TP3

Simulation des attaques (ARP, DNS, PROXY)

Dans ce TP () nous allons changer les adresses ip de notre réseau local :

- L'adresse principal de réseau : 192.168.5.0
- Ladresses de machine KALI : 192.168.5.99

- Les entrées **ARP** peut facilement être manipulées en utilisant des paquets de données falsifiées. On parle alors **d'ARP spoofing** (de l'anglais *spoof*, qui signifie *échanger*), un type **d'attaque de l'homme du milieu**, qui permet aux pirates d'échanger deux systèmes de communication en passant inaperçus.

- La finalité de l'empoisonnement **DNS** est d'acheminer les utilisateurs vers un site Web frauduleux. Par exemple, un utilisateur tape « gmail.com » dans un navigateur Web avec pour objectif d'aller consulter sa boîte email. Le DNS ayant été empoisonné, ce n'est pas la page gmail.com qui s'affiche mais une page frauduleuse choisie par l'attaquant, dans le but par exemple de récupérer les accès aux boîtes emails. Les utilisateurs saisissant le nom de domaine correct, ils ne se rendent pas compte que le site Web qu'ils visitent est un faux, une escroquerie.
- Les **proxy** (« serveurs mandataires ») permettent de faire du filtrage intelligent. Il ne permet aucune connexion sur une machine locale à partir de l'extérieur. Le proxy gère toutes les connexions au nom des machines de réseau local.
- **Toutes les manipulations sont bien évidement à réaliser dans un réseau local, dont vous êtes le propriétaire.**

1- Objectifs de ce TP :

- Implémenter quelques attaques et les tester
- Mise en place de quelques attaques en utilisant des outils d'attaques

2- Outils logiciels :

KALI Linux, utilitaire, bettercap

1. Prise en main de bettercap sur kali linux pour des simulation des attaques.

Dans la machine KALI, nous allons prendre en main l'utilitaire Bettercap (version actuelle : 2.X) pour la supervision et les attaques réseaux.

Config de base :

Pour l'installation, je vous invite à suivre leur github, qui est très bien documentée : https://github.com/bettercap/bettercap

1- Tout d'abord, pour lancer bettercap, il suffit de lancer la commande suivante :

sudo bettercap

2- Avant de commencer à voir les différentes attaques, il faut savoir que nous allons utiliser des « **modules** », et ces derniers peuvent se configurer en utilisant la commande « **set** » :

set dns.spoof.domain toto.com

Et pour lancer le module, il suffit simplement d'appeler le module et d'ajouter l'argument « **on** » ou « **off** »

dns.spoof on

2. Attaque ARP

Cette attaque consiste à empoisonner le cache ARP de la machine cible afin de permettre de router les paquets vers la machine pirate.
En effet, dans un réseau local, toutes les communications se font via les adresses MAC des machines. L'attaque ARP va permettre de modifier la table d'adresse MAC de la victime.

Il est judicieux de modifier l'adresse MAC de la passerelle par défaut de la victime afin de pouvoir récupérer tous les paquets transmis par la victime.
Attention, si vous attaquez la victime sans configurer votre machine comme « routeur », l'attaque sera appelée DOS (**Denial Of Service**). Les requêtes réalisées par la victime ne sera pas transmis au « vrai » destinataire et seront donc « drop ».

Il est donc judicieux d'activer le mode routeur du Kali en saisissant cette commande :

echo 1 > /proc/sys/net/ipv4/ip_forward

Pour trouver la cible que l'on souhaite attaquer, utiliser la commande suivante :

net.show

Notre victime sera la machine possédant l'adresse IP : **192.168.5.99** :

set arp.spoof.targets 192.168.5.99

NB : Par défaut, la cible est le réseau où se trouve l'attaquant. Dans notre cas, s'il y a pas du modification de arp.spoof.target, l'attaque ARP aurait été propagée sur le réseau 192.168.5.0/24.

Verifier la table ARP de la victime (avant l'attaque) :

L'adresse MAC commençant par « **e4-5d** » est la passerelle du réseau.
L'adresse MAC commençant par « **d8-cb** » est la machine de l'attaquant

Lorsque nous activons le module :

```
arp.spoof on
```

Nous pouvons observer la modification de la table ARP de la victime :

A partir de maintenant, nous avons effectué une attaque MITM (**M**an-**I**n-**T**he-**M**iddle). Nous pouvons « sniffer » le réseau et nous verrons apparaître les paquets transmis par la victime.

Nous pouvons dès à présent combiner cette attaque avec une attaque de type « Spoof DNS » pour vérifier son fonctionnement.

3. Attaque DNS

Un « Spoof DNS » consiste à modifier la table DNS d'un client (ou d'un réseau entier). Lorsque la victime souhaitera contacter le site web http://apache.org (nous allons prendre le site apache.org comme exemple), il sera alors redirigé vers une autre adresse IP (privée ou public) en gardant le même nom de domaine.

Nous commençons tout d'abord par configurer le spoof DNS :

```
set dns.spoof.domains apache.org
set dns.spoof.all true
dns.spoof on
```

NB : la commande « set dns.spoof.all true » permet de prendre en compte les requêtes provenant de l'extérieur (et non local à la machine).

Nous gardons bien évidement l'attaque spoof ARP en place pour cibler l'utilisateur que nous attaquons.

Rappel :

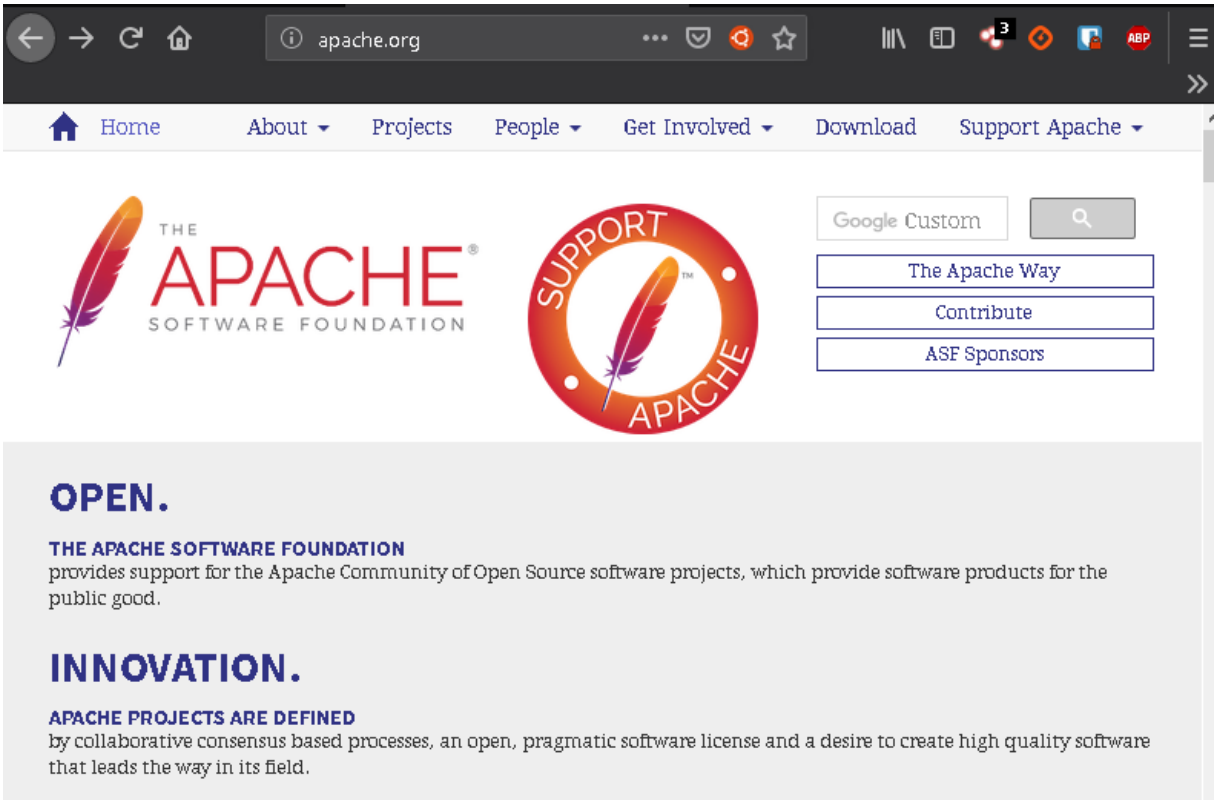
set arp.spoof.targets 192.168.5.99
arp.spoof on

Vu que nous redirigeons les flux en provenance de « apache.org » vers l'ip de l'attaquant, nous allons monter un serveur web avec Bettercap :

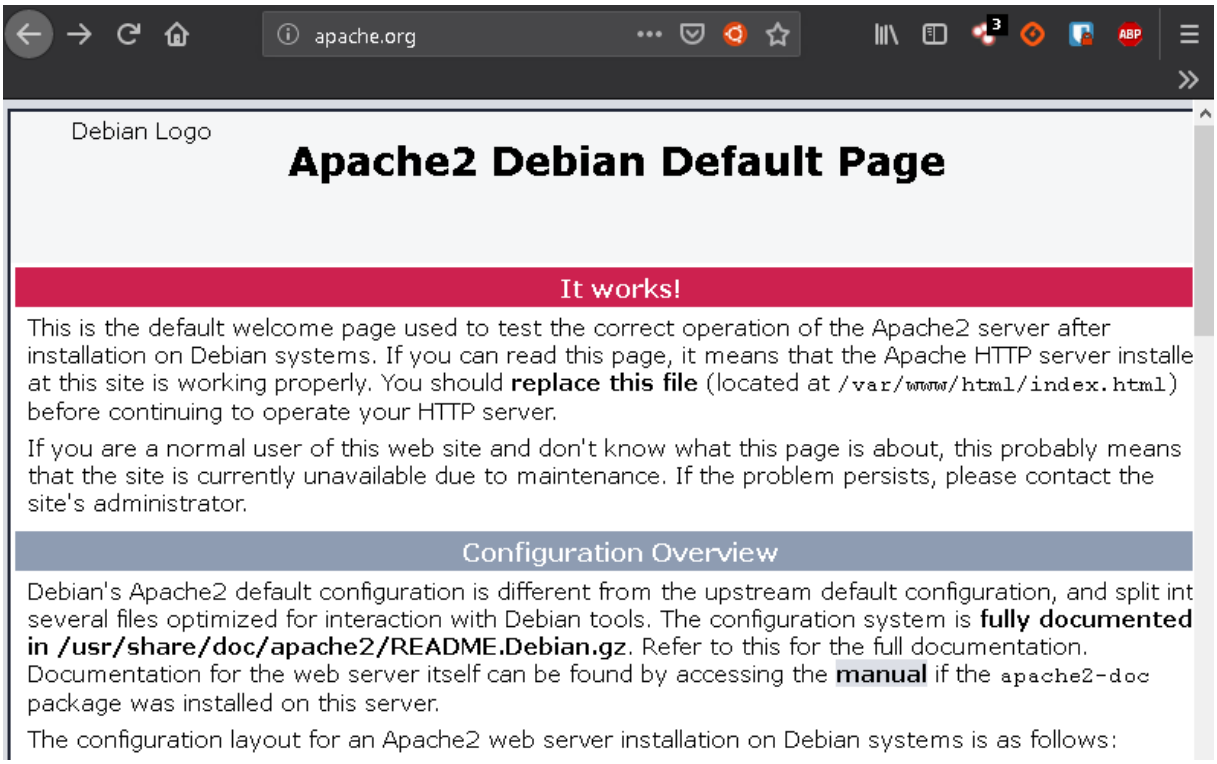
set http.server.path /var/www/html
http.server on

Attention de ne pas avoir un serveur web déjà allumé, sinon il faut modifier le port dans la configuration du serveur dans Bettercap (set http.server.port)

Avant la modification de la table ARP de la victime, lorsqu'il se rend sur apache.org il atteint la page officielle :



Après l'empoisonnement du cache ARP, la page apache.org redirige vers la page web de l'attaquant :



L'attaque DNS est maintenant terminée,

Attaque Proxy

Une attaque Proxy est le fait de pouvoir récupérer les logs de toutes les requêtes web. Il est intéressant de mettre en place un proxy lorsque l'on souhaite bloquer certains sites web dans une entreprise.
Dans notre cas, nous allons utiliser le proxy pour récupérer les pages que consulte la victime.

Tout d'abord, il faut configurer le niveau de verbosité du sniffer (false équivaut à réduit) :

```
set net.sniff.verbose false
net.sniff on
```

Puis on configure le proxy :

```
set http.proxy.sslstrip true
http.proxy on
```

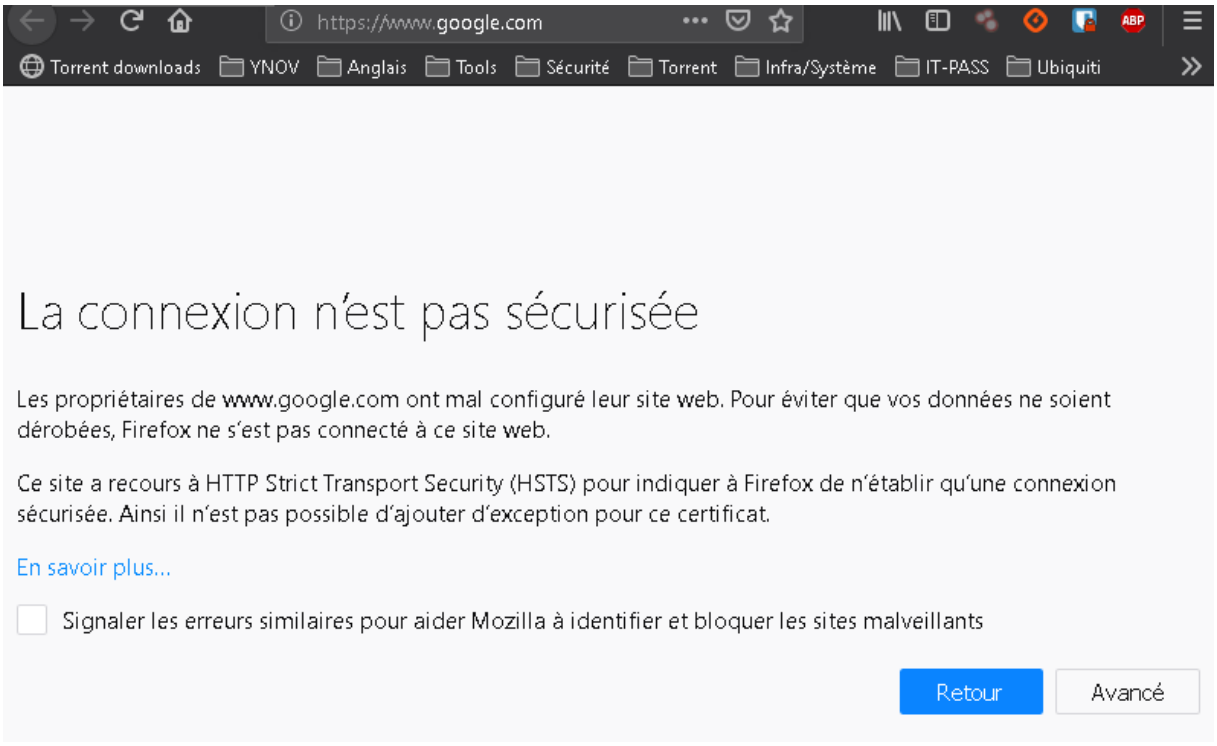
L'utilisation du paramètre « sslstrip » nous permet d'activer le proxy https, et de proposer au client un certificat généré par Bettercap.

Là encore nous gardons en place l'attaque ARP afin de cibler la victime.

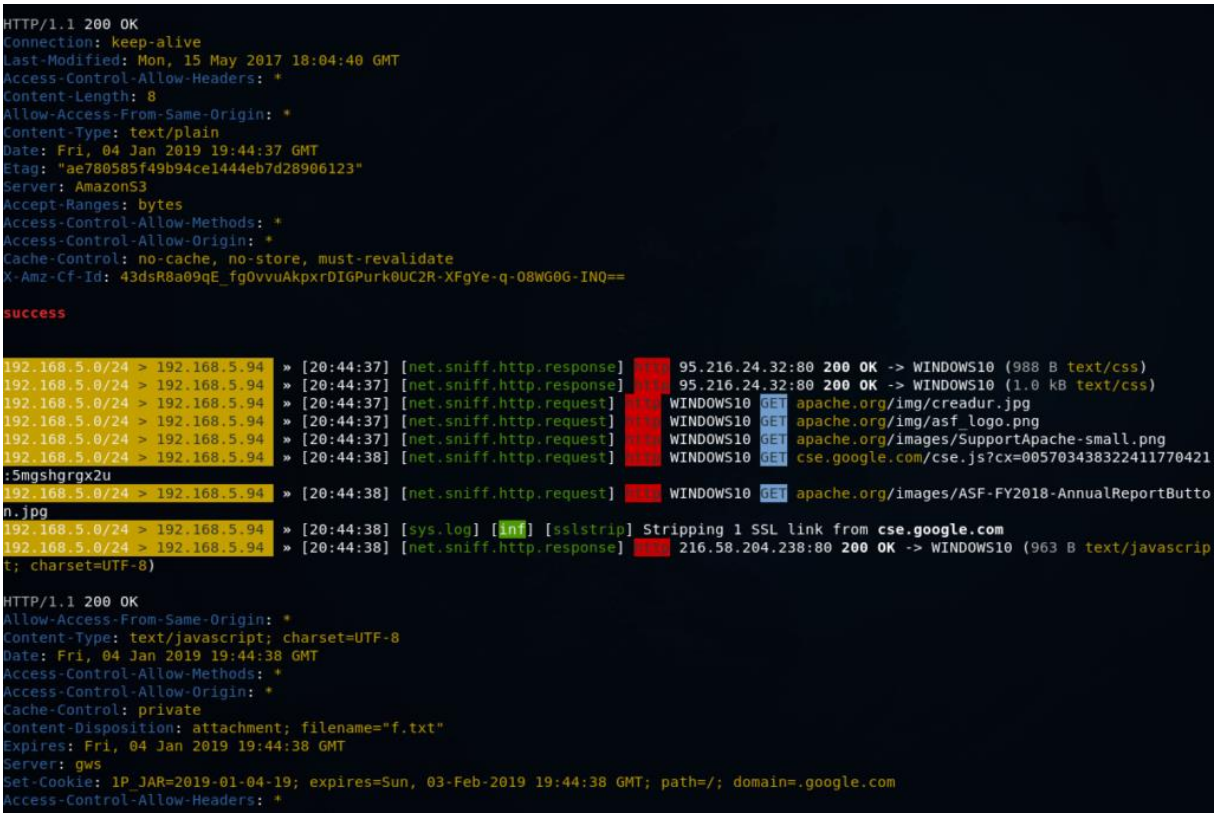
Et lorsque la victime consulte des pages web, nous avons des messages sur bettercap :

```
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
---
```

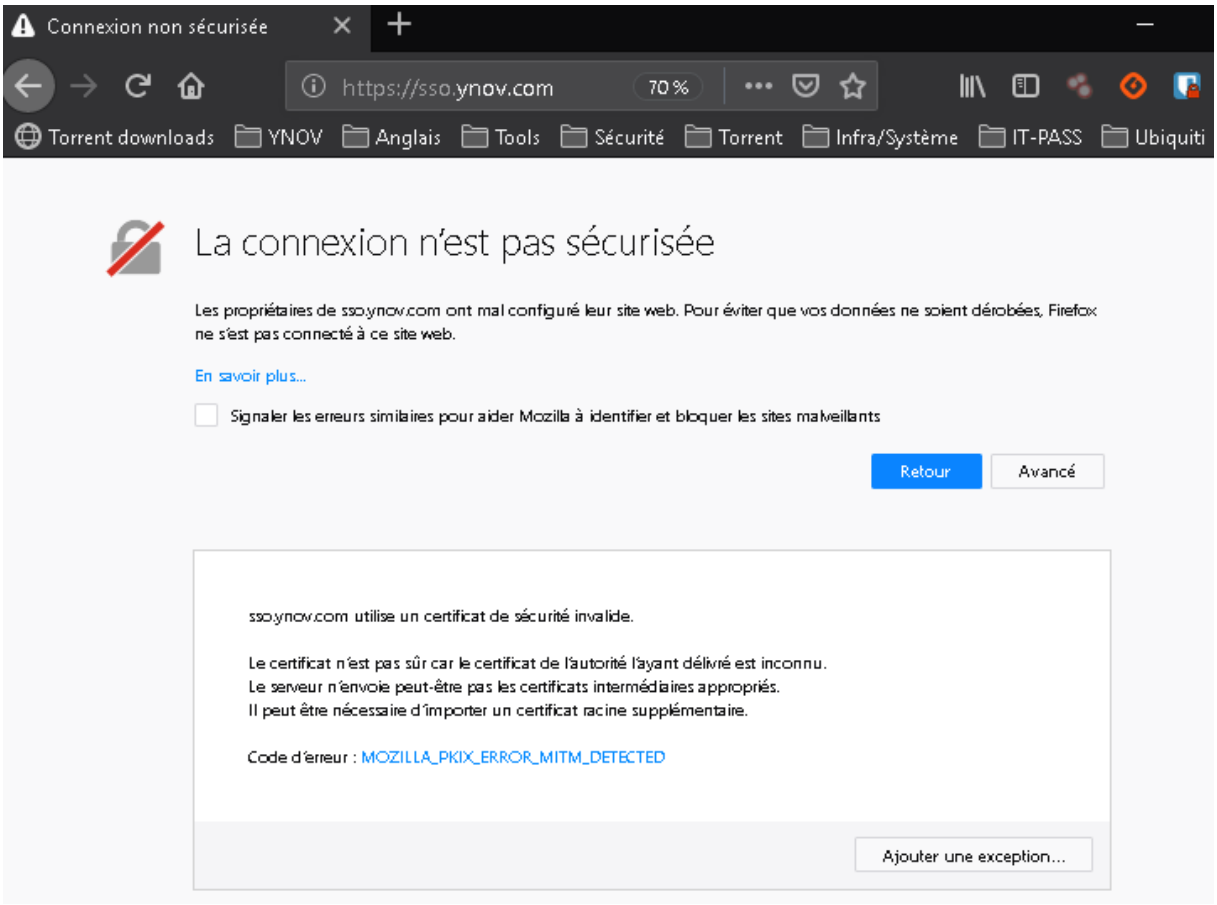
Attention aux sites en HTTPS, certains ont une protection HSTS qui permet de bloquer (ou du moins complexifier l'attaque) :



voir dans Bettercap que la victime à bien été sur google comme ce suit:



Lorsque la victime souhaite accéder à un site en HTTPS (sans HSTS) :



Une erreur de certificat apparaît sur la page du navigateur car c'est un certificat non approuvé par l'autorité de certification. Si la victime continue sans faire attention, nous pourrons alors récupérer potentiellement des informations confidentielles.
En effet, vu que nous avons généré le certificat pour le proxy, nous détenons la clef privée. Tous les flux peuvent donc être déchiffrés sans difficultés.