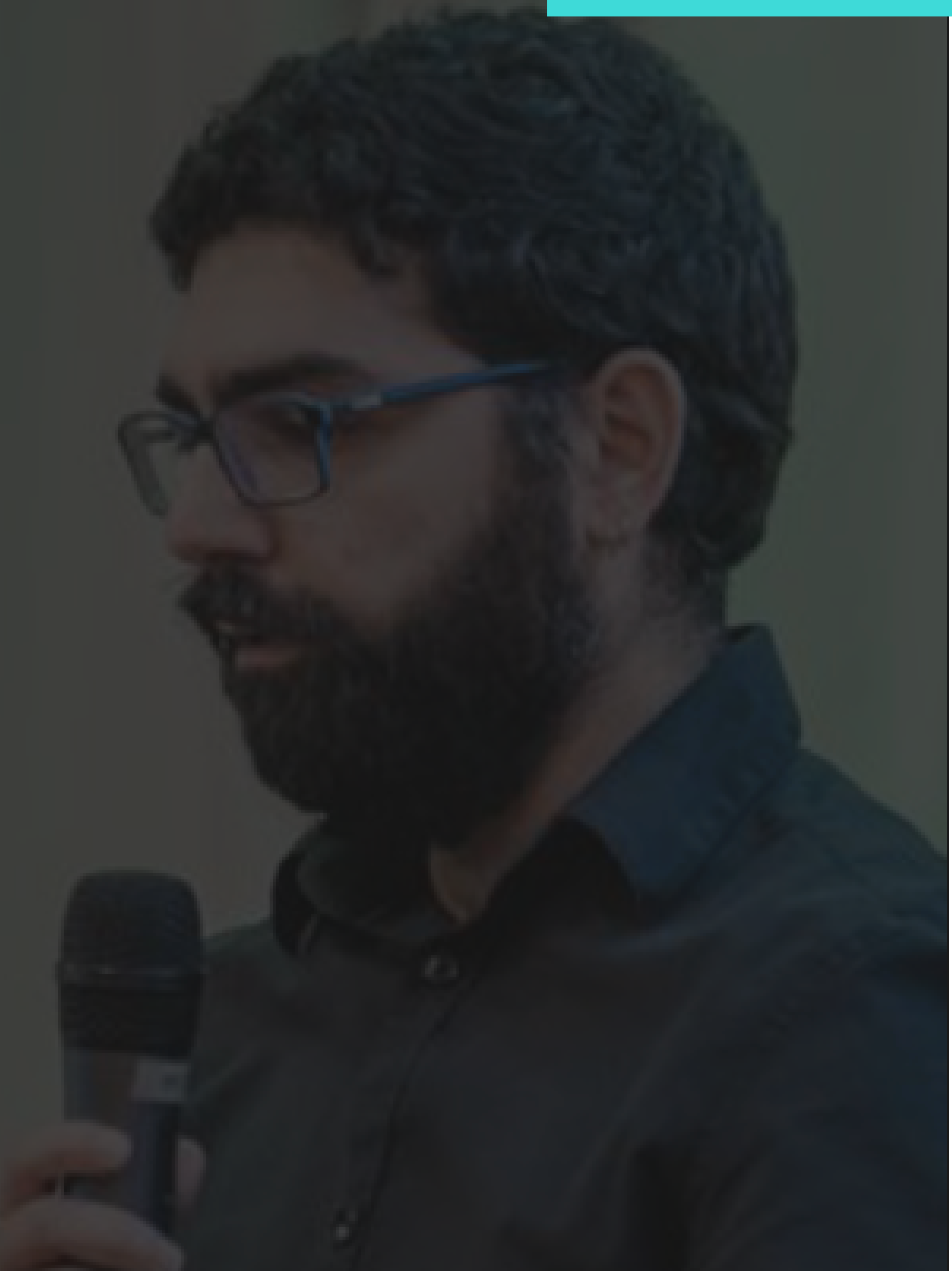
Non-Traditional Attack Techniques

HIGHLIGHTS FROM A STUDY ON SIDE-CHANNEL AND AIR-GAP ATTACKS

HackInBo® Winter Edition 2023

HACKINBO®
Winter 2023 Edition
21ª EDIZIONE



Rocco Sicilia
Who?

red team

white hat

hacking

chuck norris

$buzzword

cyber security

info security

blue team

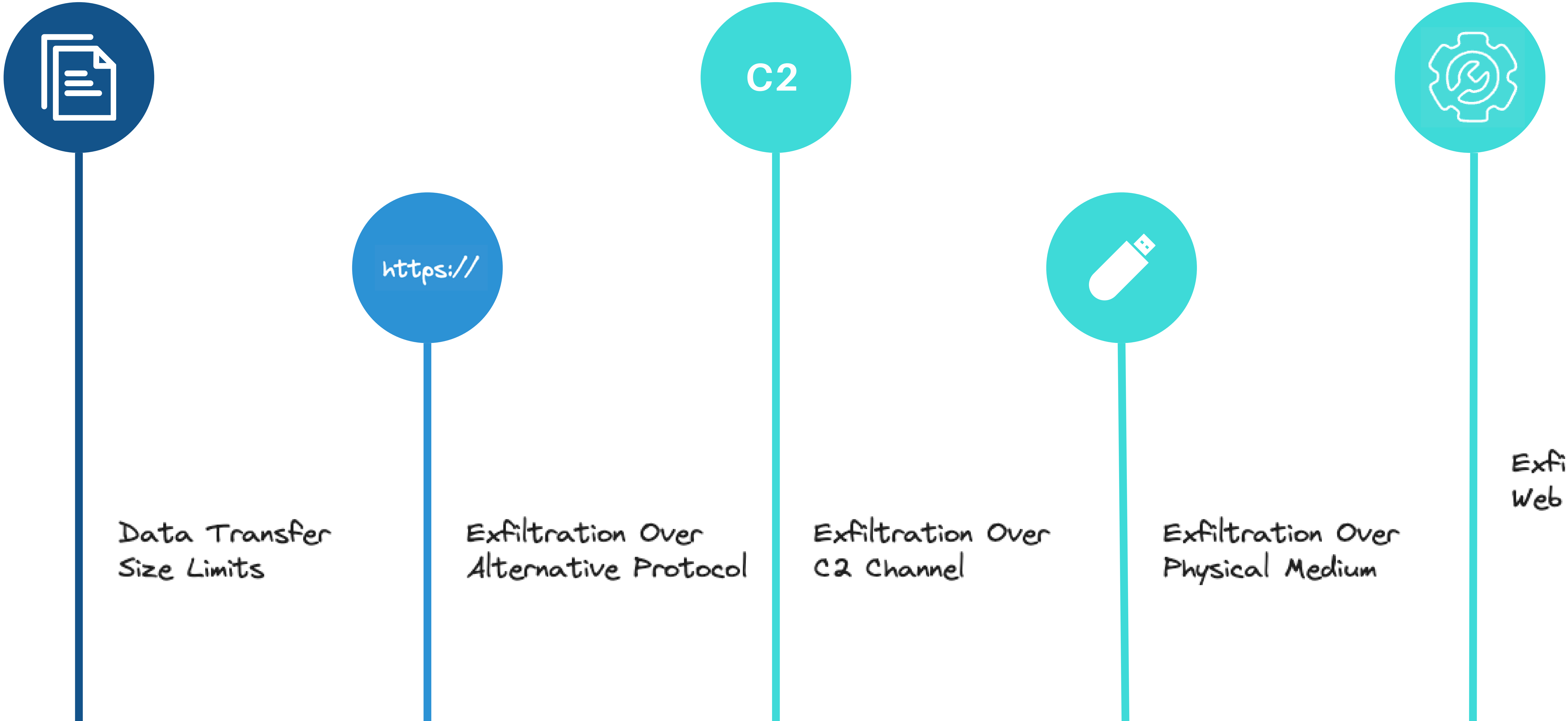
bug hunting

Agenda

1. Traditional Techniques
2. Side channel
3. Air-gap
4. Scenario
5. Lab - LED light
6. Lab - Have you heard anything?
7. Beer

Traditional Techniques

Data Exfiltration [TA0010]



Data Transfer Size Limits

Exfiltration Over Alternative Protocol

Exfiltration Over C2 Channel

Exfiltration Over Physical Medium

Exfiltration Over Web Services

Side Channel

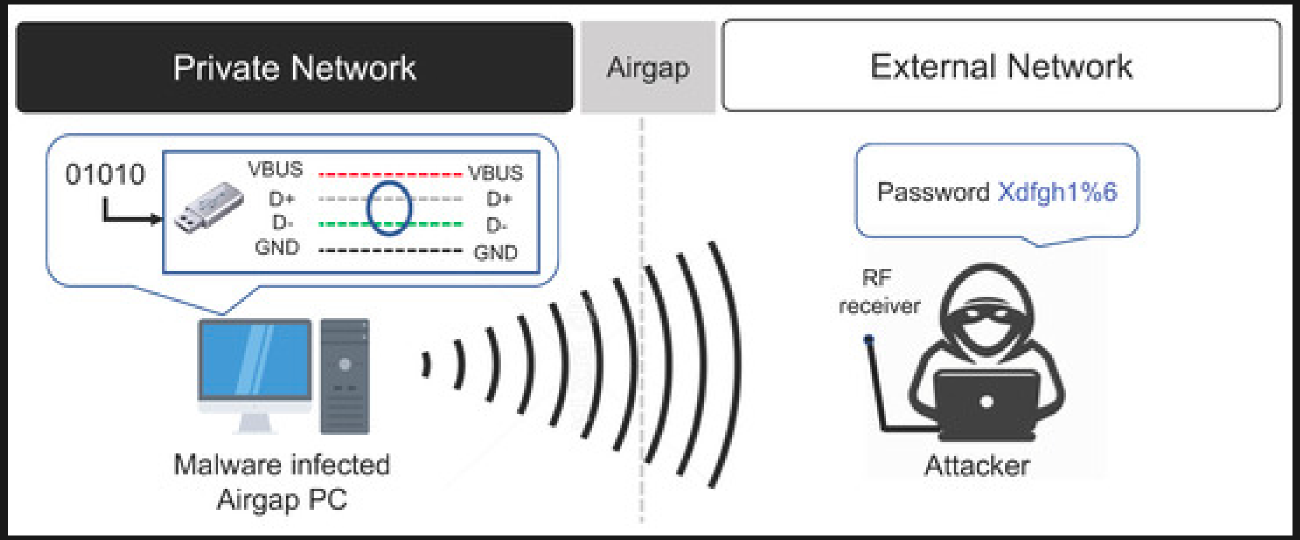
[...] attack based on extra information that can be gathered because of the fundamental way a computer protocol or algorithm is implemented [...]

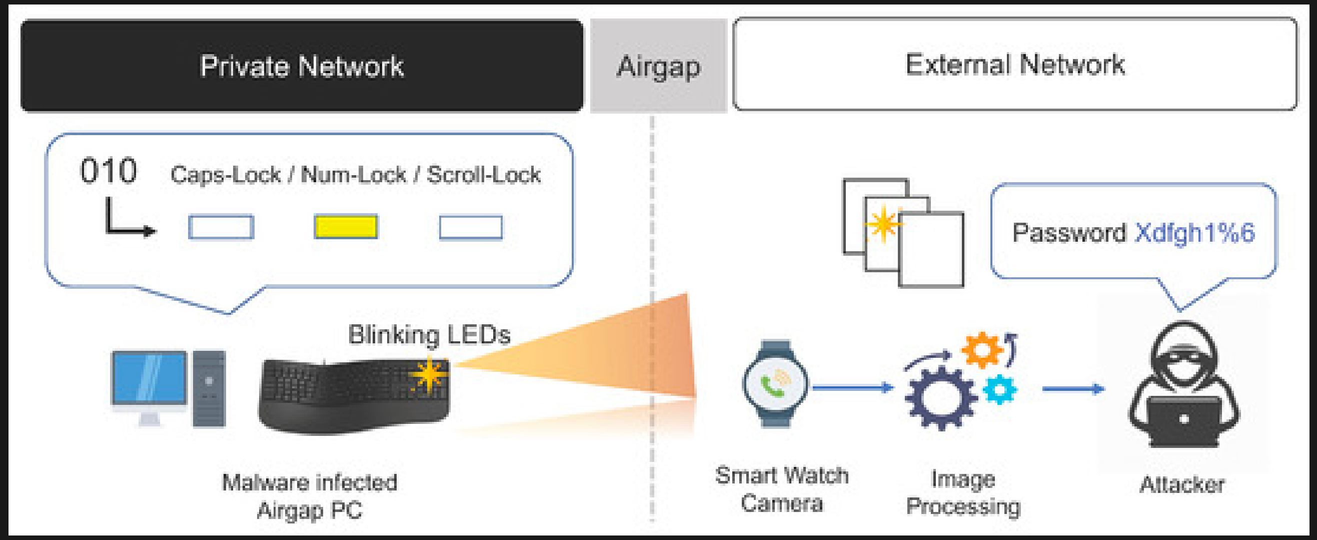
Electromagnetic Attack

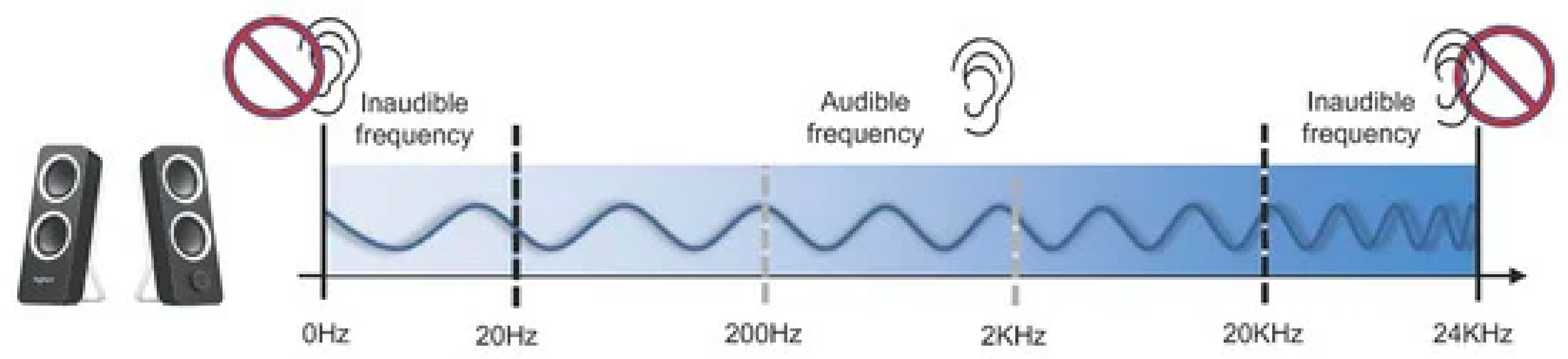
Power-Monitoring Attack

Acoustic Cryptanalysis

Air-gap

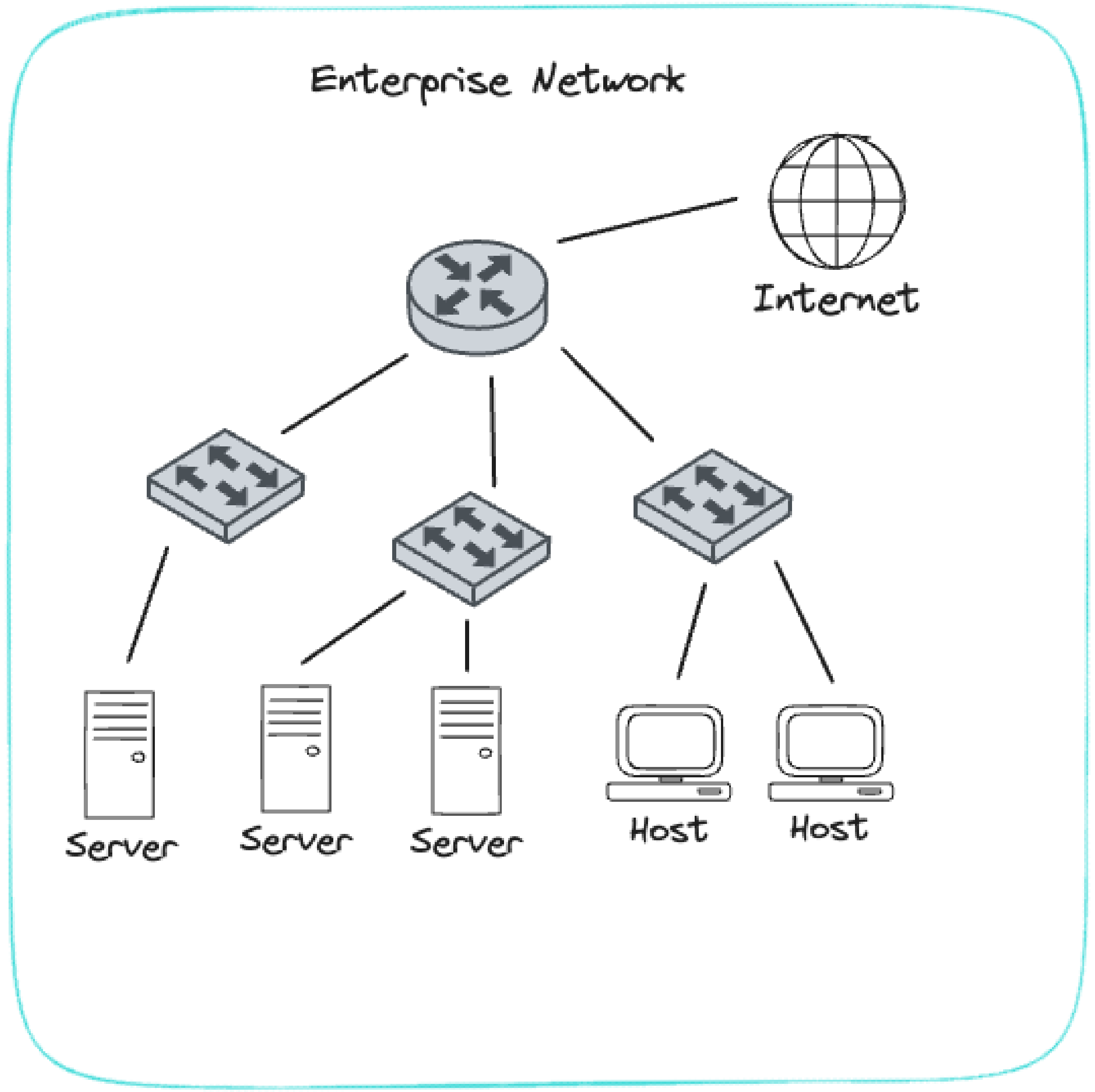


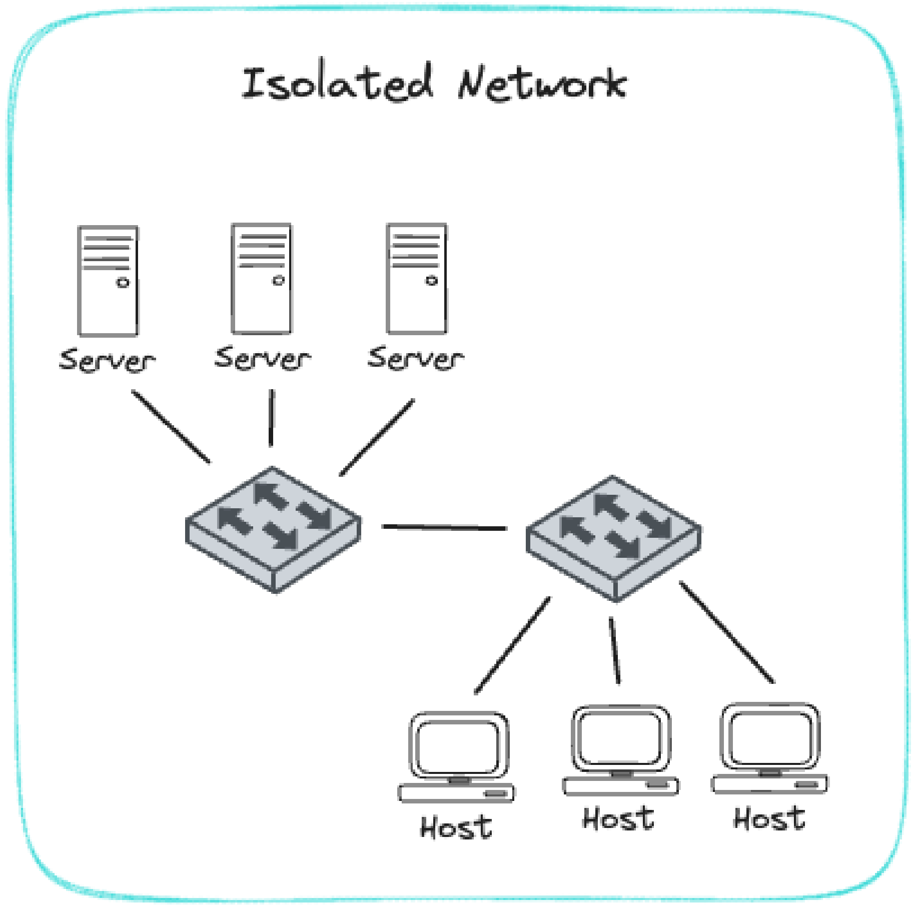




Paper: https://www.mdpi.com/1424-8220/23/6/3215

Scenario


Enterprise Network
Internet
Server
Server
Server
Host
Host


Isolated Network
Server
Server
Server
Host
Host
Host

Scenario

Scenario

led light

led light

Physical Access

Distance from the device

Time

1

constraints

led light

Physical Access

Distance from the device

Time

Job Interview

1

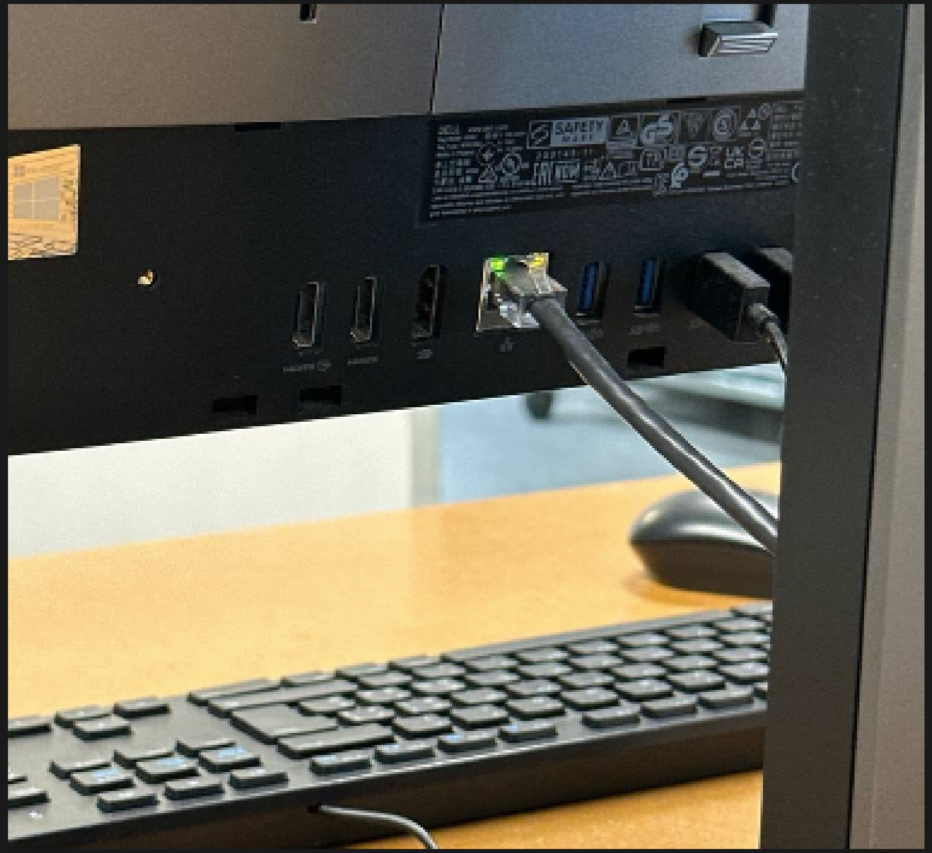
constraints

led light

Physical Access

Distance from the device

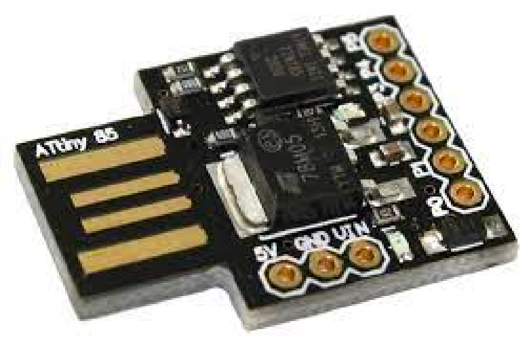
Time

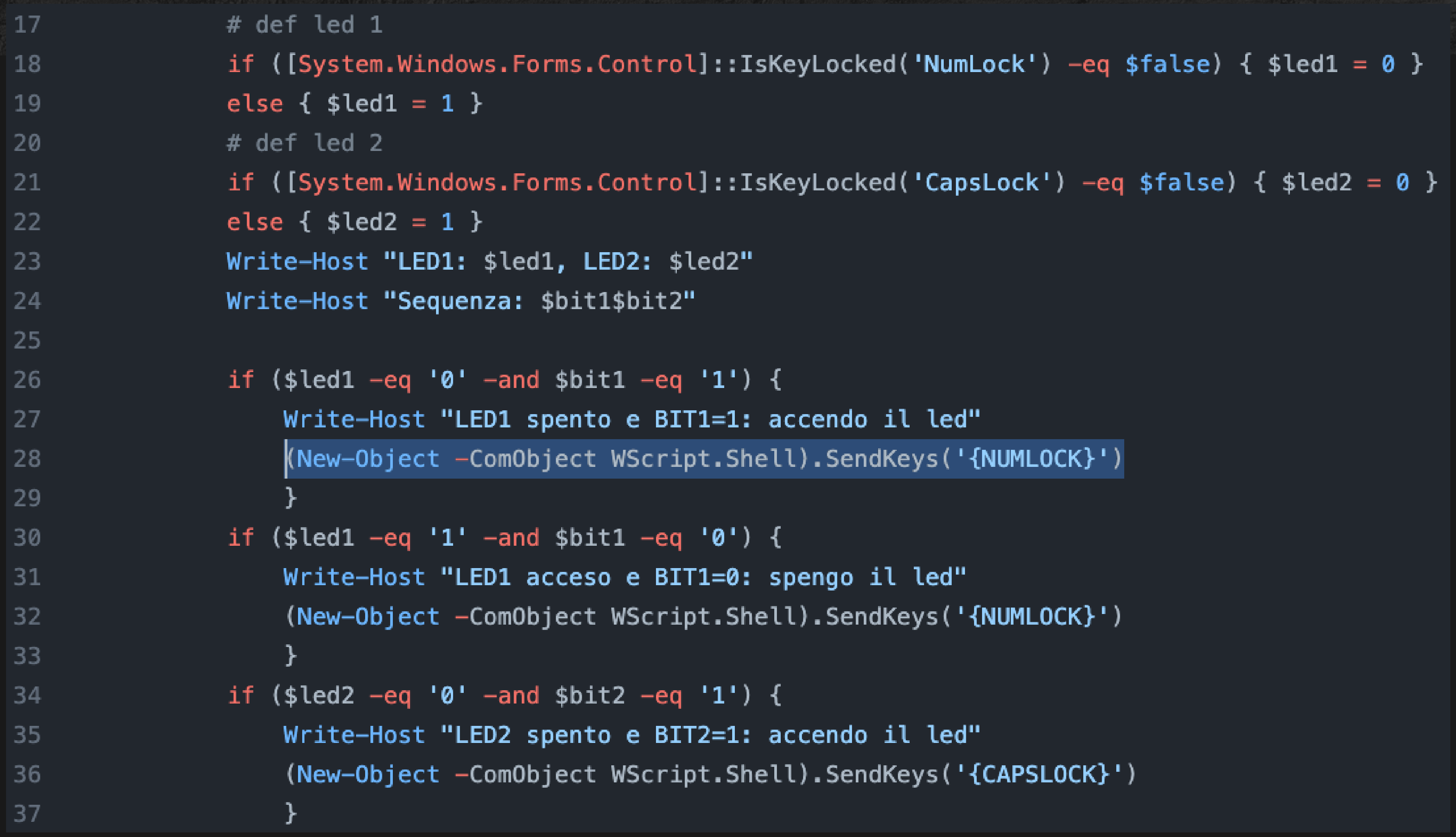


1

constraints

led light





```
        # def led 1
        if ([System.Windows.Forms.Control]::IsKeyLocked('NumLock') -eq $false) { $led1 = 0 }
        else { $led1 = 1 }
        # def led 2
        if ([System.Windows.Forms.Control]::IsKeyLocked('CapsLock') -eq $false) { $led2 = 0 }
        else { $led2 = 1 }
        Write-Host "LED1: $led1, LED2: $led2"
        Write-Host "Sequenza: $bit1$bit2"

        if ($led1 -eq '0' -and $bit1 -eq '1') {
            Write-Host "LED1 spento e BIT1=1: accendo il led"
            (New-Object -ComObject WScript.Shell).SendKeys('{NUMLOCK}')
            }
        if ($led1 -eq '1' -and $bit1 -eq '0') {
            Write-Host "LED1 acceso e BIT1=0: spengo il led"
            (New-Object -ComObject WScript.Shell).SendKeys('{NUMLOCK}')
            }
        if ($led2 -eq '0' -and $bit2 -eq '1') {
            Write-Host "LED2 spento e BIT2=1: accendo il led"
            (New-Object -ComObject WScript.Shell).SendKeys('{CAPSLOCK}')
            }
```

weapon

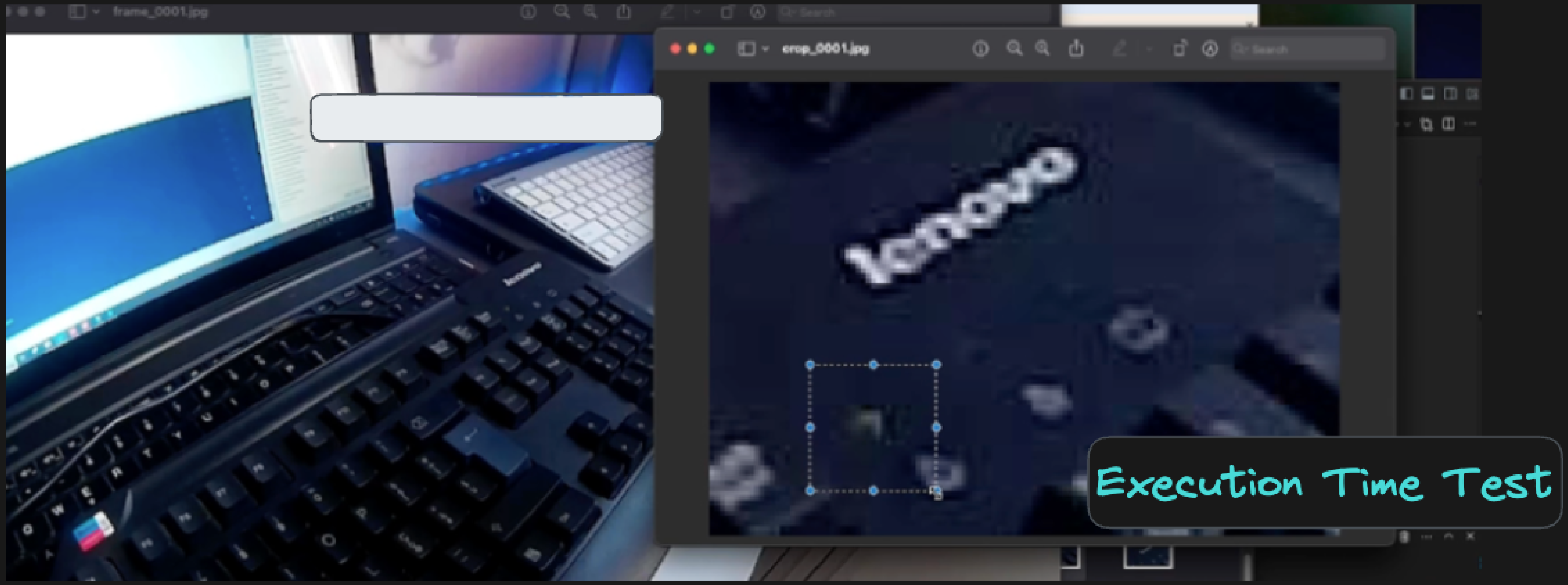
1

2

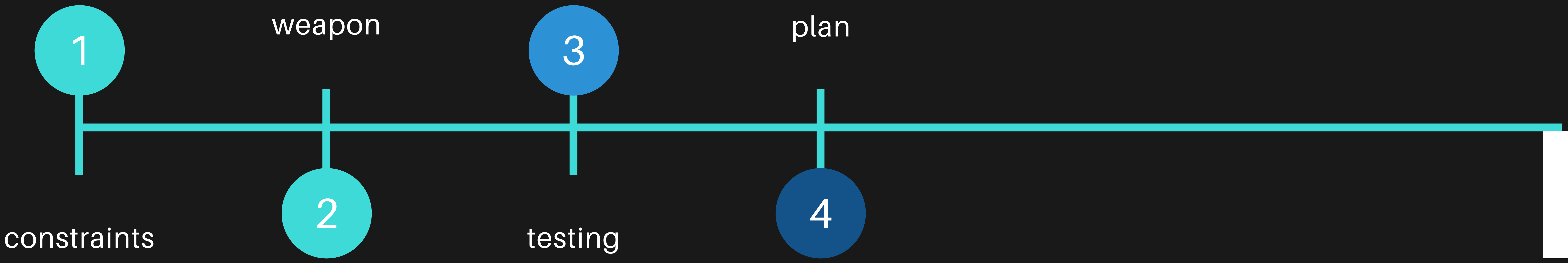
constraints

HackInBo® Winter 2023 Edition
21ª EDIZIONE

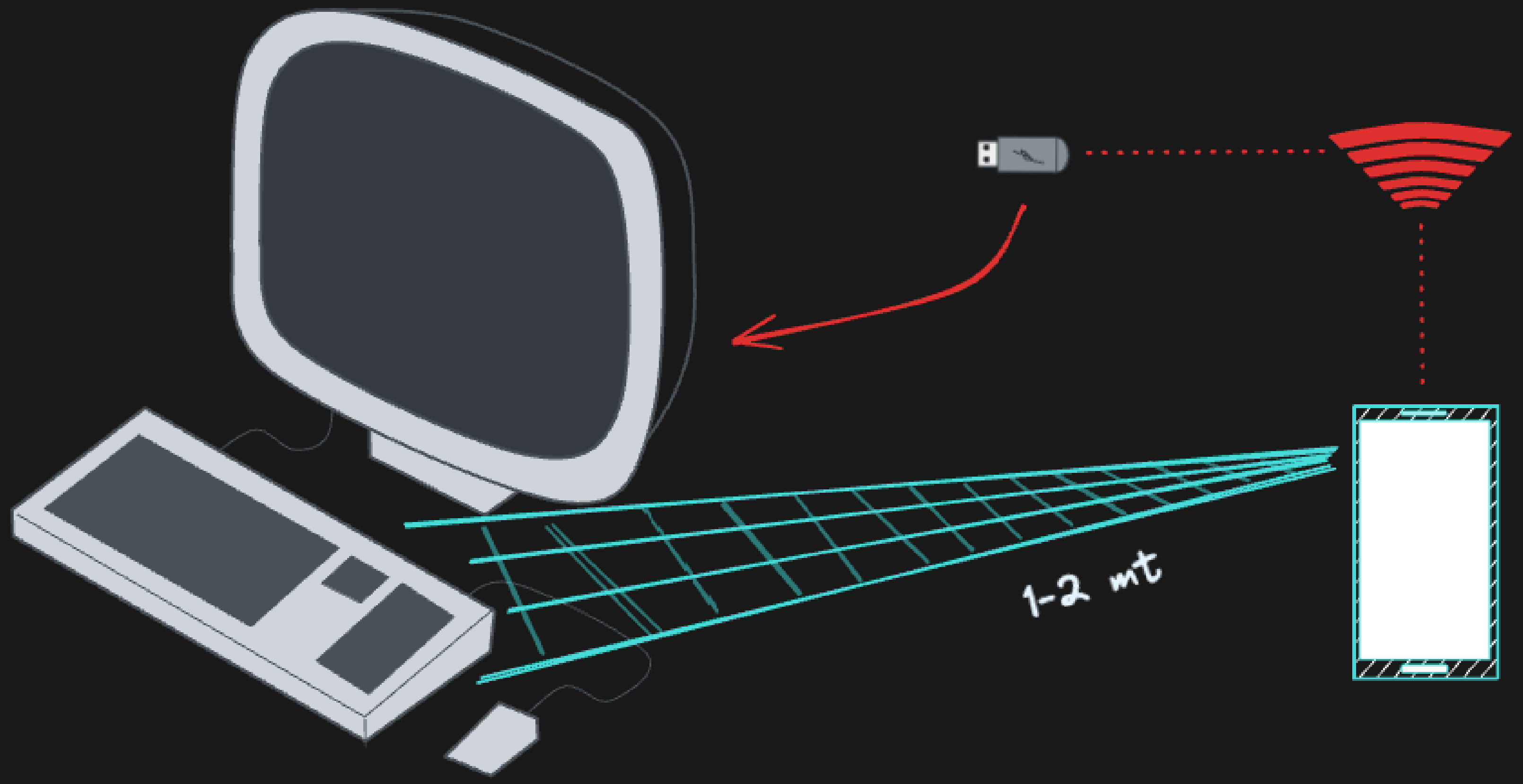
led light



Execution Time Test



led light



1 constraints

2 weapon

3 testing

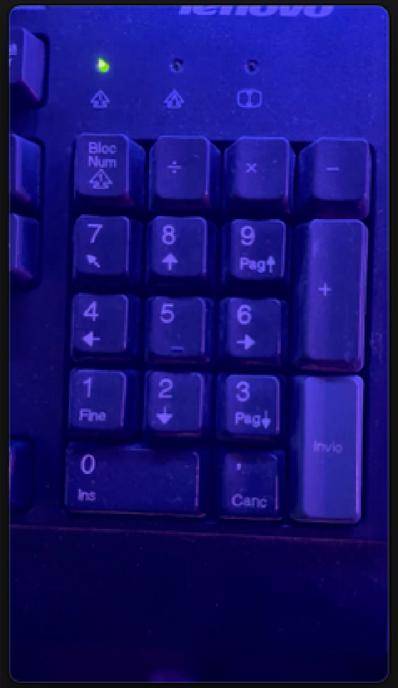
4 plan

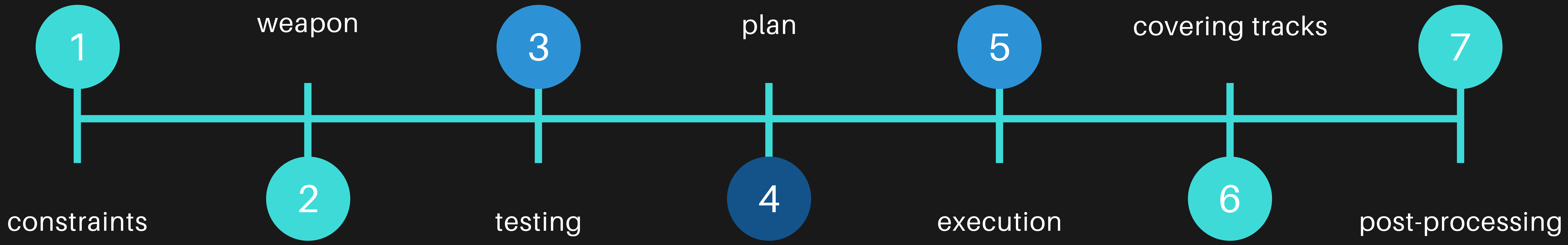
5 execution

6 covering tracks

HackInBo Winter 2023 Edition
21ª EDIZIONE

led light





1 constraints

2 weapon

3 testing

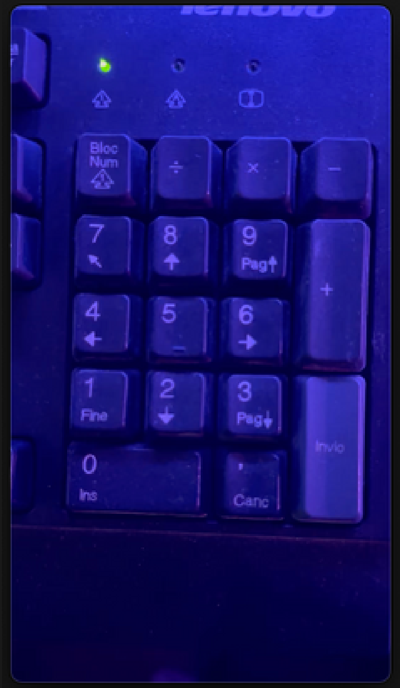
4 plan

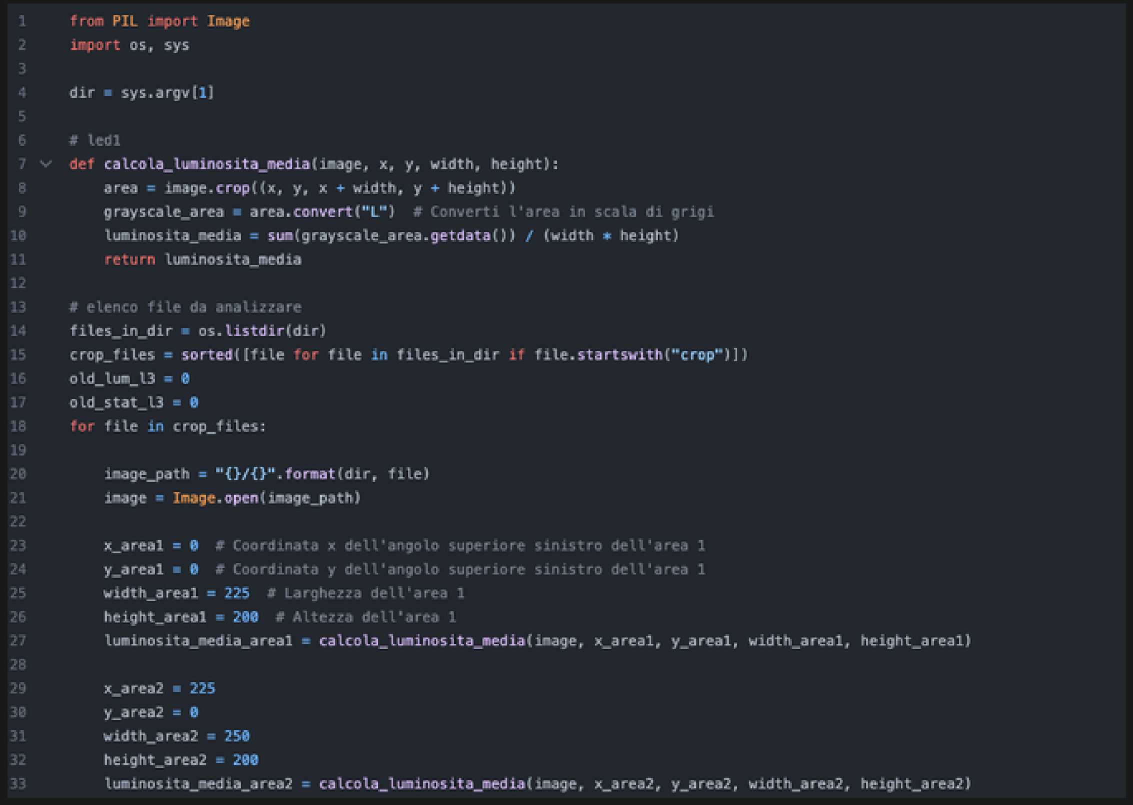
5 execution

6 covering tracks

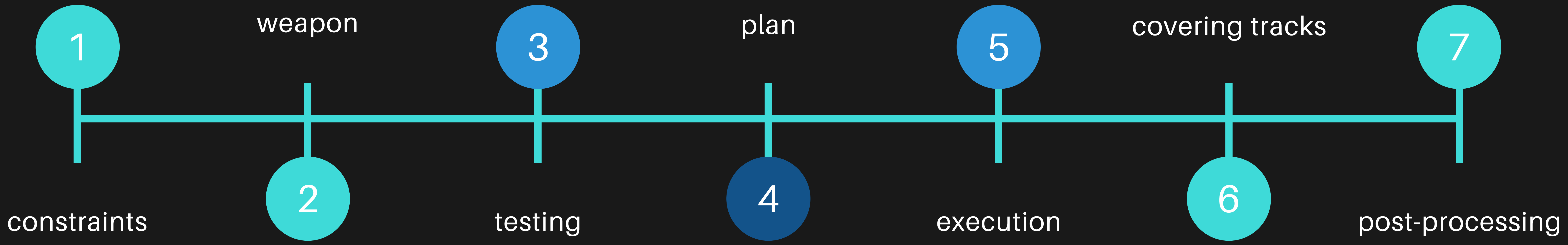
7 post-processing

led light





```
from PIL import Image
import os, sys

dir = sys.argv[1]

# led1
def calcola_luminosita_media(image, x, y, width, height):
    area = image.crop((x, y, x + width, y + height))
    grayscale_area = area.convert("L")  # Converti l'area in scala di grigi
    luminosita_media = sum(grayscale_area.getdata()) / (width * height)
    return luminosita_media

# elenco file da analizzare
files_in_dir = os.listdir(dir)
crop_files = sorted([file for file in files_in_dir if file.startswith("crop")])
old_lum_l3 = 0
old_stat_l3 = 0
for file in crop_files:

    image_path = "{}/{}".format(dir, file)
    image = Image.open(image_path)

    x_area1 = 0  # Coordinata x dell'angolo superiore sinistro dell'area 1
    y_area1 = 0  # Coordinata y dell'angolo superiore sinistro dell'area 1
    width_area1 = 225  # Larghezza dell'area 1
    height_area1 = 200  # Altezza dell'area 1
    luminosita_media_area1 = calcola_luminosita_media(image, x_area1, y_area1, width_area1, height_area1)

    x_area2 = 225
    y_area2 = 0
    width_area2 = 250
    height_area2 = 200
    luminosita_media_area2 = calcola_luminosita_media(image, x_area2, y_area2, width_area2, height_area2)
```



1 constraints

2 weapon

3 testing

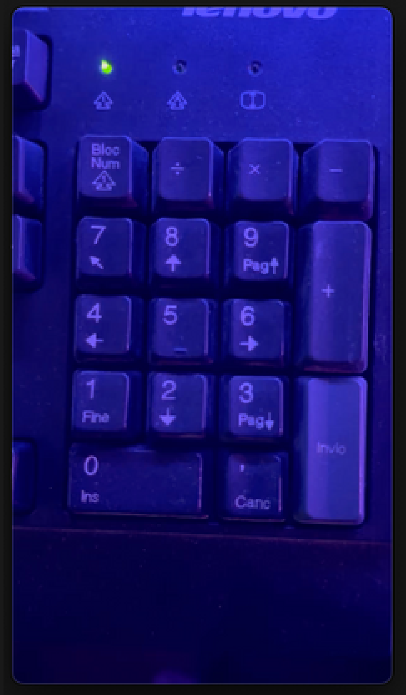
4 plan

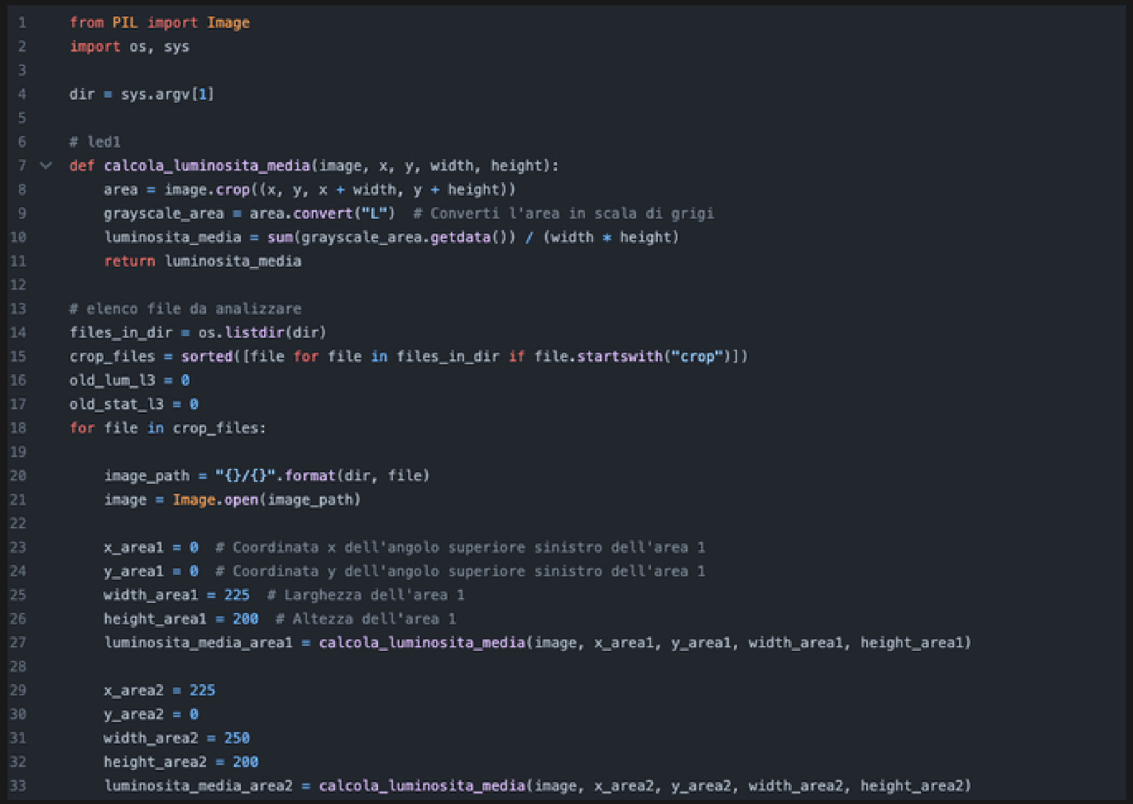
5 execution

6 covering tracks

7 post-processing

led light

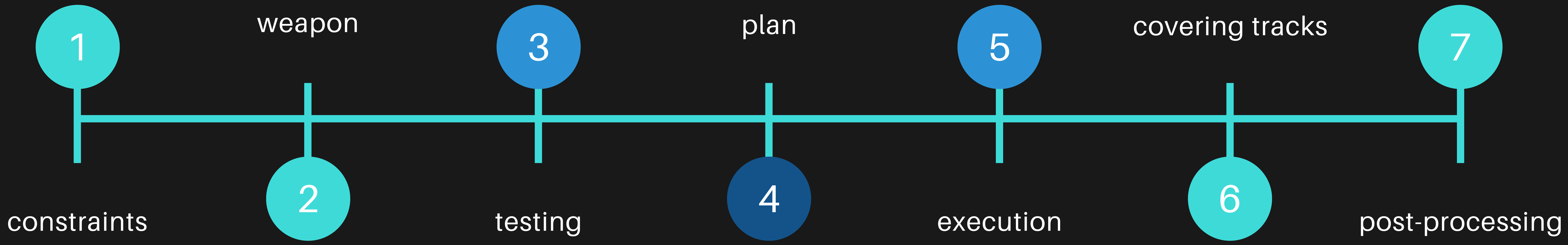




```
from PIL import Image
import os, sys

dir = sys.argv[1]

# led1
def calcola_luminosita_media(image, x, y, width, height):
    area = image.crop((x, y, x + width, y + height))
    grayscale_area = area.convert("L")  # Converti l'area in scala di grigi
    luminosita_media = sum(grayscale_area.getdata()) / (width * height)
    return luminosita_media

# elenco file da analizzare
files_in_dir = os.listdir(dir)
crop_files = sorted([file for file in files_in_dir if file.startswith("crop")])
old_lum_l3 = 0
old_stat_l3 = 0
for file in crop_files:

    image_path = "{}/{}".format(dir, file)
    image = Image.open(image_path)

    x_area1 = 0  # Coordinata x dell'angolo superiore sinistro dell'area 1
    y_area1 = 0  # Coordinata y dell'angolo superiore sinistro dell'area 1
    width_area1 = 225  # Larghezza dell'area 1
    height_area1 = 200  # Altezza dell'area 1
    luminosita_media_area1 = calcola_luminosita_media(image, x_area1, y_area1, width_area1, height_area1)

    x_area2 = 225
    y_area2 = 0
    width_area2 = 250
    height_area2 = 200
    luminosita_media_area2 = calcola_luminosita_media(image, x_area2, y_area2, width_area2, height_area2)
```

Original information

01100010 01100101 01100101 01110010



1 constraints

2 weapon

3 testing

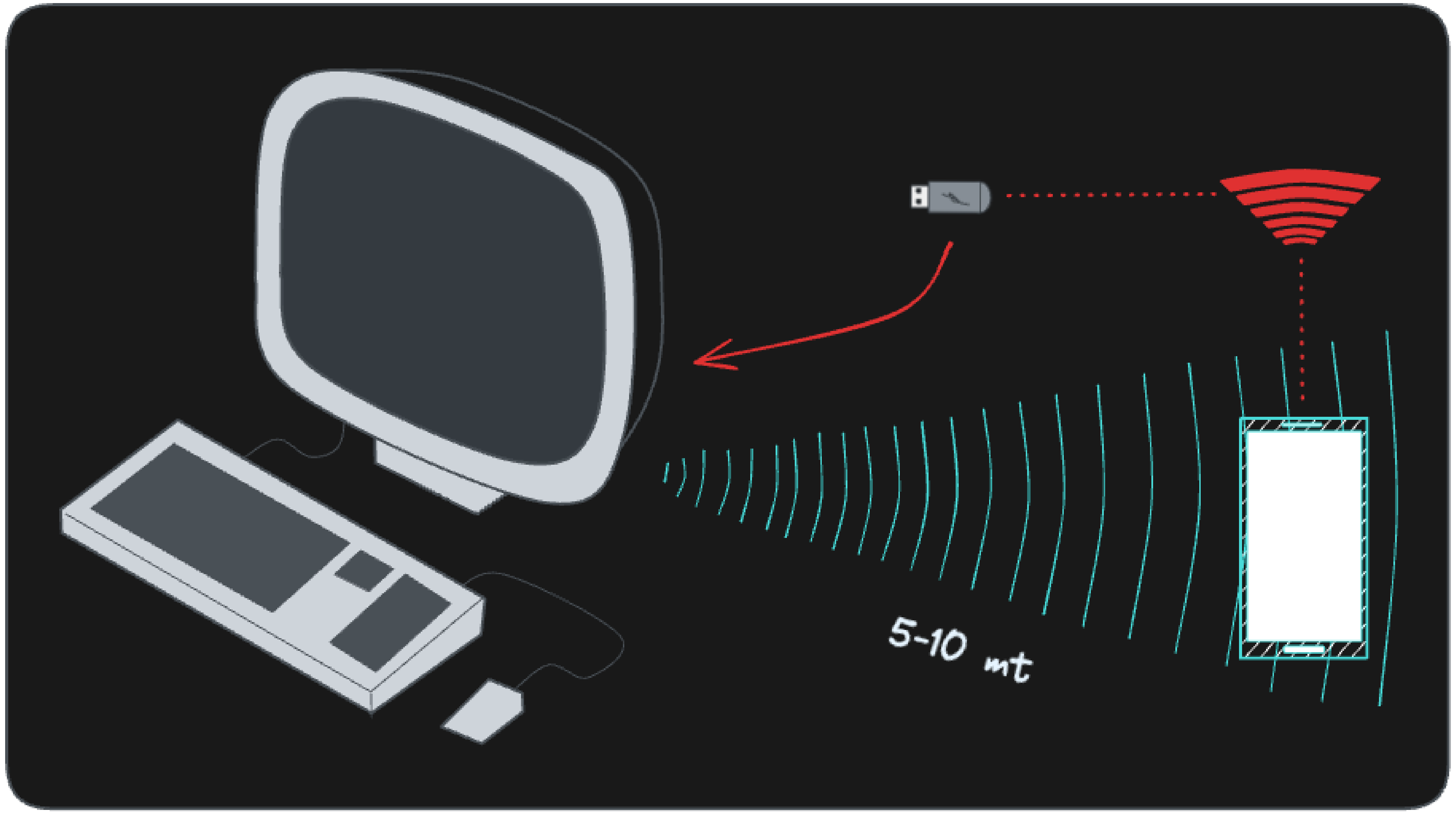
4 plan

5 execution

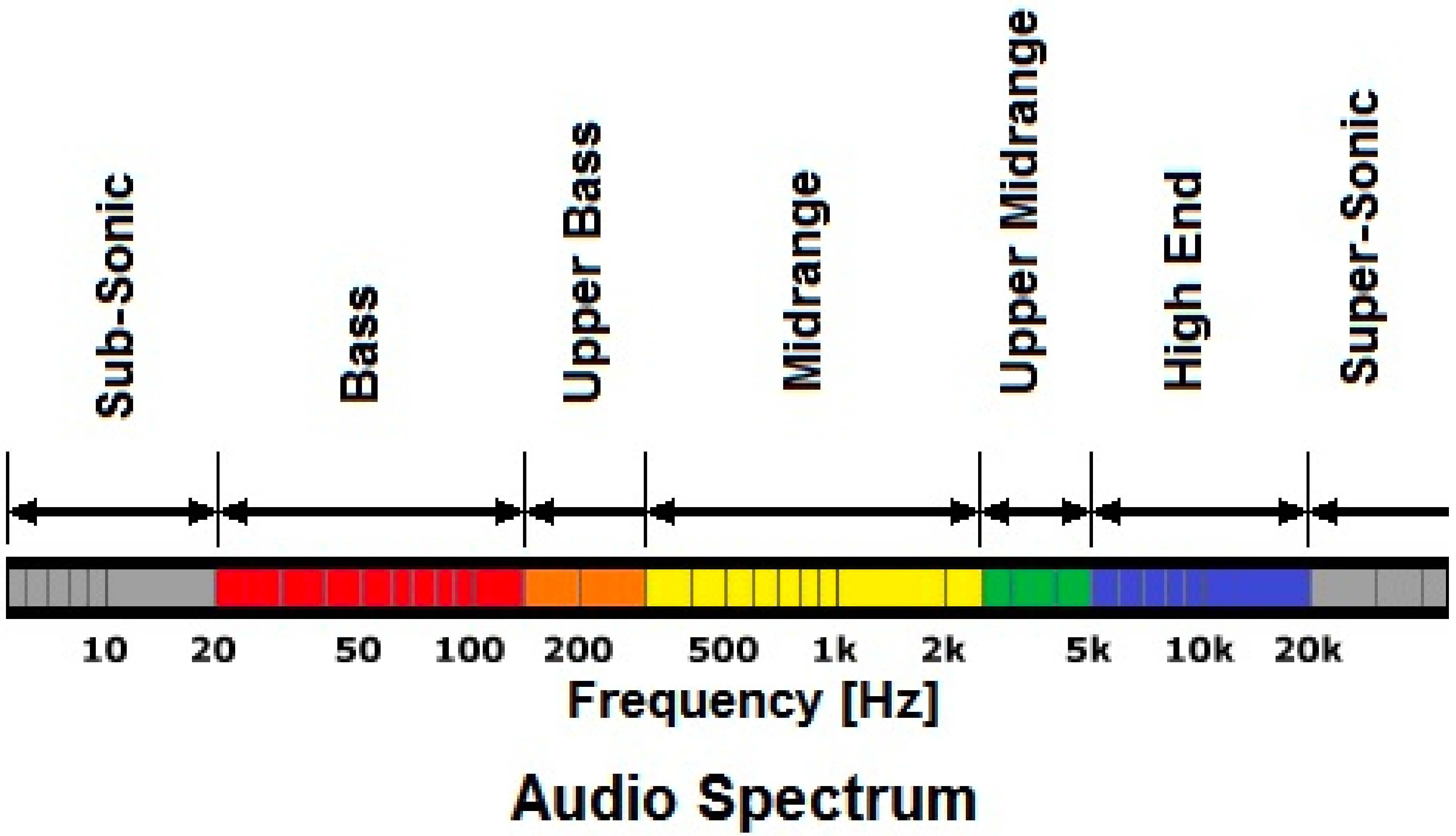
6 covering tracks

7 post-processing

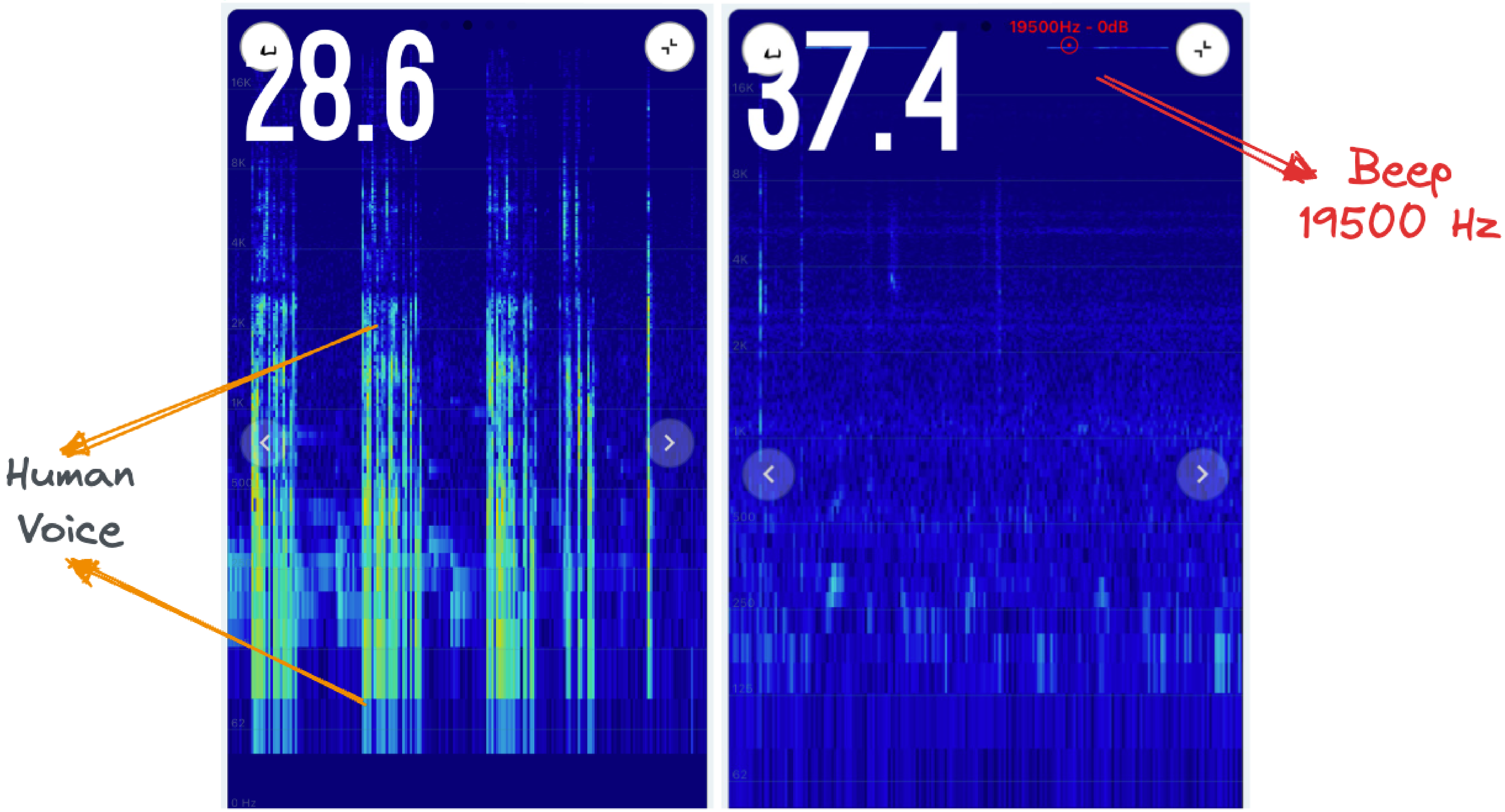
Have you heard anything?



Have you heard anything?



Have you heard anything?

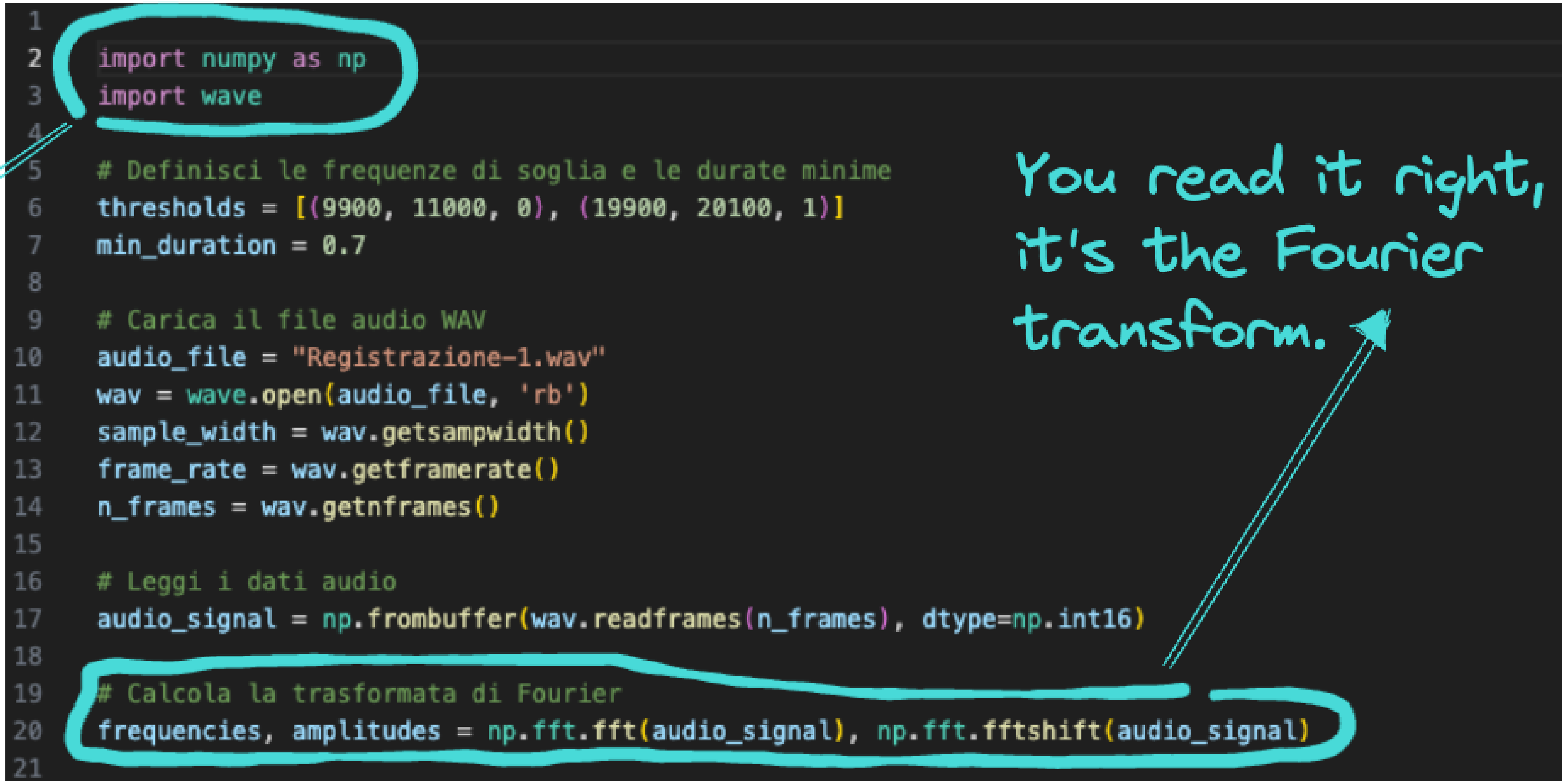


Human Voice

Beep 19500 Hz

Have you heard anything?

audio analysis



```
import numpy as np
import wave

# Definisci le frequenze di soglia e le durate minime
thresholds = [(9900, 11000, 0), (19900, 20100, 1)]
min_duration = 0.7

# Carica il file audio WAV
audio_file = "Registrazione-1.wav"
wav = wave.open(audio_file, 'rb')
sample_width = wav.getsampwidth()
frame_rate = wav.getframerate()
n_frames = wav.getnframes()

# Leggi i dati audio
audio_signal = np.frombuffer(wav.readframes(n_frames), dtype=np.int16)

# Calcola la trasformata di Fourier
frequencies, amplitudes = np.fft.fft(audio_signal), np.fft.fftshift(audio_signal)
```

You read it right, it's the Fourier transform.

Sources and references

- https://www.mdpi.com/1424-8220/23/6/3215
- https://thesecmaster.com/14-popular-air-gapped-data-exfiltration-techniques-used-to-steal-the-data/
- https://newsarchive.berkeley.edu/news/media/releases/2005/09/14_key.shtml
- https://web-assets.esetstatic.com/wls/en/papers/white-papers/eset_jumping_the_air_gap_wp.pdf
- https://ieeexplore.ieee.org/document/6999418
- https://katedavis.engr.tamu.edu/wp-content/uploads/sites/180/2022/05/2016_Q3_3.2_Metrics_Accepted.pdf
- https://ieeexplore.ieee.org/document/10197022
- https://www.bleepingcomputer.com/news/security/etherled-air-gapped-systems-leak-data-via-network-card-leds/

{Thanks}