Méthodes de détection d'attaques cybernétiques par une surveillance multicouches de communication

Andy Amoordon

UNIVERSITÉ DE LILLE

Université Gustave Eiffel Département
COSYS - Laboratoire LEOST

Thèse présentée par

Andy AMOORDON

pour obtenir le grade universitaire de Docteur

Discipline : Micro et NanoTechnologies, Acoustique et Télécommunications

---

Méthodes de détection d'attaques cybernétiques par une surveillance multicouches de communication

---

Sous la direction de :

Dr. (HDR) Virginie DENIAU, Université Gustave Eiffel VDA
Pr. Anthony FLEURY, IMT Nord Europe

Soutenance le 13/12/2022 devant le jury composé de :

Prof. Fabrice VALOIS	Examinateur, INSA Lyon (Président du jury)
Prof. Ali MANSOUR	Rapporteur, ENSTA Bretagne
Dr. (HDR) Mathieu CUNCHE	Rapporteur, Université de Lyon / Inria / CITI Lab
Dr. (HDR) Valeria LOSCRI	Examinateur, Inria Lille
Prof. Aurélien FRANCILLON	Examinateur, EURECOM Sophia Antipolis
Dr. Christophe GRANSART	Examinateur, Université Gustave Eiffel VDA

Résumé

Les réseaux sans-fil sont de plus en plus utilisés. La popularité de ces réseaux est due au fait que ces réseaux permettent de créer, modifier et étendre facilement un réseau informatique. Les réseaux sans-fil sont aussi particulièrement nécessaires pour relier des équipements mobiles tels que des montres connectées, voitures connectées et drones. Cependant, contrairement aux réseaux filaires où les transmissions sont isolées dans des câbles, les réseaux sans fil utilisent généralement des antennes omnidirectionnelles pour transmettre des signaux sur une zone de couverture déterminée par la puissance d'émission de l'antenne. Cette spécificité les rend plus vulnérables aux attaques. Il est ainsi plus facile d'écouter et d'émettre sans autorisation sur ces réseaux. Dans cette thèse, nous proposons une méthode pour détecter trois différentes attaques sur les réseaux sans-fil IEEE 802.11 (Wi-Fi). Les trois attaques sont l'attaque par faux point d'accès, l'attaque par déauthentication et l'attaque par brouillage. Dans la littérature scientifique, les méthodes existantes proposent de détecter ces attaques de manière isolée et en analysant uniquement un ou deux indicateurs.

Nous proposons une méthode utilisant des algorithmes de classification pour créer un modèle capable de détecter les trois attaques en analysant quatre indicateurs simultanément. Le modèle peut également détecter les attaques lorsqu'elles sont combinées. Concernant les données utilisées pour créer le modèle, nous avons exclusivement considéré les trames de gestion, et plus particulièrement les trames de type *beacon*, parmi les trois types de trames pouvant être émises sur un réseau Wi-Fi. Les trames de *beacon* sont régulièrement émises même en l'absence de trafic utilisateur. Nous montrons qu'une détection fondée sur les trames de type *beacon* est plus efficace. De plus, nous montrons que les variations du débit du réseau (trafic léger, moyen et intense) et de la puissance du signal de brouillage (forte, moyenne et faible) peuvent avoir un impact sur

la détection. Nous démontrons également qu'il est possible d'augmenter la précision de la détection de l'attaque par brouillage de forte intensité en prenant en compte les trames de *beacon* d'un deuxième point d'accès licite, mais éloigné du réseau. Enfin, nous présentons les résultats pour des cas particuliers tels que les transmissions Wi-Fi sur la bande 5 GHz et la détection des faux points d'accès fantômes.

Abstract

Wireless networks have nowadays become indispensable components of telecommunication infrastructures. They offer flexibility, mobility, and rapid expansion of telecommunication infrastructures. They are also particularly useful in connecting mobile devices such as connected cars, watches, and drones. However, unlike transmissions on wired networks, transmissions on wireless networks are not protected and isolated by wires, but rather emitted using omnidirectional antennas over an open spatial coverage. This makes wireless networks more vulnerable to certain types of attacks, such as unauthorized listening and emission. In this thesis, we have focused on detecting three different types of attacks on IEEE 802.11 (Wi-Fi) networks. These attacks include fake access points, jamming, and deauthentication attacks. In scientific literature, these three attacks are usually detected independently using one or two indicators.

We propose a method that uses classification algorithms to create a model that can detect the three different types of attacks by analyzing four indicators simultaneously. The model is also capable of detecting these attacks when they are combined. To create the model, we have considered only management frames of Wi-Fi networks, specifically beacon frames, as they are sent at regular intervals even in the absence of user traffic. Our findings show that basing the detection on beacon frames leads to more efficient results. We have also evaluated the impact of different variations in data rates (absence of user traffic, light, moderate, and intense user traffic) and jamming power (low, moderate, and high jamming power) on detection accuracy. Furthermore, we have found that the precision of detecting jamming attacks with high power can be improved by considering the beacon frames of a known second access point that is farther away. Finally, we present results for special cases, such as Wi-Fi transmissions in the 5 GHz band and the detection of beaconless fake access

points.

Remerciements

Mes remerciements les plus sincères vont à Dr. Mathieu Cunche, Prof. Ali Mansour, Dr. Valeria Loscri, Prof. Aurelien Francillon et Prof. Fabrice Valois pour avoir accepté être membres du jury, pour avoir relu le manuscrit de thèse et pour leurs commentaires pertinents.

À Monsieur Stéphane Ducasse et Madame Kim Loan Thai, pour leur aide dans l'obtention de ma thèse.

À Anthony Fleury, Christophe Gransart, Virgine Deniau pour m'avoir encadré pendant ces trois années et pour leurs commentaires pertinents sur mon travail.

À Monsieur le Professeur Antoine Latreille, pour m'avoir permis intervenir pendant deux ans, en tant que chargé d'enseignement, auprès de ses étudiants en L1 Droit Sciences et Innovation. Je le remercie particulièrement pour la liberté qu'il m'a laissé dans mes interventions.

À Monsieur William Gilles, Madame Irène Bouhadana, Monsieur Jean Harivel pour leur bienveillance et pour l'encadrement qu'ils m'ont fourni dans la réalisation d'une analyse juridique de ma thèse. Qu'ils trouvent ici l'expression de ma profonde reconnaissance.

Au directeur du laboratoire Laboratoire Électronique Ondes et Signaux Pour les Transports (LEOST), Monsieur Charles Tatkeu pour ses excellents conseils vers la fin de ma thèse. À Mesdames Lidwine Crampon et Corinne Davoust pour m'avoir aidé à corriger mes anomalies de badgeages lorsque j'avais des problèmes de TGV et pour toutes les petites demandes administratives qu'elles ont considérées et traitées avec célérité.

À mes collègues, Driss Aouladhadj, Artur Nogueira De Sao José, Olivier Stienne pour nos échanges scientifiques, leurs conseils et leurs explications théoriques, notamment sur la couche physique.

À Pierro Antonio (docteur en informatique et en physique) pour avoir réparé à plusieurs reprises la trottinette que j'utilisais pour venir au laboratoire. À Honoré Houpektodji pour m'avoir hébergé occasionnellement pendant ces trois années.

À Khouloud, Yue, Justin, Eva, Maria, Dana, Daven, Khesaven, Aunty Kalam, Tonton Retnon pour leur soutien moral.

À ma sœur, Mandeliya, pour avoir toujours été présente pour m'épauler et me guider. À Shawn, pour sa bonne humeur contagieuse et ses vanneries. À papa, pour son soutien inconditionnel, tant sur un plan émotionnel que financier.

Et enfin, à maman, merci pour tout ! Tu as tellement fait pour nous que quand tu n'es pas là, la vie paraît si fade... Et, comme disait si bien, Romain Gary : « Avec l'amour maternel, la vie vous fait, à l'aube, une promesse qu'elle ne tient jamais [...]. » Je ne te remercierai jamais assez. Je te dédicace toute ma thèse et tout mon parcours universitaire !

Liste des publications

— **Revue**

Amoordon, A., Deniau, V., Fleury, A., & Gransart, C. "A single supervised learning model to detect fake access points, frequency sweeping jamming and deauthentication attacks in IEEE 802.11 networks." Machine Learning with Applications 10 (2022) : 100389. **[Elsevier]**

— **Conférences internationales**

1. Amoordon, A., Deniau, V., Gransart, C., Fleury, A., & Villain, J. "A Threshold-Based Detection Approach To Detect Fake Access Points and Jamming Attacks on IEEE 802.11 Networks : Implementation, Results and Limitations." 2022 3rd URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC). IEEE, 2022.

2. Amoordon, A., Gransart, C., Deniau, V. "Characterizing Wi-Fi Man-In-the-Middle Attacks." 2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science. IEEE, 2020.

3. Amoordon, A., Gransart, C., Deniau, V., & Gesnot, C. "Finding Indicators To Detect Fake Access Point Attacks On the Physical Layer. " In URSI Benelux Forum (2019, December) (p. 1p). **[Abstract]**

— **Conférence nationale**

Amoordon, A. Gransart, C., Deniau, V., Fleury, A. "La détection par seuil des attaques Man-in-the-Middle et de brouillage sur les réseaux Wi-Fi." Neuvième Conférence Plénière du GDR ONDES les 30 novembre et 1er décembre 2021. **[Abstract]**

Table des matières

Liste d'abréviations

ACK Trame d'acquittement (*Acknowledgement frame*)

ARPANET Advanced Research Projects Agency Network

CCA-ED Clear Channel Assessment-Energy Detect

CS/CCA Carrier Sense/Clear Channel Assessment

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance

CSMA/CD Carrier Sense Multiple Access/Collision Detection

CART Classification and Regression Trees

CTS Clear To Send

DCF Distributed coordination function

DIFS Distributed Inter Frame Space

DoS Déni de service (Denial of Service)

DPO Délégué à la Protection des Données (Data Protection Officier)

DSSS Direct Sequence Spread Spectrum

FCS Frame Check Sequence

FFT Fast Fourrier Transform

FHSS Frequency Hopping Spread Spectrum

GPS Global Positioning System

IA Intelligence Artificielle

IDS Intrusion Detection System (SDI en anglais)

IF Infrarouge

IP Internet Protocol

INSEE Institut National de la Statistique et des Études Économiques

JFaible brouillage de puissance faible

JFort brouillage de puissance forte

JMoyen brouillage de puissance moyenne

KNN K-Nearest Neighbor

LDA Linear Discriminant Analysis

LLC Logical Link Control

LR Logistic Regression

MAC Medium Access Control

NB Naives Bayes

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency Division Multiple Access

OSI Open System Interconnexion

PCF Point coordination function

PDU Protocol Data Unit
PLCP Physical Layer Convergence Protocol
PMD Physical Medium Dependant
RADIUS Remote Authen- tication Dial-In User Service
RF Radio Fréquence
RGPD Règlement Général de Protection des Données
RTS Request To Send
RSSI Received Signal Strength Indicator
SDI Système de Détection Intrusion
SDR Software Defined Radio
SDU Service Data Unit
SSH Secure Shell
SSID Service Set Identifier
SVM Support Vector Machine
TCP Transmission Control Protocol
TP Taux de précision
UDP User Datagram Protocol
usrp USRPUniversal Software Radio Peripheral
VPN Virtual Private Network
WDS Wireless Distribution Sysyem
WPA Wireless Protected Access

Table des figures

Liste des tableaux

Chapitre 1

Introduction

« Il n'existe point de système parfait.»

C'est avec cette phrase célèbre que nous entamons la rédaction de ce manuscrit de thèse. "Il n'existe point de système parfait", fait référence au fait que rien n'est parfait. La perfection est considérée comme une caractéristique divine(Jambet, 2018), tandis que l'humain et ses créations restent imparfaits. Par conséquent, il est important de toujours penser à des mécanismes de protection pour sécuriser les créations humaines.

Les mécanismes de protection visent à protéger l'homme et ses créations contre les actes non intentionnels (dysfonctionnement, accidents, incidents...) et intentionnels (vol, vices, piratage...). Ces mécanismes peuvent être de différents types : juridiques, moraux, religieux et techniques. Par exemple, pour protéger un vélo contre le vol, on peut édicter une loi répressive interdisant la soustraction d'un bien d'autrui, condamner l'acte sur un plan moral, l'interdire sur un plan religieux ou utiliser des moyens techniques pour l'empêcher ou le retarder. Au Japon, l'opprobre social associé aux actes de vol est tel que ces derniers sont très rares et il est courant pour les Japonais de laisser leur vélo sans protection. Dans d'autres pays, des moyens techniques tels qu'un cadenas ou un cadre en U sont nécessaires pour protéger, retarder ou dissuader un vol. De même, la protection religieuse a été utilisée en Inde pour protéger les vaches sacrées contre l'abattage. Pendant les périodes de famine, les agriculteurs pour-

raient être tentés de les tuer pour se nourrir, ainsi la vache a été élevée au rang de sacré pour éviter cette pratique (Harris, 1978) . Il est cependant important de souligner que la protection en soi n'est pas parfaite et ne peut pas être efficace à 100%. Il est donc nécessaire d'améliorer continuellement ces mécanismes de protection.

L'informatique est une invention humaine qui résulte d'une succession d'inventions au fil des siècles. De la machine à cartes perforées de Joseph Marie Jacquard (Karwatka, 1999) à la machine à calculer à vapeur de Charles Babbage (Hyman, 1985), puis à la machine d'Alan Turing (TURING, 1950), l'informatique a connu une évolution constante. Après la Seconde Guerre mondiale, le développement de l'électronique, de l'informatique et de la technologie nucléaire ont conduit à une troisième révolution industrielle. Cette période a été marquée par la création des premiers ordinateurs et la mise en place d'ARPANET à la fin des années 1960. L'informatique s'est ensuite démocratisée dans les années 1980, tandis que la création et le développement d'Internet ont eu lieu un peu plus tard, dans les années 1990 et 2000 (Gilles et al., 2019). Cette troisième ère de notre société a entraîné l'informatisation et l'automatisation de tâches qui étaient autrefois effectuées manuellement. Elle a également vu le développement d'outils informatiques qui permettent de renforcer l'interactivité des échanges et de croiser des données (Gilles et al., 2019).

Le développement d'outils informatiques ne s'est toutefois pas fait sans crainte. Dans les années 1970, les ordinateurs étaient principalement détenus par les États et certains de ces États ont utilisé ces outils pour automatiser le traitement des fichiers de leurs administrés, créant ainsi une crainte de la part de ces derniers. Dès 1974, la population a commencé à protester contre ces pratiques étatiques (Gilles et al., 2019). En France, par exemple, à la suite de l'annonce du projet d'un Système automatisé pour les fichiers administratifs et répertoires des individus (SAFARI) (Faligot, 2007), plusieurs contestations populaires ont eu lieu. Le projet SAFARI proposait, entre autres, d'utiliser le numéro INSEE pour interconnecter ou croiser les fichiers nominatifs de l'admi-

nistration française. Les contestations ont été entendues et ce projet a finalement été retiré (Gilles et al., 2019). Cette crainte de la population a incité les dirigeants à fournir des garanties, sous forme de protections juridiques, contre le traitement des données des citoyens. La loi "Informatique et Libertés" du 6 janvier 1978, créant la Commission Nationale de l'Informatique et des Libertés (CNIL), ainsi que la loi du 17 juillet 1978, créant la Commission d'Accès aux Documents Administratifs (CADA), ont été promulguées. En 1983, la France a également ratifié la Convention pour la protection des personnes à l'égard du traitement automatisé des données à caractère personnel (dite Convention 108) (l'Europe, 1981). Plus tard, après une directive européenne 95/46/CE sur la protection des données, le Règlement Général de la Protection des Données Personnelles (RGPD) uniformisera cette protection au sein des États membres de l'Union Européenne (Gilles et al., 2019).

Internet ou l'interconnexion entre les ordinateurs permet de transférer des données entre ordinateurs à travers le monde entier. Ce faisant, Internet a permis de créer des autoroutes pour la circulation des données. Si, pendant longtemps, les équipements connectés à Internet étaient essentiellement des ordinateurs fixes utilisant des connexions câblées, depuis quelques années, nous assistons à une démultiplication des connexions sans-fils et le début d'un nouveau cycle caractérisé par la société de la donnée (FALQUE-PIERROTIN, 2014). Selon Madame Isabelle Falque-Pierrotin : « L'individu est de plus en plus pris dans un maillage extrêmement fin d'informations personnelles relayées par des objets de plus en plus communicants : téléphone portable, bracelets électroniques, dispositifs électriques, équipements de vidéosurveillance, etc... » (FALQUE-PIERROTIN, 2014).

S'opère ainsi un changement de paradigme, les données collectées sont de plus en plus précises. Si après l'activité des États, l'activité des entreprises (devenues les nouveaux pionniers de l'invention informatique) a pu être réglementée par le RGPD, il reste toutefois une petite part d'ombre que la protection juridique peine à réglementer. C'est le cas notamment des activités

d'attaquants ou de pirates informatiques qui opèrent individuellement et parfois en petits groupes. Ces personnes sont malveillantes, peu scrupuleuses et essaient, par des moyens techniques, de voler les données des utilisateurs pour les revendre ou leur causer du tort en se faisant passer pour eux. La protection juridique ne semble pas être suffisante pour protéger les utilisateurs de ces attaquants. En France, par exemple, s'il existe des dispositions répressives telles que l'article 323-1 du Code pénal français [1] ou la jurisprudence sur le vol de données [2], elles nécessitent l'établissement d'une preuve qui n'est pas aisée en cybersécurité. En effet, l'adresse Internet Protocol (IP) ne permet pas d'identifier précisément une personne, mais plutôt un ordinateur ou un équipement connecté à Internet (Grégoire, 2009). Afin d'identifier l'attaquant ou la personne responsable d'une cyberattaque, une enquête approfondie menée par l'autorité compétente est nécessaire. Dans les faits, seules les infractions les plus graves mènent à des enquêtes et donnent lieu à des sanctions. Par conséquent, pour assurer une protection efficace des réseaux et des outils informatiques contre les cyberattaques, il est nécessaire d'adopter des mesures de protection techniques.

Techniquement, le passage vers une société de la donnée implique un changement important : une grande partie de la communication s'effectue désormais par le biais de réseaux sans fil. Ces derniers sont devenus incontournables, car ils permettent d'étendre rapidement des réseaux informatiques et de connecter des équipements qui ne disposent pas d'une interface filaire. Les ordinateurs portables, les téléphones mobiles, les drones, les voitures connectées, entre autres, les utilisent pour communiquer entre eux et pour se connecter à In-

1. Article 323-1 du Code pénal, Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 60 000 € d'amende.

Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de trois ans d'emprisonnement et de 100 000 € d'amende.

Lorsque les infractions prévues aux deux premiers alinéas ont été commises contre un système de traitement automatisé de données à caractère personnel mis en œuvre par l'État, la peine est portée à cinq ans d'emprisonnement et à 150 000 € d'amende.

2. Voir jurisprudence sur le vol d'informations : notamment Cour de cassation chambre criminelle, 28 juin 2017, n°16-81.113.

ternet (Nicopolitidis et al., 2003). Les réseaux sans fil sont également utilisés dans le secteur de la sécurité et du transport, notamment pour échanger avec les centres de contrôle ou de surveillance (Ye et al., 2013). Toutefois, comparativement aux réseaux filaires, les réseaux sans fil sont plus vulnérables à certaines formes d'attaques. En effet, dans les réseaux sans fil, les transmissions ne sont pas isolées et protégées par des câbles comme dans un réseau filaire. Les transmissions sont émises par des antennes, généralement de manière omnidirectionnelle sur une couverture déterminée par la puissance d'émission de l'antenne. Pour ces raisons, il est plus facile d'écouter et d'émettre sans autorisation sur les réseaux sans fil.

La télématique, expression inventée par Nora-Minc (Walliser, 1989 ; Gilles et al., 2019), désigne la fusion de l'informatique et des télécommunications. En télématique, le modèle TCP/IP (très inspiré du datagramme de Louis Pouzin (Pouzin, 1976)) est utilisé. Le modèle TCP/IP ainsi que le modèle OSI, plus généralement, proposent une procédure d'encapsulation avant chaque transmission, qu'elle soit filaire ou sans fil. Cette procédure est grandement inspirée du système d'envoi de courrier. Dans ce système, lorsqu'une personne souhaite envoyer une lettre, cette dernière doit être mise dans une enveloppe indiquant les informations du destinataire et de l'expéditeur. Ces informations sont nécessaires pour assurer la bonne remise de la lettre au destinataire. De la même manière, le modèle OSI (Open System Interconnection) propose d'ajouter des informations à la donnée, que souhaite envoyer l'utilisateur, pendant la procédure d'encapsulation. Cette procédure se déroule en sept couches (ou étapes), au cours desquelles des entêtes contenant diverses informations sont ajoutés à la donnée à envoyer. Sur la couche 1, après le dernier entête, la donnée à envoyer et l'ensemble des entêtes (appelé trame), est convertie en suite binaire et modulée en fonction du support de transmission (filaire ou sans-fil) (Atelin, 2009). Ces informations d'entête (comme les informations sur une enveloppe) permettent la commutation et le routage de la trame dans un réseau informatique.

Dans un réseau sans-fil, la trame est convertie en suite binaire et modulée sur un canal de fréquence avant d'être émise de manière omnidirectionnelle sur une zone de couverture déterminée par la puissance d'émission de l'antenne. Tous les équipements se trouvant dans la zone de couverture réceptionnent la trame et la décodent en partant de la couche 2 à la couche 7. En s'appuyant sur les informations de la couche 2, les équipements vont déterminer s'ils sont destinataires de la transmission. Le cas échéant, l'équipement continuera le décodage. Dans le cas contraire, l'équipement devra supprimer la transmission. L'attaque par écoute consiste à décoder et à enregistrer toutes les trames émises sur un canal (même celles qui ne sont pas destinées à un équipement) (Zou et al., 2016). Pour prévenir l'attaque par écoute, l'envoyeur et le destinataire, avant toute transmission de donnée, peuvent effectuer un échange de clés de chiffrement pour sécuriser leur communication. De cette manière, les attaquants peuvent toujours écouter les trames, mais en fonction du type de chiffrement utilisé, tout ou partie des informations de la trame sera incompréhensible (Spitalnick, 2009).

Le chiffrement n'est toutefois pas la panacée et peut être contourné en effectuant, par exemple, des attaques logicielles telles que les attaques par force brute ou par dictionnaire, ou des attaques matérielles telles que l'extraction physique de la clé de chiffrement (Heckmann, 2018). L'attaque par force brute consiste à essayer toutes les possibilités jusqu'à retrouver la clef de chiffrement. L'attaque par dictionnaire consiste à essayer un petit ensemble de mots qui est réputé être le plus utilisé en espérant que le mot de passe fasse partie de cet ensemble (Sullivan, 2007). L'attaque par dictionnaire n'est pas toujours fructueuse et l'attaque par force brute, lorsqu'elle est effectuée avec des ordinateurs traditionnels (non quantiques), peut prendre beaucoup de temps et consommer beaucoup d'énergie (Vugdelija et al., s. d.). L'attaque par extraction physique de clef de chiffrement, elle, est relativement couteuse et nécessite l'utilisation d'équipements spécifiques (autre qu'un simple ordinateur) et un accès physique sur terminal ciblé (Heckmann, 2018).

Pour ces raisons, les attaquants préfèrent souvent adopter d'autres formes d'attaques plus simples à mettre en place. L'attaque *Man-in-the-Middle*, également connue sous le nom d'attaque par l'homme au milieu, est l'une d'entre elles (Mallik, 2019). Cette technique consiste à s'intercaler entre deux parties qui communiquent pour se faire passer pour l'émetteur auprès du destinataire et pour le destinataire auprès de l'émetteur. L'attaque *Man-in-the-Middle* permet à l'attaquant de contrôler la communication et de lire, modifier ou fabriquer des messages. Il existe plusieurs formes de cette attaque (Conti et al., 2016), notamment l'attaque « Man-in-the-Middle » dans un réseau sans-fil Wi-Fi (IEEE 802.11). Dans ce cas, l'attaquant tente de remplacer un point d'accès licite (Lovinger et al., 2020) en créant un faux point d'accès qui imite les informations envoyées par le point d'accès réel. L'attaquant peut alors utiliser des attaques de déni de service telles que les attaques par brouillage (*jamming* en anglais) (Poisel, 2011) ou par déauthentification (*deauthentication* en anglais) (Arora, 2018) pour déconnecter les clients du point d'accès licite et les forcer ou les inciter à se connecter à son faux point d'accès.

L'attaque par brouillage a pour objectif de diminuer le rapport signal à interférence d'un canal de communication afin de dégrader ou d'interrompre la communication (Poisel, 2011). Il existe plusieurs types de brouilleurs, mais le plus facilement accessible à l'achat est le brouilleur à balayage de fréquence. L'attaque par déauthentification consiste à exploiter un manquement de sécurité du protocole IEEE 802.11, à savoir l'absence de chiffrement des trames de contrôle. L'attaquant envoie des trames usurpées pour déconnecter les clients du point d'accès licite (Arora, 2018). Souvent, l'attaquant utilise ces deux techniques pour simuler un déni de service du point d'accès licite et ainsi contraindre ou inciter les clients à se connecter à un faux point d'accès.

Ces attaques sont dangereuses et permettent aux attaquants de voler des données sensibles, notamment concernant les sites web visités (date et heure de connexion, fréquence d'utilisation, etc.). Elles sont souvent une première étape pour d'autres attaques permettant le vol de données personnelles (telles que

les mots de passe, numéros de cartes de crédit, etc.) (Cekerevac et al., 2017). Ces trois attaques peuvent aujourd'hui être réalisées à bas coût en utilisant des logiciels en libre accès et avec du matériel peu coûteux. L'attaque *Man-in-the-Middle* peut être réalisée en utilisant la carte Wi-Fi d'un ordinateur et un logiciel libre tel que Wi-Fi Pumpkin (Team, 2018) ou Airmon-ng (D'Otreppe de Bouvette, 2016) (tous deux disponibles sous Linux). Pour l'attaque par déauthentification, il est possible d'utiliser la carte Wi-Fi d'un ordinateur et les mêmes logiciels libres que pour l'attaque *Man-in-the-Middle*. Enfin, l'attaque par brouillage peut être réalisée à l'aide d'un brouilleur qui coûte environ une centaine d'euros ou en utilisant des radios logicielles telles que le RTL-SDR (Nardi, 2019), qui coûte moins de 30 euros.

Dans cette thèse, nous avons cherché des méthodes pour détecter ces trois attaques sur des réseaux Wi-Fi. Nous avons adopté une méthode de détection dite de détection par anomalie, qui consiste à comparer des situations d'attaque avec une situation de référence sans attaque. Cette méthode est également passive, c'est-à-dire que la détection se fait en analysant les informations émises par les équipements du réseau sans rien émettre. La méthode est fondée sur les informations des deux premières couches du modèle OSI, de telle sorte que les informations personnelles et les identifiants tels que l'adresse IP des utilisateurs (se trouvant sur les couches supérieures 3 à 7) ne sont pas analysés. Enfin, le système de détection d'intrusion peut détecter les trois attaques lorsqu'elles sont réalisées séparément, mais également lorsqu'elles sont combinées ou cumulées.

La structure de ce manuscrit de thèse est la suivante :

Le **deuxième chapitre** de cette thèse commence par un bref rappel sur le fonctionnement des ordinateurs et les différents types de connexions (logiques et physiques) ainsi que les liens (sans-fil et filaires) pouvant être utilisés pour former un réseau. Nous poursuivons en détaillant le protocole IEEE 802.11 (Wi-Fi), en mettant en évidence sa couche physique et liaison de données, avant

d'aborder les différentes menaces qui pèsent sur ces réseaux. Nous présentons ensuite en détail les trois attaques considérées dans cette thèse (à savoir l'attaque par faux point d'accès, l'attaque par déauthentification et l'attaque par brouillage), ainsi que les solutions existantes dans la littérature scientifique et l'état de la technique pour détecter ces attaques. Enfin, nous concluons ce chapitre en faisant le bilan et en précisant la valeur ajoutée de notre recherche par rapport à l'existant.

Dans le **troisième chapitre** de cette thèse, nous décrivons les bancs de test que nous avons utilisés, ainsi que les différentes variations des réglages de ces bancs (telles que le trafic entre le client et le serveur, ainsi que la puissance du brouilleur) et les équipements que nous avons utilisés pour mettre en œuvre les attaques et pour capturer les données.

Dans le **quatrième chapitre** de cette thèse, nous procédons à l'analyse des données capturées pour identifier les attributs qui nous permettent de caractériser la présence d'attaques. Nous utilisons ensuite ces indicateurs pour développer un système de détection d'intrusion (SDI). La détection des attaques peut être réalisée en utilisant une méthode de prise de décision par seuil ou par algorithme de classification. Dans notre première version du SDI, nous avons opté pour la méthode de prise de décision par seuil. Cependant, cette méthode comporte des limitations qui nous ont conduits à utiliser des algorithmes de classification pour obtenir de meilleurs résultats. Nous présentons en détail la procédure de préparation des données et d'entraînement des modèles, ainsi que les résultats obtenus grâce à l'utilisation des algorithmes de classification.

Dans le **cinquième chapitre** de cette thèse, nous examinons les différentes pistes d'amélioration de la détection. Nous prenons en compte les variations des réglages des bancs de test, telles que le débit et la puissance, pour déterminer si ces variations ont un impact sur la précision de la détection. Nous abordons également des cas particuliers, tels que les transmissions sur la bande 5 GHz et les formes avancées d'attaques telles que le cumul des attaques.

Finalement, dans le **sixième et dernier chapitre** de cette thèse, nous résumons les travaux présentés dans ce manuscrit avant de mettre en évidence les problématiques que nous avons résolues et celles qui pourraient être explorées dans le cadre de futurs travaux.

Chapitre 2

Notions informatiques, la norme IEEE 802.11, description et état de l'art des attaques considérées

Dans ce chapitre, nous rappelons brièvement les notions de base liées aux ordinateurs et à leur mise en réseau, notamment les types de liens et de connexions qui peuvent être utilisés. Nous soulignons également que dans un réseau informatique, un attaquant peut avoir plusieurs points d'entrée et que, dans cette thèse, nous nous intéressons aux attaques sur la partie sans fil (Wi-Fi) du réseau. Après un rappel détaillé de la norme IEEE 802.11 (Wi-Fi), son architecture, ses différentes versions, sa couche physique et sa couche de liaison de données, nous nous intéressons aux différentes menaces qui pèsent sur ces réseaux et les trois attaques que nous cherchons à détecter dans cette thèse.

Les trois attaques en question sont l'attaque par faux point d'accès, l'attaque par déauthentification et l'attaque par brouillage. Dans ce chapitre, nous examinons les différentes formes de ces attaques et expliquons la forme d'attaque que nous considérons. Pour chaque attaque, nous fournissons une liste des outils matériels et logiciels qui peuvent être utilisés pour les réaliser. Nous détaillons ensuite, pour chaque attaque, les méthodes de détection proposées dans la littérature scientifique, ainsi que les amendements protocolaires et les solutions industrielles. À la fin du chapitre, nous présentons un bilan et ex-

pliquons nos motivations ainsi que les objectifs que nous souhaitons atteindre dans cette thèse.

2.1 Les ordinateurs

Le dictionnaire Larousse définit l'ordinateur comme étant une « machine automatique de traitement de l'information, obéissant à des programmes formés par des suites d'opérations arithmétiques et logiques ». Comme indiqué sur la Figure 2.1, l'architecture d'un ordinateur est composée d'une unité centrale de calcul, de périphériques d'entrée et de sortie, ainsi que de mémoire de stockage (primaire et secondaire). L'unité centrale de calcul est responsable de l'exécution des instructions (Laue, 2004). Les instructions peuvent être entrées directement par l'utilisateur en utilisant des périphériques d'entrée (tels que le clavier, l'écran tactile), être reçues depuis Internet, ou être copiées de la mémoire secondaire à la mémoire primaire, puis communiquées à l'unité centrale de calcul. L'exécution des instructions retourne un résultat qui peut être affiché par des périphériques de sortie tels qu'un écran ou une imprimante, ou stocké dans la mémoire primaire de manière temporaire et dans la mémoire secondaire de manière permanente (Rajaraman et al., 2014).

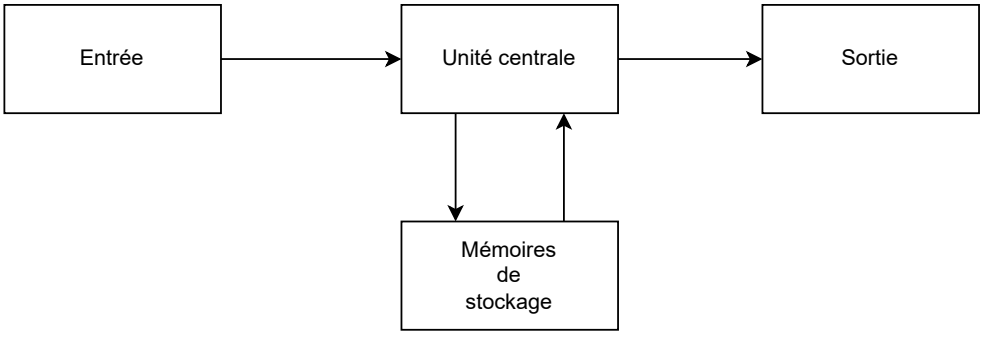

Figure 2.1 – Architecture d'un ordinateur

Source: traitement auteur : reproduction de l'architecture des ordinateurs décrite par Von Neumann en 1945 dans (Neumann, 1945)

Il existe plusieurs types d'ordinateurs : les superordinateurs, les mini-ordinateurs et les micro-ordinateurs. Chacun de ces types d'ordinateurs a également des sous-types qui diffèrent en termes de configuration (Rajaraman et al., 2014). Par exemple, un superordinateur aura une unité centrale (voire un réseau de plusieurs unités centrales) beaucoup plus puissante que celle d'un ordinateur personnel. Un serveur de données est un ordinateur conçu pour stocker une grande quantité d'informations sur le long terme, et disposera par conséquent d'une mémoire de stockage secondaire importante en comparaison avec d'autres types d'ordinateurs. De même, les objets connectés sont de très petits ordinateurs qui ont souvent pour fonction la détection d'une valeur physique (en utilisant des capteurs) et la création d'un phénomène physique (en utilisant des actionneurs). Ces objets ont une autonomie limitée qui implique une configuration plus légère et souvent l'utilisation de protocoles de réseaux sans-fil spécifiques à faible consommation d'énergie, tels que Zigbee ou LoRa (Soumyalatha, 2016 ; Reddy Maddikunta et al., 2020).

Si jadis les ordinateurs fonctionnaient principalement de manière isolée avec des instructions et des données qui étaient fournies soit par l'utilisateur, soit copiées depuis les mémoires telles que des clés USB, des CD-ROM ou des DVD-ROM, aujourd'hui les ordinateurs sont de plus en plus connectés et s'échangent directement des instructions et des données par l'intermédiaire de réseaux informatiques. Cette interconnexion a ainsi favorisé la démultiplication d'attaques (Nojeim, 2010 ; Nastasiu, 2016). Outre les attaques par clé USB empoisonnée ou par virus informatique, un attaquant peut maintenant empoisonner les informations ou les instructions lorsqu'elles circulent sur Internet. Par exemple, lorsqu'un utilisateur consulte une page web, son ordinateur demande en réalité une copie de la page web au serveur d'hébergement. La copie est transmise par Internet et sera interprétée et exécutée en local par l'ordinateur de l'utilisateur. Un attaquant, s'il contrôle un des intermédiaires d'Internet, peut empoisonner la page web et injecter des instructions illégitimes, telles qu'un logiciel espion qui sera exécuté et installé sur l'ordinateur de l'utilisateur avec la page web.

Ce logiciel permettra par la suite à l'attaquant de contrôler ou de perturber le fonctionnement de l'ordinateur (Clarke, 2009 ; Gilad et al., 2014).

Pour relier différents ordinateurs, différents types de liens et connexions peuvent être utilisés. La transmission de données sur ces liens s'effectue, majoritairement, en respectant le modèle OSI (Atelin, 2009). Ce modèle permet d'échanger des données entre ordinateurs en faisant abstraction de leur configuration et de leur fonctionnement logiciel interne tels que leur système d'exploitation ou leur architecture processeur.

2.2 La mise en réseau des ordinateurs

La mise en réseau des ordinateurs peut être effectuée à l'aide de connexions filaires ou sans fil, permettant de créer des connexions physiques ou logiques pour la transmission de données, en respectant le modèle OSI dans la plupart des cas.

2.2.1 Liens filaires vs. liens sans-fil

Les liens d'un réseau peuvent être filaires ou sans fil. Dans le cas des liens filaires, différents matériaux peuvent être utilisés pour leur création, tels que des paires de fils de cuivre torsadées, des câbles coaxiaux ou encore des fibres optiques (voir Figure 2.2) (IsaacComputerScience, s. d.). En revanche, pour les liens sans fil, les transmissions se font à travers des ondes infrarouges, acoustiques, radios ou micro, émises dans l'air, l'eau ou dans le vide. La transmission sur un fil, elle, est isolée et dirigée, ce qui permet une transmission dans un environnement contrôlé avec moins d'interférences et de collisions, car la ressource n'est pas partagée et est en libre accès, comme dans le cas des liens sans fil. Cette isolation réduit également les risques d'attaques par écoute

(a) Fibre optique

(b) Câble Ethernet, paires torsadées

(c) câble coaxial

FIGURE 2.2 – Supports de transmission filaire

Source: Fig 2.2 (a) : Chaitawat Pawapoowadon, tirée de Pixabay.com, licence Pixabay. Fig 2.2 (b) : Adrian Malec, tirée de Pixabay.com, licence Pixabay. Fig 2.2 (c) : Antonio Longueira, tirée de Pixabay.com, licence Pixabay

(IsaacComputerScience, s. d.). Pour les liens sans fil, la transmission se fait généralement via des antennes omnidirectionnelles ayant une couverture déterminée par la puissance d'émission de l'antenne (Rackley, 2011). Les transmissions se font également sur un canal dans une bande de fréquence, comme c'est le cas pour le réseau sans fil *Global Positioning System* (GPS), où les satellites émettent en permanence sur deux fréquences : 1 575,42 MHz et 1 227,60 MHz. Autre exemple, dans les réseaux sans-fil Wi-Fi, les équipements peuvent, à partir de la deuxième génération de la norme IEEE 802.11, émettre sur deux bandes de fréquences : 2,4 GHz et 5 GHz. Comme les transmissions se font de manière omnidirectionnelle et sans isolation, les liens sans fil sont plus vulnérables aux attaques, notamment à l'attaque par écoute. Sur un plan énergétique, selon (Baliga et al., 2011), les transmissions sur les liens sans-fil consommeraient 10 fois plus d'énergie que celles sur les liens filaires.

Les liens sans-fil ont toutefois l'avantage de permettre d'étendre facilement un réseau et de connecter des équipements qui ne peuvent pas être connectés en filaire. C'est le cas, par exemple, des voitures connectées, du fait de leur mobilité. Les liens sans-fil sont également utiles pour créer des réseaux informatiques dans des environnements protégés ou difficiles d'accès tels que les

forêts, les monuments culturels, les îlots et dans des environnements hostiles comme les opérations militaires à l'étranger ou dans les déserts (AGOSTA et al., 2015).

2.2.2 Connexions point-à-point vs. connexions logiques

Les liens filaires ou sans-fil permettent de créer des connexions. Les connexions peuvent être point à point ou logiques (PETERSON et al., 2007). Une connexion point à point est une connexion directe entre l'envoyeur et le destinataire du message. L'envoyeur est le terminal qui est originaire du message alors que le destinataire est le terminal auquel le message est destiné. En d'autres termes, dans une connexion point à point, il n'y a pas d'intermédiaire (voir Figure 2.3) entre l'envoyeur et le destinataire. L'envoyeur est directement connecté au destinataire. Dans une connexion logique, l'envoyeur et le destinataire ne sont pas directement connectés entre eux, mais il existe un chemin entre les deux terminaux (PUJOLLE, 2014). Le chemin est composé d'un ou plusieurs intermédiaires par lesquels transite le message avant d'arriver au destinataire (voir Figure 2.4). Les connexions point à point sont plus sécurisées du fait de l'absence d'intermédiaires entre l'envoyeur et le destinataire. Dans une connexion logique, un attaquant peut attaquer l'envoyeur et le destinataire, mais également un des intermédiaires par lequel transite le message. L'attaquant a, par conséquent, plusieurs points d'entrée pour attaquer la communication.

FIGURE 2.3 – Connexions physiques/Point à Point

Source: traitement auteur : sauf icônes sous licence open source diagrams.net

FIGURE 2.4 – Connexions logiques

Source: traitement auteur : icônes sous licence open source diagrams.net

Les connexions point à point sont généralement considérées comme plus sûres, mais leur mise en place peut être coûteuse. Par exemple, pour connecter un ordinateur en France à un ordinateur aux États-Unis en utilisant une connexion point à point, il serait nécessaire soit d'utiliser une liaison sans fil par radiofréquence (par exemple, un lien satellite) entre les deux équipements, soit de déployer un câble partiellement terrestre et partiellement sous-marin entre les deux ordinateurs (FREEMAN, 2005 ; PUJOLLE, 2014). Maintenant, si l'on souhaite connecter plusieurs ordinateurs, il faudra répéter ce processus pour chaque connexion. Pour ces raisons, les connexions point à point sont principalement utilisées pour des communications sur de courtes distances, par exemple pour envoyer une photo d'un téléphone à un autre. Pour construire de grands réseaux informatiques internationaux, les connexions logiques sont souvent privilégiées.

Internet, par exemple, est un réseau informatique qui permet d'interconnecter plusieurs ordinateurs de manière logique. Internet a été construit de la manière suivante : dans chaque pays, chaque opérateur connecte les maisons des particuliers et les entreprises entre elles pour créer un réseau de ville (GLOWNIAK, 1998). Les réseaux de ville sont ensuite connectés entre eux pour créer des réseaux départementaux, les réseaux départementaux sont connectés entre eux pour créer des réseaux régionaux, et les réseaux régionaux sont connectés entre eux pour créer des réseaux nationaux. Enfin, les réseaux nationaux sont connectés entre eux par des câbles sous-marins et des liens satellites, grâce à des opérateurs dits de *backbone* (GRALLA, 1998 ; COMER, 2018), pour

créer Internet (RYAN, 2010).

Les connexions point à point ont toutefois un avantage considérable : elles permettent plus facilement de créer des protocoles de communication propriétaires. En effet, si une entreprise souhaite que ses équipements informatiques puissent échanger entre eux avec un protocole propriétaire, elle pourra plus facilement le faire pour des communications point à point. Dans les communications point à point, il n'y a pas d'intermédiaire, et si l'envoyeur et le destinataire sont du même constructeur, l'entreprise a un contrôle sur la chaîne de communication. Par exemple, l'entreprise Apple a développé le logiciel AirDrop (ADITYA et al., 2014), qui permet aux produits de la marque Apple (iPhone, iPods et MacBook) d'échanger des données entre eux sur une courte distance. En revanche, la création de protocoles propriétaires pour router les paquets sur un réseau logique géant comme Internet est plus difficile. Il faudrait en réalité que l'envoyeur, le destinataire et l'ensemble des intermédiaires soient du même constructeur. Ces réseaux sont assez coûteux à créer et à maintenir. Il existe toutefois certains réseaux informatiques internationaux de ce type, souvent appelés Internet privé, qui sont gérés par de grandes entreprises de télécommunication. Elon Musk, par exemple, a récemment créé un réseau Internet par Satellite (Starlink) (HARRIS, 2018) qui lui permettra d'adopter des protocoles propriétaires sur toute la chaîne de transmission, de l'envoyeur au destinataire.

Les réseaux publics suivent généralement le modèle OSI. Ce modèle impose que les transmissions, que ce soit sur les connexions logiques ou point à point, soumises à une procédure d'encapsulation.

2.2.3 Le modèle OSI : la procédure encapsulation

Le modèle OSI est un modèle divisé en sept couches qui indique la procédure à suivre pour une bonne transmission des données dans un réseau informatique.

Ce modèle permet de transmettre des données en faisant abstraction du matériel, du système d'exploitation et des logiciels de l'envoyeur, du destinataire et des intermédiaires. Il permet ainsi, par exemple, à des ordinateurs de différentes marques comme Apple, Dell et Sony, qu'ils utilisent OS X, Windows ou Linux, de communiquer sans changer leur fonctionnement interne. Les sept couches, de haut en bas, sont la couche application, présentation, session, transport, réseau, liaison de données et la couche physique (voir Figure 2.5). Toutes les couches ne sont pas obligatoires et certaines couches peuvent être combinées. Cette division en couches permet de garantir une certaine indépendance entre les protocoles de chaque couche (KUMAR et al., 2014). Par conséquent, un informaticien travaillant sur la couche 3 n'a pas besoin de maîtriser toutes les informations présentes sur les couches supérieures ou inférieures à la couche 3. Seules les informations liant les couches, lorsque cela se présente (par exemple, un *flag* sur la couche 2 qui indique s'il y a une retransmission sur une couche supérieure), doivent être connues.

FIGURE 2.5 – Modèle Open Système Interconnexion - sept couches

Source: traitement auteur : reproduction du modèle Open Système Interconnexion pour lequel Hubert Zimmermann a joué un rôle clé entre 1978 et 1984 devenu la norme ISO/IEC 7498-1 :1994 (ISO/IECJTC1, 1994)

Ce modèle impose l'ajout d'entêtes à la donnée à envoyer à chaque couche, de la couche 7 à la couche 1 (KUMAR et al., 2014). Sur la couche physique, après l'ajout d'un dernier entête, l'ensemble contenant les entêtes ajoutés et

les données à envoyer est converti en suite binaire, qui est ensuite modulée et transformée en signal électrique en fonction du lien, avant d'être transmis sur ce dernier. L'ajout d'entêtes (ou la procédure d'encapsulation) proposé par le modèle OSI est similaire au système postal d'envoi de courrier. Dans le système postal d'envoi de courrier, une personne ne peut pas envoyer une lettre sans avoir préalablement mis la lettre dans une enveloppe et avoir indiqué au moins l'adresse du destinataire sur l'enveloppe. Tout comme l'adresse du destinataire qui est une information utilisée par le système de courrier pour délivrer l'enveloppe à la bonne adresse, les informations d'entêtes sont utilisées pour une bonne transmission du message au destinataire (ISO/IECJTC1, 1994).

La procédure d'encapsulation comporte toutefois deux différences notables en comparaison avec l'envoi de courrier. Premièrement, les entêtes contiennent plus d'informations que l'adresse du destinataire et de l'envoyeur. Deuxièmement, les entêtes ne sont pas ajoutés d'un seul trait. Si l'on part de la donnée à envoyer (la donnée que l'utilisateur souhaite envoyer ou instruction à envoyer), un entête est ajouté à cette donnée à la couche 7 (application) et cet ensemble forme le *Protocol Data Unit* (PDU) de la couche 7. Le PDU est composé d'une *Service Data Unit* (SDU) et d'un entête. Sur la couche 6, un nouvel entête est ajouté au PDU de la couche 7 formant le PDU de la couche 6, et ainsi de suite jusqu'à la couche 1. Certains PDU sont nommés. Ainsi, lorsque l'on utilise le terme « segment » on fait référence au PDU de la couche 4. Un « paquet » désigne le PDU de la couche 3 et une « trame » désigne le PDU de la couche 2. Par conséquent, la procédure d'encapsulation ressemble plutôt au rangement des poupées Matryoshka (voir Figure 2.6), où la plus petite poupée, P (la donnée à envoyer), est mise dans une poupée P+1 (sur lequel des informations d'entête sont indiquées). La poupée P+1 est ensuite mise dans une poupée P+2, elle-même mise dans une poupée P+3 et ainsi de suite (ISO/IECJTC1, 1994 ; Alani et al., 2014).

Les informations contenues dans les entêtes dépendent du protocole utilisé

FIGURE 2.6 – Analogie rangement - Poupée Matryoshka

Source: Igor Drondin, tirée de Pixabay.com, licence Pixabay

sur la couche. Sur la couche 2, le protocole le plus couramment utilisé est le protocole Ethernet. Sur la couche 3, c'est l'adressage par adresse IP, et sur la couche 4, ce sont les protocoles Transport Control Protocol (TCP) / User Datagram Protocol (UDP) (ALANI et al., 2014). Les informations d'entête, tout comme la donnée à envoyer, peuvent être lues, modifiées ou usurpées par un attaquant. En effet, même si la donnée à envoyer est importante et peut-être des données de vie privée, certaines informations d'entêtes, telles que l'adresse IP, peuvent révéler l'identité d'une personne sur Internet. L'anonymat sur Internet peut être tout aussi important que la vie privée pour certaines personnes (RAINIE et al., 2013). Par exemple, en analysant les adresses IP, un attaquant peut avoir accès à des informations sur l'utilisation de l'utilisateur, telles que les sites visités, l'heure de connexion, la durée de la connexion, etc. Ainsi, les méthodes de protection des données à envoyer (par exemple, le chiffrement) doivent également s'appliquer aux informations d'entêtes.

Toutefois, il n'est pas possible de chiffrer toutes les informations (entêtes et la donnée) avec le destinataire. Le chiffrement avec le destinataire ne peut protéger que les informations d'entêtes de la couche 5 à la couche 7 et la donnée à envoyer (KUMAR et al., 2014). Ces informations sont en réalité destinées à l'ordinateur du destinataire. Les informations de la couche 3 sont destinées aux routeurs (intermédiaires) formant Internet. Les informations de la couche 3 ne doivent donc pas être chiffrées avec le destinataire au risque de les rendre incompréhensibles par les routeurs. De même, les informations de la couche 2

ne peuvent être chiffrées avec le destinataire au risque de rendre les informations incompréhensibles pour les commutateurs (équipements couche 2). Ainsi, pour diminuer les risques de piratage, il est en réalité nécessaire d'opérer un cumul de chiffrement avec plusieurs équipements différents. Il est, par exemple, possible de faire un chiffrement avec un équipement couche 2, avec un serveur Virtual Private Network (VPN) et avec le destinataire (FERGUSON et al., 1998). Toutefois, ce principe de cumul de chiffrement n'est que très rarement appliqué en pratique, car il est coûteux en calcul et en bande passante.

Il est également important de souligner que le chiffrement d'une couche N ne chiffre que le SDU de cette couche. Ainsi, si un équipement opère un chiffrement couche 3, seules les informations de la couche 4 à la couche 7 et la donnée à envoyer seront chiffrées. L'entête de la couche 3 restera en clair. C'est pour cette raison que, par exemple, pour cacher son adresse I.P. (information sur l'entête de la couche 3), un chiffrement couche 3 n'est pas suffisant. Il faut en réalité rajouter un intermédiaire destinataire (serveur VPN) puis effectuer le chiffrement avec ce dernier (HAY et al., 2009). Le rajout d'un intermédiaire destinataire implique un dédoublement de la couche 3 avec l'adresse IP du site initial en SDU. Autre point important à relever concernant la couche 2, cette couche est divisée en deux sous-couches : la sous-couche *Medium Access Control* (MAC) et la sous-couche *Logical Link Control* (LLC). La sous-couche LLC est en réalité intégrée dans le SDU de la couche 2 et est donc chiffrée lorsqu'il y a un chiffrement de la couche 2 - la sous-couche MAC, elle, reste en clair (KLIAZOVICH, 2006).

Les entêtes sont en partie enlevés pendant le transfert de la donnée (par exemple commutation et routage) et à la réception par l'ordinateur destinataire de telle sorte à ce que le destinataire ne reçoit que la donnée qui lui était destinée.

2.3 La présence de multiples points d'attaque

Comme nous l'avons vu, Internet nous permet de communiquer sur de longues distances. Toutefois, ce grand réseau logique permet aussi à un attaquant d'avoir plusieurs points d'entrée dans le réseau (SHIN et al., 2018). Par exemple, dans la Figure 2.7, il y a une communication entre des utilisateurs et un serveur web. On remarque qu'un attaquant peut attaquer la communication à plusieurs endroits : il peut être directement sur l'ordinateur des utilisateurs s'il a pu, dans le passé, installer un logiciel espion ; il peut être dans le réseau local en contrôlant le routeur ou le point d'accès ; il peut être en contrôle d'un des intermédiaires se trouvant sur Internet ; ou encore, il peut être sur le serveur web.

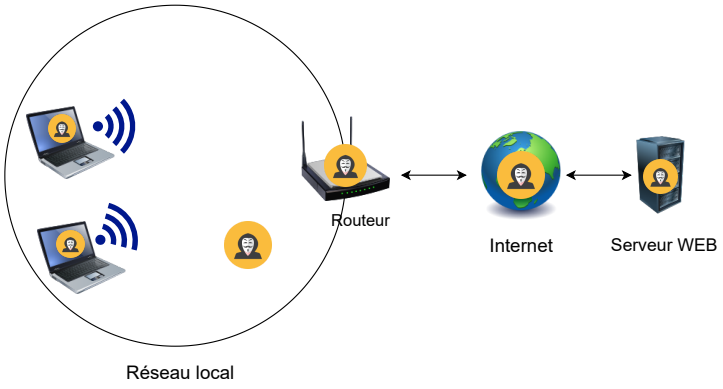

FIGURE 2.7 – Les attaquants peuvent attaquer à plusieurs endroits

Source: traitement auteur : icônes sous licence open source diagrams.net

Dans cette thèse, nous nous intéressons aux attaques réalisées dans un réseau local IEEE 802.11 (Wi-Fi) (voir Figure 2.8). Nous étudions trois attaques en particulier : l'attaque par faux point d'accès, l'attaque par déauthentification et l'attaque par brouillage. Nous cherchons à détecter ces attaques en utilisant les informations d'entête de la couche 2 et certaines informations de la couche 1. La présence de ces attaques peut également être détectée sur les couches supérieures (3 et 4). Nous avons choisi de détecter ces attaques en utilisant uniquement les informations de la couche 2 et 1, pour deux raisons. La première réside dans le fait que les informations de la couche 1 et l'entête de

la couche 2 ne sont pas chiffrées lorsque le réseau local est chiffré. En effet, le chiffrement utilisé dans un réseau local est un chiffrement de la couche 2. En nous cantonnant aux deux premières couches, nous pouvons ainsi créer un seul outil capable de détecter ces attaques, lorsqu'elles sont réalisées contre des réseaux sans-fil Wi-Fi publics (sans chiffrement) et avec chiffrement. La deuxième raison réside dans le fait que les informations des couches supérieures peuvent contenir des identifiants personnels tels que l'adresse IP. L'adresse IP est juridiquement considérée comme une donnée personnelle (GRÉGOIRE, 2009) et le traitement de ce type de données implique l'application du RGPD imposant des contraintes techniques telles que la demande et la gestion du consentement des utilisateurs du réseau, la tenue d'un registre de mise en œuvre, la mise en place, dans certains cas, d'un Délégué à la Protection des données (DPO) (VOIGT et al., 2017), etc. Concernant l'adresse MAC (information sur la couche 2), nous ne considérons que l'adresse MAC du point d'accès Wi-Fi et pas celles des équipements.

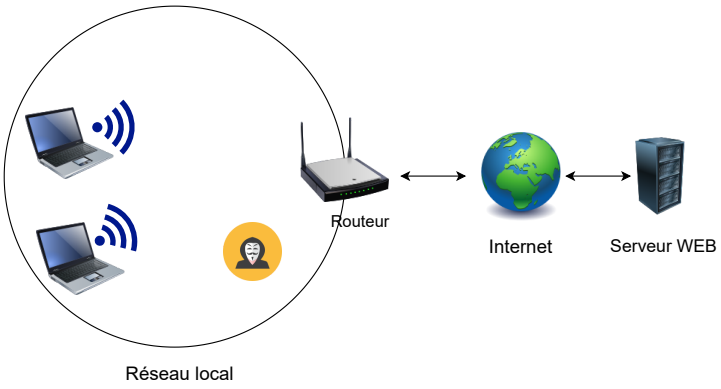

FIGURE 2.8 – Positionnement de la thèse - attaques sur les réseaux sans-fil Wi-Fi

Source: traitement auteur : icônes sous licence open source diagrams.net

Comme nous nous intéressons aux réseaux locaux IEEE 802.11 (Wi-Fi), il convient à présent de détailler cette norme.

2.4 La norme IEEE 802.11 - Wi-Fi

La norme IEEE 802.11 définit le transfert sans fil d'informations entre équipements sur de courtes et moyennes distances. Elle spécifie notamment le lien physique, les différents modes de connexion, les moyens d'accéder à la ressource et les différents types de trames pouvant être émises selon cette norme (IEEE802.11Standard, 2021). La norme originelle IEEE 802.11 a également subi plusieurs amendements modifiant la couche physique et la couche liaison de données. Les amendements majeurs de la couche physique sont présentés dans le tableau 2.1.

Version	Année	Bandes	Débit max	Génération
802.11	1997	2.4 GHz	2 Mbit/s	Originelle
802.11b	1999	2.4 GHz	11 Mbit/s	1ère
802.11a	1999	5 GHz	54 Mbit/s	2ème
802.11g	2003	2.4 GHz	54 Mbit/s	3ème
802.11n	2009	2.4 GHz, 5 GHz	150 Mbit/s	4ème
802.11ac	2013	5 GHz	3,4 Gbps	5ème
802.11ax	2021	1 GHz - 7,1 GHz	10,5 Gbps	6ème

Table 2.1 – Les différents amendements de la norme IEEE 802.11

Source: traitement auteur : forme du tableau inspirée de (Jud, 2021), données reprises de la norme IEEE 802.11 (IEEE802.11Standard, 2021), Copyright © 2021, IEEE

La norme IEEE 802.11 prévoit que différents modes de connexions peuvent être créées.

Les modes de connexions

La norme définit plusieurs modes de connexion. Les différents modes sont le mode ad-hoc (utilisant notamment le Wi-Fi Direct), le mode infrastructure, le mode système de distribution sans-fil (*Wireless Distribution System* (WDS)) et le mode Mesh. Le mode Wi-Fi ad-hoc permet à deux équipements (téléphones, ordinateurs...) de communiquer directement sans infrastructure Wi-Fi préexistante. Le mode ad-hoc a également une variante qui est le mode

ad-hoc en multi-hop qui permet de créer un réseau ad-hoc et potentiellement de le connecter à Internet (Anastasi et al., 2004). En ce qui concerne le mode infrastructure, il permet à deux ou plusieurs équipements de communiquer via un point d'accès. Le point d'accès peut également être connecté à Internet pour que les équipements puissent échanger avec d'autres équipements sur Internet (Atelin, 2008). Le mode WDS permet de créer et d'utiliser des liens sans fil Wi-Fi pour connecter des réseaux entiers et pas seulement des équipements. Ce mode permet, par exemple, de connecter deux réseaux locaux indépendants. Le mode WDS permet aussi aux équipements Wi-Fi de fonctionner en mode répéteur pour répéter la communication d'une antenne dans le but d'étendre sa couverture et d'améliorer la communication (Tulloh et al., 2017). Enfin, le mode Mesh permet de connecter des réseaux sans fil en utilisant un ensemble de points d'accès interconnectés entre eux avec plusieurs liens de redondance (Audeh, 2004). Le mode infrastructure est le mode le plus couramment utilisé pour connecter des équipements de manière sans fil dans un réseau local. Pour cette raison, dans cette thèse, nous nous intéressons uniquement aux réseaux Wi-Fi en mode infrastructure.

Pour créer différents types de connexions, les équipements disposant d'une carte Wi-Fi peuvent fonctionner selon plusieurs modes : le mode point d'accès (*master* en anglais), le mode client, le mode ad-hoc, le mode monitor et le mode répéteur. Pour créer un réseau Wi-Fi en mode infrastructure, il faut au moins un équipement en mode point d'accès. Les autres équipements doivent être en mode client. Lorsqu'un équipement est en mode point d'accès, il émet des trames de *beacon* pour informer les clients de sa zone de couverture de sa présence. Les trames de *beacon* contiennent également les fonctions et le niveau de sécurité offerts par le point d'accès (IEEE802.11Standard, 2021). Les équipements en mode client doivent s'associer au point d'accès avant toute communication. Pour s'associer, une procédure doit être suivie : les équipements doivent faire une recherche de points d'accès disponibles, faire une demande d'authentification et finalement faire une demande d'association. La recherche

de points d'accès peut être active ou passive.

Lorsque la recherche de point d'accès est passive, les clients vont utiliser les informations se trouvant dans les trames de *beacons* pour préparer la requête d'authentification. Lorsque la recherche est active, les clients vont chercher, dans leur mémoire de stockage, les détails des points d'accès sur lesquels, ils se sont déjà connectés (IEEE802.11STANDARD, 2021). Ensuite, ils envoient une requête sonde (ou *probe request* en anglais) pour demander si un point d'accès connu est actif dans les environs. Si le point d'accès est présent, il répond en émettant une réponse de sonde (ou *probe response* en anglais). Le client peut ensuite envoyer une trame d'authentification et d'association. Une fois associés, les clients peuvent échanger avec le point d'accès. Deux ou plusieurs points d'accès peuvent gérer un même réseau et donc émettre le même Service Set Identifier (SSID) ou nom de réseau. Cependant, ils n'émettent pas exactement les mêmes informations dans leurs trames de *beacon* (VAUGHAN-NICHOLS, 2003). Lorsque cela se produit, le client s'authentifie avec le point d'accès ayant les caractéristiques déclarées les plus intéressantes (par exemple, la puissance d'émission la plus élevée ou le meilleur débit). Concernant la phase d'authentification, il existe deux types d'authentification : l'authentification sans chiffrement (ou "libre") et l'authentification avec chiffrement. Le chiffrement peut être effectué par l'échange d'une clé partagée ou en utilisant des identifiants et des mots de passe uniques pour chaque client (KHASAWNEH et al., 2014 ; RUMALE et al., 2011 ; YOUNES, 2020a).

Pour fonctionner en mode point-à-point, les équipements Wi-Fi doivent être configurés en mode ad-hoc, ce qui leur permet de communiquer directement sans passer par un point d'accès intermédiaire (KLIMIASHVILI et al., 2020). En mode répéteur, l'équipement Wi-Fi répète la communication d'un point d'accès en réémettant ses trames. Enfin, le mode moniteur permet à un équipement d'écouter et d'enregistrer des trames. Il y a une différence entre le mode monitor et le mode client. Dans le mode client, une fois associée au point d'accès, la carte Wi-Fi écoute le canal, reçoit toutes les trames émises sur le canal et ne

conserve que les trames qui lui sont adressées. Dans le mode moniteur, la carte Wi-Fi ne s'associe pas préalablement à un point d'accès, elle écoute le canal, reçoit toutes les trames émises sur le canal et les conserve toutes, même celles qui ne lui sont pas destinées (IEEE802.11Standard, 2021). Ce mode est particulièrement utile pour effectuer des mesures sur le réseau, mais également pour réaliser des attaques, notamment l'attaque par écoute.

Une fois configurés dans le mode de fonctionnement requis, les équipements peuvent émettre des trames en respectant les contraintes imposées par la couche liaison de données et la couche physique.

La couche liaison de données de la norme IEEE 802.11

La couche liaison de données est divisée en deux sous-couches : la sous-couche MAC et la sous-couche LLC. Il existe plusieurs versions de la sous-couche MAC, telles que IEEE 802.11q et IEEE 802.11w (IEEE802.11Standard, 2021). Dans cette thèse, nous nous intéressons à la sous-couche MAC IEEE 802.11 dans sa version originelle. La sous-couche MAC contrôle l'accès à la ressource et s'assure de la bonne transmission des trames. La sous-couche IEEE 802 LLC gère, entre autres, l'authentification, les modes d'association et l'économie d'énergie. La norme 802.11 définit trois types de trames qui peuvent être émises par les équipements Wi-Fi : les trames de données, de contrôle et de gestion (IEEE802.11Standard, 2021). Les trames de données sont utilisées pour envoyer des données utilisateur (requête, page web, photo...), les trames de gestion sont utilisées pour gérer le réseau (la procédure d'authentification, l'envoi de trames de *beacon*, la deassociation d'un client...) et enfin, les trames de contrôle sont envoyées pour contrôler l'émission des trames de données. Les trames de contrôle incluent notamment les trames Request To Send (RTS), Clear to Send (CTS) et les trames d'acquittement (ACK). La norme IEEE 802.11 définit la structure d'une trame qui est

composée d'un entête MAC, d'un corps et d'un Frame Check Sequence (FCS) pour le contrôle d'intégrité (IEEE802.11STANDARD, 2021 ; YOUNES, 2020c). Les champs de l'entête MAC et leur fonction sont décrits dans le tableau 2.2.

Champs	Description
Frame Control	Composé de 11 sous-champs et contient les informations de contrôle (voir tableau 2.3)
Duration/ID	Information sur deux octets qui indique la valeur du *Network Allocation Vector* (NAV)
Address 1	L'adresse MAC du receveur
Address 2	L'adresse MAC du transmetteur
Address 3	L'adresse MAC du point d'accès si c'est une trame de gestion, l'adresse source ou l'adresse destination si c'est une trame de données, null si c'est une trame de contrôle
Sequence Control	Numéro qui indique les numéros de trame, détecte les trames dupliquées et met les trames en ordre pour les couches supérieures
Address 4	Indique si la trame de donnée ou de gestion est une retransmission
Frame Body	Champ variable qui contient le SDU. La taille maximale des champs de données est 2312 octets
FCS	Un champ de 4 octets contenant une information pour la détection d'erreur

TABLE 2.2 – Champs de l'entête MAC

Source: traitement auteur : description des champs de l'entête MAC telle que fournie dans la norme IEEE 802.11 accessible depuis (IEEE802.11STANDARD, 2021), Copyright © 2021, IEEE

La structure de la trame IEEE 802.11 et de l'entête MAC varie en fonction du type de trame (gestion, contrôle ou données). Les trames de gestion et de contrôle sont créées à la couche 2 et destinées uniquement aux équipements de la couche 2 (les cartes Wi-Fi dans la norme IEEE 802.11) (IEEE802.11STANDARD, 2021). Par conséquent, le champ *Frame Body* de ces types de trames n'est pas identique entre eux, ni avec celui des trames de données. Les trames de gestion ont un champ *Frame Body* réduit tandis que les trames de contrôle n'ont pas de champ *Frame Body* (voir Figure 2.9). Les

Sous-champs	Description
Protocol version	Utilisé lorsqu'il y a une différence fondamentale d'incompatibilité entre deux amendements de norme IEEE 802.11
Type	Indique le type de la trame
Subtype	Indique le sous-type de la trame
To DS	1, si destinataire est un système de distribution
From DS	1, si trame originaire d'un système de distribution
MoreFrames	Indique si plus de trames du même type sont à suivre
Retry	Indique si la trame de donnée ou de gestion est une retransmission
Power Management	Indique si l'envoyeur est en mode active ou économie d'énergie
More Data	Indique à un équipement en mode économie, que le point d'accès a plus de trames à envoyer
Security	Indique si le chiffrement et l'authentication sont utilisés
Reserved	Indique si toutes les trames de données peuvent être traitées en ordre

TABLE 2.3 – Sous-champs du champ *Frame Control*

Source: traitement auteur : description des sous-champs du champ *Frame control* telle que fournie par la norme IEEE 802.11 (IEEE802.11STANDARD, 2021), Copyright © 2021, IEEE

trames de gestion comportent tous les champs de la structure de trame IEEE 802.11, tandis que les trames de contrôle sont plus petites et ne disposent pas de tous les champs. En revanche, les trames de données respectent la structure de la trame IEEE 802.11 en tout point et peuvent avoir un *Frame Body* pouvant aller jusqu'à 2312 octets. Leur champ *Frame Body* contient le SDU de la couche 2, c'est-à-dire les entêtes de la couche 3 à la couche 7 ainsi que les données à envoyer.

Autre différence entre les différents types de trames est que seules les trames de données et certaines trames de gestion (avec l'amendement de la couche liaison de données IEEE 802.11w) peuvent être chiffrées lorsqu'un réseau Wi-

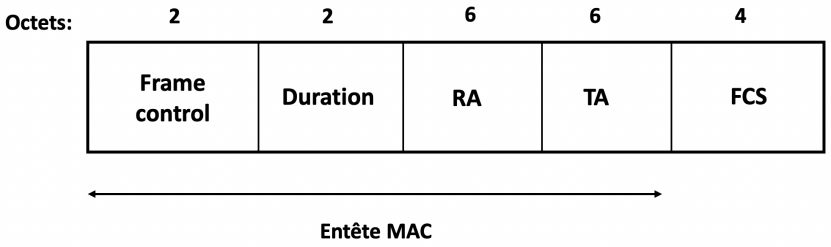

FIGURE 2.9 – Structure d'une trame de contrôle : la trame *Request To Send*

Source: traitement auteur : reproduction de structure de la trame de contrôle telle que définie par la norme IEEE 802.11 (IEEE802.11STANDARD, 2021), Copyright © 2021, IEEE

Fi est sécurisé. Les trames de contrôle ne peuvent pas être chiffrées et sont toujours envoyées en clair.

Conformément au modèle OSI, une fois les entêtes ajoutés pour les trames de données ou créées à la couche 2 lorsqu'il s'agit de trames de gestion et de contrôle, ces trames sont transmises au processus qui gère la couche physique de la norme IEEE 802.11 pour la transmission sur le lien.

La couche physique de la norme IEEE 802.11

Sur la couche physique, la trame est convertie suite binaire, puis modulée et transformée en signal électrique avant d'être émise sur un canal de fréquence donné. La couche physique détermine la conversion, la modulation et le codage canal. Il existe plusieurs amendements de la couche physique dans la norme IEEE 802.11, tels que IEEE 802.11 a, b, g ou n. La couche physique de la norme IEEE 802.11 est divisée en deux sous-couches : la sous-couche *Physical Layer Convergence Protocol* (PLCP) et la sous-couche *Physical Medium Dependent* (PMD) (IEEE802.11STANDARD, 2021). La sous-couche PLCP détermine l'amendement de la norme IEEE 802.11 à utiliser, tandis que la sous-couche PMD applique les contraintes telles que la fréquence ou la technique de modulation en fonction de la version qui a été sélectionnée. Le mode de transmission peut être différent d'une génération à une autre. Dans la ver-

sion originelle de la norme IEEE 802.11, deux modes de transmission possibles sont : le *Direct Sequence Spread Spectrum* (DSSS) et le *Frequency Hopping Spread Spectrum* (FHSS). Dans l'amendement IEEE 802.11 b, seul le mode de transmission DSSS est autorisé. Avec les autres amendements, le mode de transmission est l'*Orthogonal Frequency Division Multiplexing* (OFDM). Pour la 6ème génération du Wi-Fi, le mode de transmission est l'*Orthogonal Frequency Division Multiple Access* (OFDMA) (IEEE802.11STANDARD, 2021).

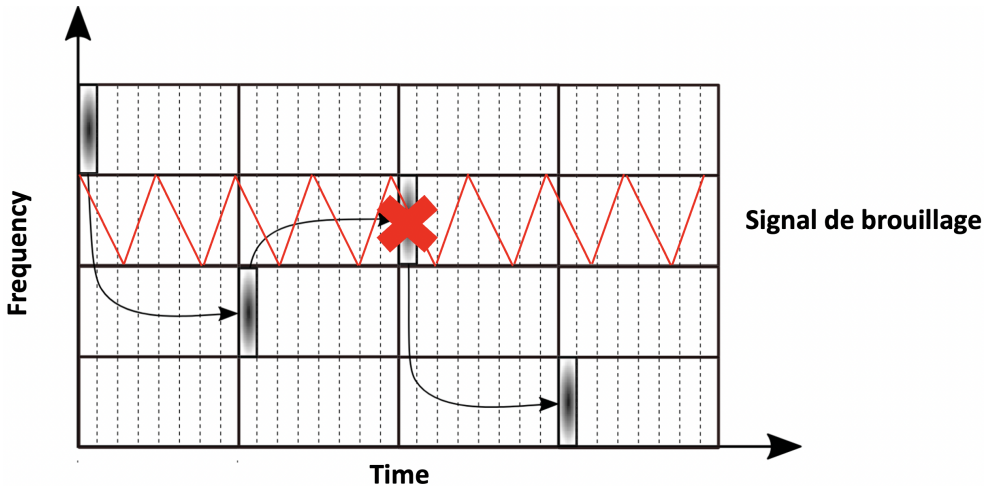

FIGURE 2.10 – *Frequency Hopping Spread Spectrum* (FHSSS) avec interférence

Source: traitement auteur : illustration modifiée avec l'accord de l'auteur principal Artur N. de São José, tirée de (JOSÉ et al., 2019), Copyright © 2019, IEEE

Le *Frequency Hopping Spread Spectrum* est une technique qui consiste à transmettre des signaux radio en alternant rapidement dans un ensemble de canaux de fréquences en fonction d'une séquence pseudo-aléatoire connue par l'émetteur et le récepteur (YOUNES, 2020b). Le FHSS est particulièrement intéressant en cas d'interférences. La transmission en FHSS permet d'avoir une transmission plus robuste aux interférences intentionnelles et non intentionnelles. En effet, comme le montre la Figure 2.10, en émettant sur plusieurs canaux de fréquences, l'équipement arrive à transmettre sans subir un impact significatif de ce type de signal d'interférence – qui n'affectent qu'un canal. La couche physique de la version originelle de la norme IEEE 802.11 indique que

la transmission de données en FHSS peut atteindre jusqu'à 2 Mbit/s (Brain et al., 2004).

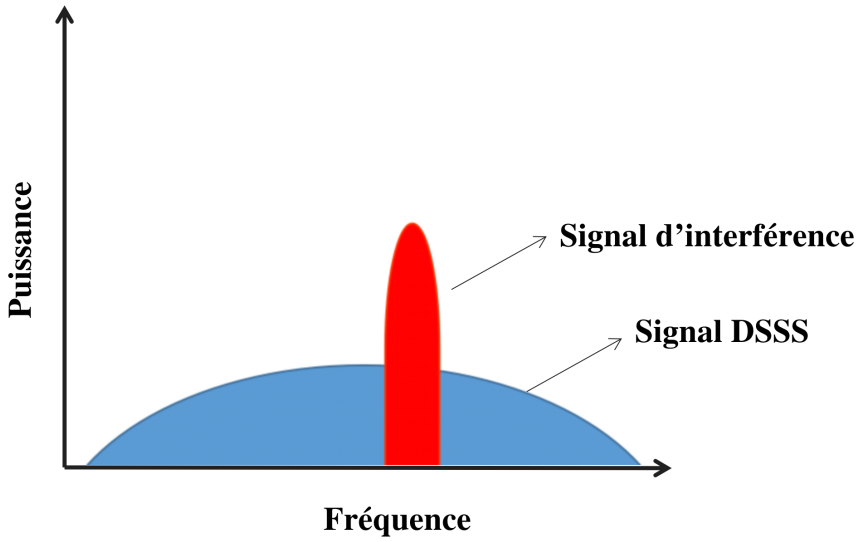

Figure 2.11 – *Direct Sequence Spread Spectrum* (DSSS) avec interférence

Source: traitement auteur : inspirée de (eroberts, 2003)

La transmission en DSSS consiste à transmettre des signaux en bande large. Une séquence connue à la fois de l'émetteur et du récepteur est utilisée pour étaler, transmettre et reconstituer le signal. La transmission en bande large permet de réduire la puissance de transmission du signal, ce qui permet aux récepteurs de distinguer plus facilement ce signal par rapport à un signal d'interférence (Younes, 2020b). En outre, comme illustré sur la Figure 2.11, en étalant le spectre, un signal d'interférence qui ne couvre qu'un canal a peu d'impact sur le signal transmis. Avec la transmission en DSSS, les informations de la partie du signal impactée peuvent être reconstituées lors du décodage du canal. Enfin, en émettant en bande large, le signal perturbe peu les autres communications car il est transmis à faible puissance. Selon la technique de codage et de modulation utilisée, il est indiqué que, avec l'amendement IEEE 802.11 b, le débit d'une transmission en DSSS varie de 1 Mb/s à 11 Mb/s (McCune, 2000).

La transmission en Orthogonal Frequency Division Multiplexing (OFDM)

est une technique de modulation et de transmission qui consiste à répartir le signal sur un grand nombre de sous-porteuses orthogonales, chacune étant modulée individuellement à bas débit. Bien que l'OFDM ne soit pas particulièrement robuste à l'interférence, elle permet d'atteindre des débits importants en transmission de données (par exemple, jusqu'à 10,5 Gbit/s avec le Wi-Fi 6) (Jun et al., 2003).

La communication Wi-Fi doit se faire sur un canal dans les bandes de fréquences de 2,4 GHz ou 5 GHz. La bande de fréquences de 2,4 GHz est divisée en 14 canaux, tandis que sur la bande de fréquences de 5 GHz, il peut y avoir de 36 à 165 canaux différents. Tous les canaux ne sont pas nettement séparables et peuvent se recouvrir. Sur la bande 2.4 GHz, par exemple, seuls trois canaux (1, 6, 11) sont nettement séparés. Pour cette raison, il est souvent recommandé d'utiliser ces canaux pour réduire les interférences non intentionnelles. La bande 2.4 GHz est également partagée avec les protocoles Bluetooth et Zigbee, ce qui peut causer des interférences lorsque ces équipements sont à proximité (Younes, 2020b). La norme régule la puissance d'émission sur les canaux. Par exemple, pour les fréquences de 2400 à 2454 MHz, la puissance maximale d'émission en intérieur et extérieur est de 100 mW. Il est important de noter que les bandes 2.4 GHz et 5 GHz sont des bandes libres de droits sur lesquelles les équipements peuvent émettre sans demander d'autorisation, à condition de respecter les contraintes législatives et réglementaires (IEEE802.11Standard, 2021).

Il convient également de noter que les amendements compatibles avec la bande 5 GHz ne proposent que des transmissions en OFDM (voir tableau 2.1). Ainsi, sur la bande 5 GHz, toutes les trois types de trames sont émises en OFDM. En revanche, sur la bande 2.4 GHz, l'amendement IEEE 802.11 b permet l'envoi de trames en DSSS. Les équipements fonctionnant sur la bande 2.4 GHz peuvent ainsi choisir, en fonction de l'état du canal et sous réserve du respect des spécifications de la norme, s'ils souhaitent émettre en DSSS ou en OFDM. Aussi, certaines trames, notamment les trames de bacon, sont toujours

émises en DSSS pour des raisons de rétrocompatibilité, notamment lorsque le point d'accès est compatible avec l'amendement IEEE 802.11 b. Enfin, il est à noter que la transmission en FHSS n'est possible qu'avec la version originelle de la norme IEEE 802.11 (IEEE802.11STANDARD, 2021).

Sur la couche physique, la trame est convertie en une suite binaire, qui est ensuite modulée et transformée en signal électrique. Une fois, le canal d'émission déterminé en fonction de la version de la norme, un équipement Wi-Fi doit vérifier que le canal est disponible avant d'émettre le signal.

L'accès à la ressource pour la transmission

Les équipements Wi-Fi communiquent en utilisant un canal de fréquence. Toutefois, dans un même réseau, plusieurs équipements Wi-Fi peuvent fonctionner sur un même canal. Pour coordonner les équipements et éviter qu'ils n'émettent tous en même temps, la norme IEEE 802.11 propose deux modes de fonctionnement : la fonction de coordination distribuée (DCF) et la fonction de coordination par point (PCF) (IEEE802.11STANDARD, 2021). La DCF est la fonction par défaut, tandis que la PCF est optionnelle. Avec la fonction PCF, un équipement devient le coordinateur des transmissions. En général, en mode infrastructure, le point d'accès est le coordinateur et il coordonne la transmission en indiquant des paramètres dans ses trames de *beacon* (BANERJI et al., 2013).

La DCF implique que les équipements déterminent par eux-mêmes s'ils peuvent accéder au canal. La méthode la plus utilisée pour déterminer l'occupation du canal est celle du *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA). Le CSMA/CA est différent du mécanisme de *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD) utilisé par les équipements sur les liens filaires (IEEE802.11STANDARD, 2021). Les équipements sans-fil fonctionnent en semi-duplex et ne peuvent, par conséquent, pas écouter

et émettre en même temps. Dans le mécanisme du CSMA/CA, les équipements adoptent un comportement qui consiste à éviter les collisions. Avant de transmettre, ils écoutent le canal pour déterminer si le canal est occupé. Deux méthodes existent pour déterminer s'il y a une transmission en cours sur le canal : le *Clear Channel Assessment-Energy Detect* (CCA-ED) et le *Carrier Sense/Clear Channel Assessment* (CS/CCA) (IEEE802.11Standard, 2021). Le CCA-ED consiste à vérifier la puissance émise sur le canal. Si la puissance mesurée est plus grande qu'un seuil appelé CCA-ED, l'équipement supposera que le canal est occupé. Avec la méthode CS/CCA, l'équipement considère que le support est occupé lorsqu'il arrive à reconnaître le préambule d'une trame.

Si le canal est libre pendant une durée égale à un *Distributed Inter Frame Space* (DIFS), alors l'équipement peut émettre la trame (IEEE802.11Standard, 2021). Le DIFS varie en fonction de l'amendement de la couche physique de la norme IEEE 802.11. Par exemple, le DIFS est de 50 µs pour l'amendement IEEE 802.11 b et de 34 µs pour l'amendement IEEE 802.11 n. Si le canal est occupé, l'équipement attendra que le canal se libère pendant une durée égale à un DIFS, puis attendra un temps déterminé par un algorithme appelé l'algorithme de *back-off* (IEEE802.11Standard, 2021). L'algorithme de *back-off* tire une valeur de compteur de manière aléatoire entre 0 et la taille de la fenêtre de contention. Si la valeur tirée est égale à 0, l'équipement peut transmettre immédiatement. Si la valeur tirée est supérieure à 0, l'équipement diminue le compteur d'une unité pour chaque *Slot time* pendant lequel le canal est libre. L'équipement ne transmettra que lorsque ce compteur atteindra 0. La valeur de la fenêtre de contention est choisie de manière dynamique en fonction du nombre de transmissions observées avec le destinataire des trames dans une fenêtre. La valeur d'une *slot time* dépend de la couche physique utilisée, ainsi que de la bande de fréquence utilisée par l'équipement Wi-Fi (IEEE802.11Standard, 2021).

Le mécanisme du CSMA/CA ne permet toutefois pas de répondre aux problèmes de l'équipement Wi-Fi caché (*Hidden node* en anglais) et de l'équipe-

ment Wi-Fi exposé (*Exposed node* en anglais) (IEEE802.11STANDARD, 2021). L'équipement Wi-Fi caché correspond au cas où un équipement Wi-Fi A est hors de portée d'un autre équipement B mais se trouve dans la zone de couverture du point d'accès. L'équipement B est en train de transmettre avec le point d'accès, mais l'équipement A ne détectera pas cette transmission et pourra transmettre alors que le canal est occupé. L'équipement exposé correspond au cas où un équipement détecte que le canal est occupé, mais en réalité, l'équipement aurait pu transmettre sa trame parce qu'elle est destinée à un équipement ne se trouvant pas dans la zone de couverture de la transmission en cours (ZHONG et al., 2015). Pour résoudre ces problèmes, la norme propose d'activer l'émission de trames de contrôle RTS/CTS et ACK. Il est important de noter que les trames RTS/CTS et ACK ne peuvent être utilisées qu'avec les trames de données et ne peuvent pas être utilisées avec les trames de gestion et encore moins avec elles-mêmes, c'est-à-dire avec les trames de contrôle.

Les réseaux Wi-Fi doivent faire face à différentes menaces.

Les différentes menaces contre le protocole Wi-Fi

Les réseaux Wi-Fi doivent faire face à plusieurs types de menaces, qu'elles soient non intentionnelles - comme des points d'accès indésirables - ou intentionnelles (YOUNES, 2020d), telles que l'écoute illégale, les attaques de déni de service, les attaques d'usurpation d'identité, les attaques contre le chiffrement et les procédures d'authentification, ainsi que les intrusions..

L'attaque par intrusion consiste à accéder sans autorisation à un réseau Wi-Fi, par exemple, les points d'accès Wi-Fi qui n'ont pas été sécurisés par leur propriétaire. Étant donné que le réseau Wi-Fi est un réseau sans-fil, le support n'est pas isolé et protégé, mais accessible à tous. Par conséquent, une personne peut accéder au réseau dans le but de commettre des actions malveillantes telles que l'utilisation frauduleuse d'une carte bancaire ou le téléchargement illégal (YOUNES, 2020d). Pour contrer ce problème, un contrôle d'accès peut

être mis en place, avec ou sans chiffrement. Une autre attaque, **l'attaque par écoute**, consiste à intercepter les transmissions radio émises sur un canal Wi-Fi pour tenter de voler des informations ou des données. L'attaque par écoute peut être contrecarrée en utilisant le chiffrement (Xia et al., 2005). Cependant, le chiffrement n'est pas une solution infaillible, car l'algorithme de chiffrement utilisé peut contenir des failles et des vulnérabilités. Les chiffrements par clé WEP et par Wi-Fi Protected Access 1 (WPA 1) ont déjà été compromis et ne sont plus recommandés (Katz, 2010). Il est fortement recommandé d'utiliser au moins le chiffrement WPA 2. Ce chiffrement peut être mis en place en utilisant une clé partagée ou en utilisant un serveur d'authentification tel que le serveur *Remote Authentication Dial-In User Service* (RADIUS) (Berghel et al., 2005).

Bien que le chiffrement WPA 2 soit recommandé, son utilisation peut comporter des risques, car les attaquants peuvent utiliser des techniques telles que *le social engineering* pour retrouver la clé partagée ou copier le certificat d'authentification RADIUS depuis un terminal pour tromper le serveur d'authentification. En outre, l'attaquant peut tenter de deviner la clé de chiffrement en utilisant des méthodes telles que l'attaque par force brute ou l'attaque par dictionnaire. **L'attaque par usurpation d'identité** consiste à prendre l'adresse MAC d'un équipement dans le réseau (Raguvaran, 2014), qu'il s'agisse d'un client ou d'un point d'accès, afin de se faire passer pour cet équipement et ainsi compromettre la sécurité du réseau. Ainsi, l'utilisateur attaqué peut être tenu responsable pour les faits de l'attaquant. Pour éviter l'usurpation d'adresse MAC, il est recommandé d'utiliser le chiffrement (Adil et al., 2020).

Nous nous intéressons, à présent, à deux grandes catégories de menaces contre les réseaux sans-fil Wi-Fi : à savoir les points d'accès non autorisés et les attaques par déni de service (DoS) (Younes, 2020d ; Noor et al., 2013). Les points d'accès indésirables peuvent être des points d'accès non autorisés qui fonctionnent sur le même canal qu'un point d'accès autorisé. L'accès à un canal Wi-Fi ne pouvant être contrôlé physiquement, il est ainsi facile de

créer un point d'accès sur un canal déjà utilisé. Ce type de point d'accès indésirable peut être détecté en utilisant une cartographie ou en analysant les trames émises sur le canal pour constater la présence de deux points d'accès (Tchakounté et al., 2019). Pour endiguer cette menace, les administrateurs de réseaux sans fil peuvent changer le canal de leur point d'accès licite (Conti et al., 2016). La cartographie peut permettre de localiser et d'éteindre un point d'accès indésirable dans des zones peu denses. Cependant, cette solution est difficilement applicable dans les zones denses. Un autre type de point d'accès indésirable est le point d'accès dit *Honeypot*. Les points d'accès Honeypot sont créés par des personnes malveillantes pour attirer des clients en offrant un accès Internet gratuit (Maimon et al., 2017). Ces personnes profitent du fait que des clients sont connectés à leur point d'accès pour collecter leurs informations. Enfin, les points d'accès en configuration *Man-in-the-Middle* sont un dernier type de point d'accès indésirable. Ces points d'accès seront détaillés ci-après et pris en compte dans cette thèse.

Les menaces de Déni de Service (DoS) peuvent être intentionnelles ou non intentionnelles (Noor et al., 2013 ; Younes, 2020d ; Pirayesh et al., 2022). Les DoS non intentionnels comprennent les perturbations involontaires sur le canal de communication. Ces dernières peuvent être dues à la présence d'un point d'accès mal configuré ou à la présence d'autres équipements (non Wi-Fi) émettant sur la même fréquence (par exemple, un four à micro-ondes, un téléphone sans fil, etc.). Les interférences involontaires sont plus fréquentes sur la bande 2,4 GHz que sur la bande 5 GHz. Les menaces de DoS peuvent être des attaques, en particulier lorsqu'elles sont intentionnelles. Les attaques de DoS peuvent être effectuées sur toutes les couches du modèle OSI. Ici, nous nous intéressons aux attaques de DoS effectuées sur la voie radio (couche 1) ou en utilisant les trames de gestion ou de contrôle (couche 2). L'attaque la plus courante sur la voie radio est l'attaque par brouillage, tandis que sur la couche 2, il s'agit, par exemple, des attaques de déauthentification et de *CTS flood* (Jamal et al., 2018).

Dans cette thèse, nous nous sommes particulièrement intéressés à trois types d'attaques : l'attaque par faux point d'accès (*Man-in-the-Middle* à la couche 2), l'attaque par brouillage sur la voie radio et l'attaque par déauthentification.

2.5 Le trio d'attaques considérées dans cette thèse

Avant d'étudier l'état de l'art et de la technique concernant les trois attaques, il est important de bien les décrire et d'indiquer les formes que nous avons prises en compte dans les travaux de cette thèse.

2.5.1 *Man-in-the-Middle* : une attaque protéiforme

L'attaque *Man-in-the-Middle* (l'attaque par l'homme au milieu) est une attaque qui consiste à se placer entre deux équipements afin de contrôler leurs communications (Bhushan et al., 2017). L'attaque *Man-in-the-Middle* peut se faire sur les différentes couches du modèle OSI (Ornaghi et al., 2003). Si l'attaquant essaie de prendre la place d'un routeur, il effectue une attaque *Man-in-the-Middle* couche 3. S'il essaie de prendre la place d'un commutateur ou d'un point accès, il effectue une attaque *Man-in-the-Middle* couche 2. Et enfin, s'il essaie de prendre la place d'un site web, d'un service de courriel, il effectue une attaque *Man-in-the-Middle* couche 7 (Ornaghi et al., 2003). Dans le cadre de nos travaux, nous avons étudié les attaques *Man-in-the-Middle* couche 2, à savoir l'attaque par la création d'un faux point d'accès. La création d'un faux point d'accès peut se faire avec ou sans redirection vers le point d'accès licite.

L'attaque par faux point d'accès se déroule en trois étapes :

1. La création d'un faux point d'accès
2. L'utilisation potentielle d'une attaque par déni de service pour inciter les clients à se connecter au faux point d'accès
3. L'association volontaire ou le transfert effectif des clients vers le faux point d'accès.

L'attaquant peut également rediriger la communication vers le point d'accès licite. Dans ce cas, l'attaque est alors dénommée l'attaque par faux point d'accès avec redirection (voir Figure 2.13). Lorsque l'attaquant effectue une attaque par faux point sans redirection (voir Figure 2.12), il dispose d'une connexion Internet autonome, soit via une connexion filaire, soit parce que son ordinateur possède deux cartes Wi-Fi : une pour effectuer l'attaque et une autre pour se connecter à un autre point d'accès que celui qu'il attaque. Dans le cas de l'attaque par faux point avec redirection, l'attaquant dispose de deux cartes Wi-Fi : une pour effectuer l'attaque et une autre pour se connecter au point d'accès licite attaqué (Wang et al., 2016).

La redirection de la communication vers le point d'accès licite est réalisée de manière logicielle. La création d'un faux point d'accès peut varier en difficulté, cela dépend du niveau de sécurité du point d'accès licite. Si le point d'accès licite n'a pas de chiffrement, l'attaque par faux point d'accès est plus facile à mettre en œuvre (Amoordon, Gransart et al., 2020). En revanche, si le point d'accès licite utilise un chiffrement WPA 2 *RADIUS* ou une clé partagée, l'attaquant devra préalablement trouver la clé de chiffrement en effectuant d'autres types d'attaques, s'il souhaite que le transfert des clients du point d'accès licite vers le faux point d'accès se passe de manière automatique (Amoordon, Gransart et al., 2020). Si l'attaquant n'arrive pas à trouver la clé de chiffrement, il peut toujours attendre que les clients se connectent par erreur sur son point d'accès.

Lorsque l'attaque par faux point d'accès est réussie, elle permet de contourner le chiffrement qu'il y avait entre les clients et le point d'accès licite. L'at-

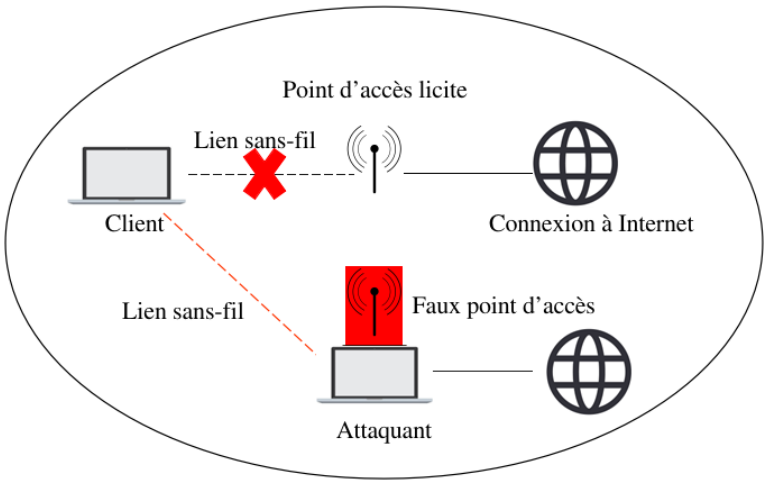

FIGURE 2.12 – Faux point d'accès sans redirection

Source: traitement auteur : illustration inspirée de (WANG et al., 2016), icône ordinateur (D3Images, tirée de freepix.com), icône Internet (conçue par Freepik, tirée de freepix.com), icône antenne (licence open source diagrams.net)

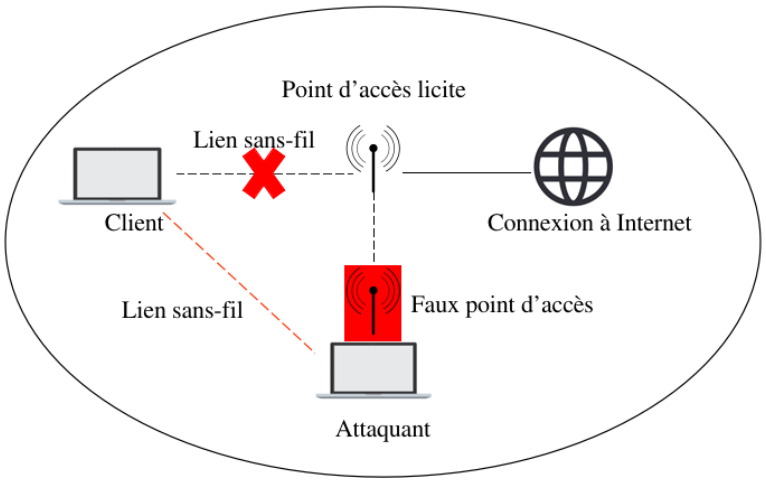

FIGURE 2.13 – Faux point d'accès avec redirection

Source: traitement auteur : illustration inspirée de (WANG et al., 2016), icône ordinateur (D3Images, tirée de freepix.com), icône Internet (conçue par Freepik, tirée de freepix.com), icône antenne (licence open source diagrams.net)

taquant devient un intermédiaire et peut lire, modifier, usurper les entêtes et la donnée à envoyer sauf s'il y a un chiffrement sur une couche supérieure. Par exemple, si un utilisateur accédant à une page web sécurisée par le protocole HTTPS est victime d'une attaque par faux point d'accès, l'attaquant pourra avoir accès aux informations de la couche 2 et 3, mais pas celles de la couche 5 à la couche 7. Sur la couche 3, il y a toutefois beaucoup d'informations inté-

ressantes telles que l'adresse IP qui va permettre à l'attaquant de connaître les sites web consultés par l'utilisateur. En connaissant les sites web consultés par l'utilisateur, l'attaquant pourra rediriger ce dernier vers une copie d'un site web et ensuite voler ses identifiants. L'attaque par faux point d'accès est souvent une première étape dans la réalisation d'attaques plus complexes (Amoordon, Gransart et al., 2020).

Il existe également une forme spéciale de l'attaque par faux point qui est plus furtive.

Forme spéciale de faux point d'accès

Lorsqu'un faux point d'accès est créé, celui-ci émet régulièrement des trames de *beacon*, rendant ainsi le point d'accès détectable. Toutefois, un attaquant peut désactiver l'émission de ces trames de *beacon* après avoir créé le faux point d'accès. Il peut ensuite utiliser une attaque par déni de service (déauthentification ou brouillage) pour déconnecter les clients du point d'accès licite. Les terminaux des clients vont alors très probablement émettre des requêtes de sondes et d'authentification pour se reconnecter au point d'accès licite. À ce moment-là, le faux point d'accès va essayer de répondre plus rapidement aux requêtes d'authentification, de sorte que les clients s'accrocheront au faux point d'accès plutôt qu'au point d'accès licite. Nous appelons cette attaque l'attaque par faux point d'accès fantôme. Nous pensons qu'elle est théoriquement possible, mais qu'elle ne peut être réalisée qu'avec une carte Wi-Fi spécifique capable de fonctionner avec une faible latence et une grande réactive afin de pouvoir répondre aux requêtes plus rapidement qu'avec une carte Wi-Fi classique.

Dans cette thèse, nous cherchons à détecter la présence d'un faux point d'accès. En adoptant une approche fondée sur la couche 2, notre méthode pourra détecter les faux points d'accès, qu'ils soient avec ou sans redirection, mais sans être capable de les différencier clairement. Pour distinguer les deux

types de faux points d'accès, il est nécessaire d'utiliser l'adresse IP, qui est une information de la couche 3, ou d'adopter une approche de détection active (voir Bratus, Cornelius et al., 2008).

Outils pour mettre en œuvre l'attaque par faux point d'accès

Plusieurs outils permettent de réaliser l'attaque par faux point d'accès. D'un point de vue matériel, une carte Wi-Fi supportant le mode point d'accès est nécessaire. La majorité des cartes Wi-Fi sur le marché sont compatibles avec ce mode. Ensuite, il faut utiliser un logiciel pour gérer le faux point d'accès. Plusieurs types de logiciels existent : notamment les scripts Scapy en Python (Dromard, 2021), les logiciels tels que Hostapd (Malinen, 2005), mdk3 (Gentili, 2009), la suite aircrack-ng (**aircrackng**) et les logiciels avec interface automatisant la création du faux point d'accès tels que Wi-Fi Pumpkin (P0cL4bs, 2015) ou Wi-Fi Pineapple (Hak5, 2008).

2.5.2 L'attaque par brouillage

L'attaque par brouillage consiste à émettre sur un canal de communication dans le but de diminuer le rapport signal à interférence pour perturber et dégrader la communication. Il existe plusieurs classifications des attaques par brouillage. Dans une étude menée par (Grover et al., 2014), les auteurs classent les attaques par brouillage en deux catégories : brouillage élémentaire et brouillage avancé. La catégorie brouillage élémentaire est elle-même divisée en deux sous-catégories : brouillage proactif et réactif. Dans la sous-catégorie brouillage proactif, on trouve les brouilleurs constants, déceptifs, réactifs et aléatoires. Nous détaillons ici les trois types de brouillage proactif ainsi que le brouillage réactif.

L'attaque par brouillage constant est une attaque qui consiste à émettre un signal de haute puissance sur une bande de fréquences en fonction d'un temps de balayage. Le signal ne représente aucune information et ne suit pas un protocole spécifique. Lorsque le signal de brouillage constant ou à balayage de fréquence est utilisé dans un réseau Wi-Fi, il a pour effet d'empêcher l'accès à la ressource, de causer des pertes de trames et d'arrêter toute communication. Cela s'applique par le fait que lorsqu'il y a un signal de brouillage constant occupant de manière permanente sur le canal, les équipements estiment que le canal est occupé, reportent et éventuellement arrêtent leurs communications (Grover et al., 2014).

Les brouilleurs déceptifs sont des brouilleurs qui émettent constamment des trames de telle sorte que les équipements pensent que le canal est toujours occupé. Contrairement aux autres types de brouilleurs, les brouilleurs déceptifs émettent de vraies trames correspondant au protocole de communication. Cependant, ils ne respectent pas les règles concernant le partage de la ressource et émettent continuellement. Nous avons remarqué, lors de nos expérimentations, que des points d'accès mal configurés peuvent être utilisés comme brouilleurs déceptifs. Les **brouilleurs réactifs** sont des équipements de brouillage qui interviennent en réponse à une action spécifique sur le réseau. Ils sont conçus pour détecter les échanges de données et générer un signal de brouillage afin de perturber la communication. Les brouilleurs réactifs offrent une efficacité supérieure à celle des brouilleurs constants, car ils ciblent spécifiquement les transmissions qui ont lieu sur le canal. Certaines cartes Wi-Fi peuvent être modifiées pour adopter ce comportement (Vanhoef et al., 2014).**Le brouilleur aléatoire**, quant à lui, oscille de manière aléatoire entre deux états : état dormant et état actif. Lorsqu'il est en état dormant, il n'émet pas de signal de brouillage, qu'il y ait du trafic ou non. Lorsqu'il est en état actif, il agit comme un brouilleur constant.

Les attaques de brouillage élémentaire et avancé n'ont pas été étudiées dans cette thèse, nous nous sommes uniquement intéressés au brouilleur constant,

car c'est le type de brouillage le plus facile à implémenter et le plus facilement accessible sur Internet. Le signal de brouillage à balayage de fréquences, couvre, de manière répétée, une bande de fréquences $[f_1, f_2]$ dans une période, T est mathématiquement représenté comme ci-dessous :

$$s(t) = A\cos\left(2\pi\left(\frac{f_2 - f_1}{2T}t + f_1\right)t\right), \quad 0 < t < T, \tag{2.1}$$

où A est l'amplitude du signal d'interférence.

(Équation 2.1 communiquée par Virginie Deniau, co-directrice de cette thèse)

Dans nos travaux, le signal de brouillage balaie la bande de fréquence $[2.4\ \text{GHz}, 2.5\ \text{GHz}]$ avec une période de $T = 10\ \mu\text{s}$.

Outils pour mettre en œuvre l'attaque par brouillage

Pour effectuer cette attaque, un attaquant peut utiliser un brouilleur clé en main, émettre le signal de brouillage préalablement implémenté en utilisant un générateur de signal arbitraire ou en utilisant une radio logicielle (Sufyan et al., 2013). Une radio logicielle est un équipement capable d'émettre et de recevoir des ondes radio. Des formes d'ondes et plus généralement des chaînes d'émission et de réception sont préalablement définies de manière logicielle avant d'être utilisées et émises par le matériel. Les langages de programmation les plus couramment utilisés pour implémenter un signal de brouillage sont notamment C, Python et Matlab. Les radios logicielles sont disponibles en plusieurs gammes (Stewart et al., 2015). Il existe, par exemple, le RTL-SDR (*RTL-SDR* s. d.) qui est peu cher et les USRP (*USRP* s. d.) qui sont relativement chères. En ce qui concerne les générateurs de signaux arbitraires, le Tektronix AWG70001A (*Tektronix AWG70001A* s. d.) peut être utilisé pour générer le signal de brouillage, bien qu'il soit plus cher que les radios logicielles.

2.5.3 L'attaque par déauthentification

L'attaque par déauthentification est une attaque de déni de service intentionnel qui consiste à envoyer plusieurs trames de déauthentification à un client afin de le déconnecter d'un point d'accès Wi-Fi (Cossa, 2014). Le protocole IEEE 802.11 définit une liste de cas et de justifications pour lesquels une trame de déauthentification peut légitimement être envoyée. Cependant, étant donné que cette trame n'est ni chiffrée ni authentifiée, elle peut facilement être usurpée par un attaquant. Ainsi, un attaquant peut se faire passer pour un point d'accès licite en indiquant simplement l'adresse MAC du point d'accès dans le champ MAC adresse source (ou SA sur la Figure 2.14) de la trame de déauthentification. Il peut ensuite envoyer plusieurs trames de déauthentification jusqu'à ce que les clients se déconnectent. Les clients n'ont aucun moyen de vérifier si le point d'accès licite est réellement l'émetteur de ces trames.

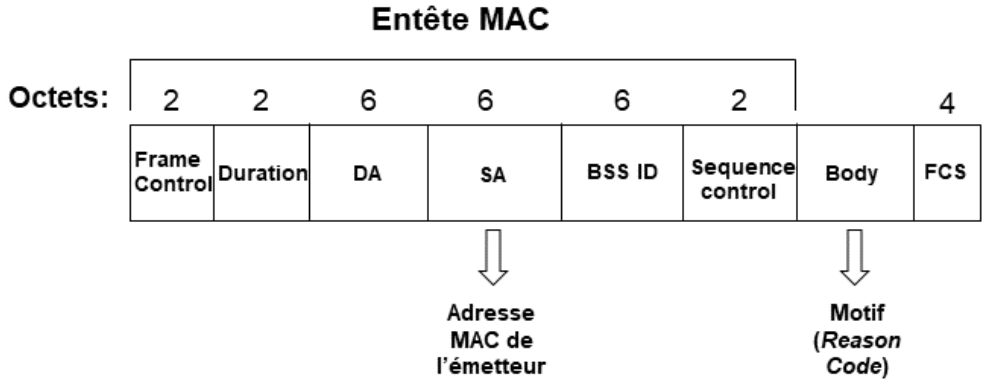

Figure 2.14 – Trame de déauthentification

Source: traitement auteur : reproduction de la trame MAC de déauthentification telle que définie par la norme IEEE 802.11 (IEEE802.11Standard, 2021), Copyright © 2021, IEEE

Outils pour mettre en œuvre l'attaque par déauthentification

Comme pour l'attaque par faux point d'accès, il est nécessaire d'utiliser une carte Wi-Fi compatible avec le mode moniteur pour effectuer cette attaque. Les logiciels pouvant être utilisés pour réaliser cette attaque incluent notamment

la suite Aircrack-ng (**aircrackng**) et le logiciel Wi-Fi Pumpkin wifipumpkin.

2.5.4 La combinaison des attaques

Dans le cadre de cette thèse, nous étudions également les combinaisons d'attaques. Après avoir créé un faux point d'accès, l'attaquant peut soit attendre passivement que les clients se connectent à son faux point d'accès, soit essayer activement de les déconnecter du point d'accès licite, en utilisant notamment l'attaque par déauthentification ou par brouillage. Nous examinons par conséquent les cas où l'attaquant combine l'attaque par faux point d'accès avec l'attaque par brouillage ou l'attaque par déauthentification

2.6 Méthodes existantes pour détecter les trois attaques

Dans cette partie, nous passerons en revue les différents travaux scientifiques et les solutions techniques existantes pour la détection des attaques par faux point d'accès, par brouillage et par déauthentification dans les réseaux Wi-Fi. Nous présenterons les méthodes de détection pour chaque type d'attaque.

2.6.1 Détection de l'attaque par faux point d'accès

L'attaque par faux point d'accès (*Man-in-the-Middle* couche 2) peut être détectée en utilisant des informations d'entête des couches supérieures (3 et 4) ou des couches basses (couches 2 et 1) (Alotaibi et al., 2016).

Méthodes utilisant les informations d'entête des couches supérieures (de 3 à 7) du modèle OSI

Dans (H. Han et al., 2011), les auteurs utilisent le temps aller-retour d'un paquet entre l'ordinateur d'un client et un serveur DNS (couche 7) pour déterminer si le client est connecté à un point d'accès licite ou à un faux point d'accès. Les auteurs soulignent qu'ils arrivent à détecter l'attaque par faux point d'accès avec une précision proche de 100 % lorsque le trafic du réseau est faible et proche de 60 % lorsque le trafic est élevé. Cette technique est facile à implémenter, mais est inefficace si, par exemple, l'attaquant dispose d'une propre connexion Internet et n'opère pas de redirection vers le point d'accès licite. Aussi, lorsque le trafic est élevé, la précision est satisfaisante et la détection est moins fiable (60% contre 100%).

Dans (Qu et al., 2010), les auteurs proposent de détecter l'attaque par faux point d'accès en utilisant le *Local Round Trip Time* calculé avec les informations dans l'entête TCP des paquets (couche 4). Cette approche est similaire à l'approche précédemment indiquée et partagent les mêmes désavantages, à savoir, le fait que la précision de la détection peut varier en fonction du trafic ou lorsqu'il y a des pertes de paquets. Dans (Nikbakhsh et al., 2012), les auteurs proposent de détecter les points d'accès en comparant les chemins IP (couche 3) des paquets. Les auteurs ne précisent toutefois pas si leur approche fonctionne lorsque l'attaquant opère une redirection vers le point d'accès licite. Dans ce cas, le routage IP s'arrêtant au point accès licite, l'affichage des routes IP (couche 3) ne permettra pas de voir la commutation (couche 2) des paquets vers le faux point d'accès avant d'arriver sur l'équipement de l'utilisateur. Pour cela, il faudrait coupler l'analyse par chemin IP avec une analyse similaire sur la couche 2.

Méthodes utilisant les informations de la couche 2

Dans la littérature scientifique, les méthodes de détection de l'attaque par faux points d'accès sont principalement fondées sur l'analyse d'informations des trames de *beacon*. Les trames de *beacon* sont un sous-type de trames de gestion. D'après la norme IEEE 802.11, un point d'accès doit envoyer des trames de *beacon* chaque 102.4 ms pour informer les clients environnants de sa présence. Les trames de *beacon* contiennent des informations telles que le nom du réseau, l'adresse MAC du point d'accès et les services proposés par le point d'accès. Quand un attaquant crée un faux point d'accès, le point d'accès coexistera avec le point d'accès licite et l'attaquant va essayer de copier et d'émettre les mêmes informations que celles présentes dans les trames de *beacon* du point d'accès licite. Certains auteurs ont proposé des méthodes (décrites ci-dessus) qui consistent à comparer différents types d'informations entre deux trames de *beacon* consécutives pour identifier des anomalies. D'autres auteurs ont proposé une analyse quantitative des trames de *beacon* reçues sur un intervalle de temps pour identifier des incohérences.

Comparaison d'informations entre deux trames de *beacon* consécutives

Une trame de *beacon* est une trame non chiffrée encapsulée dans un entête Ethernet qui contient des informations statiques comme le nom du réseau ou le SSID (Service Set IDentifier), information sur le mode du réseau (infrastructure ou ad hoc), les taux de transfert acceptés par le point d'accès et l'adresse MAC source du point d'accès licite. À côté de ces informations statiques, qui n'évoluent pas dans le temps, dans les trames de *beacon*, il y a également des informations dynamiques telles que le numéro de séquence et des informations reçues/déterminées par la couche 1 (la puissance du signal, le débit du transfert, la modulation...) qui évoluent de trame de *beacon* en trame de *beacon*. Les informations statiques sont plus facilement copiables que les informations dy-

namiques. Des auteurs ont proposé des méthodes de détection en analysant les incohérences dans les informations statiques et dynamiques entre deux trames de *beacon*.

Comparaison d'informations statiques

Dans (Bambang Setiadji et al., 2019), les auteurs proposent une méthode pour détecter la présence de faux point d'accès en analysant les adresses MAC et SSID présentes sur les trames de *beacon*. Certains attaquants débutant ne copient pas toutes les informations présentes sur la trame de *beacon*. Ils se concentrent sur le nom du réseau (SSID) parce que c'est la seule information visible par l'utilisateur. En comparant les adresses MAC et SSID entre les trames de *beacon*, la présence d'un faux point d'accès peut être détectée si l'adresse MAC reçue dans deux trames de *beacon* consécutive est différente. Cependant, cette méthode est inopérante contre les attaquants avancés qui prennent le soin de copier toutes les informations statiques.

Dans (C. Han et al., 2012), les auteurs proposent de comparer les informations telles que le nom du réseau, le type d'authentification et le type de chiffrement pour détecter la présence d'un faux point d'accès. Un réseau Wi-Fi peut être chiffré ou non chiffré. Quand le réseau est chiffré, certaines informations concernant le chiffrement (*encryption cipher*, *authentication cipher* et *authentication type*) sont indiquées dans les trames de *beacon*. Les auteurs soulignent que ces informations sont déterminées par les fournisseurs et fabricants et ajoutées par le microprogramme (*firmware*) de la carte Wi-Fi. Pour ces raisons, les auteurs indiquent que ces informations ne peuvent pas être facilement copiées. Pour copier cette information, les attaquants doivent, en réalité, modifier le microprogramme de leur carte Wi-Fi et les auteurs partent du principe que cette modification est difficile. Cependant, dans des articles scientifiques et magazines spécialisés, plusieurs auteurs présentent des tutoriels pour

modifier rapidement le microprogramme de plusieurs cartes Wi-Fi.(MEDIA, s. d. ; VANHOEF et al., 2014). Aussi, cette méthode ne fonctionne que si le réseau Wi-Fi est chiffré. Lorsque le réseau Wi-Fi est non chiffré ou public, les informations de chiffrement ne sont pas indiquées. Les réseaux Wi-Fi publics sont les réseaux Wi-Fi présents dans les aéroports, les universités, écoles publiques, les cinémas et les restaurants. Ces réseaux sont les plus vulnérables à l'attaque par faux point d'accès parce que l'attaquant n'a pas à trouver le mot de passe du réseau et peut ainsi plus facilement forcer les clients à se connecter à son faux point d'accès (AMOORDON, GRANSART et al., 2020).

Dans (C. HAN et al., 2012), les auteurs proposent de comparer les informations telles que le nom du réseau, le type d'authentification et le type de chiffrement pour détecter la présence d'un faux point d'accès. Un réseau Wi-Fi peut être chiffré ou non chiffré. Quand le réseau est chiffré, certaines informations concernant le chiffrement (*encryption cipher*, *authentication cipher* et *authentication type*) sont indiquées dans les trames de *beacon*. Les auteurs soulignent que ces informations sont déterminées par les fournisseurs et fabricants et ajoutées par le microprogramme (*firmware*) de la carte Wi-Fi. Pour ces raisons, les auteurs indiquent que ces informations ne peuvent pas être facilement copiées. Pour copier cette information, les attaquants doivent, en réalité, modifier le microprogramme de leur carte Wi-Fi et les auteurs partent du principe que cette modification est difficile. Cependant, dans des articles scientifiques et magazines spécialisés, plusieurs auteurs présentent des tutoriels pour modifier rapidement le microprogramme de plusieurs cartes Wi-Fi.(MEDIA, s. d.). Aussi, cette méthode ne fonctionne que si le réseau Wi-Fi est chiffré. Lorsque le réseau Wi-Fi est non chiffré ou public, les informations de chiffrement ne sont pas indiquées. Les réseaux Wi-Fi publics sont les réseaux Wi-Fi présents dans les aéroports, les universités, écoles publiques, les cinémas et les restaurants. Ces réseaux sont les plus vulnérables à l'attaque par faux point d'accès parce que l'attaquant n'a pas à trouver le mot de passe du réseau et peut ainsi plus facilement forcer les clients à se connecter à son faux point d'accès (AMOORDON,

Gransart et al., 2020).

Dans (C. Han et al., 2012), les auteurs proposent de comparer les informations telles que le nom du réseau, le type d'authentification et le type de chiffrement pour détecter la présence d'un faux point d'accès. Un réseau Wi-Fi peut être chiffré ou non chiffré. Quand le réseau est chiffré, certaines informations concernant le chiffrement (*encryption cipher*, *authentication cipher* et *authentication type*) sont indiquées dans les trames de *beacon*. Les auteurs soulignent que ces informations sont déterminées par les fournisseurs et fabricants et ajoutées par le microprogramme (*firmware*) de la carte Wi-Fi. Pour ces raisons, les auteurs indiquent que ces informations ne peuvent pas être facilement copiées. Pour copier cette information, les attaquants doivent, en réalité, modifier le microprogramme de leur carte Wi-Fi et les auteurs partent du principe que cette modification est difficile. Cependant, dans des articles scientifiques et magazines spécialisés, plusieurs auteurs présentent des tutoriels pour modifier rapidement le microprogramme de plusieurs cartes Wi-Fi.(Media, s. d.). Aussi, cette méthode ne fonctionne que si le réseau Wi-Fi est chiffré. Lorsque le réseau Wi-Fi est non chiffré ou public, les informations de chiffrement ne sont pas indiquées. Les réseaux Wi-Fi publics sont les réseaux Wi-Fi présents dans les aéroports, les universités, écoles publiques, les cinémas et les restaurants. Ces réseaux sont les plus vulnérables à l'attaque par faux point d'accès parce que l'attaquant n'a pas à trouver le mot de passe du réseau et peut ainsi plus facilement forcer les clients à se connecter à son faux point d'accès (Amoordon, Gransart et al., 2020).

Dans (Tang et al., 2009), les auteurs proposent de détecter le faux point d'accès en analysant les informations concernant le canal. L'information du canal n'est pas une information d'entête de la couche 2, mais une information de la couche 1 enregistrée par le pilote de la carte Wi-Fi, puis encapsulée à titre d'information, dans toutes les trames, notamment dans les trames de *beacon*. L'information que les auteurs souhaitent utiliser est la fréquence du canal sur lequel la trame a été reçue. Cette information étant déterminée et

encapsulée par le matériel à la réception, l'attaquant ne peut pas la modifier lors de la transmission. Ainsi, si cette information diffère entre deux trames de *beacon*, la présence d'un faux point d'accès peut être caractérisée. Toutefois, cette solution ne fonctionnera pas si l'attaquant a créé son point d'accès sur le même canal que le point d'accès licite. Dans ce cas précis, la carte Wi-Fi indiquera que les deux trames de *beacon* ont été reçues sur le même canal. La fréquence du canal est, ici, une information de la couche 1 récupérée par la carte Wi-Fi et encapsulée dans une trame. Nous considérons cette information comme une information de la couche 2 parce qu'elle a été, en réalité, déterminée par une carte Wi-Fi (équipement de la couche 2). Cette même information aurait pu être déterminée en utilisant, par exemple, un analyseur de spectre (équipement de la couche 1), et dans ce cas, l'information aurait été classée comme une information de la couche 1.

Comparaison d'informations dynamiques

Nous désignons les informations dynamiques, les informations qui varient dans le temps et dont les variations ne peuvent pas être facilement anticipées, prédites et, par conséquent, copiées par un attaquant.

Dans (Chumchu et al., 2011), les auteurs démontrent qu'ils arrivent à détecter la présence de faux point d'accès en analysant les différences sur le taux de transfert et de modulation. Ces informations sont déterminées et ajoutées à l'émission par le microprogramme de la carte Wi-Fi en fonction de l'état du canal Wi-Fi. Ces informations de taux de transfert et de modulation sont déterminées par l'algorithme d'adaptation de taux de transmission en fonction de l'état du canal qui peut varier avec le temps et en fonction de la position de l'équipement. Toutefois, comme le soulignent d'autres auteurs (Alotaibi et al., 2016), d'après la norme IEEE 802.11 (Wi-Fi), il existe un nombre restreint de taux de transfert et de types de modulation. De ce fait, il y a une

grande probabilité que l'algorithme de taux de transmission de la carte Wi-Fi du faux point d'accès détermine le même taux de transfert et de modulation que le point d'accès licite – surtout si le faux point d'accès est à proximité du point d'accès licite et qu'il opère sur le même canal de fréquence.

Dans (Guo et al., 2005), les auteurs démontrent qu'ils arrivent à détecter des faux points d'accès en analysant l'écart entre les numéros de séquence. Les spécifications du protocole indiquent que les trames (de gestion, de données et de contrôle) doivent être numérotées en utilisant un compteur. Le compteur appelé numéro de séquence est contrôlé et incrémenté par le microprogramme de la carte Wi-Fi. Le compteur est différent pour chaque type de trame (gestion, données ou contrôle), mais le même pour les sous-types de trame. Le numéro de séquence commence de 0 et est incrémenté de 1 jusqu'à 4095. Une fois le nombre de 4095 atteint, le numéro de séquence est réinitialisé à 0. Les auteurs démontrent qu'en fonction de leur configuration, dans une situation normale, l'écart entre deux trames de *beacon* n'est jamais plus grand que 8 et que s'il y a uniquement des trames de *beacons* (et pas d'autres trames de gestion), l'écart entre les numéros de séquences ne jamais plus petit ou plus grand que 1. L'écart entre les numéros de séquences ne peut pas être égal à 0 entre deux trames de *beacon* parce qu'il n'y a pas de retransmission de trame de *beacon*. L'écart entre les numéros de séquences peut être plus grand que 1 si le point d'accès a émis d'autres trames de gestion autre que des trames de *beacon* entre deux trames de *beacon*.

S'appuyant sur leurs observations et leur configuration, les auteurs concluent que si l'écart entre les numéros de séquences est supérieur à 8 et/ou inférieur à 1, il y a la présence d'un faux point d'accès. Ceci peut s'expliquer par le fait qu'il est difficile de copier et de suivre le numéro de séquence d'un point d'accès. Si l'attaquant n'arrive pas à synchroniser le compteur de son faux point d'accès avec celui du point d'accès licite, de grosses variations dans l'écart entre les numéros de séquences seront visibles. Même si l'attaquant arrive à synchroniser le compteur de son faux point d'accès avec celui de son

point d'accès licite, l'écart entre les numéros de séquences sera de 0 - ce qui est suffisant pour indiquer la présence d'un faux point d'accès si l'écart de 0 perdure sur de nombreuses trames. Cette méthode de détection est une des méthodes les plus efficaces pour détecter le faux point d'accès. Cependant, l'attaquant peut, en théorie, étudier l'écart entre les numéros de séquences du réseau sur une période de temps avant de créer son faux point d'accès. Il peut ainsi émettre des trames de *beacon* à des temps précis et éviter de dépasser le seuil d'écart entre les numéros de séquence observé. L'attaquant peut également utiliser des signaux de brouillage et l'envoi de trame de requêtes sondes (qui sont des trames de gestion) pour influencer le compteur du point d'accès licite.

Dans (Arackaparambil et al., 2010), les auteurs démontrent qu'ils arrivent à détecter des faux points d'accès en analysant les différences d'horodatages. Les trames sont horodatées par la carte Wi-Fi et cette information indique l'heure à laquelle la trame a été envoyée. Les auteurs indiquent qu'en analysant les différences entre horodatages, ils arrivent à distinguer entre les trames de *beacon* émises par le point d'accès licite et celles émises par le faux point d'accès. Ceci peut être expliqué par le fait que même si les deux points d'accès sont synchronisés avec le même serveur horaire, il peut toujours y avoir des différences entre les horloges. Ces différences peuvent être dues au fait que les points d'accès n'ont pas reçu la donnée de temps au même instant ou parce qu'ils n'ont pas, pour des raisons internes, pu mettre à jour leur horloge au même instant. Cette méthode est une approche cohérente, mais elle est difficile à implémenter. D'autres auteurs ont démontré que les attaquants pouvaient étudier l'horloge du point d'accès licite afin de réduire l'écart d'horloge entre le point d'accès et leur ordinateur (Bratus, Arackaparambil et al., 2011).

D'autres auteurs proposent d'analyser le nombre de *beacon* reçu sur une période de temps.

Dans (Bratus, Cornelius et al., 2008), les auteurs utilisent une méthode

inspirée de la prise d'empreinte sur paquet TCP pour créer des empreintes de plusieurs points d'accès. Les auteurs utilisent Nmap (un logiciel pour faire des audits de sécurité et la découverte de réseau) pour émettre des trames et ensuite analyser les sous-champs du champ *Frame control* de ces trames pour créer des empreintes pour chaque point d'accès. À noter que les auteurs indiquent que cette méthode fonctionne, mais ils ne précisent pas si elle fonctionne lorsqu'il y a du trafic. Aussi, c'est une méthode active qui consiste à envoyer des paquets à un certain moment pour détecter la présence d'un faux point d'accès. L'attaquant peut détecter le déclenchement du mécanisme de détection et contre-attaquer pour tenter d'influencer la prise d'empreintes ou s'endormir lorsque cela se produit.

Une analyse quantitative des trames de *beacon* reçues

Dans (Kao et al., 2014), les auteurs démontrent que des incohérences entre intervalles de trame de *beacons* peuvent être utilisées pour détecter les faux points d'accès. Dans leur expérimentation, les auteurs présument que l'attaquant a pu éliminer sa différence d'horloge, qu'il a pu contrôler l'écart entre les numéros de séquences de son point d'accès et qu'il a pu copier toutes les informations statiques se trouvant sur les trames de *beacon*. Sous ces conditions, il serait quasi impossible de détecter un faux point d'accès en se fondant uniquement sur les informations présentes sur les trames de *beacon*. En effet, toutes les informations statiques et dynamiques sont identiques. Les auteurs proposent d'analyser un grand nombre de trames de *beacons* émises sur le canal. Les trames de *beacons* doivent être envoyées à chaque 102.4 ms.

Toutefois, lorsqu'il y a un important volume de trafic, un point d'accès peut reporter l'émission des trames de *beacons* pour privilégier l'envoi de trames de données. En étudiant l'intervalle de *beacon* sur une longue période de temps, une déviation de l'intervalle de *beacon* du point d'accès pourra être détec-

tée. Si cette méthode semble cohérente, les auteurs ne détaillent pas comment l'attaquant arriverait à synchroniser l'écart entre les numéros de séquences et éliminer le décalage d'horloge de son point d'accès. Si l'attaquant arrive à faire ces manipulations, cela voudrait dire que son faux point d'accès fonctionne avec une haute précision en temps. Dans de telles conditions, l'attaquant devrait être capable de détecter des déviations de l'intervalle de *beacon* du point d'accès licite et corriger l'intervalle de son point d'accès sans que ce dernier soit détecté.

Méthodes utilisant les informations de la couche 1

Dans (Kim et al., 2012), les auteurs proposent de mesurer le *Received Signal Strength Indicator* (RSSI) des points d'accès, de le normaliser et d'utiliser des seuils pour déterminer la présence d'un faux point d'accès. Les auteurs indiquent qu'en fonction de leur configuration de test, leur méthode arrive à détecter l'attaque avec une précision de 99 % avec des faux positifs de moins de 0.1 %. Toutefois, la valeur RSSI varie très facilement en fonction du nombre de clients dans le réseau et de la distance entre le point d'accès licite et le faux point d'accès. Le Wang dans sa thèse intitulé : « *La détection de l'attaque Man-in-the-Middle en utilisant la couche physique* » (Wang et al., 2016) a proposé de détecter l'attaque *Man-in-the-Middle* en utilisant les différences de RSSI. L'auteur souligne que sa méthode fonctionne, mais elle peut être impactée par plusieurs variables (distance, le rapport signal à bruit...) et que cela peut réduire la précision de la détection.

2.6.2 Attaque par déauthentification

L'attaque par déauthentification peut être détectée sur la couche 1 et les couches supérieures (2-4) du modèle OSI (Alotaibi et al., 2016).

Méthodes utilisant les informations de la couche 2 et des couches supérieures

Comme l'attaque par faux point d'accès, l'attaque par déauthentification peut être détectée en comparant l'écart entre les numéros de séquence (Guo et al., 2005 ; Sheng et al., 2008). La trame de déauthentification est un sous-type de trame de gestion et partage le même compteur que les trames de *beacon*. Lorsque l'attaquant émet des trames de déauthentification en usurpant l'adresse MAC du point d'accès, il ne peut pas facilement copier le même numéro de séquence que le point d'accès licite (parce que c'est une information dynamique). Ainsi, en étudiant le numéro de séquence, d'importantes variations seront constatées. Ces variations peuvent être utilisées pour caractériser la présence de l'attaque par déauthentification. Ces variations ne sont pas les mêmes que pour l'attaque par faux point d'accès. Dans (Sheng et al., 2008), les auteurs proposent d'étudier le RSSI calculé sur la couche 2 et encapsulé dans les trames pour détecter la présence de trames de déauthentification. Leurs méthodes ont de bons résultats, mais le RSSI (couche 1 ou couche 2) varie en fonction de la distance entre l'attaquant et le point d'accès.

Dans (Agarwal, Biswas et al., 2013), les auteurs proposent une méthode pour détecter l'attaque par déauthentification sur les réseaux 802.11 en comparant le nombre de trames de déauthentification et le débit (calculé à partir des trames de données). Cette méthode, comme le souligne l'auteur, est facile à implémenter. Toutefois, les auteurs ne précisent pas si leur méthode a la même efficacité en l'absence de trame de données. Dans (Agarwal, Pasumarthi et al., 2016), les auteurs proposent de détecter les attaques en analysant une liste de 18 attributs extraits des couches 2, 3 et 4 du modèle OSI. Leur méthode utilise des algorithmes de classification tels que *Support Vector Machine* (SVM) et *Naïves Bayes (NB)*. Leur modèle a une précision de 98.7 % avec SVM et 95.4 % avec NB. Toutefois, leur méthode analyse cumulativement les informations présentes sur les couches 3 et 4, ce qui implique que le système de détection prenne en charge les clés de chiffrement, ce qui peut être difficile si

le chiffrement est un chiffrement RADIUS (unique pour chaque client) tel que le WPA-802.1X (WPA-EAP).(Aboba et al., 2004).Leur étude est également limitée aux attaques par déni de service et ne prennent pas en compte l'attaque par faux point d'accès

Méthodes utilisant les informations de la couche 1

Dans (Villain, Deniau, Fleury et al., 2019), les auteurs proposent d'utiliser les algorithmes d'apprentissage automatiques pour détecter les attaques par déauthentification et par brouillage dans les réseaux IEEE 802.11. Les auteurs capturent des puissances maximales des signaux qui sont transmis sur une période de temps en utilisant les équipements de laboratoires tels que des analyseurs de spectre. Ils utilisent ensuite des algorithmes de classification pour essayer de détecter les situations avec l'attaque par déauthentification ou l'attaque par brouillage. Les auteurs indiquent qu'ils arrivent à détecter les attaques avec haute précision (avec taux d'erreur de seulement 14 %). Toutefois, les auteurs ne prennent pas en compte plusieurs niveaux de trafic ni l'attaque par faux point d'accès.

2.6.3 L'attaque par brouillage

Dans la littérature scientifique, l'attaque par brouillage est détectée en utilisant des méthodes analysant les informations de la couche 2 ou les caractéristiques des signaux transmis sur la couche 1 du modèle 0SI.

Méthodes utilisant les informations de la couche 2

En utilisant les informations de la couche 2, des auteurs ont proposé des méthodes de détection de l'attaque par brouillage en analysant des trames de données ou de contrôle. L'utilisation des informations sur les trames de gestion, elle, n'est pas clairement indiquée. Ces méthodes peuvent être classées en deux catégories : des méthodes utilisant des algorithmes d'apprentissage automatique et ceux qui n'utilisent pas d'algorithmes d'apprentissage automatique.

Méthodes n'utilisant pas d'algorithmes d'apprentissage automatique

Dans (Cheng et al., 2017), les auteurs ont développé et utilisé un modèle de séries temporelles capable de détecter l'attaque par brouillage. Ils proposent d'analyser l'intervalle entre deux trames reçues, le nombre d'octets reçus dans la trame, le retard d'aller-retour et le rapport signal à interférence pour détecter l'attaque par brouillage. Ils ont également considéré le débit et l'intervalle entre les trames. La détection fondée sur ces paramètres est efficace. Toutefois, cette méthode ne fonctionne qu'en présence de trames de données et est inopérante lorsque les équipements sont inactifs et ne transmettent pas de données. Aussi, cette recherche a été menée uniquement en utilisant les outils de simulation. Dans (Reyes et al., 2013), les auteurs calculent le taux de trames endommagées, les informations sur la disponibilité du canal, le rapport de trames envoyées sur les trames reçues et la puissance du signal reçu, informations extraites des trames de données et de contrôle pour détecter l'attaque par brouillage. Les auteurs présentent des résultats intéressants pour des expérimentations en simulation (en utilisant Matlab et l'outil Fuzzy logic) et en laboratoire. Cette méthode est intéressante, mais également inopérante en cas d'absence de trames de données et de contrôle.

Méthodes implémentées avec l'utilisation d'algorithme d'apprentissage automatique

Dans (Arjoune et al., 2020), les auteurs proposent une méthode pour détecter l'attaque par brouillage en utilisant des algorithmes d'apprentissage automatique. Les attributs sont le taux de trames endommagées, le taux de trames correctement délivrées, la puissance du signal et la disponibilité du canal. Les résultats présentés par les auteurs montrent que cette méthode arrive à détecter l'attaque par brouillage avec haute précision. Avec des réseaux de neurones, leur modèle arrive à détecter l'attaque par brouillage avec une précision de 96.4% et avec l'algorithme de *Random Forest* (avec 100 estimateurs), leur modèle arrive à détecter l'attaque par brouillage avec une précision de 96.6 %. Même si leur recherche est conçue pour la détection de l'attaque par brouillage sur les réseaux 5G, des attributs similaires peuvent être trouvés sur les trames IEEE 802.11. Toutefois, comme ces informations sont extraites de trames de données, cette méthode est inopérante en absence de ces trames. Il faut également noter que les auteurs n'ont malheureusement pas clairement détaillé leur banc de test et leur procédure d'analyse des trames.

Dans (Sufyan et al., 2013), les auteurs proposent une méthode multimodale qui permet de détecter plusieurs types d'attaque par brouillage (réactive, constant, aléatoire et intelligent) en analysant la corrélation entre trois attributs : le taux de trames correctement délivrées, les variations dans la puissance du signal et la largeur d'impulsion du signal reçu. La variation de la puissance du signal et la largeur d'impulsion du signal reçu sont obtenues en utilisant une radio logicielle. Le taux de trames correctement délivrées est calculé en utilisant des trames de données - la méthode est donc inopérante en l'absence de ces trames. Dans (Puñal et al., 2014), les auteurs proposent de détecter l'attaque par brouillage sur les réseaux IEEE 802.11 (Wi-Fi) en analysant des informations sur l'entête de la couche 2 sur les trames de gestion, de contrôle et de données. Les auteurs analysent des attributs comme le temps d'inactivité, la puissance maximale du signal et le taux de trames correctement délivrées.

Les auteurs analysent également le niveau de bruit et l'occupation du canal en utilisant des cartes Wi-Fi modifiées pour détecter le signal de brouillage même en absence de trame de données. Les auteurs ont eu de très bons résultats de détection sur des bancs de test à l'intérieur, à l'extérieur et avec différentes conditions de communication entre le point d'accès et les clients.

Cette méthode permet de détecter l'attaque par brouillage en présence et en absence de trafic utilisateur, notamment grâce aux mesures du niveau de bruit et du taux d'occupation du canal avec les cartes Wi-Fi modifiées. Cependant, les auteurs indiquent qu'ils utilisent une plateforme logicielle multicouche pour rendre cette méthode de détection opérante. Plus particulièrement, ils précisent que le taux de trames correctement délivrées est en partie calculé avec des informations de la couche application et que chaque équipement Wi-Fi du réseau doit connaitre le nombre de clients environnants et un taux prédéfini pour générer des trames de sonde. Cette méthode est, par conséquent, en partie fondée sur la nécessité pour les équipements d'avoir une connaissance du réseau et des autres clients environnants. Cette méthode n'est donc pas facilement généralisable. Il semblerait également qu'elle combine la détection passive et la détection active. L'utilisation des trames de gestion n'est pas clairement indiquée dans ces travaux. Enfin, cette méthode est partiellement fondée sur des informations de la couche application qui sont indisponibles lorsque le réseau est chiffré - à moins que la plateforme soit connectée au réseau et qu'elle ait accès aux clés de chiffrement des utilisateurs.

Méthodes utilisant les données de la couche 1

Dans (Villain, Deniau, Gransart et al., 2021), les auteurs proposent d'utiliser les algorithmes d'apprentissage automatiques pour détecter l'attaque par brouillage avec faible intensité. Sur la couche 1, la détection des attaques par brouillage avec faible intensité est difficile à mettre en œuvre et les au-

teurs ont eu des résultats intéressants (plus de 94 % de détection de précision avec certains de leurs *clusters*). Les mêmes auteurs, dans des travaux déjà cités (Villain, Deniau, Fleury et al., 2019), ont également pu détecter l'attaque par brouillage avec un taux d'erreur de moins de 14 %. Dans (Bhojani et al., 2016), les auteurs proposent des méthodes pour détecter l'attaque par brouillage en utilisant une radio logicielle et en analysant un indicateur de synchronisation, une itération et un rapport adaptatif de signal à bruit plus interférence. Les auteurs ont pu déterminer des seuils qui permettent de caractériser la présence de l'attaque par brouillage.

Outre les travaux scientifiques, les groupes de travail de la norme IEEE 802.11 et certaines entreprises ont proposé respectivement des amendements et des solutions industrielles pour endiguer ou détecter ces trois attaques.

2.7 Amendements protocolaires et solutions industrielles existantes

L'introduction de l'authentification RADIUS à partir de la troisième génération de Wi-Fi permet de protéger les réseaux contre les attaques par faux points d'accès. L'authentification RADIUS implique en réalité une procédure de double authentification dans les réseaux sans fil. Cette procédure oblige le terminal à s'authentifier auprès du point d'accès, et le point d'accès à s'authentifier auprès du terminal. Pour ce faire, un certificat doit être préalablement installé sur le terminal du client. Cette solution permet d'éviter que le terminal ne se connecte à un faux point d'accès par erreur ou par force. Toutefois, la double authentification n'est pas adaptée aux réseaux Wi-Fi publics, car en principe, les utilisateurs ne prévoient pas de télécharger un certificat avant d'utiliser ces réseaux. De plus, ces réseaux sont souvent utilisés de manière imprévue.

Les amendements de la norme posent une difficulté qui réside dans le fait qu'ils impliquent souvent une mise à jour logicielle ou matérielle. Ces mises à jour ne sont pas toujours effectuées, soit par négligence, soit parce que dans certains cas, il faut changer de terminal. Ainsi, même si un amendement peut permettre d'endiguer une attaque, cette protection est inégalitaire parce que les anciens équipements qui n'ont pas bénéficié de cet amendement seront toujours vulnérables. Certaines recherches scientifiques proposent des améliorations de la norme qui sont ensuite reprises intégralement ou en partie par les groupes de travail de la norme. Par exemple, dans (Srilasak et al., 2008), les auteurs proposent une méthode de détection de l'attaque par faux point d'accès en modifiant la phase authentification. Cette approche a les mêmes limitations que l'amendement précité, notamment le fait qu'elle ne permet pas de protéger les équipements qui ne peuvent pas bénéficier des mises à jour.

Les amendements de la norme posent une difficulté qui réside dans le fait qu'ils impliquent souvent une mise à jour logicielle ou matérielle. Toutefois, ces mises à jour ne sont pas toujours effectuées, soit par négligence, soit parce que dans certains cas, il est nécessaire de changer de terminal. Ainsi, même si un amendement peut permettre d'endiguer une attaque, cette protection est inégalitaire, car les anciens équipements qui n'ont pas bénéficié de cet amendement resteront vulnérables. Certaines recherches scientifiques proposent des améliorations de la norme qui sont ensuite reprises intégralement ou en partie par les groupes de travail de la norme. Par exemple, dans (Srilasak et al., 2008), les auteurs proposent une méthode de détection de l'attaque par faux point d'accès en modifiant la phase d'authentification. Bien que cette méthode puisse être reprise, elle reste soumise aux mêmes limitations que les amendements en général, c'est-à-dire qu'elle ne permet pas de protéger les équipements qui ne peuvent pas être mis à jour.

2.7.1 Les solutions industrielles

Les entreprises proposent également des solutions pour détecter l'attaque par faux point d'accès en se fondant sur les travaux de la communauté scientifique. Par exemple, Huawei propose un système de détection d'intrusion pour les réseaux Wi-Fi qui compare les caractéristiques des trames de *beacon* avec une liste blanche ou noire pour détecter l'attaque (Huawei, 2022). Cette méthode en ligne nécessite que le système de détection d'intrusion soit connecté au réseau filaire. Cependant, il n'est pas clairement indiqué si ce système peut détecter les faux points d'accès copiant toutes les informations statiques des trames de *beacon* du point d'accès licite. Des antivirus tels que Norton proposent également un outil pour détecter les faux points d'accès dans leur suite logicielle de sécurité (IngramMicroCloudChannel, 2022). Le fonctionnement technique de cet outil n'est pas détaillé, mais il semble qu'il utilise la comparaison statique des trames de *beacon* pour détecter les attaques. En cas d'attaque détectée, l'application Norton Wi-Fi Privacy propose à l'utilisateur d'utiliser un VPN pour empêcher le vol d'informations (Tech, 2016). Cependant, l'utilisation d'un VPN peut ralentir la connexion Internet et engendrer des coûts supplémentaires.

Par ailleurs, l'utilisation d'une connexion VPN peut être bloquée sur certains réseaux Wi-Fi et ne fonctionne pas toujours de manière optimale sur les téléphones mobiles. Dans le même ordre d'idées, Avast WiFi Inspector permet également de détecter des vulnérabilités telles que la présence de faux points d'accès sur un réseau Wi-Fi (Salmi, 2017). Le fonctionnement technique de cet outil n'est pas détaillé, mais il semblerait qu'il soit similaire à celui de Norton. Enfin, l'entreprise Air-Magnet (Motorola, 2009) propose d'utiliser un système de détection d'intrusion sans-fil avec des capteurs déployés dans le réseau. Les capteurs collectent des données des couches 1 et 2 pour détecter l'attaque par faux point d'accès en déportant le calcul sur un serveur. Les informations collectées comprennent notamment l'adresse MAC, la puissance

du signal et les trames de contrôle du point d'accès. Cette méthode est cependant coûteuse à mettre en œuvre, car elle nécessite l'utilisation d'un analyseur de spectre vendu par Air-Magnet, qui coûte environ $ 3 000. De plus, cette solution n'est pas adaptée à une utilisation mobile.

2.8 Conclusion

Dans ce chapitre, nous avons brièvement rappelé le fonctionnement des ordinateurs ainsi que la mise en réseau de ces derniers en utilisant différents types de liens (sans-fil et filaire) et de connexions (physique et logique). Ensuite, nous avons détaillé le protocole IEEE 802.11, en passant en revue sa couche physique, sa couche liaison de données et ses menaces. Nous nous sommes ensuite concentrés sur trois attaques : les attaques par faux point d'accès, par brouillage et par déauthentification. Nous avons décrit les différentes formes de ces attaques avant de lister des méthodes et des solutions de détection existantes dans la littérature scientifique et l'état de la technique. Cette revue nous a permis de constater que les trois attaques sont connues, mais qu'elles sont souvent détectées de manière isolée les unes des autres, alors qu'elles peuvent être liées et combinées. De plus, ces attaques sont détectées en utilisant seulement un ou deux indicateurs. Enfin, certaines des méthodes proposées sont difficiles à mettre en œuvre, coûteuses et ne fonctionnent pas en l'absence de trames de données.

En résumé, il n'existe pas actuellement d'approche holistique, multi-indicateurs et peu coûteuse permettant de détecter les attaques, qu'elles soient réalisées séparément ou combinées – et ce même en l'absence d'envoi de données. Dans le cadre de cette thèse, nous avons travaillé sur le développement d'une telle approche et avons reproduit les attaques en laboratoire pour collecter des données. Les configurations de nos bancs de test, les variations de réglages ainsi que les matériels que nous avons utilisés sont détaillés dans le

prochain chapitre.

Chapitre 3

Expérimentations : Matériel et configurations

Dans le chapitre précédent, nous avons abordé les différents types d'attaques possibles sur les réseaux sans-fil. Nous avons ensuite examiné en détail trois types d'attaques spécifiques : les attaques par brouillage, les attaques par faux point d'accès et les attaques par déauthentification. Après avoir énuméré et détaillé les différentes formes de ces attaques, nous avons présenté les méthodes de détection actuellement disponibles dans la littérature scientifique et dans l'état de la technique. Nous avons constaté qu'il n'existe pas d'approche holistique pour détecter ces trois types d'attaques, qui peuvent pourtant être combinés. De plus, ces attaques sont souvent détectées en utilisant seulement un ou deux indicateurs, ce qui n'est pas optimal, car un attaquant peut facilement les contourner.

Afin de détecter de manière plus efficace les attaques, notre proposition consiste à adopter une approche capable de les détecter lorsqu'elles sont réalisées de manière isolée ou combinée, en utilisant plusieurs indicateurs. Nous souhaitons que cette approche de détection soit appliquée aux trames de gestion, en particulier les trames de *beacon* qui sont émises de manière régulière contrairement aux trames de données. Cette approche est basée sur la détection d'anomalies, qui consiste à comparer des situations d'attaques avec une situation de référence sans attaque. Pour mettre en œuvre cette approche, il

est nécessaire de reproduire les attaques en laboratoire, de capturer les données, d'analyser les captures pour identifier les anomalies (par exemple : des variations importantes dans une valeur lors d'une situation avec attaque, mais pas dans une situation normale), puis d'utiliser ces variations pour décider de la présence ou non d'une attaque.

Ce chapitre présente les bancs de test utilisés pour reproduire les différentes configurations, à savoir la situation normale ainsi que les situations avec les attaques par faux point d'accès, brouillage et déauthentification. Pour chaque configuration, nous détaillons le banc de test utilisé, le type/forme d'attaque sélectionné et les outils utilisés pour la réalisation de l'expérimentation. Les différentes configurations ont été reproduites dans un laboratoire simulant des conditions réelles. La combinaison des attaques sera abordée dans le chapitre 5.

3.1 La configuration de référence : état normal

La situation normale, qui correspond à situation avec absence d'attaque (voir Figure 3.1), est considérée comme la configuration de référence. Elle sert de base pour positionner les principaux éléments du réseau. Dans les autres configurations avec attaque, les équipements nécessaires à la réalisation des attaques sont ajoutés sans altérer cette configuration de référence. Il est donc important de décrire précisément tous les composants de cette configuration. Dans cette situation, une communication Wi-Fi sans perturbation intentionnelle a lieu entre un client et un serveur. Le serveur et le client sont connectés au même réseau et communiquent via un point d'accès Wi-Fi.

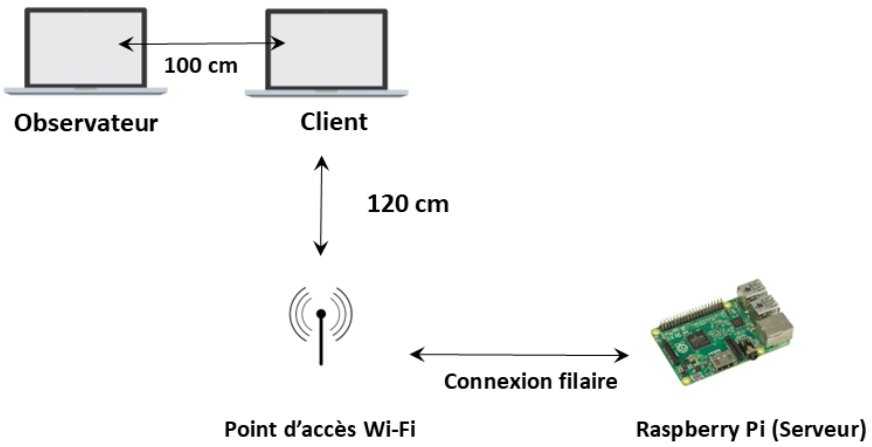

FIGURE 3.1 – Situation normale

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay)

3.1.1 Le réseau Wi-Fi : le réseau reliant tous les équipements

Le point d'accès assure la connexion des équipements filaires et sans-fil. Il dispose de deux ports Ethernet pour les équipements filaires et peut générer deux réseaux Wi-Fi simultanément sur les deux bandes Wi-Fi (2.4 GHz et 5

GHz) pour les équipements sans-fil. Le point d'accès n'est pas connecté à Internet et le serveur ainsi que le client sont connectés sur le même réseau local. Nous avons choisi cette configuration dans un réseau local pour protéger la communication de toutes fluctuations ou perturbations qu'elles pourraient subir si elles étaient relayées par plusieurs intermédiaires avant d'arriver au destinataire. Concernant, le canal utilisé, les équipements réseaux Wi-Fi fonctionnent sur le canal 13.

Pour choisir ce canal, nous avons lancé des tests. Ces tests ont révélé que le canal 13 était le moins occupé, ce qui nous a permis de réaliser nos expérimentations sans affecter d'autres équipements que ceux spécifiquement dédiés à l'expérimentation. De plus, nous avons choisi ce canal libre afin de minimiser les interférences non intentionnelles sur le réseau Wi-Fi. En effet, un canal libre est caractérisé par une faible occupation, ce qui réduit le risque d'interférences avec d'autres équipements.

Le point d'accès Wi-Fi est configuré en mode infrastructure pour assurer l'intermédiation entre les différents clients du réseau. Nous avons choisi ce mode parce que nous souhaitons précisément détecter les attaques sur les réseaux Wi-Fi en mode infrastructure. Ce mode est le plus couramment utilisé dans les réseaux sans-fil Wi-Fi, contrairement au mode ad hoc qui est majoritairement utilisé par les objets connectés (Reina et al., 2013).

Concernant l'amendement de la norme IEEE 802.11 utilisée, le point d'accès et équipements sont compatibles avec les amendements IEEE 802.11 b/g et n. Les équipements d'un réseau Wi-Fi peut utiliser plusieurs amendements de la norme IEEE 802.11 en fonction de différents critères tels que le rapport signal à bruit du canal, les utilisateurs ou les types de trames à envoyer. Le choix de l'amendement est déterminé par la sous-couche PLCP de la couche physique. D'après nos observations, les trames de données et de contrôle sont envoyées en utilisant soit la norme IEEE 802.11g ou la norme IEEE 802.11n, tandis que les trames de gestion sont toujours envoyées en utilisant la norme IEEE 802.11b.

Comme indiqué sur la Figure 3.1, nous avons relié le Raspberry Pi (sur lequel est exécuté le serveur de données) au point d'accès en utilisant une connexion filaire, c'est-à-dire, via un port Ethernet avec un câble RJ45. Le serveur est relié en filaire parce que nous souhaitons étudier l'impact des trois attaques uniquement sur la transmission sans-fil entre le client et le point d'accès. En reliant le serveur en filaire au point d'accès, nous le protégeons contre les futures attaques sans-fil que nous allons réaliser dans les situations suivantes. Concernant l'envoi de trames de *beacons*, qui doivent être envoyées de manière régulière, nous avons laissé la valeur par défaut, qui correspond à un envoi de trame de *beacon* toutes les 102,4 ms. Cette valeur par défaut est recommandée par la norme IEEE 802.11 et est la plus largement utilisée dans les réseaux Wi-Fi.

3.1.2 Génération et transmission de données via l'application Client-Serveur Iperf

L'application Iperf est un logiciel qui permet un transfert de données entre deux ordinateurs et est généralement utilisée pour effectuer des tests de performance dans un réseau informatique. Le transfert de données peut se faire avec plusieurs niveaux de débit en utilisant les deux protocoles de transport les plus connus : TCP et UDP. Lorsqu'un niveau de débit est spécifié, le logiciel envoie de la donnée de manière constante de sorte à se rapprocher autant que possible du débit indiqué. L'application se compose de deux composants : un client iPerf et un serveur iPerf. Pour simuler un transfert de données dans la situation normale, nous avons lancé un client iPerf sur l'ordinateur Client et un serveur iPerf sur le Raspberry Pi tyê B modèle 2. Le transfert de données a été réalisé du client Iperf vers le serveur Iperf.

Le Raspberry Pi type B modèle 2 est un ordinateur de petite taille conçu par des professeurs du département informatique de l'université de Cambridge.

Il est équipé d'un processeur de type ARM et fonctionne sous un système d'exploitation Linux. Il possède un port Ethernet pour une communication filaire et une interface USB sur laquelle un adaptateur Wi-Fi peut être connecté (LTD, 2015). Comme tout ordinateur relié à un réseau informatique, le Raspberry Pi peut être contrôlé à distance en utilisant le logiciel Secure Shell (SSH). Nous avons installé un serveur iPerf sur le Raspberry Pi pour le transfert de données. Le Raspberry Pi est relié au point d'accès à l'aide d'un câble RJ45. Nous avons choisi le Raspberry Pi car il dispose de suffisamment de puissance pour gérer ce type de transfert de données. Sa petite taille permet également de ne pas encombrer la configuration et de faciliter la visibilité et la malléabilité de la configuration.

L'ordinateur appelé "Client" sur la Figure 3.1 remplit deux rôles de client. D'une part, il se connecte en tant que client au point d'accès Wi-Fi et d'autre part, il exécute également le client iPerf pour la communication avec le serveur iPerf lors de la transmission de données. L'ordinateur client est connecté au point d'accès via une liaison sans fil Wi-Fi sur le canal 13 de la bande 2.4 GHz et est placé à une distance de 120 cm du point d'accès Wi-Fi. La carte Wi-Fi de l'ordinateur client reçoit les transmissions de données envoyées par le serveur iPerf via le point d'accès Wi-Fi et, après traitement, les données sont envoyées à l'application client iPerf. L'ordinateur client est placé à une distance de 100 cm du point d'accès pour garantir une meilleure couverture du réseau Wi-Fi, car la qualité de la liaison Wi-Fi diminue avec la distance. Nous avons choisi de travailler sur de courtes distances afin d'étudier l'impact des attaques lorsque le rapport signal à interférence des transmissions est élevé. Si le rapport signal à interférence des transmissions est trop faible, les transmissions seront plus facilement perturbées, en particulier lorsqu'il y a une attaque de brouillage. En limitant notre expérimentation à de courtes distances, nous prenons également des précautions pour éviter de perturber les autres réseaux et équipements lors de nos manipulations.

L'outil iPerf ne peut pas simuler un trafic réel qui varie continuellement en

fonction du protocole (TCP et UDP) ainsi que du nombre et des besoins des clients. Pour notre étude, nous avons choisi de nous concentrer sur des niveaux constants de transfert de données TCP, en considérant trois niveaux de débit : faible (1 Mb/s), moyen (50 Mb/s) et intense (100 Mb/s). Ces niveaux suffisent pour prendre en compte les nuances d'impact des attaques lorsque différents niveaux de trafic sont présents. Nous avons choisi de considérer uniquement le transfert de données TCP car cela est le plus représentatif du trafic des utilisateurs. Le serveur iPerf, qui est exécuté sur le Raspberry Pi, est lancé et contrôlé à distance depuis l'ordinateur client en utilisant les commandes dans le listing 1 (voir Annexe).

3.1.3 L'observateur : le superviseur de réseau

L'Observateur est un ordinateur portable de marque HP qui fonctionne sous le système d'exploitation Kali Linux. Nous avons connecté une carte Wi-Fi externe Riverbed AirPcap (AirPcap, s. d.) sur cet ordinateur. Cette carte possède une interface USB et peut fonctionner sur les cinq modes Wi-Fi, à savoir le mode capture (appelé *monitor* en anglais), le mode point d'accès (appelé *master* en anglais), le mode client (appelé *managed* en anglais), le mode Device to Device (appelé *Adhoc* en anglais) et le mode répéteur (appelé *repeater* en anglais). Nous avons choisi cette carte en raison de sa compatibilité avec tous les modes et les deux bandes Wi-Fi (2.4 GHz et 5 GHz). Bien que la carte Wi-Fi interne de l'ordinateur puisse être utilisée, elle était moins performante en mode monitor et provoquait des interruptions. Pour ces raisons, nous avons préféré une carte Wi-Fi externe.

La carte Wi-Fi dispose également de deux entrées pour des antennes externes, mais nous avons décidé d'utiliser seulement les antennes internes dans notre configuration intérieure à courte distance. Nous avons constaté qu'il n'y avait pas d'amélioration significative du rapport signal à interférence lorsque

les deux antennes externes étaient connectées. L'objectif de l'observateur est de capturer toutes les transmissions sur le canal 13 de la bande 2.4 GHz pendant deux minutes. À terme, nous prévoyons de remplacer l'observateur par un système de détection d'intrusion (SDI). Nous avons configuré la carte Wi-Fi Pcap en mode capture sur le canal 13 pour capturer les trames émises par le client et le point d'accès, en utilisant les commandes présentées dans le listing 2 (voir Annexe).

Nous avons utilisé le logiciel libre Wireshark pour récupérer les captures et analyser les transmissions (Siswanto et al., 2019). Ce logiciel permet de récupérer les transmissions capturées par la carte Wi-Fi sur le canal sélectionné pour ensuite les sauvegarder et les analyser. Wireshark est capable d'analyser certaines informations de la couche 1 du modèle OSI (comme la puissance du signal, le canal utilisé et la modulation utilisée), qui sont mesurées, transmises et encapsulées dans un entête radioTap par la carte Wi-Fi. Le temps de réception des trames par le logiciel Wireshark est également renseigné et disponible. De plus, Wireshark permet de faire des captures automatiques en fonction d'une minuterie ou de la taille de la capture souhaitée. En raison de ses fonctionnalités, nous avons choisi ce logiciel pour nos mesures. Nous avons coché le mode *monitor*, indiqué l'interface de la carte Pcap et indiqué une minuterie de capture de 120 secondes dans les paramètres de Wireshark.

3.1.4 Variation du débit du trafic

Nous avons reproduit la situation normale avec quatre variations (voir tableau 3.1). Une variation avec trafic faible (1 Mb/s), avec trafic moyen (50 Mb/s), avec trafic intense (100 Mb/s) et une variation sans trafic entre le client et le serveur. Pour chaque variation, l'observateur a effectué une capture de deux minutes. Concernant la situation sans trafic, bien qu'il n'y ait pas de transmission de données par le serveur iPerf, d'autres transmissions telles que

l'envoi de trames de *beacon* par le point d'accès sont toujours émises (en présence ou absence de trafic).

Variations flux de données
Pas de trafic
Trafic faible - 1 Mb/s
Trafic Moyen - 50 Mb/s
Trafic Intense - 100 Mb/s

TABLE 3.1 – Variations du flux de données

Source: traitement auteur

Nous avons utilisé les commandes, indiquées dans le listing 3 (voir Annexe), pour modifier le débit du trafic transmis par le client iPerf. Après les captures de la situation normale et de ces variations, nous avons mis en place les équipements pour effectuer l'attaque par brouillage et perturber la communication entre le client et le serveur.

3.2 La situation avec l'attaque par brouillage

Dans la situation avec attaque par brouillage, nous avons ajouté un système de génération et d'émission du signal de brouillage (voir Figure 3.2) pour perturber les transmissions du réseau Wi-Fi. Le signal de brouillage est émis par une antenne directionnelle orientée vers l'ordinateur client, à une distance d'environ 100 cm avec un angle d'approximativement 45 degrés. Avant d'émettre le signal de brouillage, nous avons choisi le type de signal, implémenté et visualisé celui-ci.

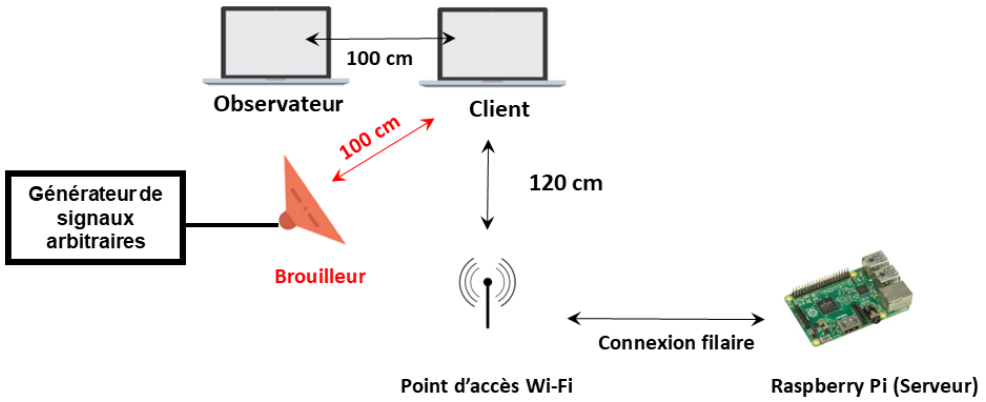

FIGURE 3.2 – La situation avec une attaque par brouillage

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay), icône antenne directionnelle (Smashicons, tirée de flaticon.com, license Flaticon)

3.2.1 Choix du signal de brouillage

Comme indiqué dans le chapitre 2, il existe plusieurs types de signaux de brouillage. Pour déterminer quel signal de brouillage utiliser pour perturber la communication Wi-Fi, nous avons commencé par identifier le type de signal utilisé dans les brouilleurs clé en main vendus dans le commerce. En effet, comme nous souhaitons nous rapprocher au plus près des outils qui sont utilisés par les attaquants, cette approche nous semblait la plus cohérente. Il s'est avéré que c'est le signal de brouillage à balayage de fréquences. Ce type de signal balaie continuellement une bande de fréquences en un temps donné (appelé temps de balayage ou temps de cycle). De plus, ce type de signal de brouillage est facilement implémentable par les attaquants pour une émission à l'aide d'une radio logicielle. Par conséquent, nous avons décidé d'utiliser ce type de signal.

En ce qui concerne les paramètres, un signal de brouillage dispose d'un temps de balayage, d'une fréquence de début de balayage, d'une fréquence de fin de balayage, d'une fréquence d'échantillonnage, d'un nombre de cycles et d'une durée de la première pente. En nous inspirant des brouilleurs vendus dans le commerce, nous avons remarqué que ces dispositifs peuvent brouiller

plusieurs bandes de fréquences correspondant à différents protocoles sans fil simultanément. Pour reproduire ce comportement, nous avons choisi une fréquence de début de balayage et une fréquence de fin de balayage qui couvrent toute la bande 2.4 GHz du protocole Wi-Fi. Nous avons alors mis La plage de fréquence de balayage du signal de brouillage à 100 MHz, ce qui permet au signal de brouillage de couvrir toutes les fréquences de 2.4 GHz à 2.5 GHz.

Le nombre de cycles du signal de brouillage est fixé à deux, mais ce signal sera émis de manière continue par un générateur de signal. La fréquence d'échantillonnage du signal est fixée à 10 Gs/s et la durée de la première pente est de 0,95 fois la longueur du cycle. En ce qui concerne le temps de balayage, c'est-à-dire dans notre cas le temps nécessaire pour que le signal balaie les 100 MHz de bande de fréquences de 2.4 GHz à 2.5 GHz, nous avons choisi une valeur de 10 µs. Pour déterminer cette valeur, nous nous sommes appuyés sur les travaux d'autres chercheurs (Romero, 2017), qui ont mené des essais de susceptibilité des communications Wi-Fi face à des signaux de brouillage avec différentes valeurs de temps de balayage allant de 0,64 µs à 50 µs. Ces travaux ont montré que l'impact du signal de brouillage est le plus important avec un temps de balayage de 10 µs, d'où notre choix.

3.2.2 L'implémentation du signal de brouillage

En nous fondant sur l'expression mathématique du signal de brouillage (voir équation 2.1), nous avons implémenté ce signal en utilisant MATLAB, un langage de programmation propriétaire multiparadigme et un environnement informatique numérique développé par MathWorks (Matlab, 2012). MATLAB permet un haut degré d'abstraction et est idéal pour l'implémentation d'équations ou de fonctions mathématiques. Nous avons donc choisi d'utiliser ce langage pour l'implémentation du signal de brouillage.

3.2.3 La visualisation et émission du signal de brouillage

Une fois le signal de brouillage implémenté, nous l'avons transféré sur un générateur de signal arbitraire, le Tektronix AWG70001A, pour le visualiser avant de l'émettre et perturber la communication. Le Tektronix AWG70001A est un générateur de signal arbitraire qui peut prendre en charge une grande variété d'exigences strictes en matière de génération de signal. Il peut générer des signaux de bande de base, infrarouge (IF) et radiofréquence (RF), avec une fréquence d'échantillonnage maximale de 50 GS/s. Il peut atteindre une plage dynamique de -75 dBv et une fréquence de jusqu'à 20 GHz (*Tektronix AWG70001A* s. d.).

Nous avons connecté une antenne directionnelle à cornet de guidage à double crête, le modèle SAS-571, sur le canal 1 du générateur de signal. Cette antenne fonctionne sur une plage de fréquence de 700 MHz à 18 GHz, ce qui la rend capable d'émettre sur la bande 2.4 GHz. Nous avons choisi cette antenne pour éviter que le signal de brouillage ne perturbe le fonctionnement d'autres réseaux Wi-Fi dans le laboratoire. En effet, le signal de brouillage couvre toute la bande 2.4 GHz, et l'utilisation d'une antenne omnidirectionnelle aurait affecté les autres réseaux opérant sur la même bande. Sur le générateur de signal, nous avons paramétré la diffusion continue du signal sur le canal 1 de l'équipement, avec un taux d'échantillonnage de 10 Gs/s pour correspondre au taux d'échantillonnage défini lors de l'implémentation du signal sous Matlab.

Avant de placer l'antenne directionnelle à 100 cm du client pour perturber les transmissions du point d'accès, nous avons visualisé le signal en utilisant un analyseur de spectre, le Keysight N9030A-526 PXA (Technologies, n.d.[a]). Le Keysight N9030A-526 PXA est un analyseur de spectre de haute performance. Il peut analyser des signaux de fréquence jusqu'à 50 GHz. Le signal de brouillage couvrant la bande de 2.4 GHz, cet équipement est adéquat pour visualiser ce signal. Nous avons ensuite relié une antenne directionnelle à l'analyseur de spectre. Sur une représentation de fréquence versus temps obtenue

après un calcul de transformation de Fourier rapide (FFT ou *Fast Fourier Transform*), nous constatons que le signal de brouillage balaie bien 100 MHz de bande en 10 μs (voir un exemple sur la Figure 3.3).

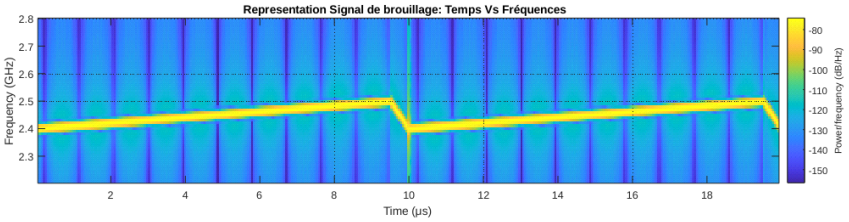

FIGURE 3.3 – Exemple de représentation du signal de brouillage

Source: Virginie Deniau, co-directrice de cette thèse, communication écrite

L'antenne émettant le signal de brouillage est positionnée à 100 cm de distance. Nous faisons varier l'atténuation du signal de brouillage (voir ci-dessous) pour simuler virtuellement une distance plus éloignée de l'antenne. En atténuant le signal de brouillage, nous diminuons le rapport signal sur bruit, ce qui peut simuler un signal de brouillage provenant de plus loin. La position et la configuration des autres équipements restent inchangées, y compris l'ordinateur client, l'observateur et le Raspberry Pi, tels que dans la situation normale. Nous faisons également varier le débit de transfert de données comme dans la situation normale. Mais cette configuration, nous faisons varier également la puissance du signal de brouillage.

3.2.4 Les variations du signal de brouillage

Pour ajuster la puissance du signal, nous atténuons le signal de brouillage en utilisant un atténuateur variable de signal, le Keysight J7211A (TECHNOLOGIES, n.d.[b]). L'atténuateur prend en entrée un signal et peut atténuer le signal en fonction d'une valeur discrète ou aléatoire. Les variations dans cette configuration sont résumées dans le tableau 3.2 :

Après les captures de la situation avec l'attaque par brouillage, nous avons

Variations	valeur
Trafic	sans trafic à trafic intense
Atténuation de la puissance de brouillage	30 db à 0 db

TABLE 3.2 – Résumé des variations de l'atténuation du signal

Source: traitement auteur

retiré les équipements utilisés pour générer le signal de brouillage avant de mettre en place les équipements nécessaires pour l'attaque par faux point d'accès.

3.3 La situation avec l'attaque par faux point d'accès

Dans la situation avec attaque par faux point d'accès, nous ajoutons un troisième ordinateur avec une carte Wi-Fi AirPcap. Sur cet ordinateur, nous créons un faux point d'accès que nous avons ensuite positionné à 120 cm du milieu entre le client et le point d'accès licite.

3.3.1 Le choix du type de faux point d'accès et sa mise en oeuvre

Concernant, le type de faux point d'accès, comme indiqué dans le chapitre 2, nous avons constaté que le faux point d'accès copiant toutes les informations statiques du point d'accès licite (sans redirection vers celui-ci) était le plus difficile à détecter. Pour cette raison, nous avons choisi de créer ce type de faux point d'accès. En plus de copier toutes les informations statiques, le faux point d'accès émet également des trames de *beacon* au même intervalle que le point

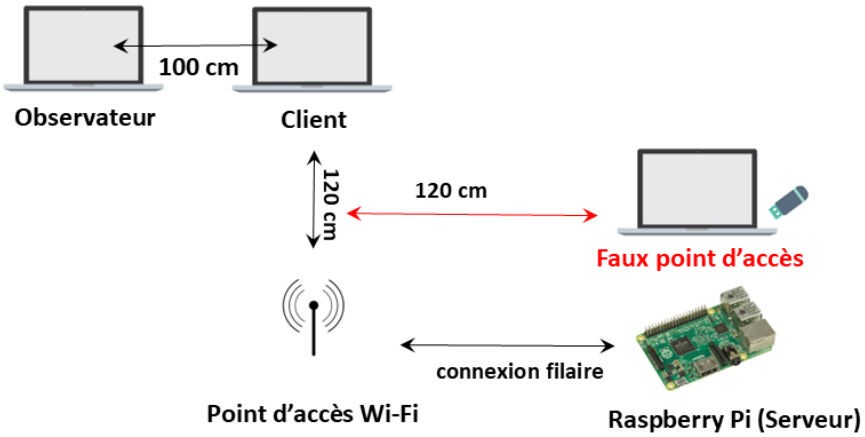

FIGURE 3.4 – Faux point d'accès

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône Internet (conçue par Freepik, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay), icône usb (conçue par rawpixel.com, tirée de freepix.com)

d'accès licite, c'est-à-dire environ toutes les 102,4 ms.

Pour créer l'attaque par faux point d'accès, nous avons utilisé le logiciel Wi-Fi Pumpkin (TEAM, 2018). Wi-Fi Pumpkin est un logiciel d'audit qui permet de tester la sécurité des réseaux Wi-Fi contre des attaques, notamment l'attaque par faux point d'accès. Le logiciel est facile à utiliser, permet de créer et de copier les informations statiques se trouvant sur l'entête Ethernet et dans la trame de *beacon*. Cependant, il est important de noter que le firmware de la carte Wi-Fi AirPcap ne nous a pas permis de copier toutes les informations statiques du point d'accès licite, notamment les informations optionnelles telles que celles propres au constructeur (*vendor specific parameters*). Nous avons supposé que ces informations étaient copiées dans la suite de nos travaux, car sans elles, le faux point d'accès serait facilement détectable. Le plus important est que nous avons pu copier, entre autres, le SSID et le BSSID. Pour copier les informations statiques des *beacons* (BSSID, SSID, etc.) et créer le faux point d'accès, nous avons utilisé les commandes présentées dans le Listing 4 (voir Annexe). Certaines informations ne peuvent pas être indiquées par les commandes de Wi-Fi Pumpkin et doivent être spécifiées dans un fichier de configuration

externe.

3.3.2 L'absence de variation des caractéristiques du faux point d'accès

Dans cette configuration, nous avons uniquement fait varier le trafic, comme dans la situation normale. Il n'y a donc pas de variation proprement dite des paramètres du faux point d'accès. Nous avons dans un premier temps tenté de modifier la puissance d'émission de la carte Wi-Fi, mais cette opération s'est révélée infructueuse en raison des limitations du firmware de la carte Wi-Fi Pcap.

Après avoir capturé les données pour cette situation, nous avons arrêté les logiciels utilisés pour l'attaque par faux point d'accès afin de lancer ceux nécessaires à l'attaque par déauthentification.

3.4 La situation avec l'attaque par déauthentification

Dans la situation d'attaque par déauthentification (Figure 3.5), nous utilisons la carte AirPcap pour générer des trames de déauthentification. Nous utilisons le même ordinateur que celui utilisé pour effectuer l'attaque par faux point d'accès, mais nous avons arrêté le faux point d'accès.

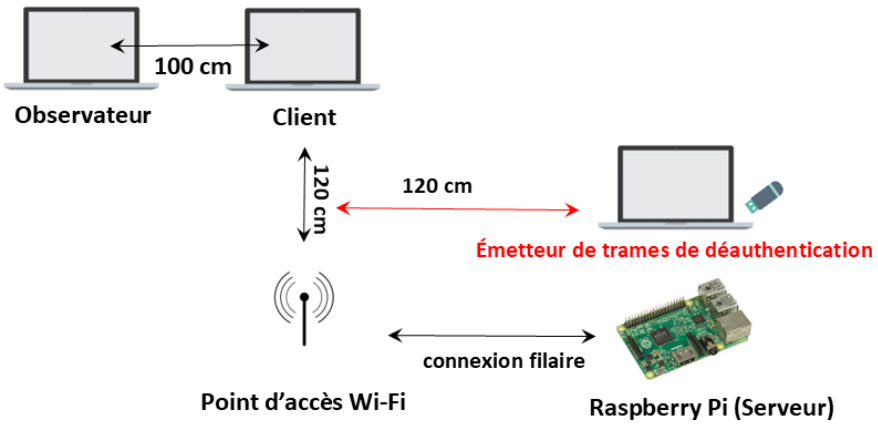

FIGURE 3.5 – Situation avec l'attaque par déauthentification

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay), icône usb (conçue par rawpixel.com, tirée de freepix.com)

3.4.1 Le choix du type d'attaque et sa mise en œuvre

Dans le cadre de cette attaque, l'attaquant tente de tromper les clients en faisant croire qu'ils ont reçu une trame de déauthentification de la part du point d'accès leur demandant de se déconnecter. Pour mener à bien cette attaque, l'attaquant doit simplement indiquer l'adresse MAC du point d'accès dans le champ *Source address* (ou SA voir Figure 2.14) de la trame de déauthentification à la place de son adresse MAC réelle. Étant donné que la trame n'est pas signée, chiffrée ou authentifiée, le client ne peut pas vérifier si elle provient bien du point d'accès. Les trames de déauthentification peuvent être émises individuellement, en blocs ou en continu.

Nous avons opté pour l'émission continue de trames de déauthentification car les attaquants utilisent souvent cette méthode pour s'assurer que les clients déconnectés ne se reconnectent pas au point d'accès licite. Il est à noter que dans nos expérimentations, nous avons observé que si les trames de déauthentification n'étaient pas émises continuellement, les clients se reconnectaient au point d'accès licite, ce qui rendait impossible la combinaison de cette attaque avec l'attaque par faux point d'accès. C'est donc pour ces raisons que nous

avons choisi d'émettre les trames en continu. Nous avons choisi la suite de logiciels Aireplay parce qu'elle permet d'usurper facilement les trames de déauthentification. Pour lancer l'attaque, nous avons utilisé les commandes dans le *listing* 5 (voir Annexe).

3.4.2 L'absence de variation de l'attaque par déauthentification

Nous avons appliqué les mêmes niveaux de trafic que dans une situation normale, à savoir : sans trafic, faible trafic, trafic moyen et trafic intense. Les variations possibles de cette attaque incluent notamment le nombre de trames de déauthentification par seconde ou le nombre de cibles visées. Nous n'avons pas modifié ces paramètres, car notre étude, dans cette thèse, se concentre uniquement sur la forme de l'attaque déauthentification qui peut être combinée avec un faux point d'accès. Cependant, il pourrait être intéressant d'explorer d'autres formes d'attaques de déauthentification, dans les travaux futurs, afin d'améliorer la performance et la complétude de notre système de détection.

Après avoir effectué les captures de données, une phase d'analyse et de traitement est requise pour détecter les éventuelles anomalies entre les différentes situations étudiées. La détection des trois types d'attaques se fondera sur ces anomalies.

3.5 Conclusion

Dans ce chapitre, nous avons présenté en détail les bancs de test pour les quatre situations distinctes : la situation normale, l'attaque par brouillage, l'attaque par faux point d'accès et l'attaque par déauthentification, ainsi que

leurs variations. La situation normale sert de référence à laquelle les attaques sont ajoutées. Pour chaque banc de test, nous avons fourni une description de l'ensemble des équipements matériels et logiciels utilisés.

La phase d'expérimentation nous a permis de collecter un ensemble de données significatif. Il est désormais nécessaire de procéder à leur analyse pour identifier les anomalies et les indicateurs pertinents qui seront utilisés pour la détection des trois types d'attaques.

Chapitre 4

Analyse des données et méthodes de détection

Dans le chapitre précédent nous avons exposé en détail les configurations expérimentales mises en place ainsi que les choix d'outils et de types ou formes d'attaques. Pour chaque configuration et variation, des captures de données ont été réalisées sur une durée de deux minutes à l'aide du logiciel Wireshark et d'une carte Wi-Fi externe connectée à l'ordinateur Observateur. Dans ce chapitre, nous présentons l'analyse de ces captures de données. Cette analyse nous a permis d'identifier différents attributs, dont l'évolution entre les différentes situations a permis de déterminer des indicateurs pertinents pour la détection des trois types d'attaques.

Ce présent chapitre est divisé en deux sections. Dans Ce chapitre se compose de deux sections distinctes. Dans la première section, nous détaillons la méthodologie que nous avons employée pour analyser les données capturées. Ces données ont été collectées par le logiciel Wireshark sous forme de trames IEEE 802.11 (Wi-Fi), qui peuvent être classées en trames de gestion, de contrôle ou de données. Notre analyse a porté sur les trames de gestion, pour lesquelles nous avons effectué un filtrage préalable en fonction du type de trame. Nous

avons ensuite examiné les informations spécifiques contenues dans ces trames, telles que la puissance du signal de la trame reçue (informations de la couche 1 du modèle OSI fournies par la carte Wi-Fi) ou l'adresse MAC source (informations de la couche 2 du modèle OSI). En effectuant cette analyse, nous avons pu étudier l'évolution de ces informations (ou attributs) en fonction des différentes situations, ce qui nous a permis d'identifier des anomalies et des différences.

Après avoir identifié les différences ou anomalies entre les situations, nous présentons dans la deuxième sous-section de ce chapitre une première implémentation d'un Système de Détection d'Intrusion (SDI). Cette première version est mise en place en utilisant des seuils de détection. Nous avons déterminé des seuils en dessous ou au-dessus desquels nous pouvons considérer la présence d'une des attaques. Cependant, cette méthode présente des limites. L'approche manuelle de détermination des seuils n'est pas optimale lorsqu'il faut tenir compte des différentes variations de situations et des formes avancées d'attaques.

En effet, dans ces cas, les différences ou anomalies que nous avons identifiées ne permettent pas facilement de distinguer les différentes situations. Ainsi, pour prendre en compte un plus grand nombre de cas d'attaques et de variations possibles, nous avons choisi d'utiliser une approche basée sur des algorithmes d'apprentissage automatique. Cette approche permet de modéliser de manière plus précise et abstraite les différentes situations. L'utilisation d'algorithmes d'apprentissage automatique nécessite une préparation des données, que nous détaillons dans la deuxième sous-section de ce chapitre, avant de présenter les résultats.

En résumé, dans ce chapitre, nous commençons par présenter l'analyse des trames avant de présenter les deux méthodes de détection.

4.1 L'analyse des trames

L'ordinateur Observateur a capturé un grand nombre de trames lors des expérimentations. Le tableau 4.1 récapitule le nombre de trames capturées pour chaque situation, toutes variations confondues.

Situation (toutes variations)	Nombre de trames	Temps total (minutes)
Situation normale	150 000	8
Situation avec attaque par brouillage	35000	32
Situation avec attaque par faux point d'accès	300 000	8
Situation avec attaque par déauthentification	350 000	8

TABLE 4.1 – Résumé des captures

Source: traitement auteur

Comme présenté dans le tableau 4.1, l'observateur a capturé environ 150 000 trames pour la situation normale, toutes variations confondues. Chaque capture a duré 2 minutes, pour un temps total de capture de 8 minutes. Nous avons ainsi constaté une quantité importante de trames capturées en peu de temps. Pour la situation d'attaque par brouillage, l'ordinateur Observateur a capturé environ 35 000 trames sur une période totale de 32 minutes, incluant les différentes variations de trafic et de puissance du signal de brouillage (voir les variations de l'attaque par brouillage dans le chapitre précédent). Pour la situation d'attaque par faux point d'accès, 300 000 trames ont été capturées en 8 minutes, tandis que pour la situation d'attaque par déauthentification, 350 000 trames ont été capturées en 8 minutes.

En nous fondant sur le nombre de trames reçues en fonction du temps, nous pouvons déjà comparer les différentes situations et identifier des différences. Si l'on compare la situation normale aux situations avec attaque par faux point

d'accès et attaque par déauthentification, nous remarquons que sur un même temps total et avec les mêmes variations, le nombre de trames est deux fois plus élevé dans la situation avec l'attaque par faux point d'accès et environ 2,3 fois plus élevé dans la situation avec l'attaque par déauthentification. Cela montre que le nombre de trames reçues par minute peut être utilisé pour différencier les situations et détecter la présence d'attaques. Toutefois, à ce stade, tous les types de trames sont confondus, et le nombre de trames reçues n'est pas encore adapté pour être utilisé comme un indicateur pour la détection des attaques.

Nous souhaitons que la détection puisse se faire exclusivement à partir des trames de gestion. Comme présenté dans le chapitre 2, les attaques par faux point d'accès et par déauthentification peuvent être détectées en utilisant des trames de gestion tandis que l'attaque par brouillage est détectée en se fondant sur les trames de données ou de contrôle. Nous préférons une détection basée sur les trames de gestion pour les trois attaques, car les trames de gestion, notamment les trames de *beacon*, sont envoyées de manière régulière, même en absence de trafic utilisateur. En nous basant sur les trames de gestion, nous nous assurons d'être capables de détecter les attaques dans un plus grand nombre de cas. Dans le chapitre 5, nous examinons la possibilité de considérer les trames de contrôle pour détecter une forme spécifique de l'attaque par faux point d'accès.

Concernant les types de trames, Les trames IEEE 802.11 sont classées en trois types : de gestion, de contrôle et de données. Les trames de gestion sont envoyées pour des opérations telles que rejoindre ou quitter un réseau, ou pour informer les autres de la présence d'un dispositif. Les trames de données contiennent des données utilisateurs tandis que les trames de contrôle sont utilisées pour contrôler l'émission et la réception des trames de données (IEEE802.11Standard, 2021). Ainsi, chaque capture contient des trames de gestion (appelées *management frames* en anglais), des trames de contrôle (*control frames* en anglais) et des trames de données (*data frames* en anglais). Afin de ne considérer qu'un seul type de trame, nous avons filtré les

données en fonction de leur type. Le type de trame est une information présente dans la trame elle-même et le filtrage peut être effectué à l'aide d'un logiciel tel que Wireshark.

Pour visualiser, analyser et filtrer les informations contenues dans les trames, nous avons utilisé le logiciel Wireshark. La Figure A.1 (voir Annexe) montre, par exemple, les trames capturées dans la situation normale avec trafic moyen. Chaque ligne représente une trame et certaines informations de la trame sont indiquées dans les colonnes par le logiciel Wireshark. Sur la même Figure A.1, on peut ainsi voir l'heure à laquelle le logiciel Wireshark a reçu la trame, l'adresse MAC source (cette adresse correspond à l'adresse MAC de l'émetteur), l'adresse MAC destination, la longueur de la trame, le numéro de séquence, la puissance du signal, le type de la trame et le sous-type de la trame (voir Figure A.1).

Le logiciel Wireshark permet également d'analyser individuellement chaque trame et de visualiser toutes les informations présentes dans la trame (voir Figure A.2 dans l'Annexe). Nous pouvons ainsi obtenir des informations telles que l'heure d'arrivée de la trame, sa taille et les informations de l'entête *Radiotap*. L'entête *Radiotap* contient des informations de la couche 1 (physique) calculées, identifiées et transmises par la carte Wi-Fi au logiciel Wireshark. Ces informations ne sont pas incluses dans la trame par l'émetteur. Ensuite, comme indiqué dans la même Figure A.2, il y a les informations de la couche 2, c'est-à-dire l'entête de la couche 2 (IEEE 802.11 Beacon frame) et le contenu (message ou instruction à envoyer) de la trame (IEEE 802.11 Wireless Management). La Figure A.2 représente une trame de type *beacon*.

Dans l'entête 802.11, on peut trouver l'adresse MAC source, l'adresse MAC destination et le contenu de la trame qui comporte des informations sur le point d'accès licite : nom du réseau, capacité, etc. Cette trame n'est pas encapsulée dans un entête des couches supérieures du modèle OSI (couches 3, 4, 5, 6 et 7) car une trame de type *beacon* est une information uniquement destinée aux

équipements fonctionnant sur la couche 2 et utilisant le protocole Ethernet (point d'accès, clients, etc.).

Nous nous limitons à l'analyse des informations de la couche 1 fournies par la carte Wi-Fi et aux informations de l'entête de la couche 2. Nous n'analysons pas les autres entêtes ou le contenu car ils sont cachés lorsque le réseau Wi-Fi est chiffré, et nous souhaitons avoir une approche de détection qui fonctionne aussi bien pour les réseaux chiffrés que pour les réseaux publics (non chiffrés). Certaines trames, notamment les trames de type *beacon*, ne sont jamais chiffrées. Pour ces types de trames, nous avons analysé à la fois le contenu de la trame (voir IEEE 802.11 Wireless Management dans la Figure A.2) et l'entête de trame (voir IEEE 802.11 Beacon frame dans la Figure A.2).

Après avoir trié les trames en fonction du type, nous avons analysé les informations des trames.

4.1.1 Analyse des trames de gestion

Le logiciel Wireshark fournit de nombreuses informations sur les trames de gestion. Cependant, nous n'avons pas analysé toutes ces informations de manière exhaustive. Nous avons effectué une étape de caractérisation des attaques pour présélectionner les informations ou attributs susceptibles d'être affectés par l'apparition des attaques. Nous avons ensuite confirmé la pertinence de ces attributs en les comparant à ceux proposés dans la littérature scientifique.

Par exemple, l'attaque par faux point d'accès se déroule en trois étapes : la création d'un faux point d'accès émettant des trames de *beacon* similaires à celles du point d'accès licite à intervalles réguliers, l'attente que des clients se connectent au faux point d'accès ou le forçage de leur déconnexion du point d'accès licite, puis le transfert effectif des clients vers le faux point d'accès. Nous pouvons déduire que, lors de cette attaque, il y aura deux points d'accès

émettant les mêmes trames de *beacon* en même temps sur le même canal. En analysant les trames de cette situation, nous devrions observer qu'il y a plus de trames de *beacons* que dans la situation normale. Il est par conséquent intéressant d'étudier l'intervalle entre deux trames de gestion. Nous avons également étudié le numéro de séquence des trames de gestion, car cette information avait été explicitement proposée dans la littérature scientifique.

Concernant l'attaque par brouillage, elle consiste à diminuer le rapport signal à interférence et donc à perturber les transmissions. En analysant les données de cette situation, nous devrions observer qu'il y a moins de trames que dans la situation normale. Il est donc intéressant d'étudier l'intervalle entre deux trames de gestion et également l'intervalle entre les numéros de séquence. Lorsqu'il y a des pertes de trames, l'intervalle entre trames et l'intervalle entre les numéros de séquence devraient être plus grands. L'effet de l'attaque par brouillage peut être étudié aussi en analysant la puissance du signal de la trame. La puissance du signal de la trame reçue devrait se dégrader en présence de l'attaque par brouillage.

En ce qui concerne l'attaque par déauthentification, elle consiste à émettre de manière excessive des trames de déauthentification. Étant donné que ces trames font partie des trames de gestion, il est intéressant d'analyser le sous-type de trame de gestion présent dans la capture ainsi que l'intervalle entre deux trames de gestion. L'intervalle moyen entre deux trames de gestion devrait diminuer car les trames de déauthentification sont envoyées de manière excessive et à de petits intervalles.

Ainsi, pour résumer, nous avons conclu qu'il fallait étudier l'évolution des attributs suivants : l'intervalle entre deux trames (*frame interval* en anglais), l'intervalle entre les numéros de séquence de deux trames consécutives (*Sequence number gap* en anglais), la puissance du signal de la trame (*Received Signal Strength Indicator* en anglais) et le sous-type de la trame (*subtype* en anglais). La plupart de ces attributs ont été proposés dans la littérature

scientifique, mais pas nécessairement pour les trois types d'attaques.

4.1.2 Évolution des attributs

Nous étudions l'évolution des différents attributs (l'intervalle entre deux trames, l'intervalle entre les numéros de séquence de deux trames consécutives, la puissance du signal de la trame et le sous-type de la trame) en fonction des différentes situations. Dans cette sous-section, nous présentons uniquement les résultats de l'analyse pour les situations avec trafic moyen. Bien que nous ayons réalisé cette analyse pour toutes les variations, nous avons choisi de présenter seulement l'évolution des attributs en fonction des situations avec trafic normal pour plus de clarté. Les résultats des autres variations seront exposés dans le chapitre suivant.

Attribut 1 : Sous-type de trame de gestion

Dans le type de trame de gestion, il existe différents sous-types, tels que les trames de *beacon* (sous-type 8), les trames de requêtes sonde (sous-type 5) et de réponse sonde (sous-type 6), conformément aux spécifications IEEE 802.11. En comparant la présence des différents sous-types de trame dans les situations où le trafic entre le client et le serveur est moyen, nous remarquons, comme le montre la Figure 4.1, que la majorité des trames reçues sont des trames de *beacon* dans la plupart des situations. La situation avec attaque de déauthentification est l'exception, car dans cette situation, la majorité des trames de gestion sont des trames de déauthentification. Nous observons également un autre type de trame présent dans les trois situations : les réponses de sonde (*probe response* en anglais).

Les réponses de sonde sont des trames de gestion envoyées par les points

d'accès pour répondre aux demandes de sonde envoyées par les clients. En comparant avec la situation normale, nous remarquons que le nombre de réponses de sonde (*probe response*) varie d'une situation à une autre. Par exemple, dans la situation avec l'attaque par brouillage fort, seulement 24 réponses de sonde ont été émises par le point d'accès, contre 87 dans la situation normale. Cela peut s'expliquer par le fait que lors de l'attaque par brouillage, le client estime que le canal est occupé et émet donc moins de demandes de sonde (et moins de trames en général). Les collisions lors de la réception pourraient être une autre raison. L'attaque par brouillage a pour finalité de dégrader la réception des signaux. Ainsi, il est possible que plus de réponses de sondes aient été envoyées, mais qu'elles n'aient pas été correctement reçues.

Bien que le nombre de réponses de sonde envoyées puisse varier d'une situation à l'autre, nous ne l'avons pas pris en compte dans notre analyse. En effet, le nombre de demandes et de réponses de sonde dépend du nombre de clients présents dans le réseau, ainsi que des paramètres de connexion des clients, tels que la connexion automatique au point d'accès favori. Elles ne sont pas émises de manière régulière. Dans notre configuration, il n'y a qu'un seul client. Cependant, si le nombre de clients augmente, les demandes et les réponses de sonde peuvent fluctuer. En effet, certains clients peuvent ne pas émettre de requêtes sondes, tandis que d'autres en émettent. Tout va dépendre de leur paramétrage. Par conséquent, ces fluctuations ne sont pas assez génériques pour être utilisées comme attribut dans notre analyse.

Nous observons également, comme attendu, sur la Figure 4.1, que dans la situation de l'attaque par déauthentification, les trames de déauthentification sont majoritaires par rapport aux autres situations. Ces trames sont présentes sur toute la capture de deux minutes et correspondent aux trames envoyées à des intervalles très courts par la carte Wi-Fi de l'attaquant. Nous avons donc conclu que, dans nos configurations, la présence d'une quantité importante de trames de déauthentification sur un petit intervalle de temps est un indicateur d'une attaque par déauthentification.

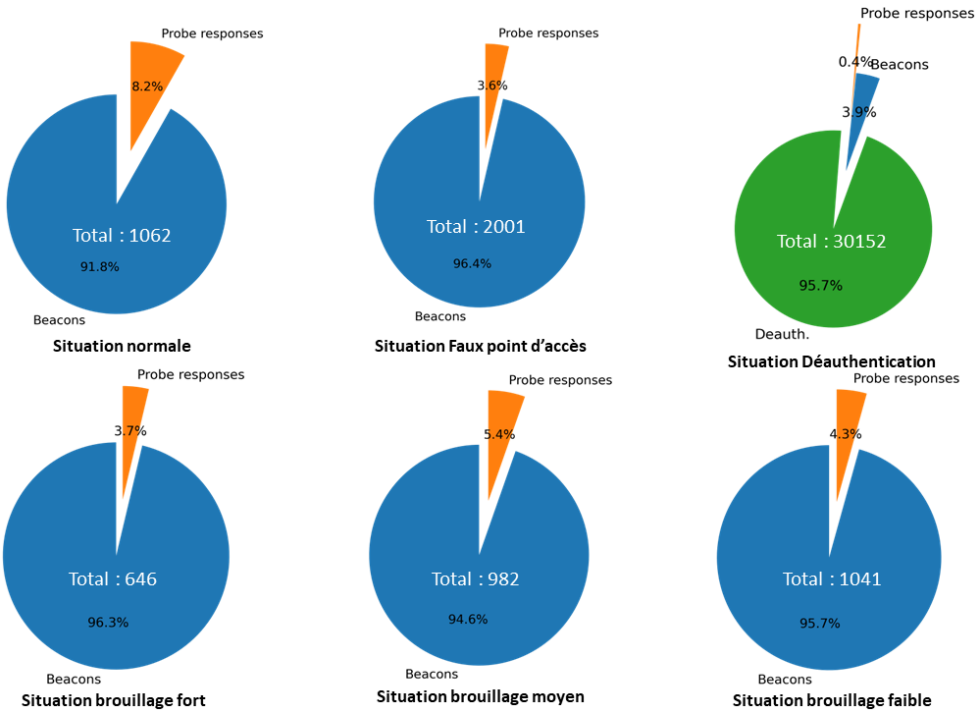

FIGURE 4.1 – Répartition des sous type de trame de gestion

Source: traitement auteur ; en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

Un autre point important à relever dans cette Figure 4.1 est que la quantité de trames de *beacon* reçues diffère pour chaque situation. Dans la situation normale, 975 trames de *beacon* ont été reçues. Dans la situation avec l'attaque par faux point d'accès, il y a eu 1928 trames de *beacon*, soit environ deux fois plus que dans la situation normale. Dans la situation avec l'attaque par déauthentification, il y a eu 1167 trames de *beacon*, soit environ 1.2 fois plus que dans la situation normale. Enfin, dans les situations avec l'attaque par brouillage (puissance faible, moyenne et forte), respectivement 1117, 1118 et 622 trames de *beacon* ont été reçues.

Nous avons donc conclu que l'attribut sous-type peut aider à détecter les situations de déauthentification, mais il est nécessaire d'étudier l'évolution de l'intervalle entre deux trames de gestion en fonction des différentes situations afin de mieux différencier les autres situations.

Attribut 2 : Intervalle entre deux trames de gestion

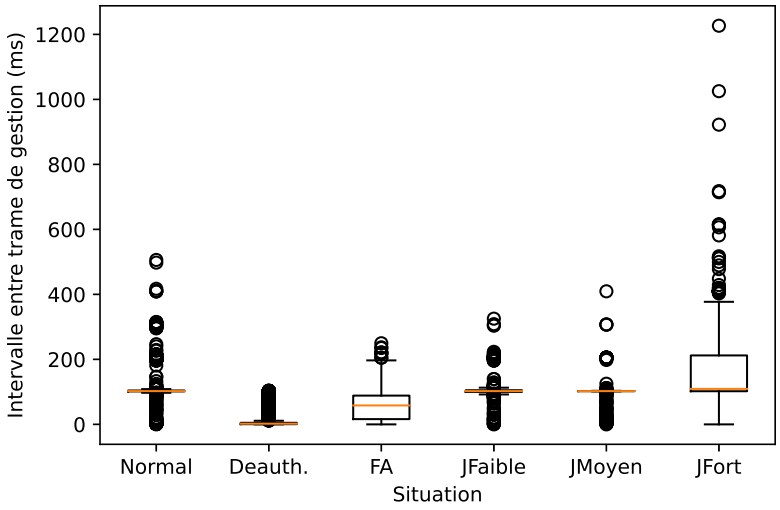

FIGURE 4.2 – Comparaison entre situations de l'intervalle entre trames de gestion

Source: traitement auteur ; en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

En examinant la Figure 4.2, nous pouvons constater que l'intervalle entre les trames n'est pas identique dans les quatre situations. Nous savons, d'après la Figure 4.1, que la plupart des trames de gestion reçue, dans la situation normale, la situation de brouillage et la situation d'attaque par faux points d'accès, sont des trames de *beacon*. Aussi, selon la norme IEEE 802.11, l'intervalle entre deux trames de *beacon* d'un point d'accès doit être d'environ **102,4 ms** (Target Beacon Transmission Time).

Dans une situation normale, l'intervalle de trame moyen (représenté par la barre orange dans la Figure 4.2) est légèrement supérieur à **102,4 ms**, avec plusieurs trames inférieures à 102 ms. Les trames reçues en dessous ou au-dessus de la barre des 102,4 ms peuvent s'expliquer par la présence d'autres types de trames que les trames de *beacons*, notamment les trames de réponse de sonde. Cependant, le pourcentage de ces trames est minime en comparaison avec les trames de *beacons* (voir Figure 4.1), et ne devrait pas affecter

considérablement la moyenne. Ces trames peuvent également correspondre à des trames de *beacons* qui ont été retardées et envoyées en une seule fois. En effet, la norme IEEE 802.11 autorise un point d'accès à différer la transmission de trames de *beacons* lorsqu'il y a beaucoup de trames de données à envoyer. Enfin, cela peut correspondre à des pertes de trames qui peuvent survenir dans toute communication.

Dans le cas d'une attaque par déauthentification, l'intervalle de trame est proche de zéro. En effet, les trames de déauthentification usurpées et envoyées par l'attaquant sont mélangées avec les autres trames envoyées par le point d'accès (les trames de déauthentification et de *beacon*). Comme indiqué précédemment, ces trames de déauthentification sont également envoyées, de manière intensive, à un intervalle bien inférieur à 102,4 ms, ce qui réduit considérablement l'intervalle moyen entre deux trames de gestion dans cette situation.

Dans le cas d'une attaque par faux point d'accès, l'attaquant a créé un faux point d'accès qui envoie des trames de *beacon* avec les mêmes informations statiques (adresse MAC source, etc.) que le point d'accès licite. Les deux points d'accès fonctionnent simultanément sur le même canal Wi-Fi et envoient les mêmes trames de *beacon*. L'observateur ne peut pas différencier les trames de *beacon* et par conséquent, il y a deux fois plus de trames de *beacon*, ce qui divise par deux l'intervalle moyen entre deux trames.

Enfin, dans le cas d'une attaque par brouillage, on pourrait s'attendre à une perte importante de trames (l'attaque par brouillage devrait fortement affecter ou annihiler toute communication). Cependant, nous avons observé que l'intervalle moyen de trame est proche de 102,4 ms dans la situation d'une attaque par brouillage de forte puissance (JFort sur la Figure). Ceci peut s'expliquer par le fait que certaines trames de gestion, notamment les trames de *beacon*, sont envoyées en utilisant l'amendement IEEE 802.11 b. Cet amendement permet l'envoi de trames sur une zone de couverture plus large, mais à un débit peu élevé. Comme les trames de *beacon* annoncent la présence d'un point d'accès,

elles sont envoyées en utilisant l'amendement IEEE 802.11 b pour informer la présence d'un point d'accès sur une plus grande couverture et ainsi toucher un plus grand nombre de clients potentiels.

La modulation prévue par cet amendement est un étalement de spectre à séquence directe. Les signaux envoyés sur un étalement de spectre sont plus résilients face aux attaques par brouillage (MARTINEZ et al., 2008). Ainsi, l'impact de l'attaque par brouillage sur les trames de *beacon* et les trames de gestion de manière générale est très faible. En revanche, les trames de données sont envoyées par le point d'accès à un débit plus élevé en utilisant généralement l'amendement 802.11n, qui est moins résilient aux interférences que l'amendement 802.11b. Ces transmissions sont perturbées et peuvent être inexistantes surtout lorsque la puissance du brouillage est forte. Nous observons également, dans les situations de brouillage, qu'il existe des intervalles de trame en dessous et au-dessus de la valeur moyenne, notamment dans les situations avec un brouillage moyen et faible. L'intervalle entre deux trames en dessous de la valeur moyenne peut s'expliquer par le fait qu'il y a des trames de réponse de sonde, mais aussi par le fait que le point d'accès détecte que le canal est occupé et met en mémoire tampon certaines trames de *beacon* de temps en temps, qu'il envoie d'un seul trait lorsque cela est possible.

L'intervalle de trame de *beacon* au-dessus de la valeur moyenne peut s'expliquer par le fait que certaines trames de *beacon* sont envoyées mais ne sont pas reçues en raison de collisions à la réception. Même si les trames de *beacon* sont émises en étalement de spectre à séquence directe, il peut toujours y avoir quelques pertes de trames de *beacon* dues à des collisions à la réception. Enfin, nous remarquons également que dans les situations de brouillage faible (JFaible sur la Figure) et de brouillage moyen (JMoyen sur la Figure), l'intervalle moyen est plus proche de 102,4 ms que dans la situation normale. Dans la figure 4.1, nous avons également noté qu'il y avait plus de trames de *beacon* dans les situations JFaible et JMoyen que dans la situation normale. Cela peut s'expliquer par le fait que dans toutes les situations, il y a un trafic de 50 Mo/s.

Ce trafic est toutefois affecté par le signal de brouillage dans les situations de brouillage faible et moyen. Dans ces situations, le point d'accès a moins besoin de différer l'envoi des trames de *beacon* pour donner la priorité aux trames de données, car le trafic est interrompu ou fortement affecté. Ce phénomène est moins visible pour la situation de l'attaque par brouillage fort (JFort) parce qu'il y a également des pertes de trames de *beacon*. Nous avons donc conclu que l'intervalle entre deux trames de gestion peut être utilisé comme indicateur pour détecter l'attaque par déauthentification et l'attaque par faux point d'accès.

La puissance du signal de la trame est également une information dont l'évolution entre situations doit être étudiée.

Attribut 3 : Puissance moyenne du signal des trames reçues

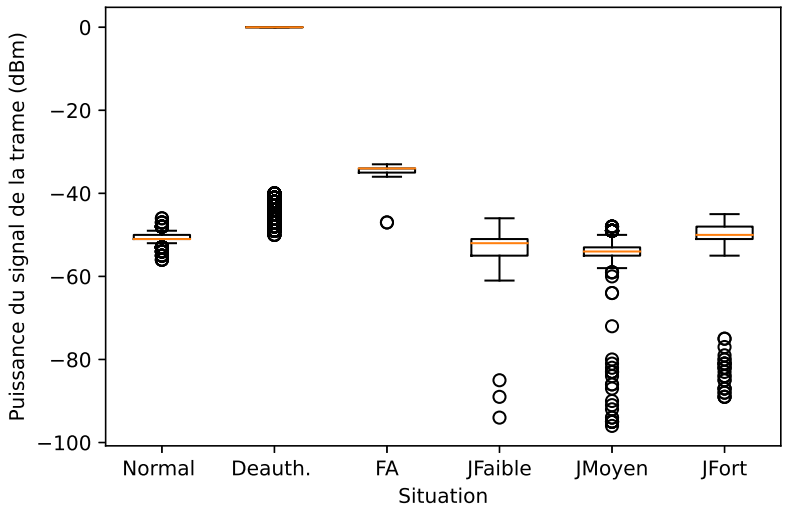

FIGURE 4.3 – Comparaison entre situation de la puissance du signal

Source: traitement auteur ; en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

Dans la figure 4.3, nous pouvons observer l'évolution de la puissance du signal des trames reçues, mesurée par le RSSI (Received Signal Strength Indi-

cation). Le RSSI représente la puissance du signal de la trame en dBm calculé par la carte Wi-Fi lorsqu'elle reçoit une trame. Nous remarquons, sur la figure 4.3, que dans la situation normale, la puissance moyenne du signal des trames reçues est d'environ -50 dBm avec un intervalle entre -56 dBm et -44 dBm. Cette moyenne et cet intervalle diffèrent entre les situations. Concernant les situations avec l'attaque par brouillage, en comparaison avec la situation normale, la moyenne du RSSI est plus basse pour la situation avec brouillage moyen, presque la même pour la situation avec brouillage faible et un peu plus élevé pour la situation avec brouillage fort. L'attaque par brouillage a pour effet de diminuer le rapport signal à bruit et par conséquent affecte la valeur RSSI des trames. L'impact est plus visible dans la situation moyenne (JMoyen). La valeur moyenne RSSI est un peu plus élevée dans la situation avec brouillage fort. Cela peut s'expliquer par le fait que dans cette situation, il y a des pertes de trames de *beacon*. Ces pertes doivent très probablement concerner les trames avec un RSSI très faible, notamment celles qui sont en dessus de -90 dBm. Ces trames sont bien reçues dans la situation JFaible et JMoyen mais pas dans la situation JFort, l'intervalle est donc réduit et la moyenne remonte.

Dans la situation d'attaque par déauthentification, il est observé que le RSSI moyen est égal à 0 dBm. Dans cette situation, les trames de déauthentification sont prédominantes, et il semble que la carte Wi-Fi de l'observateur ne puisse pas calculer la valeur RSSI de ces trames en raison de l'intervalle trop court entre elles. Comme le champ RSSI est vide pour les trames de déauthentification, nous avons remplacé ce champ par des zéros pendant le processus de prétraitement. Les valeurs non nulles correspondent au RSSI des trames de *beacon*. Dans la situation d'attaque par faux point d'accès, la valeur moyenne du RSSI est supérieure à la situation normale et est d'environ -35 dBm. Cela peut être expliqué par le fait que les trames de faux points d'accès sont prises en compte et font augmenter la valeur moyenne. On peut conclure que le RSSI pourrait être utilisé comme un indicateur pour détecter les trois attaques, mais il y aura des faux négatifs (fausses alertes), notamment dans les

situations d'attaque par brouillage faible et moyen ainsi que dans la situation normale, en raison du chevauchement des intervalles dans ces situations. Un indicateur basé sur les valeurs RSSI peut cependant être combiné avec d'autres indicateurs pour mieux différencier les situations.

Une autre information dont l'évolution peut être intéressante à étudier est l'écart entre les numéros de séquence de deux trames.

Attribut 4 : Écart moyen entre les numéros de séquence de deux trames

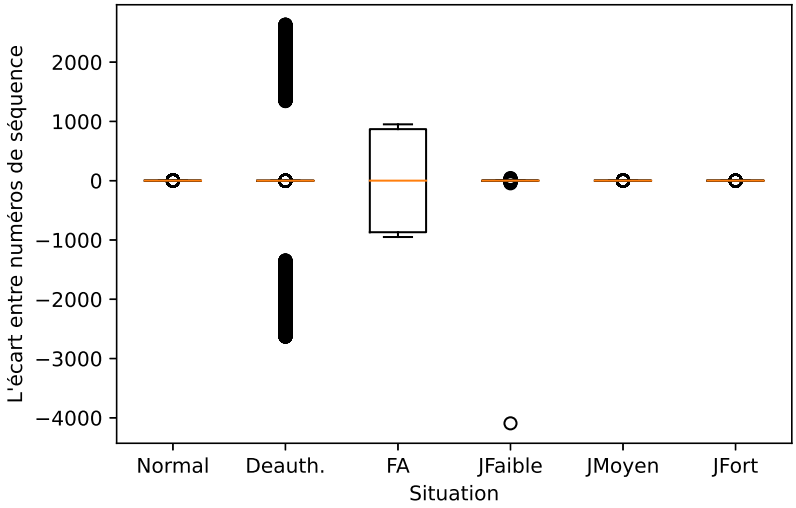

FIGURE 4.4 – Comparaison entre situation de l'écart entre les numéros de séquence

Source: traitement auteur ; en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

Sur la figure 4.4, nous observons que l'écart entre numéros de séquence diffère nettement dans les situations d'attaques par déauthentification et par faux point d'accès par rapport à la situation normale. L'écart entre numéros de séquence est défini comme l'écart entre les numéros de séquence de deux trames de gestion consécutives. Il est calculé après la capture. Dans la situation normale, la plupart des trames capturées sont des trames de *beacon* et

l'écart moyen entre numéros de séquence est autour de 1, ce qui est le comportement normal. En effet, entre deux trames de *beacon*, l'écart entre numéros de séquence doit être de 1 s'il n'y a pas eu d'autres transmissions de trames de gestion entre ces deux trames de *beacon*. L'écart entre numéros de séquence dans la situation normale peut être supérieur à 1 lorsque des trames de réponse de sonde ont été envoyées entre deux trames de *beacon*. Ces trames sont incidentes et ne sont pas envoyées à intervalles réguliers, elles ne devraient donc pas affecter l'écart moyen entre les numéros de séquence. L'écart ne peut pas être égal à 0, car les trames de *beacon* ne sont jamais retransmises.

Dans la situation d'attaque par déauthentification, nous observons un écart moyen très élevé, ce qui diffère de la situation normale. En effet, les trames de déauthentification ne sont pas envoyées par le point d'accès licite, mais par l'attaquant. Même si l'attaquant usurpe l'adresse MAC du point d'accès, comme indiqué dans le chapitre 2, il est très difficile pour lui de synchroniser son compteur de numéros de séquence avec celui du point d'accès. Ceci cause d'importantes fluctuations entre les numéros de séquence. La même observation s'applique dans la situation d'attaque par faux point d'accès, avec une légère différence dans la plage d'écart de numéro de séquence. Cela peut s'expliquer par le fait qu'il y a relativement la même quantité de trames de *beacon* émises entre le point d'accès licite et le faux point d'accès dans cette situation. L'écart est donc calculé entre une trame de *beacon* du point d'accès licite et une trame du faux point d'accès. L'écart oscille donc dans un intervalle.

Dans la situation de l'attaque déauthentification, les trames de déauthentification usurpées sont prédominantes, l'écart est donc majoritairement calculé entre les trames de déauthentification usurpées et parfois entre une trame de déauthentification usurpée et une trame de *beacon*. Cela explique une moyenne de 1 avec quelques grands écarts. En ce qui concerne les attaques par brouillage, l'écart moyen de numéro de séquence est de 1, parce que le signal de brouillage n'affecte pas les trames de gestion qui sont émises en utilisant l'amendement IEEE 802.11 b. Les écarts autour de 1 présent dans les différentes situations

correspondent aux écarts entre une trame de *beacon* et une trame de réponse de sonde.

Nous pouvons ainsi conclure que les quatre attributs peuvent être utilisés pour détecter les trois attaques. Certains de ces attributs ont déjà été proposés dans la littérature scientifique, mais souvent seul un ou deux attributs sont utilisés à la fois. Un attribut peut être contourné par un attaquant et il est donc plus judicieux d'adopter une approche de détection qui analyse plusieurs attributs simultanément.

4.2 Détection des attaques

Dans cette section, nous présentons deux méthodes que nous avons utilisées pour déterminer la présence d'attaques. La première méthode consiste à déterminer des seuils en-dessous ou au-dessus desquels la présence d'une des attaques peut être caractérisée. Cependant, cette méthode présente plusieurs limitations, ce qui nous a poussés à adopter une méthode utilisant des algorithmes de classification. La méthode utilisant des algorithmes de classification consiste à faire apprendre aux algorithmes les différentes situations et leurs variations en leur fournissant des données labellisées. Ces algorithmes seront alors en mesure de créer des modèles capables de prédire la présence d'une situation d'attaque lorsqu'ils seront présentés avec de nouvelles données.

4.2.1 Détermination des seuils

La détermination de seuils consiste à créer des indicateurs en mettant des seuils en dessous ou au-dessus desquels la présence des attaques peut être caractérisée. Le tableau 4.2 résume les seuils que nous avons déterminés après

l'analyse de données pour la détection des attaques en utilisant des trames de gestion.

Attaque	Intervalle entre trame	Intervalle entre numéro de séquence	Puissance du signal	Sous-type
Faux point d'accès	$\leq$ 52.4 ms	$>$ 8	$>$ -40 dbm	$=$ 8
Brouillage fort	$\geq$ 700 ms	Non-applicable	$<$ -59 dbm	$=$ 8
Brouillage moyen	Non-applicable	Non-applicable	$<$ -90 dbm	$=$ 8
Déauthentification	$\leq$ 2 ms	Non-applicable	0 dbm	$=$ 5

TABLE 4.2 – Trame de gestion - Seuils pour les attaques

Source: traitement auteur

Nous avons déterminé ces seuils en nous basant sur les valeurs moyennes (ligne orange) dans les différentes boîtes à moustaches (*box-plots*). Lorsqu'il y avait un chevauchement entre plusieurs situations, nous avons choisi le seuil au-delà duquel il n'y avait plus de chevauchement, en particulier pour la situation d'attaque par brouillage avec puissance moyenne. Hormis l'indicateur du sous-type de trames, les autres indicateurs sont indépendants et peuvent être utilisés seuls ou en combinaison. Par exemple, un intervalle entre deux trames inférieur ou égal à 52,3 ms, mais supérieur à 2 ms caractérise, dans nos configurations, une attaque par faux point d'accès. De même, un écart entre les numéros de séquence de deux trames supérieur à 8 caractérise également cette attaque.

En nous appuyant sur ces indicateurs, nous avons implémenté une première version d'un Système de Détection d'Intrusion en Python.

4.2.2 Première implémentation du SDI

Cette première version du SDI a été implémentée en Python et a deux modes de détection. Elle peut soit analyser les trames reçues directement depuis la carte Wi-Fi (mode *live*), soit analyser des captures sous format PCAP. Dans les deux cas, une fenêtre tampon de 20 secondes contenant des trames provenant soit directement de la carte Wi-Fi, soit d'une lecture d'un fichier PCAP, est analysée par le bloc de détection du SDI. Dans cette première version, nous avons pris en compte uniquement les attaques par faux point d'accès et par brouillage. Le bloc de détection analyse trame par trame et calcule les valeurs d'intervalle entre trame, d'écart entre les numéros de séquence et de puissance du signal avant de comparer ces valeurs aux seuils déterminés dans la sous-section précédente. Si les valeurs tombent dans les seuils déterminés, le système de détection indique la présence d'une attaque (l'état attaque est vrai si au moins un de ces opérandes est vrai). Sinon, le SDI indique une situation normale. Lorsque toutes les trames de la fenêtre tampon sont traitées, le bloc de détection analyse les prochaines 20 secondes de captures, et ce, jusqu'à l'arrêt de l'utilisateur dans le mode live et jusqu'à la fin du fichier dans le mode lecture d'un fichier PCAP.

Le SDI affiche en continu l'évolution de deux attributs : l'écart entre les numéros de séquence et l'intervalle entre deux trames. Lorsqu'il n'y a pas d'attaque, l'intervalle entre deux trames de type beacon est d'environ 102,4 ms et l'écart entre les numéros de séquence de deux trames est inférieur ou égal à 8. En situation normale, comme indiqué sur la Figure 4.5, le SDI affiche un état sans attaque.

Toutefois, lorsqu'il y a une attaque par faux point d'accès par exemple, comme indiqué dans la Figure 4.6, le SDI change d'état et passe en mode "avec attaque". Cet état peut être vérifié en analysant les valeurs d'intervalle entre deux trames et d'écart entre les numéros de séquence. Lorsqu'une attaque par faux point d'accès se produit, l'intervalle entre les trames de gestion est divisé

FIGURE 4.5 – SDI - Situation normale

Source: traitement auteur : en utilisant la bibliothèque Tkinter de Python (LUNDH, 1999)

par deux et de fortes fluctuations peuvent être observées en ce qui concerne l'écart entre les numéros de séquence.

La détection par seuils implique que les conditions doivent être strictement respectées pour que le SDI indique la présence d'une attaque. Cependant, comme nous allons le voir, les données entre les différentes situations ne suivent pas une répartition linéaire.

FIGURE 4.6 – SDI - Situation avec l'attaque par faux point d'accès

Source: traitement auteur : en utilisant la bibliothèque Tkinter de Python (LUNDH, 1999)

4.2.3 Limitations de la première implémentation du SDI

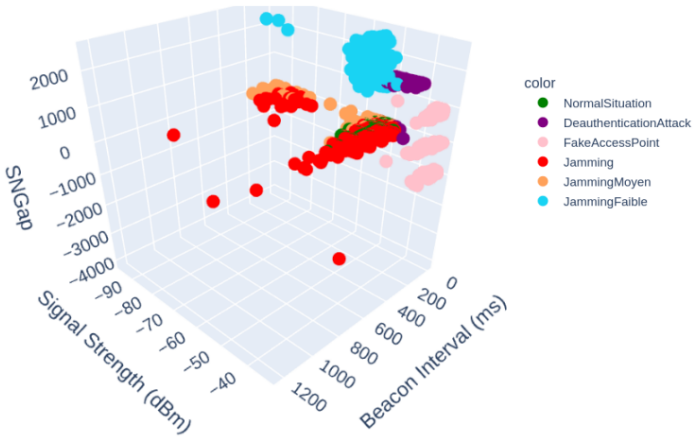

FIGURE 4.7 – Répartition dans l'espace des trames de gestion

Source: traitement auteur : en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

Nous avons utilisé les valeurs moyennes et des valeurs sans chevauchement pour déterminer les seuils. Toutefois, cela ne prend pas en compte les valeurs au-dessus ou en-dessous de la moyenne et celles dans les intervalles de valeurs en chevauchement. Cette méthode est également manuelle et fastidieuse. Dans cette première implémentation du SDI, nous avons considéré uniquement les situations de trafic moyen, mais nous devons également prendre en compte les situations avec différentes variations du trafic et de la puissance du signal de brouillage. Une autre limitation est que la première implémentation ne peut détecter qu'une seule attaque à la fois, par exemple, soit l'attaque par brouillage, soit l'attaque par déauthentification. Pour détecter plusieurs attaques, il faudrait paralléliser le processus de détection. La Figure 4.7 représente la répartition dans l'espace des valeurs des trois attributs en fonction des différentes situations. Nous remarquons qu'en réalité, seule la situation de faible brouillage peut être séparée linéairement. Les valeurs pour les autres situations sont entremêlées et ne peuvent pas être nettement séparées sur un plan linéaire.

Dans de tels cas, pour obtenir une détection plus efficace, il est préférable

d'opter pour une méthode de détection par classification. Cette méthode utilise des prédicteurs qui peuvent mieux séparer les donner et prendre en compte plusieurs indicateurs simultanément (Kreutzer et al., 2020).

L'Intelligence Artificielle (IA) est un ensemble de théories et de techniques qui permet de développer des machines capables de simuler certaines caractéristiques de l'être humain, telles que la reconnaissance de la parole, la vision ou la prise de décision. Les sous-domaines de l'IA reposent principalement sur l'apprentissage automatique, qui permet d'analyser d'importantes quantités de données et de prendre des décisions en fonction de celles-ci (Ertel, 2018). Il existe différents types d'algorithmes d'apprentissage automatique : supervisé, non supervisé, semi-supervisé, par renforcement, en ensemble, multitâches, par instance et par l'utilisation de réseaux de neurones (Mahesh, 2020). L'apprentissage supervisé consiste à entraîner un algorithme à prédire des réponses à partir d'un ensemble de données étiquetées (Bonaccorso, 2017). L'apprentissage non supervisé consiste à chercher des structures ou des modèles cachés dans des données non étiquetées. L'apprentissage semi-supervisé consiste à entraîner un algorithme avec une combinaison de données labellisées et non labellisées, ce qui est utile lorsque peu de données labellisées sont disponibles (Bonaccorso, 2017).

L'apprentissage par renforcement vise à entraîner un agent autonome, généralement représenté par un algorithme, à prendre des décisions en interagissant avec un environnement. L'objectif principal de cet agent est de maximiser les récompenses qu'il reçoit en ajustant ses actions en fonction des pénalités éventuelles qui peuvent survenir (Bonaccorso, 2017). L'apprentissage en ensemble consiste à combiner plusieurs modèles ou algorithmes pour améliorer la prédiction. L'apprentissage multitâche consiste à entraîner un modèle à effectuer simultanément plusieurs tâches liées (Zhang et al., 2018). Ainsi, au lieu d'entraîner différents modèles pour chaque tâche, l'apprentissage multitâche permet de tirer parti des informations liées et partagées entre les tâches pour améliorer la précision de détection du modèle. L'apprentissage par ins-

tance consiste à entraîner un modèle à partir d'exemples individuels sans besoin d'étiquetage (AHA et al., 1991). L'apprentissage par réseaux de neurones utilise des réseaux composés de nœuds interconnectés, appelés neurones qui traitent et transmettent des informations en utilisant des connexions pondérées, pour entraîner un modèle (LAWRENCE, 1993).

Les algorithmes d'apprentissage peuvent être classés en fonction de leur objectif, tels que la classification, le traitement du langage naturel, le clustering, etc. (BONACCORSO, 2017).En ce qui concerne les algorithmes de classification, certains sont capables de classer les données en seulement deux classes, tandis que d'autres peuvent les classer en plusieurs classes. Dans le cadre de cette thèse, nous avons utilisé des algorithmes de classification multiclasse et binaire (BONACCORSO, 2017). L'utilisation des algorithmes de classification requiert une étape préalable de préparation des données.

4.2.4 Utilisation d'algorithmes de classification

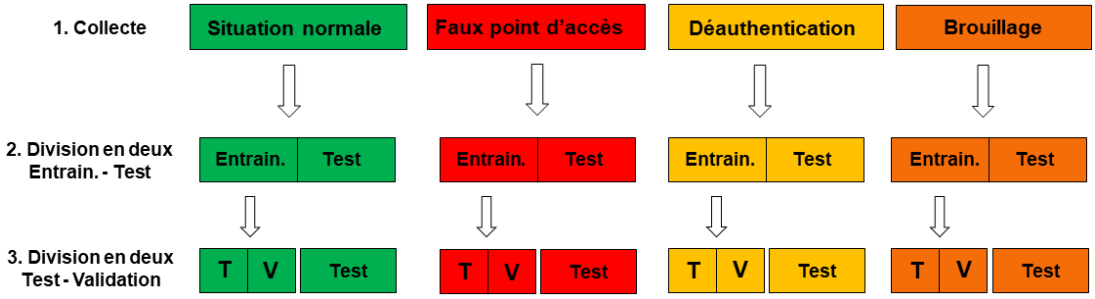

FIGURE 4.8 – Division des données

Source: traitement auteur

Dans le cadre de la préparation des données, plusieurs étapes sont nécessaires. Tout d'abord, il est essentiel d'extraire les attributs pertinents des jeux de données, c'est-à-dire les caractéristiques qui seront utilisées par les algorithmes d'apprentissage (NASTESKI, 2017). Ensuite, il est important d'étiqueter les jeux de données en attribuant des étiquettes ou des classes, afin que le mo-

dèle puisse apprendre à les classifier correctement. Enfin, il est nécessaire de diviser les jeux de données en ensembles distincts : un ensemble d'apprentissage, sur lequel le modèle sera entraîné, et un ensemble de test, qui servira à évaluer les performances du modèle une fois entraîné (NASTESKI, 2017). Le jeu d'entraînement peut également être subdivisé en deux parties afin d'obtenir un jeu de validation permettant d'évaluer le modèle et d'ajuster ses paramètres avant de procéder à son évaluation finale. Dans certaines situations où il existe un déséquilibre de données entre les classes, il peut être nécessaire d'appliquer des techniques de suréchantillonnage ou de sous-échantillonnage sur les ensembles d'apprentissage afin d'équilibrer les données (BONACCORSO, 2017).Le suréchantillonnage consiste à augmenter artificiellement le nombre d'exemples de la classe minoritaire, tandis que le sous-échantillonnage implique la réduction du nombre d'exemples de la classe majoritaire. Ces techniques visent à améliorer les performances du modèle en réduisant le biais introduit par le déséquilibre des données. Il convient ensuite de choisir l'algorithme approprié en fonction du contexte et des exigences spécifiques de l'application (BONACCORSO, 2017).

Nous avons suivi cette procédure dans cette thèse. Après avoir extrait les attributs des données, nous avons procédé à leur étiquetage en fonction de leur situation respective. Pour chaque situation, nous avons divisé l'ensemble d'attributs étiquettés en un ensemble d'entraînement, de test et de validation, comme illustré dans la Figure 4.8. Les données étiquetées pour chaque situation ont ensuite été concaténées pour entraîner et tester les modèles, comme indiqué sur la figure 4.9. Nous avons entraîné sept modèles différents en utilisant sept algorithmes d'apprentissage supervisés : *K-Nearest Neighbor* (KNN), *Random Forest*, *Classification and Regression Trees* (CART), *Logistic Regression*, *Naive Bayes*, *Support Vector Machine* (SVM) et *Linear Discriminant Analysis*.

Nous avons choisi les algorithmes de classification mentionnés ci-dessous, car ils sont largement utilisés dans la littérature scientifique pour la détection des trois types d'attaques étudiés dans cette thèse. Cela nous permet de

comparer et prendre du recul par rapport aux travaux existants. Chaque algorithme utilise sa propre méthode pour séparer et classifier les données dans leur classe correspondante. **L'algorithme KNN** détermine si une donnée se trouve dans une classe en utilisant les k plus proches voisins d'un point de requête donné (Hazzan et al., 2022). **L'algorithme CART** est basé sur la construction d'un arbre de décision qui divise les données en sous-ensembles en fonction de différentes combinaisons et valeurs. Cette procédure est répétée jusqu'à ce que les divisions les plus performantes soient identifiées, permettant ainsi de classer précisément les données. Lors de la construction de l'arbre de décision par l'algorithme CART, l'entropie est utilisée comme mesure d'impureté pour évaluer la pertinence des différentes combinaisons de variables. Une entropie élevée indique une plus grande hétérogénéité des données, ce qui rend leur classification plus difficile. Par conséquent, l'algorithme CART cherche à diviser l'ensemble de données de manière à réduire au maximum l'entropie de chaque sous-ensemble, tout en maximisant la différence d'entropie entre ces sous-ensembles (Breiman et al., 1984 ; Nasteski, 2017).

L'algorithme *Random Forest* utilise un ensemble d'arbres de décision. Chacun de ces arbres est entraîné de manière indépendante sur des sous-ensembles de données. Chaque arbre génère une prédiction ou une estimation, et c'est la combinaison de ces prédictions (ou réduction de variance) qui permet de prendre une décision finale. (Louppe, 2014). **L'algorithme de régression logistique** est utilisé pour estimer la probabilité qu'une donnée appartienne à une classe spécifique en fonction de ses attributs indépendants. Il modélise la relation entre les variables d'entrée et de sortie en utilisant une fonction logistique pour estimer la probabilité de la variable de sortie, qu'elle soit vraie ou fausse. La prédiction de la classe de la donnée est ensuite effectuée en se basant sur cette probabilité (Wright, 1995). **L'algorithme *Naïve Bayes*** est basé sur le théorème de Bayes, qui suppose que, pour une même classe, l'existence d'un attribut est indépendante de l'existence d'autres attributs. Cet algorithme calcule la probabilité que la donnée appartienne à une situation spécifique en

prenant en compte les interdépendances entre les attributs (WEBB et al., 2010).

L'algorithme *Linear Discriminant Analysis* est une technique d'apprentissage supervisé qui vise à séparer les données en utilisant des combinaisons linéaires de multiples variables indépendantes. L'objectif de l'algorithme est de trouver un discriminant linéaire qui maximise la séparation entre les différentes classes de données. Il cherche à projeter les données dans un nouvel espace de dimension inférieure tout en préservant au mieux la structure discriminante des données (BALAKRISHNAMA et al., 1998). Enfin, **l'algorithme SVM**, développé dans les années 1990, est une méthode d'apprentissage supervisé qui permet de séparer les données en utilisant des frontières de décision de telle sorte que la distance entre les données des différentes classes et la frontière soit maximale. Bien que cette notion de frontière ne soit applicable que lorsque les données sont linéairement séparables, l'algorithme SVM utilise des noyaux avec des fonctions mathématiques pour capturer des relations complexes entre les variables d'entrée et créer des frontières de décision non linéaires dans l'espace d'origine. Cette projection permet également de garantir une meilleure robustesse face au bruit et des données aberrantes (*outliers*) et donc de mieux généraliser les modèles aux nouvelles données. (BONACCORSO, 2017).

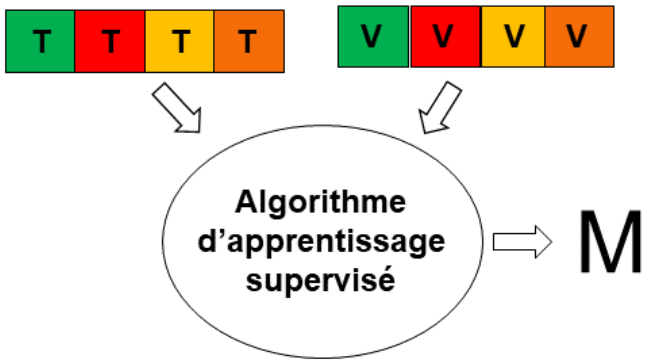

FIGURE 4.9 – Phase d'apprentissage

Source: traitement auteur

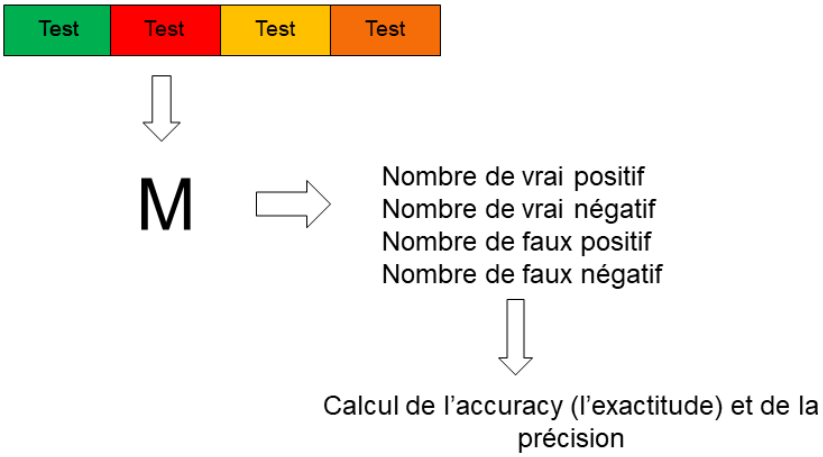

FIGURE 4.10 – Phase de test

Source: traitement auteur

4.2.5 Résultats

Chaque algorithme génère un modèle, et nous avons utilisé des jeux de données de test pour évaluer chaque modèle (voir Figure 4.10). Pendant cette phase de test, nous présentons au modèle des valeurs d'intervalle de trame de *beacon*, de RSSI, d'écart de numéros de séquence et de sous-type qui n'ont pas été utilisées lors de la phase d'entraînement. Le modèle est alors invité à prédire si ces valeurs correspondent à la situation normale, à une situation avec une attaque par déauthentification, une attaque par faux point d'accès ou une attaque par brouillage. Les nombres de vrais positifs, vrais négatifs, faux positifs et faux négatifs sont notés (PARIKH et al., 2008) et le score d'exactitude (*accuracy*), de précision, de rappel (*recall*) et le F1-score sont calculés. Le score de précision représente la proportion de vrais positifs parmi toutes les prédictions positives (vrais et faux positifs). Le score de *recall* représente la proportion de vrais positifs parmi toutes les situations réellement positives (vrais positifs et faux négatifs). Le F1-score est une moyenne harmonique de la précision et du *recall*, donnant une mesure globale de la performance du modèle(SOKOLOVA et al., 2009).

Lors de l'évaluation d'un modèle de classification, plusieurs métriques

peuvent être utilisées pour mesurer ses performances. Dans notre étude, nous avons choisi d'utiliser la précision multi-classes comme métrique de performance. Cela nous permet d'évaluer avec précision la capacité du modèle à classifier correctement les données dans les différentes classes (Grandini et al., 2020). En d'autres termes, la métrique de précision multi-classes permet d'évaluer la capacité du modèle à détecter chaque type d'attaque de manière distincte, sans les confondre avec d'autres attaques ou avec la situation normale. Par conséquent, si le modèle atteint des taux de précision proches de 100% pour toutes les classes, cela indique que le modèle est très performant dans la détection des différentes attaques sans les confondre entre elles ni avec la situation normale.

Il est toutefois important de noter que si une classe est déséquilibrée par rapport aux autres classes, cela pourrait biaiser le calcul de la précision. Par exemple, si une classe est disproportionnée en termes de données et mal classifiée dans toutes les classes, cela pourrait fausser la valeur de la précision. Comme indiqué précédemment, la distribution des données n'est pas uniforme pour toutes les classes présentes dans notre jeu de données. Par exemple, la classe correspondant à l'attaque par déauthentification est largement plus représentée que les autres classes, ce qui crée un déséquilibre dans la répartition des données et peut potentiellement influencer les résultats de précision obtenus.

Le F1-score est une métrique couramment utilisée pour détecter de telles anomalies. Dans un problème de classification multi-classes, il est recommandé d'utiliser les F1-scores micro et macro afin d'évaluer la performance globale du modèle. Dans le cadre de notre étude, nous nous concentrons principalement sur le calcul du F1-score micro, qui est une mesure pondérée de la précision et du rappel. Cette mesure s'avère particulièrement pertinente lorsque les classes présentent des déséquilibres significatifs en termes de taille (Grandini et al., 2020). Afin de ne pas trop surcharger les tableaux de résultats, nous ne présenterons que les F1-scores micro pour les deux meilleurs algorithmes et lorsque

le trafic est moyen.

Les algorithmes disposent également d'hyperparamètres qui peuvent être utilisés pour influencer le comportement ou la précision du modèle. Dans notre étude, nous avons testé les différents hyperparamètres proposés dans la bibliothèque Skicit-learn de Python en utilisant un jeu de données de validation, dans le but de déterminer les meilleurs paramètres pour chaque algorithme. Pour l'algorithme KNN, nous avons choisi de fixer le paramètre k à 5, car nous disposons de quatre attributs et qu'il est recommandé de définir k égal au nombre d'attributs (ou de dimensions) + 1. Pour l'algorithme Random Forest, nous avons décidé de fixer le nombre d'estimateurs à 100 pour augmenter le nombre d'arbres dans la forêt et ainsi améliorer la précision et l'exactitude. Pour les autres paramètres, nous avons choisi d'utiliser entropy comme *separateur*, car c'est une meilleure option pour les données non continues, et nous avons défini le paramètre *min_samples_leaf* à 3 pour élaguer (tailler) les arbres et éliminer le sur-apprentissage du modèle.

Pour l'algorithme CART, nous avons également choisi entropy comme *splitter* et avons défini le paramètre *min_samples_leaf* à 3 pour les mêmes raisons. Pour la régression logistique, nous avons opté pour le *solver* linéaire et la méthode One-Versus-Rest comme *multi_class*, car nos données contiennent plusieurs classes (ou situations) et nous avons constaté qu'un problème de classification binaire par classe fournit de meilleurs résultats de prédiction. Pour les algorithmes de régression logistique et Gaussian Naive Bayes, nous avons choisi le *solver* linéaire et la méthode One-Versus-Rest comme *multi_class* pour les mêmes raisons que pour l'algorithme CART. Enfin, pour l'analyse discriminante linéaire, nous avons sélectionné la décomposition en valeurs uniques comme *solver*, car il ne calcule pas la matrice de covariance et peut gérer plusieurs attributs.

Les résultats de détection pour les trois attaques avec un trafic moyen entre le client et le serveur sont indiqués dans le tableau 4.3. La quantité de données

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random forest	99.93%	100%	68.27%	99.82%	86.82%	70%
KNN (k=5)	99.84%	99.80%	62.68%	85.33%	93.14%	68.49%
CART	99.90%	100%	55.59%	81.02%	99.12%	68.05%
SVM	95.74%	100%	61.36%	90.80%	100%	68.36%
Logistic Regression	99.84%	99.80%	62.68%	85.33%	93.14%	68.49%
Naives Bayes	99.99%	97.20%	53.28%	44.54%	46.88%	51.16%
Linear Discriminant Analysis	100.0%	82.67%	54.79%	58.77%	72.06%	37.37%

TABLE 4.3 – Taux de précision pour détecter les attaques lorsque le trafic est moyen en utilisant les trames de gestion

***Le F1-score micro** des algorithmes les plus performants (KNN et Random Forest) est de 0.978 et 0.983 respectivement. Celui-ci confirme qu'il n'y a pas de déséquilibre de classement dans ces modèles.*

Source: traitement auteur

utilisée pour créer le modèle en fonction des différentes situations est résumée dans la figure 4.1. Les scores de précision les plus élevés ont été obtenus pour les modèles créés par les algorithmes *Random Forest*, SVM et KNN. La précision pour détecter l'attaque par déauthentification (TP_{Deauth}) et par faux point d'accès (TP_{FA}) est très élevée. Concernant l'attaque par brouillage, la précision pour détecter les attaques de brouillage de niveau moyen et faible est élevée, mais la détection de l'attaque de brouillage fort est plus faible. Les deux meilleurs scores de précision pour les attaques par déauthentification et par faux point d'accès ont été obtenus avec les algorithmes *Random Forest* et *Linear Discriminant Analysis*.

Concernant l'attaque par brouillage, les modèles créés par les algorithmes *Random Forest* et SVM sont les plus précis pour détecter les attaques de brouillage de niveau faible et moyen. En analysant la matrice de confusion, nous remarquons que, pour la détection de l'attaque de brouillage, les algorithmes

de classification, dans la majorité des cas, confondent la situation de brouillage avec la situation normale. En calculant la précision de détection de la situation normale, nous remarquons qu'elle est assez basse ($<= 70\%$), ce qui signifie que le modèle créé par l'algorithme *Random Forest*, par exemple, va considérer qu'il y a une attaque dans 30% des cas alors qu'en réalité, il n'y en avait pas. Toutefois, dans la détection de cyberattaques, la présence de faux négatifs pour la situation normale est moins grave que s'il y avait des faux négatifs pour les situations avec attaque.

Après ces résultats préliminaires prometteurs, il convient d'étudier les cas particuliers et les formes avancées d'attaques.

4.3 Conclusion

Dans ce chapitre, nous avons analysé les différentes informations présentes dans les captures pour trouver des indicateurs permettant de détecter chacune des trois attaques. Nous avons étudié deux approches de détection : une approche basée sur la détermination de seuils de détection et une autre utilisant des algorithmes de classification. En raison de la non-linéarité de la répartition de nos données, nous avons finalement opté pour l'approche basée sur les algorithmes de classification. Avec cette dernière approche, en particulier avec l'algorithme *Random Forest*, nous avons obtenu des résultats de précision de détection très satisfaisants pour les attaques par déauthentification, par faux point d'accès, ainsi que pour l'attaque de brouillage avec une puissance moyenne ou faible.

Nous devons désormais nous concentrer sur plusieurs axes d'amélioration, notamment l'amélioration de la détection de l'attaque par brouillage de forte puissance, la prise en compte des variations de trafic et leur impact sur les performances des modèles, l'étude de la détection des attaques sur la bande

5 GHz, ainsi que l'exploration de la détection de formes avancées d'attaques telles que les faux points d'accès fantômes et le cumul des attaques.

Chapitre 5

Cas particuliers et formes d'attaques avancées

Dans le chapitre précédent, nous avons procédé à l'analyse des captures obtenues lors de nos expérimentations. Nous avons filtré ces captures afin de ne considérer que les trames de gestion, puis nous avons présélectionné un ensemble d'attributs : l'intervalle entre deux trames, l'écart entre les numéros de séquence de deux trames consécutives, la puissance du signal de la trame reçue et le sous-type de trame. Nous avons étudié l'évolution de ces attributs en fonction des différentes situations, ce qui nous a permis de déterminer manuellement des seuils de détection (ou indicateurs) en dessous ou au-dessus desquels nous pouvons considérer la présence d'une attaque. Nous avons ensuite implémenté une première version d'un système de détection d'intrusion en Python, en nous appuyant sur ces seuils.

Toutefois, comme la répartition des données des différentes situations n'est pas séparable linéairement, la méthode de détection par détermination manuelle de seuils n'est pas optimale et ne permet pas d'obtenir les meilleurs résultats de précision. En effet, cette méthode peut générer beaucoup d'erreurs de détection. Pour avoir une détection plus efficace et plus précise, nous avons

choisi de nous orienter vers une détection utilisant des algorithmes de classification. Nous avons utilisé plusieurs algorithmes de classification et retenu uniquement ceux qui ont les meilleurs résultats. Nous avons constaté de très bons résultats de détection pour les attaques par faux point d'accès et par déauthentification. Concernant l'attaque par brouillage, la précision de détection dépend de la puissance du signal de brouillage. Les algorithmes sont capables de détecter avec une bonne précision les situations avec l'attaque par brouillage de puissance moyenne et faible, mais ont plus de difficultés à détecter la situation avec l'attaque par brouillage de puissance forte (environ 68 % au maximum lorsqu'il y a un trafic moyen).

Dans ce chapitre, nous étudions la prise en compte de plusieurs niveaux de trafic. Nous avons présenté, jusqu'à présent, des résultats en considérant un trafic moyen de 50 Mb/s entre le client et le serveur. Nous cherchons, dans un premier temps, à étudier l'impact des variations du trafic sur les résultats de détection, afin de savoir si la précision de détection diminue ou augmente lorsque le trafic varie. Dans un deuxième temps, nous cherchons à améliorer la détection de l'attaque par brouillage de forte puissance en étudiant les possibilités d'utilisation de radios logicielles et de prendre en compte les trames de *beacon* d'un point d'accès éloigné. Dans un troisième temps, nous prenons en compte le cas particulier des transmissions sur la bande 5 GHz. Sur cette bande, tous les types de trames sont émis en OFDM. Il convient donc d'analyser si cela peut avoir une incidence sur les résultats de détection. Dans un quatrième temps, nous étudions la détection de l'attaque par faux point d'accès fantôme en utilisant les trames de contrôle. Cette détection sur les trames de contrôle permet de détecter ce type de faux point d'accès qui n'émet pas de trame de *beacon*. Enfin, nous étudions la détection d'attaques cumulées, à savoir lorsque l'attaque par faux point d'accès est combinée soit avec une attaque par déauthentification, soit avec une attaque par brouillage.

5.1 Prise en compte des différents niveaux de trafic

Nous avons déjà présenté les résultats de détection des trois attaques dans des configurations avec trafic moyen (50 Mb/s). Il faut à présent également considérer les autres variations, notamment les cas où le trafic est faible (1 Mb/s), intense (100 Mb/s) ou même inexistant. Dans cette section, nous allons présenter, par ordre chronologique, les résultats de détection pour l'absence de trafic, le trafic léger et le trafic intense.

5.1.1 Détection des attaques en l'absence de trafic

La détection des trois attaques en l'absence de trafic revêt une importance particulière pour deux raisons. D'une part, la majorité des travaux scientifiques portant sur la détection de ces attaques ne prennent pas en compte le cas où il n'y a pas de trafic. D'autre part, comme nous l'avons souligné dans le chapitre 2, la plupart des méthodes de détection de l'attaque par brouillage sont inefficaces lorsque le trafic est absent. L'absence de trafic correspond au cas où le client et le serveur ne transfèrent pas de données via l'application iPerf. En d'autres termes, il n'y a dans ce cas pas (ou très peu) d'émission de trames de données et de contrôle. Toutefois, l'émission de trames de gestion, notamment l'envoi de trames de beacon, se poursuit. Il convient de souligner que toutes les autres variables relatives à la configuration et aux attaques sont demeurées inchangées entre les différentes variations testées.

Les résultats sont présentés dans le tableau 5.1, qui indique les taux de précision en fonction des situations et des modèles créés par les algorithmes de classification sur la base d'un jeu de trames de gestion capturées pour chaque configuration. Le jeu de trames de gestion contient **19575** trames, dont **1161**

trames pour la situation normale, **1852** trames pour la situation avec l'attaque par faux point d'accès, **14932** trames pour la situation avec l'attaque par déauthentification, **565** trames pour la situation avec brouillage de puissance forte, **1065** trames pour la situation avec brouillage de puissance moyenne et **1157** trames pour la situation avec brouillage de puissance faible. Le tableau 5.1 montre que, en l'absence de trafic, les algorithmes de classification ont des résultats plus précis.

L'algorithme *Random Forest*, par exemple, a augmenté en précision dans la détection des attaques par brouillage de puissance forte et faible. En outre, sa précision pour détecter la situation normale a également augmenté de 21,30%. Un meilleur taux de précision pour la situation normale indique qu'il y aura moins de fausses alertes (erreurs de détections). Cette amélioration peut s'expliquer par le fait que la détection des attaques se base principalement sur les trames de gestion, en particulier les trames de *beacon* qui sont régulièrement envoyées même en l'absence de trafic. La norme IEEE 802.11 permet aux points d'accès de différer et d'envoyer des trames de *beacon* de manière groupée lorsqu'il y a du trafic. Lorsqu'il n'y a pas de trafic, les trames peuvent être envoyées de manière plus précise, ce qui facilite la différenciation des situations et augmente la précision de détection.

5.1.2 Détection des attaques en présence d'un trafic léger

Dans nos expérimentations, le trafic léger correspond à un débit de 1 Mb/s. Ce débit est adapté pour envoyer rapidement des messages, des courriels et de petites photos, mais il ne permet pas de l'envoi et le téléchargement rapidement des fichiers volumineux. Le trafic de 1 Mb/s correspond à une utilisation d'environ 0,67 % du débit maximal théorique prévu par l'amendement IEEE 802.11n, qui est de 150 Mb/s.

Les résultats sont indiqués dans le tableau 5.2 sur la base d'un jeu de

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	100%	99.92%	78.24%	82.70%	99.51%	91.30%
KNN (k=5)	84.85%	99.84%	75.11%	81.27%	100%	92.39%
CART	77.46%	99.92%	56.80%	80.68%	100%	88.52%
SVM	100%	100%	74.68%	81.86%	71.06%	93.29%
Logistic Regression	99.84%	99.80%	62.68%	85.33%	93.14%	68.49%
Naives Bayes	58.24%	100%	65.87%	10%	100%	70.12%
Linear Discriminant Analysis	100%	99.45%	70%	0%	98.23%	70.60%

TABLE 5.1 – Précision de détection des attaques en l'absence de trafic

Source: traitement auteur

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.67%	100%	87.42%	97.74%	97.66%	95.94%
KNN (k=5)	99.34%	99.26%	84.13%	97.16%	97.55%	94.78%
CART	99.60%	100%	85.38%	96.29%	97.66%	94.04%
SVM	100%	100%	82.88%	98.17%	88.84%	96.21%
Logistic Regression	99.84%	99.80%	62.68%	85.33%	93.14%	68.49%
Naives Bayes	100%	100%	69.08%	39.06%	79.99%	83.62%
Linear Discriminant Analysis	100%	93.65%	42.72%	57.42%	84.80%	44.97%

TABLE 5.2 – Précision de détection des attaques en présence d'un trafic léger

Source: traitement auteur

trames de gestion contient **19819** trames de gestion dont **935** trames pour la situation normale, **2181** trames pour la situation avec l'attaque par faux point d'accès, **15123** trames pour la situation avec l'attaque par déauthentication, **629** trames pour la situation avec brouillage de puissance forte, **951** trames pour la situation avec brouillage de puissance moyenne et **1165** trames pour

la situation avec brouillage de puissance faible.

Nous constatons, dans le tableau 5.2, que de manière générale, lorsque le trafic est léger, les taux de précision pour la détection des attaques, comme dans la variation en l'absence de trafic, sont meilleurs que lorsque le trafic est moyen. Toutefois, pour la présente configuration, le modèle créé par l'algorithme SVM est généralement plus précis dans la détection des trois attaques, tandis que le modèle créé par l'algorithme *Random Forest* est plus précis pour l'attaque par brouillage avec forte puissance. Le modèle créé par l'algorithme SVM a également le taux de précision le plus élevé pour la situation normale. Il est cependant important de noter que ces résultats ont été obtenus pour un niveau de trafic très faible (1 Mb/s) et que, dans la plupart des réseaux, le trafic moyen est bien plus élevé. Après l'étude de la variation avec trafic léger, il convient de poursuivre l'analyse avec une variation impliquant un trafic plus intense.

5.1.3 Détection des attaques en présence d'un trafic Intense

Le trafic intense est un trafic de 100 Mb/s. Ce débit permet d'envoyer rapidement des messages, des courriels et des fichiers de taille conséquente (vidéos, photos, jeux...). Les résultats sont indiqués dans le tableau 5.3 sur la base d'un jeu de trames de gestion capturées pour la présente configuration. Le jeu de trames de gestion contient **19920** trames de gestion dont **935** trames pour la situation normale, **2247** trames pour la situation avec l'attaque par faux point d'accès, **15047** trames pour la situation avec l'attaque par déauthentication, **610** trames pour la situation avec brouillage de puissance forte, **1081** trames pour la situation avec brouillage de puissance moyenne et **1094** trames pour la situation avec brouillage de puissance faible.

Nous remarquons, comme indiqué dans le tableau 5.3, que les résultats sont

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	95.73%	99.71%	73.81%	88.18%	99.82%	74.2%
KNN (k=5)	94.55%	99.13%	67.72%	86.08%	100%	72.13%
CART	93.79%	99.81%	65.62%	85.74%	99.27%	70.84%
SVM	93.81%	99.77%	68.70%	90.25%	51.16%	66.49%
Logistic Regression	99.84%	99.80%	62.68%	85.33%	93.14%	68.49%
Naives Bayes	90.38%	100%	60.59%	53.69%	96.88%	53.21%
Linear Discriminant Analysis	89%	99.51%	45.12%	63.52%	90.19%	56.40%

TABLE 5.3 – Précision de détection des attaques en présence d'un trafic Intense

Source: traitement auteur

plus ou moins similaires à la variation avec trafic moyen, mais sont inférieurs aux variations sans trafic ou avec trafic léger. Dans cette variation, l'algorithme *Random Forest* est le plus performant.

En conclusion, nous pouvons affirmer que les différents algorithmes de classification sont capables de détecter les trois types d'attaques avec un taux de précision compris entre 86% et 100% pour les attaques par déauthentification, par faux point d'accès et par brouillage de faible et de moyenne puissance, et ce, pour les quatre variations de trafic étudiées (sans trafic, trafic léger, trafic moyen et trafic intense). Aujourd'hui, le trafic moyen sur les réseaux en France se situe autour de 50 Mb/s (NEVEU, 2019). Pour une représentation plus réaliste des résultats de détection, il convient de se référer aux résultats obtenus pour la variation avec un trafic moyen. En ce qui concerne l'attaque par brouillage de forte puissance, la précision de détection, toutes variations confondues, ne dépasse pas 79%. Il faut par conséquent essayer de trouver des méthodes pour améliorer la détection de l'attaque par brouillage de forte puissance.

5.2 Amélioration de la détection de l'attaque par brouillage avec forte puissance

Nous avons entrepris deux étapes pour améliorer la détection de l'attaque de brouillage à forte puissance. Tout d'abord, nous avons pris en compte et analysé les trames de *beacon* d'un deuxième point d'accès licite éloigné. Ensuite, nous avons essayé d'utiliser des équipements de laboratoire et des radios logicielles pour compléter les résultats qui, jusqu'à présent, étaient fondés sur les données de la couche 2 du modèle OSI en prenant également en compte les données de la couche 1.

5.2.1 Prise en compte de trames d'un point d'accès éloigné

Nous avons vu précédemment que les signaux émis en DSSS sont plus résistants à l'interférence et donc aux attaques par brouillage. Nous avons également constaté que les trames reçues dans la situation d'attaque par brouillage à forte puissance présentaient des caractéristiques similaires à celles reçues dans une situation normale. Aussi, d'après les matrices de confusion, les algorithmes de classification avaient tendance à classer les trames de l'attaque par brouillage à forte puissance comme étant de la situation normale, ce qui entraînait une précision de détection relativement faible pour cette situation (environ 68 % lorsqu'il y avait un trafic moyen).

Pour améliorer la détection de l'attaque par brouillage et éviter ces erreurs de classement, nous avons essayé de changer, de point d'accès de référence et à étudier l'impact de l'attaque par brouillage sur les trames de *beacon* émises par un autre point d'accès licite. Lorsque les réseaux Wi-Fi sont déployés dans des zones denses, il est courant d'utiliser plusieurs points d'accès Wi-Fi pour couvrir la zone. Ces points d'accès appartiennent tous au même réseau et doivent

émettre des trames de *beacon* portant le même nom de réseau, mais avec des adresses MAC différentes. Ces points d'accès ne sont pas des faux points d'accès (qui eux copient l'adresse MAC du point d'accès ciblé et sont généralement proches de celui-ci), mais des points d'accès licites placés à différents endroits pour étendre la couverture du réseau. Selon la norme IEEE 802.11, ces points d'accès devraient de préférence utiliser des canaux différents et être situés sur des canaux sans recouvrement de la bande 2.4 GHz.

Cependant, dans la pratique, cela n'est pas toujours possible et plusieurs points d'accès peuvent se trouver sur le même canal. Lorsque cela se produit, les points d'accès et les équipements connectés ne se causent pas systématiquement des interférences parce que les équipements écoutent le canal avant d'émettre. Cela a toutefois pour conséquence de réduire le débit du canal, car le temps disponible pour la communication doit être partagé entre plus d'équipements. Pour améliorer la détection de l'attaque par brouillage avec forte puissance, nous avons ajouté un deuxième point d'accès dans nos quatre configurations. Ce deuxième point d'accès fonctionne sur le même canal, mais est positionné en bordure du canal et a pour fonction d'étendre le réseau. Comme indiqué sur la Figure 5.1, ce point d'accès est à 714,2 cm du client. Les autres paramètres de la configuration restent inchangés.

Comme les trames de *beacon* sur la bande 2.4 GHz sont émises en utilisant l'amendement IEEE 802.11 b qui prévoit une transmission à bas débit sur une plus grande portée, il est toujours possible pour l'Observateur de recevoir les trames de *beacon* de ce point d'accès éloigné. Le deuxième point d'accès aurait pu être mis en fonctionnement sur un autre canal, mais cela aurait nécessité l'utilisation d'une deuxième carte Wi-Fi Pcap pour capturer les trames du second point d'accès. En effet, une carte Wi-Fi fonctionne sur un canal donné et ne peut pas simultanément capturer des trames sur deux canaux. Il faut soit une deuxième carte, soit faire une capture en boucle sur le canal 13, puis changer le canal de fonctionnement de la carte pour le mettre sur le canal du second point d'accès afin de capturer les trames, avant de remettre la carte

sur le canal 13. Dans les deux cas, cela aurait également posé des problèmes de synchronisation. Pour éviter ces problèmes, nous avons choisi d'ajouter le deuxième point d'accès en bordure du canal 13, ce qui nous a permis de capturer les trames des deux points d'accès avec une seule carte et d'étudier l'efficacité de cette méthode de détection.

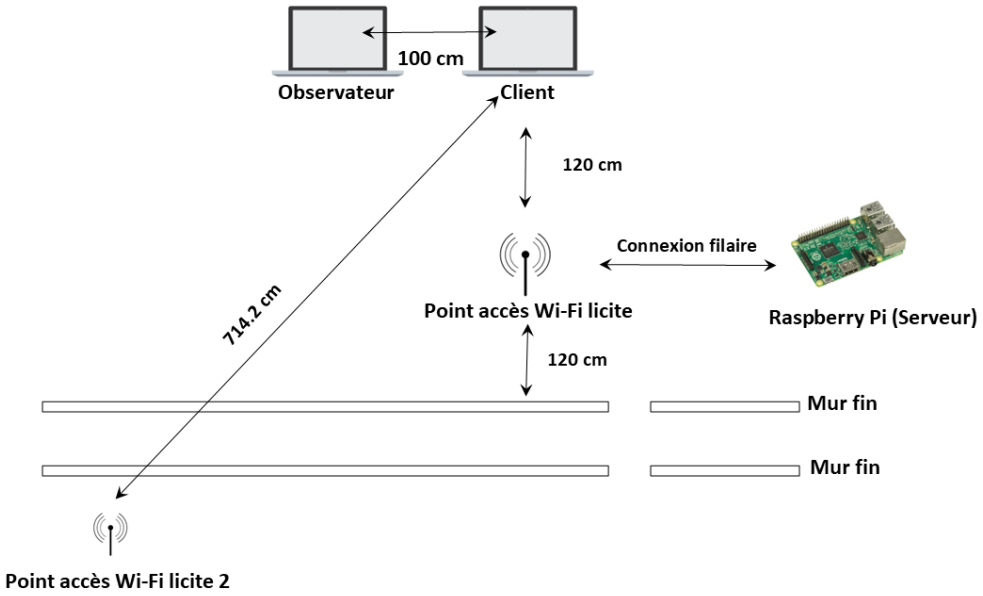

FIGURE 5.1 – Nouvelle configuration - ex : situation normale

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay)

L'idée en ajoutant un deuxième point d'accès est d'étudier l'impact de l'attaque par brouillage sur les trames de *beacon* émises à plus grande distance avec un rapport signal à interférence plus faible. Rien que dans la situation normale, nous remarquons après les expérimentations, qu'il y a une différence de 20 dBm en puissance entre les trames de *beacons* du premier point d'accès et celles du deuxième point d'accès. Par conséquent, l'impact de l'attaque par brouillage devrait être plus marqué sur ces trames de *beacon*.

Il est important de préciser que les attaques par faux point d'accès et par déauthentification sont effectuées en utilisant l'adresse MAC du premier point d'accès licite. Ces attaques n'affecteront donc pas les trames du deuxième point

d'accès. En revanche, l'attaque par brouillage impacte également les trames du deuxième point d'accès, car elle vise à dégrader le rapport signal à interférence sur un canal et les deux points d'accès fonctionnent sur le même canal. Par conséquent, pour détecter les trois types d'attaques, il est nécessaire de créer deux modèles : un premier modèle qui détectera les attaques par faux point d'accès et par déauthentification en analysant les trames du premier point d'accès licite, et un deuxième modèle qui détectera l'attaque par brouillage et ses variations en analysant les trames du deuxième point d'accès licite.

Les trames capturées ont été filtrées en fonction de l'adresse MAC des points d'accès dans une même capture avant d'être utilisées pour entraîner les modèles. Le premier modèle est capable de détecter sans difficulté les attaques par faux point d'accès et par déauthentification. Pour des raisons de lisibilité, nous présentons uniquement les résultats du deuxième modèle dans les tableaux suivants (5.4, 5.5, 5.6, 5.7). Ces tableaux présentent les taux de précision en fonction des situations et des modèles créés par les algorithmes de classification sur la base d'un jeu de trames de gestion capturées pour chaque configuration.

Pour le tableau 5.4, le jeu de donnée contient **2319** trames de gestion dont **1147** trames pour la situation normale, **12** trames pour la situation avec brouillage de puissance forte, **1160** trames pour la situation avec brouillage de puissance moyenne et **1108** trames pour la situation avec brouillage de puissance faible.

Pour le tableau 5.5, le jeu de donnée contient **3100** trames de gestion dont **1156** trames pour la situation normale, **17** trames pour la situation avec brouillage de puissance forte, **912** trames pour la situation avec brouillage de puissance moyenne et **1015** trames pour la situation avec brouillage de puissance faible.

Pour le tableau 5.6, le jeu de donnée contient **1449** trames de gestion dont **908** trames pour la situation normale, **0** trames pour la situation avec brouillage de puissance forte, **541** trames pour la situation avec brouillage de puissance

Algorithmes	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	100%	84.43%	97.75%	97.69%
KNN (k=5)	100%	84.65%	97.99%	97.53%
CART	100%	83.60%	97.60%	97.54%
SVM	-	83.16%	86.07%	99.08%
Logistic Regression	-	83.16%	86.07%	99.09%
Naives Bayes	13.04%	81.83%	92.74%	92.37%
Linear Discriminant Analysis	100%	80.96%	92.84%	78.91%

TABLE 5.4 – Précision de détection de l'attaque par brouillage en l'absence de trafic en utilisant les trames de beacon du second point d'accès licite

Les exécutions des algorithmes SVM et Régression Logistique ont rencontré des erreurs lors de la situation avec un brouillage intense. Il est, par conséquent, important de ne pas comparer ces valeurs avec les autres valeurs du tableau. Ces résultats sont uniquement fournis à titre informatif.

Source: traitement auteur

moyenne et **775** trames pour la situation avec brouillage de puissance faible.

Pour le tableau 5.7, le jeu de donnée contient **1286** trames de gestion dont **759** trames pour la situation normale, **0** trames pour la situation avec brouillage de puissance forte, **527** trames pour la situation avec brouillage de puissance moyenne et **723** trames pour la situation avec brouillage de puissance faible.

Ces résultats mettent en évidence plusieurs aspects importants. Tout d'abord, les résultats de détection pour la variation de trafic léger sont équivalents à la variation sans trafic. Aussi, dans la situation sans trafic, la précision de la détection de l'attaque par brouillage avec forte puissance est de 100%, ce qui représente une amélioration de plus de 30% par rapport au modèle qui détecte cette même attaque en utilisant les trames du point d'accès licite 1. Autre point important, la précision de détection de la situation normale est également très élevée. En effet, nous avons vu que l'attaque par brouillage avec forte puis-

Algorithmes	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	100%	92.96%	65.28%	89.99%
KNN (k=5)	100%	56.31%	98.48%	86.24%
CART	85.71%	89.82%	65.11%	90.38%
SVM	-	94.20%	62.19%	90.60%
Logistic Regression	-	94.20%	62.19%	90.60%
Naives Bayes	50%	89.37%	61.63%	81.61%
Linear Discriminant Analysis	100%	76.61%	62.10%	57.52%

TABLE 5.5 – Précision de détection d'attaques en présence de trafic léger en utilisant les trames de gestion du second point d'accès

Les exécutions des algorithmes SVM et Régression Logistique ont rencontré des erreurs lors de la situation avec un brouillage intense. Il est, par conséquent, important de ne pas comparer ces valeurs avec les autres valeurs du tableau. Ces résultats sont uniquement fournis à titre informatif.

Source: traitement auteur

sance était jusqu'ici souvent confondue avec la situation normale. En analysant les trames du second point d'accès, nous constatons que cela n'est plus le cas. Enfin, nous notons que les trames dans des situations avec brouillage de puissance moyenne et faible sont souvent confondues, ce qui diminue la précision pour ces deux variations. Cependant, cette confusion est moins préoccupante qu'une confusion avec la situation normale.

Toutefois, en ce qui concerne les variations de trafic, lorsque le trafic est moyen ou intense dans le réseau du point d'accès 1, très peu de trames de *beacon* de deuxième point d'accès (une, deux voire aucune) sont reçues. Cela est dû au fait que les trames de *beacon* sont retardées lorsqu'il y a du trafic sur le canal. Comme le point d'accès est éloigné, l'Observateur ne semble pas recevoir toutes les trames de *beacon* lorsqu'elles sont émises par bloc et d'un seul trait. Par conséquent, même si ce deuxième modèle de classification donne de bons résultats pour les variations sans trafic et avec trafic léger, il est inopérant pour

Algorithmes	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	-	96.90%	56.94%	99.58%
KNN (k=5)	-	44.57%	98.64%	88.14%
CART	-	93.65%	56.92%	99.37%
SVM	-	98.31%	56.79%	97.91%
Logistic Regression	-	98.31%	56.79%	97.91%
Naives Bayes	-	44.17%	93.14%	67.71%
Linear Discriminant Analysis	-	77.66%	56.63%	44.17%

TABLE 5.6 – Précision de détection des attaques en présence de trafic moyen en utilisant les trames de gestion du second point d'accès

***Le F1-score micro** des algorithmes les plus performants (KNN et Random Forest) est de 0.573 et 0.573 respectivement. Celui-ci confirme qu'il existe un déséquilibre de classification dans ces modèles, en particulier entre les variations de brouillage moyen et faible et qu'il existe des confusions entre ces classes.*

Source: traitement auteur

Algorithmes	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	-	97.97%	59.22%	98.76%
KNN (k=5)	-	92.54%	59.22%	89.49%
CART	-	97.61%	59.22%	99.24%
SVM	-	98.20%	59.04%	96.95%
Logistic Regression	-	98.20%	59.04%	96.95%
Naives Bayes	-	41.63%	96.41%	72.77%
Linear Discriminant Analysis	-	66.67%	58.97%	41.09%

TABLE 5.7 – Précision de détection des attaques en présence de trafic intense en utilisant les trames de gestion

Source: traitement auteur

les variations avec trafic moyen et intense en raison du manque de données disponibles.

Il faut, par conséquent, conclure que la meilleure solution serait de coupler le premier modèle, analysant les trames du premier point d'accès pour détecter les attaques par faux point d'accès, par déauthentification et par brouillage de moyenne ou faible puissance, avec un second modèle fondé sur une détection par seuil analysant les trames du second point d'accès pour détecter l'attaque par brouillage de forte puissance. Ce modèle pourra être déterminé par un seuil. Le seuil pour détecter cette attaque pourrait être, par exemple, basé sur le nombre de trames de *beacon* reçues. Si aucune trame ou seulement une ou deux trames du deuxième point d'accès licite ne sont reçues, cela pourra indiquer la présence de l'attaque par brouillage de forte puissance (pour les variations avec trafic moyen et intense). Ce seuil permettrait d'atteindre une détection de l'attaque avec un taux de précision de 100 %. Ainsi, en combinant les deux modèles, les trois types d'attaques et leurs variations peuvent être détectés avec une précision élevée.

Une autre méthode pour améliorer la détection de l'attaque par brouillage de forte puissance est de prendre en compte les données de la couche 1.

5.3 Prise en compte de données supplémentaires de couche 1

Comme nous l'avons vu dans l'état de l'art, plusieurs auteurs ont proposé de détecter les trois attaques sur la couche 1. Les informations disponibles sur la couche 1 et la couche 2 du modèle OSI ne sont pas identiques. Jusqu'à présent, nous avons utilisé des informations de l'entête de la couche 2 (adresse MAC, numéro de séquence...) et des informations de la couche 1 (par exemple, la

puissance du signal de la trame reçue) communiquées par le micrologiciel de la carte Wi-Fi au logiciel Wireshark dans un entête RadioTap. Ces informations sont des informations sur les signaux IEEE 802.11 détectés et décodés par la carte Wi-Fi. La carte Wi-Fi ne peut pas détecter les autres signaux éventuellement transmis sur le canal ; elle ne comprend que les signaux IEEE 802.11. Les captures de la couche 2 font donc abstraction des autres signaux transmis sur le canal. Les équipements de la couche 1 permettent d'obtenir des informations concernant tout type de transmission sur un canal donné. Pour obtenir ces informations, nous avons utilisé dans un premier temps un analyseur de spectre, puis dans un second temps, une radio logicielle.

5.3.1 Utilisation d'un analyseur de spectre

Dans notre configuration, nous avons ajouté un deuxième Observateur : un analyseur de spectre qui va nous permettre de capturer la puissance maximale émise sur une bande de fréquences. L'analyseur de spectre est mis en mode Keysight 89600 VSA avec les paramètres suivants : fréquence centrale de 2.45 GHz, largeur de bande de 160 MHz, résolution de la bande passante de 936 kHz et échelle de puissance de -30 dBm. Avec ces paramètres, nous avons utilisé un programme pour capturer dix mesures de niveaux de puissance maximale sur la bande de fréquence 2.4 GHz. Il convient de noter qu'une capture de niveaux de puissance peut générer rapidement une quantité importante de données. Ainsi, nous avons opté pour une mesure itérative (10 itérations) plutôt qu'une mesure sur une durée de temps (comme cela a été fait jusqu'à présent). Nous avons refait les expérimentations pour les différentes situations et variations, en effectuant à chaque fois une capture de deux minutes avec l'Observateur de la couche liaison de données et dix itérations de dix mesures de niveaux de puissance maximale avec l'Observateur de la couche 1 (voir Figure 5.2).

Nous avons ensuite analysé les données des différentes situations. Nous

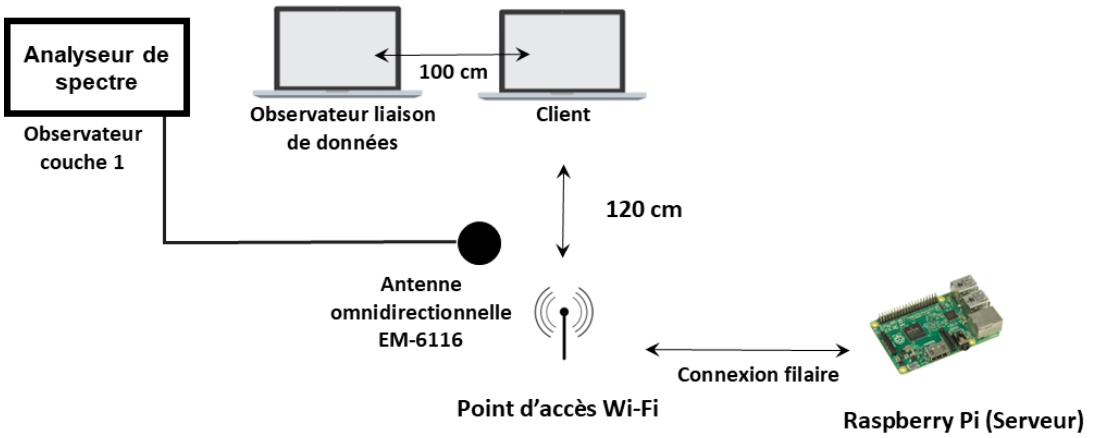

FIGURE 5.2 – L'ajout d'un Observateur couche 1 dans les expérimentations

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay)

avons tracé les niveaux maximaux de la moyenne de puissance en fonction de la fréquence. Dans la figure 5.3, nous remarquons qu'il y a un signal sur le canal 13 (2.45 GHz - 2.494 GHz) qui ressemble à un signal OFDM. Ce signal correspond à l'envoi de trames de données. Si nous observons maintenant la Figure 5.4, nous remarquons que le graphe ne ressemble pas au précédent. Sur cette figure, il y a un signal avec une puissance élevée qui est présent sur presque toute la bande. Ce signal est un signal de brouillage à balayage de fréquence à forte puissance. Nous remarquons bien sur cette même Figure que le signal est de forte puissance et qu'il a perturbé et arrêté l'envoi de trames de données. Sur la Figure 5.5, correspondant à la variation avec brouillage de moyenne puissance, la puissance moyenne du signal de brouillage sur la bande est plus basse. Enfin, la Figure 5.6 représente la situation avec un brouillage de faible puissance. Nous constatons sur cette Figure que la puissance moyenne du signal de brouillage est plus faible. Nous arrivons également à percevoir un semblant de signal OFDM sur le canal 13, ce qui correspond à un envoi de données en partie perturbé par le signal de brouillage.

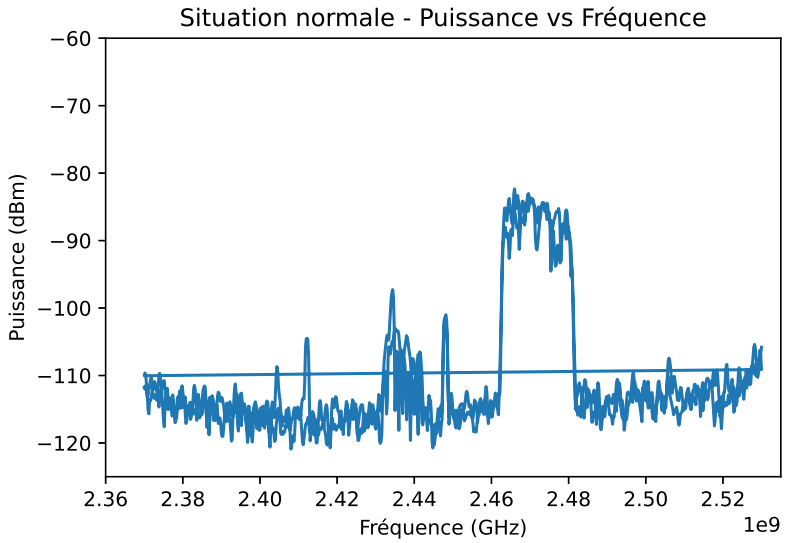

FIGURE 5.3 – Situation Normale

Source: traitement auteur : en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

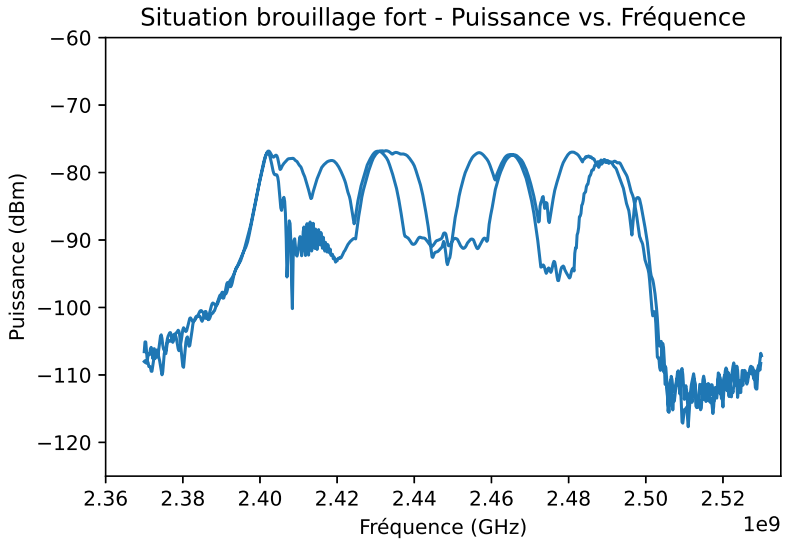

FIGURE 5.4 – Situation brouillage fort

Source: traitement auteur : en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

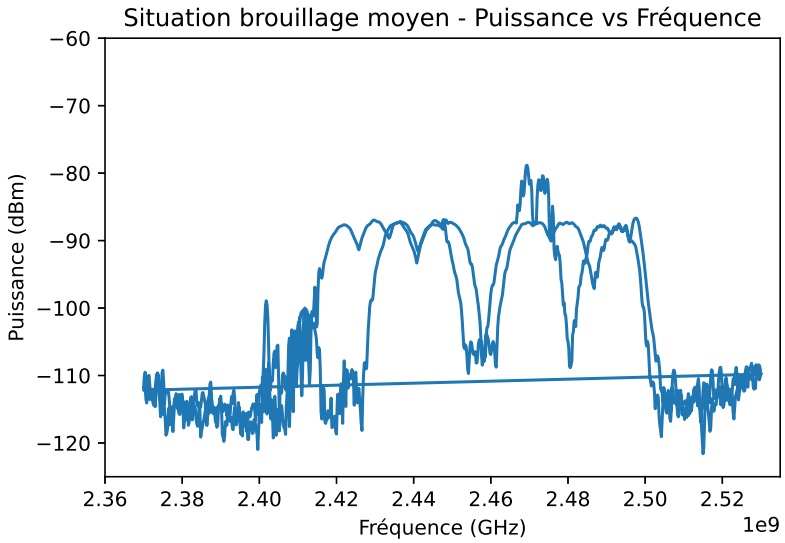

FIGURE 5.5 – Situation brouillage moyen

Source: traitement auteur : en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

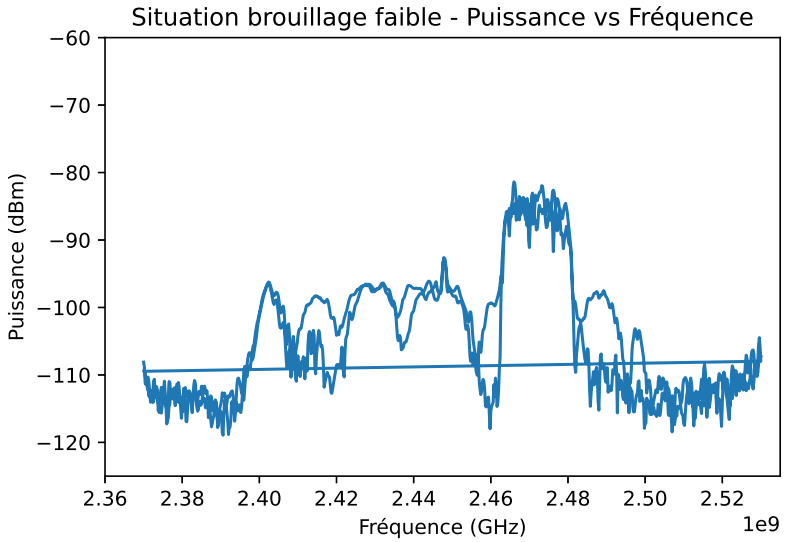

FIGURE 5.6 – Situation brouillage faible

Source: traitement auteur : en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

Dans la littérature scientifique, certains auteurs ont proposé de créer un modèle de détection d'attaque par brouillage à forte puissance en utilisant des données dérivées des niveaux de puissance maximale (voir Chapitre 2). Ces auteurs ont obtenu des taux de détection précis d'environ 96% (Villain, Deniau, Fleury et al., 2019), ce qui montre que l'utilisation d'analyseurs de spectre peut permettre de détecter avec une grande précision ce type d'attaque. Toutefois, notre objectif étant de mettre en place un système de détection d'intrusion à bas coût et facilement transportable, l'utilisation d'un analyseur de spectre n'est pas appropriée. En outre, la collecte des niveaux de puissance et des trames par deux équipements différents peut poser des problèmes de synchronisation temporelle.

En raison des limitations et des inconvénients, il est préférable d'opter pour l'utilisation de radios logicielles.

5.3.2 Utilisation des radios logicielles

Les radios logicielles sont des équipements de radiocommunication qui offrent une grande flexibilité grâce à leur capacité à être configurées à l'aide de filtres et de fonctions de traitement numérique du signal programmables sur des circuits numériques. Elles peuvent être utilisées comme émetteurs ou récepteurs et, contrairement à une carte Wi-Fi qui ne peut reconnaître et émettre que des signaux IEEE 802.11, une radio logicielle peut être configurée et programmée pour reconnaître ou émettre des signaux de différentes normes ou protocoles. Afin de reconnaître ou d'émettre un signal spécifique, il est toutefois nécessaire d'implémenter une fonction de reconnaissance et de décodage de protocole. Cette fonction peut être implémentée en utilisant plusieurs langages de programmation, tels que Python, ou en utilisant GNU Radio, un logiciel libre de droits conçu pour simplifier l'implémentation de fonctions de traitement du signal pour une utilisation avec des radios logicielles. GNU radio a pour objet

de simplifier l'implémentation des fonctions de traitement du signal.

Il est important de noter que toutes les radios logicielles n'ont pas les mêmes caractéristiques. Certaines radios logicielles peuvent être limitées en termes de taux d'échantillonnage, de puissance d'émission, de réception ou de bande de fréquences de fonctionnement (SADIKU et al., 2004). Bien que les radios logicielles soient généralement moins performantes que les analyseurs de spectre ou les générateurs de signal arbitraire, elles sont moins coûteuses et plus portables.

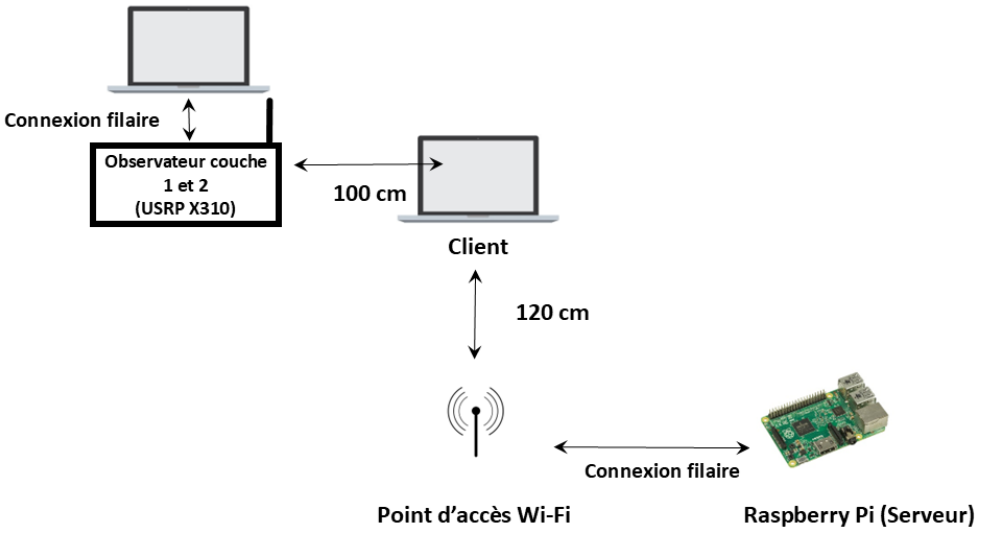

FIGURE 5.7 – Remplacement de l'observeur 2 par un observeur couche 1 et 2

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay)

Les radios logicielles sont de plus en plus utilisées pour la détection et la réalisation d'attaques (ACHAAL et al., 2022). Dans le cadre de notre travail, nous avons essayé d'utiliser une radio logicielle, la USRP X310, pour capturer des informations de la couche 1 et de la couche 2 du modèle OSI afin de détecter les trois types d'attaques. Pour les informations de la couche 1, nous cherchions à récupérer les moyennes des niveaux de puissance maximale émis sur un ensemble de fréquences. Pour les données de la couche 2, nous cherchions à récupérer les trames de gestion, de contrôle et de données. Pour capturer ces données, une chaîne capable de reconnaître et de décoder les signaux IEEE 802.11 en trames et de capturer des moyennes de niveaux de puissance maxi-

male est nécessaire. Nous avons utilisé la chaîne GNU Radio, un logiciel libre de droit implémenté par (Bloessl et al., 2013), pour la capture de trames (voir Figure 5.8) et l'émission de trames IEEE 802.11a/g/p.

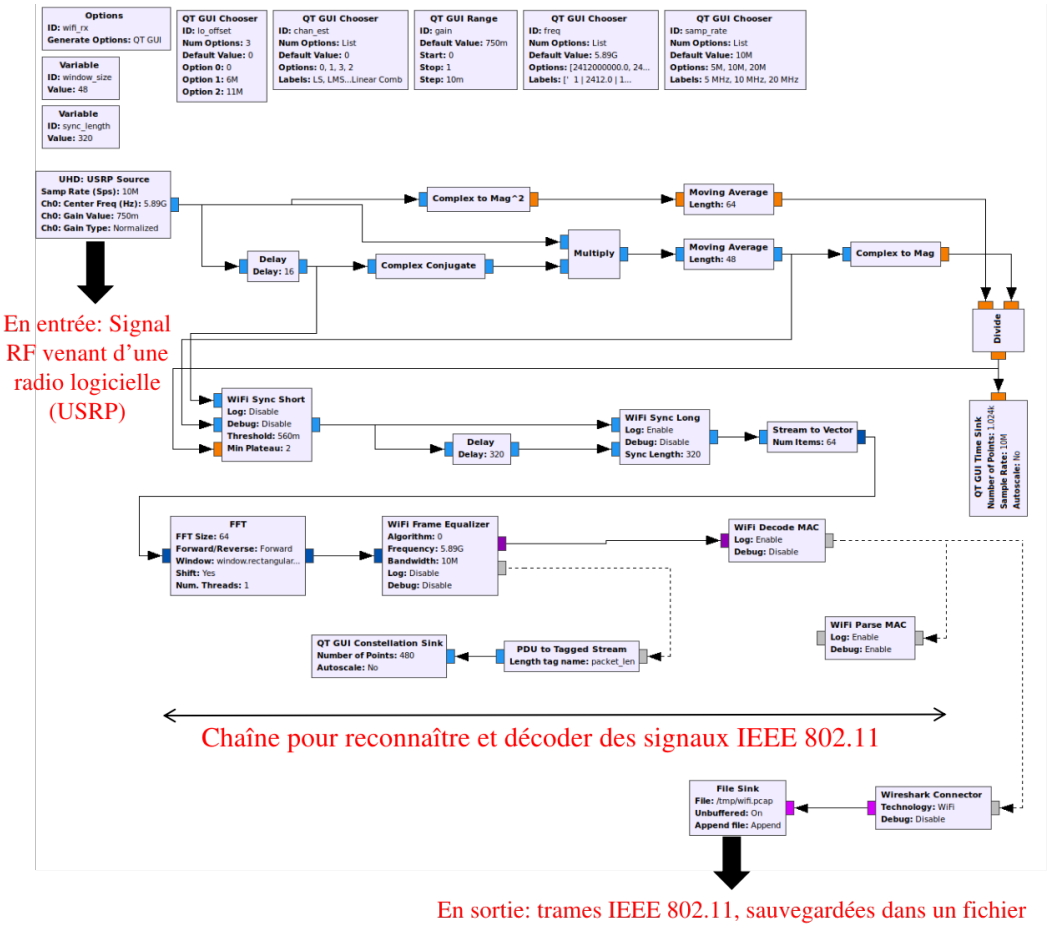

FIGURE 5.8 – Chaine de réception IEEE 802.11

Source: traitement auteur ; capture d'écran modifiée de la chaine de réception IEEE 802.11 disponible sur (Bastibl, 2015)

Toutefois, lors de nos tests avec cette chaîne, nous avons remarqué que le nombre de trames reçues n'était pas identique à celui obtenu avec l'utilisation d'une carte Wi-Fi. En effet, la chaîne était capable de reconnaître et de décoder seulement 1/10 des trames IEEE 802.11 émises sur le canal. Les auteurs de la chaîne ont indiqué que ce phénomène pouvait se produire en fonction du type de radio logicielle utilisée (Bloessl et al., 2013) et qu'ils avaient eux-mêmes reçu un nombre restreint de trames avec leur radio logicielle. Par ailleurs, lorsque nous avons utilisé les cartes Wi-Fi Pcap pour créer des attaques par déau-

thentification et par faux point d'accès, nous avons constaté que les trames émises par la carte Wi-Fi Pcap, qui ont une puissance légèrement inférieure, étaient très peu, voire pas du tout, détectées par la chaîne. En outre, le nombre de trames reçues variait considérablement entre deux expérimentations de la même situation. Pour ces raisons, nous avons conclu que cette chaîne ne pouvait pas être utilisée dans le cadre de nos expérimentations. Nous avons conclu qu'un travail important de traitement du signal serait nécessaire pour améliorer cette chaîne, l'étendre aux autres amendements de la norme IEEE 802.11 (802.11b/n, par exemple) et permettre une utilisation efficace avec différents types de radios logicielles.

Nous avons analysé, jusqu'à présent, les effets des attaques sur les transmissions IEEE 802.11 qui utilisent la bande de fréquences 2.4 GHz. Il convient désormais d'examiner spécifiquement les transmissions IEEE 802.11 sur la bande de fréquences 5 GHz.

5.4 Cas particulier du WI-Fi 5 GHz

Sur la bande 5 GHz, tous les signaux sont transmis en OFDM. Le signal OFDM est également moins résistant aux interférences. Par conséquent, nous devrions observer un impact plus important sur les trames de *beacon* émises sur la bande 5 GHz. Afin d'étudier cela, nous avons reproduit les expérimentations avec différentes configurations (voir exemple Figure 5.9), en apportant quelques légères modifications spécifiques aux transmissions sur la bande 5 GHz :

1. Les équipements ne fonctionnent plus sur le canal 13 (réservé à la bande 2.4 GHz), mais sur le canal 52.
2. Le taux d'échantillonnage du signal de brouillage est doublé pour atteindre 2 Gs/s.
3. Des câbles adaptés à la transmission sur la bande 5 GHz sont utilisés avec

le générateur de signal arbitraire pour générer le signal de brouillage.

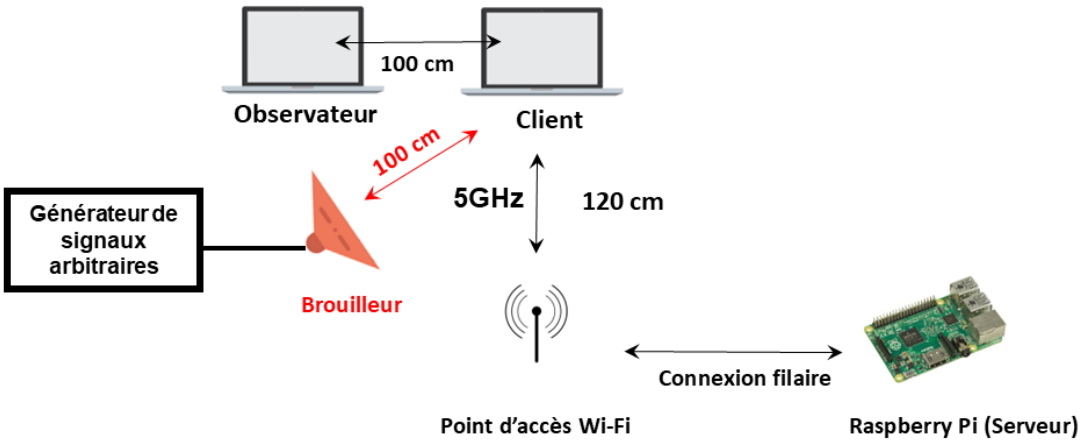

FIGURE 5.9 – Configuration - Par exemple : Situation brouillage

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay), antenne directionnelle (Smashicons, tirée de flaticon.com, license Flaticon)

En analysant les différents attributs, nous avons constaté deux différences par rapport aux transmissions sur la bande 2.4 GHz.

Premièrement, dans la situation où une attaque par brouillage de forte puissance est effectuée, très peu de trames de *beacon* sont reçues. Cela s'explique par le fait que les trames sont émises en OFDM, et sont affectées par le signal de brouillage, entraînant des pertes de trames. Bien que quelques trames de *beacon* soient reçues (une ou deux) sur les deux minutes d'expérimentation, cela ne suffit pas pour créer une boîte à moustaches et pour faire de l'apprentissage. Deuxièmement, contrairement à nos expérimentations sur la bande 2.4 GHz, nous avons remarqué que l'attribut puissance du signal de la trame reçue évolue différemment. En examinant les boîtes à moustaches, nous avons observé que les boîtes correspondant à une attaque par brouillage de moyenne et faible puissance étaient plus facilement distinguables de la situation normale qu'avec les données sur les transmissions sur la bande 2.4 GHz. Enfin, nos expérimentations ont montré que lorsqu'un signal est émis en OFDM, la carte Wi-Fi peut mesurer plus facilement le rapport signal à interférence du canal. En suivant la même procédure d'apprentissage et de test, nous avons utilisé ces

données pour créer et évaluer des modèles capables de détecter les trois types d'attaques sur la bande 5 GHz.

Les résultats sont présentés dans les tableaux 5.8 à 5.11. Ces tableaux présentent les taux de précision en fonction des situations et des modèles créés par les algorithmes de classification sur la base d'un jeu de trames de gestion capturées pour chaque configuration.

Pour le tableau 5.8, le jeu de donnée contient **21879** trames de gestion dont **1172** trames pour la situation normale, **2331** trames pour la situation avec attaque par faux point, **16032** trames pour la situation avec attaque par déauthentification, **0** trame pour la situation avec brouillage de puissance forte, **1172** trames pour la situation avec brouillage de puissance moyenne et **1172** trames pour la situation avec brouillage de puissance faible.

Pour le tableau 5.9, le jeu de donnée contient **21438** trames de gestion dont **1167** trames pour la situation normale, **2316** trames pour la situation avec attaque par faux point, **15914** trames pour la situation avec attaque par déauthentification, **0** trame pour la situation avec brouillage de puissance forte, **873** trames pour la situation avec brouillage de puissance moyenne et **1168** trames pour la situation avec brouillage de puissance faible.

Pour le tableau 5.10, le jeu de donnée contient **20912** trames de gestion dont **1018** trames pour la situation normale, **2001** trames pour la situation avec attaque par faux point, **16018** trames pour la situation avec attaque par déauthentification, **1** trame pour la situation avec brouillage de puissance forte, **890** trames pour la situation avec brouillage de puissance moyenne et **984** trames pour la situation avec brouillage de puissance faible.

Pour le tableau 5.11, le jeu de donnée contient **19769** trames de gestion dont **478** trames pour la situation normale, **1984** trames pour la situation avec attaque par faux point, **15968** trames pour la situation avec attaque par déauthentification, **1** trame pour la situation avec brouillage de puissance forte,

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	100%	100%	-	96.55%	91.58%	74.70%
KNN (k=5)	99.34%	98.50%	-	95.99%	91.71%	75.28%
CART	100%	100%	-	92.18%	91.14%	74.57%
SVM	100%	100%	-	98.91%	76.19%	74.77%
Logistic Regression	99.84%	99.80%	-	85.33%	93.14%	68.49%
Naives Bayes	100%	100%	-	51.85%	60.55%	75.54%
Linear Discriminant Analysis	100%	94.81%	-	60.60%	75.49%	34.04%

TABLE 5.8 – Précision de détection d'attaques en l'absence de trafic - cas particulier 5 GHz

Source: traitement auteur

865 trames pour la situation avec brouillage de puissance moyenne et **473** trames pour la situation avec brouillage de puissance faible.

En ce qui concerne les résultats, nous avons observé que les modèles de détection *Random Forest* et KNN ont les meilleurs résultats. Tout comme pour les transmissions sur la bande 2.4 GHz, les précisions de détection diminuent lorsque le débit de trafic augmente. Nous avons remarqué que les précisions de détection pour les variations de brouillage de moyenne et faible puissance étaient meilleures que les valeurs correspondantes pour les transmissions sur la bande 2.4 GHz. En outre, sur la bande 5 GHz, les trames ne sont pas reçues lorsque la puissance du signal est forte, ce qui rend la détection par algorithme de classification inopérante pour cette variation.

Comme pour la détection utilisant les trames d'un second point d'accès, pour résoudre ce problème, il faut créer deux modèles. Le premier modèle détecte les attaques par déauthentification, par faux point d'accès et par brouillage de moyenne et faible puissance. Ce premier modèle devrait être couplé avec un deuxième modèle qui détecte l'attaque par brouillage de forte puis-

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	96.43%	100%	-	95.84%	100%	99.32%
KNN (k=5)	88.33%	99.57%	-	97.07%	99.48%	99.32%
CART	88.52%	99.39%	-	98.85%	100%	98.17%
SVM	90.91%	100%	-	94.44%	85.76%	99.48%
Logistic Regression	99.84%	99.80%	-	85.33%	93.14%	68.49%
Naives Bayes	92.86%	99.56%	-	98.81%	97.95%	96.26%
Linear Discriminant Analysis	96.08%	99.22%	-	85.96%	67.62%	97.11%

TABLE 5.9 – Précision de détection des attaques en présence de trafic léger - cas particulier 5 GHz

Source: traitement auteur

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.93%	93.32%	-	82.03%	95.88%	87.94%
KNN (k=5)	99.87%	92.32%	-	86.08%	91.38%	84.51%
CART	99.93%	93.22%	-	76.77%	92.61%	86.67%
SVM	100%	70.82%	-	81.49%	94.69%	87.58%
Logistic Regression	99.84%	99.80%	-	85.33%	93.14%	68.49%
Naives Bayes	99.93%	91.12%	-	42.27%	72.99%	58.81%
Linear Discriminant Analysis	99.93%	55.89%	-	54.46%	82.1%	65.77%

TABLE 5.10 – Précision de détection des attaques en présence de trafic moyen - cas particulier 5 GHz

Le F1-score micro *des algorithmes les plus performants (KNN et Random Forest) est de 0.965 et 0.976 respectivement. Celui-ci confirme qu'il n'y a pas de déséquilibre de classification dans ces modèles.*

Source: traitement auteur

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	97.30%	99.45%	-	92%	97.93%	86.42%
KNN (k=5)	99.60%	98.89%	-	87.01%	95.93%	83.70%
CART	97.30%	98.90%	-	93.81%	97.14%	80%
SVM	93.81%	99.77%	-	90.25%	51.16%	66.49%
Logistic Regression	99.84%	99.80%	-	85.33%	93.14%	68.49%
Naives Bayes	98.57%	83.64%	-	79.69%	96.19%	75.86%
Linear Discriminant Analysis	100%	98.31%	-	91.04%	64.86%	89.32%

TABLE 5.11 – Précision de détection d'attaques en présence de trafic intense - cas particulier 5 GHz

Source: traitement auteur

sance en fonction d'un seuil. Le seuil peut être déterminé par rapport à un attribut, tel que le nombre de trames reçues. Si le nombre de trames reçues est nul, cela correspond à la situation de ladite attaque.

Il convient, à présent, de considérer le cas de l'attaque par faux point d'accès fantôme.

5.4.1 Détection des attaques par faux points d'accès fantômes

Les points d'accès émettent régulièrement des trames de *beacon* pour informer les clients environnants de leur présence. Les clients utilisent les informations contenues dans les trames de *beacon* pour effectuer une demande d'authentification et d'association auprès du point d'accès. Nous avons utilisé, jusqu'à présent, les trames de *beacon* pour détecter l'attaque par faux point d'accès. Toutefois, nous pensons qu'il pourrait exister une forme spéciale de cette attaque : le faux point d'accès fantôme. Ce dernier n'émettrait pas de

trames de *beacon* et essayerait directement de répondre aux demandes d'authentification et d'association des clients. L'attaque par faux point d'accès fantôme est une attaque théorique que nous n'avons pas pu mettre en œuvre, car une carte permettant d'effectuer ce type d'attaque n'est pas encore commercialisée. Néanmoins, nous craignons que cette attaque puisse devenir courante à l'avenir, ce qui nécessite la mise en place de moyens pour la détecter. En effet, la majorité des méthodes de détection des attaques par faux point d'accès reposent sur les trames de *beacon*, ce qui devrait inciter les attaquants à opter pour des méthodes alternatives. Nous estimons donc qu'il est essentiel d'étudier dès maintenant des méthodes alternatives de détection de cette attaque.

Nous proposons d'utiliser les trames de contrôle pour détecter l'attaque par faux point d'accès fantôme. Toujours dans une approche holistique de détection, nous souhaitons étudier la faisabilité de la détection des autres attaques en utilisant également les trames de contrôle. Les trames de contrôle sont des trames utilisées pour contrôler l'émission des trames de données. Parmi les trames de contrôle, il y a notamment les trames de demande de réservation de ressources (*Request to Send* en anglais), les trames d'acceptation de réservation de ressources (*Clear to send*, en anglais) et les trames d'acquittement simples et par blocs. Nous analysons uniquement les trames d'acquittement dans ces travaux, car dans la littérature scientifique, il existe déjà des études sur la détection d'une ou plusieurs attaques considérées dans ces travaux en utilisant totalement ou partiellement les informations présentes dans les trames RTS/CTS.

La détection par trame de contrôle ne nécessitant pas un changement de configuration, nous avons utilisé les données précédemment utilisées lors de transmissions sur la bande 2.4 GHz. Nous avons filtré et considéré uniquement les trames d'acquittement (ACK). Pour les attributs, nous avons conservé les quatre attributs que nous avions utilisés pour l'analyse des trames de gestion. Nous avons fait ce choix, car nous cherchons à savoir si le même ensemble de quatre attributs peut être utilisé pour détecter les trois attaques et leurs varia-

tions avec les trames de contrôle. Cela nous permettra également de comparer les résultats de la détection.

5.4.2 L'évolution des attributs

Nous étudions l'évolution des attributs, à savoir l'intervalle entre deux trames d'ACK, la puissance du signal, l'intervalle entre les numéros de séquence et le sous-type de trames de contrôle, entre les différentes situations. Pour des raisons de lisibilité, nous ne détaillons que les configurations avec un trafic moyen.

L'intervalle moyen entre deux trames de contrôle

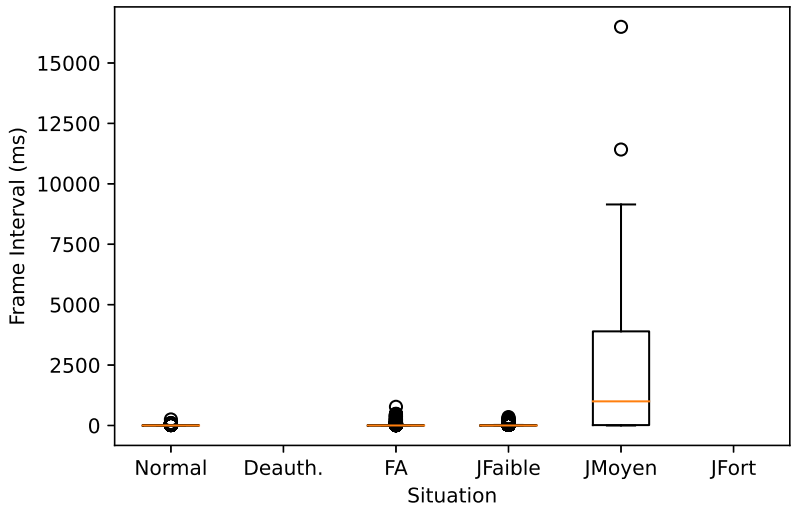

FIGURE 5.10 – Comparaison entre situations de l'intervalle entre deux trames de contrôle

Source: traitement auteur

Nous remarquons, *prima facie*, comme prévu, que dans la Figure 5.10, les trames de contrôle ne peuvent pas être utilisées pour détecter l'attaque par

déauthentification et l'attaque par brouillage (du moins lorsque cette attaque est de forte puissance, JFort sur la Figure). En effet, lorsqu'il y a une attaque par déauthentification, le client est déconnecté du point d'accès et par conséquent, le trafic entre le client et le serveur iPerf est interrompu. S'il n'y a pas de trafic, il n'y a pas d'acquittement. L'Observateur n'observe ainsi pas de trame d'acquittement. Lorsqu'il y a une attaque par brouillage avec forte puissance, le rapport signal à interférence du canal est fortement dégradé et comme les trames de données sont généralement envoyées suivant l'amendement IEEE 802.11 n, l'émission des trames de données est interrompue et l'Observateur n'observe aucune trame d'acquittement. Concernant les autres situations, nous remarquons sur cette même Figure que l'attaque par brouillage de moyenne puissance (JMoyen sur la Figure) se différencie clairement des autres par rapport à sa valeur moyenne (ligne orange). Enfin, nous pouvons constater qu'à vue d'œil, il est difficile de différencier les attaques par faux point d'accès et par brouillage de faible puissance de la situation normale.

L'écart entre les numéros de séquence

Concernant l'écart entre les numéros de séquence de deux trames de contrôle consécutives, nous avons constaté qu'il n'était pas renseigné dans les trames d'ACK. En effet, les trames de contrôle peuvent avoir un autre numéro de séquence qui évolue différemment, et c'est ce deuxième numéro de séquence qui était renseigné. Nous n'étudions pas ce deuxième numéro de séquence dans ces travaux. Par conséquent, lors de la phase de prétraitement des données, nous avons remplacé les champs nuls par la valeur 0 dans toutes les situations et variations.

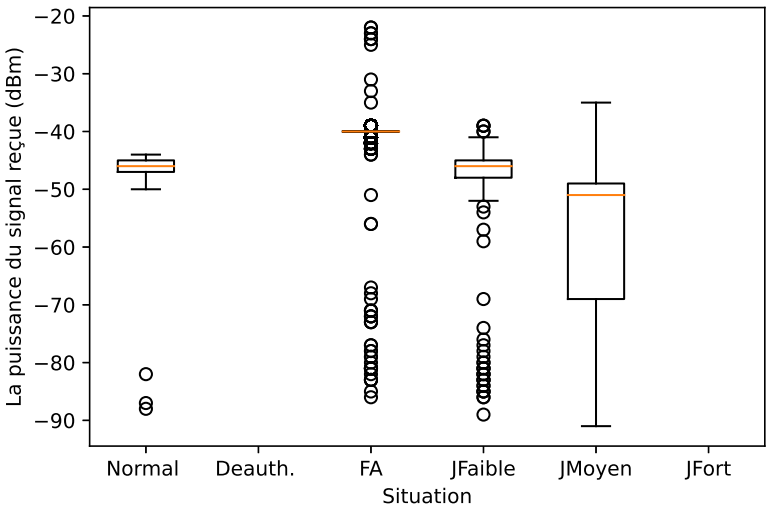

FIGURE 5.11 – Comparaison de la puissance du signal entre les situations

Source: traitement auteur ; en utilisant la bibliothèque Tkinter de Python (LUNDH, 1999)

La puissance du signal de la trame

Concernant la puissance du signal de la trame reçue, nous remarquons une différence nette de moyenne entre la situation avec l'attaque par faux point d'accès et les autres situations. Les situations avec l'attaque par brouillage faible et par brouillage moyen ont des moyennes plus ou moins similaires entre elles, mais plus basses par rapport à la situation normale. De plus, dans la situation avec le brouillage faible, il y a un éparpillement plus important des valeurs, notamment celles inférieures à -70 dBm. Après l'analyse des différentes figures (Figure 5.10 et Figure 5.11), nous avons conclu que l'attribut de puissance du signal de la trame peut être utilisé pour différencier la situation avec l'attaque par faux point d'accès de la situation normale et les situations avec des attaques par brouillage faible et moyen. L'attribut d'intervalle entre deux trames, quant à lui, peut être utilisé pour distinguer entre les situations de brouillage de moyenne et faible puissance. Nous avons ensuite utilisé ces données pour créer des modèles en utilisant les mêmes algorithmes de classification et les mêmes hyperparamètres.

Algorithmes	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	-	99.81%	-	100%	66.34%	79.34%
KNN (k=5)	-	99.67%	-	90%	66.73%	79.44%
CART	-	99.78%	-	68.75%	57.62%	75.64%
SVM	-	99.55%	-	-%	69.99%	82.23%
Logistic Regression	-	70.43%	-	83.33%	56.49%	67.32%
Naives Bayes	-	99.88%	-	7.03%	49.30%	64.70%
Linear Discriminant Analysis	-	99.76%	-	100%	23.63%	63.37%

TABLE 5.12 – Précision de détection des attaques avec trafic moyen en utilisant les trames de contrôle

Source: traitement auteur

Les résultats sont indiqués dans le tableau 5.12. Ce tableau 5.12 présente les taux de précision en fonction des situations et des modèles créés par les algorithmes de classification sur la base un jeu de trames de contrôle capturées pour cette configuration. Le jeu de donnée contient 51663 trames d'acquittement dont **22129** trames pour la situation normale, **16579** trames pour la situation avec attaque par faux point d'accès, **0** trame pour la situation avec attaque par déauthentification, **2** trames pour la situation avec attaque par brouillage de puissance forte, **31** trames pour la situation avec attaque par brouillage de puissance moyenne et **12922** trames pour la situation avec attaque par brouillage de puissance faible.

Les résultats montrent que l'attaque par faux point d'accès fantôme pourra être détectée avec une précision de 99,81 % en utilisant un modèle créé par l'algorithme *Random Forest*. Toutefois, la détection des attaques par brouillage doit être améliorée. En ce qui concerne la situation normale, l'algorithme *Random Forest* fournit le taux de précision le plus élevé, mais avec un taux d'erreur de 20 %. Ces résultats ont été obtenus en utilisant les mêmes indicateurs que

ceux utilisés pour les trames de gestion. Une étude plus approfondie sur de nouveaux indicateurs spécifiques aux trames de contrôle pourrait significativement améliorer ces résultats. La détection de l'attaque par faux point d'accès en utilisant les trames de contrôle est possible, car la présence d'un faux point d'accès ayant la même adresse MAC perturbe la transmission et entraîne des retransmissions, ce qui conduit à une perte de débit affectant l'envoi de trames de contrôle. Les trames émises par le point d'accès licite affectent également la puissance moyenne du signal de la trame reçue. Il convient aussi de noter que pour SVM, l'évaluation provoque une erreur d'exécution qui empêche le calcul de la précision pour la variation du brouillage moyen.

La détection des attaques lorsqu'elles sont réalisées de manière indépendante a été étudiée en profondeur. Il est maintenant nécessaire de se concentrer sur la détection de la combinaison d'attaques.

5.5 Détection de la combinaison des attaques

Dans cette section, nous étudions la combinaison d'attaques qui n'a pas été étudiée dans la littérature scientifique. Pour rappel, lorsqu'un attaquant crée un faux point d'accès, il peut soit attendre que les clients se connectent au faux point d'accès par inadvertance ou par erreur, soit forcer les clients à se déconnecter du point d'accès licite en utilisant une attaque par déni de service. L'attaque par déni de service peut prendre la forme d'une attaque par déauthentification ou d'une attaque par brouillage.

5.5.1 Utilisation de l'attaque par déauthentification pour déconnecter les clients

En ce qui concerne la combinaison d'une attaque par faux point d'accès et d'une attaque par déauthentification, nous avons constaté que l'attaquant ne pouvait pas utiliser la même adresse MAC que le point d'accès licite. Si l'attaquant utilise la même adresse MAC que le point d'accès licite, les trames de déauthentification qu'il émettra vont déconnecter les clients du point d'accès licite, mais vont également les empêcher de se connecter à son faux point d'accès.

Afin de faciliter la connexion des terminaux, le faux point d'accès doit être placé sur un autre canal ou utiliser une adresse MAC différente sur le même canal. Il est très facile de détecter ces deux types de faux points d'accès en analysant les informations statiques. En effet, l'analyse des champs *channel* ou *source MAC address* permettra de détecter une différence de valeur de ces attributs entre deux trames de *beacon*. L'attaque par déauthentification peut également être effectuée avec deux points d'accès ayant la même adresse MAC, à condition que les deux points d'accès ne soient pas proches. Dans notre configuration, l'attaquant est situé à proximité du point d'accès licite. Pour pouvoir effectuer la combinaison des deux attaques, nous avons changé l'adresse MAC du faux point d'accès. Nous avons ensuite reproduit toutes les situations et variations, y compris la situation avec la combinaison d'attaques, en gardant la configuration avec les transmissions sur la bande 5 GHz (voir Figure 5.12). Nous avons capturé les trames pendant deux minutes pour chaque situation et les avons analysées.

Concernant les attributs, nous avons conservé les quatre attributs et nous remarquons qu'il y a des variations différentes des autres situations, uniquement avec l'attribut intervalle entre numéros de séquence. Ceci est dû au fait que lorsque l'attaque par faux point d'accès est cumulée avec l'attaque par déauthentification, il y a un envoi excessif de trame déauthentification et la

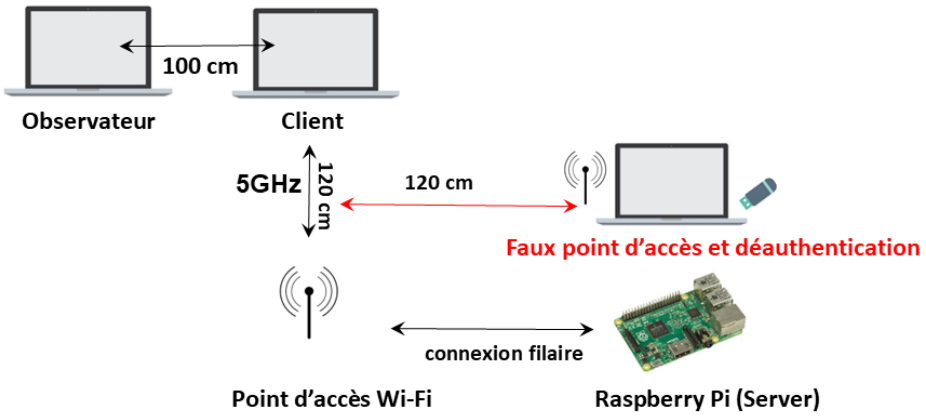

FIGURE 5.12 – Situation avec le cumul des attaques par déauthentification et par faux point d'accès

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay), icône usb (conçue par rawpixel.com, tirée de freepix.com)

présence de trame de *beacon* du faux point d'accès. L'évolution des autres attributs ne permet pas d'établir une différence visible, surtout avec la situation par l'attaque de déauthentification dont les variations d'attributs sont très proches. Après l'analyse des attributs, nous avons utilisé ces données pour créer des modèles capables de détecter les cinq situations.

Les résultats de détections sont indiqués dans les tableaux 5.13 à 5.15. Ces tableaux présentent les taux de précision en fonction des situations et des modèles créés par les algorithmes de classification sur la base d'un jeu de trames de gestion capturées pour chaque configuration.

Pour le tableau 5.13, le jeu de donnée contient **52985** trames de gestion dont **1167** trames pour la situation normale, **2316** trames pour la situation avec attaque par faux point, **15914** trames pour la situation avec attaque par déauthentification, **0** trame pour la situation avec brouillage de puissance forte, **873** trames pour la situation avec brouillage de puissance moyenne, **1168** trames pour la situation avec brouillage de puissance faible et **1168** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification.

Algorithmes	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.96%	97.59%	99.74%	-	93.16%	100%	99.16%
KNN (k=5)	99.82%	95.91%	99.83%	-	94.50%	98.87%	99.15%
CART	99.90%	97.80%	99.23%	-	92.62%	99.67%	96.88%
SVM	95.22%	94.79%	100%	-	93.11%	99.76%	98.83%
Logistic Regression	100%	99.42%	59.79%	-	85.22%	64.79%	92.11%
Naives Bayes	100%	99.14%	50.19%	-	80.61%	98.37%	96.89%
Linear Discriminant Analysis	100%	99.42%	59.79%	-	85.22%	64.78%	92.11%

TABLE 5.13 – Précision de détection d'attaques en présence d'un trafic léger

Source: traitement auteur

Pour le tableau 5.14, le jeu de donnée contient **53575** trames de gestion dont **968** trames pour la situation normale, **1928** trames pour la situation avec attaque par faux point, **16018** trames pour la situation avec attaque par déauthentification, **1** trame pour la situation avec brouillage de puissance forte, **860** trames pour la situation avec brouillage de puissance moyenne, **984** trames pour la situation avec brouillage de puissance faible et **32816** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification.

Pour le tableau 5.15, le jeu de donnée contient **51543** trames de gestion dont **478** trames pour la situation normale, **1984** trames pour la situation avec attaque par faux point, **15968** trames pour la situation avec attaque par déauthentification, **1** trame pour la situation avec brouillage de puissance forte, **865** trames pour la situation avec brouillage de puissance moyenne, **473** trames pour la situation avec brouillage de puissance faible et **31774** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification.

Algorithmes	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.89%	98.60%	93.10%	-	83.70%	99.80%	91.15%
KNN (k=5)	99.54%	98.92%	91.88%	-	76.98%	98.16%	89.88%
CART	99.90%	97.80%	99.23%	-	92.62%	99.67%	96.88%
SVM	93.58%	98.97%	92.15%	-	79.77%	100%	89.43%
Logistic Regression	93.58%	98.97%	92.15%	-	79.77%	100%	89.43%
Naives Bayes	100%	87.56%	41.21%	-	49.52%	82.09%	87.39%
Linear Discriminant Analysis	100%	90.15%	50.41%	-	49.80%	96.95%	61.36%

TABLE 5.14 – Précision de détection d'attaques en présence d'un trafic moyen

***Le F1-score micro** des algorithmes les plus performants (CART et Random Forest) est de 0.985 et 0.969. Celui-ci confirme qu'il n'y a pas de déséquilibre de classement dans ces modèles.*

Source: traitement auteur

Algorithms	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JFort}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.97%	99.03%	97.78%	-	89.59%	99.40%	90.42%
KNN (k=5)	99.77%	97.23%	95.16%	-	83.96%	95.80%	83.48%
CART	99.98%	99.15%	96.22%	-	90.57%	98.22%	84.94%
SVM	94.54%	97.95%	99.13%	-	94.40%	100%	85.83%
Logistic Regression	94.54%	97.95%	99.13%	-	94.40%	100%	85.83%
Naives Bayes	100%	94.18%	22.46%	-	63.54%	66.27%	100%
Linear Discriminant Analysis	100%	90.15%	30.68%	-	76.42%	83.19%	73.96%

TABLE 5.15 – Précision de détection d'attaques en présence de trafic intense

Source: traitement auteur

Les résultats indiquent des taux de précision excellents pour détecter le cumul des attaques par faux point d'accès et par déauthentification, pour toutes les différentes variations de trafic (de 94,54 % à 100 % selon le modèle). Pour l'attaque par brouillage avec une forte puissance, aucun taux de précision n'est disponible, car les transmissions sont effectuées sur la bande 5 GHz et la particularité de cette situation est que le signal de brouillage perturbe également les trames de *beacon*. Pour détecter cette attaque, il est nécessaire de compléter le modèle avec une détection par seuil sur la bande 5 GHz (voir les conclusions de la section 5.4) et si le cumul se fait sur la bande 2.4 GHz, de compléter avec un modèle basé de détection par seuil considérant les trames de *beacon* d'un deuxième point d'accès éloigné (voir les conclusions de la section 5.2.1). Pour les variations de trafic, les mêmes effets que dans les configurations précédentes sont observés, à savoir une diminution de la précision lorsque le trafic augmente. Le modèle créé par l'algorithme CART est le plus performant lorsque le trafic est moyen et peut être utilisé dans le SDI pour détecter les quatre situations d'attaque avec une grande précision.

Il est enfin nécessaire de détecter le cumul des attaques par faux point d'accès et par brouillage.

5.5.2 Utilisation de l'attaque par brouillage pour déconnecter les clients

L'attaquant peut également utiliser l'attaque par brouillage pour déconnecter les clients. Dans un premier temps, nous avons tenté de déconnecter les clients en utilisant une attaque par brouillage lorsque les transmissions se font sur la bande 2.4 GHz. Cependant, nous avons constaté que les terminaux ne se déconnectaient pas du point d'accès licite. En effet, bien qu'une attaque par brouillage sur la bande 2.4 GHz puisse annuler le trafic, elle n'a pas beaucoup d'impact sur les trames de *beacon*. Pour qu'une attaque par brouillage

puisse forcer les clients à se déconnecter, elle doit causer un déni de service sur le point d'accès. Tant que les clients reçoivent les trames de *beacon* sur la bande 2.4 GHz, il n'y a pas de déni de service. Nous avons donc conclu que les caractéristiques du signal de brouillage que nous avons utilisé, notamment un temps de balayage de 10 us, ne permettent pas de déconnecter les clients lorsque les transmissions se font sur la bande 2.4 GHz. Ces caractéristiques correspondent pourtant à celles des brouilleurs disponibles à l'achat sur Internet, nous avons donc étudié le cumul de l'attaque par faux point d'accès et par brouillage lorsque les transmissions se font sur la bande 5 GHz.

Sur la bande 5 GHz, toutes les trois types de trames sont émises en OFDM et subissent donc l'impact de l'attaque par brouillage. Nous avons constaté que les terminaux se déconnectent du point d'accès licite lorsqu'il y a un signal de brouillage de forte puissance. Il faut à présent déterminer si le faux point d'accès doit être sur le même canal que le point d'accès licite ou sur un canal différent. Nous avons choisi de mettre le faux point d'accès sur le même canal car c'est le type de faux point d'accès qui est le plus difficile à détecter. Pour créer le déni de service, nous orientons le signal de brouillage vers le point d'accès licite. Comme nous utilisons une antenne directionnelle, le signal de brouillage n'affecte que le point d'accès licite. Cela correspond au cas où l'attaquant se rapprocherait du point d'accès licite avec son brouilleur pour n'affecter que ce point d'accès. Nous avons ensuite programmé de nouvelles expérimentations et nous avons reproduit, sur la bande 5 GHz, les six différentes situations et leurs variations : la situation normale, l'attaque par faux point d'accès, l'attaque par déauthentification, l'attaque par brouillage, l'attaque par faux point d'accès cumulée avec l'attaque par déauthentification, l'attaque par faux point d'accès cumulée avec l'attaque par brouillage.

Nous avons cependant rencontré un problème technique lors de la réalisation de la combinaison des attaques par brouillage et par faux point d'accès. En effet, lorsque le canal est brouillé, la carte Wi-Fi Pcap peut écouter, mais se bloque dès qu'elle doit émettre sur le canal. Nous avons tenté de changer d'ordinateur,

mais cela n'a pas résolu le problème. Après plusieurs essais, nous avons décidé de déplacer le faux point d'accès loin de la zone fortement brouillée, ce qui a permis à la carte Wi-Fi de fonctionner à nouveau, comme illustré sur la Figure 5.13. Cela nous a contraints à modifier nos configurations.

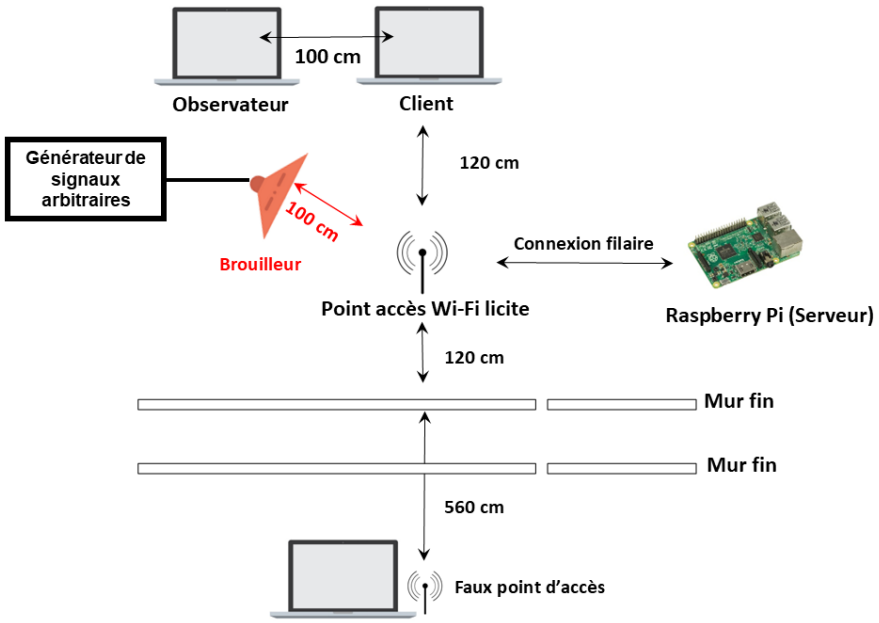

FIGURE 5.13 – Situation avec le cumul des attaques par faux point d'accès et brouillage

Source: traitement auteur : icône ordinateur (D3Images, tirée de freepix.com), icône antenne (licence open source diagrams.net), image Raspberry Pi (WikimediaImages, tirée de pixabay.com, license Pixabay), icône antenne directionnelle (Smashicons, tirée de flaticon.com, license Flaticon)

En analysant les attributs, nous avons remarqué que lorsque l'attaque par brouillage est combinée avec l'attaque par faux point d'accès, le brouillage supprime les trames de *beacon* du point d'accès licite, ce qui entraîne une absence de variation significative des valeurs moyennes des attributs tels que l'intervalle entre deux trames et l'intervalle entre deux numéros de séquence. Toutefois, l'effet de l'attaque par brouillage se manifeste clairement sur la puissance du signal de la trame reçue. Comme indiqué sur la Figure 5.14, dans la situation de cumul des attaques par brouillage et faux point d'accès (JmgFA sur la Figure), la valeur moyenne de la puissance du signal de la trame reçue est différente des autres situations.

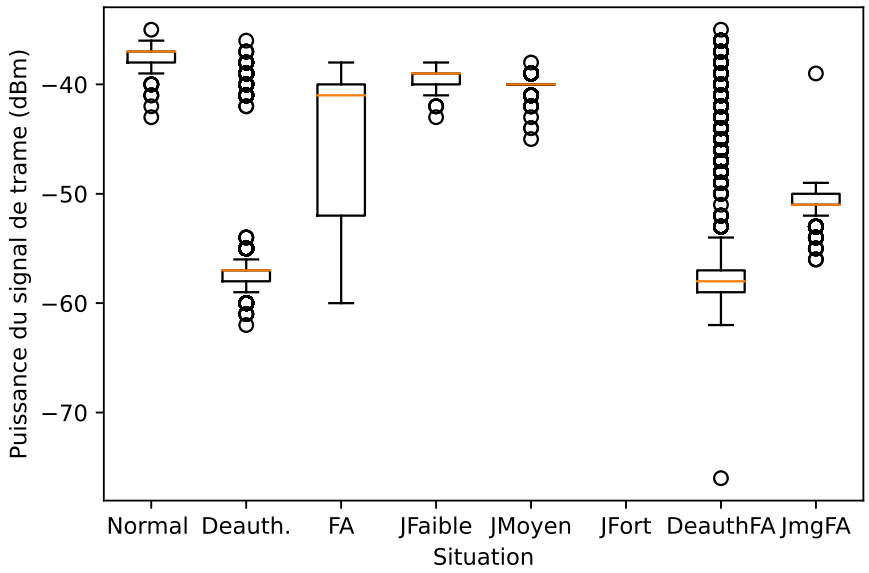

FIGURE 5.14 – Puissance de la trame reçue en fonction des différentes situations

Source: traitement auteur : en utilisant la bibliothèque Matplotlib de Python (HUNTER, 2007)

Nous avons utilisé les données pour créer des modèles permettant de détecter les six différentes situations. Les résultats de détection de ces modèles sont indiqués dans les tableaux 5.16 à 5.19. Ces tableaux présentent les taux de précision en fonction des situations et des modèles créés par les algorithmes de classification sur la base d'un jeu de trames de gestion capturées pour chaque configuration.

Pour le tableau 5.16, le jeu de donnée contient **59012** trames de gestion dont **1173** trames pour la situation normale, **2320** trames pour la situation avec attaque par faux point, **15386** trames pour la situation avec attaque par déauthentification, **2** trames pour la situation avec brouillage de puissance forte, **1040** trames pour la situation avec brouillage de puissance moyenne, **1169** trames pour la situation avec brouillage de puissance faible, **36753** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification et **1169** trames pour la situation avec cumul des attaques par faux point d'accès et par brouillage.

Pour le tableau 5.17, le jeu de donnée contient **58227** trames de gestion dont **1167** trames pour la situation normale, **2310** trames pour la situation avec attaque par faux point, **15428** trames pour la situation avec attaque par déauthentification, **1** trame pour la situation avec brouillage de puissance forte, **1074** trames pour la situation avec brouillage de puissance moyenne, **1163** trames pour la situation avec brouillage de puissance faible, **35925** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification et **1159** trames pour la situation avec cumul des attaques par faux point d'accès et par brouillage.

Pour le tableau 5.18, le jeu de donnée contient **57206** trames de gestion dont **968** trames pour la situation normale, **1663** trames pour la situation avec attaque par faux point, **15458** trames pour la situation avec attaque par déauthentification, **0** trame pour la situation avec brouillage de puissance forte, **987** trames pour la situation avec brouillage de puissance moyenne, **1056** trames pour la situation avec brouillage de puissance faible, **35905** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification et **1170** trames pour la situation avec cumul des attaques par faux point d'accès et par brouillage.

Pour le tableau 5.19, le jeu de donnée contient **52985** trames de gestion dont **1167** trames pour la situation normale, **2316** trames pour la situation avec attaque par faux point, **15914** trames pour la situation avec attaque par déauthentification, **0** trame pour la situation avec brouillage de puissance forte, **873** trames pour la situation avec brouillage de puissance moyenne, **1168** trames pour la situation avec brouillage de puissance faible, **1168** trames pour la situation avec cumul des attaques par faux point d'accès et par déauthentification et **1168** trames pour la situation avec cumul des attaques par faux point d'accès et par brouillage.

Algorithmes	$TP_{JammingFA}$	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	100%	99.06%	98.29%	99.91%	92.79%	99.83%	87.40%
KNN (k=5)	100%	97.33%	97.30%	100%	90.0 %	99.65%	86.63%
CART	100%	98.95%	96.35%	100%	90.25%	99.83%	85.69%
SVM	100%	92.95%	99.22%	100%	89.81%	100%	76.74%
Logistic Regression	100%	92.95%	99.22%	100%	89.80%	100%	76.74%
Naives Bayes	62.29%	70.84%	92.61%	32.35%	98.93%	99.65%	71.33%
Linear Discriminant Analysis	44.23%	71.09%	47.62%	45.21%	90.59%	97.60%	53.01%

TABLE 5.16 – Précision détection d'attaques en l'absence de trafic

Source: traitement auteur

Algorithmes	$TP_{JammingFA}$	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.66%	98.23%	98.61%	99.65%	99.65%	94.22%	97.55%
KNN (k=5)	99.82%	96.48%	96.10%	99.47%	93.44 %	97.26%	97.85%
CART	99.66%	98.51%	96.43%	98.69%	93.70%	100%	97.19%
SVM	99.83%	92.65%	99.20%	100%	87.27%	100%	97.52%
Logistic Regression	99.83%	92.65%	99.20%	100%	87.28%	100%	97.52%
Naives Bayes	99.65%	70.31%	76.83%	40.49%	88.33%	97.05%	92.33%
Linear Discriminant Analysis	49.31%	69.11%	41.46%	30.43%	83.12%	94.44%	95.95%

TABLE 5.17 – Précision détection d'attaques en présence de trafic léger

Source: traitement auteur

Algorithmes	TP_{JFA}	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.01%	99.35%	98.70%	94.29%	86.57%	99.24%	90.37%
KNN (k=5)	99.01%	97.71%	96.99%	89.41%	84.75 %	97.94%	89.41%
CART	99.00%	99.30%	98.36%	77.74%	87.27%	99.05%	89.09%
SVM	99.50%	92.39%	98.94%	98.88%	77.88%	100%	91.60%
Logistic Regression	99.50%	92.39%	98.94%	98.88%	77.88%	100%	91.60%
Naives Bayes	98.85%	70.00%	62.92%	11.44%	71.68%	39.4%	91.62%
Linear Discriminant Analysis	68.22%	69.45%	60.71%	1.736%	66.67%	84.30%	73.63%

TABLE 5.18 – Taux de précision pour détecter les attaques en présence de trafic moyen

Le F1-score micro *des algorithmes les plus performants (CART et Random Forest) est de 0.973 et 0.973. Celui-ci confirme qu'il n'y a pas de déséquilibre de classement dans ces modèles.*

Source: traitement auteur

Algorithmes	$TP_{JammingFA}$	$TP_{DeauthFA}$	TP_{Deauth}	TP_{FA}	TP_{JMoyen}	$TP_{JFaible}$	TP_{SN}
Random Forest	99.48%	98.54%	97.99%	97.97%	90.48%	100%	85.21%
KNN (k=5)	98.97%	96.56%	94.92%	91.60%	89.40 %	99.40%	84.17%
CART	98.80%	98.57%	97.25%	92.18%	89.69%	100%	84.88%
SVM	98.97%	92.10%	99.05%	96.76%	77.78%	100%	85.02%
Logistic Regression	43.71%	68.09%	0%	0.7792%	48.01%	69.72%	29.28%
Naives Bayes	98.97%	68.78%	41.02%	49.85%	77.24%	20.76%	79.92%
Linear Discriminant Analysis	44.23%	71.09%	47.62%	45.21%	90.59%	97.60%	53.01%

TABLE 5.19 – Précision détection d'attaques en présence de trafic intense

Source: traitement auteur

Les résultats montrent que la majorité des modèles sont capables de détecter les trois types d'attaques avec une grande précision, qu'elles soient réalisées de manière indépendante ou cumulée. L'algorithme *Random Forest* a produit le modèle le plus performant pour un trafic moyen, ce qui le rend potentiellement utilisable pour le SDI. Cependant, comme dans les sections précédentes, la précision diminue lorsque le trafic augmente.

Concernant l'attaque par brouillage avec une forte puissance, la colonne correspondante a été supprimée pour permettre une meilleure lisibilité du tableau. Si l'on souhaite détecter cette attaque, il est nécessaire de combiner le modèle avec un modèle basé sur la détection de seuil sur la bande 5 GHz (comme indiqué dans les conclusions de la section 5.4) et sur la bande 2.4 GHz, le combiner à un modèle basé sur la détection de seuil prenant en compte les trames de *beacon* d'un deuxième point d'accès éloigné (comme indiqué dans les conclusions de la section 5.2.1).

5.6 Conclusion

Dans ce chapitre, nous avons étudié la prise en compte des différents niveaux de trafic dans la détection des trois attaques. Nous avons conclu que lorsque le débit du trafic augmente, la précision de détection diminue. Nous avons ensuite étudié les possibilités d'amélioration de la détection de l'attaque par brouillage. Nous avons conclu que la détection de l'attaque par brouillage peut être améliorée en prenant en compte les trames de *beacon* d'un point d'accès licite éloigné ou en utilisant des informations de la couche 1 (attributs dérivés des puissances maximales émises sur le canal). Malheureusement, nous n'avons pas pu implémenter une chaîne GNU Radio permettant de décoder les trames IEEE 802.11 à une cadence convenable pour la détection des attaques.

Nous avons ensuite étudié le cas particulier des transmissions IEEE 802.11

sur la bande 5 GHz et avons obtenu des résultats précis pour les trois attaques sur cette bande. La détection de l'attaque par brouillage forte puissance pose toutefois un problème, car aucune trame n'est reçue dans cette situation. Nous avons donc conclu qu'il était nécessaire de compléter la détection par apprentissage automatique avec un modèle fondé sur la détection par seuil qui testera la présence régulière de trames de *beacon* pour signaler la présence de cette attaque en cas d'absence de trames. En ce qui concerne les points d'accès fantômes, une précision de 73 % peut être obtenue en utilisant les trames de contrôle. La précision peut être augmentée en effectuant une analyse plus approfondie des attributs des trames de contrôle. Enfin, nous avons étudié la détection du cumul des attaques, notamment lorsque l'attaquant utilise une attaque par faux point d'accès en combinaison avec soit une attaque par brouillage, soit une attaque par déauthentification. Nous avons conclu que la détection de ces cumuls d'attaques est possible et peut être effectuée avec précision.

Chapitre 6

Conclusion et perspectives

Les réseaux sans fil sont de plus en plus utilisés par des ordinateurs, téléphones portables, voitures et objets connectés, ainsi que par des systèmes de transport (bus, train, avion) pour communiquer avec la centrale de contrôle. Ils peuvent également être utilisés pour des communications longue distance, notamment en utilisant des satellites. La popularité de ces réseaux sans fil est principalement due au fait qu'ils permettent de créer, de gérer et d'étendre plus facilement un réseau informatique. Techniquement, cela est possible, car les réseaux sans fil ne sont pas isolés physiquement et la transmission se fait généralement de manière omnidirectionnelle dans le vide. *A contrario,* par exemple, dans les réseaux filaires, les transmissions se font à travers différents types de câbles nécessitant des liaisons et un déploiement à chaque extension du réseau. Toutefois, ces avantages peuvent être utilisés à mauvais escient par des attaquants pour perturber et attaquer ces réseaux. En effet, étant donné que l'accès au réseau ne peut être contrôlé physiquement, les attaquants peuvent plus facilement écouter ou émettre sans autorisation sur ces types de réseaux.

Dans cette thèse intitulée : « Méthodes de détection d'attaques cybernétiques par une surveillance multicouches de communication. », nous avons étudié, en particulier, les transmissions sur les réseaux sans fil Wi-Fi. Ces réseaux sont définis par la norme IEEE 802.11 et ses amendements. Nous avons consi-

déré uniquement le mode Wi-Fi infrastructure. Dans ce mode, les terminaux (ou clients) doivent se connecter à un point d'accès avant de pouvoir communiquer. Nous avons identifié deux grands types d'attaques qui peuvent fortement affecter ces réseaux : les attaques par usurpation d'identité et par déni de service. Dans cette thèse, nous avons donc considéré trois attaques, à savoir l'attaque par faux point d'accès, par brouillage et par déauthentification, qui font partie de ces deux familles. Nous avons cherché à détecter ces attaques en surveillant les informations des couches 1 et 2 du modèle OSI. L'état de l'art de la littérature scientifique nous a permis de constater que ces attaques sont connues et que plusieurs méthodes de détection ont été proposées. Toutefois, il n'existe pas d'approche holistique permettant de détecter les trois attaques avec un seul outil. En effet, ces attaques sont aujourd'hui détectées de manière isolée alors qu'elles sont pourtant liées et peuvent être cumulées. Nous avons donc défini une approche qui consiste à trouver différentes méthodes pour détecter ces attaques. Ces méthodes peuvent être classées en trois types : méthodes de prise de décision finale, de prise en considération d'indicateurs, de récupération des données et de stratégie de détection.

Concernant **les méthodes de prise en considération d'indicateurs**, nous avons constaté dans la littérature que certains auteurs proposent de détecter les attaques en utilisant un ou deux indicateurs pris séparément. Nous pensons que cette méthode n'est pas adaptée parce que les attaquants connaissent aujourd'hui ces indicateurs et peuvent, par conséquent, essayer de les contourner. Nous avons adopté une méthode qui consiste à détecter les attaques en suivant l'évolution d'un ensemble d'indicateurs. En considérant l'évolution de cet ensemble d'indicateurs, notre but est de mettre en place une méthode de prise de décision finale qui va indiquer soit la présence d'une des attaques, de toutes les attaques ou aucune attaque. Concernant **les méthodes de prise de décision finale**, nous avons testé la prise de décision par seuils ainsi que celle utilisant des algorithmes d'apprentissage automatique. Si nous avons commencé par la prise de décision par seuils, nous nous sommes rapidement ren-

dus compte que nos données ne suivaient pas une tendance linéaire, ce qui risquait de générer diverses erreurs de détection. Nous avons donc opté pour une approche de prise de décision en utilisant un algorithme de classification. Toutefois, l'étude de la méthode par détermination de seuils n'est pas complètement abandonnée, car elle peut être utilisée pour compléter la méthode avec l'utilisation d'algorithmes de classification lorsque les données sont peu ou pas disponibles.

La méthode d'apprentissage automatique nous a permis de créer un seul modèle capable de détecter les trois attaques, qu'elles soient perpétrées de manière indépendante ou cumulée, en analysant les quatre indicateurs simultanément. Nous avons créé ces modèles en utilisant plusieurs algorithmes de classification pour comparer les résultats de chaque algorithme et ne garder que le plus performant. Les algorithmes *Random Forest*, KNN ou CART ont montré d'excellents taux de précision pour détecter les attaques. Par exemple, comme indiqué à la fin du chapitre 4, l'algorithme *Random Forest* détecte, lorsque le trafic est moyen, l'attaque par déauthentification avec un taux de précision de 99,93 %, l'attaque par faux point d'accès avec un taux de 100 %, l'attaque par brouillage de moyenne puissance avec un taux de 68,27 %, par brouillage de faible puissance avec un taux de 86,82 % et par brouillage de forte puissance avec un taux de 68,27 %. Pour le trafic moyen, nous avons retenu l'algorithme *Random Forest*, car il a obtenu les meilleurs résultats.

Nous avons ensuite cherché à améliorer ces résultats préliminaires en prenant en compte des cas particuliers et des formes d'attaques avancées telles que les faux points d'accès fantômes et les attaques cumulées. Pour les cas particuliers, nous avons pris en compte les différents niveaux de trafic et les transmissions sur la bande 5 GHz. Il est important de souligner que de nombreux travaux scientifiques ne prennent pas en compte ces différents niveaux de trafic et surtout l'absence de trafic. Cependant, comme nous l'avons démontré, il y a un impact sur la décision lorsqu'il y a une variation de trafic. En effet, nos résultats ont montré que lorsque le débit du trafic augmente, la

précision de la détection diminue, car le trafic peut être priorisé sur l'envoi des trames de *beacon*. Pour l'absence de trafic, nous avons démontré que nous pouvons détecter les trois types d'attaques, en particulier l'attaque par brouillage, avec des taux de détection supérieurs à 78 %. Ceci est intéressant, car dans la littérature scientifique, de nombreux auteurs ont proposé des méthodes de détection basées sur l'analyse des trames de données, qui ne sont pas disponibles en l'absence de trafic.

Dans cette thèse, nous avons créé différents modèles pour étudier la variation de trafic. Le modèle utilisant les résultats de trafic moyen est différent de celui utilisant un trafic faible. Dans des travaux futurs, il serait intéressant d'étudier si la performance d'un modèle choisi comme référence varie lorsque le trafic varie. Par exemple, si le modèle avec trafic moyen est choisi comme référence, conserve-t-il ses performances de détection lorsque le trafic du réseau varie brusquement ? À défaut, il faudrait essayer d'intégrer les différentes variations de trafic dans un même modèle et vérifier si les algorithmes sont capables de distinguer les différentes situations. Une solution, ce serait d'utiliser un palier pour changer de modèle de référence en fonction du trafic. En ce qui concerne la variation de la puissance de brouillage, celle-ci a été intégrée dans chaque modèle, de sorte que ces questions ne se posent pas pour cette attaque. Toutefois, nous avons pris en compte seulement trois niveaux de puissance de brouillage correspondant à la position potentielle de l'attaquant. Nous pensons qu'il est nécessaire, dans des travaux futurs, de prendre en compte d'autres niveaux de puissance, de modifier le positionnement des différents composants de la configuration, de modifier le temps de balayage du signal, de détecter la position de l'attaquant (comme le projet GéoLOCalisation d'ATtaques sur réseaux sans fil (*GLO-CAT Project* s. d.)) et d'étudier d'autres types d'attaques de brouillage (réactives, deceptives...) et d'autres variations d'attaques par déauthentification pour obtenir un modèle plus complet.

Sur la transmission sur la bande 5 GHz, nous avons étudié les différences par rapport à la transmission sur la bande 2.4 GHz. Il convient de noter qu'à notre

connaissance, aucune étude comparative similaire n'a été menée dans la littérature scientifique. Nous avons constaté que sur la bande 5 GHz, les résultats de détection sont plus précis pour l'attaque par brouillage et ses variations. Cela est dû au fait que sur la bande 5 GHz, tous les types de trames sont transmis en OFDM, ce qui permet aux cartes Wi-Fi de mieux mesurer le rapport signal à bruit de ces trames de gestion. Par conséquent, l'algorithme *Random Forest* parvient à détecter l'attaque par brouillage de moyenne puissance avec un taux de détection de 82,03 % et l'attaque par brouillage de faible puissance avec un taux de détection de 95,88%, lorsque le trafic est moyen. Ces résultats sont meilleurs que ceux obtenus précédemment sur la bande 2.4 GHz. Les résultats de détection des autres types d'attaques restent relativement inchangés.

Une autre différence notable est que sur la bande 5 GHz, les trames de *beacon* sont inexistantes dans la variation de l'attaque par brouillage de forte puissance. Par conséquent, il est nécessaire de coupler la détection par classification avec la détection par seuils. Dans nos configurations, l'absence de trames de *beacon* peut correspondre uniquement à l'attaque par brouillage de forte puissance. Dans le cadre de travaux futurs, il serait pertinent d'examiner les autres situations pouvant conduire à cette problématique, notamment une situation de déni de service réel du point d'accès. Cette situation réelle devrait toujours être, *a minima*, différente de la situation avec une attaque par brouillage de forte puissance, car lorsque le point d'accès est victime d'un déni de service réel, les transmissions émanant des clients seront toujours présentes, tandis que dans la situation d'attaque par brouillage de forte puissance, aucune ou très peu de trames sont émises de la part des clients et du point d'accès.

En ce qui concerne les attaques avancées, nous avons étudié la possibilité de détecter l'attaque par faux point d'accès licite fantôme en utilisant les trames de contrôle. Bien que cette attaque ne soit pas documentée dans la littérature scientifique, elle pourrait très probablement être utilisée par des attaquants à l'avenir, surtout si la majorité des méthodes de détection pour cette attaque est fondée sur les trames de *beacon*. Nous avons obtenu des résultats satisfaisants

pour la détection de cette attaque en utilisant les trames de contrôle, avec un taux de précision jusqu'à de 99,81% pour l'algorithme *Random forest*. Dans les travaux futurs, il convient d'étudier l'évolution des numéros de séquence des trames de contrôle afin d'augmenter considérablement les taux de détection et de diminuer les fausses alertes. Il serait également intéressant de reproduire cette attaque en expérimentation, bien que les cartes Wi-Fi que nous ayons utilisées ne permettaient pas de reproduire cette attaque.

En ce qui concerne le cumul des attaques, nous avons montré que les trois attaques, lorsqu'elles étaient réalisées de manière indépendante et cumulée, pouvaient être détectées avec un seul modèle.L'algorithme *Random Forest* a pu détecter le cumul des attaques par brouillage et faux point d'accès avec un taux de 99,48 % et le cumul des attaques par déauthentification et faux point d'accès avec un taux de 99,54 %. Cependant, pour détection l'attaque par brouillage avec forte puissance, il est nécessaire de coupler notre modèle avec modèle créé par une détection par seuils. Dans les travaux futurs, il serait intéressant de reproduire le cumul des attaques sur la bande 2.4 GHz en prenant soin de modifier le temps de balayage du signal de brouillage pour pouvoir déconnecter les clients lors du cumul des attaques par déauthentification et par faux point d'accès.

Concernant la **méthode de récupérer des données**, nous avons utilisé différentes méthodes pour récupérer les données : récupérer sur la couche physique, liaison de données ou les deux. Nous avons utilisé des cartes Wi-Fi qui nous a permis d'avoir des informations de la couche 2 et certaines informations de la couche 1 du modèle OSI. Nous avons également tenté d'utiliser des radios logicielles (SDR) qui permettent de récupérer plusieurs informations synchronisées de la couche 1 et de la couche 2. Cependant, l'utilisation des SDR a été infructueuse, car la chaîne de reconnaissance et de détection des trames IEEE 802.11 que nous avons utilisée n'a pas réussi à reconnaître suffisamment de trames IEEE 802.11 pour que la méthode de détection par classification puisse être utilisée. Dans les travaux futurs, un important travail de traitement du

signal sera nécessaire pour améliorer cette chaîne et l'adapter éventuellement à diverses radios logicielles telles que le HackRF. Si aucune solution alternative n'est trouvée, il faudra rester sur une méthode de récupération de données fondée uniquement sur la couche 2, c'est-à-dire, en utilisant uniquement des cartes Wi-Fi. Si les analyseurs de spectre permettent également de récupérer des informations de la couche 1, ils sont coûteux et peu mobiles. Parmi ces trois méthodes, la méthode de récupération de données basée uniquement sur la couche 2 est aujourd'hui la plus viable. En effet, elle présente plusieurs avantages, notamment en termes de rapport coût-détection. Les cartes Wi-Fi sont moins chères et plus compactes que les radios logicielles permettant de décoder et d'analyser en quantité suffisante les trames IEEE 802.11. Par conséquent, un système de détection qui utilise cette méthode pourra être plus facilement intégré dans différents environnements à un coût raisonnable.

Tout cela nous amène aux stratégies de détection. Les réseaux Wi-Fi que nous avons considérés dans cette thèse sont des réseaux Wi-Fi en mode infrastructure et pour détecter les attaques sur ce type de réseau, nous avons pris en compte des trames d'un point d'accès du réseau. Pour améliorer la détection de l'attaque par brouillage, nous avons considéré une méthode qui prend en compte l'activité d'un deuxième point d'accès dans la détection des attaques. En prenant en compte les trames d'un deuxième point d'accès se trouvant sur le même canal, mais éloigné du point d'accès autorisé, et en combinant la détection par seuils avec des algorithmes de classification, nous avons réussi à augmenter considérablement la détection de l'attaque par brouillage avec une forte puissance jusqu'à 100 %. Dans les travaux futurs, il est nécessaire d'étudier si une variation de trafic sur le deuxième point d'accès peut affecter ces résultats. Il faut aussi considérer le problème de synchronisation dans le cas où le deuxième point d'accès se situe sur un autre canal, notamment lorsque la récupération de données se fait avec deux cartes Wi-Fi.

En envisageant des travaux futurs de manière générale, il serait opportun de considérer la réaction à adopter si l'attaquant cible le système de détection

d'intrusion (SDI) lui-même, par exemple en réalisant une attaque de déni de service sur ce dernier avant de perpétrer les attaques. À l'heure actuelle, le SDI ne sera pas en mesure d'alerter en cas d'attaque de ce type. Une solution à ce problème serait de déployer plusieurs mini-sondes du SDI connectées entre elles et capables de détecter les trois types d'attaques. Un tel déploiement en réseau éviterait les problèmes de redondance et accroîtrait la résilience en cas d'attaque du SDI.

Si les mini-sondes sont connectées à Internet, elles pourront, en tant que client léger (Amoordon et Rocha, 2019), enregistrer leur état et celui des attaques sur une blockchain privée. Sinon, elles pourraient elle-même gérer une blockchain simplifiée dans le réseau local. Cette configuration en blockchain va permettre d'identifier et de rejeter automatiquement les sondes corrompues. Toutefois, si un déploiement en réseau du SDI est choisi, il faudra recalibrer la détection par brouillage, car la mesure du rapport signal à interférence par la carte Wi-Fi varie selon sa position. Il convient également d'améliorer l'implémentation du SDI en ajoutant deux indicateurs restants et en prenant en compte les différents modèles créés dans le chapitre 4 et 5. Enfin, il est nécessaire de comparer les deux versions du logiciel en termes de précision, de complexité et de temps de détection.

Annexes

Annexe A

Captures d'écran

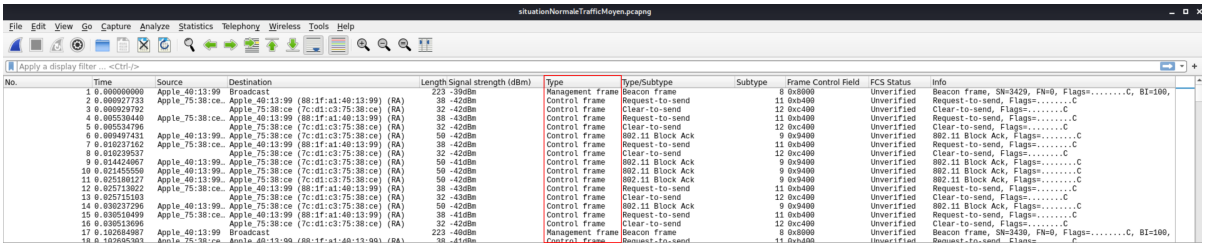

FIGURE A.1 – Example de capture Wireshark

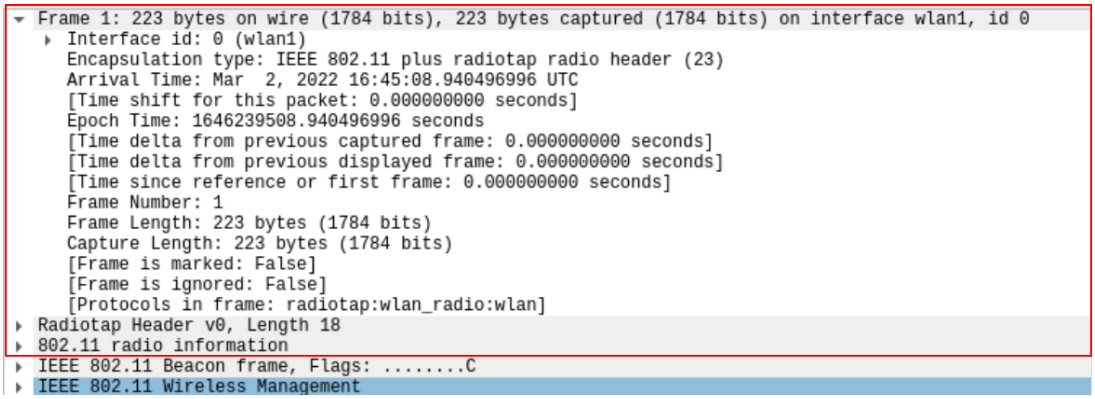

FIGURE A.2 – Example de capture Wireshark individuelle

Annexe B

Listings

Listing 1 – Commandes pour lancer le serveur iPerf à distance

```
# Se connecter sur Raspberry Pi
ssh raspberry@192.168.0.1
# Lancer le serveur iPerf depuis le raspberry
iperf3 -S
```

Listing 2 – Commandes pour mettre la carte en mode monitor

```
# Deactiver la carte Pcap

ifconfig wlan1 down

# Changer le mode d'operation en mode monitor

iwconfig wlan1 mode monitor

# Mettre la carte sur le canal 13

iwconfig wlan1 channel 13
```

```
# Reactiver la carte Pcap
ifconfig wlan1 up
```

Listing 3 – Commandes pour modifier le flux du trafic iPerf

```
iperf3 -c 192.168.1.102 -b 100M

# Pour lancer une transmission moyenne de donnees

iperf3 -c 192.168.1.102 -b 50M

# Pour lancer une transmission faible de donnees

iperf3 -c 192.168.1.102 -b 1M
```

Listing 4 – Commandes pour lancer le faux point d'accès

```
    # Lancer Wi-Fi Pumpkin

    wifipumpkin3

    # Acceder aux caracteristiques du point d'acces

     ap
    # Indiquer l'interface a utiliser (ici wlan1)

    set interface wlan1
```

```
# Indiquer l'adresse MAC du point d'acces licite a copie

set bssid 88:00:00:00:00:00

# Indiquer le nom du reseau

set ssid TER

# Indiquer la config

set hostapd_config true

# Lancer le faux point d'acces
start
```

Listing 5 – Commandes pour lancer l'attaque par déauthentification

```
# Deactiver la carte Wi-Fi

 ifconfig wlan1 down

# Mettre la carte en mode monitor

 iwconfig wlan1 mode monitor

# Reactiver la carte Wi-Fi

 ifconfig wlan1 up

# Rechercher le reseau cible
```

```
 airdump-ng wlan0mon

# Indiquer le bssid et le canal du reseau cible

 airodump-ng wlan1 --bssid 88:00:00:00:00:00 --channel 14

# Lancer l'attaque par deauthentification et emettre continuellement

aireplay-ng --deauth 0 -c ff:ff:ff:ff:ff:ff  -a 88:00:00:00:00:00 wlan1
```

Bibliographie

Aboba, B., L. Blunk, J. Vollbrecht, J. Carlson et H. Levkowetz (juin 2004). « Extensible Authentication Protocol EAP ». In : *Request for comments (RFC)* 3748.

Achaal, B., M. R. Mortada, A. Mansour et A. Nasser (2022). « Wireless Communication Attack Using SDR and Low-Cost Devices ». In : *Intelligent Decision Technologies*. Sous la dir. d'I. Czarnowski, R. J. Howlett et L. C. Jain. Singapore : Springer Nature Singapore, p. 417-428.

Adil, M., R. Khan et M. A. N. U. Ghani (2020). « Preventive techniques of phishing attacks in networks ». In : *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*. IEEE, p. 1-8.

Aditya, P., V. Erdélyi, M. Lentz, E. Shi, B. Bhattacharjee et P. Druschel (2014). « Encore : Private, context-based communication for mobile social apps ». In : *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, p. 135-148.

Agarwal, M., S. Biswas et S. Nandi (2013). « Detection of De-authentication Denial of Service attack in 802.11 networks ». In : *2013 Annual IEEE India Conference (INDICON)*. IEEE, p. 1-6.

Agarwal, M., D. Pasumarthi, S. Biswas et S. Nandi (2016). « Machine learning approach for detection of flooding DoS attacks in 802.11 networks and

attacker localization ». In : *International Journal of Machine Learning and Cybernetics* 7.6, p. 1035-1051.

Agosta, S., R. Sierra et F. Chapron (2015). « High-speed mobile communications in hostile environments ». In : *Journal of Physics : Conference Series*. T. 664. 5. IOP Publishing, p. 052001.

Aha, D. W., D. Kibler et M. K. Albert (1991). « Instance-based learning algorithms ». In : *Machine learning* 6, p. 37-66.

AirPcap, R. (s. d.). *Riverbed AirPcap website*. `https://support.riverbed.com/content/support/software/steelcentral-npm/airpcap.html`. Consulté le 05 novembre 2022.

Alani, M. M. et M. M. Alani (2014). « OSI model ». In : *Guide to OSI and TCP/IP Models*, p. 5-17.

Alotaibi, B. et K. Elleithy (2016). « Rogue access point detection : Taxonomy, challenges, and future directions ». In : *Wireless Personal Communications* 90.3, p. 1261-1290.

Amoordon, A., C. Gransart et V. Deniau (2020). « Characterizing Wi-Fi Man-In-the-Middle Attacks ». In : *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*. IEEE, p. 1-4.

Amoordon, A. et H. Rocha (2019). « Presenting Tendermint : Idiosyncrasies, weaknesses, and good practices ». In : *2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. IEEE, p. 44-49.

Anastasi, G., E. Borgia, M. Conti et E. Gregori (2004). « Wi-fi in ad hoc mode : a measurement study ». In : *Second IEEE Annual Conference on*

Pervasive Computing and Communications, 2004. Proceedings of the, p. 145-154. DOI : 10.1109/PERCOM.2004.1276853.

ARACKAPARAMBIL, C., S. BRATUS, A. SHUBINA et D. KOTZ (2010). « On the reliability of wireless fingerprinting using clock skews ». In : *Proceedings of the third ACM conference on Wireless network security*, p. 169-174.

ARJOUNE, Y., F. SALAHDINE, M. ISLAM, E. GHRIBI et N. KAABOUCH (jan. 2020). « A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication ». In : *2020 International Conference on Information Networking (ICOIN)*. Los Alamitos, CA, USA : IEEE Computer Society, p. 459-464. DOI : 10.1109/ICOIN48656.2020.9016462.

ARORA, A. (2018). « Preventing wireless deauthentication attacks over 802.11 networks ». In : *arXiv preprint arXiv :1901.07301*.

ATELIN, P. (2008). *Wi-Fi : réseaux sans fil 802.11*. Ediciones ENI.

— (2009). *Réseaux informatiques : notions fondamentales : normes, architecture, modèle OSI, TCP/IP, Ethernet, Wi-Fi,...* Editions ENI.

AUDEH, M. (2004). « Metropolitan-scale Wi-Fi mesh networks ». In : *Computer* 37.12, p. 119-121.

BALAKRISHNAMA, S. et A. GANAPATHIRAJU (1998). « Linear discriminant analysis-a brief tutorial ». In : *Institute for Signal and information Processing* 18.1998, p. 1-8.

BALIGA, J., R. AYRE, K. HINTON et R. S. TUCKER (2011). « Energy consumption in wired and wireless access networks ». In : *IEEE Communications Magazine* 49.6, p. 70-77.

Bambang Setiadji, M. Y., R. Ibrahim et A. Amiruddin (2019). « Lightweight Method for Detecting Fake Authentication Attack on Wi-Fi ». In : *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, p. 280-285. doi : 10.23919/EECSI48112.2019.8976975.

Banerji, S. et R. S. Chowdhury (2013). « On IEEE 802.11 : wireless LAN technology ». In : *arXiv preprint arXiv :1307.2661*.

Bastibl (nov. 2015). *GitHub - bastibl/gr-ieee802-11 : IEEE 802.11 a/g/p Transceiver*. https://github.com/bastibl/gr-ieee802-11. Consulté le 05 novembre 2022.

Berghel, H. et J. Uecker (2005). « WiFi attack vectors ». In : *Communications of the ACM* 48.8, p. 21-28.

Bhojani, R. et R. Joshi (2016). « An integrated approach for jammer detection using software defined radio ». In : *Procedia Computer Science* 79, p. 809-816.

Bhushan, B., G. Sahoo et A. K. Rai (2017). « Man-in-the-middle attack in wireless and computer networking—A review ». In : *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*. IEEE, p. 1-6.

Bloessl, B., M. Segata, C. Sommer et F. Dressler (2013). « An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio ». In : *Proceedings of the second workshop on Software radio implementation forum*, p. 9-16.

Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.

Brain, M., T. V. Wilson et B. Johnson (2004). « How wifi works ». In : *Dimuat turun Februari* 15, p. 2005.

Bratus, S., C. Arackaparambil et A. Shubina (avr. 2011). *Detection of Rogue APs Using Clock Skews : Does it Really Work ?* https://www.cs.dartmouth.edu/~sergey/skew/toorcon11-slides.pdf. Consulté le 10 novembre 2022.

Bratus, S., C. Cornelius, D. Kotz et D. Peebles (2008). « Active behavioral fingerprinting of wireless devices ». In : *Proceedings of the first ACM conference on Wireless network security*, p. 56-61.

Breiman, L., J. Friedman, R. Olshen et C. Stone (1984). « Cart ». In : *Classification and regression trees*.

Cekerevac, Z., Z. Dvorak, L. Prigoda et P. Cekerevac (2017). « Internet of things and the man-in-the-middle attacks–security and economic risks ». In : *MEST Journal* 5.2, p. 15-25.

Cheng, M., Y. Ling et W. B. Wu (2017). « Time Series Analysis for Jamming Attack Detection in Wireless Networks ». In : *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, p. 1-7. doi : 10.1109/GLOCOM.2017.8254000.

Chumchu, P., T. Saelim et C. Sriklauy (2011). « A new MAC address spoofing detection algorithm using PLCP header ». In : *The International Conference on Information Networking 2011 (ICOIN2011)*. IEEE, p. 48-53.

Clarke, J. (2009). *SQL injection attacks and defense*. Elsevier.

Comer, D. E. (2018). *The Internet book : everything you need to know about computer networking and how the Internet works*. Chapman et Hall/CRC.

Conti, M., N. Dragoni et V. Lesyk (2016). « A survey of man in the middle attacks ». In : *IEEE communications surveys & tutorials* 18.3, p. 2027-2051.

Cossa, D. (2014). « The Dangers of Deauthentication Attacks in an Increasingly Wireless World ». In : *Iowa State University* 537.

D'Otreppe de Bouvette, T. (2016). *Aircrack-ng*. https://www.aircrack-ng.org/. Consulté le 05 novembre 2022.

Dromard, P. (2021). *Comment créer de faux points d'accès en utilisant Scapy en Python ?* https://www.geeksforgeeks.org/how-to-create-fake-access-points-using-scapy-in-python/. Consulté le 05 novembre 2022.

eroberts (avr. 2003). *Wireless Computing*. https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/wireless-computing/int_ss.shtml. Consulté le 05 novembre 2022.

Ertel, W. (2018). *Introduction to artificial intelligence*. Springer.

Faligot, R. (2007). « Du projet Safari au contrôle biométrique : Big Brother est parmi nous ». In : *Histoire secrète de la Ve République*. La Découverte, p. 278-288.

FALQUE-PIERROTIN, I. (nov. 2014). *Commission de réflexion et de propositions sur le droit et les libertés à l'âge du numérique*. Assemblée Nationale Mercredi 26 novembre 2014 Séance de 17 heures Compte rendu n° 08.

Ferguson, P. et G. Huston (1998). « What is a VPN ? » In.

Freeman, R. L. (2005). *Fundamentals of telecommunications*. John Wiley & Sons.

GENTILI, E. (2009). *mdk3*. https://github.com/wi-fi-analyzer/mdk3-master. Consulté le 05 novembre 2022.

GILAD, Y. et A. HERZBERG (2014). « Off-path TCP injection attacks ». In : *ACM Transactions on Information and System Security (TISSEC)* 16.4, p. 1-32.

GILLES, W. et I. BOUHADANA (2019). « Quarante ans de construction du droit du numérique. Les enjeux juridiques de l'avènement d'un monde intelligent ». In : Revue internationale de droit des données et du numérique.

GLO-CAT Project (s. d.). https://cybcom.univ-gustave-eiffel.fr/projets/glocat/. Consulté le 9 décembre 2022.

GLOWNIAK, J. (1998). « History, structure, and function of the Internet ». In : *Seminars in nuclear medicine*. T. 28. 2. Elsevier, p. 135-144.

GRALLA, P. (1998). *How the Internet works*. Que Publishing.

GRANDINI, M., E. BAGLI et G. VISANI (2020). « Metrics for multi-class classification : an overview ». In : *arXiv preprint arXiv :2008.05756*.

GRÉGOIRE, S. (2009). « Le statut de l'adresse IP ». In : *LEGICOM* 2, p. 103-107.

GROVER, K., A. LIM et Q. YANG (2014). « Jamming and anti-jamming techniques in wireless networks : A survey ». In : *International Journal of Ad Hoc and Ubiquitous Computing* 17.4, p. 197-215.

GUO, F. et T.-c. CHIUEH (2005). « Sequence number-based MAC address spoof detection ». In : *International Workshop on Recent Advances in Intrusion Detection*. Springer, p. 309-329.

HAK5 (2008). *WiFi Pineapple*. https://docs.hak5.org/hc/en-us/sections/360003463354-WiFi-Pineapple. Consulté le 05 novembre 2022.

HAN, C., J. IN-JANG, J.-f. SHAO, K. CHAE, B. SEONG-SOO et S. JUNG (2012). « A Scheme of Detection and Prevention Rogue AP using Comparison Security Condition of AP ». In.

HAN, H., B. SHENG, C. C. TAN, Q. LI et S. LU (2011). « A timing-based scheme for rogue AP detection ». In : *IEEE Transactions on parallel and distributed Systems* 22.11, p. 1912-1925.

HARRIS, M. (2018). « Tech giants race to build orbital internet [news] ». In : *IEEE Spectrum* 55.6, p. 10-11.

HARRIS, M. (1978). « India's sacred cow ». In : *Human Nature* 1.2, p. 28-36.

HAY, A., P. GIANNOULIS, K. HAY et W. VERBANEC (2009). « Chapter 5 - Security and Access Configuration ». In : *Nokia Firewall, VPN, and IPSO Configuration Guide*. Sous la dir. d'A. HAY, P. GIANNOULIS, K. HAY et W. VERBANEC. Boston : Syngress, p. 165-225. DOI : https://doi.org/10.1016/B978-1-59749-286-7.00005-X.

HAZZAN, O. et K. MIKE (2022). « Teaching core principles of machine learning with a simple machine learning algorithm : the case of the KNN algorithm in a high school introduction to data science course ». In : *ACM Inroads* 13.1, p. 18-25.

HECKMANN, T. (2018). « Reverse engineering secure systems using physical attacks ». Thèse de doct. Paris Sciences et Lettres (ComUE).

HUAWEI (mai 2022). *Wireless intrusion detection*. https://support.huawei.com/enterprise/en/doc/EDOC1000141910/2b7e5365/wireless-intrusion-detection. Consulté le 05 novembre 2022.

HUNTER, J. D. (2007). « Matplotlib : A 2D graphics environment ». In : *Computing in Science & Engineering* 9.3, p. 90-95. DOI : 10.1109/MCSE.2007.55.

HYMAN, A. (1985). *Charles Babbage : Pioneer of the computer*. Princeton University Press.

IEEE802.11STANDARD (2021). « IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications ». In : *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, p. 1-4379. DOI : 10.1109/IEEESTD.2021.9363693.

INGRAMMICROCLOUDCHANNEL (juill. 2022). *Norton WiFi Privacy Man in the Middle Attack Demonstration*. www.youtube.com/watch?v=f1MEyWESD78. Consulté le 05 novembre 2022.

ISAACCOMPUTERSCIENCE (s. d.). *Wired and wireless networks*. https://isaaccomputerscience.org/concepts/net_network_wired_wireless?examBoard=all&stage=all.

ISO/IECJTC1 (nov. 1994). *Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Modèle de référence de base : Le modèle de base*. https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-1:ed-1:v2:fr. Consulté le 05 novembre 2022.

JAMAL, T., P. AMARAL, A. KHAN, A. ZAMEER, K. ULLAH et S. A. BUTT (2018). « Denial of service attack in wireless LAN ». In : *ICDS 2018* 51.

JAMBET, C. (2018). « L'Homme parfait. Métaphysique de l'âme et eschatologie selon Qāzī Saīd Qummī (m. 1695) ». In : *Annuaire de l'École pratique des*

hautes études (EPHE), Section des sciences religieuses. Résumé des conférences et travaux 125, p. 411-423.

José, A. N. d. S., V. Deniau, Ú. d. C. Resende et R. Adriano (2019). « Mitigating Intentional Electromagnetic Interferences over the GSM-R System with Adaptive Filters ». In : *2019 23rd International Conference on Applied Electromagnetics and Communications (ICECOM)*, p. 1-6. doi : 10.1109/ICECOM48045.2019.9163653.

Jud, M. (août 2021). *Le WiFi 6E arrive en Europe : est-ce que passer à la nouvelle norme en vaut la peine ?* https://www.digitec.ch/fr/page/le-wifi-6e-arrive-en-europe-est-ce-que-passer-a-la-nouvelle-norme-en-vaut-la-peine--20628. Consulté le 02 decembre 2022.

Jun, J., P. Peddabachagari et M. Sichitiu (2003). « Theoretical maximum throughput of IEEE 802.11 and its applications ». In : *Second IEEE International Symposium on Network Computing and Applications, 2003. NCA 2003.* IEEE, p. 249-256.

Kao, K. F., W. C. Chen, J. C. Chang et H. T. Chu (2014). « An Accurate Fake Access Point Detection Method Based on Deviation of Beacon Time Interval ». In : *2014 IEEE Eighth International Conference on Software Security and Reliability-Companion*, p. 1-2. doi : 10.1109/SERE-C.2014.13.

Karwatka, D. (1999). « Joseph Marie Jacquard and the punched card textile loom ». In : *Tech Directions* 59.4, p. 17.

Katz, F. H. (2010). « Wpa vs. wpa2 : Is wpa2 really an improvement on wpa ? » In : *2010 4th Annual Computer Security Conference (CSC 2010), Coastal Carolina University, Myrtle Beach, SC*.

KHASAWNEH, M., I. KAJMAN, R. ALKHUDAIDY et A. ALTHUBYANI (2014). « A survey on Wi-Fi protocols : WPA and WPA2 ». In : *International conference on security in computer networks and distributed systems*. Springer, p. 496-511.

KIM, T., H. PARK, H. JUNG et H. LEE (2012). « Online detection of fake access points using received signal strengths ». In : *2012 IEEE 75th vehicular technology conference (VTC Spring)*. IEEE, p. 1-5.

KLIAZOVICH, D. (2006). « CROSS-LAYER PERFORMANCE OPTIMIZATION IN WIRELESS LOCAL AREA NETWORKS ». Thèse de doct. University of California, Los Angeles.

KLIMIASHVILI, G., C. TAPPARELLO et W. HEINZELMAN (2020). « LoRa vs. WiFi ad hoc : A performance analysis and comparison ». In : *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, p. 654-660.

KREUTZER, R. T., M. SIRRENBERG et al. (2020). *Understanding artificial intelligence*. Springer.

KUMAR, S., S. DALAL et V. DIXIT (2014). « The OSI model : Overview on the seven layers of computer networks ». In : *International Journal of Computer Science and Information Technology Research* 2.3, p. 461-466.

L'EUROPE, C. de (jan. 1981). *Convention pour la protection des personnes à l'égard du traitement automatisé des données à caractère personnel*.

LAUE, A. (2004). « How the computer works ». In : *A companion to digital humanities*, p. 143-160.

LAWRENCE, J. (1993). *Introduction to neural networks*. California Scientific Software.

Louppe, G. (2014). « Understanding random forests : From theory to practice ». In : *arXiv preprint arXiv :1407.7502*.

Lovinger, N., T. Gerlich, Z. Martinasek et L. Malina (2020). « Detection of wireless fake access points ». In : *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, p. 113-118.

Ltd, R. P. (2015). *Buy a Raspberry Pi 2 Model B – Raspberry Pi characteristics*. https://www.raspberrypi.com/products/raspberry-pi-2-model-b/. Consulté le 05 novembre 2022.

Lundh, F. (1999). « An introduction to tkinter ». In : *URL : www. pythonware. com/library/tkinter/introduction/index. htm*. Consulté le 05 novembre 2022.

Mahesh, B. (2020). « Machine learning algorithms-a review ». In : *International Journal of Science and Research (IJSR).[Internet]* 9, p. 381-386.

Maimon, D., M. Becker, S. Patil et J. Katz (2017). « Self-protective behaviors over public WiFi networks ». In : *The {LASER} workshop : Learning from authoritative security experiment results ({LASER} 2017)*, p. 69-76.

Malinen, J. (2005). *hostapd*. https://w1.fi/hostapd/. Consulté le 05 novembre 2022.

Mallik, A. (2019). « Man-in-the-middle-attack : Understanding in simple words ». In : *Cyberspace : Jurnal Pendidikan Teknologi Informasi* 2.2, p. 109-134.

Martinez, A., U. Zurutuza, R. Uribeetxeberria, M. Fernández, J. Lizarraga, A. Serna et al. (2008). « Beacon frame spoofing attack detec-

tion in IEEE 802.11 networks ». In : *2008 Third International Conference on Availability, Reliability and Security*. IEEE, p. 520-525.

MATLAB, S. (2012). « Matlab ». In : *The MathWorks, Natick, MA*.

MCCUNE, E. (2000). « DSSS vs. FHSS narrowband interference performance issues ». In : *RF Signal Processing Magazine*.

MEDIA, H. S. M. (s. d.). *Modfied WI-Fi for ham radio*. https://www.qsl.net/kb9mwr/projects/wireless/modify.html. Consulté le 05 novembre 2022.

MOTOROLA (2009). « TIRED OF ROGUES ? Solutions for Detecting and Eliminating Rogue Wireless Networks ». In : *White paper*.

NARDI, T. (août 2019). *RTL-SDR : Seven Years Later*. https://hackaday.com/2019/07/31/rtl-sdr-seven-years-later/. Consulté le 05 novembre 2022.

NASTASIU, C.-I. (2016). « Cyber security strategies in the internet era ». In : *Scientific research and education in the air force-afases*, p. 619-624.

NASTESKI, V. (2017). « An overview of the supervised machine learning methods ». In : *Horizons. b* 4, p. 51-62.

NEUMANN, J. v. (juin 1945). *First draft of a report on the EDVAC*. http://web.mit.edu/STS.035/www/PDFs/edvac.pdf. Consulté le 10 novembre 2022.

NEVEU, L. (déc. 2019). *Internet : 30,4 Mb/s de débit moyen en France, et vous ?* https://www.futura-sciences.com/tech/actualites/internet-internet-304-mb-s-debit-moyen-france-vous-78755/. Consulté le 05 novembre 2022.

Nicopolitidis, P., M. S. Obaidat, G. I. Papadimitriou et A. S. Pomportsis (2003). *Wireless networks*. John Wiley & Sons.

Nikbakhsh, S., A. B. A. Manaf, M. Zamani et M. Janbeglou (2012). « A novel approach for rogue access point detection on the client-side ». In : *2012 26th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, p. 684-687.

Nojeim, G. T. (2010). « Cybersecurity and Freedom on the Internet ». In : *J. Nat'l Sec. L. & Pol'y* 4, p. 119.

Noor, M. M. et W. H. Hassan (2013). « Wireless networks : developments, threats and countermeasures ». In : *International Journal of Digital Information and Wireless Communications (IJDIWC)* 3.1, p. 125-140.

Ornaghi, A. et M. Valleri (2003). « Man in the middle attacks ». In : *Blackhat Conference Europe*. T. 1045.

P0cL4bs (2015). *WiFi-Pumpkin*. `https://github.com/P0cL4bs/WiFi-Pumpkin`. Consulté le 05 novembre 2022.

Parikh, R., A. Mathai, S. Parikh, G. C. Sekhar et R. Thomas (2008). « Understanding and using sensitivity, specificity and predictive values ». In : *Indian journal of ophthalmology* 56.1, p. 45.

Peterson, L. L. et B. S. Davie (2007). *Computer networks : a systems approach*. Elsevier.

Pirayesh, H. et H. Zeng (2022). « Jamming attacks and anti-jamming strategies in wireless networks : A comprehensive survey ». In : *IEEE Communications Surveys & Tutorials*.

Poisel, R. (2011). *Modern communications jamming principles and techniques*. Artech house.

Pouzin, L. (1976). « Virtual circuits vs. datagrams : technical and political problems ». In : *Proceedings of the June 7-10, 1976, national computer conference and exposition*, p. 483-494.

Pujolle, G. (2014). *Les réseaux*. Editions Eyrolles.

Puñal, O., I. Aktas, C.-J. Schnelke, G. Abidin, K. Wehrle et J. Gross (2014). « Machine learning-based jamming detection for IEEE 802.11 : Design and experimental evaluation ». In : *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, p. 1-10. doi : 10.1109/WoWMoM.2014.6918964.

Qu, G. et M. N. Michael (2010). « RAPiD : An indirect rogue access points detection system ». In : *International Performance Computing and Communications Conference*. IEEE, p. 9-16.

Rackley, S. (2011). « Wireless networking technology : From principles to successful implementation ». In.

Raguvaran, S. (2014). « Spoofing attack : Preventing in wireless networks ». In : *2014 International Conference on Communication and Signal Processing*. IEEE, p. 117-121.

Rainie, L., S. Kiesler, R. Kang, M. Madden, M. Duggan, S. Brown et al. (2013). « Anonymity, privacy, and security online ». In : *Pew Research Center* 5.

Rajaraman, V. et N. Adabala (2014). *Fundamentals of computers*. PHI Learning Pvt. Ltd.

Reddy Maddikunta, P. K., G. Srivastava, T. Reddy Gadekallu, N. Deepa et P. Boopathy (2020). « Predictive model for battery life in IoT networks ». In : *IET Intelligent Transport Systems* 14.11, p. 1388-1395.

Reina, D. G., S. L. Toral, F. Barrero, N. Bessis et E. Asimakopoulou (2013). « The role of ad hoc networks in the internet of things : A case scenario for smart environments ». In : *Internet of things and inter-cooperative computational technologies for collective intelligence*. Springer, p. 89-113.

Reyes, H. I. et N. Kaabouch (2013). « Jamming and lost link detection in wireless networks with fuzzy logic ». In : *International Journal of Scientific & Engineering Research* 4.2, p. 1-7.

Romero, G. (2017). « Identication of the impact mechanisms of the electromagnetic interferences on the Wi-Fi communications ». Thèse de doct. UNIVERSITÉ DE LILLE 1 SCIENCES ET TECHNOLOGIES.

RTL-SDR (s. d.). https://www.rtl-sdr.com/about-rtl-sdr/. Consulté le 10 novembre 2022.

Rumale, A. et D. Chaudhari (2011). « Ieee 802.11 x, and wep, eap, wpa/wpa2 ». In : *Tech. Appl* 2.6, p. 1945-1950.

Ryan, J. (2010). *A History of the Internet and the Digital Future*. Reaktion Books.

Sadiku, M. N. et C. M. Akujuobi (2004). « Software-defined radio : a brief overview ». In : *Ieee Potentials* 23.4, p. 14-15.

Salmi, D. (juin 2017). *Grâce à Wi-Fi Inspector, découvrez si votre réseau domestique est vulnérable*. https://blog.avast.com/fr/grace-a-wi-fi-inspector-decouvrez-si-votre-reseau-domestique-est-vulnerable. Consulté le 05 novembre 2022.

Sheng, Y., K. Tan, G. Chen, D. Kotz et A. Campbell (2008). « Detecting 802.11 MAC layer spoofing using received signal strength ». In : *IEEE*

INFOCOM 2008-The 27th Conference on Computer Communications. IEEE, p. 1768-1776.

Shin, Y. Y., J. K. Lee et M. Kim (2018). « Preventing state-led cyberattacks using the bright internet and internet peace principles ». In : *Journal of the Association for Information Systems* 19.3, p. 3.

Siswanto, A., A. Syukur, E. A. Kadir et al. (2019). « Network traffic monitoring and analysis using packet sniffer ». In : *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*. IEEE, p. 1-4.

Sokolova, M. et G. Lapalme (2009). « A systematic analysis of performance measures for classification tasks ». In : *Information Processing & Management* 45.4, p. 427-437.

Soumyalatha, S. G. H. (2016). « Study of IoT : understanding IoT architecture, applications, issues and challenges ». In : *1st International Conference on Innovations in Computing & Net-working (ICICN16), CSE, RRCE. International Journal of Advanced Networking & Applications*. T. 478.

Spitalnick, A. (2009). « Understanding encryption and its role in security ». In : *CPA Prac. Mgmt. F.* 5, p. 5.

Srilasak, S., K. Wongthavarawat et A. Phonphoem (2008). « Integrated wireless rogue access point detection and counterattack system ». In : *2008 International Conference on Information Security and Assurance (isa 2008)*. IEEE, p. 326-331.

Stewart, R. W., K. W. Barlee, D. S. Atkinson et L. H. Crockett (2015). *Software defined radio using MATLAB & Simulink and the RTL-SDR*. 1. Strathclyde Academic Media.

SUFYAN, N., N. A. SAQIB et M. ZIA (2013). « Detection of jamming attacks in 802.11 b wireless networks ». In : *EURASIP Journal on Wireless Communications and Networking* 2013.1, p. 1-18.

SULLIVAN, B. (2007). « Preventing a brute force or dictionary attack : how to keep the brutes away from your loot ». In : *Pridobljeno (17.4. 2014) iz CODE Project : http ://www. codeproject. com/Articles/17111/Preventing-a-Brute-Force-or-Dictionary-Attack-How*.

TANG, H.-R., R.-L. SUN et W.-Q. KONG (2009). « Wireless Intrusion Detection for defending against TCP SYN flooding attack and man-in-the-middle attack ». In : *2009 International Conference on Machine Learning and Cybernetics*. T. 3. IEEE, p. 1464-1470.

TCHAKOUNTÉ, F., M. NAKOE, B. O. YENKE et K. P. UDAGEPOLA (2019). « Recognizing illegitimate access points based on static features : a case study in a campus wifi network ». In : *Int. J. Cyber-Secur. Digit. Forensics* 8.4, p. 279-291.

TEAM, P. (sept. 2018). *Wifipumpkin3 - CLI*. `https://wifipumpkin3.github.io/`. Consulté le 12 septembre 2022.

TECH, A. T. (juin 2016). *Norton Wi-Fi Privacy VPN video - created by Symantec Australia*. `https://www.youtube.com/watch?v=QxHsiSdqKQE`. Consulté le 05 novembre 2022.

TECHNOLOGIES, K. (n.d.[a]). *Analyseur de signaux Keysight PXA N9030A-526*. `https://www.keysight.com/us/en/products/spectrum-analyzers/pxa-signal-analyzer/n9030a-526.html`. Consulté le 10 novembre 2022.

— (n.d.[b]). *Atténuateur Keysight J7211A*. `https://www.keysight.com/us/en/product/J7211A/attenuator.html`. Consulté le 12 mars 2023.

Tektronix AWG70001A (s. d.). https://www.tek.com/awg70001a. Consulté le 10 novembre 2022.

Tulloh, R., H. Putri, D. A. Nurmantris et D. D. Prihatin (2017). « Simulation Wi-Fi Network With Wireless Distribution System (Wds) Topology ». In : *International journal* 15.5.

TURING, A. M. (oct. 1950). « I.—COMPUTING MACHINERY AND INTELLIGENCE ». In : *Mind* LIX.236, p. 433-460. DOI : 10.1093/mind/LIX.236.433.

USRP (s. d.). https://www.ettus.com/all-products/usrp-software-defined-radio-devices/. Consulté le 10 novembre 2022.

Vanhoef, M. et F. Piessens (déc. 2014). « Advanced Wi-Fi Attacks Using Commodity Hardware ». In : *30th Annual Computer Security Applications Conference (ACSAC 2014)*. New Orleans, LA : ACM, p. 256-265. DOI : 10.1145/2664243.2664260.

Vaughan-Nichols, S. J. (2003). « The challenge of Wi-Fi roaming ». In : *Computer* 36.7, p. 17-19.

Villain, J., V. Deniau, A. Fleury, E. P. Simon, C. Gransart et R. Kousri (2019). « EM monitoring and classification of IEMI and protocol-based attacks on IEEE 802.11 n communication networks ». In : *IEEE Transactions on Electromagnetic Compatibility* 61.6, p. 1771-1781.

Villain, J., V. Deniau, C. Gransart, A. Fleury et E. P. Simon (2021). « Characterization of IEEE 802.11 Communications and Detection of Low-Power Jamming Attacks in Noncontrolled Environment Based on a Clustering Study ». In : *IEEE Systems Journal* 16.1, p. 683-692.

VOIGT, P. et A. VON DEM BUSSCHE (2017). « The eu general data protection regulation (gdpr) ». In : *A Practical Guide, 1st Ed., Cham : Springer International Publishing* 10.3152676, p. 10-5555.

VUGDELIJA, N., N. NEDELJKOVIĆ, N. KOJIĆ, L. LUKIĆ et M. VESIĆ (s. d.). « REVIEW OF BRUTE-FORCE ATTACK AND PROTECTION TECHNIQUES ». In : ().

WALLISER, A. (1989). « Le rapport" Nora-Minc". Histoire d'un best-seller ». In : *Vingtieme siecle. Revue d'histoire*, p. 35-47.

WANG, L. et A. M. WYGLINSKI (2016). « Detection of man-in-the-middle attacks using physical layer wireless security techniques ». In : *Wireless Communications and Mobile Computing* 16.4, p. 408-426.

WEBB, G. I., E. KEOGH et R. MIIKKULAINEN (2010). « Naive Bayes. » In : *Encyclopedia of machine learning* 15, p. 713-714.

WRIGHT, R. E. (1995). « Logistic regression. » In.

XIA, H. et J. C. BRUSTOLONI (2005). « Hardening web browsers against man-in-the-middle and eavesdropping attacks ». In : *Proceedings of the 14th international conference on World Wide Web*, p. 489-498.

YE, Y., S. CI, A. K. KATSAGGELOS, Y. LIU et Y. QIAN (2013). « Wireless video surveillance : A survey ». In : *IEEE Access* 1, p. 646-660.

YOUNES, C. : P. B. (mars 2020a). *WLAN PARTIE 3 - 1 : Les modes et les mécanismes d'association*. https://www.youtube.com/watch?v=fMx7tNOyNcY. Consulté le 05 novembre 2022.

— (mars 2020b). *WLAN PARTIE 4 : LA COUCHE PHYSIQUE 802.11*. https://www.youtube.com/watch?v=PhqES6sQAno. Consulté le 05 novembre 2022.

Younes, C. : P. B. (avr. 2020c). *WLAN PARTIE 5 : LA COUCHE LIAISON DE DONNÉES*. https://www.youtube.com/watch?v=CzhFjs4Wy4w. Consulté le 05 novembre 2022.

— (avr. 2020d). *WLAN PARTIE 6 : La sécurité dans les Réseaux Wifi*. https://www.youtube.com/watch?v=HgTUr5TRm5U. Consulté le 05 novembre 2022.

Zhang, Y. et Q. Yang (2018). « An overview of multi-task learning ». In : *National Science Review* 5.1, p. 30-43.

Zhong, Z., P. Kulkarni, F. Cao, Z. Fan et S. Armour (2015). « Issues and challenges in dense WiFi networks ». In : *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, p. 947-951.

Zou, Y., J. Zhu, X. Wang et L. Hanzo (2016). « A survey on wireless security : Technical challenges, recent advances, and future trends ». In : *Proceedings of the IEEE* 104.9, p. 1727-1765.