UMONS

Faculté des Sciences

Développement d'un pare-feu domestique

Pré-rapport de projet

Activité d' Apprentissage S-INFO-037

Rémy Decocq

Année Académique 2018-2019
Master en Sciences Informatiques, bloc 1
Faculté des Sciences, Université de Mons

Table des matières

1 Présentation de l'*Internet des Objets* **4**
- 1.1 Généralités . . . 4
- 1.2 Caractéristiques des équipements de l'*IoT* . . . 4
  - 1.2.1 Domaines d'application . . . 4
  - 1.2.2 L'environnement *smarthome* . . . 5
  - 1.2.3 Exemples d'équipements . . . 6
  - 1.2.4 Restrictions des équipements . . . 7

2 Présentations des pare-feux **8**
- 2.1 Généralités . . . 8
- 2.2 Différentes positions et fonctions dans l'architecture réseau . . . 8
  - 2.2.1 Les pare-feux niveau réseau . . . 8
  - 2.2.2 Les pare-feux niveau hôte . . . 9
- 2.3 Les types de pare-feu . . . 10
  - 2.3.1 Pare-feu de filtrage (sans état) . . . 10
  - 2.3.2 Pare-feu à état . . . 10
  - 2.3.3 Pare-feu applicatif . . . 11
  - 2.3.4 Pare-feu applicatif proxy . . . 12
  - 2.3.5 Pare-feu identifiant . . . 12
- 2.4 Systèmes de détection d'intrusion (*IDS*) . . . 13
- 2.5 Systèmes de prévention d'intrusion (*IPS*) . . . 15
- 2.6 Les pare-feux nouvelle génération . . . 15
- 2.7 Les pare-feux de référence . . . 16
  - 2.7.1 Sous forme de software . . . 16
  - 2.7.2 Sous forme d'appliance . . . 17

3 La protection dans l'*IoT* **19**
- 3.1 Motivations et exemples . . . 19
- 3.2 Description des besoins en terme de sécurité . . . 20
- 3.3 Modèle d'architecture des équipements . . . 21
- 3.4 Vulnérabilités liées à l'IoT et menaces courantes . . . 21
- 3.5 Possibilités d'amélioration . . . 21
- 3.6 Solutions existantes . . . 21
  - 3.6.1 OWASP Sec framework . . . 21

4 Les pare-feux et l'*IoT* **21**
- 4.1 Différents types d'architecture . . . 21
- 4.2 Les pare-feux domestiques . . . 21
  - 4.2.1 Caractéristiques d'un réseau domestique . . . 21
  - 4.2.2 Attaques possibles et conséquences . . . 21
- 4.3 Application et implémentation . . . 21

5 Mise en pratique : ébauche **21**

Introduction

Depuis maintenant plusieurs années, la connectivité n'a cessé d'évoluer : en se limitant au domaine de l'Internet entre 2000 et 2015, une estimation de l'augmentation du pourcentage de la population mondiale l'utilisant avoisine 40% [5] [2]. Que ce soit dans le cadre d'infrastructures de type « mainframe »ou dans le contexte des ordinateurs personnels, les technologies et équipements relatifs au réseau et aux communications sont devenus indispensables. En raison du potentiel croissant d'interconnexion, les ordinateurs et équipements mis en réseau s'exposent à davantage de menaces. Heureusement, parallèlement à cette évolution, les performances de ces machines qui rejoignent de nouveaux réseaux ont également suivi une amélioration en terme de performances. Cela a permis d'en renforcer la sécurité à plusieurs niveaux, et surtout d'intercepter efficacement les menaces étrangères liées à l'utilisation des réseaux. À l'heure actuelle, les OS utilisés classiquement sur des machines desktop fournissent un pare-feu simple (*Windows Defender*, un utilitaire fournit de base dans MAC OS X, *iptables/Netfilter* ou autre pour les distributions Linux). Ce dernier tournant en arrière plan de façon quasi invisible car il demande peu de ressources par rapport à ce qu'une machine actuelle peut offrir.

En parallèle avec la montée en puissance de ces machines de type desktop, serveurs, etc. s'est développée depuis à peu près les années 2000 la tendance de l'« Internet des Objets », ou encore plus communément abrégé IoT pour *Internet of Things*. Bien qu'assez large, cette dénomination regroupe beaucoup d'objets et de concepts, qu'ils soient virtuels ou non mais possédant un dénominateur commun : la capacité de communiquer en réseau avec d'autres équipements. Cela englobe par exemple la domotique, les outils et capteurs de mesures diverses, les imprimantes et scanneurs en réseau, etc. Tous ces éléments convergeraient idéalement vers une mise en réseau commune, leur permettant de communiquer malgré leur nature hétérogène [14]. Cela peut se faire via le réseau Internet, il n'est pas rare d'orienter ces connections vers un cloud permettant de traiter globalement et intelligemment la masse de données qu'il reçoit de ces équipements [17]. Or, comme évoqué ci-dessus, plus on s'interconnecte et plus on s'ouvre à des attaquants potentiels, ce qui pose problème si rien n'est mis en place pour s'en protéger.

Ce travail aura pour objectif premièrement de faire un état de l'art des dispositifs de protection qui sont actuellement déployés dans l'IoT et plus particulièrement dans le cadre domestique. Il s'agit d'un monde beaucoup plus hétérogène et restreint en terme de ressources que celui des ordinateurs qu'on retrouve classiquement dans ce milieu, de fait il n'est pas toujours possible de réutiliser telles quelles toutes les technologies de protection y attenant. De plus, il faut prendre en considération les spécificités d'une habitation : au centre de tous ces équipements communiquant se trouve l'habitant, une personne n'étant pas forcément habilitée à manipuler et contrôler ces nouvelles technologies. Il sera également question d'établir une vision globale des différentes dispositifs de pare-feux existant et sous quelles formes ceux-ci peuvent être implémentés dans les équipements d'une habitation classique. Deuxièmement, il sera question de mettre en pratique ces connaissances pour développer un système en lien avec ces nouvelles mesures de sécurité inhérentes à l'IoT. Celui-ci fera intervenir les connaissances acquises au préalables sur les pare-feux.

1 Présentation de l'*Internet des Objets*

1.1 Généralités

L'Internet des objets, qu'on désignera par *IoT* pour le terme plus répandu de *Internet of Things*, est une notion englobant un vaste ensemble de dispositifs. Aucune définition formelle n'est acceptée globalement, mais plusieurs organismes ont tenté d'en établir une ébauche. Par exemple, l'ITU-T (ITU Telecommunication Standardization Sector) Y.2060 le définit comme tel :

« *Global infrastructure for the society, enabling advanced services by inter-connecting (physical and virtual) things based on existing and evolving in-teroperable information and communication technologies.* »

Derrière cette définition très générale, on peut distinguer plusieurs sous-groupes d'objets connectés distincts, aux applications tant variées que hétérogènes, dans des domaines et secteurs également très différents. C'est ce qui fait la force et en même temps la faiblesse de cet ensemble d'équipements et services qu'on regroupe derrière le terme *IoT* et que des efforts considérables sont déployés pour inter-connecter au maximum. C'est un domaine d'étude intéressant car il représente littéralement ce qu'on pourrait considérer comme le futur de notre environnement technologique. De fait, les équipements que l'ont peut associer à une partie de l'IoT ont déjà fait leur apparition dans notre quotidien : en 2017 on comptait 8,4 milliards de machines en présentant les caractéristiques et les estimations pour l'année 2020 tendent vers 20,4 milliards d'objets connectés d'après la société d'analyse Gartner [10].

1.2 Caractéristiques des équipements de l'*IoT*

1.2.1 Domaines d'application

Les secteurs dans lesquels l'IoT s'est implanté ces dernières années sont nombreux et très variés : ils s'étendent de l'industrie au domaine des soins de santé en passant par la tendance des *Smart Home*. C'est ce dernier domaine qui est approfondi dans ce travail et qui sera le plus sous-entendu par la suite quand le terme *IoT* est utilisé. La Figure 1 présente une vision d'un schéma global des autres domaines qui gravitent autour du vaste monde de l'IoT.

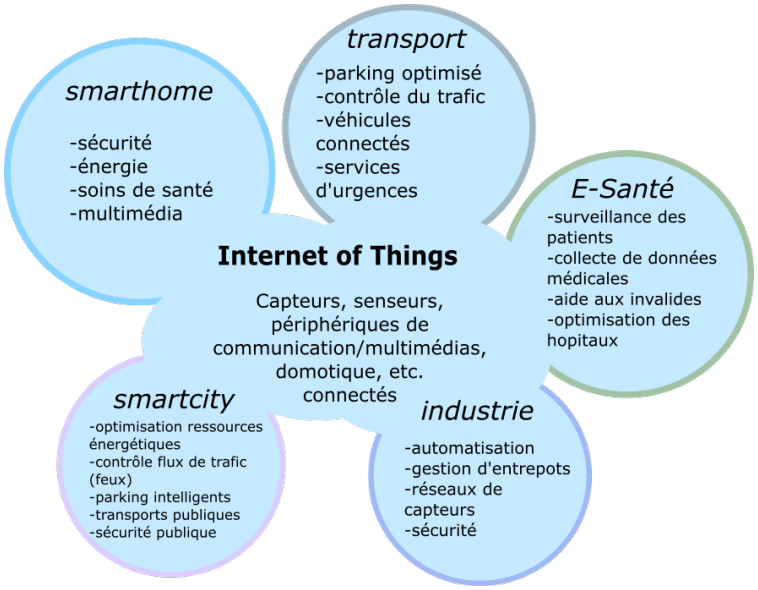

FIGURE 1 – Domaines d'application de l'IoT

1.2.2 L'environnement *smarthome*

Le terme émergeant *Smart Home/smarthome* est une fois de plus très englobant et général. Il n'en existe pas de définition formelle et communément acceptée. Basman M. Hasan et al. [9] en présentent plusieurs. Un résultat les unifiant pourrait être

« Une smarthome est un environnement lié au domicile particulier où plusieurs équipements ou sous-systèmes sont inter-connectés et où les informations qu'ils échangent sont collectées et utilisées afin de surveiller, réguler et automatiser l'écosystème du domicile ».

L'utilisateur en tant que personne physique y vivant est donc au centre de cette architecture, et y siège comme le principal intervenant. Effectivement, l'objectif global du déploiement de tous ces dispositifs est l'amélioration de sa qualité de vie. La notion d'intelligence est intrinsèquement liée avec celle de l'interconnexion de tous ces capteurs et actuateurs déployés dans l'environnement du domicile. Il s'agit d'en récolter et regrouper toutes les données en un point central doté d'une capacité de traitement plus évoluée afin qu'il puisse en tirer une optimisation globale de l'habitation (du système de sécurité, des économies d'énergie, de temps par l'automatisation, etc.).

Une certaine classification fonctionnelle peut être établie pour distinguer de façon plus concrète les différents équipements qui peuvent intervenir dans l'écosystème d'une smarthome. Elle est schématisée par la Figure 2, inspirée de [9]. Ce qui est désigné par *point d'interconnexion* peut dépendre de l'architecture réelle d'une smarthome [17]. Dans la plupart des cas il s'agit d'une machine faisant office de collecteur pour toutes les données transitant dans le domicile et de gateway vers le reste de l'Internet, éventuellement le cloud associé.

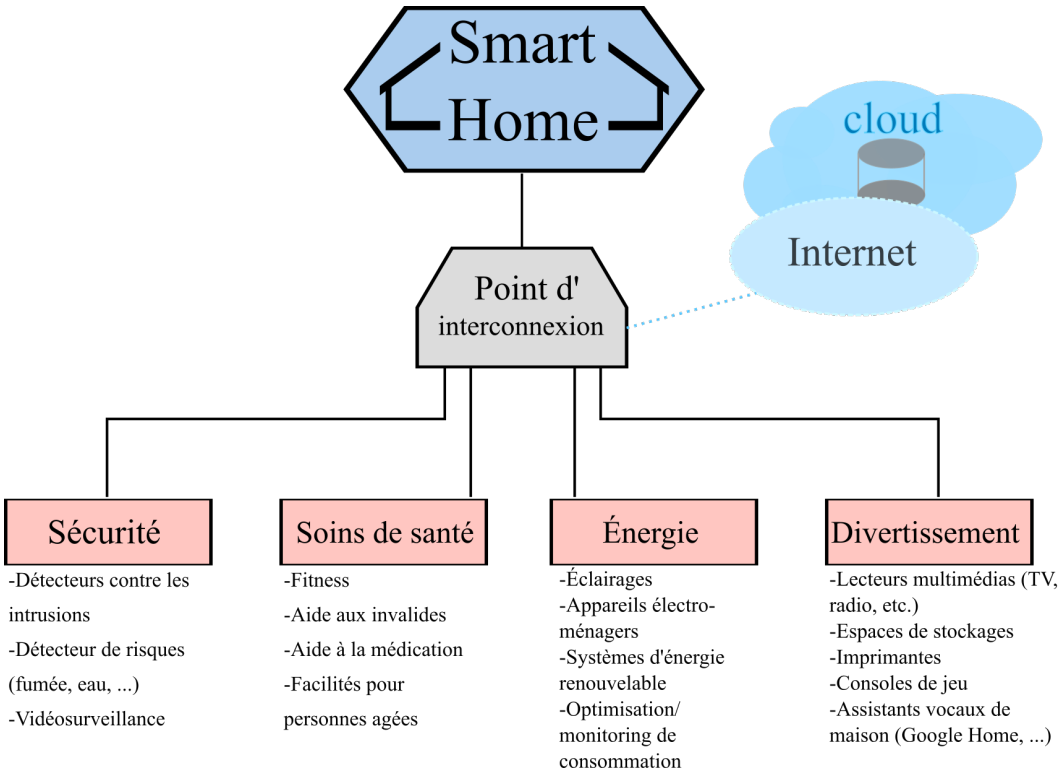

FIGURE 2 – Classification fonctionnelle des équipements *IoT* d'une smarthome

1.2.3 Exemples d'équipements

La perception de ce qu'est l'IoT par le grand public se résume énormément à l'environnement que constitue la smarthome [10]. Il s'agit d'une erreur d'incompréhension, on peut tenter de l'expliciter en analysant ce qui compose cette perception. La Figure 3 en donne une vision générale (tirée d'un sondage effectué aux États-Unis), qui va être étayée par les exemples concrets d'équipements la suivant.

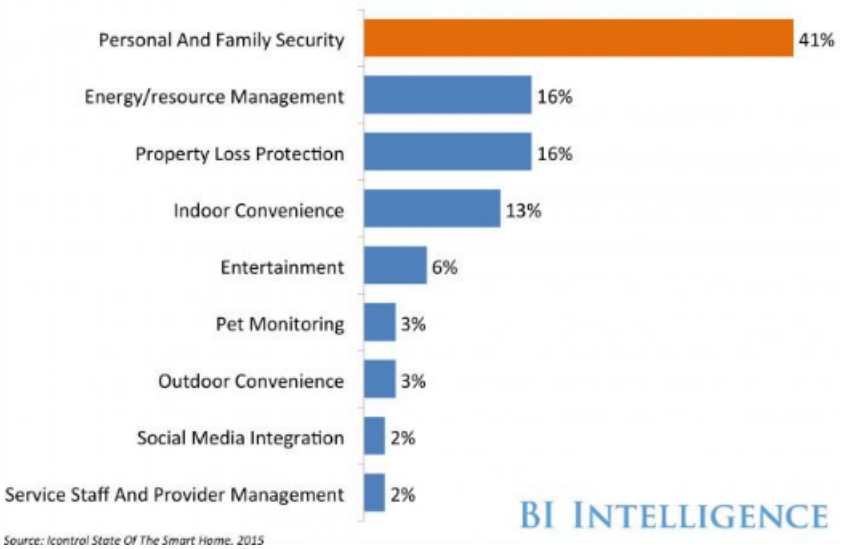

FIGURE 3 – Top des meilleurs apports des *smarthome* tel que perçu par les Américains

/—— A faire : Remplacer par une section plus pertinente (diff architectures smarthome) ——\

Appartenant à la classe sécurité

- Caméras de vidéosurveillance dites IP
- Systèmes de gestion d'alarme à distance
- Verrous de portes intelligents
- Simulateurs de présence et occupation du domicile

Appartenant à la classe soins de santé

- Surveillance des patients à leur domicile (contrôle des mesures médicales)
- Accessoires de fitness : montres, balances connectées et autres
- Outils divers d'aide aux personnes invalides (fauteuils, bracelets de secours, ...)

Appartenant à la classe des énergies

- Luminaires intelligents/automatisés contrôlables à distance
- Frigos, lave-vaisselles, etc.
- Thermostats connectés, compteurs et senseurs énergétiques

Appartenant à la classe du divertissement & multimédia

- Outils de communication : smartphones, babyphones, etc.
- SmartTV, consoles, casques connectés et lecteurs multimédia divers
- Imprimantes et scanneurs en réseau
- Assistants vocaux de maison comme le *google home*

1.2.4 Restrictions des équipements

Malgré le fait que l'ensemble des objets considérés comme appartenant à l'IoT soit très hétérogène, on peut distinguer plusieurs caractéristiques communes à beaucoup d'entre eux. Elles tendent généralement vers ce qui est vu comme une restriction par rapport à un ordinateur type classique (*desktop*). Ces éléments constituent les plus gros freins au développement de la sécurité sur de tels système [8]. Les conséquences de ces restrictions sont discutées plus en détail dans la section 3 de ce document.

Conçus pour satisfaire une unique fonction

Le meilleur exemple est celui des capteurs. Un capteur a pour objectif de faire une mesure d'une grandeur physique (température, pression, etc.), d'en tirer une valeur numérique et de faire remonter via son interface avec le réseau cette information vers une unité centrale. Cette dernière accumule ainsi des mesures provenant des nœuds distribués pour y appliquer un traitement, et c'est à ce plus haut niveau que le processus de décision a lieu s'il est nécessaire. Ce genre d'équipement est généralement minimaliste au possible et ne peut donc pas remplir d'autre tâche.

Requièrent une faible consommation énergétique

Les systèmes embarqués n'ont pas toujours accès à une source illimitée d'énergie, et auront donc une durée de vie limitée à celle de leur batterie. En conséquence, il est souhaitable d'économiser un maximum, ce qui peut se faire en réduisant les temps d'éveil de l'équipement et en optimisant le nombre d'opérations effectuées quand il tourne à plein régime. Dés lors, certains protocoles et algorithmes doivent être adaptés (relatifs aux communications réseaux mais aussi à la sécurité) [17].

Sont contraints en ressources CPU, mémoire et radio

Ces contraintes sont aussi identiques à celles des systèmes embarqués classiques. En plus de celles-ci, on peut mettre en évidence le fait que les radios (quand il s'agit d'une interface sans fil) sont assez faibles et ne permettent pas des communications à haut débit. En résulte également que des techniques comme le saut de fréquence et les algorithmes de chiffrement de type asymétrique sont plus compliquées à mettre en œuvre [15].

Sont programmés à un bas niveau d'abstraction

Étant conçus pour ne remplir que des fonctions spécifiques, certains équipements ne sont pas programmables en utilisant des langages de haut niveau. Par conséquent, ce sont souvent des boîtes noires difficilement manipulables et statiques : les mises à jour et patch de sécurité ne sont pas déployables aisément par les constructeurs [17]. Les seules interactions que l'utilisateur lambda peut avoir avec l'équipement sont celles prévues par l'interface de ce dernier s'il y en a une (pour le configurer, consulter son état, etc.).

2 Présentations des pare-feux

2.1 Généralités

Un pare-feu est un dispositif, virtuel ou matériel, qui surveille et contrôle le lien entre un réseau dit de confiance et un réseau extérieur non fiable. Typiquement, ce réseau potentiellement dangereux est Internet et la zone à protéger est le réseau interne d'une entreprise ou d'une habitation. L'existence des pare-feux est une conséquence du besoin d'outils de protection aux bordures de réseaux distincts contrôlés par des entités différentes. D'une part, il faut garantir que des données internes confidentielles restent à l'intérieur du réseau de confiance. D'autre part, il est nécessaire de filtrer les données entrantes et sortantes de ce réseau de sorte qu'aucune menace ne s'y infiltre par des flux, même initiés depuis l'intérieur du réseau de confiance. Ces filtres sont créés et assemblés à partir de *politiques* ou *règles* définies par défaut ou par les personnes compétentes liées à l'entité gérant le réseau.

2.2 Différentes positions et fonctions dans l'architecture réseau

Deux classes de pare-feux sont distinguables en fonction de ce qu'ils visent à protéger. La première correspond aux pare-feux au niveau réseau (*network firewalls*), situés en bordure de LANs, WANs et intranets. Ceux-ci, de par leur nature de barrière entre réseaux, peuvent également fournir des services plus évolués : système de NAT, gestion de *zones démilitarisées*, service DHCP, etc. [12] La seconde classe agit au niveau des nœuds du réseau eux-mêmes (*host-based firewalls*), protégeant une machine physique et non pas un réseau entier. Ces pare-feux se présentent donc sous forme de logiciels, directement intégrés au niveau du système d'exploitation ou installés à un plus haut niveau.

2.2.1 Les pare-feux niveau réseau

Afin de remplir leur fonction, ces pare-feux sont placés en bordure du réseau à protéger et sont directement liés aux machines qui font office de *gateway* (routeurs). Le trafic échangé entre les réseaux ainsi connectés passe donc par le pare-feu afin d'être analysé et filtré, ce qui peut représenter beaucoup de données à traiter. Le pare-feu doit offrir une vitesse de traitement proportionnelle aux débits et à la qualité des liens qui le traversent afin de ne pas être un goulot d'étranglement. C'est pourquoi ces pare-feux doivent être très efficaces et sont communément portés (partiellement) au niveau matériel. On parle de *hardware-based firewall appliances*, qui sont des machines physiques dont le seul objectif est de remplir les tâches d'un pare-feu le plus efficacement possible. On y retrouve deux composants : la partie applicative software ou firmware remplissant la fonction de pare-feu, qui repose sur la deuxième partie plus basse constituée de juste ce qu'il faut d'un OS particulier (*jeOS - just enough Operating System*). Cet OS est généralement propriétaire, lié au fabriquant du hardware sur lequel les deux parties opèrent et donc optimisé pour garantir les performances requises.

À plus petite échelle, dans un réseau domestique par exemple, on retrouve également des pare-feux directement implémentés dans le routeur qui fait office de *gateway* pour l'habitation. Ceux-ci sont fatalement moins efficaces et complets que les matériels spécialement dédiés à cette unique fonction.

À titre indicatif, la Figure 4 donne un exemple d'une *appliance* pare-feu issue de la gamme des Cisco ASA 55xx-X, la dernière génération que Cisco a introduit sur le marché en 2018 [3]. Il ne s'agit pas du modèle le plus performant de la gamme en comparant les performances et fonctionnalités, mais convient typiquement bien pour être déployé dans une petite entreprise (son prix avoisine 500€). Dans le cas d'une habitation, l'analogue correspondant est dans la majorité des cas le modem distribué par le fournisseur d'accès internet au domicile. La société Proximus fournit par exemple des *b-box*, que leurs clients peuvent configurer par une interface web. Dans ces configurations, il existe bien un onglet intitulé *firewall*, mais celui-ci ne présente que peu d'options de configuration comme le montre la Figure 5. Dans d'autres onglets, on peut trouver d'autres

fonctionnalités qui peuvent y être relatives : restriction d'accès pour certains nœuds en fonction de l'heure (par adresse MAC) et blocage de certains sites par domaine. Cela semble assez faible, les utilisateurs du réseau ont tout intérêt à installer sur leurs équipements personnels des pare-feux pour hôtes.

Débit d'inspection stateful (multiprotocole)	300 Mbps - 750 Mbps max
Débit max. App. Control (AVC) + IPS/NG IPS	125 Mbps
Débit max. VPN avec chiffrement 3DES/AES	100 Mbps
Nombre max. de sessions simultanées	50000

FIGURE 4 – Cisco ASA 5506H-X

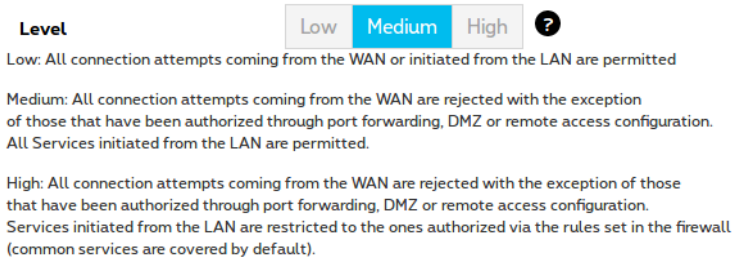

FIGURE 5 – Options de config. du pare-feu pour une b-box 3V

2.2.2 Les pare-feux niveau hôte

Du fait que les pare-feux de cette classe sont généralement déployés sur des machines de type desktop, la dénomination *pare-feu personnel* est aussi utilisée. Ces ordinateurs sont généralement munis de tels pare-feux par défaut ou peuvent tout du moins supporter leur installation par l'utilisateur si l'OS utilisé est classique (Windows, MAC OS X, ...). Le pare-feu agit jusqu'au niveau applicatif [13] et sous forme d'un *service* ou d'un *daemon* en fonction du système d'exploitation utilisé. Cela permet une proximité étroite avec l'OS et les autres processus. Le pare-feu personnel est donc capable de contrôler le trafic réseau demandé par chaque application et de détecter les menaces engendrées par certains flux. Ces dernières peuvent être entrantes ou sortantes : une machine extérieure qui tente d'établir une connexion suspecte ou un exécutable sur la machine à protéger qui initie un trafic avec une cible blacklistée, par exemple.

Les principales fonctionnalités d'un pare-feu personnel devraient être au minimum les suivantes [12]

- Bloquer les attaques et comportements dangereux du réseau extérieur : scanning des ports ouverts, attaques par fragmentation, *IP Spoofing*, etc.
- Stopper les menaces venant de l'intérieur : un exécutable comme un *malware* ou un *spyware* qui tente d'établir une connexion vers l'extérieur doit être bloqué et mis en quarantaine
- Présenter de l'automatisation car d'une part car un utilisateur non-expérimenté pour sa configuration doit quand même rester protégé et d'autre part il doit pouvoir se mettre à jour automatiquement
- Agir au niveau applicatif : un malware pourrait utiliser le port web 80 pour se répandre par exemple, pour détecter cela il faut analyser les payloads des paquets et disposer d'une base de données à jour pour y déceler des patterns malicieux
- Alerter l'utilisateur quand un événement survient, et le logger avec suffisamment d'informations pour qu'il puisse prendre une décision adaptée
- Éviter les *faux-positifs* (bloquer du trafic légitime)

Que ce soit en entreprise ou dans une habitation, il est donc probable que l'on retrouve des pare-feux appartenant à ces deux classes distinctes là où ils sont efficaces (voir Figure 6). Il ne s'agit que d'échelles différentes, qui impliquent également des intervenants différents. En entreprise, l'administrateur sécurité/réseau configure du matériel spécifique afin de sécuriser les frontières de son réseau tout en préservant ses performances. Il est aussi possible qu'il installe dans les machines internes de son réseau des pare-feux assez puissants pour empêcher les propagations des attaques déclenchées. Dans le domicile, l'habitant est sommairement protégé par le routeur fournit par son FAI qui inclut un pare-feu réseau par défaut. S'il est

un minimum expérimenté, il sera à même d'installer et configurer des pare-feux personnels sur chacun de ses équipements connectés.

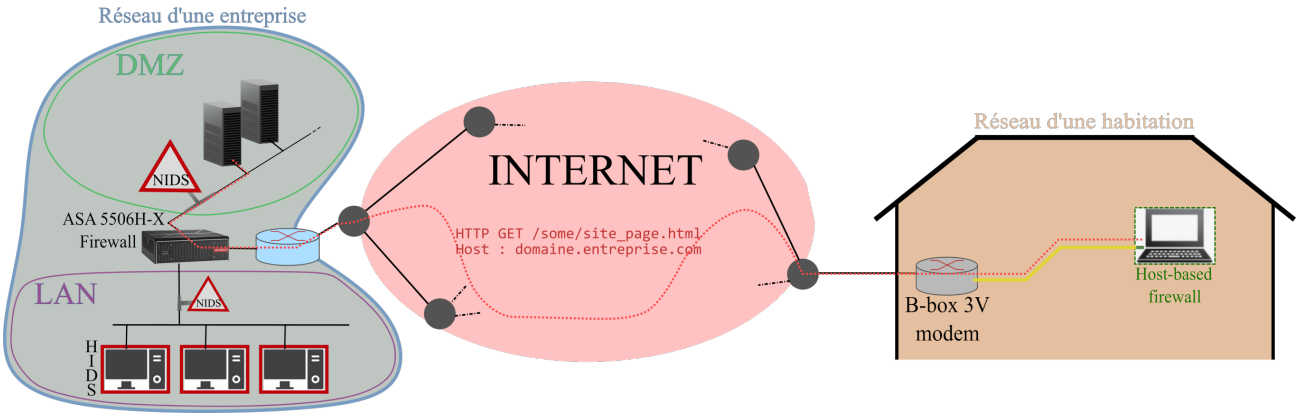

FIGURE 6 – Schéma récapitulatif des places des différents dispositifs de protection dans un réseau

2.3 Les types de pare-feu

2.3.1 Pare-feu de filtrage (sans état)

Aussi désignés dans la littérature par le terme *stateless*, ces pare-feux sont les plus rudimentaires, le premier prototype étant celui élaboré par Jeffery Mogul en 1989 [19]. Généralement, ces pare-feux sont rapides mais peu efficaces car facilement dupés. Un pare-feu de ce type inspecte chaque paquet individuellement et détermine s'il peut passer sur base d'un ensemble de règles écrites que ses en-têtes vérifient ou non. Ces règles portent sur les en-têtes TCP/IP (également UDP), les options y attenant et l'interface d'entrée. Plus en détail : les adresses IP source et destination, les ports source et destination au minimum seront soumis aux filtres du pare-feu. Il y a plusieurs problèmes avec cette approche :

- Il n'y a aucune vérification au dessus de la couche transport : la partie applicative peut contenir n'importe quoi
- Le pare-feu n'est pas dynamique : il n'apprend rien du trafic qu'il laisse passer. Aucun état n'est retenu, alors que par exemple TCP est un protocole lié à une machine à état (orienté connexion) dont il pourrait être possible de garder une trace
- Les règles sont très redondantes à écrire pour être efficace, très peu modulables et facilement sujettes aux erreurs et oublis

Les paquets ICMP peuvent également faire l'objet d'un filtrage du même acabit [12]. Ce type de pare-feu tombe en désuétude car il est trop simple à duper par des attaquants.

2.3.2 Pare-feu à état

Le but de ce type de pare-feu dit *stateful* est d'améliorer les techniques de filtrage sans état en maintenant de l'information sur les flux passant au travers, principalement TCP. Effectivement, au contraire d'UDP, TCP est un protocole qui garde un état qui se traduit dans certains champs de ses headers. Dès lors, le pare-feu peut accéder à cette information, l'analyser et l'utiliser pour en déduire dans quel état est actuellement la connexion entre les deux processus communiquant sur les deux hôtes impliqués. Cela peut se traduire par l'inspection des flags (SYN, FIN, ...), des numéros de séquences et acquittements qui permettent au pare-feu de contrôler en conséquence le contenu d'une table interne des connexions.

L'établissement d'une connexion TCP se fait comme suit : lors de la réception du premier paquet TCP par le pare-feu, celui-ci va le soumettre à ses règles de filtrage définies comme dans un pare-feu sans état. Si le paquet est valide, il est retransmis pour poursuivre librement sa route et une entrée est ajoutée dans la table interne du pare-feu. Celle-ci va changer d'état à chaque paquet de la communication reçu, jusqu'à arriver dans un état *established* correspondant à la fin du 3-way handshake TCP. Tous les paquets suivants pourront alors être traités rapidement en faisant matcher les headers à ceux de l'entrée en état estbalished dans la table (qui sont donc théoriquement également légitimes). Cette façon de procéder est plus efficace que de soumettre chaque paquet individuellement aux règles de filtrage qui peuvent être complexes et demander beaucoup de travail et donc gourmandes en temps.

Dans le cas d'une communication utilisant UDP, le contrôle possible est moins fin car il ne s'agit pas d'un protocole à état et les échanges ne sont pas bidirectionnels. Dans ce cas, lorsque le premier paquet d'une transmission arrive en entrée du pare-feu, il est soumis aux règles de filtrage. S'il est considéré comme correct, le pare-feu ajoute une entrée dans sa table pour les couples d'adresses et ports et considérera les suivants comme légitimes. Cette entrée expirera après un temps donné sans autre nouveau paquet la matchant.

2.3.3 Pare-feu applicatif

Ce type de pare-feu peut être considéré comme une extension complète aux simples pare-feux à états dans le but d'en améliorer la fiabilité. Là où ces derniers sont capables de déterminer quels protocoles sont utilisés et autorisés sur chaque port, les filtres ajoutés au niveau applicatif peuvent en plus déduire à quelles fins sont utilisés ces protocoles (l'inspection va jusqu'à la couche 7 du modèle OSI). Il s'agit cependant de plus que de simples filtres, c'est une technologie à part entière qui est utilisée : l'inspection profonde des paquets ou *deep packet inspection* (DPI). Les pare-feux utilisant la DPI mêlent les fonctionnalités d'un pare-feu à état avec les systèmes de détection et prévention d'intrusions (sous-sections 2.4 et 2.5). La décision de bloquer ou de laisser passer le paquet tient compte de ce qu'il contient réellement au niveau données et de l'interprétation qui en est faite pour déterminer s'il représente un danger.

Par exemple, les *Web Application Firewalls* (WAF) sont une sous-catégorie des pare-feux applicatifs qui opèrent un filtrage sur le trafic HTTP. Ils peuvent donc filtrer le trafic sur base de certaines règles d'accessibilités fixées en bloquant des requêtes aux URL douteuses ou considérées comme non éthiques vis-à-vis de l'organisme maintenant le réseau par lequel elles transitent. Outre cela, les WAFs permettent également de se protéger contre les attaques dites de type *parameter tampering* (injection SQL, cross-site scripting, etc.) [18], ainsi que d'inclure des filtres anti-spam pour les courriels [7].

Plus généralement, les pare-feux applicatifs permettent de se protéger des virus, vers et tentatives d'*exploits* de faiblesses connues des systèmes à protéger [1] [18]. Un autre avantage d'analyser jusqu'à la couche 7 grâce à la DPI est que certains protocoles (FTP par exemple) utilisent la couche applicative pour transmettre des informations relatives aux couches plus basses (adresses IPs, ports, etc.) et qui seront impliquées dans des communications passant par le dispositif pare-feu [7]. Comme illustré par l'exemple des WAFs, le principal défaut inhérent à leur nature réside dans le fait que pour chaque type de trafic applicatif que l'on souhaite contrôler, un travail spécifique doit être opéré (établir des règles dédiées par exemple) [4].

2.3.4 Pare-feu applicatif proxy

Ces pare-feux mêlent l'inspection au niveau applicatif et le rôle de proxy pour les flux qui sont destinés à le traverser originellement. Ainsi, il devient réellement un intermédiaire entre les deux parties communiquant, interceptant les paquets pour les analyser puis les retransmettant vers le destinataire comme s'il en était la source [12]. Avant cette retransmission, le paquet peut être inspecté du point de vue de toutes ses couches et déterminé comme étant dangereux en fonction des opérations de filtrages effectuées dans le proxy. Un contrôle très fin peut donc être opéré. Cela garantit un niveau de sécurité plus haut que les pare-feux de filtrage pur qui traitent les paquets à la volée [6].

Cette approche possède cependant trois gros désavantages :

- Clairement, l'impact sur les performances du réseau n'est pas insignifiant (surtout si le pare-feu n'est basé que sur du software pur et pas sur une machine adaptée)
- À chaque nouveau protocole applicatif que l'on souhaite pouvoir filtrer de la sorte, un nouveau dispositif proxy correspondant doit être pensé et développé pour en inspecter le trafic
- Si un proxy présente une faille de sécurité, un assaillant peut l'utiliser comme un vecteur d'attaque et prendre le contrôle du système sur lequel il tourne. S'il parvient alors à désactiver les services relatifs au pare-feu, la voie vers ce qu'on souhaitait protéger se retrouve toute ouverte

2.3.5 Pare-feu identifiant

Pour ce type de dispositif, on veut pouvoir définir des règles de filtrage en fonction de l'utilisateur ou le groupe qui se cache derrière un paquet ou un flux. Plusieurs schémas d'association existent, définissant ce à quoi la notion d'un utilisateur correspond. Par exemple certains de ces pare-feux comme authpf sous OpenBSD, qui étant placés sur un gateway, exigent qu'avant d'autoriser un flux l'utilisateur d'un hôte communiquant doit établir une connexion SSH avec ce dernier. Un autre type d'association possible peut se faire par adresse MAC.

Les pare-feux destinés aux entreprises développés par Cisco (ASA) [11] et Palo Alto Networks [13] utilisent des gestionnaires d'annuaires d'utilisateurs basés sur les services Windows Active Directory. Dans ces bases de données utilisateurs, on retrouve les différentes adresses IP associées à un utilisateur particulier ainsi que les groupes qu'ils composent. Les interactions avec le pare-feu sont transparentes à ce niveau, en effet ce dernier travaille avec les objets « utilisateurs »et non plus directement les IPs (bien que cela reste possible). Procéder de cette façon apporte plusieurs avantages [11] :

- Cela simplifie la création et la gestion des politiques de sécurité du pare-feu
- Offre la possibilité d'identifier facilement les utilisateurs utilisant le réseau
- Simplifie la surveillance des activités des utilisateurs du réseau et permet d'identifier la source de menaces

2.4 Systèmes de détection d'intrusion (*IDS*)

Ce type de systèmes, abrégé par IDS pour *intrusion detection system*, est analogue à ce qu'est un système d'alarme intérieur dans l'environnement d'une habitation : quand les dispositifs de protection mis en place à l'entrée sont contournés (une présence non permise est détectée à l'intérieur), l'alarme est lancée et des actions sont éventuellement prises en conséquence. Les IDSs sont donc des sentinelles qui surveillent le réseau interne, logiquement placées après le pare-feu dans le sens entrant. Ces systèmes se présentent sous la forme d'outils spécialisés dans l'interprétation des logs des routeurs, pare-feux, serveurs et autres agents du réseau interne. Les IDSs sont épaulés par une base de données des signatures d'attaques déjà connues et y comparent le contenu des logs afin de trouver des *patterns* qui matchent [12]. Dans une telle situation, plusieurs actions allant de simplement alerter l'administrateur réseau à couper les accès réseaux des machines peuvent être déclenchées, en fonction du degré de certitude (correspondance forte entre les signatures) et de la menace.

Tout comme les pare-feux, les IDSs peuvent être des ressources software ou reposer sur du hardware spécifique. Dans le cas ou il s'agit de software, ils sont établis sur la même machine que le pare-feu, les proxys ou autres dispositifs de bordure de réseau. S'ils se présentent sous forme d'équipements hardwares spécifiques, ils sont installés de sorte à contrôler et surveiller de près un de ces dispositifs sensibles. Le trafic tant entrant que sortant peut être analysé par les IDSs, car les attaques peuvent autant venir de l'extérieur que se déployer depuis l'intérieur (chevaux de Troie, spywares, etc.).

L'analyse des événements repose communément sur deux techniques. La première, la détection par signature (*signature detection*), utilise une base de données de signatures d'attaques déjà connues et se base sur le trafic et les patterns observés pour établir un match avec ces signatures. La seconde approche est celle de la détection d'anomalies (*anomaly detection*). Il est question d'utiliser des heuristiques afin de distinguer les situations et comportements anormaux, sur base de profils types construits par analyses statistiques, agencements de règles ou réseaux neuronaux. Outre la technique utilisée, trois catégories d'IDSs existent en fonction de leur place dans l'architecture réseau (leur mode de fonctionnement diffère en conséquence) :

- Surveillance du réseau (*NIDS* : *Network-based IDS*)
- Surveillance des systèmes du réseau (*HIDS* : *Host-based IDS*)
- Surveillance distribuée du réseau (*NIDS* : *Distributed IDS*)

Les NIDSs

Ces dispositifs surveillent le réseau ou un segment de ce réseau, sous forme d'un équipement intermédiaire de capture (ses interfaces sont en mode écoute pour l'entièreté du trafic transitant par le segment). Il est important d'avoir plusieurs unités de surveillance distinctes dans le cas où le réseau est scindé en plusieurs modules/zones. Par exemple, un serveur web d'une entreprise pourrait être infecté et servir de plateforme de lancement d'une attaque depuis le réseau interne, vers un autre module contenant des serveurs internes.

Deux manières de procéder à l'analyse sont possibles : soit en mode *in-line* soit en mode *off-line*. Le cas de la capture du trafic par interface décrit ci-dessus correspond au mode in-line. Le trafic est analysé en temps réel, ce qui permet une plus grande réactivité mais peut constituer un goulot d'étranglement si le processus de décision est coûteux. Le mode off-line est donc plus avantageux à ce niveau, puisque le processus d'analyse et décision est opéré sur des données stockées et non pas à traiter à la volée. Une inspection plus fine est alors envisageable, cependant le principal défaut de ce mode de fonctionnement reste le manque de réactivité à une attaque.

Les HIDSs

Les IDSs orientés systèmes, *Host-based IDS*, sont élaborés dans le but de protéger uniquement l'hôte sur lequel ils sont déployés et non plus un segment du réseau. Ils ciblent également de façon précise le trafic qu'ils doivent surveiller en fonction de la nature du système. Par exemple, si l'hôte ne maintient aucun service DNS, il est inutile d'analyser des requêtes DNS qui lui parviendraient pour y déceler un menace exploitant une faille connue dans le protocole DNS. Puisque les HIDSs vont s'exécuter sous forme d'un processus (*daemon*) sur la machine hôte qui elle-même fonctionne sous un OS classique, un lien doit être établi entre les deux pour que l'HIDS puisse surveiller le système et les interfaces réseaux. Certains vont même jusqu'à rechercher les intrusions dans le noyau de l'OS.

D'une part, un HIDS peut surveiller le comportement du système dynamiquement en récupérant des informations que ce dernier met à sa disposition à la manière d'un antivirus. Différents indicateurs sont à interpréter :

- Activité de la machine même : processus qui y vivent, ressources qu'ils consomment (CPU, RAM, réseau etc.), modification dans les comptes utilisateurs
- Activité des utilisateurs : commandes entrées, programmes lancés, tentatives d'accès à des ressources non autorisées, passage au compte administrateur
- Patterns d'exécution ou de procédure de déploiement des vers, virus, chevaux de Troie (shell ouvert simultanément à l'ouverture d'un fichier, accès anormaux aux interfaces réseau, ...)

D'autre part, certaines sections plus critiques du système peuvent être intéressantes à analyser finement, car c'est généralement là qu'un attaquant voulant prendre le contrôle de la machine va laisser des traces en y installant ses outils softwares. Un HIDS voulant s'assurer que des sections critiques (système de fichiers, registres, ...) ne sont pas infectées va maintenir une base de données de leur évolution dans le temps, sous forme d'attributs et sommes de contrôles (*checksums*). Ces dernières permettent d'assurer l'intégrité des ressources concernées en en comparant les valeurs régulièrement dans le temps, et notifiant les différences observées.

Les DIDSs

Les DIDSs se présentent sous la forme d'une architecture distribuée d'IDSs (surveillant des segments de réseaux et/ou hôtes), et d'une unité centrale de management qui récolte toutes les informations de ces dispositifs déployés. Ainsi, la machine centrale peut maintenir une vaste base de données centralisée, représentant l'état global du réseau à protéger. Cette approche présente plusieurs avantages :

- moins de faux positifs car plus de données pour justifier une prise de décision (agrégation des événements et attaques)
- mises à jour et distribution de la base de données des signatures aisées
- centralisation des alertes/logs, contrôle global avec une vue d'ensemble
- réponses aux événements plus efficaces, IDSs plus simples à administrer en conséquence (ajout de nouvelles règles suite à des brèches découvertes, blacklisting d'IPs, etc.)

Afin de transmettre les alertes générées jusqu'à l'unité centrale efficacement, les différents nœuds IDSs du réseau devraient communiquer de façon homogène. À cette fin, un standard a été écrit sous le nom de format IDMEF (*Intrusion Detection Message Exchange Format*, RFC 4765). Ce langage d'alertes est lisible par l'humain, utilisant le format XML. Utilisé par tous les nœuds qui effectuent la tâche de détection d'intrusion (quelque soit le type et l'implémentation), l'unité centrale qui récolte toutes les alertes peut utiliser un outil de pilotage de la sécurité du réseau global comme Prelude. Cet outil normalise, trie, agrège, corrèle et en tire les conclusions sur des décisions à prendre pour protéger le réseau, tout en fournissant une interface (et

des logs) via laquelle l'administrateur peut surveiller ce qui se passe globalement.

2.5 Systèmes de prévention d'intrusion (*IPS*)

Dans la section précédente, il était question de détection des attaques et pour certaines d'entre elles des actions à effectuer en contre-mesure pour bloquer la menace. Cela relève du domaine d'application des systèmes de prévention d'intrusion, aussi abrégé IPS pour *intrusion prevention systems*. Tout comme pour les IDSs, deux types d'IPS peuvent être distingués en fonction de leur place dans l'architecture réseau : les *host-based IPS* agissent au niveau local pour un hôte, tandis que les *network-based IPS* défendent un réseau ou un de ses segments internes. De fait, IDSs et IPSs vont de paire puisque les contre-mesures prises par un dispositif IPS sont basées sur ce que l'IDS détecte et lui transmet comme information.

En plus de lancer une alerte à destination d'un administrateur et écrire des logs complets, les actions qu'un IPS peut effectuer suite à la détection d'une menace sont généralement les suivantes :

- Jeter les paquets détectés comme contribuant à cette attaque
- Bloquer tout trafic issu de la même adresse IP/du même utilisateur que celui considéré comme attaquant (mettre fin à la connexion TCP impliquée)
- Reconfigurer le pare-feu associé pour qu'il puisse à l'avenir bloquer le trafic relatif à cette attaque
- Si l'IPS est lié à un dispositif faisant du proxy, il peut agir sur le contenu des paquets pour neutraliser la menace (par exemple ôter un fichier infecté joint à un mail)

2.6 Les pare-feux nouvelle génération

Abrégés par NGFW (*next-generation firewalls*), ces pare-feux font suite à ce qu'on distingue comme la troisième génération dans l'évolution des dispositifs pare-feux [4], c'est-à-dire ceux qui appliquent les concepts de filtrage au niveau applicatif décrits dans la sous-section 2.3.3. Les NGFW effectuent une analyse plus fine que ces derniers en utilisant les technologies avancées de *deep packet inspection*. Les services qu'offrent la nouvelle génération de pare-feux s'étendent encore au-delà de la simple fonctionnalité d'inspection des paquets, et sont devenus des systèmes très complexes implémentant d'autres dispositifs liés à la sécurité dont les suivants :

- Support des technologies de (dé)chiffrement utilisées classiquement dans les réseaux
- Les systèmes de management et contrôle sur base des identités des utilisateurs inspirés des pare-feux identifiants
- Les pare-feux destinés aux flux web (WAFs), le filtrage d'URLs selon des politiques de sécurité et éthique
- Des moyens de contre-mesures aux attaques empruntés aux IPSs
- Les environnements virtuels (*sandboxing*) de tests pour déceler dans un flux suspect une attaque non-reconnue [13]
- De vastes bases de données sur le cloud interrogées efficacement via un service spécialisé fourni avec le pare-feu (environnement WildFire et similaires)

2.7 Les pare-feux de référence

2.7.1 Sous forme de software

Ces pare-feux sont destinés à être installés et utilisés par-dessus des systèmes d'exploitation génériques tels que Windows, distributions Linux et BSD, etc. En plus de protéger la machine sur laquelle ils opèrent des menaces extérieures, certains d'entre eux intègrent d'autres fonctionnalités telles que du *traffic shaping* ou la gestion d'un système de NAT.

Windows

Windows Firewall est intégré par défaut avec les versions de l'OS Windows postérieures à *Windows XP*. Il s'agit d'un pare-feu qui se veut « user-friendly » : par défaut il présente une liste de profils préconfigurés pour que l'utilisateur puisse sélectionner celui qui convient le plus à ses besoins (*Public*, *Private* et *Domain*). Il agit comme un pare-feu à état basique mais assez complet dans sa fonction restreinte cependant, en plus d'être simple à utiliser et de disposer d'une interface graphique. Le fait qu'il soit directement incorporé dans le système permet un contrôle fin de quelles applications sont autorisées pour quels types de trafics.

Linux

Netfilter est la solution de base de filtrage de paquets inclue dans le noyau Linux. Plusieurs modules liés au noyau sont également fournis pour servir d'intermédiaire dans la configuration de Netfliter, notamment sous la forme de tables de règles en chaines à appliquer séquentiellement sur les entrées à filtrer. Ces modules sont accessibles depuis l'utilitaire *iptables*, sous les noms de ip_tables, ip6_tables, etc.

nftables a pour objectif de remplacer Netfilter à terme car il est sensé apporter plus de performances principalement. L'utilitaire permettant de faire le lien avec cette nouvelle solution implémentée dans le noyau est réduit à *nft* (réduction du nombre de module par rapport à Netfilter).

Shorewall repose sur Netfilter, faisant office de couche d'abstraction pour faciliter la configuration des règles qui est assez laborieuse avec les outils comme iptables. Il permet l'écriture des règles alimentant les tables par un mécanisme de fichiers de configuration. C'est un software complet, il propose des fonctionnalités classiques telles que du NAT, du *traffic shaping*, un support VPN et autres. Une autre fonctionnalité notable est la possibilité de partitionner le réseau en zones distinctes (ou pouvant se recouvrir l'une l'autre), permettant le contrôle de toutes les connexions entre chaque paire de zone à considérer. Cela permet notamment l'établissement aisé de zones démilitarisées (DMZ).

BSD

Packet Filter (pf) est comparable à Netfilter pour les noyaux Linux. C'est une solution de filtrage de paquets à état, incluant d'autres fonctionnalités classiques telles que la gestion de NAT, des mécanismes liés à la *qualité de service* (QoS), une gestion des logs avancée en offrant leur configuration règle par règle. D'autres utilitaires sont venus s'y greffer, tel que *authpf* qui permet un système de gateway authentifié (voir les pare-feux identifiants). Packet Filter a fait l'objet de port vers de nombreux autres systèmes d'exploitation, notamment Apple Mac OS X et iOS, NetBSD et Solaris.

2.7.2 Sous forme d'appliance

Le terme *appliance* désigne un appareil/système constitué du strict minimum de composants pour remplir au mieux une fonction spécifique. Du point de vue des dispositifs pare-feux, on peut distinguer trois catégories d'appliances :

- les pare-feux destinés aux systèmes embarqués, par exemple des équipements tels que présentés dans la section 1 de ce travail (les plus adaptés aux ressources limitées de ces équipements).
- les pare-feux logiciels destinés à fonctionner au dessus d'une couche d'abstraction du hardware spécifique (ou même sur une machine virtuelle), basés sur un noyau d'OS qui est transparent à l'utilisation
- les pare-feux reposant sur du hardware (machines dédiées vendues telles quelles) et qui utilisent le matériel et les équipements spécifiques construits à cette fin apportant de meilleures performances

Généralement, il s'agit de solutions destinées aux entreprises de par leur prix et leur nature. Effectivement, comme ce sont des équipements/dispositifs dédiés (presque) uniquement à la fonction de pare-feu (et à la sécurisation du réseau en général), ils sont souvent déployés comme des pare-feux niveau réseau. À noter que beaucoup de ces pare-feux tournent sur un OS propriétaire de l'entreprise qui les fabrique (Cisco, Check Point, etc.). Cela permet aux fabricants d'optimiser leurs produits tant au niveau performance que sécurité, restreignant leur OS au strict nécessaire (certains d'entre eux sont basés sur des noyaux déjà existants).

Appliances sur systèmes embarqués

IPFire est dérivé du noyau Linux et est capable de remplir les toutes les fonctions basiques d'un routeur utilisé en conjugaison avec un pare-feu, sous des contraintes de ressource de 1GHz pour le CPU, 1GB de RAM et 4GB d'espace disque. De fait, IPFire peut être utilisé sur des systèmes comme une carte Raspberry PI ou dans des environnements virtuels. IPFire est modulable avec des add-ons gérés par un utilitaire de management, *Pakfire*. Cependant, la version minimaliste supporte déjà des fonctionnalités telles que l'inspection des paquets à état, le proxying, la prévention d'intrusions, la segmentation en zones du réseau, la gestion de QoS, les services DHCP et DNS, etc.

IPCop est un pare-feu open-source assez minimaliste utilisant Netfilter en arrière-plan. Les pré-requis pour le faire tourner sont : 200MHz de fréquence CPU, 64 Mo de RAM et 800 Mo d'espace de stockage. Même avec cette configuration minimale, il assure les fonctionnalités de proxy et d'IPS en plus du filtrage à état classique. Des modules peuvent être ajoutés en fonction des services supplémentaires requis : services DHCP ou DNS, *trafic shaping*, liste d'accès utilisateurs, etc. IPCop est une branche divergente du projet SmoothWall, ce dernier s'étant développé de telle sorte qu'il n'est plus associable à un pare-feu *embarqué* en plus d'être passé en partie sous licence commerciale.

Appliances orientées software

pfSense se repose sur FreeBSD (bien qu'aucune connaissance sur ce dernier ne soit requise, puisque pfSense est une appliance à utiliser tel quel). Le software peut être installé sur une machine physique dédiée ou dans un environnement virtuel. Il est complètement open-source, et a la prétention d'offrir tous les services que les grandes distributions incluent dans leur appliances physiques en un seul software, gratuit et extensible. pfSense opère comme un routeur et pare-feu sur la machine dédiée, ces services étant alors configurables et mis à jour via une interface web.

Untangle est quant à lui basé sur le noyau Linux. Il est également déployable sur une machine simple dédiée ou virtuelle. Ses fonctionnalités incluent entre autres les classiques d'un pare-feu nouvelle génération (filtrage du contenu, IPS, identification utilisateur, etc.) mais aussi la possibilité de faire du cache web, du *load balancing* et du routage. Cependant, seule une version légère du software est distribuée gratuitement.

Softwares reposant sur des appliances hardware

CheckPoint Software est une entreprise qui vend des machines dédiées à la protection des réseaux ainsi que les solutions logicielles qui les accompagnent. Le software *GAiA* fait office d'OS pour les appliances hardwares vendues par CheckPoint pour lesquels il est optimisé, mais peut également être déployé sur une appliance virtuelle compatible hébergée sur une machine qui en est capable.

Cisco offre des services similaires avec ses pare-feux nouvelle génération ASA 5500-X. Le software Cisco *Adaptative Security Appliance* est l'OS dédié principalement aux machines de la famille ASA (encore une fois des solutions de virtualisation existent). La sur-couche *Cisco Firepower Threat Defence* (FTD) est un software intégrant les fonctionnalités d'ASA et des services axés IDS/IPS nouvelle génération de FirePOWER.

Ces solutions sont généralement utilisées dans des réseaux de grande taille, donc en entreprise. Les appliances hardwares sont vendues telles quelles à prix élevé, et le software fourni avec y est intégré pour en garantir les performances, tant au niveau des débits que de la détection et du blocage des menaces. Cependant, il n'est pas vain de s'y intéresser car certaines technologies sur lesquelles elles sont basées sont parfois réutilisables à plus petite échelle (dans le cadre de solutions pour protéger un réseau d'habitation entre autres). Par exemple, FirePOWER a été développé à partir du software open-source Snort et utilise les bases de données de définitions de signatures virales issues d'un utilitaire anti-virus nommé ClamWin, libre également.

3 La protection dans l'*IoT*

3.1 Motivations et exemples

À priori, dans le cadre d'une smarthome, les équipements relevant de l'IoT sont plutôt des petits accessoires, voire gadgets (montres, lampes, balances connectées par exemple). On pourrait donc se dire qu'introduire des mécanismes de sécurité avancés dans de tels petits systèmes n'est pas primordial, l'aspect fonctionnel étant plus mis à l'honneur. C'est d'ailleurs ce qu'ont fait beaucoup de constructeurs, mettant sur le marché des équipements possédant des failles de sécurité importantes ou des faiblesses au niveau du design de conception. Un exemple interpellant est le fait que pour un équipement qui possède une interface accessible par un portail de sécurité de type identification *user/password*, les valeurs d'usine par défaut sont laissées telles quelles, l'utilisateur n'étant jamais invité à les modifier [16]. Ainsi, il existe des listes établies de ces identifiants par défaut en fonction du constructeur (pour les webcams par exemple) qui, combinées avec des outils puissants comme Shodan ou Censys, constituent des brèches dans la protection de l'habitation pour les attaquants qui savent les exploiter. La Figure 7 illustre la déconcertante facilité avec laquelle il est possible d'attenter à la vie privée des utilisateurs inconscients (une unique requête a été nécessaire).

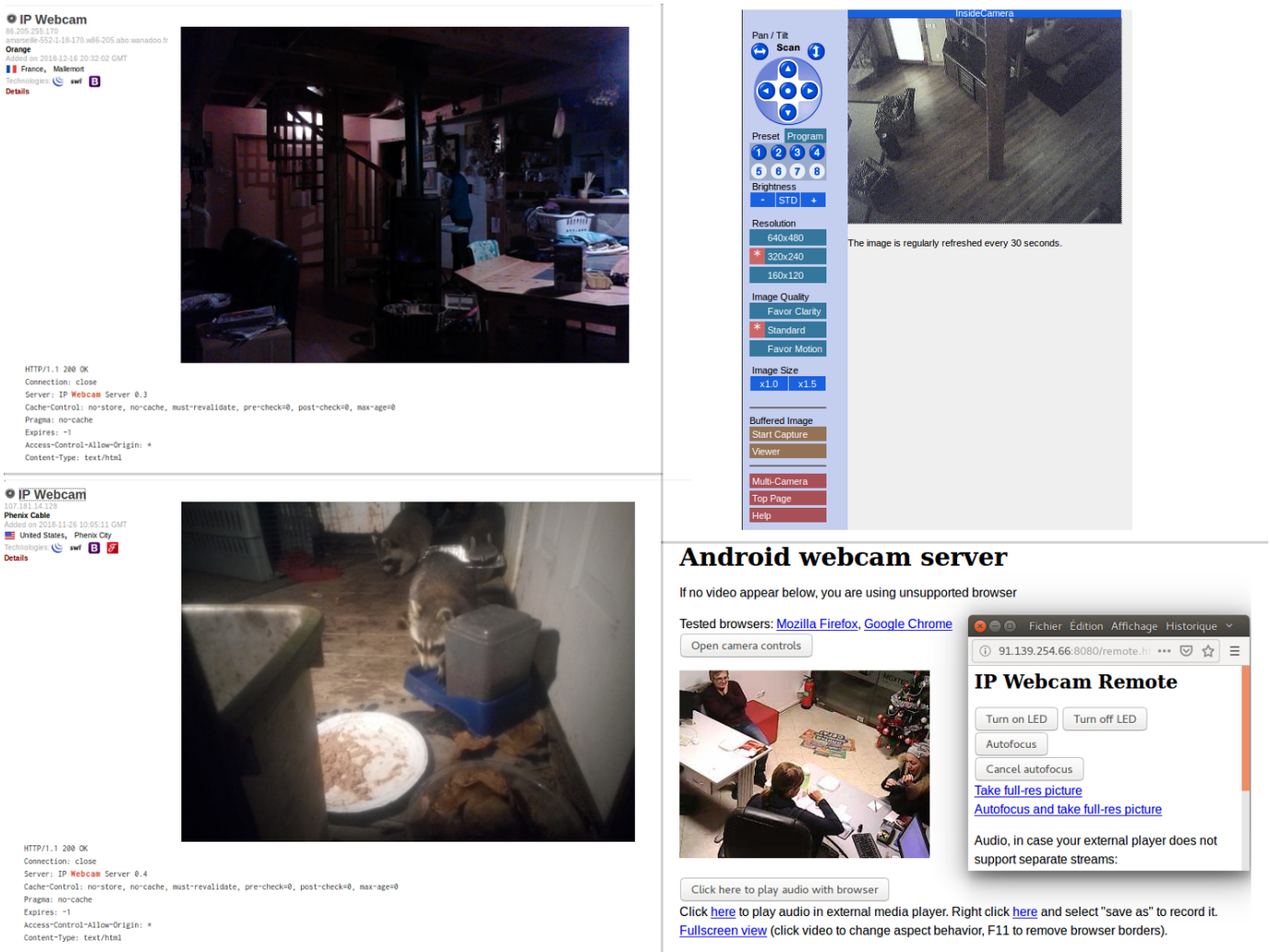

FIGURE 7 – Résultats d'une brève recherche avec Shodan ciblant des caméras en réseau non ou peu protégées

Évidemment, l'utilisateur « final »n'est pas le seul qui puisse en subir les conséquences. Ses équipements peuvent entre autres rejoindre un ensemble d'autres objets connectés infectés, reconfigurés malicieusement en vue d'effectuer une attaque de masse de type *DDOS* sur une cible étrangère au réseau domestique. C'est ce qui s'est passé avec *Mirai*, un malware décelé bien après son expansion qui a été utilisé dans plusieurs attaques menées sur le fournisseur de services DNS *Dyn*, la société d'hébergement *OVH* et bien d'autres. Mirai n'est pas un cas isolé, et les hackers ont pleinement saisi l'opportunité qu'incarne l'IoT : il existe beaucoup de failles connues et exploitables sur des équipements hétérogènes, et celles-ci sont en partie rendues publiques sur par exemple des dépôts githubs et des forums dédiés.

Les équipements qu'on retrouve dans une smarthome ne sont pas les seuls à présenter des lacunes en terme de sécurité. Encore plus grave, les systèmes connectés utilisés dans l'industrie de nos jours peuvent être attaqués, et cela peut porter lourdement à conséquence jusqu'à l'échelle nationale. Le scénario Horus le démontre particulièrement bien. Imaginé par un chercher néerlandais, il cible les installations photovoltaïques qui fournissent en énergie l'Europe, et plus spécifiquement les onduleurs connectés qui sont les dispositifs transformant le courant alternatif en continu. Pas moins de 17 failles ont été découvertes, dont certaines permettant de prendre le contrôle total des onduleurs à distance. Il serait alors possible de simuler artificiellement l'impact qu'une éclipse solaire aurait sur le réseau, qui est en temps normal compensé artificiellement. En appliquant une telle variation de puissance soudaine, le réseau entrerait dans un état de fluctuation tel qu'il ne serait plus contrôlable et une grande partie du réseau finirait par sauter. La perte suite à une telle panne, considérant qu'elle dure trois heures, s'élèverait à 4,5 milliards d'euros.

3.2 Description des besoins en terme de sécurité

Ces notions sont similaires à celles généralement mises en avant dans le domaine de la sécurité informatique. On peut cependant les affiner davantage en se restreignant au cadre un peu plus limité des équipements IoT et en tenant compte de leurs particularités. Ces grands axes sont les suivants [15] [17] :

Confidentialité

L'accès aux informations doit être limité aux personnes et systèmes qui y sont autorisés, et uniquement eux : aucun accès indésirable ne doit être possible. Dans un réseau constitué de plusieurs équipements, les transmissions entre chacun des nœuds doivent être protégées des intrus pouvant potentiellement les intercepter.

Intégrité

Les informations stockées et échangées ne doivent pas être altérées, qu'elle qu'en soit la raison (de source malveillante ou non). En plus des informations (mesures effectuées), le firmware exploitant les équipements doit également respecter cette propriété.

Disponibilité et continuité

Les systèmes doivent être fonctionnels quand cela est attendu et garantir l'accès et la bonne exécution des services qu'ils sont destinés à offrir, dans le délai attendu. La sécurité doit pouvoir être assurée en cas de panne ou dysfonctionnement de certaines parties du systèmes, et ce même dans le cas de systèmes dont les ressources sont limitées (équipements sur batterie en fin de vie par exemple).

Authentification

Seuls les utilisateurs légitimes (acquéreurs du produit entre autres) devraient être en mesure d'accéder et de configurer les systèmes, ainsi qu'obtenir aux informations sensibles qu'ils possèdent. L'authentification peut aussi être relative à une entité, comme le fabriquant certifié du produit qui tente d'accéder au système pour effectuer sa mise-à-jour ou lui appliquer un patch afin de corriger une faille dans sa sécurité.

Autorisation d'accès aux ressources

Les différents composants du système et les applications qui y sont relatives doivent avoir des privilèges d'accès au ressources limités de sorte qu'ils ne permettent que le strict minimum pour qu'ils puissent fonctionner correctement.

3.3 Modèle d'architecture des équipements

3.4 Vulnérabilités liées à l'IoT et menaces courantes

3.5 Possibilités d'amélioration

3.6 Solutions existantes

3.6.1 OWASP Sec framework

4 Les pare-feux et l'*IoT*

4.1 Différents types d'architecture

4.2 Les pare-feux domestiques

4.2.1 Caractéristiques d'un réseau domestique

4.2.2 Attaques possibles et conséquences

4.3 Application et implémentation

5 Mise en pratique : ébauche

Conclusion

Références

[1] Application firewall. Wikipedia article, URL : https://en.wikipedia.org/wiki/Application_firewall.

[2] Cable broadband technology gigabit evolution. Dernière lecture le 27/11/2018.

[3] Cisco asa. Wikipedia article, URL : https://en.wikipedia.org/wiki/Cisco_ASA, dernière lecture le 19/12/2018.

[4] Firewall(computing). Wikipedia article, URL : https://en.wikipedia.org/wiki/Firewall_(computing).

[5] Internet growth statistics par internet world stats. http://www.ti.com/lit/ml/swrb028/swrb028.pdf. Dernière lecture le 26/11/2018.

[6] Les firewalls. Community article, URL : https://www.frameip.com/firewall/.

[7] Pare-feu(informatique). Wikipedia article, URL : https://fr.wikipedia.org/wiki/Pare-feu_(informatique).

[8] Security in the Internet of Things. https://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr_security-in-the-internet-of-things.pdf, 2015.

[9] Alhafidh Basman M.Hasan and William Allen. Design and simulation of a smart home managed by an intelligent self-adaptive system. *Int. Journal of Engineering Research and Application*, 2016.

[10] Dan-Radu Berte. Defining the IoT. *De Gruyter*, 2018.

[11] Cisco. *CLI Book 2 : Cisco ASA Series Firewall CLI Configuration Guide, 9.6*, May 2018. URLs : https://www.cisco.com/c/en/us/td/docs/security/asa/asa96/configuration/firewall/asa-96-firewall-config/access-idfw.html, https://www.cisco.com/c/en/us/td/docs/security/asa/asa96/configuration/firewall/asa-96-firewall-config/access-idfw.pdf.

[12] Robert J. Shimonski, Debra Littlejohn Shinder, and Thomas W. Shinder, editors. *The Best Damn Firewall Book Period*. Syngress, Burlington, 2003.

[13] Kalle Kokko. Next-generation firewall case study. Master's thesis, South-Eastern Finland University of Applied Sciences, 2017.

[14] Sylvain Kubler, Kary Främling, and Andrea Buda. A standardized approach to deal with firewall and mobility policies in the IoT. *Pervasive Mobile Computing*, 2014.

[15] Engin Leloglu. A review of security concerns in Internet of Things. *Journal of Computer and Communications*, 2017.

[16] Manuel Leos Rivas. Securing the home IoT network. SANS Institute Reading Room, URL : https://www.sans.org/reading-room/whitepapers/hsoffice/paper/37717, 2014.

[17] Huichen Lin and Neil Bergmann. IoT privacy and security challenges for Smart Home environments. *Information (online MDPI journal)*, 2016.

[18] Dariusz Pałka and Marek Zachara. Learning web application firewall - benefits and caveats. pages 295–308, 01 2011.

[19] Alaauddin Shieha. *Application Layer Firewall Using OpenFlow*. PhD thesis, University of Colorado, 2014.