



INSTITUT UNIVERSITAIRE DE TECHNOLOGIE DE LA ROCHELLE

Département Informatique

-----

Stage de fin d'études

Étudiant

Clément Lavergne – Informatique Embarquée

Fonction

Stagiaire chercheur

Titre du sujet

Période de stage du 17/04/2017 au 23/05/2017

Entreprise

National University of Ireland Galway,  
University Road,  
Galway, Ireland

Enseignant tuteur: M. BERNDT

Maître de stage: M. SCHUKAT



Niveau de confidentialité : Aucun

Année Universitaire 2016-2017





INSTITUT UNIVERSITAIRE DE TECHNOLOGIE DE LA ROCHELLE

Département Informatique

-----

Stage de fin d'études

Étudiant

Clément Lavergne – Informatique Embarquée

Fonction

Stagiaire chercheur

Titre du sujet

Cybersécurité dans le domaine de l'embarqué

Période de stage du 17/04/2017 au 23/05/2017

Entreprise

National University of Ireland Galway,  
University Road,  
Galway, Ireland

Enseignant tuteur: M. BERNDT

Maître de stage: M. SCHUKAT



NUI Galway  
OÉ Gaillimh

Niveau de confidentialité : Aucun

Année Universitaire 2016-2017

Merci à :

M.Shukat Mickael pour son accueil chaleureux et l'ambiance de travail qu'il nous a offert

Brûler Antoine d'avoir partagé ces 10 semaines avec moi ainsi que pour les petites aides techniques

Eboueya Nadine, de permettre ce genre d'expérience à l'étranger

Aït Yachou Bastien pour son soutien outre manche

# Table des matières

Introduction.....	6
Présentation.....	7
L'université et le groupe.....	7
Outils techniques à notre disposition.....	8
Présentation globale du système à mettre en place.....	9
Planning prévisionnel et réel.....	13
Sécuriser les communications : VPN.....	14
Introduction à la cybersécurité : clés et certificats.....	14
Définition d'un VPN.....	15
Pourquoi un VPN dans ce système.....	16
Implémentation du VPN.....	16
Conclusion partielle.....	17
Mettre le système à jour.....	17
Définition et avantages d'un SSR.....	18
Connexion au SSR par le VPN.....	18
Mise à jour automatique.....	19
Conclusion partielle.....	19
Protection envers le réseau.....	20
Qu'est qu'un NIDS.....	20
Pourquoi notre système requiert un NIDS.....	21
Implémentation.....	21
Conclusion partielle.....	23
Réflexion sur le projet.....	24
Lien avec les matières.....	24
Apports techniques.....	24
Apports méthodologiques.....	25
Ressenti personnel.....	25
Conclusion.....	27
Glossaire.....	28
Bibliographie.....	29
Tables des Annexes.....	30
Résumés et mots clés.....	31
Mots Clés.....	31

# Introduction

Actuellement les objets connectés deviennent grand-public et commencent à largement se démocratiser avec la chute de leur prix et leur miniaturisation. Ces objets qui se veulent souvent minimalistes changent notre vision de l'informatique, en effet, la tendance actuelle est d'aller vers des machines toujours plus puissantes, ces objets sont à contre-courant de ce phénomène leur objectif étant d'être petit, leur puissance en est directement impactée. Quand une machine est moins puissante les protocoles utilisés pour (par exemple) le transport de données vont être également plus simple ce qui entraîne parfois des failles de sécurité. Le monde industriel fait face à ce genre de problématiques depuis un peu plus longtemps que le grand public, avec l'utilisation d'appareils répondant aux critères d'objets connectés actuels depuis déjà plus de 10 ans. C'est dans cette optique qu'un groupe de recherche nommé OSNA<sup>1</sup> s'est formé au sein du département informatique de l'université de Galway, ce groupe est composé d'enseignants, de chercheurs et d'intervenants qui travaillent sur des solutions de cybersécurité pour l'industrie.

Ma soif de voyage et de découvertes culturelles m'ont poussées à choisir d'effectuer mon stage à l'étranger, mon intérêt envers les systèmes embarqués m'a tout simplement guidé vers le stage proposé par l'université de Galway. C'est ainsi que j'ai choisi de travailler pour 10 semaines en tant que chercheur au sein du groupe OSNA. Ma mission principale consistait à sécuriser la partie embarquée d'un réseau de capteur communiquant avec un serveur qui centralise les données collectées.

Quelles sont les types de menaces pesant sur un réseau industriel ?

Comment immuniser un réseau soumis à des sources de menaces multiples ?

Dans un premier temps nous présenterons le stage en lui-même puis, nous étudierons pourquoi et comment effectuer la connexion entre les principaux éléments du système à l'aide d'un VPN<sup>2</sup>. Ensuite, nous nous intéresserons à la sécurisation du point de vue de l'obtention de paquets pour les installations et les mises à jour de notre système, à l'aide d'un dépôt privé et sécurisé. Nous continuons en nous penchant sur les techniques mises en place pour la protection envers le réseau. Pour terminer j'exposerai ma réflexion sur le projet, avant de conclure.

---

1 Open Sensor Network Authentication traduction : Authentification pour réseau ouvert de capteur

2 « virtual private network » à traduire par « réseau virtuel privé. »

# Présentation

## L'université et le groupe

L'université fût fondée en 1845 sous le nom de « Queen's College Galway » avec la construction du tout premier bâtiment, elle ouvrit ses portes en octobre 1849 pour seulement 68 étudiants sous la direction de Dr. Joseph W.Kirwan, à cette époque y était enseigné 3 cursus : Art, Médecine et droit, il y avait également une école d'agriculture et une d'ingénieur. L'université changera deux fois de nom, une fois en 1908 pour « University College Galway » et une dernière fois en 1997 pour « National University of Ireland Galway » ou « NUI Galway ». À partir des années 1990 le campus va largement s'agrandir avec la conversion de locaux adjacents mais aussi, grâce à la construction de nouveaux bâtiments pour arriver au campus actuel. Aujourd'hui la NUI Galway est classée parmi les 2 % des meilleures universités au monde (rang 249), on y enseigne plus de 50 disciplines, réparties dans cinq facultés dont celle d'ingénierie et d'informatique où nous avons travaillé.

Aujourd'hui la NUI Galway est très attractive, on y compte 12 % d'étudiant étrangers venant de 110 pays différents. Les locaux dans lesquels nous travaillons sont assez vastes mais nous savons à qui nous adresser. On y retrouve deux secrétaires qui peuvent nous aider pour tout ce qui touche à l'administratif, les bureaux de M. Schukat et M. Marvin qui dirigent notre stage et qui peuvent nous aiguiller pour tout ce qui touche de près à notre projet. Enfin deux techniciens travaillent également à notre étage pour nos questions touchant au matériel.

Mon projet prend part au sein du groupe de recherche nommé OSNA<sup>3</sup> dont l'objectif est de développer des réseaux de capteurs sécurisés permettant un traitement des données en toute sécurité à l'aide (tant que possible) d'outils open source<sup>4</sup>.



Dans le domaine industriel on voit de plus en plus émerger de nouveaux systèmes à base de capteurs qui communiquent sans fil au sein d'une entreprise. Plus récemment de grands groupes travaillant à l'aide de ces outils ont pointé du doigt un problème majeur : les risques de cyberattaque sur ce genre de réseau sont très présent et peu ou pas de solution sont en place contre cela. Pour

3 Open Sensor Network Authentication traduction : Authentification pour réseau ouvert de capteur

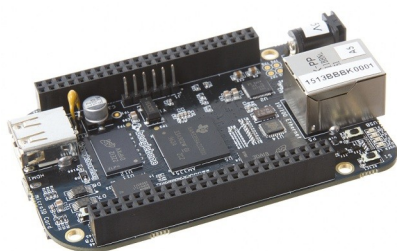
4 Un logiciel est open source lorsque la licence qui y est rattachée autorise l'étude, les modifications et la distribution du logiciel, ainsi tout (ou une partie, suivant les cas) le code source et en libre accès.

faire simple les communications entre machines sont rarement cryptées et lorsqu'elles le sont, la sécurité n'est pas optimale, de plus, rare sont les systèmes d'authentifications fiables en place dans ce type de réseau. Ces lacunes en terme de protection font que l'on ne peut garantir ni la confidentialité, ni authenticité des données qui circulent ce qui peut avoir de graves conséquences.

Le groupe de recherche est dirigé par le Dr Michael Schukat, mon maître de stage lequel est professeur à l'université et chercheur en cybersécurité pour les systèmes embarqués notamment dans le domaine médical. En plus du Dr Hugh Melvin avec qui nous avons été amené à échanger, il y a 10 autres personnes qui travaillent dans ce groupe mais nous n'avons pas eu à entre en contact. La hiérarchie au sein de ce groupe est extrêmement simple et complètement horizontale, en effet, outre le fait que le Dr Michael Schukat soit à la tête du groupe chaque intervenant, peu importe son origine est traité de la même manière : il doit donner de son temps pour faire avancer les projets sans notions de supérieur ou de responsabilités.

## Outils techniques à notre disposition

### Le BeableBone Black



Le BeableBone Black est l'outil principal de mon stage, toutes les tâches que j'ai eues à réaliser devaient l'être sur cet outil. Un BeableBone Black est une carte de développement sur laquelle l'on va utiliser une distribution de Linux. La carte possède les principales caractéristiques d'un ordinateur classique à savoir : un système d'exploitation (Linux), une carte réseau ainsi un port Ethernet pour se connecter à internet, un port HDMI pour avoir un retour sur un écran (qui n'est pas utilisé dans le projet) et un port USB. Les deux vraies différences avec un ordinateur classique sont néanmoins frappantes : la taille (10cm x 5cm), et la puissance, en effet, pour un prix tournant autour de 50€ la puissance de ses cartes est assez limitée (comparé à un ordinateur classique), c'est exactement les caractéristiques d'un système embarqué.

### Le serveur « rack »



Le serveur rack (ou serveur pour baie de serveur) est la partie du projet dont Antoine a la charge. Il s'agit d'un serveur qui a sa place au sein d'une baie de serveur (voir image). Il s'agit d'un type de machine assez particulier, ces serveurs sont beaucoup plus puissants qu'un ordinateur ordinaire car ils ont



en général la charge de nombreuses tâches plus ou moins complexes au sein d'un réseau. Ce serveur aura la charge de centraliser toutes les informations de notre système qui viendront des différents BeableBone Black, informations qui devront ensuite être mises à la disposition de l'utilisateur pour obtenir une vision globale de l'état du système en direct.



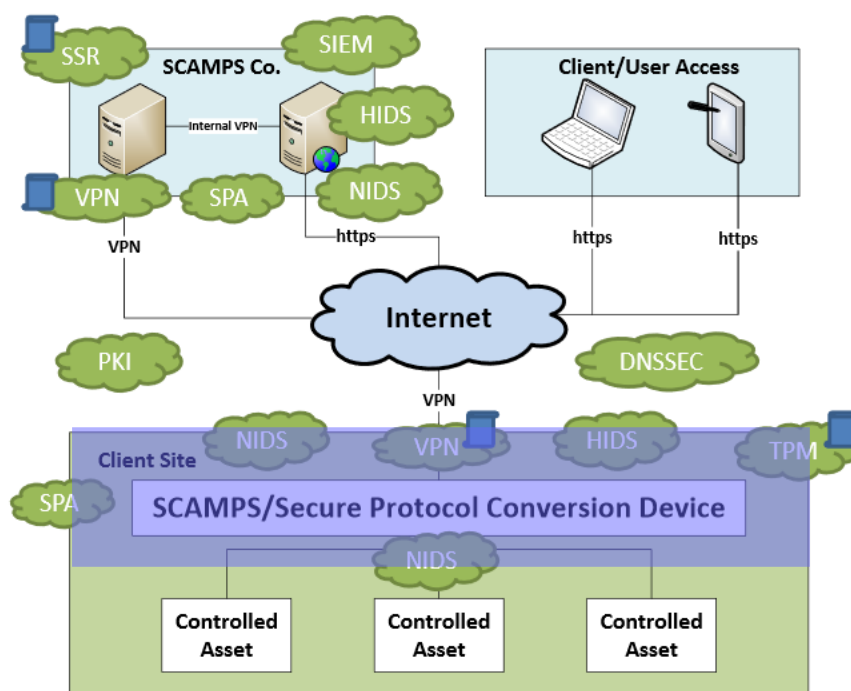
*Exemple de baie de serveur*

Présentation globale du système à mettre en place

Les différentes parties du projet

## Security Features of SCAMPS


- TPM – Trusted Platform Module
- DNSSEC – Secure DNS
- SPA – Single Packet Authentication (port knocking) - optional
- VPN – OpenVPN tunnel
- PKI – Public Key Infrastructure
- NIDS / HIDS – Network / Host Intrusion Detection System
- SSR – Secure Software Repository
- SIEM - Security Information and Event Management



4

Ci-dessus est représenté le schéma de l'architecture cible du projet de recherche, notre maître de stage n'attend pas forcément qu'absolument tout soit fait en 10 semaines c'est pour cela

que nous avons hiérarchisé les besoins présentés par ce schéma. Je m'occupe de la partie en bleu mais j'ai eu besoin de comprendre le fonctionnement de tout le système pour pouvoir y évoluer. Toutes les notions que j'ai été amené à traiter sont détaillées dans les parties qui y sont consacrées, ci-dessous je ne fait que les présenter pour donner une vue globale du projet. On retrouve donc, par ordre d'importance :

- VPN<sup>5</sup> (OpenVPN tunnel) : Permet de connecter deux machines par un tunnel, de telle sorte que les informations qui y seront échangées soient entièrement inaccessibles pour tout le réseau (intranet et internet) c'est une voie de communication privée et sécurisée.
- SSR (Secure Software Repository<sup>6</sup>) : Normalement pour se mettre un jour un ordinateur va chercher sur internet ce qu'il y a de disponible. Un SSR est un dossier présent sur le serveur (donc en interne de l'entreprise) où sont stockés les mises à jour que va aller chercher l'ordinateur client.
- PKI (Public Key Infrastructure<sup>7</sup>) : A chaque fois que l'on voit ce symbole:  cela signifie qu'un certificat<sup>8</sup> est nécessaire. Une PKI va permettre de centraliser (en interne) la création et la gestion de tous les certificats dont nous allons avoir besoin.
- NIDS (Network Intrusion Detection System<sup>9</sup>) : Ce système va permettre de surveiller les connections et autres actions qui auront lieu sur les différents réseaux qui seront mis en place, les NIDS permettent d'envoyer une alerte en cas de comportement suspect.
- SIEM (Security Information and Event Management<sup>10</sup>) : Ce système uniquement présent sur la partie « serveur » est un peu le centre de contrôle de l'administrateur, y sera stocké et rendu disponible toutes les informations concernant la sécurité et divers autres événements qui se déroulent sur le réseau.
- DNSSEC (Secure DNS<sup>11</sup>) : De manière générale un DNS est une application qui permet de

5 « virtual private network » à traduire par « réseau virtuel privé. »

6 Secure Software Repository, traduction : Dépôt sécurisé de logiciel

7 Public Key Infrastructure traduction : infrastructure public de clé

8 On reviendra sur la notion de certificat, en deux mots un certificat est d'une carte d'accès qui permet d'ouvrir certaines portes. Bien géré c'est une méthode extrêmement sécurisée et infalsifiable.

9 Network Intrusion Detection System, traduction : Système de détection d'intrusion

10 Security Information and Event Management traduction : gestion des événements et des informations de sécurité

11 Secure DNS traduction : DNS sécurisé

transformer une adresse IP (ex : 66.249.64.55) en nom (ex : google.com). Dans notre système cela permettra de simplifier la gestion du réseau (en donnant des noms aux machines) mais de manière sécurisée en empêchant un tiers de prendre un nom par exemple, ainsi chaque machine est connue. De plus pour de nombreuses raisons, certains systèmes seront peut-être amenés à changer d'adresse IP, grâce à un DNS cet événement ne sera plus un risque de dysfonctionnement.

- TMP (Trusted Platform Module<sup>12</sup>) : Un TMP est un module que l'on va directement connecter sur les BeagleBone Black qui va fonctionner en tâche de fond et qui va scruter les modifications logicielles dans l'objectif d'empêcher à d'éventuels virus de modifier l'ordinateur.
- ✕ SPA (Single Packet Authentication<sup>13</sup>) : Le SPA est un système d'authentification entre deux machines qui est intégré à OpenVPN qui est l'application que l'on utilise pour la création du tunnel VPN, il est donc inutile d'en implémenter un autre dans le système.

## Les flux d'informations

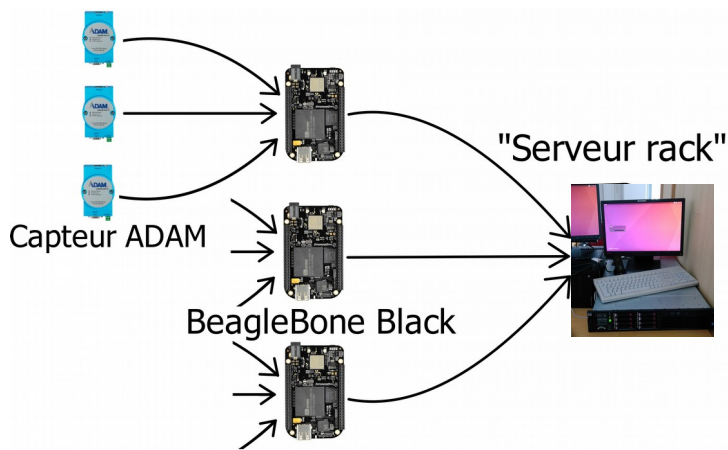
Le flux d'information principale qui est l'objet de la recherche sont des données collectées par des capteurs qui seront placés sur des machines industrielles et qui relèveront des données telle que la température ou l'humidité par exemple. Ses informations doivent remonter depuis les capteurs jusqu'à un serveur qui les rendra ensuite disponible sur une interface web accessible pour les utilisateurs. L'objet de notre stage est de réussir à sécuriser la remontée de ces données mais cela passe par de nombreuses étapes de sécurisation tout au long de l'acheminement des informations. Les capteurs sont directement reliés à des BeagleBone Black, ces derniers récupéreront les données de quelques capteurs, ensuite ils devront communiquer les dites données vers le serveur auquel ils sont reliés. (voir schéma ci-dessous)

---

DNS : Domain Name System, Système de nom de domaine.

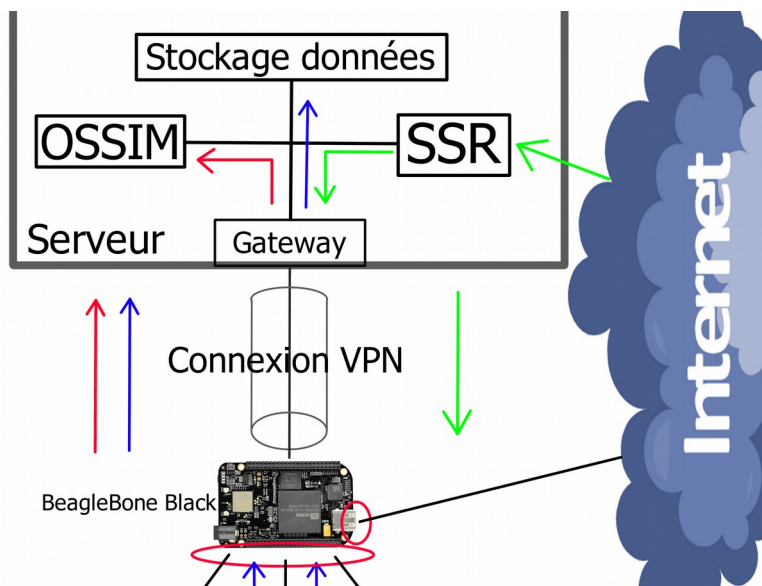
12 Trusted Platform Module traduction : module de plate-forme de confiance

13 Single Packet Authentication traduction : Authentification par paquet unique



*Représentation du flux de donnée principal*

Tous les systèmes précédemment décrit vont en plus de générer des informations, en demander, le principe même d'un NIDS est d'alerter l'utilisateur, il va donc falloir également, comme avec les données principales, acheminer, stocker et rendre disponible les alertes générées. De plus le BeagleBone Black va être relié à un SSR pour les installations et mises à jour, leur stockage se fera dans un premier temps sur le serveur, puis évidemment sur le BeagleBone Black une fois installées. Le schéma ci-dessous présente ces flux d'informations.



*Représentation des flux d'information au sein du système*

La partie haute du schéma représente le serveur et ses différentes machines virtuelles<sup>14</sup> qui ont chacune une tâche précise. La première machine virtuelle que l'on rencontre est la Gateway<sup>15</sup> cette machine virtuelle permet en quelques sortes d'aiguiller le trafic au sein du serveur. La machine virtuelle notée SSR est la machine virtuelle qui va fournir le service de mise à jour pour le

<sup>14</sup> Une machine virtuelle est un logiciel qui permet de simuler une (ou plusieurs) machine au sein d'une machine. Cela permet de diviser avec efficacité les ressources disponibles (RAM, CPU, mémoire...).

<sup>15</sup> Traduction : passerelle.

BeableBone Black. La machine virtuelle notée OSSIM<sup>16</sup> permet d'exécuter un service qui permet de surveiller les événements qui ont lieu sur le réseau. Pour en terminer avec les machines virtuelles, on peut noter celle qui permet de stocker les données principales du système en provenance des capteurs.

Les cercles rouges représentent les zones qui seront observées par le NIDS au niveau du BeableBone Black, ce sont des événements suspects sur ces interfaces de communication qui déclencheront des alertes (en fonction de règles) qui remonteront jusqu'au service OSSIM.

Les flèches bleues représentent le flux principale de donnée décrit précédemment, les rouges représentent le cheminement des alertes générées par le NIDS du BeableBone Black. Enfin les flèches vertes représentent le cheminement des paquets pour les mises à jour et pour les installations sur le BeableBone Black

## Planning prévisionnel et réel

Nom de la Tâche	Date de Début	Date de Fin	Durée	Mai							Juin			
				Avr 24	Mai 1	Mai 8	Mai 15	Mai 22	Mai 29		Juin 5	Juin 12	Juin 19	Juin 26
<b>Section 1 - Connexion VPN</b>	<b>02/05/17</b>	<b>12/05/17</b>	<b>9j</b>											
Sous-tâche 1 - Créer une connexion VPN	02/05/17	05/05/17	4j											
Sous-tâche 2 - Sécuriser la connexion à l'aide de clés et de certificats	08/05/17	12/05/17	5j											
<b>Section 2 - Mise à jour</b>	<b>15/05/17</b>	<b>02/06/17</b>	<b>15j</b>											
Sous-tâche 1 - Mettre à jour le BBB en local	15/05/17	19/05/17	5j											
Sous-tâche 2 - Mettre à jour depuis le serveur	22/05/17	26/05/17	5j											
Sous-tâche 3 - Mettre à jour de manière sécurisée	29/05/17	02/06/17	5j											
<b>Section 3 - Gestion des données</b>	<b>05/06/17</b>	<b>23/06/17</b>	<b>15j</b>											
Sous-tâche 1 - Récupérer les données en provenance des capteurs	05/06/17	09/06/17	5j											
Sous-tâche 2 - Envoyer les données au serveur	12/06/17	16/06/17	5j											
Sous-tâche 3 - Sécuriser l'envoi des données	19/06/17	23/06/17	5j											

### Planning prévisionnel

Le planning prévisionnel a été construit en fonction des informations que l'on avait le 28 avril. Notre maître de stage nous avait bien précisé que ce planning serait amené à changer en fonction de la direction que prendrait le projet. Et c'est en effet ce que s'est passé, de nouvelles tâches relativement chronophage ont été ajoutées au dépit des anciennes. Comme on peut le voir

<sup>16</sup> Open Source Security Information Management : système de gestion d'informations sur un réseau, OSSIM est une application permettant la gestion et la centralisation d'événements, d'alertes etc. Cette application permet d'avoir une vue d'ensemble sur le réseau

sur le planning réel (réalisé le 06/06/2017) ci-dessous les premières tâches étaient en fait bien plus rapide à réaliser que prévu. Puis notre maître de stage nous a demandé de mettre en place un NIDS, ces systèmes sont assez complexes et leur mise en fonctionnement se fait au cas par cas. Ainsi cette tâche a été assez longue, ensuite nous avons commencé à nous documenter sur la « plate-forme de confiance » (TPM) ainsi que sur l'acheminement des données entre les capteurs et le BeagleBone Black, ces deux tâches demandaient un temps de familiarisation à de nouveaux outils assez long d'autant que la fin du stage approchait à grands pas.

Nom de la Tâche	Date de Début	Date de Fin	Durée	Mai					Juin				
				Avr 24	Mai 1	Mai 8	Mai 15	Mai 22	Mai 29	Juin 5	Juin 12	Juin 19	Juin 26
[-] Section 1 - Connexion VPN	24/04/17	28/04/17	5j										
Sous-tâche 1 - Créer une connexion VPN	24/04/17	27/04/17	4j										
Sous-tâche 2 - Sécuriser la connexion à l'aide de clés et de certificats	28/04/17	28/04/17	1j										
[-] Section 2 - Mise à jour	02/05/17	05/05/17	4j										
Sous-tâche 1 - Mettre à jour le BBB en local	02/05/17	02/05/17	1j										
Sous-tâche 2 - Mettre à jour depuis le serveur	03/05/17	05/05/17	3j										
Sous-tâche 3 - Mettre à jour de manière sécurisée													
[-] Section 4 - NIDS	08/05/17	05/06/17	21j										
Sous-tâche 1 - Premières installations	08/05/17	12/05/17	5j										
Sous-tâche 2 - Configuration et tests	15/05/17	19/05/17	5j										
Sous-tâche 3 - Nouveau NIDS, installation et configuration	22/05/17	26/05/17	5j										
Sous-tâche 4 - Lien avec le serveur	29/05/17	05/06/17	6j										
[+] Section 4 - Gestion des données													

*Planning réel*

## Sécuriser les communications : VPN

### Introduction à la cybersécurité : clés et certificats

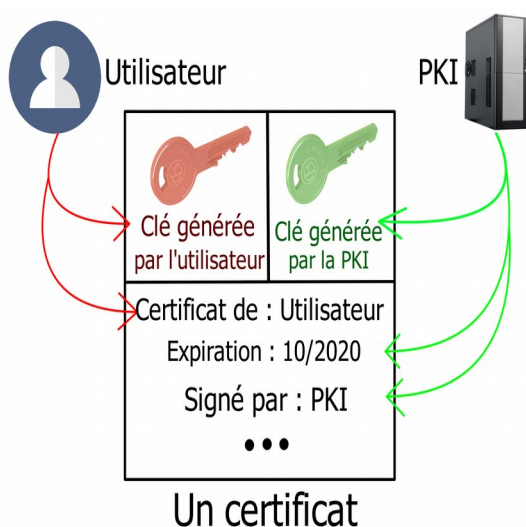
De manière factuelle une clé est une très longue suite de caractère générés à l'aide d'une fonction de hachage. Une telle fonction permet d'obtenir une suite de caractère de longueur fixe à partir d'une autre, le principal avantage de ces fonctions est que la chaîne de caractère obtenue (qui peut servir de clé) est toujours unique, ainsi deux clés sont toujours différentes. Ces clés uniques sont donc utilisées pour s'authentifier mais également pour crypter des messages ou des données.

Un certificat est une structure qui va contenir différentes données dont la principale est une clé. Le certificat va permettre de stocker avec cette clé des informations la concernant, par exemple

son propriétaire mais aussi et surtout un certificat va pouvoir être signé. La signature des certificats est gérée par une entité indépendante (PKI<sup>17</sup>) à qui l'on va fournir des certificats à compléter. Les informations ajoutées les plus importantes sont :

- La date d'expiration de la signature (jusqu'à quelle date le certificat sera valide)
- L'identité de l'infrastructure qui a signé de certificat
- Une clé de signature

Le certificat ainsi signé peut être alors utilisé et chaque entité qui sera amenée à utiliser la clé principale du certificat pourra vérifier à l'aide de la clé de signature que la clé principale est bien valide et certifiée.



*Représentation symbolique d'un certificat.*

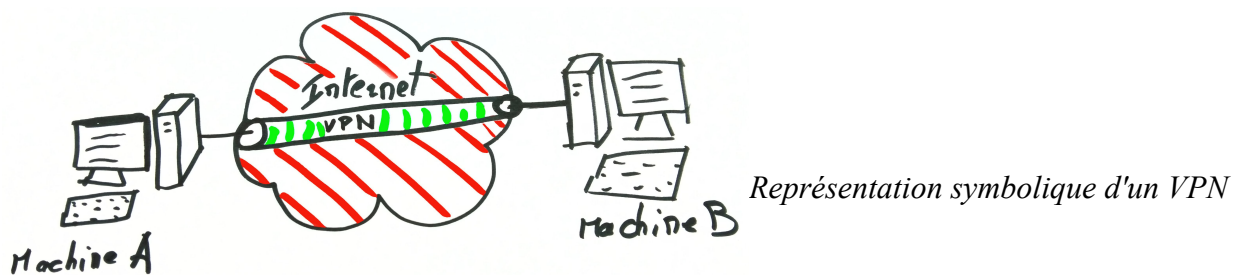
## Définition d'un VPN

Un VPN<sup>18</sup> est un réseau que l'on va construire au sein d'un autre réseau dans l'objectif de privatiser une connexion, c'est-à-dire créer un canal privilégié entre deux machines (ou deux sous-réseaux) de telle sorte que la communication ne soit pas corrompible. En effet la majorité des protocoles VPN, en plus de créer cette connexion privée vont crypter les informations qui transitent par le tunnel ainsi créé, on peut donc utiliser cette connexion pour communiquer de manière très sécurisée.

<sup>17</sup> Public Key Infrastructure, voir présentation globale

<sup>18</sup> « virtual private network » à traduire par « réseau virtuel privé. »





## Pourquoi un VPN dans ce système

Les communications au sein de notre système ne passent pas par internet, malgré tout nous avons tout de même besoin d'établir une connexion VPN. Nos machines communiquent au sein de l'intranet de l'entreprise où elle seront installées, de manière générale on considère qu'un intranet est une zone sécurisée car en théorie tout le trafic entrant est filtré par le par-feux de l'entreprise. Malgré cela il est assez utopique de penser que toutes les entreprises possèdent un réseau intranet sécurisé, à l'abri de toute tentative d'intrusion. Il existe en effet de nombreuses failles d'autant plus que des connexions sans fils sont présentes dans le système que l'on met en place. La connexion VPN que nous allons mettre en place ne permettra pas de passer au travers d'internet mais bien au travers de l'intranet de l'entreprise, ainsi les communications peuvent être garanties comme sûr peut importe où est installé le système. La connexion VPN va être le canal privilégié de toutes les communications futurs de notre projet ainsi, une fois mise en place nous pourrions éventuellement couper toute autre interface de communications et être certain de la fiabilité des échanges

## Implémentation du VPN



Avant notre arrivée notre tuteur de stage a déjà tenté sans succès une connexion VPN entre le BeableBone Black et le serveur VPN à l'aide de l'application openVPN.

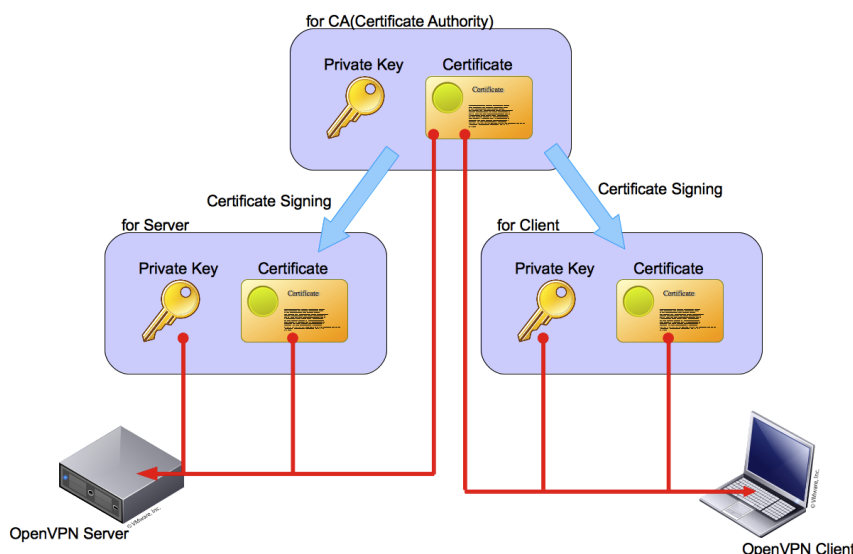
**OpenVPN :** OpenVPN est un logiciel open-source qui permet la création de tunnels sécurisés entre machines. Mais son intérêt ne s'arrête pas là, en effet l'initialisation de la connexion est elle-même sécurisée à l'aide d'un système de clés et de certificats de telle sorte que l'on soit absolument certain qu'uniquement les machines autorisées puissent se connecter.

La première étape consistait à installer l'application sur le BeableBone Black et à la configurer, ensuite il fallait établir la connexion avec le serveur, pour cela il faut récupérer les clés et les certificats que l'on a générés sur le serveur, cette tâche a été réalisée manuellement mais à terme nous devons développer une PKI<sup>19</sup> où l'on ira chercher clés et certificats. L'idée est la

<sup>19</sup> « Public Key Infrastructure » voir présentation du système.



suivante : le serveur génère deux clés et deux certificats, une paire pour lui et une pour le client (car la génération de telles entités sur le BeableBone Black demanderait trop de temps) chaque clé est associée à un certificat et celui-ci est signé par la PKI du système.



*Gestion des clé et des certificat avec OpenVPN, cf introduction aux clés et certificats*

La clé et le certificat client sont ensuite placés sur le client (le BeableBone Black) et on peut initialiser la connexion. Lors de la connexion le serveur vérifie si la clé et le certificat présentés par le client sont valide et le client fait de même avec ceux de serveur, ainsi une fois la connexion effectuée chaque partie sait qu'il est connecté à la machine qu'il souhaite.

## Conclusion partielle

L'implémentation du tunnel VPN n'était pas la première étape pour rien, en effet au sein notre système cette connexion est la seule qui permet de communiquer en toute sécurité. C'est une étape absolument cruciale qui permet d'être maintenant sûr que l'on pourra implémenter d'autres systèmes communiquant des données mais aussi en toute logique d'être certain de l'intégrité des données, ce qui est encore une fois crucial pour un système. La première utilisation de ce VPN va être la récupération de paquet pour l'installation et la mise à jour des logiciels sur le BeableBone Black.

## Mettre le système à jour

## Définition et avantages d'un SSR<sup>20</sup>

Pour définir ce qu'est un logiciel de dépôt sécurisé, il faut dans un premier temps comprendre le système de dépôt pour les mises à jour sous Linux. De base lorsque l'on souhaite installer un logiciel sous Linux le système va interroger différents serveurs et leur demander s'il possède le logiciel souhaité, dans l'affirmatif, le système Linux va pouvoir télécharger le logiciel. Ce mécanisme est également utilisé lorsqu'un système souhaite se mettre à jour, il va vérifier pour chacun de ses logiciels s'il possède la dernière version et recommencer le même mécanisme.

L'intérêt d'un dépôt sécurisé de logiciel est qu'il est privé, en effet au lieu d'aller demander à plusieurs serveurs pour notre logiciel, avec un SSR il n'y a qu'une seule source à interroger. Les logiciels présents sur ce dépôt sont gérés manuellement, c'est l'administrateur qui va décider de ce qui sera disponible en téléchargement. De ce fait, il devient presque impossible pour une personne mal intentionnée de proposer de fausses mises à jour.

*Représentation du système de mise à jour, avant (à gauche) et après (à droite)*

Comme on peut le voir sur cette illustration un système classique (à gauche) peut-être vulnérable lors de mises à jour automatiques qui pourraient être générées par un pirate se faisant passer pour un serveur Linux. Grâce au système que l'on doit mettre en place ce type d'attaque devient très compliqué voir impossible, en effet la source de mise à jour où va aller chercher le BeagleBone Black n'est plus un serveur externe mais une partition sur le serveur rack, partition qui n'est peuplée que de paquets que l'administrateur aura préalablement accepté. Pour qu'un paquet corrompu atteigne le BeagleBone Black il faudrait tromper la vigilance de l'administrateur ce qui est quasiment impossible si ce dernier vérifie méthodologiquement quels paquets il accepte (en allant sur le site web du diffuseur par exemple).

## Connexion au SSR par le VPN

Le SSR a été mis en place sur le serveur, la suite est assez simple, modifier le fichier de mon système sur lequel se trouvent toutes les sources utilisées. Pour cela il a fallu retirer toutes les sources ordinaires pour enfin en ajouter une qui serait la seule, le SSR sur le serveur. On peut établir une liste de toutes les sources à télécharger sur le SSR à partir des sources que l'on n'utilisait préalablement mais cette liste va être amenée à évoluer au fur et à mesure que l'on va implémenter de nouvelles applications il faudra donc récupérer la liste des sources du BeagleBone Black une fois le projet fini pour connaître toutes les sources utilisées. Il faudra également trier les sources utiles

---

<sup>20</sup> Secure Software Repository, traduction : Dépôt sécurisé de logiciel

de celles que l'on utilisera pas pour alléger la mémoire du BeagleBone Black.

La connexion au SSR devait elle aussi être sécurisée, c'est à ce moment qu'Antoine a mis en place la PKI sur le serveur, à l'aide de cette dernière nous avons pu réaliser la connexion au dépôt de logiciels à l'aide d'un certificat de telle sorte d'être certain que seul les entités autorisées puissent se connecter. Pour pouvoir réaliser une connexion au dépôt il faut d'abord générer une clé unique sur la PKI puis utiliser cette clé depuis le BeagleBone Black pour se connecter. Une fois la connexion établie l'utilisation de la clé est limitée à la machine qui l'a utilisée. Personne ne pourra se connecter à moins qu'une clé lui soit spécialement fabriquée.

Il peut sembler que l'on aurait pu faire le tri des sources pour les mises à jour directement sur le BeagleBone Black mais cela n'aurait pas été judicieux, en effet avec le système que nous avons mis en place nous centralisons les sources car il y aura plusieurs clients pour les sources.

## Mise à jour automatique

Une fois que les BeagleBone Black seront mis en marche sur le système un client aimerait ne jamais avoir à y faire les mises à jour, le système doit être autonome et faire remonter de l'information. Les mises à jour doivent donc se faire automatiquement. Il y a donc était envisagé dans un premier temps d'utiliser le logiciel cron qui permet de gérer des tâches qui fonctionnent en arrière plan. Cron permet de demander au système d'exécuter certaines tâches (démarrer une application, supprimer des fichiers etc.) décrites dans des scripts, à des moments définis par des dates, des heures ou encore des intervalles de temps (tous les mois, toutes les minutes). Ce logiciel est utilisé pour l'administration de nombreux éléments allant des e-mails aux mises à jour.

Mais en cherchant comment créer une tâche qui ferait les mises à jour il s'est avéré qu'une application avait déjà été développée à cet effet : cron-apt. Cette application une fois configurée fait les mises à jour du système tous les jours par exemple. Il y a une grammaire qui permet de configurer quand va s'exécuter le processus de mise à jour.

## Conclusion partielle

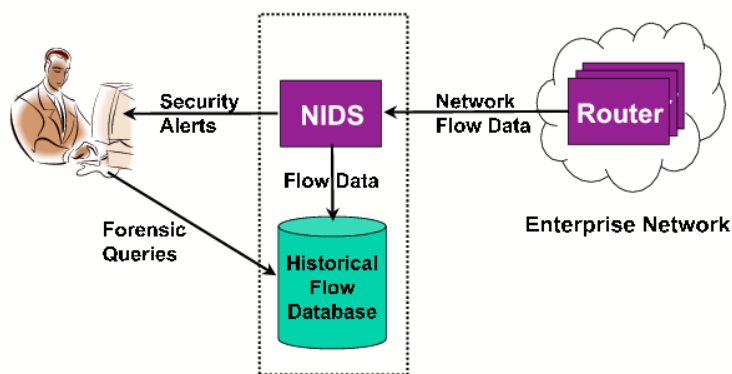
Maintenant que cette partie du projet est mise en place nous sommes en mesure d'affirmer que les mises à jour sur le BeagleBone Black se font automatiquement et surtout de manière sécurisée. Ainsi un administrateur réseau n'aura pas à installer manuellement les mises à jour sur chaque BeagleBone Black. Grâce au SSR la seule tâche consiste à valider les paquets proposés par

les serveurs Linux. Ce système propose également un autre avantage, le fait de pouvoir décider quels paquets seront mises à jour cela permet de limiter très grandement la quantité de donnée au sein de notre système embarqué, les petits ordinateurs n'en fonctionnerons que mieux. Nous l'avons vu, le réseau extérieur peut être synonyme de danger, protéger nos mises à jour et une chose mais se protéger de toute attaque en provenance d'internet en est une autre, c'est l'objet de la partie suivante.

## Protection envers le réseau

### Qu'est qu'un NIDS

Un NIDS est une application qui s'implémente sur un serveur ou sur une machine connectée à un réseau dont l'objectif va être de prévenir toute tentative d'intrusion non souhaitée. Ces systèmes se veulent « intelligent » dans le sens où ils ont vocation à différencier les attaques malveillantes des connexions légitimes en utilisant un système de pattern qui reconnaît de quel type de connexion il s'agit. Un NIDS possède également tout un système de gradation des attaques qui déclenchent différentes réponses. Bien qu'ils fonctionnent en autonomie une fois configurés et lancés, ils vont communiquer avec l'administrateur réseau par e-mail, par fichiers d'historiques etc. (voir schéma) pour l'informer des différents événements qui se sont passés (historique) mais aussi des événements en cours lorsque ceux-ci présentent un caractère urgent (alerte de niveau élevée). De manière générale un NIDS va être placé sur une machine à l'entrée du réseau de manière à scruter le trafic entrant. L'installation d'une application de détection d'intrusion se fait en deux parties, la première consiste à installer et configurer l'application en elle-même la seconde, également cruciale, consiste à définir les règles que va exploiter l'application. L'idée va être de définir comment le NIDS va devoir se comporter en fonction des connexions qu'il va analyser, il va donc falloir définir quels types de connexion représentent des dangers potentiels et quels autres sont à ignorer. Cette étape de création des règles qui vont régir le NIDS était à la base réalisée manuellement mais cette tâche est devenue bien trop complexe tant le nombre de règles à créer est important, c'est pour cela qu'il existe des programmes qui vont servir à créer ses règles à partir des patterns connus de connexion mal intentionnés. Une fois les règles implémentées l'administrateur n'a plus qu'à rajouter quelques règles propres à son réseau, enfin, le NIDS est prêt.



*Communication entre le NIDS et l'administrateur<sup>21</sup>*

## Pourquoi notre système requiert un NIDS

Encore une fois nous n'allons pas utiliser l'application exactement comme elle est censée l'être. En effet, comme déjà présenté notre système n'aura pas accès directement à l'entrée du réseau de l'entreprise et nous considérons donc que celui-ci peut ne pas être complètement sécurisé, or notre système doit l'être. Le NIDS que j'ai à installer sur le BeableBone Black ne sera là que pour analyser les connexions destinées à ce dernier et non à tout le réseau. De cette manière notre système est sécurisé comme s'il était directement connecté à internet indépendamment du pare-feu<sup>22</sup> de l'entreprise.

## Implémentation

Du fait de l'utilisation peut conventionnelle que l'on fait de notre NIDS une contrainte et très présente : la RAM<sup>23</sup> utilisée par ce type d'application est souvent assez important d'autant que les NIDS sont prévus pour s'exécuter sûr de puissant serveur, or ici je travaille sur un petit ordinateur, le choix de l'application de NIDS sera donc crucial pour ne pas surmener mon système.

## Application PSAD : Mauvaise piste

J'ai dans un premier temps installé et configuré une application nommée « psad » qui est une application de NIDS, ainsi que quelques applications prévues pour la gestion des règles décrites précédemment. Nous avons passé plusieurs jours de travail sur ce système sans que celui-ci ne soit capable de donner une quelconque alerte. De plus notre maître de stage a rajouté une nouvelle

21 Traductions : router = routeur, entreprise network = réseau de l'entreprise, Network flow data = flux de données du réseau, flow data = flux de données, Security Alerts = alerte de sécurité, Historical Flow Database = Base de donnée de l'historique des flux, Forensic Queries = Requête d'historique

22 Un pare-feu est une zone du réseau par laquelle tout le trafic entrant et sortant transite, c'est ici qu'il va y être filtré

23 Ou mémoire vive, qui correspond à la limite de nombre calcul dont est capable le système à un instant t.

condition : le système de NIDS devra être capable d'envoyer les alertes déclenchées au serveur sur une application particulière nommée OSSIM, de manière à centraliser les différentes alertes. Après quelques recherches il s'est avéré qu'aucune explication n'était disponible quant à la démarche à suivre pour faire un lien entre psad et OSSIM je décide d'abandonner psad pour un NIDS plus connu pour lequel la démarche à suivre pour faire un lien avec OSSIM est disponible : SNORT.

De plus le fonctionnement des règles de psad pour les alertes paraît inapproprié. Les outils pour la création des règles prennent en compte davantage les règles d'iptables<sup>24</sup> que des patterns connus ou autres règles de la communauté.

## Application SNORT : Solution utilisée

Le choix de SNORT n'a pas été fait par hasard, en effet c'est un NIDS adapté aux systèmes embarqués très connu et souvent utilisé dans ce genre de projet. De plus M. Shukat et M. Melvin ont déjà été amenés à utiliser cette application lors de tests préliminaires ou sur d'autres projets. SNORT est un NIDS il répond donc à toutes les caractéristiques décrites précédemment mais il peut



également servir à empêcher des intrusions (pas seulement à alerter). Comme dit précédemment SNORT est très rependu de ce fait une communauté active développe des collections de règles déjà établies, et c'est un point crucial car ce sont les règles qui définissent à quel point le NIDS va être efficace.

*Logo de SNORT le NIDS utilisé*

Après plusieurs tentatives il semblerait que le BeagleBone Black ne possède pas assez de mémoire pour ingérer les nombreuses règles (plus de 30 000) automatiquement implémentées par les outils conventionnels. Suite à un entretien avec mon maître de stage la décision est prise de rédiger les règles manuellement.

Nous expliquions plus tôt qu'implémenter toutes les règles manuellement était une tâche trop complexe il est vrai que nous ne pouvons pas prétendre remplacer toutes les règles existantes à la main. Mais nous n'utilisons pas ce NIDS de manière classique, de ce fait la création des règles pourra être personnalisée. Nous ne souhaitons pas administrer l'entrée d'un réseau d'entreprise (utilisation « classique ») mais plutôt prévenir de toute connexion sur le BeagleBone Black. Les règles à ajouter furent donc assez basique puisqu'elles visaient à déclencher une alerte pour toute

---

<sup>24</sup> Iptable est une application implémentée de base sous Linux qui fait office de pare-feu en triant les connexions entrantes et sortantes

connexion ne provenant pas du tunnel VPN. Il a fallu ensuite ajouter quelques règles concernant les communications avec les capteurs. En effet bien que nous n'avions pas encore abordé cette partie du projet, les BeagleBone Black ont pour objectif d'être reliées à des capteurs dont il récupérera les données. L'ajout des règles s'est fait deux parties, dans un premier temps accepter les communications utilisant le protocole Modbus<sup>25</sup> (le protocole utilisé par les capteurs pour communiquer) et dans un deuxième temps ajouter des règles prévenant d'éventuelles attaques qui passeraient par le protocole Modbus.

## Lien avec le serveur

Une fois SNORT installé et configuré il faut réussir à communiquer les alertes détectées. En effet l'administrateur réseau ne veut pas devoir vérifier manuellement sur chaque BeagleBone Black si des alertes ont été détectées car en plus d'être chrono-phage cette stratégie serait hautement inefficace.

Comme décrit au début de ce rapport le serveur possède une machine virtuelle sur laquelle l'application OSSIM s'exécute en continu. OSSIM est un projet open source visant à proposer un système de gestion d'événement au sein d'un réseau, plus simplement, cette application permet de centraliser de nombreuses informations concernant la sécurité et d'autres événements provenant d'un réseau. Mon objectif a donc été de faire remonter les alertes déclenchées sur le BeagleBone Black jusqu'à la machine virtuelle où OSSIM est exécuté, de manière à centraliser (dans le cas où il y aurait plusieurs BeagleBone Black) toutes les alertes.

Pour acheminer les alertes nous avons utilisé le logiciel Rsyslog, logiciel adapté au transfert de journaux d'événements. Ce logiciel fonctionne de base sur ma distribution de Linux, il a fallu configurer SNORT pour que les alertes générées ne soit plus stockées dans un fichier de log<sup>26</sup> en local mais envoyés dans la file d'attente de Rsyslog pour l'envoi vers le serveur. Enfin il a fallu reconfigurer Rsyslog pour qu'il envoie les alertes générées, vers la machine virtuelle et cela, évidemment au travers du VPN.

## Conclusion partielle

Une fois le lien avec le serveur effectué, l'administrateur réseau à l'aide de OSSIM possède

---

<sup>25</sup> Modbus est un protocole de communication qui permet des communication extrêmement simplifiées avec des capteurs.

<sup>26</sup> Un fichier de log est un fichier qui va contenir l'historique des messages émis par un système

une vue d'ensemble assez exhaustive en temps réel des événements malveillants qui se déroule (ou se sont déroulés) sur tous les BeagleBone Black du système. Notre système communique avec le serveur au travers d'un VPN sécurisé où toutes les mises à jour passent. Il reste néanmoins une source de faiblesse de notre système, en effet, si quelqu'un réussit à modifier le contenu du BeagleBone Black (en modifiant le contenu de la carte SD par exemple) il pourrait infecter le système de l'intérieur, bien que ce scénario soit moins probable qu'une attaque au travers d'internet (car il faudrait que le pirate pénètre physiquement dans l'entreprise) il faut également gérer ce cas. Malheureusement, à l'heure où j'écris ces lignes cette partie n'est qu'à l'état d'embryon et ne possède donc pas sa place dans ce rapport.

## Réflexion sur le projet

### Lien avec les matières

Durant mon stage j'ai été à de nombreuses reprises en contact avec des notions abordées pendant notre formation à l'IUT<sup>27</sup>. La matière principale de mon stage était le travail sur le BeagleBone Black sur lequel le système d'exploitation n'était autre qu'une distribution de Linux, nous avons été au travers de différentes matières amenés à utiliser Linux. Plus précisément mes notions de réseaux m'ont été cruciales, évidemment sur la partie traitant du NIDS, qui je le rappelle et une protection du réseau. Les notions de réseau ont été également utiles lors de la connexion VPN, ou même tout simplement lorsqu'il a fallu relier le BeagleBone Black à internet.

Il y a également une partie peu abordée dans le rapport mais qui nous a occupé un certain temps, où l'on a développé un module qui permet notamment de sécuriser le démarrage (ou boot) du BeagleBone Black. Pour cela il a fallu utiliser les notions de boot sous linux dans un environnement embarqué abordées en SEE4<sup>28</sup>. Cette partie du cours est passée très rapidement et je n'en ai retenu que des principes globaux qui certes m'ont été utiles mais mes connaissances restaient superficielles.

### Apports techniques

Ses 10 semaines de stage ont été pour moi très formatrices techniquement, en effet j'ai été amené à travailler dans des domaines assez variés d'autant que les tâches réalisées ont été souvent assez courtes. Je pense avoir énormément progressé en réseau, en effet bien que les cours dispensés à

---

<sup>27</sup> IUT : Institut Universitaire Technologique

<sup>28</sup> SEE : Systèmes d'exploitations embarqués, semestre 4



L'IUT soit complet mes notions dans ce domaine était très médiocre, j'ai réappris des notions, le progrès est grand car je partais avec peu. Cette idée de progrès important dû à ma méconnaissance des sujets abordés est assez présent dans le bilan technique de ce stage en réseau donc mais aussi avec tout ce qui touche aux commandes Linux ainsi qu'à toutes les notions spécifiques de ce rapport.

## Apports méthodologiques

Bien que les projets sur lesquels nous avons travaillé à l'IUT nous ont permis de découvrir le monde professionnel et de nous familiariser avec un modèle de travail nouveau, le stage a été pour moi une autre étape, bien plus important. J'ai appris à me fixer des objectifs à très court terme (une journée) et donc à découper une tâche importante en de nombreuses tâches réalisables rapidement. J'ai aussi appris à me retrouver seul devant un problème et à devoir le surmonter quant-bien même je n'en avais pas la compétence. Justement, apprendre par moi-même des notions parfois relativement techniques fut assez nouveau et difficile mais j'y suis parvenu, parfois laborieusement il faut l'admettre.

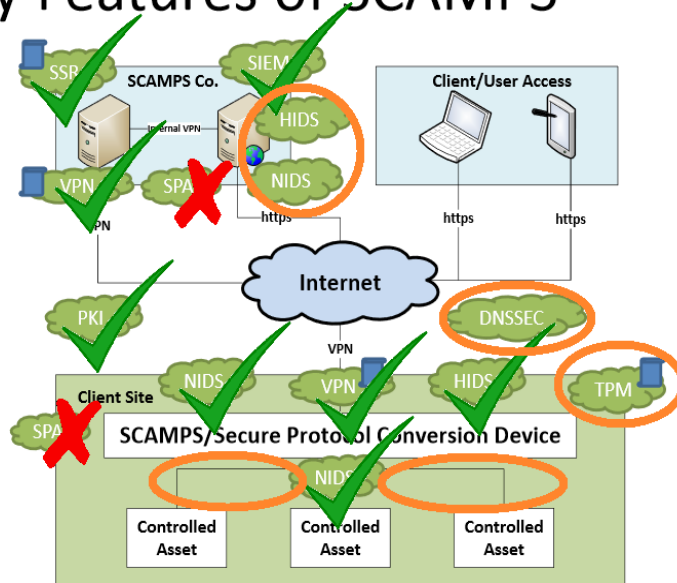
## Ressenti personnel

Avoir l'impression d'avoir produit un travail sur lequel d'autres chercheurs de l'organisation pourront s'appuyer est, je doit l'avouer, assez valorisant. Nous avons réalisé en 10 semaines une partie relativement importante de l'objectif global fixé pour cette recherche.

Pour et-teiller mes propos j'ai annoté le schéma que l'on nous avait fournis à notre arrivée. En vert les tâches réalisées, en rouge celles jugées inutiles et en orange les tâches non terminées.

# Security Features of SCAMPS

- TPM – Trusted Platform Module
- DNSSEC – Secure DNS
- SPA – Single Packet Authentication (port knocking) - optional
- VPN – OpenVPN tunnel
- PKI – Public Key Infrastructure
- NIDS / HIDS – Network / Host Intrusion Detection System
- SSR – Secure Software Repository
- SIEM - Security Information and Event Management



4

## Présentation du projet annoté

Bien que la progression du projet durant ce stage fut bien réelle et qu'il m'a apporté beaucoup, le domaine concerné par le stage n'était au final absolument pas adapté à mes goûts. En effet que ce soit le réseau, ou l'administration de système sous Linux, ces deux domaines ne m'intéressent que très peu et ce n'est absolument pas ma vision (ni celle de l'IUT selon moi) de l'informatique embarquée. La partie Linux de notre enseignement durant le semestre 4 concernant les systèmes embarqués reste anecdotique en comparaison des enseignements en C par exemple que je m'attendais (à tort donc) à retrouver durant le stage.

# Conclusion

Les objectifs du stage étaient nombreux d'autant que les sources et les formes potentielles d'attaque sur un réseau sont extrêmement variés, de se fait tous les objectifs atteignables n'ont pas été remplie tant le projet de recherche est vaste. Le DNS privé et sécurisé ainsi que sa gestion n'ont pas été abordés, le NIDS sur le serveur n'a pas été mis en place, la plate-forme de confiance pour le BeagleBone Black n'a pas été installer bien que les recherches à le sujet ai été débutées. Le flux de donnée principal décrit au début du rapport permettant l'acheminement des données qui représentaient le centre du projet n'ont pas non-plus pu être terminés.

Malgré cela nous avons mis en place bon nombre des parties cruciales du projet toutes centrées sur la sécurisation des communications ce qui correspond bel et bien aux objectifs principaux de ce stage. Nous avons mise en place une communication VPN, outil centrale et indispensable de manière à communiquer en toute sécurité entre nos différents systèmes, cette connexion VPN a été la colonne vertébrale de notre système permettant la mise en place des systèmes suivants. La première utilité de cette connexion VPN fut l'automatisation des mises à jour de manière sécurisée, en effet aller chercher nos mises à jour et installations de programme sur internet n'étant absolument pas sécurisé nous avons mis en place un système de dépôt de logiciel sécurisé permettant au BeagleBone Black de se mettre à jour en toute sérénité depuis le serveur. Enfin nous avons mis en place un système de NIDS, qui permet de surveiller les communications qui sont établies avec le monde extérieur et tel sorte à envoyer des alertes sur le serveur en cas de menace potentielle.

Le système que nous avons mis en place et certainement une bonne base pour les itérations futures du projet de recherche, tel qu'il est maintenant il ne pourrait pas être mis en place dans un cas réel pour plusieurs raisons, en effet comme dis précédemment nous n'avons pas effectué toutes les tâches nécessaires pour terminer le projet, hormis cela, les tâches que nous avons réalisées comportent également des défauts. Les règles SNORT utilisées sont assez peu complètes, les alertes reçues sur le serveur sont donc assez générique et ne renseigne que trop peu sur la nature exacte de l'alerte. La connexion VPN se fait tel que expliqué se fait à l'aide d'un certificat, pour que ce certificat soit vraiment sûr il faudrait le faire signer par notre PKI, le problème, dans notre système actuel nous ne pouvons utiliser la-dite PKI qu'au travers du VPN. Il faudrait donc adapter la PKI pour que l'on puisse l'utiliser depuis le réseau sans compromettre son intégrité.

# Glossaire

- VPN : « **v**irtual **p**rivate **n**etwork » à traduire par « réseau virtuel privé. » désigne la connexion entre deux machines ou réseaux de manière privée et souvent encryptée.
- BeableBone Black : Le BeableBone Black est l'outil principal de mon stage, c'est une carte de développement sur laquelle l'on va utiliser une distribution de Linux. La carte possède les principales caractéristiques d'un ordinateur classique pour une taille minime
- SSR : Secure Software Repository, traduction : Dépôt sécurisé de logiciel. Il s'agit d'un espace mémoire où l'on va déposer des logiciels de telle sorte à les rendre disponibles pour d'autres machines.
- PKI « Public Key Infrastructure » à traduire par « Infrastructure publique de clé » permet de centraliser la création et la gestion de certificats et de clés
- NIDS « Network Intrusion Detection System » à traduire par « Système de détection d'intrusion » est un type d'application permettant de surveiller les réseaux et d'alerter l'administrateur en cas de problème.
- OSSIM « Open Source Security Information Management » à traduire par « système de gestion d'informations sur un réseau » OSSIM est une application permettant la gestion et la centralisation d'événements, d'alertes etc. Cette application permet d'avoir une vue d'ensemble sur le réseau

# Bibliographie

Introduction to Public Key Infrastructure de Johannes A.Buchmann, Avangelos Karatsiolis et Alexandre Wiesmaire

Site de l'université : [www.nuigalway.ie/](http://www.nuigalway.ie/)

Groupe de recherche : <http://www.osna-solutions.com>

Site logiciel VPN : <https://openvpn.net>

Site NIDS SNORT : <http://snort.org/>

OSSIM : <https://www.alienvault.com/products/ossim>

Ressources pour le BeagleBone Black : <http://beagleboard.org/black>  
<http://derekmolloy.ie/beaglebone/>

Ressources pour PSAD : <http://cipherdyne.org/psad/>

Lien en OSSIM et SNORT : [https://s3.amazonaws.com/snort-org-site/production/document\\_files/files/000/000/113/original/Integrating\\_Snort\\_and\\_OSSIM-rev2.pdf](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/113/original/Integrating_Snort_and_OSSIM-rev2.pdf)

## Tables des Annexes

<i>Annexe 1 : Documentation pour le groupe de projet.....</i>	<i>30</i>
---	-----------

## Résumés et mots clés

J'ai réalisé mon stage au sein de l'université de Galway nommée NUIG laquelle est le berceau d'un groupe de recherche : OSNA. Ce groupe de recherche vise à développer et à proposer des solutions de cybersécurité pour des réseaux de capteurs industriels. En effet le nombre de capteur au sein des entreprises ne fait qu'augmenter alors que l'on cherche à toujours plus automatiser les productions tout en gardant un œil sur les informations émanant la chaîne de production. Malgré tout il n'existe que très peu de solution proposant aux systèmes embarqués d'être sécurisé.

Durant ce stage nous avons été amené à sécuriser différents aspects d'un BeagleBone Black qui est un petit ordinateur servant, dans notre système, de relais entre les capteurs et le serveur centralisant toutes les données. La protection de cet ordinateur fut abordé sous trois angles principaux : La création d'un tunnel VPN permettant au petit ordinateur de communiquer avec le serveur de manière privée et sécurisé et cela au travers de n'importe quel réseau (internet ou intranet) ; la mise en place d'un système d'installation et de mise à jour de logiciel depuis le serveur vers le BeagleBone Black au travers du VPN et non depuis internet et enfin la mise en place d'une application de NIDS permettant de surveiller les connexions sur le BeagleBone Black.

Grâce à ce stage nous avons pu développer une partie d'un système qui permettra, à terme aux entreprises de posséder les réseaux de capteur plus sûr et ainsi de protéger des données sensibles tout en gardant une vision d'ensemble sur les événements qui ont lieu au sein de leur société.

I made my internship in the university for Galway called NUIG which had created a research group: OSNA. This research group try to develop and to show solution of cyber-security for industrial sensors network. Actually we can observe an important increase of the number of sensors in our company because we want to make our production more and more automatic while keeping an eyes on data whose come from our assembly-line. There is not a lot of solution that offer for embedded system a security layer for communication.

During this internship we had to make secure different part of a BeagleBone Black which is a tiny computer used, in this project, as relay for the data coming from the sensor to the server which centralize all the data. The protection of this tiny computer has been done in three points: we made a VPN connection which allow the BeagleBone Black to communicate with the server via a private and secure way no matter though which network it pass by (internet or intranet); we built a system of installation and update from the server to the BeagleBone Black trough the VPN tunnel and not from the internet; and finally we set up an NIDS application for watching connections coming on the BeagleBone Black.

Thanks to the internship we were able to develop a part of a big system which will allow company to have secure sensors network, more safe, and to protect themselves their important data while having a global view of their system.

## Mots Clés

Systèmes embarqués  
Cybersécurité  
Industrie  
Capteurs

NIDS  
Mise à jour sécurisé  
Tunnel VPN  
BeagleBone Black

## Documentation about BeagleBoneBlack

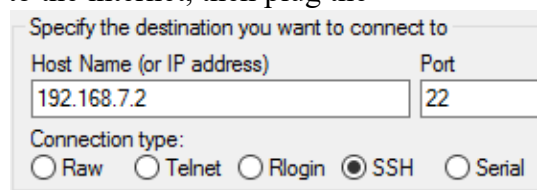
### 1. VPN connection

#### a) OS installation

- First you need to download a good version of BeagleBoneBlack OS (Jessie, Debian 8.7), find a direct download [here](#) (all the images available here: <https://beagleboard.org/latest-images>)
- Then flash the downloaded file in the SDCard of the BeagleBone Black
  - If you are using Windows: [Flashing using Windows](#)
  - If you are using Linux type: `dd bs=1M if=SOURCE of=TARGET`  
SOURCE: The .img file downloaded  
TARGET: Your SDCard, use `lsblk` command before and after having plugged the Card to get it name (if there are several same for the Card, use the one without number)  
finally execute: `sync`
- For more information see: <http://beagleboard.org/getting-started#books>

#### b) Prepare the BeagleBone Black

- Before plugging the USB cable plug the Ethernet one to the internet, then plug the USB to your computer
- Using [PuTTY](#) make a direct SSH connection through the USB
- Now you can connect to the BBB (id: debian pwd: tempwd) and get the IP address (`ifconfig`) of the BBB to make SSH connection by Ethernet.
- Now type: `apt update` and `apt upgrade` (can be long)
- Then, install OpenVPN: `apt install openvpn` and check if you got the right



```
debian@beaglebone:~$ openvpn --version
OpenVPN 2.3.4 arm-unknown-linux-gnueabi
(OpenSSL 1.0.1.4) [LZO] [x264] [zlib] [uuid] [system] [SHA512]
```



version (openVPN -version) you need 2.3.4 or higher

- If you don't have the good version, try:

`apt upgrade openvpn`

#### c) Make the VPN connection

- First you will need 3 files (client.key client.crt ca.crt) provided by the server (make a safe transaction, by USB or FTP for example) copy them in /etc/openvpn
- Now configure the connection: `nano /etc/openvpn/client.conf`  
Check that the field ca, cert, key contain the right name. Also put the server address in the remote field.
- Now you can execute: `openvpn client.conf` to start the VPN connection
- In `ifconfig` you will find a new entry names tun0

#### d) In case that the snort installation make the openvpn initialization on boot crash you will need to:

- Add those 3 lines in /etc/init.d/openvpn at line 43 :  
`sudo mkdir -p /dev/net`  
`sudo mknod /dev/net/tun c 10 200`  
`sudo chmod 600 /dev/net/tun`
- If openvpn won't start, put my initialization script named "initOpenVPN.conf" in /etc/init/

## 2) Secure update

#### a) cron apt

- First instal cron-apt : `apt install cron-apt`
- Default setting are basic but enough for what we want, nothing to add, we have an update every day at 4'o clock
- Only thing remaining to make it work, every time we want t to check if the system need an upgrade then type: `apt-get dist-upgrade` then, you can choose to upgrade or not.
- For more information: [https://debian-administration.org/article/162/A\\_short\\_introduction\\_to\\_cron-apt](https://debian-administration.org/article/162/A_short_introduction_to_cron-apt)

#### b) Change update source

- Now we have to change the directory used for update
- Go to `/etc/apt` and edit `sources.list` by removing everything (or adding a `#` at the beginning of every line) and add one line:
- `deb http://SERVER_IP CODENAME DIRECTORY`  
`SERVER_IP`: The IP used for hosting update  
`CODENAME`: The version's name of Ubuntu on the BBB, so "jessie"  
`DIRECTORY`: The name in the server of the directory where update are available

Our configuration : `deb http://10.10.10.11/ jessie main`

- Now we can use the a key to unlock the repository:
- `sudo apt-key adv --keyserver KEY_SERVER --recv-key XXXXXXXX`  
`XXXXXXXXXX`: The key generated, on the PKI by the administrator of the server.  
`KEY_SERVER`: ip Adresse of the key server (we used 10.10.10.13)

### 3) NIDS

- I tried to use `psad` and `fwsnort` as NIDS for our BBB but once is installed couldn't make `psad` work and it's look really hard to make a connection with OSSIM.
- For more information: <http://bodhizazen.net/Tutorials/psad>

#### a. Installing SNORT

- For the installation of SNORT I followed [the official guide](#) while thinking about my aim, so for the first step don't install Barnyard2 witch will be useless and nether PulledPork witch make crash the BBB, I made a script that make a big part of the job named "install snort"
- Some advices :
  - Many steps on "Network Card Configuration" are useless on our platform
  - When you need to choose, always follow the Ubuntu 14 instructions
  - Page 9 an ip address is needed, I used 192.168.10.0/24
  - Page 18 for making your init script executable you have to use `initctl` command, in this aim install `upstart` with `apt`
- More information: [www.snort.org](http://www.snort.org)

#### b. Link with OSSIM

- First you need to make a backup of `/etc/rsyslog.conf` and `/etc/snort/snort.conf`
- now go in `/etc/rsyslog.conf` and add the following line at line 23:

```
# Added for OSSIM integration with snort
```

```
$SystemLogRateLimitInterval 10
```

```
$SystemLogRateLimitBurst 500
```

```
#$SystemLogSocketFlowControl on
```

```
#$AddUnixListenSocket /var/snort/dev/log
```

```
local1.info @@10.10.10.12:514
```

- Now go in /etc/snort/snort.conf and add those line at step6 :

```
# syslog
```

```
output alert_fast: snort.fast
```

```
output alert_syslog: LOG_LOCAL1 LOG_INFO
```

- now reboot the system and everything normally work

Find [here](#) the tutorial I followed

c. Add some rules

- The rules we add to make for snort are pretty basic because we just want to block all the traffic from eth0, I make and find those rules:

```
#Network scan alerts
```

```
alert tcp any any -> $HOME_NET any (msg:"FIN Scan"; flags: F; sid: 1000001; rev:001;  
classtype:network-scan;)
```

```
alert tcp any any -> $HOME_NET any (msg:"NULL Scan"; flags: 0; sid: 1000002; rev:001;  
classtype:network-scan;)
```

```
alert tcp any any -> $HOME_NET any (msg:"SYN attack"; flags:S,12; sid: 1000003; rev:001;  
classtype:network-scan;)
```

```
alert tcp any any -> $HOME_NET any (msg:"XMUS attack"; flags:FPU; sid:1000004; rev:001;  
classtype:network-scan;)
```

```
#default alerts by protocol
```

```
alert icmp any any -> $HOME_NET any (msg:"ICMP detected"; sid:10000005; rev:001;  
classtype:icmp-event;)
```

```
alert tcp any any -> $HOME_NET any (msg:"non-scan tcp detected"; sid:10000006; rev:001;  
classtype:tcp-connection;)
```

```
alert ip any any -> $HOME_NET any (msg:"ip detected"; sid:10000007; rev:001;)
```

```
alert udp any any -> $HOME_NET any (msg:"udp detected"; sid:10000008; rev:001;)
```

More information about making rules:

- [Manual](#)
- one tutorial : [http://www.vorant.com/files/EZ\\_Snort\\_Rules.pdf](http://www.vorant.com/files/EZ_Snort_Rules.pdf)
- scan alert explanations and examples:  
[http://www.sersc.org/journals/IJFGCN/vol9\\_no6/32.pdf](http://www.sersc.org/journals/IJFGCN/vol9_no6/32.pdf)

Hyperlinks:

- ➔ Direct download for Jessie : <https://debian.beagleboard.org/images/bone-debian-8.7-iot-armhf-2017-03-19-4gb.img.xz>
- ➔ Flashing with windows :  
[http://elinux.org/RPi\\_Easy\\_SD\\_Card\\_Setup#Flashing\\_the\\_SD\\_Card\\_using\\_Windows](http://elinux.org/RPi_Easy_SD_Card_Setup#Flashing_the_SD_Card_using_Windows)
- ➔ PuTTY: <http://www.putty.org/>
- ➔ Connect a new repository : <https://www.digitalocean.com/community/tutorials/how-to-use-reprepro-for-a-secure-package-repository-on-ubuntu-14-04#install-a-package-from-our-new-repository>
- ➔ SNORT installation : [https://s3.amazonaws.com/snort-org-site/production/document\\_files/files/000/000/122/original/Snort\\_2.9.9.x\\_on\\_Ubuntu\\_14-16.pdf](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/122/original/Snort_2.9.9.x_on_Ubuntu_14-16.pdf)
- ➔ Snort-Ossim link tutorial: [https://s3.amazonaws.com/snort-org-site/production/document\\_files/files/000/000/113/original/Integrating\\_Snort\\_and\\_OSSIM-rev2.pdf](https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/113/original/Integrating_Snort_and_OSSIM-rev2.pdf)
- ➔ Snort manual for rules: <http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node31.html>