HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Sécurité des réseaux sans fil

Laboratoire WEP avancé

Professeur
Abraham Rubinstein
abraham.rubinstein@heig-vd.ch

Assistant
Yohan Martini
yohan.martini@heig-vd.ch

Février 2018 – Juin 2018

Pour cette partie pratique, vous devez être capable de :

1. Déchiffrer **manuellement** des trames WEP utilisant Python et Scapy
2. Chiffrer **manuellement** des trames WEP utilisant Python et Scapy
3. Forger des fragments protégés avec WEP afin d'obtenir une keystream de longueur plus grande que 8 octets
4. Contourner l'authentification « clé partagée » (bonus)

Vous allez devoir faire des recherches sur internet pour apprendre à utiliser Scapy et la suite aircrack pour vos manipulations. Il est **fortement conseillé** d'employer une distribution Kali. Si vous utilisez une VM, il vous faudra une interface WiFi usb, disponible sur demande.

ATTENTION : Pour vos manipulations, il pourrait être important de bien fixer le canal lors de vos captures et vos injections. Si vous en avez besoin, la méthode la plus sure est d'utiliser l'option :

`--channel` de `airodump-ng`

et de garder la fenêtre d'airodump ouverte en permanence pendant que vos scripts tournent ou vos manipulations sont effectuées.

Pour les interfaces Alfa **AWUS036ACH (interfaces noires)**, **il faut activer la compatibilité USB 3.0 sur votre VM**. Il faudra faire les manipulations suivantes pour les configurer en mode monitor (**pour les autres interfaces, se renseigner sur Internet**) :

Installer le driver (disponible sur Kali. Pour d'autres distributions, il faudra probablement le compiler à partir des sources) :

```
sudo apt-get install realtek-rtl88xxau-dkms
```

Ensuite, pour passer en mode monitor :

Mettre l'interface "down"

```
sudo ip link set wlan0 down
```

#Configurer le mode monitor

```
sudo iwconfig wlan0 mode monitor
```

A la fin de cette procédure, vous aurez une interface « wlan0 » en mode monitor (et non pas wlan0mon comme c'est souvent le cas avec d'autres interfaces).

Si vous devez compiler le driver :

```
git clone https://github.com/astsam/rtl8812au.git
cd rtl8812au
make
sudo make install
```

1 Déchiffrement manuel de WEP

Dans cette partie, vous allez récupérer le script Python « `manual-decryption.py` » disponible sur `eistore`. Il vous faudra également le fichier de capture « `arp.cap` » contenant un message arp chiffré avec WEP et la librairie « `rc4.py` » pour générer les keystreams indispensables pour chiffrer/déchiffrer WEP. Tous les fichiers doivent être copiés dans le même répertoire local sur vos machines.

a) Ouvrir le fichier de capture « `arp.cap` » avec Wireshark
b) Utiliser Wireshark poou déchiffrer la capture. Pour cela, il faut configurer dans Wireshark la clé de chiffrement/déchiffrement WEP (Dans Wireshark : Preferences→Protocols→IEEE 802.11→Decryption Keys). Il faut également activer le déchiffrement dans la fenêtre IEEE 802.11 (« Enable decryption »). Vous trouverez la clé dans le script Python « `manual-decryption.py` »
c) Exécuter le script avec « `python manual-decryption.py` »
d) Comparer la sortie du script avec la capture text déchiffrée par Wireshark
e) Analyser le fonctionnement du script

2 Chiffrement manuel de WEP

Utilisant le script « `manual-decryption.py` » comme guide, créer un nouveau script « `manual-encryption.py` » capable de chiffrer un message, l'enregistrer dans un fichier pcap et l'envoyer.
Vous devrez donc créer votre message, calculer le contrôle d'intégrité (ICV), et les chiffrer (voir slides du cours pour les détails).

Quelques éléments à considérer :

- Vous pouvez utiliser la même trame fournie comme « template » pour votre trame forgée (conseillé). Il faudra mettre à jour le champ de données qui transporte le message (`wepdata`) et le contrôle d'intégrite (`icv`).
- Le champ « `wepdata` » accepte des données en format text.
- Le champ « `icv` » accepte des données en format « long ».
- Vous pouvez vous guider à partir du script fourni pour les différentes conversions de formats qui pourraient être nécessaires.
- Vous pouvez exporter votre nouvelle trame en format pcap utilisant Scapy et ensuite, l'importer dans Wireshark. Si Wireshark est capable de déchiffrer votre trame forgée, elle est correcte !

3 Fragmentation

Dans cette partie, vous allez enrichir votre script développé dans la partie précédente pour chiffrer 3 fragments.

Quelques éléments à considérer :

- Chaque fragment est numéroté. La première trame d'une suite de fragments a toujours le numéro de fragment à 0. Une trame entière (sans fragmentation) comporte aussi le numéro de fragment égal à 0
- Pour incrémenter le compteur de fragments, vous pouvez utiliser le champ « SC » de la trame. Par exemple : « `trame.SC += 1` »
- Tous les fragments sauf le dernier ont le bit « more fragments » à 1, pour indiquer qu'un nouveau fragment va être reçu
- Le champ qui contient le bit « more fragments » est disponible en Scapy dans le champ « `FCfield` ». Il faudra donc manipuler ce champ pour vos fragments. Ce même champ est visible dans Wireshark dans IEEE 802.11 Data → Frame Control Field → Flags
- Pour vérifier que cette partie fonctionne, vous pouvez importer vos fragments dans Wireshark, qui doit être capable de les recomposer
- Pour un test encore plus intéressant (optionnel), vous pouvez utiliser un AP (disponible sur demande) et envoyer vos fragments. Pour que l'AP accepte vous données injectées, il faudra faire une « fake authentication » que vous pouvez faire avec `aireplay-ng`
- Si l'AP accepte vos fragments, il les recomposera et les retransmettra en une seule trame non-fragmentée !

4 Shared-key fake authentication

ATTENTION : il y aura un bonus de 0.5 points dans le TE1 pour la première équipe qui rendra cet exercice terminé. Un bonus de 0.2 points pour les équipes suivantes

Cet exercice nécessite l'utilisation d'un AP WEP configuré en mode d'authentification clé partagée (disponible sur demande).

Le but c'est de réussir une authentification auprès de l'AP sans connaître la clé WEP. Vous devrez :

- Configurer l'AP en WEP avec une clé connue, configurer l'authentification clé partagée et capturer un processus d'authentification et l'enregistrer dans un fichier .cap (voir fichier exemple « exercice_4.cap » dans eistore).
- Ensuite, vous devez développer un script capable d'utiliser votre capture pour réaliser une authentification **sans utiliser la clé WEP**.

<u>Quelques éléments à considérer</u> :

- La plupart des problèmes provient souvent du formatage de l'ICV (endianness, format, etc.). C'est donc une voie à explorer si vous n'arrivez pas à communiquer correctement avec l'AP (voir les scripts de base, chercher sur Internet, etc.)
- Pour le challenge « authentification », vous devez « entamer une conversation » avec l'AP. Scapy fournit une commande capable d'envoyer un seul paquet, en attendre la réponse et continuer l'exécution du script
- Attention à la taille du challenge et de la réponse chiffrée… !!!

Livrables

Un fichier zip contenant :

- Script de chiffrement WEP **<u>abondamment commenté/documenté</u>**
  - Fichier pcap généré par votre script contenant la trame chiffrée
  - Capture d'écran de votre trame importée et déchiffré par Wireshark
- Script de fragmentation **<u>abondamment commenté/documenté</u>**
  - Fichier pcap généré par votre script contenant les fragments
  - Capture d'écran de vos trames importées et déchiffrés par Wireshark
- (**Challenge <u>optionnel</u>**) Script d'authentification clé partagée **<u>abondamment commenté/documenté</u>**
  - Capture d'écran de votre authentification vue par Wireshark

Echéance

Le 9 avril 2018 à 18h00