République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Fin d'études

En vue de l'obtention du diplôme Master recherche en Informatique

Option : Réseaux & Sécurité

Thème

Théorie des jeux et analyse des graphes d'attaques dans le contexte de l'IoT

Présenté par :

Yousfi Mayliss & Bounehar Lynda

Devant le jury composé de :

Président :	Dr.N. Bouadem	M.C.B U. A/Mira Béjaia.
Examinateur :	Dr.F.Zidani	M.C.B U. A/Mira Béjaia
Promotrice :	Dr.L.HAMZA	M.C.A U. A/Mira Béjaia

Année Universitaire : 2022/2023

*** Remerciements ***

Nous remercions le bon ***Dieu****, tout puissant, pour nous avoir accordé la force, et nous avoir guidés sur le bon chemin ;*

Nous exprimerons notre sincère gratitude envers notre encadreur M^{me} ***HAMZA Lamia****, pour ses qualités humaines et professionnelles remarquables. Sa précieuse aide, , conseils avisés et remarques pertinentes ont grandement contribué à la réalisation de ce travail ;*

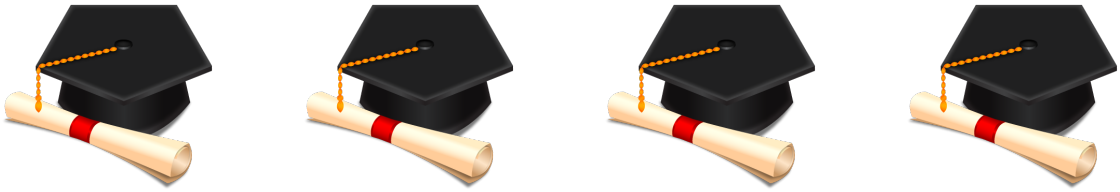
Nos vifs remerciements vont également aux membres du jury pour avoir accepté dévaluer notre travail et d'apporter les corrections nécessaires afin d'améliorer la qualité du manuscrit ;

Nous sommes extrêmement reconnaissants envers nos parents et nos familles qui nous ont toujours soutenus tout au long de notre parcours ;

Nous remerciements tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Nous sommes reconnaissants envers nos enseignants.

Nous remercions nos amis pour leur soutien ainsi que les membre d'option RS (RN).



Dédicace

Je dédie ce modeste travail :

À ceux qui sont les plus chers à mon cœur : mon père et ma mère. Leur soutien indéfectible, ils ont été les bougies illuminant mon chemin vers la réussite. Je leurs suis infiniment reconnaissante.

À ma sœur et mon frère, merci d'avoir toujours été là pour moi.

À mon encadrante Docteur Hamza Lamia qui nous a vraiment motivé et guidé tout au long de notre travail.

Mes pensées vont également à mes grands-parents, que Dieu ait leurs âmes. J'aurais aimé qu'ils soient présents en ce jour pour partager ma joie et ma réussite .

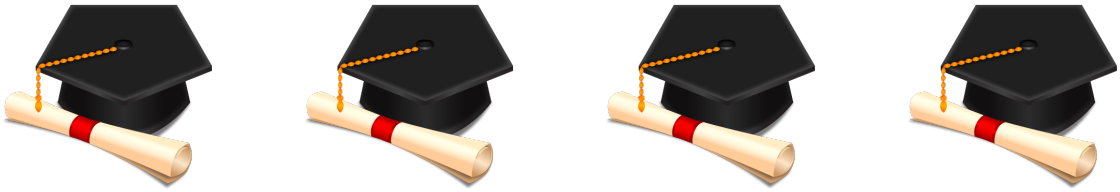
Toute ma famille sans exception oncles, tantes, cousins et cousines .

À mes chères copines : Mayssa, Karima, Dihia, Imene et Rahil, merci pour vos encouragements .

À mon aimable amie et binôme, Lynda et sa famille .

A tous ceux qui sèment le bonheur dans mon chemin.

Mayliss



Dédicace

Ce projet, je le dédie avec une profonde affection :

À ma chère maman, mon paradis, la prunelle de mes yeux, la source de ma joie et mon bonheur, tu es ma lune et le fil d'espoir qui illumine mon chemin. À toi, ma moitié.

À celui qui m'a fait une femme, ma source de vie, d'amour et d'affectation, à mon support qui était toujours à mes cotés pour me soutenir et m'encourager, à mon prince papa.

À mes chères sœurs Zahra et Mélissa, vous êtes mes complices, mes confidentes et mes épaules sur lesquelles je peux m'appuyer et mes partenaires dans les bons et les mauvais moments.

À ma chère tante Chafia, tu es un rayon de soleil dans ma vie. Ta bienveillance, ta sagesse et ta générosité m'ont toujours inspiré. Tu es une source de réconfort et d'encouragement, et je suis honoré d'avoir une tante aussi exceptionnelle que toi.

À Mes chers grands-pères, mes pensées se dirigent vers vous, même si vous n'êtes plus physiquement présents à mes côtés.

À ma copine Dihia, qui a toujours été là pour moi, prêts à tendre une main secourable et à partager mes joies et mes peines.

A mon aimable amie et binôme Mayliss pour les efforts contribué de sa part dans ce travail et son soutien tout au long de notre parcours universitaire.

Enfin, je souhaite dédier ce travail à ma famille entière maternelle et paternelle.

Lynda

Table des matières

Liste des tableaux

Table des figures

List of Algorithms

Liste des abréviations

AGA-POSG *Analyse Graphe Attaques - Partially Observable Stochastic Game*

BDD *Base des données*

COBANOT *Cost and Budget Aware Network Hardening for IoT*

CVSS *Common Vulnerability Scoring System*

DDS *Data Distribution Service*

FTP *File Transfer Protocol*

GPS *Global Positioning System*

HMM *Hidden Markov Model*

HP *Honey Pot*

HTTP *Hypertext Transfer Protocol*

IDS *Intrusion Detection System*

IP *Internet Protocol*

IPv4 *Internet Protocol version 4*

IPv6 *Internet Protocol version 6*

IoBT *Internet of Battlefield Things*

IoT *Internet of Things*

M2M *Machine-to-Machine*

MulVAL *Multihost, Multistage Vulnerability Analysis*

MDP *Markov Decision Process*

MQTT *Message Queuing Telemetry Transport*

POSG *Partially Observable Stochastic Game*

POMDP *Partially Observable Markov Decision Process*

RF *Radio Frequency*

RFID *Radio Frequency Identification*

SMS *Short Message Service*

SSH *Secure Shell*

SSHD *Secure Shell Daemon*

SQL *Structured Query Language*

TCP *Transmission Control Protocol*

WiFi *Wireless Fidelity*

WSN *Wireless Sensor Network*

Introduction générale

Dans le monde interconnecté d'aujourd'hui, un nouveau concept révolutionnaire est en plein essor, connu sous le nom d'Internet des objets (IoT).Cette technologie offre de nombreuses applications, fonctions et services facilitant la vie quotidienne dans une variété de domaines. L'IoT vise à imprégner notre environnement quotidien et ses objets, en reliant le monde physique au monde numérique et en permettant aux personnes et aux appareils d'être connectés à tout moment, n'importe où, avec n'importe quoi et n'importe qui [1].

L'Internet des objets (IoT) présente des risques importants en termes de sécurité. Avec l'augmentation de nombre des appareils connectés, les opportunités pour les cybercriminels d'exploiter les vulnérabilités et de compromettre les systèmes se multiplient. Les dispositifs IoT sont souvent conçus avec des mesures de sécurité insuffisantes, ce qui les expose à des attaques. Les failles de sécurité dans l'IoT peuvent avoir des conséquences graves, allant de l'accès non autorisé à des données personnelles et financières, jusqu'à la perturbation des infrastructures critiques. De plus, la collecte massive de données par les appareils IoT soulève des préoccupations en matière de vie privée et de protection des informations sensibles. Il est donc primordial de mettre en place des mesures de sécurité solides pour protéger ces réseaux.

Un graphe d'attaques est un formalisme général utilisé pour modéliser les failles de sécurité d'un système et toutes les séquences possibles d'exploits qu'un intrus peut utiliser pour atteindre un objectif spécifique [2]. La plupart des chercheurs se concentrent sur la génération de graphe d'attaques en raison de leur taille et de leur complexité, mais peu d'attention a été accordée à leur analyse. Il ne suffit pas de générer le graphe d'attaques d'un réseau, il est également nécessaire de déterminer les vulnérabilités suffisantes à supprimer ou à patcher afin de mettre fin aux attaques. Ainsi, une analyse complète accompagne la génération des graphes d'attaques.

Notre projet de fin de cycle se concentre sur la problématique de l'analyse des graphes d'attaques pour la protection des réseaux IoT. Nous proposons une nouvelle approche basée sur la théorie des jeux, qui facilite la gestion de la sécurité du réseau pour les administrateurs. Dans un premier temps, nous modélisons la situation sous forme d'un jeu, puis nous procédons à l'analyse du graphe d'attaques en appliquant notre méthode. Cette approche novatrice permettra aux administrateurs de mieux comprendre les schémas d'attaques potentiels et de prendre des décisions éclairées pour améliorer la sécurité de leur réseau IoT.

Afin de bien mener notre projet de fin de cycle, nous avons organisé notre mémoire en quatre

chapitres comme suit :

Dans le premier chapitre " Internet des objets ", nous présenterons les notions fondamentales de l'IOT à savoir sa définition, ses composantes, son architecture, la technologie dans l'IoT, ses caractéristiques, etc.

Dans le deuxième chapitre " Théorie des Jeux " , inclus quelques notions de base (définitions, théorèmes et propositions) sur la théorie des jeux que nous avons utilisé pour la réalisation de notre approche.

Le troisième chapitre " Travaux antérieurs sur l'analyse des graphes d'attaques " comporte notre problématique et quelques travaux déjà réalisés sur l'analyse des graphes d'attaques.

Enfin, le dernier chapitre " Attack Graph Analysis by Partially Observable Stochastic Game (AGA-POSG) " nous aborderons la démarche suivie pour la réalisation de notre approche par la définition de notre jeu stochastique partiellement observable (les joueurs, les actions, les observation, fonction de transition, fonction de récompense , etc). Ensuite, nous développerons les étapes de notre approche, et nous conclurons par une évaluation et une discussion.

Une conclusion et perspectives terminerons ce mémoire.

Chapitre 1

Internet des Objets (IdO)

1.1 Introduction

En 1999, Kevin Ashton responsable de marketing chez Procter et Gamble a utilisé pour la première fois le terme :Internet of Things (IoT) afin de décrire des objets équipés de puces RFID (Radio Frequency Identification) permettant de les identifié de manière unique. Cela l'a amené à travailler sur un système universel et ouvert pour connecter les objets à Internet [3]. L'Internet des objets consiste en un réseau mondial d'objets qui peuvent être connecter à Internet et qui sont en mesure de transmettre des informations et de recevoir des commandes. Cette technologie permet de connecter le monde physique et le monde virtuel, offrant ainsi une multitude de possibilités pour la création de nouveaux scénarios. Dans ce chapitre, nous allons étudier en détail le concept de l'Internet des Objets.

1.2 Définition d'Internet des objets

Il n'y a pas une définition standardisée et unifiée de l'Internet des Objets. Certaines définitions mettent l'accent sur les aspects techniques de l'IoT, tandis que d'autres se concentrent sur son utilisation et ses caractéristiques.

1.2.1 Définition 1

L'IoT est définit comme étant un réseau d'objet physiques munis de capteurs, d'actionneurs et d'interfaces de communication, qui leur permettent de communiquer, de collecter et d'échanger des informations dans le but de réaliser des tâches spécifiques en vue d'améliorer la qualité de vie [4].

1.2.2 Définition 2

L'IoT est un réseau qui permet de connecter des objets à Internet et de faciliter la communication et l'échange d'informations entre eux, en utilisant des protocoles adaptés à différents types de dispositifs [5].

1.3 Composante de l'IoT

L'IoT est un système qui résulte de l'intégration de multiples composants, ce qui peut engendrer une complexité atténuée par l'interopérabilité. La gestion des interfaces est donc cruciale pour assurer son bon fonctionnement. Dans ce qui suit nous allons présenter les principales composantes de l'IoT.

1.3.1 Capteur

Un capteur est un appareil qui mesure une propriété physique en détectant des informations particulières dans le monde physique. Il peut s'agir de lumière, d'humidité, de mouvement, de pression, de température ou de toute autre condition environnementale [6].

1.3.2 Actionneur

Un actionneur est un simple moteur qui peut être utilisé pour déplacer ou commander un mécanisme ou un système, sur la base d'un ensemble spécifique d'instructions [6].

1.3.3 Réseau de capteur

Les capteurs sont équipés de dispositifs sans fil pour communiquer entre eux, mais cela ne suffit pas à rendre le réseau de capteurs accessible de manière interopérable, transparente et simplifiée. Pour atteindre cet objectif, les capteurs doivent également être organisés en réseau. Ce qui caractérise un réseau de capteurs, c'est que ses éléments sont de petits appareils dotés de capacités de transmission sans fil [6].

1.3.4 Energie

La principale contrainte des capteurs concerne leurs alimentations en énergie. La durée de vie autonome des nœuds est mesurée en termes d'années [6].

1.3.5 Connectivité

Les objets connectés utilisent une antenne Radio Fréquence pour communiquer avec un ou plusieurs réseaux. Ils peuvent envoyer des informations telles que leur identité, leur état, des alertes ou des données de capteurs, ainsi que recevoir des commandes d'action et des données. Le module de connectivité gère également le cycle de vie de l'objet, notamment l'authentification, l'enregistrement, la mise en service, la mise à jour et la suppression de l'objet du réseau [6].

1.4 Architecture de l'IoT

Depuis les dernières années, le concept de l'IoT a été étudié, mais certains aspects de ce domaine ne sont pas encore clairement définis. Prenons exemple, il n'y a pas une architecture standardisée et spécifique pour l'IoT. Cependant, une architecture à trois couches est largement acceptable, même si elle n'est pas totalement compatible, cette dernière se compose de la couche de perception, la couche réseau et la couche application comme le présente la Figure 1.1.

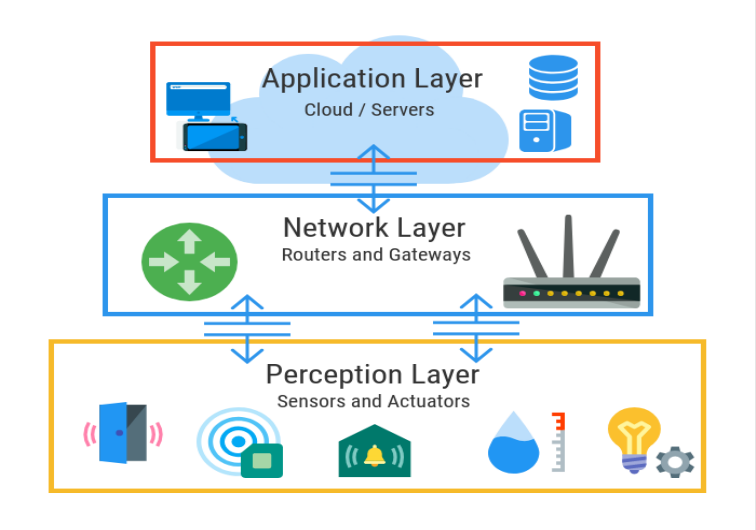


FIGURE 1.1 – Les trois couches de l'IoT [10].

1.4.1 La couche de perception

La tâche principale de la couche de perception est de donner une signification physique à chaque objet. Elle se compose de capteurs de données sous différentes formes, comme des étiquettes RFID ou d'autres réseaux de capteurs qui peuvent détecter la température, l'humidité, la vitesse et l'emplacement des objets, etc. Cette couche recueille les informations utiles sur les objets à partir des dispositifs de détection qui leur sont associés et les convertit en signaux numériques qui sont ensuite transmis à la couche réseau pour la suite des opérations [8].

1.4.2 La couche réseau

L'objectif de cette couche est de recevoir les informations utiles sous forme de signaux numériques de la couche perception et de les transmettre aux systèmes de traitement de la couche intergiciel par le biais de moyens de transmission tels que WiFi, Bluetooth, , ZigBee,3G, etc. avec des protocoles tels que IPv4, IPv6, MQTT (Message Queuing Telemetry Transport), DDS (Data Distribution Service), etc [8].

1.4.3 La couche d'application

La couche application est charger d'analyser les informations reçues de la couche réseau et de fournir des applications pour résoudre divers défis technologiques [8].

1.5 La technologie dans l'IoT

L'IoT permet d'interconnecter différents Objets intelligents via l'Internet. Le principal facteur du concept de l'IoT est l'intégration de différentes technologies à savoir le RFID, les technologies mobiles, TCP/IP, etc ; ces derniers permettent d'identifier des Objets, capter, stocker, traiter, et transférer des données dans les environnements physiques, mais aussi entre des contextes physiques et des univers virtuels. En effet, bien qu'il existe plusieurs technologies utilisées dans le fonctionnement de l'IoT, les technologies clés de l'IoT sont définies ci-dessous [5] :

1.5.1 RFID

RFID (Radio Frequency Identification) est une technologie sans fil utilisé pour identifier les Objets, elle englobe toutes les technologies qui utilisent des ondes radio pour identifier automatiquement des Objets ou des personnes. Elle permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui émet des ondes radio. Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des Objets, ou pour identifier ces Objets à distance. L'étiquette contient des informations stockées électroniquement pouvant être lues à distance.

1.5.2 WSN

WSN (Wireless Sensor Network)est un ensemble de nœuds qui communiquent sans fil organisés en un réseau coopératif. Chaque nœud possède une capacite de traitementet peut contenir différents types de mémoires, un émetteur-récepteur RF et une source d'alimentation. Il peut aussi tenir compte des divers capteurs et actionneurs. Comme son nom l'indique.

1.5.3 M2M

M2M (Machine-to-Machine) est l'association des technologies de l'information et de la communication avec des Objets intelligents dans le but de donner à ces derniers les moyens d'interagir sans intervention humaine avec le système d'information d'une organisation ou d'une entreprise.

1.6 Caractéristique de l'IoT

Les caractéristiques fondamentales de l'IoT sont les suivantes [9] :

1.6.1 Interconnectivité

En ce qui concerne l'IoT, tout peut être interconnecté avec l'infrastructure mondiale d'information et de communication.

1.6.2 Services liés aux objets

L'IoT est capable de fournir des services liés aux objets dans les limites des objets, tels que la protection de la vie privée et la cohérence sémantique entre les objets physiques et les objets virtuels qui leur sont associés. Afin de fournir des services liés aux objets dans le cadre des contraintes des objets, les technologies du monde physique et du monde de l'information vont toutes deux évoluer.

1.6.3 Hétérogénéité

Les appareils de l'IoT sont hétérogènes car ils sont basés sur des plateformes matérielles et des réseaux différents. Ils peuvent interagir avec d'autres appareils et d'autres réseaux.

1.6.4 Changements dynamiques

L'état des appareils change de manière dynamique (connectés et/ou déconnectés), ainsi que leur contexte, y compris leur emplacement et leur vitesse. En outre, le nombre de dispositifs peut changer de manière dynamique.

1.6.5 Énorme échelle

Le nombre d'appareils à gérer et qui communiquent entre eux sera d'au moins un ordre de grandeur supérieur à celui des appareils connectés à l'Internet actuel. La gestion des données générées et leur interprétation à des fins d'application seront encore plus critiques. Cela concerne la sémantique des données, ainsi qu'un traitement efficace des données.

1.6.6 Sécurité

Lorsque nous utilisons l'IoT à notre avantage, il est essentiel de ne pas négliger la sécurité. Cela inclut la sécurité de nos données personnelles et la sécurité de notre bien-être physique. Sécuriser les terminaux, les réseaux et les données qui circulent à travers tout cela signifie créer un paradigme de sécurité qui s'adaptera.

1.6.7 Connectivité

La connectivité permet l'accessibilité et la compatibilité du réseau. L'accessibilité consiste à se connecter à un réseau, tandis que la compatibilité fournit la capacité commune de consommer

et de produire des données.

1.7 Domaine d'application de l'IoT

IoT est une technologie innovante en plein essor, offrant une multitude d'application, de fonctionnalité et de services dans la vie quotidienne, ceci permettra l'émergence d'espaces intelligents, la Figure 1.2 illustre les domaines d'application dans le domaine IoT.

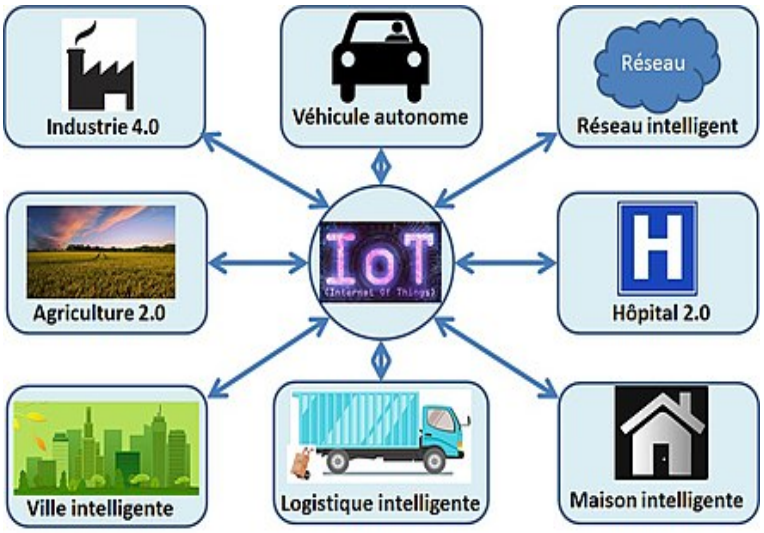


FIGURE 1.2 – Les domaines d'exploitation de IoT [11].

Nous allons énumérer brièvement quelques exemples d'application de l'IoT [12] [13] :

1.7.1 Les villes intelligentes

Les villes intelligentes peuvent bénéficier de l'IoT en fournissant de nouvelles capacités et fonctionnalités tout en réduisant considérablement l'intervention humaine. Cela peut inclure une meilleure gestion des réseaux de services publics (électricité, gaz, eau, etc.) grâce à une surveillance continue en temps réel et précise. Les solutions IoT peuvent également améliorer l'efficacité de l'utilisation des ressources et la qualité de vie des citoyens grâce à une combinaison intelligente des infrastructures à différents niveaux hiérarchiques (ville, quartier, bâtiment, etc.). Voici un résumé des cas d'utilisation les plus courants qui ont déjà été mis en place dans les villes intelligentes à travers le monde :

- **IoT dans la circulation routière :** les villes intelligentes utilisent l'IoT pour développer des solutions de circulation routière intelligentes. Ces solutions utilisent différents types de capteurs et intègrent les données GPS (Global Positioning System) des smartphones des conducteurs pour contrôler le nombre, l'emplacement et la vitesse des véhicules. Les

feux de signalisation intelligents connectés à une plateforme de gestion cloud permettent de surveiller les durées de feu vert et de changer automatiquement les feux en fonction de la situation actuelle du trafic pour éviter la congestion. Les données historiques sont utilisées pour prévoir les points de congestion potentiels et prendre des mesures pour les éviter.

- **Sécurité des citoyens :** les technologies de villes intelligentes basées sur l'IoT offrent des moyens de surveillance en temps réel, d'analyse et de prise de décision pour améliorer la sécurité publique. En intégrant les données des capteurs acoustiques et des caméras de vidéo surveillance déployées dans toute la ville avec les données provenant des réseaux sociaux, ces solutions peuvent prédire les scènes de crime potentielles. De cette manière, la police peut intervenir pour prévenir les infractions potentielles.

- **Stationnement intelligents :** les solutions de stationnement intelligent utilisent les données GPS des smartphones des conducteurs pour créer une carte de stationnement en temps réel qui indique les emplacements libres ou occupés. Lorsqu'une place de stationnement se libère à proximité, les conducteurs reçoivent une notification et peuvent utiliser la carte sur leur téléphone pour trouver plus rapidement et facilement une place de stationnement, évitant ainsi de conduire sans but à la recherche d'une place.

- **Les bâtiments à haute efficacité énergétique :** la technologie IoT facilite la tâche des bâtiments ayant une infrastructure ancienne pour économiser de l'énergie et améliorer leur durabilité. Les systèmes de gestion intelligente de l'énergie pour les bâtiments, tels que la climatisation et l'éclairage, utilisent des dispositifs IoT pour connecter des systèmes de chauffage et de sécurité incendie différents et non standardisés à une application centrale de gestion. Selon les recherches, les bâtiments commerciaux gaspillent jusqu'à 35% de l'énergie qu'ils consomment, donc les économies réalisées avec un système de gestion de l'énergie pour les bâtiments intelligents peuvent être considérables.

- **Gestion des déchets :** les solutions de villes intelligentes basées sur l'IoT aident à optimiser les horaires de collecte des déchets en suivant les niveaux de déchets, fournissant ainsi une optimisation de l'itinéraire et des analyses opérationnelles. Chaque conteneur à déchets est équipé d'un capteur qui recueille des données sur le niveau de remplissage. Lorsque le niveau de remplissage atteint un certain seuil, la solution de gestion des déchets envoie une notification à l'application mobile du conducteur de camion pour vider le conteneur plein, évitant ainsi de vider des conteneurs à moitié pleins.

1.7.2 La domotique

Une maison intelligente connectée est une résidence équipée de divers capteurs, de systèmes et de dispositifs qui permettent aux propriétaires de contrôler, surveiller à distance et automatiser

les différentes fonctions de leur maison. Cette technologie permet de créer un environnement plus confortable et sécurisé pour les habitants, tout en facilitant leur vie quotidienne tels que :

- **Services de sécurité :** les systèmes de sécurité sont souvent destinés à offrir des services conçus pour surveiller, détecter et contrôler les menaces de sécurité et de sûreté. Les systèmes de sécurité et de sûreté intelligents pour la maison vont généralement des services de surveillance d'entrée à distance aux systèmes qui reconnaissent automatiquement les menaces physiques, à savoir un incendie ou un cambriolage, et prennent automatiquement les mesures correspondantes. Ce domaine comprend des fonctionnalités qui prennent en charge les systèmes d'alarme, les caméras et les serrures de porte intelligentes.

- **Gestion de l'énergie :** les systèmes de gestion d'énergie pour les maisons ont pour objectifs d'optimiser la consommation d'énergie et d'en assurer une gestion efficace, tout en utilisant des technologies telles que les compteurs intelligents et les systèmes d'éclairage adaptatifs. Les architectures de système associées à ce domaine peuvent utiliser des systèmes multi-agents intelligents et des stratégies de contrôle pour prédire et maximiser automatiquement l'efficacité énergétique et le confort de l'utilisateur.

- **Divertissement :** les systèmes connectés de maisons intelligentes ont pour but de rendre le divertissement plus accessible et confortable pour les occupants en proposant un contenu de divertissement personnalisé et des services de communication sociale. Le secteur du divertissement est principalement représenté par des systèmes de haut-parleurs intelligents, des téléviseurs connectés et des consoles de jeux.

1.7.3 L'industrie

L'utilisation de l'IoT dans le secteur industriel offre de nombreux avantages pout résoudre les problèmes liée à ce dernier. L'objectif principal consiste à utiliser des capteurs et des dispositifs connectés pour collecter des données en temps réel sur les équipements, les processus de production et les opérations logistique. Cela permet aux entreprises d'améliorer l'efficacité de leurs opérations d'une manière approfondie, d'optimiser la production et la qualité des produits, d'assurer la traçabilité et renforcer la sécurité des employés.

1.7.4 Energie

Le système IoT fournit un moyen puissant pour gérer le coût de la consommation d'énergie et d'optimiser la production des entreprises. La gestion de l'énergie est devenue vitale pour les services publics et les installations, pour cette raison l'IoT est considérée comme essentielle pour répondre à la demande croissante en énergie, et elle permet de réduire les coûts tout en améliorant la gestion de l'énergie. Voici quelques-uns des principaux cas d'utilisation de IoT dans le domaine de l'energie :

- **Réparation et maintenance :** en utilisant des capteurs surveillés à distance en temps réel, il est possible de détecter toute anomalie dans l'efficacité énergétique, telle qu'une absence de succès ou une diminution inhabituelle. Ces résultats peuvent être consultés instantanément depuis n'importe quel appareil connecté à Internet, et les utilisateurs peuvent être alertés de manière inhabituelle par SMS ou e-mail en cas de variation anormale de l'efficacité énergétique.

- **Collecte facile des données de consommation :** Les compteurs connectés éliminent la collecte manuelle des relevés et le traitement des données de consommation d'eau, de gaz et d'électricité. Ils transmettent instantanément les données via le réseau Sigfox [36], sans configuration préalable. Les compteurs fonctionnent pendant des années sans intervention des utilisateurs ou remplacement de batterie. Cela permet de surveiller en temps réel la consommation, détecter les fuites et pannes, automatiser la facturation et la gestion, et contrôler les services à distance.

- **Surveillance des poteaux électriques :** est essentielle pour prévenir leur basculement causé par des événements tels que les vents violents, les accidents de la route ou les mouvements de terrain, qui peuvent causer une tension mécanique et une rupture de câble. Afin d'empêcher la chute totale du poteau, il est important d'identifier rapidement les problèmes potentiels et de prendre des mesures correctives. L'utilisation de dispositifs IoT pour surveiller les poteaux électriques à distance permet de mettre en place une maintenance préventive en effectuant des mesures régulières et en envoyant des alertes en cas de basculement important. Grâce à cette surveillance proactive, les équipes de maintenance peuvent agir rapidement pour empêcher la chute du poteau et déterminer les réparations nécessaires.

1.8 Sécurité dans l'IoT

L'IoT est un concept de réseau dans lequel des objets et des capteurs peuvent communiquer directement entre eux sans intervention humaine. Son objectif est de faciliter la vie des utilisateurs en connectant divers appareils tels que des ordinateurs, des voitures et des électroménagers. Cependant, le déploiement à grande échelle de l'IoT pose un défi majeur en matière de sécurité tels que la confidentialité, l'autorisation, la vérification, le contrôle d'accès et la gestion des informations. Les applications IoT ont simplifié la vie en étant sensibles, adaptatives et réactives aux besoins humains, mais cela peut compromettre la vie privée des utilisateurs et divulguer leurs informations personnelles. La sécurité est donc essentielle pour assurer la confiance des utilisateurs en termes de confidentialité et de contrôle des informations personnelles. Ainsi, la résolution de ces problèmes de sécurité est cruciale pour le développement de l'IoT [13].

La Figure 1.3 met en évidence les principaux défis à relever pour renforcer la sécurité dans l'environnement de l'IoT [14].

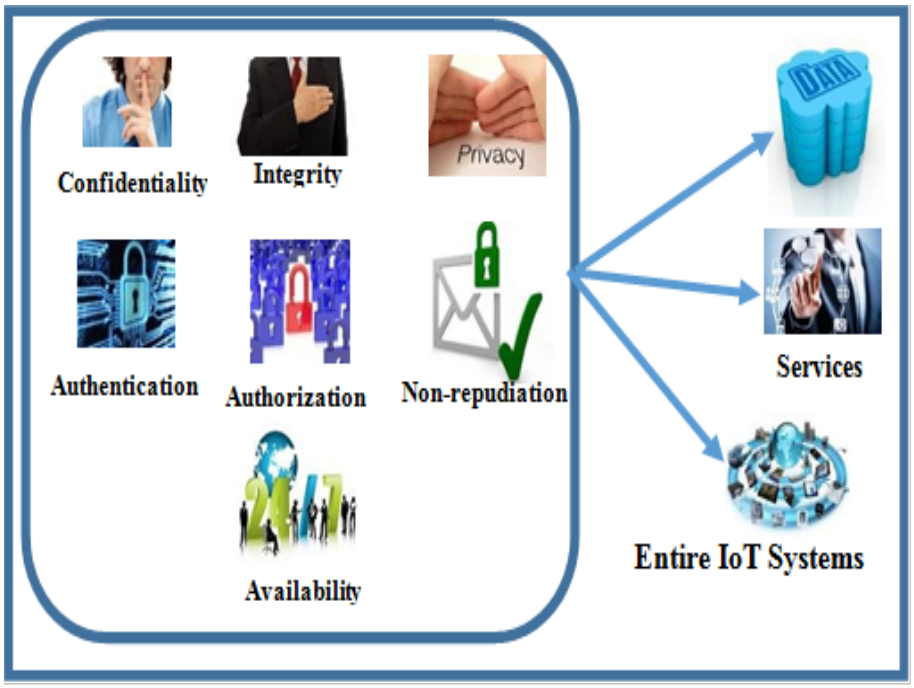


FIGURE 1.3 – Les exigences de la sécurité [14].

1.8.1 Confidentialité

La confidentialité est essentielle pour garantir aux entités autorisées l'accès et la modification des données. Dans l'environnement IoT, l'autorisation est donnée non seulement aux utilisateurs, mais aussi aux objets. Pour assurer la confidentialité, il est important de mettre en place un mécanisme de contrôle d'accès et un processus d'authentification d'objet. Par ailleurs, dans un scénario IoT, il faudra également décrire un langage de requête approprié pour permettre aux applications de récupérer les informations souhaitées à partir d'un flux de données.

1.8.2 Intégrité

Il s'agit de la véracité, de l'honnêteté et de la fiabilité des données. Avec le nombre croissant d'appareils et d'utilisateurs connectés à l'environnement IoT, l'intégrité devient un enjeu majeur en matière de sécurité. Un adversaire ne peut pas altérer les données d'une transaction sans que le système ne détecte les modifications. L'identité des appareils est complexe et il est très difficile de déterminer la source originale des données. . Il y a une confusion dans l'utilisation des dispositifs et des données de confiance. Les mesures de protection des données par des mots de passe sont insuffisantes dans les technologies IoT et des solutions de calcul de confiance doivent être mises en place pour garantir l'intégrité des données et des appareils.

1.8.3 Authentification

L'authentification consiste à vérifier l'identité de l'utilisateur ou de l'entité qui participe à la communication. Elle fonctionne conjointement avec l'intégrité, la confidentialité et l'autorisation. Bien que le nombre d'appareils connectés à Internet ne cesse d'augmenter, la scalabilité représente une menace importante pour l'authentification des appareils. Il est donc impératif de proposer une architecture ou un mécanisme capable de gérer de manière sécurisée la scalabilité des appareils dans l'environnement IoT. Pour ce faire, des infrastructures d'authentification adaptées aux scénarios IoT sont nécessaires.

1.8.4 Autorisation

L'autorisation consiste à permettre à tout utilisateur ou périphérique d'accéder aux informations de l'environnement IoT. La permission est délivrée avec l'identité de l'appareil ou de l'utilisateur. Avec une identité appropriée, n'importe qui peut accéder aux informations de l'environnement IoT, mais sans autorisation, personne ne peut accéder aux données ou services de cet environnement. Pour cela, il est primordial d'avoir un mécanisme d'autorisation efficace, et les chercheurs doivent également relever le défi de renforcer la vérification de l'identité.

1.8.5 Non-répudiation

C'est l'assurance qu'une personne ne peut pas nier quelque chose. Dans le contexte de l'IoT, un nœud ne peut pas nier l'envoi d'un message ou d'une information à un autre nœud ou à l'utilisateur. La propriété des données deviendra un problème majeur dans l'environnement IoT.

1.8.6 Disponibilité

La disponibilité est un aspect crucial de l'Internet des Objets, qui englobe la récupération et la fiabilité des données. En raison de la nature hautement distribuée de l'environnement IoT, une quantité explosive de données est disponible partout. Lorsque tous les appareils sont connectés à Internet, ils peuvent générer des données et les stocker n'importe où. Ainsi, n'importe qui peut être suivi ou tracé sans leur consentement ou leur connaissance. Pour garantir la disponibilité des données et des services dans cet environnement, il est essentiel de développer des algorithmes appropriés.

1.8.7 Vie privée

C'est le droit d'une entité (personne) de déterminer la quantité d'informations qu'elle est prête à partager avec d'autres personnes. En raison de la connectivité croissante de l'IoT, la protection de la vie privée est devenue un enjeu majeur. Beaucoup d'informations personnelles peuvent être collectées à l'insu de l'utilisateur, et le contrôle de leur diffusion est difficile à assurer dans le

contexte actuel. Les utilisateurs du système IoT doivent donc gérer leurs propres données. Les propriétaires doivent conscients de qui utilise leurs données et dans quel but. Les chercheurs devraient élaborer un cadre général pour la protection de la vie privée dans l'IoT ainsi que de techniques innovantes pour garantir la scalabilité dans un environnement IoT hétérogène.

1.9 Défis de l'IoT

L'IoT est un concept bénéfique et prometteur qui peut résoudre efficacement des problèmes de suivi et de surveillance dans divers domaines. Toutefois, son acceptabilité et sa maturité soulèvent des questions importantes. Parmi les défis les plus importants, on peut citer [9] [14] :

1.9.1 Fiabilité

La fiabilité est essentielle pour garantir le bon fonctionnement et la disponibilité des informations et des services dans les systèmes IoT. Elle est encore plus critique dans les applications de réponse d'urgence. Pour assurer une distribution d'informations fiable, la fiabilité doit être implémentée dans les logiciels et le matériel à travers toutes les couches IoT. La communication sous-jacente doit être fiable pour éviter les retards, les pertes de données et les décisions erronées.

1.9.2 Scalabilité

L'IoT implique de connecter un grand nombre d'appareils intelligents au réseau, ce qui peut causer des problèmes d'adressage, de gestion de l'information et de services. Ces problèmes deviennent de plus en plus difficiles à gérer à mesure que le nombre d'appareils connectés augmente. Pour répondre à ce défi, l'IoT doit être capable de gérer efficacement des environnements de petite et grande échelle, assurant ainsi une gestion optimale et sécurisée des dispositifs connectés. Auto-configuration : les dispositifs IoT doivent être conçus pour pouvoir se configurer automatiquement et ainsi s'adapter à leur environnement sans intervention manuelle de la part de l'utilisateur.

1.9.3 Hétérogénéité

Dans le contexte de l'IoT, les objets connectés ont des capacités de collecte, de traitement et de communication qui varient d'un appareil à l'autre, ce qui peut créer une hétérogénéité entre les appareils. Afin de permettre une communication et une coopération efficaces entre ces différents types d'objets connectés, il est essentiel qu'ils disposent d'une norme de communication commune. Cela permettra aux appareils de différents types de communiquer et de travailler ensemble de manière cohérente et coordonnée pour atteindre des objectifs communs.

1.9.4 Interprétation des données

Dans le domaine de l'IoT, il est essentiel de prendre en compte le contexte dans lequel les capteurs opèrent pour interpréter correctement les données collectées et tirer des conclusions pertinentes.

1.9.5 Sécurité et confidentialité des données personnelles

Dans l'IoT, le réseau d'objets intelligents connectés à internet présente un défi majeur en termes de sécurité et de confidentialité des données personnelles. Les utilisateurs peuvent avoir besoin de restreindre l'accès à certaines informations ou de bloquer des communications ou des transactions pour protéger leurs données confidentielles contre les concurrents. Cela implique la nécessité de mettre en place des mesures de protection adéquates pour prévenir les atteintes à la vie privée et les violations des données.

1.9.6 Tolérance aux pannes

Dans l'IoT, les objets intelligents ou les appareils sont dynamiques et le contexte peut changer rapidement. Il est donc crucial que le réseau continue à fonctionner correctement et de manière automatique pour s'adapter aux conditions changeantes. Ainsi, la conception de l'IoT doit tenir compte de la tolérance aux pannes et de la robustesse pour garantir la continuité de service en cas de défaillance d'un ou plusieurs appareils connectés.

1.9.7 Volume de stockage

Dans l'IoT, les objets intelligents peuvent collecter de petites quantités de données ou des volumes de données énormes en fonction du scénario et du contexte. Pour cela, le stockage des données est un aspect crucial à prendre en compte dans la conception d'un système IoT.

1.9.8 Solution optimisée en termes d'énergie

Le réseau de l'IoT est composé d'un grand nombre d'appareils interconnectés qui nécessitent une quantité considérable d'énergie pour maintenir le réseau en activité. Par conséquent, l'optimisation de l'énergie est un aspect majeur de l'IoT.

1.10 Avantages et inconvénients de l'IoT

1.10.1 Avantages

L'IoT est une technologie innovante qui permet de connecter des objets physique au monde numérique, cela permet aux utilisateurs de collecter des données en temps réel sur ces objets, de

surveiller leur état et de les contrôler à distance. Cette technologie a connu une adoption croissante en raison des nombreux avantages qu'elle offre, nous allons citer quelques avantages dans ce qui suit [15] [16] :

1.10.1.1 Communication

Grâce à l'IoT, une communication entre les différents appareils est possible, ce qui permet une connexion permanente et une transparence totale avec moins d'inefficacité et une meilleure qualité.

1.10.1.2 Collecte de donnée améliorée

La collecte de données contemporaine présente des lacunes en raison de ses limitations et de sa conception axée sur une utilisation passive. Cependant, l'IoT améliore considérablement cette situation en permettant aux humains d'analyser le monde qui les entoure de manière plus approfondie et d'obtenir ainsi une vision globale et précise de toutes les données nécessaires.

1.10.1.3 Automatisation et contrôle

Les machines automatisent et contrôlent une quantité considérable d'informations sans intervention humaine, ce qui permet d'obtenir des résultats plus rapides et précis.

1.10.1.4 Economie de l'argent et du temps

Grâce à l'utilisation de capteurs intelligents, l'IoT permet une surveillance précise de divers aspects de notre vie quotidienne pour des applications variées, ce qui permet d'économiser de l'argent et du temps tout en améliorant l'efficacité et la durabilité des activités humaines.

1.10.1.5 Optimisation de la technologie

L'IoT permet d'obtenir des données fonctionnelles et opérationnelles précieuses, qui peuvent être utilisés pour améliorer l'expérience client, mais également pour optimiser l'utilisation des périphériques et améliorer la technologie dans son ensemble.

1.10.1.6 L'engagement client

Les analyses ont des limites et des imperfections qui affectent leur précision, et l'engagement client est souvent passif. L'IoT vient révolutionner ce domaine, les entreprises peuvent obtenir un engagement plus interactif et performant avec leur public en utilisant des données plus précises et des interactions plus riches.

1.10.1.7 Réduction des déchets

L'IoT permet une meilleure identification des domaines à améliorer en fournissant des informations plus précises et en temps réel, contrairement aux analyses classique qui ne donnent souvent qu'un aperçu superficiel.

1.10.1.8 Nouvelles opportunité d'affaires

La technologie IoT offre de nouvelles perspectives d'activités commerciales qui simulent la croissance économique et génèrent des opportunités d'emploi supplémentaires.

1.10.2 Inconvénients

Malgré les nombreux avantages qu'offre l'IoT, il est crucial de reconnaitre qu'il comporte également des inconvénients notamment [15] [16] :

1.10.2.1 Complexité

Les systèmes IoT sont souvent considérés comme complexes car ils utilisent une variété de technologies et de nouvelles innovations pour leur conception, leur déploiement et leur maintenance. Cela peut rendre leur gestion difficile et nécessite une attention particulière pour éviter les problèmes qui peuvent survenir en cas de défaillance ou d'erreur.

1.10.2.2 Comptabilité

L'IoT implique l'interconnexion d'appareils provenant de différents fabricants, chacun ayant sa propre façon de concevoir ses produits ce qui peut rendre leur interconnexion difficile. Actuellement, il n'existe pas de normes internationales universelles de compatibilité pour les équipements de marquage et de surveillance utilisés dans l'IoT. Pour cette raison les entreprises qui déploient ses systèmes doivent être conscientes des problèmes qui peuvent survenir et travailler aves diligence pour les résoudre afin d'assurer le bon fonctionnement de leurs systèmes.

1.10.2.3 Sécurité/Vie privé

L'IoT crée un écosystème de dispositifs constamment connectés communiquant sur des réseaux. Malgré les mesures de sécurité, le système offre un peu de contrôle laissant les utilisateurs exposés à divers types d'attaquants. De plus, comme beaucoup de données relatives au contexte seront transmises par les capteurs intelligents, il y a un risque élevé de perte de données sensibles et privées.

1.10.2.4 Flexibilité

La flexibilité est une préoccupation cruciale pour de nombreuses personnes lorsqu'il s'agit de systèmes IoT. Elles craignent que les systèmes ne soient pas facilement intégrables entre eux, ce qui peut mener des conflits ou des blocages.

1.10.2.5 Coût

la mise en place d'un système IoT peut également être un inconvénient majeur pour certaines entreprises en raison des coûts élevés associés au développement, le déploiement et la maintenance d'un réseau de dispositifs connectés.

1.10.2.6 La technologie prend le contrôle de la vie

L'utilisation de la technologie est de plus en plus importante dans notre vie quotidienne et elle va probablement continuer à s'étendre avec l'IoT. La génération plus jeune a déjà une forte dépendance à la technologie pour toutes sortes d'activités, ce qui peut influencer les routines quotidiennes des utilisateurs.

1.11 Conclusion

Au cours des dernières années, l'Internet des Objets est devenu l'une des technologies les plus importantes du 21ème siècle, permettant la connectivité des objets du quotidien tels que les appareils électroménagers, les voitures, les thermostats, etc., avec Internet. Dans ce chapitre nous avons présenté une étude détaillée sur l'Internet des Objets, notamment sa définition, ses composantes, ses technologies et son architecture. Puis nous avons cité quelques domaines d'application d'Internet des Objets. Finalement, nous avons parlés brièvement sur leurs avantages et leurs inconvénients. Le prochain chapitre abordera les concepts de base de la théorie des jeux .

Chapitre 2

Théorie des jeux

2.1 Introduction

La théorie des jeux est une branche des mathématiques qui étudie les modèles de décision stratégique et les interactions entre les agents. L'histoire de cette discipline remonte aux années 1940 grâce aux contributions de John Von Neumann et Oskar Morgenstern depuis la publication de leur ouvrage " ***Theorie Of Game and Economic Behavior*** " en 1944. Dans les années 1950, John Nash a également apporté une contribution majeure à la théorie des jeux qui se focalise sur des situations impliquant des joueurs qui interagissent entre eux [17].

Ce chapitre commencera par la description des termes usuels de la théorie des jeux avant d'expliquer les concepts de base. Bien que tous les types de jeux ne soient pas couverts dans la suite, seules les notions nécessaires à savoir les jeux stochastiques, pour notre problématique qui se concentre sur l'analyse et la génération des graphes d'attaques dans le contexte IoT, seront présentées.

2.2 Définition d'un jeu

Un jeu est une situation dans laquelle des joueurs doivent choisir parmi un ensemble d'action possible, dans un cadre défini par des règles préétablies. Chaque résultat des choix effectués par les joueurs forme une issue du jeu, qui est associée à un gain pour chacun des participants. Ces résultats dépendent de l'interaction entre les choix des différents joueurs avec la possibilité que le hasard intervienne [18].

2.3 Composantes d'un jeu

Dans ce qui suit, nous allons décrire les éléments clés qui composent un jeu :

2.3.1 Joueurs

Un joueur **i** est un acteur ou une entité pouvant être une personne, une entreprise, un gouvernement, une cellule, un virus...etc, agissant dans leur propre intérêt selon le principe de la rationalité. [40].

2.3.2 Actions

Une action représente une décision prise par un joueur dans le cadre d'un jeu. L'ensemble des actions permises au joueur $\boldsymbol{i}$ est défini comme suit [18] :

$$A_i = \{\ a_1, a_2, \ldots, a_n\ \}.$$

2.3.3 Stratégies

Une stratégie est un ensemble d'actions qu'un joueur choisit de prendre à chaque instant dans le cadre d'un jeu donné. Un profil de stratégies $S = \{s_1, s_2, ...s_n\}$ où $s_1 \in S_i$ pour $i = 1$ à n, n est un n-uplet ordonné composé d'une stratégie pour chacun des n joueurs participants au jeu [18].

Il existe deux types de stratégies [20] :

1.3.3.1 Stratégies pures

Une stratégie pure du joueur $\boldsymbol{i}$ est un plan d'actions qui prescrit une action de ce joueur à chaque fois qu'il est susceptible de jouer.

1.3.3.2 Stratégies mixtes

Une stratégie mixte est une stratégie où le joueur choisit au hasard le coup qu'il joue parmi les coups possibles. Cela revient à attribuer une certaine distribution de probabilités sur l'ensemble des stratégies pures du jeu. Ainsi, l'ensemble des stratégies mixtes d'un joueur $\boldsymbol{i}$ est défini comme suit :

$$\Delta_i = \Big\{\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R} \mid \alpha_j \in \{0, 1\}, \forall j = 1, \ldots, n, \textstyle\sum_{j=1}^{n_i} \alpha_j = 1\Big\}.$$

D'où :

n_i : est le nombre de stratégies pures du joueur i.

α_i : est la probabilité que la stratégie s_i soit jouée.

2.3.4 Utilité

Chaque joueur $\boldsymbol{i}$ est associé à une fonction d'utilité, qui représente le bénéfice qu'il reçoit en fonction des stratégies choisies par lui-même et par les autres joueurs [18], la fonction d'utilité est donnée comme suit :

$$\mathbf{u}_i : S_1 \times \ldots \times S_n \to \mathbb{R}.$$

2.3.5 Issue

Une issue d'un jeu à n joueurs est un n-uplet $(U_1(S), \ldots, U_n(S))$ ordonné et composé des revenus (gains) des n joueurs participants au jeu lorsque le profil de stratégies $S = (s_1, s_2, \ldots, s_n)$ est joué.

Un jeu est décrit comme suit [18] :

$$\langle I, \{S_i\}_{i \in I}, \{U_i\}_{i \in I} \rangle.$$

Où : $\mathbf{I}$ = l'ensemble des joueurs,
$\mathbf{S}_i$ = l'ensemble des stratégies.
$\mathbf{U}_i$ = l'ensemble des utilités.

2.4 Types des jeux

Etant donnée la multitude de situations conflictuelles et l'utilisation diverse de la théorie des jeux, il existe plusieurs catégories de jeux qui peuvent être classées en fonction de différents critères.

2.4.1 Classification selon les relations entre les joueurs

La théorie des jeux se distingue en deux branches, les jeux non coopératifs et les jeux coopératifs.

1.4.1.1 Jeux coopératifs / non coopératifs

Dans les jeux coopératifs, les joueurs travaillent ensemble en formant des groupes également appelés des coalitions et prennent des mesures communes afin d'atteindre leurs objectifs. En revanche, dans les jeux non coopératifs, les joueurs sont supposés choisir leurs actions individuellement, en cherchant égoïstement à atteindre leurs propres objectifs et à maximiser leurs propres profits [18].

2.4.2 Classification selon le nombre de coups

Les jeux sous forme normale et les jeux sous forme extensive sont deux modèle différents utilisés en théorie des jeux pour décrire les interactions stratégiques entre les joueurs [20].

1.4.2.1 Jeux sous forme normale (Stratégique)

Un jeu sous forme normale également appelé sous forme stratégique est résumé par la collection : $\Gamma = (N, S, U)$ où :

N : est l'ensemble de joueurs.

S : représente l'ensemble de combinaisons stratégiques.

U : une fonction d'utilité définie sur S.

Ces jeux se déroulent en un seul coup. La forme normale d'un jeu peut être utiliser dans le cas ou les joueurs interviennent simultanément. Dans le cas des jeux finis, les jeux sous forme normale sont caractérisés par une matrice de paiement qui spécifie les gains ou les pertes pour chaque joueur en fonction de ses actions et celles des autres joueurs [20].

Exemple 1 :

Un homme (Joueur1) et une femme (Joueur2) aillant au cinéma. Une fois sur place, ils doivent choisir entre aller voir un documentaire ou une comédie, l'un des deux préfèrent les documentaires et l'autre les comédies, mais tous les deux préfèrent voir un film ensemble que séparément.

Les stratégies disponibles pour chacun des joueurs en considérant qu'ils font leur choix simultanément sont alors :

- Aller voir un documentaire, noté " D ".
- Aller voir une comédie, noté " C ".

Nous avons donc un jeu non coopératif avec :

1) N = {Joueur1, Joueur2}.

2) $S = S_1 = S_2 = \{D, C\}$.

3) Gains : représentent le niveau de plaisir atteint par le joueur.

Le tableau 2.1 représente la forme stratégique de ce jeu :

Joueur1	Joueur2	
	D	C
D	(2,3)	(1,1)
C	(1,1)	(3,2)

TABLE 2.1 – Guerre des sexes représentées sous forme stratégique [19].

1.4.2.2 Jeux sous forme extensive (étendue)

La forme extensive est utilisée dans le cas où les joueurs prennent des décisions de manière séquentielle. Cette forme symbolise en effet très bien l'idée de succession et d'enchainement des coups se modélise par l'arbre de Kuhn. La forme extensive est d'avantage utilisé dans les jeux à information parfaite. La forme extensive d'un jeu spécifie les données suivantes [18] :

- Les joueurs concernés par le jeu.
- Les moments où chaque joueur aura joue à jouer.
- Les actions possibles de chaque joueur au moment où il joue.
- L'information dont dispose chaque joueur au moment où il joue.
- Les paiement des joueurs pour chacune des combinaisons possibles.

La Figure 2.1représente un exemple de la forme extensive d'un jeu.

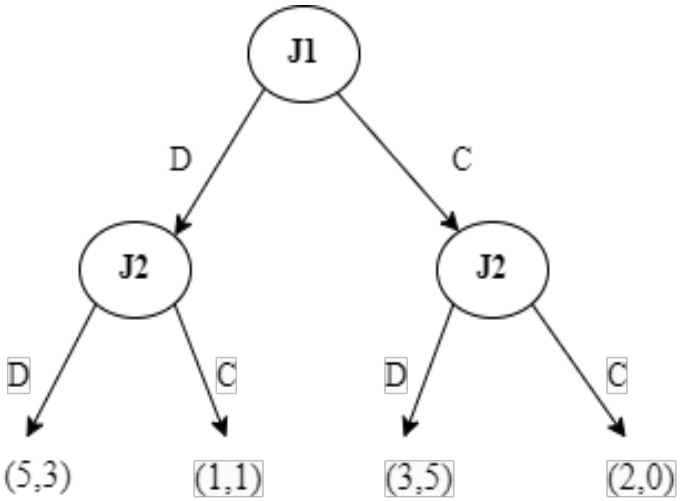


FIGURE 2.1 – Forme extensive d'un jeu [17].

2.4.3 Classification selon l'information que possède chaque joueur

1.4.3.1 Jeux à information complète / incomplète

Un jeu est dit à information complète si tous les joueurs connaissent les règles de jeu. Par ailleurs, un jeu est dit à information incomplète, si au moins un des joueurs ne connait pas entièrement les règles de jeu [20].

1.4.3.2 Jeux à information parfaite / imparfaite

Dans les jeux à information parfaite, tous les joueurs disposent à chaque étape de jeu d'information complète sur tous les mouvements effectues aux étapes précédentes y compris les mouvements personnels et les mouvements aléatoire. En contrepartie, dans les jeux à information imparfaite à un certains stade de jeu, les joueurs n'auront que des informations partielles ou aucune information sur les mouvements effectues aux étapes précédentes [21].

2.4.4 Classification selon les gains (ou utilité)

1.4.4.1 Jeux à somme nulle / non nulle

Les jeux à somme nulle représentent les jeux dont les intérêts des joueurs sont complètement opposés. Ce qui est gagné par un joueur est perdu par d'autre joueur, il n'y a pas de coopération possible. Ce type de jeu ne permet pas de présenter toujours la réalité, pour cela il est nécessaire d'introduire les jeux à somme non nulle c'est-à-dire les gains diffèrent des pertes [18].

2.4.5 Jeux finis

Un jeu est dit fini, si tous les ensembles de stratégies des joueurs sont finis, C'est-à-dire :

$$\langle I, S_i i \in I, u_i i \in I \rangle, \text{ où } |S_i| < \infty, \forall i. \text{ [19].}$$

2.4.6 Jeux stochastiques

Un jeu stochastique est défini par le tuple [22] :

$$< A_g, A_i : \ i = 1...|A_g|, \ S, \ T >.$$

A_g : Ensemble fini des joueurs.

A_i : Ensemble fini des stratégies du joueur i ($i \in A_g$).

R_i : Fonction de récompense du joueur i, $R_i(a) \to \mathbb{R}$.

S :Ensemble fini des états du jeu.

T :Fonction de transition du jeu, $T : S \times A \times S \to [0, 1]$ Elle indique la probabilité de passer d'un état $s \in S$ à un état $s' \in S$ en exécutant l'action $a \in A$. La notation de l'action jointe n'est pas nécessaire, car l'action a est déjà définie comme un élément de l'ensemble des actions A. Cette fonction vérifie la propriété de Markov (le futur ne dépend que du présent), c'est pourquoi les jeux stochastiques sont souvent appelés jeux de Markov.

Les jeux stochastiques sont des jeux ou les résultats possibles dépends à la fois des décisions des joueurs et de facteurs aléatoires c'est-à-dire les jeux stochastiques multi-agents ou les agents prennent des décisions en fonction de leur propre information et des informations des autres joueurs. Ces jeux sont utilisés pour modéliser des situations ou l'incertitude est un facteur important, tels que l'économie, la finance, etc.

2.4.7 Jeux stochastiques partiellement observable (POSG)

Un jeu stochastique partiellement observable (Partially Observable Stochastic Game POSG) entre deux joueurs est un modèle mathématique utilisé en théorie des jeux pour modéliser les situations où plusieurs joueurs interagissent dans un environnement incertain et partiellement observable. Un POSG est définit comme étant un tuple (N, S, A, O, P, R) avec [23] :

N = {1, 2} est l'ensemble des joueurs, le joueur 1 étant le défenseur du réseau et le joueur 2 l'attaquant.

S est un ensemble fini de tous les états possibles.

$A = A_1 \times A_2$ est l'espace d'action du jeu, où A_1 et A_2 sont les espaces d'action du défenseur et de l'attaquant, respectivement. Une action $A_1^t \in A_1$ jouée par le défenseur au temps t définit l'état du réseau. Une action $A_2^t \in A_2$ au temps t détermine la récompense des deux joueurs.

O = O1 × O2 est un ensemble fini d'observations.

$P = [p_{i,j}]$ est un ensemble de probabilités de transition d'état et d'observation de Markov. $Pr(s_j, o \mid s_i, a_1, a_2)$ est la probabilité de transition vers l'état s_j et l'observation o à partir de l'état s_i sous l'action conjointe (a_1, a_2).

$R = \{R_1, R_2\}$, où $R_1 + R_2 = 0$. $R_1 : S \times A \to R$ est la fonction de récompense pour le défenseur et R_2 est la fonction de récompense pour l'attaquant.

2.5 Quelques concepts de solutions

2.5.1 Elimination de stratégies dominées

Dans tout jeu, une stratégie d'un joueur "domine strictement/faiblement" une autre stratégie si elle est strictement/faiblement supérieur, quoi que fassent les autres joueurs [18].

Définition 1.5.1

- Une stratégie est strictement dominée pour le joueur i, s'il existe une autre stratégie qui est strictement meilleure que quellques soient les stratégies des autres joueurs. i.e [18].
- Une stratégie est faiblement dominée pour le joueur i, s'il existe une autre stratégie qui est au moins aussi bonne quelles que soient les stratégies des autres joueurs et strictement meilleure que pour au moins une combinaison de stratégies. i.e [18].

$$u_i(s'_i, s_{-i}) \geq u_i(s_i, s_{-i}),\ \forall\ s_i \in S_{-i}$$
$$u_i(s'_i, s_{-i}) > u_i(s_i, s_{-i}),\ \text{pour au moins un } s_{-i} \in S_{-i}.$$

Exemple 2 :
Considérant le jeu non coopératif le tableau 2.2 [18] :

Joueur1	Joueur2	
	u	v
x	(2,1)	(0,1)
y	(3,1)	(4,2)

TABLE 2.2 – Forme normale d'un jeu [18].

- La stratégie x est strictement dominée pour le joueur 1.
- La stratégie u est faiblement dominée pour le joueur 2.

2.5.2 Equilibre de Nash

L'équilibre de Nash est une notion très importante dans la théorie des jeux. C'est une situation dans laquelle aucun des joueurs ne souhaite modifier son comportement étant donné le comportement de l'autre c'est-à-dire une situation telle qu'aucun joueur n'a intérêt à dévier (seul) de la situation obtenue.

Définition 1.5.2 Equilibre de Nash pure

un équilibre de Nash du jeu J est un profil de stratégies
$\mathbf{s}^* = (\mathbf{s}_1^*, ..., \mathbf{s}_n^*) \in \mathbf{S}$ tel que pour tout joueur $i \in \mathbf{N}$, et pour toute stratégie $\mathbf{s}' \in \mathbf{S}_i$ on a :

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}'),\ \forall\ s_i \in S_i,\ \forall i \in N \text{ [18].}$$

Définition 1.5.3 Equilibre de Nash mixte

Une situation $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*) \in \Delta = \prod_{k=1}^{N} \Delta A_k$ est un équilibre de Nash d'un jeu en stratégies mixtes, si pour chaque joueur $k \in N$, nous avons :

$$u_i(\alpha_i^*, \alpha_{-i}^*) \geq u_i(\beta_i^*, \alpha_i^*), \forall \beta_i \in \Delta_i, \forall i = 1, \ldots, N.$$

Avec α_i l'action jouée par le joueur i et α_{-i} le profil d'action jouée par tous les joueurs à l'exception du joueur i [17].

Proposition 1.5.1

Tout équilibre de Nash en stratégies pures et aussi un équilibre de Nash en stratégie mixtes [18].

Théorème 1.5.1

Tout jeu fini, admet au moins un équilibre de Nash en stratégies mixtes [18].

2.6 Conclusion

Dans ce chapitre, nous avons introduit les concepts clés de la théorie des jeux. Grâce à celle-ci, nous pouvons modéliser les interactions entre les attaquants et les défenseurs dans un réseau, ce qui nous permet d'évaluer les conséquences des actions et des réactions des différentes parties engagées sur la sécurité du système. En comprenant les stratégies adoptées par les attaquants, nous sommes en mesure de concevoir des contre-mesures efficaces visant à protéger les systèmes.

Le prochain chapitre abordera quelques travaux sur l'analyse des graphes d'attaques dans un contexte IoT.

Chapitre 3

Quelques travaux antérieurs sur l'analyse des graphes d'attaques et l'Internet des objets

3.1 Introduction

L'IoT est une tendance technologique majeure qui a vu le jour au cours des dernières années. Les réseaux IoT sont des réseaux de dispositifs connectés qui permettent la communication et l'échange de données entre des objets physiques et virtuels. Afin de maintenir le fonctionnement efficace de ces réseaux, il est nécessaire de les protéger contre les éventuelles attaques susceptibles de les perturber ou de les endommager.

Actuellement l'attaquant développe des scenarios d'attaques qui sont représentés dans la littérature par des graphes d'attaques. Ces derniers déterminent l'ensemble des privilèges et les contrôles d'accès acquis par l'attaquant en exploitant les vulnérabilités existantes dans un réseau et énumèrent les chemins d'attaques possibles qui portent dommage à la sécurité des réseaux informatiques.

Les graphes d'attaques sont un outil précieux pour les défenseurs des réseaux, illustrant les chemins qu'un attaquant peut emprunter pour accéder à un réseau ciblé [24]. Par ailleurs, ce graphe peut être obtenu à partir de la topologie du réseau et les informations sur les vulnérabilités de chaque machine.

L'analyse des graphes d'attaques est cruciale pour la sécurité informatique, car elle permet de révéler les vulnérabilités qui pourraient compromettre le système. Les attaquants peuvent exploiter ces vulnérabilités pour mener leurs attaques. Par conséquent, l'analyse des graphes d'attaques permet de comprendre les scénarios d'attaques possibles et d'identifier les mesures de protection à mettre en place pour prévenir ou atténuer les risques de sécurité.

Notre objectif dans ce travail est de développer une méthode innovante basée sur la théorie des jeux pour analyser des graphes d'attaques dans les systèmes IoT, afin de proposer une stratégie de défense optimale contre les attaques malveillantes.

3.2 Concepts de base

3.2.1 Vulnérabilité

C'est une faille, ou une erreur (bug) qui peut mener à un compromis imprévu dans le système de sécurité, elle est vue comme un trou pouvant être exploité par l'attaquant pour s'introduire dans le système cible.

Exemple1 : Faille d'injection SQL
Les failles d'injection SQL consistent en l'insuffisantes de validation des entrées afin de s'assurer que les données entrées par les utilisateurs ne s'immiscent pas dans le code, se faisant ainsi passer pour du code côté serveur.

3.2.2 Attaque informatique

Une attaque est l'exploitation d'une faille d'un système informatique à des fins non reconnues par l'exploitant du système, généralement ils sont préjudiciables (dommageables) [25].Les motivations pour les attaques peuvent être de divers types qui sont :

- Obtenir un accès au système.
- Perturber le bon fonctionnement d'un service.
- Avoir des informations sur l'organisation (entreprise de l'utilisateur, etc.).
- Utiliser le système de l'utilisateur en tant qu'intermédiaire pour déclencher une attaque.
- Récupérer des données d'un système informatique.
- etc.

Exemple 1 : Attaque par injection SQL

Le principe d'une injection SQL est de fermer la requête prématurément en insérant une simple quotte " ' " puis en complétant la requête afin d'obtenir une autre réponse que celle qui doit normalement être donnée à l'utilisateur tel qu'il est illustré dans la Figure 3.1.

3.2.2.1 Types d'attaques informatique

Les attaquants utilisent plusieurs techniques d'attaques, qui peuvent être classées en trois catégories différentes [25] : Les attaques directes, les attaques indirectes par rebond et les attaques indirectes par réponse.

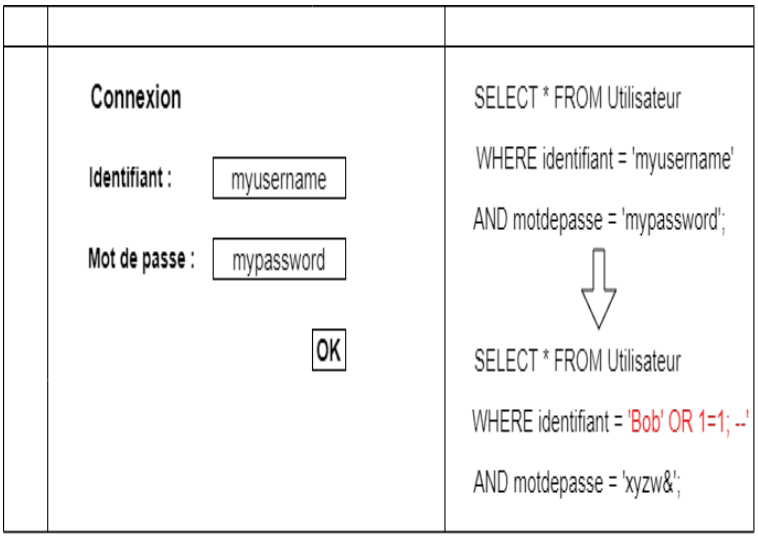


FIGURE 3.1 – Vulnérabilité d'injection SQL [26].

A. Attaques directes

Il s'agit de l'attaque la plus simple à réaliser, l'attaquant utilise directement son ordinateur pour attaquer sa victime. En effet, les programmes de hack qu'ils utilisent ne sont que faiblement paramétrable, et la plupart de ces logiciels envoient directement les paquets à la victime. La Figure 3.2 représente l'attaque directe.

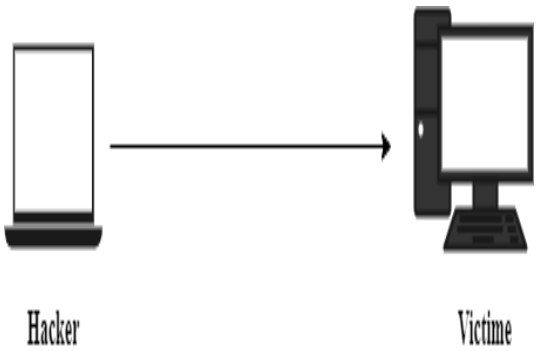


FIGURE 3.2 – Attaque directe.

B. Attaques indirectes par rebond

Ce type d'attaque consiste à utiliser un ordinateur intermédiaire qui va recevoir les paquets d'attaques et qui répercute l'attaque vers la victime. En effet le rebond a deux avantages :

- Masquer l'identité de l'attaquant (l'adresse IP).
- Utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant (CPU, bande passante, etc.).

La Figure 3.3 montre une attaque indirecte par rebond.

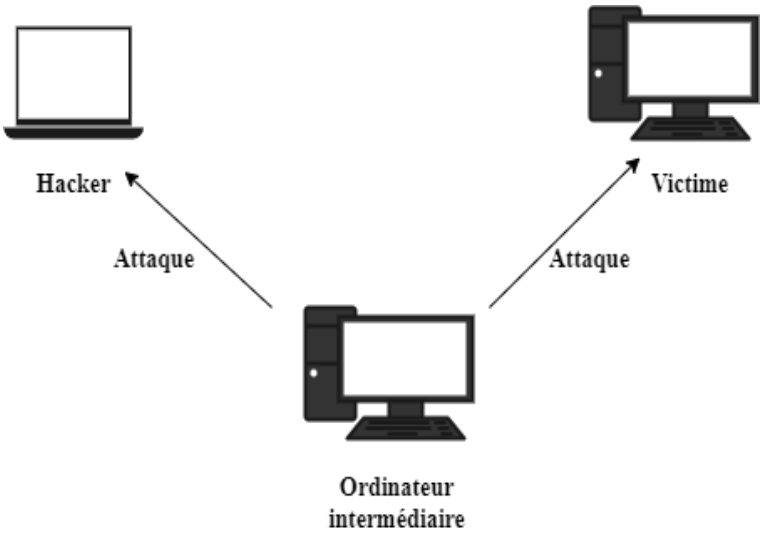


FIGURE 3.3 – Attaque indirecte par rebond.

C. Attaques indirectes par réponse

Cette attaque est une variante de l'attaque par rebond. Elle offre les mêmes avantages, du point de vue du pirate. Mais au lieu d'envoyer une attaque à l'ordinateur, il lui envoie une requête dont la réponse sera envoyée à la victime. La Figure 3.4 représente une attaque indirecte par réponse.

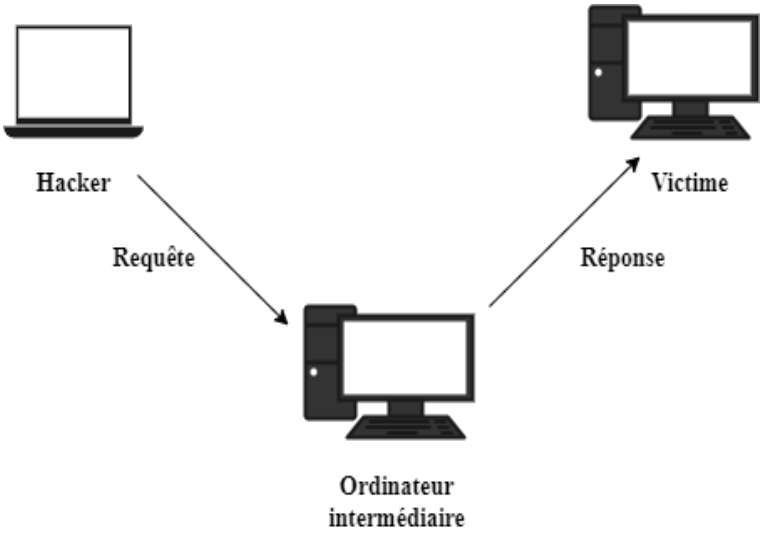


FIGURE 3.4 – Attaque indirecte par réponse.

3.2.3 Exploit

Un exploit est une série d'étapes qui tirent parti d'une ou plusieurs vulnérabilités d'un système cible pour exposer des fonctionnalités spécifiques à l'attaquant [27]. Les exploits peuvent être

définis comme un prédicat de la forme : v (hs,hd) d'où

- v : la vulnérabilité exploitée.
- hs : l'hôte source, c'est-à-dire l'hôte qui commet l'exploit.
- hd : la cible de l'exploit.

3.2.4 Pré-condition

C'est un ensemble de propriétés du système qui doivent exister pour qu'un exploit soit réussi. Une pré-condition initiale est une propriété du système qui existe de façon intégrée et qui n'est pas apparue après une exploitation. Les types de pré-condition sont [27] :

- Statuts/services : la cible détient ou annonce des versions particulières d'exploitation, de logiciels systèmes, ou se trouve dans un état matériel/logiciel particulier.
- Accessibilité : la cible est accessible.
- Les capacités du pirate : le pirate a des capacités particulières, comme la capacité d'exécuter un processus sur une cible, l'accès à des outils ou des niveaux de privilèges.

3.2.5 Post-condition

Le succès à voiler un exploit donne lieu à une ou plusieurs postconditions. Ces conditions peuvent également constituer les préconditions d'autres exploits [27].

3.2.6 Scénario d'attaques

Un scénario d'attaque est un ensemble d'étapes et des manières dont l'attaquant peut utiliser les vulnérabilités du système afin d'atteindre son objectif.

Exemple intuitif

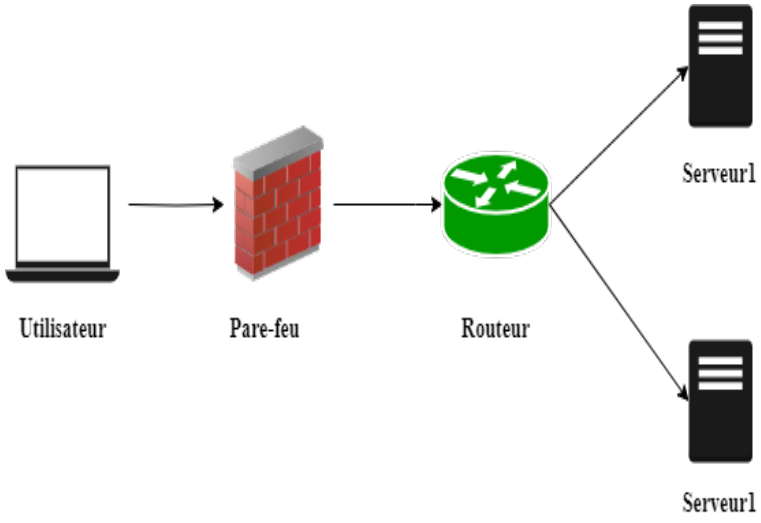


FIGURE 3.5 – Exemple d'un réseau [28].

Soient trois machines, qui représentent respectivement,l'utilisateur, le serveur web et le serveur de base de données. Le pare feu permet d'envoyer des requêtes " http " et "ssh " de l'utilisateur vers le serveur web. Au cours du fonctionnement normal, l'utilisateur effectue une requêtes " http" au serveur 1, qui passe par le pare feu. Le serveur web accède au serveur de base de données pour obtenir les données dont il a besoin et puis, communiquent avec l'utilisateur avec " http ". Si l'utilisateur tente d'accéder au serveur 2 directement, le pare-feu bloque la communication. Une demande de "ssh" de l'utilisateur au serveur 2 est considérée comme un comportement anormal qui est bloqué par le pare-feu. En outre, la base de données sur le serveur 2 aurait des données privées des utilisateurs autres que celui de l'utilisateur. Sauf si une attaque par injection de commande est lancée avec succès sur le serveur web pour le compromettre. Ensuite, à l'aide d'un invité de commande compromit, une attaque par injection SQL est lancée sur la base de données du serveur2. La donnée est siphonnée au serveur 1 puis l'utilisateur. Le réseau correspondant à cet exemple est représenté par la Figure 3.5 [28].

3.2.7 Graphe d'attaques

3.2.7.1 Définition 1

Un graphe d'attaques est une représentation précise de tous les chemins qui, à travers un système, aboutissent à un état où un intrus a réussi à atteindre son objectif.

Il existe deux formes populaires de graphe d'attaque. Le premier est un graphe direct ou les nœuds représentent les états du réseau (généralement spécifiée par les attributs de réseau concernés tels que la connectivité entre la source, la victime et le privilège d'accès d'un attaquant dans l'état) et les arêtes représentent les exploits qui transforment un état en un état plus compromis, montrant finalement une attaque réussie. Une deuxième forme est un graphe direct ou les nœuds représentent les conséquences d'avoir une précondition qui permet une postcondition d'exploit [37].

Selon Albanese et al. [29], un graphe d'attaques est défini comme suit :

3.2.7.2 Définition 2

Plus formellement : Étant donné un ensemble d'exploits E, un ensemble de conditions de sécurité C, une relation d'exigence $RT \subseteq C \times E$ (qui indique qu'un exploit E exige un ensemble de conditions C pour qu'il soit réussi), et une relation d'implication $Ri \subseteq E \times C$ (qui indique que la violation d'un exploit E implique et donne lieu à un ensemble de postconditions C). Un graphe d'attaques G est le graphe orienté $G = (E \cup C, RT \cup Ri)$, où $E \cup C$ est l'ensemble des sommets et $RT \cup Ri$ l'ensemble des arêtes. Un exemple de cette définition est représenté par la Figure 3.6.

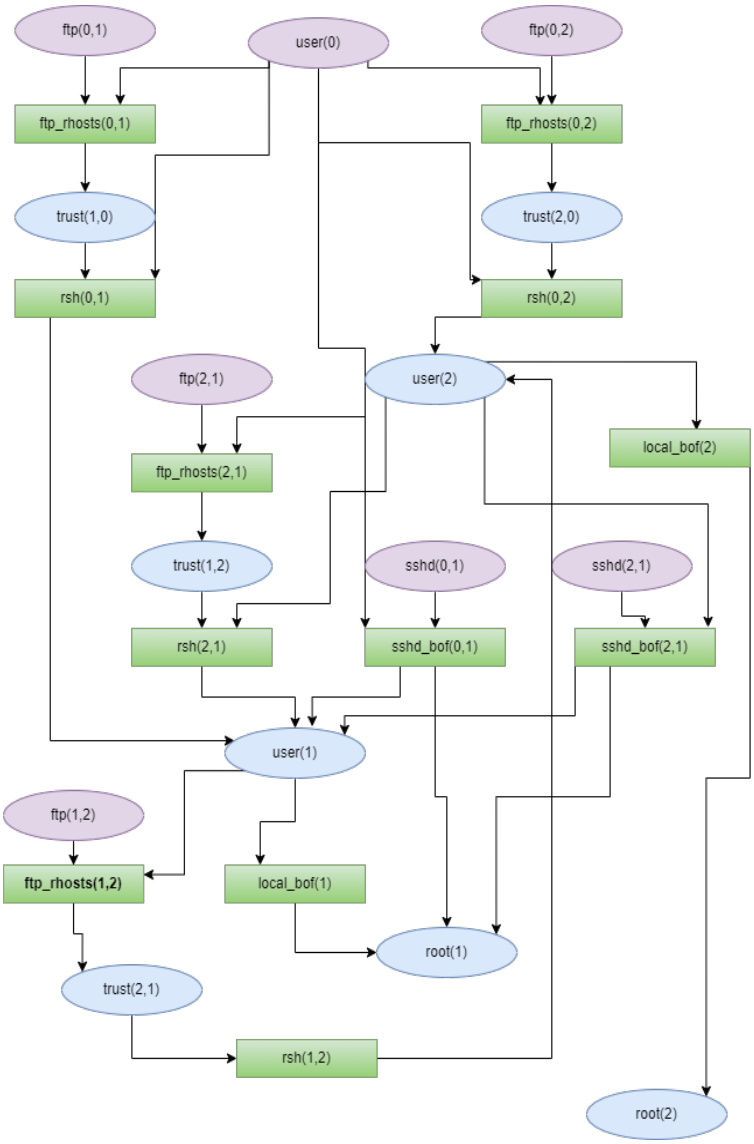


FIGURE 3.6 – Exemple de graphe d'attaques, comprenant les conditions initiales (ovales violets), les exploits (rectangles verts) et les conditions intermédiaires (ovales bleus) [29].

3.3 Quelques travaux antérieurs sur l'analyse des graphes d'attaques

Dans cette section, notre attention est portée sur les travaux qui se concentrent sur l'analyse des graphes d'attaques. Nous allons présenter plusieurs travaux qui utilisent des graphes d'attaques pour résoudre les différents problèmes liés à la sécurité informatique.

3.3.1 Game theory approach for analysing attack graphs

Bouafia et Hamza (2022) [30] ont proposé une nouvelle méthode d'analyse des graphes d'attaques qui s'appuie sur la théorie des jeux ou ils cherchent à aider l'administrateur à protéger son réseau contre les attaques menées par les intrus. Ils ont modélisé cette situation sous forme d'un jeu non coopératif à deux joueurs et à somme non nulle avec un ensembles de stratégies finies. Les auteurs ont pris comme exemple d'application de leur approche la Figure 3.7.

Les auteurs montrent par la Figure 3.7 un exemple d'un réseau composé d'un pare-feu qui sépare un réseau interne d'un réseau externe. Le réseau externe se compose d'un ordinateur de l'attaquant (intru) connecté via Internet à un routeur qui permet d'établir la connexion avec le réseau externe composé de deux ordinateurs : A qui exécute les services ftp et sshd et l'ordinateur B qui exécute ftp et la base de données. Un IDS qui voit le trafic du réseau entre le réseau interne et externe. L'attaquant lance son attaque en commençant par un seul ordinateur qui se trouve dans le réseau externe. Plus précisément, le but de l'attaquant est de perturber le fonctionnement de la base de données de B ; pour atteindre cet objectif, il a besoin d'un accès root à ce dernier.

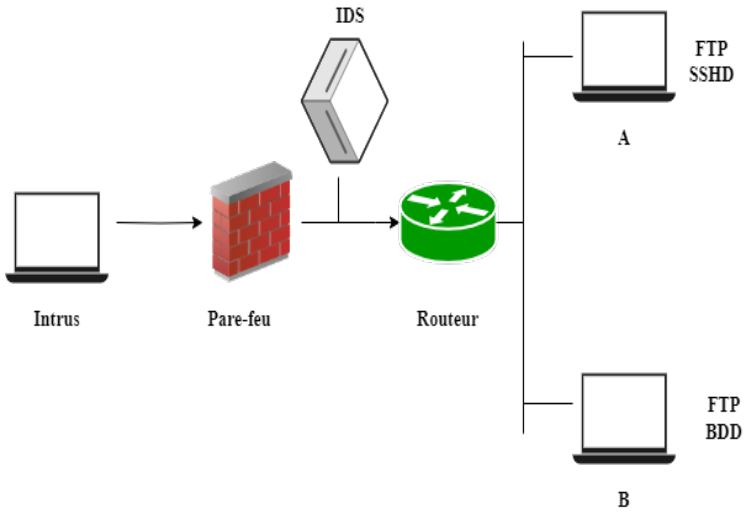


FIGURE 3.7 – Exemple de réseau [30].

Le jeu se déroule entre deux joueurs : l'attaquant comme 1er joueur et l'administrateur comme 2ème joueur.

Chaque joueur possède ses propres actions qui sont définit comme suit :

- Les stratégies de l'attaquant sont : S1= {sshdbuffer overflow, ftp rhost,local buffer overflow}.

Les coûts associés à ces actions sont présentés dans le tableau 3.1.

- Les stratégies de l'administrateur sont : S2= {Generate alarm, IP blocking,Isolate host,Kill process,No defence}.

Les coûts associés à ces actions sont présentés dans le tableau 3.2.

Dans cet article les auteurs ont défini la fonction d'utilité des deux joueurs comme suit :

Stratégies d'attaque	coûts
sshdbuffer overflow	10
Ftp rhost	5
Local buffer overflow	9

TABLE 3.1 – Tableau des stratégies d'attaques [30].

Stratégies de défense	coûts
Generatealarm	8
IP blocking	5
Isolatehost	10
Kill process	4
No defence	0

TABLE 3.2 – Tableau des stratégies de défense [30].

$$U_i : S1 \times S2 \to \mathbb{R}$$
$u_1 : S1 \times S2 \to R$ est la fonction d'utilité de l'attaquant.
$(s_i{}^1, s_J^2) \to u_1(s_i{}^1, s_J^2) = a_{ij}, i = 1 \dots n; j = 1 \dots m.$
$u_2 : S1 \times S2 \to R$ est la fonction d'utilité de défendeur.
$(s_i{}^1, s_J^2) \to u_1(s_i{}^1, s_J^2) = b_{ij}, i = 1 \dots n; j = 1 \dots m.$. Avec $s_i^1 \in S1$ et $s_J^2 \in S2$.

La solution proposée par les réalisateurs de cet article consiste à représenter ce jeu sous forme normale c'est à dire un tableau de deux dimensions qui contient l'ensemble des joueurs, leurs stratégies et les issues correspondantes pour ce dernier. Ensuite, ils ont opté pour l'analyse de graphe d'attaques en utilisant l'algorithme d'élimination itératif des stratégies dominées 1, avec l'élimination des lignes correspondant aux stratégies dominées de l'attaquant et les colonnes correspondante aux stratégies dominées de l'administrateur. Sur le graphe d'attaques,ils suppriment les nœuds correspondants à ces derniers.

A la fin, ils obtiennent l'ensemble des stratégies non dominées ; dans cet exemple le résultat correspond à l'équilibre de Nash.

La Figure 3.8 représente le graphe des attaques de l'exemple étudier. Les nœuds bleus représentent les ordinateurs de l'attaquant et d'administrateur et les nœuds roses représentent les services exécutés dans tout le réseau et que l'intrus utilisera comme stratégies d'attaques pour atteindre son but.

Bouafia et Hamza ont appliqué le processus d'élimination des stratégies dominante, ils ont supprimé les stratégies "local buffer overflow et sshd buffer overflow " cette élimination a permet d'obtenir un nouveau graphe d'attaque avec des vulnérabilités minimales représenté par la Figure 3.9.

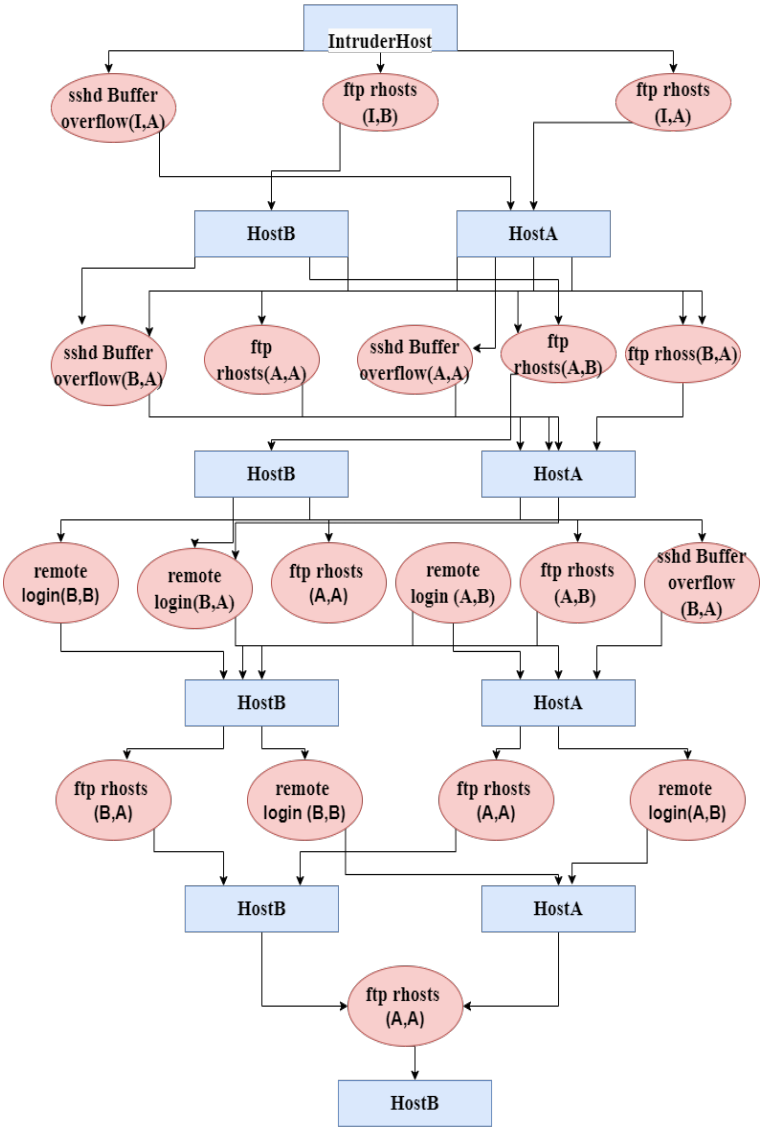


FIGURE 3.8 – Graphe d'attaques correspondant à la topologie du réseau étudié [30].

3.3.2 A game-theoretic framework for dynamic cyber deception in Internet of Battlefield Things (IoBT)

Anwar et al. (2019) [23] ont présenté une méthode de la théorie des jeux pour la cyber tromperie dans le but de protéger les nœuds vitaux des réseaux informatiques. Ils ont modélisé l'interaction dynamique contradictoire entre l'administrateur et l'attaquant par un jeu stochastique partiellement observable POSG. Leur approche consiste à insérer des *pots de miel*[1], appelés aussi Honey pots (HP), pour tromper l'attaquant et identifier les vulnérabilités dans le graphe d'attaques à

1. est un type de technologie de tromperie conçu pour attirer les attaquants potentiels dans un piège, il s'agit d'un système ou un réseau leurre qui semble être une cible légitime mais qui en fait mise en place dans le but de détecter, de compromettre la sécurité dans un système réel.

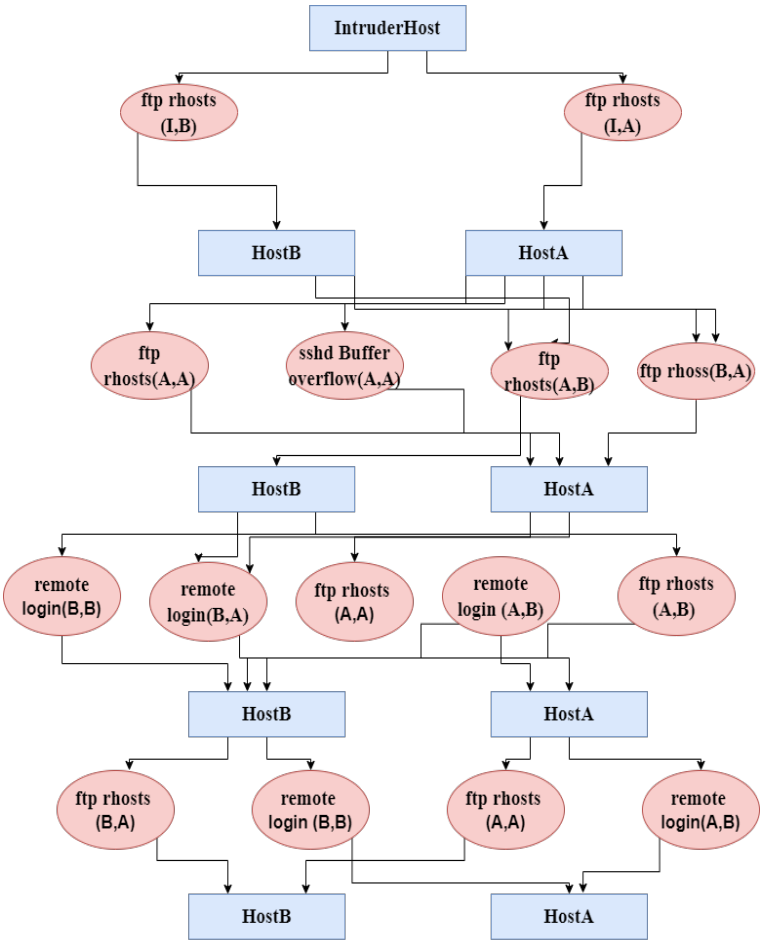


FIGURE 3.9 – Graphe d'attaques analysé [23].

l'aide de l'utilisation de POMDP (Partially Observable Markov Decision Processes), utilisé dans le domaine de l'IoT (Internet des Objets), ainsi que dans le domaine militaire de l'IoBT [2]. Les joueurs disposent d'informations partielles, et le défenseur cherche un déploiement optimal des pots de miel pour minimiser ses pertes dans le réseau avec un minimum de coûts, tandis que l'attaquant tente d'exploiter les vulnérabilités existantes pour infliger un maximum de dégâts et éviter l'interaction avec les pots de miel.

L'approche de Anewar et al. [23] est modélisée comme un jeu non coopératif, fini (à cause de la taille fixe des graphes d'attaques) et à somme nulle. Pour ce faire, les auteurs ont défini le graphe d'attaques comme suit : G (V, E) de sorte que V représente les nœuds (hôtes) et E représente

2. Internet des Objets dans le domaine militaire qui fait également référence aux appareils utiles pour les batailles militaires qui peuvent communiquer sur des réseaux tactiques autres qu'Internet.

les arêtes. Une arête est dirigée une fois l'attaquant atteint un hôte (v) depuis un autre hôte (u) en exploitant les vulnérabilités existantes. Dans cet article, le graphe d'attaques est créé à partir de la configuration des nœuds et les vulnérabilités actuelles de réseau. Il est généré à partir de type d'hôte des nœuds et les vulnérabilités associées à ce dernier. Par la suite, un jeu stochastique partiellement observables est défini comme un tuple (N, S, A, O, P, R) où :

- $N = \{1, 2\}$ sachant que "1" représente l'administrateur qui est le 1er joueur, "2" représente l'attaquant comme 2^{e}.

- S fait référence à l'ensemble fini des états, où les auteurs ont défini l'espace d'états ainsi : un état $s \in S$, $s = (E_s, I_s, T_s)$, avec :

- E_s : est l'ensemble de vulnérabilités observé par les deux joueurs.
- I_s : identificateur de chaque vulnérabilité qui détermine si elle appartient à un nœud normal ou à un pot de miel.
- T_s : l'ensemble de vulnérabilité exploiter jusqu'à s ; l'attaquant sait exactement les vulnérabilités qui ont été exploité et la structure de graphe d'attaques. Le graphe d'attaques évolue de l'état s à l'état s'.

$A = A_1 \times A_2$ est l'ensemble des actions des deux joueurs, d'où une action $a_1 \in A_1$ jouée par le défenseur à l'instant t détermine l'état du réseau. Par ailleurs, une action $a_2 \in A_2$ jouée par l'attaquant à l'instant t définit la récompense des deux joueurs.

A) Premièrement : les actions de défenseur

L'administrateur utilise MulVal [3] pour générer automatiquement le graphe d'attaques, et vu l'incertitude de l'attaquant sur l'état réel du graphe d'attaques, cela permet au défenseur d'augmenter cette dernière en jouant a_1 : insertion d'un pont du miel. Pour cela, l'ensemble de vulnérabilités change et automatiquement le graphe change. Un certain coût $C_1 = (s, a_1)$ est associé à chaque action de joueur 1. $N(N-1)$ est le nombre d'actions pures du défenseur où $N = |V|$.

B) Deuxièmement : les actions de l'attaquant

L'attaquant sait que le réseau observé est trompeur et ne connait pas ou les Honey pots sont placés. Compte tenu des informations imparfaites collectées lors de l'étape de reconnaissance, l'action a2 de l'attaquant consiste à sélectionner la vulnérabilité à exploiter ultérieurement.

- $O = O_1 \times O_2$ est l'ensemble fini des observations, ces dernières sont asymétriques pour les deux joueurs.

3. c'est un logiciel basé sur l'analyse des vulnérabilités. Il sert à détecter les attaques en analysant les failles potentielles dans les systèmes informatiques et en identifiant les activités malveillantes à partir des événements observés.

- $P = [P_{i,j}]$ est la matrice de probabilité de transition d'état et d'observation Markovienne (dépend uniquement de l'état courant) dans laquelle $Pr(s^j,\ o|s^i,\ a_1, a_2)$ représente la probabilité de transition à l'état s^j et l'observation o de l'état s^i dans le cadre des actions (a_1, a_2).

- R = {R1, R2} la fonction d'utilité des deux joueurs, telle que R1 + R2 = 0, c'est-à-dire que les gains du premier joueur représentent les pertes du deuxième et vice versa. On a R1 : S × A → R, où S et A sont respectivement l'ensemble des stratégies et l'ensemble des actions disponibles pour les joueurs, et R est l'ensemble des réels.

Le but du défenseur dans un jeu stochastique partiellement observable est de maximiser la somme attendue des récompenses actualisées qui peut être obtenue à partir d'un état s en suivant une politique donnée. Cela peut être défini à l'aide de l'équation de Bellman pour un processus de décision partiellement observable (POMDP).

Vu que dans les POSGs, la valeur d'un état est cachée, le but du défenseur est de maximiser la valeur de chaque croyance qu'il a pour chaque état $s \in S$. De plus, le jeu étant fini, cela montre qu'il existe au moins un équilibre de Nash.

Le problème de POSG est un problème difficile qui peut être résolu à l'aide de différentes techniques telles que les algorithmes existants pour les processus de décision de Markov partiellement observables. Dans ce contexte, A. Hansen et al. (2004) ont proposé une approche de résolution de problème POSG qui consiste à utiliser des opérateurs de programmation dynamique pour résoudre le POMDP [4]. L'agent utilise des croyances d'état pour modéliser son incertitude. Effectivement, dans la formulation proposée, chaque joueur a son propre espace d'état de croyance, qui est une distribution de probabilité définie sur ce dernier. En convertissant un POMDP en MDP(Marcov Decesion Process) [5] défini sur l'espace de croyances qui contient toutes les croyances possibles sur l'état courant en réduisant la dimensionnalité de l'espace état-croyance pour les deux joueurs. Cette approche combine l'opérateur de programmation dynamique avec l'élimination itérative des stratégies dominées, ce qui permet d'obtenir une politique optimale pour POMDP. En effet, la limitation du nombre d'actions pures pour chaque joueur et le nombre restreint de mouvements séquentiels réduisent la complexité du jeu. Il est souvent coûteux en termes de calculs pour chaque joueur de raisonner sur tous les états possibles. De plus, chaque joueur doit modéliser récursivement les croyances des autres joueurs dans la solution de jeu.

3.3.3 Cost-Aware Securing of IoT Systems Using Attack Graphs

Yagit et al. (2018) [31] ont présenté une méthode pour le renforcement de la sécurité dans les systèmes IoT en utilisant des graphes d'attaques et en tenant compte le budget associé à ces

4. Il est utilisé dans le domaine d'apprentissage par renforcement où l'agent de prise de décision ne dispose pas d'informations complètes sur l'état actuel de l'environnement.

5. Il s'agit d'un modèle mathématique qui décrit la prise de décision séquentielle dans des environnements stochastiques.

systèmes. Vu que les systèmes IoT peuvent engendrer un très grand nombre de dispositifs ce qui rends leurs analyse difficile (compliquer) avec l'utilisation des graphes d'attaques normales ; pour cela les auteurs ont opté pour les graphes d'attaques compact pour réaliser leur approche.

Les graphes d'attaques compact sont des graphes dirigés qui contiennent des nœuds représentant des vulnérabilités appelé exploit et la condition de sécurité qui peut être de deux types différents à savoir les conditions initiales (préconditions) qui doivent être remplie avant l'exécution d'un exploit et les conditions intermédiaires (sont à la fois les préconditions et les postconditions de certains exploits) qui doivent être remplie durant l'exploitation d'un exploit. Ces nœuds sont reliés par des arêtes dirigés qui représentent des relations d'exigence et d'implicite entre les exploits et les conditions de sécurité. Les graphes d'attaques compact sont utilisés pour modéliser les attaques multi-étapes et évaluer le niveau de sécurité de manière quantitative.

Les graphes d'attaques montrent tous les chemins d'attaques qui peuvent être utilisés par un attaquant, pour cela il existe deux façons pour les éliminer à savoir : La désactivation de l'une des conditions de sécurité du chemin d'attaque. Corriger l'un des exploits du chemin d'attaque. Cela ne peut pas être suffisant dans certains cas où il y a des contraintes matérielles ou logicielles qui empêche la correction complète de la vulnérabilité.

Yagit et al [31]. ont utilisé le COBANOT qui représente une approche permettant de minimiser le coût de la sécurisation des systèmes IoT en utilisant les graphes d'attaques compact. Il se compose de deux phases. Dans la première phase, le but est d'extraire tous les chemins d'attaque (de l'attaquant vers la cible) possible à partir de graphe d'attaques en utilisant un algorithme à rebours et d'éviter tous les chemins cycliques pour garantir l'extraction du chemin d'attaque, et calculer la probabilité de succès pour chemin d'attaque trouvé. Une métrique de sécurité (SM) est obtenue en considérant les probabilités de succès de tous les chemins. La deuxième phase consiste à calculer les contributions de chaque exploit et les conditions initiale du chemin d'attaque, en prenant en compte le cout de suppression et la contribution ; de coup certains sont identifié comme des menaces potentielles qui sont ensuite éliminer et la métrique de sécurité est recalculer. Ce processus se poursuit jusqu'un ce que le réseau soit entièrement sécuriser.

Finalement, la complexité de COBANOT dépend de nombre de nœuds, les arêtes de graphe d'attaques, le nombre de chemin d'attaques, le nombre d'exploit et les conditions initiales et la contribution moyenne de chaque exploit.

3.3.4 Vulnerability association evaluation of Internet of thing devices based on attack graph

Ma et al. (2022) [32] ont présenté une méthode d'évaluation de la sécurité des dispositifs de l'Internet des objets basé sur les graphes d'attaques. L'approche est subdivisée en quatre étapes, comme suit :

1. Construction et génération de graphe d'attaques.
2. Evaluation des associations de vulnérabilités.

3. Utilisation de HMM(Hidden Markov Model) pour modéliser les états de vulnérabilités.
4. Evaluation des risques et recommandations de sécurités.

La méthode de modélisation de Markov caché et d'évaluation des vulnérabilités proposée dans cet article vise à évaluer les vulnérabilités des dispositifs IoT en utilisant un graphe d'attaques. Les auteurs ont construit un environnement de topologie de réseau qui comprend trois sous-réseaux interconnectés par des routeurs. Le réseau comprend six hôtes numérotés H1-H6, cinq caméras numérotées C1-C5, trois imprimantes numérotées P1-P3, le routeur R et le serveur S. Les différents sous-réseaux sont protégés par des pares-feux pour empêcher les attaquants d'attaquer directement les dispositifs cibles, mais ils doivent utiliser des vulnérabilités multiples sur différents dispositifs du système ou des attaques en plusieurs étapes sur des dispositifs pour attaquer l'environnement réseau. La Figure 3.10 illustre la topologie du réseau.

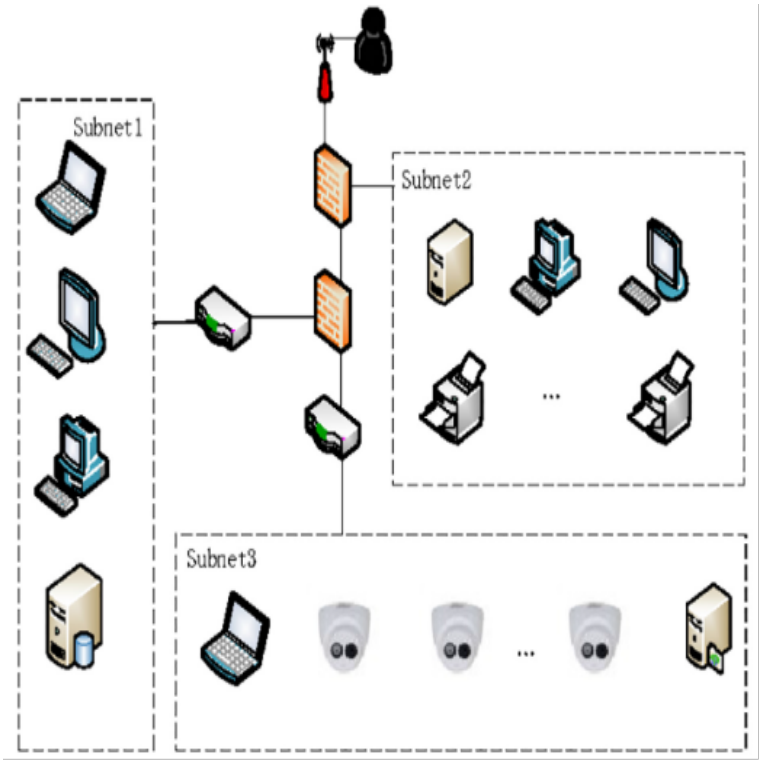


FIGURE 3.10 – Topologie de réseau [32].

1. Construction et génération de graphe d'attaques

Dans le graphe d'attaques traditionnel, les nœuds représentent les privilèges obtenus par l'attaquants et les arrêtes représentes les vulnérabilités logicielles utilisés par l'attaquant pour atteindre son objectif ; ceci peut lui permettre d'obtenir l'accès à d'autre dispositifs. Par ailleurs, dans cet article, les auteurs ont défini les nœuds de graphe d'attaques comme étant l'ensemble des dispositifs et les vulnérabilités liées à ce dernier ; et les arêtes représentent le processus d'attaque. De plus, pour la génération de graphes d'attaques, les auteurs ont défini tous d'abord l'ensemble

des vulnérabilités de chaque dispositifs IoT et les différents chemins qu'un attaquant peut utiliser pour atteindre ses objectifs, donc le graphe d'attaque est généré à partir des informations pertinentes sur les vulnérabilités dans l'équipement et la corrélation entre les dispositifs ainsi que la probabilité de réussite d'une attaque et ces conséquences. Le graphe d'attaque est parcouru à partir de l'état cible, si ceci est réussi donc il existe des vulnérabilités dans le système. La Figure 3.11 illustre le graphe d'attaques obtenu :

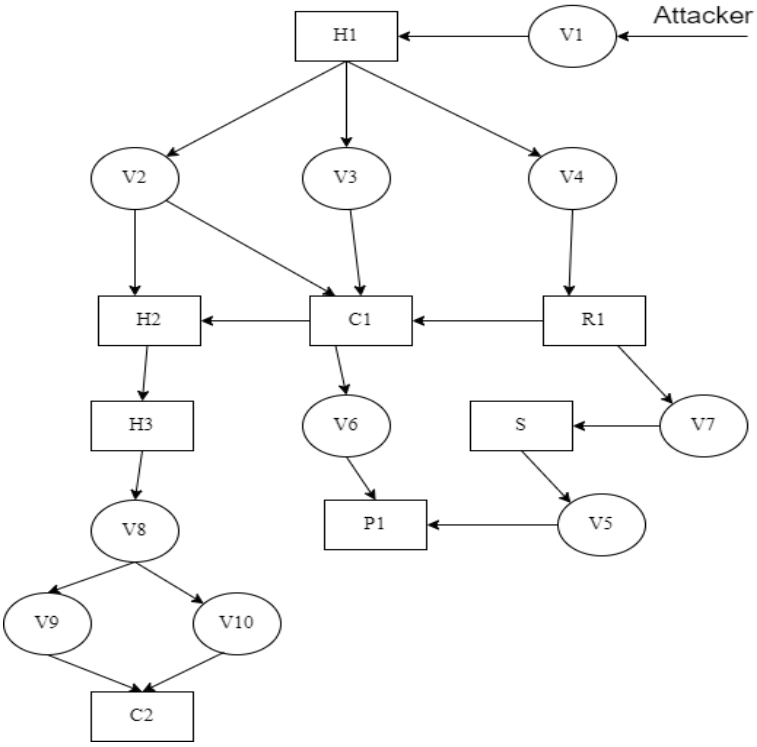


FIGURE 3.11 – Graphe d'attaques des vulnérabilités des dispositifs [32].

Ce diagramme représente les différentes étapes d'un attaquant qui pourraient être lancées contre un dispositif IoT, ainsi que les chemins possibles. Les nœuds représentent les vulnérabilités, les points d'entrée et les actions de l'attaquant, tandis que les liens représentent les relations de causalité entre ces étapes.

2.Evaluation des associations de vulnérabilités

Les vulnérabilités dans les dispositifs IoT peuvent permettre aux attaquants d'obtenir des privilèges supplémentaires de contourner les mesures de sécurité et accéder aux systèmes cibles. Pour évaluer ces vulnérabilités, les auteurs ont utilisé une approche qui consiste à construire l'environnement topologique de réseau afin d'obtenir la relation d'accessibilité entre les dispositifs. Ensuite, l'outil d'analyse de vulnérabilité Nessus [32] est utilisé pour scanner chaque nœud de dispositif dans le réseau cible et obtenir les informations de vulnérabilité sur le dispositif.

3. Utilisation de HMM pour modéliser les états de vulnérabilités

De nombreuses méthodes d'évaluation ne peuvent pas décrire qualitativement les attaques à plusieurs niveaux et la relation entre les comportements d'attaque, tel que la méthode CVSS(Common Vulnerability Scoring System) qui évalue les vulnérabilités et leurs degrés de risque par groupe d'attributs pour les montrer sous de nombreux aspects. Cependant, il ne peut évaluer une vulnérabilité que de manière isolée, sans prendre en compte le chemin de l'attaque, la séquence de l'attaque et la relation entre les nœuds de vulnérabilité avant et après l'attaque. Pour cela, les auteurs envisagent d'utiliser le HMM pour résoudre ce problème. Le HMM est un modèle probabiliste qui est utilisé pour modéliser la relation entre les états du dispositif après avoir obtenu la relation d'association entre les nœuds du dispositif, c'est-à-dire que la transition d'état dans l'environnement du réseau peut être réalisée par HMM. Les états dans le HMM sont cachés et inobservables, ce qui signifie que l'on ne peut pas observer directement les états mais seulement les symboles qui sont produits par ces états. L'état caché a deux probabilités : la probabilité de transition entre les états cachés et la probabilité de production des symboles observés pour chaque état caché.

4. Evaluation des risques et recommandations de sécurité

Cet article propose de manière créative l'évaluation de la corrélation de vulnérabilités des dispositifs Iot basé sur le graphe d'attaques. Voici les principaux résultats et conclusion de l'étude :

- Les auteurs ont pu identifier l'ensemble des vulnérabilités potentielles en fonction de la corrélation entre les équipements IoT.

- Le graphe d'attaques est généré par MulVAL, il montre les différents chemins d'attaques possibles pour chaque dispositif IoT en tenant compte la probabilité d'exploitation et la gravité des conséquences.

- La disponibilité et les conséquences de l'attaque des vulnérabilités sont analysés pour obtenir la valeur de risque des failles dans les dispositifs IoT.

- Les auteurs ont proposé l'utilisation de HMM pour représenter plus précisément la relation entre les équipements Iot et évalué efficacement le risque de sécurité des dispositifs dans les différents scénarios d'attaques. Cette méthode permet de mieux comprendre la complexité des relations entre les dispositifs de l'IoT.

Ma et al. [32]ont proposé plusieurs recommandations pour améliorer la sécurité des dispositifs IoT.

3.3.5 The Internet of Things Network Penetration Testing Model Using Attack Graph Analysis

Almazrouei et Magalingam (2022) [33] ont présenté une nouvelle approche qui concentre principalement sur l'utilisation de l'analyse des graphes d'attaques pour tester la sécurité des réseaux IoT.Dans cet article, les auteurs ont développé un modèle de test de pénétration, ils ont proposé deux algorithmes dont le premier consiste à générer tous les chemins d'attaques pour les dispositifs IoT en commençant par la collecte des données à partir de la topologie du réseau, la base de données des nœuds et des vulnérabilités et déterminer l'ensemble de vulnérabilités et les points d'accès potentiels associer à ce dernier. En utilisant la base de données communes des vulnérabilités (CVSS) les auteurs ont peu d'identifier le niveau de sécurité de chaque appareil ainsi qu'ils ont déterminer les chemins d'attaques critiques et les nœuds vitaux cela été décrit dans le deuxième algorithme.

Afin de montrer la flexibilité de leur modèle, Almazrouei et Magalingam [33].ont pris un exemple sur un réseau IoT composé de trois appareils qui sont vulnérables et ils ont pu de détecter les vulnérabilités zero-day dans ces derniers, les résultats ont également montré que le modèle était capable d'identifier les nœuds critique et les chemins d'attaques.

3.3.6 Attack Graph Generation with Machine Learning for Network Security

Moon et al. (2022) [34] ont proposé une méthode basée sur l'apprentissage automatique pour générer les graphes d'attaques afin de sécuriser, de renforcer la sécurité des réseaux et donne aux professionnels de sécurité la possibilité de prendre des mesures préventives pour protéger le réseau contre les diverses attaques. Les graphes d'attaques sont un moyen efficace de modélisation des risques de sécurité pour un réseau informatique en identifiants les vulnérabilités et les chemins d'attaques potentiels. Cependant, générer un graphe d'attaques nécessite une base de données de vulnérabilités et une grande capacité de calculs pour tous les nœuds connecter au réseau (la corrélation entre les nœuds) ce qui est couteux et laborieux, cela a conduit à transformer le problème de génération de ces derniers comme un problème d'apprentissage automatique à sorties multiples et de classification binaires qui est une méthode d'extraction de caractéristiques et d'expression pour apprendre un modèle de génération de graphes d'attaque à l'aide d'informations sur la topologie du réseau, le système et le chemin d'attaque.

Ils ont opté pour la classification binaire afin d'évaluer la probabilité de réussite d'une attaque entre chaque paire de nœuds afin de déterminer l'existence d'une vulnérabilité. En autre, Les auteurs ont également utilisé la classification multi-sortie pour une analyse plus fine des risques potentiels dans le réseau. Les modèles d'apprentissage automatique et d'apprentissage profond sont employés pour résoudre ces problèmes afin de les entrainer pour prédire le chemin d'attaque en utilisant le graphe d'attaques générer à partir d'une base de données des vulnérabilités.

3.4 Classification des travaux antérieurs

Afin de comprendre les différents articles étudiés dans ce chapitre en traitant le problème d'analyse des graphes d'attaques , le diagramme 3.12 représente une classification réalisée sur les divers méthodes d'analyse. Ce diagramme met en évidence les différentes approches utilisées dans la littérature pour détecter et prévenir les attaques informatiques basées sur les graphes d'attaques.

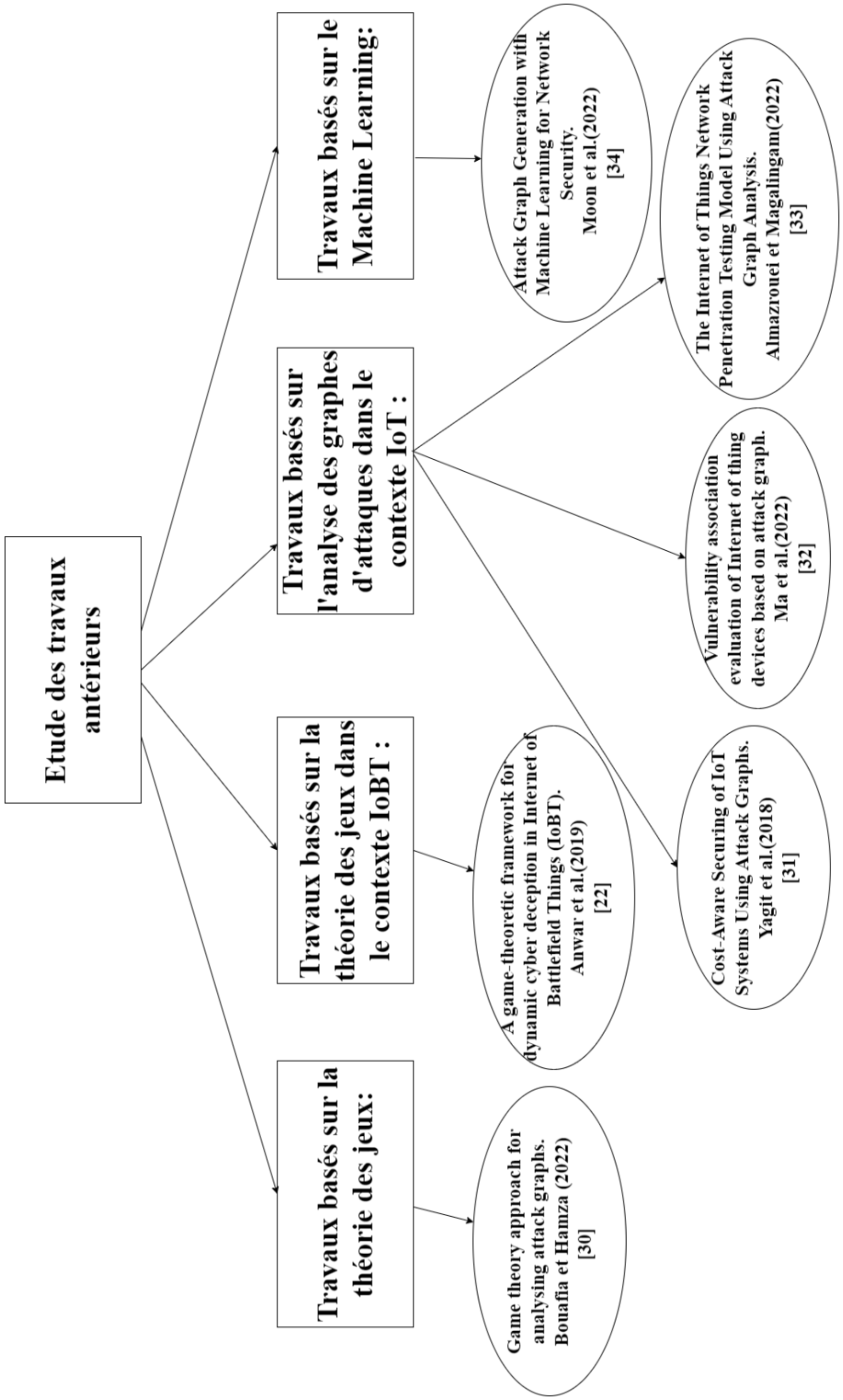


FIGURE 3.12 – Diagramme de classification des travaux antérieurs étudiés.

3.5 Conclusion

Ce chapitre a examiné quelques travaux existants dans la littérature sur l'analyse des graphes d'attaques. Les différents modèles présentés ont permis d'obtenir un nouveau point de vue sur la problématique de l'analyse des graphes d'attaques, de mieux appréhender son importance ainsi que les méthodes pour les traiter et les résoudre. Dans le prochain chapitre, nous présenterons notre approche.

Chapitre 4

Attack Graph Analysis by Partially Observable Stochastic Game (AGA-POSG)

4.1 Introduction

La sécurisation des réseaux IoT présente des défis spécifiques en raison de la nature complexe et interconnectée des dispositifs IoT. Les vulnérabilités et les risques de sécurité sont exacerbés dans un environnement IoT en raison du grand nombre de périphériques connectés et des échanges continus de données.

Dans ce contexte, notre proposition, Attack Graph Analysis by Partially Obesrvable Stochastic Game (AGA-POSG), offre une approche adaptée à la sécurisation des réseaux IoT. Cette approche nous a permis de modéliser les interactions entre les acteurs du réseau IoT, tels que les périphériques, les utilisateurs et les attaquants potentiels. Elle permet à l'administrateur de prendre des décisions éclairées pour renforcer la sécurité du réseau IoT.

4.2 Démarche proposée

Notre proposition vise à permettre à l'administrateur de prendre connaissance des différentes vulnérabilités menaçantes son système et de choisir la méthode de correction qui convient le mieux, en fonction de ses contraintes de coût. Cette solution est adaptée aux besoins spécifiques des réseaux IoT.

La technique que nous avons développée repose sur l'utilisation d'un graphe d'attaques qui représente tous les chemins possibles qu'un attaquant peut emprunter pour atteindre sa cible. Notre approche consiste à analyser ce graphe d'attaques afin de minimiser les pertes subies par l'administrateur et de l'aider à prendre des décisions éclairées pour renforcer la sécurité de son réseau.

Considérant un réseau IoT, un intrus cherche à infiltrer le réseau dans le but de perturber certaines fonctionnalités et causer le plus de dommages possible pour satisfaire ses propres besoins.

Cette perturbation est réalisée à travers une série d'attaques. De son côté l'administrateur du réseau tente de minimiser les pertes et de maximiser les gains en réagissant aux attaques lancées par l'intrus. L'objectif de l'administrateur est donc de protéger le réseau en prenant des mesures pour contrer les attaques et réduire les dommages potentiels.

La situation se modélise sous forme d'un jeu stochastique partiellement observable, non coopératif à deux joueurs (attaquant et administrateur) à somme nulle et fini. Notre jeu englobe :

- Le jeu stochastique partiellement observable permet de modéliser des scénarios où l'information disponible est limitée et où l'incertitude est présente c'est à dire les agents ne peuvent pas avoir une vue complète de l'état du système à tout moment, ainsi que les dispositifs de l'IoT peuvent augmenter ou diminuer au fur et à mesure. De plus, l'incertitude est une caractéristique inhérente aux environnements de cyberdéfense. Les défenseurs ne disposent souvent que d'informations partielles sur les actions et les intentions des attaquants et les résultats des actions peuvent être soumis à des facteurs aléatoires. Les jeux stochastiques partiellement observables permettent aux défenseurs de prendre en compte cette incertitude et de modéliser les différentes possibilités d'actions des attaquants et les résultats correspondants.
- Le non-coopératif due à la contradiction entre les deux joueurs un attaque et l'autre défend.
- Deux joueurs car il y a deux individus un administrateur de réseau IoT et un attaquant.
- A somme nulle, car les gains d'un joueur représentent les pertes de l'autre joueur.
- Le jeu est fini car les stratégies des deux joueurs sont finies.

4.3 AGA-POSG par l'exemple

Pour la bonne compréhension de notre proposition, nous appliquons notre démarche sur l'exemple de la Figure 4.1 qui présente un réseau IoT composé d'un pare-feu et un routeur qui sépare le sous réseau 1 du sous réseau 2, le sous réseau1 contient une caméra de surveillance, un capteur 1 et une tablette. Par ailleurs, le sous réseau 2 contient une imprimante, un téléphone et un capteur 2. Nous supposons que le but de l'intrus est d'accéder aux informations sensibles de la tablette.

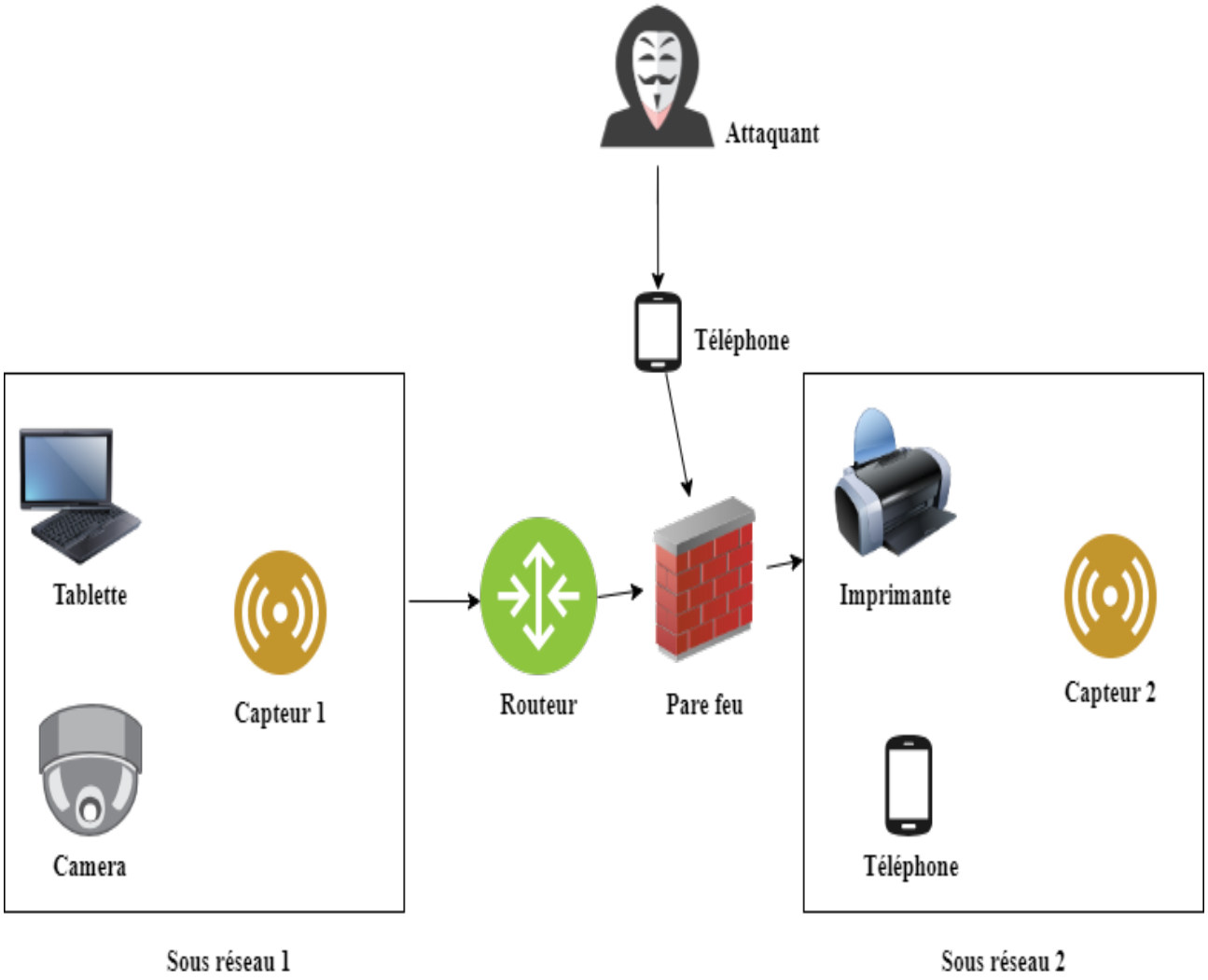


FIGURE 4.1 – Exemple d'un réseau IoT .

A l'aide de la base de données des vulnérabilités NVD(National Vulnerability Database) [38] et CVE(Common Vulnerabilities and Exposures) [39], nous avons obtenu le tableau 4.1 qui représente les dispositifs, leurs identifiant, le niveau de risque associer à chaque vulnérabilité d'un dispositif et le nom de leurs vulnérabilités.

Dispositif	CVE-ID	Nom de la vulnérabilité	Niveau de risque	Type de vulnérabilité
IP Caméra (V1)	CVE-2017-10796	TP-Link NC 250 vulnérabilité de sécurité	faible	problème d'autorisation
Capteur 1 (V3)	CVE-2002-0237	ISS BLACK ICE DEFENDER,BLACKICE AGENT, REAL SECURE SERVE SENSOR BUFFER OVERFLOW.	élevé	Débordement de tompon
Capteur 2 (V2)	CVE-2018-14890	Vectra Networks Cognito Brain and Sensor before 4.2 contains a cross-site scripting (XSS)	faible	cross-site scripting (XSS)
Routeur (V4)	CVE-2017-14415	D-Link DIR-850L REV A XSS.	moyen	cross-site scripting (XSS)
Tablette (V5)	CVE-2018-18784	ZZcms 8.3 SQL injection.	moyen	SQL Injection
Imprimante (V6)	CVE-2002-0237	Cyber Ark Viewfinity élévation de privélège via l'option " ajouter une imprimante "	élevé	élévation de privélège
Téléphone (V7)	CVE-2022-47480	Autorisation manquante.	moyen	Vulnérabilité de de type déni service DOS

TABLE 4.1 – Dispositifs, vulnérabilités et niveaux de risque de l'exemple étudié.

4.3.1 Modélisation

Notre modèle est défini comme suit :

- **Les joueurs**

  L'ensemble des joueurs est N = {Attaquant ; Administrateur}.

- **Les actions**

  L'ensemble des actions (stratégies) $A = A_1 \times A_2$, où A_1 représente les stratégies de l'attaquant et A_2 représente les stratégies de l'administrateur.

  Les actions de l'intrus sont :

A_1= {Débordement de tampon, Cross-Site Scripting (XSS), Injection SQL, Problème d'autorisation, Élévation de privilège, Déni de service (DoS)}

Les définitions des actions de l'intrus sont donnés comme suit :

- **Débordement de tampon :** un attaquant peut exploiter cette vulnérabilité pour exécuter du code malveillant sur le système cible, prendre le contrôle du système ou causer un plantage du système.

- **Cross-Site Scripting (XSS) :** un attaquant peut exploiter cette vulnérabilité pour injecter du code malveillant dans une page web et l'exécuter sur le navigateur de la victime. Cela peut lui permettre de voler des informations sensibles de la victime ou de prendre le contrôle de son compte.

- **Injection SQL :** un attaquant peut exploiter cette vulnérabilité pour accéder à des informations confidentielles dans une base de données, modifier des données ou exécuter du code malveillant sur le système cible.

- **Problème d'autorisation :** un attaquant peut exploiter cette vulnérabilité pour accéder à des ressources ou des fonctionnalités du système auxquelles il n'est pas autorisé. Cela peut lui permettre d'accéder à des informations sensibles ou de prendre le contrôle du système.

- **Élévation de privilège :** un attaquant peut exploiter cette vulnérabilité pour obtenir des privilèges supplémentaires sur le système cible. Cela peut lui permettre d'accéder à des informations sensibles ou de prendre le contrôle du système.

- **Déni de service (DoS) :** un attaquant peut exploiter cette vulnérabilité pour perturber le fonctionnement normal du système cible en le surchargeant de trafic ou en utilisant des techniques d'épuisement de ressources.

Par ailleurs, les actions de l'administrateur sont :

A_2 = {Generate Alarm, IP Blocking, Isolate Host, Kill Process, No defence}

Les définitions des actions de l'administrateur sont données comme suit :

- **Generate Alarm :** Cette action peut être utilisée pour générer une alerte lorsque des tentatives d'attaque sont détectées sur le réseau IoT. Cela permettra aux défenseurs de prendre des mesures de protection avant que l'attaque ne cause des dommages graves.

- **IP Blocking :** Cette action permet de bloquer les adresses IP suspectes ou connues pour être utilisées par des attaquants. En bloquant ces adresses IP, les attaquants ne pourront pas se connecter aux appareils IoT vulnérables.

- **Isolate Host :** Cette action permet d'isoler les appareils vulnérables du reste du réseau, de sorte que les attaquants ne peuvent pas accéder à ces appareils depuis le réseau.

- **kill Process :** Cette action peut être utilisée pour arrêter les processus malveillants qui tentent d'exploiter les vulnérabilités sur les appareils IoT. Cela permettra d'empêcher les attaquants d'atteindre leur objectif.

- **No defence :** Cette action n'est pas recommandée car elle signifie qu'aucune mesure de sécurité n'est prise pour protéger le réseau IoT.

- **Les états**

  Un ensemble "S" décrit l'état du système à chaque stratégie jouée que ce soit par l'attaquant ou par le défenseur. Dans notre cas, chaque stratégie jouer par les deux joueurs, les états possibles sont :

  S = $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$où :

s_1 : état initial.
s_2 : attaquant réussit à exploiter la vulnérabilité d'autorisation sur la caméra.
s_3 : attaquant réussit à exploiter la vulnérabilité XSS sur le capteur 2.
s_4 : attaquant réussit à exploiter la vulnérabilité de débordement de tampon sur le capteur 1.
s_5 : attaquant réussit à exploiter la vulnérabilité XSS du routeur.
s_6 : attaquant réussit à effectuer une injection SQL sur la tablette.
s_7 : attaquant réussit l'élévation de privilège sur l'imprimante.
s_8 : attaquant réussit à effectuer une attaque de déni de service (DoS) sur le téléphone.

- **Les observations**

  À l'instant $t = 0$, $O = O_1 \times O_2 = \emptyset$.

  À l'instant t, lorsque l'attaquant réussit une attaque, O_2 devient un ensemble vide et O_1 devient un ensemble contenant l'élément "attaque réussie". En revanche, dans le cas où l'attaque est détectée, cela implique que O_1 devient un ensemble vide et O_2 devient un ensemble contenant "tentative d'attaque détectée".

- **Probabilités de transition**

  La probabilité de transition peut dépendre de plusieurs facteurs, tels que l'état actuel, les actions prises, les informations disponibles comme les vulnérabilités existantes dans chaque dispositif dans notre cas nous allons s'intéresser au niveau de risque de chaque vulnérabilité citer dans le tableau 4.1. La Figure 4.2 représente les différentes transitions d'états.

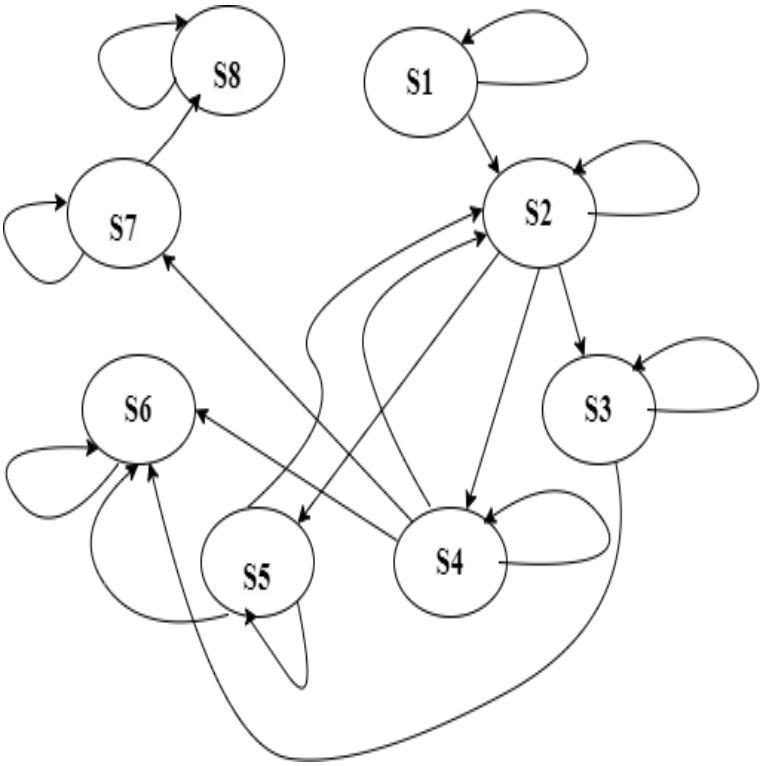


FIGURE 4.2 – Grpahe d'états-transitions.

Nous citons ci-dessous deux exemples de probabilité de transition :

- T(s_1, Problème d'autorisation, No Defence,O_1, O_2, s_2)= 0.2

Cela signifie que la probabilité de passer de l'état s_1 à l'état s_2 en l'absence de défense contre le problème d'autorisation est de 0.2.

- T(s_1, Problème d'autorisation, Isolate Host, O_1,O_2,s_1)= 0.8

Cela signifie que la probabilité de rester dans l'état s_1 après avoir isolé l'hôte en réponse au problème d'autorisation est de 0.8 (Pour n'importe quelle action de l'administrateur sauf No Defence).

- **Fonction de récompense**

$$R_1(s, a_1, a_2) = \alpha_{1\{I_s(a_1)=\text{"accès à un dispositif"}\}} - \eta_{1\{I_s(a_1)\neq\text{"accès à un dispositif"}\}} - C_1(s, a_1) + C_2(s, a_2) \quad (4.1)$$

α_1 : Coefficient de récompense pour l'attaquant lorsque l'accès à un dispositif est réussi.

η_1 : Coefficient de pénalité pour l'attaquant lorsque l'accès à un dispositif est échoué.

I(condition) : Fonction indicatrice qui renvoie 1 si la condition est vraie et 0 sinon.

a_1 : Action de l'attaquant (joueur 1).

a_2 : Action de l'administrateur (joueur 2).

$C_1(s, a_1)$: Coût associé à l'action de l'attaquant (joueur 1).

$C_2(s, a_2)$: Coût associé à l'action de l'administrateur (joueur 2).

D'où R_1 représente la fonction de récompense de l'attaquant et R_2 est la fonction de récompense de l'administrateur. Avec : $R_1 = -R_2$.

$\forall\ s\ \in\ S$ et s = $s_1...s_8$, les actions de l'intrus utilisées contre ces attaques et les coûts associés sont représentés dans le tableau 4.2 :

(État, Stratégies d'attaque)	Coûts
(s, Débordement de tampon)	10
(s, Cross-Site Scripting (XSS))	5
(s, Injection SQL)	9
(s, Problème d'autorisation)	6
(s, Élévation de privilège)	8
(s, Déni de service (DoS))	7

TABLE 4.2 – Tableau des stratégies d'attaques.

$\forall\ s\ \in\ S$ et s = $s_1...s_8$, les coûts associés à ces actions sont représentés dans le tableau 4.3.

(État, Stratégies de défense)	Coûts
(s,Generate Alarm)	8
(s,IP Blocking)	5
(s,Isolate Host)	10
(s,Kill Process)	4
(s,No defence)	0

TABLE 4.3 – Tableau des stratégies de défense.

Pour calculer R, nous utilisons la formule de Anwar et al. [23] qui correspond à 4.1.

Nous substituons les valeurs données dans la formule pour chaque combinaison d'état (s) et d'actions (a_1, a_2). Voici le calcul de R pour chaque cas :

Débordement de tampon :

R_1(s, Débordement de tampon, Generate Alarm) = 0.8 * 0 - 0.2 * 1 +10 - 8 = 1.8

R_1 (s, Débordement de tampon, IP Blocking) = 0.8 * 0 -0.2 * 1 + 10 - 5 = 4.8

R_1(s, Débordement de tampon, Isolate Host) = 0.8 *0 - 0.2 * 1 +10 - 10 = -0.2

R_1(s, Débordement de tampon, Kill Process) = 0.8 * 0 - 0.2 * 1 + 10 - 4 = 5.8

R_1(s, Débordement de tampon, No defence) = 0.8 * 1 - 0.2 * 0 + 10 - 0 = 10.8

XSS :

R_1 (s, Cross-Site Scripting (XSS), Generate Alarm) = 0.8 * 0-0.2 * 1+ 5 - 8 =-3.2

R_1 (s, Cross-Site Scripting (XSS), IP Blocking) = 0.8 * 0- 0.2 * 1+ 5 - 5 = -0.2
R_1 (s, Cross-Site Scripting (XSS), Isolate Host) = 0.8 * 0 - 0.2 * 1 + 5 - 10 = -5.2
R_1 (s, Cross-Site Scripting (XSS), Kill Process) = 0.8 * 0 - 0.2 * 1 + 5 - 4 = 0.8
R_1 (s, Cross-Site Scripting (XSS), No defence) = 0.8 * 1 - 0.2 * 0 + 5 - 0 = 5.8

Injection SQL :

R_1 (s, Injection SQL, Generate Alarm) = 0.8 * 0 - 0.2 * 1 + 9 - 8 = 0.8
R_1(s, Injection SQL, IP Blocking) = 0.8 * 0 - 0.2 * 1 + 9 - 5 = 3.8
R_1(s, Injection SQL, Isolate Host) = 0.8 * 0 - 0.2 * 1 + 9 - 10 = -1.2
R_1 (s, Injection SQL, Kill Process) = 0.8 * 0 - 0.2 * 1 + 9 - 4 = 4.8
R_1(s, Injection SQL, No defence) = 0.8 * 1 - 0.2 * 0 + 9 - 0 = 9.8

Problème d'autorisation :

R_1 (s, Problème d'autorisation, Generate Alarm) = 0.8 * 0 -0.2 * 1 + 6 - 8 = -2.2
R_1 (s, Problème d'autorisation, IP Blocking) = 0.8 * 0 - 0.2 * 1 +6 - 5 = 0.8
R_1(s, Problème d'autorisation, Isolate Host) = 0.8 * 0 - 0.2 * 1 + 6 - 10 = -4.2
R_1 (s, Problème d'autorisation, Kill Process) = 0.8 * 0 - 0.2 * 1 +6 - 4 = 1.8
R_1(s, Problème d'autorisation, No defence) = 0.8 *1 - 0.2 * 0+ 6 - 0 = 6.8

Élévation de privilège :

R_1 (s, Élévation de privilège, Generate Alarm) = 0.8 * 0- 0.2 * 1 +8 - 8 = -0.2
R_1 (s, Élévation de privilège, IP Blocking) = 0.8 * 0 - 0.2 * 1 + 8 - 5 = 2.8
R_1 (s, Élévation de privilège, Isolate Host) = 0.8 * 0 - 0.2 * 1 +8- 10 = -2.2
R_1 (s, Élévation de privilège, Kill Process) = 0.8 * 0 - 0.2 * 1 + 8 - 4 = 3.8
R_1 (s, Élévation de privilège, No defence) = 0.8 * 1- 0.2 * 0 + 8- 0 =8.8

Déni de service (DoS) :

R_1(s, Déni de service (DoS), Generate Alarm) = 0.8 * 0- 0.2 * 1 +7 - 8 = -1.2
R_1 (s, Déni de service (DoS), IP Blocking) = 0.8 * 0 - 0.2 * 1 + 7 - 5 = 1.8
R_1 (s, Déni de service (DoS), Isolate Host) = 0.8 * 0 - 0.2 * 1 +7 - 10 = -3.2
R_1 (s, Déni de service (DoS), Kill Process) = 0.8 * 0 - 0.2 * 1 + 7 - 4 = -2.8
R_1(s, Déni de service (DoS), No defence) = 0.8 * 1- 0.2 * 0 + 7 - 0 = 7.8

4.3.2 Représentation de jeu sous forme normale

Après avoir connu les stratégies de chaque joueur et les coûts associés, la forme normale correspondante au jeu est représentée dans le tableau 4.4 :

Attaquant	Administrateur				
	Generate Alarm	IP Blocking	Isolate Host	Kill Process	No defence
Débordemant de tampon	(1.8, -1.8)	(4.8, -4.8)	(-0.2, 0.2)	(5.8, -5.8)	(10.8, -10.8)
Cross-Site Scripting (XSS)	(-3.2, 3.2)	(-0.2, 0.2)	(-5.2, 5.2)	(0.8, -0.8)	(5.8, -5.8)
Injection SQL	(0.8, -0.8)	(3.8, -3.8)	(-1.2, 1.2)	(4.8, -4.8)	(9.8, -9.8)
Problème d'autorisation	(-2.2, 2.2)	(0.8, -0.8)	(-4.2, 4.2)	(1.8, -1.8)	(6.8, -6.8)
Élévation de privilège	(-0.2, 0.2)	(2.8, -2.8)	(-2.2, 2.2)	(3.8, -3.8)	(8.8, -8.8)
Déni de service (DoS)	(-1.2, 1.2)	(1.8, -1.8)	(-3.2, 3.2)	(2.8, -2.8)	(7.8, -7.8)

TABLE 4.4 – Forme normale du jeu.

4.3.3 Analyse du graphe d'attaques

Pour la phase d'analyse, nous appliquons l'algorithme 1 à notre exemple.

Algorithm 1 Algorithme d'élimination itérée des stratégies dominées

1: **Étape 1 :** $A_i^0 = A_i$;
2: **Étape 2 :** $A_i^1 = \{\text{stratégies non dominées}\}$, ensemble des stratégies non dominées.
3: **for** $K = 1$ to ∞ **do**
4: **Étape** $K+1 : A_i^{K+1} = s_i \in A_i^k, \nexists y_i \in A_i^k, \forall a_i, f_i(y_i, a_{-i}) > f_i(x_i, x_{-i})$
5: **end for**
6: **Étape** $\infty : A_i^\infty = \bigcap_k A_i^k$.

Une fois que nous avons terminé le processus itératif d'élimination des stratégies dominées, nous obtenons l'ensemble des stratégies non dominées. Les résultats du jeu qui font partie de l'ensemble $\prod_{i=1}^{N} X_i^\infty$ correspondent à des équilibres en stratégies non dominées.

4.3.4 Déroulement de l'algorithme d'élimination itérée des stratégies dominées

Étape 1 :

$A_1^0 =$ {Débordement de tampon, Cross-Site Scripting (XSS), Injection SQL, Problème d'autorisation, Élévation de privilège, Déni de service (DoS)}
$A_2^0 =$ {Generate Alarm, IP Blocking, Isolate Host, Kill Process, No defence}

Nous allons se concentrer sur les colonnes **Kill process** et **No defence** du tableau 4.4 :
- Si le joueur 1 (attaquant) joue Débouredement de tompon le joueur 2 (administrateur) à le choix entre un gain de (-5.8) et un gain de(-10.8).
- Si le joueur 1 joue XSS, le joueur 2 à le choix entre un gain de (-0.8) et un gain de (-5.8).
- Si le joueur 1 joue SQL, le Joueur 2 a le choix entre un gain de (-4.8) et un gain de (-9.8).
- Si le joueur 1 joue Problème d'autorisation, le Joueur 2 a le choix entre un gain de (-1.8) et un gain de (-6.8).
- Si le joueur 1 joue Elévation de privélège, le Joueur 2 a le choix entre un gain de (-3.8) et un gain de (-8.8).
- Si le joueur 1 joue DoS, le Joueur 2 a le choix entre un gain de (-2.8) et un gain de (-7.8).
- Puisque (-5.8)>(-10.8), (-0.8)>(-5.8), (-4.8)>(-9.8), (-1.8)>(-6.8), (-3.8)>(-8.8) et (-2.8)>(-7.8) alors nous pouvons supprimer la colonne **No defense**.

Ce qui nous donne le tableau 4.5 :

Attaquant	Administrateur			
	Generate Alarm	IP Blocking	Isolate Host	Kill Process
Débordemant de tampon	(1.8, -1.8)	(4.8, -4.8)	(-0.2, 0.2)	(5.8, -5.8)
Cross-Site Scripting (XSS)	(-3.2, 3.2)	(-0.2, 0.2)	(-5.2, 5.2)	(0.8, -0.8)
Injection SQL	(0.8, -0.8)	(3.8, -3.8)	(-1.2, 1.2)	(4.8, -4.8)
Problème d'autorisation	(-2.2, 2.2)	(0.8, -0.8)	(-4.2, 4.2)	(1.8, -1.8)
Élévation de privilège	(-0.2, 0.2)	(2.8, -2.8)	(-2.2, 2.2)	(3.8, -3.8)
Déni de service (DoS)	(-1.2, 1.2)	(1.8, -1.8)	(-3.2, 3.2)	(2.8, -2.8)

TABLE 4.5 – Forme normale obtenue après l'élimination de la stratégie *No defence*.

Étape 2 :

$A_1^1 =$ {Débordement de tampon, Cross-Site Scripting (XSS), Injection SQL, Problème d'autorisation, Élévation de privilège, Déni de service (DoS)}

$A_2^1 =$ {Generate Alarm, IP Blocking, Isolate Host, Kill Process}

Nous allons se concentrer sur les colonnes **Kill process** et **IP Blocking** du tableau 4.5.

- Si le joueur 1 (attaquant) joue Débouredement de tompon le joueur 2 (administrateur) à le choix entre un gain de (-4,8) et un gain de (-5,8).

- Si le joueur 1 joue XSS, le joueur 2 à le choix entre un gain de (0.2) et un gain de (-0.8).

- Si le joueur 1 joue SQL, le Joueur 2 a le choix entre un gain de (-3.8) et un gain de (-4.8).

- Si le joueur 1 joue Problème d'autorisation, le Joueur 2 a le choix entre un gain de (-0.8) et un gain de (-1.8).

- Si le joueur 1 joue Elévation de privélège, le Joueur 2 a le choix entre un gain de (-2.8) et un gain de (-3.8).

- Si le joueur 1 joue DoS, le Joueur 2 a le choix entre un gain de (-1.8) et un gain de (-2.8).

Puisque (-4.8)>(-5.8), (-0.2)>(-0.8), (-3.8)>(-4.8), (-0.8)>(-1.8), (-2.8)>(-3.8) et (-1.8)>(-2.8) alors nous pouvons supprimer la colonne **Kill Process**.

Ce qui nous donne le tableau 4.6 :

Attaquant	Administrateur		
	Generate Alarm	IP Blocking	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(4.8, -4.8)	(-0.2, 0.2)
Cross-Site Scripting (XSS)	(-3.2, 3.2)	(-0.2, 0.2)	(-5.2, 5.2)
Injection SQL	(0.8, -0.8)	(3.8, -3.8)	(-1.2, 1.2)
Problème d'autorisation	(-2.2, 2.2)	(0.8, -0.8)	(-4.2, 4.2)
Élévation de privilège	(-0.2, 0.2)	(2.8 , -2.8)	(-2.2, 2.2)
Déni de service (DoS)	(-1.2, 1.2)	(1.8, -1.8)	(-3.2, 3.2)

TABLE 4.6 – Forme normale obtenue après l'élimination de la stratégie *Kill Process.*

Etape 3 :

$A_1^2 =$ {Débordement de tampon, Cross-Site Scripting (XSS), Injection SQL, Problème d'autorisation, Élévation de privilège, Déni de service (DoS)}

$A_2^2 =$ {Generate Alarm, IP Blocking, Isolate Host}

Nous allons se concentrer sur les colonnes **Generate alarm** et **IP Blocking** du tableau 4.6.

- Si le joueur 1 (attaquant) joue Débouredement de tompon, le joueur 2 (administrateur) à le choix entre un gain de (-1,8) et un gain de (-4,8).

- Si le joueur 1 joue XSS, le joueur 2 à le choix entre un gain de (3.2) et un gain de (0.2).

- Si le joueur 1 joue SQL, le Joueur 2 a le choix entre un gain de (-0.8) et un gain de (-3.8).

- Si le joueur 1 joue Problème d'autorisation, le Joueur 2 a le choix entre un gain de (2.2) et un gain de (-0.8).

- Si le joueur 1 joue Elévation de privélège, le Joueur 2 a le choix entre un gain de (0.2) et un gain de (-2.8).

- Si le joueur 1 joue DoS, le Joueur 2 a le choix entre un gain de (1.2) et un gain de (-1.8).

Puisque (-1.8)>(-4.8), (3.2)>(0.2), (-0.8)>(-3.8), (2.2)>(-0.8), (0.2)>(-2.8) et (1.2)>(-1.8) alors nous pouvons supprimer la colonne **IP Blocking.**

Ce qui nous donne le tableau 4.7 :

Attaquant	Administrateur	
	Generate Alarm	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(-0.2, 0.2)
Cross-Site Scripting (XSS)	(-3.2, 3.2)	(-5.2, 5.2)
Injection SQL	(0.8, -0.8)	(-1.2, 1.2)
Problème d'autorisation	(-2.2, 2.2)	(-4.2, 4.2)
Élévation de privilège	(-0.2, 0.2)	(-2.2, 2.2)
Déni de service (DoS)	(-1.2, 1.2)	(-3.2, 3.2)

TABLE 4.7 – Forme normale obtenue après l'élimination de la stratégie *IP Blocking.*

Etape 4 :

$A_1^3 =$ {Débordement de tampon, Cross-Site Scripting (XSS), Injection SQL, Problème d'autorisation, Élévation de privilège, Déni de service (DoS)}

$A_2^3 =$ {Generate Alarm, Isolate Host}

Nous allons se concentrer sur les lignes **Cross-Site Scripting (XSS)** et **Problème d'autorisation** du tableau 4.7.

- Si le joueur 2 joue Generate Alarm , le joueur 1 a le choix entre un gain de (-3.2) et un gain de (-2.2).

- Si le joueur 2 joue Isolate Host, le joueur 1 a le choix entre un gain de (-5.2) et un gain de (-4.2).

Puisque (-2.2)>(-3.2), (-4.2)>(-5.2) alors nous pouvons supprimer la ligne **XSS**.

Ce qui nous donne le tableau 4.8 :

Attaquant	Administrateur	
	Generate Alarm	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(-0.2, 0.2)
Injection SQL	(0.8, -0.8)	(-1.2, 1.2)
Problème d'autorisation	(-2.2, 2.2)	(-4.2, 4.2)
Élévation de privilège	(-0.2, 0.2)	(-2.2, 2.2)
Déni de service (DoS)	(-1.2, 1.2)	(-3.2, 3.2)

TABLE 4.8 – Forme normale obtenue après l'élimination de la stratégie *XSS.*

Etape 5 :

$A_1^4 =$ {Débordement de tampon, Injection SQL,
Problème d'autorisation, Élévation de privilège, Déni de service (DoS)}

$A_2^4 =$ {Generate Alarm, Isolate Host}

Nous allons se concentrer sur les lignes **Problème d'autorisation** et **Déni de service (DoS)** du tableau 4.8.

- Si le joueur 2 joue Generate Alarm, le joueur 1 a le choix entre un gain de (-2.2) et un gain de (-1.2).

- Si le joueur 2 joue Isolate Host, le joueur 1 a le choix entre un gain de (-4.2) et un gain de (-3.2).

Puisque (-1.2)>(-2.2), (-3.2)>(-4.2) alors nous pouvons supprimer la ligne **Problème d'autorisation**.

Ce qui nous donne le tableau 4.9 :

Attaquant	Administrateur	
	Generate Alarm	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(-0.2, 0.2)
Injection SQL	(0.8, -0.8)	(-1.2, 1.2)
Élévation de privilège	(-0.2, 0.2)	(-2.2, 2.2)
Déni de service (DoS)	(-1.2, 1.2)	(-3.2, 3.2)

TABLE 4.9 – Forme normale obtenue après l'élimination de la stratégie *Problème d'autorisation.*

Etape 6 :

$A_1^5 =$ {Débordement de tampon, Injection SQL, Élévation de privilège, Déni de service (DoS)}
$A_2^5 =$ {Generate Alarm, Isolate Host}

Nous allons se concentrer sur les lignes **DoS** et **Elélévation de privilège** du tableau 4.9.
- Si le joueur 2 joue Generate Alarm, le joueur 1 a le choix entre un gain de (-1.2) et un gain de (-0.2).

- Si le joueur 2 joue Isolate Host, le joueur 1 a le choix entre un gain de (-3.2) et un gain de (-2.2).

Puisque (-0.2)>(-1.2) et (-2.2)>(-3.2) alors nous supprimons la ligne **DoS**.

Ce qui nous donne le tableau 4.10 :

Attaquant	Administrateur	
	Generate Alarm	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(-0.2, 0.2)
Injection SQL	(0.8, -0.8)	(-1.2, 1.2)
Élévation de privilège	(-0.2, 0.2)	(-2.2, 2.2)

TABLE 4.10 – Forme normale obtenue après l'élimination de la stratégie *DoS.*

Etape 7 :

$A_1^6 =$ {Débordement de tampon, Injection SQL, Élévation de privilège}
$A_2^6 =$ {Generate Alarm, Isolate Host}

Nous allons se concentrer sur les lignes **Elévation de privilège** et **Injection SQL** de tableau 4.10.
- Si le joueur 2 joue Generate Alarm, le joueur 1 a le choix entre un gain de (-0.2) et un gain de (0.8).

- Si le joueur 2 joue Isolate Host, le joueur 1 a le choix entre un gain de (-2.2) et un gain de (-1.2).

Puisque (0.8)>(-0.2), (-1.2)>(-2.2) alors nous pouvons supprimer la ligne **Elévation de privilège**.

Ce qui nous donne le tableau 4.11 :

Attaquant	Administrateur	
	Generate Alarm	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(-0.2, 0.2)
Injection SQL	(0.8, -0.8)	(-1.2, 1.2)

TABLE 4.11 – Forme normale obtenue après l'élimination de la stratégie *Elévation de privilège.*

Etape 8 :

$A_1^7 =$ {Débordement de tampon, Injection SQL}
$A_2^7 =$ {Generate Alarm, Isolate Host}

Nous allons se concentrer sur les lignes **Débordement de tampon** et **Injection SQL** du tableau 4.11.
- Si le joueur 2 joue Generate Alarm, le joueur 1 a le choix entre un gain de (1.8) et un gain de (0.8).

- Si le joueur 2 joue Isolate Host, le joueur 1 a le choix entre un gain de (-0.2) et un gain de (-1.2).

Puisque (1.8)>(0.8), (-0.2)>(-1.2) alors nous pouvons supprimer la ligne **Injection SQL**.

Ce qui nous donne le tableau 4.12 :

Attaquant	Administrateur	
	Generate Alarm	Isolate Host
Débordemant de tampon	(1.8, -1.8)	(-0.2 , 0.2)

TABLE 4.12 – Forme normale obtenue après élimination de la stratégie *Injection SQL* .

Etape 9 :

$A_1^8 =$ {Débordement de tampon}
$A_2^8 =$ {Generate Alarm, Isolate Host}

Nous allons se concentrer sur les colonnes **Generate Alarm** et **Isolate Host** du tableau 4.12.
- Si le joueur 1 (attaquant) joue Débouredement de tampon le joueur 2 (administrateur) à le choix entre un gain de (-1,8) et un gain de (0.2).

Puisque (-1.8)>(0.2) alors nous pouvons supprimer la colonne **Generate Alarm**.

Ce qui nous donne le tableau 4.13 :

Attaquant	Administrateur
	Isolate Host
Débordemant de tampon	(-0.2 , 0.2)

TABLE 4.13 – Forme normale obtenue après l'élimination de la stratégie *Generate Alarm*.

Ainsi, nous conclurons que le jeu a un équilibre de Nash qui est le profile de stratégies (Débordemant de tampon, Isolate Host) = (-0.2 , 0.2).

4.3.5 Coût des chemins

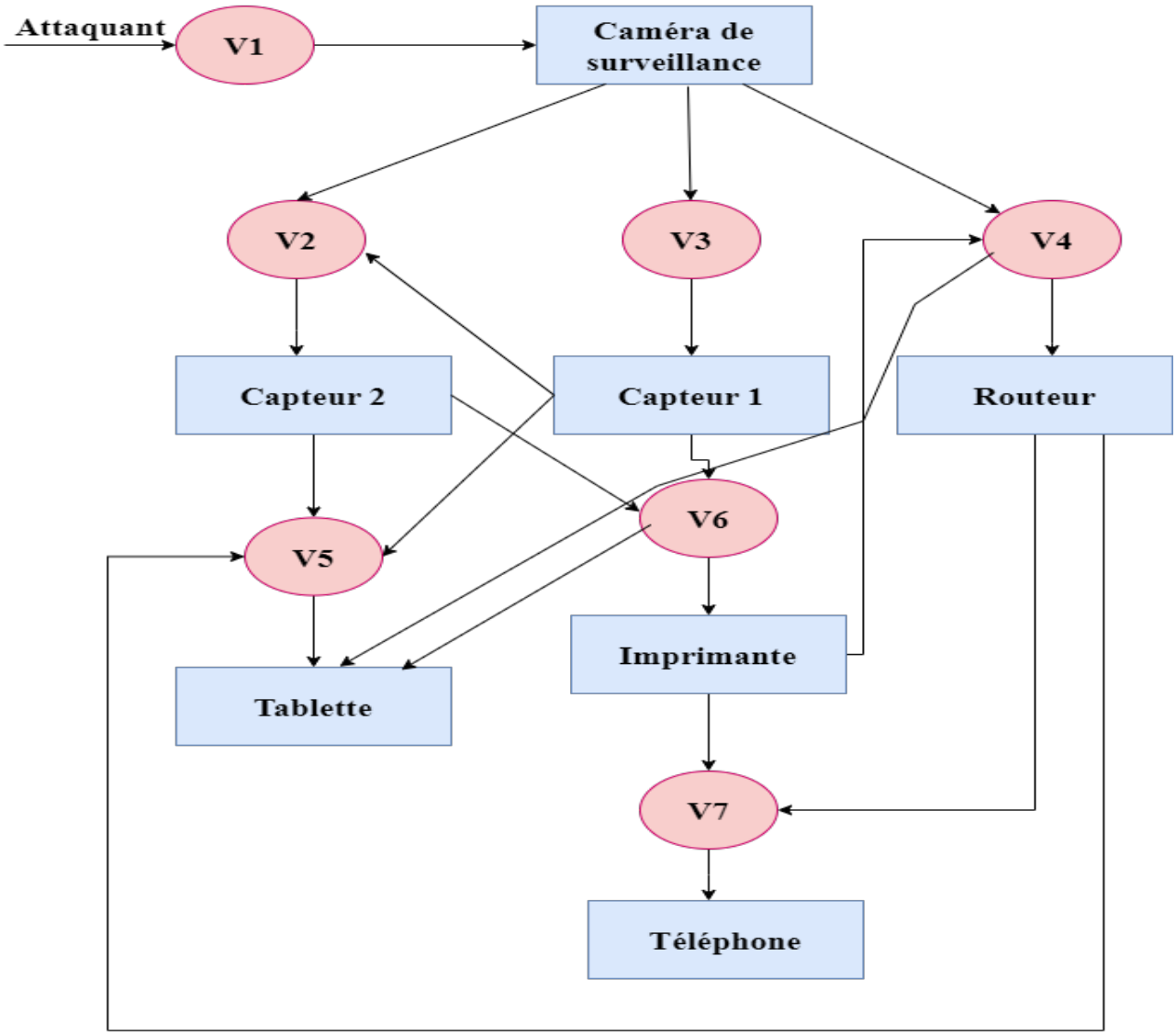


FIGURE 4.3 – Graphe d'attaques correspondant à la topologie du réseau étudiée.

La Figure 4.3 représente le graphe d'attaques que nous avons construit et qui va être analysé. Les rectangles représentent les dispositifs de la topologie représentée par la Figure 4.1 et les cercles représentent les vulnérabilités associer à chaque dispositif et que l'intrus utilise comme stratégies d'attaques. Lorsque nous appliquons le processus d'élimination des stratégies dominées sur le graphe, certains dispositifs seront supprimés tels que la tablette. Afin de prendre des décisions éclairées, il est crucial de déterminer les coûts associés à la suppression d'un chemin spécifique dans ce contexte. Cette évaluation des coûts est réalisée en suivant une démarche qui commence par l'identification du risque de sécurité, noté R.
R est calculé par la formule définie par Feng chen et al. [35] :

$$R = \frac{1}{K} \times W + (1 - W) \sum_{i=1}^{m} \frac{1}{l_i} \tag{4.2}$$

Où :

m : nombre de chemins d'attaques

l_i : distance du chemin d'attaque i

K : différentes vulnérabilités composant les chemins d'attaques

W : probabilité de résistance de la connaissance de l'attaquant

Pour calculer le coût de chaque chemin, nous définissons la formule suivante :

$$\text{Cout}(\text{chemin}_i) = \sum_{j=1}^{n} D \times \frac{R}{\text{Nb}} \tag{4.3}$$

Où :

n : nombre de vulnérabilités composant un chemin donné

D : somme des degrés de vulnérabilités composant les chemins

R : risque de sécurité

Nb : nombre des différentes vulnérabilités présentes dans les chemins

Lorsque nous appliquons l'étape d'élimination des stratégies dans un graphe d'attaques, nous cherchons à identifier tous les chemins qui contiennent des vulnérabilités qui ont été éliminées. L'objectif est ensuite de calculer le coût associé à chaque chemin, afin de déterminer quel chemin peut être supprimé ceci correspond à celui qui a un coût minimum.

Dans notre cas, il existe plusieurs chemins d'attaques et nous nous sommes concentrés sur les chemins qui mènent à la tablette cible. En suivant l'ordre d'élimination des stratégies dominées, nous avons obtenu les chemins suivants :

$\text{Chemin}_1 = \{\text{V4, tablette}\}$

$\text{Chemin}_2 = \{\text{V1, IP caméra, V4, tablette}\}$

$\text{Chemin}_3 = \{\text{V6, tablette}\}$

$\text{Chemin}_4 = \{\text{V6, imprimante, V4, tablette}\}$

$\text{Chemin}_5 = \{\text{V5, tablette}\}$

$\text{Chemin}_6 = \{\text{V1, IP caméra, V4, routeur, V5, tablette}\}$

$\text{Chemin}_7 = \{\text{V1, IP caméra, V2, capteur2, V5, tablette}\}$

$\text{Chemin}_8 = \{\text{V1, IP caméra, V2, capteur2, V6, tablette}\}$

Le risque de sécurité R on le calcule à partir de la formule définie dans 4.2 :
$l_1 = 1; l_2 = 2; l_3 = 1; l_4 = 2; l_5 = 1; l_6 = 3; l_7 = 3; l_8 = 3; m = 11; k = 6; w = 0.5.$

$$R = \frac{1}{6} \times 0.5 + (1 - 0.5) \times \left(\frac{1}{1} + \frac{1}{2} + \frac{1}{1} + \frac{1}{2} + \frac{1}{1} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3}\right) = 2.58$$

Le tableau 4.14 présente le degré de chaque vulnérabilité où le nombre de vulnérabilité égale à 6 :

Les différentes vulnérabilités	Nombre d'arcs liés
Problème d'autorisation (V1)	1
Cross-Site Scripting (XSS) (V2)	3
Cross-Site Scripting (XSS) (V4)	4
Injection SQL (V5)	4
Élévation de privilège (V6)	4
Déni de service (DoS) (V7)	3

TABLE 4.14 – Tableau des degrés de chaque vulnirabilité.

En appliquant la formule du calcul des coûts définie dans 4.3 :

$$\begin{aligned}
\text{Chemin}_1 &= \frac{(4) \times 2.58}{6} = 1.72 \\
\text{Chemin}_2 &= \frac{(1+4) \times 2.58}{6} = 2.15 \\
\text{Chemin}_3 &= \frac{(4) \times 2.58}{6} = 1.72 \\
\text{Chemin}_4 &= \frac{(4+4) \times 2.58}{6} = 3.44 \\
\text{Chemin}_5 &= \frac{(4) \times 2.58}{6} = 1.72 \\
\text{Chemin}_6 &= \frac{(1+4+4) \times 2.58}{6} = 3.87 \\
\text{Chemin}_7 &= \frac{(1+3+4) \times 2.58}{6} = 3.44 \\
\text{Chemin}_8 &= \frac{(1+3+4) \times 2.58}{6} = 3.44
\end{aligned}$$

Parmi les différents chemins à traiter, les plus appropriés dans cet exemple sont le Chemin_1, le Chemin_3 et le Chemin_5. Supposons que nous décidions de supprimer le premier chemin. En conséquence, les vulnérabilités V4 sera éliminée, et le graphe mis à jour est illustré dans la Figure 4.4.

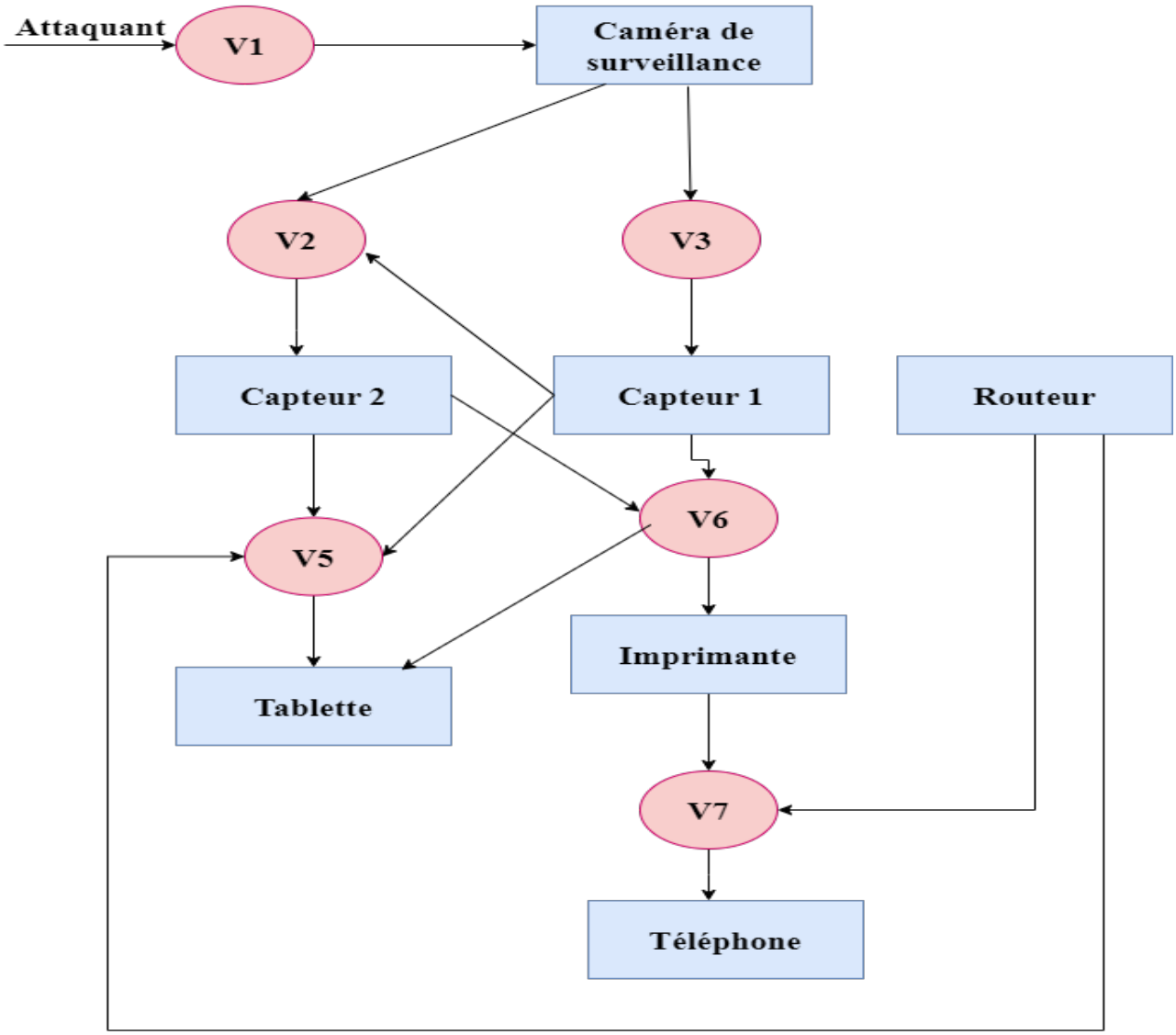


FIGURE 4.4 – Graphe d'attaques après la suppression du V4.

Prenant un autre exemple : supprimer le deuxième chemin, la vulnérabilité V6 sera éliminée, et le graphe mis à jour est illustré dans la Figure 4.5.

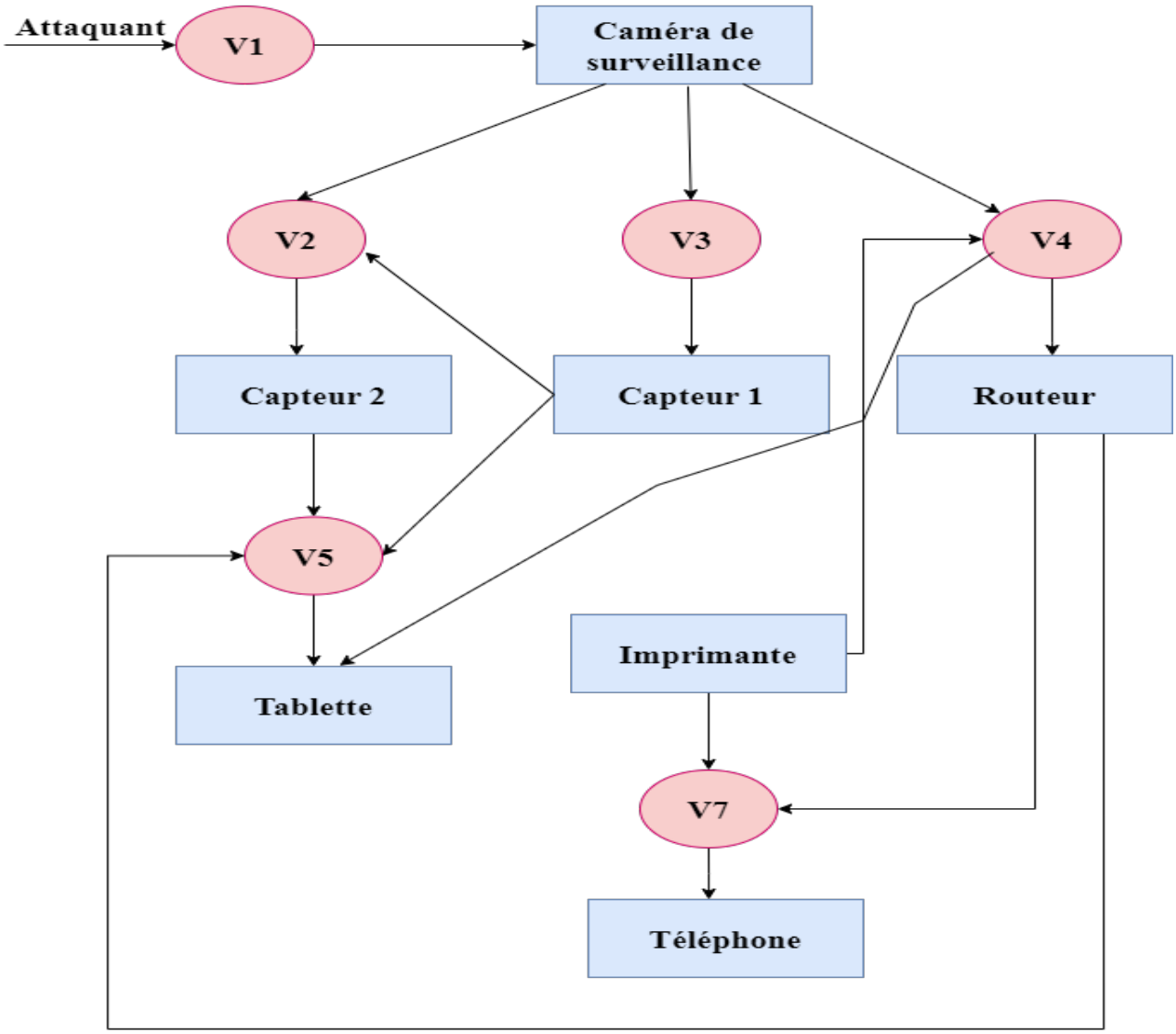


FIGURE 4.5 – Graphe d'attaques après la suppression V6.

Un autre exemple consiste à supprimer le quatrième chemin, la vulnérabilité V5 sera éliminée, et le graphe mis à jour est illustré dans la Figure 4.6.

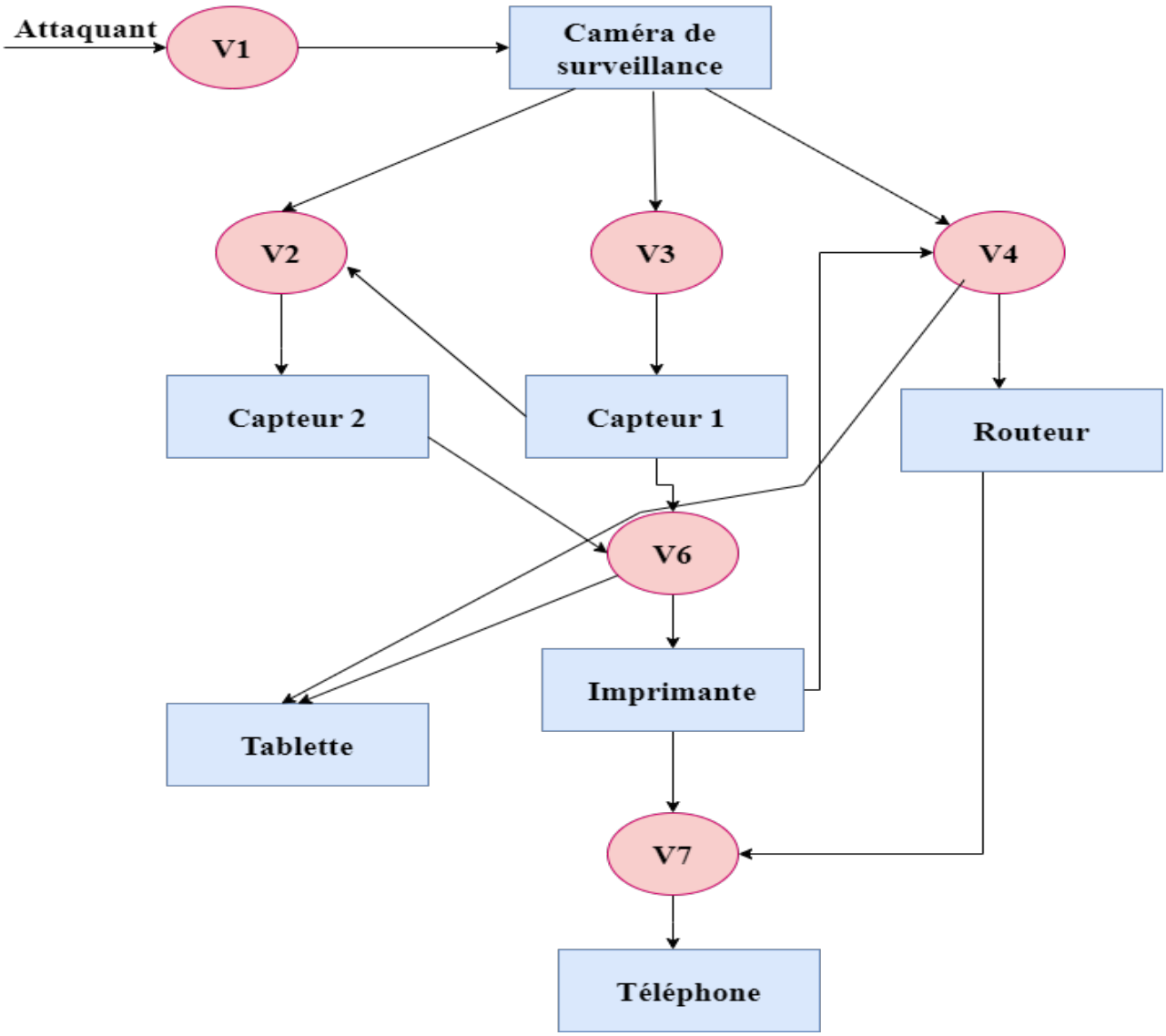


FIGURE 4.6 – Graphe d'attaques après la suppression du V5.

4.4 Evaluation

Après avoir analysé le graphe d'attaques et supprimé les vulnérabilités présentes dans les chemins 1, 3 et 5, nous pouvons prendre une décision éclairée concernant les vulnérabilités à traiter. En se référant au tableau 4.1 (Existence de vulnérabilités dans le dispositif), nous constatons que V4 et V5 présentent un niveau de risque moyen, tandis que V6 présente un niveau de risque élevé.

Il est préférable de supprimer la vulnérabilité V4 plutôt que les vulnérabilité V5 et V6, car elle apparaît dans plusieurs chemins.

En supprimant V4, nous pourrons réduire le risque potentiel sur plusieurs chemins, ce qui aura un impact plus significatif sur la sécurité globale du réseau IoT.

Pour les vulnérabilités restantes, à savoir V5 et V6, nous recommandons de mettre en place des

mesures de sécurité déjà recommandées. Pour la V5, il est recommandé de limiter les privilèges de la base de données en accordant uniquement les privilèges nécessaires aux utilisateurs de la base de données. Évitez d'utiliser un compte avec des privilèges d'administration pour les opérations quotidiennes afin de réduire le risque d'exploitation de l'injection SQL.

Afin de renforcer la sécurité de V6, il est recommandé de mettre en place des mécanismes de surveillance et d'audit pour détecter les activités anormales. Cela peut être réalisé en surveillant les journaux d'événements de l'imprimante et en utilisant des solutions de détection des intrusions. Par ailleurs, il est essentiel de maintenir le pare-feu de l'imprimante à jour en installant régulièrement les dernières versions fournies par le fabricant. Ces mises à jour du pare-feu contiennent souvent des correctifs de sécurité essentiels.

En mettant en œuvre ces mesures de sécurité adaptatives, nous renforcons la sécurité de ce réseau IoT en présentant les risques potentiels associés à V5 et V6, tout en supprimant la vulnérabilité V4 pour un impact plus global.

4.5 Discussion

Notre approche repose sur la théorie des jeux et propose un modèle de sécurité informatique adapté aux réseaux IoT. Ce modèle permet d'analyser les graphes d'attaques afin de déterminer la meilleure stratégie de défense, visant à renforcer la sécurité des réseaux IoT. De plus, notre approche vise également à identifier la meilleure stratégie d'attaque, en minimisant les pertes potentielles pour l'administrateur.

Le jeu se déroule entre deux entités, l'attaquant et l'administrateur, ce qui en fait un jeu à deux joueurs. Chacun des joueurs est rationnel dans son comportement et cherche à maximiser ses gains personnels, ce qui explique l'absence de coopération entre eux. Ainsi, la somme des gains des deux joueurs (pertes ou récompenses) est égale à zéro, d'où le terme "jeu à somme nulle". les joueurs ont une connaissance partielle de l'état du jeu. Cela signifie que chaque joueur peut avoir une perception limitée de l'environnement, ce qui ajoute une dimension d'incertitude et une prise de décision basée sur des probabilités pour cela c'est un jeu stochastique partiellemnt observable.

Afin d'obtenir de bonne solution, nous avons présenter ce jeu sous forme normale car c'est la plus adaptée à notre cas puis nous avons choisi d'utiliser la méthode d'élimination des stratégies dominées pour déterminer les meilleures stratégies de défense et d'attaque dans notre analyse des graphes d'attaques. Cette méthode nous permet de simplifier le problème en éliminant les stratégies qui sont clairement inférieures à d'autres. Ensuite, nous calculons le coût associé à chaque chemin d'attaque pour résoudre le problème d'analyse des graphes d'attaques. Cette approche nous permet d'évaluer les différents chemins possibles.

Notre approche présente plusieurs avantages tels que : elle tient compte de l'incertitude et de la partialité de l'information cela permet de prendre des décisions fondées sur des probabilités et d'ajuster les stratégies en fonction des nouvelles informations disponibles. Nous avons conçu un modèle de sécurité informatique qui se concentre spécifiquement sur les réseaux IoT et tient

compte de leurs caractéristiques particulières. Grâce à notre approche, nous sommes capables de relever les défis de sécurité uniques auxquels sont confrontés les réseaux IoT, et nous pouvons proposer des stratégies de défense et d'attaque adaptées à cet environnement.

4.6 Conclusion

Ce chapitre marque la dernière étape de notre projet, où nous avons présenté le fonctionnement de notre proposition en illustrant son application sur un exemple de graphe d'attaques. En résumé, notre travail ouvre de nouvelles perspectives pour renforcer la sécurité des réseaux IoT et fournir aux administrateurs des outils efficaces pour gérer les menaces et protéger leurs systèmes de manière proactive.

Conclusion générale et perspectives

L'Internet des objets (IoT) marque une évolution significative de l'Internet en offrant de multiples possibilités et bénéfices. En connectant les objets physiques à l'Internet, l'IoT facilite une interconnexion étendue et une interaction dynamique entre les dispositifs et les systèmes, ouvrant ainsi de nouvelles voies pour l'innovation et l'amélioration de notre vie quotidienne.

Cependant, l'Internet des objets (IoT) fait face à des défis significatifs, notamment en ce qui concerne la sécurité et la confidentialité des données car la connectivité accrue signifie également une plus grande vulnérabilité aux cybers attaques. Ces défis doivent être résolus pour garantir la protection des données et préserver la confiance des utilisateurs dans un réseau IoT.

Nous avons proposé une nouvelle approche basée sur la théorie des jeux pour analyser les graphes d'attaques et réduire les vulnérabilités du réseau IoT. Nous avons transformé le problème de sécurité en un jeu stochastique partiellement observable fini à deux joueurs, non coopératif et à somme nulle. Ce jeu a été représenté sous forme normale, ce qui nous a permis d'utiliser les méthodes de résolution de la théorie des jeux, notamment l'élimination des stratégies dominées. En utilisant cette approche, nous avons calculé les coûts des différents chemins possibles dans le graphe d'attaques vers la cible. Les chemins avec des coûts élevés ont été identifiés et des mesures de sécurité adaptées ont été prises pour les éliminer ou les atténuer. L'efficacité de cette approche a été démontrée à travers une évaluation sur un exemple de graphe d'attaques dans le contexte IoT. Les résultats ont montré que notre méthode était efficace pour résoudre les problèmes d'analyse des graphes d'attaques et pour réduire les vulnérabilités dans un environnement IoT.

Comme perspective d'amélioration pour notre approche consiste à intégrer l'apprentissage par renforcement afin de prédire les nouvelles attaques et les chemins d'attaque dans le réseau IoT. Cela permettrait d'améliorer la capacité prédictive et de prendre des mesures de sécurité plus efficaces en réponse aux menaces émergentes. En s'inspirant du modèle AGA-POSG, cette extension renforcerait notre capacité à analyser les graphes d'attaques et à réduire les vulnérabilités du réseau IoT.

Bibliographie

[1] G.Lampropoulos, K.Siakas & T.Anastasiadis, (2019), Internet of Things in the Context of Industry 4.0 : An Overview, International Journal of Entrepreneurial Knowledge, 7(1), 4-19, doi : 10.2478/ijek-2019-0001.

[2] V.Mehta, C.Bartzis, H.Zhu, E.Clarke & J.Wing, (2006), Ranking attack graphs, In Recent Advances in Intrusion Detection : 9th International Symposium, RAID 2006 Hamburg, Germany, September 20-22, 2006 Proceedings 9, 127-144, Springer Berlin Heidelberg.

[3] S.Li, L.D.Xu & S.Zhao, (2015), *The internet of things :a survey*, Information systems frontiers, Springer Science+Business Media New York, 17 :243-259, DOI 10.1007/s10796-014-9492-7.

[4] P.J.Benghozi, S.Bureau & F.Massit-Folea, (2008), *L'Internet des objets. Quels enjeux pour les Européens ?*, Documents scientifiques de niveau recherche, hal-00405070

[5] R.Saad, (2016), *Modèle collaboratif pour l'Internet of Things (IoT)*, Thése de doctorat, Université du Québec à Chicoutimi.

[6] S.Droua & K.Terir, *Gestion de la confidentialité des données pour les dispositifs IOT(Internet of Things)* , Mémoire de master, Informatique légale et multimédia, Université de Jijel.

[7] L.Ouabba & S.Mehah, (2021), *Internet of Things, protocoles de communication et simulation d'un scénario[maison intelligente]* , Mémoire de master , Administration et sécurité des réseaux, Université de Bejaia.

[8] M.U.Farooq, M.Waseem, S.Mazhar, A.Khairi & T.Kamal, (2015), *A review on internet of things (IoT)*, International journal of computer applications, 113(1), 1-7.

[9] K.K.Patel, S.M.Patel & P.Scholar, (2016), *Internet of things-IOT :definition, characteristics, architecture, enabling technologies, application & future challenges*, International journal of engineering science and computing, 6(5).

[10] Y.Abbassi & H.Benlahmer, (2021), *Un aperçu sur la sécurité de l'internet des objets (IOT)*, In Colloque sur les Objets et systèmes Connectés-COC'2021.

[11] A.Al-Fuqaha, M.Guizani, M.Mohammadi, M.Aledhari & M.Ayyash, (2015), *Internet of things : A survey on enabling technologies, protocols and applications*, IEEE communications surveys tutorials, 17(4), 2347-2376.

[12] Y.Perwej, K.Haq, F.Parwej, M.Mumdouh & M.Hassan, (2019), *The internet of things (IoT) and its application domains*, International Journal of Computer Applications, 975(8887), 182.

[13] F.A.Alaba, M.Othman, I.A.T.Hashem & F.Alotaibi, (2017), *Internet of Things security : A survey, Journal of Network and Computer Applications*, 88, 10-28.

[14] R. Shantha Mary Joshitta & L.Arockiam, (2016), *Security in IoT environment : a survey*, International Journal of Information Technology and Mechanical Engineering, 2(7), 1-8.

[15] S.G.H.Soumyalatha, (2016), *Study of IoT : understanding IoT architecture, applications, issues and challenges*, In 1st International Conference on Innovations in Computing Networking (ICICN16), CSE, RRCE, International Journal of Advanced Networking & Applications, Bengaluru, Karnataka, (Vol. 478).

[16] A.Tiwary, M.Mahato, A.Chidar, M.K.Chandrol, M.Shrivastava & M.Tripathi, (2018), *Internet of Things (IoT) : Research, architectures and applications*, International Journal on Future Revolution in Computer Science & Communication Engineering, 4(3), 23-27.

[17] A.Bouzidi & M.Kacel, (2013), *Etude d'un raffinement de l'équilibre de Nash*, Mémoire de Master, Université de TIZI-OUZOU.

[18] H. Slimani, (2022), *Théorie des jeux et applications*, Support de cours, Master 2 RS, Université de Bejaïa.

[19] S.Berri & M.Bouhaddi, (2012), *Théorie des jeux appliquée à la sécurité des réseaux ad hoc*, Mémoire de Master, Université de Béjaia.

[20] E.Elkind & J.Rothe, (2016), *Economics and computation : an introduction to algorithmic game theory, computational social choice and fair division*,Springer Texts in Business and Economics, 135-193, DOI 10.1007/978-3-662-47904-9.

[21] J.C.Harsanyi, (1995), *Games with incomplete information* , The American Economic Review, 85(3), 291-303.

[22] M.A.Hamila, E.Grislin, R.Mandiau & A.I.Mouaddib, (2010), *Les jeux stochastiques : un modèle de coordination multi-agents*, Document de recherche, In Proceedings of RFIA.

[23] A.H.Anwar, C.Kamhoua & N.Leslie, (2019), *A game-theoretic framework for dynamic cyber deception in internet of battlefield things*, In Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems :Computing, Networking and Services,November 12–14, 2019, Houston, TX, USA. ACM, New York, NY, USA, (pp. 522-526).

[24] K.Ingols, R.Lippmann & K.Piwowarski, (2006), *Practical attack graph generation for network defense*, IEEE, In 2006 22nd Annual Computer Security Applications Conference (ACSAC'06), 121-130.

[25] H.Lamia, (2022), *Sécurité des Réseaux*, Support de cours, Université de Béjaia, https ://elearning.univ-bejaia.dz/course/view.php ?id=15032.

[26] Z.S.Alwan & M.F.Younis, (2017), *Detection and prevention of SQL injection attack : a survey*, International Journal of Computer Science and Mobile Computing, 6(8), 5-17.

[27] R.Yahiaoui, (2022), *Application des alliances pour l'analyse des graphes d'attaques*, Mémoire de Master, Universtité de Béjaia, Département informatique.

[28] V.Shandilya, C.B.Simmons & S.Shiva, (2014), *Use of attack graphs in securitysystems*, Journal of Computer Networks and Communications, 2014.

[29] M.Albanese, S.Jajodia & S.Noel, (2012), *Time-efficient and cost-effective network hardeningusingattack graphs*, In IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), 1-12.

[30] K.Bouafia L.Hamza, (2022), *Game theory approach for analysing attack graphs*, International Journal of Information and Computer Security, 19(3-4), 305-320.

[31] B.Yiğit, G.Gür, F.Alagöz & B.Tellenbach, (2019), *Cost-aware securing of IoT systems using attack graphs*, Ad Hoc Networks, Preprint submitted to Elsevier, 86, 23-35.

[32] Y.Ma, Y.Wu, D.Yu, L.Ding & Y.Chen, (2022), *Vulnerability association evaluation of Internet of thing devices based on attack graph*, International Journal of Distributed Sensor Networks, 18(5), 15501329221097817.

[33] O.Almazrouei & P.Magalingam, (2022), *The Internet of Things Network Penetration Testing Model Using Attack Graph Analysis*, In 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 360-368), IEEE.

[34] K.Koo, D.Moon, J.H.Huh, S.H.Jung & H.Lee, (2022), *Attack Graph Generation with Machine Learning for Network Security*, Electronics, In Proceedings of the 16th International Conference on Multimedia Information Technology and Applications (MITA 2020), Yeosu, Korea, 20–21 November 2020, 11(9), 1332.

[35] F.Chen, D.Liu, Y.Zhang & J.Su, (2010), *A scalable approach to analyzing network security using compact attack graphs*, Journal of Networks, 5(5), 543.

[36] https ://www.sigfox.com/what-is-sigfox/. Consulté le 23/04/2023.

[37] https ://www.xmcyber.com/glossary/what-are-attack-graphs/. Consulté le 27/10/2022.

[38] https ://nvd.nist.gov/. Consulté le 27/10/2022.

[39] `https://cve.mitre.org/cve/search_cve_list.html`. Consulté le 27/10/2022.

[40] S.Rabhi & M.Chalal, (2019), *Problème de Contrôle Optimal par l'Approche de La Théorie des Jeux*, Mémoire de Master, Université de TIZI-OUZOU.

Résumé L'Internet des objets (IdO) est une technologie révolutionnaire qui a transformé notre façon d'interagir avec le monde qui nous entoure. Il englobe un réseau de dispositifs physiques connectés. Cependant, cette interconnexion soulève également des défis majeurs l'un de ces défis réside dans la sécurité. Avec des milliards d'appareils connectés, la protection des données sensibles et la prévention des cyberattaques deviennent primordiales. Les dispositifs IoT sont souvent vulnérables aux attaques. Dans ce mémoire, nous avons proposé une nouvelle approche d'analyse des graphes d'attaques basée sur la théorie des jeux. Cette méthode consiste à transformer un problème de sécurité d'un réseau IoT en un jeu stochastique partiellement observable (POSG) finis à deux joueurs et à extraire les meilleures stratégies pour chacun des deux joueurs. Des mesures de sécurité ont été mises en place pour éliminer ou atténuer les chemins d'attaque identifiés avec des coûts engendrés dans le graphe vers la cible. Ce mémoire vise à utiliser des méthodes de la théorie des jeux pour résoudre le problème d'analyse des graphes d'attaques.

Mots-clés– IdO, vulnérabilité, analyse des graphes d'attaques, théorie de jeux, POSG, AGA-POSG.

Abstract The Internet of Things (IoT) is a revolutionary technology that has transformed the way we interact with the world around us. It encompasses a network of connected physical devices. However, this interconnectedness also raises major challenges - one of which is security. With billions of devices connected, protecting sensitive data and preventing cyber-attacks are becoming paramount. IoT devices are often vulnerable to attack. In this document, we proposed a new approach to analysing attack graphs based on game theory. This method consists of transforming an IoT network security problem into a finite two-player stochastic partially observable game (POSG) and extracting the best strategies for each of the two players. Security measures were implemented to eliminate or mitigate identified attack paths with costs incurred in the graph to the target. This thesis aims to use game-theoretic methods to solve the problem of analysing attack graphs.

Key-words– IoT, vulnerability, analysis of attack graphs, game theory, POSG,AGA-POSG.