



**Université Bretagne Sud**

Master 1 Ingénierie de Systèmes Complexes

Spécialité Cybersécurité des Systèmes Embarqués

**Promotion 2019-2020**

Étude du protocole BLE

**Stage Master 1**

Gidon Rémi

*Avril/Juin 2020*

## Contents

<b>1</b>	<b>Objets connectés</b>	<b>6</b>
1.1	Architecture . . . . .	6
1.2	Protocoles . . . . .	6
1.3	Marché . . . . .	6
<b>2</b>	<b>Bluetooth Low energy</b>	<b>7</b>
2.1	Differences . . . . .	7
2.2	Protocole . . . . .	8
2.3	Connexion . . . . .	10
2.4	Versions . . . . .	10
<b>3</b>	<b>Vulnerabilites</b>	<b>12</b>
<b>4</b>	<b>Outils offensif</b>	<b>13</b>
4.1	Logiciels . . . . .	13
4.2	Matériels . . . . .	13
<b>5</b>	<b>Mirage</b>	<b>15</b>
5.1	Presentation . . . . .	15
5.2	Integration . . . . .	15
<b>6</b>	<b>Spécifications</b>	<b>16</b>
6.1	Fonctionnalités . . . . .	16
6.2	Architecture . . . . .	17
6.3	Interface . . . . .	18
6.4	Tests . . . . .	19
6.5	Livrables . . . . .	19
<b>7</b>	<b>Preuve de concept</b>	<b>20</b>
7.1	Sniffing . . . . .	20
7.2	Localization . . . . .	20
7.3	MITM . . . . .	21

## Tableaux

## Figures

Dans le cadre du master CSSE nous étudions l'internet des objets (IoT) et leurs aspects sécurité. Le protocole réseau sans fil Bluetooth Low Energy (BLE) permet une consommation réduite pour les objets fonctionnant sur batterie, visant notamment les objets connectés. Même si intégré à la spécification Bluetooth, il est incompatible avec les autres variantes de celui-ci.

Aujourd'hui intégré dans la plupart des appareils de bureautique, il est rapidement devenu populaire dans l'internet des objets.

La première itération du BLE (sortie en 2011) ne répond plus aux exigences de sécurité contemporaine et même si le protocole a su évoluer depuis pour répondre à ces besoins, beaucoup d'appareils utilisent encore la version originale n'intégrant pas ces mécanismes.

Ce sont pour la plupart des appareils conçus pour fonctionner sur batterie et communiquer en point à point. On va retrouver les capteurs corporels pour santé ou fitness mais également des mécanismes plus sensibles tels des cadenas ou serrures. Les communications (incluant parfois des données personnelles) peuvent être interceptées, voir modifiées pour réaliser une action non voulue (ouverture de cadenas).

## 1 Objets connectés

Avec l'explosion de l'internet de objets (TODO chiffres) toute une flopée d'objet du quotidien ont été augmentés pour permettre la communication avec d'autres systèmes informatiques dont nos smartphones ou encore des serveurs distants (via notre WiFi). Ces objets dits intelligents étendent leur équivalent mécanique en intégrant des composants électroniques, permettant notamment le contrôle à distance. Cependant ces améliorations engendrent une augmentation de la surface d'attaque: les objets connectés sont confrontés aux mêmes challenges que ceux des systèmes informatiques traditionnels en plus de leur fonction primaire.

TODO sécurité plus en plus prise en compte mais secondaire tj

### 1.1 Architecture

#### Point à point

Architecture réseau domotique

- simple: appareil non relié au réseau, dépendant gateway utilisateur, remplissant une fonction d'augmentation seule (smart lock)

#### Réseau

- avancée: appareil s'appuyant sur un réseau domotique pour réaliser ses fonctions, relié à une gateway "sûre" hub

### 1.2 Protocoles

Protocoles généraux supportés par tout appareil (smartphone notamment) et peu cher WiFi (WLAN) ~50m: Local = remplace câbles pour appareils fixes dans pièces / appart BLE (WPAN) ~10m: Personnel = remplace câbles pour appareils portables personnels NFC

Protocoles spécifiques conçus pour ces réseaux Zigbee Zwave Thread ANT(+)

### 1.3 Marché

Première génération point à point "smart"

Seconde génération réseaux IoT

#### BLE

Gadgets (fitness)

Domotique (IoT)

Entreprise / warehouse / smart city (beacons)

## 2 Bluetooth Low energy

Le protocole a été désigné par Nokia et d'autres entreprises pour répondre au besoin d'un protocole sans fil peu gourmand en énergie pour les périphériques personnels (téléphone portable, montre, casque). Nommé Wibree, il a été intégré au standard Bluetooth sous le nom *Low Energy*.

Le Bluetooth ne comprend pas seulement un protocole mais une multitude d'entre eux (BR, EDR, HS) qui ont en commun de permettre la communication (et l'échange de données) sans fil avec des périphériques personnels. Ils font partie des protocoles WPAN (réseau personnel sans fil) et leur distance d'émission est de quelques mètres jusqu'à 30 mètres.

La spécification Bluetooth 4.0, sortie en 2011, intègre le protocole LE (Low Energy) et permet au Bluetooth de toucher le marché des systèmes embarqués, fonctionnant sur batterie.

### 2.1 Differences

Les autres protocoles du Bluetooth sont principalement connus et utilisés pour le transfert de contenu multimédia, que ce soit des fichiers entre ordinateurs comme de la musique avec un casque ou encore une voiture. Ils fonctionnent avec une connexion continue et un transfert en flux.

Le BLE, visant à réduire la consommation d'énergie, n'établit pas de connexion continue. L'appareil reste la plupart du temps en mode veille, pouvant émettre des annonces, dans l'attente d'une connexion qui aura pour effet d'arrêter la transmission d'annonce. Pour chaque requête reçue, une réponse pourra être renvoyée directement ou une notification mise en place périodiquement.

Les appareils BLE et Bluetooth BR/EDR ne sont pas compatibles, n'utilisant pas les mêmes technologies, protocoles et répondant à des besoins différents (voir @tbl:bluetooth-use-case).

Besoin	Flux données	Transmission données	Localisation	Réseau capteurs
<b>Appareils</b>	ordinateur, smartphone, casque, enceinte, voiture	accessoires bureautique ou fitness, équipement médical	beacon, IPS, inventaire	automatisation, surveillance, domotique
<b>Topologie</b>	point à point	point à point	diffusion (1 à N)	mesh (N à N)
<b>Technologie</b>	Bluetooth BR/EDR	Bluetooth LE	Bluetooth LE	Bluetooth LE

: Cas d'utilisation et protocole Bluetooth adapté {#tbl:bluetooth-use-case}

## 2.2 Protocole

Pour permettre une interoperabilité maximale entre les appareils BLE, le standard defini 4 profils en fonction du role de l'appareil: Peripheral, Central, Broadcaster, Observer. Ces roles constituent le *GAP* (*Generic Access Profile*).

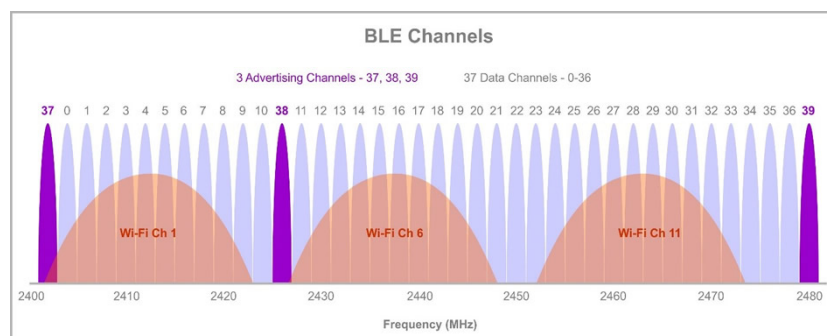
Chaque appareil se conformant au standard ne doit implementer qu'un seul de ces roles a la fois.

Le *Broadcaster* ne communique qu'avec des annonces, on ne peut pas s'y connecter. Ce mode est tres populaire pour les beacons. L'*Observer* est sont opposé, il ne fait qu'ecouter les annonces, n'etabliera jamais de connexion.

Le *Peripheral* et le *Central* forment la seconde pair et permettent la mise en place d'une architecture client-serveur. Le *Peripheral* joue le role du serveur et est dit *esclave* du *Central* qui endosse le rôle du client et *maître*.

L'esclave transmet des annonces jusqu'a recevoir une connexion d'un maitre, apres quoi il arrete de s'annoncer car il ne peut etre connecté qu'a un maitre a la fois. Le maitre ecoute les annonces d'esclave (annonces connectables) pour se connecter, puis interroge ses services via le *GATT* (*Generic Attribute*).

### Couche physique



{#fig:ble-

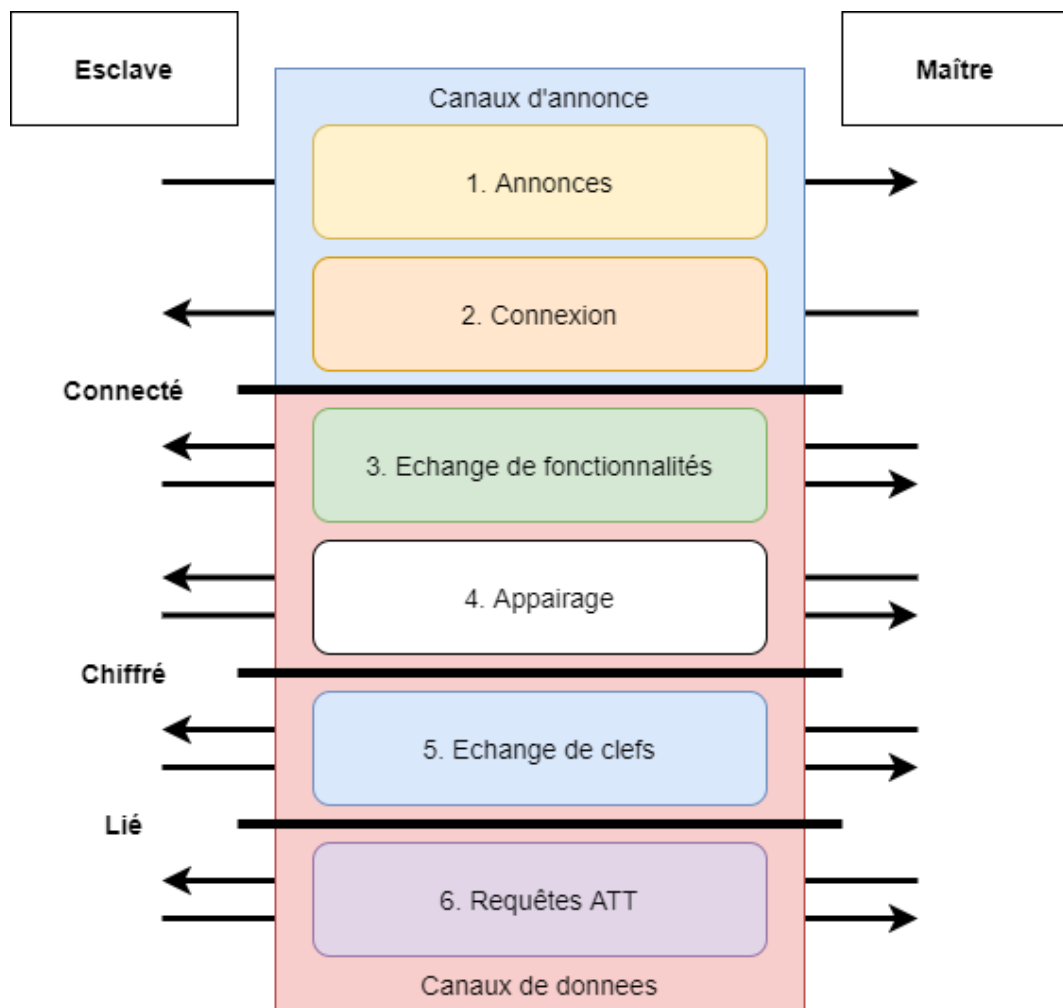
channels width=90%}

40 channels 3 adv 37 data hopping car tres utilisé donc bruit

### Couche logique

L2CAP GAP GATT SMP ATT





{#fig:ble-

conn width=50%}

Le protocole se decompose en 3 phases:

- annonce
- connexion
- echanges

**Annonces** advertisements

adv packet structure GAP AD type [https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/02/10/bluetooth\\_advertisin-hGsf](https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/02/10/bluetooth_advertisin-hGsf) <https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/>

Les beacons bluetooth ont beaucoup fait parler d'eux que ce soit pour la localisation de marchandise dans les entrepot, de personnes dans les magasins ou de publicites cibles dans les villes.

Ils se basent sur un mode special d'annonce non connectable emit en diffusion (*broadcast*) a tout les appareils a proximite. L'annonce transporte directement les donnees utiles suivant le format *GAP*.

## 2.3 Connexion

phase 1 feature exchange <https://www.bluetooth.com/blog/bluetooth-pairing-part-1-pairing-feature-exchange/>

phase 2 key generation method

### Just Works

[https://www.bluetooth.com/blog/bluetooth-pairing-part-2-key-generation-methods/?utm\\_campaign=developer&utm\\_source=internal&utm\\_medium=blog&utm\\_content=bluetooth-pairing-part-1-pairing-feature-exchange](https://www.bluetooth.com/blog/bluetooth-pairing-part-2-key-generation-methods/?utm_campaign=developer&utm_source=internal&utm_medium=blog&utm_content=bluetooth-pairing-part-1-pairing-feature-exchange)

phase 3 temp key generation and short/long term key derivation

## Communication

GATT & ATT proto <https://fr.mathworks.com/help/comm/examples/modeling-of-ble-devices-with-heart-rate-profile.html>

interop via profiles (API commune) -> GATT protocole

All Bluetooth Low Energy devices use the Generic Attribute Profile (GATT). The application programming interface offered by a Bluetooth Low Energy aware operating system will typically be based around GATT concepts.[44] GATT has the following terminology:

Client A device that initiates GATT commands and requests, and accepts responses, for example, a computer or smartphone. Server A device that receives GATT commands and requests, and returns responses, for example, a temperature sensor. Characteristic A data value transferred between client and server, for example, the current battery voltage. Service A collection of related characteristics, which operate together to perform a particular function. For instance, the Health Thermometer service includes characteristics for a temperature measurement value, and a time interval between measurements. Descriptor A descriptor provides additional information about a characteristic. For instance, a temperature value characteristic may have an indication of its units (e.g. Celsius), and the maximum and minimum values which the sensor can measure. Descriptors are optional – each characteristic can have any number of descriptors.

fonctionnement apparaige (phases)

## 2.4 Versions

4.0 arrivee BLE Version visee par le PoC

4.2

5.1

specification du Bt pour les systemes embarques, bcp utilise dans objets connectes

4.0 arrivee

4.2 securite

5.0 mesh networks for home automation or sensor networks use bluetooth mesh profile General Access Profile (GAP)

5.1 localisation

Utilisation “abusive” dans les objets connectes ?

### **3 Vulnerabilites**

COnfidentialite Appairage

Authentification Appairage

...

## 4 Outils offensif

### 4.1 Logiciels

Etude des communications bluetooth: Wireshark Scappy etc Possible avec n'importe quel chip Bt déjà sur la machine ou dongle USB pour une étude du trafic interne et des appareils émetteurs des adv.

Interceptions des communications

- BTLE (C)
- BTLEJack (lib python + firmware C)
- Mirage (framework python)

Propriétaires:

- nRF sniffer
- nRF Connect
- smartRF (TI)

Attaques

- GATTacker (NodeJS) MiTM
- BTLEJuice (NodeJS) MiTM
- BTLEJack (Jamming/ Hijacking)
- Mirage (MiTM / jam / hijack / crack)

### 4.2 Matériels

We can BLE dedicated devices to sniff or modify it. Internal Bt chips can only adv or connect to peripherals but never scan or modify it. They only see internal traffic (locked firmware).

Full featured HackRF PandwaRF Ubertooth

BLE HCI Dongle nRF52840 (<https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840>)

- <https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF52840-Dongle> Some using CSR8510 (<https://www.qualcomm.com/products/csr8510>)
- Adafruit Bluetooth 4.0 USB Module (<https://www.adafruit.com/product/1327>)
- <https://www.amazon.co.uk/CSR8510-Bluetooth-Adapter-Classic-Headset/dp/B01G92CNY8>

Qualcomm, Broadcom, Realtek, NordicSemiconductor ... Featured in documentation is Qualcomm one

Sniffer

- Ubertooth One (\$\$)
- BTLEJack BBC Micro:Bit, Bluefruit, Waveshare BLE400, nRF51822 Eval kit (tweak) (<https://github.com/virtualabs/btlejack>)
- Bluefruit <https://www.adafruit.com/product/2269> (limited)

- nRF51 <https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF51-Dongle> (close)
- TI CC2540 USB Dongle BLE sniffer (<http://www.ti.com/tool/CC2540EMK-USB>)
- Crazy Radio PA 2.4GHz (<https://store.bitcraze.io/collections/kits/products/crazyradio-pa>)

#### Board

- HackRF
- PandwaRF

## 5 Mirage

### 5.1 Présentation

### 5.2 Integration

Si je me concentre sur Mirage, cela restreint pas mal les outils possible:

- dongle BLE HCI standard
- sniffer BLE adaptable avec BTLEJack (micro:bit, bluefruit, ble400, nRF51)  
Les appareils dépendent des besoin, dans mon cas il me faudrait:
- inventaire: Sniffer (BTLEJack)
- obtention d'informations (crack, mit): dongle HCI x2 (un slave et un master, a voir si un BTLEJack peut remplacer un HCI)
- localisation / tracking (rssi + autres méthodes): Mirage ne permet pas cela nativement mais les informations demandées doivent être récupérables dans le framework pour l'implémenter manuellement (RSSI, angle antenne ?). Cela demande au minima un dongle HCI, meme si les travaux trouvés sur le sujet utilisent un sniffer Bluefruit. Dans les travaux étudiés, la localisation demande 3+ appareils BLE pour permettre la trilatération

Il me manque donc a voir si un BTLEJack peut remplacer un HCI dans l'attaque MITM, ainsi que trouver des informations pour implémenter la localisation IPS avec Mirage. Mirage supporte également d'autres appareils (comme Ubertooth) mais leurs fonctionnalités ne nous sont pas nécessaires, un sniffer flashé avec BTLEJack suffit (et coute moins cher). Pour les sniffers BTLEJack éligibles:

- Bluefruit et nRF51 (~20e) demandent reprogrammation via un "external SWD" (assez cher + 100e)
- la carte BBC Micro:bit (20e, non vendue en France directement) permet une reprogrammation sans appareil supplémentaire, et semble donc la plus simple

Pour résumer:

- dongle BLE (<https://www.adafruit.com/product/1327> / <https://www.amazon.co.uk/CSR8510-Bluetooth-Adapter-Classic-Headset/dp/B01G92CNY8>)
- carte Micro:Bit (<https://microbit.org/buy/>)

## 6 Spécifications

Sujet: Mettre en place des attaques sur le protocole *Bluetooth Low Energy* (Bluetooth Smart)

### 6.1 Fonctionnalités

La preuve de concept devra fournir plusieurs fonctionnalités offensive décritent ci-après.

#### Repérage

Inventaire des appareils et connexions BLE à proximité.

- Écoute des annonces sur les 3 canaux publicitaires pour récupérer les appareils émetteurs.
- Écoute des communications sur les 37 canaux de données pour répertorier celles active.

#### Localisation

Localisation des appareils BLE alentours.

- Écoute passive des annonces pour extraire le calibrage du signal et calculer la distance à partir de la puissance du signal reçu.
- Si le calibrage n'est pas émit dans l'annonce, établissement d'une connexion pour récupérer la valeur si disponible.

Opération répétables autant de fois que voulu pour améliorer la précision de la localisation (minimum 3 mesures pour une position).

#### Identification

Connexion directe à un appareil via son adresse bluetooth pour extraire toutes les données exposées.

- Écoute optionnelle des annonces pour identifier un esclave cible.
- Requête de connexion à la cible en tant que maître.
- Récupération des informations standardisées (GAP/GATT) ainsi que services et attributs propriétaires.

#### Interception

Interception de communications et possible déchiffrement des trames.

- Écoute des communications sur les 37 canaux de données.
- Récupération de l'adresse d'accès et des paramètres d'appairage (carte des canaux, temps et nombre de sauts, etc).
- Synchronisation avec la communication et écoute des trames.
- Si la communication est chiffrée et la phase d'appairage passée, déconnexion des appareils via brouillage des communication jusqu'au temps mort.



- Écoute des canaux d'annonce: attente d'un appairage en supposant qu'il provienne des appareils précédemment déconnectés.
- Récupération des informations cryptographique pour déchiffrer la connexion seulement si celle-ci n'utilise pas une clef à long terme déjà établie ou une connexion sécurisée (BLE 4.2).
- Écoute des communications et déchiffrement des trames à la volée.

### Modification

Attaque *man in the middle* par clonage et usurpation d'un appareil BLE pour modifier les données échangées.

- Écoute passive des annonces de l'esclave cible de l'usurpation pour retransmission ultérieure et récupération de l'adresse bluetooth.
- Connexion à l'esclave cible d'usurpation pour qu'il n'émette plus d'annonces.
- Changement de l'adresse de l'usurpateur en celle de l'esclave usurpé et réémission des annonces précédemment capturées.
- Attente de la connexion du maître.
- Appairage entre l'usurpateur et le maître.
- Retransmission des communications entre le maître et l'esclave par l'usurpateur.

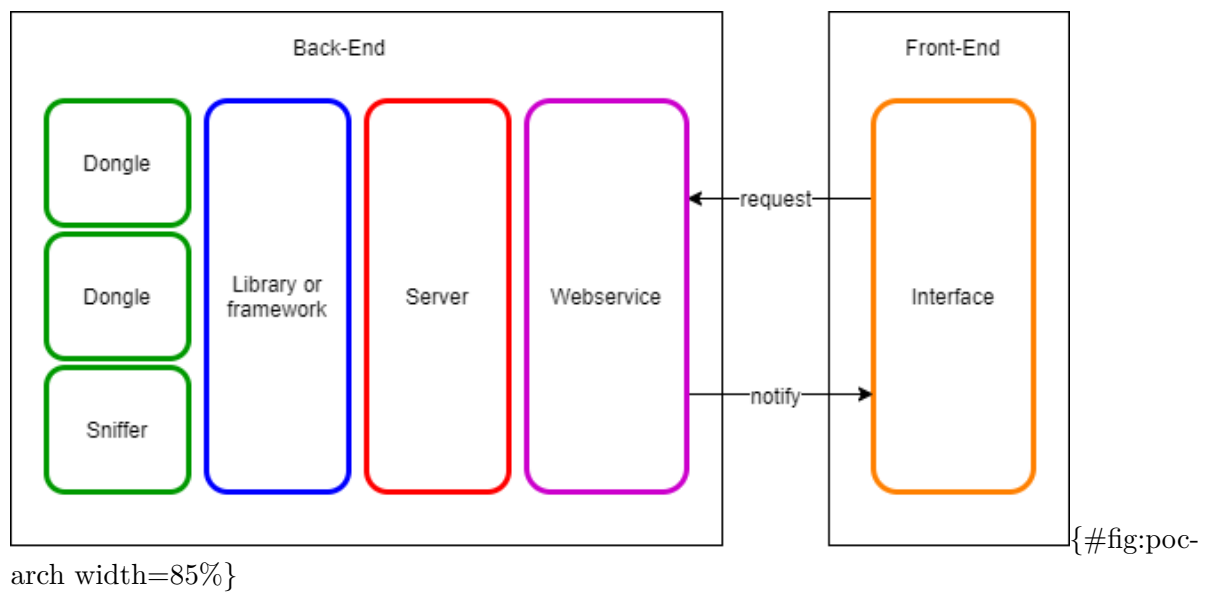
Il sera par la suite envisageable d'associer plusieurs fonctionnalités pour réaliser des scénarios différents. Ce peut être par exemple l'usurpation d'un appareil suite au brouillage lors de l'interception des communications entre 2 appareils.

## 6.2 Architecture

Le système se compose d'un front-end fournissant une interface utilisateur affichant les appareils BLE et les actions possible ainsi qu'un back-end permettant la réalisation des actions implémentées.

Le back-end se compose d'un service web (en violet sur @fig:poc-arch) pour communiquer avec le front-end, il transmet les requêtes au serveur (en rouge) qui se base sur un framework BLE offensif (en bleu) pour les traiter. Le framework BLE offensif utilise plusieurs appareils BLE (en vert) pour mener à bien les attaques.

Le serveur orchestre les attaques même si il ne les implémentent pas lui-même.



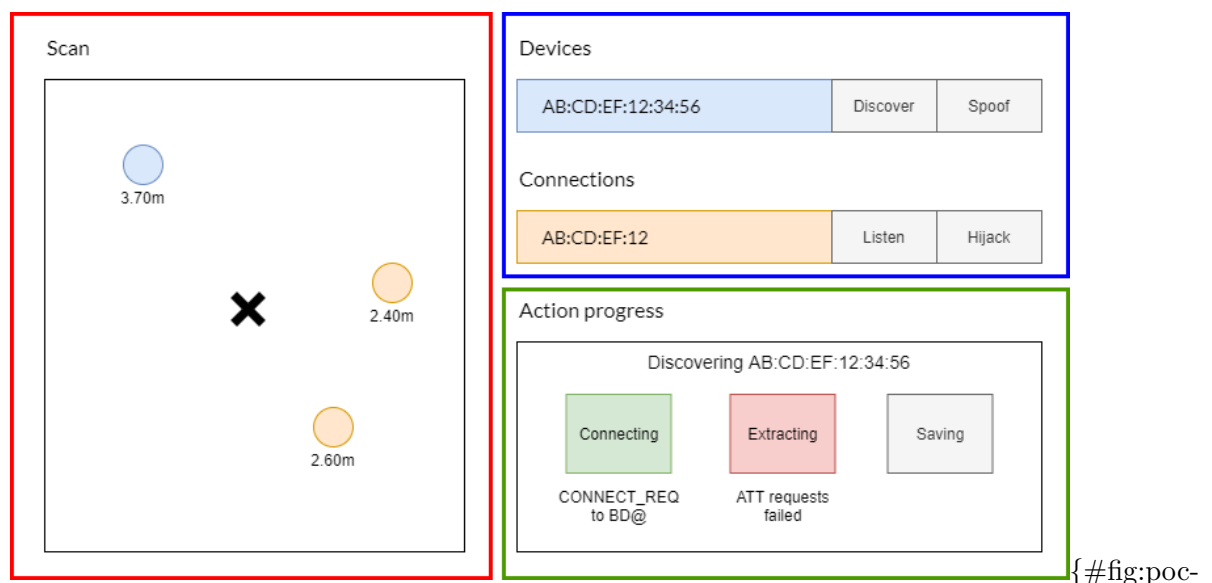
### 6.3 Interface

On retrouve la carte des appareils et connexions identifiés avec leur distance et position estimée par rapport au système (voir @fig:poc-ui: zone rouge *Scan*).

Pour chaque cible (appareil ou connexion), des attaques sont disponibles:

- Récupération du profil ou modification des transmissions par usurpation pour un appareil BLE émettant des annonces (zone bleue *Devices*).
- Déconnexion des appareils ou interception des communications entre deux appareils appairés (zone bleue *Connections*).

Une troisième section permet de suivre le déroulement de l'attaque choisie (zone verte *Action progress*). Celle-ci est découpée en phases, dès que la phase courante est terminée sans erreur (carré vert), la phase suivante est exécutée. Lorsqu'une phase échoue l'attaque s'arrête et le message d'erreur est affiché en dessous (carré rouge).



ui width=85%}

## 6.4 Tests

Il est possible de tester toutes les attaques en mettant en place un réseau BLE de test. Toutes les attaques ne ciblent jamais plus de 2 appareils BLE. Il est possible de reproduire les conditions attendues dans l'attaque en imitant un esclave et un maître BLE avec des requêtes et réponses préprogrammées. Sur chaque attaque demande des conditions de départ différentes, les appareils peuvent être en attente (émettant des annonces), en appairage ou connectés.

Une fois notre réseau test mis en place, l'attaque est exécutée sur celui-ci et les résultats obtenus comparés par rapport à ceux préprogrammés dans le test.

Il est possible d'automatiser ces tests avec 5 appareils (4 dongles et 1 sniffer) branchés à la machine réalisant ceux-ci. Le sniffer réalise la plupart des tâches purement offensive, 2 dongles mettent en place le réseau test pendant que les 2 autres permettent l'usurpation d'identité.

## 6.5 Livrables

Code source du système fonctionnel: comprend l'intégration de l'outils offensive, le serveur et client pour l'interface ainsi qu'un moyen de déployer le système (Docker).

Documentation du système: rédigée en langage spécifique (markdown, rst) et déployable avec un outils (Sphinx, pandoc), documentation développeur pour mettre en place le système et documenter les choix techniques.

Rapport de projet: rédigé avec un outils spécifique (LaTeX, pandoc), rendue au format PDF, comprend une étude du contexte, analyse de l'existant et de faisabilité puis mise en place de la preuve de concept.

## 7 Preuve de concept

### 7.1 Sniffing

### 7.2 Localization

#### Fingerprinting

A partir d'une liste de beacons et leurs position, calcul la position se rapprochant le plus d'un des beacons (a partir du RSSI).

Demande de pouvoir etabli la liste des beacons et les identifie de facon sure. Si le systeme est mit en place pour cet effet on s'assurera qu'ils soient identifiables (MAC unique par exemple) mais dans notre cas de recuperation d'information, les appareils peuvent mettre en place des mesures contre le tracage comme la generation d'adresse mac aleatoire. Il est possible d'utiliser le profile GATT pour identifier un appareil, combiner avec le RSSI dans le temps et les déplacements (capteurs) on peut esperer distinguer deux profils GATT identiques.

~ beacons coverage

Le beacon le plus proche

#### RSSI / TOA

~ m

Trilateration determines the position of an object by understanding its distance from three known reference points. In the case of Bluetooth, locators estimate their distance to any given asset tag based on the received signal strength from the tag

#### AOA / AOD

~ cm

Basee sur le nouveau systeme d'angle du BLE 5.1 Demande du materiel en plus (Multiple antennes directionnelles pour former une matrice) Differentes facon de calculee (angle arrivee, angle depart ...)

<https://www.bluetooth.com/blog/bluetooth-positioning-systems/> [https://www.bluetooth.com/bluetooth-resources/enhancing-bluetooth-location-services-with-direction-finding/?utm\\_campaign=location-services&utm\\_source=internal&utm\\_medium=blog&utm\\_content=bluetooth-positioning-systems](https://www.bluetooth.com/bluetooth-resources/enhancing-bluetooth-location-services-with-direction-finding/?utm_campaign=location-services&utm_source=internal&utm_medium=blog&utm_content=bluetooth-positioning-systems)

#### Ajouter de la precision

Fusionner les resultats avec un filtre kalmann:

- dead reckoning
- trilateration / triangulation

Ou RSS (range) + AOA (direction)

**RSS**

1. Scan devices BTLEJack sniffer
2. find settings (rssi, txPower / measured power ...) Tx Power service 0x1804 and Tx Power Level Characteristic 0x2A07
3. calculate distance (in a circle around you)  $10^{\frac{txPower - RSSI}{10 * N}}$  N = loss factor (between 2 and 4), 0 for optimal conditions
4. cross multiple references to determine a position (trilateration) repeat 3 times to 3 devices get OUR position

**AOA****7.3 MITM**