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

List of Tables

List of Figures

Dans le cadre du master CSSE nous etudions l'internet des objets (IoT) et leurs aspects securite. Le Bluetooth Low Energy (BLE) est une specification du protocole Bluetooth pour les objets fonctionnant sur batterie, visant notamment les objets connectes.
Standardise, gratuit et integré dans la plupart des appareils de bureautique (laptop, smartphone) il est rapidement devenu populaire dans l'internet des objets.
La première itération du BLE est principalement un portage du procole Bluetooth vers une couche pysique *Low Energy*. Celle-ci intègre des mesures de sécurité aujourd'hui desuètes et manque de fonctionnalités (topologies autres que point à point, localisation précise). Même si le protocole à su évoluer depuis pour repondre à ces besoins, beaucoup d'appareils *première génération* utilisent encore la version originale n'intégrant pas ces mécanismes.
Ce sont pour la plupart des appareils conçus pour fonctionner en point à point avec un smartphone ou ordinateur comme les montres connectés, les capteurs corporels fitness, les termomètres, serrures ou cadenas, etc. Les données personnelles peuvent etres interceptées et les actions modifiées (ouverture de cadenas, par exemple).

1 Objets connectés

Avec l'explosion de l'internet de objets (TODO chiffres) toute une floppée d'objet du quotidien ont étés augmentés pour permettre la communication avec d'autres systemes informatique dont nos smartphones ou encore des serveurs distants (via notre réseau privé). Ces objets dits intelligents étendent leur équivalent mécanique en intégrant des composants éléctroniques, 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 securite plus en plus pris en compte mais secondaire tj

1.1 Architecture

Point à point

Architecture reseau domotique

- simple: appareil non relie au reseau, dependant gateway utilisateur, remplissant une fonction d'augmentation seul (smart lock)

Réseau

- avancee: appareil s'appuyant sur un reseau domotique pour realiser ses fonctions, relie a une gateway "sure" hub

1.2 Protocoles

Protocoles generaux supportes par tout appareil (smartphone notamment) et peu cher WiFi (WLAN) ~50m: Local = remplace cables pour appareils fixes dans pieces / appart BLE (WPAN) ~10m: Personnal = remplace cables pour appareils portable personnels NFC

Protocoles specifiques concus pour ces reseaux Zigbee Zwave Thread ANT(+)

1.3 Marché

Premiere generation point a point "smart"

Seconde generation networks IoT

BLE

Gadgets (fitness)

Domotique (IoT)

Entreprise / warehouse / smart city (beacons)

2 Bluetooth Low energy

multi bande master - slave packet based

2.1 Stack

2.2 Finding

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.bluetooth.com/specifications/assigned-numbers/generic-access-profile/

2.3 pairing

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

phase 2 key generation method

JustWorks

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

2.4 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)

2.5 BLE

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 ?

Flaws

Downgrade “SC” The SC field is a 1-bit flag that is set to one to request LE Secure Connection pairing. The possible resulting pairing mechanisms are if both devices support LE Secure Connections, use LE Secure Connections and otherwise use LE legacy pairing. So this flag is an indicator to determine Phase 2 pairing method.

\newpage{}

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 deja sur la machine ou dongle USB pour une etude du traffic interne et des appareils emettants des adv.

Interceptions des communications

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

Proprietaires:

- nRF sniffer
- nRF Connect
- smartRF (TI)

Attaques

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

4.2 Materiels

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 Presentation

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'implementer 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 implementer 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 eligibles:

- 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 qui sont décritent ci-après.

Repérage

Inventaire des appareils et connexions BLE à proximité.

- Écoute des annonces sur les 3 canaux publicitaires pour récuperer 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, établissment d'une connexion pour récuperer 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 precedement déconnectés.
- Récupération des informations cryptographique pour déchiffrer la connexion seulement si celle-ci n'utilise pas une clef a long terme deja établie ou une connexion securisée (BLE 4.2).
- Écoute des communications et déchiffrement des trames à la volée.

Modification

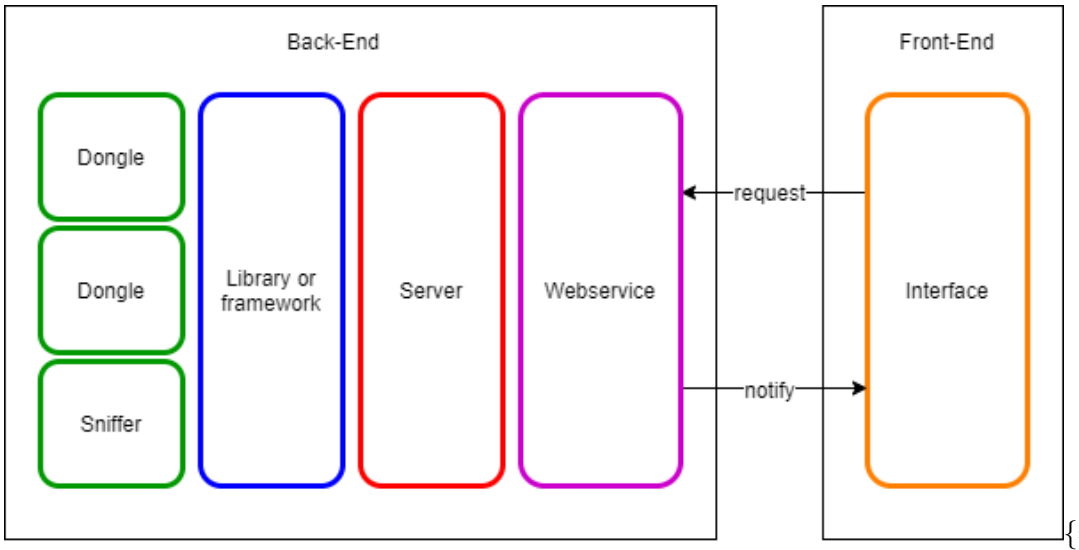
Attaque *man in the middle* par clonage et usurpation d'un appareil BLE pour modifier les données echangées.

- Écoute passive des annonces de l'esclave cible de l'usurpation pour retransmission ultérieur 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 précédement 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 implementées.
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 framwork 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.



{#fig:poc-arch width=85%}

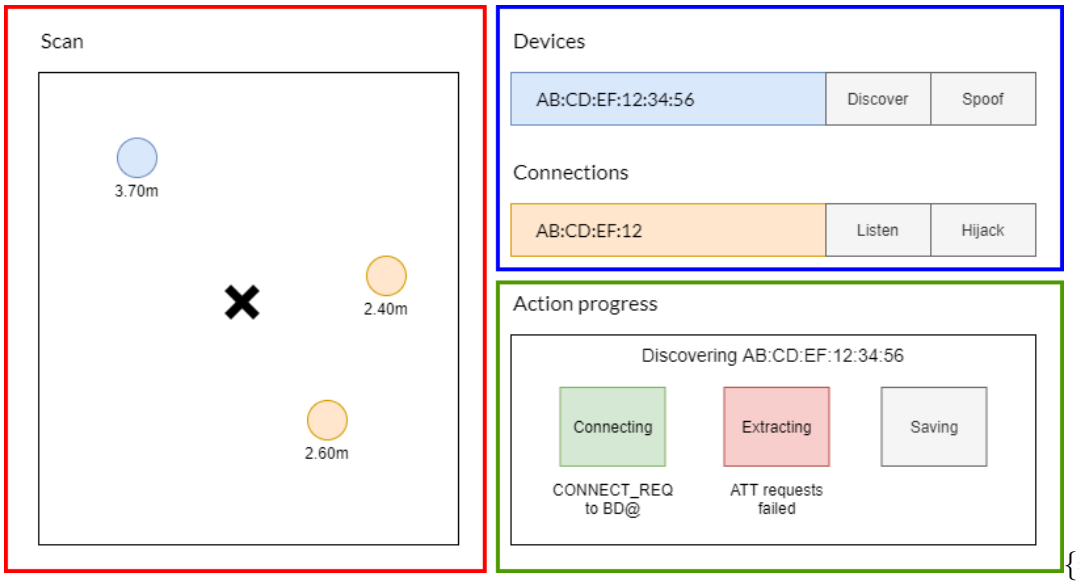
6.3 Interface

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 transimissions par usurpation pour un appareil BLE emettant des annonces (zone bleue *Devices*).
- Déconnexion des appareils ou interception des communications entre deux appareils appairés (zone bleue *Connections*).

Une troisieme section permet de suivre le déroulement de l'attaque chosie (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).



{#fig:poc-

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 executée 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 tàches 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 etablir la liste des beacons et les identifies 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 deplacements (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

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 ^ ((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