A BRIEF HISTORY OF WHAD

(or how we ended up creating an ecosystem of opensource tools for wireless security)

Romain Cayre, Damien Cauquil

WHO ARE WE ?

Romain Cayre

- Lecturer at INSA Toulouse
- Researcher at LAAS-CNRS
- Main research thematics:
  - Wireless Security
  - Embedded systems Security
  - Internet of Things Security

Damien Cauquil

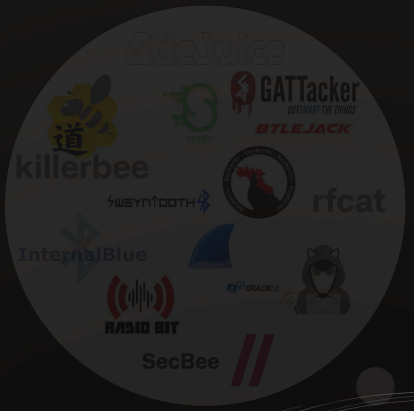
- Security Engineer at Quarkslab
- Developer of BTLEJack & BTLEJuice
- Specialized in:
  - Wireless Security
  - Hardware Security
  - Embedded Systems Security

HOW IT ALL BEGAN

1 Internship at LAAS-CNRS

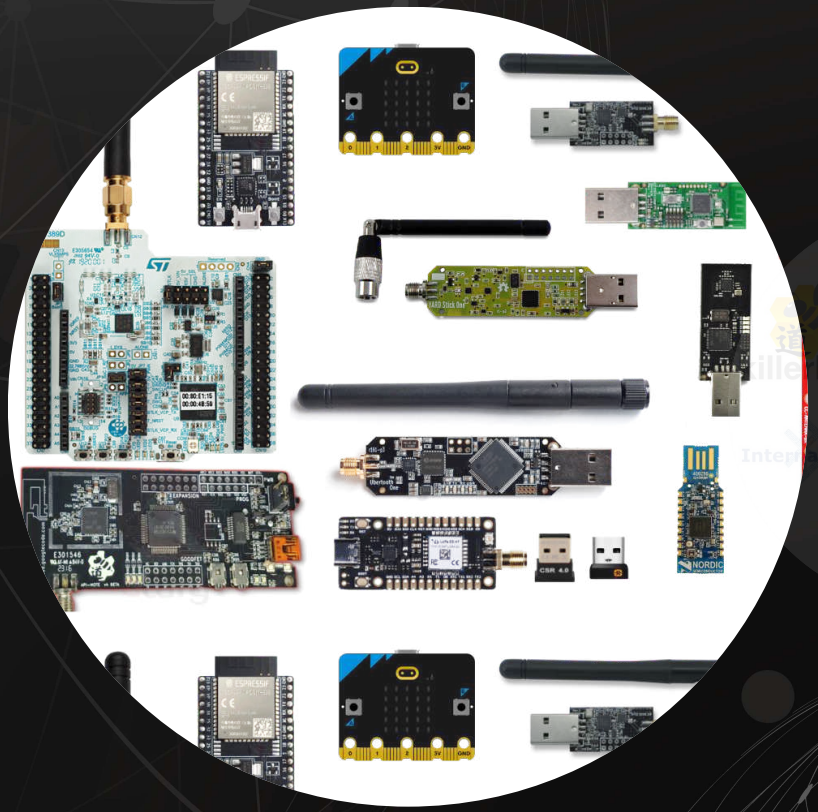
Goal: Evaluation of a protocol agnostic IDS for IoT against existing threats.

My task is to implement a set of realistic and diversified wireless attacks to attack IoT wireless protocols.



Multiple heterogeneous protocols

- Fast and chaotic deployment
- Keen competition to win the IoT market
- Fast evolution with new features every year



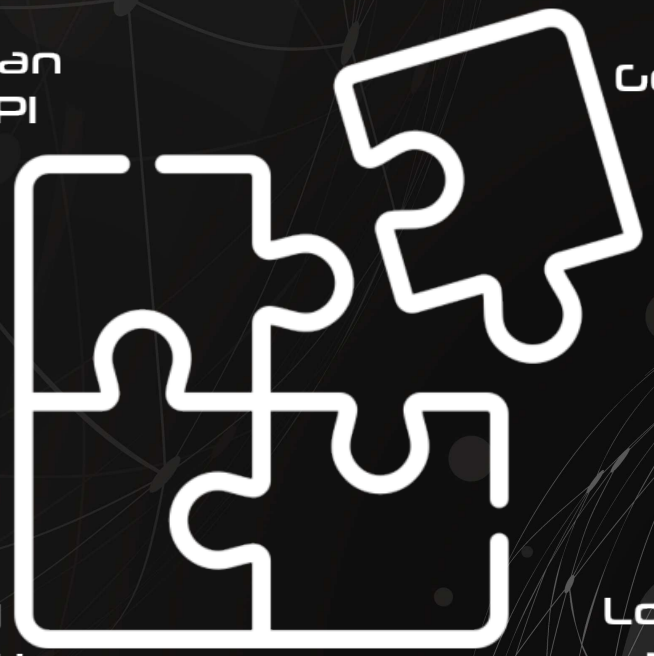
Various offensive hardware

- Increasing in number and variety
- Not suited for wireless security without offensive firmwares
- Provide heterogeneous capabilities & features

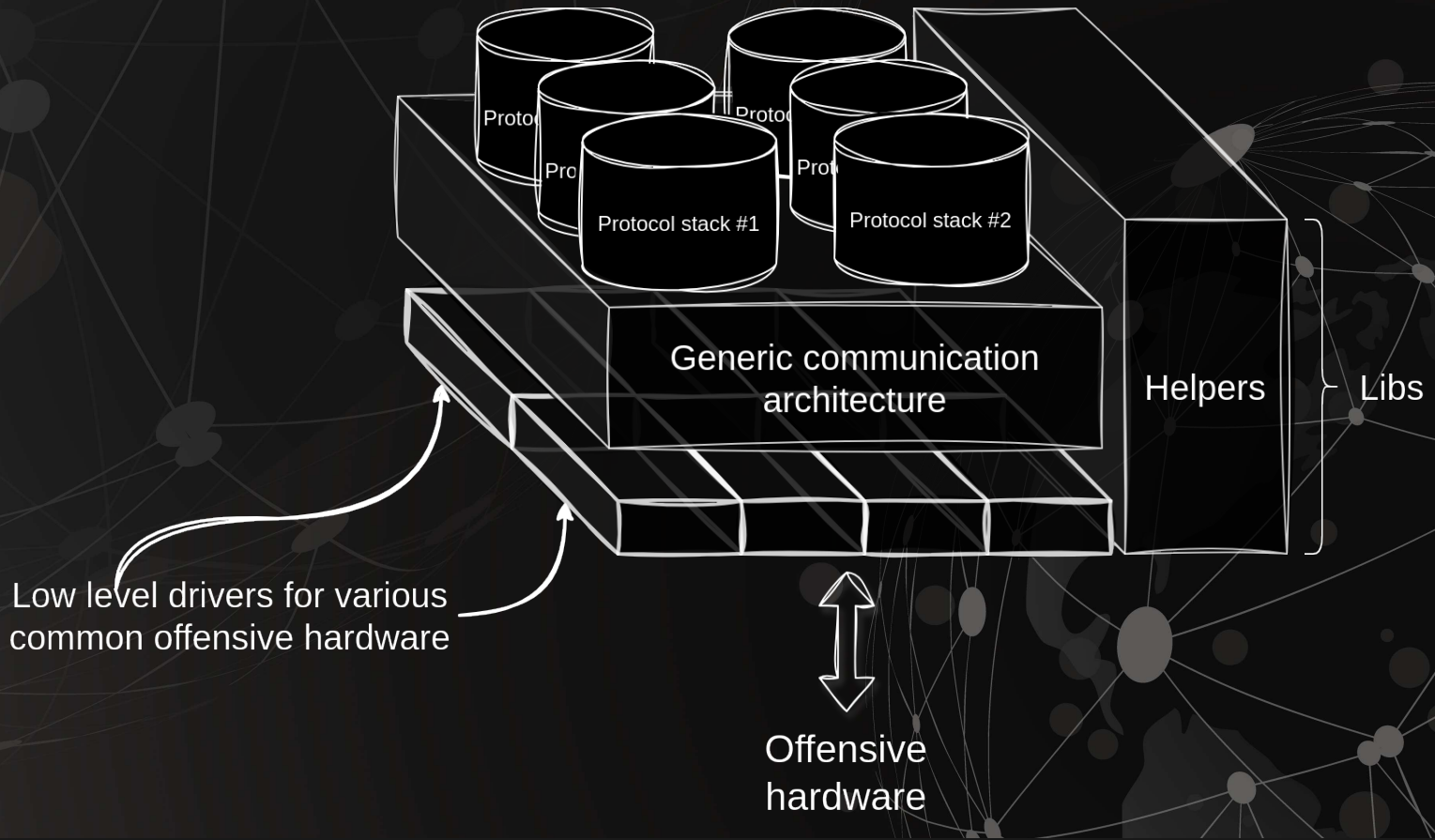
Various offensive software

- Increasing in number and variety & everyone reinvent the wheel
- Mutually incompatible in term of API, file formats & hardware support
- Use of high level libraries not designed for offensive security

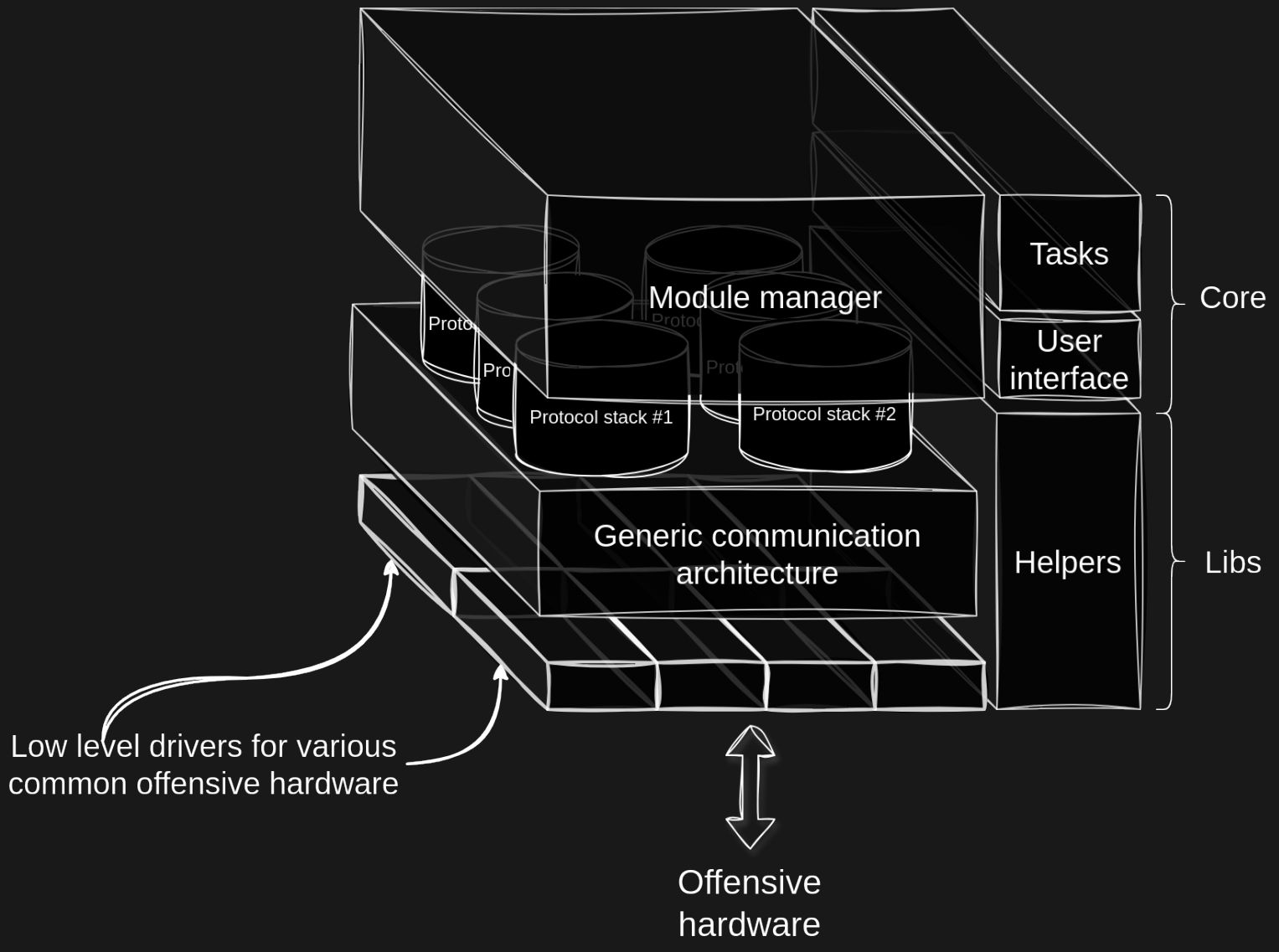
MIRAGE: MAIN GUIDELINES



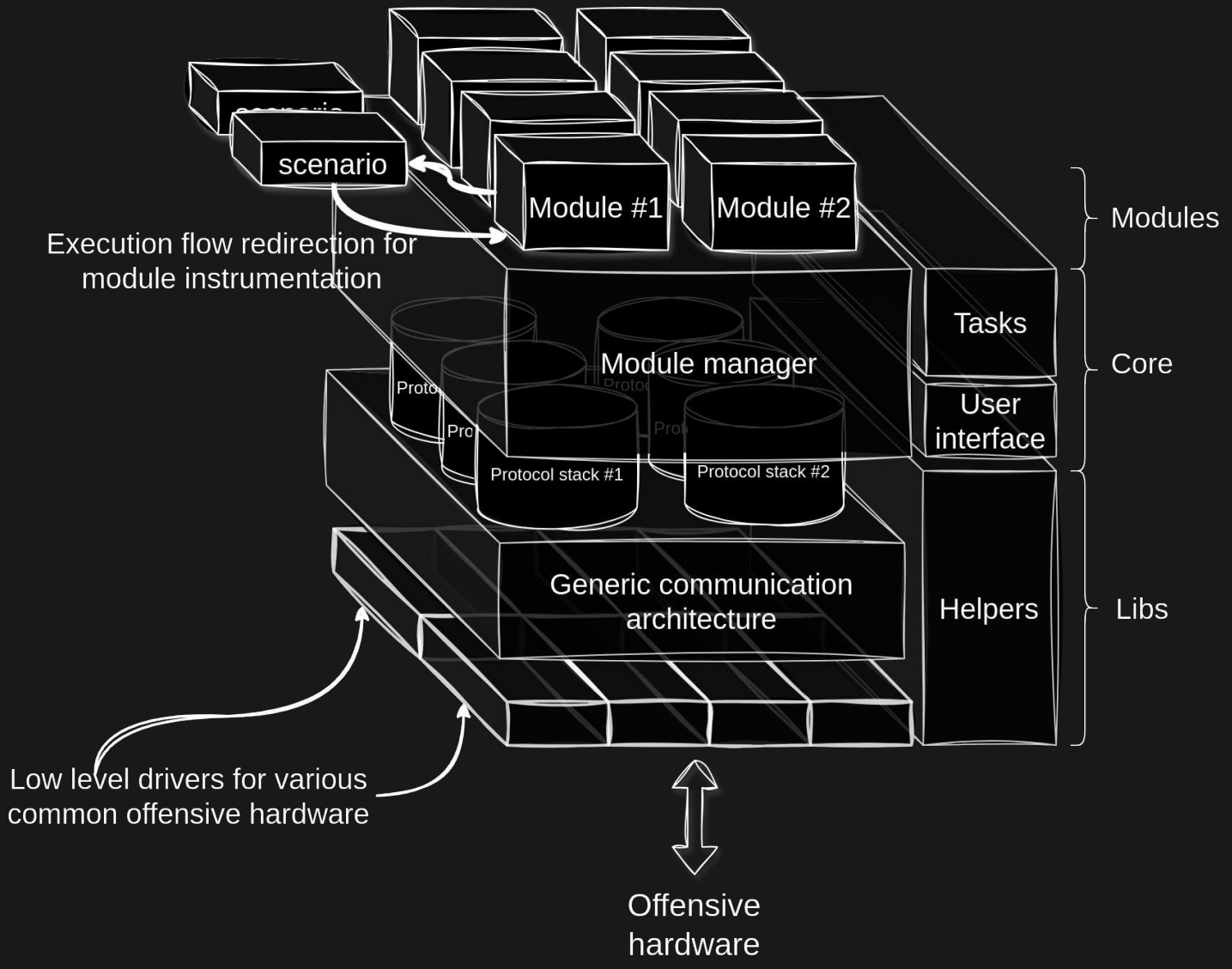
MIRAGE: ARCHITECTURE



MIRAGE: ARCHITECTURE

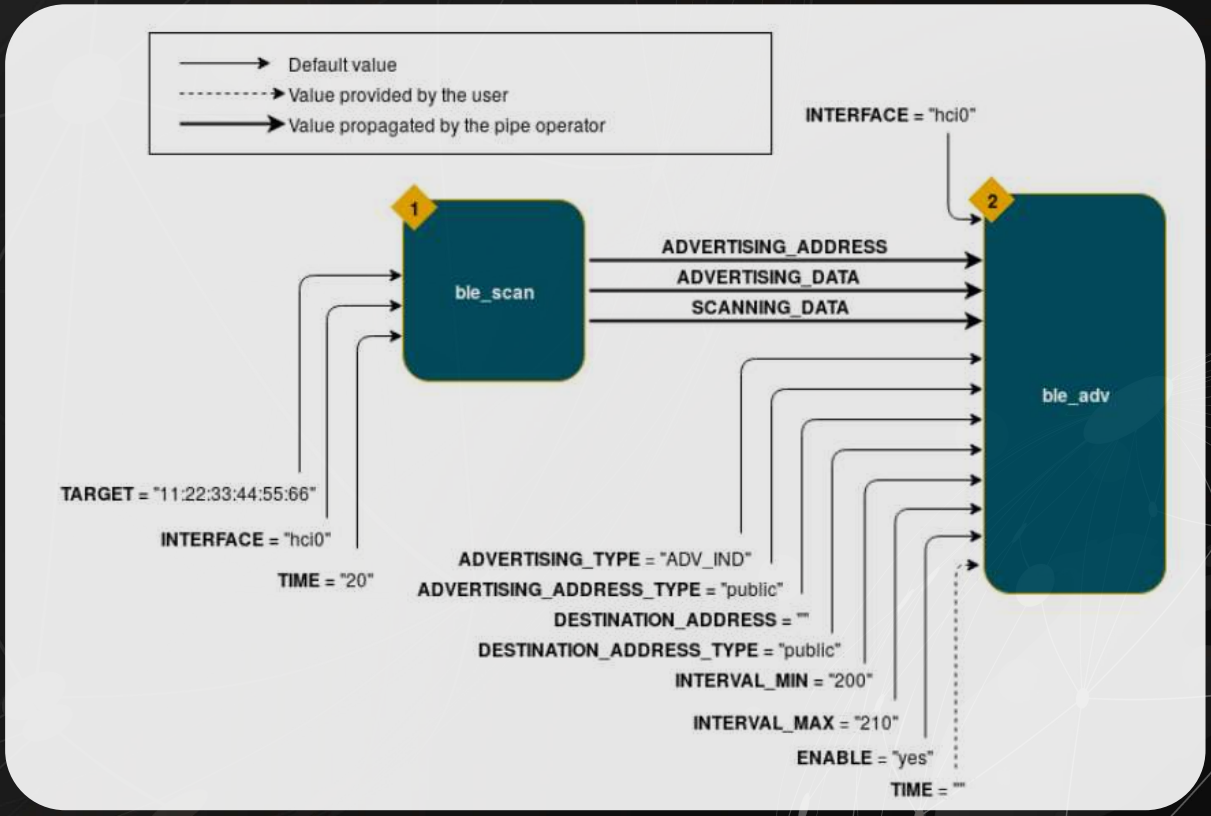

Tasks
Module manager
User interface
Core
Protocol stack #1
Protocol stack #2
Generic communication architecture
Helpers
Libs
Low level drivers for various common offensive hardware
Offensive hardware

MIRAGE: ARCHITECTURE


scenario
Module #1
Module #2
Execution flow redirection for module instrumentation
Modules
Tasks
Module manager
Core
User interface
Protocol stack #1
Protocol stack #2
Generic communication architecture
Helpers
Libs
Low level drivers for various common offensive hardware
Offensive hardware

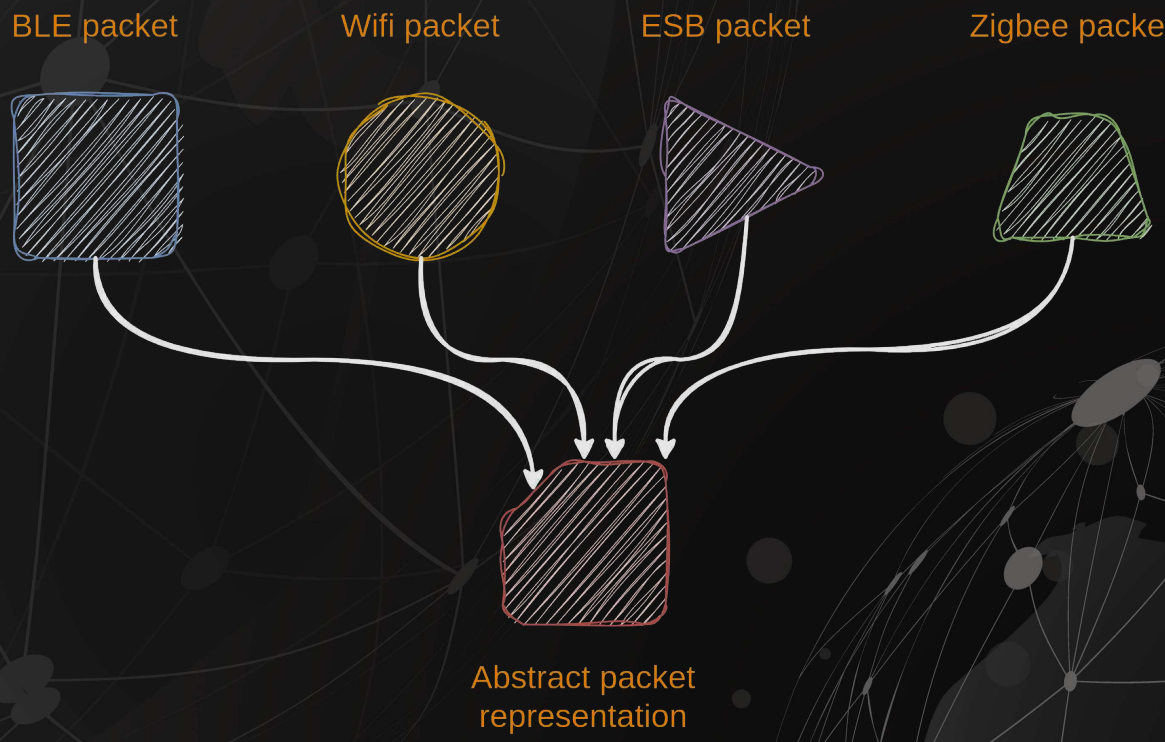
MIRAGE: A FEW INTERESTING IDEAS

```
sudo mirage "ble_scan | ble_adv"
```



- Mirage is splitted into modules
- Every module can perform a simple operation or implement a basic attack
- Modules can be combined using a chaining operator (similar to UNIX pipe)

MIRAGE: A FEW INTERESTING IDEAS

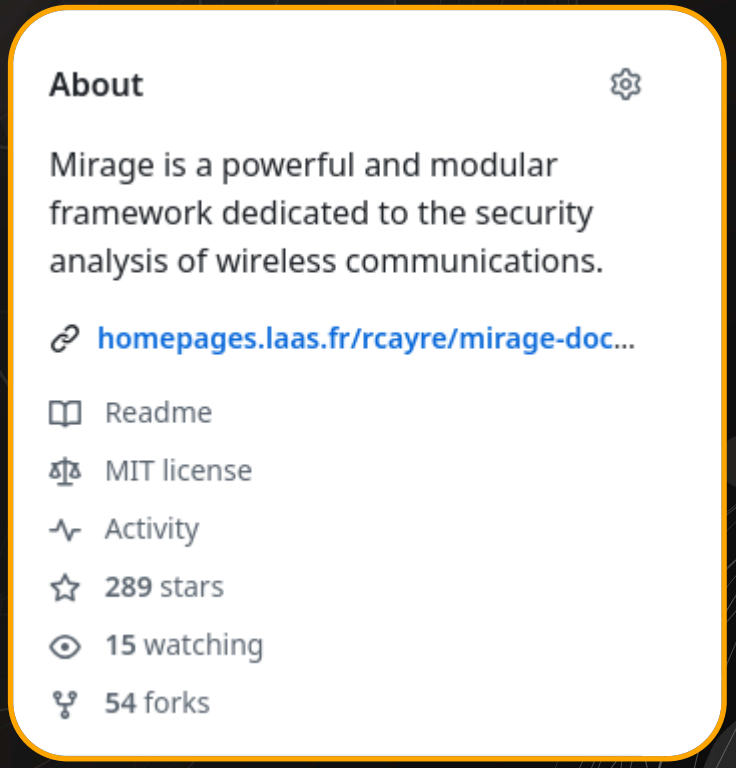


- Every packets are harmonised into a single abstract representation
- Mirage provides a unified API to transmit and receive these packets

MIRAGE: LIMITATIONS

- Native UNIX pipe not directly used → complex command lines
- Abstract packet representation lacks flexibility → tons of translation layers !
- Concepts specific to Mirage (scenarios, tasks, modules) → high learning curve
- Components tightly coupled → hard to use outside of Mirage
- **Does not change the way we write offensive firmwares → new drivers to write**

DESPITE THESE LIMITATIONS...



Clearly, the tool has raised interest in the community !

😀

1 Internship at LAAS-CNRS

Goal: Evaluation of a protocol agnostic IDS for IoT against existing threats.

My task is to implement a set of realistic and diversified wireless attacks to attack IoT wireless protocols.

2 Mirage public releases

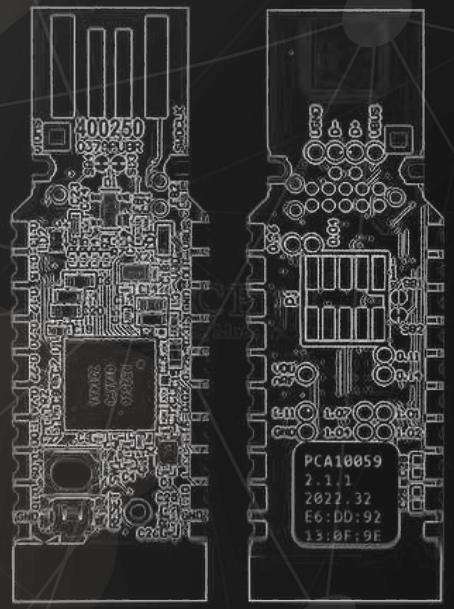
First release at SSTIC 2019 (v1.0)
BLE support only

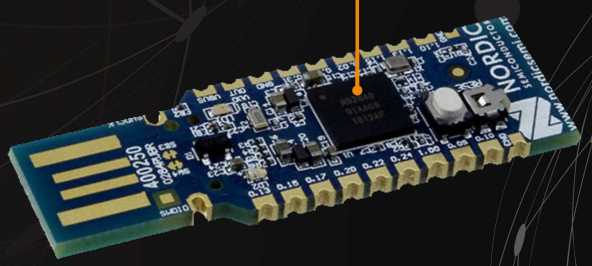
Second release at ISSRE 2019 (v1.1)
support for BLE, WiFi, ZigBee, ESB, Mosart and basic InfraRed protocols

DIVING INTO OFFENSIVE FIRMWARES DEVELOPMENT


1
Internship at LAAS-CNRS
Goal: Evaluation of a protocol agnostic IDS for IoT against existing threats.
My task is to implement a set of realistic and diversified wireless attacks to attack IoT wireless protocols.
2
Mirage public releases
First release at SSTIC 2019 (v1.0)
BLE support only
Second release at ISSRE 2019 (v1.1)
support for BLE, WiFi, ZigBee, ESB, Mosart and basic InfraRed protocols
3
Low level attacks
During my PhD & postdoc research work, I work on complex attacks requiring very low level control
Damien Cauquil & I start working together on the implementation of offensive firmwares for nRF52 & ESP32 SoCs.

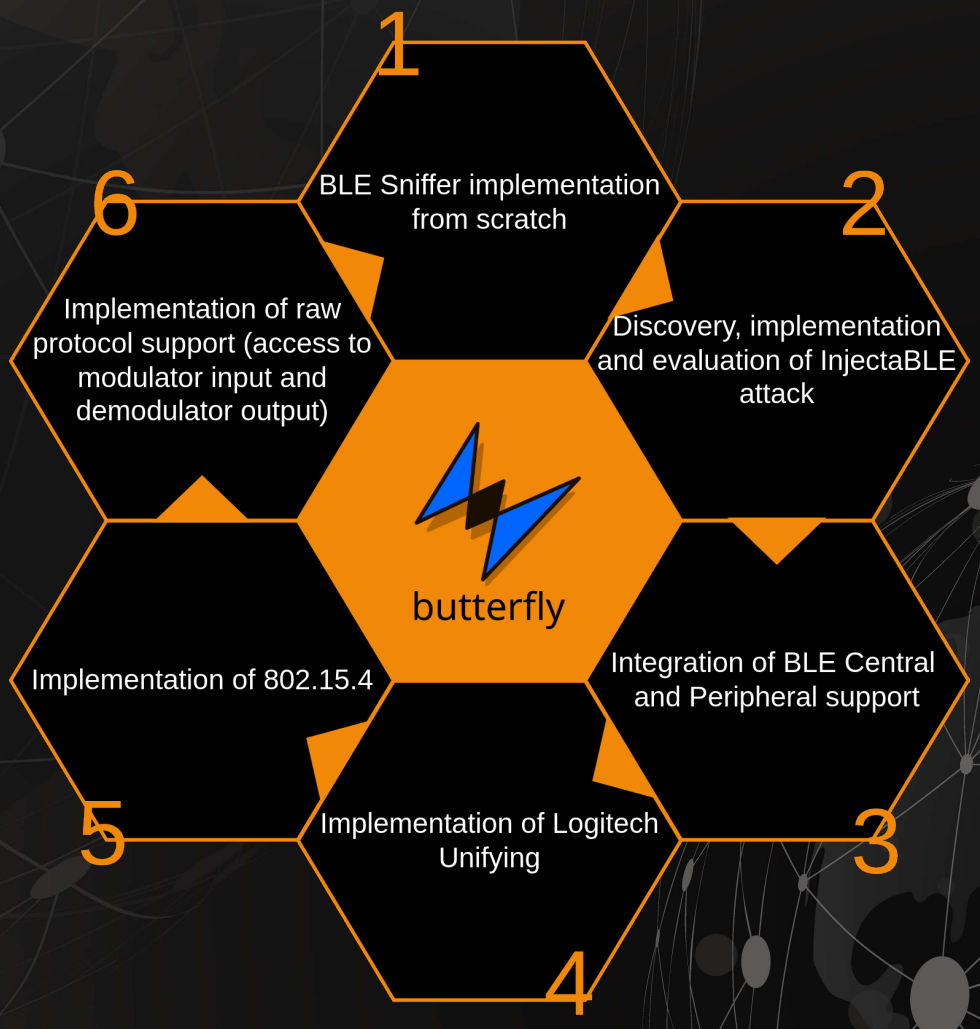
MEET NRF52840 DONGLE ...



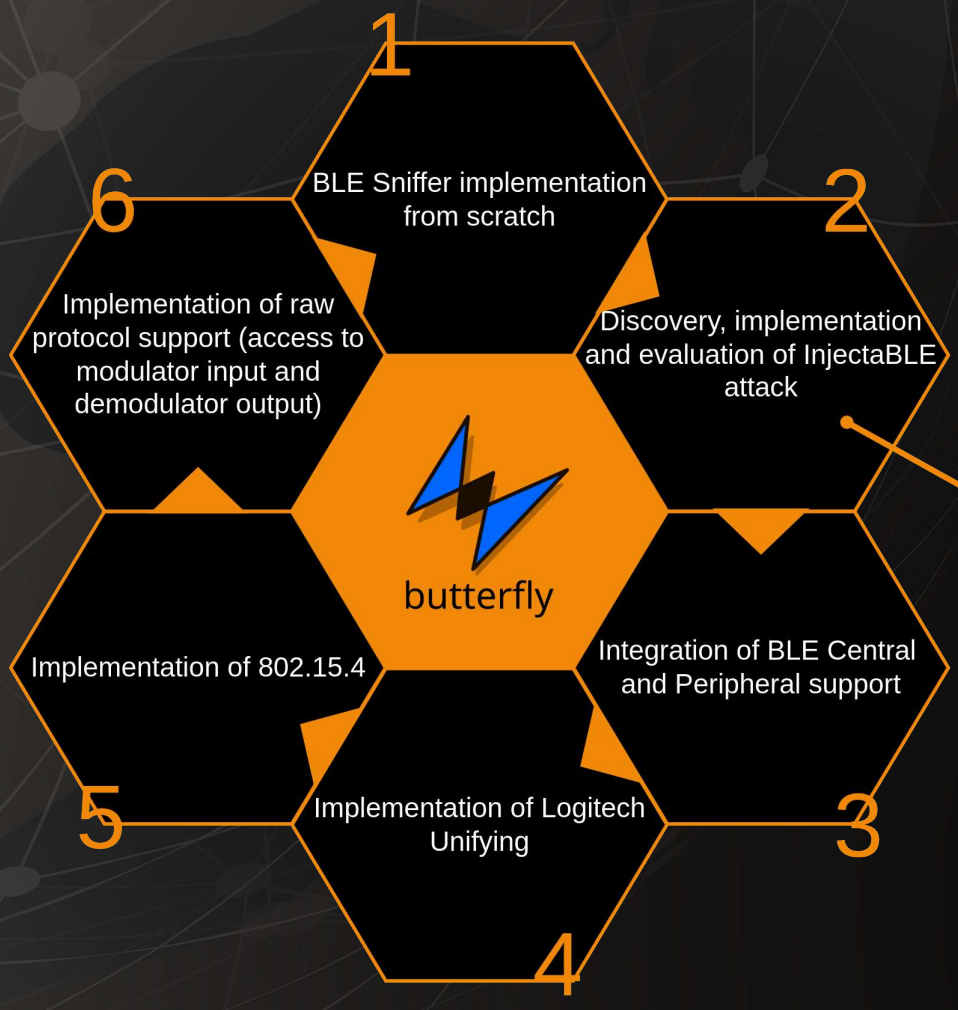


- **Nordic Semiconductor description:**
  - 64 MHz Cortex-M4 with FPU
  - 1 MB Flash, 256 KB RAM
  - 2.4 GHz Transceiver
  - 2 Mbps, 1 Mbps, Long Range
  - Bluetooth Low Energy, Bluetooth mesh
  - ANT, 802.15.4, Thread, Zigbee
- **My description:**
  - Super cheap (~10€)
  - Hacker friendly (Travis Goodspeed's nRF24 promiscuous mode hack still works !)
  - Multi-protocol & fast RX/TX switch time

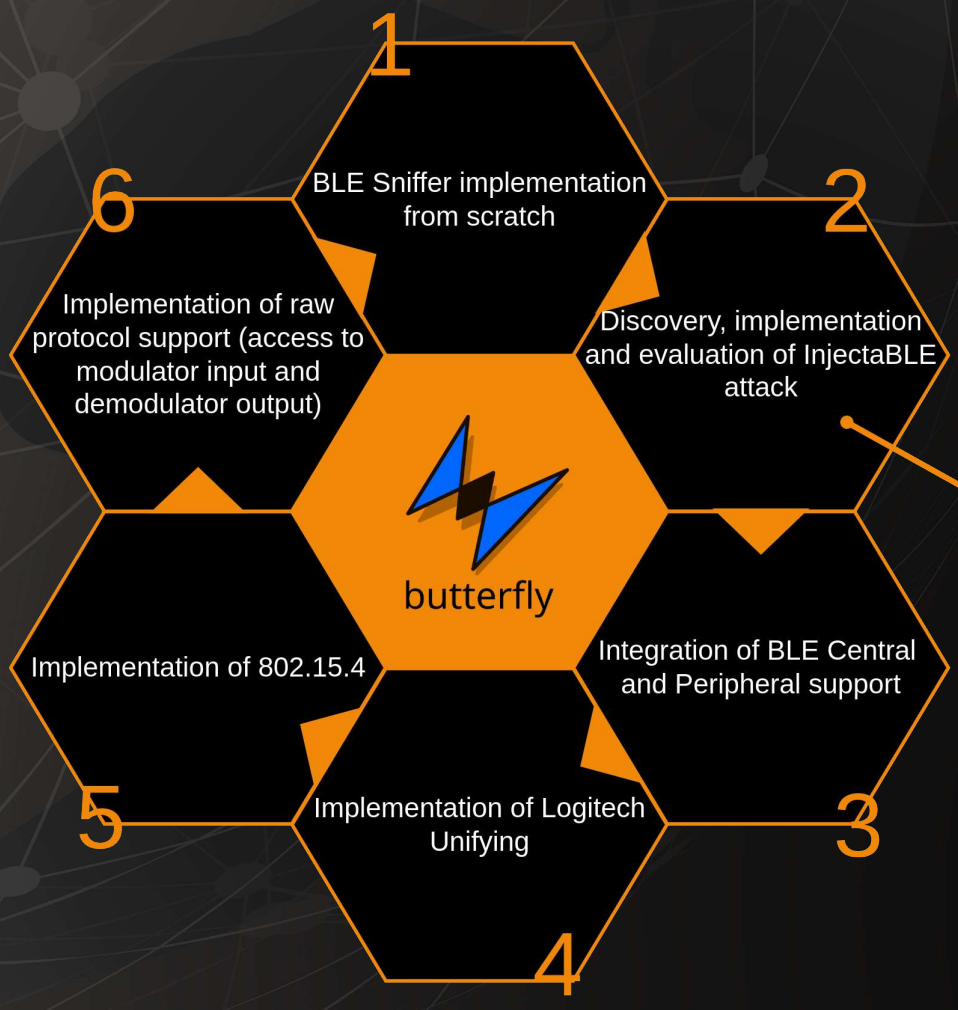
BUTTERFLY: OFFENSIVE FIRMWARE FOR NRF52

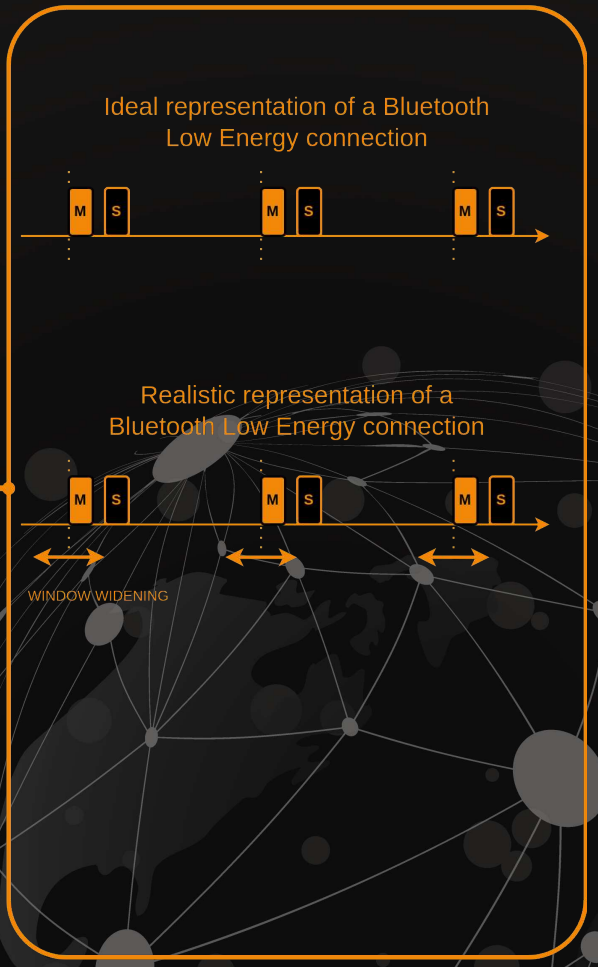


BUTTERFLY: INJECTABLE ATTACK

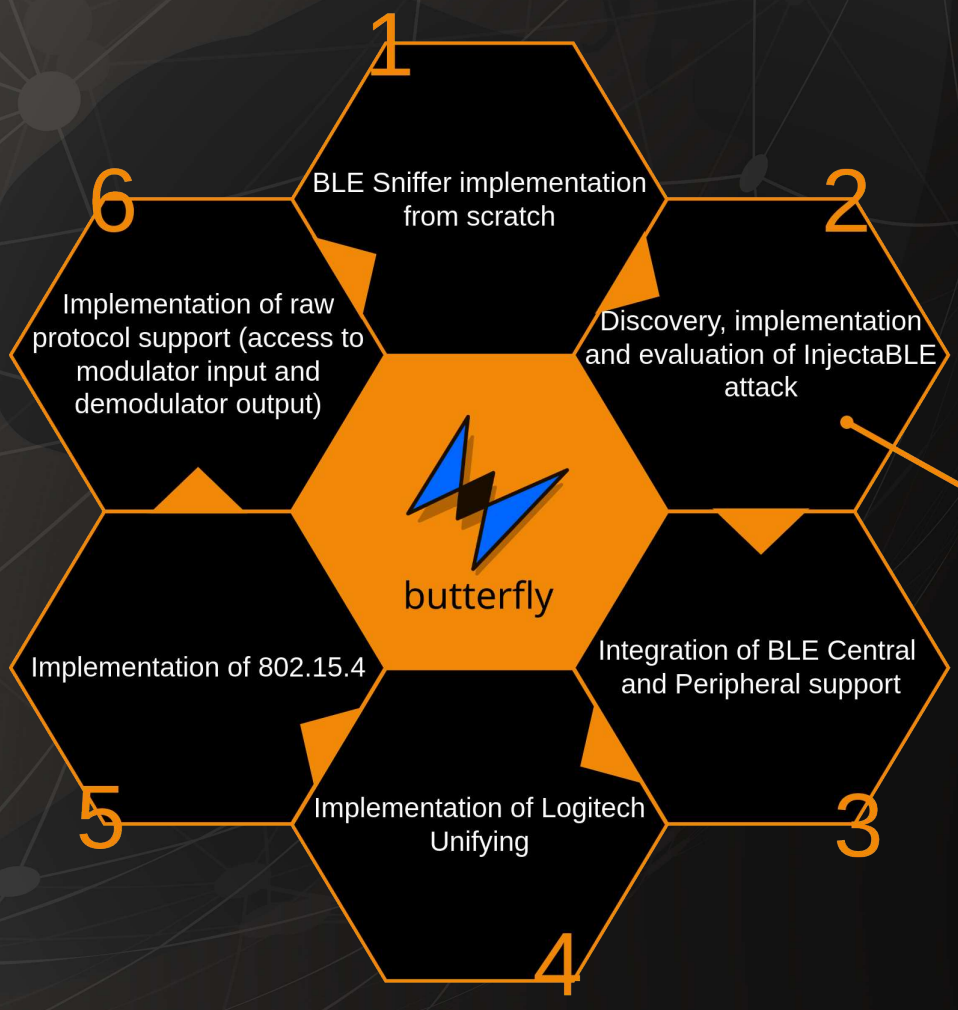


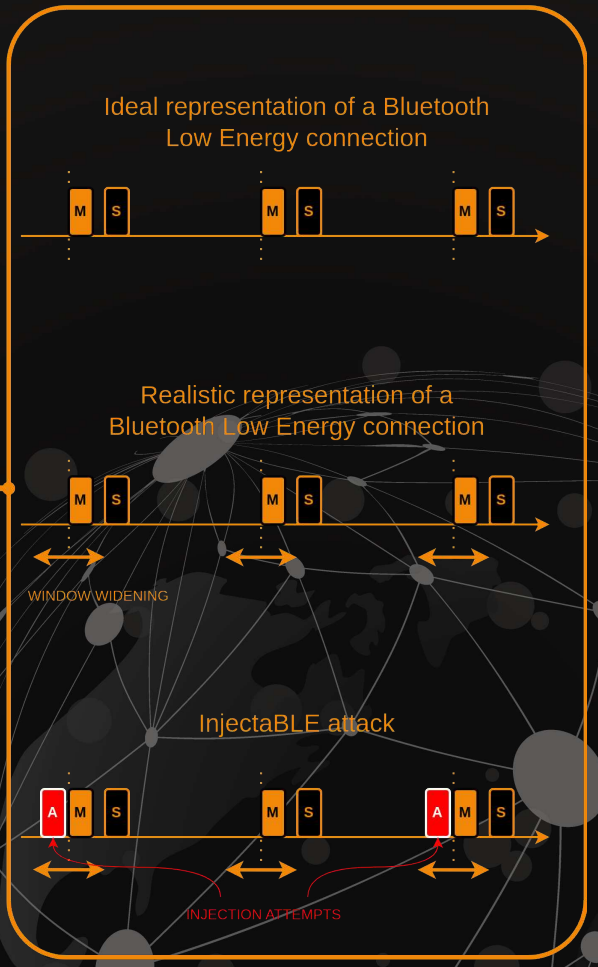
BUTTERFLY: INJECTABLE ATTACK



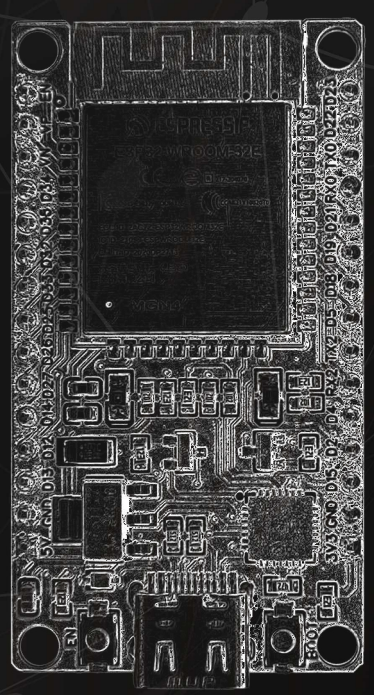


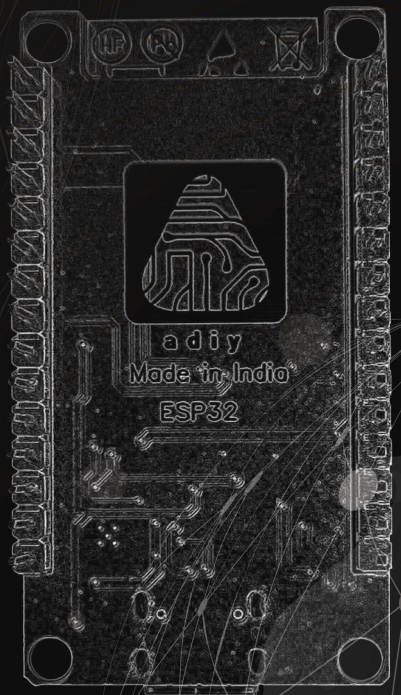
BUTTERFLY: INJECTABLE ATTACK



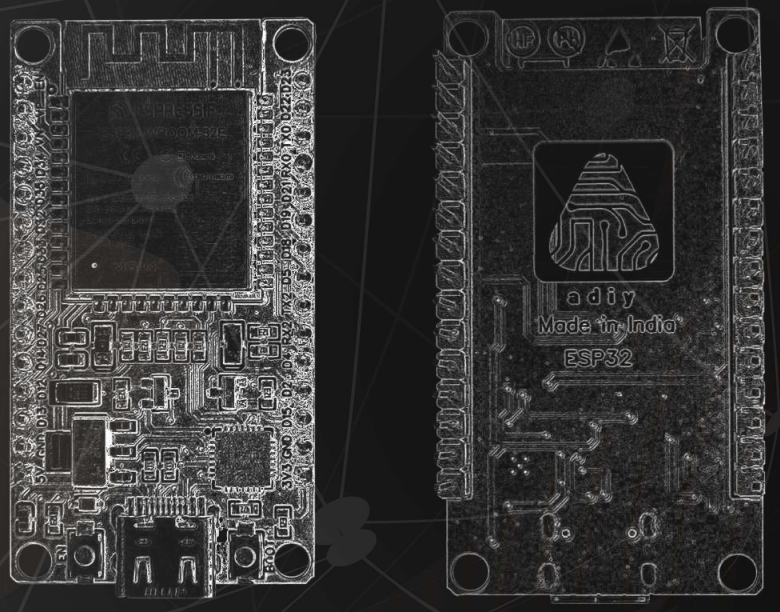


ESPWN32





ESPWN32



TYPICAL BLE STACK

HOST

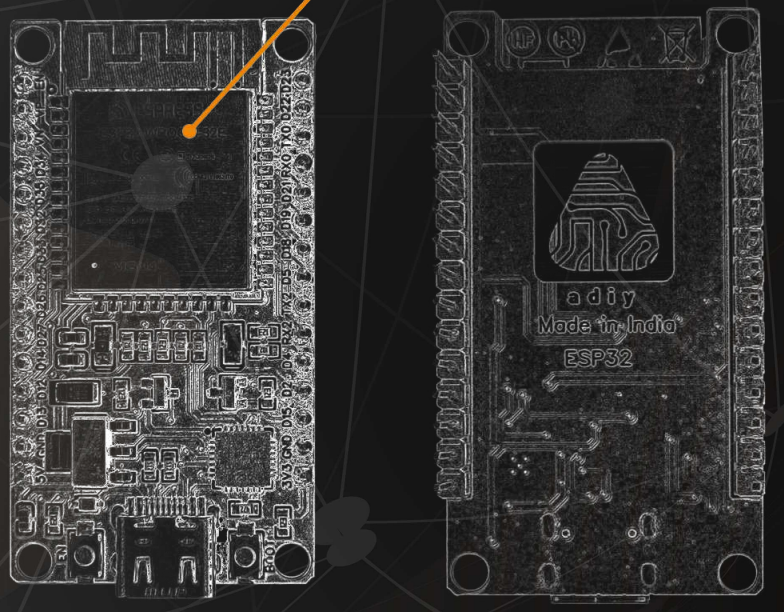
Applicative & Security Layers

HOST / CONTROLLER INTERFACE (HCI)

CONTROLLER

Physical & Link Layers

ESPWN32



ESPRESSIF IMPLEMENTATION

USER APPLICATION

NIMBLE (HOST)

VIRTUAL HOST CONTROLLER INTERFACE

INTERNAL ROM (SOFTWARE)

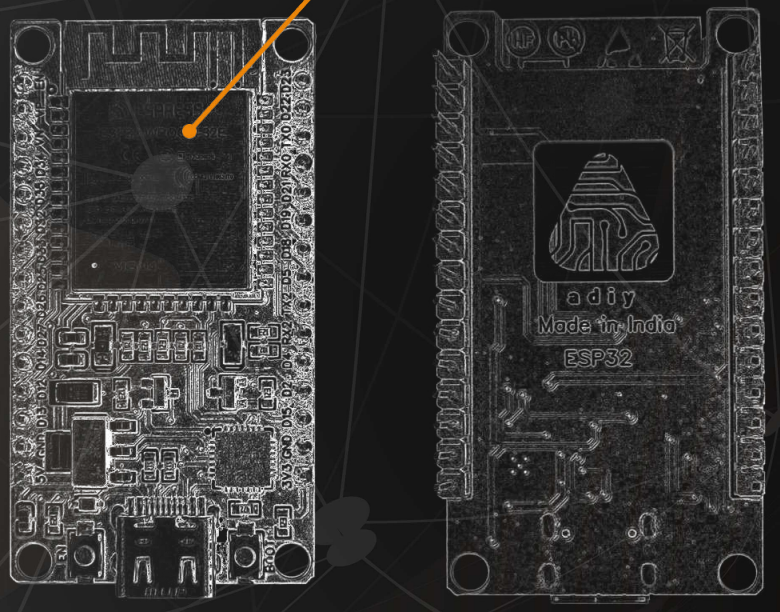
Reception()

Transmission()

SetChannel()

BLE CORE (RF HARDWARE)

ESPWN32



ESPRESSIF IMPLEMENTATION

USER APPLICATION

PERFORM AN ACTION

NIMBLE (HOST)

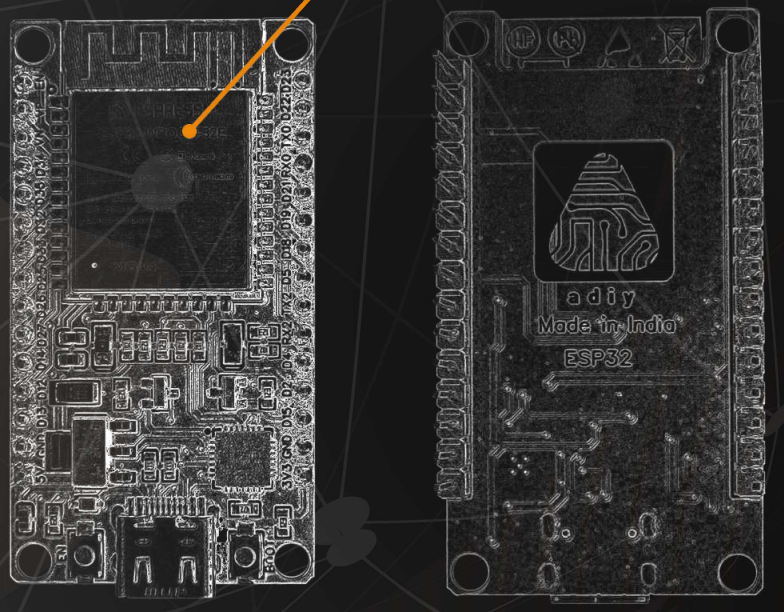
VIRTUAL HOST CONTROLLER INTERFACE

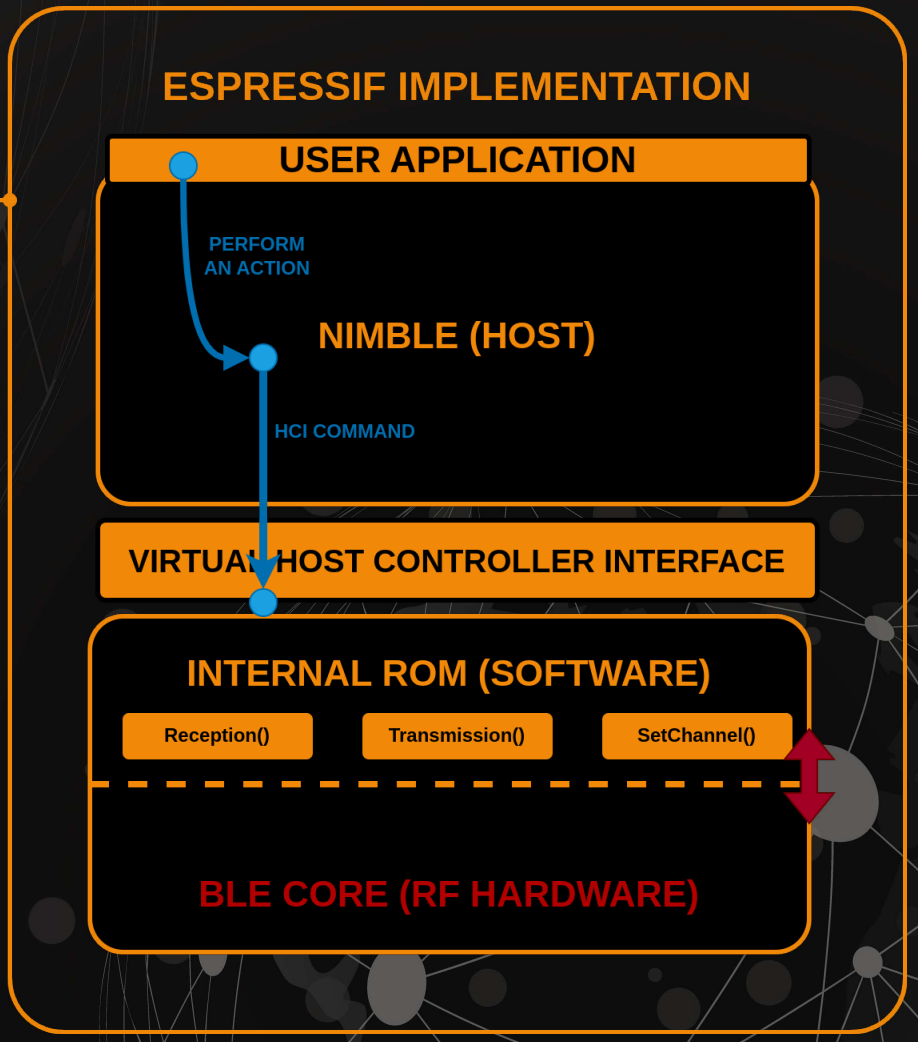
INTERNAL ROM (SOFTWARE)

Reception() Transmission() SetChannel()

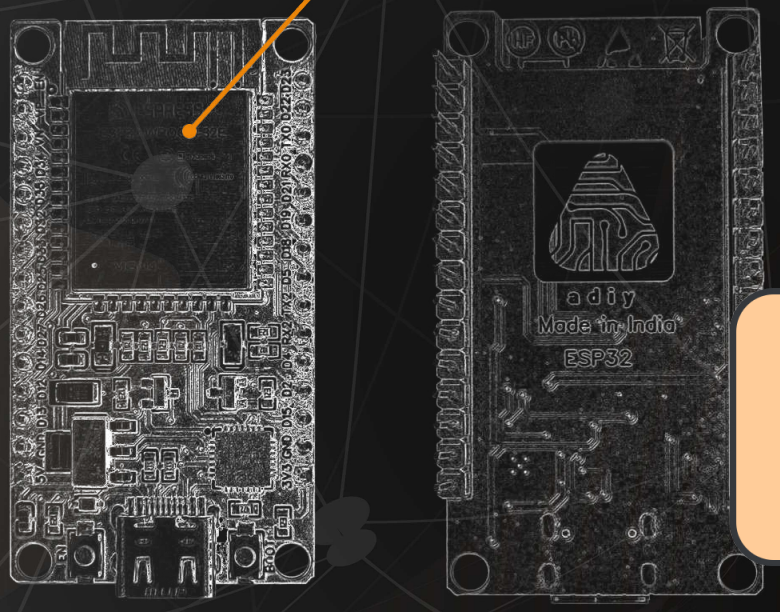
BLE CORE (RF HARDWARE)

ESPWN32


adiy
Made in India
ESP32


ESPRESSIF IMPLEMENTATION
USER APPLICATION
PERFORM
AN ACTION
NIMBLE (HOST)
HCI COMMAND
VIRTUAL HOST CONTROLLER INTERFACE
INTERNAL ROM (SOFTWARE)
Reception()
Transmission()
SetChannel()
BLE CORE (RF HARDWARE)

ESPWN32



ESPRESSIF IMPLEMENTATION

USER APPLICATION

PERFORM AN ACTION

NIMBLE (HOST)

HCI COMMAND

VIRTUAL HOST CONTROLLER INTERFACE

RAM

INDIRECT CALL

Reception

Transmission

...

INTERNAL ROM (SOFTWARE)

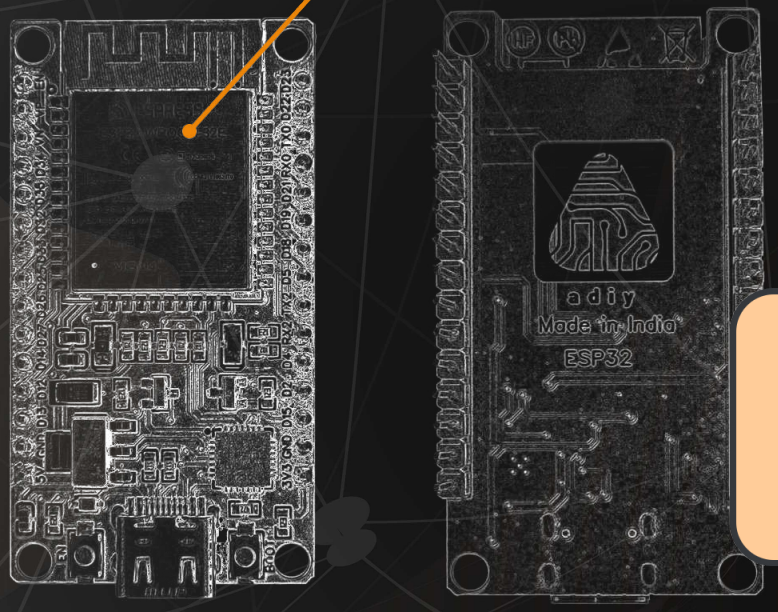
Reception()

Transmission()

SetChannel()

BLE CORE (RF HARDWARE)

ESPWN32



ESPRESSIF IMPLEMENTATION

USER APPLICATION

PERFORM AN ACTION

NIMBLE (HOST)

HCI COMMAND

VIRTUAL HOST CONTROLLER INTERFACE

RAM

INDIRECT CALL

Reception

Transmission

...

INTERNAL ROM (SOFTWARE)

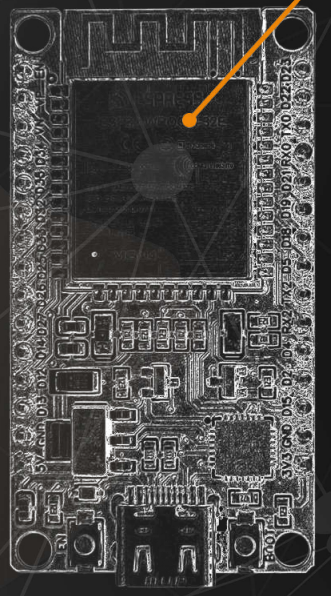
Reception()

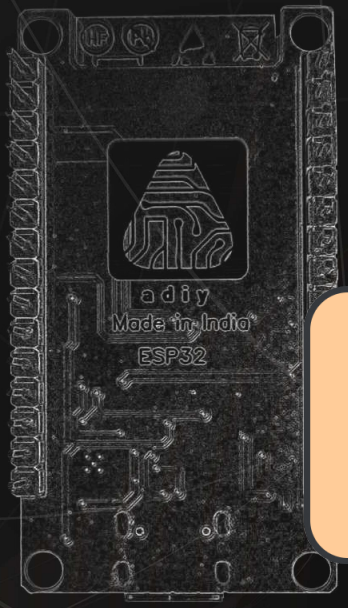
Transmission()

SetChannel()

BLE CORE (RF HARDWARE)

ESPWN32





HOOKING STRATEGY

HookRX()

USER APPLICATION

HookTX()

NIMBLE (HOST)

RAM

HookRX

HookTX

...

VIRTUAL HOST CONTROLLER INTERFACE

INTERNAL ROM (SOFTWARE)

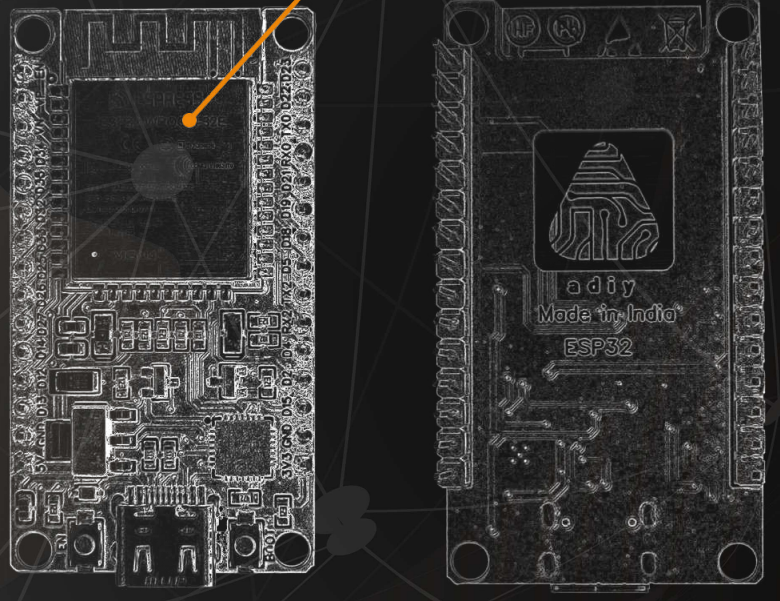
Reception()

Transmission()

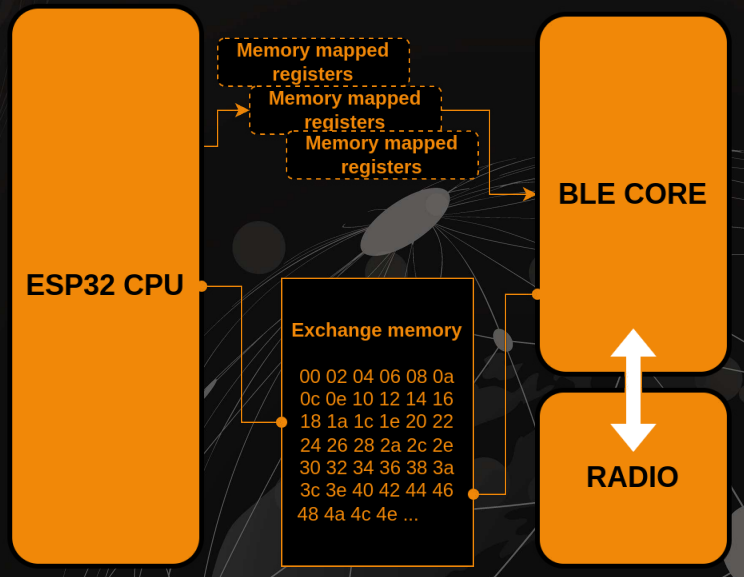
SetChannel()

BLE CORE (RF HARDWARE)

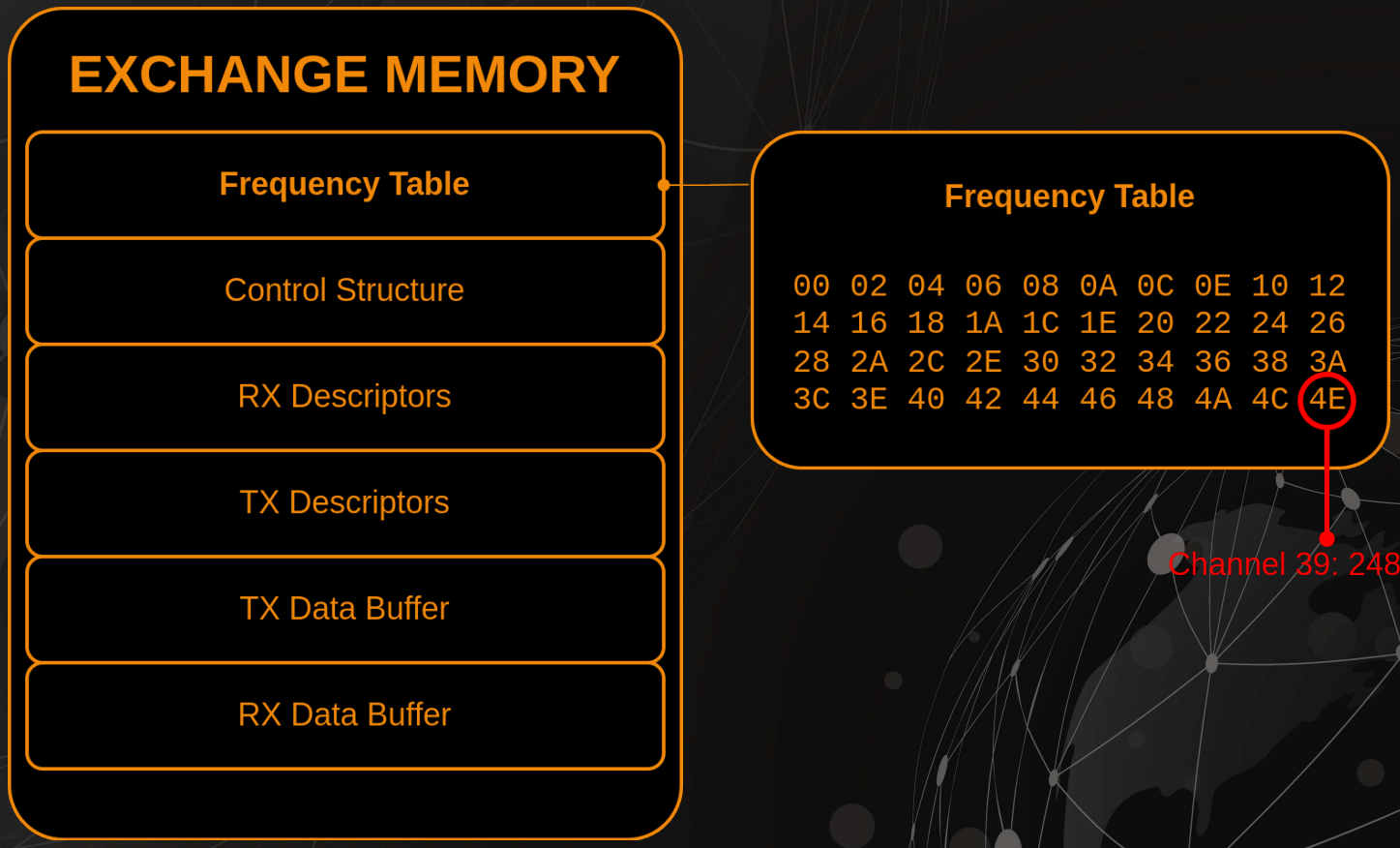
ESPWN32



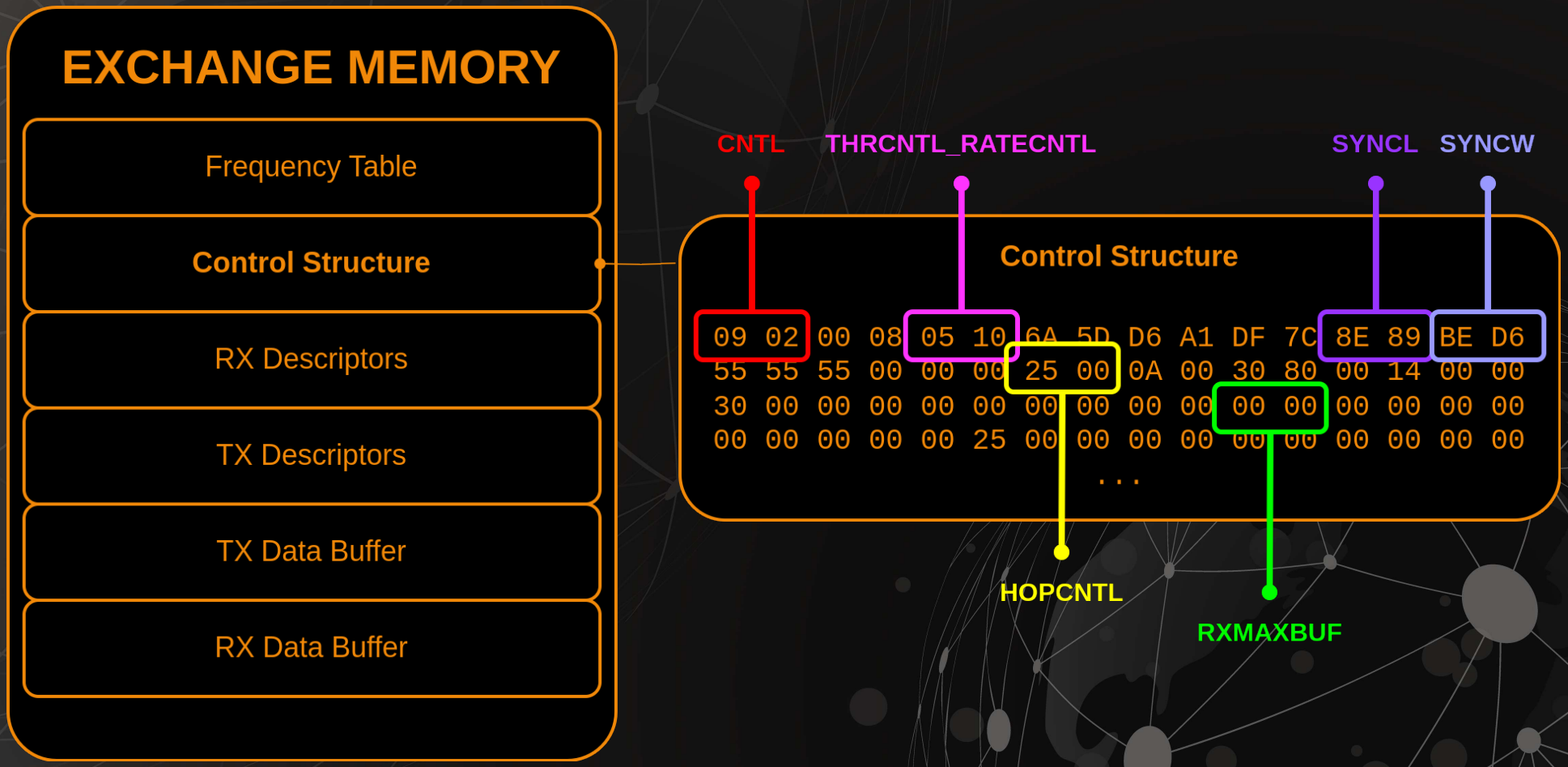
DIVERTING BLE CORE



ESPWN32


EXCHANGE MEMORY
Frequency Table
Control Structure
RX Descriptors
TX Descriptors
TX Data Buffer
RX Data Buffer
Frequency Table
00 02 04 06 08 0A 0C 0E 10 12
14 16 18 1A 1C 1E 20 22 24 26
28 2A 2C 2E 30 32 34 36 38 3A
3C 3E 40 42 44 46 48 4A 4C 4E
Channel 39: 2480MHz

ESPWN32

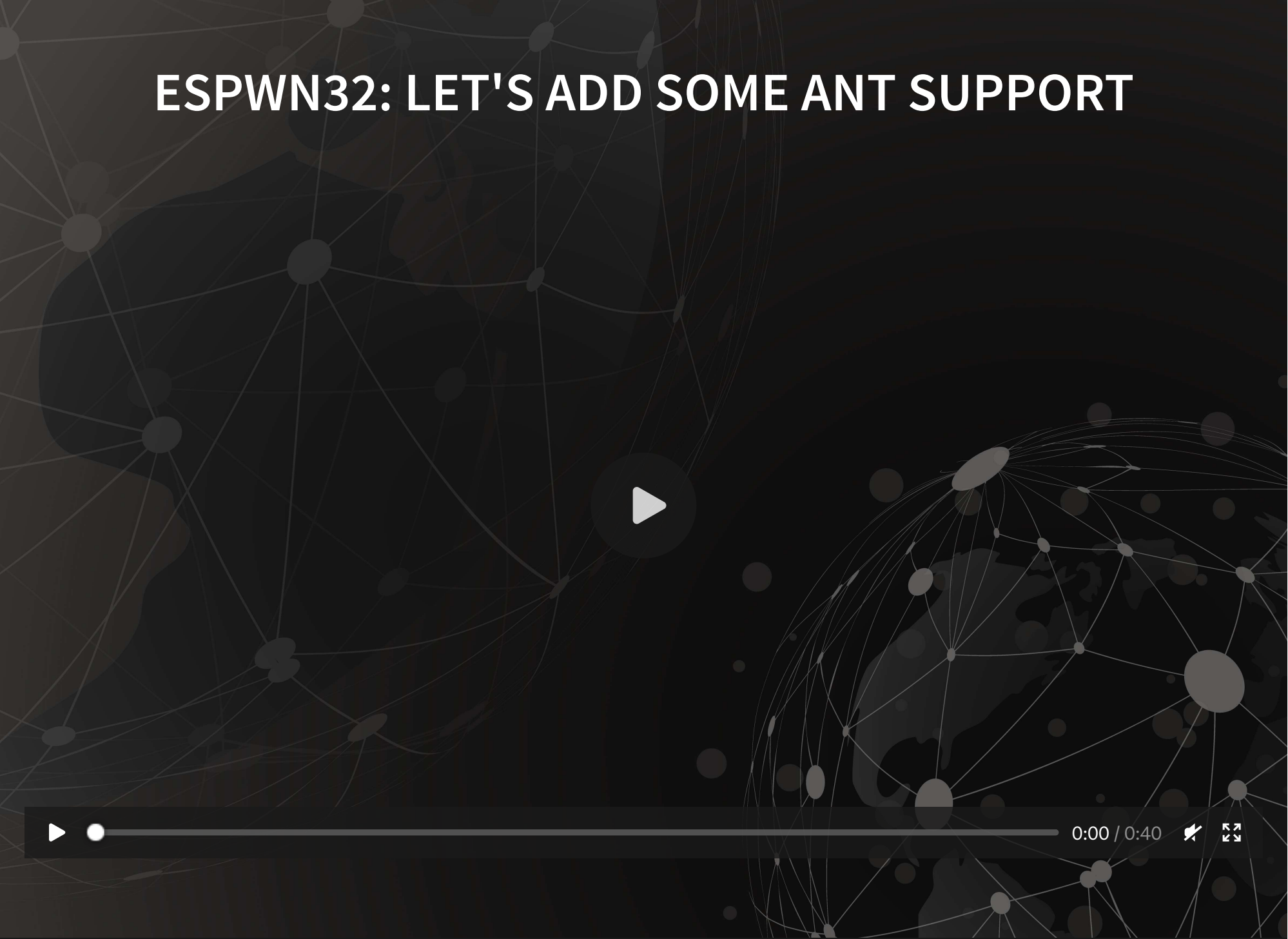

EXCHANGE MEMORY
Frequency Table
Control Structure
RX Descriptors
TX Descriptors
TX Data Buffer
RX Data Buffer
CNTL
THRCNTL_RATECNTL
SYNCL
SYNCW
Control Structure
09 02 00 08 05 10 6A 5D D6 A1 DF 7C 8E 89 BE D6
55 55 55 00 00 00 25 00 0A 00 30 80 00 14 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 25 00 00 00 00 00 00 00 00 00 00
...
HOPCNTL
RXMAXBUF

LET'S SUMMARIZE

- You can hook every low level function
- You can arbitrarily change the datarate, the frequency, the sync word and the timing
- You got access to the demodulator output and modulator input

= you can add support for protocols not natively supported by the hardware !

ESPWN32: LET'S ADD SOME ANT SUPPORT



CUSTOM FIRMWARES : NEW FEATURES

New capabilities unlocked
Injection into a BLE connection

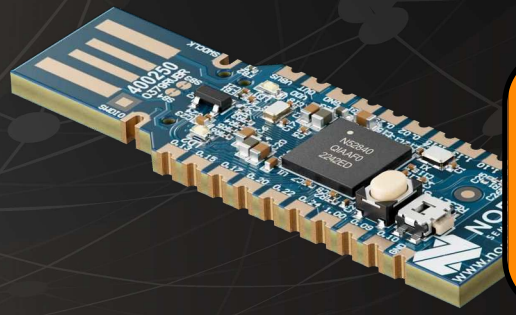
New capabilities unlocked
BLE Link Layer injection & sniffing

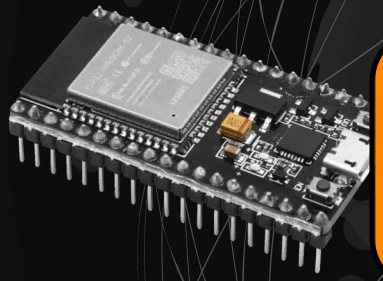
New capabilities unlocked
Low level Zigbee primitives

New capabilities unlocked
Continuous Jamming

New capabilities unlocked
Multi-protocol support

New capabilities unlocked
Multi-protocol support





A FEW OBSERVATIONS

- Mirage lacks some flexibility to perform low level attacks
- If we need to play with low level stuff, writing firmwares is the easiest way
- Writing firmware is costly: C language, complex tricks to play with low level stuff
- On the other hand, Transceivers can be pushed to their limits with funny hacks
- Can we take advantage of multi-protocol support to make great & cheap hardware ?

At this point, the question is... how can we do better ?

WHAD ABOUT REWRITING THE IDEAL TOOL FROM SCRATCH ?

LET'S DESCRIBE THE TOOL OF OUR DREAM...

PROTOCOLS

- Native support of multiple protocols
- Can be easily extended to add new ones
- Facilitate code reuse when protocols share similarities
- Full control over the protocol stack, including the lowest layers

LET'S DESCRIBE THE TOOL OF OUR DREAM...

PROTOCOLS

- Native support of multiple protocols
- Can be easily extended to add new ones
- Facilitate code reuse when protocols share similarities
- Full control over the protocol stack, including the lowest layers

HARDWARE

- Natively compatible with existing & future tools
- Considerably facilitates the development of new firmwares
- Adaptable to hardware capabilities

LET'S DESCRIBE THE TOOL OF OUR DREAM...

PROTOCOLS

- Native support of multiple protocols
- Can be easily extended to add new ones
- Facilitate code reuse when protocols share similarities
- Full control over the protocol stack, including the lowest layers

HARDWARE

- Natively compatible with existing & future tools
- Considerably facilitates the development of new firmwares
- Adaptable to hardware capabilities

SOFTWARE

- Can be used as a library in a script or another program
- Provide a set of easy-to-use command-line tools
- Allow to perform complex workflows by chaining CLI tools together
- Easy to extend, well-documented, user-friendly
- Relies on robust and common tools & libraries (scapy, wireshark, ...)

1 Internship at LAAS-CNRS

Goal: Evaluation of a protocol agnostic IDS for IoT against existing threats.

My task is to implement a set of realistic and diversified wireless attacks to attack IoT wireless protocols.

2 Mirage public releases

First release at SSTIC 2019 (v1.0)
BLE support only

Second release at ISSRE 2019 (v1.1)
support for BLE, WiFi, ZigBee, ESB, Mosart and basic InfraRed protocols

3 Low level attacks

During my PhD & postdoc research work, I work on complex attacks requiring very low level control

Damien Cauquil & I start working together on the implementation of offensive firmwares for nRF52 & ESP32 SoCs.

4 WHAD development

During my postdoc, Damien and I start working together on the tool we wish we had during our respective work

2 years of hard work before the first release at DEFCON 32 !

THE CORE IDEA: WHAD-PROTOCOL

Designing an extensible Host / Device communication protocol which is:

- **Well defined:** properly defined in protobuf & entirely documented
- **Generic:** covers all the existing wireless capabilities
- **Modular:** supports multiple wireless protocols
- **Evolutive:** designed to be extended and improved
- **User-friendly:** comes with C, C++ & Python parsing libraries

LET THE HARDWARE DO HARDWARE STUFF

HOST SIDE

- Flexible protocol stacks
- Packets dissectors & builders
- Attacks & tools

FIRMWARE SIDE

CPU & MEMORY LIMITATIONS

- Atomic wireless primitives
- Time-critical operations
- Minimal logic

OFFLOAD COMPLEXITY

DEVICE DISCOVERY AND CAPABILITIES

protobuf
Protocol Buffers

GENERIC

DEVICE

Discovery

Protocol version

Firmware info

Supported domains

DOMAINS

PROTOCOL1

Capabilities

SCAN INJECT JAM

Commands

PROTOCOL2

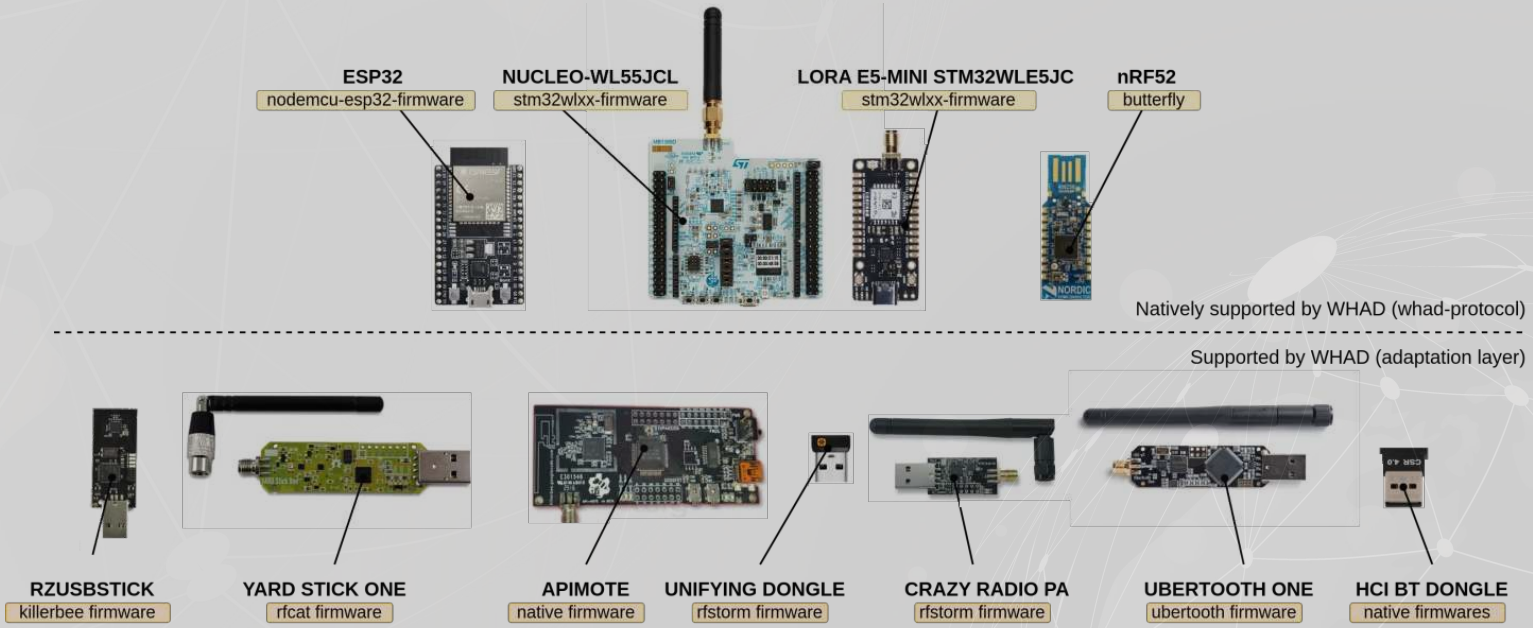
Capabilities

HIJACK SIMU.

Commands

...

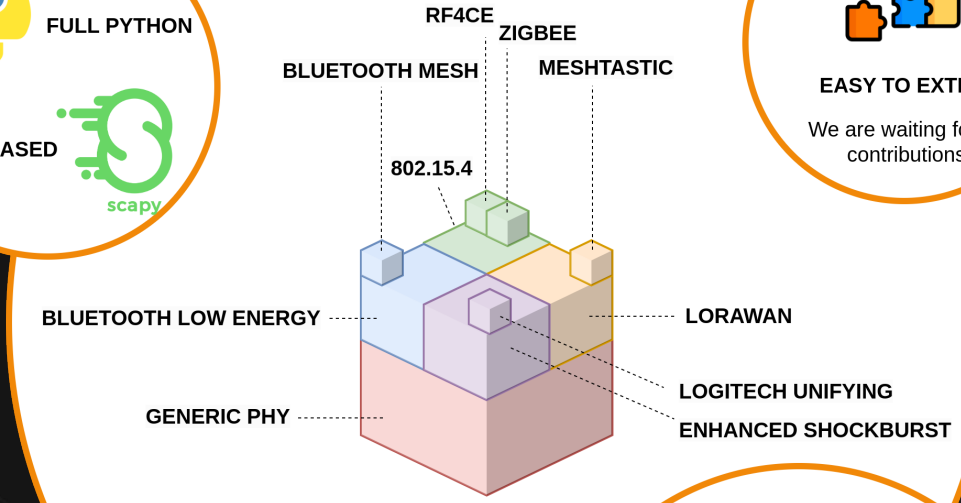
COMPATIBLE HARDWARE



SUPPORTED PROTOCOLS

FULL PYTHON

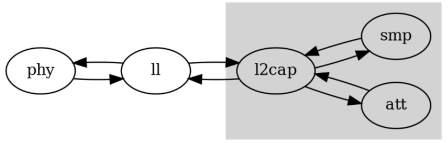
SCAPY-BASED



EASY TO EXTEND

We are waiting for your contributions ;)

FLEXIBLE STACK MODEL

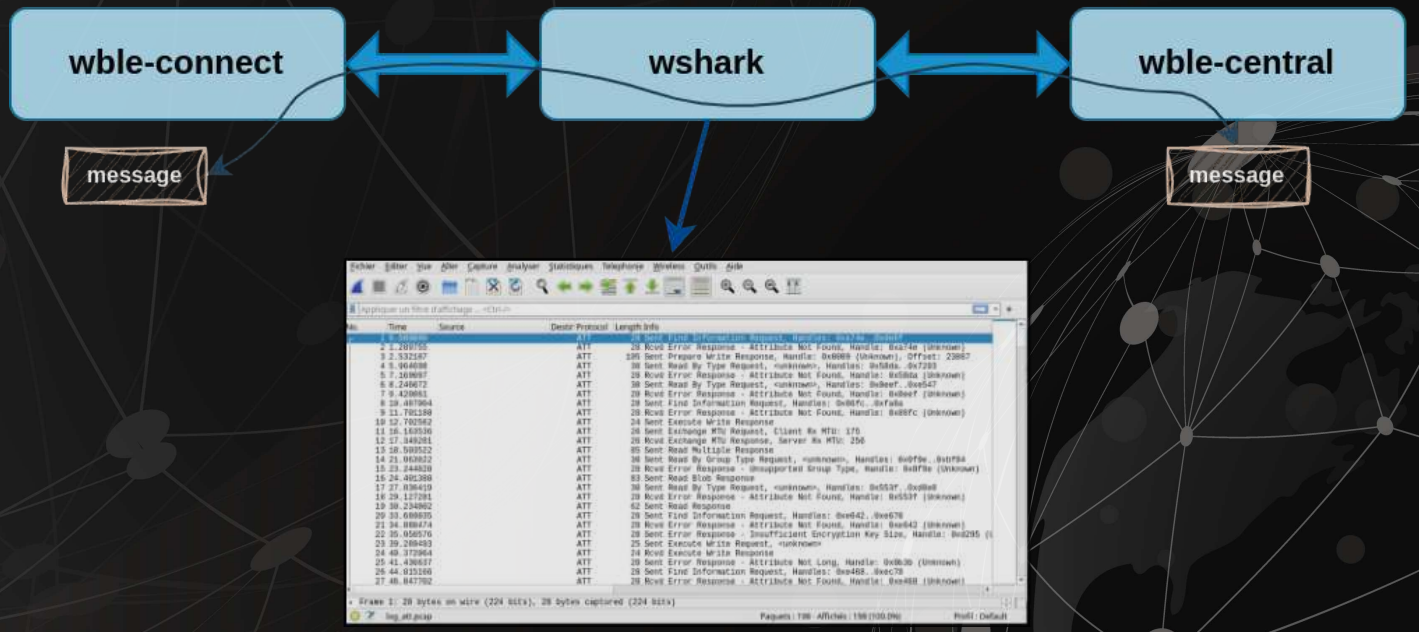


- easy to modify a specific layer
- protocol stack state snapshot
- easy to implement unit tests

IT'S WAAAAY MORE THAN A PROTOCOL !

THE RETURN OF THE PIPE

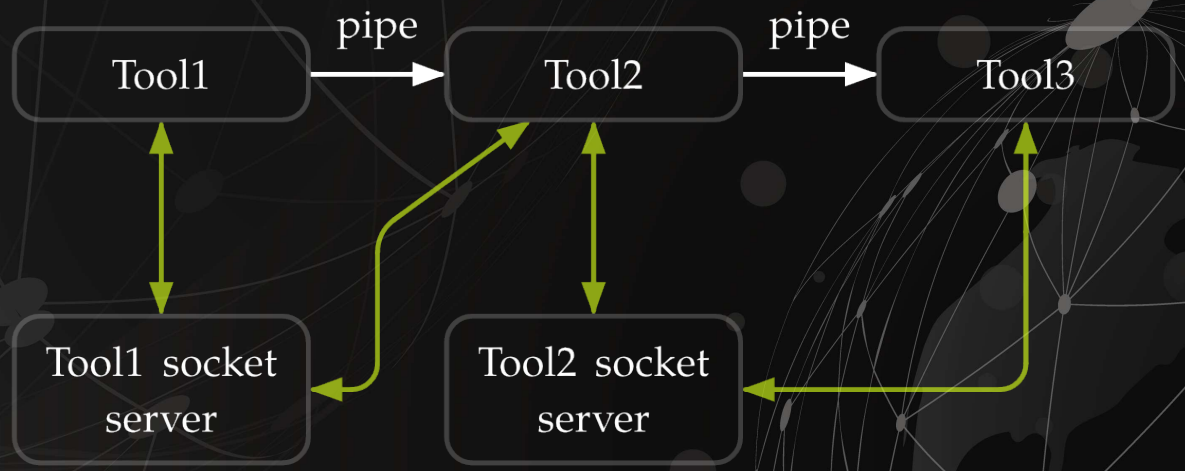
```
$ wble-connect -i hci0 00:11:22:33:44:55 | wshark | wble-central profile
```



PIPE INTERNALS

Pipe is uni-directional, but radio implies bi-directionnality

→ **Damien found a very cool design:** a reference to an UNIX socket acting as a proxy to the radio is propagated on the standard output between tools




Demo time !

INTERACTION WITH A ZIGBEE NETWORK

EMULATING BLE & LOGITECH UNIFYING DEVICES

Let's build a Logitech Unifying to BLE proxy !

(BECAUSE WHY NOT.)

EMULATING BLE & LOGITECH UNIFYING DEVICES

PACKETS, BUT OVER THE WIRE

1 Internship at LAAS-CNRS

Goal: Evaluation of a protocol agnostic IDS for IoT against existing threats.

My task is to implement a set of realistic and diversified wireless attacks to attack IoT wireless protocols.

2 Mirage public releases

First release at SSTIC 2019 (v1.0)
BLE support only

Second release at ISSRE 2019 (v1.1)
support for BLE, WiFi, ZigBee, ESB, Mosart and basic InfraRed protocols

3 Low level attacks

During my PhD & postdoc research work, I work on complex attacks requiring very low level control

Damien Cauquil & I start working together on the implementation of offensive firmwares for nRF52 & ESP32 SoCs.

4 WHAD development

During my postdoc, Damien and I start working together on the tool we wish we had during our respective work

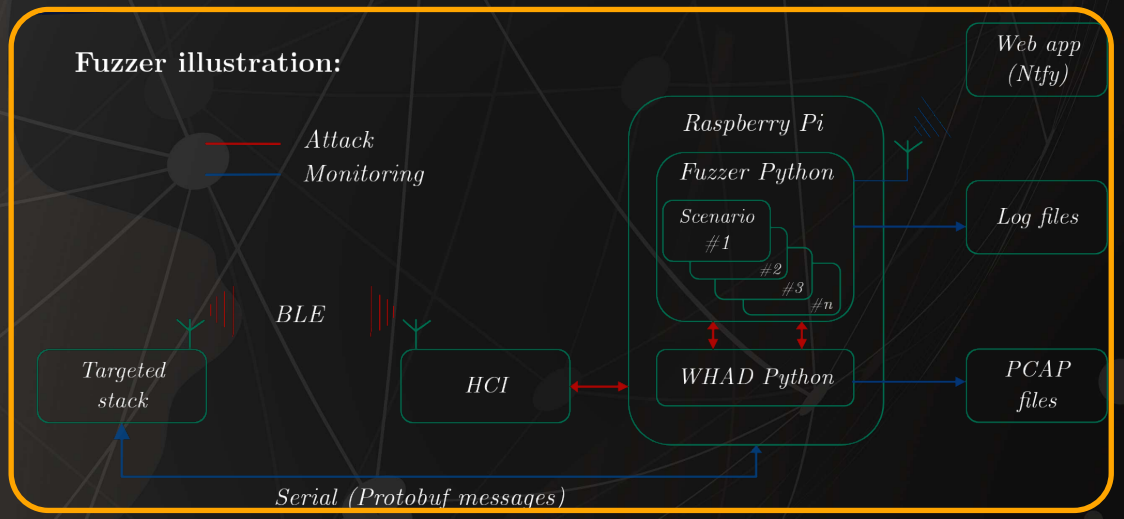
2 years of hard work before the first release at DEFCON 32 !

5 Since the release...

We used it in a ton of cool stuff ! :)

Research, Pentest, Teaching, Challenges...

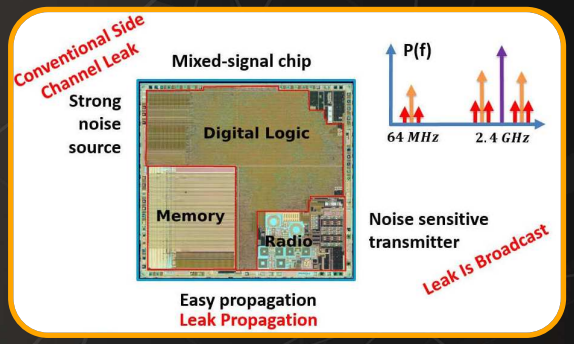
RESEARCH: FUZZING OF BLE GATT LAYER



- Internship of Baptiste Boyer at Quarkslab
- Identification and analysis of several non compliances and vulnerabilities:
  - Permissions bypass
  - Denial of Service
  - Out of Bonds write
- CVE assigned for Apache NimBLE stack: CVE-2024-24746
- Talk presented at Hardwear.io 2024

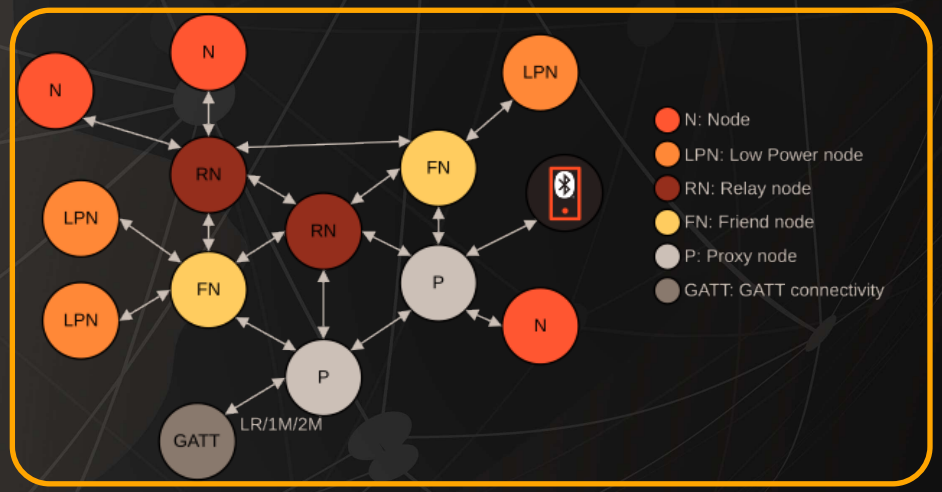
RESEARCH: BLUESCREAM, APPLYING SCREAMING CHANNELS TO BLUETOOTH LOW ENERGY

Screaming Channel attack: side channel attacks at large distance against mixed-signal chips used in modern connected devices (intermodulation of the leak signal with the RF carrier)



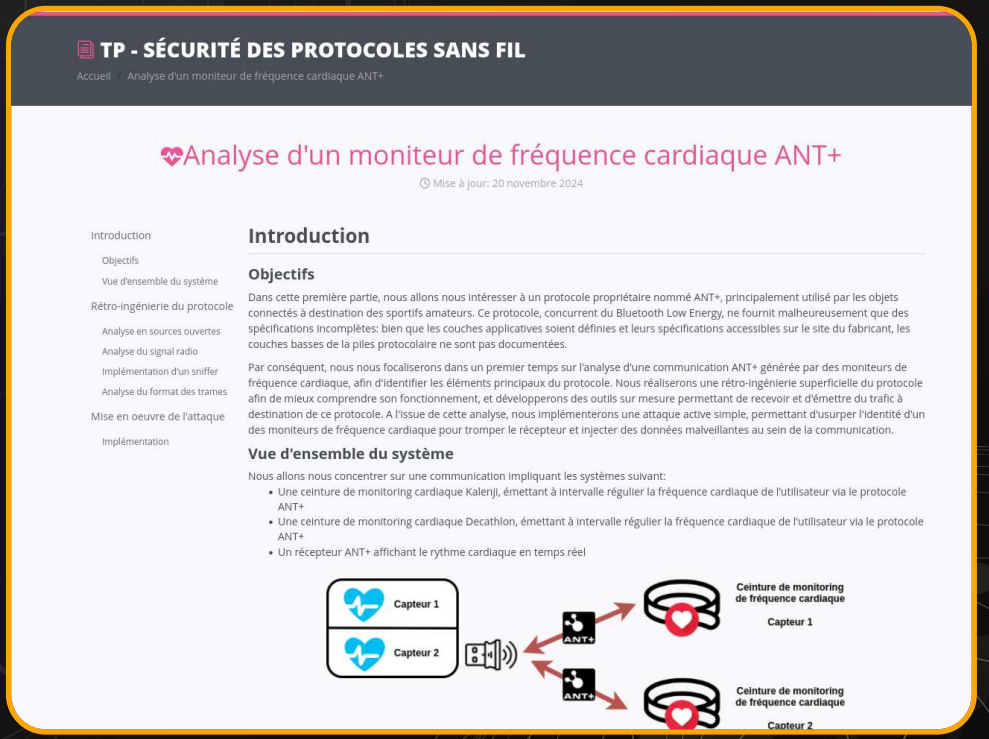
- Amazing contribution of Pierre Ayoub PhD thesis (EURECOM)
- Original attack (Camurati et al., CCS 2018) demonstrated in laboratory conditions: is it a realistic threat when a "real" wireless protocol is used ?
- We showed the feasibility of retrieving the Long Term Key by exploiting the leak during the session key derivation in a BLE connection
- WHAD allowed to:
  - inject carefully crafted LL packets (to trigger the leak during the session key derivation)
  - reliably trigger the leak recording with an SDR
- Paper presented at ACSAC'24

RESEARCH: SECURITY ANALYSIS OF BLUETOOTH MESH ROUTING MECHANISMS



- Internship of Elies Tali at LAAS-CNRS
- Implementation of a BLE Mesh protocol stack on top of WHAD BLE stack
- Analysis of the specification, leading to:
  - the identification of 2 vulnerabilities
  - the design of 3 attacks (nodes discovery & denial of service)
- Paper presented at THCon and SSTIC 2025 !

TEACHING: WIRELESS SECURITY LAB FOR TLS-SEC



TP - SÉCURITÉ DES PROTOCOLES SANS FIL

Accueil / Analyse d'un moniteur de fréquence cardiaque ANT+

Analyse d'un moniteur de fréquence cardiaque ANT+

Mise à jour: 20 novembre 2024

Introduction

Objectifs

Dans cette première partie, nous allons nous intéresser à un protocole propriétaire nommé ANT+, principalement utilisé par les objets connectés à destination des sportifs amateurs. Ce protocole, concurrent du Bluetooth Low Energy, ne fournit malheureusement que des spécifications incomplètes: bien que les couches applicatives soient définies et leurs spécifications accessibles sur le site du fabricant, les couches basses de la piles protocolaire ne sont pas documentées.

Par conséquent, nous nous focaliserons dans un premier temps sur l'analyse d'une communication ANT+ générée par des moniteurs de fréquence cardiaque, afin d'identifier les éléments principaux du protocole. Nous réaliserons une rétro-ingénierie superficielle du protocole afin de mieux comprendre son fonctionnement, et développerons des outils sur mesure permettant de recevoir et d'émettre du trafic à destination de ce protocole. A l'issue de cette analyse, nous implémenterons une attaque active simple, permettant d'usurper l'identité d'un des moniteurs de fréquence cardiaque pour tromper le récepteur et injecter des données malveillantes au sein de la communication.

Vue d'ensemble du système

Nous allons nous concentrer sur une communication impliquant les systèmes suivant:

- Une ceinture de monitoring cardiaque Kalenji, émettant à intervalle régulier la fréquence cardiaque de l'utilisateur via le protocole ANT+
- Une ceinture de monitoring cardiaque Decathlon, émettant à intervalle régulier la fréquence cardiaque de l'utilisateur via le protocole ANT+
- Un récepteur ANT+ affichant le rythme cardiaque en temps réel

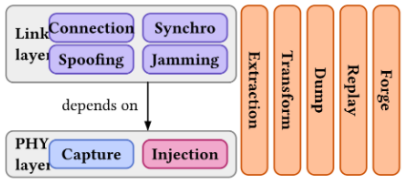
Il s'agit d'un protocole propriétaire basé sur le protocole Enhanced ShockBurst de Nordic Semiconductors, auquel ont été ajoutés un mécanisme de saut de fréquence simple afin de réduire les problématiques liées aux interférences dans la bande ISM 2.4GHz. Le protocole reprend de nombreuses caractéristiques du protocole ShockBurst, telles que la topologie en étoile, composée de deux types de noeuds:

BUT ALSO ...

- Used internally at Quarkslab for several security audits
- 3 workshops at BruCON, Hardwear.io and Ph0wn
  - ...we plan to do more trainings soon, stay tuned !
- Heavily used in **Hardware CTF** at Hardwear.io for two consecutive years:
  - BLE challenges were 100% emulated with WHAD
  - LoRaWAN challenge was also emulated with WHAD
- Used to implement the **Hardware CTF** at Ph0wn 2024
  - Picobox Revolution challenge
- Various academic papers in preparation
  - (see you at SSTIC 2025 !)

A FEW COOL PERSPECTIVES

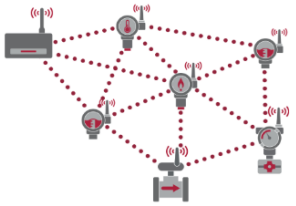
SYSTEMATIZATION OF WIRELESS ATTACKS



(see you at SSTIC 2025 !)

IMPLEMENTATION OF INDUSTRIAL PROTOCOLS

WirelessHART

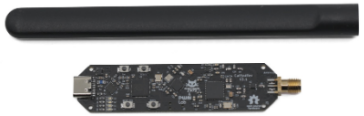


SIMPLE. RELIABLE. SECURE.

(so they say)

A IN-DEPTH ANALYSIS OF INTERACTIONS BETWEEN DIFFERENT PROTOCOLS

AN UNIVERSAL MULTI-PROTOCOL FIRMWARE

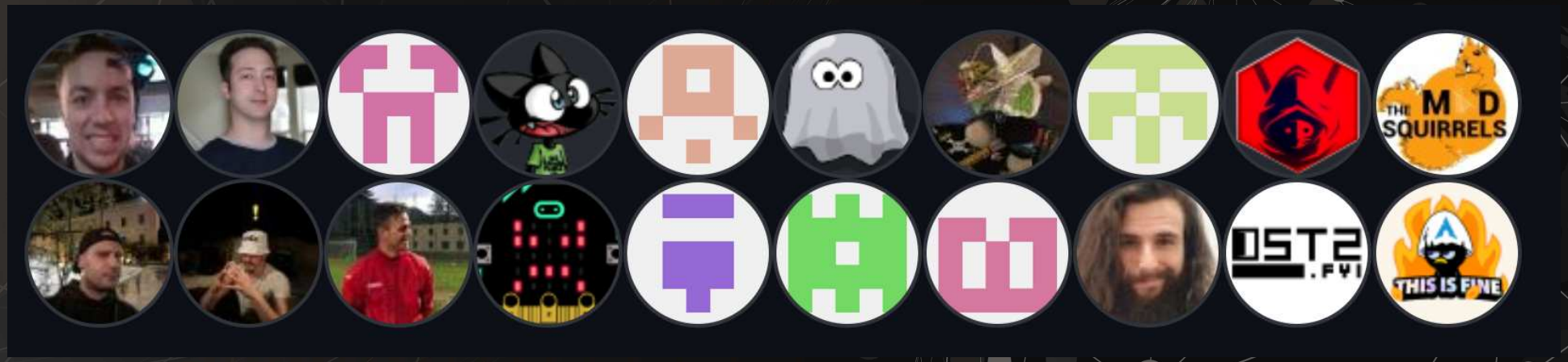


WHAD-SDR

LET'S CONCLUDE THIS TALK

This a team work, and I have a ton of people to acknowledge:

- Damien Cauquil, of course
- Jonathan Roux, Florent Galtier, Vincent Nicomette and Guillaume Auriol for their support during my PhD
- Aurélien Francillon, Pierre Ayoub, Aurélien Hernandez for the cool research work we did at EURECOM
- All the security researchers who tested WHAD and provide us feedbacks: jduck, Mike Ryan, Xeno Kovah, Slawomir Jasek, Jiska Klassen, Axelle Apvrille, MadSquirrels, Fenrisfulsur 🙏



CALL FOR CONTRIBUTIONS

We want to give this project to the community, so don't hesitate to:

- Create compatible firmwares for unsupported hardware
- Report bugs and issues on GitHub
- Help writing documentation
- Add support for more protocols !
- Spread the word and tell everyone to use it 🥰

Everything is open-source (MIT License)

THANKS FOR YOUR ATTENTION !

```
$ pip install whad
```

- Website available at: https://whad.io
- Documentation available on ReadTheDocs
- Github organization @ github.com/whad-team
  - Main library and CLI tools available in repository whad-client
  - WHAD Protocol available in repository whad-protocol
  - Firmwares available in sub-repos