Activité 2 – Attaque MITM d'un service HTTP et mise en place de contre-mesures

Propriétés	Description
Intitulé long	Ce TP a pour but de simuler une attaque de l'homme du milieu sur un service HTTP afin de démontrer le besoin de chiffrement et l'utilité d'utiliser le protocole HTTPS
Formation(s) concernée(s)	BTS Services Informatiques aux Organisations
Matière(s)	Bloc 3 SISR – Cybersécurité des services informatiques
Présentation	Après avoir remobilisé les savoirs fondamentaux en matière de cryptographie, ce TP permet de mettre en évidence la grande faiblesse du protocole HTTP et la nécessité d'utiliser un protocole permettant le chiffrement.
Compétences	• Protéger les données à caractère personnel. • Garantir la disponibilité, l'intégrité et la confidentialité des services informatiques et des données de l'organisation face à des cyberattaques. • Assurer la cybersécurité d'une infrastructure réseau, d'un système, d'un service. • Assurer la cybersécurité d'une solution applicative.
Savoirs	• Typologie des risques et leurs impacts. • Principe de la sécurité : disponibilité, intégrité et confidentialité. • Chiffrement, authentification et preuve : principes et techniques. • Sécurité des applications Web : risques, menaces et protocoles. • Cybersécurité : bonnes pratiques, normes et standards. • Vulnérabilités du protocole HTTPS et contre-mesures.
Prérequis	Connaissances de base concernant l'administration d'un système GNU/Linux, fondamentaux en matière de cryptographie, de certificat, fondamentaux réseaux (Ethernet, IP, TCP).
Outils	Kali, metasploitable, mutillidae, conteneurs sur Docker (Laboratoire n°2) https://forge.aeif.fr/btssio-labos-kali/lab2
Mots-clés	Kali, cryptographie, chiffrement, certificat, exploitation de vulnérabilités, remédiations, hygiène numérique, respect des bonnes pratiques.
Durée	2 heures
Auteurs	Apollonie Raffalli et Patrice Dignan avec la relecture de Valérie Martinez Laboratoire sur Docker : Apollonie Raffalli avec les tests de Christelle Thiry

Préambule

Le TP proposé est uniquement à visée pédagogique. Son objectif est l'analyse de failles liées à l'usage de certains protocoles réseaux afin de proposer une amélioration de la sécurité informatique d'un système d'information et de l'hygiène numérique des étudiants. Il permet également l'acquisition de compétences associées au bloc 3 Cybersécurité SISR du BTS SIO.

Les outils abordés dans ce support sont uniquement utilisés à des fins éthiques (Ethical Hacking) et pédagogiques. Leur usage est formellement interdit en dehors de ce cadre sur un réseau tiers sans autorisation explicite.

Pour rappel, l'article 323-1 du code pénal stipule que le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 60 000 € d'amende.

Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système, la peine est de trois ans d'emprisonnement et de 100 000 € d'amende.

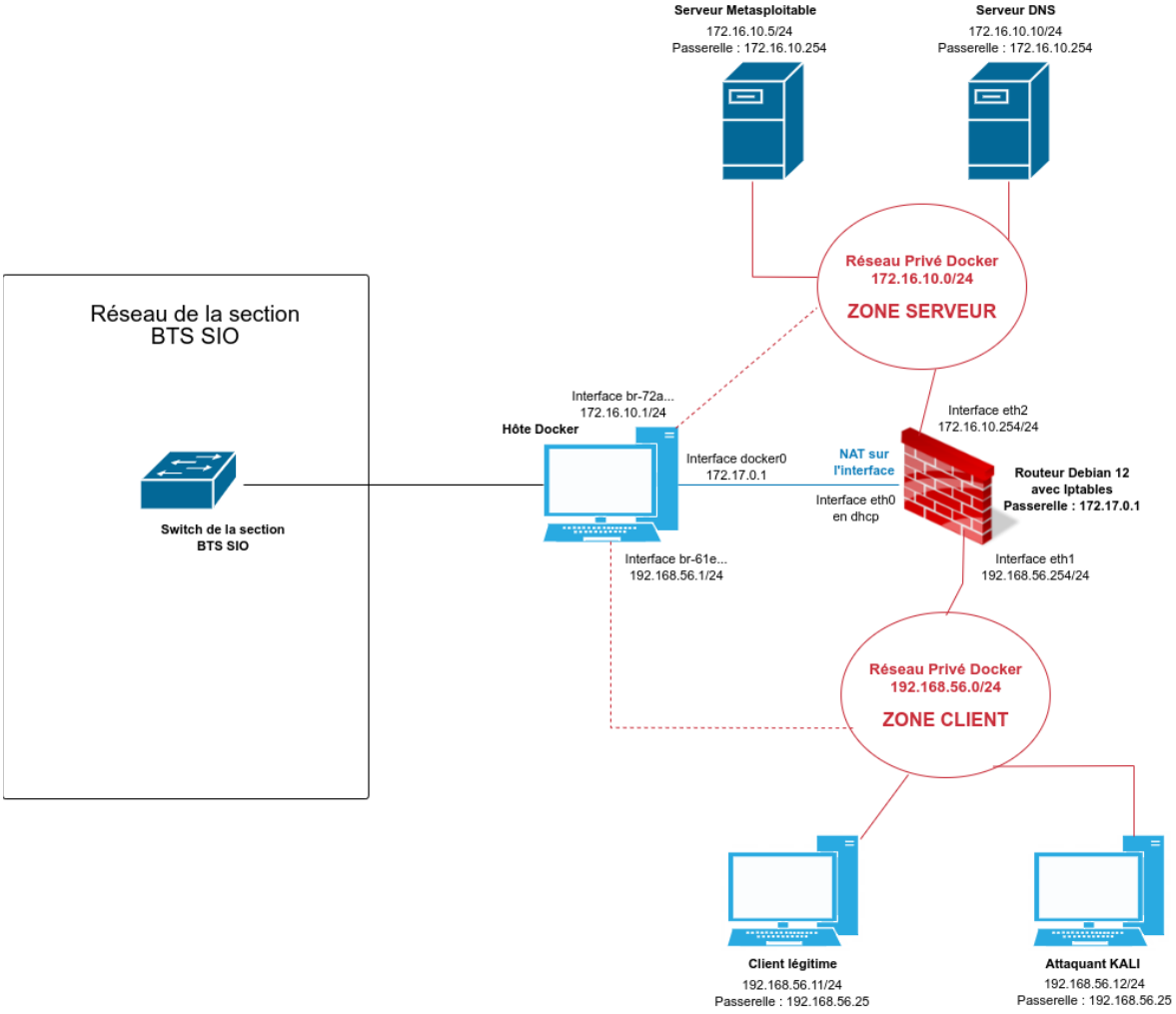
Plateforme utilisée

Le TP qui vous est proposé utilise le laboratoire 2 contenant 5 conteneurs pré-configurées. **Pour rappel :**

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 »)

Toutes les machines sont accessibles en SSH (sur le port 22 à partir de l'hôte qui les héberge) et à partir de l'extérieur.

La machine Kali Linux et le client Debian bénéficient d'une interface graphique accessible via un bureau à distance (protocole RDP sur le port 3389 à partir de l'hôte qui les héberge) et à partir de l'extérieur.



Voici les comptes, les mots de passe et les numéros de ports **accessibles de l'extérieur**, vous permettant d'accéder aux différents conteneurs :

Intitulé de la machine	Nom d'utilisateur	Mot de passe	Ports SSH	Port RDP
Serveur sous Debian 12	etusio	Fghijkl1234*	12222	
Client sous Debian 12	etusio	Fghijkl1234*	22222	23389
Attaquant sous Kali Linux 2023.3	etusio	Fghijkl1234*	32222	33389
Routeur sous Debian 12	etusio	Fghijkl1234*	42222	
Serveur Metasploitable	msfadmin	msfadmin	52222	

Lorsque des commandes nécessitant des privilèges administrateurs seront utilisées, il sera nécessaire d'utiliser la commande **sudo**.

```
etusio@srvssh:~$ sudo service ssh restart
```

Voici une représentation logique de la maquette proposée dans le cadre de ce laboratoire :

- Lancer le laboratoire : **bash gestion_lab2.sh -c**
- Vérifier que les 5 conteneurs sont actifs : **docker ps**

```
CONTAINER ID   IMAGE                      COMMAND                  CREATED      STATUS           PORTS
                                                      NAMES
dd0866a4c36d   tleemcjr/metasploitable2   "sh -c '/bin/service…"   6 days ago   Up About a minute
                                                      metasploitable-lab2
4136eb106769   aporaf/kalirolling:lab2    "/lib/systemd/system…"   6 days ago   Up About a minute   22/tcp, 3389/tcp
                                                      kali-lab2
e879e10e57a3   aporaf/clientdebian12:lab2    "/lib/systemd/system…"   6 days ago   Up About a minute   22/tcp, 3389/tcp
                                                      client-lab2
7c1ef0ce2e65   aporaf/serveurdebian12:lab2   "/lib/systemd/system…"   6 days ago   Up About a minute   22/tcp, 53/tcp, 53/udp
                                                      serveur-lab2
13a1ffaae620   aporaf/routeurdebian12:lab2   "/lib/systemd/system…"   6 days ago   Up About a minute   0.0.0.0:12222->12222/tcp, :::12222->12222/tcp, 0.0.0.0:22222->22222/tcp, :::22222->22222/tcp
, 0.0.0.0:23389->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
```

Attaque MITM d'un service HTTP

Découverte des hôtes et services présents sur un réseau local

- En tant qu'attaquant, la première étape consiste à recueillir des informations sur le réseau dans lequel nous nous trouvons. Ainsi, à l'aide de l'outil nmap présent sur Kali Linux, nous allons réaliser un scan des réseaux logiques locaux.

```
etusio@kali:~$ nmap -sP 192.168.56.0/24
etusio@kali:~$ nmap -sP 172.16.10.0/24
```

- Puis scanner les différents hôtes afin de savoir quels ports sont ouverts sur ceux-ci et quels services sont proposés. Par exemple, sur la zone des serveurs :

.

```
etusio@kali:~$ nmap -sV 172.16.10.5
etusio@kali:~$ nmap -sV 172.16.10.10
etusio@kali:~$ nmap -sV 172.16.10.254
```

Voici un extrait du résultat obtenu à l'aide de cette commande sur le serveur metasploitable d'adresse IP 172.16.10.5 :

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-08 14:32 CEST
Nmap scan report for 172.16.10.5
Host is up (0.00037s 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)
...
```

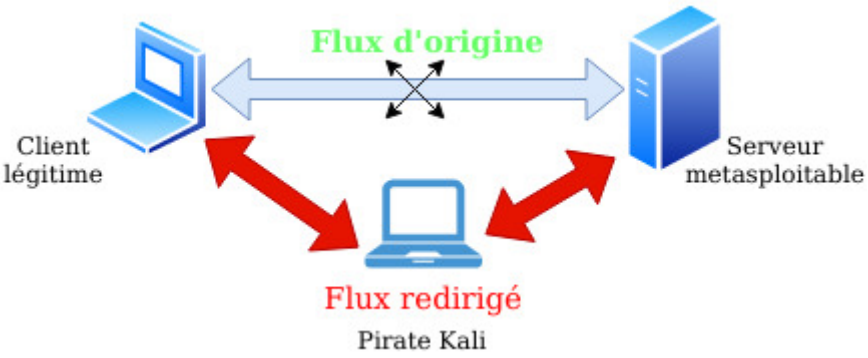
Nous constatons que de nombreux ports sont ouverts, ce qui est logique puisqu'il s'agit d'un serveur intentionnellement vulnérable.

Si la commande « nmap » est trop longue, enclenchez plutôt les étapes suivantes :

- nmap 172.16.10.5 pour récupérer les ports
- nmap -sV 172.16.10.5 -p 80 pour récupérer les informations sur le service HTTP

Nous ciblons dans ce TP plus particulièrement le serveur HTTP : le serveur metasploitable accueille un site Web Mutillidae conçu pour identifier et tester les failles de sécurité.

Le principe de l'attaque est le suivant : le pirate Kali empoisonne le cache ARP du client légitime et récupère le mot de passe de son compte Mutillidae via une connexion non sécurisée HTTP.



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

Une personne malveillante s'est donc introduite sur le réseau dans le but de récupérer entre autres des informations confidentielles dont des noms d'utilisateurs et mots de passe disposant de privilèges.

Après avoir analysé l'architecture réseau et découvert l'existence d'un serveur Web, elle décide de réaliser une attaque Man in the Middle afin d'obtenir un accès sur ce dernier.

Pour rappel, l'empoisonnement du cache ARP permet de falsifier le cache ARP de la victime en associant, par exemple, l'adresse IP de la passerelle à l'adresse MAC du pirate. Ainsi, tout le flux passe par la machine du pirate qui peut se mettre en écoute avec un logiciel de capture de trames. Mais ce n'est souvent pas suffisant, car il est nécessaire que le flux en retour passe également par la machine du pirate.

Ici, nous voulons récupérer tout le trafic entre le client (192.168.56.11) et le serveur (172.16.10.5). Mais les adresses MAC qui doivent être falsifiées sont celles du client légitime et du routeur (et non du serveur qui est sur un réseau logique distant) car, pour atteindre le serveur, le client passe obligatoirement par la passerelle.

Les messages ARP piratés n'iront pas au-delà des limites du réseau logique local, l'attaque doit donc être lancée à partir d'un appareil connecté au réseau logique local. Il n'est pas impossible pour quelqu'un d'externe de lancer une attaque sur l'ARP, mais il devra d'abord compromettre à distance un système local via d'autres moyens. Une personne interne, en revanche, aura seulement besoin d'un accès au réseau et de quelques outils facilement disponibles.

Sur Kali, il est possible d'utiliser les outils Ettercap ou arpspoof pour réaliser l'empoisonnement du cache ARP. Nous avons utilisé Ettercap dans le TP MITM SSH, nous utiliserons arpspoof dans celui-ci.

Préalables

Configuration de la machine Kali du pirate qui doit jouer le rôle de routeur :

↗️ Activez le routage sur cette machine. Pour cela, ouvrir le fichier /etc/sysctl.conf, enlever le commentaire devant la ligne suivante et sauvegarder le fichier :

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

↗️ Il faut ensuite exécuter la commande suivante pour recharger les paramètres système :
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
	192.168.56.254

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
	192.168.56.11

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

ADRESSE MAC	ADRESSE IP
	192.168.56.12

Modification de la configuration du site mutillidae sur le serveur Metasploit

- Se connecter sur le serveur Metasploitable : **ssh -o HostKeyAlgorithms=ssh-rsa msfadmin@172.16.10.5**
- Ouvrir le fichier config.inc dans /var/www/mutillidae (sudo nano /var/www/mutillidae/config.inc) et modifiez le contenu de la variable dbname par la valeur suivante : $dbname = 'owasp10'.
  Pour pouvoir utiliser nano saisir « export TERM=xterm ». Vous pouvez également utiliser vim.tiny.
- Recharger apache2 : **sudo /etc/init.d/apache2 reload**. Il s'agit d'une très vielle distribution, le démarrage des services étaient opérés par le **démon init system V**.

- Créer un compte sur le site: http://172.16.10.5/mutillidae/ ou http://srvm.local.sio.fr/mutillidae.

Empoisonnement du cache ARP via arpspoof

L'étape suivante consiste à réaliser l'empoisonnement ARP. *Pour rappel, on va envoyer des paquets de façon constante car si une communication est effectuée entre le serveur et le client, ce dernier va à nouveau mettre à jour sa table ARP et cette fois-ci avec les informations correctes. Il faut donc remettre continuellement à jour la table ARP du client et de la passerelle pour qu'elles restent falsifiées.*

Pour cela, suivre les étapes suivantes depuis la machine Kali :

- Ouvrez un premier terminal puis saisissez la commande suivante :

sudo arpspoof -t @ip-client-victime @ip-passerelle

On peut traduire cette ligne de commande par « fait moi passer pour l'adresse IP de la passerelle auprès du client ». Cette commande redirige le trafic entre l'IP du client victime et la passerelle vers la machine pirate (qui a lancé la commande).

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

ADRESSE MAC	ADRESSE IP

Ouvrez un second terminal puis saisissez la commande suivante :
sudo arpspoof -t @ip-passerelle @ip-client-victime

On peut traduire cette ligne de commande de la part du pirate par « fait moi passer pour l'adresse IP du client auprès de la passerelle ». Cette commande redirige le trafic entre la passerelle et l'IP du client victime vers la machine pirate (qui a lancé la commande).

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

ADRESSE MAC	ADRESSE IP

Note

Pour ne pas ouvrir deux terminaux et pour récupérer ensuite la main sur chaque terminal, il est possible d'envoyer le résultat de la commande vers « /dev/null » (après avoir ouvert une « console administrateur » avec la commande « sudo su ».
sudo arpspoof -t @ip-client-victime @ip-passerelle > /dev/null 2>&1 &

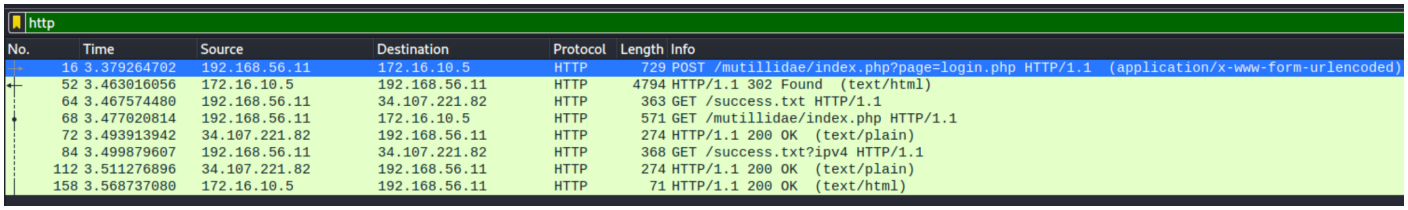
Ne pas oublier d'arrêter les processus ensuite via la commande « Kill num_processus ».

Capture et inspection de trames
Le pirate lance maintenant une capture de trame et attend que le client légitime s'authentifie sur l'application Mutillidae de la machine Metasploitable en HTTP pour capturer le mot de passe saisi puisque ce dernier circule en clair.

Depuis la machine kali, ouvrez le logiciel Wireshark puis configurez une écoute sur le protocole HTTP.
Si besoin, sudo dpkg-reconfigure wireshark-common puis répondre « Yes » à Should non-superusers be able to capture packets? [yes/no].

Depuis la machine cliente victime, connectez-vous au site Mutillidae avec le compte créé précédemment.

Des trames HTTP doivent apparaître sur wireshark :



Pour trouver facilement les identifiants, positionnez-vous sur la première requête de connexion au site mutillidae, puis cliquez sur « Analyser / suivre / HTTP stream » :

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

Note

Les organisations doivent nécessairement prendre en compte la confidentialité des échanges sur un site comportant une page d'authentification.

L'article 32 du RGPD porte sur l'obligation pour une organisation de mettre en œuvre des solutions sécurisées : « Compte tenu de l'état des connaissances, des coûts de mise en œuvre et de la nature, de la portée, du contexte et des finalités du traitement ainsi que des risques, dont le degré de probabilité et de gravité varie, pour les droits et libertés des personnes physiques, le responsable du traitement et le sous-traitant mettent en œuvre les mesures techniques et organisationnelles appropriées afin de garantir un niveau de sécurité adapté au risque, y compris entre autres, selon les besoins »

Plus loin, cet article fait référence au chiffrement.

Mise en place de Contre-mesures

Le chiffrement HTTPS

Il est considéré ici que les notions de chiffrement et notamment de chiffrement asymétrique sont acquises ainsi que les notions relatives aux certificats.

L'objectif est ici de faire évoluer le site Mutillidae en HTTPS afin d'instaurer une connexion chiffrée et sécurisée entre le serveur et les terminaux. Pour cela, il est nécessaire de configurer un virtualhost sur l'application Mutillidae en suivant les étapes suivantes depuis la machine Metasploitable.

Nous utiliserons le certificat par défaut fourni par Apache. En production, il sera nécessaire d'obtenir un certificat signé par une autorité de certification "publique".

- Activez le module ssl : sudo a2enmod ssl

- Créez dans le répertoire /etc/apache2/sites-available le fichier default-ssl en y mettant le contenu suivant :

```
<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>
```

- Activez le VirtualHost : sudo a2ensite default-ssl et sudo /etc/init.d/apache2 force-reload

- Ouvrez le fichier htaccess situé à la racine de l'application de Mutillidae (/var/www/mutillidae/.htacess) et mettez en commentaire les trois lignes commençant par php_flag (la commande php_flag utilisée pour activer ou désactiver une option n'est n'est pas reconnue dans ce contexte) en ajoutant le caractère # devant :

```
## The following section disables PHP magic quoting feature.
## Turning these on will cause issues with Mutillidae.
## Note: Turning these on should NEVER be relied on as a method for securing ag$
## As of PHP 6 these options will be removed for exactley that reason.

## Donated by Kenny Kurtz
#php_flag magic_quotes_gpc off
#php_flag magic_quotes_sybase off
#php_flag magic_quotes_runtime off
```

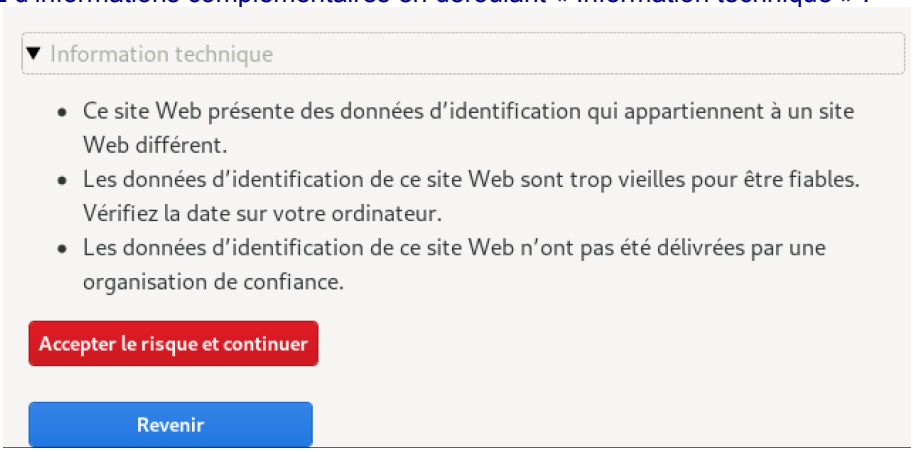
- Redémarrez apache 2 : sudo /etc/init.d/apache2 restart

Depuis la machine client légitime

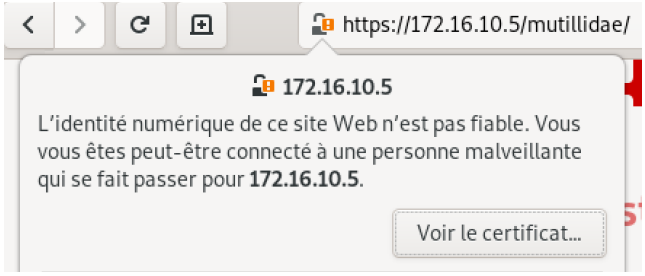
- Connectez-vous au site Mutillidae en utilisant le protocole HTTPS. *N'utilisez pas Firefox qui est trop sécurisé par défaut pour accéder à un site configuré avec une version de protocole SSL/TLS trop ancienne mais le navigateur « epiphany » :* ***Applications/Internet/Web***

Un message d'alerte s'affiche :

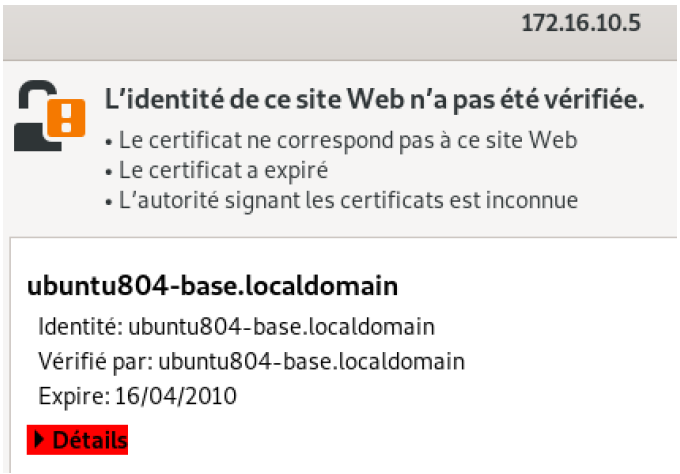
Vous disposez d'informations complémentaires en déroulant « Information technique » :



Le cadenas précédent l'URL montre effectivement un point d'interrogation :



Vous pouvez visualisez les détails du certificat en cliquant sur « voir le certificat » :



Selon la CNIL (https://www.cnil.fr/fr/securite-chiffrer-garantir-lintegrite-ou-signer), lors de la réception d'un certificat électronique, Il est essentiel de vérifier que le certificat contient une indication d'usage conforme à ce qui est attendu, qu'il est valide et non révoqué, et qu'il possède une chaîne de certification correcte à tous les niveaux ».

Q6. Quels sont les rôles du certificat côté serveur ?

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

↗️ Acceptez le risque et continuez quand même (chose à ne pas faire, bien évidemment sur un vrai site en ligne)

↗️ Reproduisez l'attaque.

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) *:*

- filtrez sur tcp.port == 443 ;
- repérez la trame « hello » du protocole

```
34 12.224299152  172.16.10.5          192.168.56.11         TLSv1     1011 Server Hello, Certificate, Server Hello Done
```

- Activez le menu « Analyse / Suivre / Flux TCP

Mesures pour détecter l'empoisonnement du cache ARP

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

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

L'empoisonnement ARP utilisé ici n'est possible que grâce à l'émission de messages non sollicités à tout le réseau, qu'on appelle « gratuitous ARP » qui a, tout de même, l'inconvénient de générer beaucoup d'activité sur le réseau.

Des outils permettent de contrôler les modifications du cache ARP afin de vérifier les modifications suspectes. On peut citer l'exemple de l'outil arpwatch qui surveille le réseau en continu et peuvent alerter un admin si des signes d'empoisonnement du cache ARP sont détectés. Toutefois, il est nécessaire de faire attention aux faux positifs qui peuvent créer beaucoup d'alertes indésirables.

Ces outils de surveillance seront configurés dans une prochaine activité.

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