PWNAGOTCHI: DEEP REINFORCEMENT LEARNING FOR WIFI PWNING!

BOUSSETA HATIM

GCSE

@ENSA-TETOUAN

What is pwnagotchi ?

Pwnagotchi is a small device that uses AI to learn from nearby WiFi networks. It runs on a Raspberry Pi Zero W and uses a tool called bettercap. It tries to collect as much WiFi password data as possible by listening to networks or forcing devices to reconnect. The data it collects is saved in files that can be used later with tools like hashcat to try to crack the passwords.

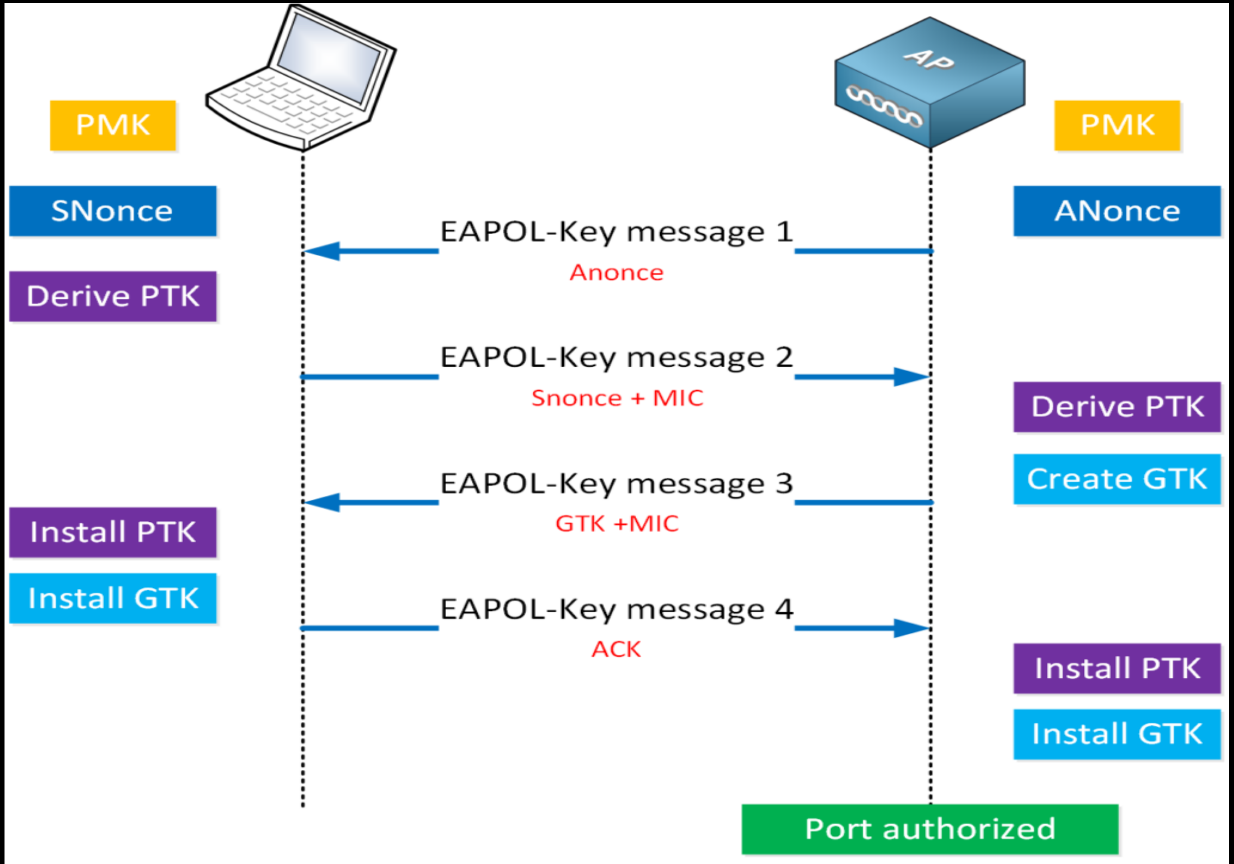
Pwnagotchi uses an AI model based on ^A2C (Advantage Actor-Critic)^, which is a type of ^reinforcement learning^.

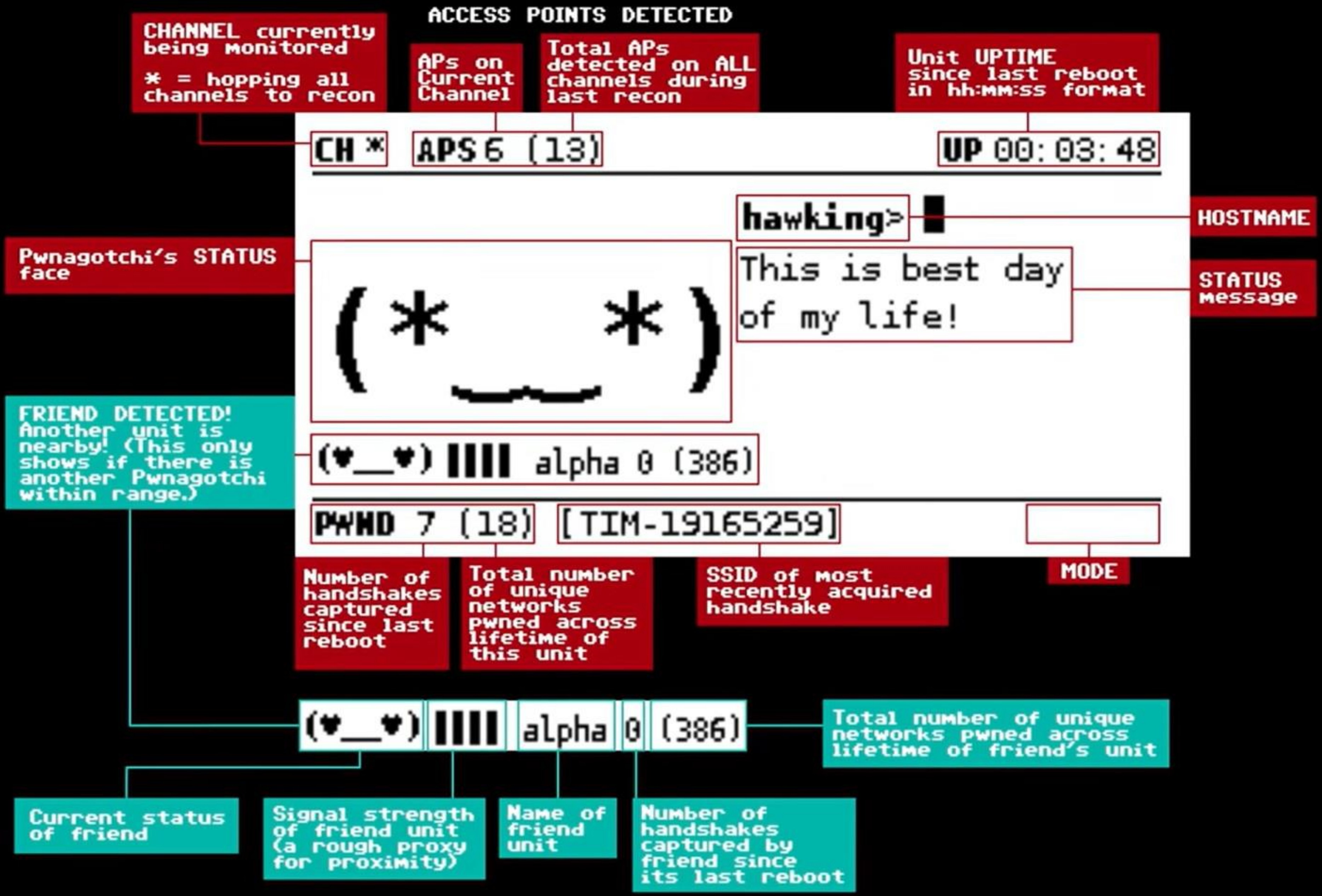
In simple words: it learns by trying different actions (like scanning or attacking networks), sees what works best to get more WiFi handshakes, and improves over time. A2C helps it choose smarter actions based on past success.

How does pwnagotchi work ?

- *When an AP and a device establish a WI-FI connection , they exchange special data packets , called a handshake in the WPA AND WPA2 Wireless protocol .*

- *How the handshake work?*





ACCESS POINTS DETECTED

CHANNEL currently being monitored

* = hopping all channels to recon

APs on Current Channel

Total APs detected on ALL channels during last recon

Unit UPTIME since last reboot in hh:mm:ss format

```
CH *   APS 6 (13)                              UP 00:03:48

                               hawking> █

                               This is best day
( *__* )                       of my life!

(♥__♥) ▌▌▌▌ alpha 0 (386)

PWND 7 (18)   [TIM-19165259]
```

HOSTNAME

Pwnagotchi's STATUS face

STATUS message

FRIEND DETECTED! Another unit is nearby! (This only shows if there is another Pwnagotchi within range.)

Number of handshakes captured since last reboot

Total number of unique networks pwned across lifetime of this unit

SSID of most recently acquired handshake

MODE

```
(♥__♥) ▌▌▌▌ alpha 0 (386)
```

Total number of unique networks pwned across lifetime of friend's unit

Current status of friend

Signal strength of friend unit (a rough proxy for proximity)

Name of friend unit

Number of handshakes captured by friend since its last reboot

- ***Personality and moods :***

(⇀‿‿↼) sleeping

This is the state the unit will start from. Moreover, from time to time your Pwnagotchi will also perform naps of a few seconds while hopping among WiFi channels

(≖‿‿≖) awakening

The unit is in the last seconds of its nap

(◕‿‿◕) awake / normal

This face is the neutral awake status of the unit. It’ll be used to smooth the transition between other moods and in general when there’s no external cause of either positive or negative moods. It can also be used, randomly, when another unit is encountered for the first time.

(☉⚆), (⚆☉) observing (neutral mood)

Your Pwnagotchi is waiting and observing what bettercap can find on all the channels it’s hopping on.

(◕‿◕), (◕‿◕) observing (happy)

When there’s one or multiple units nearby and their cumulative bond counter is greater or equal than the personality.bond_encounters_factor, this will be the unit’s face while observing.

(°▃▃°) intense

The unit is sending an association frame to an access point in order to force it to leak the PMKID.

⌐■_■) cool

The unit is deauthenticating a client station from an access point. This face can also be picked randomly when meeting another unit for the first time.

(•‿‿•) happy

Your Pwnagotchi is happy in one of the following cases:

-The AI just finished loading and it’s ready.

-Valid key material for an access point has just been captured.

-In MANU mode, if the last session was short or if any handshake has been captured during it.

-When another unit is met and the bond level is high enough.

(╥☁╥) sad

If there are no friendly units around and the amount of consecutive inactive epochs reached personality.sad_num_epochs.

(ب__ب) lonely

If your Pwnagotchi just lost contact with a friendly unit that was nearby, or if the amount of missed interactions with access points or client stations (the amount of times it tried to send some type of packet but missed the target because it isn’t in range anymore)

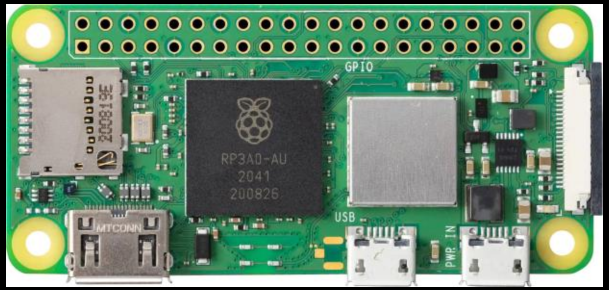
(☓‿‿☓) broken

Your unit is rebooting either as a coping strategy for the blindness bug, or after installing an update.

(#__#) debugging

Used for debug and test messages on screen.

Basic requirements for running Pwnagotchi :

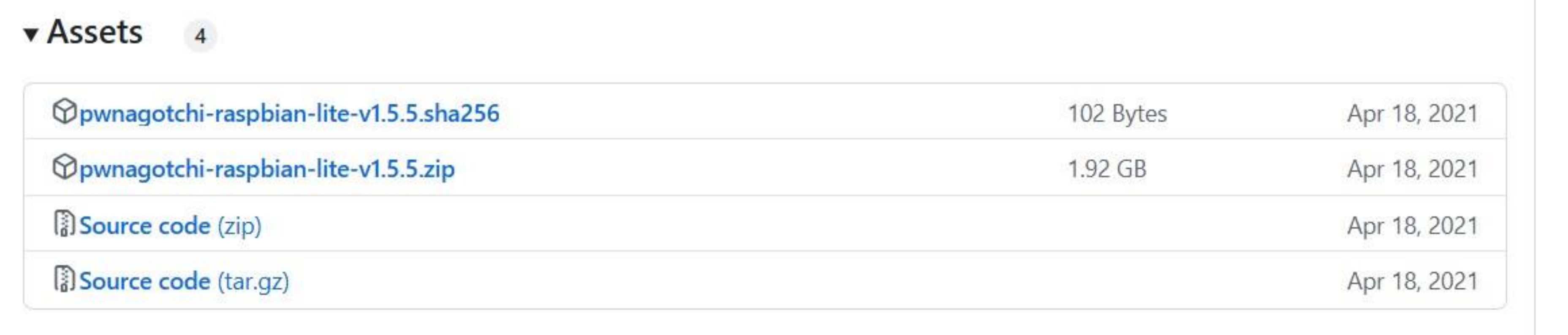


Hardware:

- **Raspberry Pi Zero W.**
- **MicroSD card** (at least 8GB)
- **Power source** (portable battery or USB)
- SD card USB adapter.
- **eInk display** (optional for face display)
- **Case** (optional, for protection)

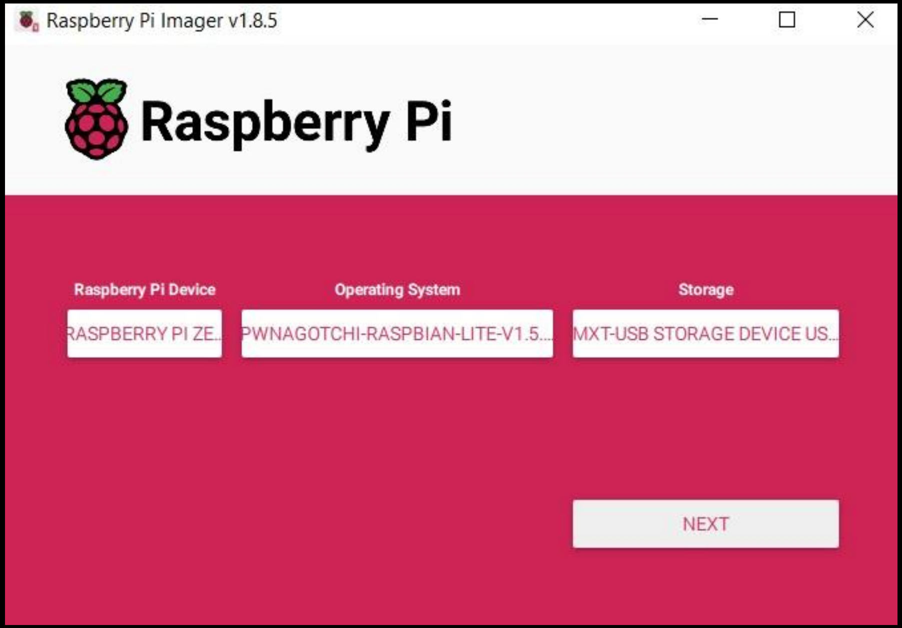
-Steps to Install Pwnagotchi:

https://github.com/evilsocket/pwnagotchi/releasesf



▾ Assets 4

pwnagotchi-raspbian-lite-v1.5.5.sha256	102 Bytes	Apr 18, 2021
pwnagotchi-raspbian-lite-v1.5.5.zip	1.92 GB	Apr 18, 2021
Source code (zip)		Apr 18, 2021
Source code (tar.gz)		Apr 18, 2021



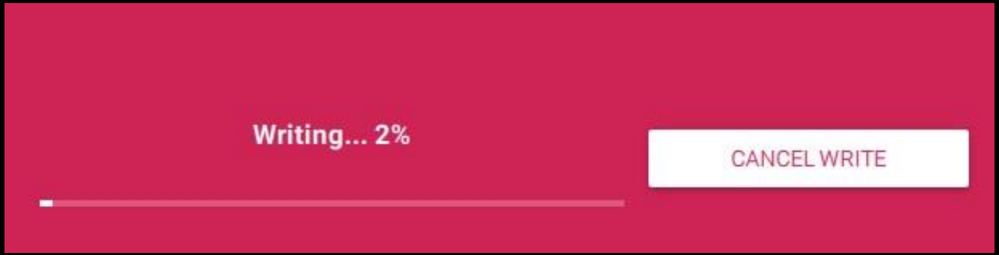
Raspberry Pi Imager v1.8.5

Raspberry Pi

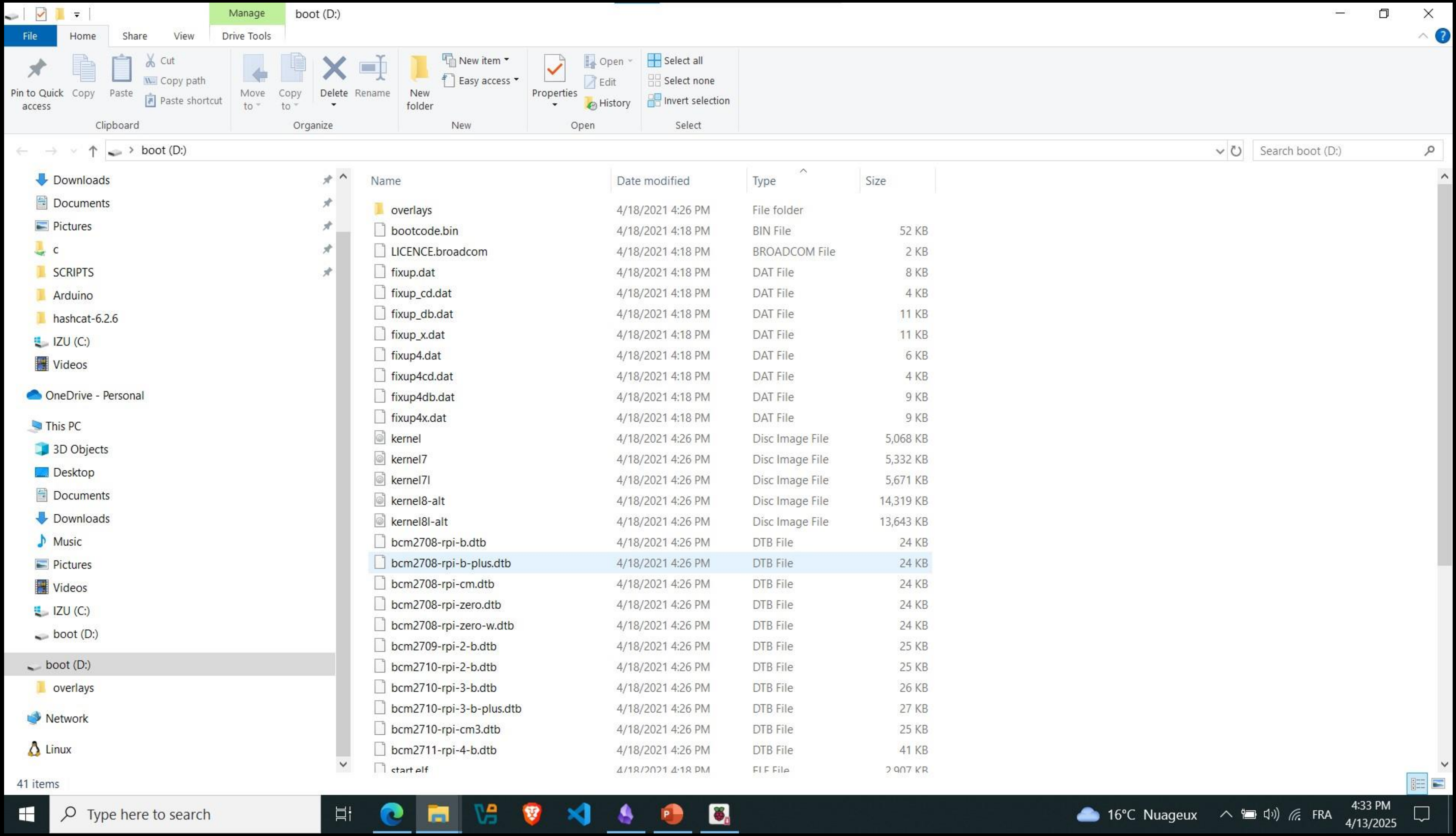
Raspberry Pi Device: RASPBERRY PI ZE...

Operating System: PWNAGOTCHI-RASPBIAN-LITE-V1.5...

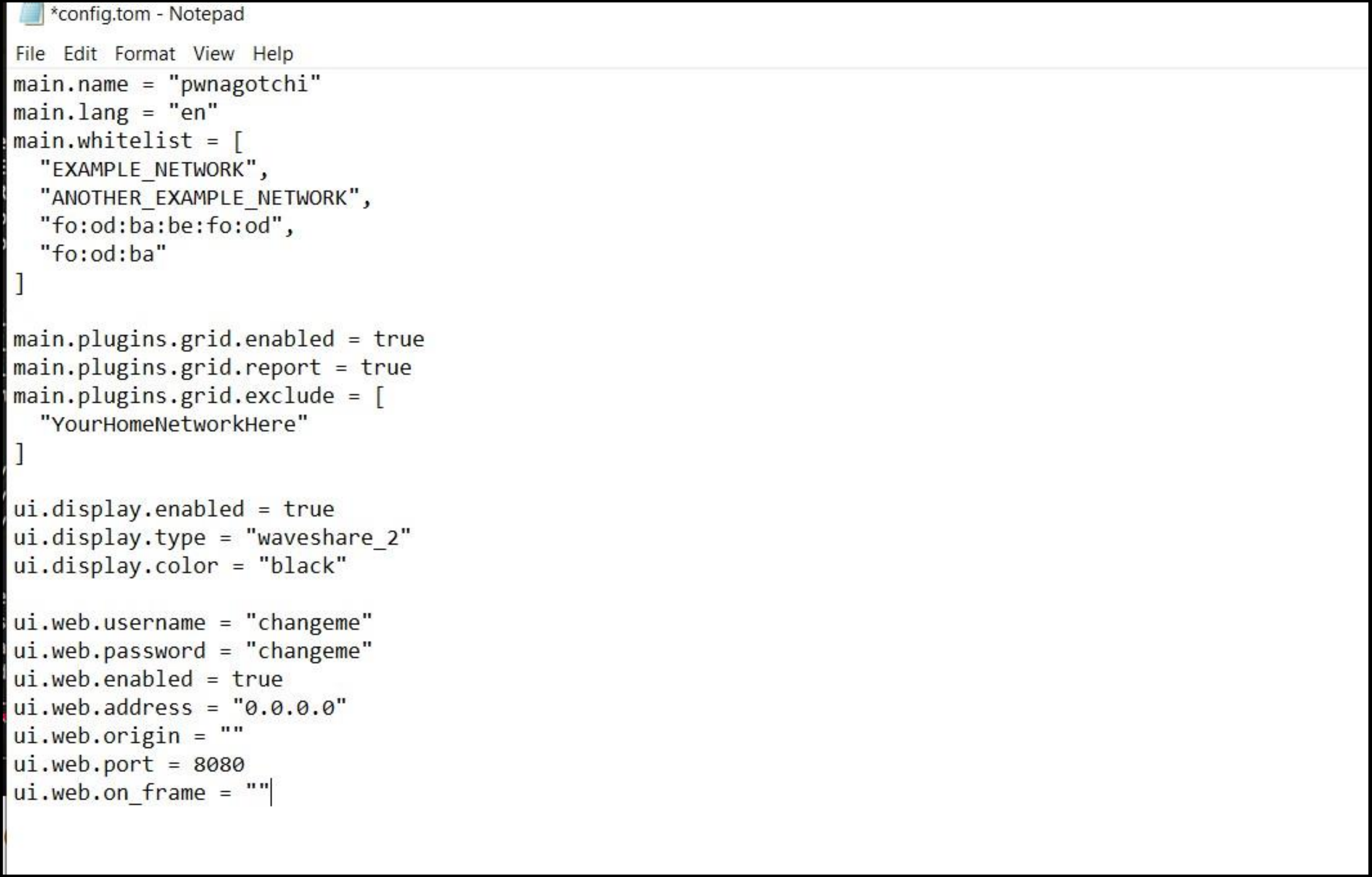
Storage: MXT-USB STORAGE DEVICE US...



Writing... 2%



Name	Date modified	Type	Size
overlays	4/18/2021 4:26 PM	File folder	
bootcode.bin	4/18/2021 4:18 PM	BIN File	52 KB
LICENCE.broadcom	4/18/2021 4:18 PM	BROADCOM File	2 KB
fixup.dat	4/18/2021 4:18 PM	DAT File	8 KB
fixup_cd.dat	4/18/2021 4:18 PM	DAT File	4 KB
fixup_db.dat	4/18/2021 4:18 PM	DAT File	11 KB
fixup_x.dat	4/18/2021 4:18 PM	DAT File	11 KB
fixup4.dat	4/18/2021 4:18 PM	DAT File	6 KB
fixup4cd.dat	4/18/2021 4:18 PM	DAT File	4 KB
fixup4db.dat	4/18/2021 4:18 PM	DAT File	9 KB
fixup4x.dat	4/18/2021 4:18 PM	DAT File	9 KB
kernel	4/18/2021 4:26 PM	Disc Image File	5,068 KB
kernel7	4/18/2021 4:26 PM	Disc Image File	5,332 KB
kernel7l	4/18/2021 4:26 PM	Disc Image File	5,671 KB
kernel8-alt	4/18/2021 4:26 PM	Disc Image File	14,319 KB
kernel8l-alt	4/18/2021 4:26 PM	Disc Image File	13,643 KB
bcm2708-rpi-b.dtb	4/18/2021 4:26 PM	DTB File	24 KB
bcm2708-rpi-b-plus.dtb	4/18/2021 4:26 PM	DTB File	24 KB
bcm2708-rpi-cm.dtb	4/18/2021 4:26 PM	DTB File	24 KB
bcm2708-rpi-zero.dtb	4/18/2021 4:26 PM	DTB File	24 KB
bcm2708-rpi-zero-w.dtb	4/18/2021 4:26 PM	DTB File	24 KB
bcm2709-rpi-2-b.dtb	4/18/2021 4:26 PM	DTB File	25 KB
bcm2710-rpi-2-b.dtb	4/18/2021 4:26 PM	DTB File	25 KB
bcm2710-rpi-3-b.dtb	4/18/2021 4:26 PM	DTB File	26 KB
bcm2710-rpi-3-b-plus.dtb	4/18/2021 4:26 PM	DTB File	27 KB
bcm2710-rpi-cm3.dtb	4/18/2021 4:26 PM	DTB File	25 KB
bcm2711-rpi-4-b.dtb	4/18/2021 4:26 PM	DTB File	41 KB
start.elf	4/18/2021 4:18 PM	ELF File	2,907 KB



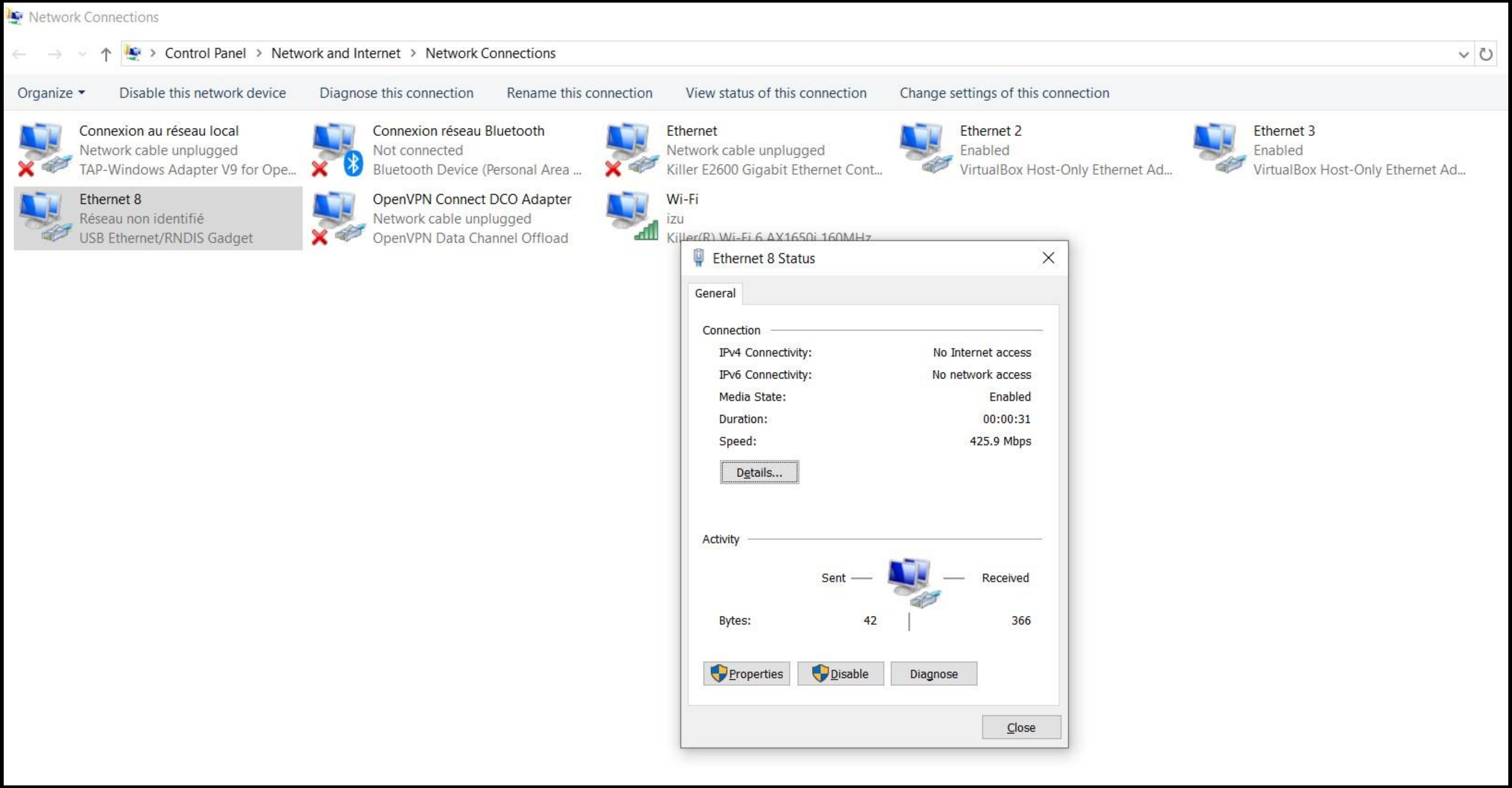
```
main.name = "pwnagotchi"
main.lang = "en"
main.whitelist = [
  "EXAMPLE_NETWORK",
  "ANOTHER_EXAMPLE_NETWORK",
  "fo:od:ba:be:fo:od",
  "fo:od:ba"
]

main.plugins.grid.enabled = true
main.plugins.grid.report = true
main.plugins.grid.exclude = [
  "YourHomeNetworkHere"
]

ui.display.enabled = true
ui.display.type = "waveshare_2"
ui.display.color = "black"

ui.web.username = "changeme"
ui.web.password = "changeme"
ui.web.enabled = true
ui.web.address = "0.0.0.0"
ui.web.origin = ""
ui.web.port = 8080
ui.web.on_frame = ""
```

-Insert the SD card and plug the raspberry .


Network Connections
Control Panel > Network and Internet > Network Connections
Organize
Disable this network device
Diagnose this connection
Rename this connection
View status of this connection
Change settings of this connection

Connexion au réseau local
Network cable unplugged
TAP-Windows Adapter V9 for Ope...

Connexion réseau Bluetooth
Not connected
Bluetooth Device (Personal Area ...

Ethernet
Network cable unplugged
Killer E2600 Gigabit Ethernet Cont...

Ethernet 2
Enabled
VirtualBox Host-Only Ethernet Ad...

Ethernet 3
Enabled
VirtualBox Host-Only Ethernet Ad...

Ethernet 8
Réseau non identifié
USB Ethernet/RNDIS Gadget

OpenVPN Connect DCO Adapter
Network cable unplugged
OpenVPN Data Channel Offload

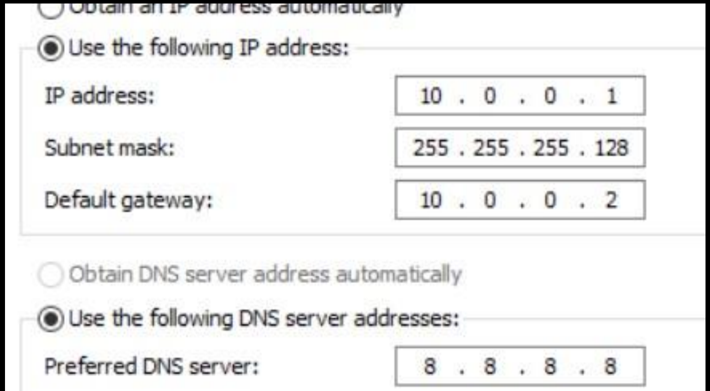
Wi-Fi
izu
Killer(R) Wi-Fi 6 AX1650i 160MHz

Ethernet 8 Status

General

Connection
IPv4 Connectivity: No Internet access
IPv6 Connectivity: No network access
Media State: Enabled
Duration: 00:00:31
Speed: 425.9 Mbps

Activity
Sent — Received
Bytes: 42 | 366



○ Obtain an IP address automatically

● Use the following IP address:

IP address: 10 . 0 . 0 . 1

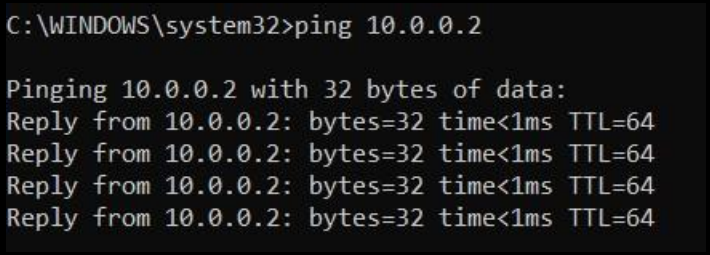
Subnet mask: 255 . 255 . 255 . 128

Default gateway: 10 . 0 . 0 . 2

○ Obtain DNS server address automatically

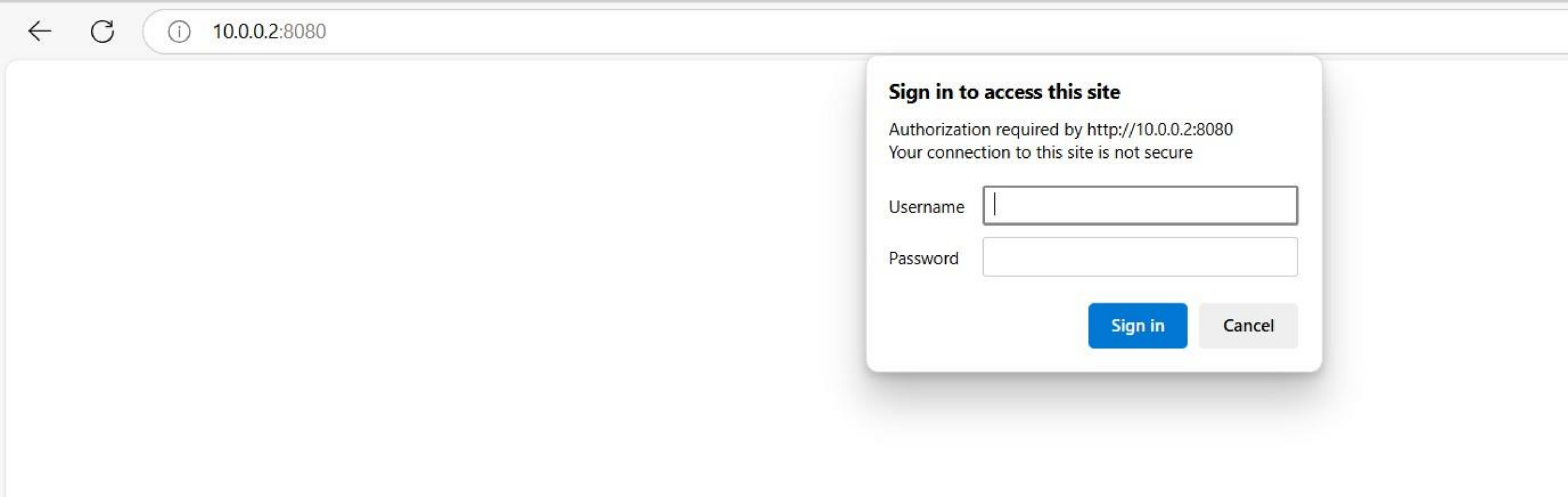
● Use the following DNS server addresses:

Preferred DNS server: 8 . 8 . 8 . 8



```
C:\WINDOWS\system32>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time<1ms TTL=64
Reply from 10.0.0.2: bytes=32 time<1ms TTL=64
Reply from 10.0.0.2: bytes=32 time<1ms TTL=64
Reply from 10.0.0.2: bytes=32 time<1ms TTL=64
```



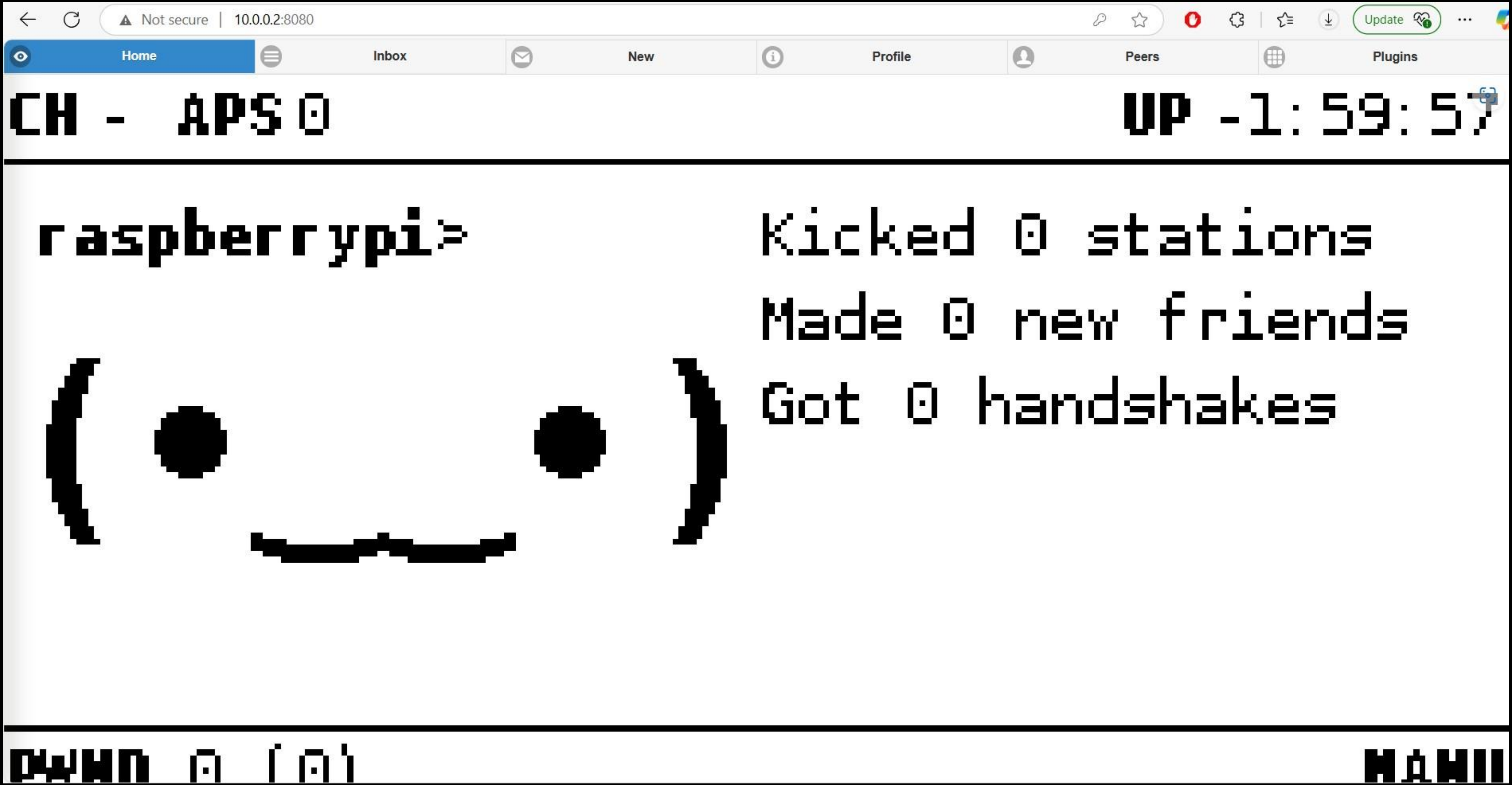
10.0.0.2:8080

Sign in to access this site

Authorization required by http://10.0.0.2:8080
Your connection to this site is not secure

Username

Password

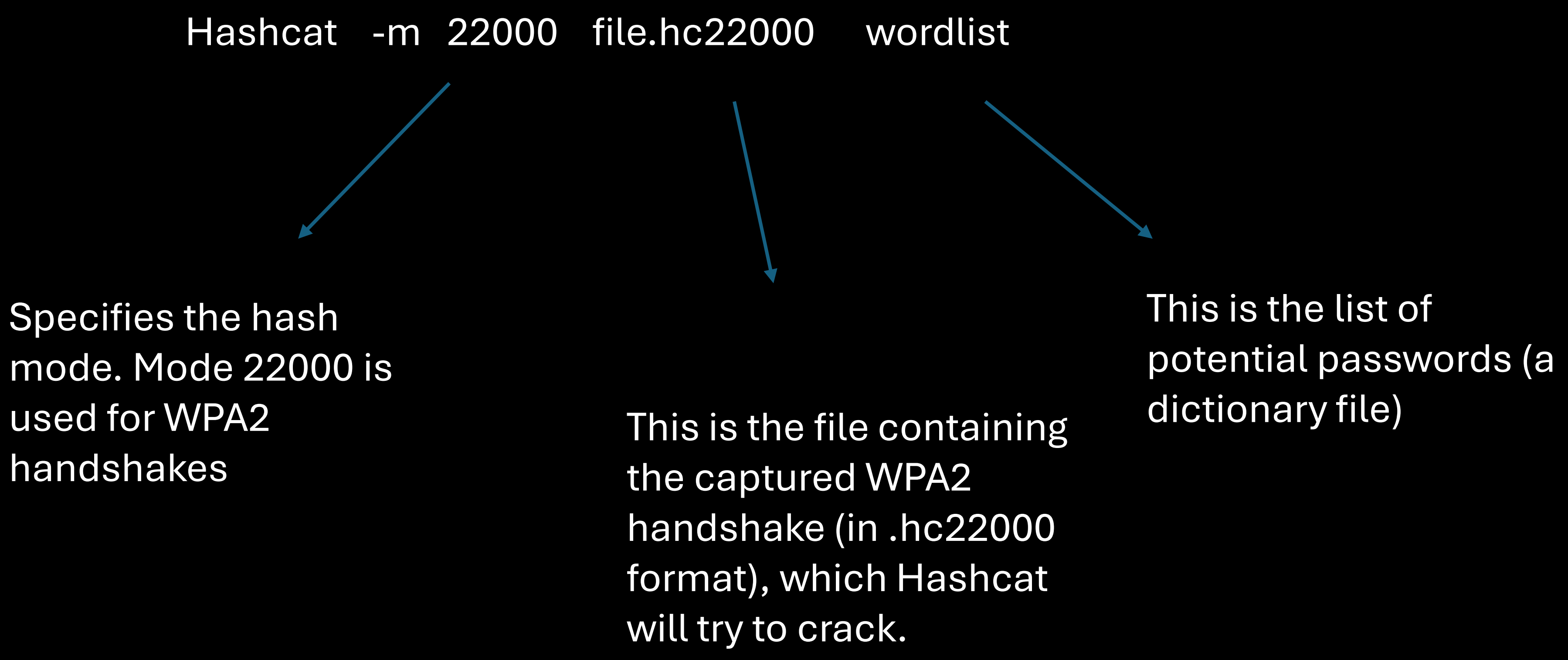


```
CH -  APS 0                                   UP -1:59:57

raspberrypi>                 Kicked 0 stations
                             Made 0 new friends
                             Got 0 handshakes

PWND 0 (0)                                           MANU
```

Hashcat command :



```
Hashcat   -m   22000   file.hc22000    wordlist
```

Specifies the hash mode. Mode 22000 is used for WPA2 handshakes

This is the file containing the captured WPA2 handshake (in .hc22000 format), which Hashcat will try to crack.

This is the list of potential passwords (a dictionary file)

Application :

-**Access Point (AP)** name is **izu.**

-The password is **MuZhlo9n%8!G**

-This password appears strong because it includes a mix of uppercase letters, lowercase letters, numbers, and special characters. However, it can be cracked in seconds using a powerful **GPU** and a good **wordlist** (like those from Seclists on GitHub). The reason is that with the right hardware and wordlist, tools like **Hashcat** can quickly try many potential passwords and break even complex ones in a short time.

FIN .