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

Sécurité des réseaux sans fil

Laboratoire WPA 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. Extraire à partir d'une capture Wireshark les données nécessaires pour dériver les clés de chiffrement et intégrité WPA utilisant Scapy
2. Coder votre propre version d'aircrack pour trouver la passphrase d'un réseau WPA à partir d'une capture utilisant Python et Scapy
3. Coder votre propre version d'airodump et aireplay pour déauthentifier un client, sniffer un handshake et l'utiliser pour trouver une passphrase WPA utilisant Python et Scapy (Option bonus)

Il est **fortement conseillé** d'employer une distribution Kali. Si vous utilisez une VM, il vous faudra une interface WiFi usb, disponible sur demande.

Les fichiers nécessaires pour ce laboratoire sont disponibles sur :

`//eistore1/profs/ARS/cours/SWI/2.Labo/3.WPA/`

ATTENTION : Pour l'exercice 3 (bonus), il est très 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.

1 Obtention des paramètres pour la dérivation des clés WPA

Dans cette première partie, vous allez récupérer le script Python `wpa_key_derivation.py` disponible sur eistore1. Il vous faudra également le fichier de capture `wpa_handshake.cap` contenant un processus d'authentification WPA. Vous aurez aussi besoin du fichier `pbkdf2_math.py`. Tous ces fichiers doivent être copiés dans le même répertoire.

a) Ouvrir le fichier de capture `wpa_handshake.cap` avec Wireshark
b) Exécuter le script avec python `wpa_key_derivation.py`
c) Essayer d'identifier les valeurs affichées par le script dans la capture Wireshark
d) Analyser le fonctionnement du script. En particulier, **faire attention** à la variable `data` qui contient la payload de la trame et la comparer aux données de la quatrième trame du 4-way handshake. Lire la fin de ce document pour l'explication de la différence.
e) **Modifier le script** pour qu'il récupère automatiquement, à partir de la capture, les valeurs qui se trouvent actuellement codées en dur (ssid, APmac, Clientmac, nonces…)

2 Scaircrack (aircrack basé sur Scapy)

Aircrack utilise le quatrième message du 4-way handshake pour tester les passphrases contenues dans un dictionnaire. Ce message ne contient pas de données chiffrées mais il est authentifié avec un MIC qui peut être exploité comme « oracle » pour tester des clés différentes obtenues des passphrases du dictionnaire.

Utilisant le script `wpa_key_derivation.py` comme guide, créer un nouveau script `scaircrack.py` qui doit être capable de :

a) Lire une passphrase à partir d'un fichier (wordlist)
b) Dériver les clés à partir de la passphrase que vous venez de lire et des autres éléments nécessaires contenus dans la capture (cf exercice 1)
c) Récupérer le MIC du dernier message du 4-way handshake dans la capture
d) Avec les clés dérivées à partir de la passphrase, nonces, etc., calculer le MIC du dernier message du 4-way handshake à l'aide de l'algorithme Michael (cf l'explication à la fin de ce document)
e) Comparer les deux MIC
    a. Identiques → La passphrase utilisée est correcte
    b. Différents → Essayer avec une nouvelle passphrase

3 Scairodump (Bonus 0.5 points dans le TE2)

Modifier votre script de cracking pour qu'il soit capable de faire les mêmes opérations que le script précédant mais sans utiliser une capture Wireshark. Pour cela, il faudra donc sniffer un 4-way handshake utilisant Scapy et refaire toutes les opérations de la partie 2 pour obtenir la passphrase. Le script doit implémenter la possibilité de déauthentifier un client pour stimuler le 4-way handshake. Cette déauthentification doit aussi être implémentée avec Scapy.

Quelques détails importants

- Le calcul du MIC peut utiliser MD5 (WPA) ou SHA-1 (WPA2). Le 4-way handshake contient les informations nécessaires dans le champ Key Information
- La commande : `a2b_hex(variable)` est équivalente à `variable.decode("hex")`
- La commande `b2a_hex(variable)` est équivalente à `variable.encode("hex")`
- Le dernier message du 4-way handshake contient un MIC dans sa payload. Pour calculer vous-même votre MIC, vous devez mettre **les octets du MIC** dans cette payload à `\x00`

Livrables

Un fichier zip contenant :

- Script `wpa_key_derivation.py` **modifié** pour la récupération automatique des paramètres à partir de la capture
- Script `scaircrack.py` **abondamment commenté/documenté** + fichier wordlist
    - Capture d'écran de votre script en action
- (Bonus) Script `scairodump.py` **abondamment commenté/documenté**
    - Capture d'écran de votre script en action

Echéance

Le 8 mai 2018 à 18h00