath-ehci1.1: ATH EHCI
[   37.940000] ath-ehci1 ath-ehci1.1: new USB bus registered, assigned bus number 2
[   37.952000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.956000] ehci_reset Port Status 1c000000
[   37.960000] ath-ehci1 ath-ehci1.1: irq 3, io mem 0x1b400000
[   37.968000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.976000] ehci_reset Port Status 1c000000
```

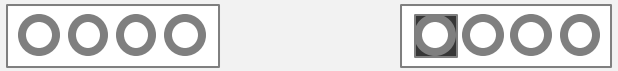
Dumping firmware

```
000000f0: 0000 00000000 00000000 00000000    ................
ap135> md -
00000000: 20a03ccd 545fc332 aba03ccd 545fc332     .<.T_.2..<.T_.2
00000010: aba03ccd 545fc332 aba03ccd 545fc332    ..<.T_.2..<.T_.2
00000020: 001ad042 335a0e78 037ad821 8f7a0000    .Z.x.z.!.z..
00000030: 8f7b0004 001a9182 409a1000 001bd902    .{......@.......
00000040: accd 545fc332 aba03ccd 545fc332    ..<.T_.2..<.T_.2
00000050: aba03ccd 545fc332 aba03ccd 542    ..<.T_.2..<.T_.2
00000060: 00000000 00000000 00000000 00000000    ................
00000070: 00000000 0000000000000 00000000    ................
00000080: aba03ccd 545fc332 aba03ccd 545fc332    ..<..<.T_.2
00000090: aba03ccd 545fc332 aba03ccd 545fc332    ..<.T_.2..<.T_.2
000000a0: 00000000 00000000 00000000 000    ................
000000b0: 00000000 00000000 00000000 00000000    ................
000000c0: aba03ccd 5452 aba03ccd 545fc332    ..<.T_.2..<.T_.2
000000d0: aba03ccd 545fc332 aba03ccd 545fc332    ..2..<.T_.2
000000e0: 00000000 00000000 00000000 00000000    ................
000000f0: 00000000 00000000 0000000000000    ................
```

Agenda

1. What is UART?
2. **Identifying UART**
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. Interactive root console

Identifying UART – identify pin out [1/2]

Potential UART Pin out	
Tools required	▪ Crocodile clip wire (x4) ▪ Multimeter
Identify GND port	1. Ensure the device is powered **off** 2. Use "Continuity Test" Mode on Multimeter 3. Touch one probe of multimeter onto an arbitrary metal on the board, and test the GND port by touching the other end of multimeter 4. The GND pin can be identified if a continuous "beep" sound from multimeter is noted

Identifying UART – identify pin out [2/2]

Identify VCC port	1. Ensure the device is powered **on** 2. Use "DC Voltage Mode", with 20V Max on Multimeter 3. Touch black probe onto GND port, and test the VCC port by touching the red probe 4. The VCC port can be found if the multimeter measures a constant voltage (usually 3.3V or 5V)
Identify TX port	1. Ensure the device is powered **on** 2. Use "DC Voltage Mode", with 20V Max on Multimeter 3. Touch black probe onto GND port, and test the TX port by touching the red probe 4. The TX port can be found if the multimeter measures a fluctuate voltage (around 1.8V - 2.1V) and then returning to VCC voltage (due to debugging data sent over via TX port)
Identify RX port	1. Ensure the device is powered **on** 2. Use "DC Voltage Mode", with 20V Max on Multimeter 3. Touch black probe onto GND port, and test the TX port by touching the red probe 4. The TX port can be found if the multimeter measures a small fluctuate voltage

Identifying UART – verify baud rate by Logic Analyzer [1/4]

What is baud rate?

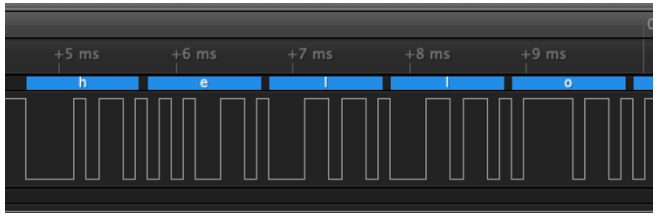
- Baud rate is the measure of the speed at which data is transmitted over a communication channel, typically expressed in bits per second (bps)
- It determines the number of signal or symbol changes that occur per second, allowing devices to effectively communicate with each other by synchronizing the rate at which data is sent and received.

Tools required

- Logic analyzer
- Machine with PulseView and corresponding driver installed
- Connecting wires

Setup

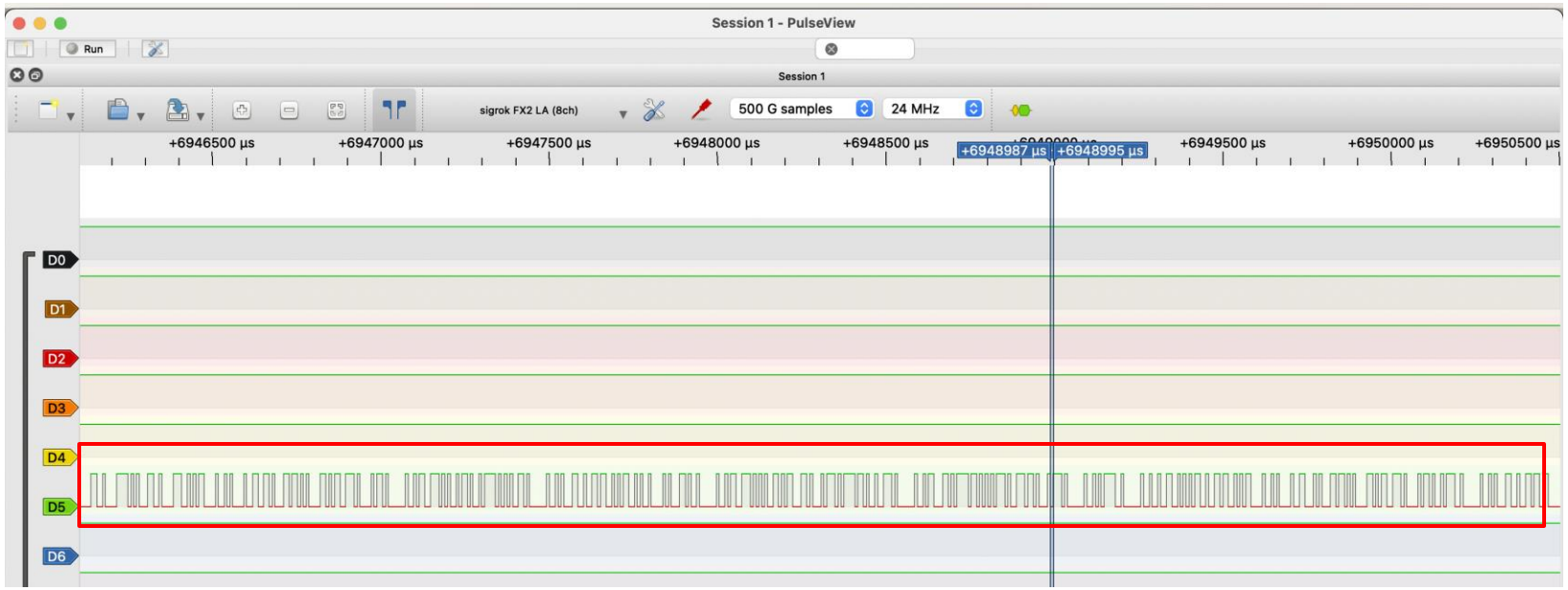
1. Connect TX of device to any channel on the logic analyzer
2. Connect GND of device to GND on Logic Analyzer
3. Install PulseView (and all dependencies) on your machine
4. Connect Logic Analyzer and machine

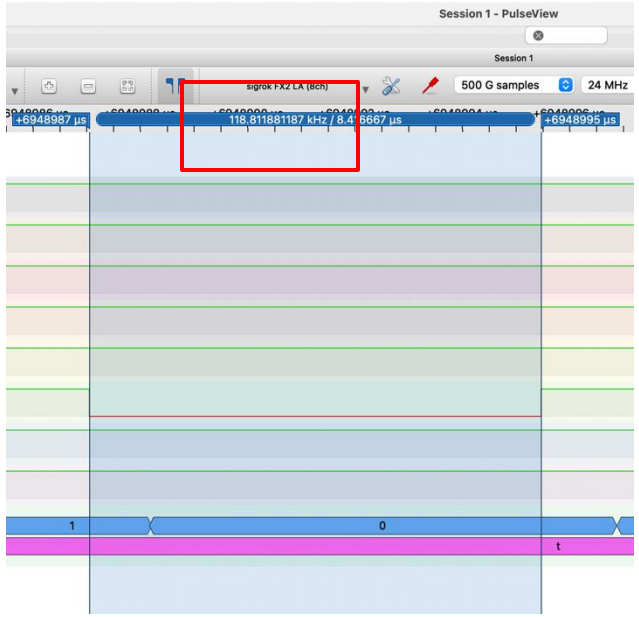


Identifying UART – verify baud rate by Logic Analyzer [2/4]

Procedures

1. Open up PulseView, and ensure the logic analyzer is connected and selected
2. Choose 500M samples and 24MHz^ initially
3. Click run, and turn on the device
4. There should be a signal appearing on your selected channel
5. Zoom into the smallest trough, and measure the time interval of the trough. This is used for rough estimation of signal frequency
6. Approximate value to the closest common Baud rate (e.g. 115200 in our demo)



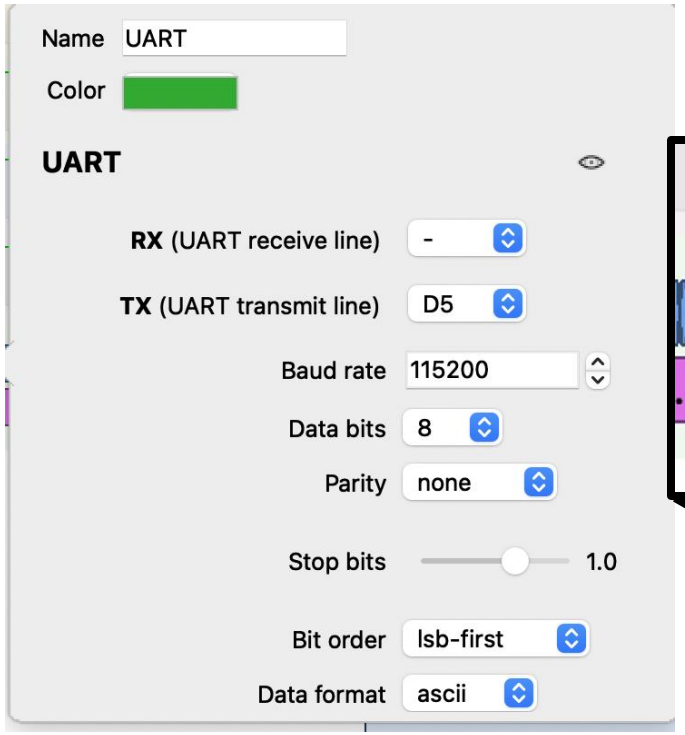


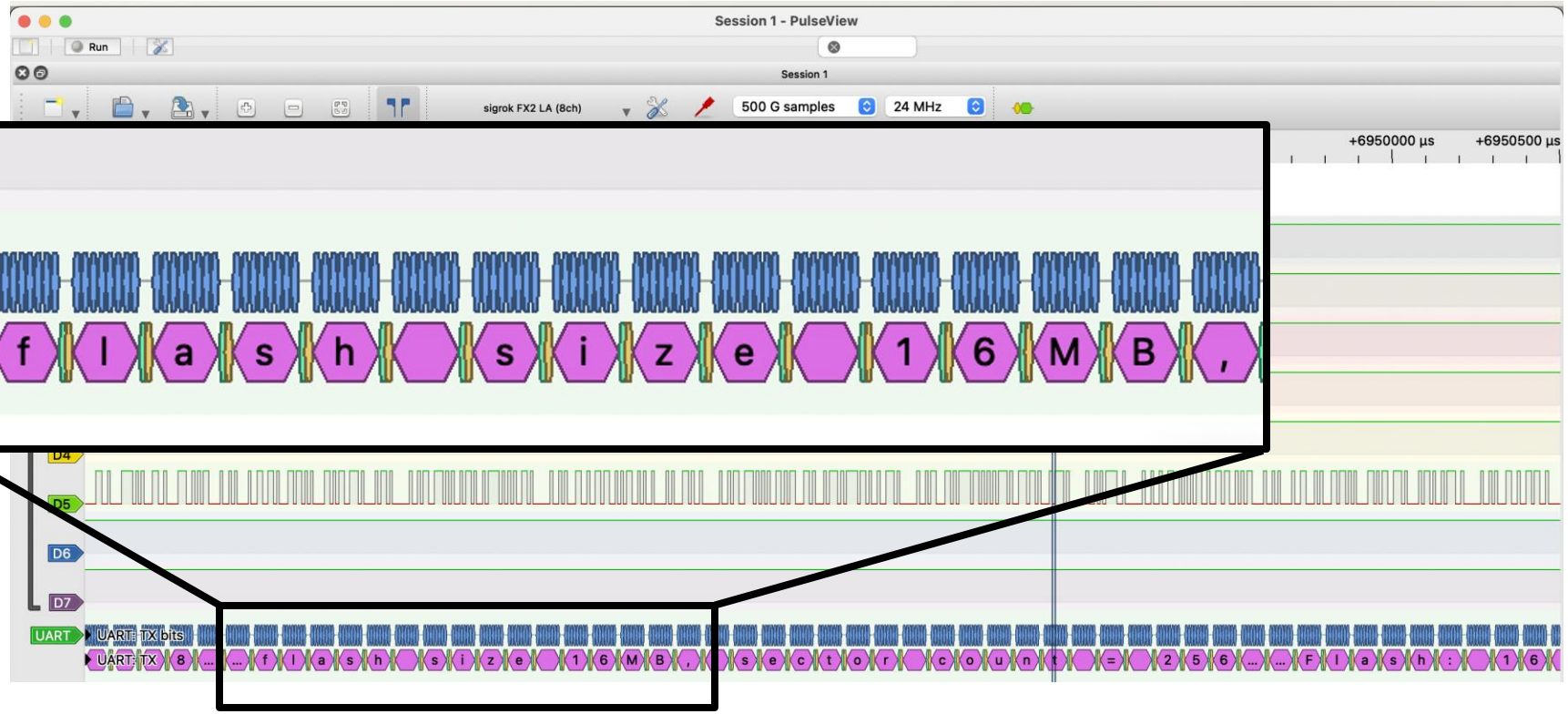
^ Be reminded to check the maximum supported frequency on your logic analyzer and set to highest available

Identifying UART – verify baud rate by Logic Analyzer [3/4]

Procedures

7. To read the decoded content from TX of device, select “add protocol header”, and choose UART
8. Select the channel as TX, with Baud rate set to the approximated value, and Data format to ASCII
9. Now you are able to view the decoded content from TX
10. If the content displayed in garbled, try with different Data bit, Parity and Stop bit

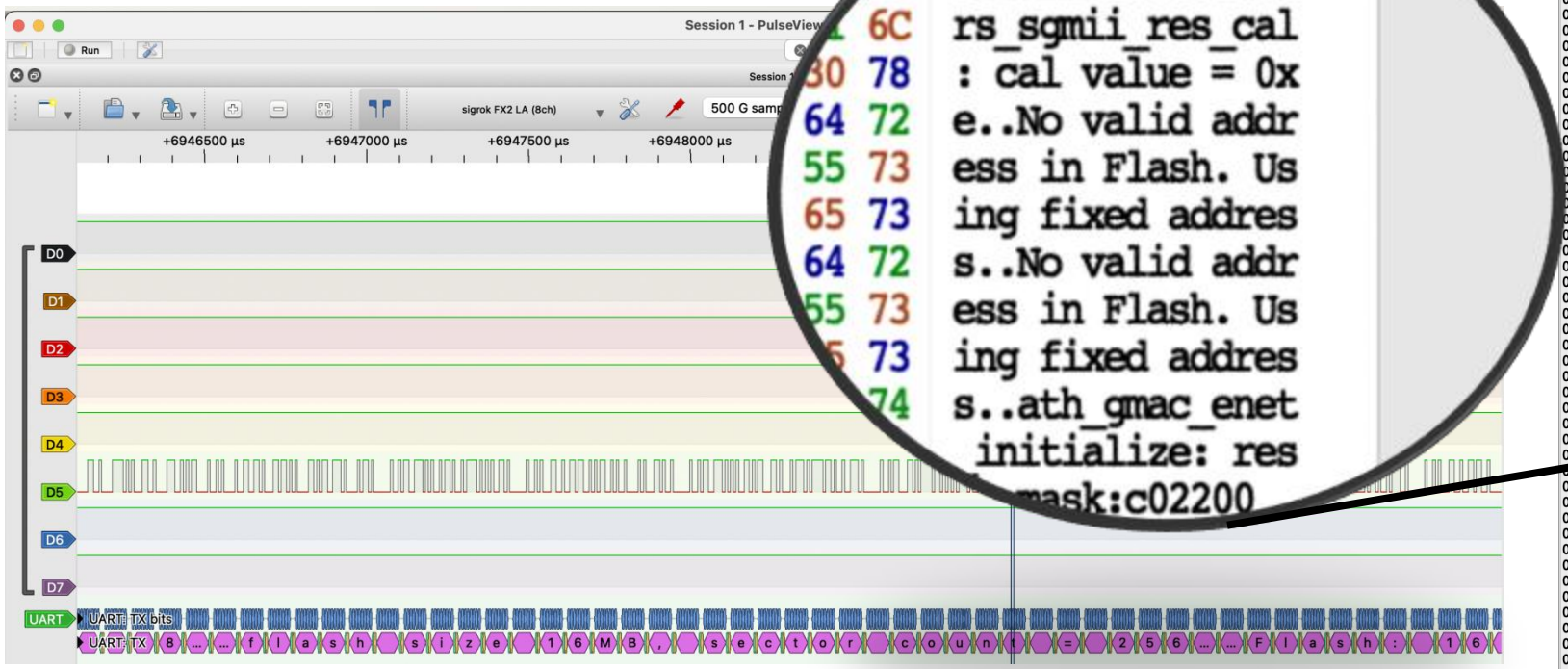


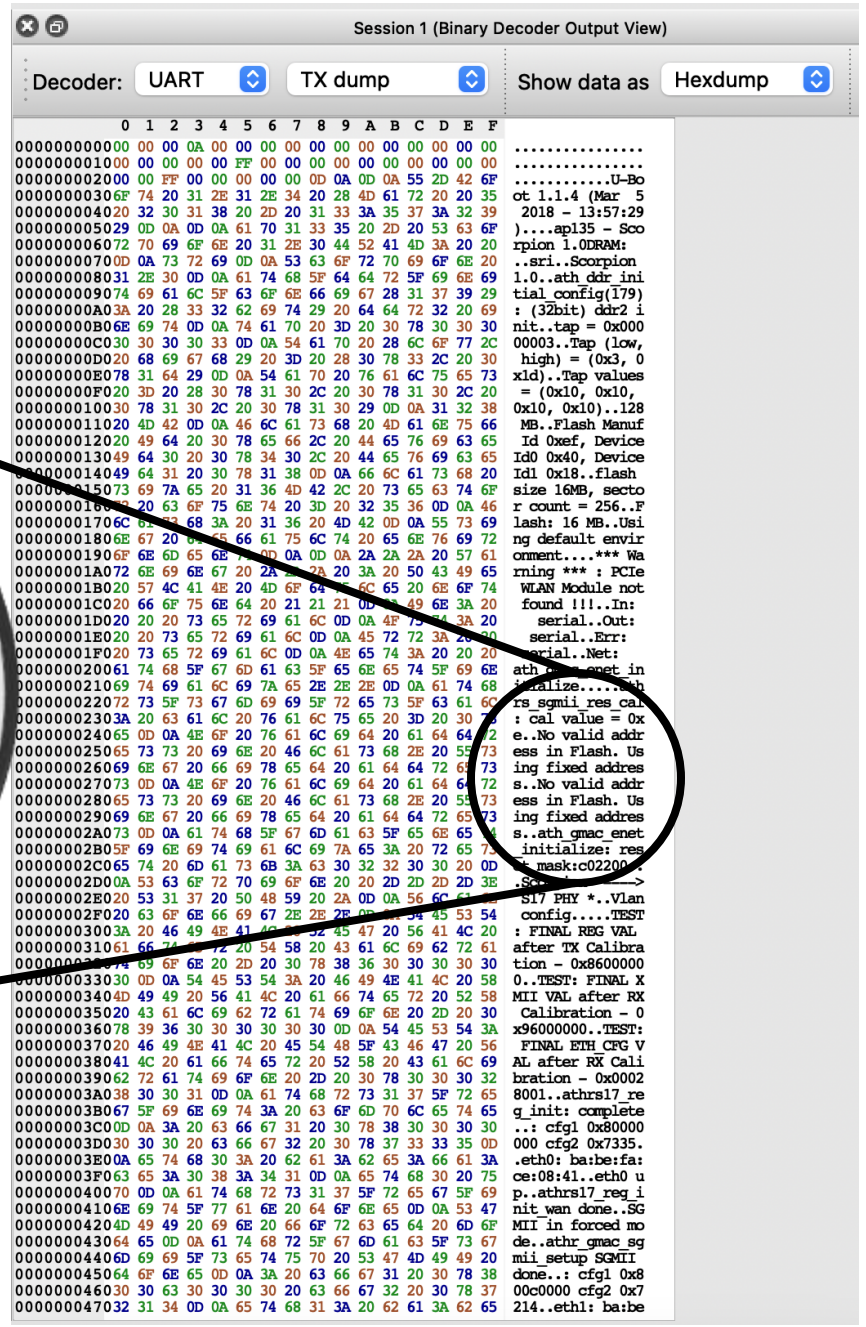


Identifying UART – verify baud rate by Logic Analyzer [4/4]

Procedures

11. Alternatively, you can also leverage “binary decoder output view” and view the Hexdump content





Agenda

1. What is UART?
2. Identifying UART
3. **Connecting to UART**
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. Interactive root console

Connecting to UART

FTDI

Bus Pirate

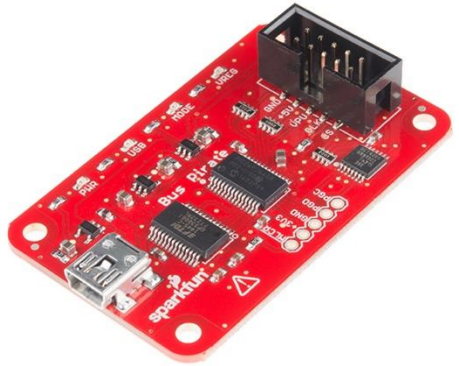
Raspberry Pi

Flipper Zero

Connecting to UART via Bus Pirate

Setup

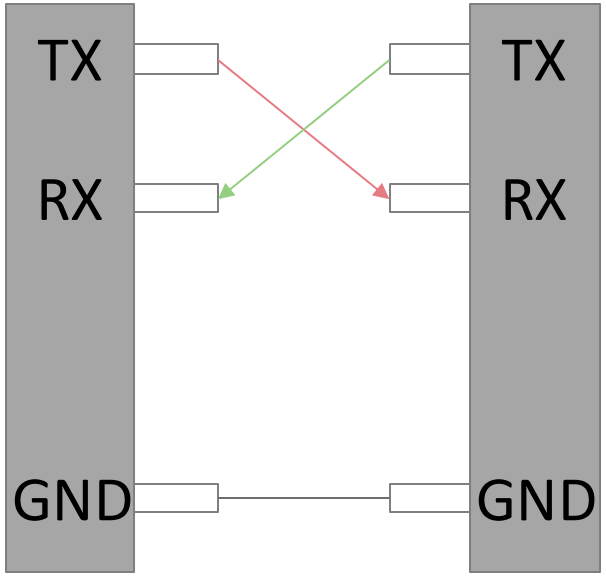
1. Connect TX of device to MISO (Master In Slave Out) of Bus Pirate
2. Connect RX of device to MOSI of Bus Pirate
3. Connect GND of device to GND of Bus Pirate
4. Connect Bus Pirate to your machine



Bus Pirate

Procedures

1. On your machine, open console, and list out candidates of USB ports:
   Linux: `ls /dev/usb*`
   MacOS: `ls /dev/tty.usb*`
2. To open serial connection to bus pirate, input either of the following commands:
   - `screen <bus-pirate-serial-port> 115200`
   - `picocom -b 115200 <bus-pirate-serial-port>`
3. Input m, then select 3 (UART), and configure with the baud rate, data bit, parity bit, stop bit you found in previous session
4. Select 2 for the power
5. Input (1)Transparent UART bridge and interact with UART on the device



You got the shell!

```
[   37.284000] set port 22 on
[   37.624000] fuse init (API version 7.12)
[   37.804000] usbcore: registered new interface driver usbfs
[   37.812000] usbcore: registered new interface driver hub
[   37.824000] usbcore: registered new device driver usb
[   37.848000] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[   37.856000] Port Status 1c000004
[   37.860000] ath-ehci ath-ehci.0: ATH EHCI
[   37.864000] ath-ehci ath-ehci.0: new USB bus registered, assigned bus number 1
[   37.872000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.880000] ehci_reset Port Status 1c000000
[   37.884000] ath-ehci ath-ehci.0: irq 3, io mem 0x1b000000
[   37.892000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.896000] ehci_reset Port Status 1c000000
[   37.916000] ath-ehci ath-ehci.0: USB 2.0 started, EHCI 1.00
[   37.920000] usb usb1: configuration #1 chosen from 1 choice
[   37.924000] hub 1-0:1.0: USB hub found
[   37.928000] hub 1-0:1.0: 1 port detected
[   37.936000] Port Status 1c000000
[   37.936000] ath-ehci1 ath-ehci1.1: ATH EHCI
[   37.940000] ath-ehci1 ath-ehci1.1: new USB bus registered, assigned bus number 2
[   37.952000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.956000] ehci_reset Port Status 1c000000
[   37.960000] ath-ehci1 ath-ehci1.1: irq 3, io mem 0x1b400000
[   37.968000] ehci_reset Intialize USB CONTROLLER in host mode: 13
[   37.976000] ehci_reset Port Status 1c000000
[   37.992000] ath-ehci1 ath-ehci1.1: USB 2.0 started, EHCI 1.00
[   37.996000] usb usb2: configuration #1 chosen from 1 choice
[   38.004000] hub 2-0:1.0: USB hub found
[   38.004000] hub 2-0:1.0: 1 port detected
[   38.204000] SCSI subsystem initialized
[   38.468000] Initializing USB Mass Storage driver...
[   38.472000] usbcore: registered new interface driver usb-storage
[   38.476000] USB Mass Storage support registered.
[   38.556000] kcg 333 :GPL NetUSB up!
[   38.772000] kc   90 : run_telnetDBGDServer start
[   38.776000] kc  227 : init_DebugD end
[   38.780000] INFO17E0: NetUSB 1.02.65.5, 0002061F : Apr 21 2015 15:30:36
[   38.784000] INFO17E2: 7437: Archer C7 v2 :Archer C7 v2
[   38.792000] INFO17E3:  AUTH ISOC
[   38.796000] INFO17E4:  filterAudio
[   38.796000] usbcore: registered new interface driver KC NetUSB General Driver
```

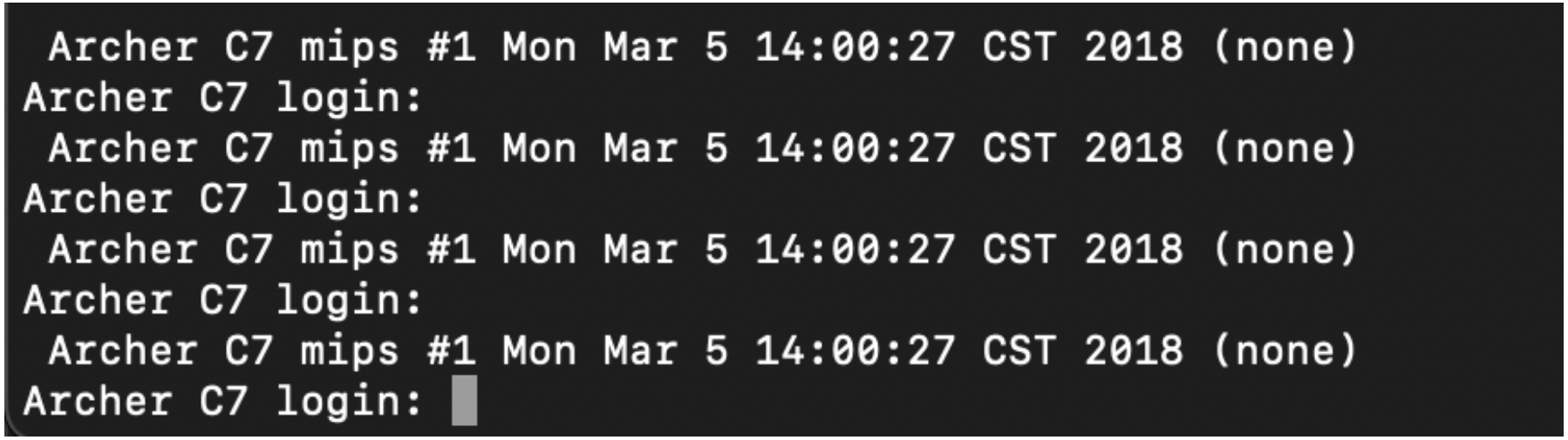
Agenda

1. What is UART?
2. Identifying UART
3. Connecting to UART
4. **Interactive console via Bus Pirate**
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. Interactive root console

Interactive console via Bus Pirate

Procedures

1. Launch the transparent UART bridge mode on Bus pirate, allow the boot flow to complete
2. You may now find yourself on the login console after the boot is completed, but how can one get the credentials to authenticate?
3. Authenticate with the obtained credentials
   - default password (root:root, admin:admin, etc)
   - web UI admin password
   - decrypted password from /etc/shadow from firmware analysis



Agenda

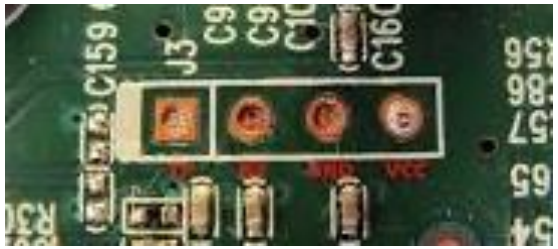
1. What is UART?
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. **Methods to dump firmware**
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. Interactive root console

Methods to dump firmware

JTAG

SPI flash

UART (U-boot)



Check our other workbenches for walkthrough on jTAG and SPI flash!

Online repository from vendor

TL-WR902AC(US)_V3.6_0.9.1 Build 220329		Download
Published Date: 2022-10-26	Language: Multi-language	File Size: 7.96 MB

Modifications and Bug Fixes:
Optimized total performance.

Agenda

1. What is UART?
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. **Extract hash and crack password from official firmware repository**
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. Interactive root console

Extract hash and crack password from official firmware repository [1/2]

Procedures

1. Download the firmware from vendor's website
2. Extract the filesystem from the firmware binary with binwalk
   `binwalk -e <firmware_binary_path>`

```
srlabs@srlabs:~/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_180305$ binwalk -e ArcherC7v2_en_eu_3_15_3_up_boot\(180305
\).bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             TP-Link firmware header, firmware version: 1.-15188.3, image version: "", product ID: 0x0, product version: -956
301310, kernel load address: 0x0, kernel entry point: 0x80002000, kernel offset: 16384512, kernel length: 512, rootfs offset: 855873, rootfs l
ength: 1048576, bootloader offset: 15204352, bootloader length: 0
71520         0x11760         Certificate in DER format (x509 v3), header length: 4, sequence length: 64
98560         0x18100         U-Boot version string, "U-Boot 1.1.4 (Mar  5 2018 - 13:57:29)"
98736         0x181B0         CRC32 polynomial table, big endian
131584        0x20200         TP-Link firmware header, firmware version: 0.0.3, image version: "", product ID: 0x0, product version: -95630131
0, kernel load address: 0x0, kernel entry point: 0x80002000, kernel offset: 16252928, kernel length: 512, rootfs offset: 855873, rootfs length
: 1048576, bootloader offset: 15204352, bootloader length: 0
132096        0x20400         LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes, uncompressed size: 2451644 bytes

WARNING: Symlink points outside of the extraction directory: /home/srlabs/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_
180305/_ArcherC7v2_en_eu_3_15_3_up_boot(180305).bin.extracted/squashfs-root/etc/passwd -> /tmp/passwd; changing link target to /dev/null for s
ecurity purposes.

WARNING: Symlink points outside of the extraction directory: /home/srlabs/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_
180305/_ArcherC7v2_en_eu_3_15_3_up_boot(180305).bin.extracted/squashfs-root/etc/ppp/conn-script -> /tmp/conn-script; changing link target to /
dev/null for security purposes.

WARNING: Symlink points outside of the extraction directory: /home/srlabs/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_
180305/_ArcherC7v2_en_eu_3_15_3_up_boot(180305).bin.extracted/squashfs-root/bin/iptables-xml -> /lzm/qca_dual/rootfs.build.2.6.31_ap135v2_wifi
/sbin/iptables-multi; changing link target to /dev/null for security purposes.
1180160       0x120200        Squashfs filesystem, little endian, version 4.0, compression:lzma, size: 9878520 bytes, 789 inodes, blocksize: 1
31072 bytes, created: 2018-03-05 06:16:10

srlabs@srlabs:~/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_180305$
```

Extract hash and crack password from official firmware repository [2/2]

Procedures

3. Extract the hash from /etc/shadow
4. Search the hash from online, or crack it with your hashcat

```
srlabs@srlabs:~/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_180305$ ls -l _ArcherC7v2_en_eu_3_15_3_up_boot\(180305\).bin.extracted/squashfs-root/
total 52
drwxr-xr-x  2 srlabs srlabs 4096 Aug 27 12:44 bin
drwxr-xr-x  3 srlabs srlabs 4096 Mar  5  2018 dev
drwxr-xr-x  9 srlabs srlabs 4096 Aug 27 12:44 etc
drwxr-xr-x  5 srlabs srlabs 4096 Mar  5  2018 lib
lrwxrwxrwx  1 srlabs srlabs   11 Aug 27 12:44 linuxrc -> bin/busybox
drwxr-xr-x  2 srlabs srlabs 4096 Mar  5  2018 mnt
drwxr-xr-x  2 srlabs srlabs 4096 Mar  5  2018 proc
drwxr-xr-x  2 srlabs srlabs 4096 Mar  5  2018 root
drwxr-xr-x  2 srlabs srlabs 4096 Mar  5  2018 sbin
drwxr-xr-x  2 srlabs srlabs 4096 Mar  5  2018 sys
drwxr-xr-x  2 srlabs srlabs 4096 Mar  5  2018 tmp
drwxr-xr-x  4 srlabs srlabs 4096 Mar  5  2018 usr
drwxr-xr-x  3 srlabs srlabs 4096 Mar  5  2018 var
drwxr-xr-x 10 srlabs srlabs 4096 Mar  5  2018 web
srlabs@srlabs:~/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_180305$ cat _ArcherC7v2_en_eu_3_15_3_up_boot\(180305\).bin.extracted/squashfs-root/etc/shadow
root:$1$GTN.gpri$DlSyKvZKMR9A9Uj9e9wR3/:15502:0:99999:7:::
srlabs@srlabs:~/Workspace/device-testing/tp-link-ac1750v2/analysis/Archer C7(EU)_V2_180305$
```

Googling the hash[1]

I had to use --force to get it to work on my VM.
Attack type, bruteforce=3 : -a 3
Set the variable for the mask attack, we using just lower case letters: -1 ?l
md5 crypt 1 : -m 500
hash file: hash.txt
mask to use (know the password uses admin): ?1?1?1?1admin

1GTN.gpri$DlSyKvZKMR9A9Uj9e9wR3/:sohoadmin

[1] https://blog.xynos.co.uk/2020/03/hashcat-fun.html

Agenda

1. What is UART?
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. **U-boot and interactive terminal**
8. Exercise
9. Alternative boot
10. Interactive root console

What is U-Boot

Description	▪ Official name: Das U-Boot (the Universal Boot Loader) ▪ Architectures supported: M6800, ARM, Blackfin, MicroBlaze, IBM S360, My66, MOS 6502, ARM64, MIPS, Nios, SuperH, PPC, RISC-V, x86 ▪ GitHub repository: https://github.com/u-boot/u-boot
Functionality	▪ First-stage and second-stage bootloader ▪ Intended start-up flow for board ▪ Stores important configuration parameters (e.g. IP address of TFTP server)

Information retrievable	▪ Firmware, which could contain:
	  ▪ Passwords and Hashes
	  ▪ Usernames
	  ▪ Sensitive Public-private key pairs
	  ▪ IP addresses pre-configured for communication

U-boot via Bus Pirate

Procedures

- [For TP-Link devices] repetitively type “tpl” and enter in the beginning of boot -> dump firmware
- [For other devices] check for documentations on u-boot

Common commands available

- autoload
- autostart
- baudrate
- bootargs
- bootcmd
- bootdelay
- bootfile
- ethaddr
- ipaddr
- loadaddr
- serverip
- silent
- help
- printenv

```
version - print monitor ver
ap135> printenv
bootargs=console=ttyS0,115200 root=31:02 rootfstype=jffs2 init=/sbin/init mtdparts=ath-nor0:256k(u-boot),64k(u-b
bootcmd=bootf020000
bootdelay=1
baudrate=115200
ethaddr=0xba:0xbe:0xfa:0xce:0x08:0x41
ipaddr=192.168.1.111
serverip=192.1100
dir=
lu=tftp 0x80060000 ${dir}u-boot.bin&&erase 0x9f000000 +$filesize&&cp.b $fileaddr 0x9f000000 $filesize
tp 0x80060000 ${dir}ap135${bc}-jffs2&&erase 0x9f050000 +0x630000&&cp.b $fileaddr 0x9f050000 $filesize
lk=tftp 0x8000 ${dir}vmlinux${bc}.lzma.uImage&&erase 0x9f680000 +$filesize&&cp.b $fileaddr 0x9f680000 esize
stdin=serial
stdout=serial
stderr=serial
ethact=eth0
```

Agenda

1. What is UART?
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. **Exercise**
9. Alternative boot
10. Interactive root console

Exercise – Dump root credentials from U-boot

Hints

- Here are a few keywords that may spark your thoughts:
  - md (memory display)
  - cp (copy)
  - binwalk

```
000000f0: 0000 00000000 00000000 00000000    ................
ap135> md -
00000000: 20a03ccd 545fc332 aba03ccd 545fc332     .<.T_.2..<.T_.2
00000010: aba03ccd 545fc332 aba03ccd 545fc332    ..<.T_.2..<.T_.2
00000020: 001ad042 335a0e78 037ad821 8f7a0000    .Z.x.z.!.z..
00000030: 8f7b0004 001a9182 409a1000 001bd902    .{......@.......
00000040: accd 545fc332 aba03ccd 545fc332    ..<.T_.2..<.T_.2
00000050: aba03ccd 545fc332 aba03ccd 542    ..<.T_.2..<.T_.2
00000060: 00000000 00000000 00000000 00000000    ................
00000070: 00000000 0000000000000 00000000    ................
00000080: aba03ccd 545fc332 aba03ccd 545fc332    ..<..<.T_.2
00000090: aba03ccd 545fc332 aba03ccd 545fc332    ..<.T_.2..<.T_.2
000000a0: 00000000 00000000 00000000 000    ................
000000b0: 00000000 00000000 00000000 00000000    ................
000000c0: aba03ccd 5452 aba03ccd 545fc332    ..<.T_.2..<.T_.2
000000d0: aba03ccd 545fc332 aba03ccd 545fc332    ..2..<.T_.2
000000e0: 00000000 00000000 00000000 00000000    ................
000000f0: 00000000 00000000 0000000000000    ................
```

Agenda

1. What is UART?
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. **Alternative boot**
10. Interactive root console

Alternative boot if all these failed

USB live boot

```
[   37.928000] hub 1-0:1.0: 1 port detected
[   37.936000] Port Status 1c000000
[   37.936000] ath-ehci1 ath-ehci1.1: ATH EHCI
[   37.940000] ath-ehci1 ath-ehci1.1: new USB bus registered,
[   37.952000] ehci_reset Intialize USB CONTROLLER in host mod
[   37.956000] ehci_reset Port Status 1c000000
[   37.960000] ath-ehci1 ath-ehci1.1: irq 3, io mem 0x1b400000
[   37.968000] ehci_reset Intialize USB CONTROLLER in host mod
[   37.976000] ehci_reset Port Status 1c000000
[   37.992000] ath-ehci1 ath-ehci1.1: USB 2.0 started, EHCI 1.
[   37.996000] usb usb2: configuration #1 chosen from 1 choice
[   38.004000] hub 2-0:1.0: USB hub found
[   38.004000] hub 2-0:1.0: 1 port detected
[   38.204000] SCSI subsystem initialized
[   38.468000] Initializing USB Mass Storage driver...
[   38.472000] usbcore: registered new interface driver usb-st
[   38.476000] USB Mass Storage support registered.
[   38.556000] kcg 333 :GPL NetUSB up!
[   38.772000] kc   90 : run_telnetDBGDServer start
[   38.776000] kc  227 : init_DebugD end
[   38.780000] INFO17E0: NetUSB 1.02.65.5, 0002061F : Apr 21 2
[   38.784000] INFO17E2: 7437: Archer C7 v2 :Archer C7 v2
[   38.792000] INFO17E3:  AUTH ISOC
[   38.796000] INFO17E4:  filterAudio
```

TFTP boot

```
ap135>
ap135> help
?       - alias for 'help'
bootm   - boot application image from memory
cp      - memory copy
crc32   - checksum calculation
erase   - erase FLASH memory
flinfo  - print FLASH meinformation
go      - start application at address 'addr'
help    - print online help
    - memory modify (auto-incrementing)
mw      - memory write (fill)
ntenv- print environment variablesaddress)
progmac - Set ethernet MAC addresses
protect - enable sable FLASH write protection
reset   - Perform RESET of the CPU
setenv  - set environment variables
tftpboot- bimage via network using TFTP protocol
version - print monitor version
```

Agenda

1. What is UART?
2. Identifying UART
3. Connecting to UART
4. Interactive console via Bus Pirate
5. Methods to dump firmware
6. Extract hash and crack password from official firmware repository
7. U-boot and interactive terminal
8. Exercise
9. Alternative boot
10. **Interactive root console**

Interactive root console

Procedures	1. Authenticate with the obtained credentials
What's next?	▪ Thorough enumeration on filesystem, and look for sensitive information ▪ Inspect services installed (e.g. available services on busybox) ▪ Dynamic analysis (e.g. running processes, interaction with other devices, communication with external IPs)