UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE

APPLIQUÉES

PAR

BENJAMIN DUMISANI MALAMU

VERS UNE CYBERTROMPERIE RÉSILIENTE : CONCEPTION D'UNE APPROCHE BASÉE SUR LA REDONDANCE DES POTS DE MIEL DIVERSIFIÉS.

JANVIER 2026

REMERCIEMENTS

En préambule, je remercie l'Éternel Dieu tout-puissant de m'avoir donné le souffle de vie jusqu'aujourd'hui pour la présentation de ce mémoire.

J'adresse mes plus respectueux remerciements à ma directrice de recherche, Professeure Hakima Ould-Slimane, pour son mentorat, sa confiance et son soutien financier, morale à mon égard. Sa rigueur, sa patience ainsi que son esprit critique m'ont permis à bien mener ma recherche. Je lui suis infiniment reconnaissant pour les opportunités qu'elle m'a accordées durant ma maîtrise. De tout cœur, je lui dis merci.

Je tiens à exprimer toute ma gratitude au personnel de l'UQTR, particulièrement le corps professoral ainsi que les administratifs du département de mathématiques et informatique.

J'adresse mes remerciements à tous les étudiants avec qui j'ai eu à collaborer et, plus précisément, à Babacar Dieng pour sa fraternité.

Je remercie ma famille en République démocratique du Congo et spécialement en mon frère Elisha Tshibuabua Dumisani, qui m'a soutenu financièrement, ainsi qu'à mon cousin Kato Nsangwa Mumba, qui m'a accueilli ici au Canada.

Enfin je remercie toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

DÉDICACÉ

Je dédie ce mémoire à mon frère, Elisha Tshibuabua Dumisani. Merci d'avoir soutenu mes études de maitrise ; ton aide n'a pas seulement allégé mon parcours, elle a rendu possible un rêve qui me tenait à cœur.

Cette thèse témoigne toute ma reconnaissance et ma gratitude pour toi.

Ton frère.

TABLE DES MATIÈRES

LISTES DES TABLEAUX

LISTES DE FIGURES

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

TDR : Threat Detection and Reponse

IDS : Intrusion Detection Systems

ICS : Industrial Control Systems

IoT : Internet of Things

IIoT : Industrial Internet Of Things

IoHT : Internet of Health Things

IoBT: Internet of Battlefield Things

WSN: Wireless Sensor Networks

SDN: Software Defined Networks

OSN: Online Social Networks

MTD: Moving Target Defense

LIHP: Low Interaction Honey Pots

MIHP: Medium Interaction Honey Pots

HIHP: High Interaction Honey Pots

VM: Virtual Machine

FTP: File Transfer Protocol

SSH: Secure Shell

IP: Internet Protocol

ML: Machine learning

APT: Advanced Persistent Threat

GBO: Game-Base Optimizer

TBM: Tolerance-Based Model

WAN: Wide Area Network

LAN: Local Area Network

DMZ : Zone démilitarisée

RÉSUMÉ

De nos jours, la cybertromperie est un élément essentiel des systèmes de défense proactifs et réactifs. Les équipements militaires sont de plus en plus connectés à Internet et la sécurité des réseaux de nouvelle génération devient de plus en plus complexe. Ce vaste réseau de tactiques militaires devient ainsi exposé et vulnérable, ouvrant la voie aux cyberattaques. La sécurité d'un tel réseau est donc primordiale.

Dans cet article, nous proposons une approche de cybertromperie basée sur la théorie des jeux pour l'allocation de pots de miel diversifiés, c'est des systèmes factices conçus pour tromper l'attaquant et observer son comportement. Afin de garantir une haute disponibilité de notre mesure de sécurité, nous optons pour une redondance, entendue comme la duplication intentionnelle de ressource de défense. Afin d'assurer la continuité du service même en cas de première attaque réussie. Nous mobilisons également la cybersécurité et la cyberdéfense, respectivement la protection des systèmes informatiques et l'ensemble des stratégies actives visant à contrer ou ralentir l'adversaire. Nous analysons ainsi l'impact sur la récompense du défenseur, avec ou sans probabilité de succès de l'attaquant. L'optimisation sera résolue à l'aide de l'équilibre de Nash, un concept de théorie des jeux décrivant une situation où aucun acteur n'a intérêt à modifier seul sa stratégie. Concernant le placement des pots de miel, nous adoptons un déploiement avec redondance pour garantir la sécurité des nœuds critiques.

Nos résultats montrent que la redondance renforce la capacité à diversifier le pot de miel, assurant ainsi la sécurité du réseau à long terme.

Mots clés : pots de miel, cybersécurité, cyberdéfense, redondance, théorie de jeux

CHAPITRE I

INTRODUCTION

1.1.Contexte

La protection des systèmes et des actifs critiques est devenue une préoccupation majeure dans notre monde interconnecté, où une seule perturbation peut se répercuter sur plusieurs secteurs de la société. Par exemple, en 2019, le département d'État américain a publié un document sur sa stratégie de déception militaire pour les opérations multidomaines [106], incluant la sécurité des réseaux [2]. Cette évolution souligne que les cybermenaces, qu'elles visent des infrastructures militaires, industrielles ou civiles [107], sont désormais omniprésentes, difficiles à anticiper et de plus en plus sophistiquées.

Avec l'essor des réseaux de nouvelle génération (5G, IoT, cloud computing), la surface d'attaque s'élargit considérablement. Leur nature interconnectée augmente les surfaces d'attaque, nécessitant des mécanismes avancés de détection, de prévention et de réponse. Dans cette dynamique, les forces armées intègrent désormais pleinement le combat cybernétique comme mode d'action à part entière, coordonné avec les opérations traditionnelles dans une manœuvre globale

1.2. Problématique et motivation.

Dans ce contexte, les menaces cyber sont devenues permanentes et en évolution continues. Profitant des opportunités offertes par le cyberespace, la multiplicité des équipements connectés. Le nombre et l'intensité des attaques menées dans ce milieu ne cessent de croître, visant le profit financier, la captation de données, la manipulation des opinions ; contrôler l'accès à des armements et équipements importants ou encore protéger des informations confidentielles, des centres de données et des zones de commandement stratégiques a pour point commun la mise en place de mesures de sécurité renforcées.

La réponse stratégique de pays consiste à renforcer sa capacité à détecter et prévenir les actes de cybermalveillance et à y répondre. Et chaque pays s'appuie sur des moyens de cyberdéfense forts et résilients pour accomplir les trois tâches fondamentales que sont la dissuasion et la défense, la gestion des crises et la sécurité proactive. Pour être à l'abri des attaques, une sécurité quasi intégrale est requise [108].

Cependant, les technologies traditionnelles de pots de miel sont non seulement passives, mais également inefficaces face à des adversaires expérimentés [109] et face aux risques de piratage, d'espionnage, qu'il s'agisse de ports, de systèmes radars d'aéroport, de gares ferroviaires ou d'entrepôts routiers, de transport et de logistique, de sabotage de sites militaires sensibles ou de sites nucléaires. Ces sites sont confrontés à des problèmes de sécurité courants. Outre leur configuration, souvent étendue et dont le périmètre est difficile à surveiller, de nombreuses contributions des parties prenantes sont nécessaires pour contrôler et optimiser les flux de trafic.

1.3. Objectifs

L'objectif de ce mémoire est de proposer une architecture capable de pérenniser mes mesures de sécurité dans un réseau en exploitant la redondance des pots de miel. Dans [105], la diversification des logiciels est mise en avant : toutefois, nous soulignons qu'un pot de miel peut rencontrer un dysfonctionnement interne et s'éteindre sans être attaqué. D'où l'importance de la technique de redondance dans le placement des pots de miel. Le principe de cette technique est d'utiliser plusieurs configurations différentes des pots de miel sur différentes architectures matérielles et la défense en profondeur appliquée à la cybersécurité, qui consistent en un ensemble de systèmes de sécurité dont les périmètres d'action sont relayés. L'idée principale est simple : un attaquant exploitant une vulnérabilité spécifique à une architecture matérielle aura peu de chances de compromettre une machine de configuration différente. Ainsi, si un pot de miel est contourné ou attaqué, la présence d'un second, doté d'une configuration alternative, permet de maintenir la capacité de détection et de tromperie.

La cybertromperie ne se limite pas à défendre un réseau contre les attaques : elle aide aussi les systèmes de défense à analyser les méthodes utilisées par le réseau. Puisque les attaquants collectent des informations sur les réseaux avant de lancer des activités malveillantes [110] à l'aide d'outils comme Nmap [111], la déception est d'une grande importance pour

déformer l'état du réseau et les informations qui pourraient être collectées pendant la phase de reconnaissance [105].

En réponse à l'accroissement et à l'évolution des menaces cyber sur les organisations, la mise en place de mécanismes de sécurité redondants et robustes sont une nécessité. En particulier, l'adoption de méthodes de défense en profondeur s'impose comme un impératif pour les organisations modernes qui font face à des acteurs avec des moyens et une motivation élevée. Concernant la redondance avec diversification, nous nous basons sur une idée simple : un attaquant qui cible une vulnérabilité d'une machine qui fonctionne sur une architecture matérielle particulière a peu de chance de fonctionner sur une autre machine fonctionnant sur une autre architecture matérielle.

La technologie rend beaucoup de choses possibles, mais possibilité ne rime pas toujours avec sécurité. À mesure que les cybermenaces augmentent en volume et en sophistication ; la technologie devient essentielle pour répondre aux besoins de défense de pays face à ce fléau.

Pour modéliser l'interaction stratégique entre l'attaquant et le défenseur, nous faisons appel à la théorie des jeux, qui permet d'analyser et d'optimiser l'attribution de pots de miel destinés à protéger efficacement les sites de valeur. La tromperie, en tant que mécanisme intentionnel visant à présenter de fausses informations ou à masquer des informations vraies [112], dépend fortement de ce que chaque joueur peut observer [1], ce qui motive encore davantage l'utilisation d'un cadre théorique formel.

1.4.Contribution

La nouveauté principale de ce travail réside dans la formulation d'une approche de cybertromperie intégrant une redondance diversifiée des pots de miel, un aspect encore peu étudié dans la littérature. Contrairement aux approches classiques centrées sur la seule diversification logicielle, notre contribution met en évidence l'intérêt d'une duplication hétérogène des pots de miel afin d'assurer la continuité des capacités de détection en cas de défaillance ou de contournement d'un premier dispositif. Nous proposons pour cela une modélisation formelle en jeu à somme nulle, permettant d'évaluer l'impact de cette redondance sur les stratégies mixtes optimales obtenues à l'équilibre de Nash. Les résultats issus de la simulation montrent que la redondance diversifiée améliore significativement la résilience du mécanisme de tromperie, offrant ainsi un cadre robuste pour renforcer la pérennité de la cybertromperie dans les environnements critiques.

1.5.Organisation du document

La suite de ce mémoire est organisée comme suit : le chapitre deux porte sur les notions et concepts préliminaires de la tromperie défensive et les pots de miel. Le chapitre trois porte sur l'état de l'art concernant la cybertromperie. Le quatrième chapitre présente la contribution et la méthodologie utilisée ainsi que l'expérimentation et les résultats numériques. Et enfin, le dernier chapitre porte sur la conclusion générale et perspective.

En annexe de ce travail, nous présentons notre article scientifique intitulé GAME THEORY : CYBER DECEPTION BASED ON THE REDUNDANCY OF DIVERSIFIED HONEYPOTS qui a été soumis, accepté et publié au journal Procedia Computer Science (The 20th International Conference on Future Networks and Communications (FNC2025) August 4-6, 2025, Leuven, Belgium).

Lien de téléchargement : https://authors.elsevier.com/sd/article/S1877050925022124

CHAPITRE II

NOTIONS ET CONCEPTS PRÉLIMINAIRES

2.1. Section 1 : La tromperie défensive dans la cybersécurité et cyberdéfense

La tromperie défensive est une approche prometteuse pour la cyberdéfense. Grâce à la tromperie défensive, un défenseur peut anticiper et prévenir les attaques en trompant ou en leurrant un attaquant, ou en cachant certaines de ses ressources[8].

Le concept conventionnel de tromperie militaire fait référence aux actions entreprises pour tromper intentionnellement un ennemi sur ses forces et ses faiblesses, ses intentions et ses tactiques [8]. Le défenseur utilise la tromperie pour manipuler les actions de l'ennemi afin de faire avancer sa mission[8]. La tromperie défensive s'applique à un large éventail d'interactions entre un attaquant et un défenseur dans des situations de conflit[8]. Le concept de la tromperie défensive initié dans les environnements militaires a été introduit dans les cyberdomaines sous le nom de cybertromperie.

Almeshekah et Spafford [29] ont affiné le concept de cybertromperie dans [30] comme « des actions planifiées prises pour induire en erreur et/ou confondre les attaquants et les amener ainsi à prendre (ou à ne pas prendre) des actions spécifiques qui aident les défenses de sécurité informatique ». Bien que le concept de tromperie soit hautement multidisciplinaire [31], l'idée commune est d'induire une entité en erreur afin qu'elle forme une fausse croyance et de contrôler son comportement en fonction de cette fausse croyance . Par conséquent, lorsque la tromperie est exécutée avec succès, une personne trompée répond de manière sous-optimale, ce qui offre des avantages dans la réalisation du but du trompeur.

La tromperie peut être appliquée lorsqu'un objet réel et vrai existe ou qu'aucun objet réel n'existe. Pour être plus précis, un défenseur peut utiliser de vrais objets ou informations, mais peut vouloir les cacher en mentant ou en fournissant de fausses informations à son égard. Par exemple, même si un système utilise Windows comme système d'exploitation, mais peut mentir, il utilise Unix. Mais le système utilise effectivement un certain système d'exploitation.

D'autre part, un faux objet, qui n'existe nulle part, peut être créé dans le but de créer de l'incertitude ou de la confusion, ce qui peut conduire un adversaire à prendre une décision sous-optimale ou médiocre, comme des jetons de miel ou des fichiers de miel.

Dans « L'art de la guerre » [32], le traité militaire le plus important et le plus célèbre d'Asie depuis deux mille ans, Sun Tzu observe : « Toute guerre est basée sur la tromperie. Par conséquent, lorsque nous sommes capables d'attaquer, nous devons paraître incapables ; lorsque nous utilisons nos forces, nous devons paraître inactifs.» De nos jours, 2 500 ans plus tard, la tromperie peut toujours être appliquée efficacement dans la guerre contre la cybercriminalité en plus des opérations militaires modernes. La tromperie est un excellent moyen d'obtenir des informations sur l'adversaire [33] . Bien qu'il existe de nombreuses approches de sécurité, comme le pare-feu, les systèmes de détection d'intrusion pour reconnaître et empêcher les acteurs malveillants d'accéder aux ressources critique, elles ne constituent toujours pas des solutions de défense actives. Au lieu d'attendre que les attaquants s'introduisent dans le système réseau et de les bloquer rapidement, la technologie de cybertromperie, connue sous le nom de « pot de miel » de nouvelle génération, est adoptée pour se faire passer pour de véritables ressources réseau afin d'attirer les pirates. Grâce à cette stratégie proactive, des pièges cybernétiques et des systèmes de leurre sont déployés à certains endroits du réseau pour consommer l'effort et le temps des attaquants[34].

Au cours des dernières décennies, les concepts de tromperie ont connu une popularité croissante dans la sécurité de l'information avec le concept de pot de miel pour tromper les pirates informatiques potentiels dans un piège. La cybertromperie a énormément reçu l'attention des chercheurs universitaires et industriels [33] . Elle apporte une efficacité dans la détection précoce des attaques et donne plus d'obstacles aux attaquants pendant leur phase de reconnaissance.

La cybersécurité est devenue une priorité de grande importance pour les chefs de système de sécurité des entreprises, et cela depuis plusieurs années. Car au fil du temps, les attaquants ont développé une ingéniosité rendant les attaques plus sophistiquées. Les premiers chercheurs ont aidé les équipes de sécurité à créer différents types de pare-feu qui empêchaient les adversaires d'entrer dans le réseau. Ces mesures de sécurité ont fonctionné pendant un certain temps jusqu'à ce que les attaquants apprennent à contourner ces pare-feu.

Les chercheurs en sécurité ont cherché à trouver d'autres moyens de vaincre les attaquants, et ils ont créé des technologies de déception[35]. Les mécanismes de sécurité

précédents étaient conçus pour répondre aux menaces et aux logiciels malveillants dans un laps de temps déterminé. Le flux continu et constant de logiciels malveillants bloque désormais le système de détection et pénètre très facilement dans le réseau interne. La menace « zero day » est un exemple de ces logiciels malveillants sophistiqués qui peuvent facilement éviter tout antivirus et pénétrer dans le système sans être détectés. Les systèmes de détection et de réponse aux menaces (Threat Detection and Reponse : TDR) aident à détecter et à vaincre les attaquants potentiels. Le cadre défend le système contre de nombreux types de logiciels malveillants nuisibles[36].

L'idée de base des technologies de tromperie est de piéger les adversaires et de confondre les actifs réels et les actifs factices. Les experts en sécurité ont créé des couches fictives du réseau en même temps que la couche réelle du réseau pour tromper les attaquants. De nombreux attaquants sont tombés dans ces couches trompeuses au début, mais ont ensuite trouvé des moyens d'ignorer les technologies de tromperie. Les responsables de la mise en œuvre de la sécurité tentent à présent de créer des technologies de tromperie plus transparentes et plus efficaces qui peuvent rendre les adversaires plus méfiants à l'égard des actifs du réseau[37].

2.1.1. Outils de tromperie

Dans le passé, les techniques et outils de sécurité comprenaient généralement des pare-feu et des systèmes de détection d'intrusion (IDS). Ces systèmes fonctionnaient bien, mais, dans le cas de réseaux plus vastes où le trafic quotidien se chiffre en millions, ces réseaux échouaient gravement. Par exemple, un serveur normal d'une grande entreprise reçoit chaque jour des millions d'appels de fichiers provenant de différentes régions. Il est pratiquement impossible d'identifier l'attaquant parmi ces millions. Les outils de tromperie n'étaient pas la seule et unique solution proposée par les chercheurs en sécurité après avoir découvert ce problème. Les données massives (big data) ont été proposées comme solution. Cette solution fonctionnait à petite échelle. Lorsque le trafic quotidien a dépassé une limite normale, les données enregistrées ont augmenté et se sont transformées en mégadonnées. Les entreprises ont commencé à employer des experts en mégadonnées pour mettre en œuvre des techniques, telles que l'exploration de données, afin de découvrir des données utiles parmi la masse de données. L'utilisation des mégadonnées a tellement augmenté qu'il est devenu presque impossible de repérer les données utiles parmi la masse de données. La technologie de déception a créé un changement de paradigme, car elle a modifié la tendance traditionnelle de l'utilisation des mégadonnées[38]. Elle a transformé les mégadonnées en données utiles en les

réduisant à un niveau minimal. Le problème s'en trouve considérablement réduit. Les entreprises n'ont plus besoin de fouiller dans des millions d'entrées pour savoir qui a attaqué leur système. Elles peuvent directement trouver le pirate en examinant le trafic par l'intermédiaire de pots de miel.

Les pirates informatiques utilisaient le même schéma traditionnel pour attaquer et pénétrer dans les réseaux. Ce schéma était bien connu et il était facile de les détecter et de les vaincre, mais les nouvelles technologies ont aidé les attaquants à pénétrer dans le réseau en utilisant différents modèles, ce qui rend non seulement difficile la détection par les responsables de la sécurité, mais aussi la mise en échec par les cadres de sécurité.

Les outils de tromperie ont trois taches principales :

- Ils examinent les schémas des attaquants à l'aide de différentes méthodes d'investigations
- Ils recueillent des informations sur les schémas d'attaque
- Ils contrecarrent ces attaques en utilisant le même schéma que celui utilisé par l'intrus, ce qui facilite leur mise en échec.

Les machines utilisent des leurres, un type de pare-feu. Les attaquants sont dirigés vers un système leurre qui lit le modèle d'attaque et prépare une réponse en fonction de ce modèle. Le logiciel malveillant doit passer par ce leurre avant d'entrer dans le système[39].

2.1.2. Principe de conception

Selon [8], les principes est présenté sous quelques aspects :

- Quel attaquant tromper : ce principe de conception demande de déterminer quel type d'attaquant un défenseur veut tromper. Par exemple, si le défenseur vise à tromper les attaquants effectuant des attaques de reconnaissance en tant qu'attaquants extérieurs, il peut viser à les tromper en fournissant de fausses informations sur une configuration système. Par conséquent, déterminer quel attaquant tromper revient à décider quels attaquants cibler par le défenseur. Étant donné que le développement d'une technique de tromperie défensive entraîne un coût. Il convient d'effectuer une analyse et une enquête approfondies pour déterminer si un attaquant spécifique doit être empêché ou détecté par une technique de tromperie défensive en termes de coût, d'efficacité et d'efficience du déploiement de la technique.

- Quand tromper : ce principe de conception fait référence à la détermination du moment où une technique de tromperie doit être utilisée en termes de stade d'attaque d'un attaquant dans la chaîne de destruction cybernétique [40], [41]. Les six étapes de la chaîne de destruction cybernétique comprennent la reconnaissance, la livraison, l'exploitation, le commandement et le contrôle, le mouvement latéral et l'exfiltration des données [41]. Les attaquants extérieurs effectuent principalement des attaques au stade des étapes de reconnaissance et de livraison dans le but de pénétrer un système cible. D'autre part, les attaquants internes visent à effectuer des attaques pour exfiltrer des informations confidentielles vers l'extérieur, des parties non autorisées.
- Comment tromper : la décision de conception sur la façon de tromper en utilisant la tromperie défensive est liée au type de technique de tromperie défensive à utiliser. Cependant, la technique spécifique à utiliser pour la tromperie défensive est plus liée à la technologie à utiliser pour réaliser la tromperie. Certains exemples de technologies de tromperie défensive incluent les pots de miel, les fichiers de miel, les jetons de miel, les faux correctifs, les fausses topologies de réseau, les fausses clés ou les fichiers appâts [9].

2.1.3. Principaux avantages des techniques de tromperie défensive

Il s’agit de :

- La tromperie défensive est relativement rentable par rapport à d'autres stratégies de défense, car son coût de déploiement est relativement faible tout en montrant une efficacité relativement élevée pour tromper les attaquants. Par exemple, les fichiers miel ou les jetons miel sont relativement simples à déployer et à maintenir, par rapport à d'autres mécanismes de cybersécurité traditionnels, tels que le contrôle d'accès ou la détection d'intrusion.
- La tromperie défensive fournit des services de défense complémentaires à d'autres mécanismes de défense hérités, tels que la détection ou la prévention des intrusions. Par exemple, les pots de miel sont bien connus comme un mécanisme de surveillance efficace qui peut fournir des fonctionnalités d'attaque supplémentaires pour améliorer la détection des intrusions ainsi que pour protéger de manière proactive un système contre les intrusions avant qu'elles ne lancent réellement des attaques contre des cibles.
- Divers types de techniques de tromperie défensive peuvent être déployés sur plusieurs couches de systèmes, telles que les couches réseau, système, application et données [9],

sans modifications importantes des architectures système existantes. La capacité de déploiement élevée des techniques de tromperie défensive offre également une flexibilité de conception ainsi qu'une capacité de défense en profondeur avec plusieurs couches de protection.

- Les techniques de cybertromperie automatisées, telles que l'obscurcissement des informations de session, le flux de données et l'obscurcissement du code du logiciel, ont considérablement amélioré la sécurité pour contrer les attaques automatisées.

2.1.4. Principales mises en garde des techniques de tromperie défensive

Puisqu'il est très difficile d'obtenir la motivation, l'intention et l'objectif d'un attaquant lors du lancement et de l'exécution d'une attaque particulière, il n'est pas trivial de choisir une stratégie de tromperie défensive optimale basée sur le risque estimé et avantage associé à une stratégie de tromperie défensive.

- La plupart des techniques Honey-X (par exemple, les pots de miel, les fichiers de miel, les jetons de miel) visent à tromper les attaquants pour qu'ils fassent des choix sous-optimaux ou médiocres lors du lancement d'attaques par de fausses informations. Cela peut introduire des procédures supplémentaires ou des protocoles pour les utilisateurs normaux ou un défenseur à ne pas confondre avec eux.
- En raison de la nature de la déception, l'efficacité de la tromperie défensive nécessite un processus continu de configuration, de reconfiguration et de mise en œuvre. Sinon, les adversaires seraient capables de distinguer facilement les dispositifs trompeurs à moins que la tromperie ne soit automatisée, comme les techniques d'obscurcissement des logiciels.
- Avec l'émergence d'attaquants intelligents, il existe une demande croissante pour des pots de miel hautement interactifs ou des stratégies de tromperie élaborées qui ne peuvent pas être facilement identifiées par les attaquants. Cela apporte un coût de défense plus élevé et plus de difficulté dans la gestion d'un système de défense qu'un système traditionnel sans mécanisme de tromperie défensive.

2.1.5. Domaines d'application de la tromperie défensive

2.1.5.1. Un réseau d'entreprise

Un réseau d'entreprise est un système commun configuré de manière homogène pour fonctionner dans une configuration statique [113]. Par conséquent, un attaquant peut facilement planifier et exécuter son attaque avec succès et par conséquent, pénétré dans le système sans éprouver de grandes difficultés, mais en utilisant des stratégies dynamiques pour vaincre les stratégies de défense.

2.1.5.2. Systèmes cyberphysiques généraux

Un système cyberphysique est un système qui peut fournir des communications entre les humains via des cybercapacités et des infrastructures physiques ou des plates-formes [42]. Le système cyberphysique a été perfectionné avec les cybercapacités de communication, de mise en réseau, de détection et d'informatique ainsi que les capacités physiques avec des matériaux, des capteurs, des actionneurs et du matériel. Le système cyberphysique se distingue uniquement des autres plates-formes en raison de la présence d'aspects cybernétiques et physiques des systèmes et de leur coordination[42].

Les systèmes cyberphysiques comprennent les réseaux de capteurs sans fil, l'Internet des objets (IoT), les réseaux définis par logiciel (SDN) et les systèmes de contrôle industriels (ICS) [43], [44]. Bien que nous abordions chacun d'eux dans des sections distinctes, nous incluons cette section en particulier pour discuter des techniques de tromperie défensive conçues pour les « systèmes cyber-physiques généraux » sans spécifier une plate-forme particulière.

2.1.5.3. Environnements Web basés sur l'infonuagique

Les applications Web basées sur l'infonuagique deviennent plus courantes et plus populaires que jamais pour traiter des tâches critiques qui placent la sécurité au premier plan. La tromperie défensive est l'une des directions prometteuses pour se défendre contre les attaques Web [45]. Les techniques de tromperie défensive peuvent être utilisées pour se défendre contre les attaques Web avancées qui ne peuvent pas être bien gérées par les mécanismes existants de détection ou de prévention des intrusions basés sur des signatures ou des anomalies [45].

2.1.5.4.Internet des objets (IoT)

Les environnements de réseau IoT deviennent très populaires et ont été reconnus comme l'un des CPS capables de fournir des services efficaces aux utilisateurs. L'IoT s'est également spécialisé dans des domaines particuliers, tels que l'Internet des objets de combat (IoBT) [46], l'Internet industriel des objets (IIoT) [47] ou l'Internet des objets de santé (IoHT) [48]. En outre, l'IoT englobe les idées de plusieurs environnements de réseau des systèmes cyberphysiques, tels que les réseaux de capteurs sans fil (WSN), les réseaux mobiles ad hoc (MANET) ou les environnements de ville intelligente composés de capteurs et d'autres machines intelligentes [49]. L'IoT a été populairement considéré comme une plate-forme principale de déploiement des technologies de cybertromperie [50].

2.1.5.5. Réseaux définis par logiciel (SDN)

Le paradigme SDN sépare le traitement du plan de données (par exemple, le transfert de paquets) du traitement du plan de contrôle (par exemple, les décisions de routage) [51]. Le protocole OpenFlow [52] agit comme une API entre les commutateurs du réseau et un décideur logiquement centralisé, appelé contrôleur OpenFlow. Dans ce protocole, les commutateurs réseau mettent en cache les règles de flux du plan de données.

Lorsqu'un commutateur reçoit un paquet et ne sait pas comment le transmettre selon ses règles mises en cache, il envoie une demande « d'élévation » contenant le paquet d'origine et une demande de guidage au contrôleur. Le contrôleur examine le paquet et envoie un ensemble de règles que le commutateur doit ajouter au cache du plan de données pour transmettre les paquets [53].

2.1.5.6. Réseaux sans fil

Les communications sans fil sont omniprésentes de nos jours et plus courantes que les communications filaires en raison de leur déploiement facile et de leur efficacité. Cependant, lorsque les ressources réseau sont rares, les contraintes de bande passante ou le support sans fil peu fiable deviennent les principaux problèmes à résoudre. Des techniques de tromperie défensive basées sur la théorie des jeux ou basées sur ML sont également proposées pour se

défendre contre divers types d'attaques exploitant les vulnérabilités des environnements de réseau sans fil.

2.1.5.7. Réseaux sociaux en ligne

En raison des grandes échelles d'OSN et de leur influence significative sur les aspects sociaux, économiques et politiques, les communautés d'IA et de ML ont reconnu de grands défis dans le développement de techniques de tromperie défensive dans les plates-formes OSN. Par conséquent, des efforts importants ont été déployés pour sécuriser OSN contre les utilisateurs malveillants (par exemple, les faux utilisateurs et les spammeurs) en développant diverses techniques de tromperie défensive sociales [54],[55], [56], [57].

2.1.6. Distinctions entre tromperie défensive et autres techniques de défense similaires

La tromperie défensive est souvent mentionnée comme défense de cible mobile ou vice versa. De plus, l'obscurcissement a été utilisé pour atteindre le même objectif que la tromperie défensive. Dans cette section, nous discutons de la manière dont les rôles et les natures de ces trois techniques diffèrent les unes des autres ainsi que de la manière dont elles se chevauchent dans leurs fonctionnalités et leurs objectifs[8].

2.1.6.1. Distinction avec la défense de cible mobile (MTD)

MTD est similaire à la tromperie défensive en ce qui concerne son objectif d'accroître la confusion ou l'incertitude des attaquants, ce qui a pour effet de dissuader l'escalade de leurs attaques à un niveau supérieur ou de faire échouer leurs attaques. Cependant, la principale distinction est que MTD n'utilise aucune fausse information pour tromper activement les attaquants, tandis que la tromperie défensive implique souvent l'utilisation de faux objets ou de fausses informations pour que les attaquants forment de fausses croyances et soient induits en erreur pour prendre des décisions d'attaque sous-optimales ou médiocres.

MTD associe sa fonctionnalité clé principalement à la façon de modifier les configurations du système de manière plus efficace et efficiente, tandis que la tromperie défensive implique davantage l'exploitation de la manipulation de la perception de l'attaquant. Si la tromperie défensive est bien déployée sur la base d'une formulation solide de manipulation

de la perception de l'attaquant, il peut être plus rentable que le déploiement de MTD. Ward et al. [58] considéraient le MTD comme faisant partie de tromperie défensive, tandis que Cho et al. [59] ont traité de tromperie défensive comme faisant partie de MTD. La distinction entre ces deux ne sont pas claires, car la tromperie défensive peut être déployée à l'aide de techniques MTD, telles que le caractère aléatoire ou les changements dynamiques des configurations du système (par exemple, l'attribution dynamique de leurres dans un réseau) alors qu'il peut ne pas utiliser de fausses informations tout le temps (par exemple, des appâts ou omissions, tels qu'une absence de réponse).

2.1.6.2. Distinction avec obscurcissement

Des techniques d'obscurcissement ont été utilisées par un attaquant pour obscurcir des logiciels malveillants [60], des virus métamorphiques [61] ou du code JavaScript malveillant [62].

Du point de vue d'un défenseur, des techniques d'obscurcissement ont été utilisées pour obscurcir des informations privées [63], ou du bytecode Java pour la décompilation [64]. Bien que les techniques d'obfuscation soient souvent mentionnées comme faisant partie des techniques de tromperie défensive, elles ont été étudiées comme un domaine de recherche distinct pendant des décennies.

Chan et Yang[64] ont mené une enquête approfondie sur les techniques d'obscurcissement améliorant la sécurité du système et ont discuté des principaux objectifs des techniques d'obscurcissement comme : (i) augmenter la difficulté de l'ingénierie inverse de la logique du programme ; (ii) atténuer l'exploitabilité des vulnérabilités par les attaquants ; (iii) empêcher les modifications par des parties non autorisées ; et (iv) cacher des données ou des informations. Bien que le concept de tromperie défensive ait gagné en popularité ces derniers temps alors que les techniques d'obscurcissement ont été appliquées pendant des décennies, il est évident que l'objectif de l'obscurcissement défensif est bien aligné avec celui de la tromperie défensive.

Cependant, une différence subtile réside dans le fait que la tromperie défensive implique des aspects cognitifs de la perception d'un attaquant, tandis que l'obscurcissement défensif est directement impliqué dans la réalisation des objectifs de sécurité (par exemple, la confidentialité, l'intégrité des données, la disponibilité). Les techniques d'obscurcissement défensif sont également utilisées en combinaison avec la diversification [64] (par exemple,

diversifier les systèmes, les configurations de réseau ou les composants), qui est un concept bien connu utilisé par les techniques MTD.

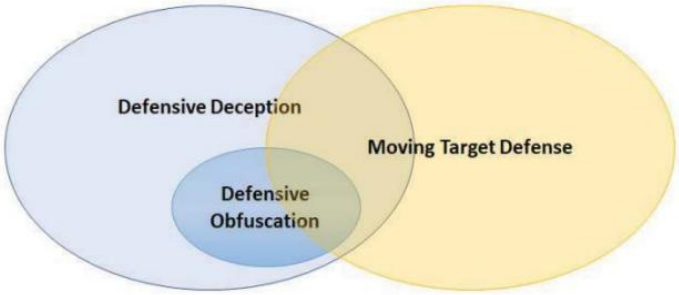

Figure 1 : Relations entre la tromperie défensive, la défense de cible mobile et l'obscurcissement défensif[8].

Sur la base de notre compréhension des fonctionnalités et des objectifs de tromperie défensive, MTD et de l'obscurcissement défensif, nous aimerions maintenir notre point de vue basé sur la figure 1, montrant les relations entre eux. Notre point de vue est qu'un obscurcissement défensif peut appartenir soit à MTD, soit à la tromperie défensive. En outre, il existe également une zone de chevauchement de MTD et de la tromperie défensive où certaines techniques défensives nécessitent l'utilisation des fonctionnalités des deux concepts.

2.1.7. Principaux types d'action de la cyberdéfense

La cyberdéfense basée sur la tromperie repose sur deux principaux types d'actions : **la simulation et la dissimulation** d'informations[65].

2.1.7.1. La dissimulation d'informations

Sont couramment utilisées pour cacher des informations. Les méthodes courantes comprennent le masquage, le reconditionnement et l'éblouissement[66].

2.1.7.1.1. Le masquage

Cet effet est résolu lorsque la tromperie est utilisée pour changer l'affichage d'un objet réel en quelque chose d'autre. Le masquage tente de cacher ou d'effacer des informations cruciales de la cible afin d'échapper à la détection. Les techniques de masquage des données, telles que le brassage, la substitution et le cryptage, sont couramment utilisées pour protéger les informations critiques, telles que les données personnelles identifiables.

2.1.7.1.2. Le reconditionnement

Fait référence à la transformation des caractéristiques clés de la cible de manière à ce qu'elles semblent non pertinentes ou différentes de l'original, dans l'espoir de détourner l'attention de l'attaque de la cible. Le saut d'adresse IP [66] est une technique récente de reconditionnement qui fait que les hôtes d'un réseau ont constamment des adresses IP différentes et que les flux du réseau ne peuvent pas être facilement suivis par les adversaires.

2.1.7.1.3. L'éblouissement

Brouille ou obscurcis les éléments clés de la cible sans les supprimer, afin de confondre la cible avec d'autres objets. Par exemple, l'obscurcissement des logiciels masque les sections critiques du code source ou du code en cours d'exécution et a été utilisé comme une pratique courante pour se défendre contre la rétro-ingénierie[67], [68].

2.1.7.2.La simulation de l'information

Elle implique des techniques de mimétisme, d'invention et de leurre [69]. L'objectif de la simulation est de créer et utiliser de fausses informations pour distraire et tromper les adversaires.

La simulation fournit une arène pour intégrer des environnements opérationnels avec des modèles de comportement humain pour analyser la vulnérabilité d'une structure de réseau compte tenu des différents types d'utilisateurs. Les simulations donnent un aperçu des effets possibles de l'utilisation et de la mauvaise utilisation des cybersystèmes [71].

La simulation n'implique généralement que la représentation des périphériques, des technologies et de la communication entre eux tout en manquant certains éléments essentiels d'un scénario cybernétique. D'autres ont proposé d'adopter une approche en direct, virtuelle et constructive pour la construction d'un environnement représentatif[72].

Elle implique des techniques de mimétisme, d'invention et de leurre [69]. L'objectif de la simulation est de créer et utiliser de fausses informations pour distraire et tromper les adversaires.

2.1.7.2.1. Mimétisme

Création d'une fausse entité par imitation en reproduisant les caractéristiques clés ou l'identité d'une entité réelle. Il s'agit de l'une des méthodes de simulation les plus utilisées. Par exemple, un pot de miel peut imiter un site Web réel[70].

2.1.7.2.2. L'invention

Création d'entités inexistantes avec des éléments clés et caractéristiques essentielles qui semblent réels. Il existe des différences subtiles entre la mimique et l'invention. Alors que l'exigence principale de la mimique est de créer une fausse entité qui doit ressembler à l'entité originale, l'invention crée une nouvelle entité qui semble réaliste.

2.1.7.2.3. Le leurre

Le leurre est une méthode largement utilisée parmi toutes les techniques de simulation. Un leurre est normalement un objet qui ressemble ou se comporte comme un objet réel. Il est utilisé pour détourner l'attention vers quelque chose de différent de la cible réelle. Par exemple, un serveur web leurre peut être utilisé pour attirer les attaquants. Le leurre, le mimétisme et l'invention sont souvent utilisés ensemble. Un serveur leurre peut imiter la fonction d'un serveur réel d'une organisation (par exemple, un pot de miel) ou être inventé dans le seul but de distraire l'attention.

En outre, nous avons :

- La Simulation (cyber) en direct : des acteurs réels interagissent avec des systèmes physiques d'ordinateurs réels connectés à des ordinateurs réels, et généralement isolés des réseaux.
- La Simulation virtuelle (cyber) : les acteurs réels interagissent avec l'émulation ou la simulation de réseaux ou d'acteurs émulés ou simulés interagissant avec des réseaux réels et généralement isolés.
- La Simulation constructive (cyber) : des acteurs simulés ou émulés interagissent avec des simulations ou des émulations de réseaux.

Et cette taxonomie est la plus fréquemment utilisée pour la tromperie dans le domaine de la sécurité de l'information.

La cybersécurité a été plus efficace en utilisant des outils de tromperie. Depuis les années 90, les responsables de la mise en œuvre des mesures de sécurité ont recours à deux grands outils de tromperie : les pots de miel et les honeynets. Les pots de miel sont de pièges ou de leurres utilisés dans les réseaux qui se présentent comme cible potentielle des attaquants. Ils se présentent généralement comme un serveur de grande valeur ou un actif réel de la cible. Les cyberattaquants tentent, sans le savoir, d'accéder aux informations apparemment utiles contenues dans les pots de miel. Pendant qu'ils sont occupés à leur attaque, ces pots de miel recueillent des informations sur les schémas de l'intrus et sur les moyens qu'il utilise pour obtenir des informations du système[73].

Lorsqu'un attaquant tente d'accéder au pot de miel et échoue, il se rend compte qu'une sécurité plus stricte doit être mise en œuvre en raison de l'importance de ce système pour l'entreprise. Cela le pousse à envahir le système à l'aide d'outils avancés et sophistiqués. La plupart des grandes organisations de haute sécurité du monde utilisent des pots de miel pour tromper les attaquants. Les informations recueillies par ces pots de miel sont utilisées par les chercheurs en sécurité pour concevoir des cadres de sécurités plus sûrs et plus sophistiqués contre les cyberattaques[74].

2.2. Section 2 : Pots de miel

2.2.1. Définition

Il existe plusieurs définitions des pots de miel. Certains les considèrent comme un système permettant de confondre les attaquants et d'inspecter leurs activités, tandis que d'autres les considèrent comme un système de surveillance de l'environnement technologique de détection des attaques ou systèmes réels formés pour être attaqués[75].

Un pot de miel est un système de défense actif complémentaire pour la sécurité du réseau. Il piège les attaques, enregistre les informations d'intrusion sur les outils et les activités du processus de piratage et empêche les attaques sortant du système compromis. Intégré à d'autres solutions de sécurité, un pot de miel peut résoudre de nombreux dilemmes traditionnels[76].

Dans [75], le pot de miel est défini comme étant un système de non-production, utilisé pour exploiter l'attaquant et remarquer les techniques et actions d'attaques.

Il peut être considéré comme un système informatique connecté à un réseau pour inspecter les vulnérabilités d'un ordinateur ou d'un réseau complet[77].

2.2.2. Fonctionnement

Un pot de miel est conçu pour ressembler à un véritable atout et contient généralement un ordinateur, des données et une application. Le système est isolé, aucun utilisateur n'est autorisé à accéder au pot de miel et il est étroitement surveillé. Tout accès au pot de miel est considéré comme hostile. C'est pourquoi chaque accès au système est évalué et étudié de près afin de recueillir des informations utiles à son sujet. Les attaquants sont occupés à enregistrer des activités pendant que le système collecte des informations à leur sujet. L'ensemble du mécanisme de cybersécurité détourne les attaquants des actifs réels. Les cybercriminels utilisent également des pots de miel pour distraire les chercheurs, en utilisant de faux pots de miel pour étudier le comportement des chercheurs en sécurité[37].

Cela permet aux intrus à utiliser des stratégies d'attaque différentes et plus efficaces. Les pots de miel frauduleux fonctionnent directement comme des leurres dans le système et diffusent de fausses informations, perturbant la recherche, ce qui aide finalement les cybercriminels dans leur objectif. Les chercheurs en sécurité sont désormais plus attentifs au

mécanisme de sécurité et sont désormais conscients des différentes stratégies utilisées par les cybercriminels pour différents types de cybercrimes[37].

2.2.3. Classification de pot de miel

2.2.3.1. En fonction de leur utilisation

Il existe deux types principaux de pots de miel. Les pots de miel de recherche et les pots de miel de production.

2.2.3.1.1. Pots de miel de production

Les pots de miel de production sont des pots de miel plus importants et efficaces placés à l'intérieur du réseau de production. Ils sont implantés avec l'IDS. Les données placées dans les pots de miel de production sont très similaires aux données réelles, mais pas utiles pour les attaquants. Pendant que les attaquants sont occupés par les pots de miel de production, les administrateurs étudient les schémas d'attaque et découvrent tout type de vulnérabilités dans le système de production [37].

2.2.3.1.2. Pots de miel de recherche

Les pots de miel de recherche sont les leurres qui collectent des informations sur les schémas d'attaque et fournissent les informations aux analystes de données. Les analystes de données placent des données uniques dans les pots de miel de recherche qui collectent et analysent les données sur les attaquants et leurs modèles. Il peut y avoir plus d'un attaquant essayant d'envahir un système à la fois, et les pots de miel de recherche tentent également de découvrir les connexions entre ces attaquants. Les pots de miel de recherche sont simplement conçus pour collecter des informations sur les activités de l'intrus, tel que défini ci-dessus, à des fins de recherche. Bien que toutes les informations collectées par d'autres pots de miel soient également utilisées à des fins de recherche, le pot de miel de recherche utilise ces informations pour aider les chercheurs à détruire les intrus[37].

Les pots de miel de recherche peuvent être mis en œuvre par les forces de l'ordre pour recueillir des informations sur les criminels et leurs activités. Ils sont conçus pour garder l'attaquant engagé autant que possible ; plus l'interaction est importante, plus l'information est bénéfique à des fins de recherche[37].

2.2.3.2. En fonction du niveau d'interaction

Il existe d'autres types de pots de miel qui sont utilisés dans différents domaines de sécurité : les Pots de miel purs, les pots de miel à faible interaction (LIHP), pots de miel à interaction moyenne (MIHP) et pots de miel à forte interaction (HIHP), Pots de miel à haute interaction basée sur VM

2.2.3.2.1. Pots de miel purs

Un pot de miel pur est une copie complète d'un système de production. Le pot de miel contient généralement des données en miroir qui montrent à l'attaquant que le pot de miel contient une grande quantité de données sensibles. L'attaquant croit avoir accédé à un véritable système de production et pourra désormais obtenir des informations utiles alors qu'en réalité, il est tombé dans un piège. Par exemple, un pot de miel pur est une instance d'un site Web d'investissement et bancaire où des fonctions complètes d'un site Web bancaire sont disponibles. Les fonctions de ce site Web sont si réelles qu'il est presque impossible pour un attaquant novice de remarquer qu'il a été dans un pot de miel. La seule différence entre un système de production et un pot de miel pur est que le pot de miel pur n'a pas la capacité de se connecter à une vraie base de données et à de vrais clients. Ce pot de miel collecte beaucoup de données sur l'attaquant et lui fait perdre son temps et ses ressources[37].

2.2.3.2.2. Les pots de miel à faible interaction (LIHP) :

Comme son nom l'indique, un pot de miel à faible interaction est conçu pour avoir moins d'interaction avec l'attaquant. Ces pots de miel contiennent des services de base fréquemment demandés par l'attaquant. Ce type de pot de miel est généralement facile à entretenir et moins risqué. [dix]. Il est relativement peu coûteux, car il n'offre pas les fonctionnalités comme le système de production. L'objectif fondamental d'un pot de miel à faible interaction est de distraire l'attaquant des ressources réelles. Ces pots de miel sont évolutifs et sont conçus pour modifier leurs capacités et être mis à niveau, ils ne fournissent aucun accès au système d'exploitation à l'attaquant. Les pots de miel à faible interaction sont généralement mis en œuvre dans les entreprises où la sécurité n'est pas un très gros problème. La plupart des organisations de domaine général à petite échelle ont des pots de miel à faible interaction où les attaquants sont également censés avoir des compétences intermédiaires[78].

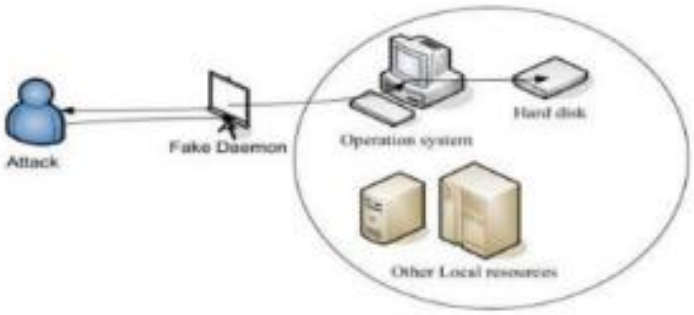

Fig. 2 : Pots de miel à faible interaction

2.2.3.2.3. Les pots de miel à interaction moyenne (MIHP) :

Le niveau d'interaction de ces pots de miel se situe entre les deux précédents. Un système d'exploitation n'est pas complètement émulé, mais un service de couche applicative entier est implémenté dans des pots de miel à interaction moyenne. Si un Honeynet contient à la fois des pots de miel à faible et à forte interaction, nous l'appelons un Honeynet à interaction moyenne. Comme exemple de pots de miel à interaction moyenne, nous pouvons citer le pot de miel proposé par [79], qui utilise Telnet et SSH pour en savoir plus sur les sessions d'attaque. [80] a également été proposé un réseau de miel comprenant un serveur pot de miel à faible interaction en façade et un serveur pot de miel à forte interaction en façade, qui est censé être un réseau de miel à interaction moyenne.

2.2.3.2.4. Les pots de miel à forte interaction (HIHP) :

Les pots de miel à interaction élevée offrent un environnement réaliste qui imite de près les systèmes et services réels. Ils facilitent une interaction étendue avec les attaquants potentiels, ce qui les rend inestimables pour recueillir des informations détaillées sur les techniques et stratégies d'attaque. Cependant, leur complexité nécessite une gestion prudente. Ces pots de miel émulent toutes les parties d'un système et tous ses services. Par conséquent, l'adversaire peut difficilement distinguer ces pots de miel d'un système réel. Cependant, si l'adversaire parvient à compromettre ces pots de miel, il peut les exploiter pour lancer de puissantes attaques contre le réseau. Par conséquent, la conception et la mise en œuvre de pots de miel à haute interaction nécessitent plus d'attention [81]. Un pot de miel à interaction élevée est une copie d'un système de production avec une capacité d'interaction maximale. Ces pots de miel ont généralement un temps de réponse lent afin que l'attaque de l'intrus puisse être

ralentie tout en lui faisant croire qu'il se trouve dans un système de production réel. Ces pots de miel sont utilisés dans des organisations où la recherche et le développement sont la principale priorité. Les organisations avec un niveau de sécurité moyen utilisent des pots de miel à interaction élevée [37].

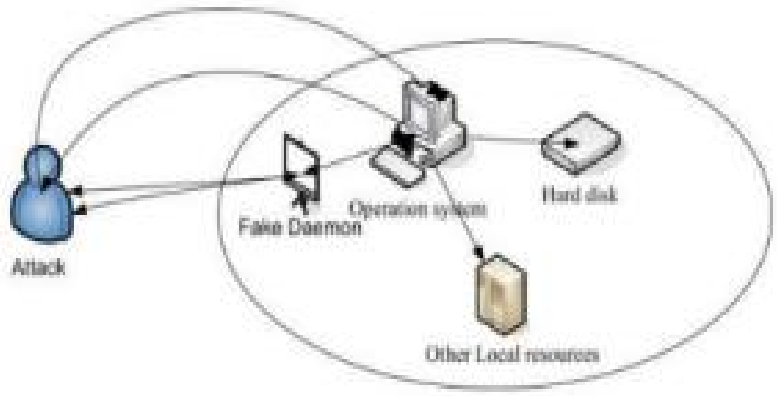

Fig. 3 : Pots de miel à forte interaction

2.2.3.2.5. Pots de miel à haute interaction basée sur VM

Les auteurs estiment que les pots de miel basés sur VM sont la technologie la plus récente et également la plus pérenne pour les HIHP. Pot de miel VM peut être construit à l'aide d'un logiciel de virtualisation comme VMWare Workstation, Qemu-KVM, Virtualbox, Parallels Bureau ou Linux en mode utilisateur.

C'est pourquoi il est devenu courant de ne différencier que les pots de miel bas et hauts, comme le recommande Lance Spitzner. Les pots de miel du serveur attendent que les attaquants initient la communication, tandis que les pots de miel du client recherchent activement des entités malveillantes potentielles et demandent une interaction.

2.2.3.3. En fonction de leurs méthodes de mise en œuvre

2.2.3.3.1. Pot de miel physique

Ce type de pot de miel est implémenté sur une machine distincte et possède une adresse IP unique. L'implémentation de cette classe est plutôt difficile et prend du temps, et sa sécurité nécessite une supervision et une attention particulières. Lorsque le réseau doit prendre en charge un espace d'adressage étendu, l'utilisation de pots de miel physiques n'est pas abordable. IoTPOT [82] est un exemple de pot de miel physique.

2.2.3.3.2. Pot de miel virtuel

Les pots de miel virtuels ne nécessitaient pas de machine physique dédiée pour leur mise en œuvre. Plusieurs pots de miel virtuels pouvaient être hébergés sur un seul serveur physique, ce qui en faisait un choix rentable et efficace. Le temps de mise en œuvre était souvent réduit et le développement du réseau était simplifié par rapport aux pots de miel physiques. Par exemple, HoneyIo4 [83]était un exemple de pot de miel virtuel.

Le développement des pots de miel repose désormais largement sur les progrès de la technologie de virtualisation. Les pots de miel virtuels offrent plusieurs avantages précieux : facilité de maintenance, configuration dynamique et anti-détection[84].

La virtualisation facilite la maintenance. Tout d'abord, grâce à la technologie de virtualisation, une machine physique peut héberger simultanément plusieurs pots de miel virtuels, ce qui peut améliorer considérablement l'efficacité des ressources. Ensuite, la tâche fastidieuse de déploiement des pots de miel à grande échelle est considérablement réduite grâce à l'utilisation de techniques de virtualisation qui ne prennent que quelques minutes à déployer plutôt que plusieurs heures sur des machines physiques, par exemple, le pot de miel physique conçu par Cliff Stoll en 1986 [85].

La virtualisation facilite également la configuration dynamique, qui est souvent utilisée pour réduire les temps de réponse à des événements spécifiques. Comme indiqué précédemment, les pots de miel dynamiques peuvent être utilisés pour cloner des systèmes de production et se synchroniser avec leurs modifications en temps opportun. Ils peuvent également être utilisés pour enquêter sur des intrusions en modifiant leur propre état en fonction des exigences de la recherche d'attaque [84].

L'anti-détection vise à éviter que le pot de miel ne soit détecté. La technologie de virtualisation offre plusieurs moyens de cacher à la fois le leurre et le capteur. Le capteur est tout d'abord caché, par l'introspection de la mémoire de la machine virtuelle qui facilite la surveillance « prête à l'emploi ». Cela améliore la capacité de surveillance furtive des HIH. Deuxièmement, comme l'approche de goupillage brutal est facile à détecter par un adversaire qualifié, les systèmes de pot de miel dynamiques redirigent souvent le trafic sortant vers le Honeynet pour l'anti-détection. Pour les pots de miel à fonction limitée, l'anti-détection se concentre sur le camouflage du fait que le leurre est un pot de miel [84].

2.2.3.4. En fonction de leurs uniformités de leurs leurres

2.2.3.4.1. Pots de miel homogènes

Ces pots de miel utilisent des leurres similaires dans le réseau. Ils n'utilisent qu'un seul type de piège pour tromper l'adversaire. Les performances de ces pots de miel sont limitées et ils ne peuvent détecter et retarder que des types d'attaques spécifiques. [86] a proposé une équipe de pots de miel homogènes formant un Honeynet qui collecte des informations utiles pour les IDS .

2.2.3.4.2. Pots de miel hétérogènes

Ces pots de miel utilisent des leurres hétérogènes et différents types d'outils de sécurité. Par conséquent, ils sont plus puissants pour détecter les attaques que le type précédent. Par exemple [87] conçut un réseau de pots de miel hétérogènes pour obtenir des données critiques auprès des adversaires.

2.2.3.5. En fonction de leur consistance

2.2.3.5.1. Pots de miel statiques

Les pots de miel statiques ont une certaine configuration et agissent toujours de la même manière pour différents adversaires et à différents moments. Leur comportement est fixe malgré différentes conditions de réseau et sous différents types d'attaques. Par conséquent, l'adversaire peut les suspecter et découvrir qu'il s'agit de leurres. La plupart des pots de miel mentionnés dans autoreftab:types sont des pots de miel statiques. Par exemple, ThingPot [88] est un pot de miel statique conçu pour les plateformes IoT.

2.2.3.5.2. Pot de miel dynamique

Ces pots de miel offrent une plus grande flexibilité par rapport aux pots de miel statiques. Ils peuvent s'adapter de manière dynamique aux changements d'état du réseau et modifier leur comportement en réponse aux attaques polymorphes. Par exemple [89] a introduit un pot de miel dynamique qui peut ajuster sa configuration dans diverses situations, notamment en réponse aux attaques par empreintes digitales.

2.2.3.6. En fonction du côté courant

Selon le côté courant des pots de miel, ils sont classés en deux types :

2.2.3.6.1. Pots de miel côté serveur

Ces pots de miel tentent d'identifier les vulnérabilités côté serveur et les fuites de sécurité qu'un serveur peut avoir. Ils tentent de protéger les serveurs critiques d'un réseau. L'adversaire qui lance une attaque contre les pots de miel côté serveur agit en tant que client. Par conséquent, les pots de miel côté serveur sont également des outils utiles pour détecter les clients malveillants dans un réseau. IoTPOT [82] est un pot de miel de serveurs qui vise à attirer les attaques basées sur Telnet.

2.2.3.6.2. Pots de miel côté client

Ces pots de miel sont conçus pour trouver des serveurs malveillants et également essayer d'identifier les vulnérabilités côté client. Un pot de miel côté client recherche les serveurs suspects, leur envoie une requête, puis analyse la réponse reçue. Si la réponse est anormale, les serveurs malveillants peuvent être détectés et le pot de miel peut identifier les vulnérabilités côté client qu'ils exploitent. Un pot de miel client est proposé par [90], qui agit comme un navigateur Web pour trouver les pages Web malveillantes et protéger les utilisateurs légaux.

2.2.3.7. En fonction du niveau d’activité

Le niveau d'activité des différents pots de miel n'est pas le même. À cet égard, les pots de miel sont classés en deux types :

2.2.3.7.1. Pots de miel passifs

L'objectif de ces pots de miel est de collecter des informations auprès de l'adversaire sans garantir la protection des autres systèmes. Un pot de miel passif attend que les connexions des adversaires enregistrent leurs informations et ne demande pas beaucoup d'efforts pour les attirer. Par exemple, [91] propose un pot de miel passif qui sert de service FTP pour détecter les logiciels malveillants .

2.2.3.7.2. Pots de miel actifs

Les pots de miel actifs sont conçus pour attirer les adversaires potentiels loin des systèmes critiques en les recherchant et en s'engageant de manière proactive avec eux. Contrairement aux pots de miel passifs, les pots de miel actifs utilisent diverses méthodes pour attirer et tromper les attaquants potentiels [92]. Un exemple de pot de miel actif peut être trouvé dans le travail proposé par [93].

3.2.4. **Autres types de pots de miel**

3.2.4.1. **Honeywebs** [94]

El-Kosairy et Azer ont proposé un cadre de serveur Web qui fournit un pare-feu d'application Web avec un Honeyweb pour détecter les trafics malveillants et transmettre les trafics suspects aux serveurs de pots de miel. Dans la définition du jeu, l'action principale d'un défenseur consiste à implémenter une machine virtuelle (VM) pour consommer les efforts et les ressources de l'attaquant.

Avantages et inconvénients[8] : Les toiles de miel sont rarement utilisées, en particulier sur la base d'approches basées sur la théorie des jeux. Ils sont utilisés pour aider les pots de miel avec des jetons de miel ou des fichiers de miel. Développer de fausses pages Web coûte moins cher alors que son rôle ressemble plus à un assistant pour soutenir les pots de miel. Par conséquent, les toiles de miel ne peuvent pas être utilisées sans d'autres technologies défensives.

3.2.4.2. **Pot de miel industriel**

L’environnement industriel constitue un champ d’application intéressant pour le HP. Leur système support Modbus, FTP, Telnet, http est basé honeyd. quatre ans plus tard, le premier pot de miel industriel le plus connu est Conpot 71 publié par Rist en 2013.

3.2.4.3. **Honeywords**

Proposé par Jules et Rivest pour confondre les attaquants lorsqu'ils craquent un fichier de mot de passe haché volé en cachant le vrai mot de passe parmi une liste de « faux ». Leur schéma augmente les bases de données de mots de passe avec un (N - 1) faux identifiant supplémentaire. Si la base de données est volée et fissurée, les attaquants sont confrontés à N mots de passe différents à choisir parmi lesquels un seul d'entre eux est le bon. Cependant, s'ils utilisent l'un des faux, le système déclenche une alarme alertant les administrateurs système que la base de données a été fissurée [29].

3.2.4.4. **Honeynet**

Est un exemple de pot de miel à haute interaction. Honeynet est un réseau de deux pots de miel ou plus qui travaillent ensemble pour tromper et piéger les attaquants[95]. Considérer un attaquant et un système de honeynets (c'est-à-dire un défenseur) comme des acteurs dans un jeu stratégique non coopératif. Le cadre est basé sur de vastes jeux d'informations imparfaites[8].

3.2.4.5. **Honeyd**

Honeyd est un outil populaire développé par Niels Provos, qui offre un moyen simple d'émuler des services offerts par plusieurs machines sur un seul PC. Il s'agit d'un pot de miel à faible interaction [114].

Les pots de miel, comme toute technologie de sécurité réseau, doivent être configurés. Traditionnellement, qu'il s'agisse d'une interaction faible, moyenne ou élevée des pots de miel, ils sont normalement configurés manuellement. Le pot de miel statique actuel est confronté aux mêmes exigences. Même si la configuration et le déploiement n'étaient pas suffisants, nous avons besoin de quelqu'un pour maintenir le pot de miel et cela peut être plus difficile. C'est pour résoudre ce problème que l'idée des pots de miel dynamique est née. Cela semble peut-être une solution plug and play. Nous pouvons simplement le brancher sur le réseau et le pot de miel fait tout le travail à notre place [98].

Applications principales :

- Distraction : Honeyd est utilisé principalement à deux fins. Si un réseau n'a que trois serveurs réels, mais qu'un serveur exécute Honeyd, le réseau apparaîtra exécutant des

centaines de serveurs pour un pirate. Le pirate devra alors faire plus de recherches (éventuellement par le biais de l'ingénierie sociale) afin de déterminer quels serveurs sont réels, ou le pirate peut se faire prendre dans un pot de miel. Dans tous les cas, le pirate sera ralenti ou éventuellement attrapé.

- Pot de miel : Honeyd tire son nom de sa capacité à être utilisé comme pot de miel. Sur un réseau, tout le trafic normal doit se faire uniquement vers et depuis des serveurs valides. Ainsi, un administrateur réseau exécutant Honeyd peut surveiller ses journaux pour voir s'il y a du trafic vers les hôtes virtuels configurés par Honeyd. Tout trafic allant vers ces serveurs virtuels peut être considéré comme hautement suspect.

Les pots de miel dynamiques sont tous très demandés, notamment le nombre de pots de miel à déployer, la manière de les déployer et leur apparence qui déterminera automatiquement afin qu'ils puissent se fondre dans l'environnement de l'organisation. Mieux encore, les pots de miel déployés peuvent changer et s'adapter à l'environnement de l'organisation. Lorsque nous changeons de routeur dans notre réseau, les routeurs de pot de miel changent également. L'objectif final est une application et une solution où nous nous connectons simplement à notre réseau, puis il apprendra l'environnement, déploiera les pots de miel et capturera les dommages causés par l'attaquant [96].

3.2.5. **Anti-Honeypot**

Comme tout logiciel, les pots de miel sont sujets à des bogues logiciels. Les pots de miel à faible et moyenne interaction sont généralement construits de manière compacte. Ils visent à simuler un type très spécifique de système, généralement ciblé pour détecter un type spécifique d'attaques. Le pot de miel à faible interaction le plus simple est un simple programme de journalisation écoutant sur le port 22 (protocole SSH) et enregistrant toutes les données reçues dans un fichier journal. Cela facilite le développement, le déploiement et la maintenance des pots de miel. Cependant, cela les rend également plus faciles à reconnaître.

Les attaquants construisent des empreintes digitales complexes basées sur la notion de services connexes. Si un hôte réseau A offre les services X et Y, il est courant que l'hôte A offre également les services W, E et R. Un hôte réseau réel aurait alors l'ensemble des services (ou un grand sous-ensemble) tandis que le pot de miel pourrait n'offrir que X et Y. Cela conduit à une détection facile par les attaquants [83].

Les pots de miel peuvent être considérés comme un moyen de capturer et d'analyser les comportements malveillants et le trafic sur les systèmes informatiques en réseau. Pour y parvenir, les HIHP sont souvent implémentés comme de vrais systèmes d'exploitation avec de vrais logiciels. Ce système d'exploitation est soit exécuté sur une plate-forme de virtualisation, soit sur du matériel réel. Dans le premier cas, les outils d'introspection et de collecte de données sont intégrés au pot de miel. Dans ce dernier, certains crochets sont utilisés pour surveiller le système d'exploitation virtualisé. Dans les deux cas, ces systèmes de surveillance laissent des traces dans le système. Des exemples de telles traces sont des composants matériels irréguliers, un comportement inhabituel (par exemple, de grandes latences d'appel système) et des indices de logiciel de surveillance[97].

3.2.6. **Construction du Pot de miel**

Un pot de miel peut être construit sous de nombreuses formes, soit sous la forme d'une machine physique, d'une machine virtuelle (VM) ou d'un hôte virtuel émulé à l'aide d'un honeyd. Un pot de miel physique est une véritable plate-forme informatique qui possède sa propre adresse IP valide[95].

Par exemple, un hôte virtuel peut être programmé pour contenir un script Perl qui émule un service Sendmail [95].

3.2.7. **Avantages et inconvénients des pots de miel** [99]

Les avantages et les inconvénients des pots de miel ont été évalués par plusieurs chercheurs [100], [101], [102]. Nous résumons les différents aspects comme suit.

3.2.7.1. **Avantages**

- Collecte de données précieuses : Les pots de miel collectent des données qui ne sont pas polluées par le bruit des activités de production et qui sont généralement de grande valeur. Cela rend les ensembles de données plus petits et l'analyse des données moins complexe.
- Indépendamment de la charge de travail : Cela signifie qu'ils sont indépendants de la charge de travail subie par les systèmes de production des stratégies inconnues et des exploits zéro-day seront identifiés.

- Réduction des faux positifs et négatifs : les pots de miel client vérifient les attaques en détectant les changements d'état du système.
- Flexibilité : cela signifie que des outils de pot de miel bien ajustés peuvent être utilisés pour des tâches spécifiques, ce qui pourrait en outre réduire la charge redondante.

3.2.7.2. **Inconvénients**

- Champ de vision limité : tant que les attaquants n'envoient aucun paquet au pot de miel, il ne sera pas au courant de toute activité non autorisée sur les systèmes de production.
- Les pots de miel à faible interaction émulent des services tardifs : ce qui signifie que leurs services peuvent se comporter différemment des services réels, qui peuvent être utilisés pour prendre les empreintes digitales des pots de miel et, par conséquent, les détecter.
- Risque pour l'environnement : comme nous l'avons vu, plus le niveau d'interaction est élevé, plus le risque d'utilisation abusive est élevé.
- En supposant que nous réussissions à attirer l'attaquant dans notre pot de miel, nous devons être en mesure de le tromper en permanence sur le fait qu'il se trouve dans le système réel. Dans cette situation, les attaquants sont dans une position où ils ont la capacité de mener des activités de contre-tromperie en se comportant d'une manière différente de ce qu'ils feraient dans un environnement réel [103].

3.2.8. **Placement des pots de miel**[96]

3.2.8.1. **Derrière le pare-feu**

La grande majorité des pots de miel sont installés à l'intérieur des pare-feu "Fig4". Afin qu'ils puissent être mieux contrôlés, bien qu'il soit possible de les installer à l'extérieur des pare-feu[96].

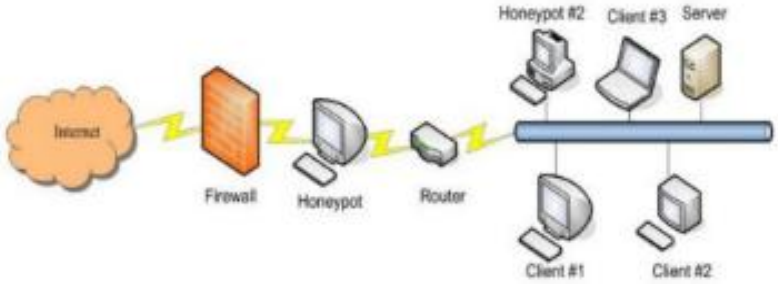

Figure 4 : Pot de miel derrière le pare-feu[96]

Par pare-feu supplémentaire. De même, la signature IDS ne peut pas générer d'alarme chaque fois que le pot de miel est attaqué ou scanné. Le plus gros problème survient lorsque le pot de miel interne est compromis par un attaquant externe. En conséquence, ce trafic ne sera plus arrêté en tant que trafic de pot de miel ; ainsi, l'attaquant a accès au réseau interne via le pot de miel. En fait, la principale raison de placer un pot de miel à l'intérieur du pare-feu est d'analyser les attaquants internes. De plus, avec un pot de miel interne, il est également possible de trouver une mauvaise configuration du pare-feu[104].

3.2.8.2. **Devant le pare-feu**

Cependant, l'utilisation d'un système compromis devant le pare-feu réduit les risques du réseau interne. Un rôle joué par le pare-feu dans un pot de miel est à l'opposé de celui d'un pare-feu normal. En effet, le pare-feu du pot de miel limite ce que le système renvoie et permet à tout le trafic d'entrer depuis Internet, au lieu de cela, limite ce qui entre dans un système depuis Internet. En tentant un pirate informatique dans un système, un pot de miel sert plusieurs objectifs : l'administrateur peut observer le pirate informatique utiliser les vulnérabilités du système ; par conséquent, la compréhension des faiblesses peut être utile pour reconcevoir le système afin d'atteindre notre objectif[96].

.

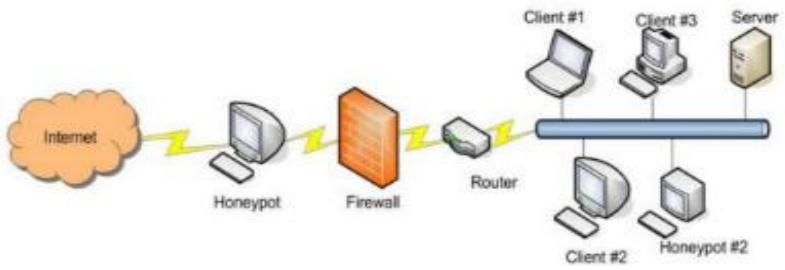

Figure 5 : Pot de miel devant le pare-feu[96]

3.2.9. **Considérations juridiques et éthiques**

La légalité des pots de miel demeure en discussion. Le recours aux pots de miel pour lutter notamment contre la distribution de contenus protégés par le droit d'auteur ainsi que le risque de complicité et de la collecte de données à caractère personnel demeurent encore un sujet à débattre.

La gestion des risques liés au déploiement des pots de miel implique également la prise en compte des implications légales et réglementaires, telles que les lois sur la confidentialité

des données, les restrictions sur l'utilisation de l'attirail de sécurité informatique, et les accords de partage d'informations avec les autorités compétentes.

CHAPITRE III

ÉTATS DE L'ART

Les mécanismes de sécurité, tels que le contrôle d'accès et la détection d'intrusion, aident à faire face aux menaces extérieures et intérieures, mais résistent de manière inadéquate aux attaquants qui peuvent subvertir les contrôles ou lancer de nouvelles attaques. La tromperie est une ligne de défense distincte visant à contrecarrer les attaquants potentiels.

Depuis que les avantages de tirer parti des idées fondamentales de la tromperie défensive ont été réalisés dans la communauté de recherche sur la cybersécurité, une bonne quantité de recherches sur la tromperie défensive a été explorée. Dans cette partie du mémoire, nous allons présenter de manière non exhaustive quelques travaux traitant le problème de placement de pots de miel dans la tromperie défensive en tenant compte de différentes approches.

Notre recherche se base sur des travaux antérieurs en matière de cybertromperie, de cybersécurité, la théorie de jeux, ainsi que sur les placements de pots de miel.

a. Pot de miel

Dans [1], les auteurs développent une approche de tromperie en deux phases : une politique proactive d'allocation de pots de miel, suivie d'une déception réactive s'ajustant en fonction des alertes IDS. L'approche décrite dans [2] propose un algorithme évolutif pour allouer les pots de miel dans un graphe d'attaque, modélisé comme un jeu à somme nulle entre un défenseur et un attaquant. Le travail [3] introduit un modèle de jeu dynamique pour analyser l'interaction entre pots de miel et attaques APT dans l'IIoT. Dans [4], les auteurs utilisent un cadre théorique de jeu pour intégrer les pots de miel comme mécanismes de tromperie dans les environnements IoT.

b. Théorie de jeu

Les travaux [5] et [6] montrent que la théorie des jeux est fréquemment mobilisée pour traiter des problématiques de sécurité dans l'IoT. Dans [13], une revue détaillée des modèles de jeux appliqués aux systèmes de sécurité logiciels illustre la diversité des applications existantes. Les auteurs de [14] exploitent l'équilibre de Nash pour identifier les tentatives

d'attaque et évaluer l'efficacité des stratégies défensives. Plusieurs travaux, dont [15] et [16], démontrent l'efficacité de la théorie des jeux pour la cybertromperie, notamment dans l'élaboration de stratégies de tromperie adaptatives.

Le travail [17] met en place un modèle à somme nulle où l'allocation des pots de miel est déterminée par l'équilibre de Nash sur un graphe d'attaque. Dans [18], une approche de cybertromperie à deux niveaux est proposée, optimisant le placement des leurres en tenant compte de la topologie du réseau et de l'importance des nœuds.

D'autres contributions, telles que GBO (Game-Based Optimizer) proposé dans [21], modélisent le comportement de l'adversaire via la compétition de Stackelberg pour déterminer le nombre optimal de pots de miel. Le modèle TBM (Tolerance-Based Model) [22] évalue le nombre de pots de miel permettant de rendre l'attaque non rentable pour l'adversaire.

Enfin, plusieurs travaux se sont intéressés aux liens entre jeux dynamiques et processus décisionnels markoviens. [27] montre que les politiques optimales d'un processus de décision markovien peuvent être calculées efficacement. À l'inverse, Conitzer et Sandholm ont démontré la complexité du problème d'existence d'un équilibre de Nash pur dans un jeu stochastique, qui est PSPACE-difficile [28].

Notre travail s'appuie notamment sur [105], qui propose une diversification logicielle et un placement optimisé sous contrainte de ressources. Nous étendons cette direction en ajoutant un aspect clé et encore peu étudiée : la redondance hétérogène des pots de miel, couplée à un cadre théorique rigoureux.

c. Cybersécurité et cybertromperie

Almeshekah et Spafford [7] analysent les processus de tromperie à travers trois phases (planification, mise en œuvre, suivi), mais leur approche reste limitée à un spectre réduit de types d'attaquants. Rowe et Rrushi [8] classifient les techniques de tromperie (camouflage, faux, retards, etc.) et évaluent leur efficacité, mais l'intégration de la théorie des jeux y est marginale.

Han et al. [9] proposent une taxonomie complète des techniques de tromperie selon l'objectif, la couche et le mode de déploiement. Toutefois, leur revue des modèles théoriques est partielle. Pawlick et al. [10] fournissent une analyse approfondie des approches de tromperie

théorique (Stackelberg, Nash, jeux de signalisation), mais leur travail ne couvre pas les environnements cyber réels ni les approches combinant ML et théorie des jeux. Enfin, Lu et al. [11] présentent une synthèse récente, mais centrée sur une fraction limitée de la littérature existante.

Pawlick et al. [10] ont mené une enquête approfondie sur les taxonomies de la tromperie défensive et les techniques de tromperie défensive de la théorie des jeux qui ont été utilisées pour la cybersécurité et la confidentialité. Les auteurs ont discuté des six principaux types différents de catégories de déception : perturbation, défense de cible mobile, obscurcissement, mélange, miel-X et engagement de l'attaquant. Leur article a passé en revue 24 articles publiés entre 2008 et 2018 et a défini des taxonomies associées pour développer leur propre classification des techniques de tromperie défensive de la théorie des jeux. Ce travail est intéressant pour traiter la défense des cibles mobiles et l'obscurcissement comme des sous-catégories sous tromperie défensive. Cet article a discuté des approches théoriques des jeux courantes utilisées pour développer des techniques de tromperie défensive, telles que Stackelberg, Nash et les théories des jeux de signalisation. Cependant, l'étude et l'analyse des techniques défensives de la théorie des jeux existantes menées dans cet article se limitent à l'analyse de la théorie des jeux sans tenir compte des environnements de réseau réalistes où les techniques de tromperie défensive basées sur ML ou une combinaison de ces deux (c'est-à-dire la théorie des jeux et ML) peuvent fournir des informations plus utiles et des directions de recherche prometteuses.

Récemment, Lu et al. [11] ont mené une brève enquête basée sur les processus de tromperie défensive consistant en trois phases : la planification de la déception, la mise en œuvre et le déploiement de la déception, et le suivi et l'évaluation de la déception. Les auteurs ont discuté des techniques de tromperie basées sur la dissimulation d'informations pour cacher les informations réelles et la simulation d'informations pour se concentrer sur les attaquants. Ce travail a brièvement discuté de la théorie des jeux dans la tromperie défensive et s'est principalement concentré sur la discussion des défis et des limites de la recherche actuelle. Cependant, seule une petite fraction de la littérature a été incluse. De plus, cet article n'aborde pas les approches tromperie défensive basées sur le ML.

Des travaux tels que celui de Virvilis et al. [12] se concentrent sur des techniques partielles de tromperie visant à atténuer les attaques APT.

Synthèse comparative

Pour clarifier ce que notre approche apporte de nouveau, nous comparons qualitativement les contributions existantes selon quatre critères pertinents :

1. Placement optimisé des pots de miel
2. Diversification logicielle
3. Redondance des pots de miel (continuité en cas de défaillance)
4. Modélisation théorique (jeu à somme nulle + Nash)

Travaux existants	Placement optimisé	Diversification	Redondance	Modélisation théorique
[1], [2], [3], [4]	Oui (partiel)	Non	Non	Oui
[17], [18], [21], [22]	Oui	Parfois	Non	Oui
[9], [10], [11]	Non (revues)	Mentionnée	Non	Oui (revue)
[105]	Oui	Oui	Non	Oui
Notre approche	Oui (optimisation Nash)	Oui (diversification logicielle)	Oui (nouveau)	Oui (jeu à somme + stratégies mixtes

Tableau 1 : Tableau de synthèse comparative

Contrairement aux travaux antérieurs qui se concentrent soit sur le placement, soit sur la diversification, notre approche introduit la combinaison inédite de : Diversification logicielle, Redondance hétérogène (continuité malgré contournement ou panne), Optimisation théorique des stratégies mixtes par équilibre de Nash, Analyse d'impact sur la récompense du défenseur.

C'est cette fusion cohérente de mécanismes diversifiés, redondants et formellement optimisés qui constitue l'originalité et la valeur ajoutée majeure de votre contribution

CHAPITRE IV

MÉTHODOLOGIE ET EXPÉRIMENTATION

Dans ce chapitre, nous formulons le modèle de jeu entre le défenseur et l'attaquant, et le modèle du réseau.

Dans la conception des mécanismes de cybersécurité, de nombreux algorithmes d'optimisation ont été proposés pour améliorer la répartition des ressources de défense et le placement optimal des pots de miel. Par exemple les algorithmes génétiques exploitent les principes de l'évolution naturelle, croisement et mutation pour rechercher des solutions quasi optimales dans un vaste espace d'actions. Les algorithmes de colonie de fourmis permettent d'optimiser des chemins ou des topologies de réseau en simulant la communication par phéromones. D'autres méthodes, telles que le recuit simulé ou la recherche tabou, sont également utilisées pour éviter les minima locaux et améliorer la stabilité des solutions.

Un autre modèle est proposé par [19] pour trouver le nombre optimal de pots de miel sans les paramètres perçus par l'adversaire. Le défenseur du réseau dans le modèle précédent doit disposer d'informations sur les coûts et les avantages perçus par l'adversaire. Un filet de miel est modélisé comme une urne contenant trois types de perles de couleurs différentes. Nous appelons ce modèle URN. Lorsque l'adversaire attaque un hôte, une perle est retirée de l'urne. Ce modèle suppose que l'adversaire réussit s'il peut lancer une attaque contre au moins un des hôtes vulnérables. Le défenseur du réseau peut calculer le nombre de pots de miel requis pour quantifier le seuil de taux d'attaque réussi de l'adversaire.

Pour certains adversaires professionnels, se connecter à quelques pots de miel ne révèle pas leur identité. Ils acceptent donc de communiquer avec un nombre précis de pots de miel. [20]ont proposé un modèle d'urne similaire, appelé URNt (URN avec seuil), qui considère également un nombre acceptable de connexions aux pots de miel. La probabilité d'une attaque réussie est calculée, puis le défenseur du réseau peut prendre la meilleure décision pour choisir la méthode de tromperie appropriée en examinant la probabilité de victoire de l'adversaire. Les modèles d'urnes précédents ne prennent pas en compte le processus de sondage effectué par l'adversaire avant de lancer les attaques.

[23]ont mené de recherche pour détecter les valeurs aberrantes à partir des données de pot de miel SSH collectées à l'aide de la technique d'optimisation par essaim de particules qui appartient à la catégorie des méthodes de détection des valeurs aberrantes basées sur les clusters. Avec des expériences et des résultats, l'optimisation par essaim de particules montre les meilleurs résultats dans la détection des valeurs aberrantes et a la meilleure fonction de coût par rapport à d'autres algorithmes basés sur des clusters tels que l'algorithme génétique et l'algorithme d'évolution différentielle.

[24] ont utilisé l'algorithme de Pareto pour optimiser la procédure de résolution de l'équilibre de Nash, ce qui réduit la complexité de calcul. [25] ont proposé une approche de théorie des jeux pour atteindre un système de défense coopérative économe en énergie. Spécifiquement, pour accroître la sécurité des données dans le cloud de capteurs, un système à deux couches problème de détection assistée par passerelle et de décision de défense impliquant plusieurs systèmes de détection d'intrusion utilisant un jeu évolutif est formulé, qui optimise la stratégie de détection pour réduire la consommation d'énergie et réduire les messages d'alerte.

[26] ont analysé la stratégie de défense optimale dans différents états de sécurité avec le processus de décision de Markov et utilisé la programmation linéaire algorithme pour calculer la solution optimale, mais Huang et al a traité les stratégies d'attaque-défense et les gains comme étant publics connaissances, et il n'existe qu'un seul type pour l'attaquant/ défenseur respectivement. En substance, il appartient toujours à la catégorie de jeu d'information complète.

Cependant, ces approches classiques restent centrées sur une optimisation statique. Elles cherchent une configuration optimale sans considérer la présence d'un adversaire intelligent qui observe, apprend et adapte sa stratégie d'attaque. Or, dans le domaine de la cybertromperie, le défenseur n'affronte pas un environnement fixe, mais un attaquant rationnel qui cherche à maximiser son propre gain tout en minimisant celui du défenseur.

C'est précisément dans ce contexte que la théorie des jeux s'impose comme un outil puissant. Elle permet de modéliser la dualité stratégique entre l'attaquant et le défenseur sous forme d'un jeu à somme nulle, où les actions de l'un influencent directement les gains de l'autre. Le défenseur doit alors élaborer une stratégie proactive et adaptative, afin de rendre son comportement imprévisible et de dissuader l'attaquant.

5.1. Modèle Réseau

Le réseau est représenté par un graphe $G = (V, E)$, où :

- V est l'ensemble des nœuds, incluant des nœuds stratégiques.
- E est l'ensemble des arêtes représentant les connexions entre nœuds.
- Les pots de miel redondants sont placés devant les nœuds critiques.

Paramètres **:**

- H: Ensemble des pots de miel.
- $V = V_s \cup V_s$: Nœuds simples et stratégiques.
- SW : Logiciels déployés sur les pots de miel.
- B : l'ensemble fini de tous les exploits disponibles pour l'attaquant
- R_h : Nombre de pot de miel redondants pour chaque nœud.
- $SWV[s, v]$: Probabilité qu'un logiciel s ait une vulnérabilité v.
- $P[s, v]$: Indique si l'exploit b peut exploiter la vulnérabilité v.

Matrice d'affectation des logiciels :

$$HSW[h, s] = \begin{cases} 1, & \textbf{si le logiciel } s \textbf{ est de´ploye´ sur le pot de miel } h \\ 0, & \textbf{sinon} \end{cases}$$

5.2. Modèle du Défenseur

Le défenseur protège les nœuds stratégiques en déployant des pots de miel diversifiés et redondants. Il choisit les logiciels pour chaque pot de miel.

Stratégie du défenseur :

$$\boldsymbol{A_d} = \{ y_{s_h} \in [0,1] \mid \sum_{s_h \in SW} y_{s_h} = 1 \}, \quad \forall\, h \in H$$

- y_{s_h} : Probabilité d'affectation du logiciel s_h au pot de miel h.
- $R_h \geq 1$: Redondance assurée par plusieurs pots de miel pour chaque nœud stratégique.

Coût de la défense :

Le coût dépend des logiciels choisis :

$$C_D(a_d) = \sum_{h \in H} \sum_{h \in H} c_s y_{s_h} \quad (4.1)$$

5.3. Modèle de l'Attaquant

L'attaquant cherche à compromettre les nœuds stratégiques en utilisant des exploits contre les vulnérabilités.

Stratégie de l'attaquant :

$$\boldsymbol{A_a} = \{x_b \in [0,1] \mid \sum_{b \in B} x_b = 1\}$$

- x_b : Probabilité d'utiliser l'exploit b.

Coût de l'attaque :

$$C_D(a_d) = \sum_{b \in B} c_b x_b \quad (4.2)$$

5.4. Formulation du modèle de jeu

Un jeu à deux joueurs à somme nulle est défini comme un tuple Γ (K, A, R), où :

- K = {Défenseur, Attaquant} : Ensemble des deux joueurs.
  - d : Défenseur.
  - a : Attaquant.
- Espace d'actions $\mathcal{A} = \mathcal{A}_d$ x $\mathcal{A}_a$
  - $\mathcal{A}_d$: Espace d'actions du défenseur, $\boldsymbol{A_d} = \{y_{s_h} \in [0,1] \mid \sum_{s_h \in SW} y_{s_h} = 1\}$

$\mathcal{A}_a$: Espace d'actions de l'attaquant,

$$\boldsymbol{A_a} = \{x_b \in [0,1] \mid \sum_{b \in B} x_b = 1\}$$

x_b : Probabilité d'utiliser l'exploit b.

5.5. Modélisation avec Redondance

Nous supposons que chaque nœud stratégique h est protégé par plusieurs pots de miel H_h, chacun ayant un logiciel différent.

- H_h : Ensemble des pots de miel placés sur le nœud h.
- $SWV[b, v]$: Probabilité qu'un logiciel s_h possède la vulnérabilité v.
- $P[s_{h'}, v]$: Indique si l'exploit b peut attaquer la vulnérabilité v.
- R_h=$|H_h|$: Nombre de pot de miel redondants sur le nœud h.
- x_b : Probabilité que l'attaquant utilise l'exploit b.
- y_{sh} : Probabilité que le défenseur affecte le logiciel sh au pot de miel h.

5.6. Matrice de Gain avec Redondance

L'attaquant doit franchir tous les pots de miel avant de compromettre le nœud stratégique h, ce qui réduit ses chances de succès. Le gain de l'attaquant devient :

$$U_A(a_a, a_d) = \sum_{b \in B} x_b \sum_{h \in H} P[b, v_h] \cdot \left(1 - \prod_{h' \in H} (1 - SWV[s_{h'}, v_h])\right) \cdot V_h - c_b \cdot x_b \quad (4.3)$$

Où :

- $\prod_{h' \in H}(1 - SWV[s_{h'}, v_h])$: représente la probabilité que tous les pots de miel résistent à l'attaque.
- Plus R_h augmente, plus cette probabilité est élevée, réduisant les gains de l'attaquant.

Le gain du défenseur est alors :

$$U_D(a_d, a_a) = -U_A(a_a, a_d) \sum_{\mathrm{h} \in H} x_b \sum_{\mathrm{h}' \in H} c_{s_{h'}} y_{s_{h'}} \quad (4.4)$$

5.7. Optimisation linéaire pour l'équilibre de Nash

L'équilibre de Nash est atteint lorsqu'aucun joueur ne peut améliorer son gain en modifiant unilatéralement sa stratégie, c'est-à-dire lorsque le défenseur et l'attaquant adoptent des stratégies mixtes optimales qui se stabilisent.

Nous allons utiliser la programmation linéaire définie précédemment pour trouver cet équilibre.

5.7.1. Définition de l'Équilibre de Nash

L'équilibre de Nash implique que :

- L'attaquant choisit une distribution optimale x_b des exploits pour maximiser son gain minimal.
- Le défenseur choisit une distribution optimale y_{s_h} des logiciels sur les pots de miel pour maximiser son propre gain minimal.
- Les gains des deux joueurs sont égaux en équilibre, soit $u_A^* = u_D^*$.

5.7.2. **Formulation en Problème de Programmation linéaire**

Nous avons deux programmes linéaires :

- **Programme de l'Attaquant** *(Maximisation de son gain minimal)*

$$\max U_A$$

Sous contraintes :

$$U_A \leq \sum_{b \in B} x_b \sum_{h \in H} P[b, v_h] \cdot \left(1 - \prod_{h' \in H} (1 - SWV[s_{h'}, v_h])\right) \cdot V_h - c_b \cdot x_b \qquad (4.5)$$

$$\sum_{b \in B} x_b = 1, \qquad x_b \geq 0 \qquad \forall x_b \in B$$

- **Programme du Défenseur** *(Maximisation de son gain minimal)*

$$\max U_D$$

Sous contraintes :

$$U_D \leq -\sum_{b \in B} x_b \sum_{h \in H} P[b, v_h] \cdot \left(1 - \prod_{h' \in H} (1 - SWV[s_{h'}, v_h])\right) \cdot V_h - \sum_{h \in H} \sum_{s_h \in SW} c_{s_h} \cdot y_{s_h} \qquad (4.6)$$

$$\sum_{s_h \in SW} y_{sh} = 1, \qquad y_{sh} \geq 0 \qquad \forall h \in H$$

5.7.3. Résolution de l'Équilibre de Nash

L'équilibre de Nash est trouvé en résolvant simultanément ces deux programmes linéaires en introduisant une formulation en minimax.

Nous recherchons un point-selle où les gains de l'attaquant et du défenseur sont égaux :

$$\max_{y \in \Delta_{SWN}} \min_{x \in \Delta_B} U_D(x, y) = \min_{x \in \Delta_B} \max_{y \in \Delta_{SWN}} U_A(x, y) \quad (4.7)$$

En utilisant l'approche duale de la programmation linéaire, nous obtenons les conditions suivantes :

- Égalisation des gains :

$$u_A^* = u_D^* \quad (4.8)$$

Ce qui signifie que les solutions x_b^* et $y_{s_h}^*$ trouvées doivent respecter cette contrainte.

- Les stratégies mixtes doivent respecter les contraintes linéaires**.**

Méthode de Résolution :

1. Résolution du programme linéaire de l'attaquant pour obtenir x_b^*.
2. Résolution du programme linéaire du défenseur pour obtenir $y_{s_h}^*$.
3. Vérification de l'égalité des gains $u_A^* = u_D^*$.
    - Si l'égalité est respectée, alors (x_b^* , $y_{s_h}^*$) est un équilibre de Nash.
    - Sinon, ajuster les distributions jusqu'à convergence.

Donc, avec plus de redondance (R_h élevé) :

- Le coût de l'attaquant augmente (plus d'exploits nécessaires).
- L'utilité du défenseur augmente, car l'attaquant doit investir plus de ressources.
- L'équilibre de Nash favorise une meilleure répartition des pots de miel, augmentant ainsi la sécurité.

Et avec moins de redondance ($R_h = 1$) :

- L'attaquant trouve plus facilement une faille.

- Le coût du défenseur est plus faible, mais l'attaquant obtient une meilleure récompense.
- L'équilibre de Nash peut être moins favorable au défenseur.

L'équilibre de Nash est obtenu en résolvant simultanément les programmes linéaires de l'attaquant et du défenseur. La redondance des pots de miel diversifiés rend l'attaque plus coûteuse et moins efficace, améliorant ainsi la récompense du défenseur en équilibre.

5.8. Fonction de récompense

Nous définissons les fonctions de récompense du défenseur et de l'attaquant en intégrant la redondance des pots de miel diversifiés.

5.8.1. Récompense de l'Attaquant $U_A(a_a, a_d)$

L'attaquant cherche à maximiser la valeur des nœuds compromis tout en minimisant le coût de l'attaque.

L'efficacité de l'attaque dépend :

- Des exploits b choisis
- Des vulnérabilités associées
- De la résistance offerte par les pots de miel redondants

La probabilité que l'attaquant réussisse à compromettre le nœud h est réduite par la redondance R_h.

$$U_A(a_a, a_d) = \sum_{\mathrm{b} \in B} x_b \sum_{\mathrm{h} \in H} \mathrm{P}[\mathrm{b}, v_h] \,.\, (1 - \prod\nolimits_{h\prime \in H} (1 - SWV[s_{h'}, v_h]))\ V_h - c_b x_b \quad (4.9)$$

Où :

- $\mathrm{P}[\mathrm{b}, v_h]$: Indique si l'exploit b peut exploiter la vulnérabilité v_h.
- $\prod_{h\prime \in H}(1 - SWV[s_{h'}, v_h])$: représente la probabilité que tous les pots de miel résistent à l'attaque.
- Plus V_h Valeur du nœud h s'il est compromis.
- $c_b x_b$: Coût de l'attaque (utilisation des exploits).

Impact de la redondance :

- Plus R_h est grand, plus la probabilité d'échec de l'attaque augmente, réduisant ainsi U_A.
- L'attaquant doit utiliser plus d'exploits, augmentant son coût total.

5.8.2. Récompense du Défenseur $U_D(a_d, a_a)$

Le défenseur cherche à minimiser l'impact de l'attaque et à réduire les coûts de défense.

$$U_D(a_d, a_a) = -U_A(a_a, a_d) - \sum_{h \in H} x_b \sum_{h' \in H} c_{s_{h'}} y_{s_{h'}} \quad (4.10)$$

Avec :

- $-U_A(a_a, a_d)$: Perte due aux attaques réussies.
- $\sum_{h \in H} x_b \sum_{h' \in H} c_{s_{h'}} y_{s_{h'}}$: Coût d'installation et de maintenance des pots de miel.

Impact de la redondance :

- Augmenter R_h augmente le coût du défenseur, mais diminue les succès de l'attaquant, améliorant la sécurité.

Équilibre entre coût et sécurité : Une forte redondance est bénéfique lorsque la valeur des nœuds à protéger est élevée.

5.9. Application de la redondance dans le placement des pots de miel diversifié

Pour garantir la résilience et la durabilité dans le placement de pots de miels tel que proposé par [8], il est impérieux de ne pas se fier uniquement à la diversification des types de pots de miel. Car, de que le pot de miel est corrompu, la voix reste libre pour l'attaquant et il peut tout se permettre.

Par conséquent, nous proposons au-delà de la diversification de pot de miel, la redondance dans le placement de pot de miel. La redondance de pot de miel est le fait de mettre plusieurs pots de miel dans une même arrête pour que, si l'un tombe en panne, l'autre le remplace. Cela permet de ne pas avoir d'interruption dans la sécurisation du réseau ou du nœud cible. En d'autres termes, elle permet d'assurer la disponibilité de mesure de sécurité en cas de panne ou corruption réussi d'un pot de miel. Si l'un est compromis, l'attaquant sera buté au

deuxième pot de miel de configuration différente. Cela va retarder son avancement et permettre au défenseur de prendre les dispositions utiles pour la sécurité du réseau.

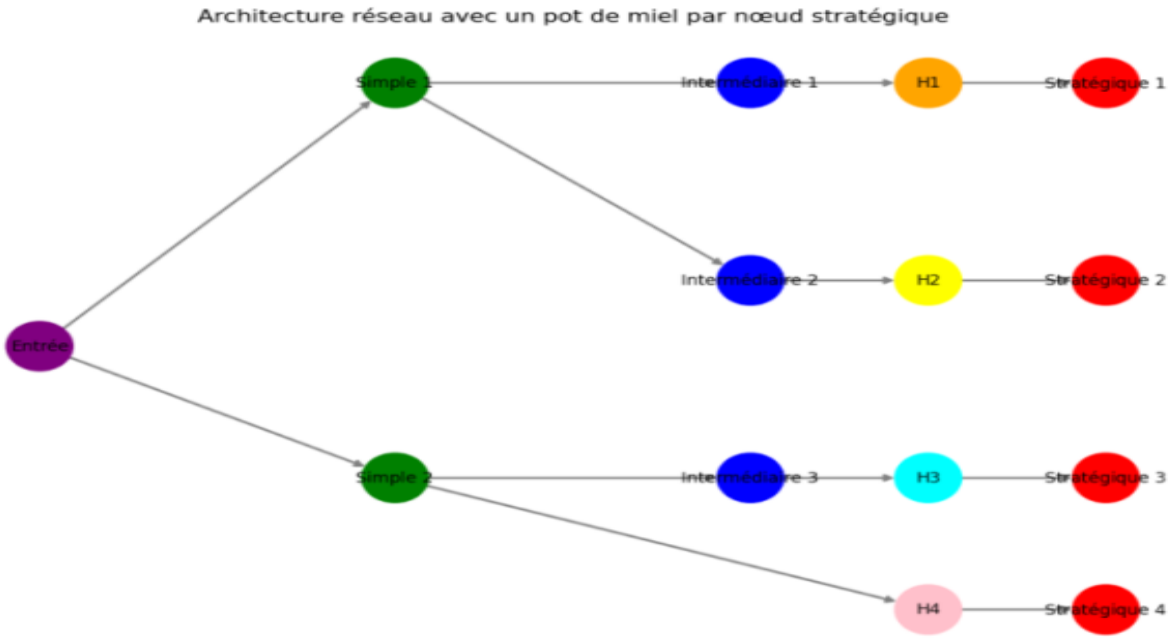

Figure 6 : architecture réseau sans redondance de pot de miel diversifié

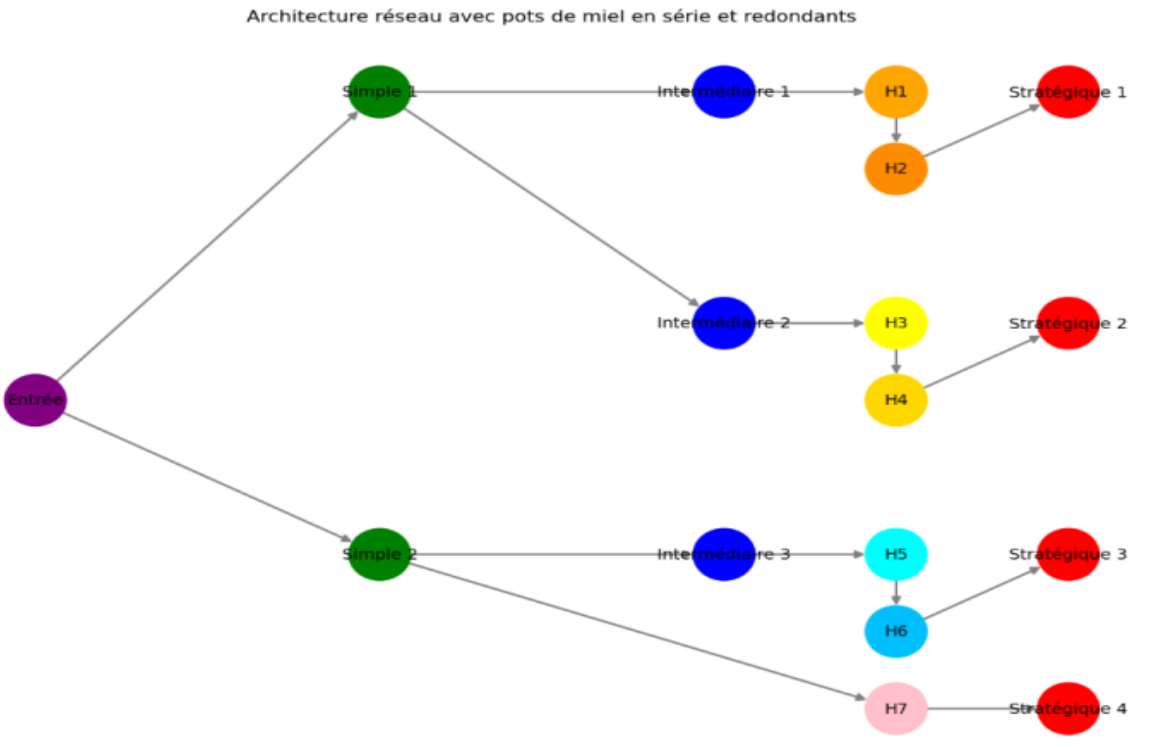

Fig. 7 : architecture réseau avec redondance de pot de miel diversifié

Partant de la même topologie, dans la figure 1 où nous avons un nœud d'entré, deux nœuds simples, trois nœuds intermédiaires et 4 nœuds stratégiques. Nous remarquons que les

4 nœuds stratégiques ont chacun un seul pot de miel. Cela étant, si l'attaquant réussi à compromettre l'unique pot de miel, directement il est en face du nœud stratégique.

En outre, les pots de miel ne sont pas seulement soumis aux attaques de l'extérieur. Un pot de miel peut s'éteindre parce qu'il a été débranché du réseau électrique, il peut subir un court-circuit, un sabotage à l'interne et il tombe en panne, exposant ainsi le nœud qu'il est censé sécurisé. Dans la figure 1, si l'unique pot de miel tombe en panne à cause d'une erreur interne telle qu'on les a énumérés, nous remarquons que l'attaquant aura le champ libre, le menant directement sur sa cible.

Pourtant dans la figure 2, la mise en place de la redondance de pot miel, résout le problème évoquer dans le paragraphe précèdent. Nous pouvons remarquer que si le premier pot de miel arriver être corrompu ou s'il tombé à cause d'une défaisance interne, l'attaquant sera face au second pot de miel qui est en redondance avec le premier. Par conséquent, notre nœud cible est toujours en sécurité.

Certes la redondance va engendrer un cout d'exploitation supplémentaire auprès du défenseur. Néanmoins, elle renforce la sécurité du réseau en augmentant une seconde couche de protection. Donc si le premier honeypot est attaqué grâce à une vulnérabilité, l'attaquant est confronté au second honeypot, donnant du temps au défenseur de revoir sa politique de sécurité.

Récompense de l'attaquant : La récompense de l'attaquant est calculée comme : 1−moyenne des probabilités de succes1.

Cela reflète le fait que plus la probabilité de succès des pots de miel est élevée, moins l'attaquant a de chances de réussir son attaque, donc la récompense de l'attaquant diminue.

Graphique combiné : Le graphique affiche à la fois la récompense du défenseur et de l'attaquant en fonction du nombre de pots de miel redondants. Vous devriez voir que la récompense du défenseur augmente avec la redondance, tandis que la récompense de l'attaquant tend à diminuer vers 0.

Cependant, le cout opérationnel de la mise en exploitation, la maintenance, correctifs et reconfigurations des honeypots vont augmenter. D'autres parts, l'attaquant est face à la double ceinture de sécurité, car il devra fournir doublement les efforts pour franchir une barrière de sécurité.

Nous allons structurer le problème de manière à représenter les actions et les paiements des deux joueurs, puis utiliser un algorithme d'optimisation pour trouver l'équilibre de Nash.

5.9.1. Notations et définitions :

- H : Ensemble des pots de miel
- SW: Ensemble des types de logiciels, | SW | est le nombre de type de logiciels
- V : Ensemble des vulnérabilités, | V | est le nombre de vulnérabilités
- B : Ensemble des exploits, | B| est le nombre d'exploits
- NSW : Matrice de dimension |H| x |SW| représentant l'affectation des logiciels aux pots de miel
- SWV : Matrice de dimension |SW| x |V| où $0 \leq \mathrm{SWV[i,j]} \leq 1$, représentant la probabilité qu'un logiciel s_i ait une vulnérabilité v_j.
- P : Matrice binaire de dimension |B| x |V| où $P_{[i,j]}=1$ si l'exploit b_i peut exploiter la vulnérabilité v_j.

5.9.2. Actions et coûts

- Espace d'action de l'attaquant ($\mathrm{A_a}$) :

$$\mathrm{A_a} = B \tag{4.11}$$

- Espace d'action du défenseur ($\mathrm{A_d}$) :

$$\mathrm{A_d} = \mathrm{SW}^H \tag{4.12}$$

- Coût de l'attaque :

$$c_b(a_a) = \sum_{b \in B} c_b \,.\, a_a \text{ (b)} \tag{4.13}$$

Où c_b est le cout associé à l'utilisation de l'exploit b et a_a(b) est l'indicateur d'utilisation de b.

- Cout de la défense :

$$c_b(a_d) = \sum_{h \in H} c_{sh} \tag{4.14}$$

Où c_{sh} est le cout associé à l'utilisation du logiciel s_h pour le pot de miel h

5.9.3. Fonction de Récompense

La diversité des pots de miel étant cruciale pour éviter des attaques répétées avec les mêmes exploits. Cela étant, nous allons utiliser l'algorithme gourmand. Et bien que la redondance des pots de miel augmente la protection, elle entraine également un coût supplémentaire.

L'algorithme gourmand peut être utilisé pour maximiser la diversité tout en contrôlant le coût. À chaque étape, on choisit l'affectation de type de logiciel pour le pot de miel qui minimise l'impact sur le coût global tout en assurant la diversité. Cela peut être formalisé ainsi:

- À chaque itération, pour chaque pot de miel h, nous choisissons un logiciel s_h de sorte que $s_h \notin \{s'_h \mid \text{h}' \text{ voisin de h}\}$ (afin de garantir la diversité), tout en minimisant le coût total de défense.

5.9.3.1. Récompense de l'attaquant ($\boldsymbol{\mathcal{R}_a}$) :

L'attaquant veut maximiser ses gains tout en minimisant ses coûts. La récompense de l'attaquant est proportionnelle à la probabilité de succès de ses exploits sur les pots de miel. Elle est définie comme suit :

$$\mathcal{R}_{\text{a}}(\text{a}_{\text{a}}, \text{a}_{\text{d}}) = \sum_{b \in B} P(b, \text{a}_{\text{d}}).\, gain\ (b, \text{a}_{\text{d}}) - \text{c}_{\text{b}}(\text{a}_{\text{a}}) \tag{4.15}$$

Avec $gain(b, \text{a}_{\text{d}}) = \sum_{h \in H} \text{V}_{\text{h}}.\, \text{p}_{\text{succès}}\ (b, h)$

Où

- $P(b, \text{a}_{\text{d}})$ est la probabilité que l'exploit b réussisse, étant fonction des logiciels déployés par le défenseur.
- $gain\ (b, \text{a}_{\text{d}})$ est le gain que l'attaquant obtient si l'exploit b réussit.
- Le terme $\text{c}_{\text{b}}(\text{a}_{\text{a}})$ représente le coût de l'attaque
- Chaque honeypot a une valeur stratégique V_{h}
- Plus la probabilité de succès est élevée, plus l'attaquant gagne
- Si un honeypot est sécurisé avec un logiciel différent ou redondant, $\text{p}_{\text{succès}}$ diminue et donc le gain diminue.

5.9.3.2.*Récompense du défenseur* ($\boldsymbol{\mathcal{R}_d}$) :

Le défenseur veut minimiser les risques tout en équilibrant le coût de la défense. Sa récompense peut être exprimée par :

$$\mathcal{R}_{\mathrm{d}}(\mathrm{a_a}, \mathrm{a_d}) = \sum_{h \in H} (1 - P(b, \mathrm{a_d})).\, gain\ défense\ - \mathrm{c_d}(\mathrm{a_d}) \tag{4.16}$$

Où

- $(1 - P(b, \mathrm{a_d}))$ représente l'efficacité du logiciel $\mathrm{s_h}$ du pot de miel h contre l'exploit b.
- $gain\ défense$ est le gain que le défenseur obtient pour chaque exploit non réussit par l'attaquant.
- Le terme $\mathrm{c_b}(\mathrm{a_d})$ représente le coût de la défense.

5.10. Résultats numériques

Dans cette section, nous présentons nos résultats numériques qui valident l'approche proposé sur l'attribution de pots de miel de manière optimal en usant de la redondance afin de maximiser le gain du défenseur. Notre analyse permet aussi de voir l'impact de la redondance dans la récompense du défenseur.

Considérons une topologie réseau à 10 nœuds ayant 1 seul nœud d'entré en mauve, 2 nœuds simples en verts, 3 nœuds intermédiaires en bleu et 4 nœuds stratégiques en rouge. Chaque nœud simple a une valeur de 50, nœuds intermédiaires à une valeur de 150 et 3 nœuds stratégique ont la valeur respective de 200 et le dernier nœud stratégique a la valeur de 250 et le nœud d'entrée 50.

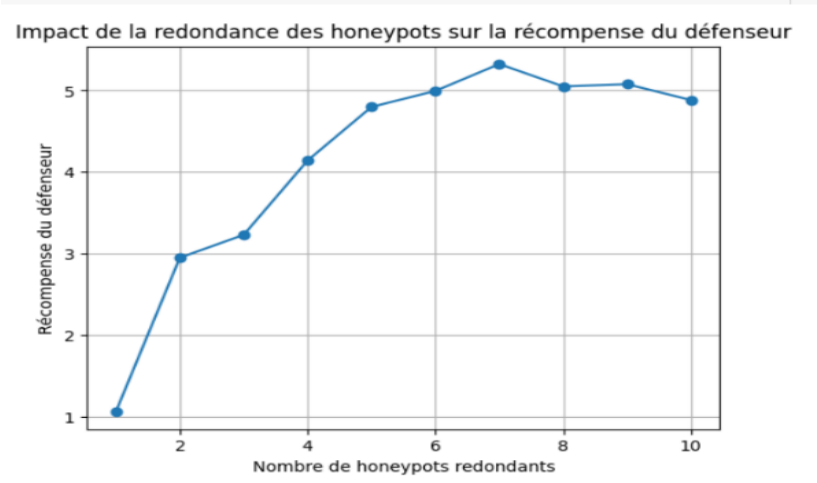

Fig. 8 : Avec probabilité de réussite de l'attaquant

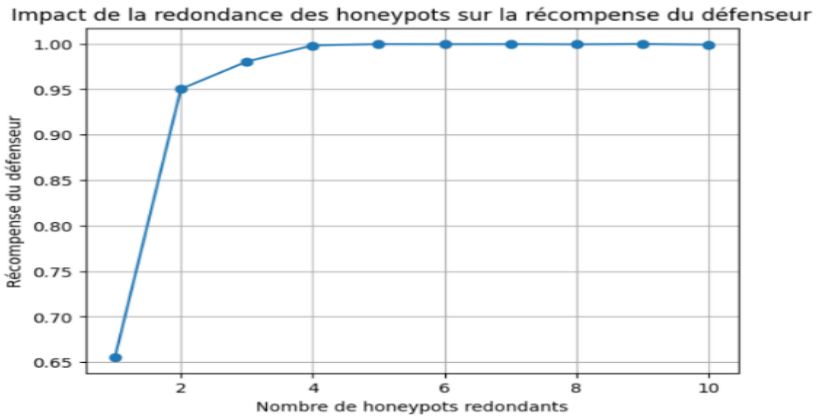

Fig. 9 : Sans probabilité de réussite de l'attaquant

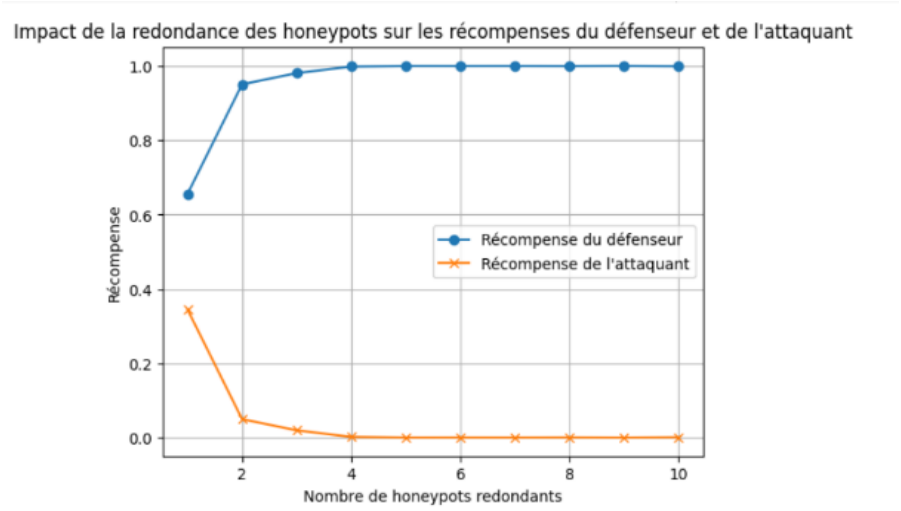

Fig. 10 : Evolution de la récompense du défenseur ainsi que celle de l'attaquant.

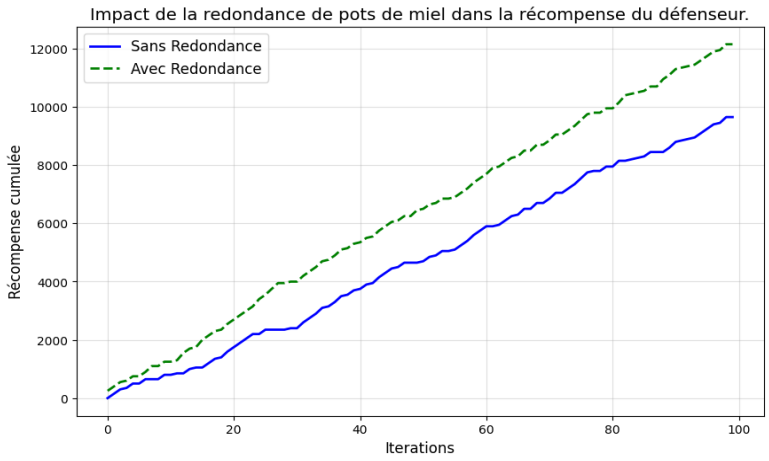

Fig.11 : l'impact de la redondance de pots de miel dans la récompense du défenseur.

A l'aide de l'algorithme de fictitious play, un jeu est simulé entre les deux joueurs. L'attaquant cible les nœuds pour maximiser ses gains. Le défenseur utilise une stratégie mixte pour protéger les nœuds critiques. Le récompense par niveau de redondance est de 0.5 et le niveau de redondance de 0 à 10. La fig.8 nous montre la récompense du défenseur avec probabilité de réussite de l'attaquant. Et dans la fig.9, on peut voir l'impact de la récompense du défenseur sans probabilité de réussite de l'attaquant.

En utilisant une simulation aléatoire, nous examinons l'impact de la probabilité de réussite de l'attaquant sur la récompense du défenseur dans la fig. 12 et 13, tout en tenant compte de la redondance, par exemple, avec de probabilité de l'ordre de 0.1, 0.3, 0.5, et 0.7. Dans la fig.12, à mesure que p_{succes} augmente (de 0.1 à 0.7), la récompense du défenseur diminue rapidement. Les courbes illustrent que le défenseur est plus vulnérable lorsque l'attaquant a une forte probabilité de réussir. Dans la fig.12, les courbes montrent une récompense plus élevée pour le défenseur, même lorsque p_{succes} est élevé. La redondance réduit efficacement l'impact négatif des probabilités élevées de réussite de l'attaquant.

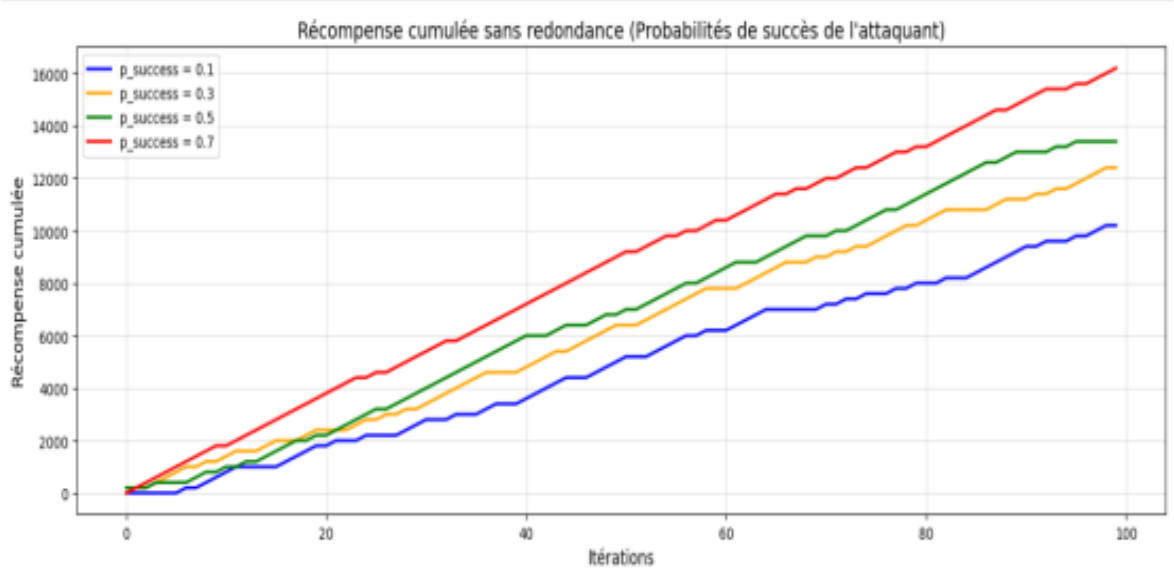

Fig.12 : Récompense cumulée sans redondance (Probabilités de succès de l'attaquant)

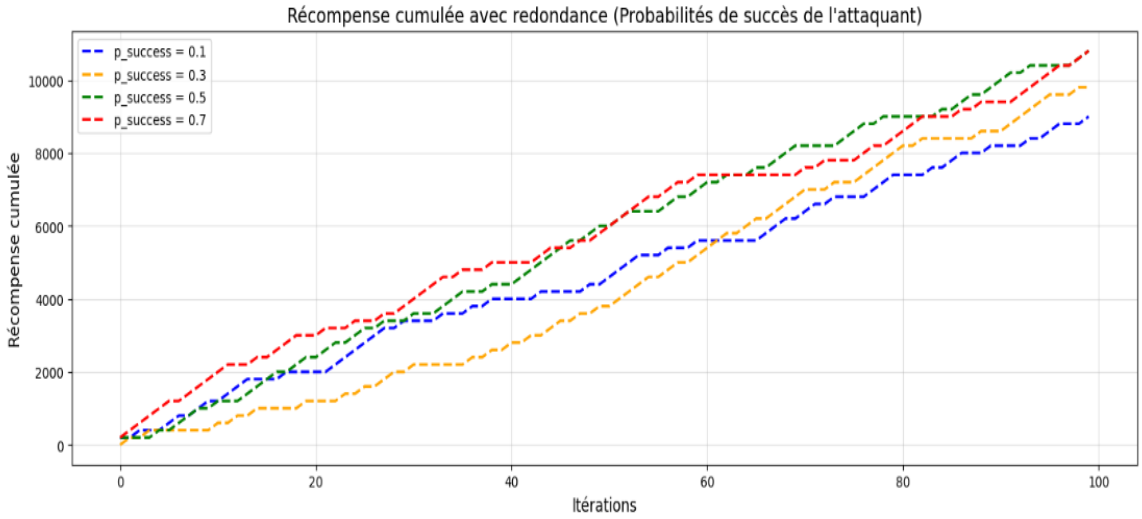

Fig.13 : Récompense cumulée avec redondance (Probabilités de succès de l'attaquant)

De même dans la fig. 11, la ligne bleu nous montre que les pots de miel sont placés de manière non redondante et la ligne verte nous montre que les pots de miel sont placés de manière redondante sur des arêtes stratégiques. La courbe verte (avec redondance) montre une augmentation significative de la récompense cumulative au fil des itérations. Cela démontre que l'utilisation de la redondance améliore l'efficacité de la stratégie défensive, augmentant la probabilité d'intercepter l'attaquant.

Avec un coût d'un honeypot estimé à 1 unité arbitraire, nous avons 4 nœuds stratégiques à protéger sans redondance, le coût total étant de 4 unités pour les 4 honeypots. Avec redondance, le coût sera de 8 unités pour 8 honeypots. En raison de 2 honeypots par nœud stratégique, nous pouvons simuler et afficher ce coût croissant en fonction du niveau de redondance, de 1 à 3 honeypots par nœud stratégique. Comme le montre la figure 14 et 15, l'augmentation du coût de défense est linéaire avec le niveau de redondance des honeypots par nœud stratégique.

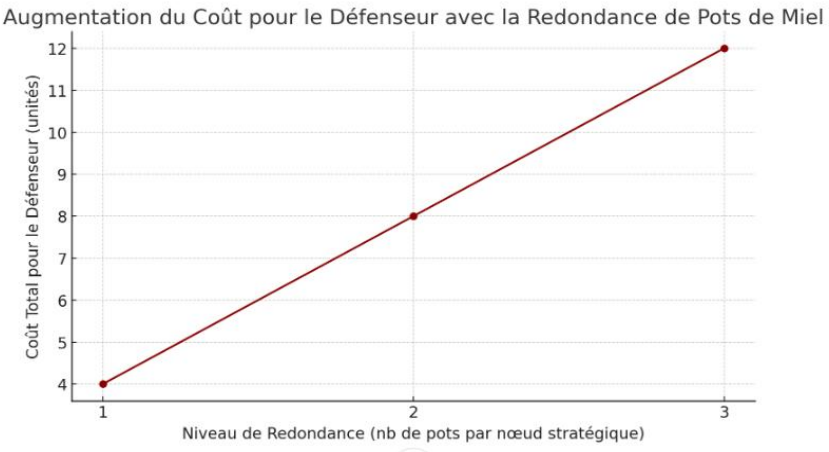

Fig. 14 : Coût et ratio de redondance sans probabilité de succès

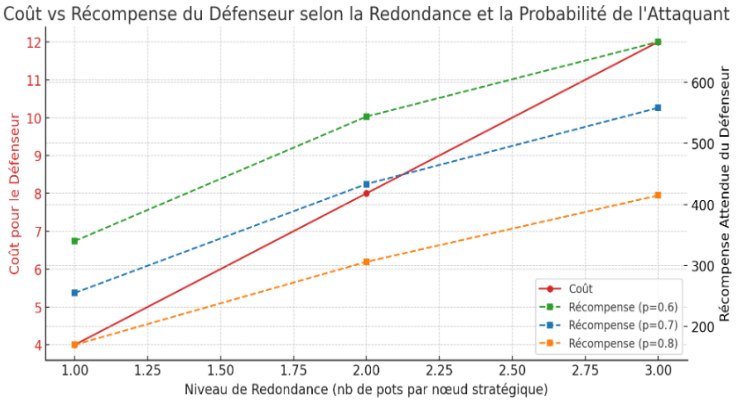

Fig. 15 : Coût et ratio de redondance avec probabilité de succès

La redondance améliore la sécurité, mais elle engendre un coût supplémentaire significatif pour le défenseur. Comparée à différentes probabilités de réussite de l'attaquant (par exemple, 0,6, 0,7, 0,8), comme le montre la figure 15, plus l'attaquant est efficace (p élevé), plus le défenseur est incité à augmenter le nombre de honeypots pour compenser.

5.11. Déploiement expérimental sur infrastructure réelle

Dans la première partie de ce chapitre, un développement mathématique approfondi suivi d'une simulation à l'aide de l'algorithme fictituis play a été présenté, il devient important qu'on puisse étendre cette analyse théorique en pratique. Ce passage de déploiement informatique avec des équipement réels permettra de valider le modèle en condition concrète.

Cette partie permet d'évaluer la robustesse, l'efficacité et la faisabilité opérationnelle des stratégies de défense étudiées, en tenant compte de contrainte de ressource qui sont limitées. Elle va constituer un pont crucial entre la modélisation abstraite et les applications pratiques de la cybersécurité et cyberdéfense.

Description du réseau

Nous avons mis en place un réseau virtuel, subdiviser en 3 zones avec un nombre réduit de nœud et qui est composé de :

1. WAN

   - D'un ordinateur virtuel sous Kali Lunix, utilisé par l'attaquant

- D’un ordinateur Windows utilisé pour l’accès distant par le défenseur
- D’un ordinateur virtuel sous Kali linux pour le défenseur
- Putty (un client SSH gratuit qu’on installe sur Windows)
- Nmap (un logiciel pour scanner les ports)

2. LAN (192.168.100.0/24)
    - D’un ordinateur virtuel sous Windows
3. DMZ (192.168.200.0/24)
    - Un ordinateur sous Lunix avec Pentbox. (Ecrit en Ruby, pentbox est une application qui peut être configurer comme un pot de miel. Disposant de plusieurs outils, elle est utilisée pour le test d’intrusion.
    - Un ordinateur sous Windows avec KFSensor. (KFSensor, étant un système de pot de miel avancé, il agit pour attirer et détecter les tentatives d’intrusions en simulant de service système vulnérable. Personnalisable, il agit sur les ports TCP, ICMP UDP.
    - D’un serveur virtuel Windows 2019, où nous avons ajouté le rôle IIS, permettant l’accès distant sur le port 80 (http, https) et aussi nous avons activé le OpenSSH pour l’accès distant avec le port 22 (SSH)

Et ensuite, nous avons un VM avec PFSense installé. Pfsense est un système d’exploitation gratuit pour le routeur et le pare-feu. Nous allons l’utiliser pour créer notre DMZ et LAN. Il peut aussi être utiliser comme serveur DHCP, serveur DNS Serveur VPN.

Sur ce, nous allons déployer premièrement une architecture simple et après études, nous allons ajouter les pots de miel.

- Architecture 1 : nous mettons en place notre LAN, DMZ et nous le connectons au WAN grâce à notre Pfsense. Et nous mettons en place aussi la machine Kali-linux pour l’attaquant ainsi que celui du défenseur.

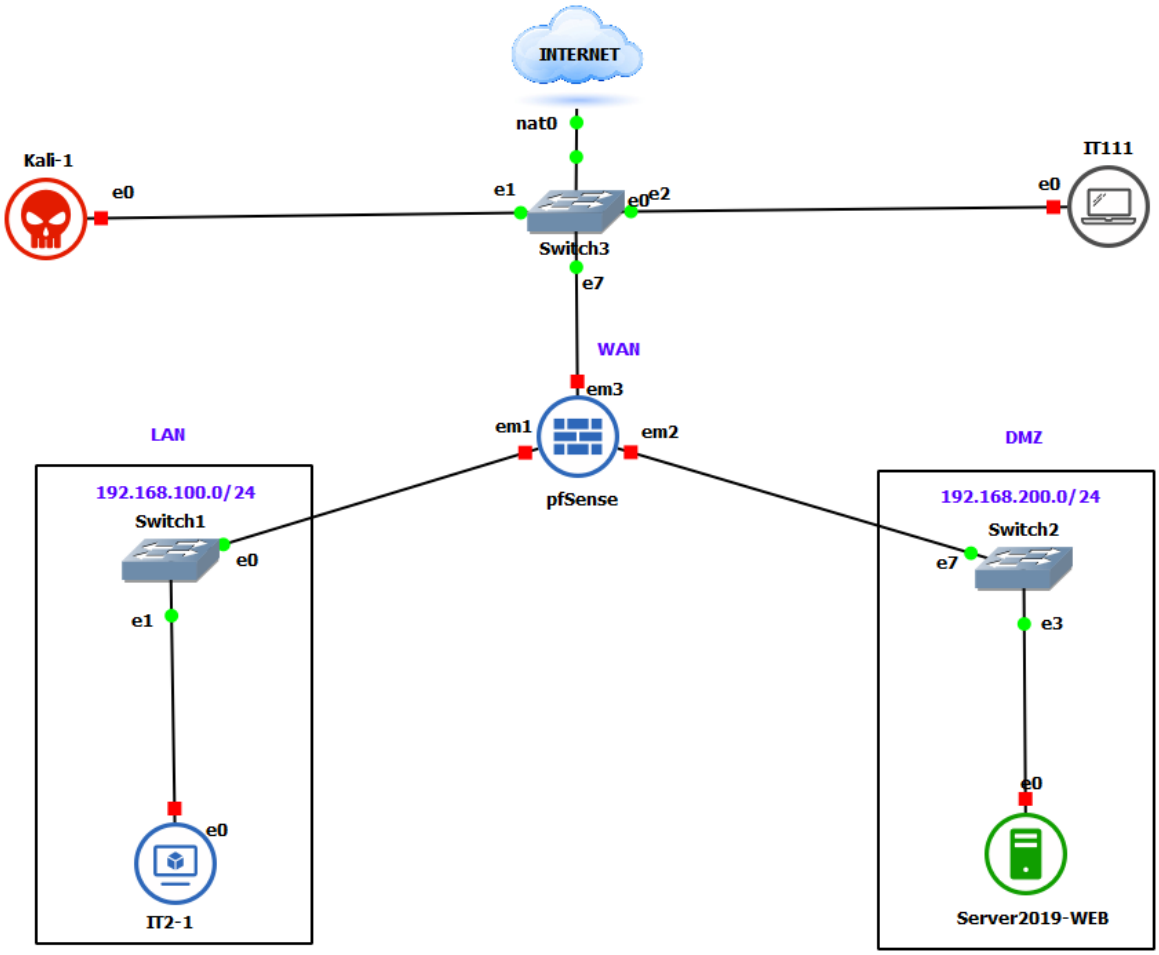

Figure 15 : Architecture sans pot de miel

Dans notre Pfsense, nous configurons le port 22 (SSH) et le port 80 (http) pour permettre l'accès distant, comme l'indique l'image ci-dessous.

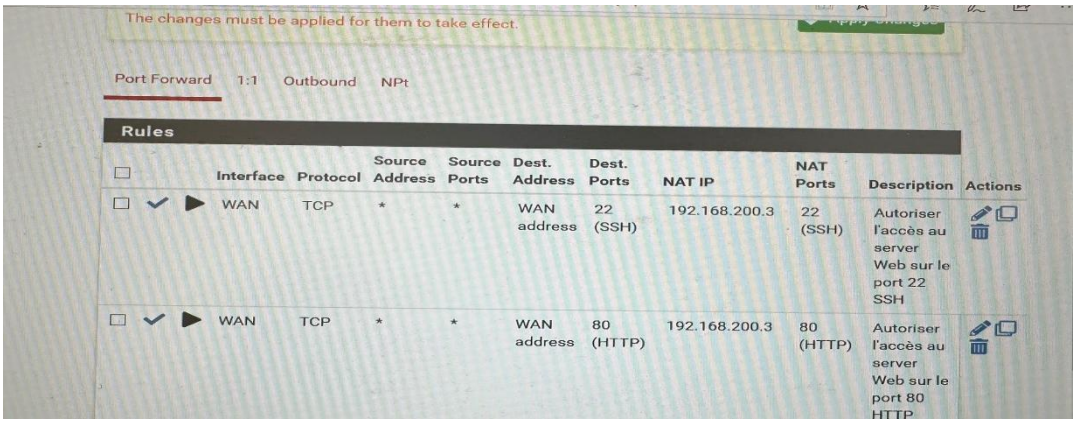

Figure 16 : Configuration SSH et http sous Pfsense

- ✓ Autorisation de l'accès au server web sur le port 22 SSH
- ✓ Autorisation de l'accès au server web sur le port 22 SSH

Ensuite nous allons utiliser Nmap pour scanner notre réseau. Au niveau de l'ordinateur du défenseur depuis le WAN, nous avons scanné réseau et nous pouvons remarquer les ports qui sont ouvert et accessible depuis le WAN.

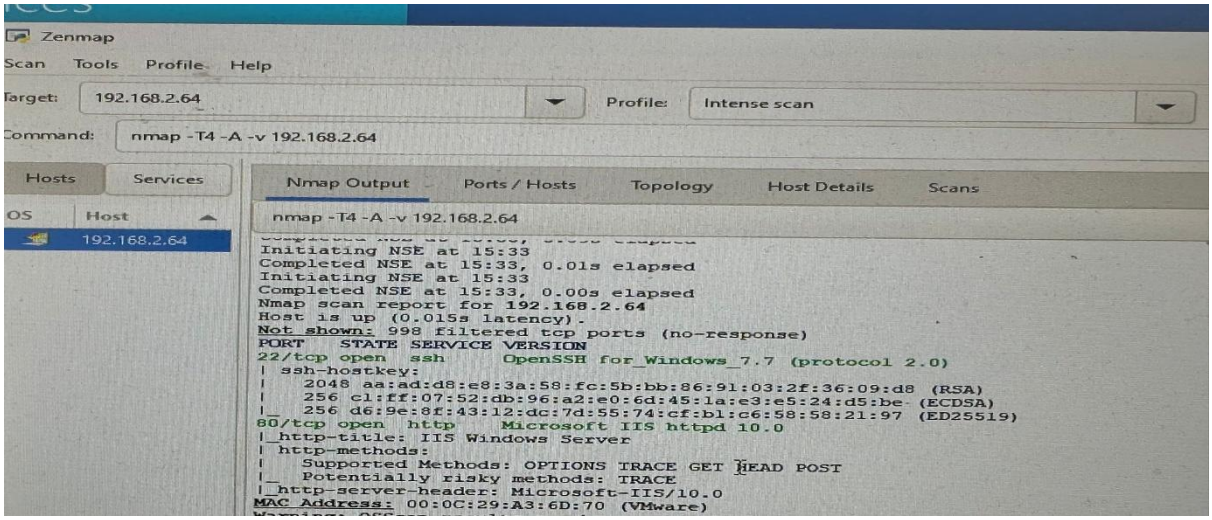

Figure 17 : Scan du réseau avec Nmap du défenseur

Nous remarquons que les ports 22 et 80 sont ouverts. Nous allons tenter une connexion distante sur le port 80

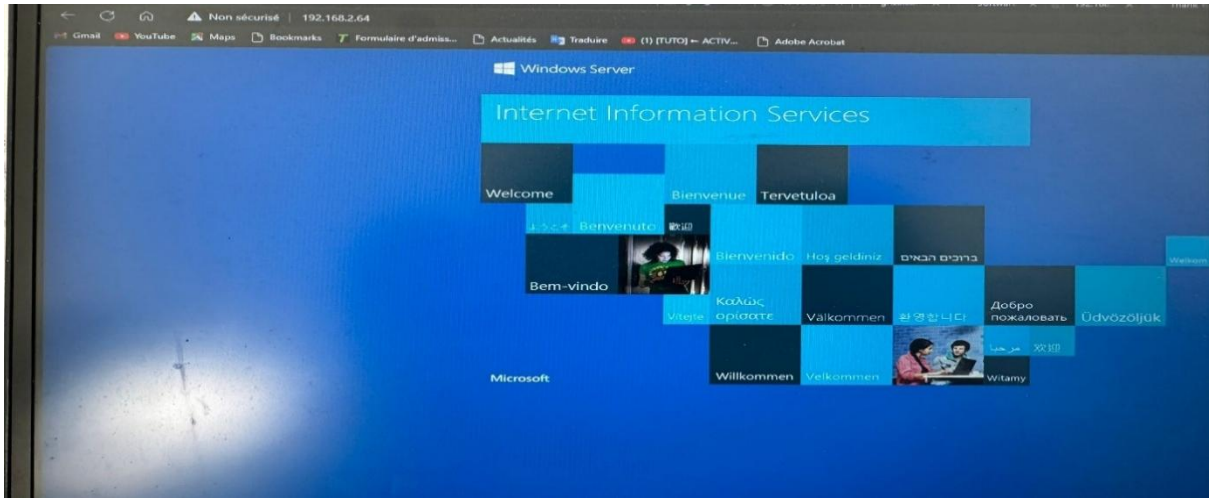

Figure 18 : Connexion distante sur server web

Nous pouvons constater que le port 80 HTTP est bel et bien accessible depuis le WAN. Cette architecture permet au défenseur d'accéder à distance à partir du WAN, et accéder à notre server Web au niveau du DMZ.

Néanmoins, cette ouverture vers l'extérieur ouvre aussi une surface d'attaque pour les agents malveillant. Car si l'attaquant arriver à obtenir l'IP publique, il lui sera facile d'accéder et de prendre la main sur le server. Etant donné qu'on sous-entend que l'attaquant est en possession de l'adresse IP publique de notre réseau. Il va d'abord scanner l'IP pour voir quels sont les ports qui sont ouvert et quels services y sont rattachés et enfin mener des attaques vers le DMZ.

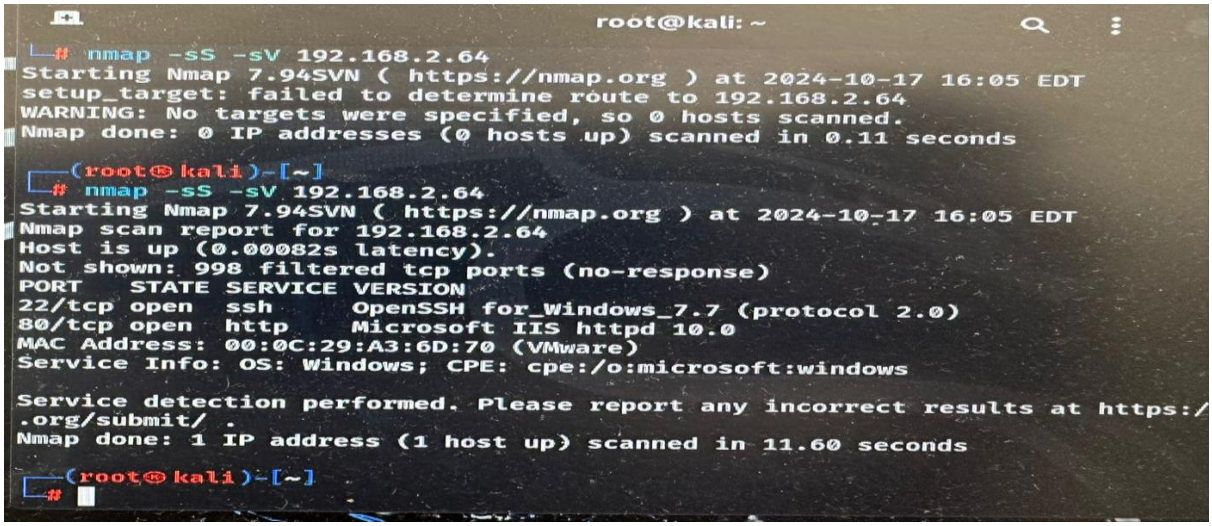

Figure 19 : Scan sous Kali Linux avec Nmap par l'attaquant.

Après avoir découvert que le port 22 et 80 sont ouvert. Il va lancer des attaques. Ici, il lance un attaque Forte Brute afin d'avoir accès aux noms d'utilisateur ainsi que leurs mots de passe. Premièrement, il attaque avec Hydra, puis s'en vient l'attaque avec Medusa

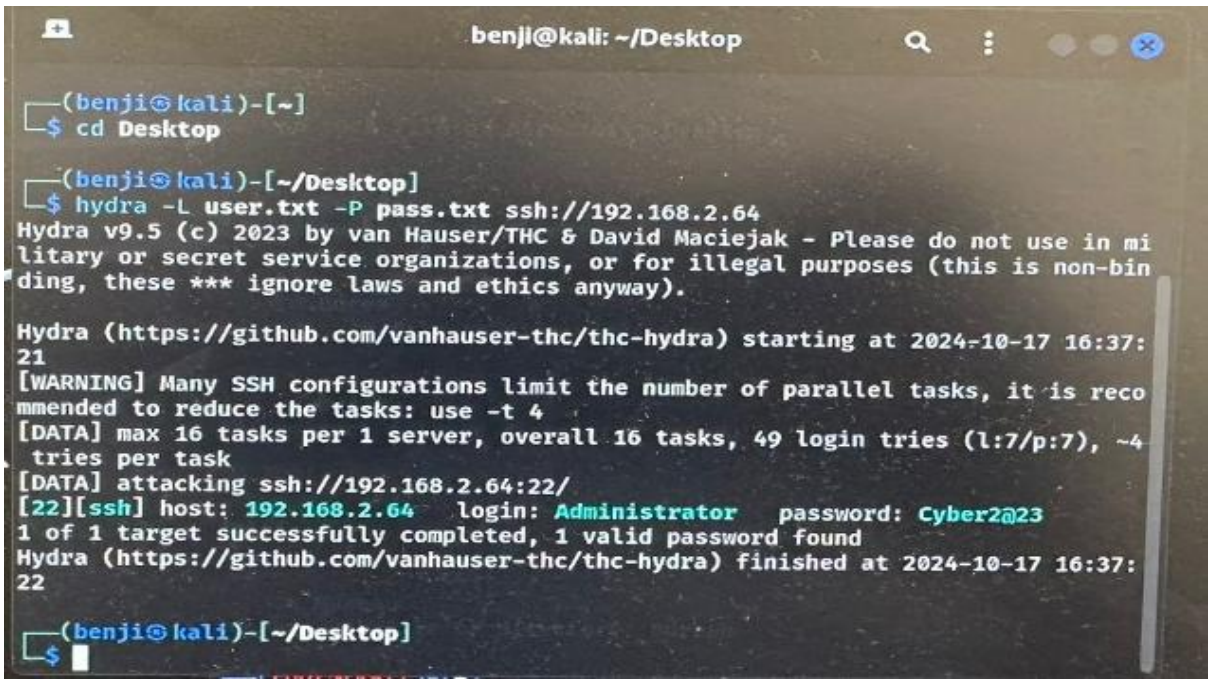

Figure 20 : Attaque avec Hydra

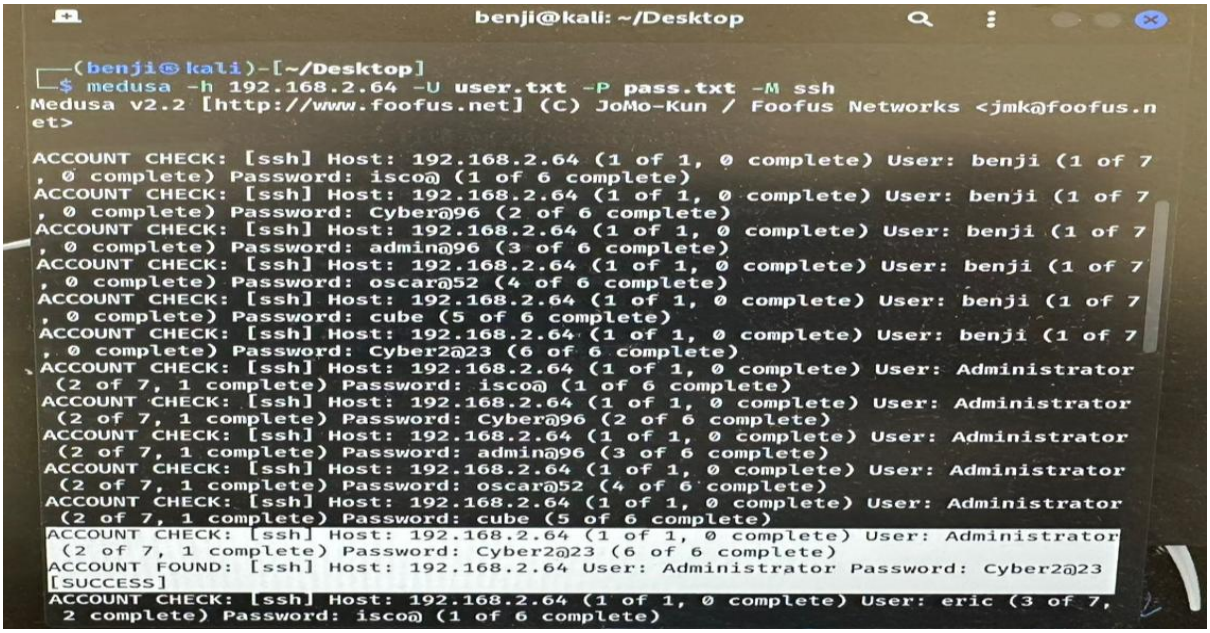

Figure 21 : Attaque avec Medusa

On peut voir que les attaques menaient par l'attaquant, lui ont permis d'avoir le nom d'utilisateur de notre server web au niveau du DMZ en claire. Avec ça, il peut se connecter à distance et faire ce qu'il veut.

Solution 1

Dans la solution 1, nous avons mis en place un pot de miel Pentbox sous kali lunix dans notre DMZ. Et nous avons activé le port 80 HTTP pour servir d'appât.

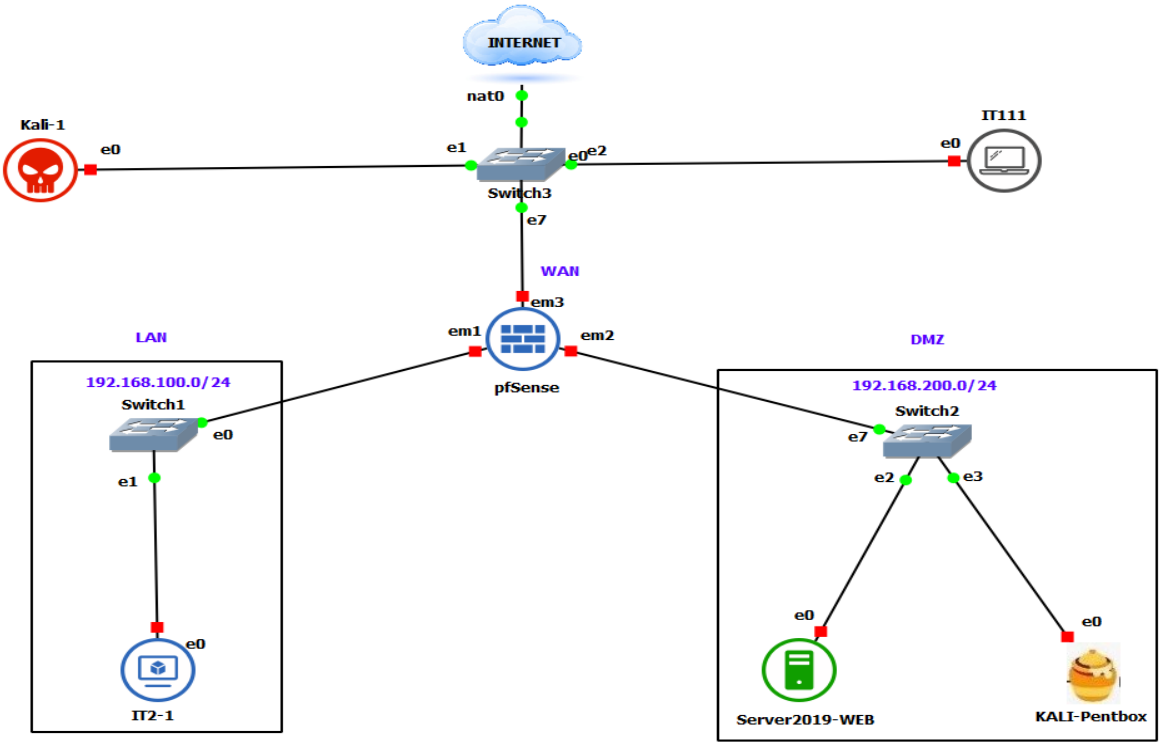

Figure 22 : Architecture avec déploiement du premier pot de miel dans le DMZ

Nous allons à présent passer à l'installation du pot de miel Pentbox dans un ordinateur sous Kali lunix, pour activer le port 80. Afin de donner un accès distant par SSH.

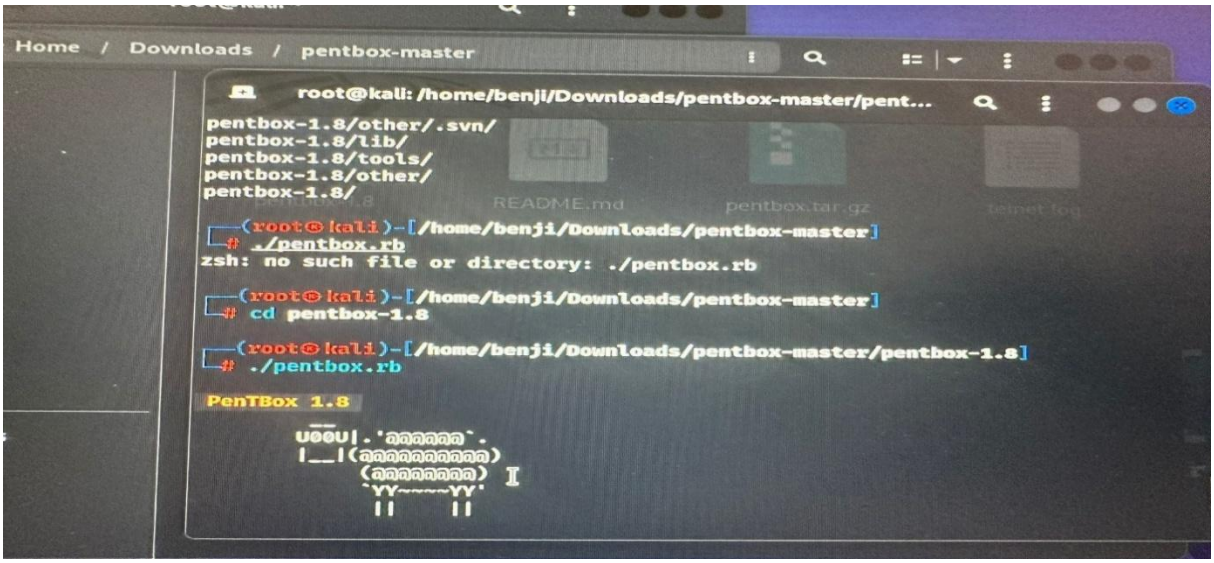

Figure 23 : Phase 1 de l'installation du Pentbox

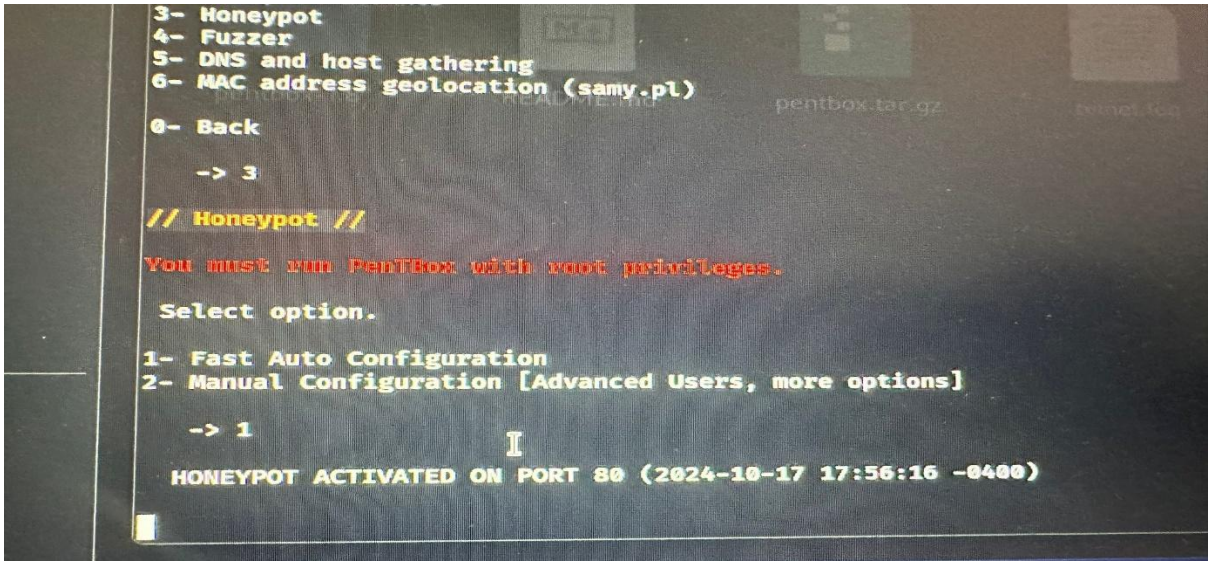

Figure 24 : Phase 2 de l'installation du Pentbox

Sur la figure phase 2, on peut bien voir que le port 80 de notre pot de miel est activé. Et au niveau de notre Pfsense, nous configurons l'accès distant via le protocole 80 de http

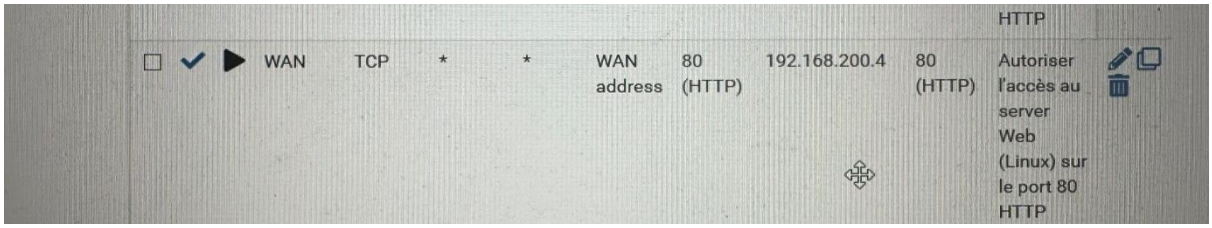

Figure 25 : Activation accès distant sur le port 80 dans pfsense

Observons maintenant le comportement du pot miel après tentative d'intrusion sur le port 80

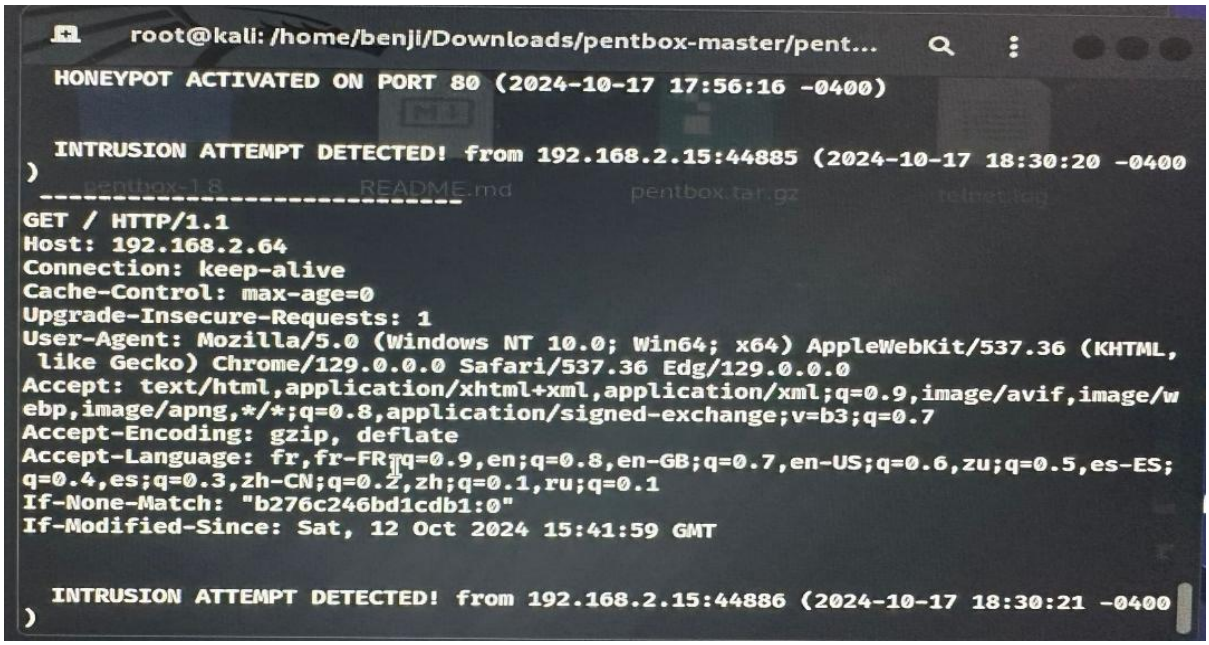

Figure 26 : Détection d'intrusion par le pot de miel Pentbox

Nous constatons que notre pot de miel réagit instantanément aux tentatives d'intrusions. Il donne l'heure de l'attaque, l'adresse IP de l'attaquant, le port d'attaque ; le système d'exploitation de l'attaquant…

Nous ouvrons un autre port, le 23 Telnet au niveau de notre pot de miel Pentbox.

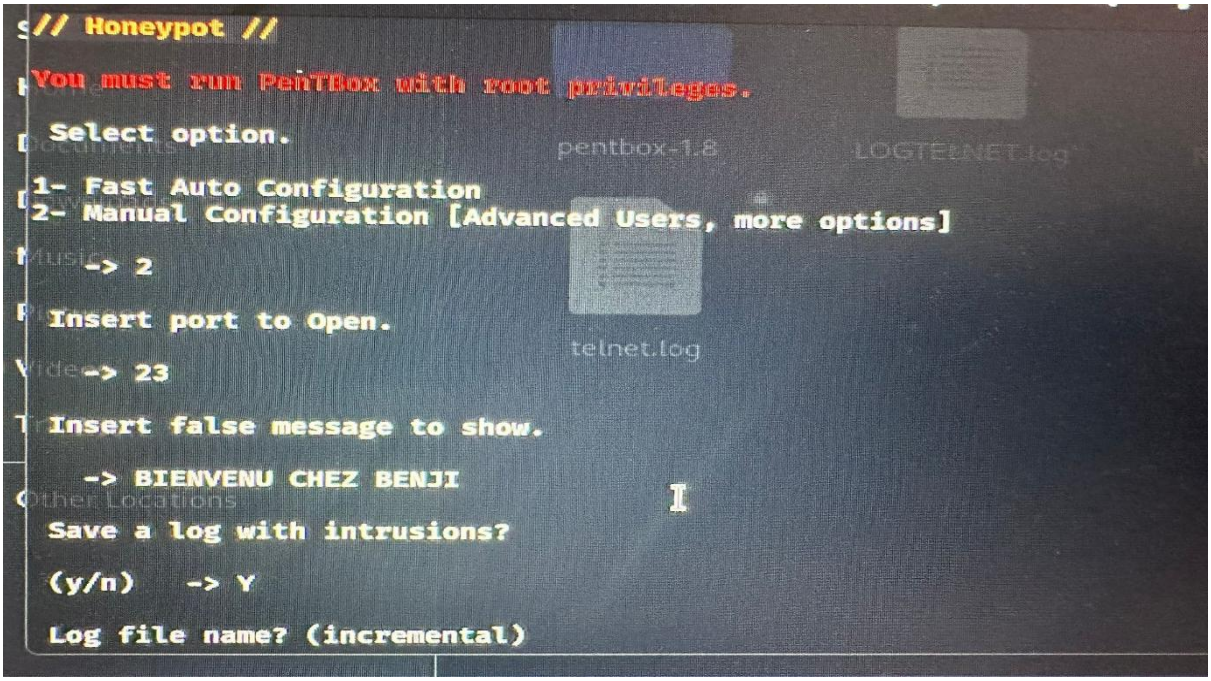

Figure 27 : Phase 1 de la configuration du telnet sur Pentbox

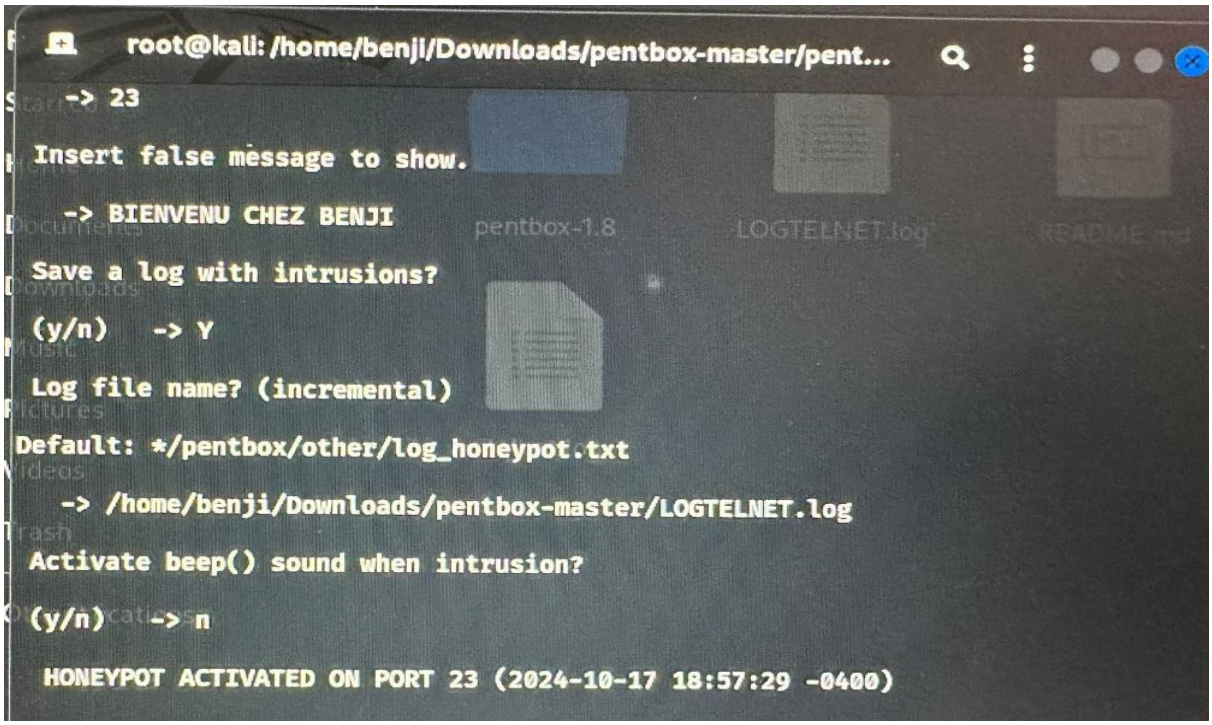

Figure 28 : Phase 2 de la configuration du telnet sur Pentbox

Sur la figure ci-haut, nous pouvons remarquer bien que le port 23 est activé. Et au niveau de notre Pfsense, nous maintenons l'accès distant via le protocole 80 de HTTP

Observons maintenant le comportement du pot miel après tentative d'intrusion sur le port 80.

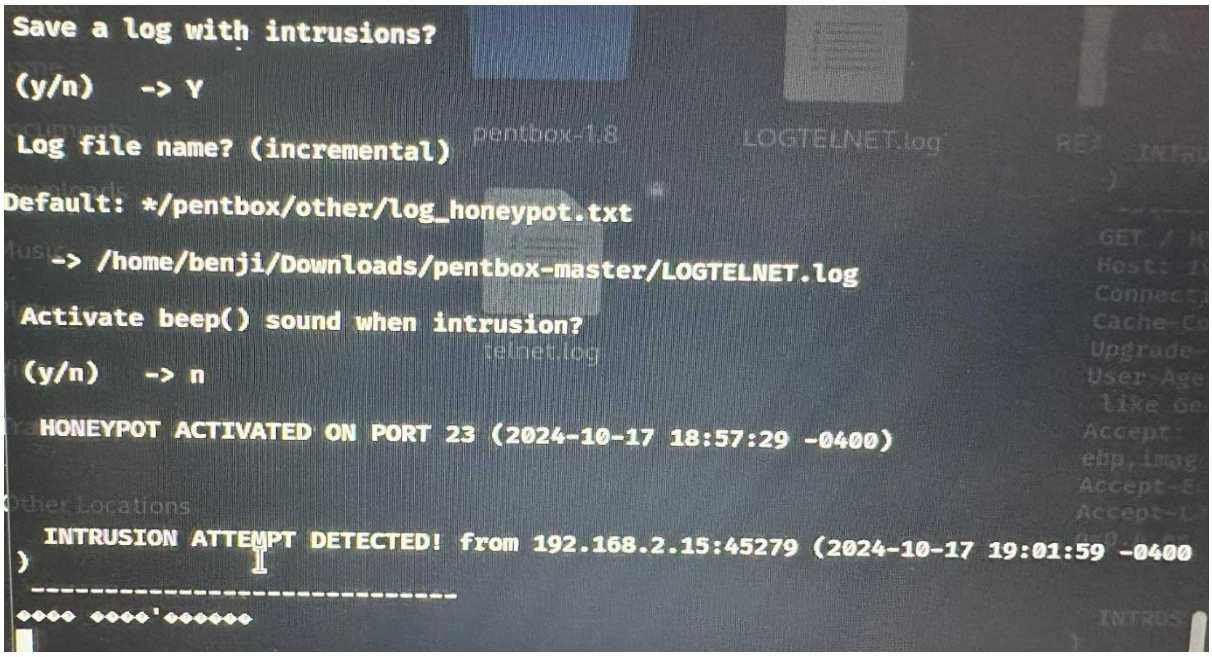

Figure 29 : Détection d'intrusion via le pot de miel Pentbox

Nous constatons la sensibilité de notre pot de miel. Dès qu'il y'a une tentative de connexion via les ports ouverts en son sein, directement il le signale et nous pouvons voir en claire, la description de l'attaquant.

Néanmoins, cette solution est faillible, car les attaques ne peuvent pas venir seulement de l'extérieur. Admettons qu'il a une tope au sein de l'entreprise, et il se permet de mettre notre pot de miel à l'arrêt. Notre server sera exposé et l'attaquant dans ses tentatives, va avoir l'accès.

Solution 2

Dans la solution 2, étant donné qu'une menace peut aussi venir de l'intérieur. Un agent peut être au service de l'ennemi et débrancher ou causer l'interruption de notre pot de miel, mettant à l'arrêt notre système de détection. Par conséquent, cette coupure nous rend aveugle, car on aura plus de moyen de surveillance ou de détection les intrusions. Ainsi donc, nous avons mis en place un deuxième pot de miel KFsensor sous windows dans notre DMZ. Il va fonctionner de façon indépendant du premier pot de miel. Avec ce deuxième pot de miel, nous allons activer la totalité de port.

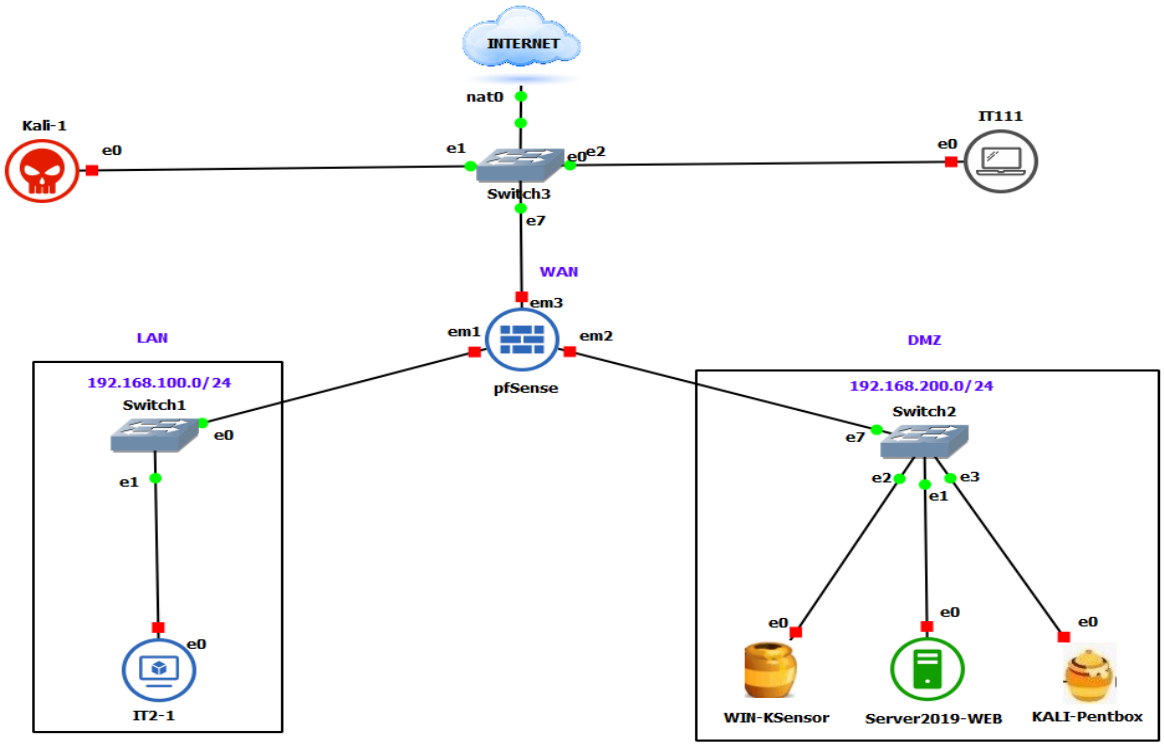

Figure 30 : Architecture avec déploiement du second pot de miel dans le DMZ

Après avoir télécharger et installer notre deuxième pot de miel, nous maintenons l'accès au niveau de notre Pfsense en laissant tout le port ouvert pour expérimenter plusieurs tentatives au niveau de différents ports.

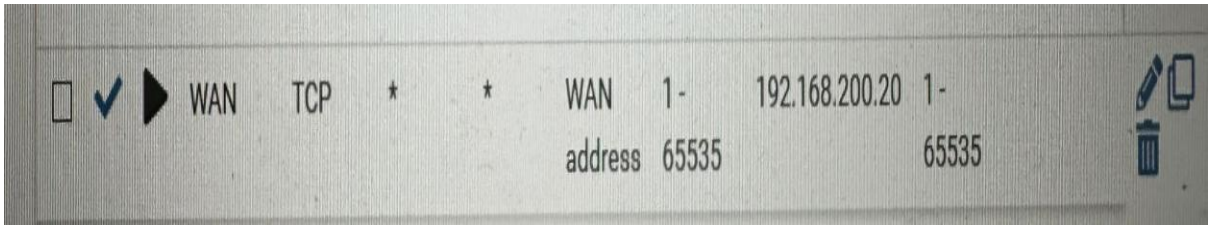

Figure 31 : Activation de tous les ports au niveau du pfsense.

Vérifions d'abord les pour ouvert au niveau de nos pots de miel. On peut voir les trois ports ouverts sur notre pot de miel Pentbox sous Kali lunix. Nous avons le port 443, 80 ainsi que 50. Cela étant, toute tentative d'intrusions via ce 3 port sera détecté par notre pot de miel.

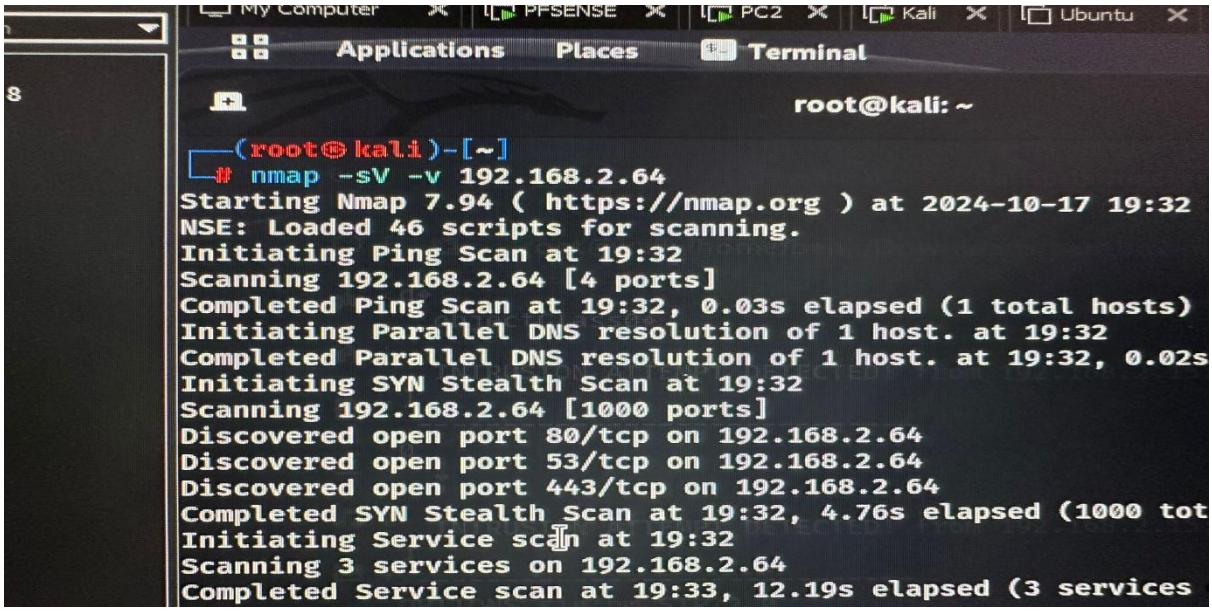

Figure 32 : Vérification de port ouvert avec Nmap.

Ensuite, nous allons vérifier les ports ouverts sur notre pot de miel KFsensor qui est installé sur un ordinateur Windows.

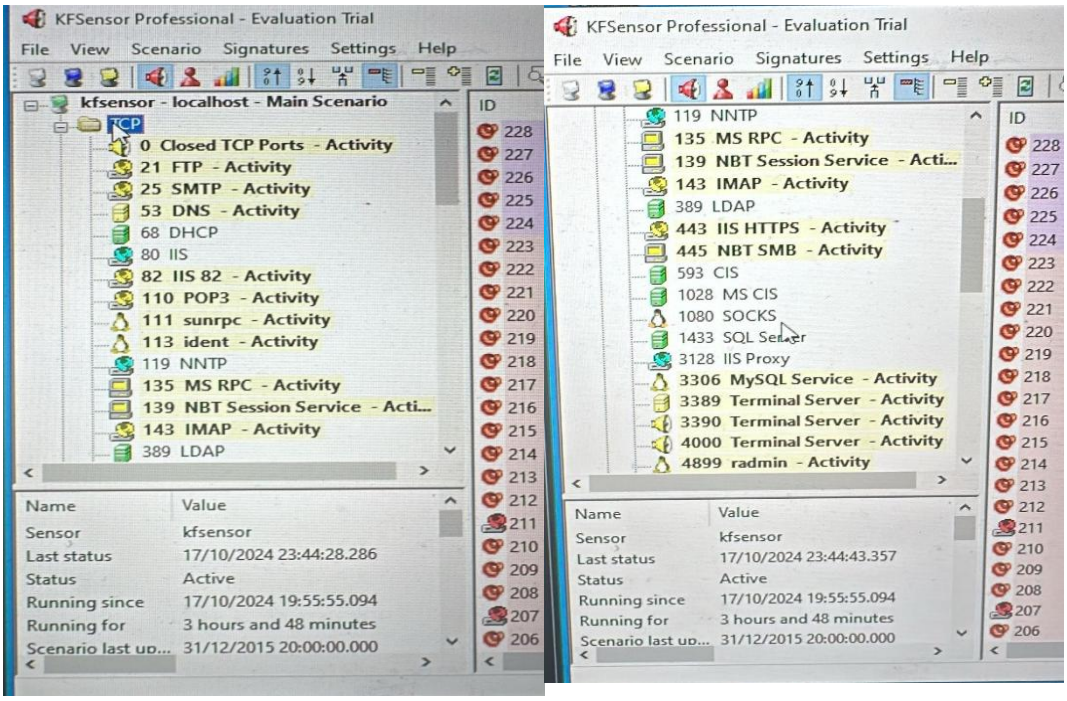

Figure 33 : Vérification de port ouvert avec KFSensor

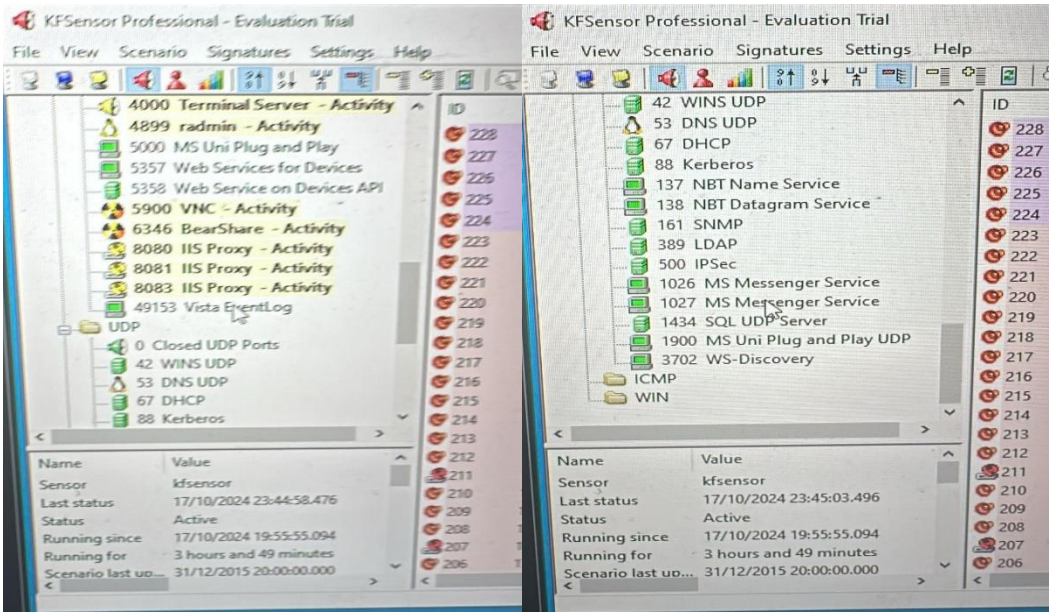

Figure 34 : Vérification de port ouvert avec KFSensor (suite)

Nous remarquons que tous les ports en vert sont ouverts.

Quand nous tentons une tentative d'accès sur le port 80 à partir de l'ordinateur attaquant, nous constatons qu'au même moment, nos deux pots de miel, installés séparément, ont émis des signales des tentatives d'intrusion.

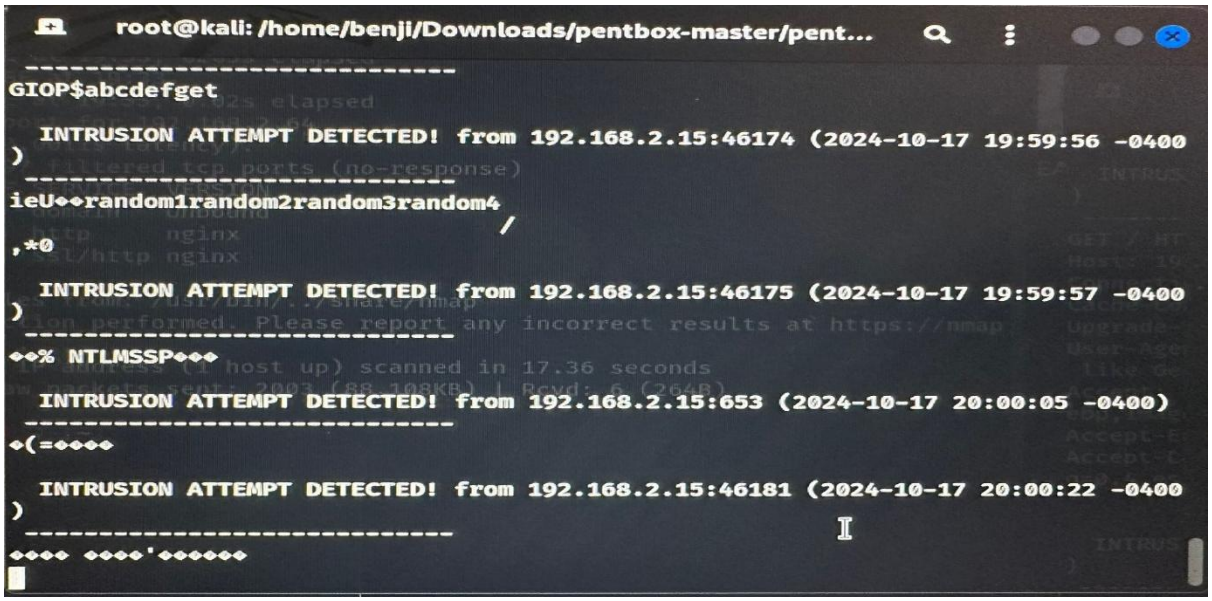

Figure 35 : Détection d'intrusion avec le premier pot de miel Pentbox

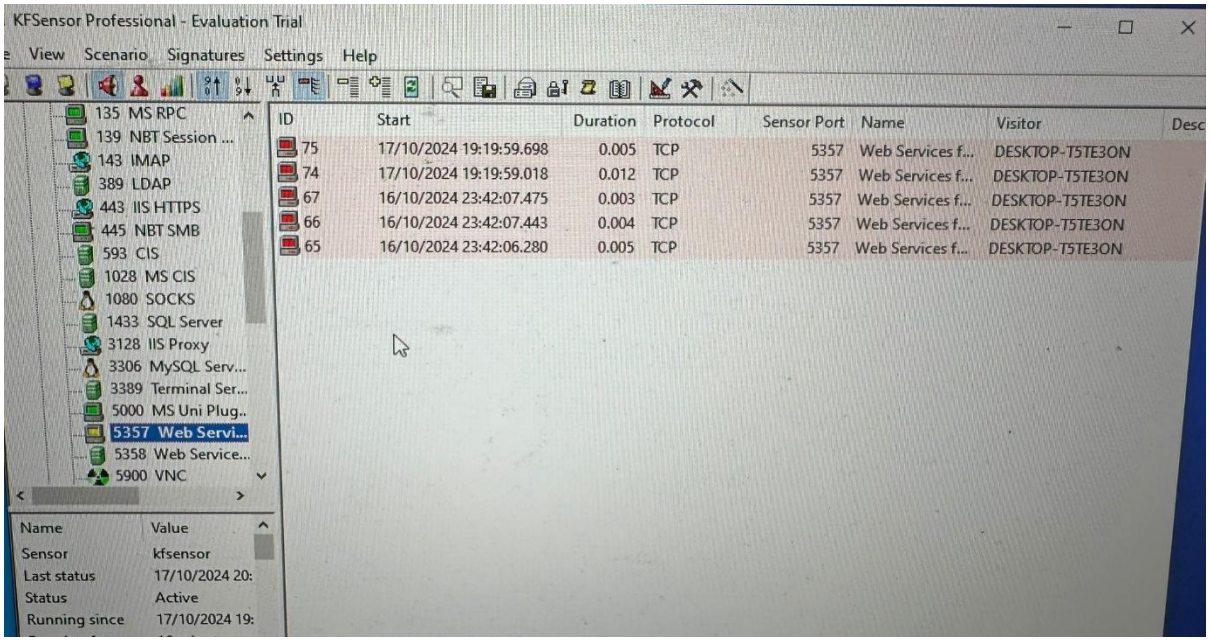

Figure 36 : Détection d'intrusion avec le second pot de miel KFSensor

Nous allons interrompre brusquement les services du pot de miel Pentbox qui est sur Kali lunix, faisant allusion à une attaque interne venant d'un employé au service de l'ennemi qui va mettre à l'arrêt complet notre pot de miel Pentbox. Tout en laissant le Kfsensor actif.

Nous remarquons que le pot de miel Kfsensor continue à nous alerter en cas de tentatives d'intrusions.

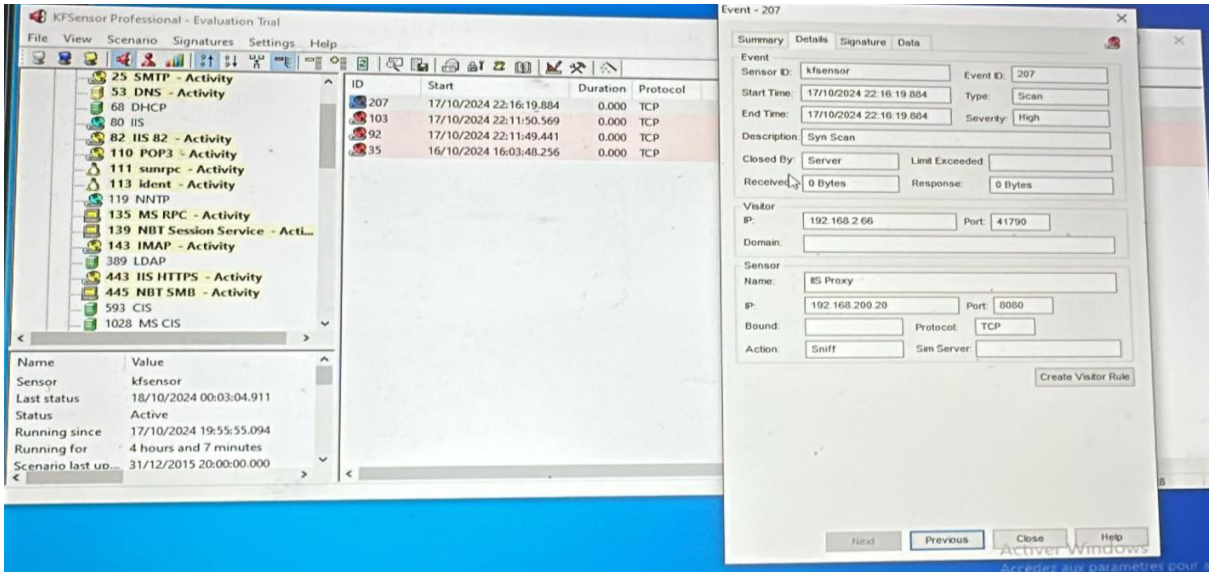

Figure 37 : Détection de la tentative d'intrusion sur le pot de miel KFSensor

Et même quand nous laçons 3 tentatives d'intrusion au même moment (FTP, DNS,SMTP), nous constatons que notre second pot de miel est toujours actif

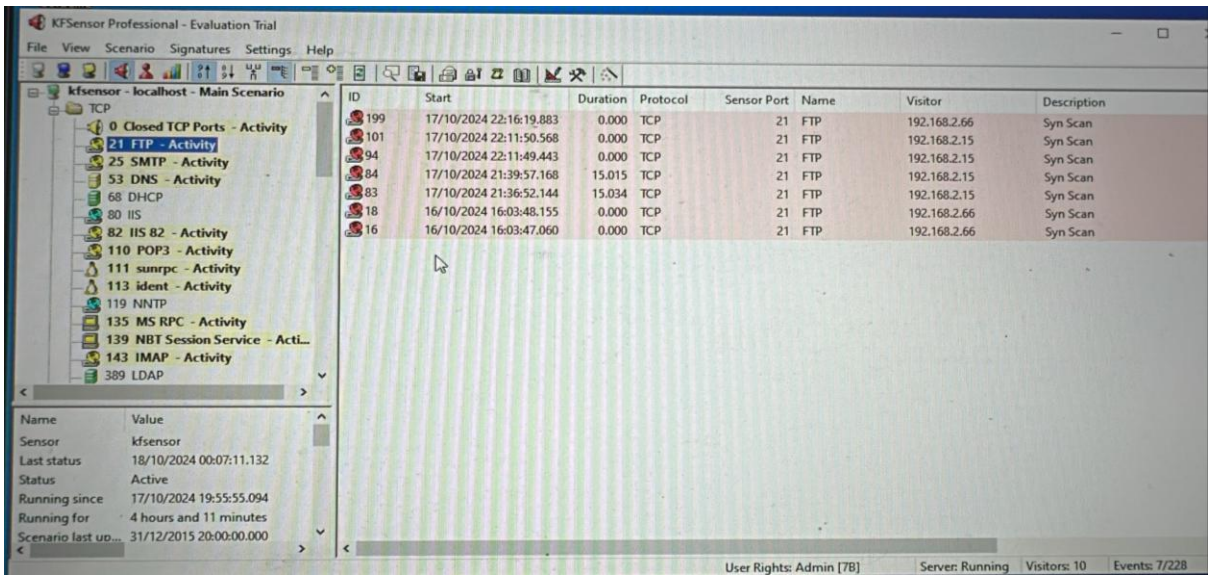

Figure 38 : Détection de la tentative d'intrusion sur le port FTP

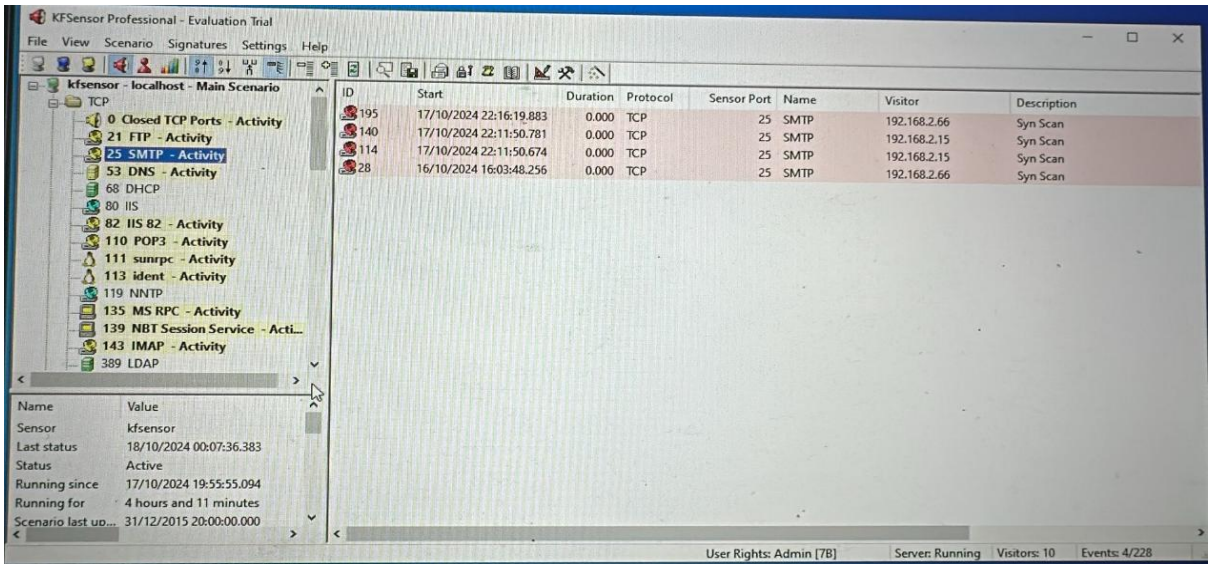

Figure 39 : Détection de la tentative d'intrusion sur le port SMTP

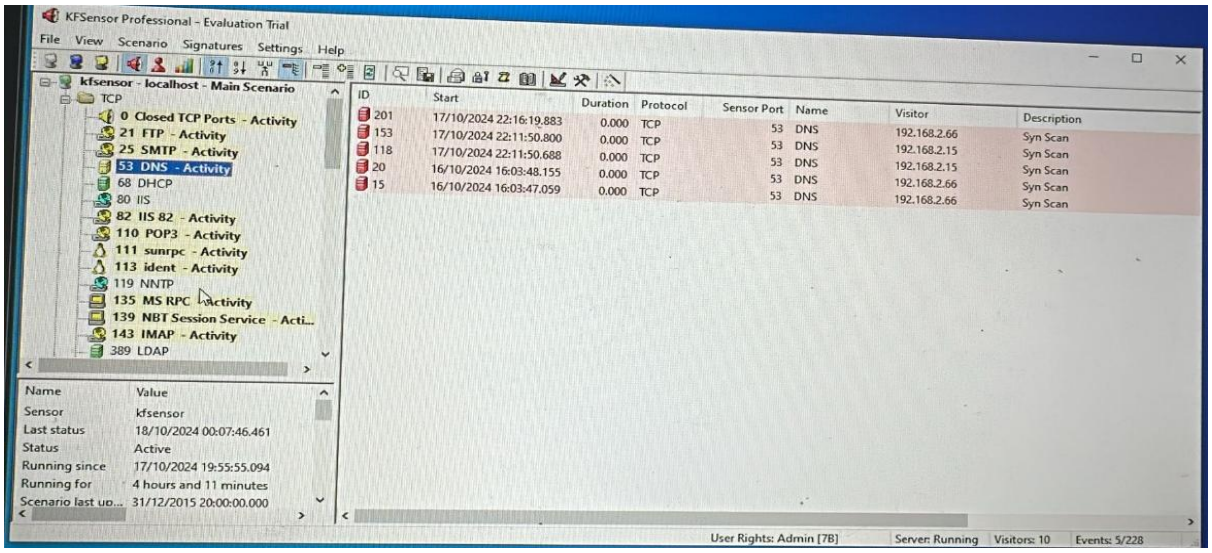

Figure 40 : Détection de la tentative d'intrusion sur le port DNS

5.12. Conclusion

Dans la première partie de ce chapitre, nous avons proposé une nouvelle approche de la théorie de jeux basée sur la tromperie avec une politique de redondance dans le placement de pots de miel afin rendre la tâche plus difficile aux attaquants. Un jeu à somme nulle a été formulé entre les deux joueurs et l'équilibre de Nash a été obtenu en stratégie mixte dans le placement de pots de miel.

A l'aide de l'algorithme fictitious play, nous avons examiné l'impact de la récompense du défenseur dans l'utilisateur de la redondance de pot de miel. Et nos analyses montrent que les nœuds qui ont de pots de miel en redondance, sont les mieux sécurisés. L'impact de leur

récompense du défenseur s'accroit au fur et à mesure que la redondance est accentuée. Et quand la probabilité de réussite de l'attaquant augmente, cela influence aussi en baisse la récompense du défenseur.

Nos résultats montrent que la redondance vient renforcer la sécurité de nœuds stratégiques et augmente en même temps la récompense du défenseur.

Dans la seconde partie, nous avons mis en pratique ce qui était abstrait dans la première partie. Nous avons configuré un réseau, subdiviser en 3 parties (WAN, LAN et DMZ).

D'abord, nous avons laissé fonctionner notre réseau sans pot de miel. Et nous avons orchestré une attaque par force brute, nous avons découvert que notre réseau était aveugle, car il n'a pas su signaler cette attaque.

Ainsi donc, nous avons déployé le premier pot de miel dans notre DMZ pour signaler toute tentative d'intrusion. Notre pot de miel à fonctionner convenablement. Mais nous avons ténu compte qu'il peut y avoir une attaque à l'interne, venant d'un agent qui est au service de l'ennemi, qui peut interrompre le service de notre pot de miel, rendant ainsi la détection d'intrusion impossible.

C'est alors que nous avons proposé un second pot de miel qui va fonctionner en parallèle et de manière autonome par rapport au premier pot de miel. C'est qui fait que si le premier pot de miel est attaqué ou cessé de fonctionner, le second continuera à détecter.

Nous retenons la solution de la mise en place d'une redondance de mécanisme de sécurité est une solution efficace, car elle obéit à la règle de la disponibilité de mesure de sécurité. Car si notre premier pot de miel subit une attaque interne ou externe, le temps qu'on remet le premier pot de miel à l'état, le second sera toujours en service.

CHAPITRE V

CONCLUSION

L'essor rapide des équipements connectés et des objets intelligents (IoT) a profondément transformé les infrastructures numériques, tout en accentuant leur exposition aux cybermenaces. Dans ce contexte, la sécurisation des réseaux, qu'ils soient civils ou militaires, exige des mécanismes de défense avancés capables de faire face à des attaques de plus en plus sophistiquées et adaptatives. Les approches de sécurité traditionnelles montrent aujourd'hui leurs limites, ce qui justifie l'intégration de nouveaux paradigmes tels que la cybertromperie.

Dans ce travail, nous avons tout d'abord réalisé un état de l'art approfondi sur l'utilisation de la théorie des jeux dans la cybersécurité, en particulier pour l'optimisation du placement des honeypots. Nous avons montré comment ces mécanismes de tromperie constituent un complément stratégique aux solutions classiques de sécurité en permettant de détourner les attaquants de leurs véritables cibles, d'observer leurs comportements et d'améliorer en continu les politiques de défense. Nous avons également présenté une classification des honeypots, leur fonctionnement, leurs avantages et leurs limites, ainsi que les enjeux liés à leur déploiement dans des environnements réseau complexes. L'apport principal de ce travail réside dans la proposition d'un cadre novateur fondé sur la redondance de honeypots diversifiés. Contrairement aux travaux existants qui se limitent soit au simple déploiement de honeypots, soit à leur diversification logicielle, notre approche combine ces deux dimensions dans une perspective de haute disponibilité et de résilience. À travers une modélisation en théorie des jeux, nous avons formalisé l'interaction stratégique entre l'attaquant et le défenseur, en intégrant la notion de tromperie proactive et les fonctions de récompense associées. Cette modélisation permet de mieux comprendre l'impact de la redondance sur l'équilibre du jeu et sur les gains respectifs des joueurs. Les résultats numériques obtenus confirment l'efficacité de l'approche proposée : la redondance des honeypots améliore significativement le gain du défenseur tout en réduisant celui de l'attaquant. Ces résultats mettent en évidence que la duplication stratégique de pots de miel, associée à leur hétérogénéité, renforce la capacité du système à absorber les attaques et à maintenir un niveau de surveillance constant, même en cas de compromission ou de défaillance d'un honeypot. Cette propriété confère au réseau une meilleure résilience opérationnelle. Par ailleurs, la mise en œuvre d'un environnement expérimental basé sur des machines virtuelles a permis de valider concrètement le concept de

redondance. Les scénarios testés ont démontré que l'arrêt brutal d'un honeypot unique rend le système aveugle face aux attaques, tandis que la présence d'un second honeypot indépendant assure la continuité de la détection et de la remontée des alertes. Cette validation expérimentale renforce la pertinence pratique du modèle théorique proposé. Comparée aux travaux antérieurs [16], [18] et [8], notre contribution se distingue par sa capacité à garantir la disponibilité des mécanismes de tromperie, tout en optimisant les stratégies de défense à travers la théorie des jeux. Elle s'inscrit ainsi comme une avancée significative dans le domaine de la cyber-déception, en proposant une solution robuste et pérenne pour la protection des infrastructures critiques. Cette approche a d'ailleurs été validée scientifiquement à travers la publication de notre article intitulé *"Game Theory: Cyber Deception Based on the Redundancy of Diversified Honeypots"* dans la revue *Procedia Computer Science* https://authors.elsevier.com/sd/article/S1877050925022124 .

Néanmoins, certaines limites ouvrent des perspectives de recherche intéressantes. Dans ce travail, le coût de la redondance a été modélisé de manière linéaire. L'introduction de fonctions de coût plus réalistes, hiérarchisées selon le type de honeypot et son niveau d'interaction, permettrait d'affiner l'analyse du compromis entre efficacité défensive et contraintes budgétaires. De plus, l'extension du modèle vers un cadre dynamique constitue une piste prometteuse. L'intégration de jeux stochastiques ou de techniques d'apprentissage adaptatif, telles que le Deep Fictitious Play ou l'apprentissage par renforcement, permettrait de prendre en compte l'évolution temporelle des stratégies d'attaque et de défense. Une telle approche offrirait la possibilité d'ajuster dynamiquement le nombre et le type de honeypots en fonction de la valeur des nœuds, de leur centralité dans le réseau et du niveau de menace observé. La mise en place de mécanismes prédictifs pour anticiper les attaques futures et déclencher automatiquement une redondance adaptative constitue également une direction de recherche pertinente.

En définitive, cette étude démontre que la redondance diversifiée des honeypots constitue un levier essentiel pour renforcer la résilience des réseaux critiques et assurer la continuité des stratégies de cybertromperie. En combinant diversification logicielle, redondance hétérogène et modélisation par la théorie des jeux, nous proposons un cadre méthodologique robuste capable d'accompagner l'évolution des menaces cybernétiques. Ce travail ouvre ainsi la voie à de futures recherches visant à concevoir des systèmes de défense autonomes, intelligents et adaptatifs, capables de faire face aux défis croissants de la cybersécurité moderne.

BIBLIOGRAPHIE

[1] A. H. Anwar, C. A. Kamhoua, N. O. Leslie, and C. Kiekintveld, "Honeypot allocation for cyber deception under uncertainty," *IEEE Transactions on Network and Service Management,* vol. 19, no. 3, pp. 3438-3452, 2022.

[2] A. H. Anwar, C. Kamhoua, and N. Leslie, "Honeypot allocation over attack graphs in cyber deception games," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, 2020: IEEE, pp. 502-506.

[3] W. Tian, M. Du, X. Ji, G. Liu, Y. Dai, and Z. Han, "Honeypot detection strategy against advanced persistent threats in industrial internet of things: A prospect theoretic game," *IEEE Internet of Things Journal,* vol. 8, no. 24, pp. 17372-17381, 2021.

[4] Q. D. La, T. Q. Quek, J. Lee, S. Jin, and H. Zhu, "Deceptive attack and defense game in honeypot-enabled networks for the internet of things," *IEEE Internet of Things Journal,* vol. 3, no. 6, pp. 1025-1035, 2016.

[5] S. Kim, "Game Theory Solutions for the Internet of Things: Emerging Research and Opportunities: Emerging Research and Opportunities," 2017.

[6] J. Antoniou and J. Antoniou, "Game theory and networking," *Game Theory, the Internet of Things and 5G Networks: Utilizing Game Theoretic Models to Characterize Challenging Scenarios,* pp. 1-20, 2020.

[7] M. H. Almeshekah and E. H. Spafford, "Planning and integrating deception into computer security defenses," in *Proceedings of the 2014 New Security Paradigms Workshop*, 2014, pp. 127-138.

[8] M. Zhu, A. H. Anwar, Z. Wan, J.-H. Cho, C. A. Kamhoua, and M. P. Singh, "A survey of defensive deception: Approaches using game theory and machine learning," *IEEE Communications Surveys & Tutorials,* vol. 23, no. 4, pp. 2460-2493, 2021.

[9] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Computing Surveys (CSUR),* vol. 51, no. 4, pp. 1-36, 2018.

[10] T. E. Carroll and D. Grosu, "A game theoretic investigation of deception in network security," *Security and Communication Networks,* vol. 4, no. 10, pp. 1162-1172, 2011.

[11] Z. Lu, C. Wang, and S. Zhao, "Cyber deception for computer and network security: Survey and challenges," *arXiv preprint arXiv:2007.14497,* 2020.

[12] N. Virvilis, B. Vanautgaerden, and O. S. Serrano, "Changing the game: The art of deceiving sophisticated attackers," in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, 2014: IEEE, pp. 87-97.

[13] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR),* vol. 45, no. 3, pp. 1-39, 2013.

[14] R. K. Shrivastava, S. Ramakrishna, and C. Hota, "Game theory based modified naïve-bayes algorithm to detect DoS attacks using Honeypot," in *2019 IEEE 16th India Council International Conference (INDICON)*, 2019: IEEE, pp. 1-4.

[15] J. Pawlick and Q. Zhu, *Game theory for cyber deception*. Springer, 2021.

[16] A. H. Anwar and C. Kamhoua, "Game theory on attack graph for cyber deception," in *International conference on decision and game theory for security*, 2020: Springer, pp. 445-456.

[17] A. H. Anwar, N. O. Leslie, and C. A. Kamhoua, "Honeypot allocation for cyber deception in internet of battlefield things systems," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, 2021: IEEE, pp. 1005-1010.

[18] A. A. Adebayo and D. B. Rawat, "Cyber deception for wireless network virtualization using stackelberg game theory," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021: IEEE, pp. 1-6.

[19] M. Crouse, "Performance analysis of cyber deception using probabilistic models," Wake Forest University, 2012.
[20] M. Crouse, B. Prosser, and E. W. Fulp, "Probabilistic performance analysis of moving target and deception reconnaissance defenses," in *Proceedings of the Second ACM Workshop on Moving Target Defense*, 2015, pp. 21-29.
[21] D. Fraunholz and H. D. Schotten, "Strategic defense and attack in deception based network security," in *2018 International Conference on Information Networking (ICOIN)*, 2018: IEEE, pp. 156-161.
[22] N. C. Rowe, E. J. Custy, and B. T. Duong, "Defending cyberspace with fake honeypots," *J. Comput.,* vol. 2, no. 2, pp. 25-36, 2007.
[23] M. Sithara, M. Chandran, and G. Padmavathi, "Outlier detection in secure shell honeypot using particle swarm optimization technique," *International Journal of Advanced Networking and Applications,* vol. 9, no. 3, pp. 3443-3450, 2017.
[24] E. Serra, S. Jajodia, A. Pugliese, A. Rullo, and V. Subrahmanian, "Pareto-optimal adversarial defense of enterprise systems," *ACM Transactions on Information and System Security (TISSEC),* vol. 17, no. 3, pp. 1-39, 2015.
[25] J. Liu, J. Yu, and S. Shen, "Energy-efficient two-layer cooperative defense scheme to secure sensor-clouds," *IEEE Transactions on Information Forensics and Security,* vol. 13, no. 2, pp. 408-420, 2017.
[26] J. Huang, H. Zhang, and J. Wang, "Markov evolutionary games for network defense strategy selection," *IEEE Access,* vol. 5, pp. 19505-19516, 2017.
[27] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Mathematics of operations research,* vol. 12, no. 3, pp. 441-450, 1987.
[28] V. Conitzer and T. Sandholm, "New complexity results about Nash equilibria," *Games and Economic Behavior,* vol. 63, no. 2, pp. 621-641, 2008.
[29] M. H. Almeshekah and E. H. Spafford, "Cyber security deception," *Cyber Deception: Building the Scientific Foundation,* pp. 23-50, 2016.
[30] J. J. Yuill, "Defensive computer-security deception operations: Processes, principles and techniques," 2007.
[31] Z. Guo, J.-H. Cho, R. Chen, S. Sengupta, M. Hong, and T. Mitra, "Online social deception and its countermeasures: A survey," *Ieee Access,* vol. 9, pp. 1770-1806, 2020.
[32] S. Tzu, "The art of war," in *Strategic Studies*: Routledge, 2008, pp. 63-91.
[33] D. Fraunholz *et al.*, "Demystifying deception technology: A survey," *arXiv preprint arXiv:1804.06196,* 2018.
[34] P. T. Duy, H. Do Hoang, N. H. Khoa, and V.-H. Pham, "Fool your enemies: Enable cyber deception and moving target defense for intrusion detection in SDN," in *2022 21st International Symposium on Communications and Information Technologies (ISCIT)*, 2022: IEEE, pp. 27-32.
[35] R. Joshi and A. Sardana, *Honeypots: a new paradigm to information security*. CRC Press, 2011.
[36] J. M. F. Marín, J. Á. M. Naranjo, and L. G. Casado, "Honeypots and honeynets: Analysis and case study," in *Handbook of research on digital crime, cyberspace security, and information assurance*: IGI Global, 2015, pp. 452-482.
[37] A. Yarali and F. G. Sahawneh, "Deception: Technologies and strategy for cybersecurity," in *2019 IEEE International Conference on Smart Cloud (SmartCloud)*, 2019: IEEE, pp. 110-120.
[38] N. C. Rowe, "Deception in defense of computer systems from cyber attack," in *Cyber warfare and cyber terrorism*: IGI global, 2007, pp. 97-104.
[39] B. Endicott-Popovsky, J. Narvaez, C. Seifert, D. A. Frincke, L. R. O'Neil, and C. Aval, "Use of deception to improve client honeypot detection of drive-by-download attacks," in *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience: 5th International Conference, FAC 2009 Held as Part of HCI International 2009 San Diego, CA, USA, July 19-24, 2009 Proceedings 5*, 2009: Springer, pp. 138-147.
[40] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings 15*, 2014: Springer, pp. 63-72.

[41] H. Okhravi *et al.*, "Survey of cyber moving targets," *Lincoln Laboratory-Massachusetts Institute of Technology Technical Report,* 2013.
[42] R. Poovendran, "Cyber–physical systems: Close encounters between two parallel worlds [point of view]," *Proceedings of the IEEE,* vol. 98, no. 8, pp. 1363-1366, 2010.
[43] P. Kathiravelu and L. Veiga, "SD-CPS: taming the challenges of cyber-physical systems with a software-defined approach," in *2017 Fourth International Conference on Software Defined Systems (SDS)*, 2017: IEEE, pp. 6-13.
[44] D. Serpanos, "The cyber-physical systems revolution," *Computer,* vol. 51, no. 3, pp. 70-73, 2018.
[45] A. M. Efendi, Z. Ibrahim, M. A. Zawawi, F. A. Rahim, N. M. Pahri, and A. Ismail, "A survey on deception techniques for securing web application," in *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 2019: IEEE, pp. 328-331.
[46] A. Kott, A. Swami, and B. J. West, "The internet of battle things," *Computer,* vol. 49, no. 12, pp. 70-75, 2016.
[47] S. Pinto, T. Gomes, J. Pereira, J. Cabral, and A. Tavares, "IIoTEED: An enhanced, trusted execution environment for industrial IoT edge devices," *IEEE Internet Computing,* vol. 21, no. 1, pp. 40-47, 2017.
[48] J. J. Rodrigues *et al.*, "Enabling technologies for the internet of health things," *Ieee Access,* vol. 6, pp. 13129-13141, 2018.
[49] M. Kocakulak and I. Butun, "An overview of Wireless Sensor Networks towards internet of things," in *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*, 2017: Ieee, pp. 1-6.
[50] A. Alshammari, D. B. Rawat, M. Garuba, C. A. Kamhoua, and L. L. Njilla, "Deception for cyber adversaries: status, challenges, and perspectives," *Modeling and Design of Secure Internet of Things,* pp. 141-160, 2020.
[51] D. C. MacFarland and C. A. Shue, "The SDN shuffle: Creating a moving-target defense using host-based software-defined networking," in *Proceedings of the second ACM workshop on moving target defense*, 2015, pp. 37-41.
[52] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 151-152.
[53] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," *ACM SIGCOMM computer communication review,* vol. 43, no. 4, pp. 7-12, 2013.
[54] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 435-442.
[55] K. Lee, B. Eoff, and J. Caverlee, "Seven months with the devils: A long-term study of content polluters on twitter," in *Proceedings of the international AAAI conference on web and social media*, 2011, vol. 5, no. 1, pp. 185-192.
[56] P. R. Badri Satya, K. Lee, D. Lee, T. Tran, and J. Zhang, "Uncovering fake likers in online social networks," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 2365-2370.
[57] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th annual computer security applications conference*, 2010, pp. 1-9.
[58] B. C. Ward *et al.*, "Survey of cyber moving targets second edition," *MIT Lincoln Laboratory Lexington United States, Tech. Rep,* 2018.
[59] J.-H. Cho *et al.*, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys & Tutorials,* vol. 22, no. 1, pp. 709-745, 2020.
[60] I. You and K. Yim, "Malware obfuscation techniques: A brief survey," in *2010 International conference on broadband, wireless computing, communication and applications*, 2010: IEEE, pp. 297-300.

[61] J.-M. Borello and L. Mé, "Code obfuscation techniques for metamorphic viruses," *Journal in Computer Virology,* vol. 4, no. 3, pp. 211-220, 2008.
[62] W. Xu, F. Zhang, and S. Zhu, "The power of obfuscation techniques in malicious JavaScript code: A measurement study," in *2012 7th International Conference on Malicious and Unwanted Software*, 2012: IEEE, pp. 9-16.
[63] C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *IFIP annual conference on data and applications security and privacy*, 2007: Springer, pp. 47-60.
[64] J.-T. Chan and W. Yang, "Advanced obfuscation techniques for Java bytecode," *Journal of systems and software,* vol. 71, no. 1-2, pp. 1-10, 2004.
[65] B. Whaley, "Toward a general theory of deception," *The Journal of Strategic Studies,* vol. 5, no. 1, pp. 178-192, 1982.
[66] S. Grazioli and S. L. Jarvenpaa, "Deceived: Under target online," *Communications of the ACM,* vol. 46, no. 12, pp. 196-205, 2003.
[67] N. Kuzurin, A. Shokurov, N. Varnovsky, and V. Zakharov, "On the concept of software obfuscation in computer security," in *Information Security: 10th International Conference, ISC 2007, Valparaíso, Chile, October 9-12, 2007. Proceedings 10*, 2007: Springer, pp. 281-298.
[68] B. Barak, "On the (im) possibility of software obfuscation," *Proc. of Advances in Cryptology (CRYPTO2001), 8,* pp. 1-18, 2001.
[69] J. B. Bell and B. Whaley, *Cheating and deception*. Routledge, 2017.
[70] L. Spitzner, "Honeypots: Sticking it to hackers," *NETWORK MAGAZINE-SAN FRANCISCO-,* vol. 18, no. 4, pp. 48-51, 2003.
[71] H. Kavak, J. J. Padilla, D. Vernon-Bido, S. Y. Diallo, R. Gore, and S. Shetty, "Simulation for cybersecurity: state of the art and future directions," *Journal of Cybersecurity,* vol. 7, no. 1, p. tyab005, 2021.
[72] R. Píbil, V. Lisý, C. Kiekintveld, B. Bošanský, and M. Pěchouček, "Game theoretic model of strategic honeypot selection in computer networks," in *Decision and Game Theory for Security: Third International Conference, GameSec 2012, Budapest, Hungary, November 5-6, 2012. Proceedings 3*, 2012: Springer, pp. 201-220.
[73] S. Garfinkel, "All about honeypots and honeynets," ed: CSO, 2003.
[74] M. Rouse, "Intrusion detection system (IDS)," *TechTarget [online]. Newton,* 2018.
[75] A. Sharma, "Honeypots in Network Security," *Int. J. Technol. Res. Appl,* vol. 1, pp. 7-12, 2013.
[76] F. Zhang, S. Zhou, Z. Qin, and J. Liu, "Honeypot: a supplemented active defense system for network security," in *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2003: IEEE, pp. 231-235.
[77] H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, "A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks," *computers & security,* vol. 25, no. 4, pp. 274-288, 2006.
[78] K. Sadasivam, B. Samudrala, and T. A. Yang, "Design of network security projects using honeypots," *Journal of Computing Sciences in Colleges,* vol. 20, no. 4, pp. 282-293, 2005.
[79] D. Fraunholz, D. Krohmer, S. D. Anton, and H. D. Schotten, "Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot," in *2017 international conference on cyber security and protection of digital services (cyber security)*, 2017: IEEE, pp. 1-7.
[80] F. Ja'fari, S. Mostafavi, K. Mizanian, and E. Jafari, "An intelligent botnet blocking approach in software defined networks using honeypots," *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, pp. 2993-3016, 2021.
[81] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A comprehensive survey on cyber deception techniques to improve honeypot performance," *Computers & Security,* p. 103792, 2024.
[82] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: A novel honeypot for revealing current IoT threats," *Journal of Information Processing,* vol. 24, no. 3, pp. 522-533, 2016.
[83] A. G. Manzanares, "Honeyio4: the construction of a virtual, low-interaction iot honeypot," Universitat Politècnica de Catalunya. Escola Politècnica Superior d …, 2017.

[84] W. Fan, Z. Du, D. Fernández, and V. A. Villagra, "Enabling an anatomic view to investigate honeypot systems: A survey," *IEEE Systems Journal,* vol. 12, no. 4, pp. 3906-3919, 2017.
[85] C. Stoll, *The cuckoo's egg: tracking a spy through the maze of computer espionage*. Simon and Schuster, 2005.
[86] Z. A. Khan and U. Abbasi, "Reputation management using honeypots for intrusion detection in the internet of things," *Electronics,* vol. 9, no. 3, p. 415, 2020.
[87] W. Fan, Z. Du, M. Smith-Creasey, and D. Fernandez, "Honeydoc: an efficient honeypot architecture enabling all-round design," *IEEE journal on selected areas in communications,* vol. 37, no. 3, pp. 683-697, 2019.
[88] M. Wang, J. Santillan, and F. Kuipers, "ThingPot: an interactive Internet-of-Things honeypot," *arXiv preprint arXiv:1807.04114,* 2018.
[89] N. Naik, C. Shang, P. Jenkins, and Q. Shen, "D-FRI-Honeypot: A secure sting operation for hacking the hackers using dynamic fuzzy rule interpolation," *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 5, no. 6, pp. 893-907, 2020.
[90] A. Zarras, "The art of false alarms in the game of deception: Leveraging fake honeypots for enhanced security," in *2014 International Carnahan Conference on Security Technology (ICCST)*, 2014: IEEE, pp. 1-6.
[91] V. A. Perevozchikov, T. A. Shaymardanov, and I. V. Chugunkov, "New techniques of malware detection using FTP Honeypot systems," in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2017: IEEE, pp. 204-207.
[92] A. McCarthy, E. Ghadafi, P. Andriotis, and P. Legg, "Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey," *Journal of Cybersecurity and Privacy,* vol. 2, no. 1, pp. 154-190, 2022.
[93] S. Kumar, R. Sehgal, and J. Bhatia, "Hybrid honeypot framework for malware collection and analysis," in *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, 2012: IEEE, pp. 1-5.
[94] A. El-Kosairy and M. A. Azer, "A new Web deception system framework," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 2018: IEEE, pp. 1-10.
[95] W. Z. A. Zakaria and M. L. M. Kiah, "A review on artificial intelligence techniques for developing intelligent honeypot," in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, 2012, vol. 2: IEEE, pp. 696-701.
[96] R. Honarbakhsh and O. Zakaria, "A Survey on Dynamic Honeypots," *International Journal of Information and Electronics Engineering,* vol. 2, no. 2, p. 233, 2012.
[97] S. Laurén, V. Leppänen, S. Rauti, and J. Uitto, "A Survey on Anti-honeypot and Anti-introspection Methods," *Recent Advances in Information Systems and Technologies,* vol. 2, pp. 11-13, 2017.
[98] I. Kuwatly, M. Sraj, Z. Al Masri, and H. Artail, "A dynamic honeypot design for intrusion detection," in *The IEEE/ACS International Conference onPervasive Services, 2004. ICPS 2004. Proceedings.*, 2004: IEEE, pp. 95-104.
[99] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A survey on honeypot software and data analysis," *arXiv preprint arXiv:1608.06249,* 2016.
[100] S. Lance and R. Marty, "The value of honeypots, part one: Definitions and values of honeypots," ed: April, 2001.
[101] I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *Proceedings of the 45th annual southeast regional conference*, 2007, pp. 321-326.
[102] T. Kaur, V. Malhotra, and D. Singh, "Comparison of network security tools-firewall, intrusion detection system and Honeypot," *Int. J. Enhanced Res. Sci. Technol. Eng,* vol. 200204, 2014.
[103] L. Spitzner, "The value of honeypots, part one: Definitions and values of honeypots," *Security Focus,* 2001.
[104] A. A. Ashour, *Designing high interaction windows honeynets*. University of Louisville, 2006.
[105] A. H. Anwar and C. A. Kamhoua, "Cyber deception using honeypot allocation and diversity: A game theoretic approach," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022: IEEE, pp. 543-549.

[106] D. ARMY, "Army support to military deception," *FM,* vol. 3, p. 4, 2019.

[107] V. Daniel, *Cyberattaque et cyberdéfense*. Lavoisier, 2011.

[108] M. Capó, A. Pérez, and J. A. Lozano, "An efficient K-means clustering algorithm for massive data," *arXiv preprint arXiv:1801.02949,* 2018.

[109] N. Cifranic, R. A. Hallman, J. Romero-Mariona, B. Souza, T. Calton, and G. Coca, "Decepti-SCADA: A cyber deception framework for active defense of networked critical infrastructures," *Internet of Things,* vol. 12, p. 100320, 2020.

[110] N. C. Rowe and H. C. Goh, "Thwarting cyber-attack reconnaissance with inconsistency and deception," in *2007 IEEE SMC Information Assurance and Security Workshop*, 2007: IEEE, pp. 151-158.

[111] G. F. Lyon, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[112] O. Tsemogne, Y. Hayel, C. Kamhoua, and G. Deugoué, "Game-theoretic modeling of cyber deception against epidemic botnets in internet of things," *IEEE Internet of Things Journal,* vol. 9, no. 4, pp. 2678-2687, 2021.

[113] A. H. Anwar, N. O. Leslie, and C. A. Kamhoua, "Honeypot allocation for cyber deception in internet of battlefield things systems," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, 2021: IEEE, pp. 1005-1010.

[114] C. Leita, K. Mermoud, and M. Dacier, "Scriptgen: an automated script generation tool for honeyd," in *21st Annual Computer Security Applications Conference (ACSAC'05)*, 2005: IEEE, pp. 12 pp.-214.

[115] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[116] J. F. Nash Jr, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences,* vol. 36, no. 1, pp. 48-49, 1950.

Annexe

ARTICLE SCIENTIFIQUE

Les auteurs : Malamu Dumisani Benjamin et Ould-Slimane Hakima

Soumis au journal Procedia Computer Science (La 20e Conférence internationale sur les réseaux et les communications du futur (FNC2025) Aout 4-6, 2025, Leuven, Belgique). http://cs-conferences.acadiau.ca/fnc-25/

Numéro papier : 265 (2025) 107–115

Soumis le 11 avril 2025

Accepté le 03 mai 2025

Publié le 21 Août 2025 (https://authors.elsevier.com/sd/article/S1877050925022124)

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 265 (2025) 107–115

Procedia Computer Science

www.elsevier.com/locate/procedia

The 20th International Conference on Future Networks and Communications (FNC)
August 4-6, 2025, Leuven, Belgium

Game Theory: Cyber Deception Based on the Redundancy of Diversified Honeypots

Benjamin Malamu Dumisani[a]*, Hakima Ould-Slimane[b]

[a,b]Université du Québec à Trois-Rivières, 3351, Bd des forges, Trois-Rivières G8Z 4M3, Québec Canada

Abstract

Today, cyber deception is an essential element of proactive and reactive defense systems. Military equipment is increasingly connected to the Internet, and next-generation network security is becoming increasingly complex. This vast network of military tactics thus becomes exposed and vulnerable, opening a battlefield against cyberattacks. The security of such a network is therefore paramount. In this paper, we propose a game-theoretic cyber deception approach for the allocation of diversified honeypots. To ensure high availability of our security measure, we opt for diversified redundancy in honeypot placement. We will analyze the impact on the defender's reward, with or without the attacker's success probability. The optimization will be solved using the Nash balance. Regarding honeypot placement, we adopt a deployment with redundancy to guarantee the security of critical nodes. Our results show that redundancy strengthens the ability to diversify the honeypot, thus ensuring long-term network security.

Keywords: Honeypots, Cybersecurity, Cyberdefense, Redundancy, Game theory;

1. Introduction

Protecting critical systems and assets has become a major concern in our interconnected world, where a single disruption can ripple across multiple sectors of society. For example, in 2019 the US Department published a

* Corresponding author.
E-mail address: benjamin.malamu.dumisani@uqtr.ca

10.1016/j.procs.2025.07.162

document on its military deception strategy for multi-domain operations [17], including network security [18]. Cyber threats to country security are very difficult to anticipate, to destroy, to coerce, and increasingly frequent. Targets may be military, industrial or civilian [5]. With the rise of next- generation networks (5G, IoT, cloud computing), cyber threats are intensifying. Their interconnected nature increases attack surfaces, requiring advanced detection, prevention, and response mechanisms.

In most cases, cyber threat solutions do not consider the probability of attacker success and others minimize the fact that an attack can come from within the company itself. To address the problem of sustainability and resilience of security measures, redundancy is part of the solution to be implemented. To avoid attacks, near-total security is needed [10], yet traditional honeypots are often passive and ineffective against skilled adversaries [1], leaving critical infrastructures like transport, logistics, and military or nuclear sites vulnerable to threats such as piracy, espionage, and sabotage. In addition to their configuration, which is often extensive and with a perimeter difficult to monitor, there are numerous inputs from stakeholders to control and traffic flows to be optimized. In this work, we will propose an approach to cyber deception based on diverse honeypot redundancy, firstly formulating a zero-sum game model and modelling the actions of each actor. Next, we will solve the mixed strategy optimization problems with Nash equilibrium. We define the reward function of each player, and we will simulate a zero-sum game with the fictitious play algorithm to evaluate the reward of players. And we will evaluate the impact of redundancy in the placement of diversified honeypots in the software configuration. In [8] software diversification is highlighted but after our reflections we found that a honey pot can meet an internal malfunction and shut down without this being attacked, Hence the importance of redundancy technique in the placement of honey pots. The principle of this technique is to use several different configurations of honeypots on different hardware architectures and the defense in depth applied to cybersecurity which consists in a set of security systems whose perimeters of action are relayed. The goal of this architecture is to ensure that if a honeypot is attacked or bypassed by an attacker, this attack will be ineffective against the second honeypot having a different configuration than the one being attacked. As attackers gather information about networks before launching malicious activities [9] using tools like the Nmap [15], Deception is of great importance in distorting the state of the network and the information that could be collected during the recognition phase [8].

2. Related work

Our research builds upon previous studies in cyber deception, cybersecurity, game theory, and honeypot placement.

Defensive deception techniques aim to conceal valuable network information to create uncertainty for attackers [14]. Deception has historically provided a strategic advantage in conflicts [1] and remains a key cyber defense mechanism that misleads attackers with false information [3]. The success of deception depends on what both attackers and defenders can observe [16]. However, most cybersecurity studies focus on information dissemination and confidentiality rather than the impact of an intruder on a system's physical components. Game theory provides a mathematical framework for decision-making under uncertainty [2]. Recent works [21,22] apply game theory to IoT security, while others explore Nash equilibrium-based deception strategies [20]. A zero-sum game model has been proposed where honeypots serve as a deception tool for defenders [23]. Research demonstrates that game theory effectively optimizes cyber deception strategies to counter adversaries in different network security contexts [12,13].

Honeypots act as decoy systems that trap attackers while remaining invisible to legitimate users [2]. Existing approaches propose proactive and reactive honeypot placement strategies [16], evolutionary algorithms for honeypot allocation [18], and strategic game models for defending industrial networks [24]. However, most security strategies fail to account for evolving attacker tactics, potentially increasing their success rate. [8] proposed a cyber-deception approach using honeypot allocation and software diversity to enhance network security and considered a trade-off between the level of software diversity and operational cost in using different types of honeypots.

By going through these various related works, we find that security measures are deployed without considering the current and future ingenuity of attackers, which can give them a chance of success. Therefore, we propose a solution based on diversified redundancy to ensure the sustainability of the security mechanism in the face of attackers' ingenuity and their chances of success during their first attacks. Our contribution in this paper can be summarized as follows: we first propose a game-theoretic approach for honeypot allocation. We model a zero-sum game to find the Nash equilibrium between the two players. Then, by applying redundancy, we will evaluate its

impact on the defender's reward by considering the cost. Finally, we provide the numerical results of the approach to show its impact on the defender's reward.

3. Modeling the network

In this section, we formulate the game model between the defender and the attacker, and the network model.

3.1. Model Network

The network is represented by a graph $G = (V, E)$,Where : V is the set of nodes, including strategic nodes. E is the set of edges representing the connections between nodes. Redundant honeypots are placed in front of critical nodes. and we have H represents the set of honeypots, while V consists of both simple and strategic nodes, denoted as $V = V_s \bigcup V_r$. SW refers to the software deployed on honeypots, and R_h indicates the number of redundant honeypots assigned to each node. The SWV matrix, $SWV[s, v]$, represents the probability that a software s has a vulnerability v, whereas $P[s, v]$ specifies whether an exploit b can successfully exploit the vulnerability v. With software assignment matrix: $HSW[h, s] = \begin{cases} 1, & \text{si le logiciel } s \text{ est déployé sur le honeypot } h \\ 0, & \text{sinon} \end{cases}$

3.2. *Defender Model and Attacker Model*

The defender secures strategic nodes by deploying diverse and redundant honeypots, selecting software for each. His strategy is defined as $A_d = \{y_{s_h} \in [0,1] \mid \sum_{s_h \in SW} y_{s_h} = 1\}, \forall\, h \in H$, where y_{s_h} represents the probability of assigning software s_h to honeypot h. Redundancy is ensured with $R_h \geq 1$, reinforcing protection through multiple honeypots per strategic node. The defense cost is given by $C_D(a_d) = \sum_{h<H} \sum_{h<H} c_s y_{s_h}$, depending on the selected software. The attacker aims to compromise critical nodes by exploiting vulnerabilities. His strategy is defined as $A_a = \{x_b \in [0,1] \mid \sum_{b \in B} x_b = 1\}$, where x_b represents the probability of using exploit b. The attack cost is given by $C_D(a_d) = \sum_{b \in B} c_b x_b$

4. Formulation of the game model

A two-player zero-sum game is defined as a tuple Γ (K, A, R), where K = {Defender, Attacker} represents the set of two players (d: Defender and a: Attacker). The action space is given by $\mathcal{A} = \mathcal{A}_d$ x $\mathcal{A}_a$, where $A_d = \{y_{s_h} \in [0,1] \mid \sum_{s_h \in SW} y_{s_h} = 1\}$ defines the defender's strategy, and $A_a = \{x_b \in [0,1] \mid \sum_{b \in B} x_b = 1\}$ represents the attacker's strategy, with $x_b$denoting the probability of using exploit b.

4.1. *Modeling with Redundancy and Gain Matrix with Redundancy*

Each strategic node h is protected by multiple honeypots H_h, each running different software. The probability that softwares_h has vulnerability v is given by $SWV[b, v]$, while $P[s_{h'}, v]$ indicates whether exploit b can attack vulnerability v. The redundancy level is defined as R_h=|H_h|. The attacker selects an exploit with probability x_b, while the defender assigns software sh to honeypot h with probability y_{sh}.

The attacker must pass all honeypots before compromising the strategic node h, reducing his chances of success. The attacker's gain becomes:

$$U_A(a_a, a_d) = \sum_{b \in B} x_b \sum_{h \in H} \mathrm{P}.[b, v_h] .(1 - \prod_{h \in H} (1 - SWV[s_{h'}, v_h])) \; V_h - c_b.x_b \qquad (1)$$

Where $\prod_{h \in H}(1 - SWV[s_{h'}, v_h])$ represents the probability that all honeypots resist the attack. Increasing R_h

enhances this probability, reducing the attacker's gain.
The gain of the defender is then:

$$U_D(a_d, a_a) = -U_A(a_a, a_d) \sum_{b \in B} x_b \sum_{h' \in H} c_{s_{h'}} y_{s_{h'}} \quad (2)$$

4.2. *Linear optimization for Nash equilibrium*

The Nash balance is achieved when no player can improve his or her winnings by unilaterally modifying his or her strategy, i.e., when the defender and attacker adopt optimal mixed strategies that stabilize themselves. The Nash balance implies that the attacker chooses an optimal distribution x_b of exploits to maximize his minimum gain. The defender chooses an optimal distribution y_{s_h} of the software on the honeypots to maximize his own minimum gain. The winnings of both players are equal in balance, either $u_A^* = u_D^*$.

4.3.2. ***Linear Programming Problem Formulation***

- *Attack Program (Maximization of his Minimum Gain)*

$$\max U_A$$

Under constraints:

$$U_A \leq \sum_{b \in B} x_b \sum_{h \in H} P[b, v_h](1 - \prod_{h' \in H} (1 - SWV[s_{h'}, v_h]))\, V_h - c_b x_b \quad (3)$$

$$\sum_{b \in B} x_b = 1, \qquad x_b \geq 0 \quad \forall x_b \in B$$

- *Defender's program (Maximization of his minimum Gain)*

Under constraints:

$$U_D \leq -\sum_{b \in B} x_h \sum_{h \in H} P[b, v_h](1 - \prod_{h' \in H} (1 - SWV[s_{h'}, v_h]))\, V_h - \sum_{h \in B} \sum_{h \in H} c_{s_{h'}} y_{s_{h'}} \quad (4)$$

$$\sum_{s_h \in SW} y_{sh} = 1, \qquad y_{sh} \geq 0 \quad \forall h \in H$$

4.3.3. Solving the Nash balance and reward Function with redundancy of diversified honeypots

The Nash equilibrium is found by simultaneously solving these two linear programs by introducing a minimax formulation. We're looking for a saddle-point where the winnings of the attacker and the defender are equal:

$$\max_{y} \min_{x} U_D(a_d, a_a) = \min_{x} \max_{y} U_A(a_a, a_d) \quad (5)$$

By applying the dual linear programming approach, we obtain the gain equalization condition: $u_A^* = u_D^*$. This implies that the solutions x_b^* and $y_{s_h}^*$ must satisfy this linear constraint.

Reward of the Attacker $U_A(a_a, a_d)$: The attacker seeks to maximize the value of the compromised nodes while minimizing the cost of the attack. The effectiveness of the attack depends on the exploits b choosen; the associated vulnerabilities and the resistance offered by redundant honeypots. The probability that the attacker succeeds in compromising the node h is reduced by redundancy R_h.

$$U_A(a_a, a_d) = \sum_{b \in B} x_b \sum_{h \in H} \mathrm{P}[b, v_h] \cdot (1 - \prod_{h' \in H} (1 - SWV[s_{h'}, v_h]))\ V_h - c_b x_b \quad (6)$$

With $\mathrm{P}[b, v_h]$ the Indicates whether the exploit can exploit the vulnerability v_h. $\prod_{h' \in H}(1 - SWV[s_{h'}, v_h]))$ is the represents the probability that all honeypots will resist the attack. If V_h Value of the node h if it is compromised. $c_b x_b$ is the cost of the attack (use of exploits). The impact of redundancy shows that as R_h increases, the probability of attack failure also rises, thereby reducing U_A. Consequently, the attacker must use more exploits, increasing their overall cost.

Defender's Award $U_D(a_d, a_a)$: The defender seeks to minimize the impact of the attack and reduce the cost of defense.

$$U_D(a_d, a_a) = -U_A(a_a, a_d) - \sum_{h \in H} x_b \sum_{h' \in H} c_{s_h, y_{s_{h'}}} \quad (7)$$

The defender's loss consists of $-U_A(a_a, a_d)$ from successful attacks and $\sum_{h \in H} x_b \sum_{h' \in H} c_{s_h, y_{s_{h'}}}$ for honeypot deployment and maintenance. Increasing R_h raises costs but reduces attack success, enhancing security. High redundancy is optimal when protecting high-value nodes.

5. Application of redundancy in the placement of diversified honeypots

To ensure resilience and sustainability in the placement of honeypots as proposed by [8], it is imperative not to rely solely on the diversification of the types of honeypots. Because, since the honey pot is corrupted, the voice remains free for the attacker and he can afford everything.

Therefore, we offer beyond the honey pot diversification, redundancy in the honey pot placement. Honey pot redundancy is the act of putting several honey pots in the same stop so that if one breaks down, the other replaces it. This ensures that there is no interruption in securing the target network or node. In other words, it ensures the availability of safety measures in the event of a successful breakdown or corruption of a honey pot. If one is compromised, the attacker will be whacked at the second honey pot of different configuration. This will delay his progress and allow the defender to make the necessary arrangements for the security of the network.

Fig. 1: (a) network architecture without redundancy of diversified honeypot; (b) network architecture with redundancy of diversified honeypot

Starting from the same topology, in Fig. 1 (a) where we have one input node, two single nodes, three intermediate nodes and four strategic nodes. We notice that the 4 strategic nodes each have a single pot of honey. That being, if the attacker succeeds in compromising the single honeypot, directly he is in front of the strategic node.

In addition, honey pots are not only subjected to attacks from the outside. A honey jar can turn off because it's been disconnected from the power grid, it can be shorted, it can be sabotaged internally and it breaks down, exposing the node it's supposed to be secured. In Fig. 1 (a), if the single pot of honey fails due to an internal error as listed, we notice that the attacker will have the field clear, leading him directly to his target. Yet in Fig. 1 (b), the

setting up of honey pot redundancy, solves the problem mentioned in the previous paragraph. We can notice that if the first honey pot arrives be corrupted or if it falls due to an internal dislocation, the attacker will be facing the second honey pot which is in redundancy with the first. Therefore, our target node is always safe.

Certainly, the redundancy will generate an additional operating cost for the defender. However, it enhances network security by increasing a second layer of protection. So, if the first honeypot is attacked through a vulnerability, the attacker is confronted with the second honeypot, giving the defender time to review his security policy.

Attacker reward: The attacker's reward is calculated as: 1−average probability of success1. This reflects the fact that the higher the probability of success of the honeypots, the less likely the attacker is to succeed, so the attacker's reward decreases.

5.1. Notations and definitions:

H is the set of honeypots, where $|H|$ represents the number of honeypots; SW is the set of software types, with | SW | denoting the number of software types; V is the set of vulnerabilities, with | V | representing their total number; and B is the set of exploits, with | B| indicating the number of exploits. The HSW matrix, of dimension $|H|$ x $|SW|$, defines the assignment of software to honeypots, while the SWV matrix, of dimension $|SW|$ x $|V|$, where $0 \leq SWV[i,j] \leq 1$, represents the probability that a software s_i has a vulnerability v_j. Additionally, P is a binary matrix of dimension $|B| \times |V|$, where $P[i,j] = 1$ if the exploit b_i can exploit the vulnerability v_j.

5.2. *Actions and costs*

The attacker's action space (A_a) is defined by the set of B exploits. The defender's action space (A_d) is made up of SW software deployed on honeypots H. The cost of the attack is calculated as the sum of the costs associated with the use of the exploits $c_b(a_a)=\sum_{b\in B} c_b \cdot a_a(b)$, where c_b is the cost of exploit b and $a_a(b)$ is an indicator of its use. The cost of the defense is given by $c_b(a_d)=\sum_{h\in H} c_{sh}$, where c_{sh} is the cost associated with the use of software s_h for honeypot h.

5.3. *Reward Function*

The diversity of honeypots is essential to avoid repeated attacks using the same exploits. To achieve this, we will use the greedy algorithm, which maximizes diversity while controlling costs. At each step, the software type assignment for each honeypot is selected to minimize the impact on the overall cost while ensuring diversity. Specifically, at each iteration, for each honeypot h, we choose a software s_h such that $s_h \notin \{s'_h \mid h' \text{ neighbor of } h\}$, to guarantee diversity, while minimizing the total defense cost.

Attacker reward ($\mathcal{R}_a$): The attacker wants to maximize his gains while minimizing his costs. The attacker's reward is proportional to the probability of success of his exploits on the honeypots. It is defined as follows

$$\mathcal{R}_a(a_a, a_d) = \sum_{b \in B} P(b, a_d).gain\ (b, a_d) - c_b(a_a) \tag{8}$$

With $P(b, a_d)$ is the probability that exploit b will succeed, depending on the software deployed by the defender. $gain(b, a_d)$ is the gain the attacker gets if exploit b succeeds. The term $c_b(a_a)$ represents the cost of the attack.

His reward can be expressed *by:*

$$\mathcal{R}_d(a_a, a_d) = \sum_{h \in H} (1 - P(b, a_d)).gain\ défense\ - c_d(a_d) \tag{9}$$

With $(1 - P(b, a_d))$ represents the effectiveness of the honeypot software s_h against exploit b. $gain\ defense$ is the win the defender gets for each failed exploit by the attacker. The term $c_b(a_d)$ represents the cost of defense.

6. Digital results

In this section, we present our numerical results that validate the proposed approach to the allocation of honeypots optimally by using redundancy to maximize the gain of the defender. Our analysis ale network topology having 1 single input node in purple, 2 single nodes in green, 3 intermediate nodes in blue and 4 strategic nodes in red. Each single node has a value of 50, intermediate nodes to a value of 150 and 3 strategic nodes have a respective value of 200 and the last strategic node has a value of 250 and the input node 50.

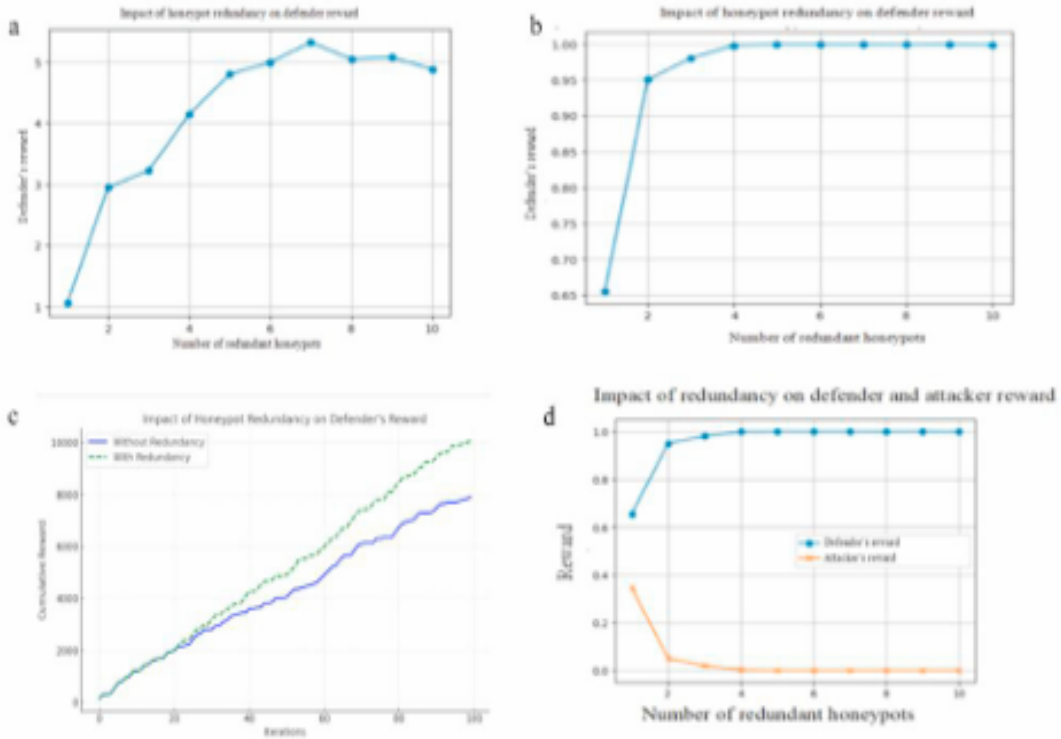

Fig. 2: (a) with probability of success of the attacker; (b) without probability of success of the attacker; (c) evolution of the reward of the defender as well as that of the attacker; (d) combination chart the impact of redundancy of honeypots in the Defender's reward and attacker's.

The fictitious play algorithm allows us to model and analyze the strategic interaction between an attacker and a defender in a network. We simulate a game between the two players. The attacker targets the nodes to maximize his win. The defender uses a mixed strategy to protect the critical nodes. The reward per redundancy level is 0.5 and the redundancy level is 0 to 10. Fig. 1 (a) shows the reward of the defender with probability of success of the attacker. And in fig.1 (b), we can see the impact of the reward of the defender without the probability of success of the attacker Using random simulation, we examine the impact of the attacker's probability of success on the reward of the defender in fig. 3 (a) and (b), while taking into account the redundancy, for example, with probability in the order of 0.1, 0.3, 0.5, and 0.7. In fig. 3 (a), as p_{succes} it increases (from 0.1 to 0.7), the reward of the defender decreases rapidly. The curves show that the defender is most vulnerable when the attacker has a high probability of success. In fig.2 (c), the curves show a higher reward for the defender, even when $p_{successes}$ is high. Redundancy effectively reduces the negative impact of the attacker's high probability of success.

Similarly in fig. 1 (c), the blue line shows us that the honey pots are placed in a non-redundant manner and the green line shows us that the honey pots are placed in a redundant manner on strategic edges. The green line (with redundancy) shows a significant increase in cumulative reward over iterations. This shows that the use of redundancy improves the effectiveness of the defensive strategy, increasing the probability of intercepting the attacker. In fig. 1(d), the graph shows the defender's and attacker's reward as a function of the number of redundant honeypots. The defender's reward increases with redundancy, while the attacker's reward tends to 0.

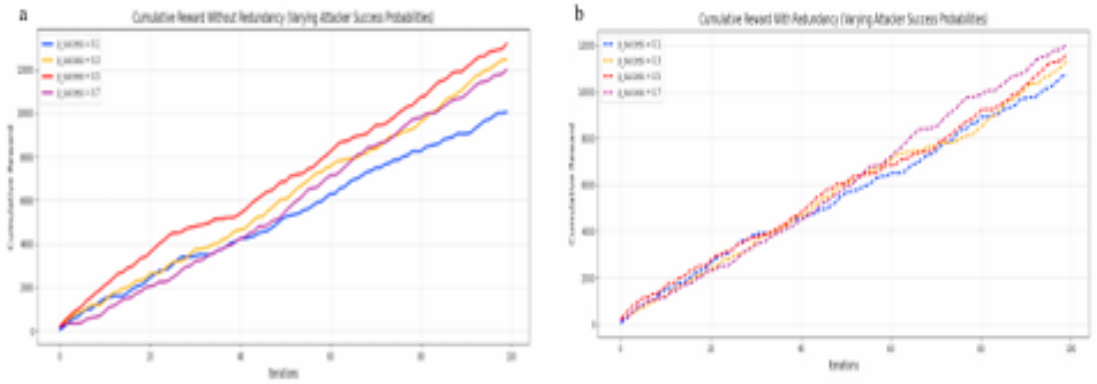

Fig.3: (a) cumulative reward without redundancy (Attacker's probability of success); (b) Accumulated reward with redundancy (Attacker's odds of success)

With a cost of a honeypot estimated at 1 arbitrary unit. We have 4 strategic nodes to protect without redundancy, the total cost is units for the 4 honeypots. With redundancy, the cost will be 8 units for 8 honeypots. Because of 2 honeypots per strategic node. We can simulate and display this increasing cost as a function of the redundancy level from 1 to 3 honeypots per strategic node. As we can see in Fig. 4, the increase in defense cost is linear with the redundancy level of honeypots per strategic node.

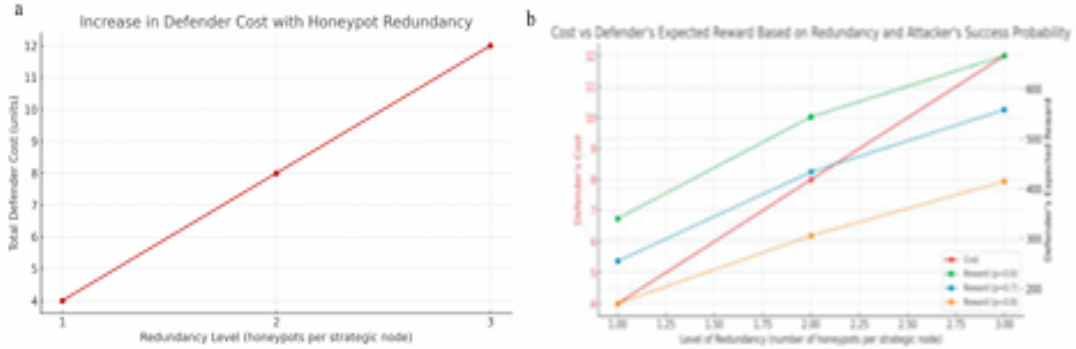

Fig. 4: (a) Cost and redundancy ratio without probability of success; (b) Cost and redundancy ratio with probability of success

Redundancy improves security, but it incurs a significant additional cost for the defender. Compared to different attacker success probabilities (e.g., 0.6, 0.7, 0.8), as shown in Fig. 4(b), the more efficient the attacker is (higher p), the greater the defender's incentive to increase the number of honeypots to compensate.

Compared to the work of [16], [18] which simply proposes the deployment of honeypots and [8] which is simply limited to the diversification of honeypots, our approach is better because it aims at high availability of security measure.

7. Conclusion

In this work, we proposed a new approach to game theory based on deception with a policy of redundancy in the placement of honeypots to make the task more difficult for attackers. A zero-sum game was formulated between the two players and the Nash balance was obtained in mixed strategy in the placement of honeypots.

Using the fictitious play algorithm, we examined the impact of the defender reward in the user of the honeypot redundancy. And our analysis shows that the nodes that have redundant honeypots, are the most secure. The impact of their defender reward increases as redundancy increases. And when the attacker's probability of success increases, it also decreases the reward of the defender.

Our results show that redundancy enhances the security of strategic nodes and at the same time increases the reward to the defender.

Acknowledgements

The authors would like to thank Mrs. Suzanne Mukundi Busara for English editing this paper.

References

[1] N. Cifranic, R. A. Hallman, J. Romero-Mariona, B. Souza, T. Calton, and G. Coke, . "Decepti-SCADA: A cyber deception framework for active defense of networked critical infrastructures," Internet of Things, vol. 12, p. 100320, 2020.

[2] K. Durkota, V. Lisý, B. Bošanský, C. Kiekintveld, and M. Pěchouček, "Hardening networks against strategic attackers using attack graph games," Computers & Security, vol. 87, p. 101578, 2019.

[3] O. Tsemogne, Y. Hayel, C. Kamhoua, and G. Deugoué, . "Game-theoretic modeling of cyber deception against epidemic botnets in internet of things," IEEE Internet of Things Journal, Vol. 9, no. 4, pp. 2678-2687, 2021.

[4] T. Başar and G. J. Olsder, Dynamic noncooperative game theory. SIAM, 1998.

[5] V. Daniel, Cyber Attack and Cyber Defense. Lavoisier, 2011.

[6] F. Douzet and A. Géry, "Cyberspace is used, first of all, to wage war. The Proliferation, Security and Stability of Cyberspace," (in Fr), Herodotus, Vol. 177-178, no. 2, pp. 329-350, 2020, doi: 10.3917/her.177.0329.

[7] J. Nash Jr, "Equilibrium Points in N-Person Games," in Essays on Game Theory: Edward Elgar Publishing, 1996, pp. 9-9.

[8] A.H. Anwar and C. A. Kamhoua, "Cyber deception using honeypot allocation and diversity: A game theoretic approach," in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 2022: IEEE, pp. 543-549.

[9] N. C. Rowe and H. C. Goh, "Thwarting cyber-attack recognition with inconsistency and deception," in 2007 IEEE SMC Information Assurance and Security Workshop, 2007: IEEE, pp. 151-158.

[10] Mr. Capó, A. Pérez, and J. A. Lozano, "An efficient K-means clustering algorithm for massive data," arXiv preprint arXiv:1801.02949, 2018.

[11] H. Çeker, J. Zhuang, S. Upadhyaya, Q. D. La, and B.-H. Soong, "Deception-based game-based approach to mitigate-DoS-attacks," in GameSec: 7th International Conference, New York, NY, USA, November 2-4, 2016, Proceedings 7, 2016: Springer, pp. 18-38.

[12] J. Pawlick and Q. Zhu, Game theory for cyber deception. Springer, 2021.

[13] A.H. Anwar and C. Kamhoua, "Game theory on attack graph for cyber deception," in International conference on decision and game theory for security, 2020: Springer, pp. 445-456.

[14] H. Çeker, J. Zhuang, S. Upadhyaya, Q. D. La, and B.-H. Soong, "Deception-based game-based approach to mitigate-DoS-attacks," in GameSec: 7th International Conference, New York, NY, USA, November 2-4, 2016, Proceedings 7, 2016: Springer, pp. 18-38.

[15] G. F. Lyon, Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure, 2009.

[16] A.H. Anwar, C. Kamhoua, N. O. Leslie, and C. Kiekintveld, "Honeypot allocation for cyber deception under uncertainty," IEEE Transactions on Network and Service Management, Vol. 19, no. 3, pp. 3438-3452, 2022.

[17] D. ARMY, "Army support to military deception," FM, vol. 3, p. 4, 2019.

[18] A.H. Anwar, C. Kamhoua, and N. Leslie, "Honeypot allocation over attack graphs in cyber deception games," in 2020 International Conference on Computing, Networking and Communications (ICNC), 2020: IEEE, pp. 502-506.

[19] Mr. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," ACM Computing Surveys (CSUR), vol. 45, no. 3, pp. 1-39, 2013.

[20] R.K. Shrivastava, S. Ramakrishna, and C. Hota, , "Game theory based on modified naïve-bayes algorithm to detect DoS attacks using Honeypot," in 2019 IEEE 16th India Council International Conference (INDICON), 2019: IEEE, pp. 1-4.

[21] S. Kim, . "Game Theory Solutions for the Internet of Things: Emerging Research and Opportunities: Emerging Research and Opportunities," 2017.

[22] J. Antoniou, Game theory, the internet of things and 5G networks. Springer, 2020.

[23] Q. D. There, T. Q. Quek, J. Lee, S. Jin, and H. Zhu, , "Deceptive attack things," IEEE Internet of Things Journal, Vol. 3, no. 6, pp. 1025-1035, 2016.

[24] W. Tian, M. From, X. Ji, G. Liu, Y. Dai, and Z. Han, "Honeypot detection strategy against advanced persistent threats in industrial internet of things: A prospect theoretic game," IEEE Internet of Things Journal, Vol. 8, no. 24, pp. 17372-17381, 2021.

[25] A.H. Anwar, N. O. Leslie, and C. A. Kamhoua, "Honeypot allocation for cyber deception in the internet of battlefield things systems," in MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), 2021: IEEE, pp. 1005-1010.