Le sidejacking avec pycookiejsinject

Adil Alhima (adilalhima[at]gmail.com)

Blog: http://actunux.blogspot.com

facebook[dot]com/adilgeek

Table de matière :

Introduction :

un grand nombre d'internautes néglige la nécessité de l'utilisation du protocole https qui assure le chiffrement des données échangées entre leurs navigateurs et les serveurs web, lors de la consultation des emails(Hotmail, Yahoo..) ou l'accès aux sessions des réseaux sociaux favoris(facebook, twitter..) par le protocole http via des réseaux publics(hotspots, wifi), le risque d'être victime d'une attaque de type sidejacking est très élevé, dans cet article je vais essayer d'expliquer la facilité d'avoir des accès non autorisés aux sessions d'une victime connectée sur le même réseau LAN que l'attaquant.

Qu'est ce que le sidejacking :

Le sidejacking est une attaque très connue dans les milieux de la sécurité informatique, elle date à l'apparition du principe des cookies, elle se résume par la récupération d'une copie des cookies de sessions de la victime, et de les utiliser pour une usurpation d'identité afin de se logger comme si l'identifiant et le mot de passe ont été validés.

Présentation de pycookiejsinject :

Pycookiejsinject est un sniffer programmé par diogo mónica, il génère un script javascript à injecter dans la barre d'adresse du navigateur, disponible en téléchargement sur le lien suivant « https://github.com/diogomonica/py-cookieJsInjection », contrairement aux autres utilitaires dédiés au sidejacking comme hamster/ferret et firesheep, pycookiejsinject a les avantages suivants:

-écrit en python, langage installé par défaut dans la majorité des distributions Gnu/Linux.

-dépend seulement des deux paquets scpay et tcpdump.

-disponible pour Gnu/Linux et OSX version 10.6+.

-tous les navigateurs peuvent t'être utilisé pour cette attaque.

Le fonctionnement de pycookiejsinject est en ligne de commande, les paramètres passés sont l'interface réseau choisie pour le sniff des cookies de sessions et l'autre facultative pour filtrer un domaine précis, par exemple:

```
root@adhima-laptop:~#python cookiejsinject wlan0 -domaine
```

-wlan0: est l'interface réseau dont cookiejsinject se met à l'écoute du trafic, pour capture des cookies de sessions.
-domaine: l'ajout de ce paramètre permet de filtrer les cookies capturer selon le domaine.

Démonstration :

La démonstration expliquée ci-dessous est testée sur une distribution Debian squeeze avec un navigateur iceweasel, réalisée sur mon propre réseau local dont ma propre session facebook utilisant le protocole http, est la cible, cette session ouverte depuis une machine avec l'adresse ip 192.168.1.14 par le navigateur IE8.(image1)

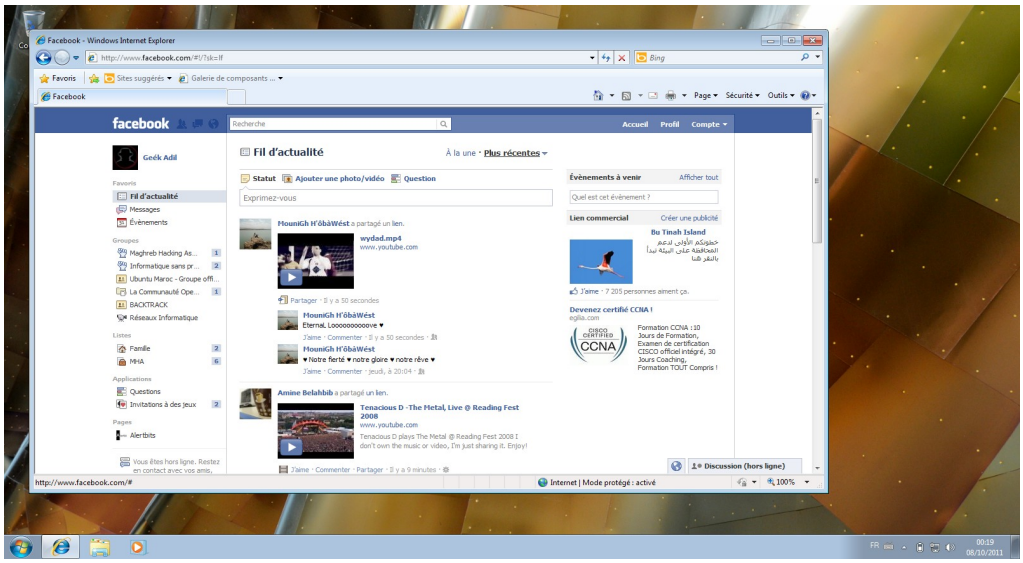


image1

-L'attaque est composée de deux partie:

1-la première est une attaque de type empoisonnement du cache ARP (ARP cache poisoning), le but est de faire passer le trafic de la victime sur la carte réseau de la t'attaquant, arpspoof de Dsniff est l'utilitaire utilisé.

2-la deuxième partie de l'attaque consiste à mettre pycookiejsinject en sniff sur la carte réseau connectée au réseau LAN, pour interception des cookies de sessions échangés entre le navigateur de la victime et le serveur web, pycookiejsinject retourne un résultat comme un script javascript contenant les cookies de sessions de la victime, il suffit de copier et coller le résultat sur la barre d'adresse du navigateur, et de valider avec la touche Entrée, de préférence il est conseillé de vider le cache du navigateur avant de faire valider le script retourné par pycookiejsinject.

Exemple:

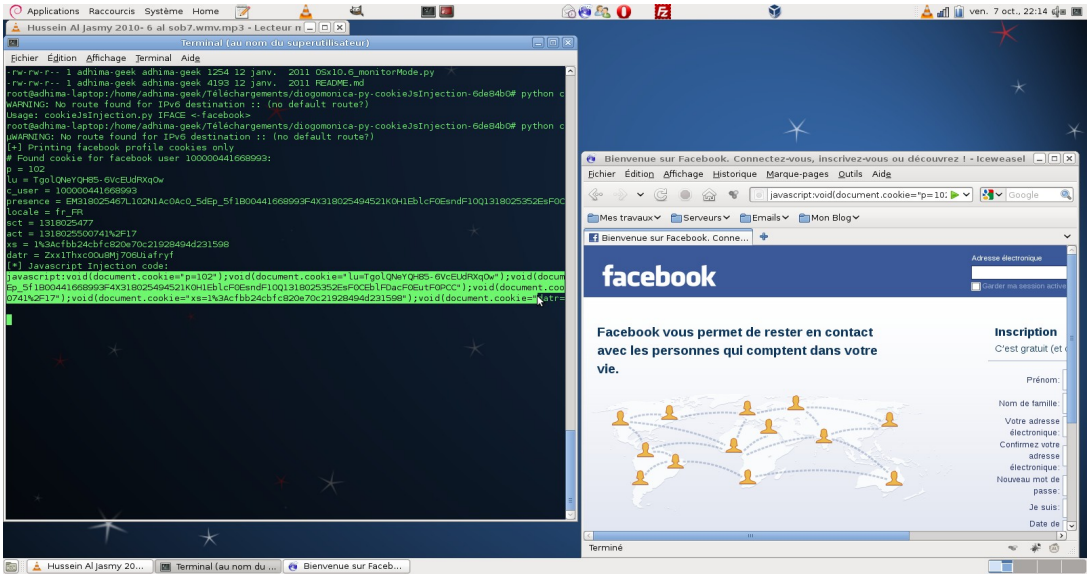
1-)empoisonnement du cache ARP.

```
root@adhima-laptop:~#arpspoof -t 192.168.1.1 192.168.1.14

root@adhima-laptop:~#arpspoof -t 192.168.1.14 192.168.1.1

root@adhima-laptop:~#echo 1 > /proc/sys/net/ipv4/ip_forward
```

2-)sniff des cookies de sessions de la victime.

```
root@adhima-laptop:/home/adhima-geek/diogomonica-py-
cookieJsInjection-6de84b0#python cookieJsInjection.py
wlan0 -facebook
```

En ouvrant ma propre session facebook sur la machine avec l'adresse ip 192.168.1.14, pycookiejsinject retourne un résultat comme sur l'image ci-dessous (image2).



[image2]

comme décrit en haut, il suffit de copier et coller le script javascript sur la barre d'adresse du navigateur, puis la touche Entrée, et enfin entrer le site dont on a sniffer les cookies, pour notre exemple c'est www.facebook.com, la session s'ouvre comme si l'identifiant et le mot de passe ont été entrés correctement.

remarque:

le cookie capturé en haut est c_user, utilisé par facebook comme cookie de session.

Protection :

-configurer dans les paramètres de la session, l'utilisation du protocole https, s'il est proposé par le serveur web.
-utiliser xarp, tool qui permet la détection des attaques empoisonnement du cache ARP.
-des extensions sur firefox, permettent de forcer l'utilisation de l'https avec les sites web, comme HTTPS everywhere.