Master 1 Informatique

PJI - Projet Individuel - Sujet no 104

Systèmes de détection d'intrusion pour l'Internet des Objets

Auteurs :
M. Théo Plockyn
M. Rémy Debue

Encadrant :
Pr. Gilles Grimaud

Version 0.5 du
2 mai 2016

Remerciements

Nous remercions tout d'abord l'équipe pédagogique, administrative et intervenants du Master 1 informatique de nous avoir encadré, aidé et assuré les enseignements dont nous avons disposé cette année.

Nous tenons aussi à remercier et à témoigner notre reconnaissance aux personnes suivantes :

Gilles Grimaud, notre encadrant, pour nous avoir proposé le sujet, nous avoir suivi et conseillé tout au long de ce projet.

Michaël Hauspie, pour ses consignes et sa participation dans les décisions du déroulement du projet.

Nadir Cherifi, pour son aide précieuse et ses connaissances des technologies utilisées qui nous ont débloqué à plusieurs reprises.

Samuel Hym, François Serman, Christophe Bacara, Quentin Bergougnoux, et toute l'équipe 2XS pour leur accueil sympathique et leur soutien tout au long de ce projet.

Table des matières

Table des figures

Liste des sigles et acronymes

6LoWPAN	*IPv**6** **Lo**w power **W**ireless **P**ersonal **A**rea **N**etworks*
LoWPAN	***Lo**w power **W**ireless **P**ersonal **A**rea **N**etworks*
IRCICA	**I**nstitut de **r**echerche sur les **c**omposants logiciels et matériels pour l'**i**nformation et la **c**ommunication **a**vancée de Lille
2XS	*e**X**tra **S**mall e**X**tra **S**afe* – L'équipe de recherche
CFS	***C**offee **F**ile **S**ystem* – Le système de fichier de Contiki
DSL	***D**omain **S**pecific **L**anguage* – Langage dédié
IDS	***Intrusion Detection System*** – Système de Detection d'Intrusions
PJI	**P**roj**j**et **i**ndividuel
RFC	***R**equest **F**or **C**omments* – Documents de spécifications
OS	***O**perating **S**ystem* – Système d'exploitation

Introduction

Dans le cadre de notre cursus en Master Informatique à Lille 1, nous avons eu lopportunité de réaliser un projet sur lensemble du semestre appelé PJI. Chaque étudiant ou binôme pouvait choisir un sujet sur lequel travailler parmi une liste mais également proposer le sien. Nous avons choisi de nous intéresser à un sujet proche de l'informatique embarquée, plus particulièrement dans le domaine de l'Internet des Objets. Notre sujet se porte sur la détection d'attaques dans un réseau 6LoWPAN.

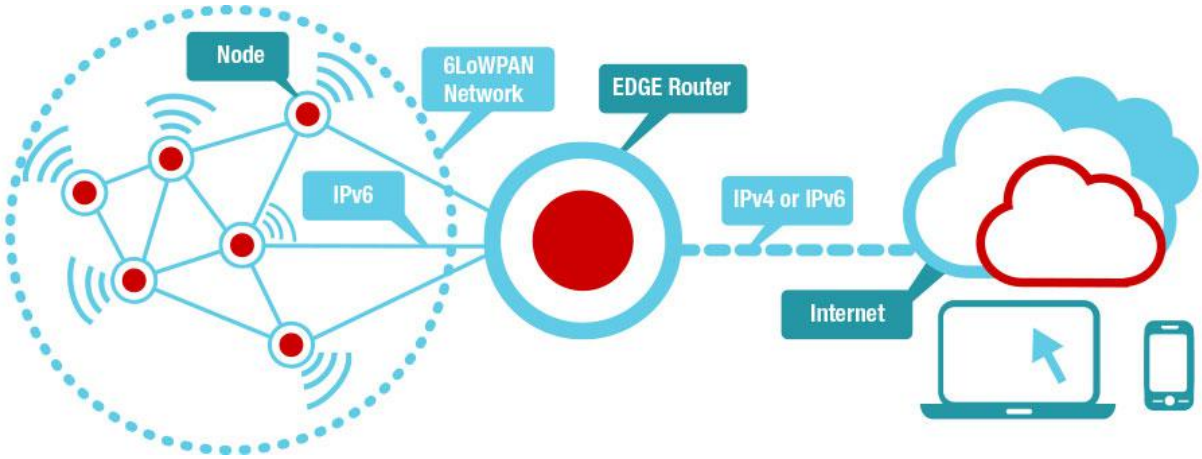

FIGURE 1 – Diagramme d'explication de 6LoWPAN.

L'équipe de recherche proposant ce sujet est le groupe 2XS **eXtra Small eXtra Safe** composée de notamment **Gilles GRIMAUD** notre encadrant, **Michael HAUSPIE** son collègue proche de ce sujet et bien sûr le reste de l'équipe. L'équipe se focalise sur les problématiques de sécurité dans les systèmes embarqués contraints, notamment fournir des solutions logicielles prouvées.

Chapitre 1

Contexte du sujet

Notre projet est de produire une sonde qui renifle le trafic réseau dans le contexte de l'Internet des Objets. Cette sonde est un noeud dans ce réseau, et a pour but de transmettre des informations utiles à la sécurisation du réseau, et dans une certaine mesure à l'analyse de ces informations.

1.1 Analyse de l'existant

1.1.1 Internet des objets

L'Internet des Objets, ou Internet of Things en anglais, correspond à l'extension d'internet aux éléments ou lieux du monde physique, là où l'internet habituel s'arrête au domaine du virtuel. Cette technologie est implantée dans notre société avec diverses applications comme la domotique, le médical, la gestion des déchets, mais pas limité à ceux là. Notre projet s'inscrit donc dans cet univers puisque les sondes surveillent le trafic de différents éléments d'un sous-réseau d'objets physiques, d'un bâtiment par exemple.
L'Internet des objets regroupe différents modes de communications entre les nuds d'un réseau tels que le Wi-Fi, le courant porteur ou le bluetooth.

1.1.2 Technologies de communication

6LoWPAN est une spécification du principe des LoWPAN, c'est à dire un ensemble d'équipements aux ressources limitées, puissance, autonomie entre autres, reliés dans un réseau au débit limité. Typiquement, ces réseaux sont constitués d'un grand nombre d'éléments ou nuds dans le réseau. Basé sur l'IPv6, quelques problèmes se posent avec la spécification standard de celui ci. Ce protocole de communication possède une taille d'entête importante, couplée aux contraintes de tailles de paquets imposées, cela pose des soucis de

fragmentation et de réassamblage excessif pour des contrôleurs aux capacités limitées.
La spécification de 6LoWPAN et ses RFC (4919 et 4944) définissent donc des solutions à ces problèmes, et on peut aujourd'hui utiliser plusieurs implémentations de 6LoWPAN, tel que ZigBee. Linky, le nouveau compteur communicant d'ERDF utilise cette technologie. Bien sûr, on peut trouver pléthore de projets et d'objets de domotique se servant de la spécification et de ses implémentations pour communiquer.

1.1.3 Sécurité des communications

Les différentes RFC définissent un ensemble de consignes sur l'implémentation de la sécurité des réseaux 6LoWPAN notamment sur les différentes couches de la pile protocolaire.

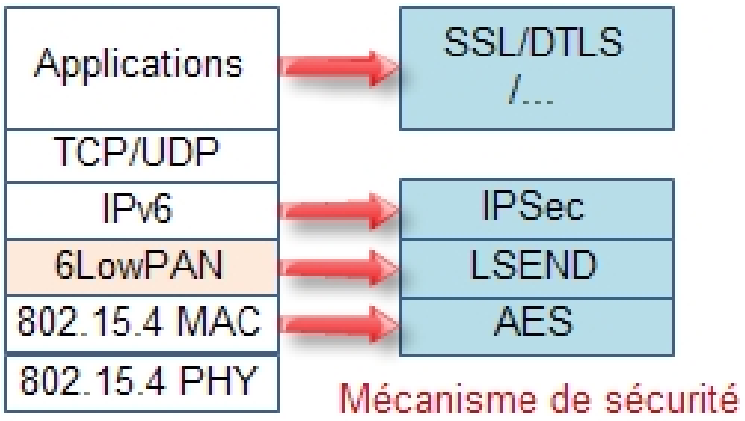

FIGURE 1.1 – Diagramme d'explication de la sécurité des couches de 6LoWPAN.

— Sur la couche MAC : l'algorithme AES doit être utilisé pour sécuriser la couche liaison.
— Sur la couche réseau : l'utilisation d'IPsec est possible, mais coûteuse et un échange de clés habituel n'est pas possible. Une extension du protocole SEND – ***SE****cure* ***N****eighbor* ***D****iscovery protocol* – (RFC 3971) permettant de sécuriser ce mécanisme a été mis en place pour les réseaux 6LoWPAN, appelé LSEND – ***L****ightweight* ***SE****cure* ***N****eighbor* ***D****iscovery protocol* –.
— Sur la couche application : une solution possible est de mettre en place la sécurisation via SSL.

Dans les faits, la sécurité étant difficile à mettre en place à cause des contraintes de l'embarqué, elle est parfois insuffisante pour garantir un échange de données sécurisé.

1.2 Objectif du projet

1.2.1 Détail du projet

Nous avons expliqué l'intitulé du projet mais non pourquoi ces sondes peuvent être utiles. Les sondes sont des noeuds, ou mote dans le jargon de Contiki, qui vont renifler et analyser le trafic circulant, car les informations qu'elles récupèrent permettent la détection d'intrusions.
En effet, le trafic circulant et les paquets sont soumis à des formats spécifiques, contenant des informations qui doivent s'y conformer, ou au contraire faire des écarts vis à vis du format, ce qui constitue une anomalie, et possiblement une attaque.
La difficulté de cette analyse réside autant dans les contraintes du matériel, qui est limité en puissance et en mémoire, que dans les solutions mises en places par le réseau pour pallier à ces contraintes, par exemple la compression d'entête.

1.2.2 Où s'inscrit le projet ?

Comme illustré précédemment, l'Internet des Objets trouve son utilité dans de nombreux domaines. Certains domaines, comme l'industrie ou le médical, font circuler des données sensibles sur le réseau, et des personnes mal intentionnées pourraient causer de graves problèmes sans être détecté si le réseau n'est pas protégé.
Un exemple un peu plus léger est l'histoire du soi-disant hacker qui avait changé le nombre de places des panneaux de parking par des injures. La personne a simplement usurpé l'identité de la machine qui met à jour les places, et a envoyé des paquets falsifiés contenant ces injures.

1.3 Réponse à un besoin de l'équipe

1.3.1 Focalisation sur la sécurité par 2XS

L'équipe de recherche 2XS se focalise sur la création de systèmes sûrs, par le biais de différents mécanismes, dont les preuves formelles de programmes, le développement de librairies, et bien d'autres projets visant à sécuriser les systèmes.

Mais la problématique de la sécurité ne s'arrête pas à la frontière du système en lui-même, il communique avec d'autres. C'est là que les failles et les fuites d'informations sont les plus nombreuses, même si l'intégrité du système n'est pas en cause.
L'un de leurs projets pour répondre à ce problème est Discus.

1.3.2 Discus

Discus est une architecture d'IDS – Système de détection d'intrusion – massivement distribuée, qui est configurée grâce à un DSL – Langage dédié – Discus-script. Le principe de Discus est d'abstraire la définition des contraintes de sécurité sur un réseau, qu'il soit ethernet, bluetooth ou Wi-Fi.
Notre projet est donc directement en rapport avec celui-ci, fournissant la couche matérielle nécessaire à Discus pour analyser le réseau afin d'y appliquer ces contraintes de sécurité.

1.4 Technologies et systèmes utilisés

Pour développer notre sonde renifleuse, nous avons utilisé plusieurs outils que nous allons présenter ici :

1.4.1 Contiki

Contiki est un système d'exploitation léger et flexible avec pour cible les capteurs miniatures en réseau. Ses atouts sont sa flexibilité, sa portabilité, sa faible consommation énergétique, et surtout dans notre cas, son support des protocoles IPv6 et 6LoWPAN. Il répond à une attention importante de la communauté scientifique portée aux réseaux de capteurs sans fil. Il a été créé par une équipe de recherche du centre suédois de recherche scientifique SICS.

1.4.2 Outils de simulations

Notre sonde a été créée dans un environnement de développement fourni par le site officiel de Contiki, la machine virtuelle InstantContiki3.0, en utilisant principalement pour les tests l'outil de simulation Cooja.
Cooja est un simulateur de matériel pour Contiki permettant de créer virtuellement un réseau de capteurs, de les positionner à notre envie et de charger les différents programmes pour les noeuds (ou motes dans le jargon Contiki) à la volée. Nous avons donc passé beaucoup de temps à l'utiliser pour tester notre programme dans des situations réalistes.

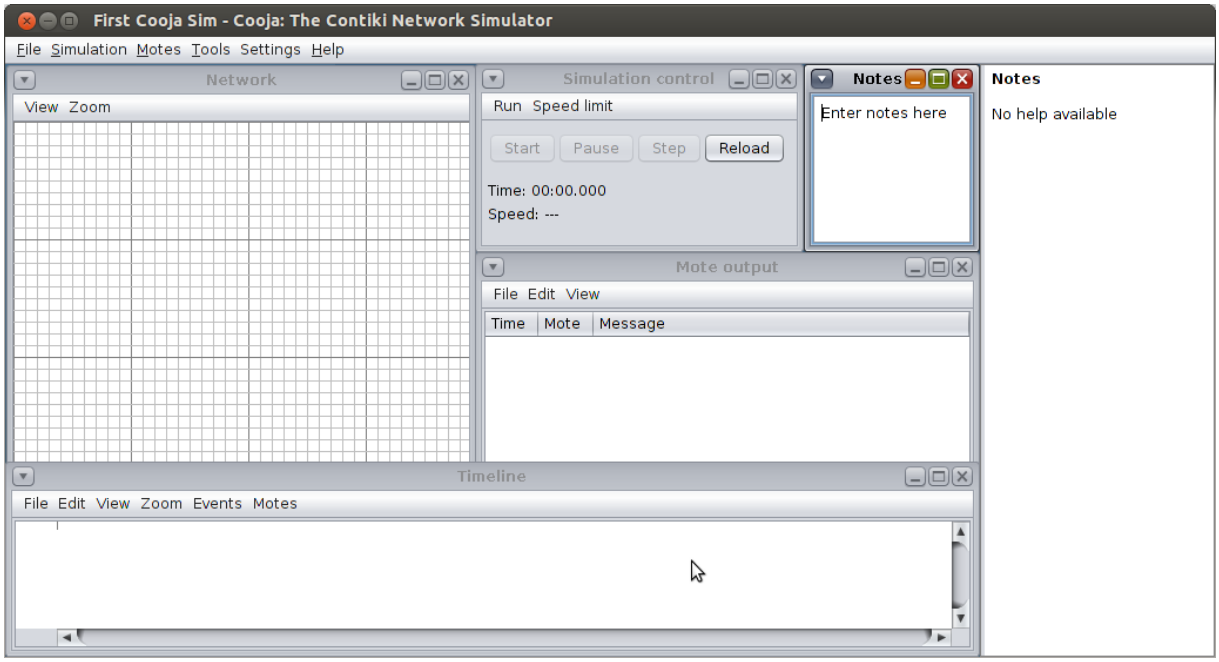

FIGURE 1.2 – Capture d'écran de Cooja.

1.4.3 Langage C embarqué et sa chaîne de compilation

La création de la sonde s'est fait sur Contiki et le programme a donc dû être adapté aux contraintes du matériel pour lequel il est créé. Pour cela, nous devons rendre le code le plus léger et proche du matériel, et cela passe par l'absence des librairies standard du langage C, par exemple stdlib ou unistd.
La compilation des programmes se fait avec des versions de GCC spécifiques aux architectures matérielles que nous utilisons.

1.4.4 Git

Git est un gestionnaire de version de projet. Celui-ci permet de synchroniser le travail de notre binôme.
Nous avons choisi d'utiliser la plateforme GitHub pour accueillir notre dépôt Git, afin de faciliter l'accès à notre code.

Chapitre 2

Explications techniques

2.1 Contiki

2.1.1 Pile réseau de Contiki

2.1.2 Systèmes de stockage Contiki

Contiki file system

- Expliquer comment écrire dedans
- Expliquer pourquoi on ne l'a pas utilisé

Volatile

Listes

- Expliquer comment écrire dedans
- Expliquer pourquoi on ne l'a pas utilisé

Buffers cycliques

- Expliquer comment écrire dedans
- Expliquer pourquoi on l'a utilisé

2.2 6LoWPAN

2.2.1 Compression des headers

- Pourquoi compresser ?
- Comment ça marche (avec images)

2.2.2 Attaques possibles

- Qui nous intéressent pas directement
  - Attaques passives (écoute)
  - Brouillage des ondes
  - Inondation de paquets
- Qui nous intéressent
  - Spoofing
  - Paquets dupliqués
  - Paquets fabriqués
  - Sybil attack

Nos possibles solutions du coup

Chapitre 3

Déroulement du projet

3.1 Prise en main du sujet et des technologies

Après ces explications techniques, il est temps d'expliquer comment le projet s'est déroulé pour nous. Au début, il nous a fallu prendre en main les différents outils pour travailler.

3.1.1 Contiki

Comme nous avons déjà vu, Contiki est un système d'exploitation qui vise les systèmes embarqués. Notre expérience avec l'embarqué était limitée bien que non-nulle, aussi la prise en main s'est faite assez rapidement, même si difficilement.
Certains concepts, comme les proto-threads utilisés par Contiki, sont assez proches d'autres concepts présents dans les systèmes d'exploitation habituels, néanmoins la découverte des fonctionnalités de Contiki a pris du temps car il est complet et offre les principales caractéristiques et fonctionnalités d'un système habituel.
Ceci est l'une des raisons, avec celles déjà évoquées pour IPv6 et 6LoWPAN, du choix de Contiki plutôt que FreeRTOS ou TinyOS.

3.1.2 Chaîne de compilation et pilotes

Les découvertes ici ont été plutôt rapides puisque la machine virtuelle InstantContiki3.0 contient la chaîne de compilation nécessaire à la compilation des projets sur les différentes architectures supportées par Contiki. L'installation à la main de la chaîne de compilation est bien documentée sur les différents tutoriels concernant Contiki présents sur Internet.
Les pilotes des différents contrôleurs radios sont par contre difficiles à prendre en main,

aussi avons nous choisi de nous concentrer sur les contrôleurs cc2420 de Texas Instrument car déjà présent dans d'autres projets que nous avons passés en revue.

3.2 Programme développé

3.2.1 Fonctionnement du projet

3.2.2 État du projet

3.3 Retours d'expérience

Pendant ce projet, nous avons pu apprendre beaucoup de choses, développer nos compétences et de produire un programme, dont qui peut bénéficier d'évolutions sur le court et long terme.

3.3.1 Évolutions à court terme

A court terme, il est évidemment possible d'ajouter d'autres détections d'attaques en fonction des besoins, mais ce qui est le plus utile à l'équipe de recherche est d'intégrer notre sonde dans Discus, leur système de détection d'intrusion. La sonde peut fournir les informations dont Discus a besoin pour faire respecter les contraintes énoncée dans le script adéquat. Les vérifications d'attaques sont donc effectée par le système qui reçoit les informations, et non par les sondes directement, ce qui allège la charge de travail sur les capteurs aux capacités restreintes. Aussi, les nouvelles vérifications d'intrusion pourront s'écrire et se faire sans reprogrammer les sondes.

3.3.2 Évolutions à long terme

Sur un plus long terme, il a été pensé d'éventuellement faire communiquer les sondes entre elles afin de créer une grille de capteurs. Cela permettrait, grâce aux différents RSSI pour un seul paquet captés par les sondes, de localiser les différents acteurs du réseau, et donc de localiser l'attaquant lors d'une anomalie.

3.3.3 Challenges

Ce projet a été très intéressant et instructif, mais nous avons dû faire face à plusieurs difficultés et challenges lors de son déroulement. Bien que ces contraintes aient pu parfois

nous ralentir, elles furent instructives à plusieurs niveaux.
Tout d'abord, le sujet du projet se concentre sur des domaines et technologies dont nous n'étions pas très familiers. Le monde de l'informatique embarquée est fait de contraintes auxquelles il faut s'adapter pour être productif, les retours beaucoup moins verbeux lors d'erreurs, les limites de mémoire, de puissance, et parfois l'absence de librairies pour rendre le code assez léger pour la plateforme sont quelques exemples de difficultés lorsqu'on découvre l'embarqué. S'y adapter n'est pas un obstacle en soi, mais il faut bien se préparer et ne pas avoir peur de prendre son temps pour cela.
Les découvertes étaient nettement plus nombreuses dans les technologies employées, notamment au niveau des systèmes d'exploitation embarqués, comme Contiki, et leurs technologies de communication. Nous avons lu beaucoup de spécifications, de RFC et de documentation, et en rétrospective, nous aurions eu plus de facilité à établir un plan d'approche, organiser nos découvertes pour éviter la confusion. Par exemple, prendre du temps pour bien se renseigner sur Contiki, puis lorsque l'outil est maîtrisé, se renseigner sur 6LoWPAN, et ainsi de suite.
Malgré nos recherches en profondeur, nous sommes parfois tombés sur des incohérences dans la documentation, ou des explications pas assez claires, notamment sur le buffer de paquets qui traite différemment les paquets réseau selon qu'ils soient entrants ou sortants. Pour pallier à ce souci, nous avons recherché plus d'informations sur des sites et des encyclopédies en ligne (wikis) universitaires qui se sont penchés sur Contiki et ont écrit de bons tutoriels.
Parfois, certains exemples de code présents dans Contiki ne sont pas assez commentés, ce qui peut être problématique pour apprendre comment certains programmes sont faits. C'est pourquoi il ne faut pas hésiter à rechercher sur des forums des discussions traitant de ces sujets et à contacter certains interlocuteurs pour demander de l'aide.

Conclusion et perspectives

Notre sujet de projet était de contribuer à un système de détection d'intrusion pour l'Internet des Objets. La principale difficulté de ce sujet était les nombreuses attaques possibles sur le réseau, notamment le détournement de routage pour l'écoute, le vol d'identité et la falsification de paquets. En partant de cette problématique, nous avons produit une sonde, un noeud sur le réseau qui écoute le trafic radio, afin de construire par dessus plusieurs couches de détections.

De ce fait, nous procurons à l'équipe de recherche un outil de base pour développer la sécurité réseau dans l'IDS de l'équipe – Discus, en rajoutant la plateforme radio aux autres déjà existantes.

En travaillant sur ce projet, nous avons pu découvrir des technologies que nous n'avons pas l'habitude de voir lors de nos cours. Cela nous a permis d'élargir nos horizons vis à vis des domaines de l'informatique, de tester de nouveaux paradigmes.
Nous avons pu nous améliorer sur notre travail en autonomie, nous entraîner à la recherche d'information, de documentation. Cet entraînement est aussi un regard vers ce qu'est le monde de la recherche, qui a permis de le découvrir ou le redécouvrir.