Année 2022-2023

SAÉ Cyber 4.0 Sécurisation d'un SI

Cahier de SAÉ

Version 1.0

Appartient à :	En équipe avec :
Nom : HIRSCH Prénom : Matéo Groupe : Groupe 0	Nom : ECOTIERE Prénom : Léo
Nom de votre équipe : Los 2 leones de oro	

Informations générales

Répartition en groupes

5 équipes de 4 étudiants et une équipe de deux avec Léo et Matéo

Emploi du temps

Semaine 1 : 9h-12h 13h-17h sauf jeudi 9h-12h
Semaine 2 : 9h-12h 13h-17h sauf jeudi 9h-12h
Semaine 3 : 9h-12h 13h-17h jeudi libre et vendredi soutenance

Evaluation

- Au fil de la progression, après validation de chaque tâche
- Remplissage de votre cahier de SAÉ qui sera rendu et noté
- Soutenance d'une heure par équipe soit 15 min par étudiant.

Matériel par équipe

- 2 Firewalls Stormshield
- 1 Switch
- 1 Borne WiFi
- 5 PC tour
- 2 Portables

Documentation

- https://documentation.stormshield.eu/

Tâches à réaliser

Bilan

A la fin de votre SAÉ, vous devrez répartir 80h de travail x 4 personnes soit 320 heures-homme (soit 80h x 2 personnes qui font 160h) dans ce tableau et indiquer votre évaluation de l'accomplissement de chaque tâche en pourcentage de réalisation.

Tâches	Heures-homme	Pourcentage de réalisation
Mise en place d'une infrastructure sécurisée	2x1h=2h	100%
Installation et configuration d'un firewall Stormshield	2x2h=4h	100%
Installation et configuration d'un serveur HTTP/HTTPS et d'un serveur FTP/FTPS	2x6h=12h	100%
Authentification transparente par certificat SSL	2x6h=12h	100%
Mettre en place un IDS	2x9h=19h	100%
Attaque sur le Wifi	2x3h=6h	100%
Utilisation de scanners de vulnérabilité	2x10h=20h	100%
Attaque Man in The Middle	2x0.5h=1h	100% (0% Bonus)
Contre-mesures pour le MiM	2x4h=8h	100%
Supervision du réseau	2x2.5h=5h	100%
Mise en place d'une architecture Single Sign-On	2x6h=12h	100%
Mise en place d'un VPN SSL pour clients distants	2x2h=4h	100%
Mise en place d'un VPN IPSEC site à site	2x3.5h=7h	100%
TOTAL	112h / 160h	100% / 100%

Ce projet comptabilise 160 heures-hommes pour un binôme. Or, nous l'avons réalisé dans sa totalité en 112 heures-hommes soit une différence de 48 heures-hommes → 24 heures par personne (projet finit un jour en avance).

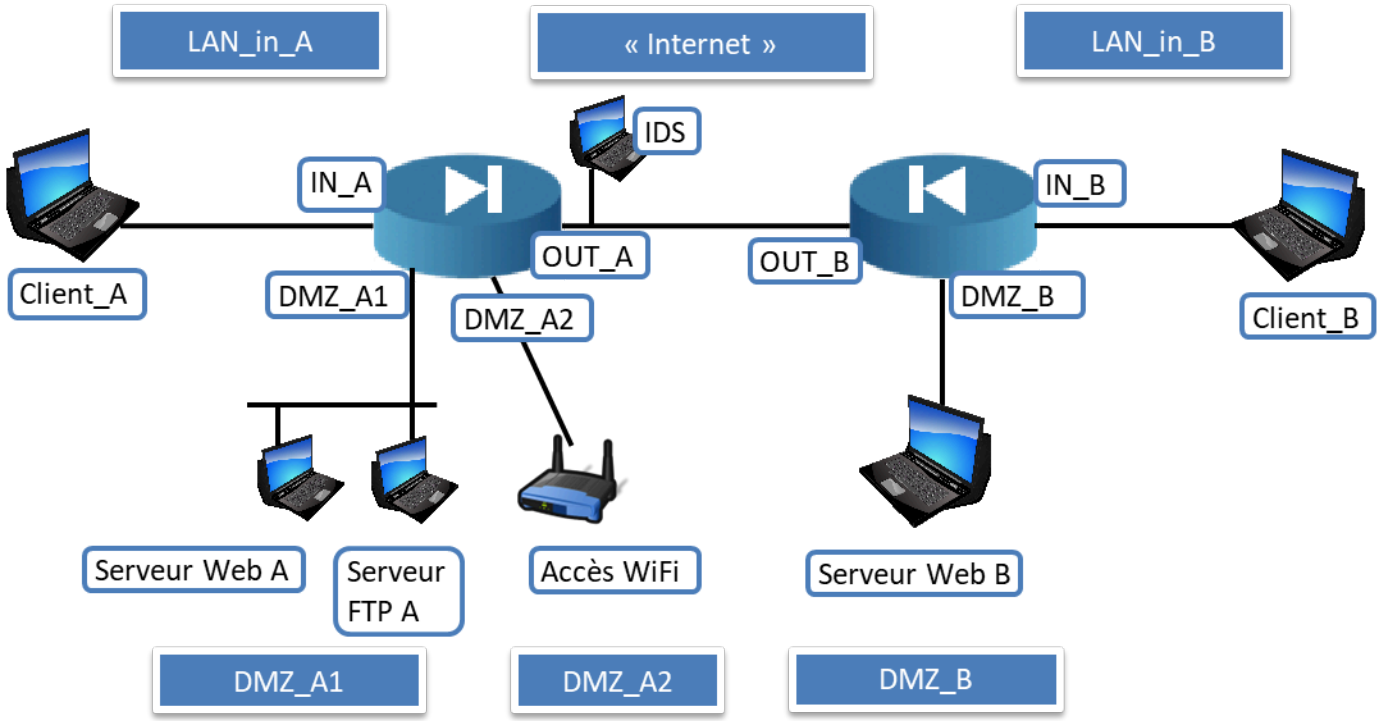
Détails des tâches à réaliser

Tâche 1. Mise en place d'une infrastructure sécurisée

Liste des personnes impliquées avec pourcentage de répartition	1h/pers -> 2h heures-hommes
HIRSCH Matéo 50% ECOTIERE Léo 50%	

Estimation du temps passé sur cette tâche en heure-homme : 2h

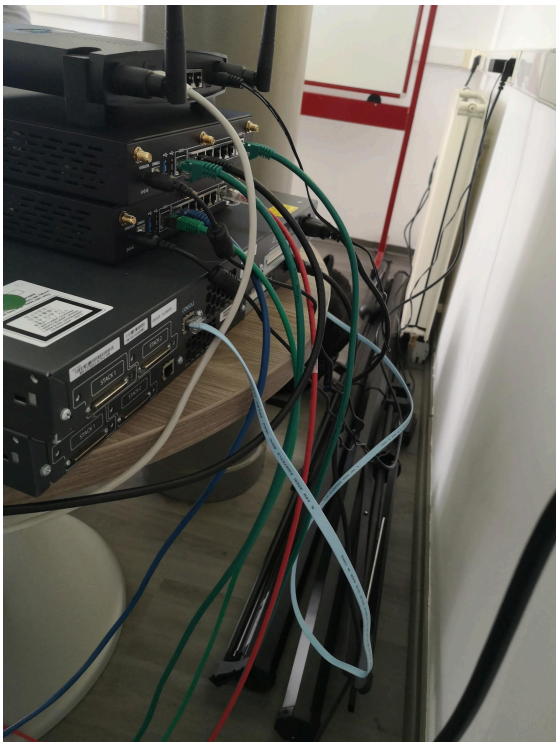
Objectif : Mettre en place l'infrastructure réseau suivante :



Rapport

(Expliquez votre démarche, dessinez un plan IP, insérez des photos de votre architecture avec identification de chaque machine, photo des écrans de configuration IP, etc.)

Voici quelques photos de notre câblage :

Nous avons fait le choix de travailler que sur des machines portables afin de pouvoir s'installer dans une salle libre et être au calme pour travailler.

Voici le plan IP que nous avons établi pour l'ensemble des réseaux :

Réseau LAN A	192.168.1.0/24
Client A	192.168.1.1
Firewall IN A	192.168.1.254

Réseau DMZ A	10.0.0.0/24
Serveur FTP/WEB A	10.0.0.1
Borne WIFI A	10.0.0.100
Firewall DMZ A	10.0.0.254

Réseau WAN	87.10.10.0/24
IDS	87.10.10.100
Firewall OUT A	87.10.10.1

Firewall OUT B	87.10.10.2
Serveur DMZ A Virtuel	87.10.10.11
Client A Virtuel	87.10.10.10
Serveur DMZ B Virtuel	87.10.10.21
Client B Virtuel	87.10.10.20
Virtual Machine Metasploitable Virtuel	87.10.10.50

Réseau LAN B	192.168.2.0/24
Client B	192.168.2.1
Firewall IN B	192.168.2.254

Réseau DMZ B	10.0.0.0/24
Serveur WEB B	10.0.0.1
Firewall DMZ B	10.0.0.254
Virtual Machine Metasploitable	10.0.0.10
Virtual Machine Kali	10.0.0.11
Virtual Machine Windows Server 2019	10.0.0.100
Virtual Machine Windows 10 Professional	10.0.0.200 (puis .201)

Tâche 2.Configuration des firewalls Stormshields

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2h/pers -> 4h heures-hommes

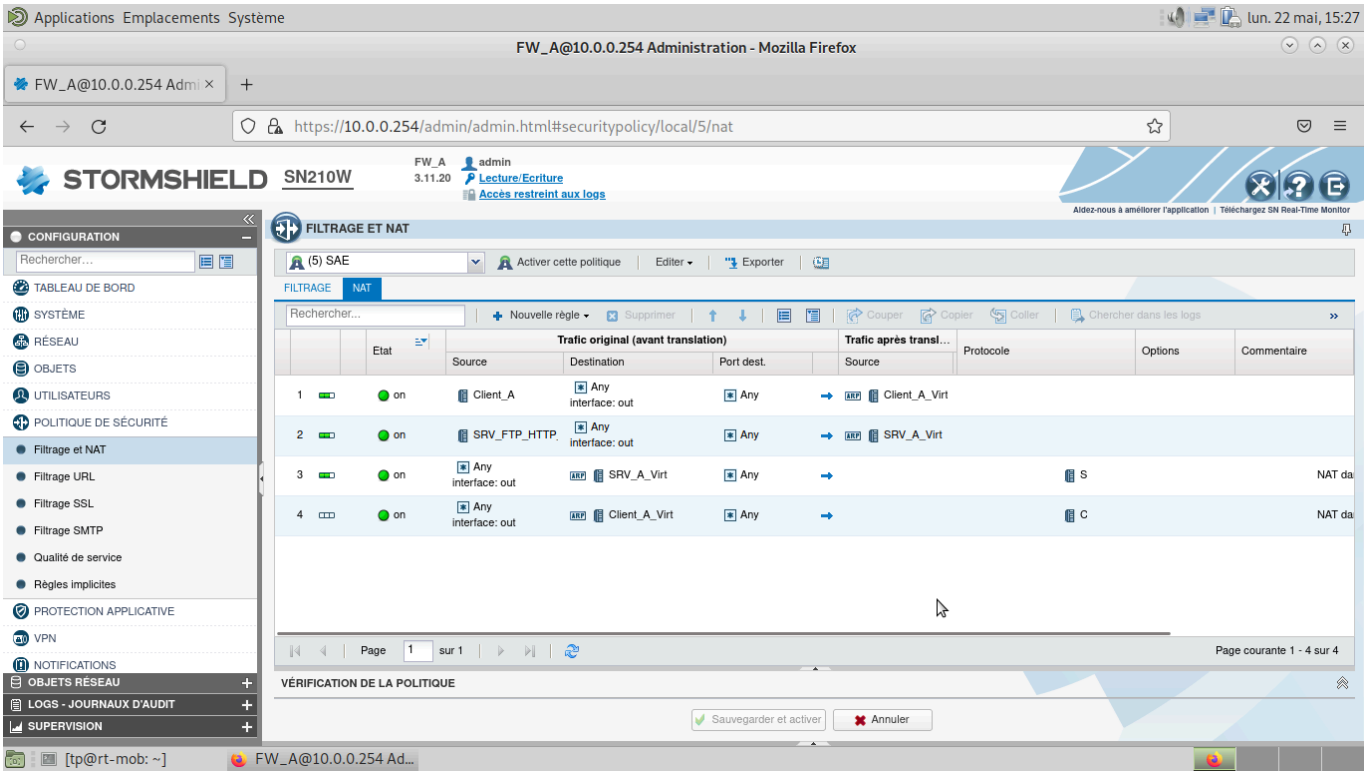
Estimation du temps passé sur cette tâche en heure-homme : 2 heures

Objectif : Configuration des firewalls pour protéger les réseaux internes et DMZ

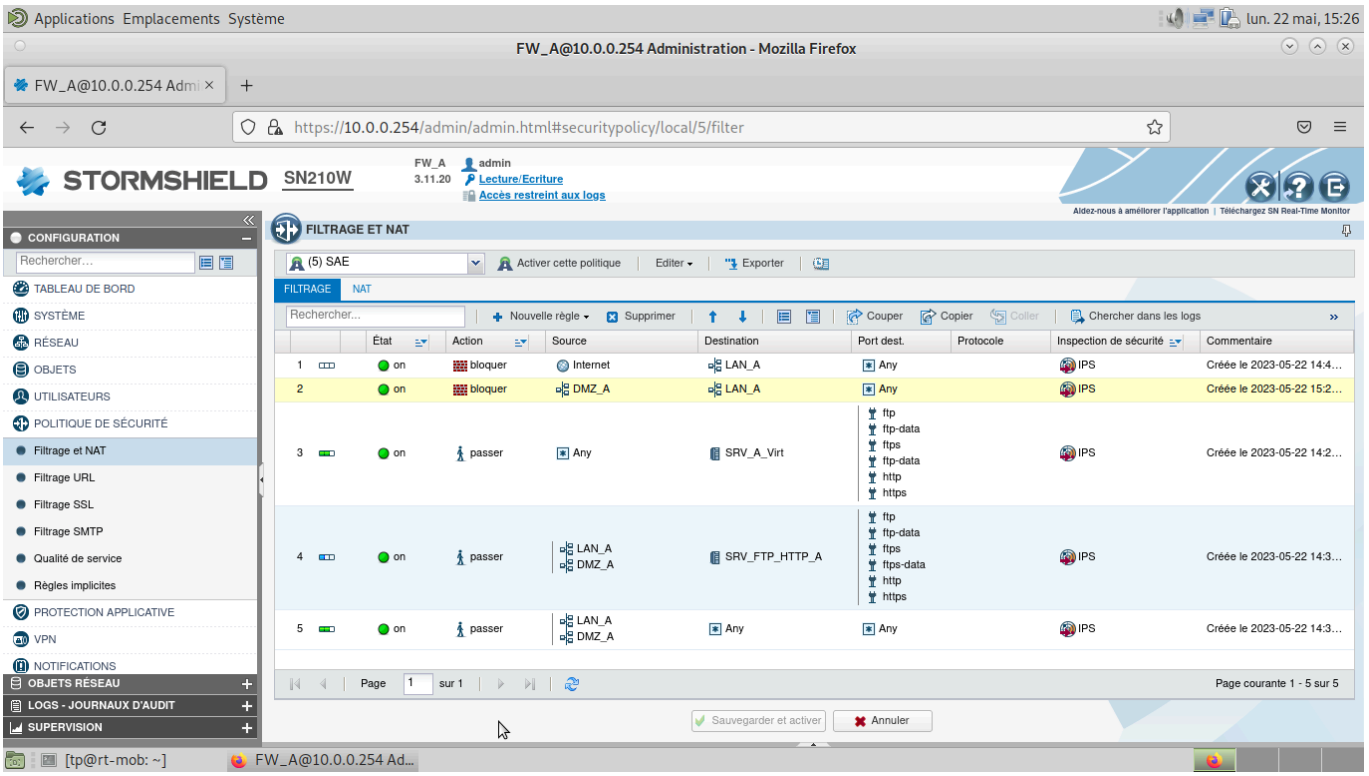
Sous-tâches	Evaluation prof
Mettre en place une politique de NAT -> OK Permettre l'accès aux serveurs uniquement sur les ports concernés -> OK Interdire l'établissement d'une connexion sur les réseaux internes depuis les réseaux externes et les DMZ -> OK *Autorisez l'accès à DMZ_A1 depuis DMZ_A2 -> NE PAS FAIRE*	100%
Testez l'accès aux serveurs	100%

Rapport

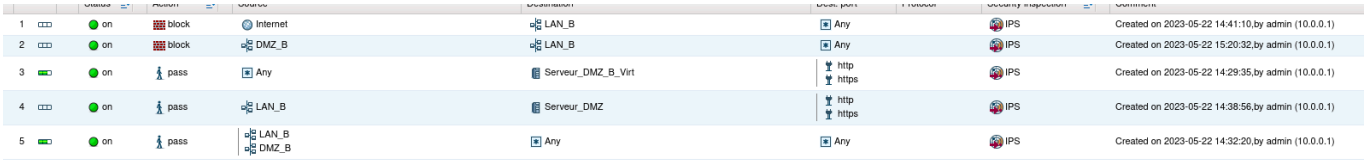
(Expliquez votre démarche, insérez les captures d'écran des menus NAT et Filtrage, de vos tests, etc.)



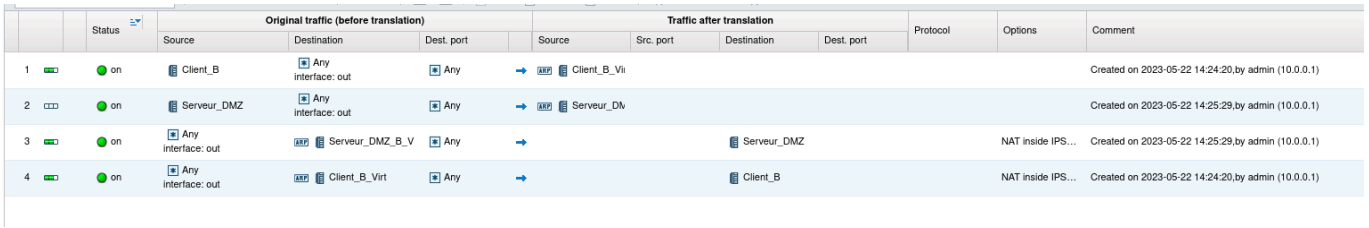
Règles de NAT sur FW A



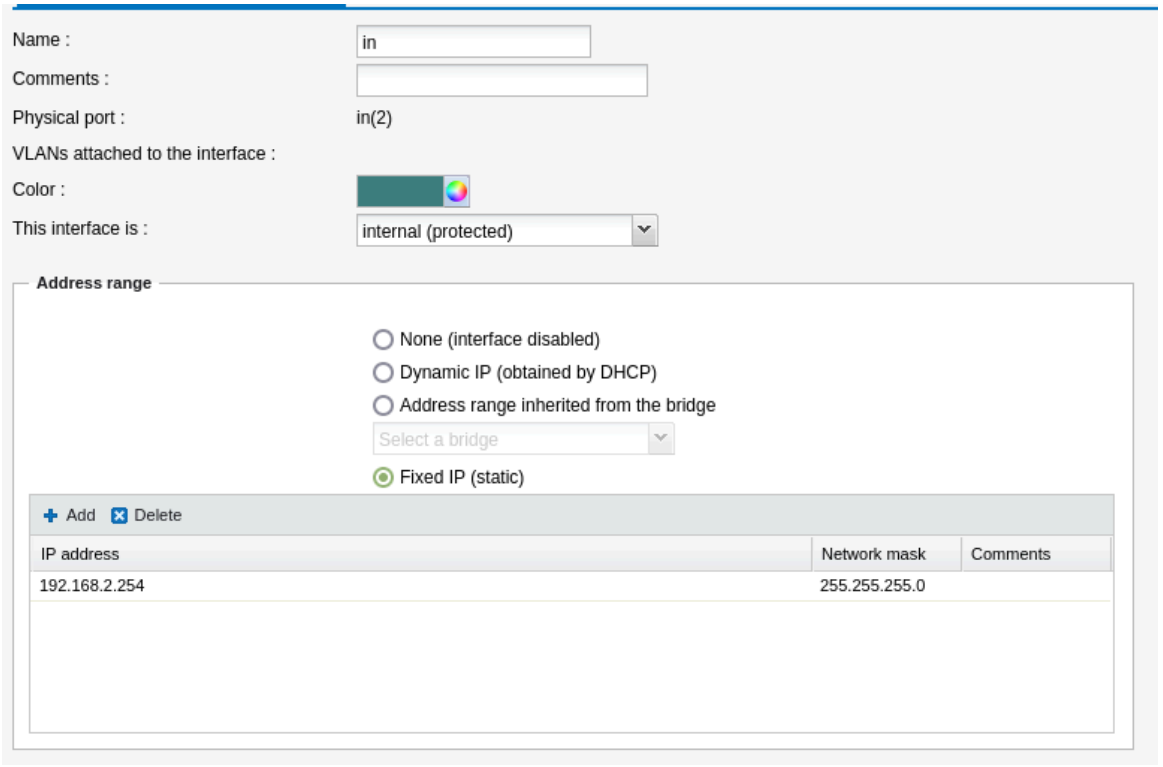
Règles de filtrage FW A



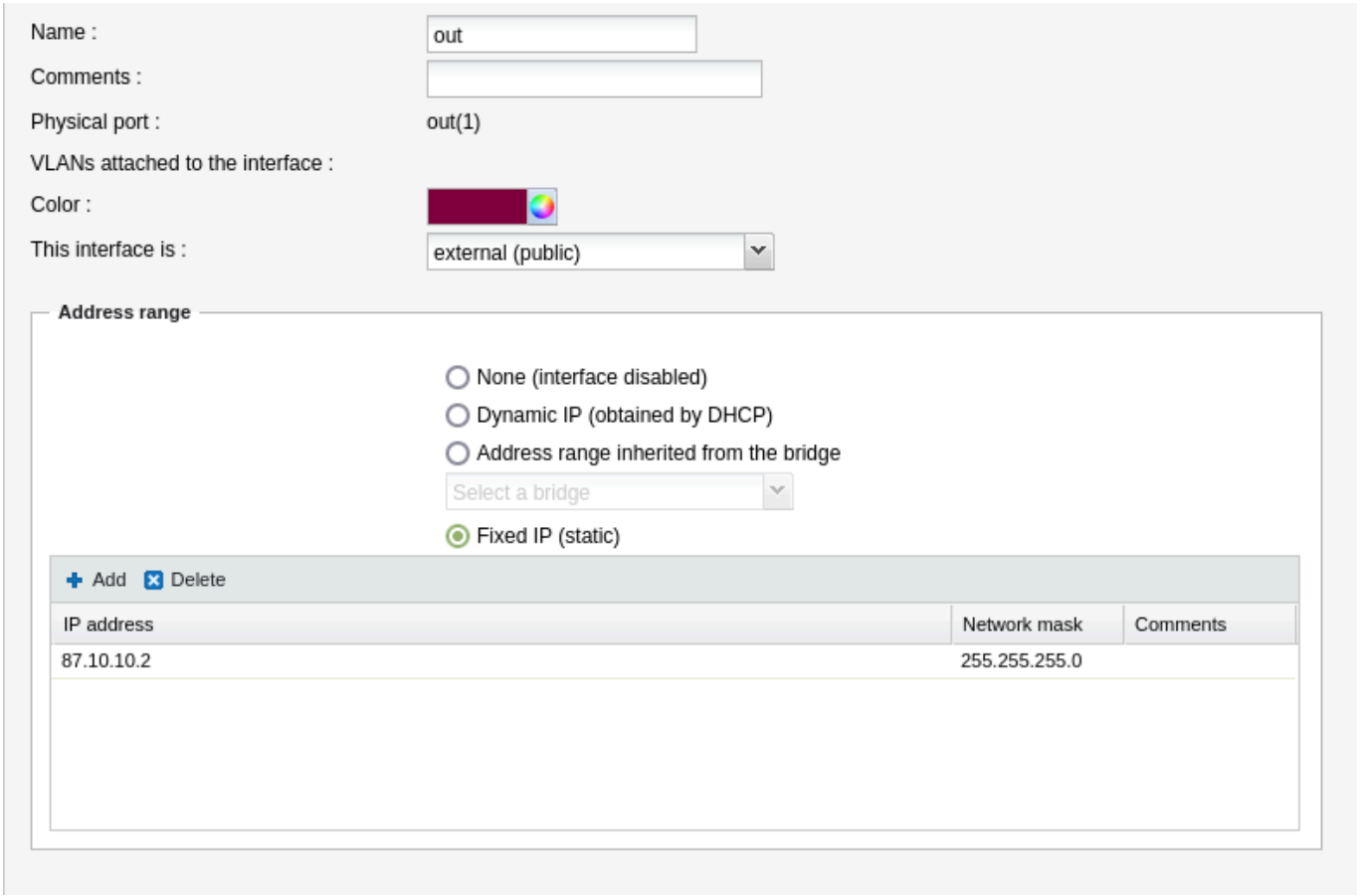
Règles de filtrage FW B



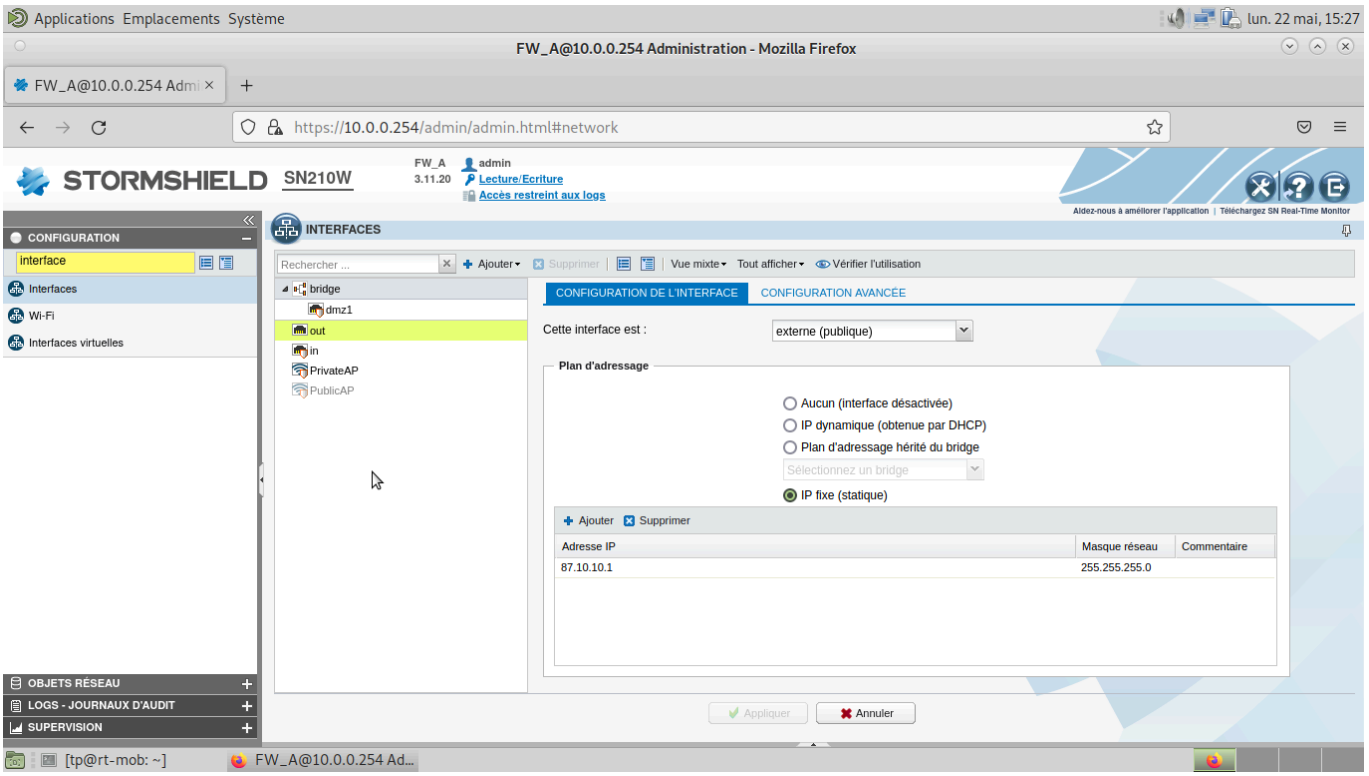
Règles de NAT FW B



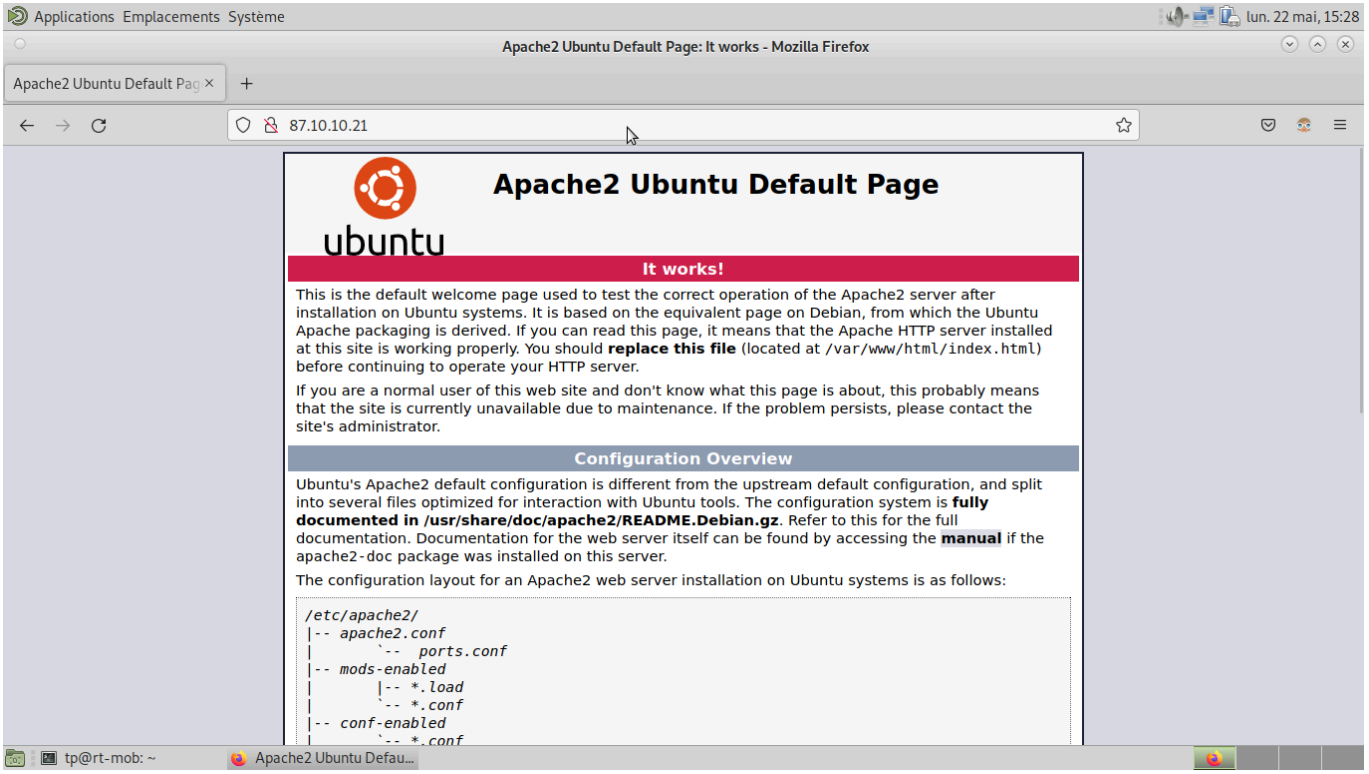
Configuration interface in FW B



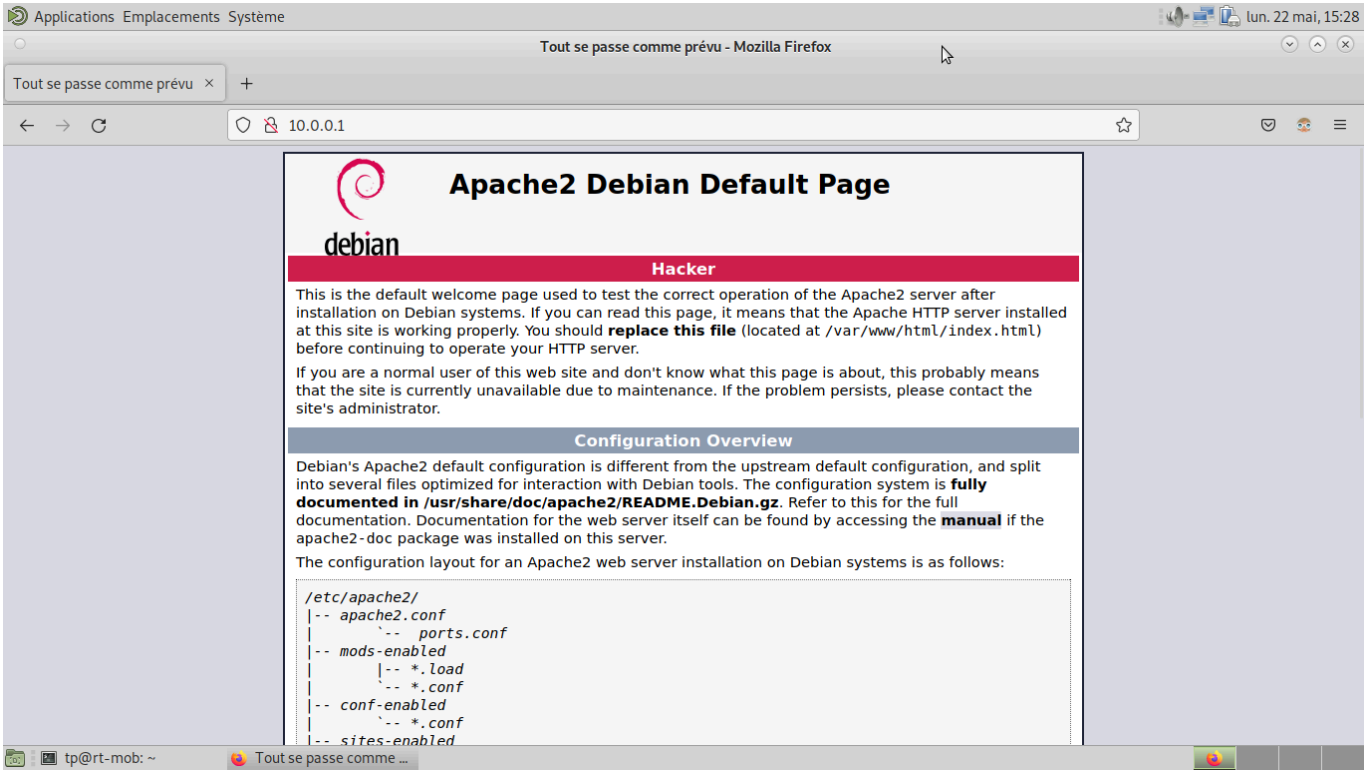
Configuration interface out FW B



Configuration interfaces FW A



Accès serveur WEB B (@IP virtuelle) depuis Client A



Accès serveur WEB A depuis Client A

Pour cette partie de configuration, nous avons eu du mal à établir les bonnes règles de NAT et filtrage afin d'avoir accès dans les bon sens de communications. Les indications du sujet étant très ouvertes, nous devions réfléchir aux règles adéquates afin d'obtenir une communication fonctionnelle.

Suite à de nombreuses réflexions et plusieurs tests, nous avons pu établir les bonnes règles. L'accès aux différents réseaux se fait dans le respect des "règles de l'art". Les serveurs sont accessibles de partout en fonction de leur adresses IP (virtuelles : NAT ; ou physique : LAN).

Tâche 3 Serveurs HTTP/HTTPS et serveur FTP/FTPS

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	6h/pers = 12 heures-hommes

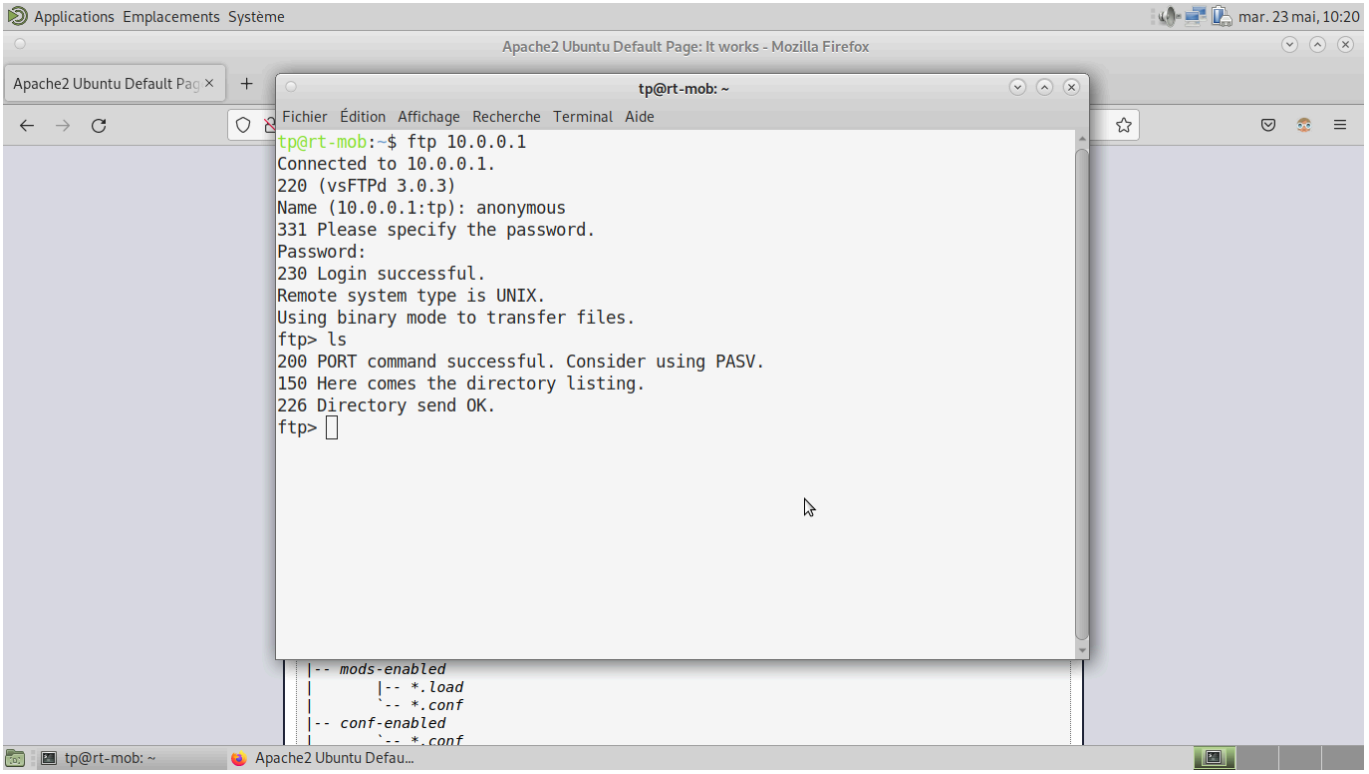
Estimation du temps passé sur cette tâche en heure-homme : 2 heures

Objectif : Configuration des firewalls pour protéger les réseaux internes et DMZ

Sous-tâches	Evaluation prof
Installez les serveurs http -> ok Installez le serveur FTP -> ok Activez HTTPS et FTPS -> ok Mettez à disposition un fichier sur le serveur FTP -> ok	100%
Installez un CMS et créez un petit site web -> ok	100%
Testez l'accès à vos serveurs -> ok	100%

Rapport

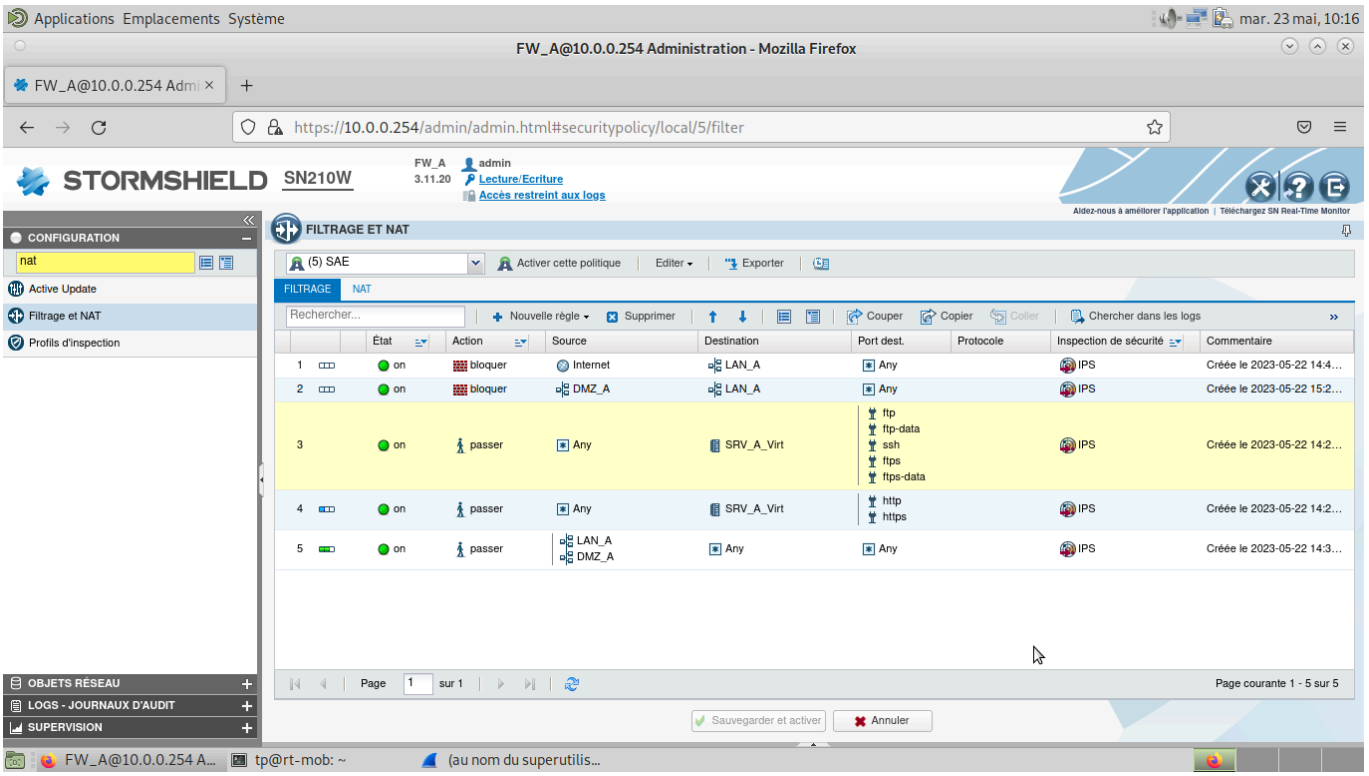
(Expliquez votre démarche, écrivez les commandes principales que vous avez tapées, insérez les captures d'écran de vos tests, etc.)



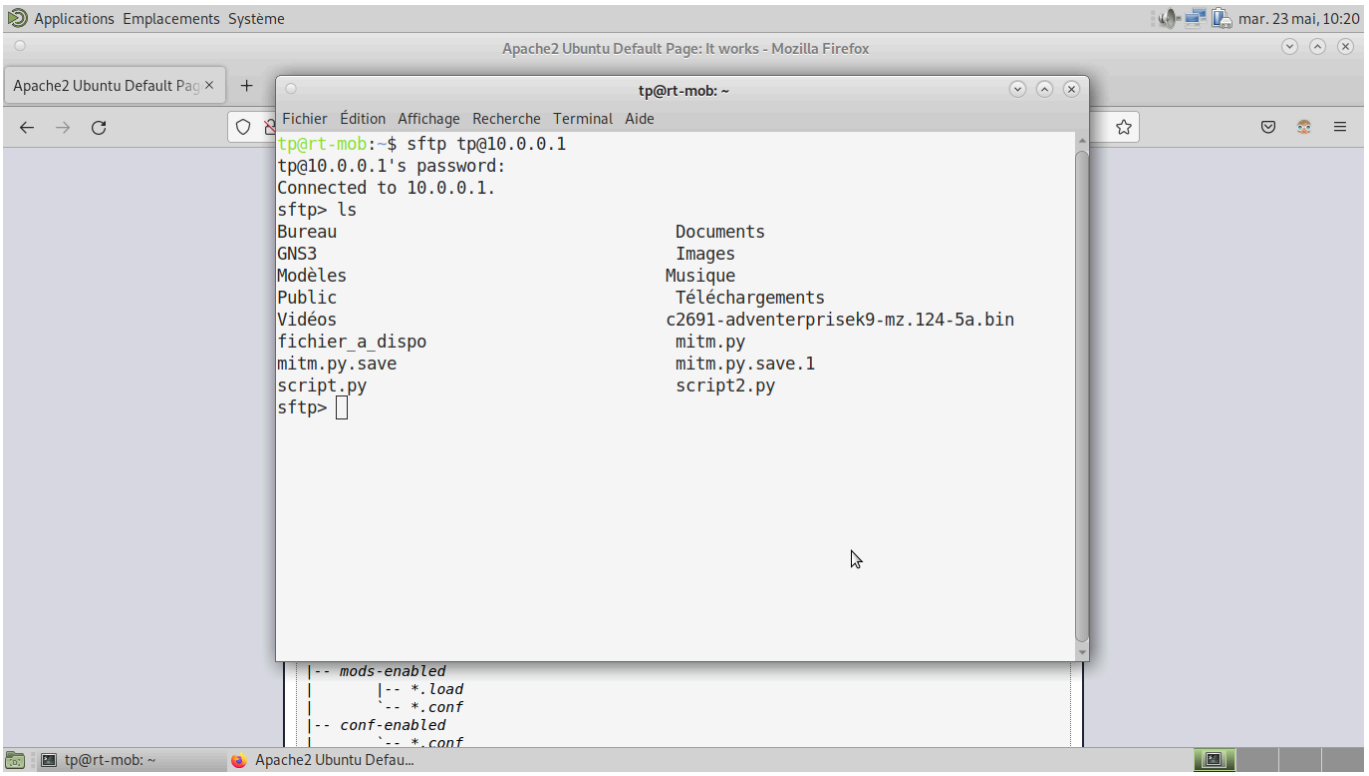
Connexion du LAN A sur le serveur FTP (en local)

```
root@rt-mob:/home/tp# ftp 87.10.10.11
Connected to 87.10.10.11.
220 (vsFTPd 3.0.3)
Name (87.10.10.11:tp): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Connexion du LAN B sur le serveur FTP (distant)



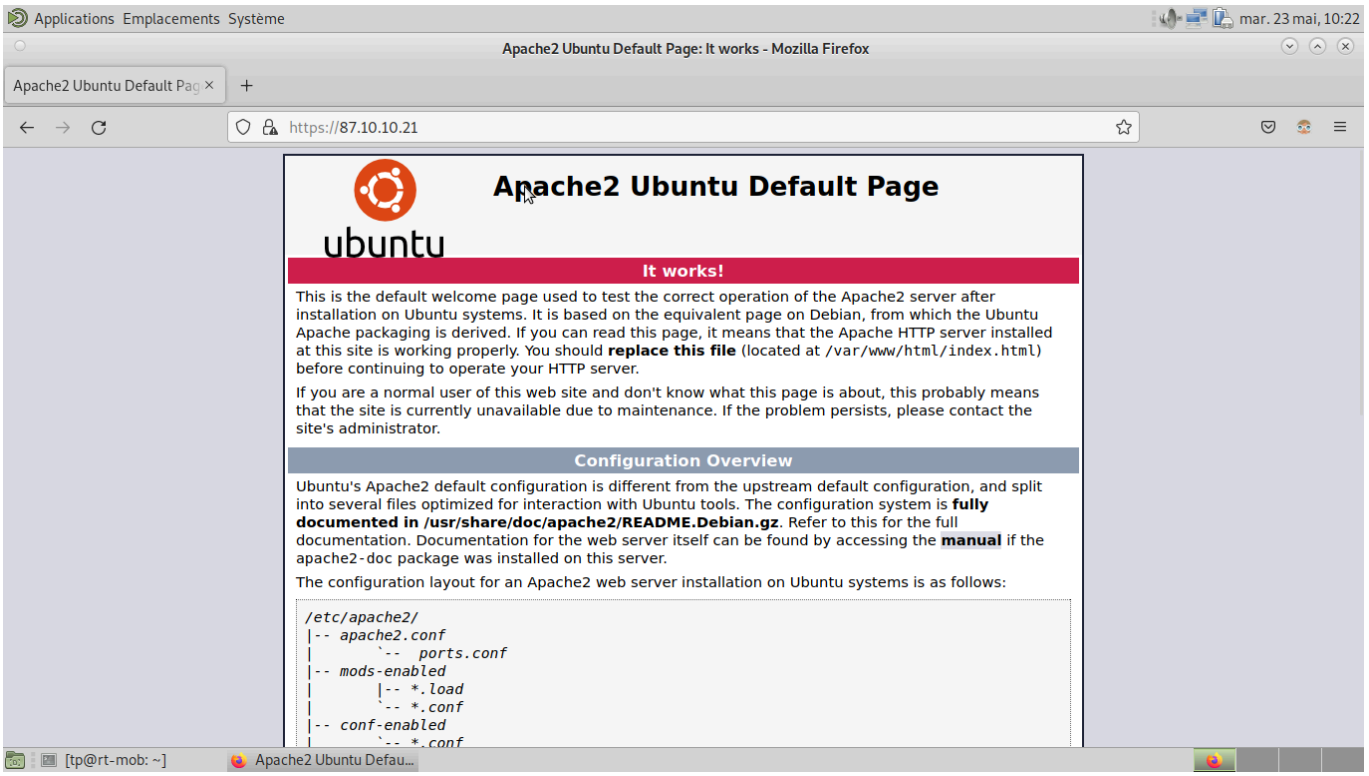
Politique de filtrage avec règle d'accès au serveur FTP



Connexion du LAN A au serveur SFTP (local)
avec le "fichier à disposition"

Connexion depuis le LAN B au serveur SFTP (distant)

Connexion du LAN B au serveur Web A en HTTPS (distant)

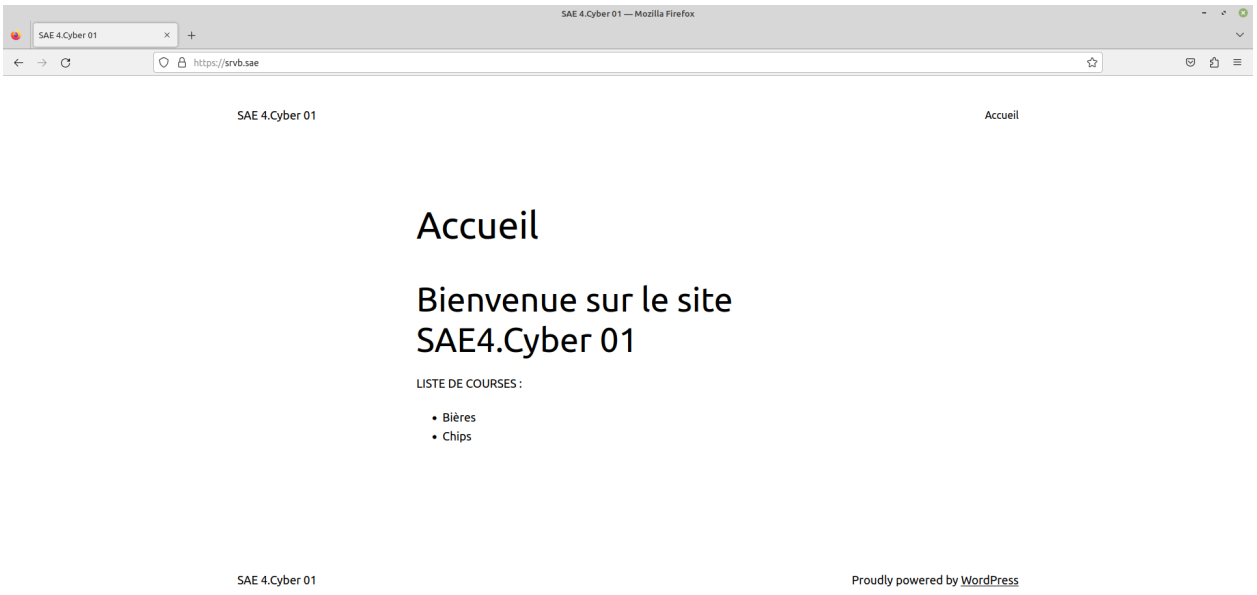


Connexion du LAN A au serveur Web B en HTTPS (distant)

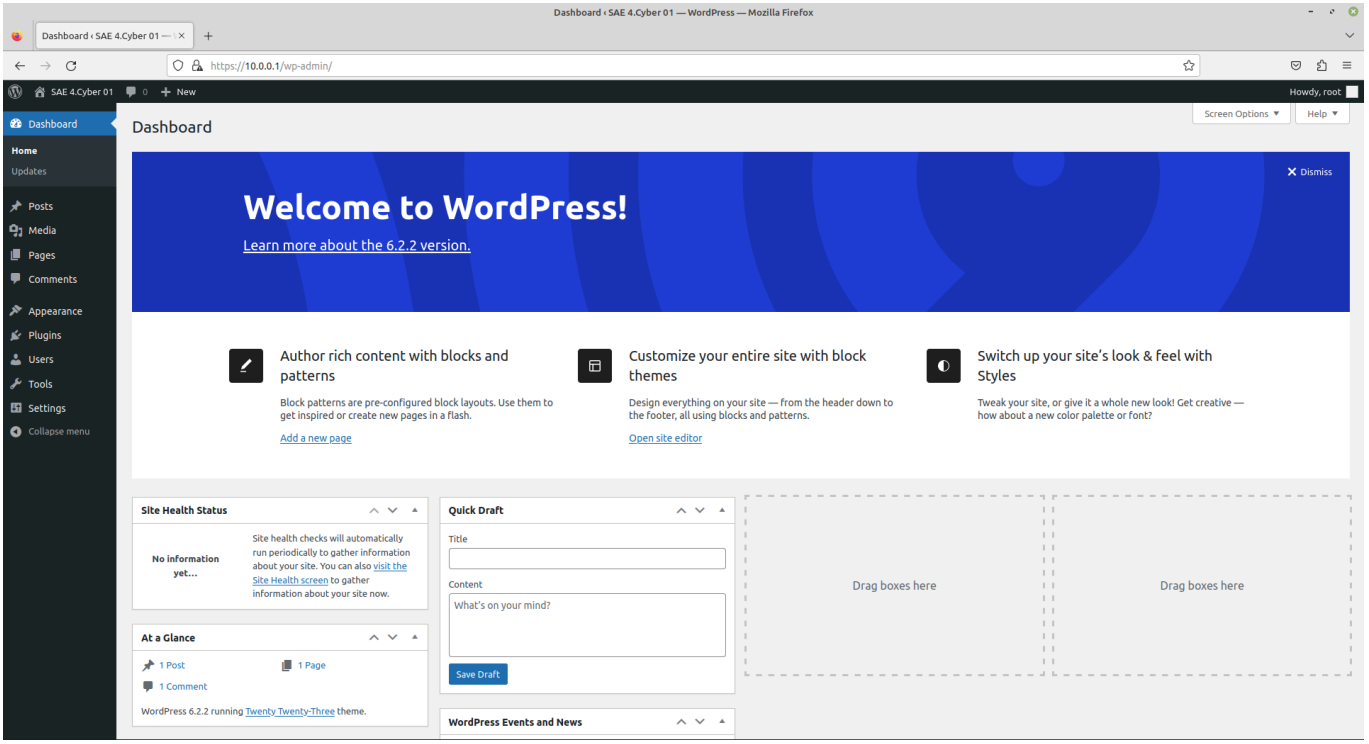
Pour ces étapes nous avons fait des choix de configurations assez drastiques : le service FTP ne sera utilisé que par l'utilisateur Anonymous dans un cadre de lecture seule, tandis que le SFTP est réservé aux utilisateurs enregistrés pour des actions de lecture/écriture.

Pour l'HTTPS il suffit d'utiliser les commandes "a2enmod & a2ensite" pour activer le service (on utilise ici les clés auto-signées pré-définies).

La politique de filtrage n'autorise que les connexions FTP/SFTP & HTTP/HTTPS depuis l'extérieur sur l'@IP virtuelle du serveur FTP/Web.
NB : il a été nécessaire d'autoriser aussi le protocole SSH dans le filtrage pour pouvoir accéder au SFTP depuis l'extérieur. Ce détail nous a fait rencontrer beaucoup de difficulté puisque le firewall n'autorise pas SSH avec les port par défauts "ftps & ftps-data (ports 990 & 989).



Page d'accueil du CMS



Page d'administration du CMS

Pour le CMS nous avons fait le choix d'utiliser Wordpress, le leader dans ce domaine et opensource. Afin que le site soit accessible de l'internet mais aussi du LAN nous avons créer 2 Virtual Hosts sur Apache2, l'un écoutant sur l@IP du LAN et l'autre sur l'@IP de son interface WAN.

D'une manière plus générale afin qu'aucun avertissement de certificat ne soit lévée nous avons défini le CN du certificat sur un FQDN. Nous avons renseigner dans le fichier /etc/hosts de chaque client la correspondance entre l'@IP du CMS (selon où se situe le client) et sont FQDN. Ainsi plus aucune erreur ne sera levée pour un "BAD_CERT_DOMAIN".

Tâche 4 Authentification transparent par certificat SSL

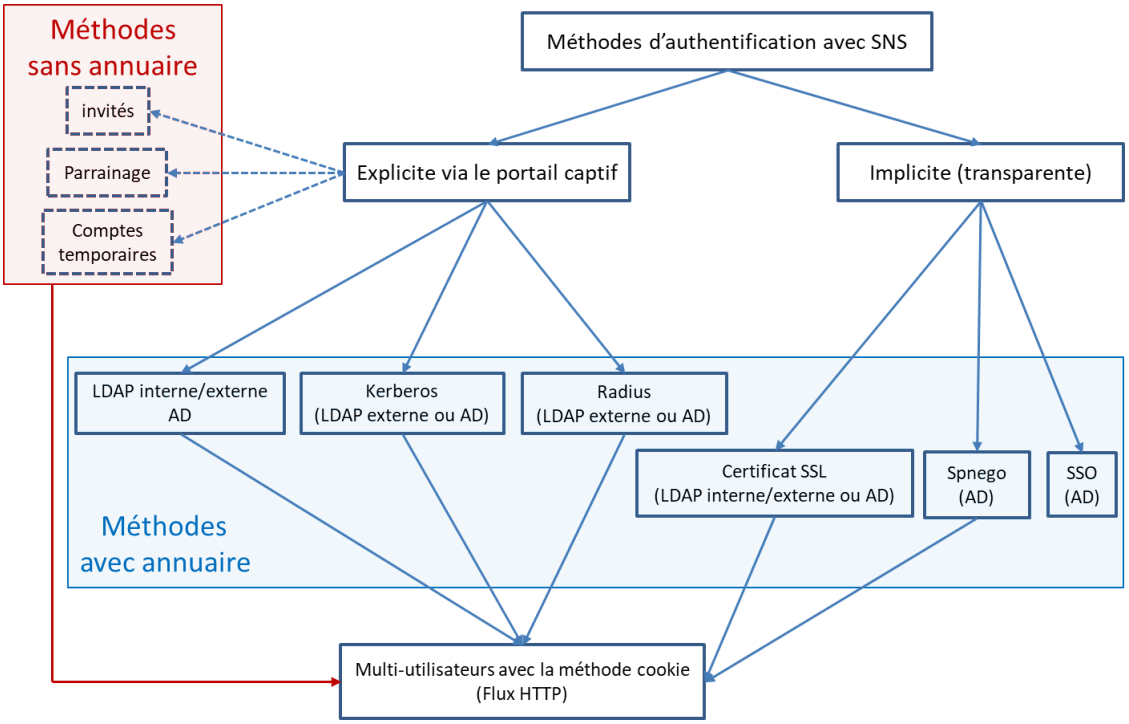
Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2x6h=12 heure-homme

Estimation du temps passé sur cette tâche en heure-homme : 10h

Objectif : Mettre en place une authentification transparente pour les utilisateurs

Les firewalls implémentent plusieurs méthodes d'authentification qui peuvent être classées en deux catégories :

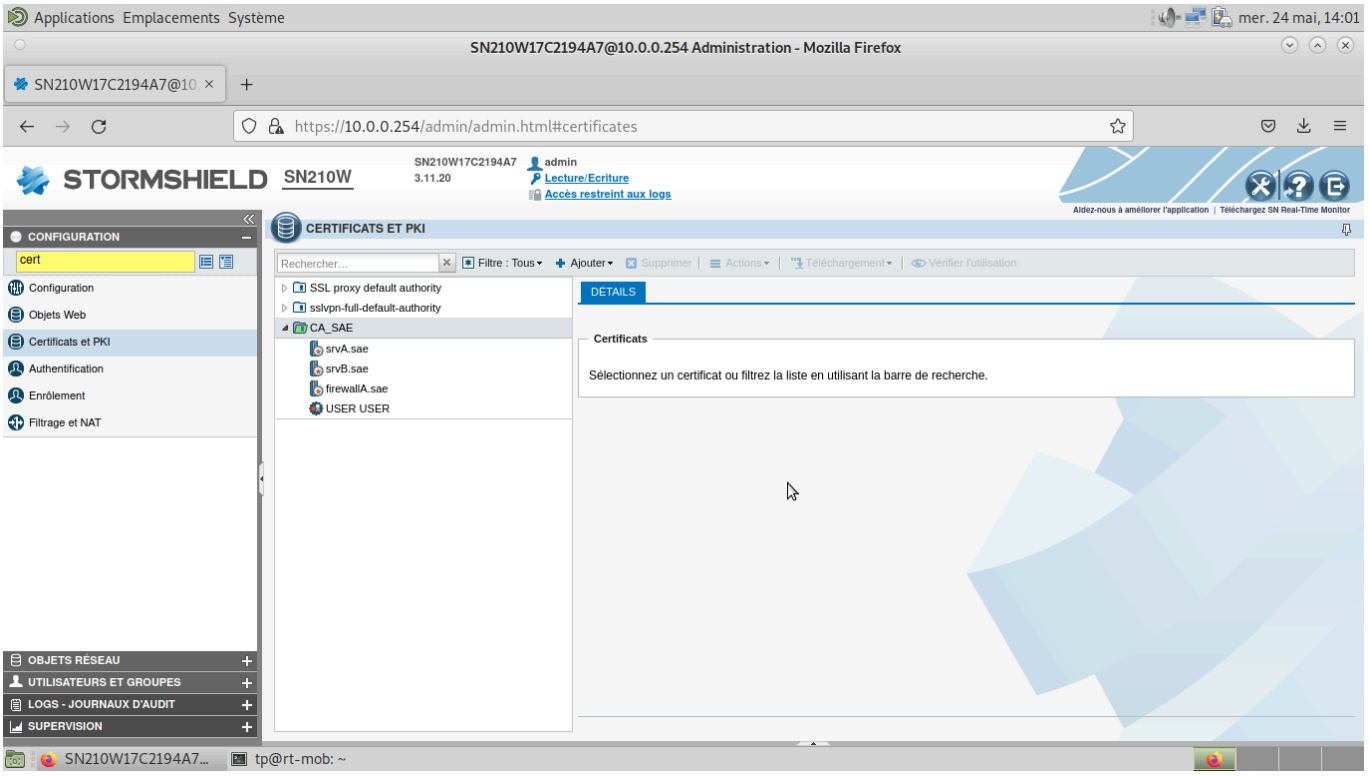
- Les méthodes explicites via le portail captif : l'utilisateur est redirigé vers le portail captif pour saisir un couple identifiant/mot de passe.
- Les méthodes implicites (transparentes) : l'authentification est transparente vis-à-vis de l'utilisateur qui n'a pas besoin de saisir son couple identifiant/mot de passe explicitement pour accéder au réseau.



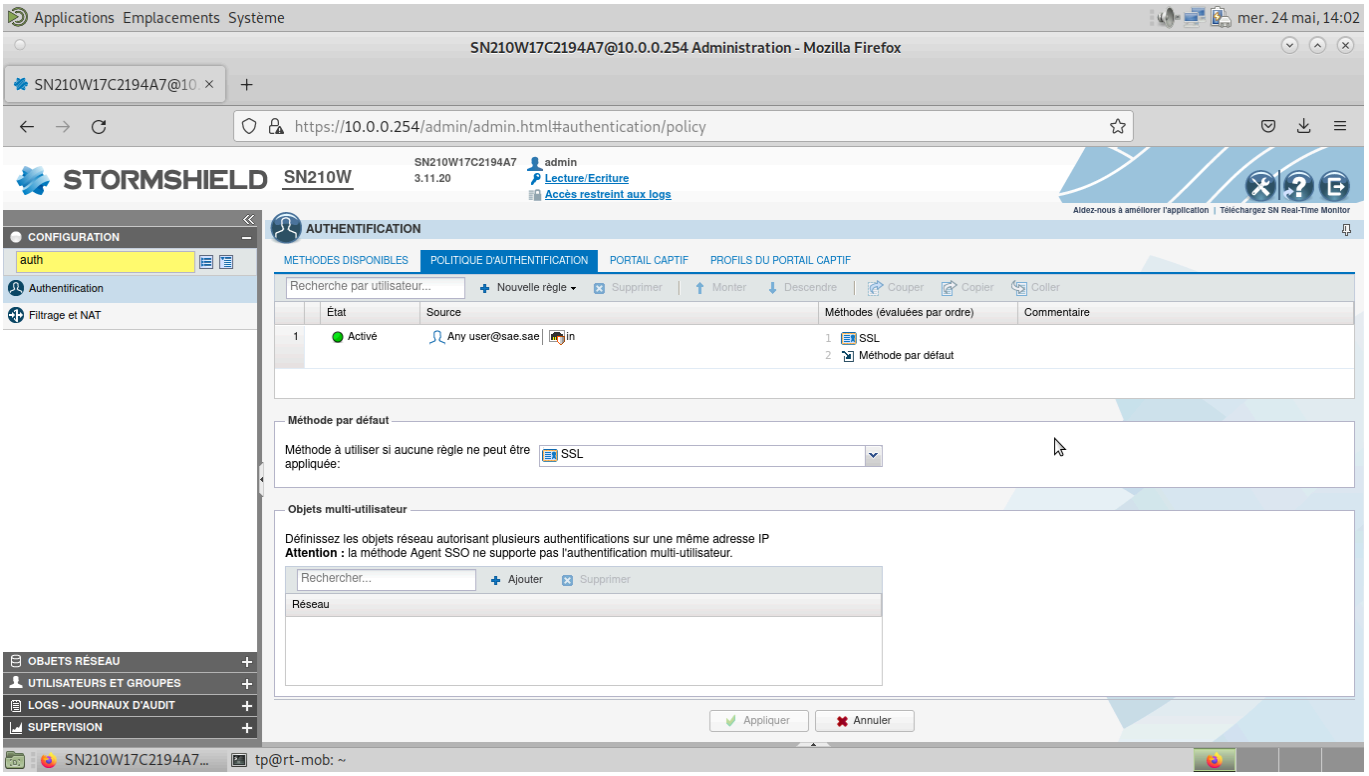
Sous-tâches	Evaluation prof
Création d'une autorité racine -> ok	100%
Activer l'authentification par certificat SSL -> ok	100%
Importez le certificat dans le navigateur -> ok	100%
Testez votre configuration -> ok	100%

Rapport

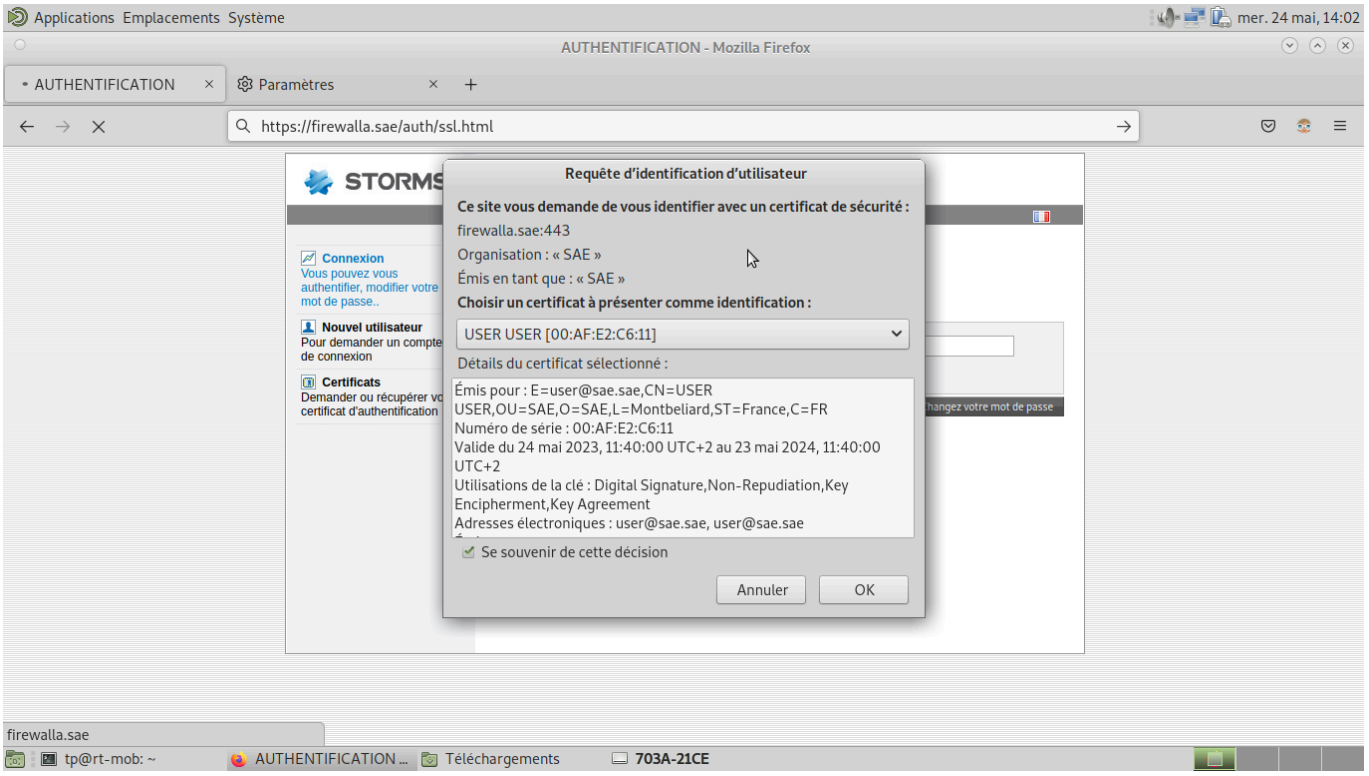
(Expliquez votre démarche, insérez les captures d'écran de votre configuration, de vos tests, etc.)



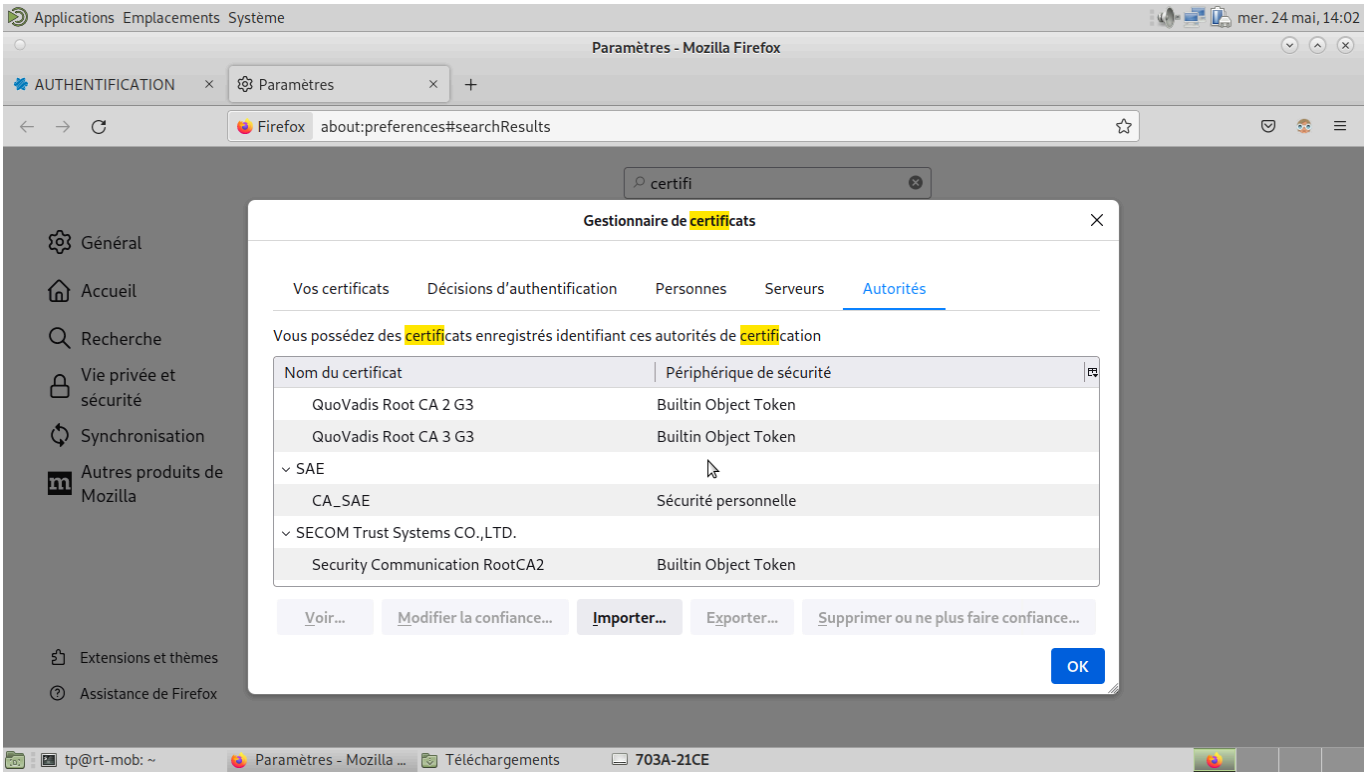
Résumé de la CA et des certificats



Authentification définie sur le paramètre SSL avec règle d'authentification



Résumé du certificat utilisateur renseigné dans le navigateur client



Présence du certificat de la CA dans le navigateur client

A présent l'utilisateur peut accéder à tout les sites et services réglementés par le portail captif. Par exemple des sites web, des connexions SSH/FTP, ou encore un simple ping.

Tâche 5 Mettre en place un IDS et le tester

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	9h/pers = 18 heures-hommes

Estimation du temps passé sur cette tâche en heure-homme : 20h

Objectif : Installation et configuration de Snort

Vous devrez être capable de détecter les évènements suivants :

- Tentative de connexion SSH sur vos serveurs depuis l'extérieur
- Attaque DoS sur votre serveur Web avec des GET requests
- Détection des URIs non-normalisées
- Détection d'un login raté sur le serveur FTP
- Détection d'une attaque DoS avec TCP SYN
- Détection de paquets fragmentés de taille < 500 ou > 2000

Sous-tâches	Evaluation prof
Installation de Snort -> ok	100%
Création des règles -> ok	100%
Test des règles -> ok	100%

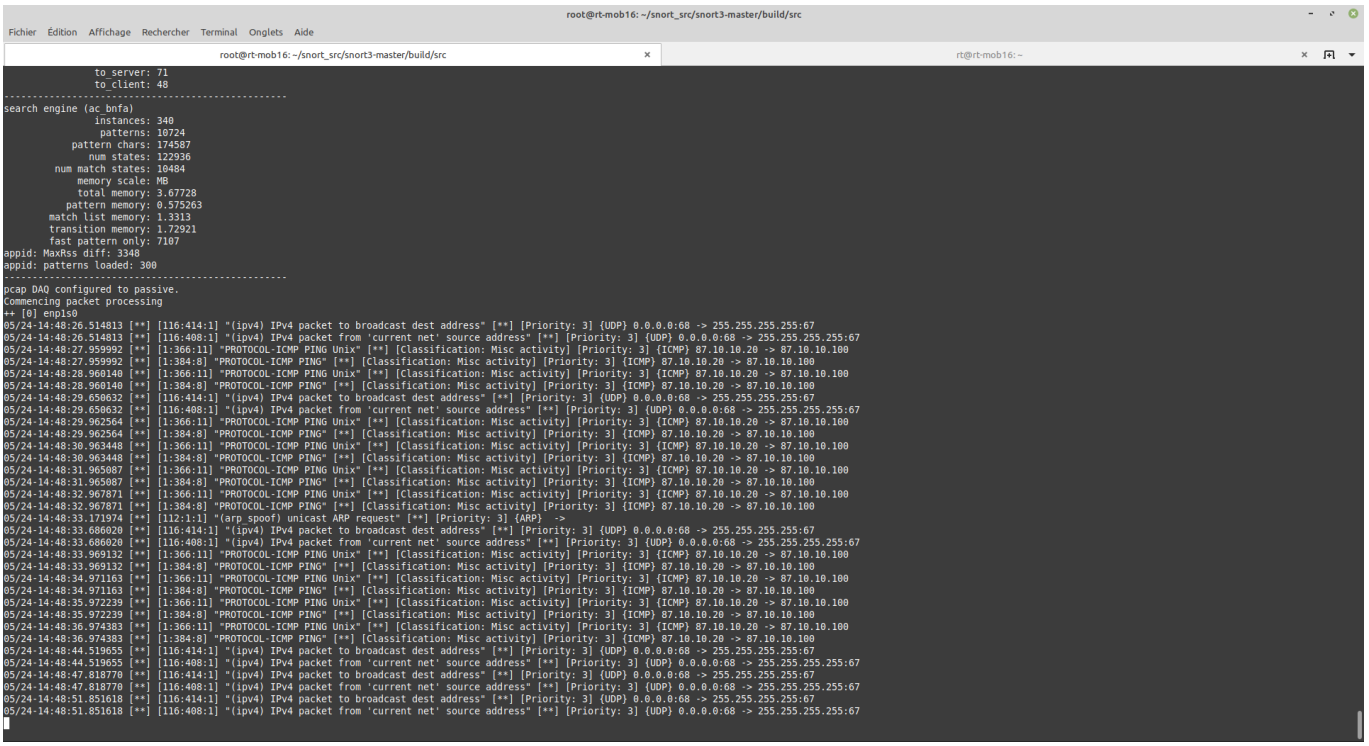
Rapport

(Expliquez votre démarche, le format d'une règle, écrivez vos règles, insérez les captures d'écran des résultats de détection de Snort, etc.)

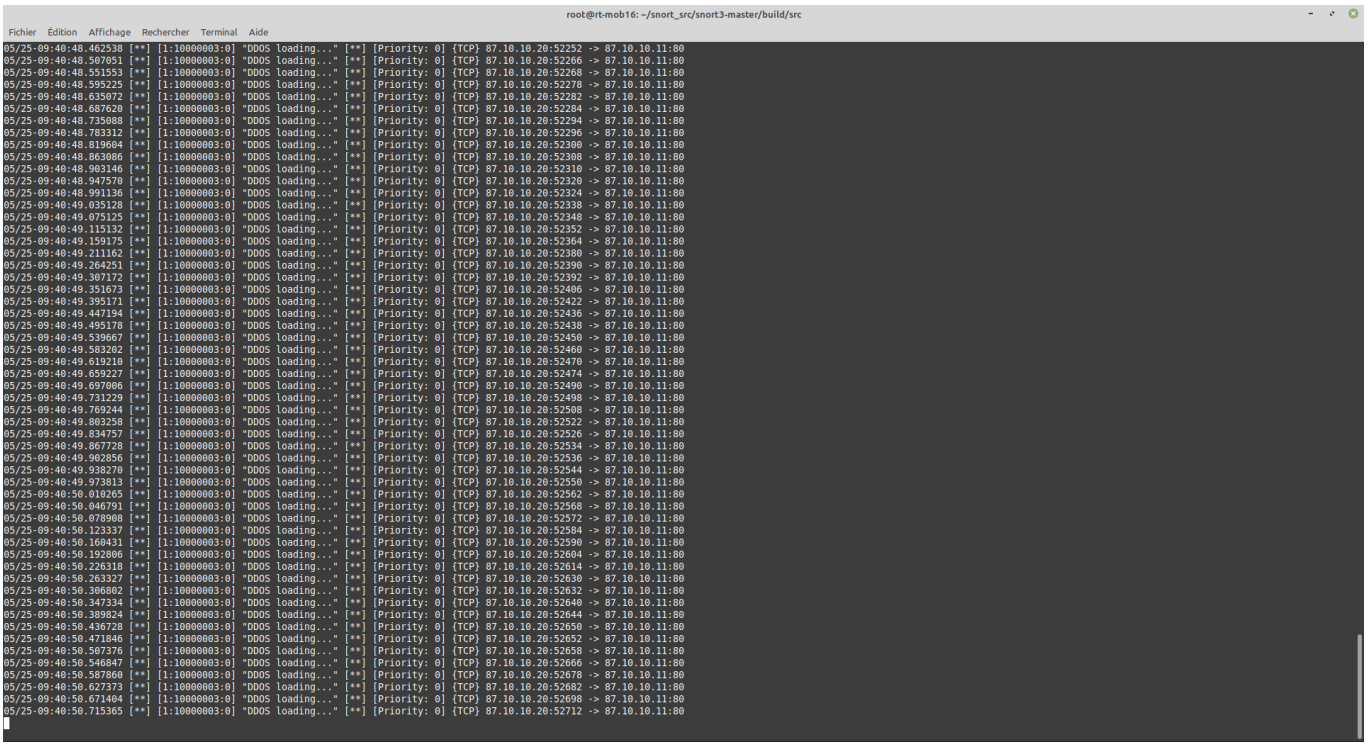
```
root@rt-mob16:~/snort_src/snort3-master/build/src# nano  /etc/systemd/system/snort3-nic.service
root@rt-mob16:~/snort_src/snort3-master/build/src# systemctl enable --now snort3-nic.service
root@rt-mob16:~/snort_src/snort3-master/build/src# systemctl status snort3-nic.service
● snort3-nic.service - Set Snort 3 NIC in promiscuous mod and Disable GRO, LRO on boot
     Loaded: loaded (/etc/systemd/system/snort3-nic.service; enabled; vendor preset: enabled)
     Active: active (exited) since Wed 2023-05-24 14:29:44 CEST; 17s ago
    Process: 61526 ExecStart=/sbin/ip link set dev enp1s0 promisc on (code=exited, status=0/SUCCESS)
    Process: 61534 ExecStart=/sbin/ethtool -K enp1s0 gro off lro off (code=exited, status=0/SUCCESS)
   Main PID: 61534 (code=exited, status=0/SUCCESS)

mai 24 14:29:44 rt-mob16 systemd[1]: Starting Set Snort 3 NIC in promiscuous mod and Disable GRO, LRO on boot...
mai 24 14:29:44 rt-mob16 ethtool[61534]: Cannot change large-receive-offload
mai 24 14:29:44 rt-mob16 systemd[1]: Finished Set Snort 3 NIC in promiscuous mod and Disable GRO, LRO on boot.
root@rt-mob16:~/snort_src/snort3-master/build/src#
```

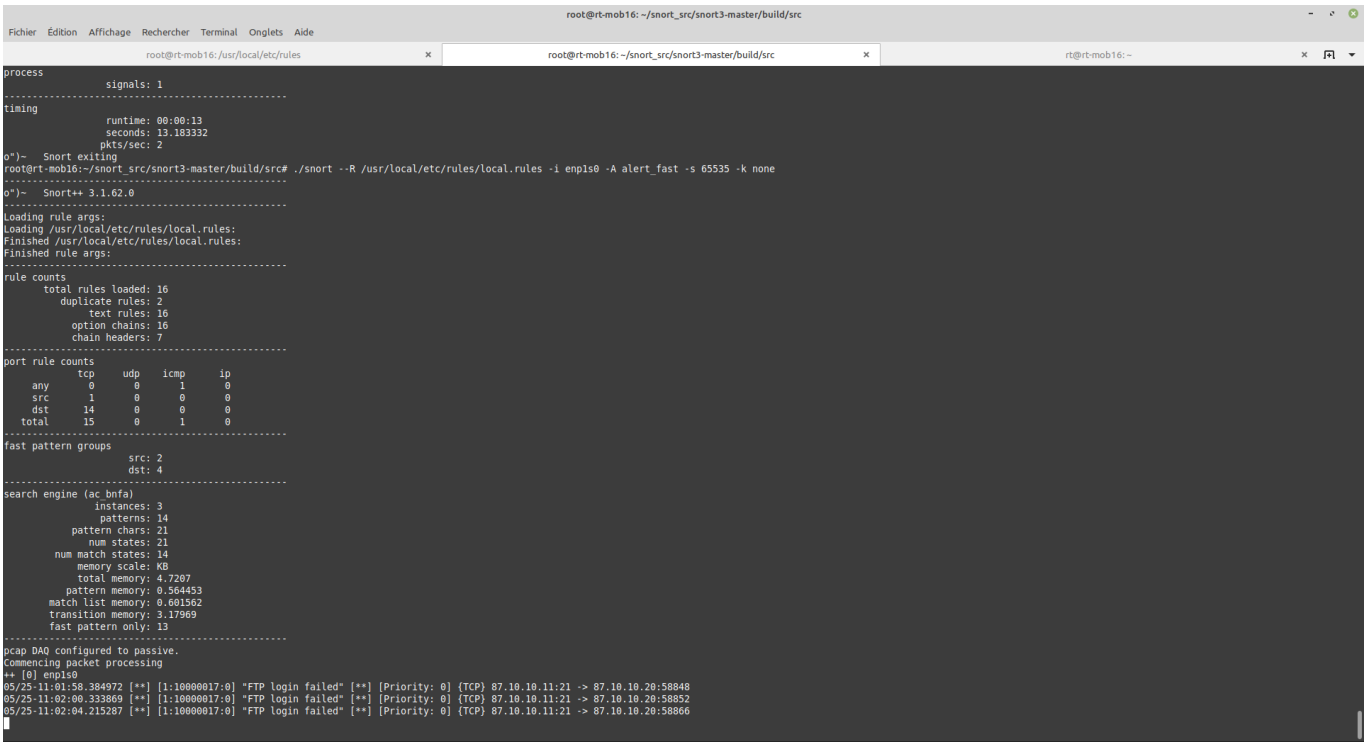
Snort3 configuré et opérationnel



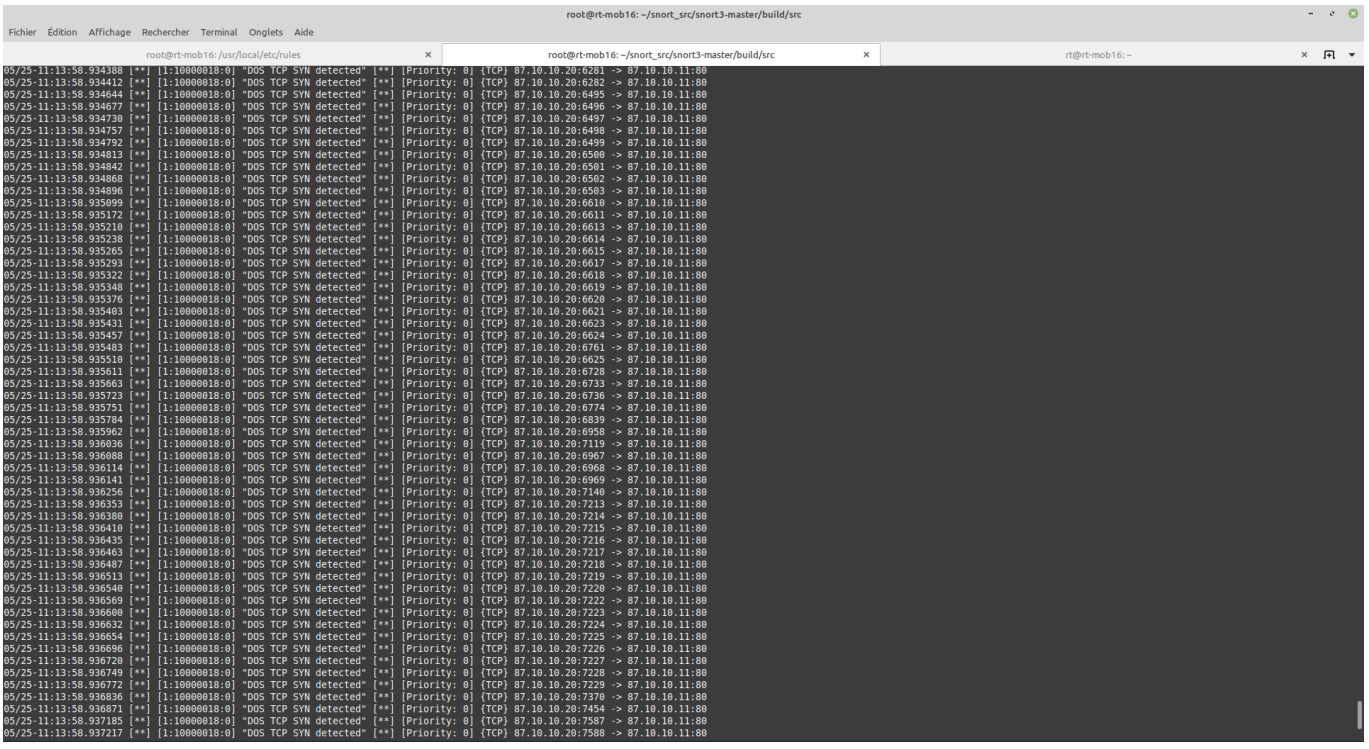
Alerte trafic ICMP sur Snort3



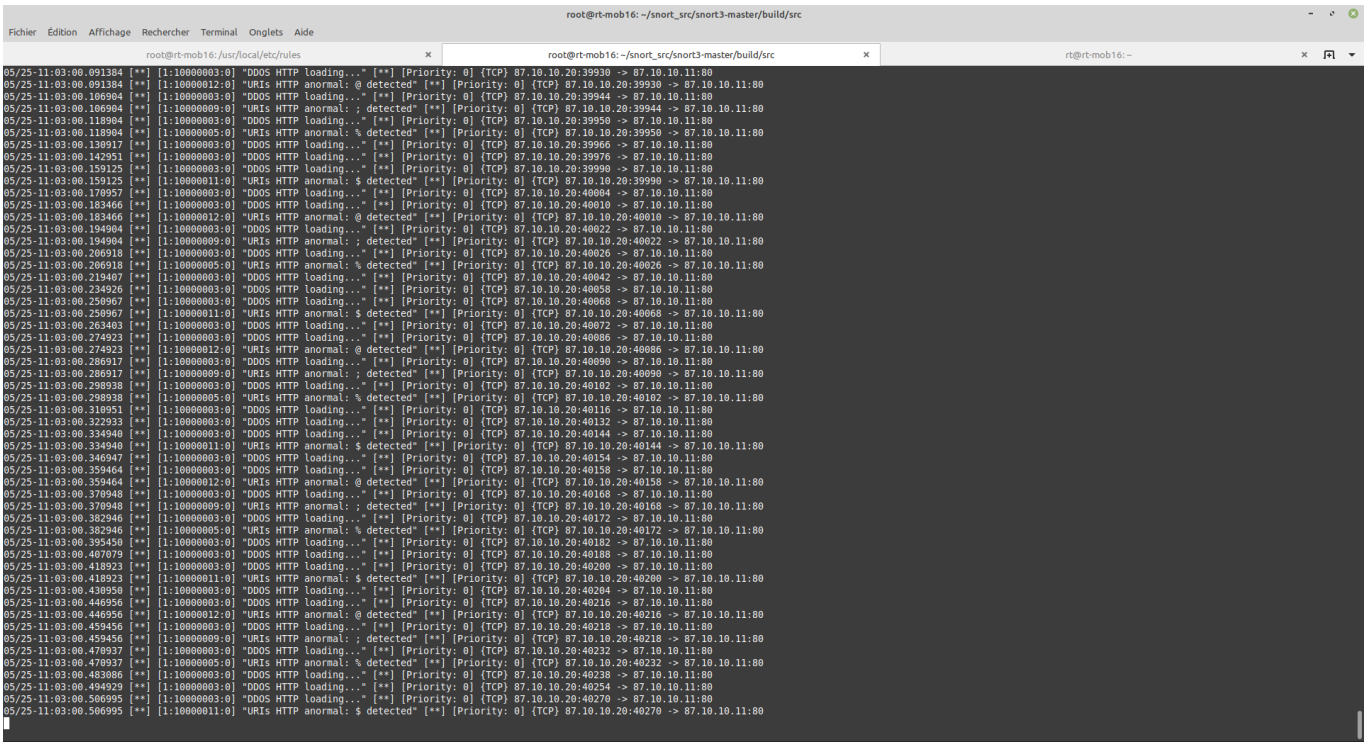
Alerte possible DDoS par requête GET sur Snort3



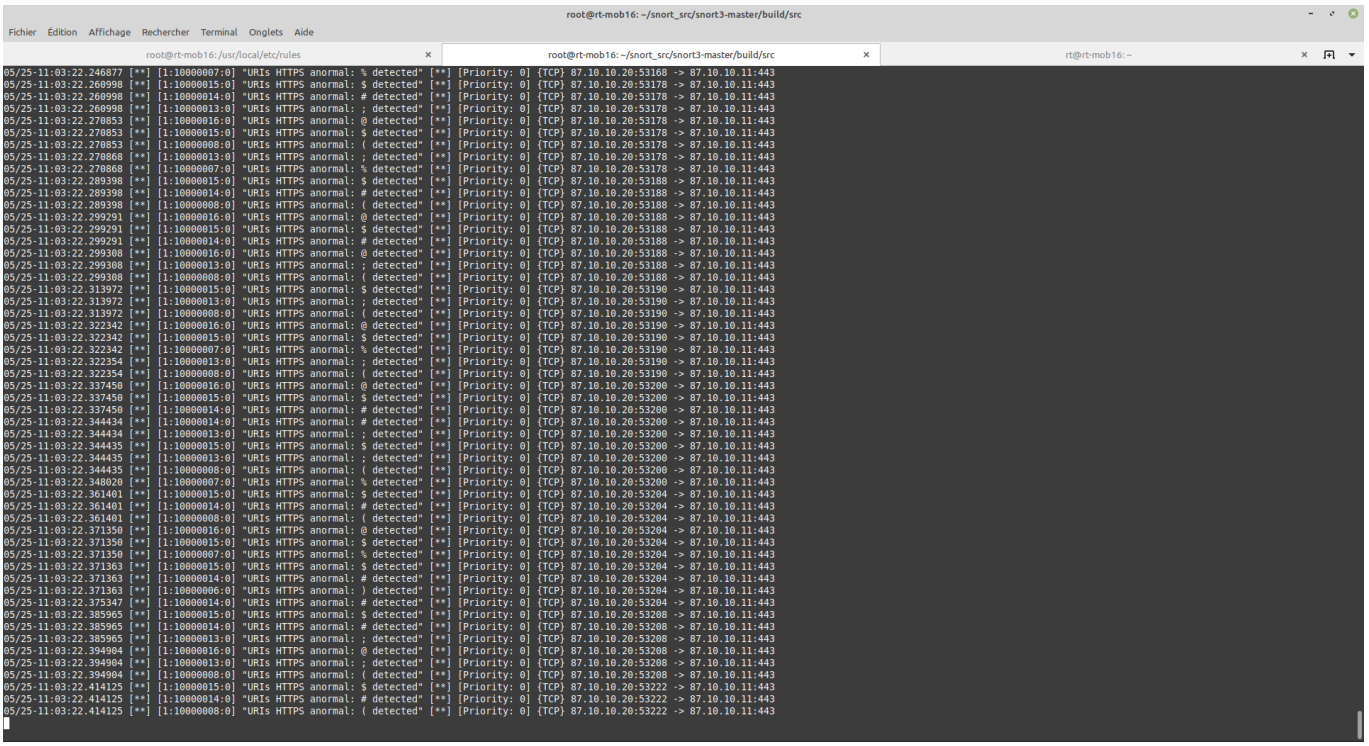
Alerte échec de connexion FTP sur Snort3



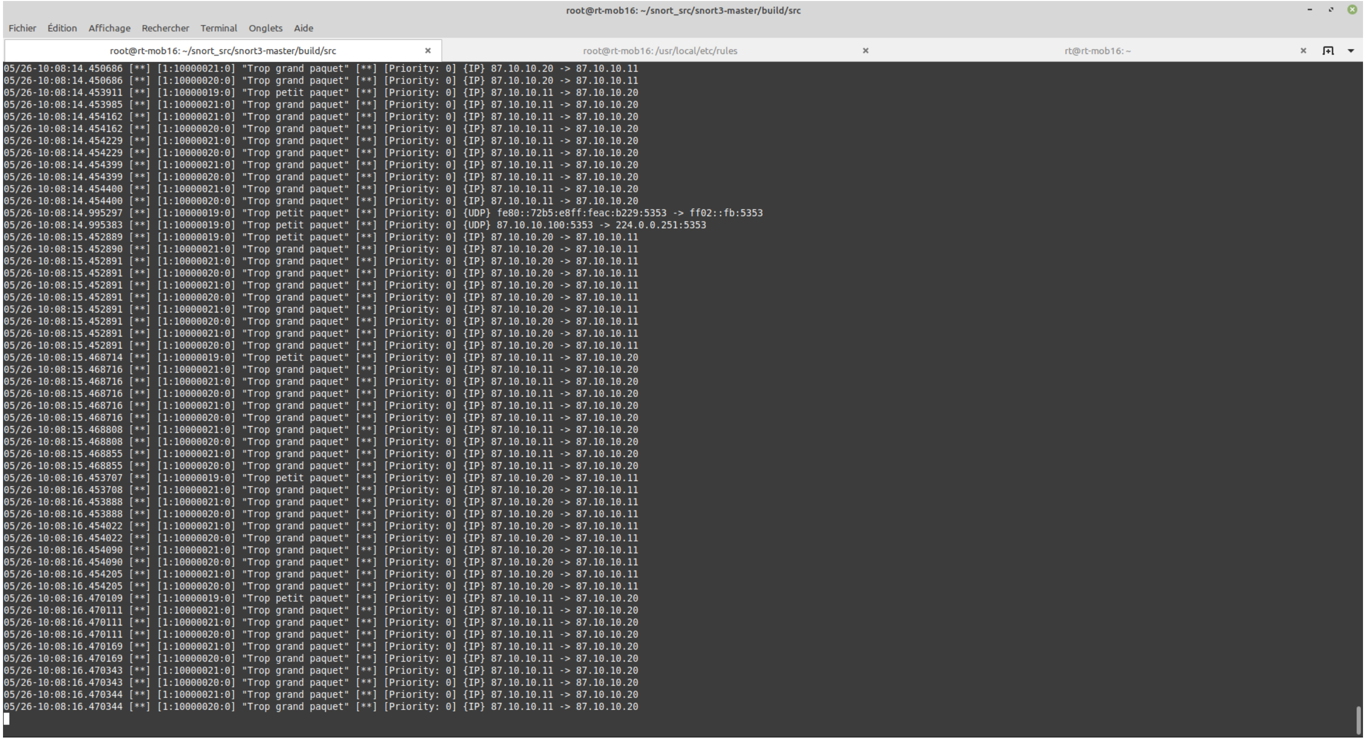
Alerte possible DDoS par SYN flag sur Snort3



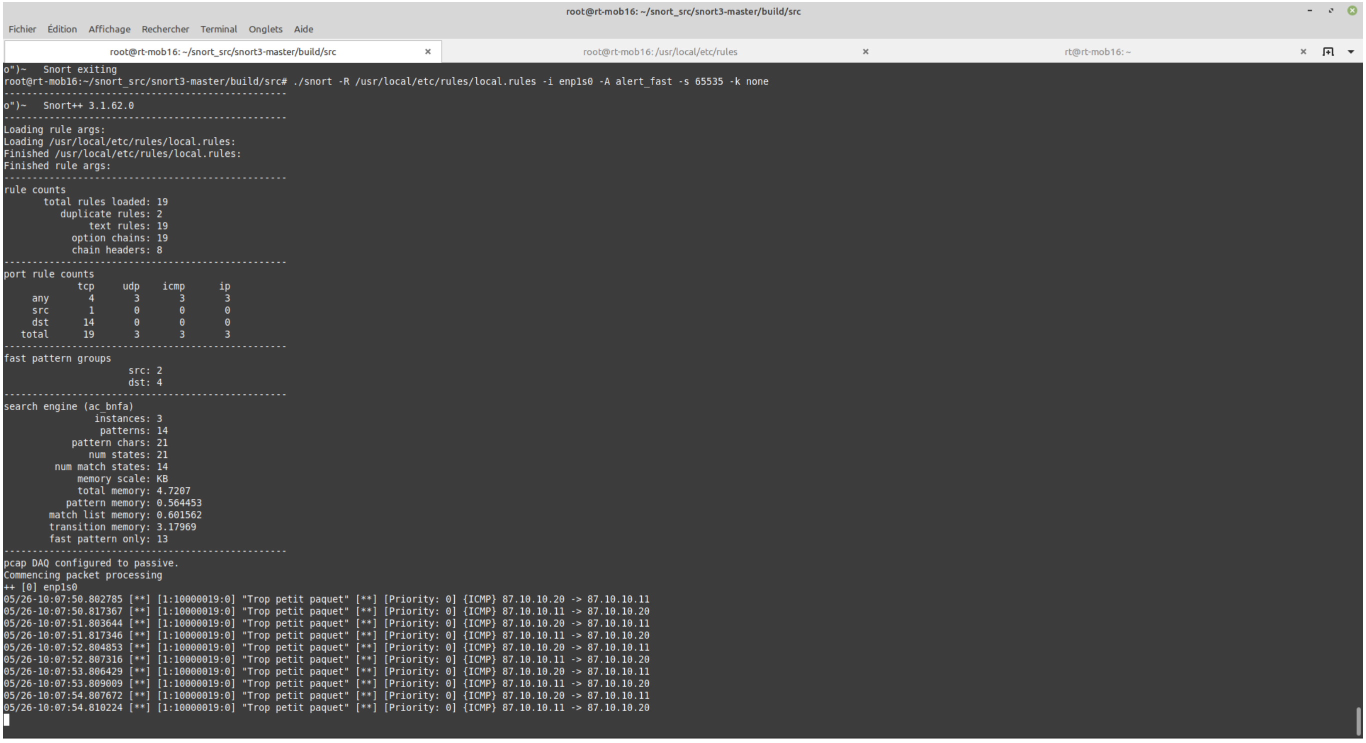
Alerte détection d'URI non-normalisés en HTTP sur Snort3



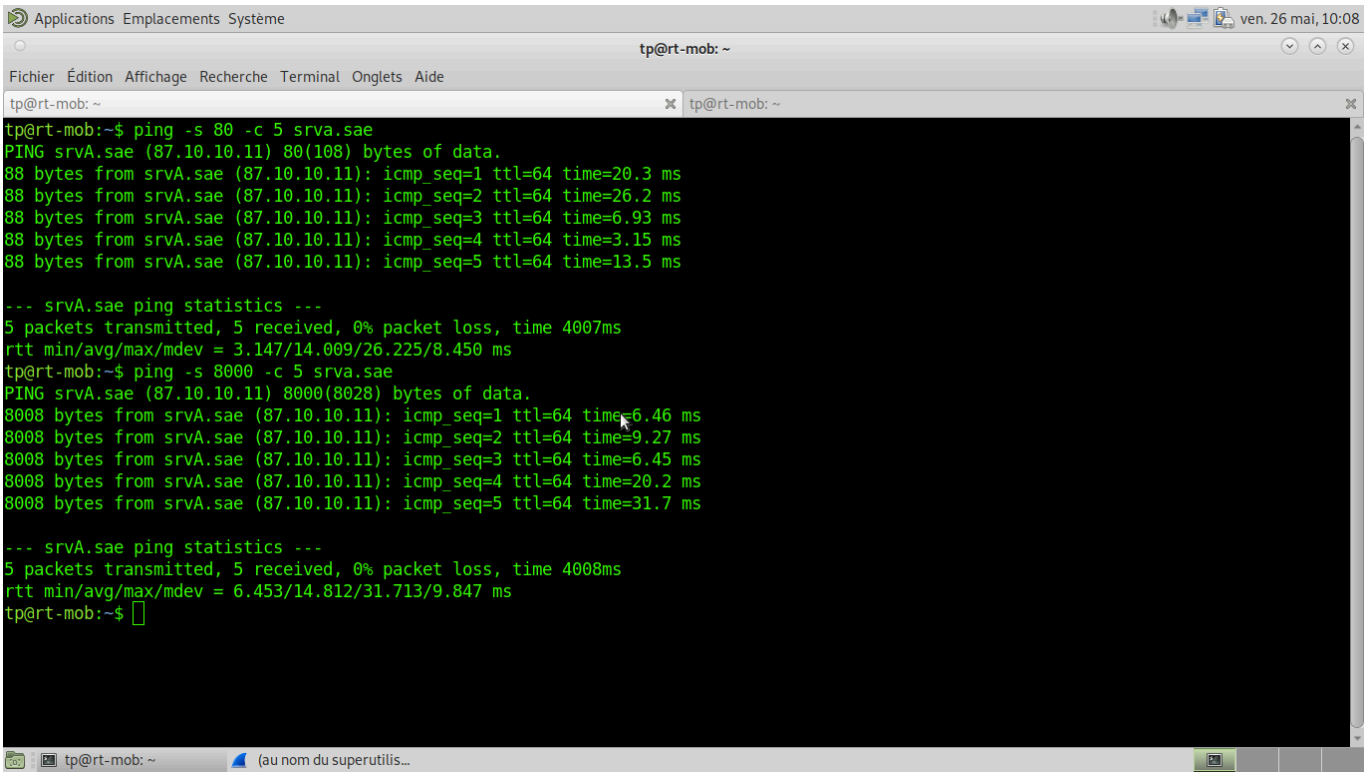
Alerte détection d'URI non-normalisés en HTTPS sur Snort3



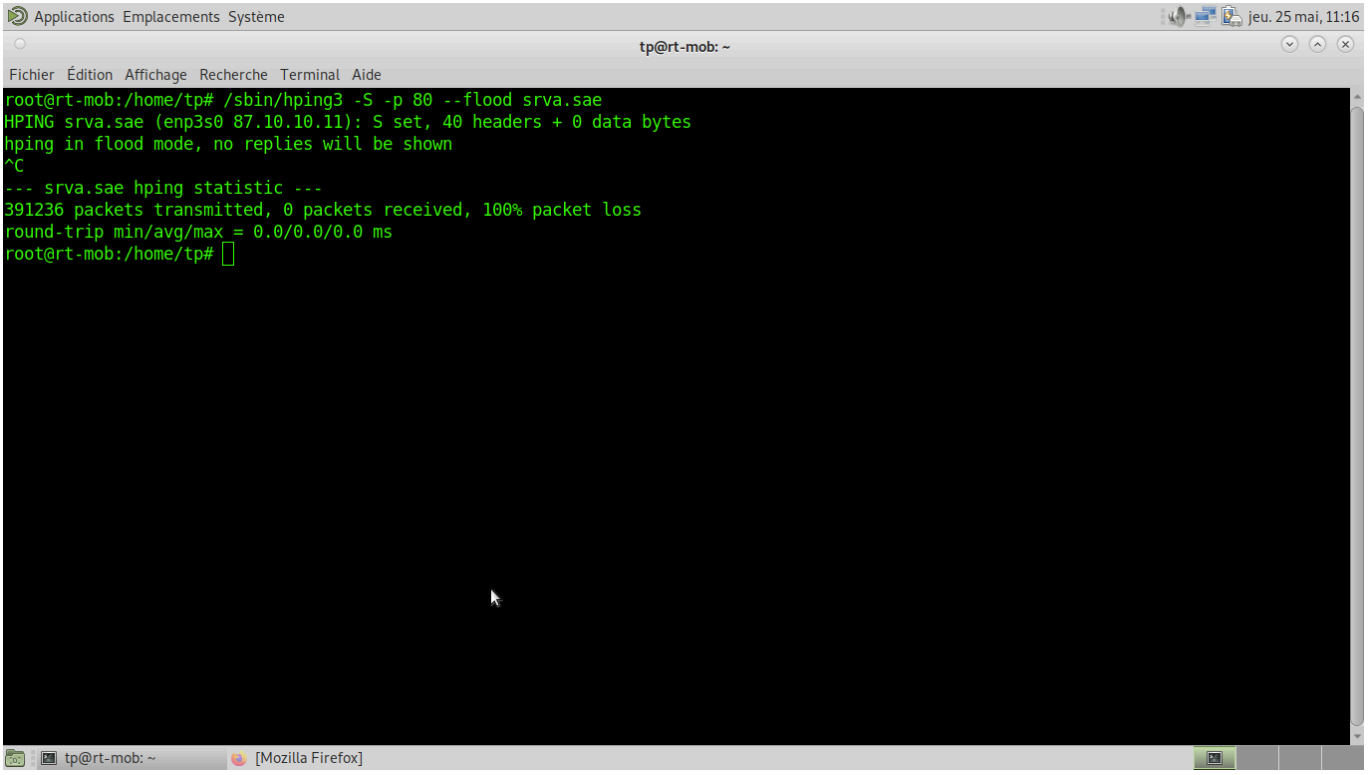
Alerte taille de fragmentation trop grande sur Snort3



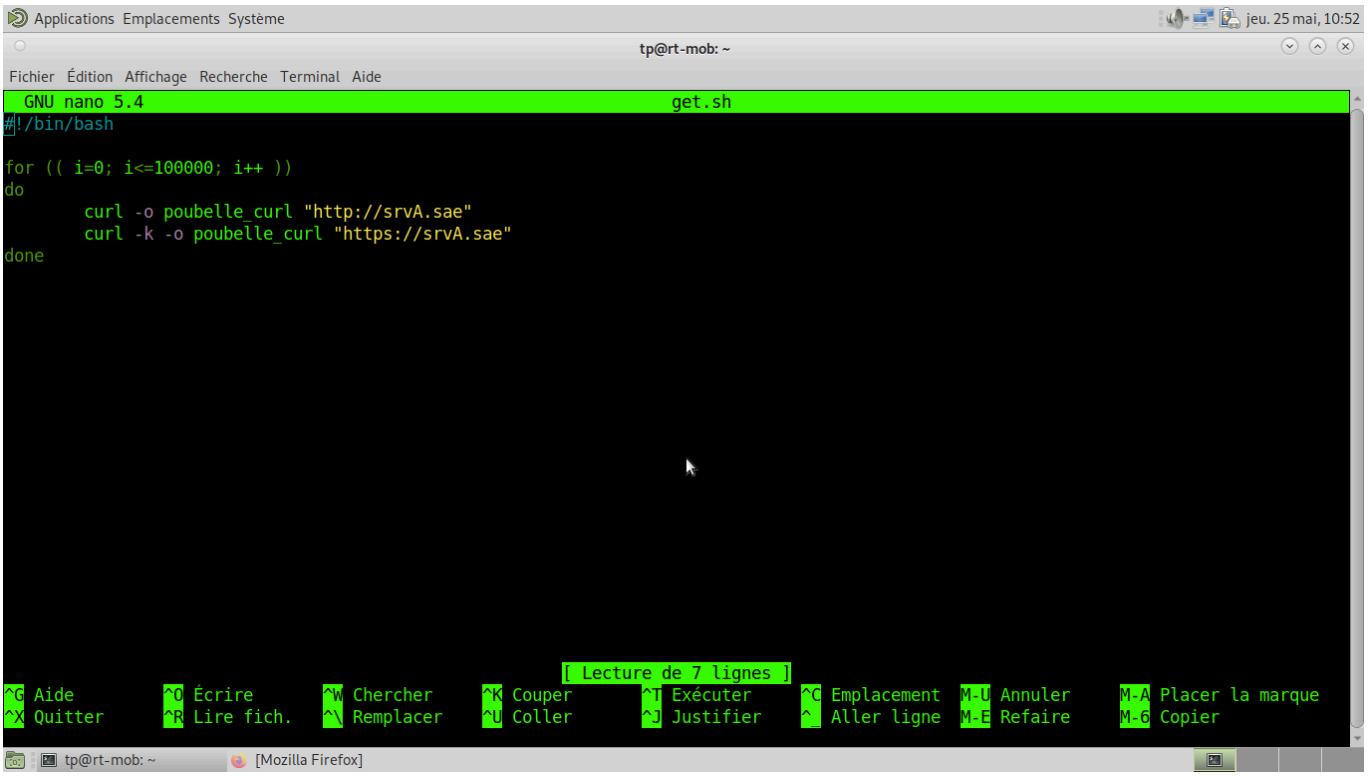
Alerte taille de fragmentation trop petite sur Snort3



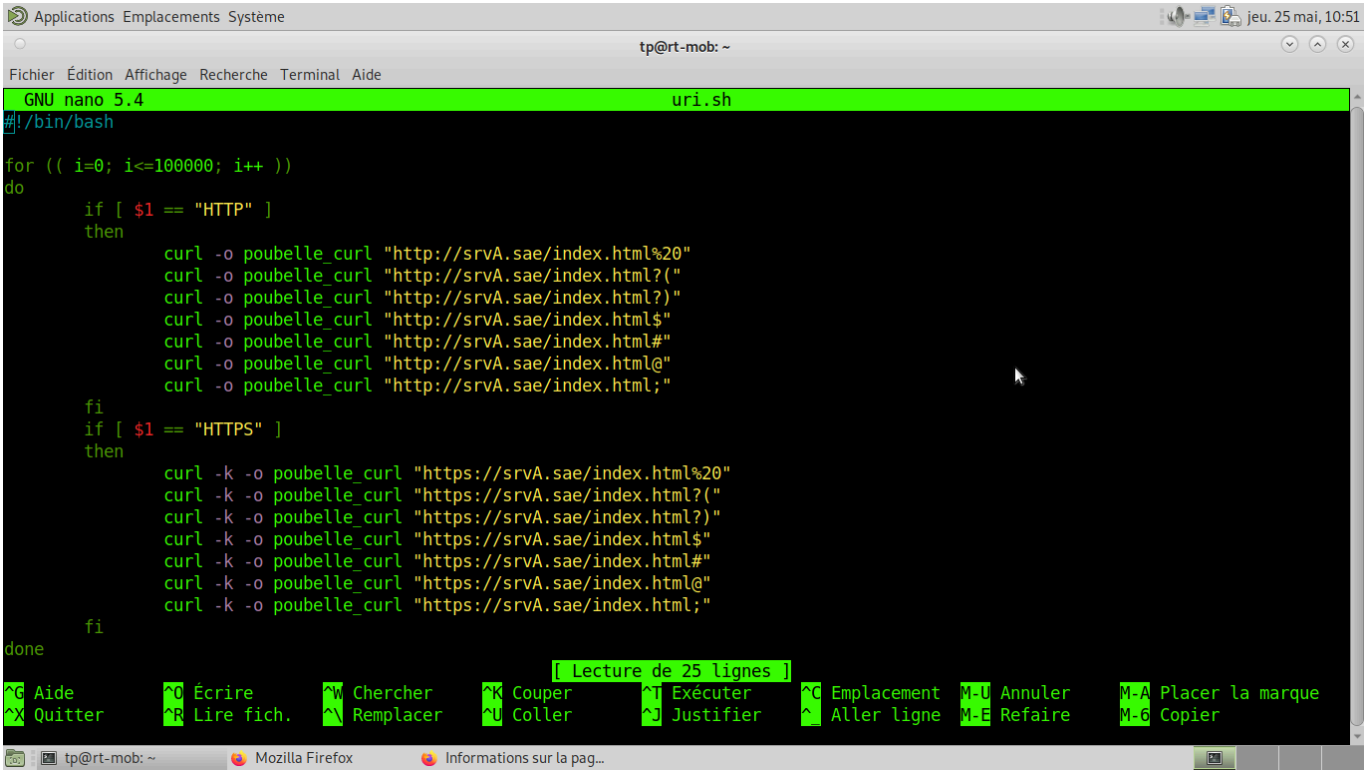
Test utilisé pour la fragmentation



Test utilisé pour le SYN DDoS



Test utilisé pour le GET DDoS



Test utilisé pour les URIs non-normalisés

Pour cette tâche-ci, nous n'avons rencontré que très peu d'erreurs mais ces dernières se sont avérées extrêmement bloquantes.

Tout d'abord la version de Snort utilisée ici est la 3.1, nous avions commencé avec la 2.9 mais celle-ci ne disposait pas de toutes les fonctionnalités dont nous avions besoin. De plus, lors du test des règles implémentées, certaines se recoupent (notamment le SYN DDoS & le GET DDoS), ce qui provoque un "recouvrement des règles" et une seule alerte est levée.

Afin d'éviter d'avoir des alertes à la moindre requête GET ou à un paquet flaggé SYN il a fallu implémenter en plus un contrôle du débit, ce contrôle possède une syntaxe très changeante selon les versions ce qui nous a longtemps induit en erreur.

Dans un souci de simplicité (pas de besoin particulier dans cette SAE), Snort3 intercepte les paquets de toutes origines et de toutes destinations via un port-mirroring sur le switch (Cisco Catalyst 3750v2) auquel il est relié. Il a aussi fallu créer des "règles test" afin de contrôler notamment la fragmentation des paquets (longueur max du fragment 2000 octets et MTU à 1514 octets).

Pour le contrôle de la fragmentation nous avons trop longtemps essayer d'utiliser le PréProcesseur Frag3, finalement la solution retenue aura été l'option de règle "fragoffset", à diviser en 2 règles distinctes (inférieure et supérieure).

Tâche 6 Attaque sur le Wifi

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	3h/pers = 6 heures-hommes

Estimation du temps passé sur cette tâche en heure-homme : 12 heures-hommes

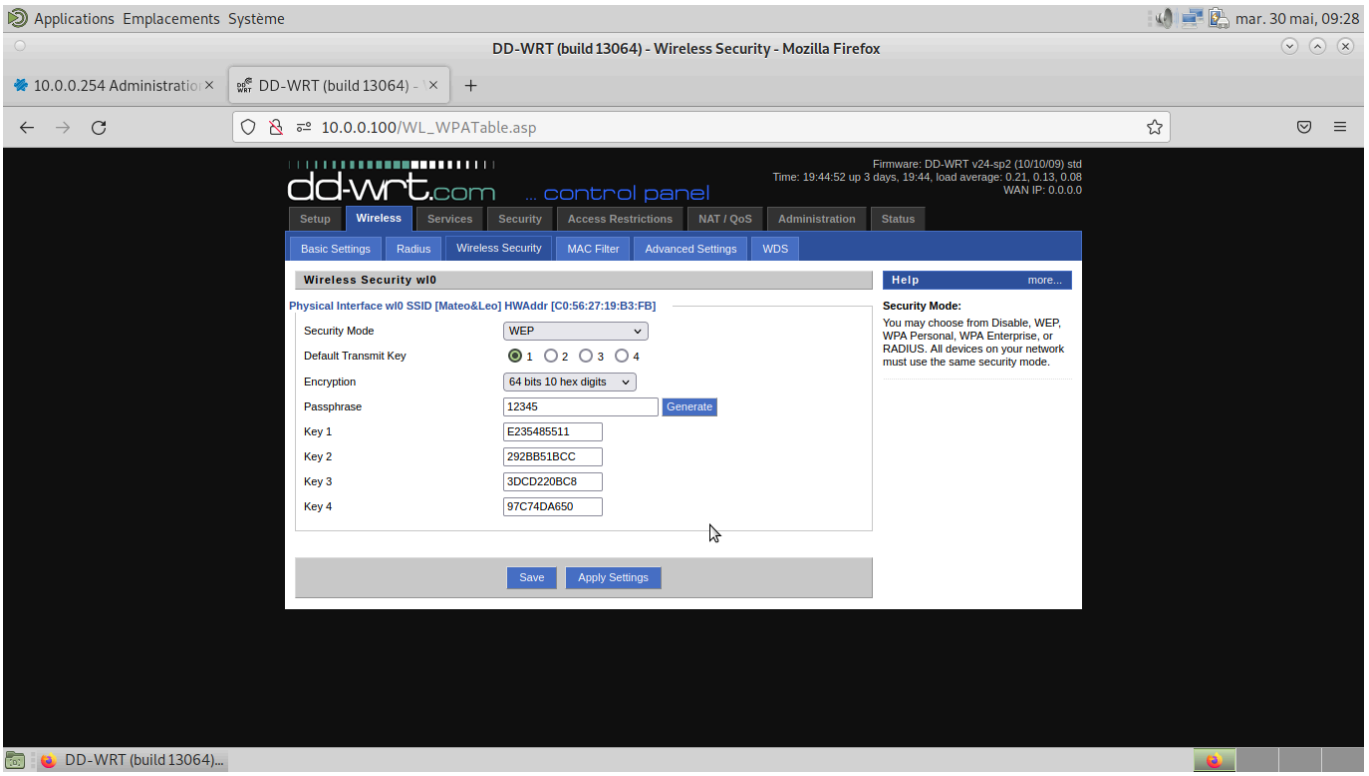
Objectif : Mettre en place des attaques sur le WEP et sur le WPA avec une Linksys puis avec un SNS

Sous-tâches	Evaluation prof
Mise en place du WEP sur Linksys -> ok	100%
Cassage de la clé WEP sur Linksys -> ok	100%
Mise en place du WPA sur Linksys -> ok	100%
Cassage du WPA sur Linksys -> ok	100%
Mise en place du WEP sur Stormshield SNS -> ok	100%
Cassage de la clé WEP sur Stormshield SNS -> ok	100%
Mise en place du WPA sur Stormshield SNS -> ok	100%
Cassage du WPA sur Stormshield SNS -> ok	100%

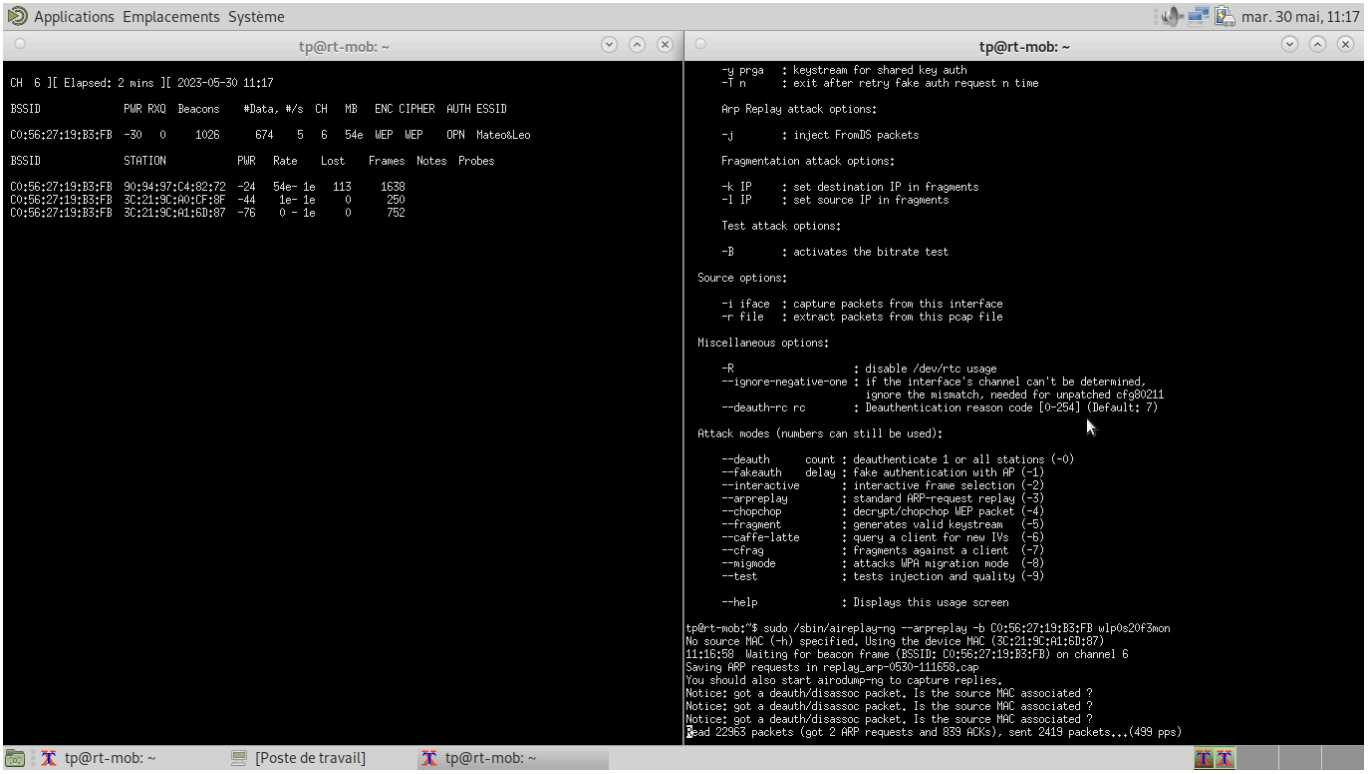
Rapport

(Expliquez votre démarche, le fonctionnement de WEP et de WPA, le principe mis en place par le cracker, etc.)

Le Wired Equivalent Privacy (WEP) utilise un système de chiffrement symétrique obsolète, ce qui permet via une longue (en fonction du trafic sur le LAN) écoute l'interception de paquets transitants sur le réseau afin de tenter d'effectuer des collisions et d'ainsi récupérer caractère par caractère la clé de chiffrement.
NB : Environ 10 000 paquets IVs pour une clé 64bits et environ 40 000 pour 128bits



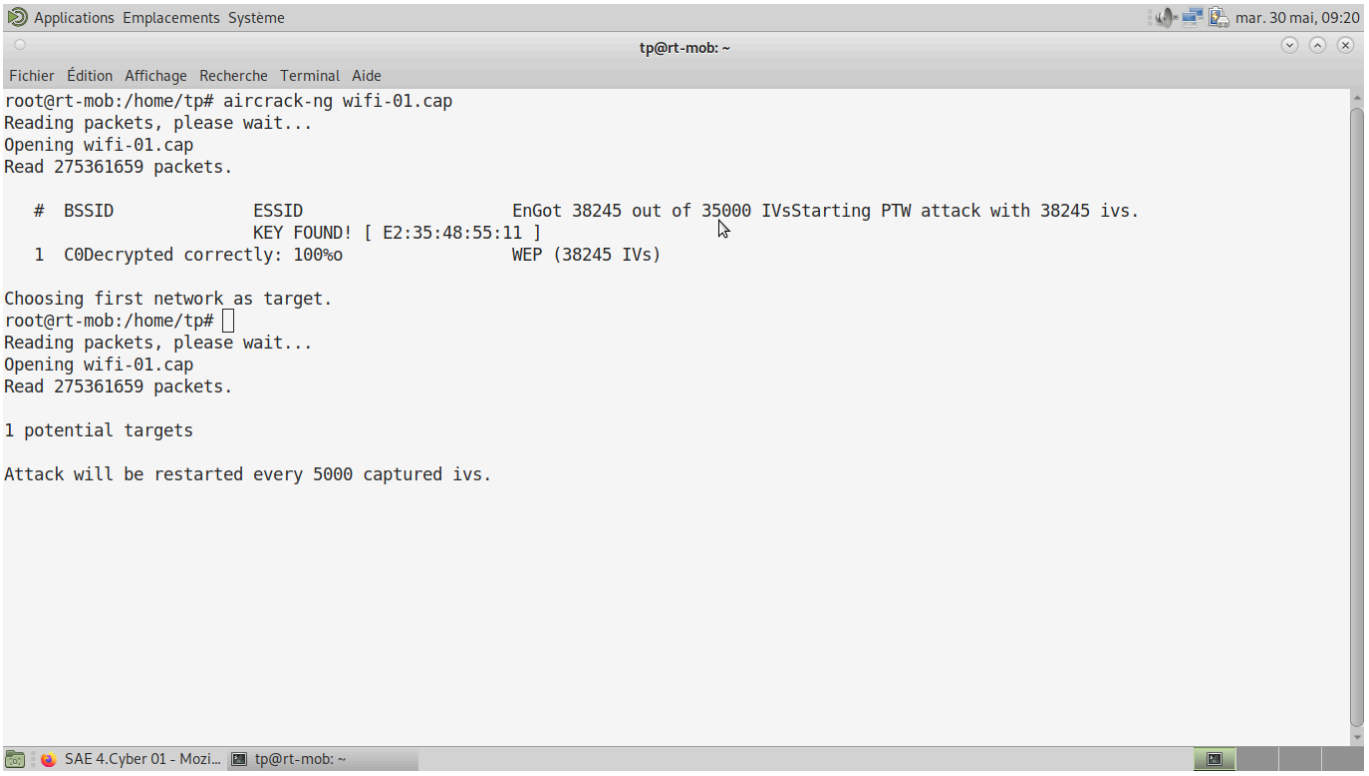
Mise en place du WEP sur la borne LinkSys avec utilisation de la première clée



Sniffing du réseau avec "airodump-ng" et génération de trafic avec "aireplay-ng" avec la méthode "arpreplay-ng".

Cette dernière duplique le trafic ARP émit légitimement par des clients afin de gonfler le volume de données transitant sur la bande passante.

Commande pour le sniffing : airodump-ng –bssid [MAC-AP] -c [CHANNEL-AP] wlp3s0f20mon

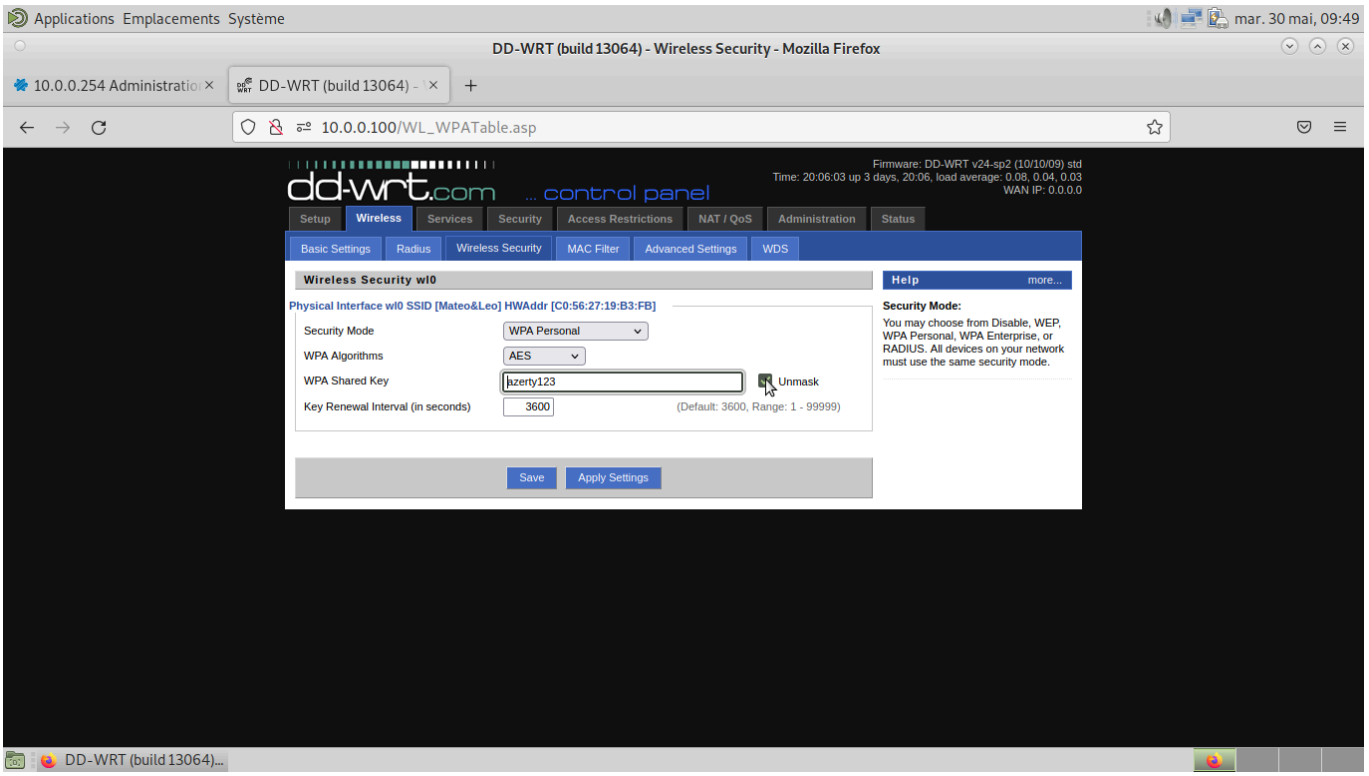


Après avoir obtenu suffisamment de Data on lance aircrack-ng avec notre fichier de capture. Aircrack-ng va donc chercher à provoquer des collisions dans les paquets Data pour récupérer la clé de chiffrement.
Obtention de la clée, ici : E235485511

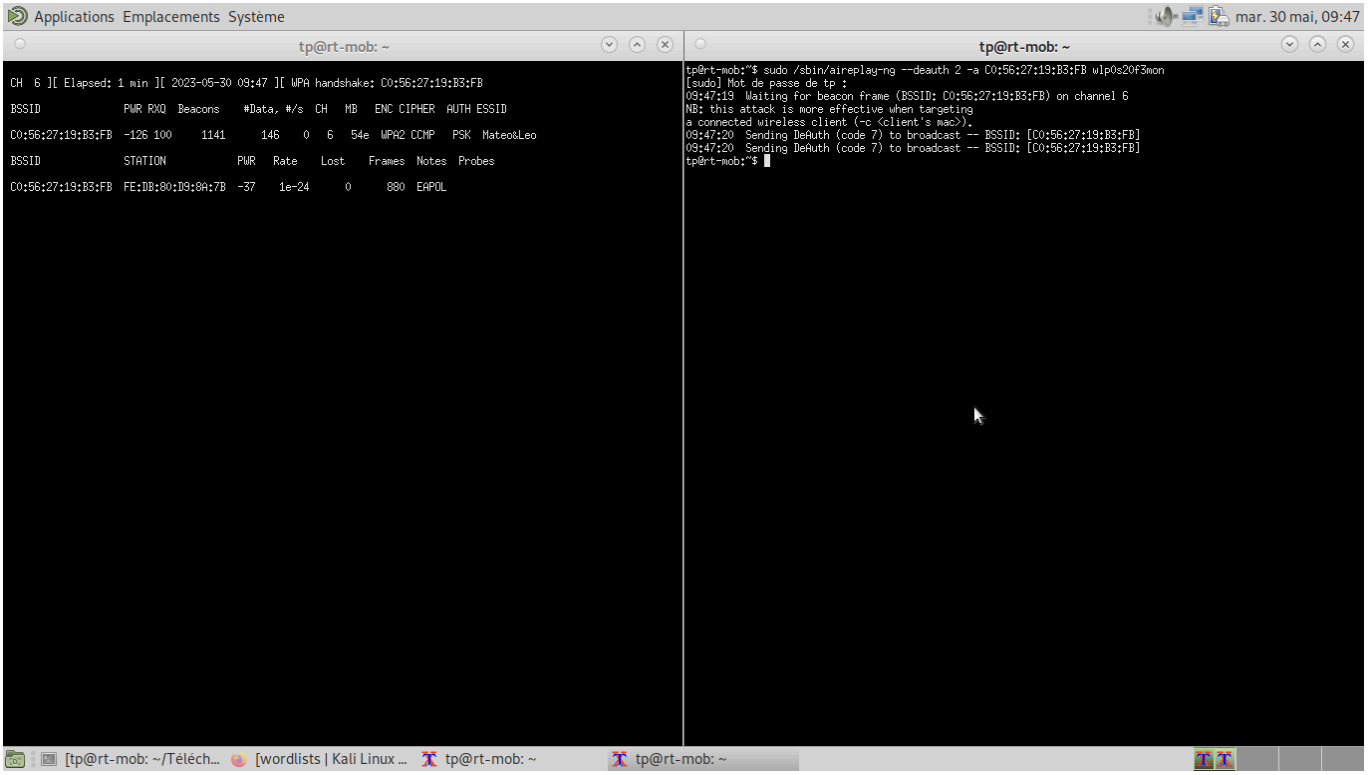
Ayant rencontré des problèmes de connexion de la part des clients sur l'Access Point Wi-Fi lorsqu'il utilisait le protocole WEP nous n'avions que d'autre choix d'attendre…
Nous l'avons donc laissé tourner tout le week-end, ce qui explique les environ 40 000 IVs !
Néanmoins cela nous a facilité la tâche derrière lors du "crackage" à proprement parler.

Wi-fi Protected Access (WPA) utilise la méthode chiffrement Temporal Key Integrity Protocol (TKIP) qui génère une clef "aléatoire" et temporaire pour chiffrer la connexion lors du handshake (échange d'informations initial). La clé TKIP étant distribuée aux clients dans le handshake, si nous l'interceptons nous pouvons "brute-forcer" le mot-de-passe en chiffrant chaque élément d'une wordlist avec cette même clef TKIP et en recherchant les collisions avec le mot-de-passe présent dans le handshake.

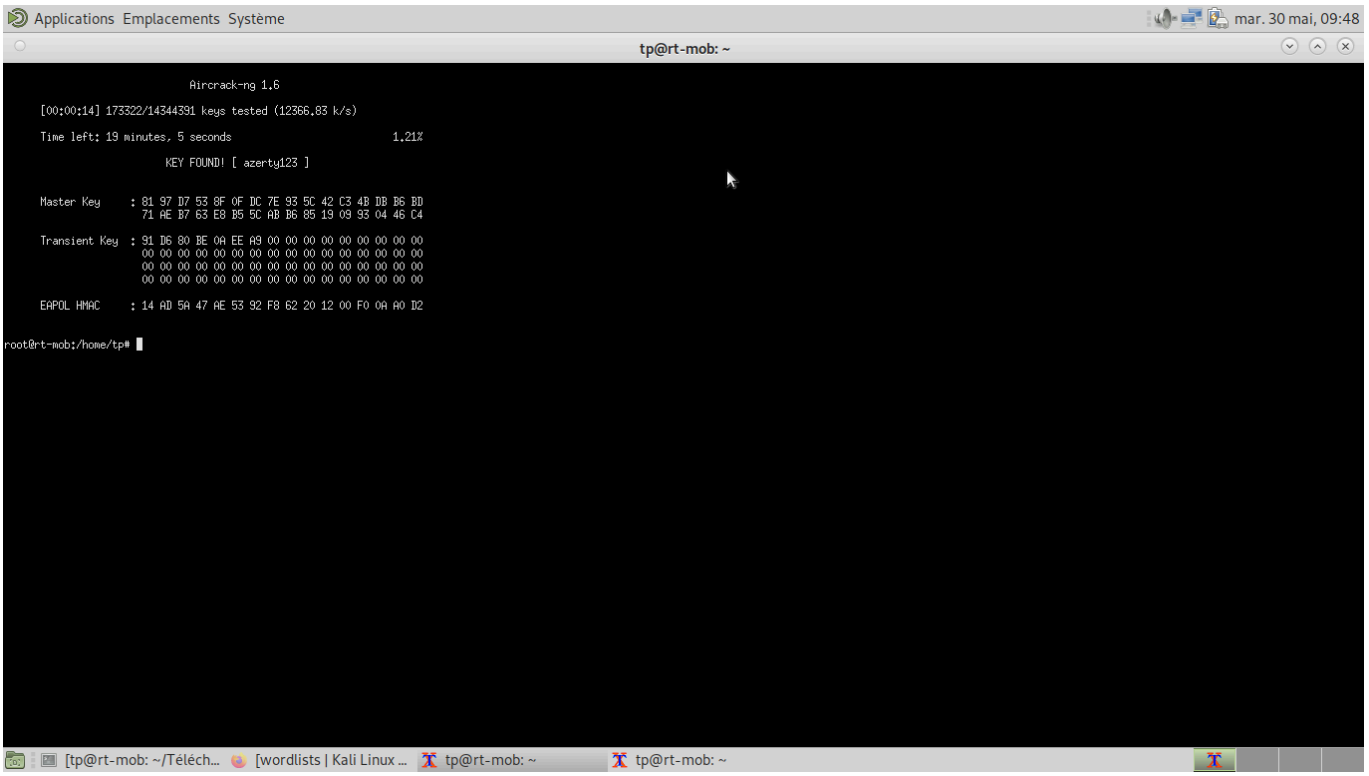
Ce principe nécessite néanmoins de pourvoir intercepter un échange de connexion entre un client et l'Access Point.



Mise en place du WPA sur la borne LinkSys avec "azerty123" comme mot-de-passe. Ce mot-de-passe est notamment présent dans la wordlist "rockyou", très célèbre.

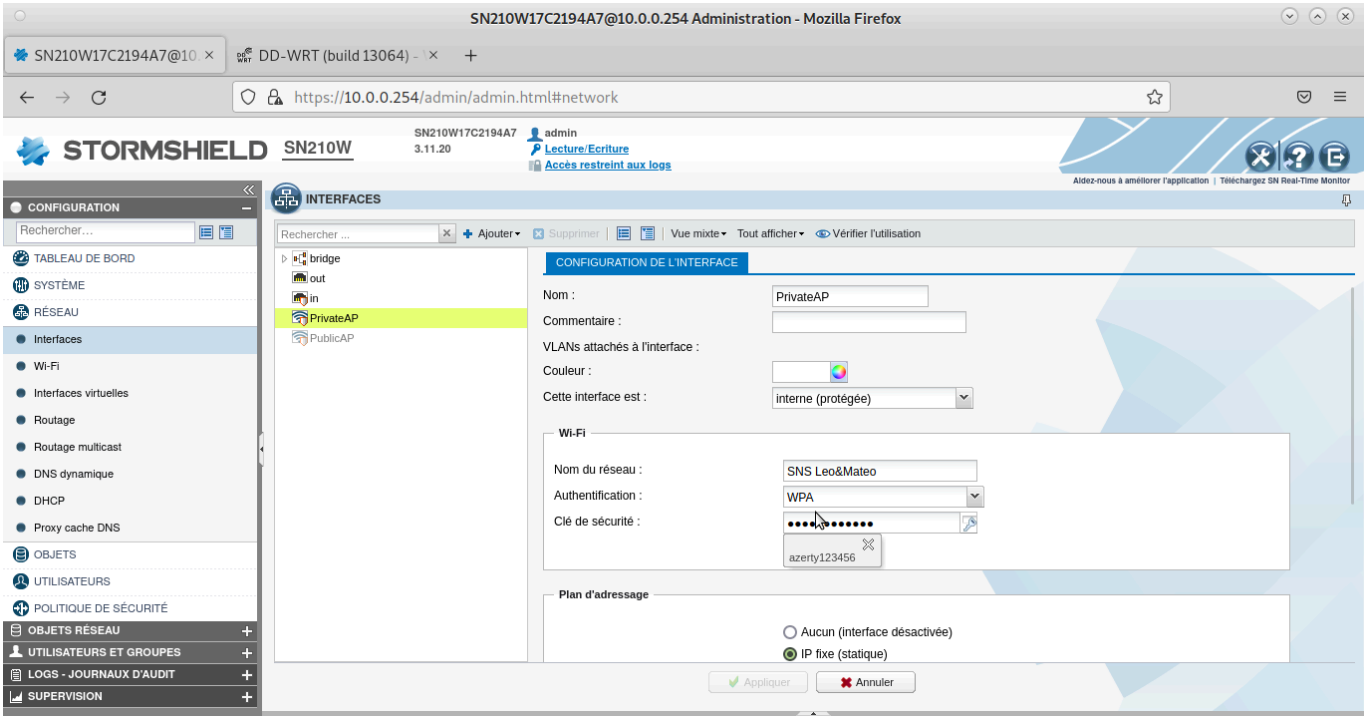


Sniffing du réseau avec "airodump-ng" et envoi de paquets de désauthentification via "aireplay-ng" avec la méthode deauth. Ces déconnexions forcent les clients à se reconnecter et nous pouvons ainsi intercepter le handshake utilisé lors de l'échange pour la connexion d'un client. Il peut être nécessaire de réitérer l'opération plusieurs fois.

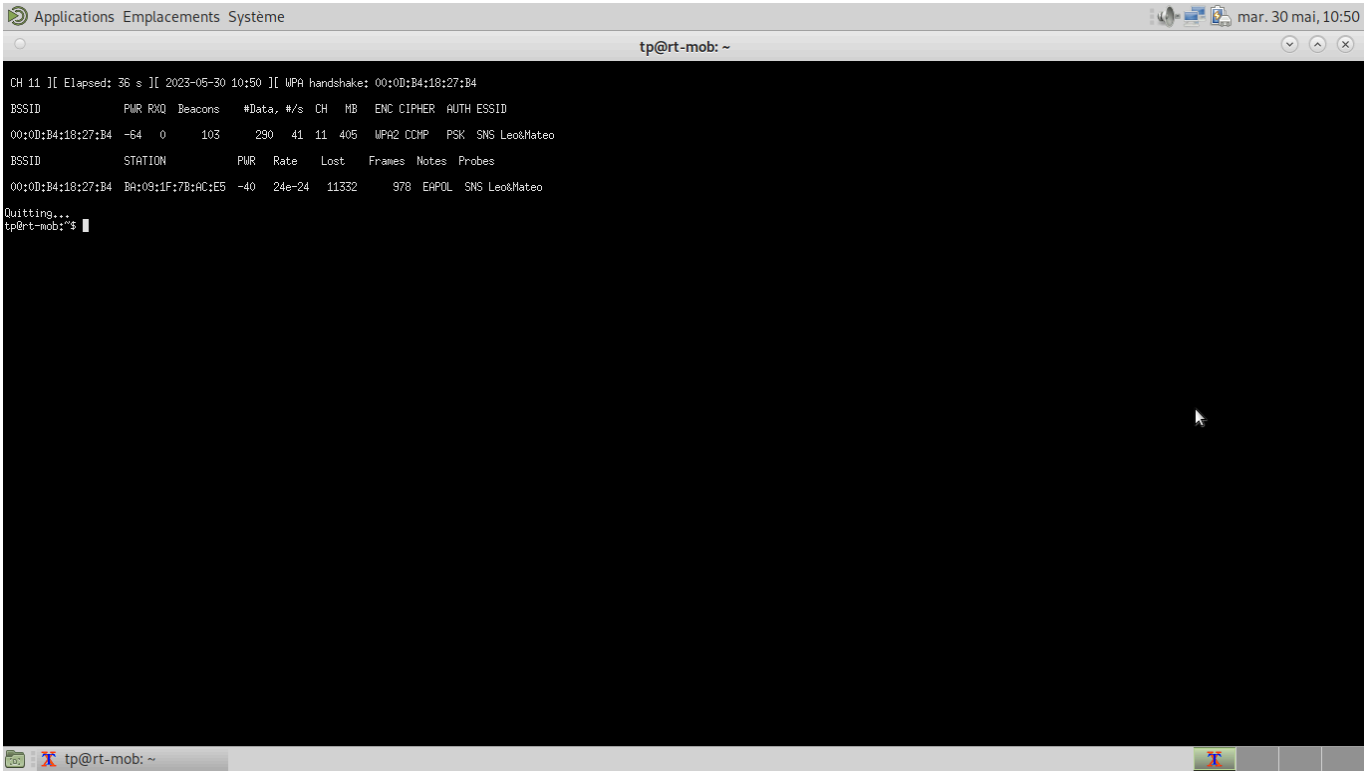


Utilisation de aircrack-ng et de la wordlist rockyou pour casser le mot-de-passe une fois le handshake récupéré dans les paquets de capture.

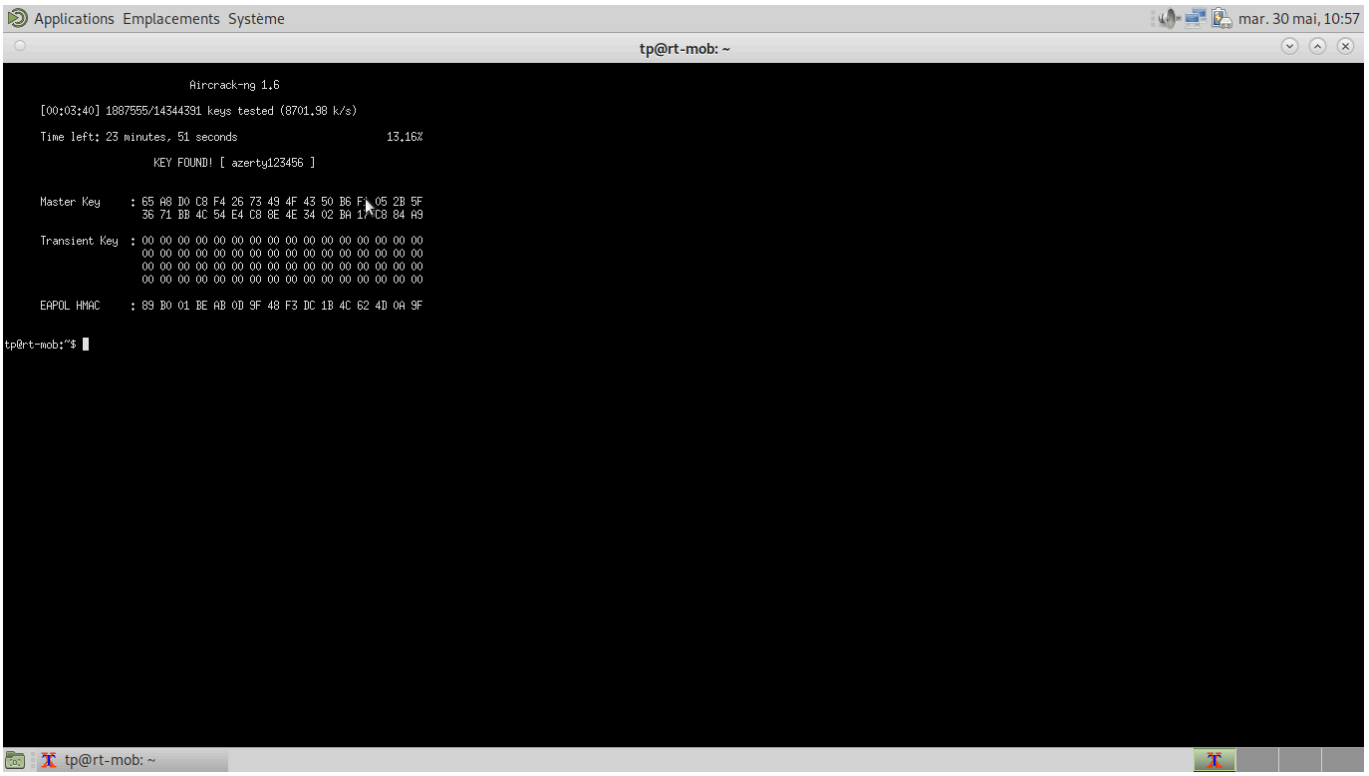
Commande pour retrouvé le mot-de-passe :
aircrack-ng [NOM].cap -w rockyou.txt -o password.txt



Mise en place du WPA sur le SNS avec le mot-de-passe "azerty123456"



Même dynamique pour le WPA sur la borne LinkSys, mais cette fois-ci : interception d'une connexion "légitime" d'un client -> obtention du handshake WPA sans envoi de paquets frauduleux



Utilisation de aircrack-ng et de la wordlist rockyou de la même manière que pour la borne LinkSys.

Tâche 7 Utilisation de scanneurs de vulnérabilité

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	10h/pers = 20 heures-hommes

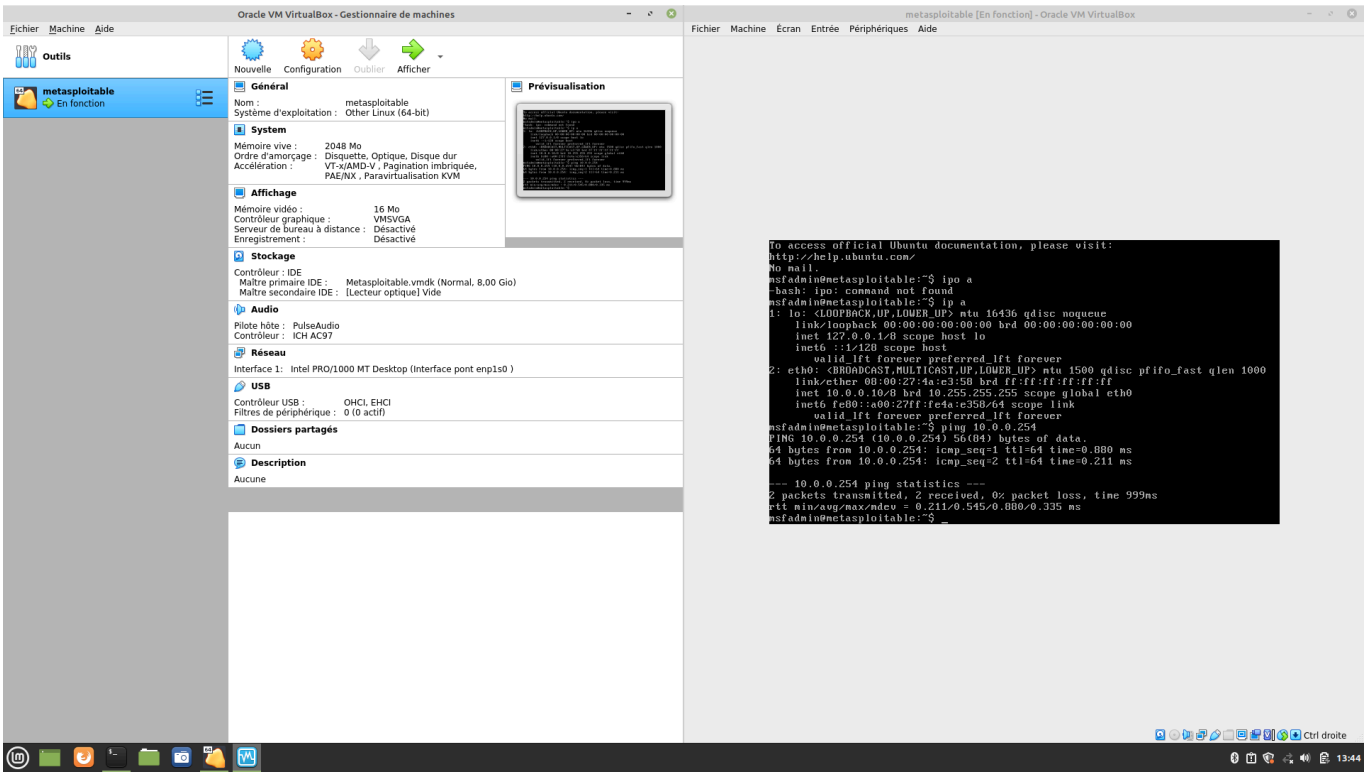
Estimation du temps passé sur cette tâche en heure-homme : 12heures-homme

Objectif : Réaliser plusieurs évaluations de la sécurité des serveurs

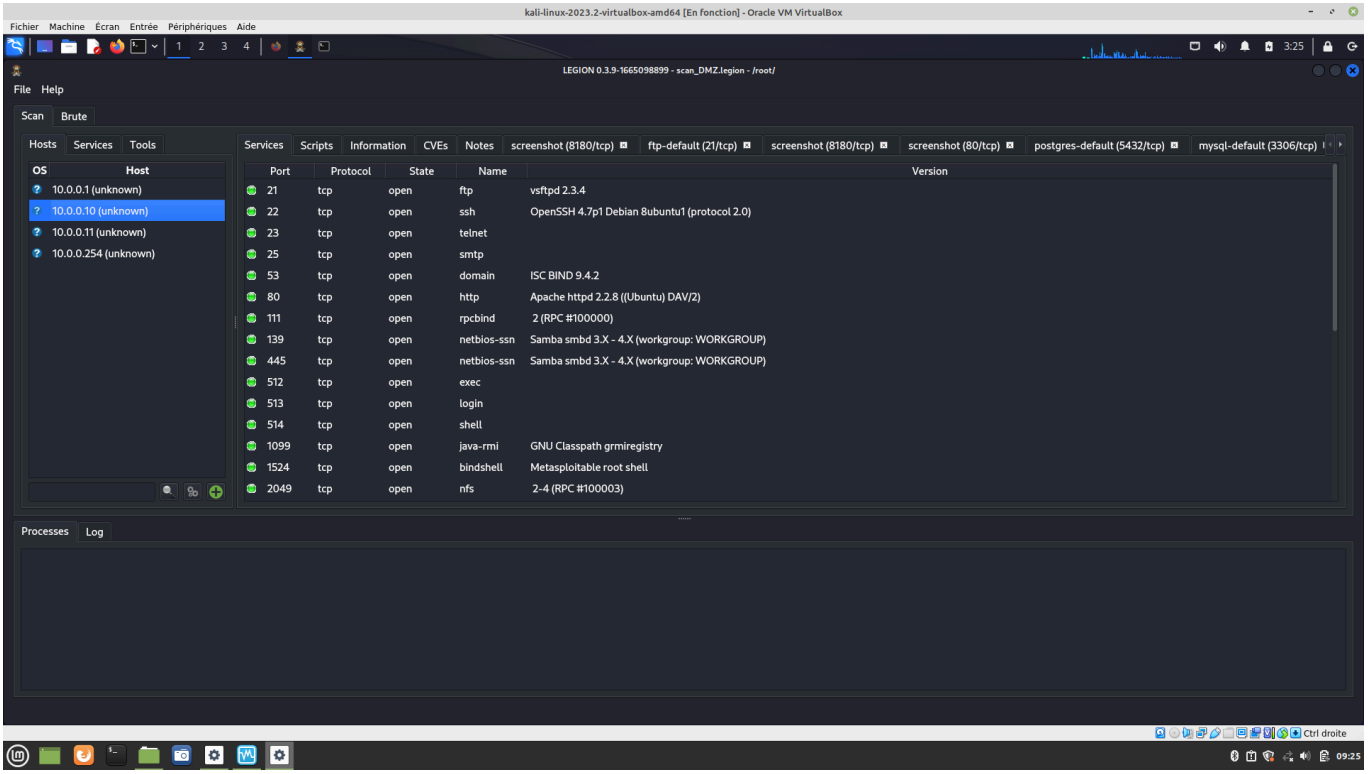
Sous-tâches	Evaluation prof
Installez dans la DMZ une machine/VM metasploitable -> ok	100%
Faites une évaluation de la sécurité avec le SNS Stormshield -> NOT POSSIBLE	
Installez et utilisez SCNR -> ok	100%
Installez et utilisez Legion -> ok	100%
Installez et utilisez Nuclei -> ok	100%
Installez et utilisez Nikto -> ok	100%
Placez les scanners dans la DMZ, puis à l'extérieur -> ok	100%

Rapport

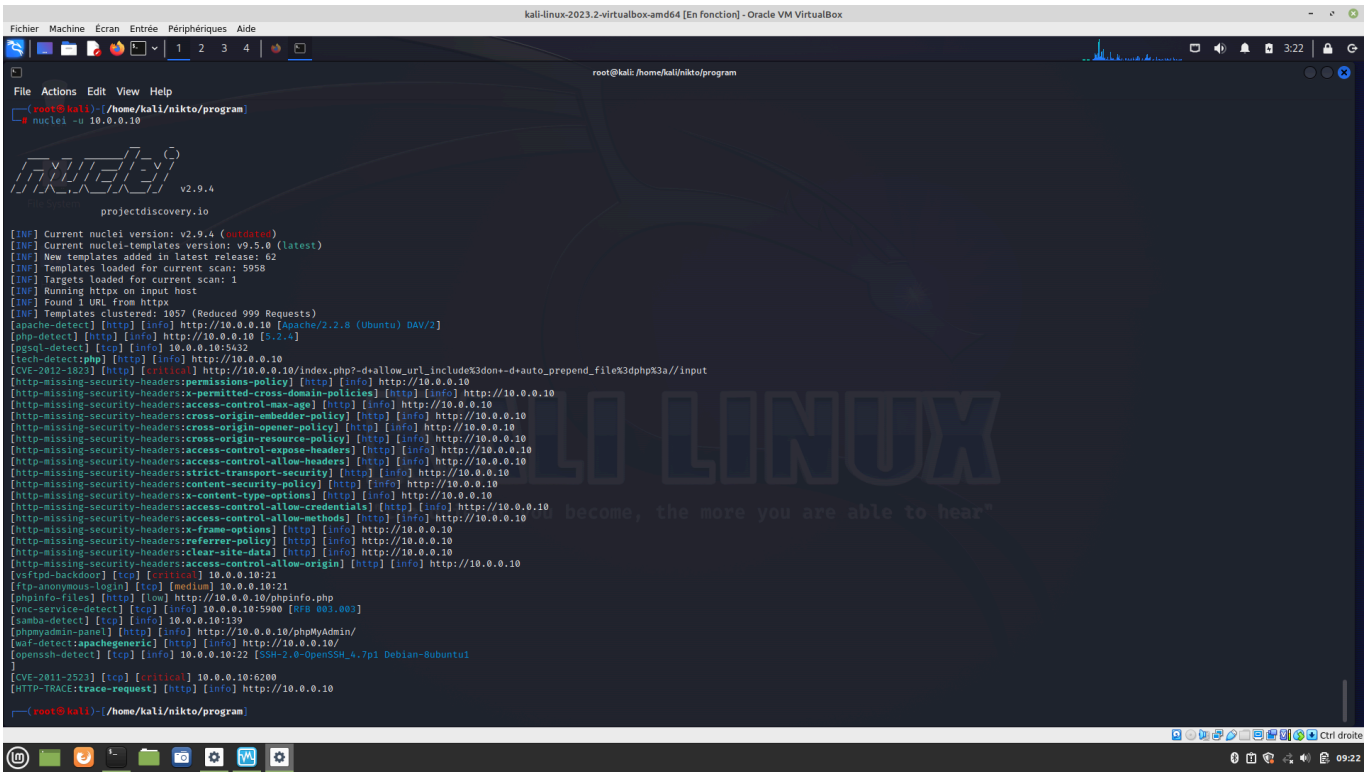
(Expliquez votre démarche, captures d'écrans des installations, listez le résultat des scans, etc.)



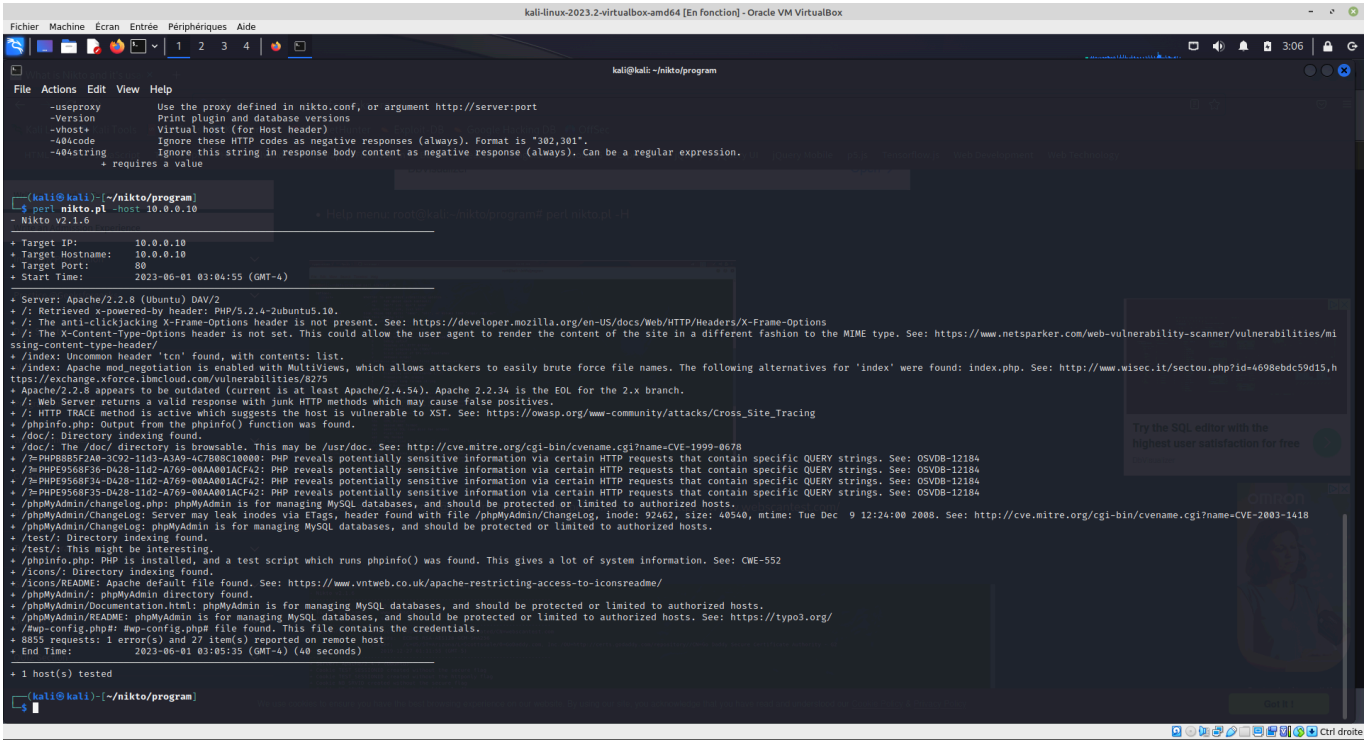
Capture montrant l'installation de la machine "metasploitable" et de virtualbox dans le serveur du Site B.



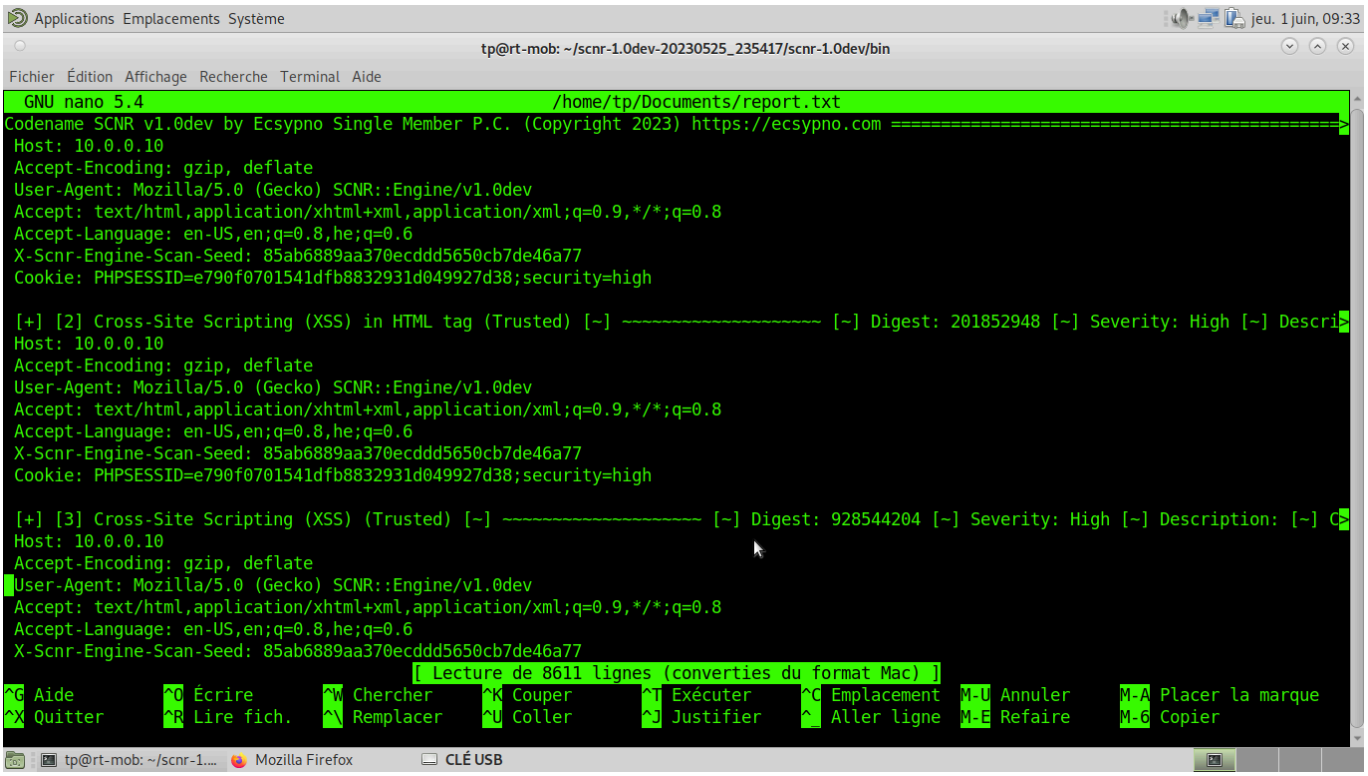
Capture du scanner Légion dans la DMZ du Site B.



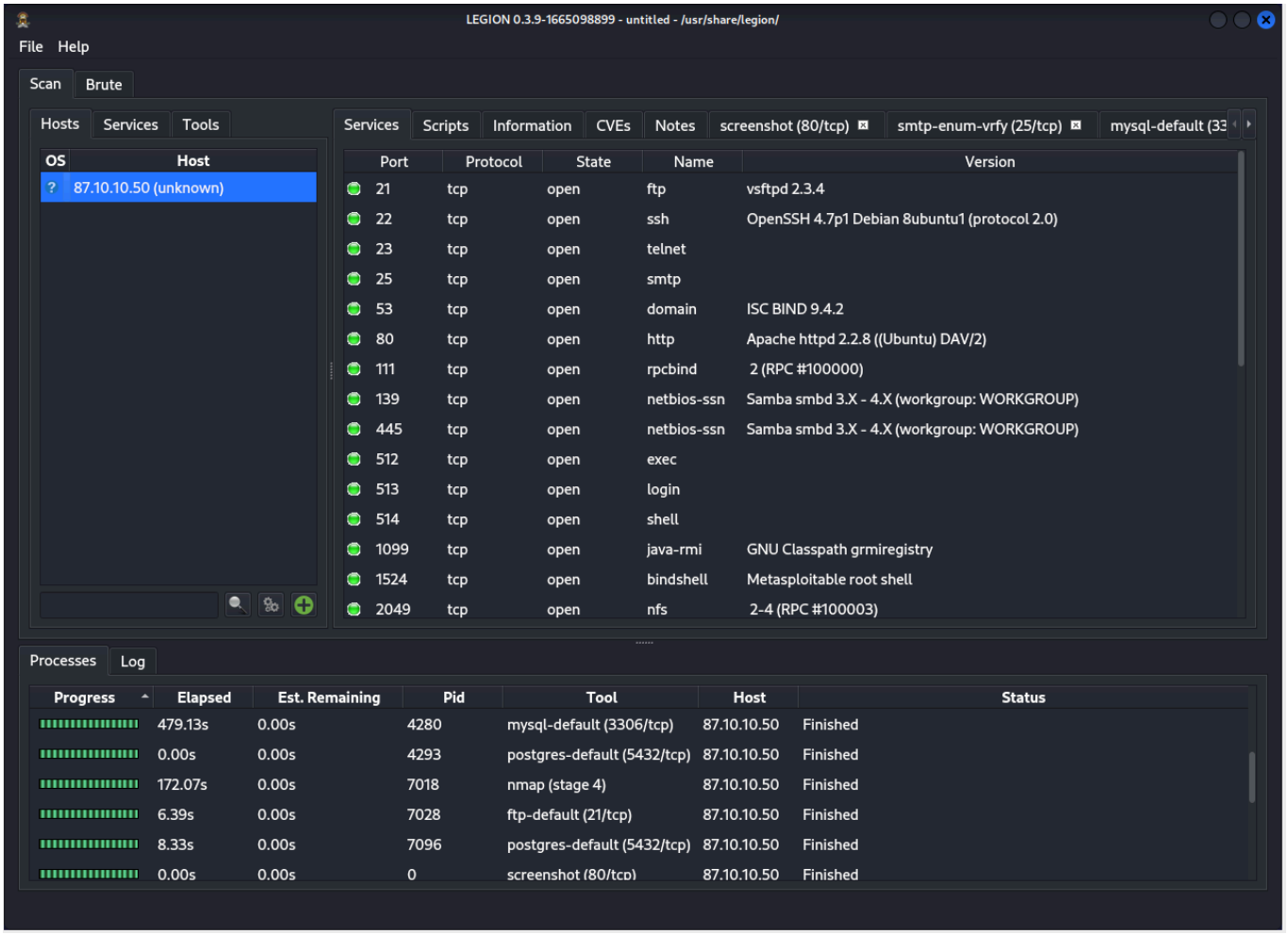
Capture du scanner Nuclei dans la DMZ du Site B.



Capture du scanner Nikto dans la DMZ du Site B.



Capture du scanner SCNR dans la DMZ du Site B



Capture scanner Legion dans "internet"

```
┌──(kali㉿kali)-[~]
└─$ cat ~/nuclei.txt
[php-detect] [http] [info] http://87.10.10.50 [5.2.4]
[apache-detect] [http] [info] http://87.10.10.50 [Apache/2.2.8 (Ubuntu) DAV/2]
[waf-detect:apachegeneric] [http] [info] http://87.10.10.50/
[tech-detect:php] [http] [info] http://87.10.10.50
[phpmyadmin-panel] [http] [info] http://87.10.10.50/phpMyAdmin/
[vsftpd-backdoor] [tcp] [critical] 87.10.10.50:21
[pgsql-detect] [tcp] [info] 87.10.10.50:5432
[openssh-detect] [tcp] [info] 87.10.10.50:22 [SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
]
[http-missing-security-headers:clear-site-data] [http] [info] http://87.10.10.50
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://87.10.10.50
[http-missing-security-headers:access-control-allow-credentials] [http] [info] http://87.10.10.50
[http-missing-security-headers:access-control-expose-headers] [http] [info] http://87.10.10.50
[http-missing-security-headers:access-control-max-age] [http] [info] http://87.10.10.50
[http-missing-security-headers:access-control-allow-headers] [http] [info] http://87.10.10.50
[http-missing-security-headers:content-security-policy] [http] [info] http://87.10.10.50
[http-missing-security-headers:x-frame-options] [http] [info] http://87.10.10.50
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://87.10.10.50
[http-missing-security-headers:strict-transport-security] [http] [info] http://87.10.10.50
[http-missing-security-headers:permissions-policy] [http] [info] http://87.10.10.50
[http-missing-security-headers:x-content-type-options] [http] [info] http://87.10.10.50
[http-missing-security-headers:access-control-allow-origin] [http] [info] http://87.10.10.50
[http-missing-security-headers:access-control-allow-methods] [http] [info] http://87.10.10.50
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://87.10.10.50
[http-missing-security-headers:referrer-policy] [http] [info] http://87.10.10.50
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://87.10.10.50
[vnc-service-detect] [tcp] [info] 87.10.10.50:5900 [RFB 003.003]
[CVE-2012-1823] [http] [critical] http://87.10.10.50/index.php?-d+allow_url_include%3don+-d+auto_prepend_file%3dphp%3a//input
[samba-detect] [tcp] [info] 87.10.10.50:139
[phpinfo-files] [http] [low] http://87.10.10.50/phpinfo.php
[CVE-2020-1938] [tcp] [critical] 87.10.10.50:8009
[HTTP-TRACE:trace-request] [http] [info] http://87.10.10.50
[CVE-2011-2523] [tcp] [critical] 87.10.10.50:6200
[ftp-anonymous-login] [tcp] [medium] 87.10.10.50:21
```

Capture scanner Nuclei dans "internet"

```
rt@rt-mob16:~/nikto/program$ cat ~/nikto.txt
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          87.10.10.50
+ Target Hostname:    87.10.10.50
+ Target Port:        80
+ Start Time:         2023-06-01 09:10:45 (GMT2)
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: index.php
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ SCAN TERMINATED:  20 error(s) and 7 item(s) reported on remote host
+ End Time:           2023-06-01 09:10:51 (GMT2) (6 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
rt@rt-mob16:~/nikto/program$
```

Capture scanner Nikto dans "internet"

```
[~] 648 issues have been detected.

[+]   1 | Publicly writable directory at http://87.10.10.50/dav/SCNR_Engine-3931cf8b2b91d0da053d7484d4b7ada1 in server.
[+]   2 | File Inclusion at http://87.10.10.50/mutillidae/ in link input `page` using GET.
[+]   3 | File Inclusion at http://87.10.10.50/mutillidae/index.php in link input `page` using GET.
[+]   4 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/Scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `template` using GET.
[+]   5 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/Scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `template` u
sing GET.
[+]   6 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1/scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `template` using
GET.
[+]   7 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1/WebHome in link input `template` using GET.
[+]   8 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/ManagingWebs in link input `template` using GET.
[+]   9 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/ManagingWebs in link input `param1` using GET.
[+]  10 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWikiscnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1/WebHome in link input `template` using GET.
[+]  11 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiWeb in link input `param1` using GET.
[+]  12 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiWeb in link input `template` using GET.
[+]  13 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiAuthentication in link input `template` using GET.
[+]  14 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiAuthentication in link input `param1` using GET.
[+]  15 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiUpgradeNotes in link input `param1` using GET.
[+]  16 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiUpgradeNotes in link input `template` using GET.
[+]  17 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiInstallationNotes in link input `template` using GET.
[+]  18 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiInstallationNotes in link input `param1` using GET.
[+]  19 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiAdministration in link input `template` using GET.
[+]  20 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiAdministration in link input `param1` using GET.
[+]  21 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/RenameTopic in link input `template` using GET.
[+]  22 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/RenameTopic in link input `param1` using GET.
[+]  23 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiNotificationOfChanges in link input `param1` using GET.
[+]  24 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiNotificationOfChanges in link input `template` using GET.
[+]  25 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiTemplatingSystem in link input `param1` using GET.
[+]  26 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiTemplatingSystem in link input `template` using GET.
[+]  27 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/HandlingTopics in link input `param1` using GET.
[+]  28 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/HandlingTopics in link input `template` using GET.
[+]  29 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/WebChangesNotify in link input `param1` using GET.
[+]  30 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/WebChangesNotify in link input `template` using GET.
[+]  31 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiFormTemplate in link input `param1` using GET.
[+]  32 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiFormTemplate in link input `template` using GET.
[+]  33 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiTemplateSystem in link input `param1` using GET.
[+]  34 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiTemplateSystem in link input `template` using GET.
[+]  35 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/DocsATWikiFileSystem in link input `param1` using GET.
[+]  36 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/DocsATWikiFileSystem in link input `template` using GET.
[+]  37 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiVariablesExamples in link input `param1` using GET.
[+]  38 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiVariablesExamples in link input `template` using GET.
[+]  39 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/MonitoringSiteActivity in link input `template` using GET.
[+]  40 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/MonitoringSiteActivity in link input `param1` using GET.
[+]  41 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiPages in link input `param1` using GET.
[+]  42 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/TWiki/TWikiPages in link input `template` using GET.
[+]  43 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/Scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `template` us
ing GET.
[+]  44 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/Scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `param1` usin
g GET.
[+]  45 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `param1` usin
g GET.
[+]  46 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `template` us
ing GET.
[+]  47 | Cross-Site Scripting (XSS) in script context at http://87.10.10.50/twiki/bin/oops/Main/Scnr_engine_sink_tracer_3931cf8b2b91d0da053d7484d4b7ada1 in link input `param1` using GET.
```

Capture scanner SCNR dans "internet"

La machine metasploitable a été configurée en Bridge sur l'interface reliée à la DMZ du Serveur Web/FTP/SSH présent sur le Site B.

Nous avons fait le choix d'utiliser SCNR/Nikto/Légion au lieu des scanners proposés car nous avons rencontré des difficultés d'installation (paquet linux trop ancien + miroir installation plus à jour).
Nous ne pouvions pas effectuer de "apt-get upgrade" afin de rafraîchir le contenu des dépôts puisque l'un de ces dépôts à été fermé il y a plus d'un an…
Nous n'avons pas trouvé la manière pour ne pas le contacter lors de l'exécution. D'où notre choix différent des propositions.

Lors des scans, nous avons remarqué des différences entre un scan interne au LAN et un autre effectué depuis "internet". Ces dernières sont expliquées par le filtrage présent sur le firewall. Cela permet de ne pas pouvoir scanner tout le réseau interne depuis le réseau externe. Du point de vue du hacker, cela rend les attaques plus difficiles depuis le réseau externe, la première phase étant toujours l'énumération, ce pare-feu ralenti les recherches et donc la rapidité de l'attaque..

SCNR :

SCNR nous à particulièrement posé soucis lors de son installation. Rien de bien clair ni constructif pour ce rapport…

Son utilisation à elle été ultra-performante, que ce soit depuis "internet" ou bien le LAN. Son plus gros défaut est aussi son principal atout : la rigueur et la curiosité. Nous n'avons pas indiqué de test particulier à effectuer en appelant SCNR comme suivant : ./sncr http://[MSFable-NAT-IP]

SCNR a donc scanné chaque pages, découvertent par une énumération complètes des pages du sites et des ses sous-domaines, avec chaque méthode qu'il connaissait. Le scan a duré au total plus de 20h puisque la machine est volontairement sur-vérolée.

Nikto :

Nikto ne nous a posé aucun souci, que ce soit pour l'installation ou l'utilisation de l'outil.

Effectivement son usage est simplicime : perl nikto.pl -host [MSFable-NAT-IP]

Nikto se concentre sur le scan WEB et en particulier les XSS (Cross-Site Scripting). Il nous recense ici quelques failles trouvées sur la machine cible. L'outil permet un argument de niveau de scan d'effectuer un scan plus ou moins approfondi et donc d'obtenir plus ou moins de résultats.

Nuclei :

Tout comme Nikto, Nuclei n'a pas été problématique du tout ! Son utilisation est tout aussi simpliste :

nuclei -u [MSFable-NAT-IP]

Nuclei à un champ de scan extrêmement large en relevant les failles d'une multitudes de services et protocoles. Avec certes un accent prononcé sur l'analyse web.

Légion :

Légion été préalablement installé sur la distribution Kali Linux de Offensive Security, pas de soucis ni d'action à ce niveau là.

Son utilisation est très similaire à celle de "nmap", il reprend d'ailleurs l'affichage de "Zenmap" la version graphique de "nmap". Ses résultats sont nombreux et variés. Tout comme Nuclei il scanne différent services et protocoles.

Tâche 8 Réalisation d'une attaque MitM

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	0.5h/pers = 1 heures-hommes

Estimation du temps passé sur cette tâche en heure-homme : 2 heures-hommes

Objectif : Vol d'une connexion HTTP

Installez une machine dans votre DMZ en la branchant sur un switch. Faites une attaque par empoisonnement ARP pour usurper l'adresse ARP du serveur Web et affichez une page différente. Puis, faites une redirection de la connexion du client vers le vrai serveur. Le client ne s'apercevra plus qu'il y a le pirate entre lui et le serveur.

Bonus : Modifiez des données de la page HTML envoyée au client.

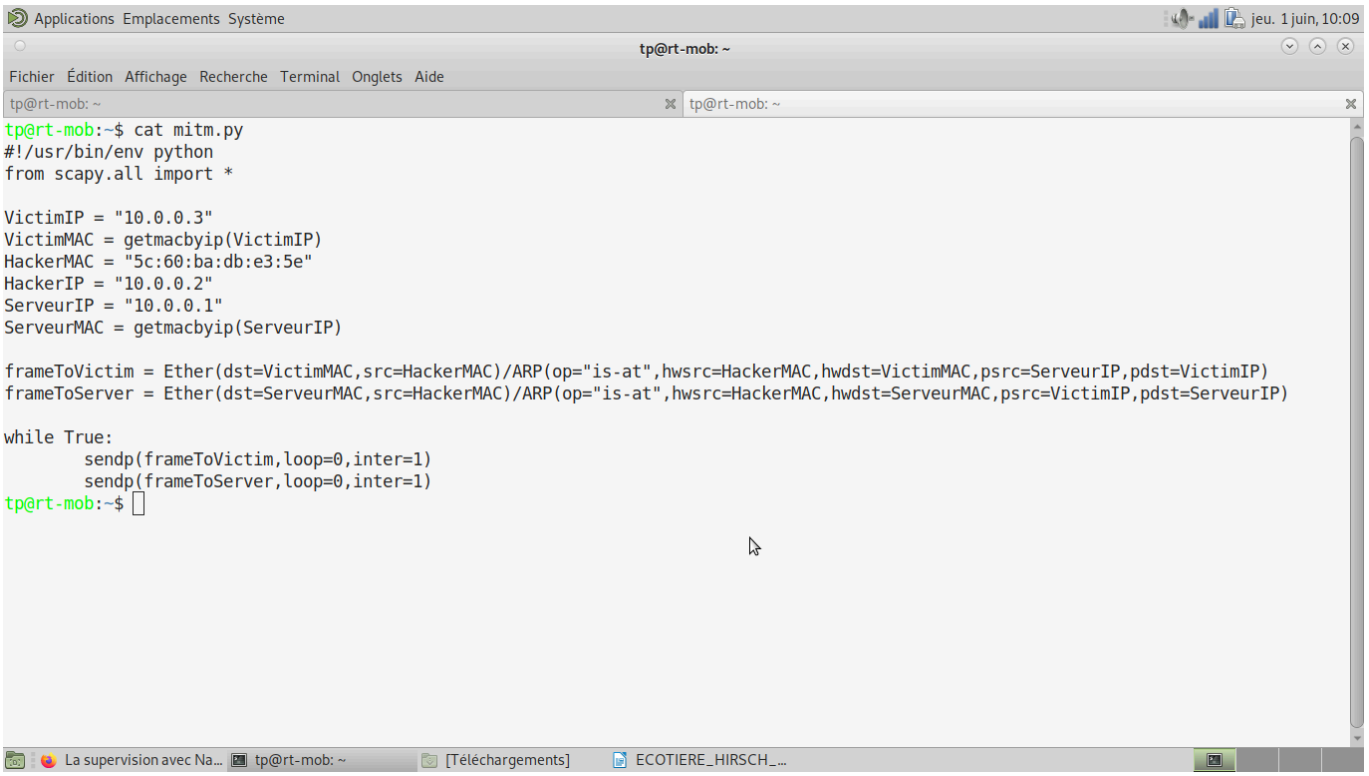
Vous devrez utiliser une application pouvant forger des paquets ARP comme Scapy ou Arp-sk par exemple.

Sous-tâches	Évaluation prof
Installez un forgeur de paquets ARP -> ok	100%
Usurpation de l'adresse ARP du serveur -> ok	100%
Redirection de la connexion -> ok	100%
Bonus : modification des données HTML -> nok	0%

Rapport

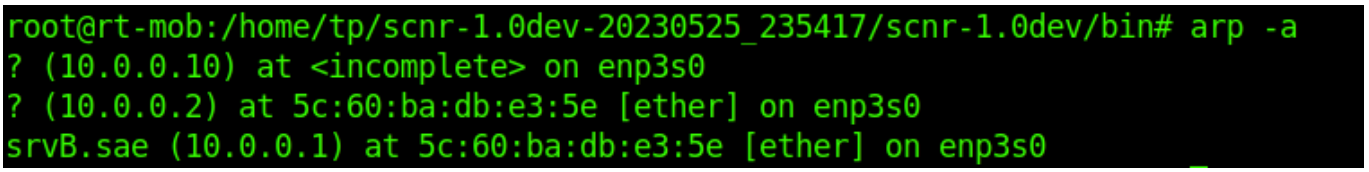
(Expliquez votre attaque, captures d'écrans des installations, de l'usurpation de la connexion et de sa redirection, code source ou commande de la modification de l'HTML, etc.)

Pour cette attaque, nous avons fait le choix d'utiliser un forgeur de paquet ARP comme Scapy. C'est un outil que nous maîtrisons pour ce type d'attaque (déjà utilisé en TP).



Capture du script python avec l'utilisation de scapy

Avec ce script, nous envoyons des paquets ARP de type "is-at" (aussi appelé "gratuitous ARP") en noyant le réseau de ces requêtes illégitimes. Cela a pour but de modifier la table arp du serveur et du client en précisant que pour joindre l'un ou l'autre, le trafic sera redirigé vers le hackeur (Man In The Middle).



Capture table ARP de la victime

Sur la capture ci-dessus, nous pouvons voir que l'adresse mac correspondant à l'adresse IP du serveur est la même que celle correspondant à l'adresse IP du hackeur. L'attaque a donc bien fonctionné.

Afin de rendre cela invisible auprès du client, nous avons activer l'IP Forward sur le hackeur. Par conséquent, lorsque le client veut se rendre sur la page WEB du serveur, il passe par le hackeur qui le redirige vers le serveur. En lançant un sniffer tel que "tcpdump" ou encore "wireshark" on peut intercepter tous les échanges. Néanmoins les services utilisant des chiffrements de bout-en-bout tels que SSH ou SFTP ne sont pas visibles en clair.

Tâche 9 Contre-mesures contre des attaques MitM

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2x4h=8 heures-hommes

Estimation du temps passé sur cette tâche en heure-homme : 4h

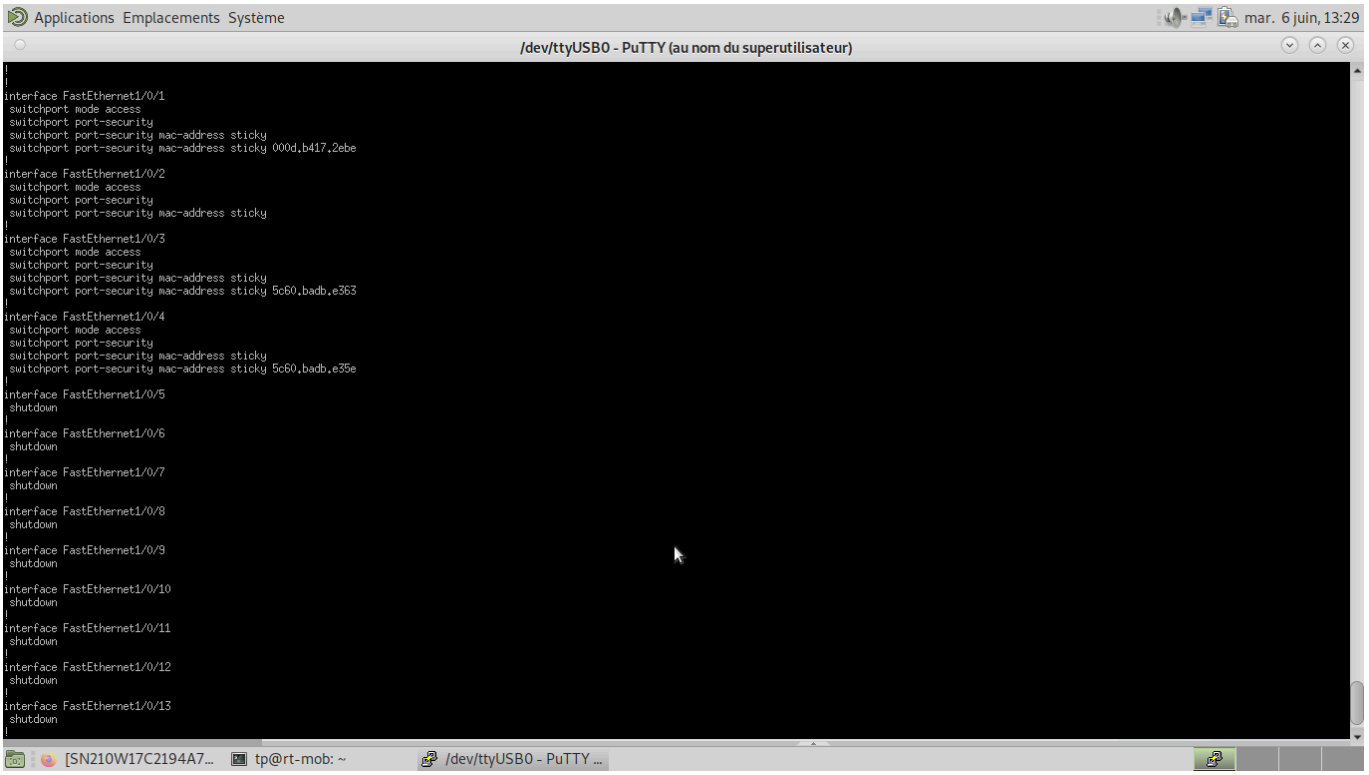
Objectif : Sécurisation de vos LAN contre le MiM

Proposez et mettez en place une ou plusieurs solutions permettant de détecter et de contrer des attaques MiM basées sur de l'usurpation ARP sur vos LANs et testez-les avec la tâche 9.
Pour la détection vous pouvez utiliser ARP Watch et la tâche 11. Pour se protéger des attaques, utilisez les fonctionnalités de votre commutateur.

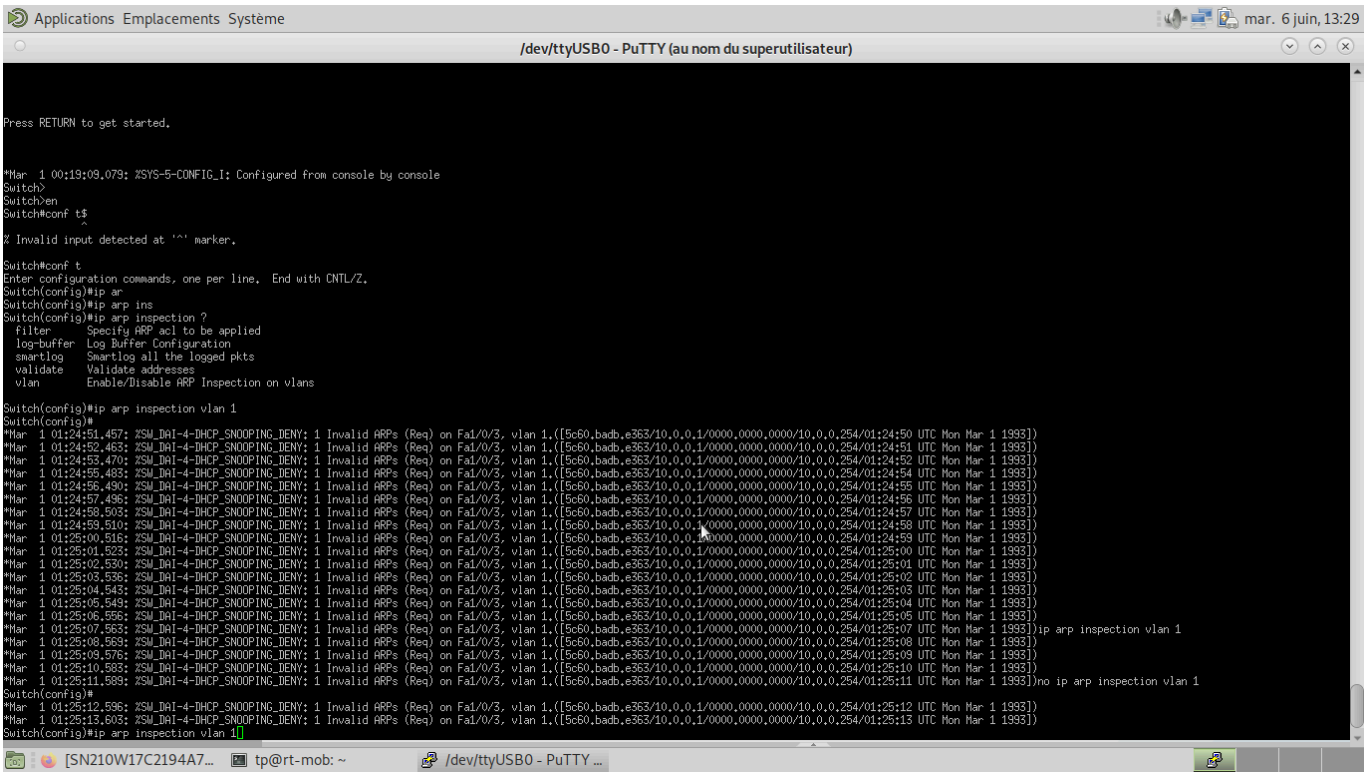
Sous-tâches	Evaluation prof
Description de la ou des solutions -> ok	100%
Mise en place des solutions de détection -> ok	100%
Mise en place de la protection -> ok	100%

Rapport

(Expliquez votre méthode, captures d'écrans des tests, etc.)



Afin d'empêcher tout intru de se connecter sur le commutateur il est nécessaire de fermer les ports non utilisés ("shutdown") et d'appliquer une sécurité sur ceux utilisés



Pour l'inspection la commande "ip arp inspection vlan 1" fonctionne à merveille en surveillant et recensant notamment les paquets "Gratuitous ARP"

Par définition un MitM est effectué depuis un PC appartenant au hacker, ou ayant une main mise dessus (auquel cas il s'agit d'un autre problème de sécurité…). Chaque carte réseau dispose d'une adresse MAC unique.
Nous configurons donc sur les interfaces utilisées par des clients/serveurs légitimes une politique de filtrage MAC. Cela consiste à apprendre automatiquement l'adresse au démarrage du commutateur si l'appareil est branché ("mac-address sticky") et à n'autoriser que cette adresse à émettre sur ce port. Toute violation entraîne une fermeture administrative du port avec ré-ouverture manuelle obligatoire.

Tâche 10 Supervision du réseau

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2x2.5h= 5 heure-homme

Estimation du temps passé sur cette tâche en heure-homme : 6h

Objectif : Mettre en place les outils de supervision de réseau Nagios et Cacti

Pour cette tâche, nous vous laissons une plus grande autonomie, à vous de nous proposer ce que vous pensez utile de monitorer dans votre réseau.
Nous vous donnons quand même quelques pistes par exemple, de monitorer toutes les machines et tous les services que vous avez installés, installer NCPA, les débits en entrée du firewall, générer des rapports, etc.

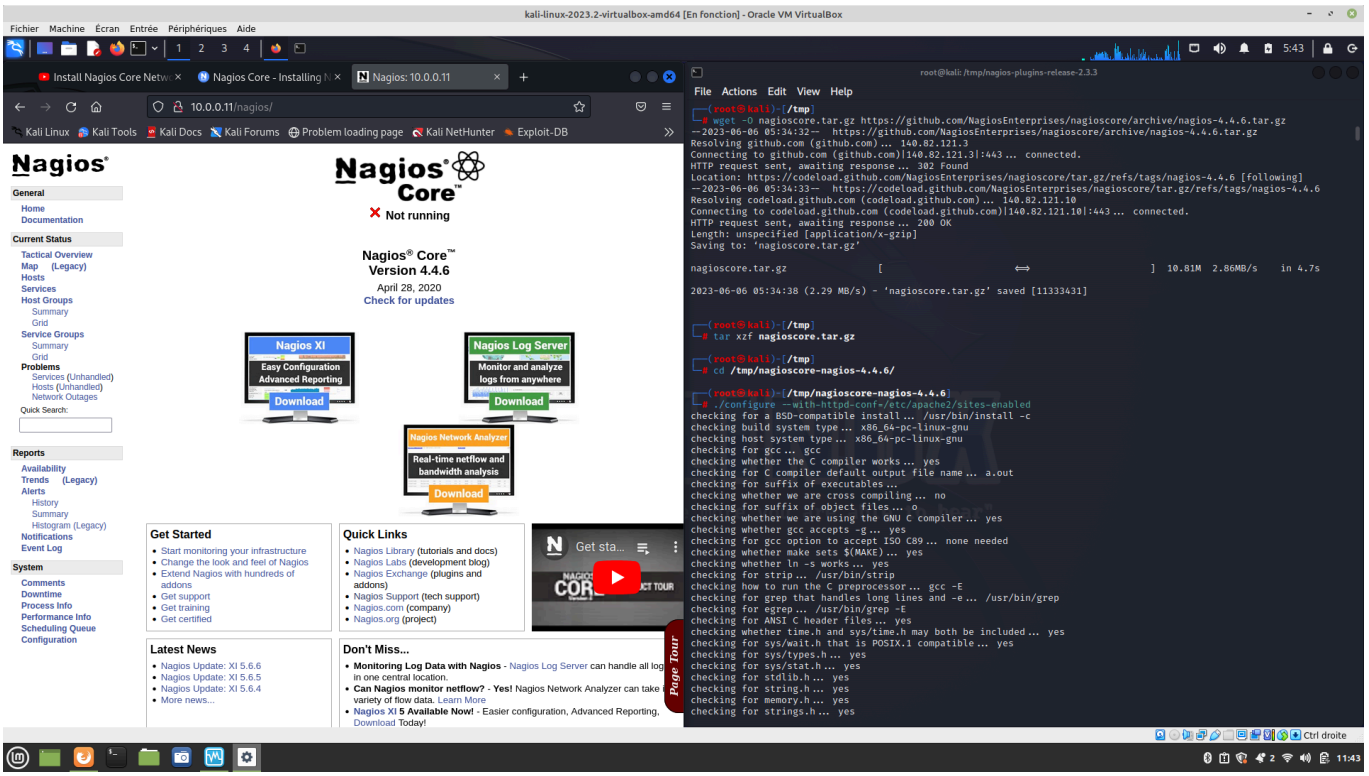
Sous-tâches	Evaluation prof
Installation et configuration -> ok	100%
Mise en place de la supervision -> ok	100%
Génération de rapports -> ok (Nagios)	100%

Rapport

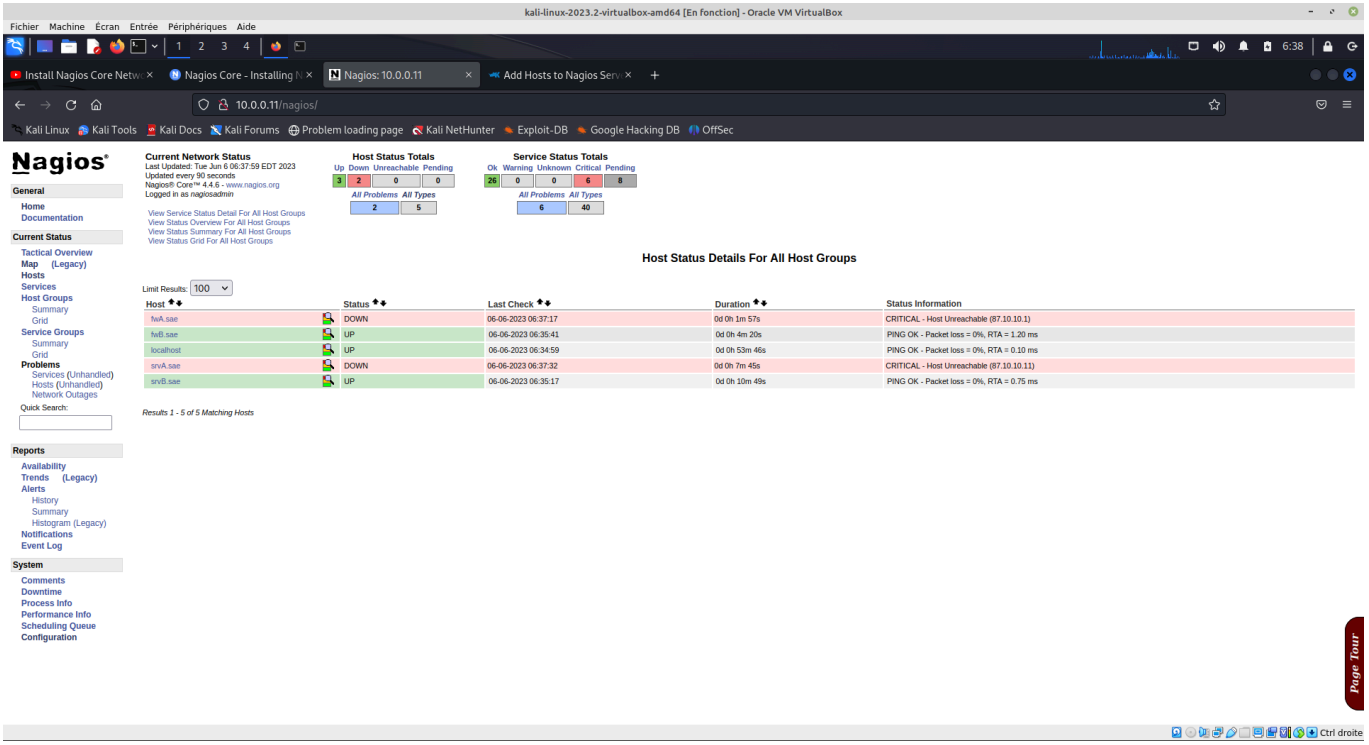
(Expliquez votre méthode, captures d'écrans des tests, etc.)

Pour cette partie de la SAE, la liberté de choix nous était donnée. Nous avons choisi d'installer Cacti et Nagios qui sont deux outils de supervision de réseau et fournissent des indicateurs en temps réel de l'infrastructure supervisée.

Capture des graphes d'analyse de Cacti.



Capture de l'installation de Nagios



Capture de la liste des hôtes sur Nagios

Host State Breakdowns:

Host	% Time Up	% Time Down	% Time Unreachable	% Time Undetermined
fwA.sae	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
fwB.sae	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
localhost	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
srvA.sae	0.000% (0.000%)	18.037% (100.000%)	0.000% (0.000%)	81.963%
srvB.sae	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
Average	0.000% (0.000%)	3.607% (20.000%)	0.000% (0.000%)	96.393%

Capture des rapports générés par Nagios.
(pas assez de données pour avoir des chiffres significatifs sur la capture)

Dans cette partie, nous avons rencontré plusieurs problèmes. Le principal étant que les miroirs debian présents sur nos PC n'étaient plus maintenus et par conséquent nous ne pouvions plus installer les derniers paquets à jour. Afin de pallier cela, nous avons installé et monté une VM Kali sur le serveur du Site B afin de pouvoir installer les différents services demandés.

Un autre problème est que nous avons rencontré des difficultés concernant la configuration de Cacti car il doit être lié à une base de données. Or cette partie nous a offert quelques complexités, Cacti ne voulait pas interroger la base de données qu'on lui avait renseignée auparavant, une modification d'un fichier de configuration php a réglé le problème.

D'un point de vue supervision, nous avons fait le choix de superviser les 2 serveurs, les 2 firewalls ainsi que l'IDS car ce sont les différents éléments principaux de notre infrastructure, en plus de cela l'IDS est un élément de sécurité critique. Nous avons également supervisé les différents services installés sur les machines afin de vérifier leur état de fonctionnement.

De plus, nous avons ajouté différentes données à superviser comme un ping régulier pour vérifier que la machine est toujours joignable, les fréquences de connexion des utilisateurs ou encore la gestion des ressources consommées.
En effet, lors d'une action non légitime d'une tierce personne, la machine visée pourrait voir sa consommation de ressources augmenter (ex: le DDoS).

Tâche 11 Mise en place d'une architecture Single Sign-On

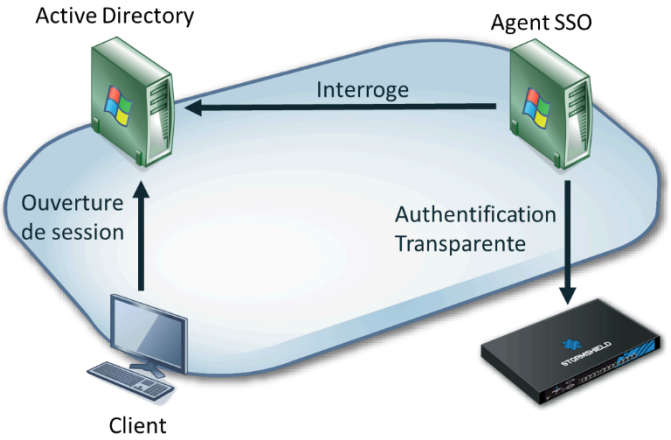
Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2x6h = 12 heure-homme

Estimation du temps passé sur cette tâche en heure-homme : 16h

Objectif : Permettre aux clients de passer le proxy sans authentification explicite

L'authentification par la méthode agent SSO permet d'authentifier les utilisateurs dès l'ouverture d'une session sur le domaine, elle se déroule en 3 étapes.

L'ouverture de session du client sur le domaine va générer un évènement d'authentification répliqué sur l'ensemble des contrôleurs de domaine Active Directory d'un même domaine. Ces évènements portent les ID 4624 ou 4768 sur les serveurs Windows 2008, 2012 et 2016.



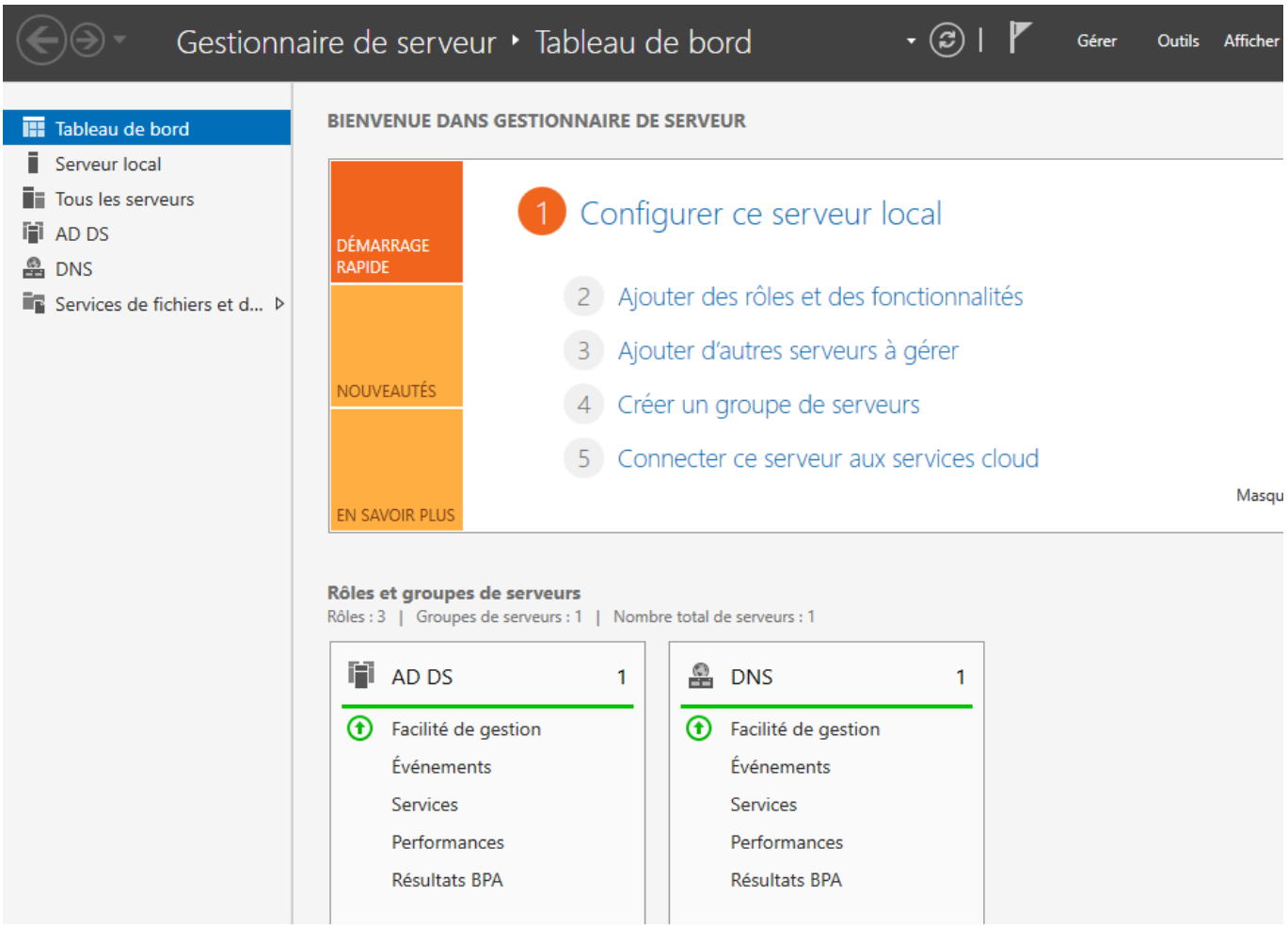
L'agent SSO va ensuite consulter les journaux d'évènements du contrôleur de domaine. Sur réception d'un nouvel événement, les informations liées à l'adresse IP et au nom du client sont transmises au firewall afin de les ajouter à la table des utilisateurs authentifiés.

Les échanges entre l'agent et le firewall utilisent le port 1301/TCP et sont chiffrés grâce au protocole SSL, algorithme PSK-AES256-CBC-SHA.

L'authentification doit être robuste au changement de l'adresse IP de la machine client.

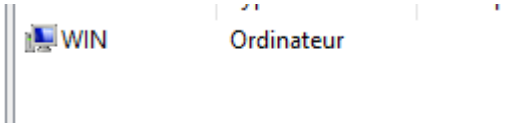
Sous-tâches	Evaluation prof
Installation d'un serveur Active Directory -> ok	100%
Installation d'un agent SSO sur une machine -> ok	100%
Configuration de la machine de client -> ok	100%
Changement de l'adresse IP de la machine -> ok	100%

Pour cette partie, nous avons le choix d'installer deux VM : 1 Windows serveur 2019 & 1 Windows 10 pro. Ce choix se justifie par le fait que beaucoup de groupes ont les mêmes plans IP et par conséquent un conflit d'adresses allait se mettre en place si nous nous branchions sur les proxmox comme proposé initialement.

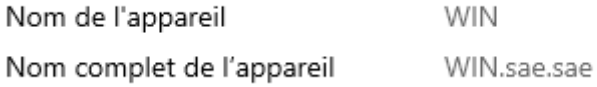


Installation du serveur Active Directory (AD) sur Windows Server 2019 via le Gestionnaire de Serveur intégré à ce dernier.

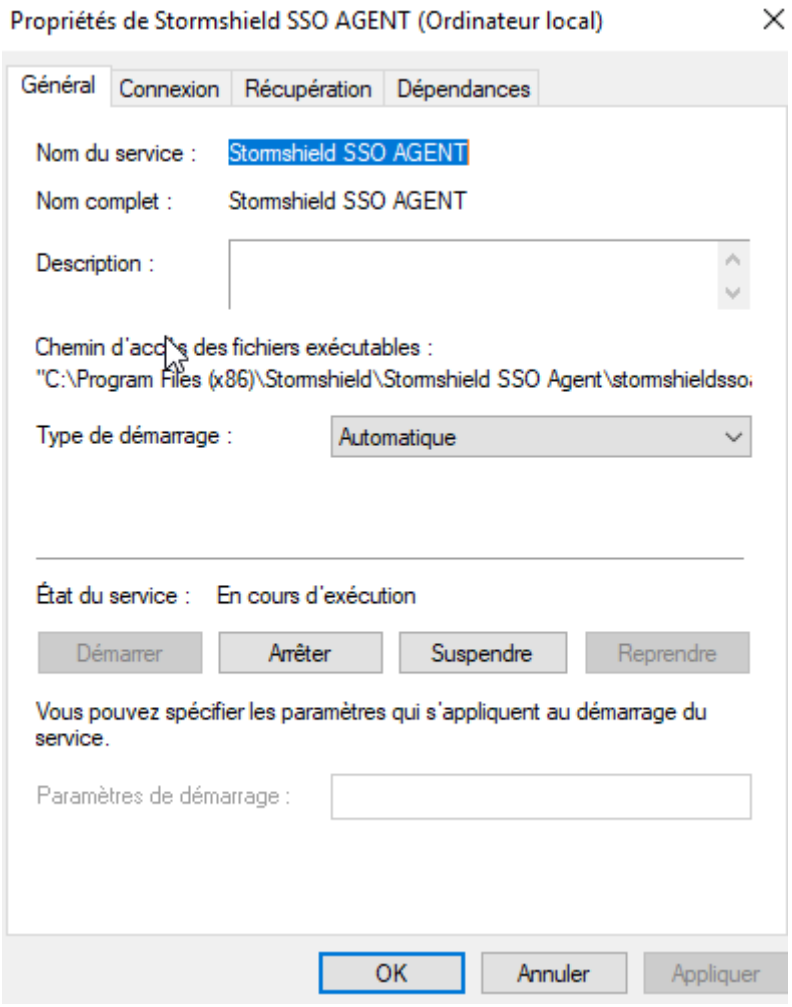
Création d'un utilisateur "test" dans l'AD.



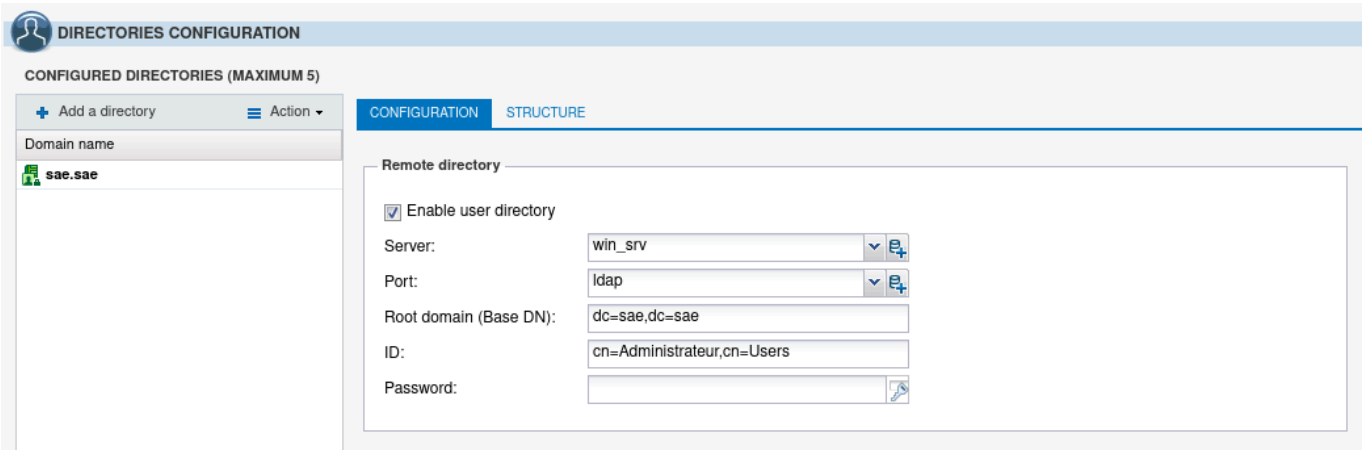
Intégration du PC Win10 dans l'AD.



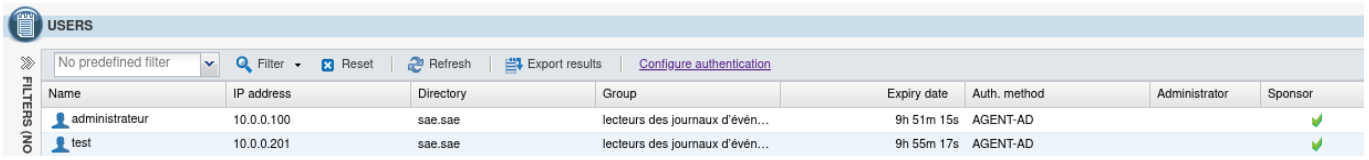
Intégration du Win 10 client dans le domaine "sae.sae".



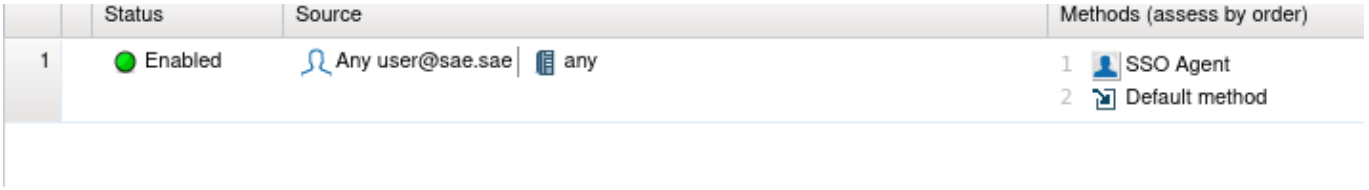
Démarrage du service en mode automatique (au lancement) "Stormshield Agent SSO" sur le serveur Windows.



Connexion de l'annuaire AD sur le firewall avec le compte administrateur et mot-de-passe défini dans l'agent.



Liste des utilisateurs connectés sur le réseau.



Règle d'authentification via SSO Agent sur le Firewall.

Dans un premier temps, nous avons configuré le service AD (Active Directory) sur le serveur Windows Serveur. Lors de cette installation nous avons créé une nouvelle forêt du domaine "sae.sae".
Nous avons créé un utilisateur "test" et nous les avons ajoutés au groupe "Utilisateurs du domaine" en plus d'avoir intégrer l'administrateur au groupe de "Lecteurs des journaux d'évènement".
Nous avons également intégré le PC client dans le domaine afin de pouvoir authentifier les utilisateurs créés précédemment. Nous n'avons pas rencontré de problème lors de cette étape.

Par la suite, nous avons installé le service "Stormshield Agent SSO" sur le serveur Windows afin de pouvoir communiquer les éléments de l'AD à notre Firewall.
Nous avons aussi changer l'adresse IP du client Win 10 afin de vérifier que l'utilisateur restait authentifié (.200 -> .201)

Sur le firewall, nous avons lié l'annuaire LDAP de l'AD via le SSO Agent. Puis, nous avons créé une règle d'authentification via "SSO Agent" afin d'authentifier les utilisateurs automatiquement sur le réseau.

Sur cette partie, nous avons perdu un peu de temps car les logs du FireWall ne sont pas affichés en temps réel (parfois plus de 5 minutes de délai). Nous pensions alors que notre configuration n'était pas bonne et nous faisions des modifications inutiles.

Tâche 12 Configuration d'un VPN SSL pour clients distants

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2x2h = 4 heure-homme

Estimation du temps passé sur cette tâche en heure-homme :

Objectif : Mettre en place un VPN SSL sur le site A pour le client du site B

Mettre en place un VPN SSL complet en utilisant un client OpenVPN pour Linux et le client Stormshield pour Windows.

Sous-tâches	Evaluation prof
Configuration d'un annuaire -> ok	100%
Génération d'un certificat -> ok	100%
Mettre en place les règles de filtrage et de NAT -> ok	100%
Configuration du service VPN SSL sur le Stormshield -> ok	100%
Installation et paramétrage des clients -> ok	100%
Tests de connexion -> ok	100%

Rapport

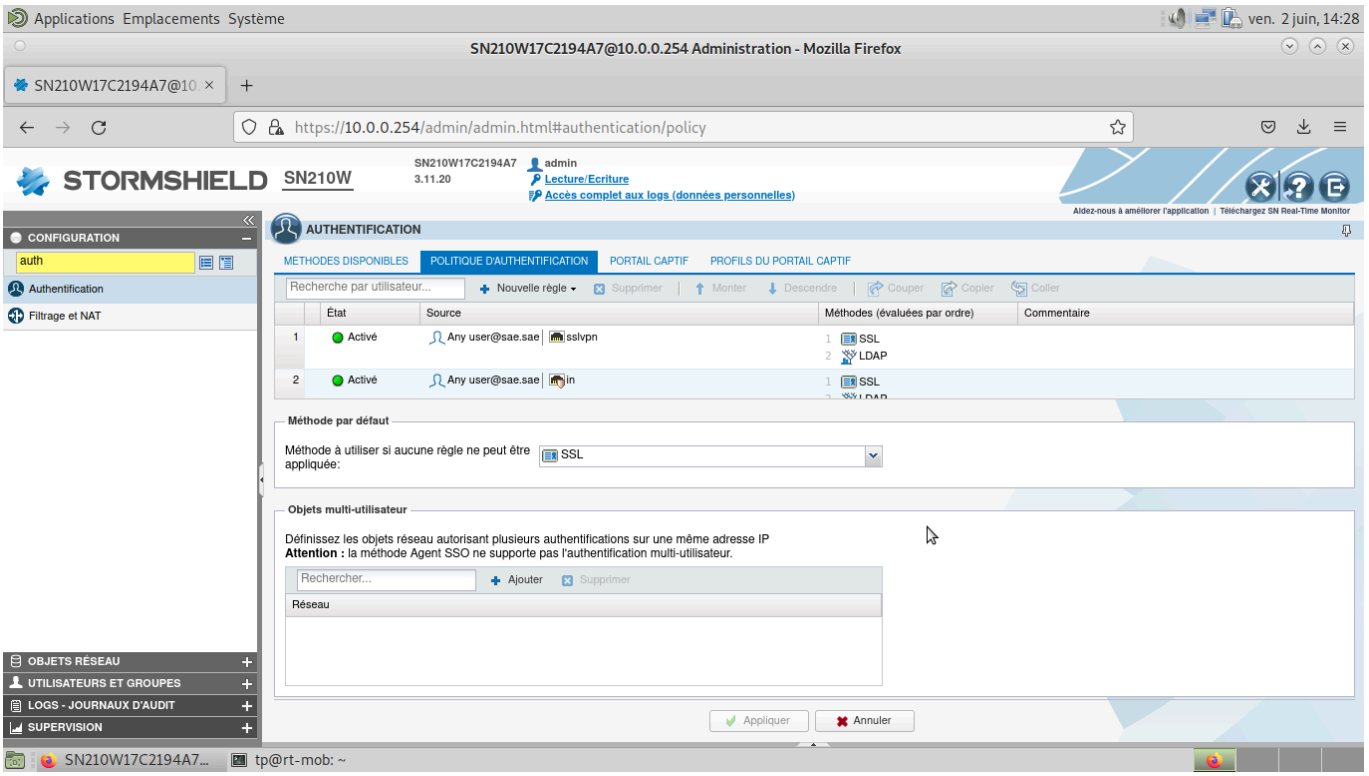
(Captures d'écrans de la configuration Stormshield et des clients, etc.)

```
root@rt-mob:/home/tp/Téléchargements# openvpn openvpn_client.ovpn
2023-06-02 14:27:14 WARNING: Compression for receiving enabled. Compression has been used in the past to break encryption. Sent packets
 are not compressed unless "allow-compression yes" is also set.
2023-06-02 14:27:14 DEPRECATED OPTION: --cipher set to 'AES-128-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future Op
enVPN version will ignore --cipher for cipher negotiations. Add 'AES-128-CBC' to --data-ciphers or change --cipher 'AES-128-CBC' to --d
ata-ciphers-fallback 'AES-128-CBC' to silence this warning.
Enter Auth Username: user
Enter Auth Password: *********
```

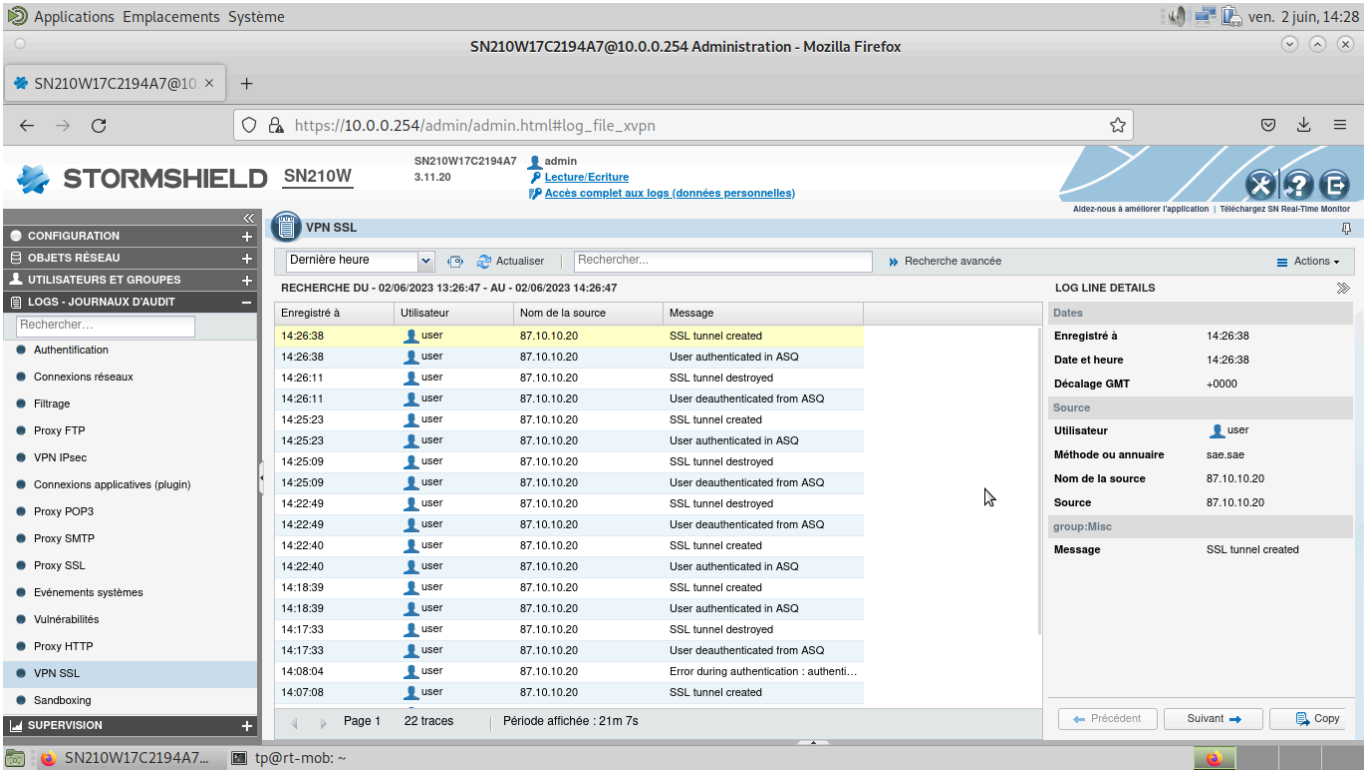
Connexion au VPN SSL avec méthode SSL+EAP (certificat + login/password)

```
8: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 172.10.20.6 peer 172.10.20.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::c71a:6055:4ef5:8a60/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
tp@rt-mob:~$ ip r
default via 192.168.2.254 dev enp3s0 onlink
10.0.0.0/8 via 172.10.20.5 dev tun0
169.254.0.0/16 dev enp3s0 scope link metric 1000
172.10.10.0/24 via 172.10.20.5 dev tun0
172.10.10.1 via 172.10.20.5 dev tun0
172.10.20.0/24 via 172.10.20.5 dev tun0
172.10.20.1 via 172.10.20.5 dev tun0
172.10.20.5 dev tun0 proto kernel scope link src 172.10.20.6
192.168.1.0/24 via 172.10.20.5 dev tun0
192.168.2.0/24 dev enp3s0 proto kernel scope link src 192.168.2.1
```

Affichage l'interface tunnel et de ses routes associées



Règles d'authentification par certificat pour le VPN SSL via l'interface dédiée



Logs du VPN SSL lors de la connexion et la création du tunnel

Le VPN SSL consiste en la création d'un tunnel entre un serveur et un client avec une vérification des certificats respectifs auprès de la Certificate Autority (CA) et de sa Certificate Revocation List (CRL).

La méthode d'authentification Extensible Authentication Protocol (EAP) permet d'ajouter une surcouche en recoupant une paire de login/password avec un annuaire afin d'autoriser certains utilisateurs seulement à se connecter, ou simplement ajouter une sécurité supplémentaire.

Tâche 13 Configuration d'un VPN IPSEC site à site

Liste des personnes impliquées avec pourcentage de répartition	
HIRSCH Matéo 50% ECOTIERE Léo 50%	2x3.5h = 7 heure-homme

Estimation du temps passé sur cette tâche en heure-homme :

Objectif : Mettre en place un VPN IPSEC entre vos deux sites

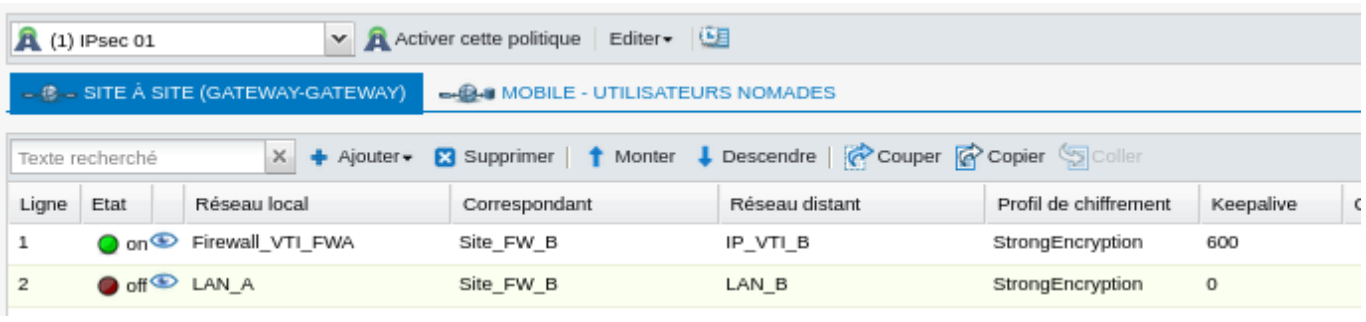
Vous commencerez par mettre en place un tunnel VPN IPSEC simple entre vos deux LANs, une fois testé et validé, vous mettrez en place un VPN utilisant les Virtual Tunneling Interface (VTI) pour relier dans un seul tunnel vos 4 réseaux (LAN A, DMZ A, LAN B et DMZ B).

Sous-tâches	Evaluation prof
Mettez en place un tunnel VPN entre vos deux LANs	100%
Testez et faites valider	100%
Mettez en place un tunnel entre tous vos réseaux en utilisant les VTI	100%
Testez et faites valider	100%
Utilisation des certificats pour l'authentification des SNSs	100%
Testez et faites valider	100%

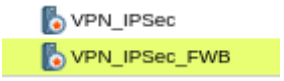
Rapport

(Captures d'écrans de la configuration Stormshield et des clients, etc.)

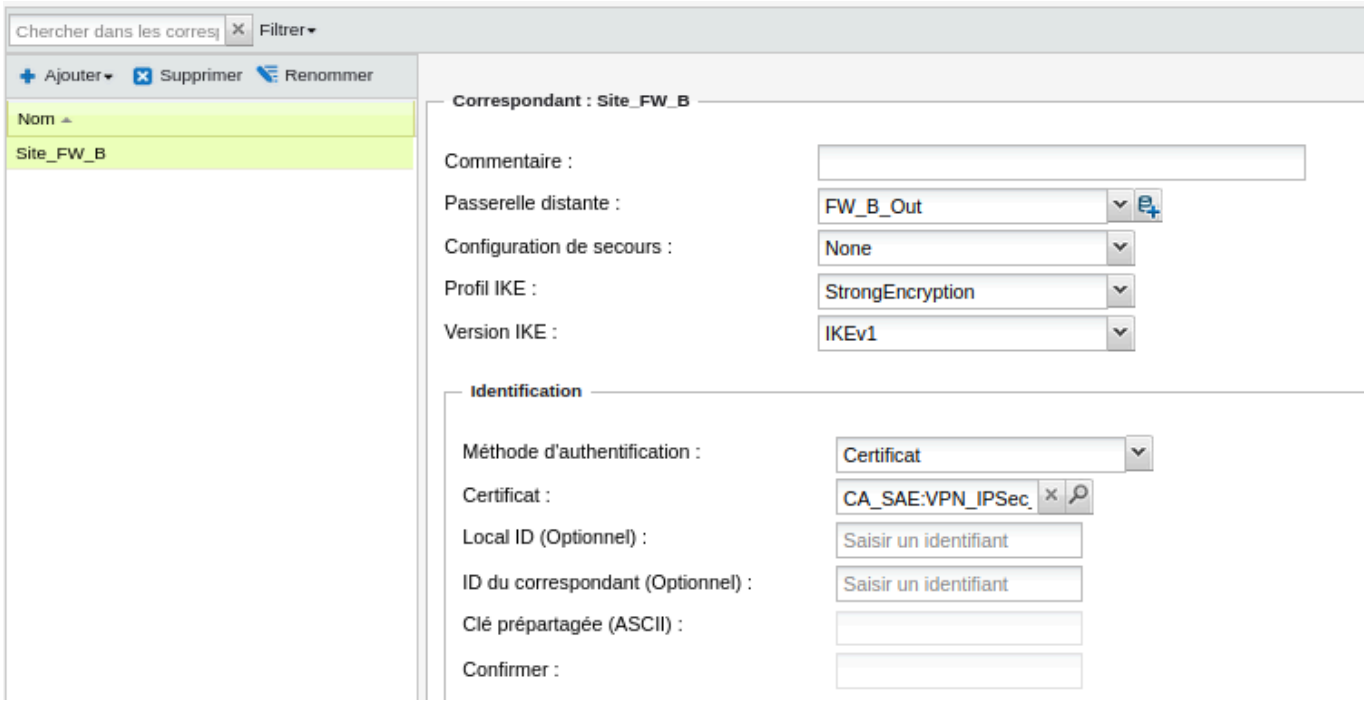
Pour établir un VPN site à site nous devons faire une configuration symétrique (c'est pourquoi nous ne vous montrons qu'un côté de la configuration) sur chaque extrémité de ce dernier. A noter qu'un VPN IPsec ne monte pas un tunnel comme un VPN SSL par exemple. Le VPN IPsec est une liaison IP sécurisée et chiffrée.



Configuration des 2 VPN IPsec sur le firewall du Site A.

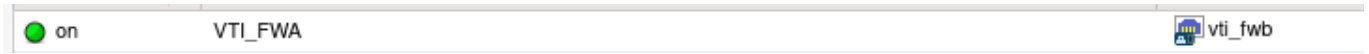


Certificats pour VPN IPsec avec méthode d'authentification certificat.

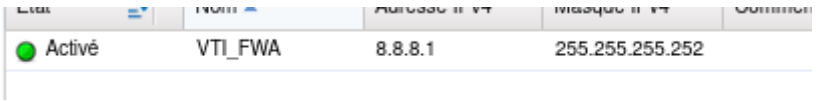


Configuration du correspondant IKEv1 sur le firewall du Site A.

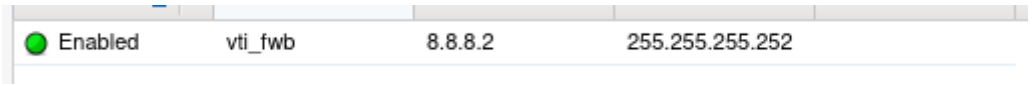
Route statique pour IPsec via VTI sur le firewall du Site A.



Route de retour pour IPsec via VTI sur le firewall du Site A.



Création du VTI sur le firewall du Site A.



Création du VTI sur le firewall du Site B.

Établissement de la connexion IPsec.

VPN IPsec site à site entre les deux LANs :

Pour établir un VPN site à site nous avons fait la même configuration sur chaque firewall. C'est-à-dire une création de correspondants et de la connexion. Nous avons créé des correspondants IKEv1 (Internet Key Exchange v1) avec une clé PSK afin de chiffrer la liaison.
Nous avons également configuré des règles de filtrages en autorisant le flux entre les 2 LANs lorsqu'il ne passe **que** par le VPN.

VPN IPsec avec VTI :

Pour établir ce VPN nous avons créé sur chaque firewall une VTI (Virtual Tunnel Interface) afin de faire passer le trafic des réseaux internes (IN + DMZ) de chaque firewall sur le VPN.
Après avoir créé les deux VTI, nous avons configuré le VPN en précisant cette fois-ci que les réseaux locaux et distants sont les VTI respectives. Nous avons utilisé la même configuration du correspondant IKEv1 que pour le VPN IPsec précédent.
Par la suite, nous avons établi des routes afin de préciser que les réseaux locaux doivent passer par les VTI et par conséquent sur le VPN. Des routes de retour ont été également mises en place.

VPN IPsec avec VTI avec certificats :

Pour ce VPN, la configuration est la même que le précédent a une seule différence : le correspondant IKEv1 est configuré avec méthode d'authentification par certificat.
Pour ce faire, sur le firewall du Site A nous avons renseigné dans la configuration du correspondant (à savoir le firewall du Site B) le certificat de ce dernier et inversement sur le deuxième firewall.
Par conséquent, lors de l'établissement du VPN, le firewall du Site A peut certifier le firewall du Site B et de même manière pour le deuxième firewall, cela permet de s'assurer de l'authenticité du correspondant.