CHAPITRE 5

CRITIQUES ET INTERPRÉTATION

Dans ce dernier chapitre, nous allons interpréter et critiquer nos résultats. Nous exhiberons les avantages et les inconvénients de nos travaux. Enfin, à la dernière section, nous réfléchirons à de possibles nouvelles solutions afin de lutter plus efficacement contre les réseaux de zombies.

5.1 INTERPRÉTATIONS DES GRAPHES D'ÉVOLUTIONS

En analysant le graphe phylogénique, on observe une tendance à l'augmentation du nombre de fonctionnalités des réseaux de zombies. En effet, plus le rayon d'un nœud est grand, plus son nombre de fonctionnalités est élevé. Ces programmes malveillants deviennent en général de plus en plus complexes. Nous supposons que cette augmentation de la complexité provient de la réutilisation de fonctionnalités (pas nécessairement une réutilisation de code). En effet, lorsqu'un réseau fonctionne bien, des attaquants peuvent reprendre le concept et les principes de fonctionnement. Ils rajoutent alors de nouvelles fonctionnalités afin de l'améliorer. Ce type de graphe nous permet aussi de démontrer une certaine proximité des programmes. En effet, nous relevons que Chuck Norris et Psyb0t ont un nombre de fonctionnalités très proches et que la totalité des fonctionnalités de Psyb0t a été transmise à Chuck Norris. Cette proximité a été démontrée par des analyses de code Čeleda *et al.* (2010).

Nous observons aussi que des programmes plus complexes tels que Mirai ont énormément de points communs avec les logiciels plus anciens. Ce phénomène est dû à la réutilisation systématique de certaines fonctionnalités vitales pour un programme malveillant. Cela s'explique très bien avec les multi-graphes de propagation. Par exemple, à la figure 3.3, nous observons que la fonctionnalité « SYN Flood » introduite par Hydra a été réutilisée par la totalité des réseaux de zombies ayant pour objectif la création d'attaques de déni de service distribuées. Il en va de même pour la fonctionnalité « d'UDP flood » mise en place par Psyb0t. Dans cette famille de réseaux de zombies, nous pouvons observer que chaque fonctionnalité permettant la mise en place d'une attaque par déni de service a été propagée à tous les réseaux futurs.

La seule fonctionnalité n'ayant pas été propagée est l'attaque par « ICMP flood ». Cette attaque est très peu efficace et peut facilement être arrêtée par les fournisseurs d'accès CloudFlare (2019). On peut aussi observer sur la figure 3.4 le remplacement de certaines fonctionnalités comme la génération d'adresses IP par liste manuelle qui s'est transformée en liste automatique puis en génération aléatoire. Ainsi, nous observons l'apparition et l'abandon de fonctionnalités. Nous pouvons supposer qu'une fonctionnalité va se propager aux générations futures si elle est vraiment utile pour atteindre un but similaire et si la difficulté d'implémentation n'est pas trop élevée par rapport à son gain.

Par exemple, nous avons vu qu'Hydra avait transmis la fonctionnalité d'attaque « SYN Flood » à neuf des seize programmes malveillants de notre échantillon. La fonctionnalité n'a été transmise qu'à des programmes ayant pour objectifs de mettre en place des attaques de déni de service. Nous voyons qu'elle n'a pas été transmise à Darlloz qui avait pour objectif de faire miner ses victimes.

Une fonctionnalité peut disparaître si une autre plus efficace peut la remplacer. Nous pouvons observer que l'attaque par dictionnaire a progressivement été substituée par l'utilisation d'un dictionnaire pondéré et par l'exploitation de CVE. Ces deux techniques sont beaucoup plus

efficaces que la première. Nous observons le même phénomène pour les méthodes de génération d'adresses IP à scanner.

Ainsi, nous pouvons supposer que la transmission de fonctionnalité au sein des programmes malveillants va suivre une sorte de sélection naturelle basée sur l'objectif de ces programmes, l'efficacité des fonctionnalités ainsi que la difficulté d'implémentation. Les sources de certains de ces programmes étant disponibles en libre accès sur Internet, il devient très facile pour certaines fonctionnalités de se propager au travers des générations futures.

Le dernier point intéressant de cette représentation en multi-graphe de propagation est la possibilité de voir rapidement les programmes ayant introduit le plus de fonctionnalités et quels sont celles qui se sont le plus propagées.

Sur le premier graphe on peut observer que VPNFilter a introduit de nombreuses fonctionnalités d'attaques, comme par exemple la mise en place de VPN inversé, la possibilité d'espionner les réseaux locaux des appareils infectés ou encore la surveillance de système SCADA, toutes très dangereuses. Ce programme est considéré comme une menace persistante avancée (APT) mise en place par la Russie afin de nuire à l'Ukraine Cimpanu (2018b,a).

La plupart des fonctionnalités de ce ver ne se retrouveront pas dans la majorité des programmes malveillants subséquent. Il est extrêmement difficile d'implémenter ces fonctionnalités, ainsi, nous ne les verrons que dans de futures grosses attaques mises en place par d'autres gouvernements.

Sur la figure 3.4 , une simple observation montre que les fonctionnalités d'attaques contre les architectures MIPS mises en place par Hydra ont été très largement propagées. Il en est de même pour les fonctionnalités permettant de cibler les architectures ARM et x86/64 mises en place par Aidra et Carna. On peut aussi constater la proximité de certains programmes malveillants alors que leurs objectifs peuvent être opposés. Par exemple, Wifacth et Hajimé sont les deux

seuls réseaux de zombies ayant une architecture décentralisée en P2P. D'un côté, Wifatch a été créé par une équipe de « White Hat ». Lorsque Wifatch infecte un hôte, il supprime les autres programmes malveillants qu'il identifie, ferme certains ports et va afficher un message dans les logs demandant à l'utilisateur de changer son mot de passe. En conséquence, il va aider à protéger les utilisateurs. À l'opposé, Hajime va infecter ses hôtes en rajoutant des portes dérobées pouvant permettre au maître du réseau d'installer rapidement d'autres programmes malveillants. Les deux programmes utilisent des fonctionnalités proches, mais pour des objectifs opposés.

5.2 CRITIQUES ET PISTES D'AMÉLIORATION POUR LES REPRÉSENTATIONS

Notre taxonomie permet de décrire correctement les programmes étudiés et peut facilement être étendue et améliorée. Comme nous le mentionnons dans le chapitre précédent, certaines fonctionnalités existent et ont été observées dans certains réseaux de zombies (IoT ou non) mais nous ne les avons pas insérés, car aucun des programmes que nous avions étudiés ne les utilisait. Cependant, la taxonomie est faite de sorte qu'il soit possible de rajouter des niveaux et des taxons sans avoir à modifier l'organisation générale de la taxonomie.

Ainsi, une piste d'amélioration pour cette étude est d'analyser plus de programmes malveillants créant des réseaux de zombies d'objets connectés. Ainsi, nous pourrions intégrer plus de fonctionnalités dans notre taxonomie et donc avoir une idée plus précise de l'évolution de ces programmes malveillants. En effet, nous n'avons étudié que 16 familles de réseaux de zombies d'objets connectés. Or en utilisant une méthode d'analyse systématique pour les résultats académiques et techniques, nous pourrions augmenter sensiblement la quantité des informations pour étudier ces programmes.

Enfin, cette étude a été menée au début de l'année 2019. Ainsi, nous pouvons l'étendre avec

l'ensemble des vers apparus en 2019. De plus, les informations sur certains vers peu influents n'étaient que peu référencées à cette époque et en effectuant les mêmes recherches sur Google, nous avons pu découvrir d'autres programmes malveillants pouvant être ajoutés à notre étude.

L'avantage de notre représentation est de pouvoir faire un graphe pour chaque famille de fonctionnalités et ainsi avoir des graphes plus lisibles. Le seul problème est que plus le nombre de programmes étudiés augmente et plus les graphes deviennent grands et donc moins lisibles. Ainsi, il pourra être utile de compléter ces représentations avec diverses statistiques comme par exemple le temps de vie de chaque fonctionnalité ou le nombre de vers qu'ils implémentent.

5.3 INTERPRÉTATIONS DES SIMULATIONS DE PROPAGATIONS D'INFECTIONS

Notre modèle de propagation des infections permet d'observer les influences de chaque fonctionnalités sur l'efficacité des réseaux de zombies. Grâce aux simulations faites avec ce modèles, nous avons pu constater plusieurs phénomènes intéressants.

Le premier est la divisions de la taille des réseaux de zombies de manière presque proportionnelles aux nombre de réseaux de zombies de même nature. Ce genre de phénomène a pu être observé après que le code source de Mirai ait été divulgué. En effet, plusieurs groupes ont récupéré le code et l'on adapté pour créer leur propre réseaux de zombies.

Depuis ce temps, on peut observer une saturation des réseaux de zombies d'objets connectés et de plus en plus cherchent à exploiter des vulnérabilités nouvelles afin de pouvoir voler des zombies aux autres réseaux. La division n'est cependant pas parfaitement proportionnelle du au caractère aléatoire de la distributions des victimes et de la méthode de scan. On observe d'ailleurs une variation importante entre les diverses simulations, expliquant aussi les différences entre modèles et observations.

Le deuxième phénomène que nous avons pu observer est la grande différence d'efficacité entre une méthode de scan séquentiel et une méthode de scan aléatoire. La première a une croissance linéaire, tandis que la seconde présente à partir d'un moment, une croissance exponentielle. Cette différence majeure s'explique par le fait qu'avec une méthode de scan séquentielle, seul le premier zombie fait avancer l'exploration de l'espace IP. En effet, tous les nouveaux zombies recommencent à scanner depuis le début. Cette méthode de scan est très basique, intuitive et simple à mettre en place. Cependant, on voit qu'elle est très mauvaise.

Le scan aléatoire possède une croissance exponentielle car l'ensemble des nouveaux zombies se mettent eux aussi à scanner de manière aléatoire. On peut ainsi approximer cette méthode de propagation par une simple expérience, où à chaque tour, on piocherait une boule de couleur dans un sac contenant des boules blanches et des boules noires. Si la boule piochée est blanche, alors on la conserve or du sac et au tour suivant on pioche autant de fois qu'on a de boules blanches. Ici, l'espérance du nombre de boules blanches au tour t équivaut à l'espérance du nombre de zombies que possède notre réseau au tour t. On peut aisément la calculer avec une suite récursive, prenant en compte le nombre de victimes ainsi que la taille totale de la population. Cette équation est la suivante : $p_{t+1} = p_t.(\frac{V-p_t}{T} + 1)$, avec V le nombre total d'objet vulnérables, T le nombre total d'appareils et p_t l'espérance de la population au temps t.

Enfin, nous avons pu observer les effets du mécanisme de prévention des réseaux de zombies. Cet effet est très utilisé depuis l'apparition de Mirai afin de garantir un monopole des ressources des appareils exploités. Nous avons pu observer que, lorsqu'un réseau de zombie A peut exploiter une faille présente dans une population d'objets (ayant déjà été infectés ou non), que les autres réseaux de zombies ne peuvent pas exploiter, alors le réseau A va surpasser tous les autres réseaux. Il va pouvoir infecter tous les objets de la population vulnérable, et les protéger contres les autres réseaux de zombie. Ces derniers vont donc progressivement perdre de la puissance jusqu'à un minimum. Ceci est corroboré par d'autres observations faites au cours de nos travaux.

En effet, nous avons observé une augmentation graduelle du niveau de complexité des programmes malveillants dirigés contre les objets connectés. Cette augmentation de la complexité s'accompagne d'une élévation de leur dangerosité et une meilleure adaptabilité de ces programmes. De plus, nous pouvons observer que certaines caractéristiques telles que la détection d'environnement virtualisé proviennent des logiciels malveillants attaquant traditionnellement les ordinateurs classiques et les téléphones intelligents. On peut donc supposer que les programmes malveillants vont continuer à évoluer et que dans quelques années nous pourrons observer des programmes ciblant plusieurs plateformes. Cela commence déjà à se voir avec VPNFilter qui était capable d'attaquer et d'exploiter d'autres objets du réseau local de sa victime.

Notre analyse montre aussi que les attaques par dictionnaire contre les protocoles SSH et Telnet sont les plus nombreuses et font partie des stratégies les plus efficaces. Ce genre d'attaque a commencé en 2008 et est toujours utilisé. Cette stratégie a notamment permis à Mirai de mettre en place un réseau de zombies suffisamment grand pour créer la plus grande attaque de déni de service jamais enregistrée.

Une méthode simple pour protéger les objets connectés de ce genre d'attaque est de fermer les ports non utilisés et d'utiliser un mot de passe aléatoire. Cette méthode est tellement efficace que les réseaux de zombies ont commencé à l'utiliser pour sécuriser leurs hôtes afin que d'autres programmes ne puissent pas les corrompre.

Cependant, nous avons aussi constaté l'augmentation du nombre d'utilisations de CVE pour exploiter des objets connectés. Ces dernières, bien que plus complexes à implémenter, permettent une expansion plus rapide du réseau de zombies et sont bien plus efficaces qu'une utilisation de dictionnaire. Ce type d'exploit ne peut pas être stoppé par l'utilisation d'un mot de passe fort.

Cette observation permet de prédire que les divers réseaux de zombies vont exploiter de plus en plus de failles et délivrer de plus en plus de patchs de ces dernières, afin de garder leur monopole.

De plus, de manière générale, chaque exploit rajouté permet d'augmenter le nombre de victimes potentielles. Enfin, on peut prédire que dans quelques années, on observera une fusion entre réseaux de zombies traditionnel (ciblant PC et serveurs) et d'objets connectés.

En effet, lorsque la course aux exploits commencera, les réseaux devront être très modulaires afin de pouvoir constamment rajouter des modules d'exploitation de failles et les patch correspondants. Or pour créer des réseaux de zombies classiques, il existe deux grandes méthodes : diffuser des pourriels ou exploiter des failles non patchées. Ainsi, il serait idiot de ne pas rajouter dans la liste des vulnérabilités exploitées quelques exploits pour rajouter des PC et serveurs. Enfin, la tailles des réseaux de zombies d'objets connectés permet de diffuser plus largement et facilement des pourriels, pouvant ainsi aider à déployer des réseaux de zombies plus classiques.

5.4 CRITIQUES

Comme nous l'avons expliqué, notre modèle permet de rendre compte de manière précise les effets de chaque fonctionnalités jouant un rôle dans la phase de recrutement d'un réseau de zombie. Cependant, il reste encore beaucoup de comportements à implémenter. Par exemple, nous implémenterons la stratégie d'exploit avec un dictionnaire pondéré, permettant de passer moins de temps à essayer d'exploiter un appareil. Enfin, lorsque nous aurons implémenté et tester la majeur partie des fonctionnalités existantes, nous implémenterons de nouvelles stratégies comme par exemple la version distribuée de ZMAP Adrian *et al.* (2014). Le but sera de trouver des stratégies toujours plus efficaces. Il nous faudra ensuite tous les tester et déterminer la combinaison de fonctionnalité et de stratégies permettant de dominer les autres.

De plus, il nous faut maintenant réussir à déterminer le temps que prennent chacune de ces fonctionnalités pour des réseaux comme Mirai et confronter notre modèle avec la réalité. En effet, pour l'instant nous n'avons fait que des expériences théoriques, permettant de mieux comprendre

l'impact de certaines stratégies. Maintenant, il nous faut être capable de correctement paramétrer ces dernières afin de pouvoir aider à la prédiction de la propagation de réseaux de zombies.

Enfin, le dernier point négatif de notre modèle est son temps d'exécution. En effet, le modèle de jeu en tour par tour est très gourmand en temps processeurs et nécessite donc beaucoup de temps si nous souhaitons modéliser de grandes populations sur des temps longs.

5.5 DE NOUVELLES SOLUTIONS ?

Le but ultime de nos travaux est d'aider à la création de nouvelles solutions afin de diminuer les impacts néfastes des réseaux de zombies. Ainsi, nous allons détailler ici quelques idées et pistes de réflexions pouvant servir de base pour de futurs travaux.

5.5.1 PRÉVOIR LES FUTURS AVANCÉES

Afin de pouvoir combattre efficacement les impacts néfastes des réseaux de zombies, nous pensons qu'il est important de bien décrire ces réseaux et leurs évolutions. Le but est ensuite de pouvoir prévoir certaines évolutions et commencer à préparer des contre-mesures avant que ces dernières n'apparaissent. Dans la section précédente, nous avons prédit que, dans le futur, les réseaux de zombies exploiteraient de plus en plus les failles CVE pour toucher de plus en plus d'objets différents et pour augmenter leur vitesse d'exploitation. Mais, il est complexe de prévoir l'apparition de tels failles et de trouver les patchs correspondant sans trouver les failles.

A terme, il pourrait être intéressant d'observer le nombre de vulnérabilités qui seront corrigé par les réseaux de zombies et celles corrigés par les utilisateurs et les constructeurs. En effet, cela fait plus de dix ans que les mêmes failles sont toujours exploitées. On observe aussi que se sont les réseaux de zombies qui ont commencé à les corriger.

Ainsi, notre modèle n'aidera que peu à traiter cette partie. Tout au plus il pourra aider à comprendre l'utilisation de certaines failles plus que d'autres. Cela ne sera possible que pour les failles dont on connais les proportions d'objets vulnérables. Cela ne fonctionnera pas pour prédire l'apparition de vulnérabilité 0-day ou leur nature. Notre modèle encourage les chercheurs à les trouver avant les entités malveillante, mais n'aide pas à les découvrir.

Cependant, notre modèle peut aider à prévoir les évolutions d'autres techniques en rapport avec le recrutement. En effet, grâce à notre modèle, nous pouvons imaginer diverses stratégies d'exploration de l'espace IP et tester leurs efficacités. Nous pouvons aussi imaginer diverses méthodes d'infections ou d'organisation des réseaux. Par exemple, on peut imaginer plusieurs version d'un scan utilisant ZMAP Durumeric *et al.* (2014). On pourrait avoir une stratégie où chaque zombie est indépendant, une stratégie ou divise l'espace IP en un certain nombre de cycles et où l'on attribue un cycle à chaque zombie ou encore une stratégie ou chaque zombie scan non pas une adresse IP aléatoire mais un groupe d'adresse. Dans les deux derniers cas, si les stratégies s'avèrent pertinentes, notre modèle pourrait aussi aider à déterminer leurs paramètres optimaux.

5.5.2 LES RÉSEAUX DE BIENVEILLANCE

Comme nous l'avons dit plutôt, le but ultime de nos recherches est de pouvoir aider à combattre les impacts néfastes des réseaux de zombies. Cependant, plutôt que de mettre en place des méthodes curatives avec des systèmes de détection d'intrusion, nous proposons d'essayer de mettre en place une méthode préventive.

En effet, nous pensons que déterminer les meilleurs stratégies de scan, d'infection et de réplication peuvent servir à mettre en place un système de mise à jour distribué et obligatoire. Plutôt que de simplement étudier et détecter l'ensemble des stratégies des réseaux de zombies, nous

proposons d'utiliser ces fonctionnalité pour combattre ces réseaux malveillants. Ce système aurait une architecture hybride, avec des serveurs centraux contrôlés par des organismes d'état. L'ensemble des objets mis à jour formeront un réseau pair à pair pour faciliter la propagation des mise à jour et la robustesse du système.

Comme nous l'avons mentionné plus tôt, Mirai est toujours le programme malveillant le plus utilisé pour mettre en place des réseaux de zombie. Or ce dernier est actif depuis 2016 et les correctifs pour s'en prémunir sont extrêmement simple. Ceci nous montre que les utilisateurs ne mettent pas à jour leurs objets connectés et que la corruption de leurs objets à un impact suffisamment faible pour qu'ils ne cherchent pas à changer d'objets ou à les mettre à jour.

Ainsi, au lieu que la puissance de calcul volée par les réseaux de zombies ne servent à alimenter des attaques de dénis de services, nous proposons de l'utiliser afin de combattre les programmes à l'origine de ces attaques. Notre système fonctionnera comme un réseau de zombie, mais ne distribuera que des patchs et n'aura aucune charge malveillante. Ce système devra chercher l'ensemble des objets vulnérables à une faille, puis les corriger. De plus, il pourra récolter certaines informations sur les objets qu'il mettra à jour, comme l'architecture, la marque, le modèle etc. Cela permettra de pouvoir rapidement le retrouver et le mettre à jour, si une nouvelle faille l'affectant est découverte subséquemment.

Ce système de mise à jour comprendrait deux grandes phases. La première correspond au déploiement initial. Ici,le système utilisera les meilleurs techniques de scan, d'infection et de duplication afin de pouvoir s'installer dans un maximum d'objets connectés. En s'installant, il corrigera l'ensemble des vulnérabilités connues pour un appareil et devra faire une mise à jour du firmware de l'objet afin d'être très difficilement désinstallable. Lors de cette phase, chaque objet nouvellement rajouté au système de mise à jour, se mettra lui aussi à chercher des objets vulnérables afin que la couverture de notre système soit maximale.

Lorsque l'ensemble des adresses IP auront été scannées plusieurs fois sans que de nouveaux objets ne rentre dans notre système, ce dernier se mettra en veille. Bien sûr, chaque objet aura un identifiant unique et ne recevra et n'installera que des mise à jour signée par notre système, à l'aide d'un mécanisme de signature électronique robuste.

Aussi, chaque objet transmettra des informations le concernant au système central. Cela permettra d'organiser la seconde phase de vie de notre système. Cette dernière phase correspond à la vie de notre système et sera répétée à chaque fois qu'une faille sera détecter et qu'il faudra propager son correctif. Ici, il ne sera plus question d'utiliser une méthode de scan pseudo aléatoire, mais une propagation organisée et informée. En effet, grâce aux données récoltées, nous pourrons aisément établir un ordre de propagation des correctifs, basées sur l'importance de chaque objets. Ce classement peut être similaire un algorithme de « Page-Rank » comme celui mis en place par les divers moteurs de recherches.

Une fois l'ordre de mise à jour généré, les serveurs centraux distribueront rapidement le correctif aux objets les plus important, ainsi qu'une liste des autres objets à patcher. Ainsi, la propagation serait extrêmement rapide et la puissance que pourrait obtenir un réseau de zombie utilisant cette faille, diminuerait de manière exponentielle. Enfin, si un serveur avec une connexion 10 Gbps peut scanner l'ensemble des adresses IP en moins de cinq minutes Adrian *et al.* (2014), alors une dizaine d'entre eux, ainsi qu'un réseaux pair à pair devrait être capable de propager un correctif en quelques minutes. Une telle vitesse de propagation permettrait de rendre impossible l'utilisation d'une faille associée à une patch pour créer un réseau de zombie.

Actuellement, il existe quelques projet ayant un objectif similaire, comme par exemple AntibIoTic De Donno *et al.* (2018a) ou le projet de Jerkins et al. Jerkins (2017). Le problème actuel de ce genre de projet, est qu'ils sont complètement illégaux dans la majeure partie des pays. De plus, ces derniers vont en général récupérer le code d'un programme malveillant et supprimer la partie malveillante pour ensuite aider à combattre ce même vers. Cela n'est pas forcément

la méthode la plus efficace et nous proposons l'idée d'un système basé sur les évolutions des programmes malveillants afin de créer le système le plus efficace possible. Cependant, ce projet n'est aujourd'hui qu'une idée et son efficacité supposée n'est encore que purement spéculative. Il nous faut encore perfectionner notre modèle pour ensuite pouvoir simuler la mise en place d'un tel système de mise à jour.

CONCLUSION

Dans ce mémoire, nous avons analyser la sécurité fournies par divers protocole de communication pour objets connectés. Nous avons remarqué, que sous certaines conditions particulière, ces dernières pouvaient fournir une protection suffisante. Nous avons aussi remarqué que l'ensemble des dommages provoqués par le manque de sécurités des objets connectés ne venaient pas de ces protocoles, mais plutôt de l'exploitation d'objets directement connectés au réseau Internet.

Ainsi, nous avons identifié comme problème majeur la formation et l'utilisation de réseaux de zombies d'objets connectes. Afin de trouver de nouvelles pistes pour combattre ces réseaux malveillants, nous avons développé deux outils d'analyse et de modélisation afin de mieux comprendre l'évolution de ces programmes.

Notre première contribution est une taxonomie qui permet de mieux décrire les réseaux de zombies composés d'objets connectés. Celle-ci, basée sur les fonctionnalités utilisées par les réseaux de zombies observés, permet de créer une nouvelle représentation des évolutions des programmes malveillants. En effet, la représentation sous forme de graphe de propagation permet de rapidement détecter les fonctionnalités intégrées dans divers réseaux de zombies. Cela permet aussi de rapidement identifier les fonctionnalités les moins efficaces, très peu propagées aux travers des réseaux étudiés.

Afin de mieux comprendre pourquoi certaines fonctionnalités ont tendance à disparaître et d'autres à rester, nous avons développé un modèle permettant de simuler de manière fine l'impact de diverses fonctionnalités sur l'efficacité des réseaux de zombies. Ce modèle simule, sous forme d'un jeu en tour par tour, le phénomène de propagation des réseaux de zombies.

Cela nous a permis de modéliser plusieurs stratégies et de montrer leurs effets sur la vitesse de propagation des infections. En effet, nous avons pu modéliser et comparer les stratégies de scan séquentiel et aléatoire. Nous avons aussi réussi à mettre en compétition ces deux stratégies. Nos résultats montre que le scan aléatoire est bien plus efficace que le scan séquentiel. Ceci nous permet de partiellement confirmer notre hypothèse voulant que les fonctionnalités les plus efficaces ont tendance à mieux ce propager.

Nos modèles sont encore améliorable et il reste encore de nombreuses expériences à faire afin de confirmer totalement nos hypothèses concernant l'évolution des réseaux de zombies. Cependant, nos modèles peuvent déjà être utilisé afin de simuler de nouveaux comportement, n'ayant pas été observé. Cela peut nous permettre d'anticiper les évolutions future des réseaux de zombies. De plus cela ouvre la voie vers de nouveaux système de corrections de failles. En effet, nous avons évoqué l'idée d'un système de mise à jour distribué et obligatoire afin de combattre les réseaux de zombie. Nos futur travaux concerneront l'amélioration de nos modèles ainsi que la modélisation d'un tel système de mise à jour. Enfin, nous évoquerons les aspects légaux et éthiques de ce projet.

Pour conclure, nous répétons, comme de nombreux chercheurs avant nous, que les configurations et les mots de passe par défauts, connus de tous, sont le problème majeur de la sécurité des objets connectés. Le second est le faible taux de mise à jour des appareils, qui ne sont pas maintenus par les constructeurs, ou difficile à mettre à jour par les utilisateurs. Cela entraîne l'apparition de réseaux de zombies, exploitant des failles découverte il y a plus de dix ans. Il est donc important de se concentrer sur ces problèmes, et d'adopter des stratégies cohérentes pour

lutter contre les effets indésirables de ce manque de sécurité. C'est dans cette optique que nous avons évoqué la piste d'un système de mise à jour obligatoire, géré par des agences d'état et contrôlé par les sociétés civiles.