Faille dans le protocole WPA-WPA2

Guides et Tutoriels

Gentoo-Québec

version 2.00

Copyright du document

Note au lecteur

Nous ne sommes pas responsables des bris ou des pertes de données.

Ce document a été créé avec LaTeX 2_ε.

Table des matières

1 Introduction

Au cours de cet article, nous allons voir la possibilité de faiblesse dans le protocole WPA/WPA2 qui pourrait être utilisé pour découvrir une clé WPA/WPA2.

2 Mise en Contexte

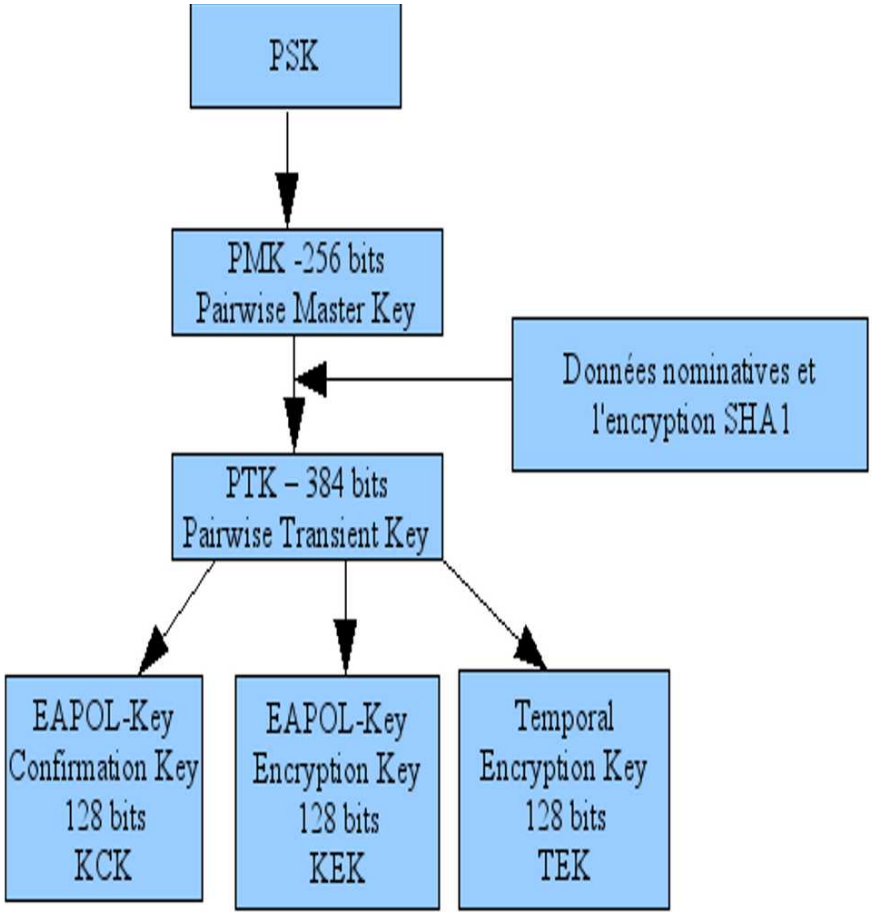
Tout d'abord, cette possibilité de faiblesse peut être contrée facilement si vous utilisez l'algorithme d'encryption(AES/CCMP) combiné avec une clé hexadécimal composée de lettres et de chiffres d'une longueur entre 8 caractères et 63 caractères. De plus, vous devez vous assurer que votre clé ne forme pas de mot qui pourrait se retrouver dans un dictionnaire.

3 Description technique des clés utilisées par WPA/WPA2

Tout d'abord, le protocole WPA/WPA2 utilise une série de clé de cryptage qui sont généralisées et aléatoire tout au long du protocole. De plus, la durée de vie des clés aléatoire est aussi limitée.Cette méthode rend impossible la détection d'une clé WPA/WPA2 contrairement au protocole WEP.

En terme clair, même si on écoute sur un réseau qui utilise l'encryption WPA/WPA2, nous n'augmentons pas nos chances pour autant(c'est techniquement une demi-vérité). De plus, je ne vais pas aborder l'aspect de l'algorithme TKIP, car à mon avis, cet algorithme existe seulement pour faire fonctionner des vieilles cartes réseaux Wifi qui supportent seulement l'encyption WEP avec l'algorithme RC4.

Fig. 1: *Description des clés généralisées et aléatoire*



1. PSK correspond à la Pre-Shared Key, soit la clé alphanumérique que WPA2 utilise.
2. PMK correspond à la PSK.
3. PTK correspond à la combinaison de la PMK, suivi de l'adresse MAC du Acces Point(Routeur), l'adresse MAC du client, nombre aléatoire côté Acces Point(ANonce),nombre aléatoire côté client(SNonce) le tout crypté avec l'algorithme SHA1. Ces données sont considérées sont des données nominative.
4. La clé PTK est en fait séparée en 3 clés distinctes.
5. KCK correspond à la clé de confirmation qui est en fait la clé d'authentification des messages (MIC).

6. KEK correspond à la clé qui est utilisée pour encrypter les données lors du Handshake.
7. TEK correspond à la clé qui est utilisée pour encrypter les données après le Handshake.

4 Description technique du Handshake WPA/WPA2.

Le protocole utilise un 4 Way HandShake pour pouvoir démarrer une communication sécurisée entre un client et un Acces Point. En effet, c'est le seul moment dans la communication où il s'échange des informations pour pouvoir se connecter avec succès et c'est la partie la plus importante du protocole et surtout la plus compliquée. De plus, c'est précisément à ce moment là que les Hackeurs veulent collecter des informations précieuses qu'on verra ensemble plus tard.

Le 4 Way HandShake se compose de 4 parties :

1. Accord au niveau du protocole de sécurité utilisé(Agreeing on the security policy)
2. (802.X Authentification)
3. Génération des clés et distributions (Key hierarchy and distribution)
4. (RSNA data confidentiality and integrity)

4.1 Étape 1 : Agreeing on the security policy

Cette étape se passe de la manière suivante :

1. Le client intérroge le Acces Point pour savoir quels sont les protocoles de sécurité qui sont supportés par celui-ci.Par exemple (Encryption WPA2,algorithme CCMP).
2. Le Acess Point envoit une trame qui contient en clair sa Mac Adress et son nombre aléatoire côté Acces Point(Random ANONCE) au Client.
3. Le client envoit une requête d'association à l'Acces Point et comme de quoi c'est conforme et il passe à l'étape 2.

4.2 Étape 2 : 802.X Authentification

Cette étape se passe de la manière suivante :

1. Le Acces Point envoit une requête au client pour qu'il s'identifie en utilisant le protocole choisi.
2. Ensuite il y a un échange de 2 trames pour s'échanger la Master Key (MK).Cette clé va être identique autant du côté Acces Point que du côté client.
3. Le client envoit aussi une trame qui contient en clair sa Mac Adress et son nombre aléatoire côté client (Random SNONCE).

4.3 Étape 3 : Key hierarchy and distribution

Cette étape est vraiment le coeur de la communication.

Avant de commencer cette partie, il faut se rappeler ceci :

Cette étape permet de créer des clés temporaires pour la session qui a passé l'étape 2. Ces clés ont des durées de vie limités et à chaque intervalle de X secondes, ils vont se regénérer automatiquement, aussi longtemps qu'une demande de déconnexion ne sera pas demandée.

Cette étape se passe de la manière suivante :

Tout d'abord la clé PMK est comme vous le savez la clé PSK. De plus, cette clé PSK doit être identique et présente autant du côté du Acces Point que du côté du client. Cela étant dit, on continu notre exploration.

Partie Client :

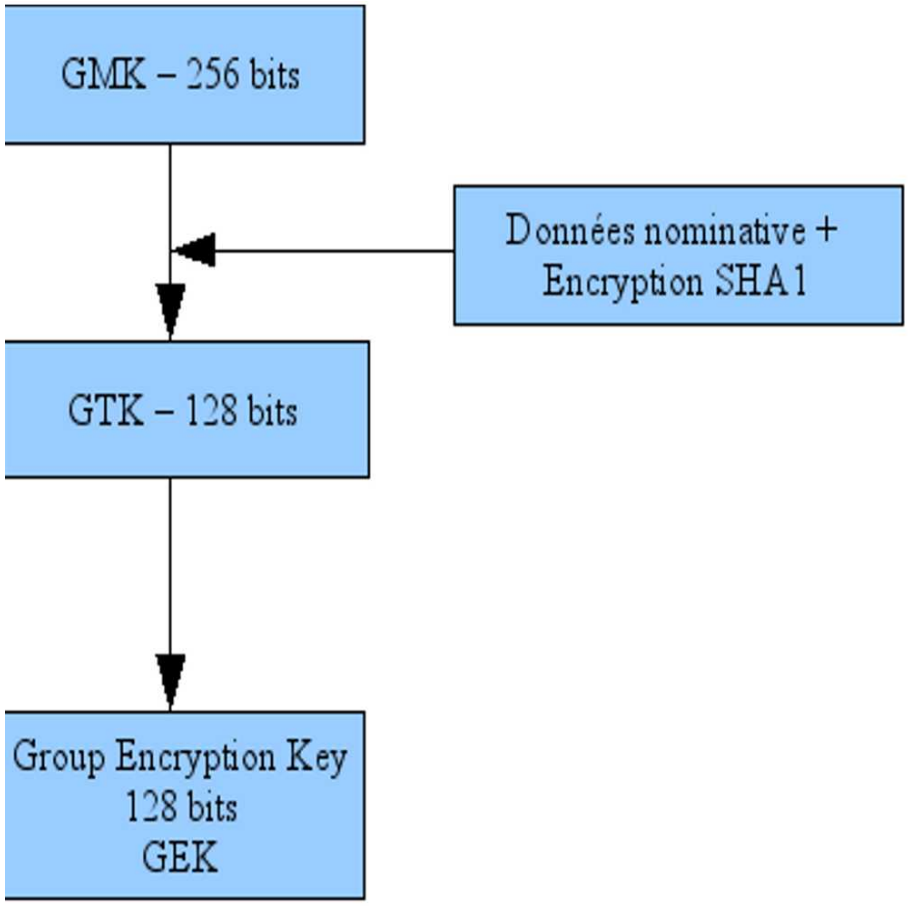
1. Génération de la clé PTK à partir de la clé PSK avec les données nominatives.
2. Génération des clés résultantes KCK,KEK et TEK.
3. Encryption de la partie message avec la clé KEK. Le résultat sera envoyé dans un champ précis dans la trame (MIC).
4. Envoit d'une trame vers le Acces Point.

Partie Acces Point :

1. Réception de la trame venant du client.Puisque le Acces Point veut vérifier l'identité du client, il va générer les clés PTK,KCK,KEK et TEK.
2. Décryptage du champ MIC pour vérifier que le client a bien encrypté le message avec les bonnes clés PTK,KCK,KEK et TEK.
3. Le Acces Point envoit une trame au client qui contient une nouvelle clé, soit la GTK (Group Transient Key).

Cette clé est composée des éléments suivant (Données nominatives) : Mac adresse de l'Acces Point,Nombre aléatoire venant du AP le GNonce, une clé aléatoire GMK(Group Master Key) le tout encrypté avec SHA1. De plus, il y a bien sûr le champ MIC qui a été calculé avec le message et surtout il a été encrypté avec la clé KCK.

Fig. 2: *Description du Handshake à l'étape 3*



Ensuite, le client reçoit cette trame et la première tâche qu'il exécute, c'est de vérifier le champ Mic pour savoir si le Acces Point a bien la bonne valeur de clé PMK et par conséquent les bonnes valeurs des clés PTK,KCK,KEK,TEK. On voit bien qu'il y a autre niveau de clé pour assurer la sécurité du protocole et c'est encore des clés aléatoires.

4.4 Étape 4 : RSNA data confidentiality and integrity

La dernière étape consiste à confirmer que le Handshake c'est réalisé avec succès.

Le Acces Point et le client vont s'échanger une trame qui va contenir la clé GEK (Group Encryption Key) qui va être utilisée par la suite pour crypter les données.

De plus, cette fameuse clé(GEK) va être regénérée à tous les 3600 secondes maximum pour s'assurer d'avoir un changement de clé régulièrement.

Voilà ce qui complète un HandShake de départ.

5 Qu'est-ce qui se passe quand le délai d'expiration de la clé GTK est atteint ?

Ce processus se découpe en 2 grandes étapes :

Étape 1 :

Côté Client :

1. Le client prend connaissance que sa clé GTK n'est plus valide pour la communication en cours.
2. Le client envoit une trame à l'Acces Point pour lui indiquer qu'il veut renouveller sa clé GTK. Côté Acces Point :

1. Le Acces Point envoit une trame au client avec un nouveau Nombre aléatoire venant du Acces Point le AP(Random ANONCE).
2. Le Acces Point envoit aussi la nouvelle valeur de la clé GTK le tout encrypté avec la clé KEK. De plus, il y a bien sûr le champ MIC qui a été calculé avec le message et surtout il a été encrypté avec la clé KCK.
3. Le client reçoit cette trame et la première tâche qu'il exécute, c'est de vérifier le champ Mic pour savoir si le Acces Point a bien la bonne valeur de clé PMK et par conséquent les bonnes valeurs des cléè PTK,KCK,KEK,TEK.
4. Il va décrypté la nouvelle clé GTK en utilisant la clé KEK.

Étape 2 : La dernière étape consiste à confirmer que la demande de changement de clés c'est réalisé avec succès. Ensuite les tramisssions de données peuvent recommencer normalement.

6 Alors oû ce trouve la possibilité de faiblesse dans ce protocole ?

Tout d'abord, il ya une belle formule pour expliquer le tout. En effet, la base est celle-ci : Formule 1 :

```
PTK = (PMK, Min(Mac Adresse du Acces Point,STA_MAC) ||

       MAX(Mac Adresse du Acces Point,STA_MAC)      ||

       MIN(Nombre aléatoire venant du Acess Point,Nombre aléatoire venant du Client)

       MAX(Nombre aléatoire venant du Acess Point,Nombre aléatoire venant du Client)
```

Note à propos de cette formule :
- La variable PMK est en fait la variable PSK.
- La variable STA_MAC doit être considérée comme ayant la valeur 0,car elle existe seulement quand on fait du Wifi en mode Ad-hoc.

7 Alors est-ce que vous commencez à comprendre pourquoi un Hacker veut à tout prix collecter les 4 étapes du Handshake ?

Nous avons vu plus tôt que les variables sont envoyés en clair dans les trames sauf la variable PMK. Techniquement, après une capture du 4 ways HandShake entre un client et son Acess Point, le Hacker pour commencer à essayer de trouver la clé PSK, car il a tout ce qu'il faut pour attaquer.

Par contre, comme vous pouvez le constater, il n'y a aucun moyen de déduire la clé PSK. Alors on doit utiliser une force brute pour casser la clé.

Le logiciel Aircrack utilise les 4 étapes pour collecter les informations nécessaire et ensuite le logiciel passe à l'attaque.

Le type d'attaque est de type dictionnaire, car vu qu'il n'y a aucun moyen de déduire la clé PSK, le logiciel utilise un fichier .txt qui contient des dizaines de milliers de mots et il va essayer chaque mot qui correspond à la clé PSK qui a été inséré dans la formule pour générer la clé PTK.

Le seul moyen de savoir si la clé PSK suggérée est valide, Aircrack doit décrypter

le champ MIC avec la clé KCK qui à été générée avec sa PTK venant de l'étape 2 lors du 4 HandShake.

De plus, il y a aussi un problème de taille au niveau technique qui est causé par la conception du protocole WPA2. En effet, le protocole WPA/WPA2 utilise un algorithme qui complique la décryption de clé WPA/WPA2 de façon importante.

Le protocole utilise la formule suivante :

Formule 2 :

PSK = PMK = PBKDF2(password,SSID,longueur du SSID,4096,256).

Où PBKDF2 est une méthode utilisée par PKCS#5, 4096 est le nombre possible de hashes par clé et 256 est la longueur de la clé PMK.

En terme clair, pour chaque mot dans le dictionnaire, Aircrack doit effectuer la formule 4096 fois, car pour chaque mot, il y a 4096 hash possible. Alors ce genre de calcul ralentit pas mal la vitesse de Aircrack. On parle d'envrion 300 mots à la seconde que Aircrack peut tester. Petit rappel à propos des hash, on peut les crypter seulement, on ne peut pas les décrypter.

De plus, ce qui suit va intéresser plusieurs personnes. Comme vous le savez, pour avoir un chance de décrypter une clé WPA/WPA2, on doit capter un 4 Way HandShake et cela peut prendre des jours si vous écoutez un réseau où le client reste tout le temps branché. Aircrack met à la disposition une option qui permet d'envoyer une trame au Acces Point qui va faire en sorte que le client va se faire déconnecter du Acess Point contre son gré. Et si vous êtes chanceux, le client va vouloir se reconnecter automatiquement...alors vous allez pouvoir capter votre 4 Way HandShake.

En terminant, j'espère que vous avez compris quelque chose dans cet article, par contre je sais que c'est assez compliqué merci.