**Année 2023-2024**

**SAÉ Cyber 4.0 Sécurisation d’un SI**

**Cahier de SAÉ**

**Version 1.0**



| **Appartient à :** | **En équipe avec :** |
| --- | --- |
| **Nom : Gambier**  **Prénom : Maxence** | **Nom : Wittig**  **Prénom : Antonin** |
|  | |
| **Nom : Vadam**  **Prénom : Julien** | **Nom : Paquelet**  **Prénom : Étienne** |
| **Nom de votre équipe : TGAI Infinity** | |

# Informations générales

# Répartition en groupes

6 équipes de 4 étudiants et 1 équipe de trois étudiants

# 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 en solo de 10 mins par étudiant + 5 mins de question.

# Matériel par équipe

• 2 Firewalls Stormshield

• 1 Switch

• 1 Borne WiFi

• 5 PC tour

• 2 Portables

# Documentation

* Moodle
* <https://documentation.stormshield.eu/>

**Tâches à réaliser**

1. Mise en place d’une infrastructure sécurisée
2. Installation et configuration d’un firewall Stormshield
3. Installation et configuration d’un serveur HTTP/HTTPS et d’un serveur FTP/FTPS
4. Authentification transparente par certificat SSL
5. Mettre en place un IDS
6. Attaque sur le Wifi
7. Utilisation de scanneurs de vulnérabilité
8. Attaque Man in The Middle
9. Contre-mesures pour le MiM
10. Supervision du réseau
11. Mise en place d’une architecture Single Sign-On
12. Mise en place d’un VPN SSL pour clients distants
13. Mise en place d’un VPN IPSEC site à site

**Gestion de votre projet**

Créez un Trello de votre projet estimez la durée de chaque tâche et sous-tâche et affectez-les entre vous. Partagez le Trello avec les enseignants.

# Bilan

A la fin de votre SAÉ, vous devrez répartir 80h de travail x 4 personnes soit 320 heures-homme 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 | 1H | 100% |
| Installation et configuration d’un firewall Stormshield | 7H | 100% |
| Installation et configuration d’un serveur HTTP/HTTPS et d’un serveur FTP/FTPS | 36H | 100% |
| Authentification transparente par certificat SSL | 20H | 100% |
| Mettre en place un IDS | 72H | 100% |
| Attaque sur le Wifi | 24H | 100% |
| Utilisation de scanners de vulnérabilité | 39H | 100% |
| Attaque Man in The Middle | 5H | 100% |
| Contre-mesures pour le MiM | 4H | 100% |
| Supervision du réseau | 2H | 50% |
| Mise en place d’une architecture Single Sign-On | - | 0% |
| Mise en place d’un VPN SSL pour clients distants | 1H | 100% |
| Mise en place d’un VPN IPSEC site à site | 3H | 30% |

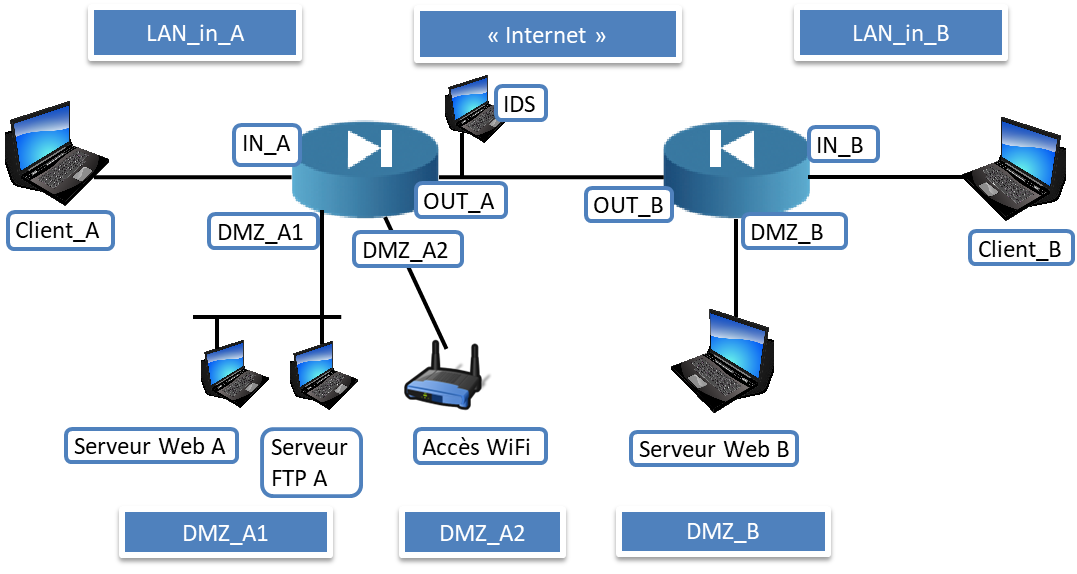
**Détails des tâches à réaliser**

**Tâche 1. Mise en place d’une infrastructure sécurisée (1,5 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| Gambier Maxence | 100% |

Estimation du temps passé sur cette tâche en heure-homme : 1H

**Objectif : Mettre en place l’infrastructure réseau suivante :**

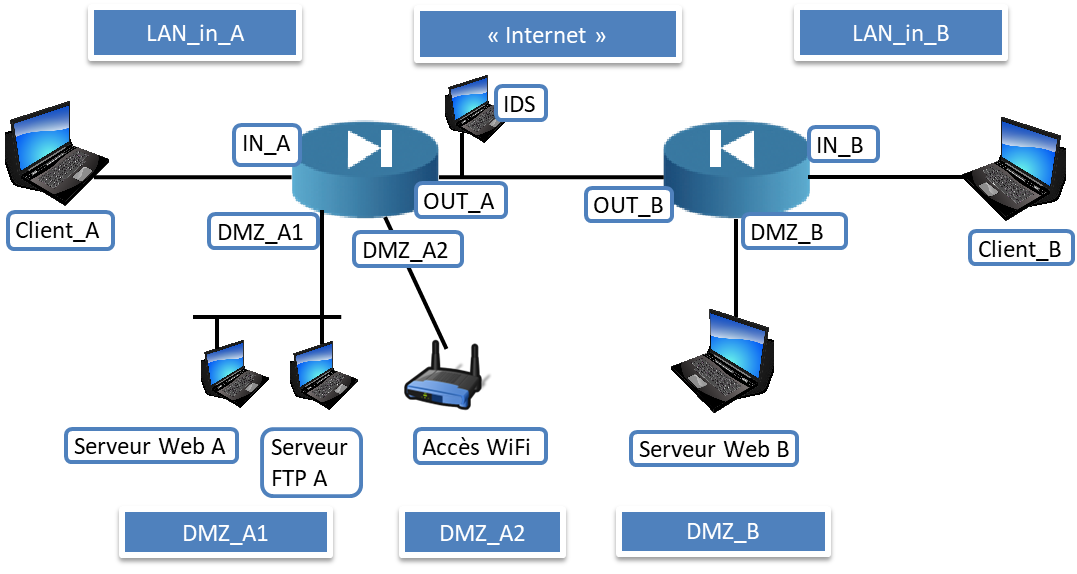


**Rapport**

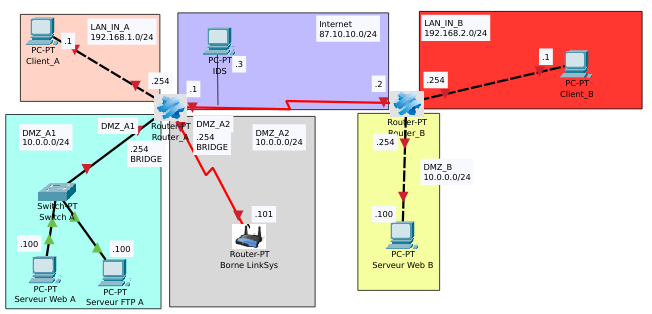
*(Expliquez votre démarche, dessinez un plan IP, insérez des photos de votre architecture avec identification de chaque machine, photo des écrans de configuration IP, etc.)*

**Étape 1) Refaire le schéma réseau**

Pour commencer la tâche numéro 1, qui est une tâche simple mais importante pour la poursuite du projet, j’ai décidé de refaire le schéma réseau à partir de celui donné dans le cahier de Saé.



*Illustration 1.1 : Schéma réseau de base*



*Illustration 1.2 : Nouveau schéma réseau*

Le schéma a été réalisé de manière à le rendre plus compréhensible aux autres membres du groupes.

J’y ai également implémenté les IPs que nous pourrons utiliser.

**Étape 2) Plan d’adressage IP**

Pour continuer cette tâche, j’ai réaliser le plan d’adressage IP suivant en commençant par les plages d’adresses IPs :

| **Nom du réseau** | **Plage d’adresse** |
| --- | --- |
| LAN\_IN\_A | 192.168.1.0/24 |
| DMZ\_A1 | 10.0.0.0/24 |
| DMZ\_A2 | 10.0.0.0/24 |
| Internet | 87.10.10.0/24 |
| LAN\_IN\_B | 192.168.2.0/24 |
| DMZ\_B | 10.0.0.0/24 |

Par la suite, une fois les plages attribué, j’ai pu attribué une IP par machine comme suivant :

| **Machines / Interfaces** | **Adresse IP** |
| --- | --- |
| IN\_A | 192.168.1.254 |
| Client\_A | 192.168.1.1 |
| DMZ\_A1 | 10.0.0.254 |
| Serveur Web A | 10.0.0.100 |
| Serveur FTP | 10.0.0.100 |
| DMZ\_A2 | 10.0.0.254 |
| Borne Wifi | 10.0.0.101 |
| OUT\_A | 87.10.10.1 |
| OUT\_B | 87.10.10.2 |
| IDS | 87.10.10.3 |
| LAN\_IN\_B | 192.168.2.254 |
| Client\_B | 192.168.2.1 |
| DMZ\_B | 10.0.0.254 |
| Serveur Web B | 10.0.0.100 |

Ces plages d’adressages ont été choisies par rapport aux habitudes prises lors des TP basé sur Stormshield. Nous avons mis des adresses LAN différentes entre LAN\_IN\_A et LAN\_IN\_B pour éviter les conflits d’adresses IP tandis que nous avons mis les mêmes IPs dans les DMZ car celles-ci sont privées et ne seront connues qu' en interne.

**Étape 3) Mise en place physique de l’infrastructure**

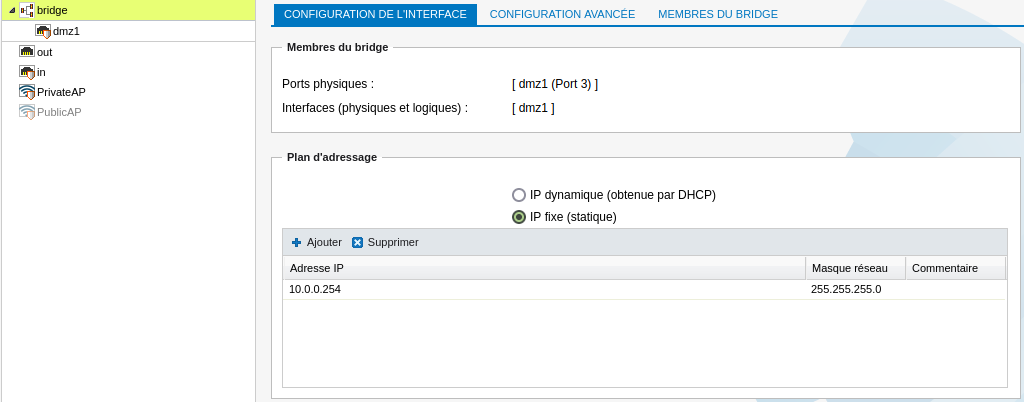
Une fois le schéma réalisé et le plan d’adressage aussi, j’ai commencé à mettre en place notre infrastructure « réel ».

J’ai donc effectué les branchements par rapport au schéma réseau et réalisé les configurations appropriées à chaque appareil.

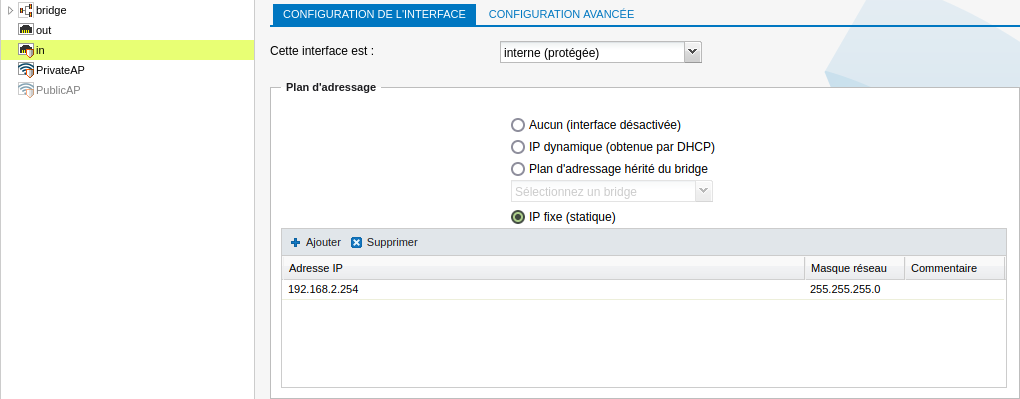
J’ai eu 2 problèmes lors de la mises en place de l’infrastructure :

* Mise en place DMZ\_A1 et DMZ\_A2 : Nous ne pouvions pas mettre 2 DMZ sur nos routeurs Stormshield car ceux-ci sont des SN210W. Ce modèle de stormshield ne possède que 1 port DMZ, 6 Port LAN et 1 port WAN. Nous avions eu comme première idée de transformer 1 port LAN en DMZ mais celle-ci est impossible car le Stormshield reconnaît les 6 ports LAN en 1 seul interface IN comparé à d'autres modèles qui ont une interface par port et chacun est indépendant. Nous avons donc choisis de réunir ces 2 DMZ en un seul.
* Ajout d’un switch : Nous avons rajouté un switch dans le réseau internet auquel on reliera les ports OUT\_A et OUT\_B ainsi que l’IDS car sans switch celui-ci ne peut pas être ajouté à l’infrastructure sans switch.

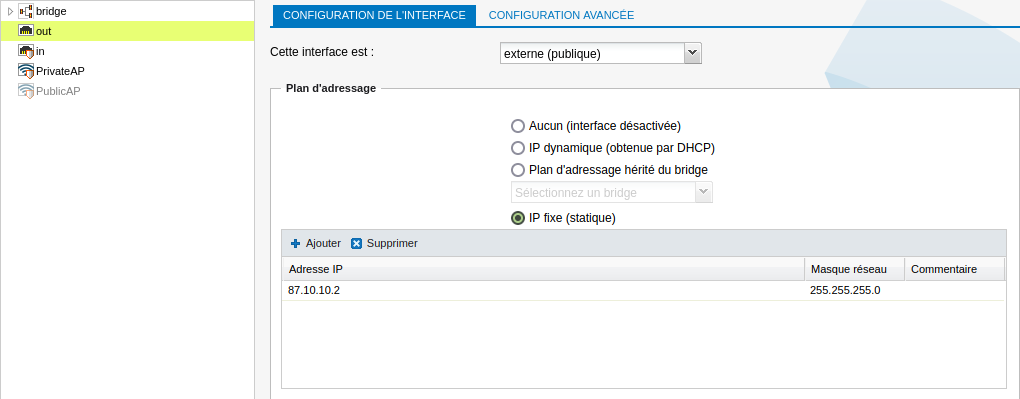
Une fois tout cela terminé nous avons pu commencer à repartir les autres tâches et à les débuter.

****

*Illustration 1.3 : Configuration interface DMZ\_B*

****

*Illustration 1.4 : Configuration interface IN\_B*

****

*Illustration 1.5 : Configuration Interface OUT\_B*

**

*Illustration 1.6 : Configuration IP Serveur Web A*

**Tâche 2 Configuration des firewalls Stormshields (3 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| VADAM Julien  PAQUELET Etienne | 50%  50% |

Estimation du temps passé sur cette tâche en heure-homme : 7H

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

| Sous-tâches | Evaluation prof |
| --- | --- |
| Mettre en place une politique de NAT |  |
| Permettre l’accès aux serveurs uniquement sur les ports concernés |  |
| Interdire l’établissement d’une connexion sur les réseaux internes depuis les réseaux externes et les DMZ |  |
| Autorisez l’accès à DMZ\_A1 depuis DMZ\_A2 |  |
| Testez l’accès aux serveurs |  |

**Rapport**

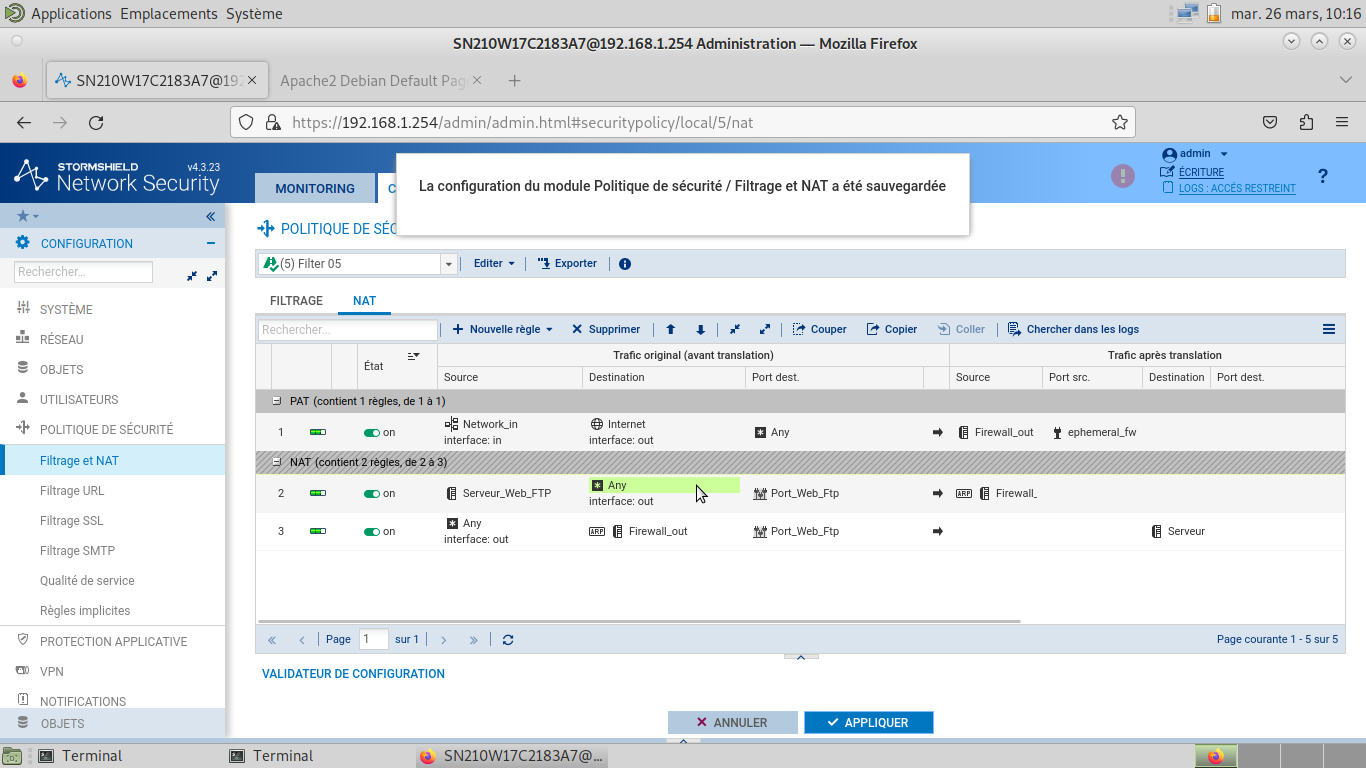
*(Expliquez votre démarche, insérez les captures d’écran des menus NAT et Filtrage, de vos tests, etc.)*

Le but de cette tâche est de configurer le filtrage et la NAT sur nos firewall et par la suite de sécuriser le réseau interne. Nous avons suivi le plan d’adressage IP (voir Tâche 1) afin de réaliser le filtrage et la NAT.

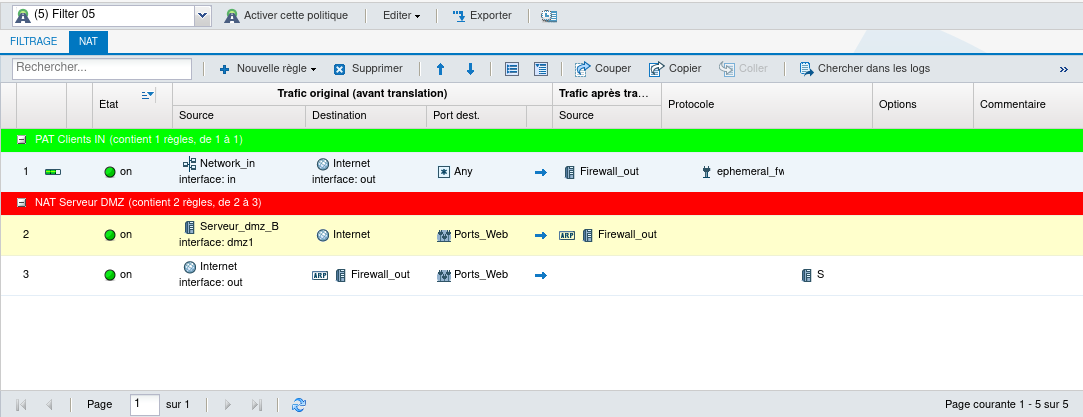
Nous avons tout d’abord configurer la NAT :

* Mise en place d’une PAT sur les 2 firewalls pour les utilisateurs internes.
* Mise en place d’une NAT pour le réseau DMZ de chaque firewall sur l’interface externe pour les ports http et https.

**Étape 1) Mise en place d’une PAT sur le firewall A**

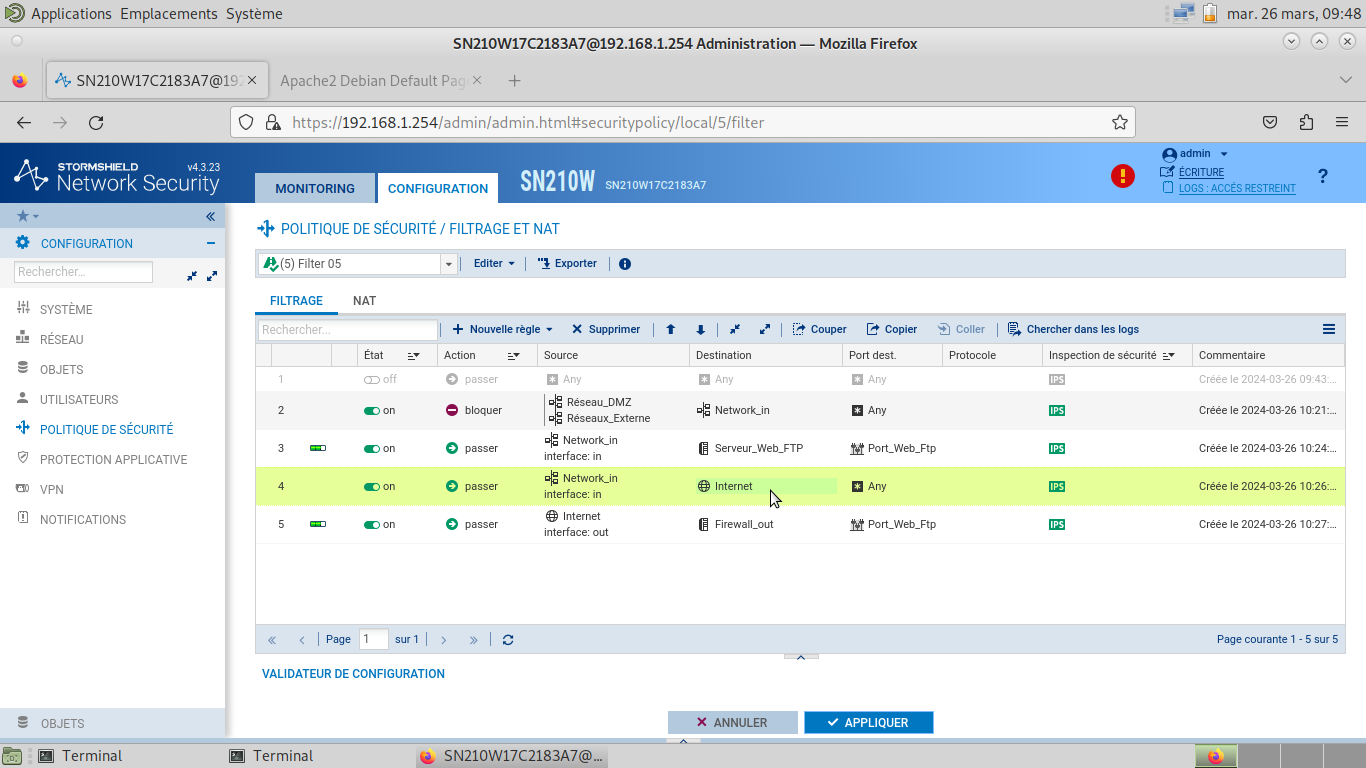


Mise en place de la même PAT sur le firewall B



* Mise en place du filtrage.

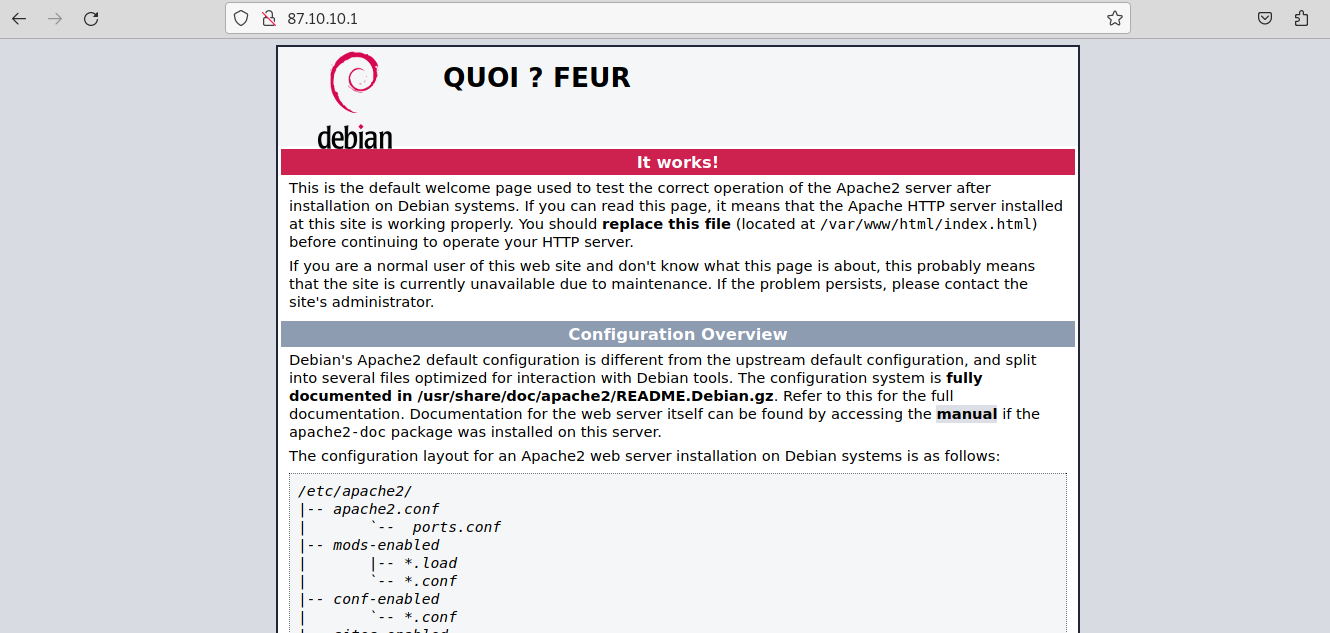
Le filtrage reste quasiment le même sur les deux firewalls, nous vous présentons la configuration du firewall A :



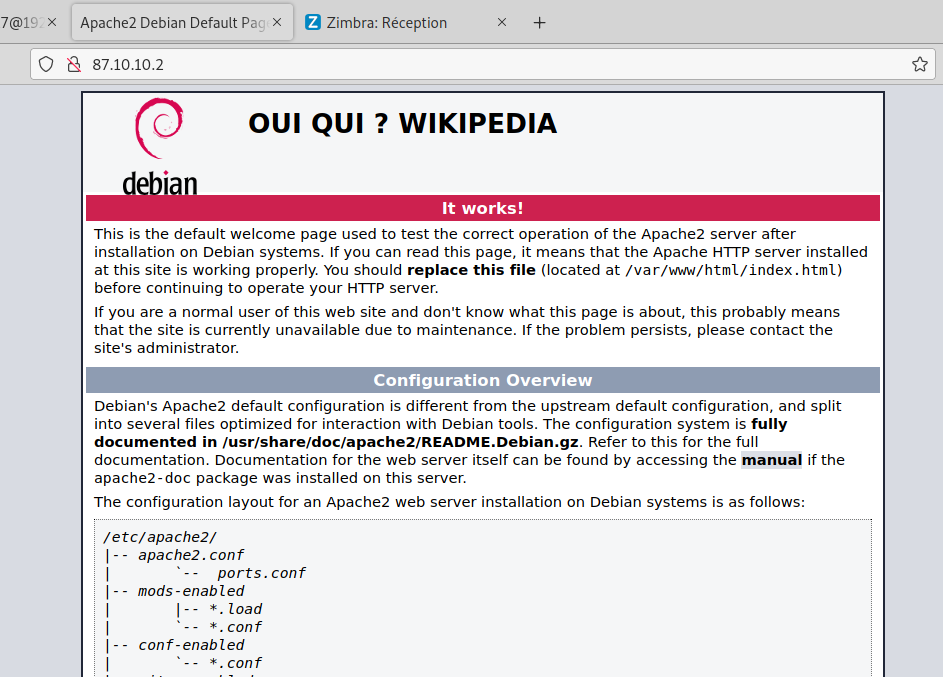
Ici, les PC présent dans le réseau interne peuvent accéder au serveur web/ftp et internet

Les paquets venant du réseau externe et du réseau de la DMZ ne peuvent pas accéder au réseau interne. Les paquets en provenance d’internet peuvent accéder à l’interface “out” du pare-feu.

Afin de tester ces règles, nous pouvons nous connecter sur l’interface out du pare feu A depuis le client B :



Nous pouvons également nous connecter sur l’interface out du pare feu B depuis le Client A :



Notre configuration fonctionne alors. Cependant, durant la configuration, nous avons eu quelques problèmes. Le premier problème était que nous n’arrivions pas à faire communiquer nos deux pare-feux sans filtrage. Pour cela, nous avons repris la configuration à zéro, puis réalisé le filtrage et la NAT étapes par étapes, en commençant par la NAT et en vérifiant le comportement escompté de chaque machine avec la configuration mise en place. Cette méthode nous à permis de trouver le problème et le solutionné.

**Tâche 3 Serveurs HTTP/HTTPS et serveur FTP/FTPS (7,35 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| WITTIG Antonin | 100% |

Estimation du temps passé sur cette tâche en heure-homme : 36H

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

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

**Rapport**

**Installer les serveurs HTTP :**

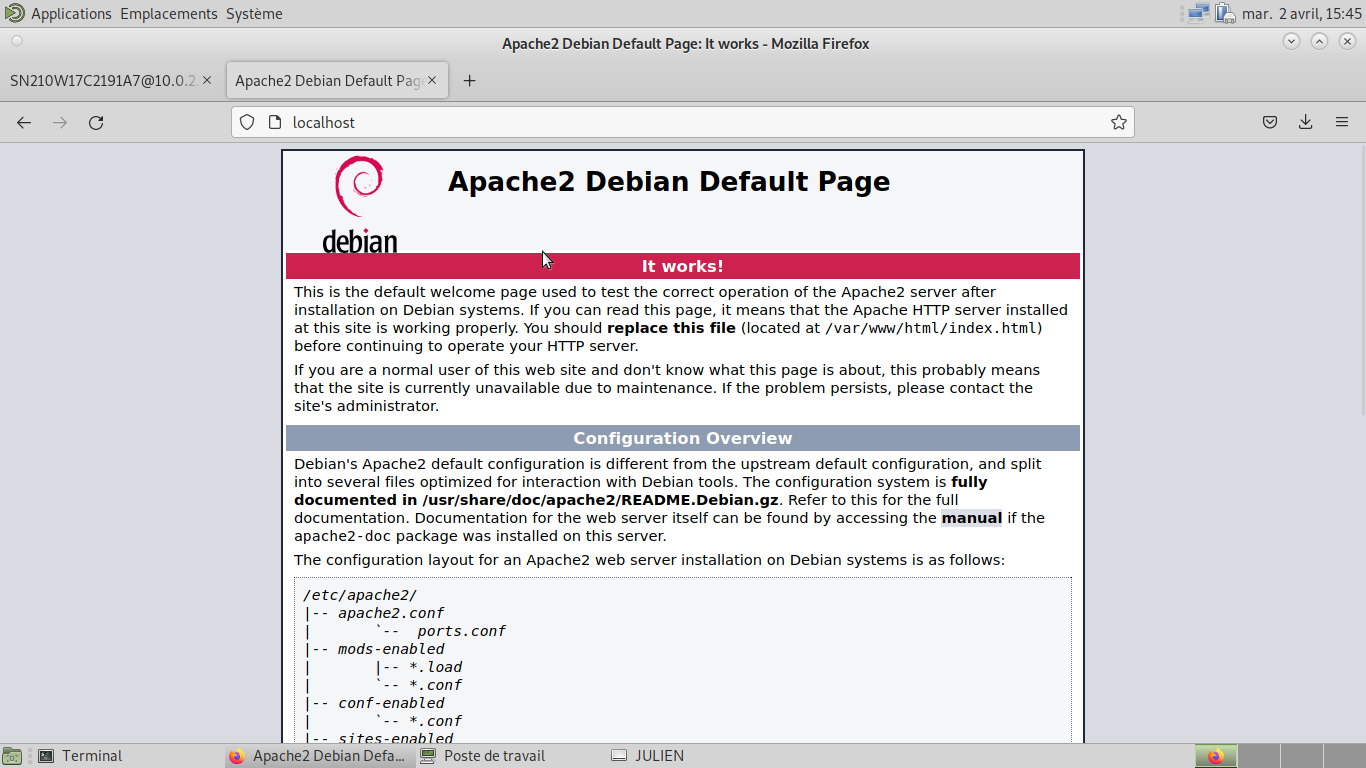
Pour commencer, il faut installer le paquet Apache2 :

| *sudo apt-get install apache2* |
| --- |

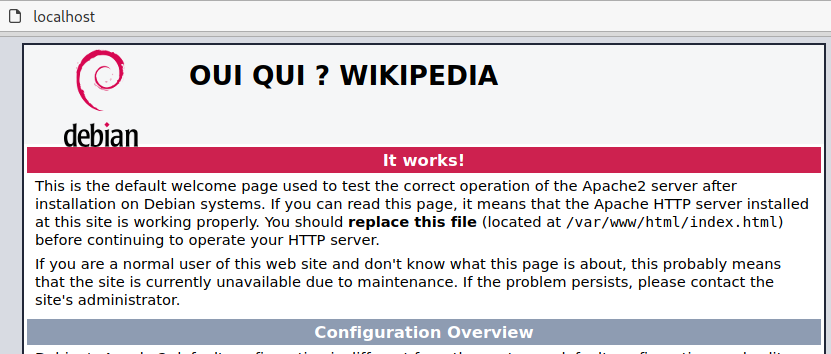
Puis il faut activer le service en effectuant cette commande :

| *sudo systemctl start apache2* |
| --- |

Une fois ces deux étapes effectuées, nous pouvons accéder au site par défaut en renseignant l'adresse IP de notre machine serveur ou en mettant localhost :



capture d'écran du site sur le serveur web B



capture d'écran du site sur le serveur A

**Activer HTTPS :**

Pour activer HTTPS, il faut générer un certificat SSL/TLS. Pour cela, il faut installer l'outil OpenSSL afin de générer ce certificat :

| *sudo apt install openssl* |
| --- |

Puis on génère le certificat avec cette commande :

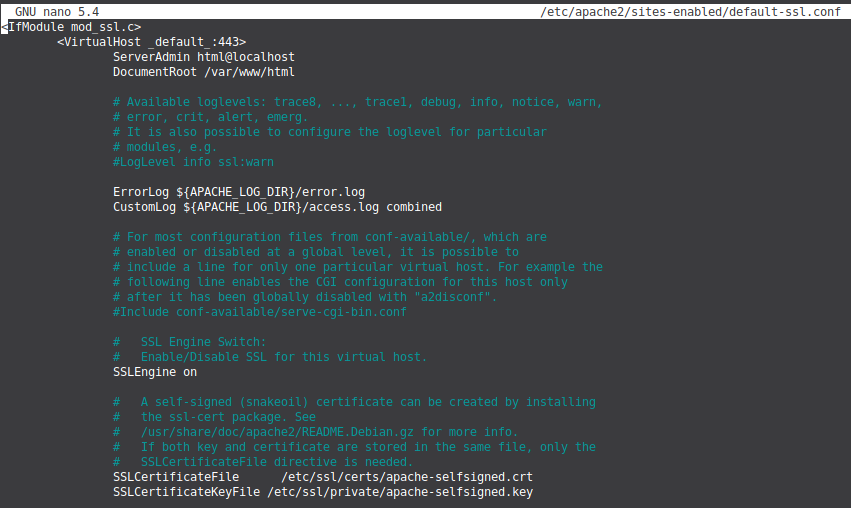
| *sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt* |
| --- |

Cette commande génère un certificat auto-signé et une clé privée. Le certificat est valable pendant 365 jours et utilise une clé RSA de 2048 bits. La clé privée est enregistrée dans /etc/ssl/private/ sous le nom apache-selfsigned.key, le certificat est enregistré dans /etc/ssl/certs/ sous le nom apache-selfsigned.crt. L'option -nodes est utilisée pour ne pas crypter la clé privée avec un mot de passe.

Une fois le certificat auto-signé généré ainsi que la clé privée, il faut activer le module SSL d'Apache :

| *sudo a2enmod ssl* |
| --- |

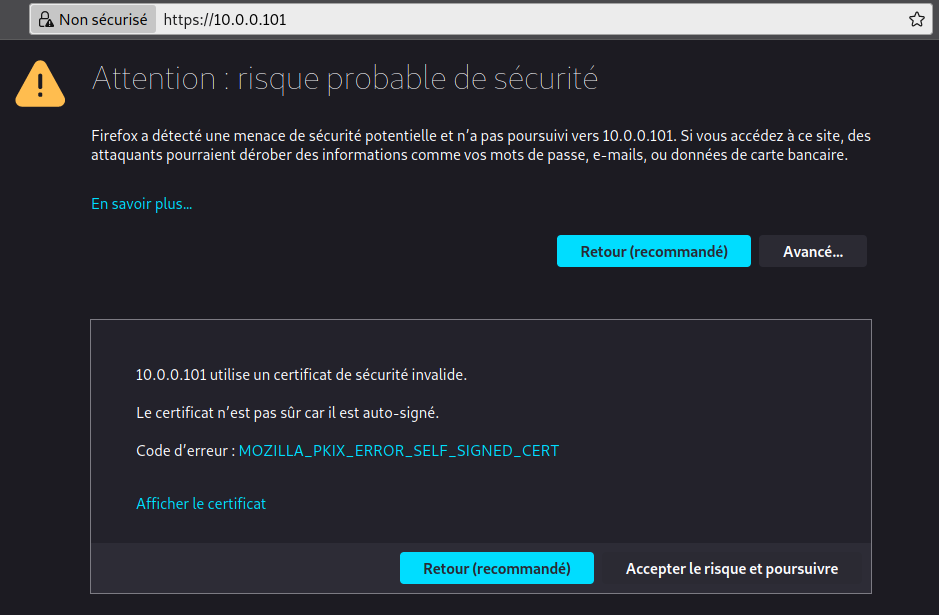
Et modifier le fichier de configuration du site web, /etc/apache2/sites-available/default-ssl.conf en modifiant les lignes SSLCertificateFile et SSLCertificateKeyFile avec la bonne direction de fichier :



Ensuite il faut activer le site pour pouvoir activer la nouvelle configuration qu'on vient de faire :

| *sudo a2ensite default-ssl.conf* |
| --- |

Maintenant quand on se connecte au site en https, il faut accepter l'utilisation du certificat auto-signé pour avoir une connexion sécurisé :



On voit un code d'erreur car c'est un certificat auto-signé donc la sécurité n'est pas maximale, il faudrait faire un certificat SSL authentifié par une entité de certification.

**Mise en place du CMS :**

Comme CMS nous avons choisi d'utiliser WordPress.

Pour cela il fallait mettre en place un serveur LAMP ( L pour linux, A pour apache, M pour Mariadb et P pour PHP )

**PHP :**

Dans un premier temps il faut télécharger le paquet php pour pouvoir exécuter les scripts intégré dans la page web :

| *sudo apt-get install php* |
| --- |

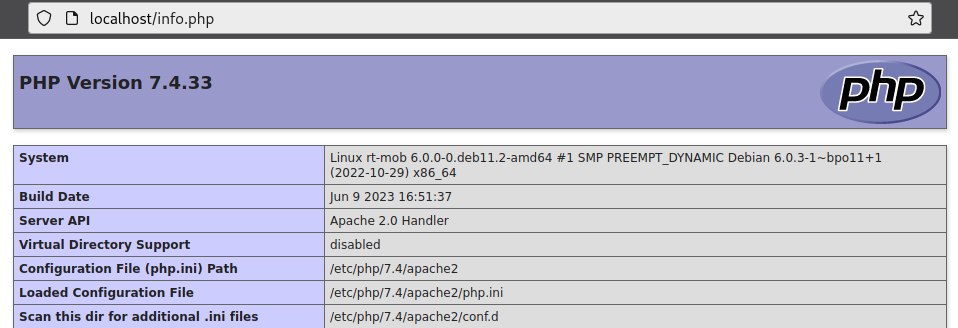
Il faut ajouter quelques paquets spécifiques pour que php puisse interagir avec l'instance Mariadb :

*sudo apt-get install -y php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath*

Pour vérifier que php s'est bien installé nous pouvons créer une page web avec le code php suivant dedans :

| <?php  phpinfo();  ?> |
| --- |

Ce qui nous donne :



**Mariadb :**

Il faut installer le paquet Mariadb :

| *sudo apt-get install mariadb-server* |
| --- |

Ensuite on ajoute un mot de passe à l'instance afin de sécuriser celui-ci.

Pour se connecter à la base de donnée, li faut utiliser la commande suivante puis le mot de passe :

| *sudo mariadb -u root -p* |
| --- |

**WordPress :**

WordPress s'appuie sur une base de données afin de stocker toutes les informations liées à la configuration et aux contenus.

Dans un premier temps, il faut créer une base de données :

| *CREATE DATABASE TG\_AI\_Infinity* |
| --- |

Puis il faut créer l'utilisateur qui sera administrateur de la base de données :

| *CREATE USER 'admin' @ 'localhost' IDENTIFIED BY 'admin' ;* |
| --- |

Ensuite il faut donner les droits à l'utilisateur sur la base de données :

| *GRANT ALL PRIVILEGES ON TG\_AI\_Infinity.\* TO admin@localhost ;* |
| --- |

Ensuite, il faut exécuter la commande suivante pour actualiser les droits sur la base de données :

| *FLUSH PRIVILEGES ;* |
| --- |

Pour finir, on va remplacer la page par défaut d'Apache par WordPress. Pour cela il faut télécharger l'archive zip de WordPress

| *wget* [*https://wordpress.org/latest.zip*](https://wordpress.org/latest.zip) |
| --- |

Décompresser l'archive

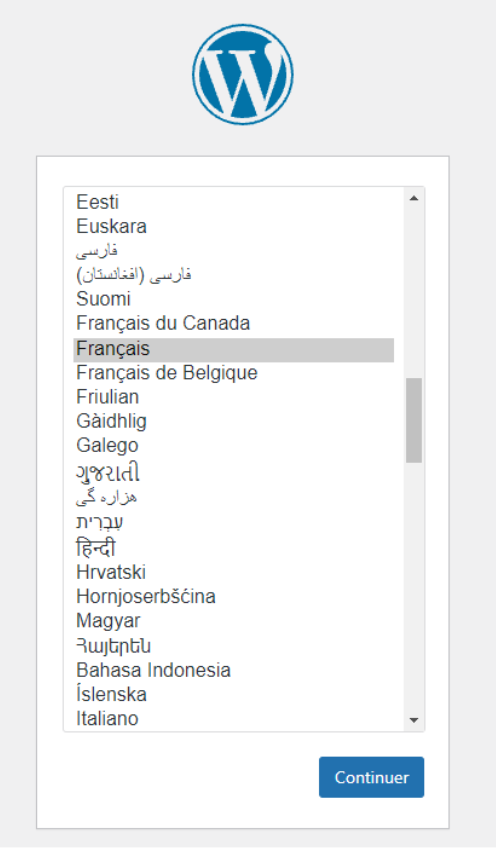
| *unzip latest.zip -d /var/www/html* |
| --- |

On se déplace dans /var/www/html et on déplace tout le contenu de wordpress dans la racine du site.

On doit donner les droits à l'utilisateur 'www.data' qui correspond à apache :

| *chown -R www-data:www.data /vaar/www/html* |
| --- |

Il ne reste plus qu'a configurer wordpress en nous rendant sur notre page web :

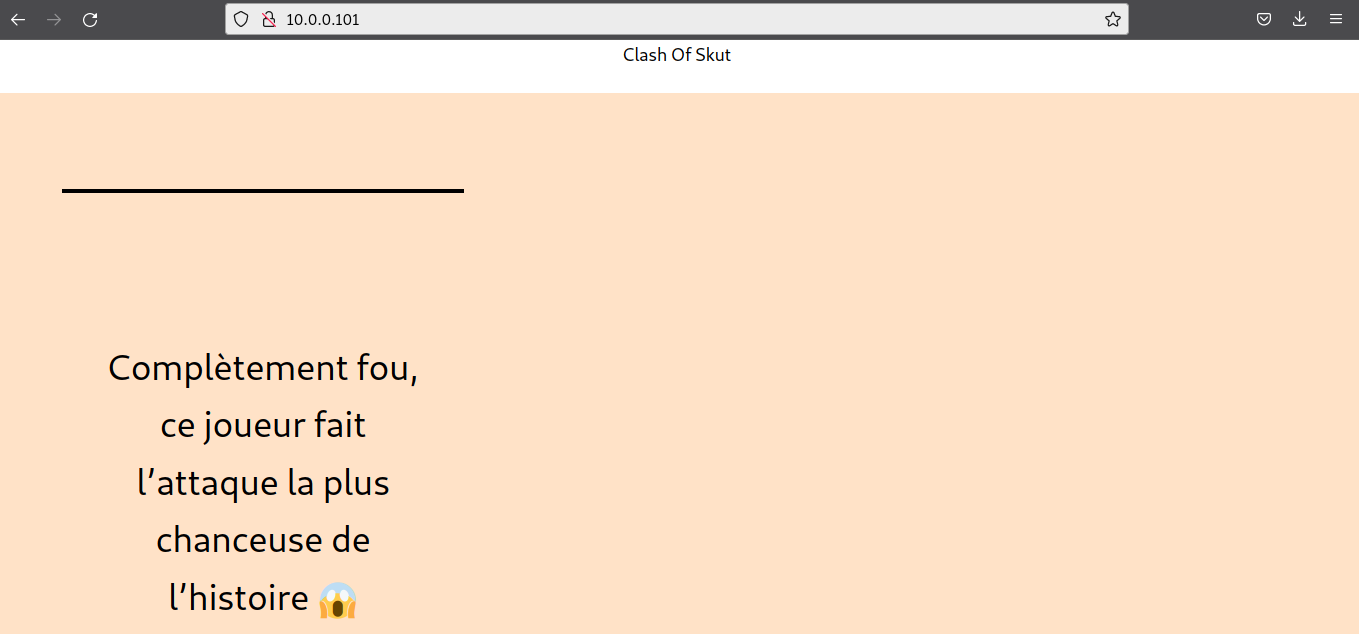


On nous demande de renseigner un nom de domaine avec notre identifiant, mot de passe avec le préfixe des tables :



Après avoir finit la configuration, on peut accéder à l'interface d'administration de wordpress :

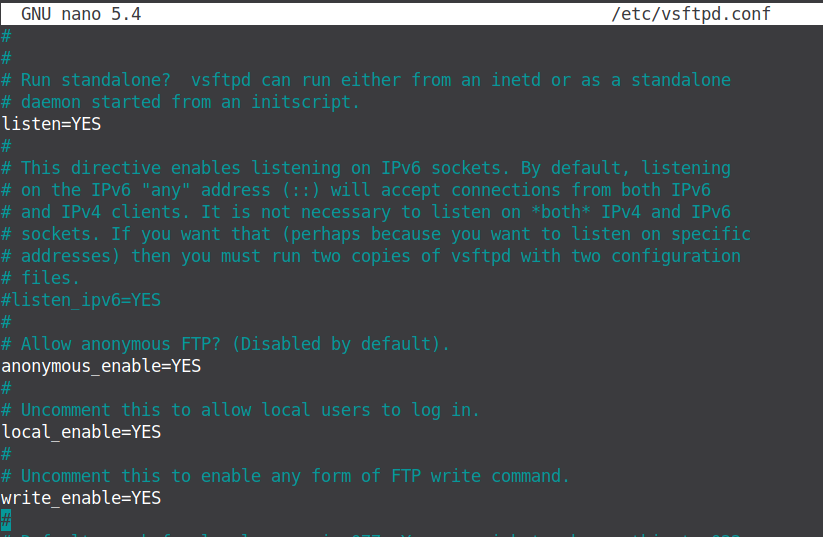
Et personnaliser notre site comme ci-dessous :



**Mise en place du serveur FTP :**

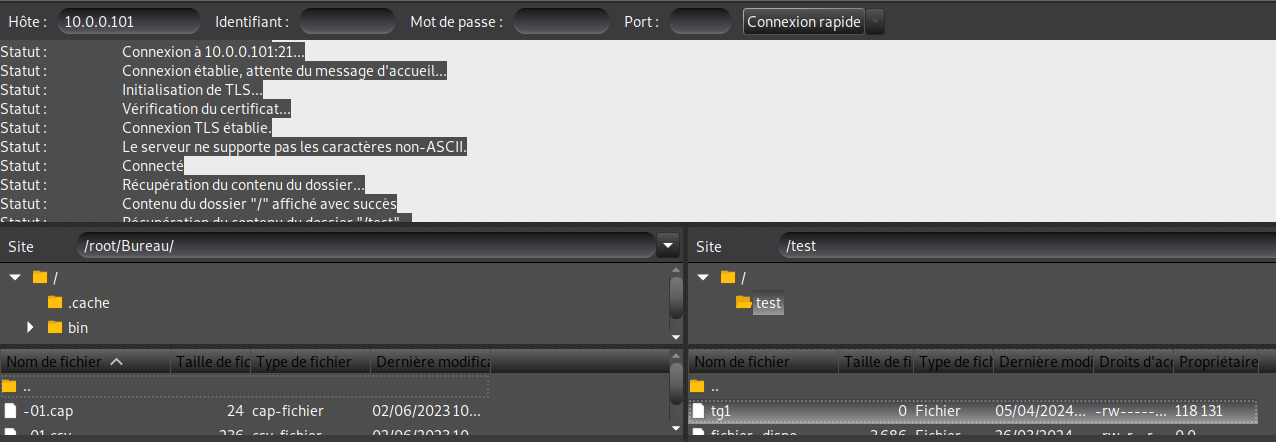
Pour commencer, il faut installer le paquet vsftpd puis activer le service.

Puis il faut aller dans les fichier de conf afin d'activer certaine fonctionnalités comme anonymous\_enable pour autoriser les connexions anonymes :

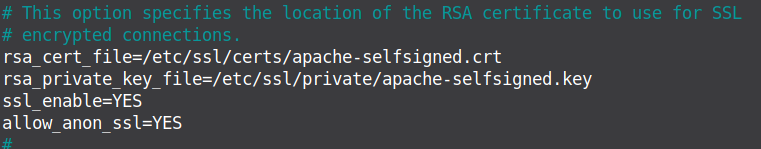


On peut voir ci-dessous qu'on peut se connecter au serveur ftp sur fizalle en ftp :

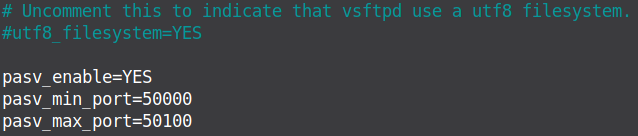
**Mise en place du serveur FTPS :**

****

Pour ftps, on va réutiliser les certificats générés pour https. On va éditer le fichier de conf en modifiant ces lignes en ajoutant le chemin de la clé privée, le certificat et en activant SSL comme ci-dessous :

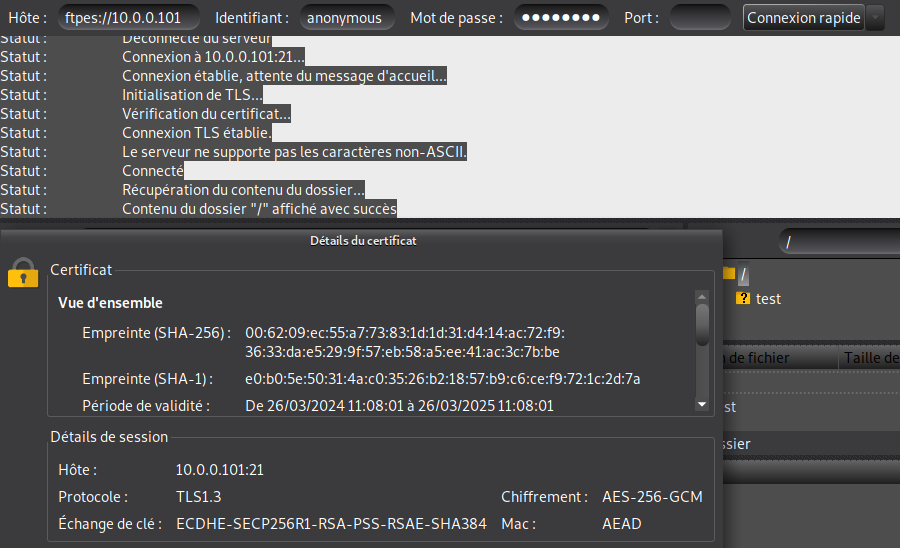


Il faut également ajouter une plage de port :



On ajoute une plage de port sur les firewall car le trafic est ensuite redirigé hors du port 21, j'ai donc fait une règle autorisant une plage de ports de 50000 à 50100 pour pouvoir continuer à communiquer avec le serveur après la validation du certificat.

Quand on se connecte au serveur en ftps avec filezilla, on peut consulter le certificat de sécurité et accéder aux fichiers du serveur :



**Tâche 4 Authentification transparent par certificat SSL (9 points)**

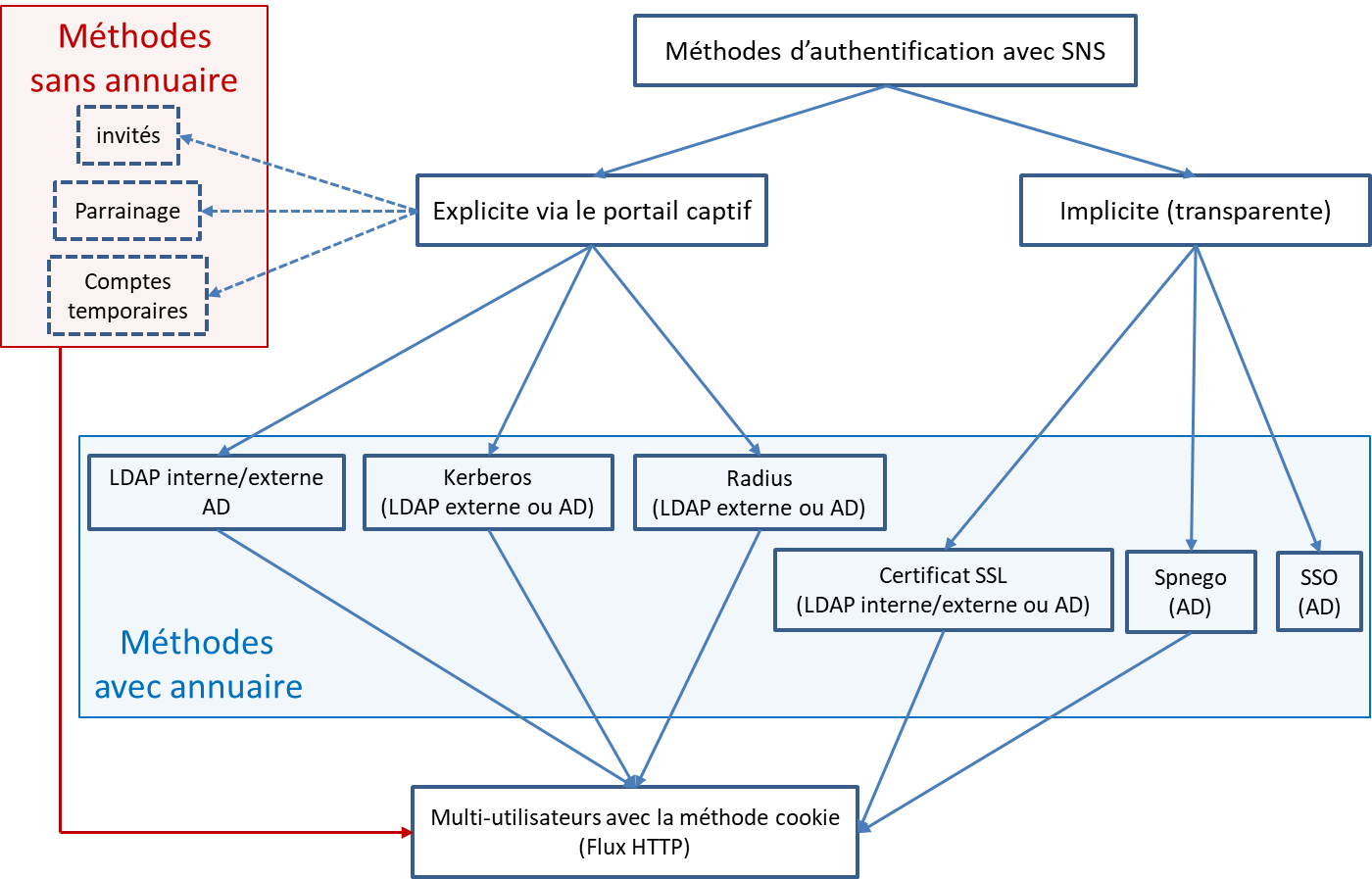
| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| PAQUELET Étienne  VADAM Julien | 33%  67% |

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

**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 |  |
| Activer l’authentification par certificat SSL |  |
| Importez le certificat dans le navigateur |  |
| Testez votre configuration |  |

**Rapport**

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

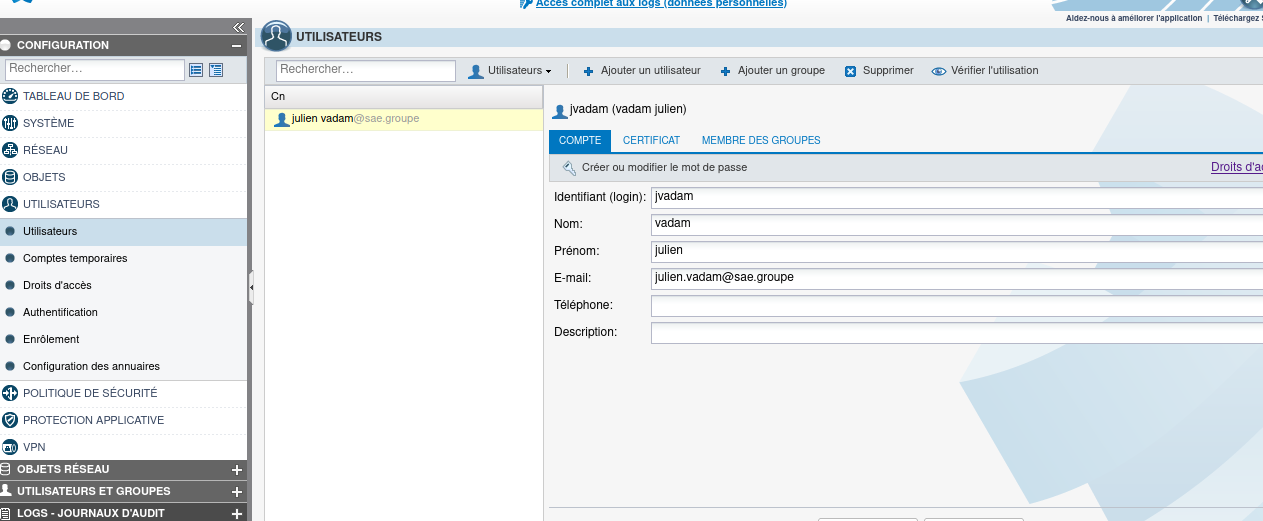
Le but de cette tâche est de réaliser une authentification sur un portail captif des utilisateurs du réseau interne qui veulent accéder au serveur externe.

Nous avons réalisé cette authentification sur le firewall B

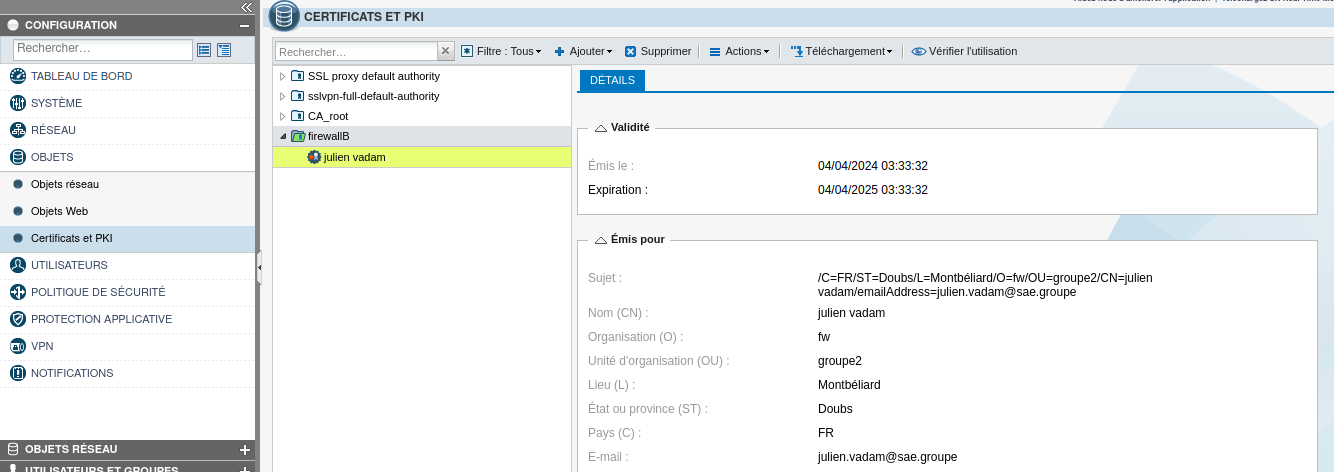
Afin d’activer cette authentification, il faut tout d’abord créer un annuaire LDAP interne :



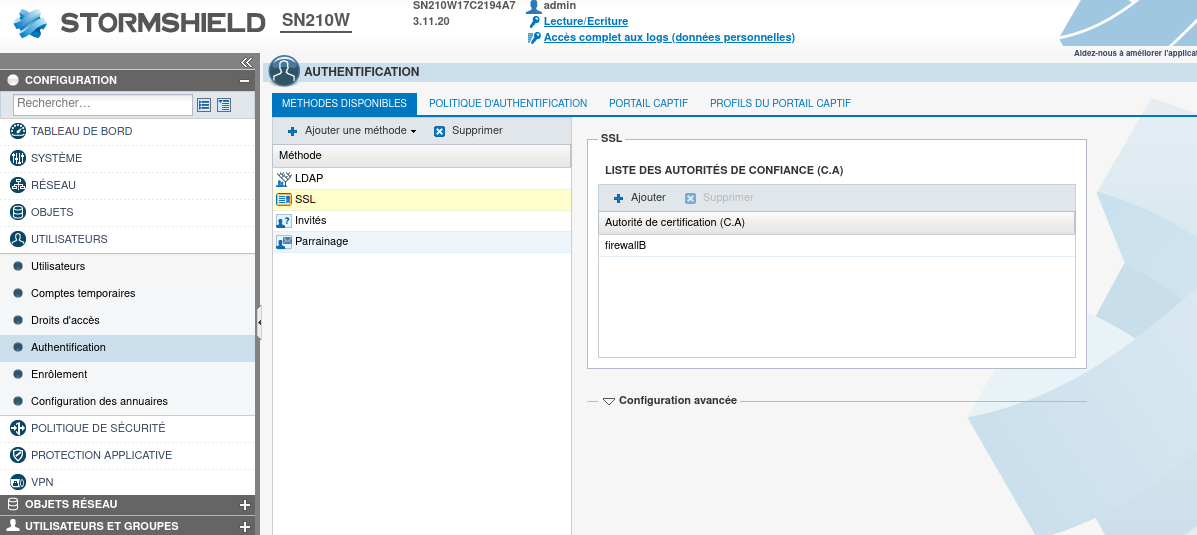
Une fois cela fait, nous avons créé dans cet annuaire un utilisateur :



Une fois créé, nous avons ensuite ajouté une autorité de certification racine (Root-Ca). Cette autorité signe son propre certificat ainsi que les certificats utilisateurs, machines… Suite à la création de cette Root-CA, nous avons créé le certificat utilisateurs de l’utilisateur : “julien vadam”.

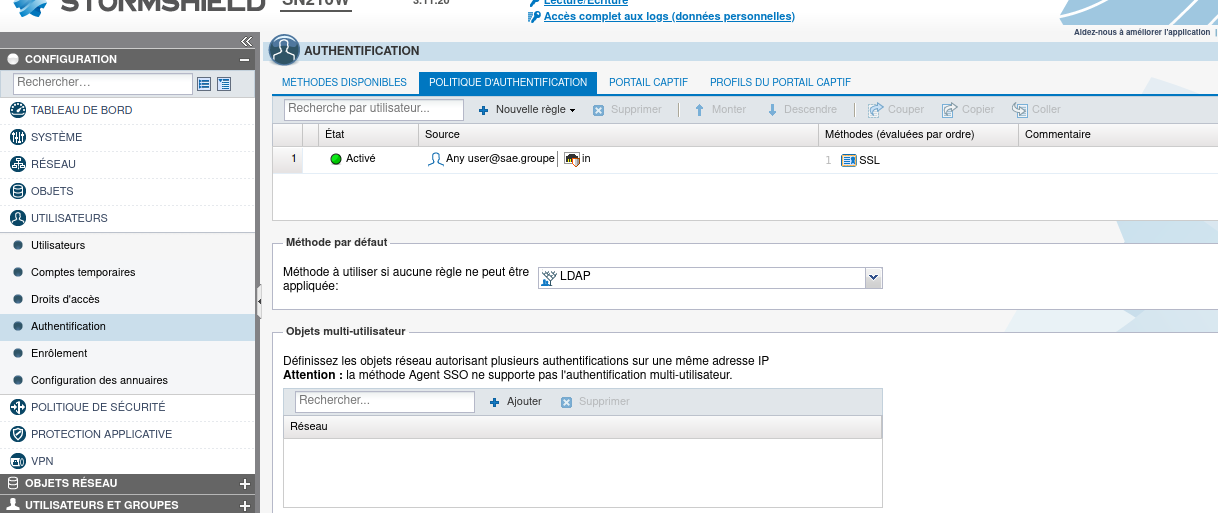
**

Nous avons modifié par la suite la méthode d’authentification dans le menu *Utilisateurs-> Authentification> Méthode d’authentification* en ajoutant la méthode SSL et en ajoutant en argument l’autorité de certification racine *firewallB*. Ainsi, pour chaque authentification par certificat, le stormshield ira demander à l’autorité de certification racine *firewallB* si le certificat utilisateur est valide.

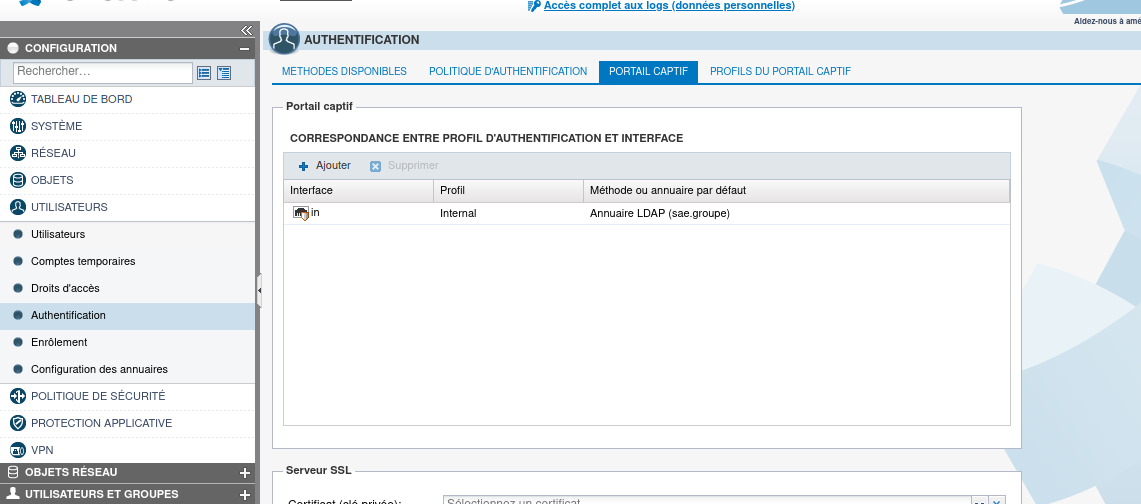


Ensuite, nous sommes allés dans le menu Politique d’Authentification et avons mis comme méthode d’authentification par défaut : SSL.

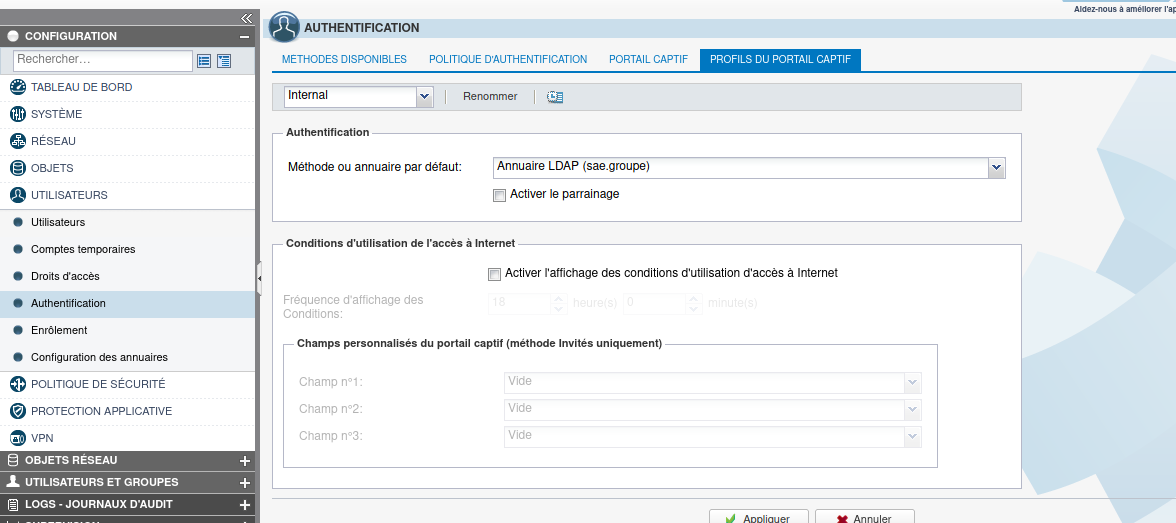
J’ai également ajouté 2 règles d’authentification : L’utilisateur authuser-x sur l’interface interne du pare feu doit s’authentifier via SSL afin d’accéder à internet :



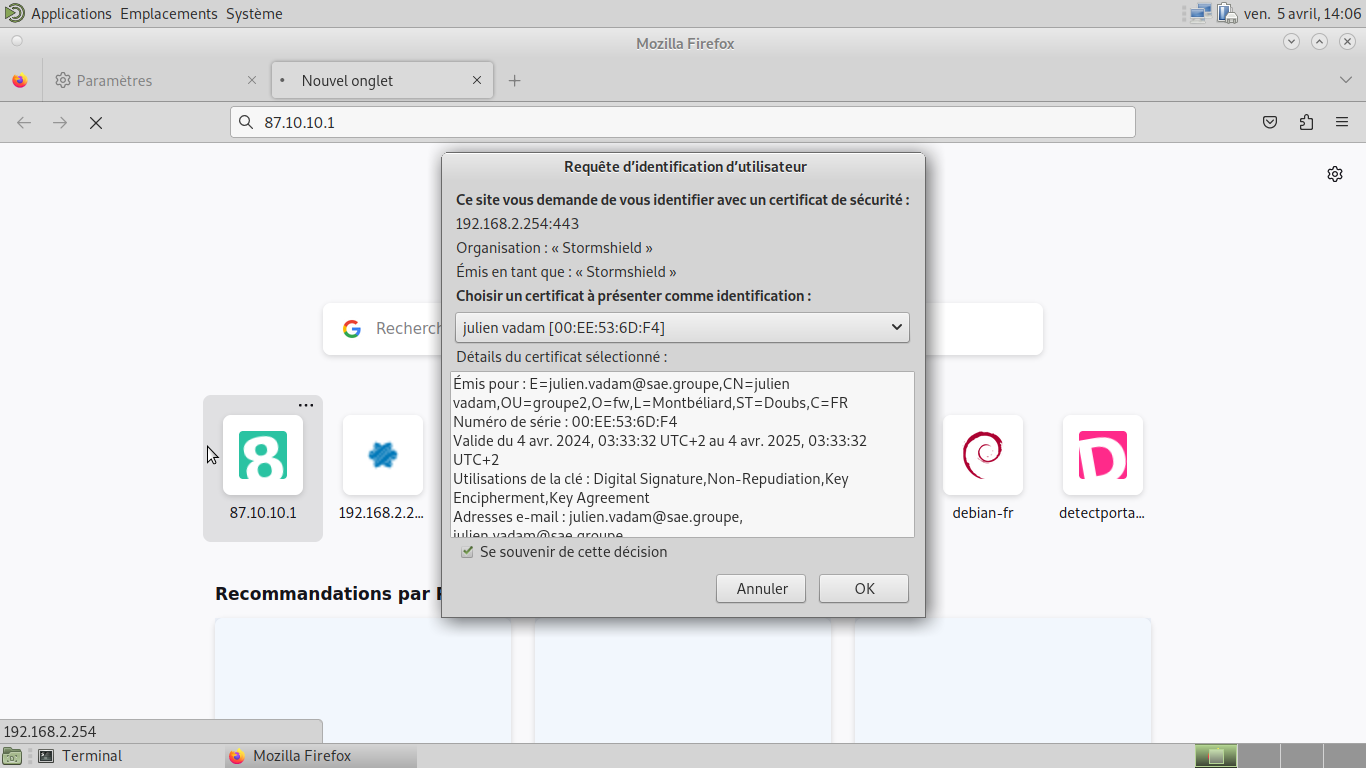
Dans l’onglet Portail Captif, nous avons ensuite ajouté une correspondance entre l’interface par laquelle les utilisateurs se connectent pour accéder à internet, soit l’interface interne du stormshield et le profil d’authentification “internal”



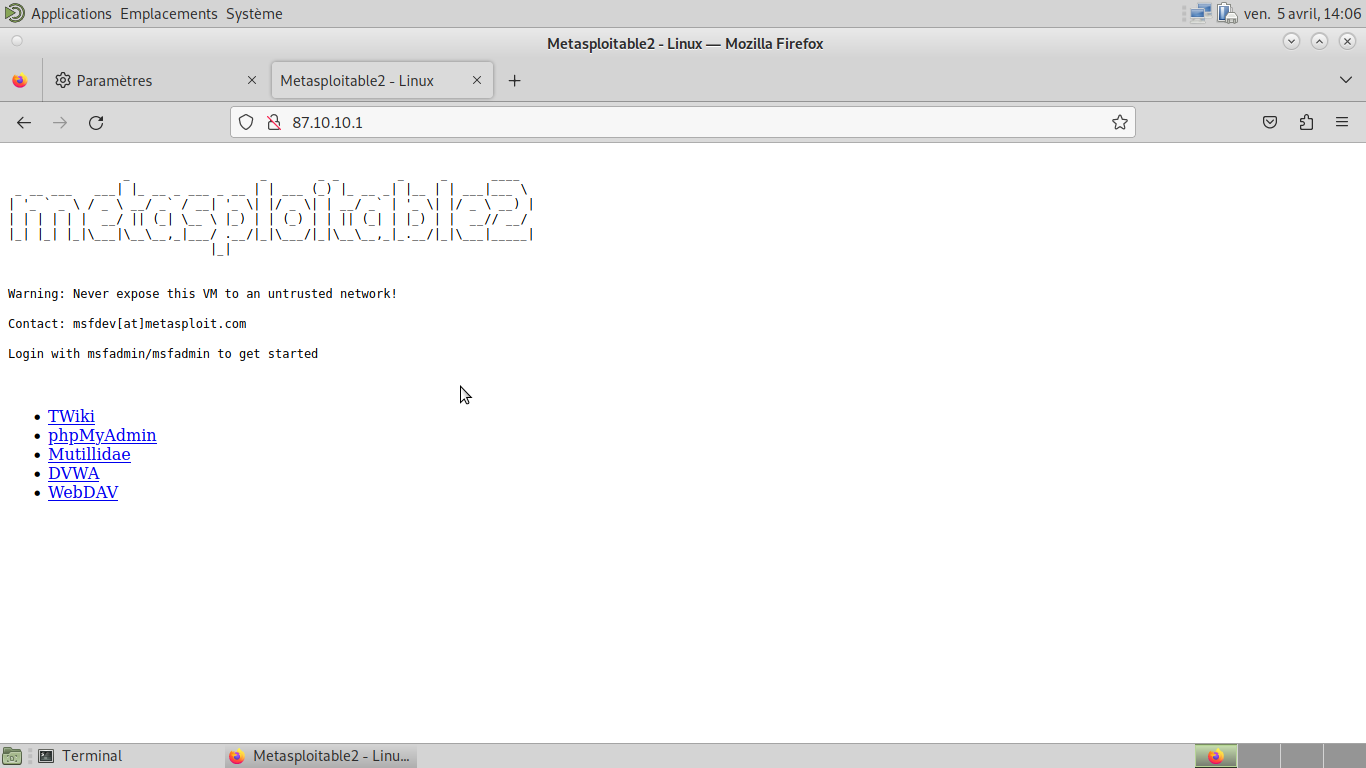
Une fois cela effectué, dans l’onglet Profil du portail captif, nous avons vérifié que la méthode par défaut où trouver les utilisateurs était l’Annuaire LDAP sae.sae



Une fois cela fait, nous avons créé une règle d’authentification dans *Politique de sécurité→ filtrage et NAT* afin que les utilisateurs non authentifiés soient redirigés vers un portail captif.

Nous pouvons vérifier que l’authentification par certificat fonctionne par cette fenêtre qui s’affiche lorsqu’on essaye d’accéder au site du pare feu A depuis le réseau interne du pare feu B : 

Une fois qu’on a choisi le certificat avec lequel s’authentifier, on peut accéder à la page librement :



**Tâche 5 Mettre en place un IDS et le tester (13,5 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| Gambier Maxence | 100% |

Estimation du temps passé sur cette tâche en heure-homme : 72H

**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 |  |
| Création des règles |  |
| Test des règles |  |

**Rapport**

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

**Étape 1) Installation et configuration de Snort**

Pour commencer la tâche 5, j’ai du installer le système de détection d’intrusion Snort. Celui-ci est système qui fonctionne avec beaucoup de paquet différents afin de pouvoir faire de la capture et de l’analyse de paquet.

Pour l’installation de snort, nous ne pouvons pas l’installer de manière simple comme la plupart des paquets car celui-ci a eu un changement de licence récemment et il a donc été retiré de la majorité des bibliothèques et notamment celle de Debian que l’on utilise.

J’ai donc suivi le tuto suivant afin d’installer et configurer Snort3 <https://www.howtoforge.com/install-and-configure-snort-3-on-ubuntu-22-04/#create-snort-custom-rules>

Pour résumé rapidement l’installation, celle-ci va être principalement basé avec la commande git qui permet de récupérer des fichiers sur github directement sur la machine.

Pour faire l’installation, il faut avant tout installé tout les paquets requis dont snort est dépendant

apt install build-essential libpcap-dev libpcre3-dev libnet1-dev zlib1g-dev luajit hwloc libdnet-dev libdumbnet-dev bison flex liblzma-dev openssl libssl-dev pkg-config libhwloc-dev cmake cpputest libsqlite3-dev uuid-dev libcmocka-dev libnetfilter-queue-dev libmnl-dev autotools-dev libluajit-5.1-dev libunwind-dev libfl-dev -y

Une fois cela fait, j’ai utilisé le tutoriel du lien plus haut et suivi les étapes une par une afin d’installer et configurer correctement Snort.

**Étape 2) Mise en place du port mirroring sur le switch**

Afin que snort analyse les paquets, il faut que les paquets se duplique vers la machine IDS.

Pour cela, j’ai mis en place du port mirroring sur le switch du réseau Internet. Celui-ci permettra de définir les 2 ports sources relié aux 2 stormshields et 1 port de destination sur celui relié à l’IDS.

Pour les ports sources, j’ai décidé d’y définir tout les ports sauf celui de l’IDS afin de pouvoir analyser tout les paquets mêmes si on rajoute une partie de réseau ou de nouvelles machines.

#Mise en place des ports sources#

Switch(config)# monitor session 1 source interface GigaEthernet2/0/1 – 4

Switch(config)# monitor session 1 source interface GigaEthernet2/0/6 – 24

#Mise en place des ports de destinations#

Switch(config)# monitor session 1 destination interface GigaEthernet2/0/5

Une fois cela mis en place, les paquets traversant le swicth seront automatiquement dupliqué et redirigé vers l’IDS.

**Étape 3) Écriture des règles et tests**

Pour que snort agisse selon la consigne, nous devons mettre en places des règles personnalisé en local. J’ai donc créer le fichier « local.rules » qui aura toutes mes règles personnalisés et qui sera utilisé quand on lancera snort.

Les règles sous snort sont sous un format en en 2 parties :

* En tête définissant l’action à faire quand un trafic correspond à la règle. On y retrouve l’action (alert dans nos cas), le protocole (majoritairement tcp), les plages d’adresses concernés, le port et l’opérateur de direction « -> »
* Options de la règle qui défini les actions comme le message à affiché dans snort, le sid (identifiant de règle), la priorité, le contenu de la trame, …

Voici la liste des règles ainsi que le/les test(s) allant avec chacune.

Pour vérifier chaque règle nous allons lancer Snort sur l’interface réseau utilisé sur l’IDS en lui précisant son fichier de configuration ainsi que les règles personnalisé qu’il utilisera.

[max@max](about:blank):~$ sudo snort -c /usr/local/etc/snort/snort.lua -R /usr/local/etc/rules/local.rules -i enp3s0 -A alert\_fast -s 65535 -k none

**A) Détection de connexion SSH**

Cette règle doit permettre la détection de connexion SSH depuis l’extérieur du réseau. Pour cela, nous allons tenter une connexion SSH sur une interface de notre stormshield depuis un pc attaquant dans le réseau B.

[root@hack](about:blank):~# ssh [admin@87.10.10.1](about:blank)

La règle de détection est formé de cette manière :

#SSH

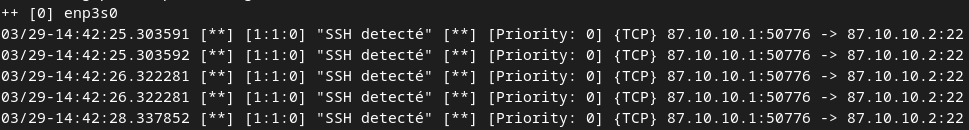
alert tcp any any → $HOME\_NET 22 (

msg:“SSH détecté” ;

sid:1 ;

)

Cette règle analyse les paquets de n’importe qui allant dans le réseau Internet sur le port 22 et renvoie donc une alerte avec les options de règle qu’on lui a attribuées.



**B) Détection d’attaque DOS sur le serveur web avec des HTTP GET**

Cette règle doit permettre la détection d’attaque DoS sur notre serveur web avec des HTTP GET. Pour cela, nous avons été sur la page depuis un pc attaquant sur le réseau B et nous avons rechargé la page web rapidement sans enregistrement de cache afin de créer assez rapidement des requêtes « HTTP GET ».

La règle de détection est formé de cette manière :

#DoS HTTP GET

alert tcp any any → $HOME\_NET $HTTP\_PORTS (

msg:“DoS GET HTTP détecté” ;

http\_method ;

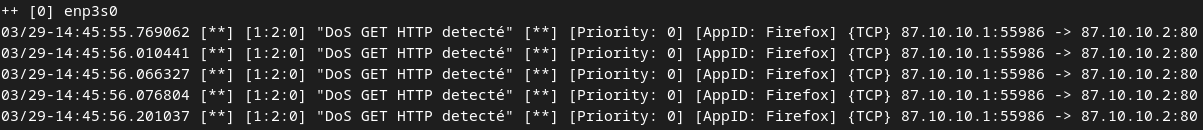
content :“GET” ;

detection\_filter:track by\_src, count 5, seconds 10 ;

sid:2 ;

)

Cette règle analyse les paquets de n’importe qui en direction de nos stormshield sur les ports HTTP et renvoie donc une alerte avec les options msg et sid qu’on lui a attribué. En option supplémentaire on a « http\_method » qui permet de créer une alert qui se produit lors d’une requête http, « content :“GET” » qui va préciser le contenu de la requête HTTP et « detection\_filter:track by\_src, count 5, seconds 10 » qui est un filtre de détection précisant qu’il faut suivre les connexions par source et de déclencher l’alerte si on reçoit au moins 5 paquets correspondant à la règle en l’espace de 10 secondes.



**C) Détection des URIs non-normalisées**

Cette règle doit permettre la détection des URIs non-normalisées. Pour cela, nous avons simulé un contournement des règles en place en essayant d’accéder au serveur web en rajoutant « admin » et « users » à la fin de l’URL.

[root@hack](about:blank):~# curl -X GET “<http://87.10.10.2/admin>”

<!DOCTYPE HTML PUBLIC “-//IETF//DTD HTML 2.0//EN”>

<html><head>

<title>404 Not Found</title>

</head><body>

<h1>Not Found</h1>

<p>The requested URL was not found on this server.</p>

<hr>

<address>Apache/2.4.38 (Debian) Server at 87.10.10.2 Port 80</address>

</body></html>

[root@hack](about:blank):~# curl -X GET “<http://87.10.10.2/users>”

<!DOCTYPE HTML PUBLIC “-//IETF//DTD HTML 2.0//EN”>

<html><head>

<title>404 Not Found</title>

</head><body>

<h1>Not Found</h1>

<p>The requested URL was not found on this server.</p>

<hr>

<address>Apache/2.4.38 (Debian) Server at 87.10.10.2 Port 80</address>

</body></html>

La règle de détection est formé de cette manière :

#URIs non normalisées

alert tcp any any → $HOME\_NET $HTTP\_PORTS (

msg :“URI non normalisée détecté”

http\_raw\_uri ;

content :“/admin”,nocase ;

sid:3;

)

alert tcp any any → $HOME\_NET $HTTP\_PORTS (

msg :“URI non normalisée détecté”

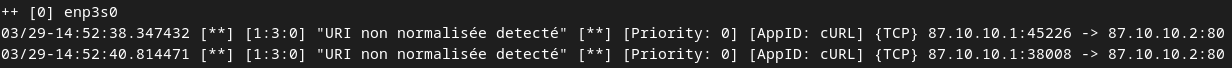
http\_raw\_uri ;

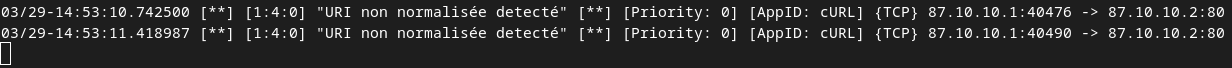
content :“/users”,nocase ;

sid:4;

)

Cette règle analyse les paquets de n’importe qui en direction de nos stormshield sur les ports HTTP et renvoie donc une alerte avec les options msg et sid qu’on lui a attribué. En option supplémentaire, on a « http\_raw\_uri » qui spécifie que snort analysera l’URI dans les paquets HTTP et « content :“/users”,nocase » qui va vérifier si le contenu de l’URI contient « /users » (idem avec /admin) et le nocase veut dire qu’il effectuera cela peu importe les majuscules ou minuscules de « users » ou « admin ».





*On différencie les 2 règles par rapport à leurs sid : 3 pour /admin et 4 pour /users.*

**D) Détection d’un login raté sur le serveur FTP**

Cette règle doit permettre la détection des login raté lors d’une connexion FTP. Pour cela, nous allons nous connecter en ftp au serveur FTP depuis le client B en faisant une erreur volontaire de login depuis filezilla

La règle de détection est formé comme celle-ci :

#Login raté FTP

alert tcp any any → any any (

msg :“FTP Login raté” ;

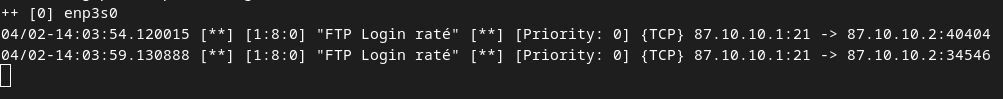
service:ftp ;

content :“530”, nocase ;

sid:8 ;

)

Cette règle analyse les paquets venant de n’importe qui en direction de n’importe qui et renvoie donc une alerte. En option supplémentaire, on a « service:ftp » qui permet de définir le service sur lequel la règle se fait et « content :“530”, nocase » vient définir la règle sur les paquet ftp qui contienne « 530 » qui est le numéro d’erreur d’un login raté.



**E) Détection d’une attaque DoS avec TCP SYN**

Cette règle permet la détection d’une attaque DoS avec TCP SYN qui consiste à envoyer un grand nombre de paquet TCP SYN à un serveur pour le rendre indisponible. Pour cela, nous allons envoyer un grand nombre de paquet TCP SYN à l’aide de la commande hping3 vers le serveur Web B avec une machine attaquante dans le réseau A sur le port 80 avec une intervalle de 1sec entre chaque paquet.

[root@hack](about:blank):~# hping3 87.10.10.2 –syn -p 80 -i 1

La règle est formé comme celle-ci :

#DoS TCP SYN

alert tcp any any → $HOME\_NET any (

msg :“DoS TCP SYN détecté” ;

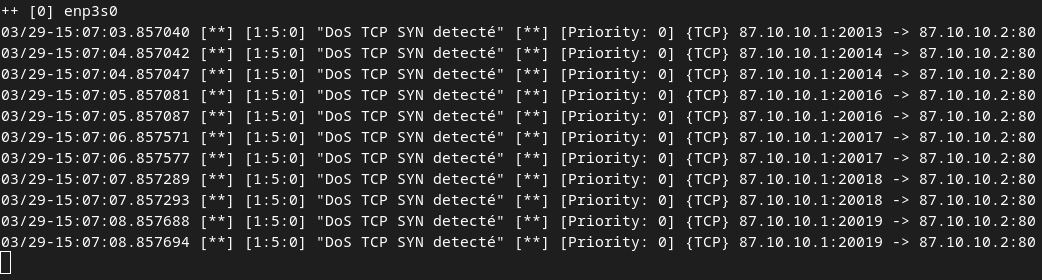
flags:S ;

detection\_filter : track by\_src, count 5, seconds 10 ;

sid:5 ;

)

Cette règle analyse les paquets venant de n’importe qui en direction des stormshield et renvoie donc une alerte. En option supplémentaire, on a « flags:s » qui cible la règle sur les paquets de type TCP SYN.



**F) Détection de paquet fragmenté de taille < 500 et > 1000**

Cette règle permet la détection de paquet fragmenté ayant une taille inférieur à 500 et ceux supérieurs à 1000. Pour cela, nous allons envoyer des paquets fragmenté de différentes tailles avec la commande hping3.

[root@client](about:blank):~# hping3 -c 87.10.10.1 –data 400 -f

La règle est formé comme celle-ci :

#Paquets fragmentés

alert ip any any → $HOME\_NET any (

msg :“Paquet fragmeté détecté” ;

fragbits:M ;

dsize :<500 ;

sid:6;

)

alert ip any any → $HOME\_NET any (

msg :“Paquet fragmeté détecté” ;

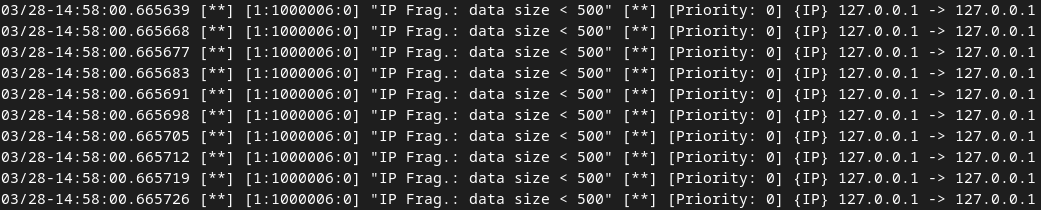
fragbits:M ;

dsize :>1000;

sid:7;

)

Cette règle analyse les paquets ip venant de n’importe qui en direction de nos stormshield. En option supplémentaire, on a « fragbits:M » qui vient dire qu’il faut analyser les paquets fragmenté et « dsize » qui précise la taille du paquet.

**

**Étape 4) Problèmes et options supplémentaire**

Durant la tâche, j’ai eu différents problème :

* Paquet FTP : Snort ne détecte pas toujours les paquets FTP car le serveur ne renvoi pas toujours la même erreur (530 Login failed ou 500 OOPS)
* Paquet fragmenté : Lors des envoie de paquet fragmenté, snort ne les détecte pas car ils sont bloqués par les Stormshields. Cependant, il est possible d’utilisé une capture wireshark prise en local pour la faire analyser dans snort.

J’ai également mis en place un fichier de log « alert\_fast.txt » afin de pouvoir démarrer snort en arrière plan et éviter les latences de terminal.

| snort -c /usr/local/etc/snort/snort.lua -i enp3s0 -l . -D |
| --- |

**Tâche 6 Attaque sur le Wifi (4,5 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| VADAM Julien | 100% |

Estimation du temps passé sur cette tâche en heure-homme : 24H

**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 |  |
| Cassage de la clé WEP sur Linksys |  |
| Mise en place du WPA sur Linksys |  |
| Cassage du WPA sur Linksys |  |
| Mise en place du WEP sur Stormshield SNS |  |
| Cassage de la clé WEP sur Stormshield SNS |  |
| Mise en place du WPA sur Stormshield SNS |  |
| Cassage du WPA sur Stormshield SNS |  |

**Rapport**

1. **Crackage WEP sur Linksys :**

Afin de réaliser cette tâche, j’ai tout d’abord mis en place le routeur Linksys sur le réseau de la DMZ A, j’ai relié le Firewall à un switch, enfin j’ai relié le Linksys (sur les ports LAN) au switch.

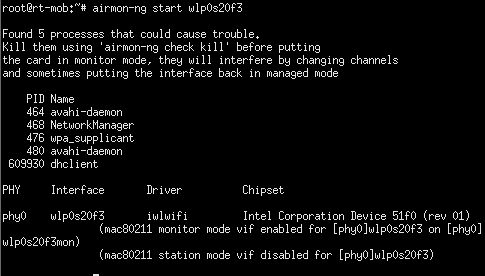
J’ai ensuite modifié la configuration du Linksys afin de lui mettre l’adressage correspondant en 10.0.0.112 et j’ai également mis le routeur en passerelle DHCP (pour obtenir une addresse IP du port DMZ du firewall).

Ensuite, j’ai mis la sécurité du routeur en WEP avec la phrase de passe « 1234 » puis j’ai généré les clés.

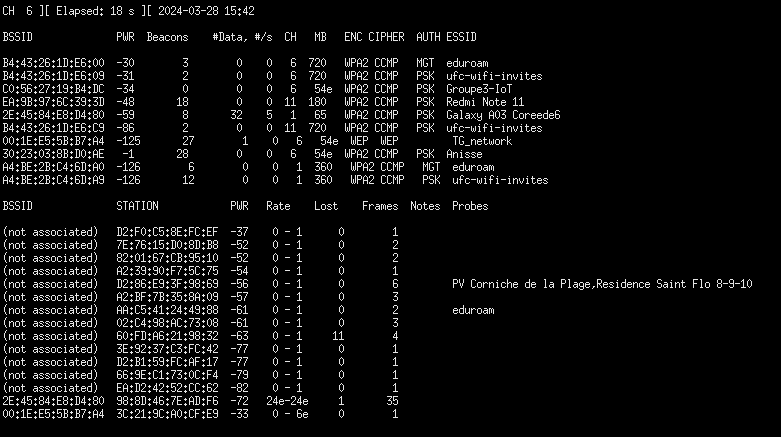
Pour craquer ce mot de passe, j’ai d’abord connecté un PC portable sur le réseau du routeur afin de pouvoir analyser les échanges de données (ça sera la victime).

J’ai donc utiliser le logiciel « aircrack-ng » pour réaliser ces crackages.

Tout d’abord j’ai dû activer la carte wifi en mode moniteur en faisant la commande :

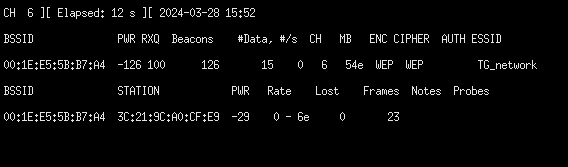


Ensuite j’ai fait la commande suivante en renseignant l’interface utilisée pour voir les réseaux à proximité “***airodump-ng wlp0s20f3mon”*** :

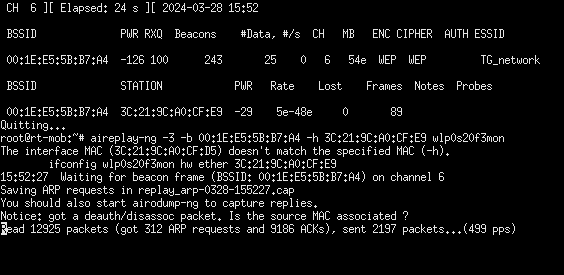


Nous pouvons voir les réseaux wifi à proximité ainsi que leur algorithme de chiffrement.

J’ai donc choisi le réseau que nous voulions cracker avec son BSSID en faisant la commande suivante, les captures réseaux sont ainsi enregistrées localement en vue d’être crackées plus tard grâce à l’option -w : ”***airodump-ng -c 6 –bssid 00:1E:E5:5B:B7:A4 -w /root/Documents/captures/***” :



J’ai ensuite dû attendre que la quantité de data augmente, en utilisant la commande aireplay, j’ai donc effectué un DOS de requêtes ARP enfin d’augmenter les échanges également pour accélérer l’attaque :

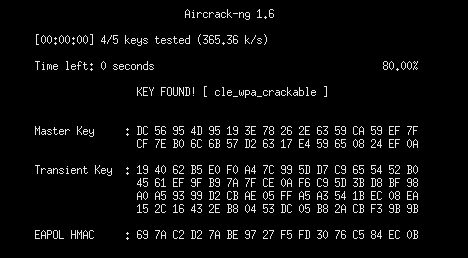


Après un certain temps j’ai récupéré la clé WEP recherchée en utilisant la commande ***aircrack-ng <fichier\_capture>*** :

# 

# Crackage WPA sur Linksys :

Pour cracker la clé pré-partagée WPA, j’ai effectué exactement le même protocole sauf que la dernière commande « aircrack-ng » avait besoin d’un dictionnaire, j’en ai donc créé un petit pour la forme car rockyou.txt ne contenait pas ce mot de passe, voici le clé trouvée par le logiciel :

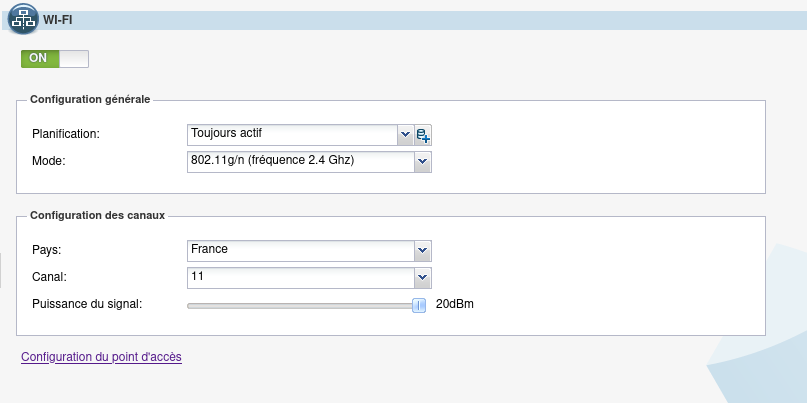


En temps normal, le dictionnaire est beaucoup plus gros et le temps pour cracker le mot de passe est beaucoup plus long, si jamais le mot de passe n’est pas dans le dictionnaire, l’attaque échoue.

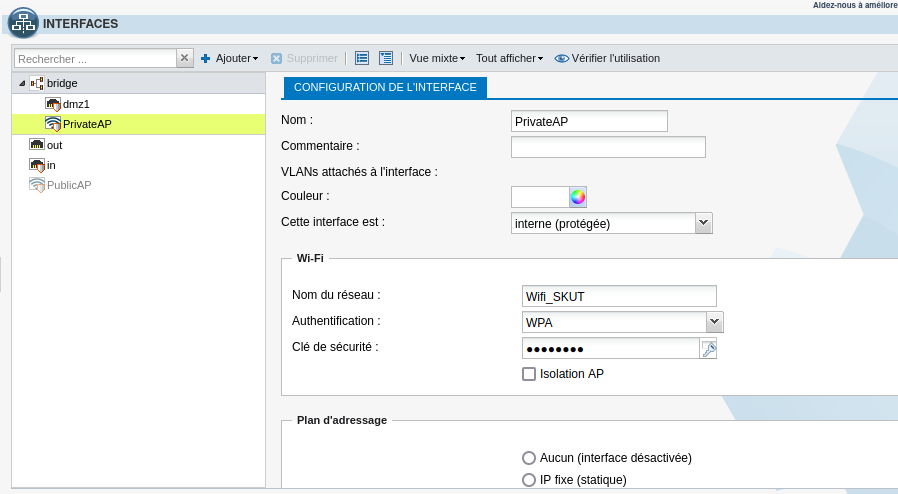
# Crackage WPA2 sur Stormshield :

***PS :*** Sur le Stormshield il n’y a pas de chiffrement WEP ; le Stormshield n’active également pas le chiffrement WPA correctement (il est affiché en chiffrement WPA2 dans aircrack-ng).

J’ai tout d’abord mis en place le wifi sur le Stormshield :



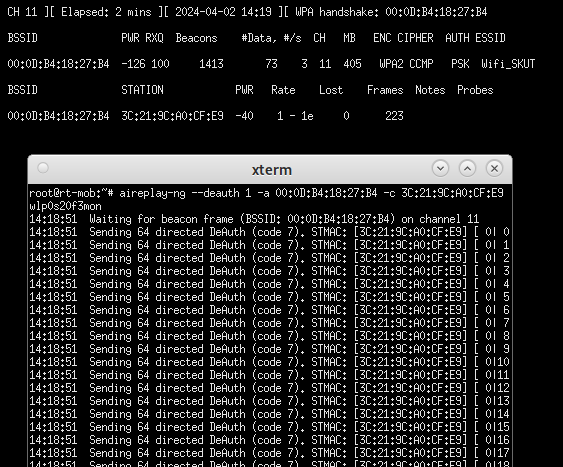
Puis j’ai mis l’interface wifi dans le bridge avec la protection WPA(2) :



Puis j’enregistre les captures dans un dossier :



On déconnecte ensuite le client pour obtenir le handshake grâce à airplay :



Il suffit ensuite de cracker la clé avec un dictionnaire mis à disposition avec la commande ***aircrack-ng <capture> -w <dictionnaire>*** :



**Tâche 7 Utilisation de scanneurs de vulnérabilité (13,5 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| PAQUELET Etienne | 100% |

Estimation du temps passé sur cette tâche en heure-homme : 39H

**Objectif : Réaliser plusieurs évaluations de la sécurité des serveurs**

| Sous-tâches | Evaluation prof |
| --- | --- |
| Installez dans la DMZ une machine/VM metasploitable |  |
| Installez et utilisez SCNR |  |
| Installez et utilisez Legion |  |
| Installez et utilisez Nuclei |  |
| Installez et utilisez Nikto |  |
| Placez les scanners dans la DMZ, puis à l’extérieur |  |

**Rapport**

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

**Legion** : Ce scanner ne m’a pas posé problème, il a été simple d’utilisation puisqu’il est présent nativement dans kali. Il scanne les vulnérabilités des services ainsi que les ports de la machine ciblée.

Son utilisation ressemble fortement à celle de nmap puisque son interface graphique ressemble à celle de zenmap.

**Nuclei** : Aucun problème d’utilisation ni d’installation.

Son utilisation est simpliste : *nuclei -u [host targeted]*

Il réalise plus particulièrement les analyses web mais scanne également les vulnérabilités/failles de la machine.

**Nikto** : Présent également nativement dans kali, il ne m’a pas posé de problèmes pour son

utilisation. Son utilisation est également simple : *nikto -host [host targeted]*

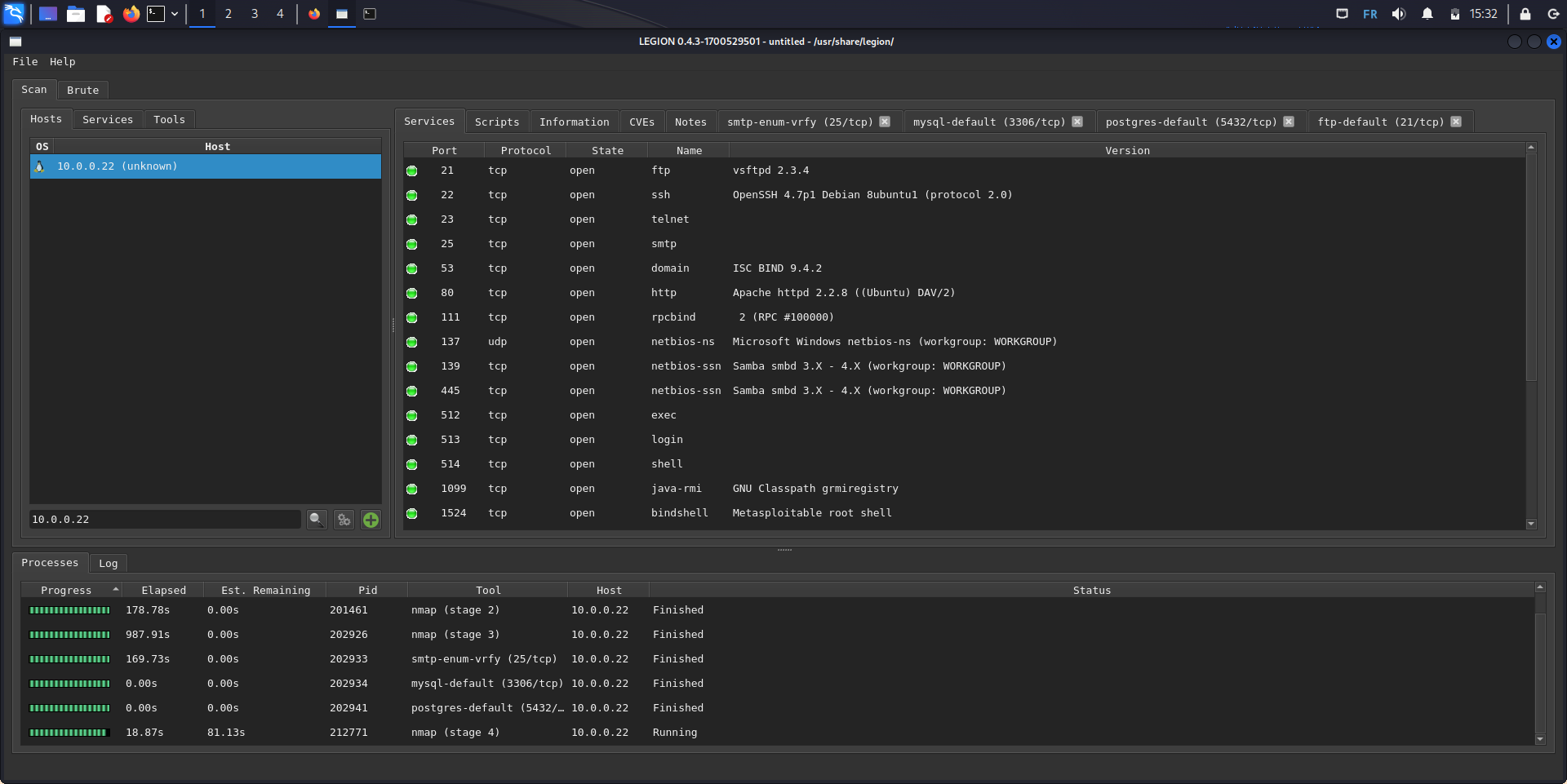
Nikto réalise principalement les scans web et XSS (Cross Site Scripting). Il m’a résumé les failles et informations qu’il a trouvé sur la cible.

**SCNR** : Ce scanner a été quelque peu plus problématique au niveau de l’installation. J’ai dû créer une nouvelle VM Kali puisqu’il refusait de s’installer sur la première VM Kali. De plus, j’ai utilisé un script fourni sur github afin de l’installer plus facilement que montré sur le site officiel.

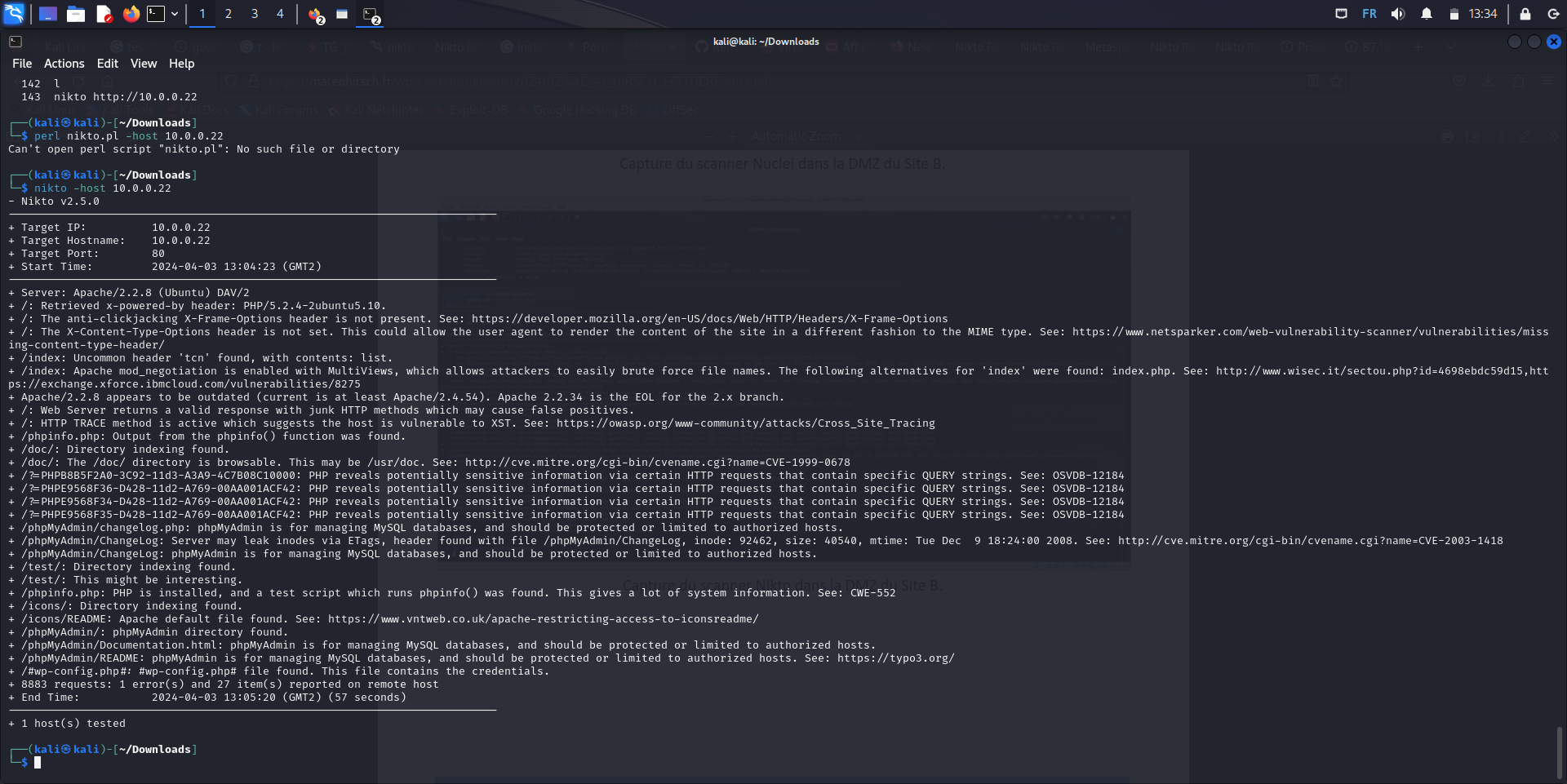
Son utilisation est plutôt simple : *./scnr-v4.1/bin/scnr [host targeted]*

Cependant, vu que cet outil est très « curieux », il scan chaque page et ainsi, le scan est long (5 heures environ pour le scan interne et 7h pour le scan externe), mais fournit beaucoup d'informations sur chaque vulnérabilité découverte sur la machine cible dans le rapport.

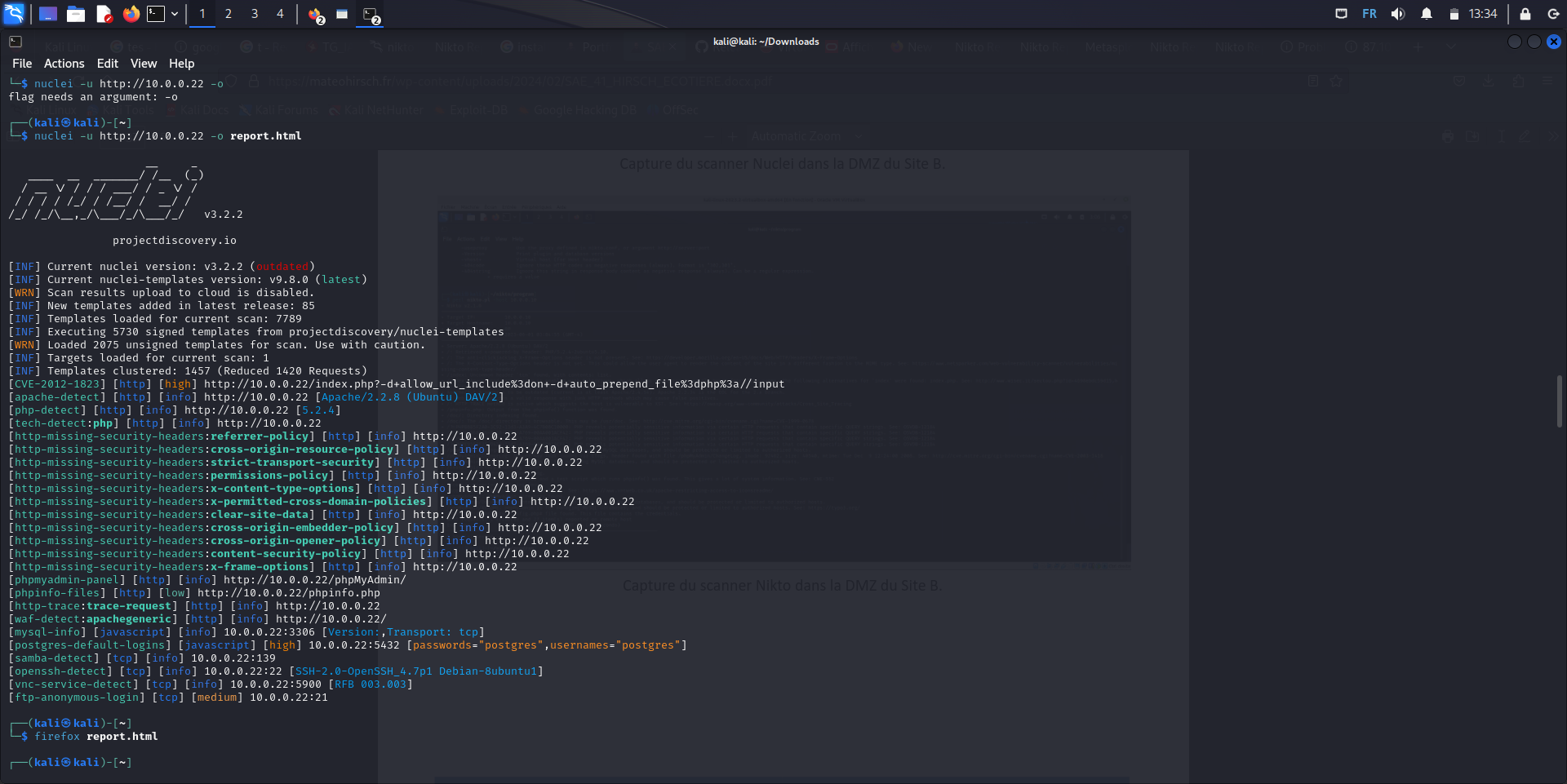
J’ai tout d’abord configuré la VM Metasploitable en bridge sur l’interface DMZ du pare-feu A. Puis, j’ai effectué un scan interne en utilisant les scanner Légion, Nuclei, Nikto et Codename SCNR. Il n’y a pas eu trop de difficultés à installer ces outils hormis pour SCNR. Ensuite, j’ai scanné cette même machine depuis l’extérieur du réseau, c'est-à-dire, depuis « internet ». J’ai pu remarquer qu’il y avait une différence entre le scan interne et externe. Cela peut s’expliquer par le pare-feu qui bloque et ou ralentit via le filtrage certains scans depuis l’extérieur. Ce pare-feu rend les attaques plus difficiles pour les attaquants vu que via le filtrage, les recherches lors de la phase d’énumération sont ralenties.



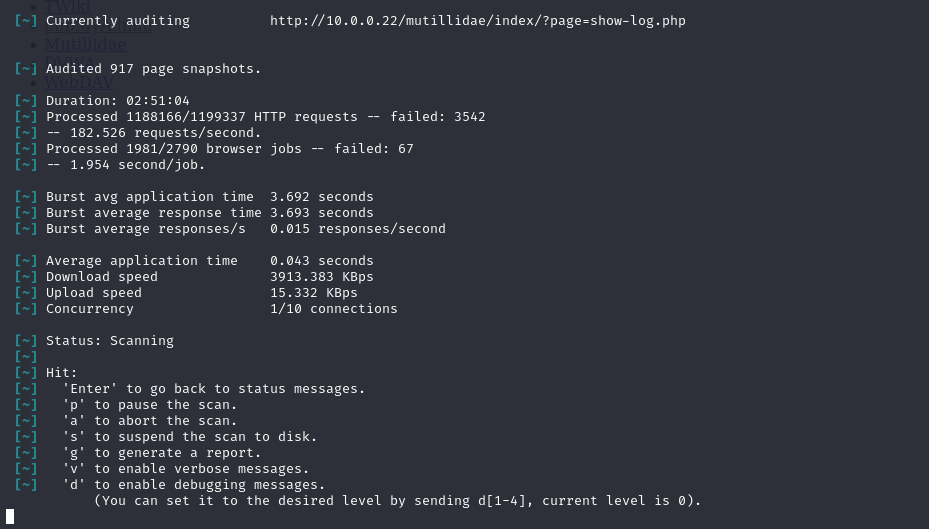
*Scan Legion dans la DMZ*



*Scan nikto dans la DMZ*

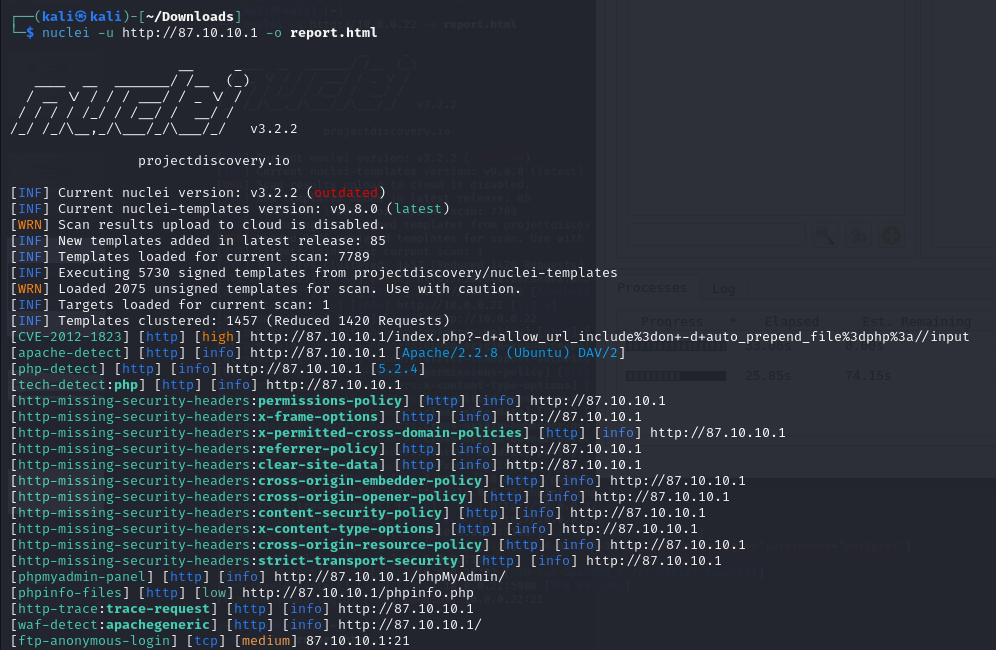
**

*Scan nuclei dans la DMZ*

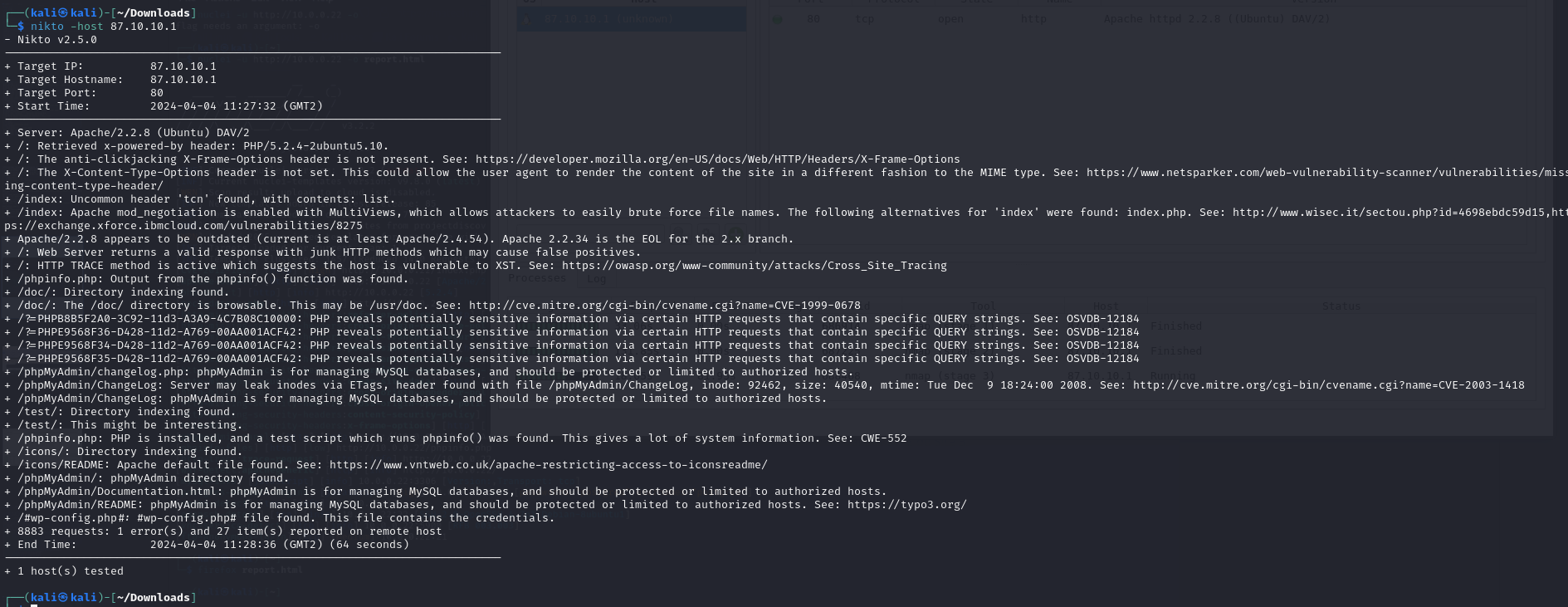
****

*Scan SCNR depuis la DMZ*

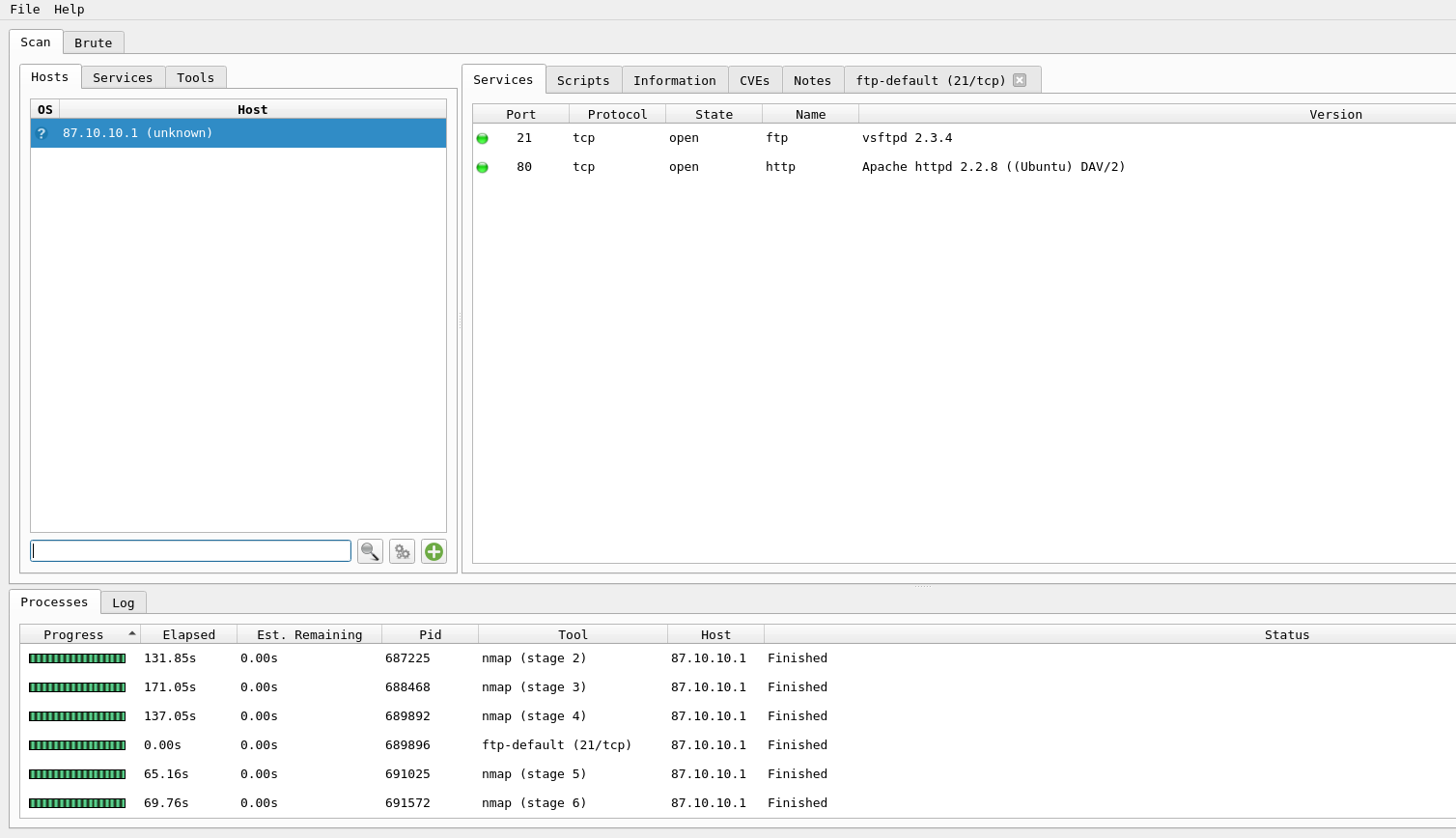
**Scan externe sur 87.10.10.1**

****

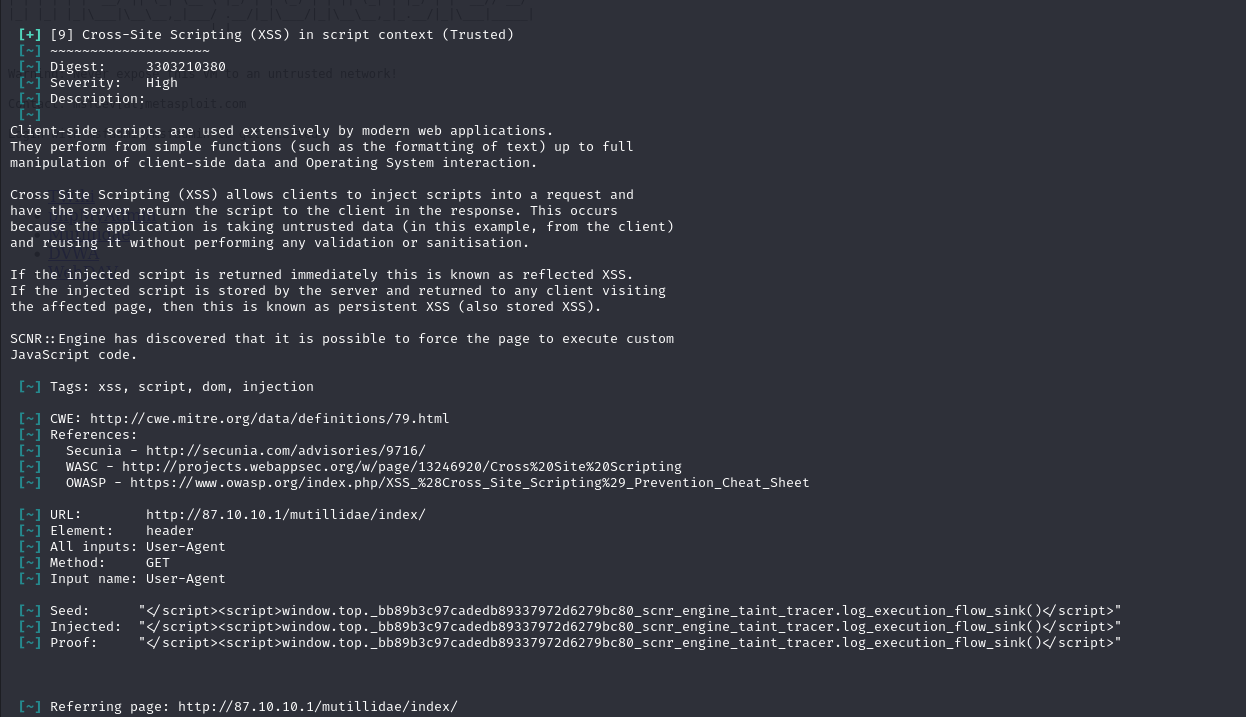
*Scan nuclei depuis “internet”*

**

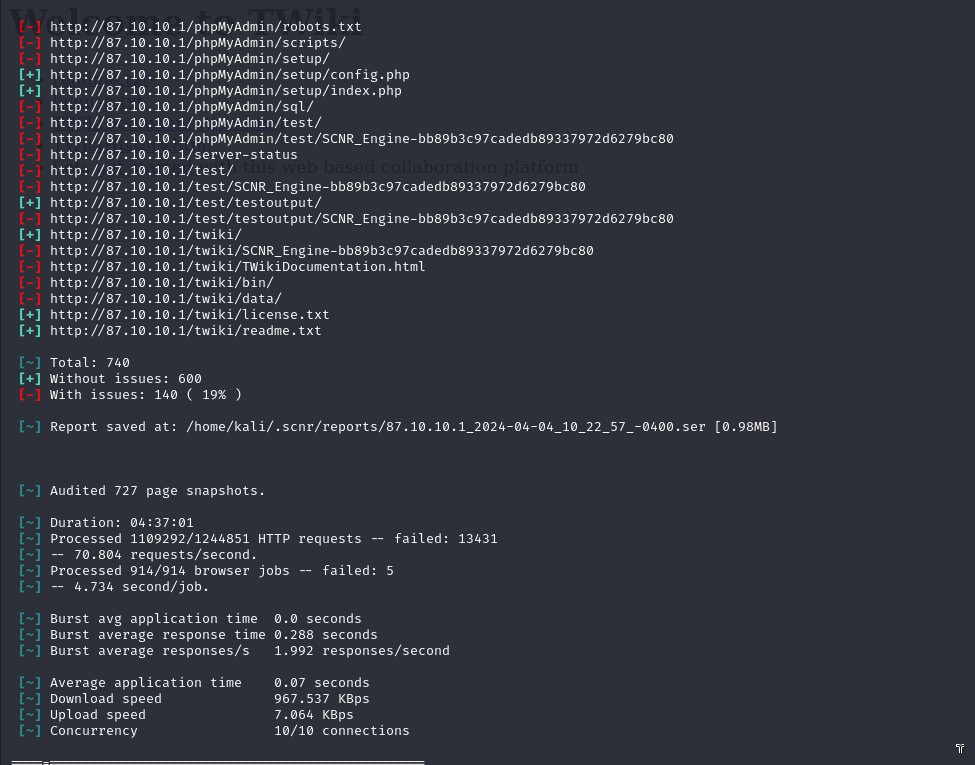
*Scan nikto depuis “internet”*

**

*Scan Legion depuis “internet”*

**

*Scan SCNR Codename depuis “internet”*

**

*Scan SCNR Codename depuis “internet”*

**Tâche 8 Réalisation d’une attaque MiM (3,75 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| VADAM Julien  WITTIG Antonin | 50%  50% |

Estimation du temps passé sur cette tâche en heure-homme : 5H

**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 de s’apercevra plus qu’il y a le pirate entre lui et le serveur.

Modifiez des données de la page HTML envoyée au client.

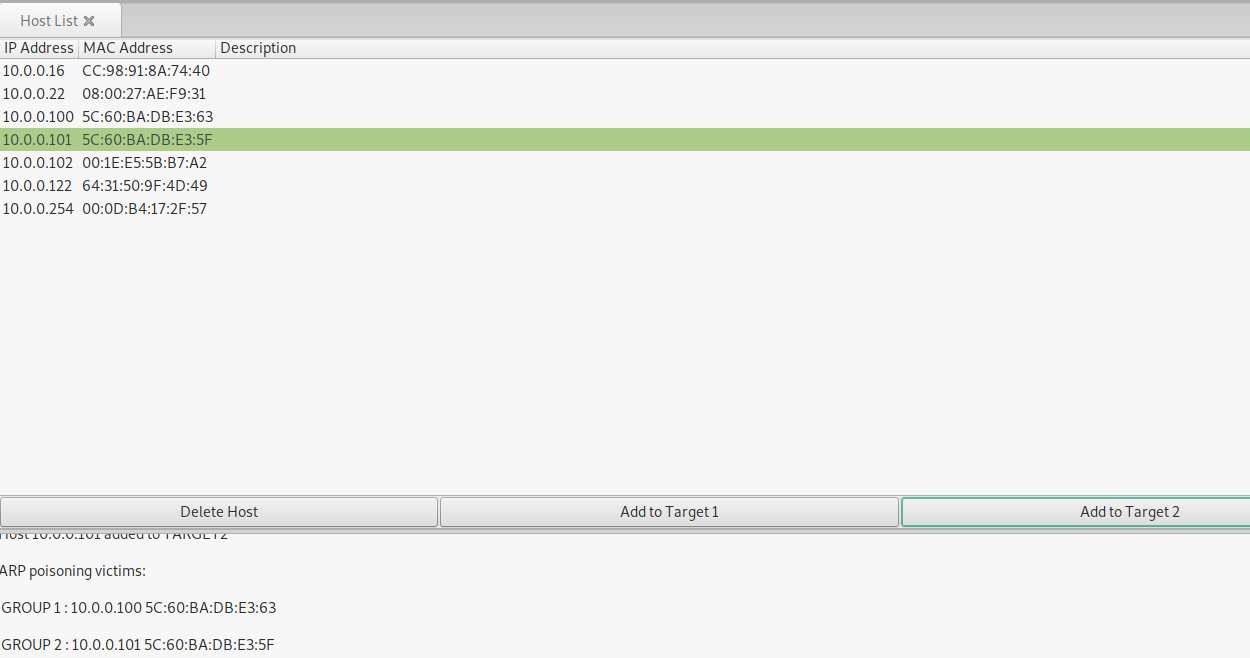
Vous devrez utiliser une application pouvant forger des paquets ARP comme Scapy ou Arp-sk par exemple.

| Sous-tâches | Evaluation prof |
| --- | --- |
| Installez un forgeur de paquets ARP |  |
| Usurpation de l’adresse ARP du serveur |  |
| Redirection de la connexion |  |
| Modification des données HTML |  |

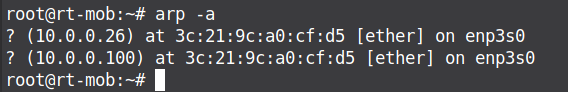
**Rapport**

Afin de réaliser cette tâche, nous avons tout d’abord installé ettercap sur le PC du hacker afin de faire de l’ARP poisonning.

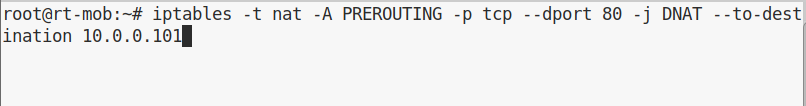
Pour corrompre les tables ARP, nous avons tout d’abord choisi nos deux victimes sur ettercap (nous allons donc nous incruster dans leur communications) :



Nous pouvons voir que la table ARP du serveur a été modifié :

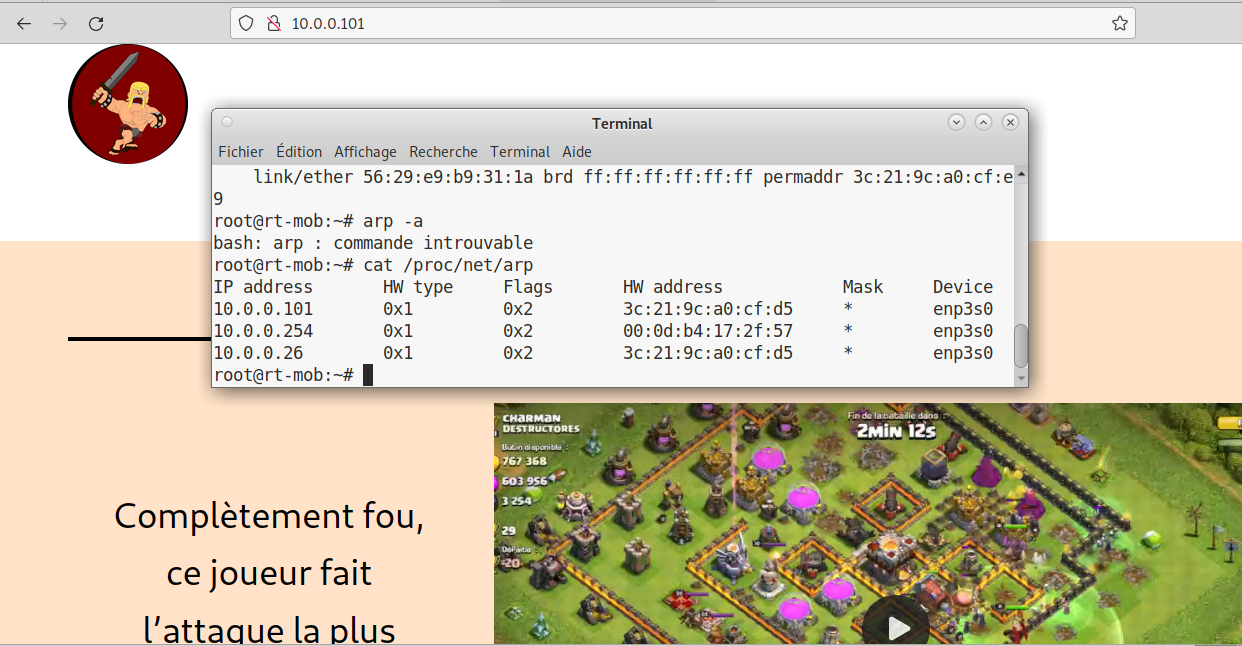


Pour activer le routage et donc assurer la communication entre le client et le serveur, nous devons exécuter la commande suivante :

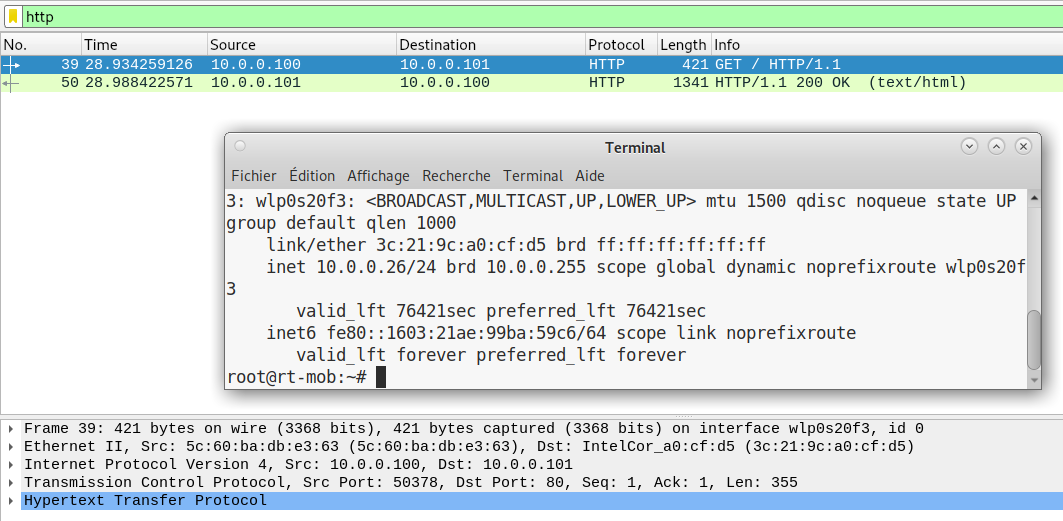


Cette commande configure une règle dans la table NAT pour rediriger les paquets TCP entrants sur le port 80 vers l’adresse IP 10.0.0.101 avant qu’ils ne soient routés.

Voici la connexion du client vers le serveur en ayant les tables ARP corrompues, le pirate peut donc inspecter tout le trafic et assurer la communication de bout en bout grâce au routage :



Voici la capture Wireshark de la requête du client vers le serveur en témoigne les différentes adresses IP affichées à l’écran :



**Tâche 9 Contre-mesures contre des attaques MiM (6 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| WITTIG Antonin  VADAM Julien | 50%  50% |

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 |  |
| Mise en place des solutions de détection |  |
| Mise en place de la protection |  |

**Rapport**

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

1. **Contre Mesures généralistes :**

- Utilisations de protocoles sécurisés tels que https, ftps…

- Mise à jour des logicielles.

- Utilisation de VPN.

- Surveillance du réseau…

1. **Désactivation ARP gratuitous :**

Exécuter la commande suivante : «echo 1>  /*proc/*sys/net/ipv4/conf/all/drop\_gratuitous\_arp».

Celle-ci désactive l’arp gratuitous et minimise donc le risque d’une attaque MiM.

1. **Limitation des adresses MAC sur le commutateur :**

Pour éviter les attaques MiM et ainsi empêcher l’usurpation des adresses IP, nous allons activer le mode : « ***switchport port-security maximum adresses*** » sur les switchs cisco.

Ce mode permet de limiter le nombre de MAC qui peuvent se connecter au switch sur l’interface souhaitée.

Pour limiter l’accès au port, il a déjà fallu faire la commande « ***switchport mode access***» puis de faire cette liste de commandes :

| - ***switchport port-security***  - ***switchport port-security maximum 1***  - ***switchport port-security violation shutdown***  - ***switchport port-security mac-address <*mac*>*** |
| --- |

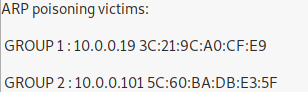


Nous pouvons voir que la configuration déployée sur Gi1/0/13 à bien fonctionné, les logs du bas de l’image indiquent bien que l’interface s’est fermée car nous n’avions pas l’adresse MAC acceptée par la configuration, il est donc impossible de se connecter au switch sans connaître les adresses MAC acceptées pour essayer d’écouter le réseau et de faire une attaque Man in the Middle.

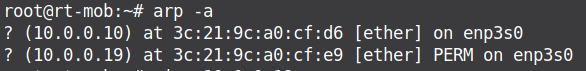
1. **Figer la correspondance entre Mac et IP (ip fixes) :**

Configurer les hôtes pour attribuer des adresses IP spécifiques à des adresses MAC uniques avec la commande « ***arp -s <@ip> <@MAC>*** » sur linux par exemple.

Après avoir figé l’adresse MAC et IP sur le client pour le serveur et sur le serveur pour le client, puis effectué l’ARP poisoning comme ci-dessous :



On voit que la table ARP n’a pas été modifié et donc que la contre mesure a fonctionné :



1. **Installer ARP Watch :**

Installez d’abord le paquet arpwatch, ensuite faites « ***systemctl enable arpwatch@enp3s0*** ».

Cette commande permet d’activer le service arpwatch pour une interface réseau spécifique en l’occurrence l’interface enp3s0, il ne reste plus qu’à démarrer ce service.

Arpwatch va créer une base de données adresses IP ↔ adresses MAC que l’on va pourvoir consulter dans /*var/lib/arpwatch/enp3s0* comme ci-dessous :



**Tâche 10 Supervision du réseau (3,75 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| WITTIG Antonin | 100% |

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

**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 piste 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 |  |
| Mise en place de la supervision |  |
| Génération de rapports |  |

**Rapport**

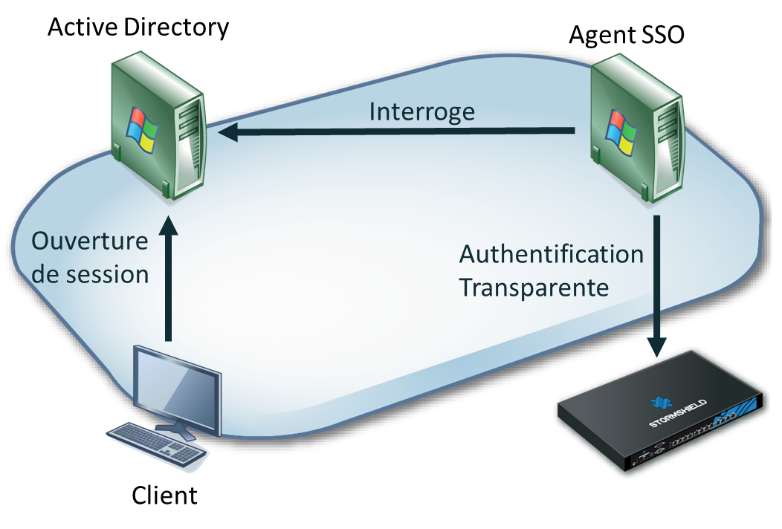
*(Expliquez votre méthode, captures d’écrans des tests, etc.)*

**Tâche 11 Mise en place d’une architecture Single Sign-On (9 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| NON FAITES |  |

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

**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 évènement, 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 |  |
| Installation d’un agent SSO sur une machine |  |
| Configuration de la machine de client |  |
| Changement de l’adresse IP de la machine |  |

**Tâche 12 Configuration d’un VPN SSL pour clients distants (6 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| GAMBIER Maxence  WITTIG Antonin | 50%  50% |

Estimation du temps passé sur cette tâche en heure-homme : 1H

**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 |  |
| Génération d’un certificat |  |
| Mettre en place les règles de filtrage et de NAT |  |
| Configuration du service VPN SSL sur le Stormshield |  |
| Installation et paramétrage des clients |  |
| Tests de connexion |  |

**Rapport**

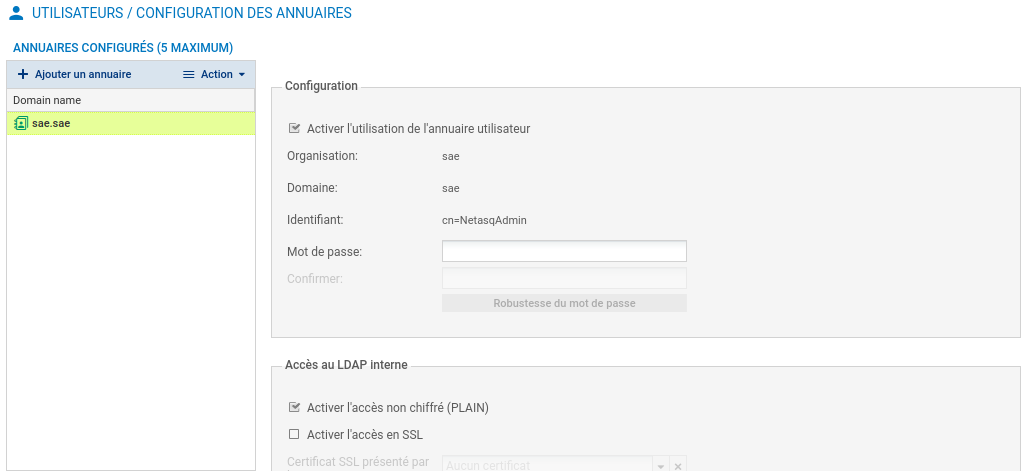
Le but de cette tâche était de mettre en place un VPN SSL sur le site A pour le client du site B.

Celui-ci sera mis en place sur le Stormshield A et nous y ferons différentes configurations pour le bon fonctionnement du VPN SSL :

* Configurer un annuaire
* Créer des utilisateurs sur l’annuaire
* Configurer le service VPN SSL sur le stormshield
* Mettre en place les règles de filtrage et de NAT
* Installer et paramétrer les clients

**Étape 1) Configuration d’un annuaire**

Pour commencer cette tâche, il a fallu mettre en place un annuaire sur le Stormshield. Celui-ci nous servira de répertoire pour les utilisateurs de la page d'authentification de celui-ci

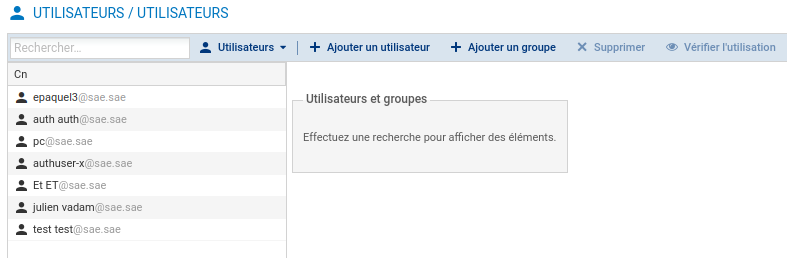


*Illustration 12.1 : Annuaire sur le Stormshield A*

Pour notre projet, nous l’avons nommé de manière simple “sae.sae” et nous avons laissé l’identifiant par défaut.

**Étape 2) Création d’utilisateurs sur l’annuaire**

Une fois cela fait, nous avons créé différents utilisateurs reliés à l’annuaire du portail captif et qui nous serviront également pour l’authentification au VPN SSL.

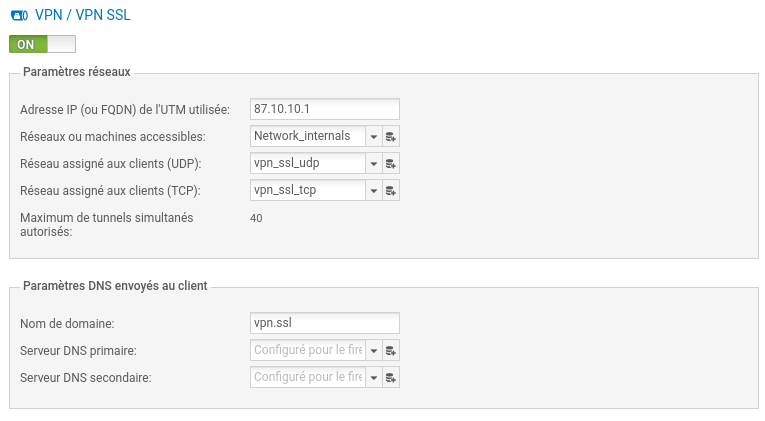


*Illustration 12.2 : Liste utilisateurs de l’annuaire sur le Stormshield A*

Nous avons donc un total de 7 utilisateurs et notamment un utilisateur “test” avec pour mot de passe “test1234” afin d’effectuer tous les tests de manière simple.

**Étape 3) Configurer le service VPN SSL sur le stormshield**

Une fois les étapes préliminaires faites, nous passons à la configuration du service sur le stormshield.

  
*Illustration 12.3 : Configuration du VPN SSL sur le Stormshield A*

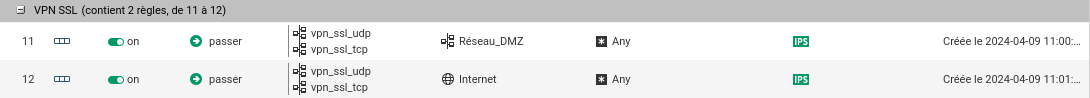
Pour cela, nous avons d’abord attribué l’adresse IP de l’UTM utilisé, c'est -à -dire l’adresse publique de notre routeur Stormshield (87.10.10.1 dans notre cas). Nous avons spécifié les réseaux qui seront accessibles lors d’une connexion au VPN SSL notamment les réseaux internes dans notre cas.

Nous devons ensuite spécifier les réseaux clients UDP et TCP. Pour cela, nous avons créer 2 objets “vpn\_ssl\_udp” et “vpn\_ssl\_tcp” qui sont donc 2 objets réseaux avec 2 adresses différentes et non trouvable sur le réseau pour éviter les conflits d’adresse IP (192.168.3.0/24 pour udp, 192.168.4.0/24 pour tcp)

Nous avons mis également un nom de domaine “vpn.ssl” et pour le reste nous avons laissé tout par défaut.

**Étape 4) Mettre en place les règles de filtrage et de NAT**

Comme nous utilisons des adresses différentes du réseau, nous avons mis en place des règles de filtrage et de NAT afin qu’une fois connecté au VPN, on ait accès aux services du réseau A comme si nous étions en interne.

**

*Illustration 12.4 : Règles de filtrage du VPN SSL*

Ces règles autorisent les réseaux “vpn\_ssl\_udp” et “vpn\_ssl\_tcp” à accéder au réseau DMZ et à Internet sur n’importe quel port comme les utilisateurs du réseau interne de base.

**

*Illustration 12.5 : Règle PAT pour le VPN SSL*

Pour la NAT, nous avons mis en place une règle PAT qui translate les paquets des réseaux vpn vers internet en des paquets qui viennent du Firewall\_OUT sur un port éphémère vers n’importe qui.

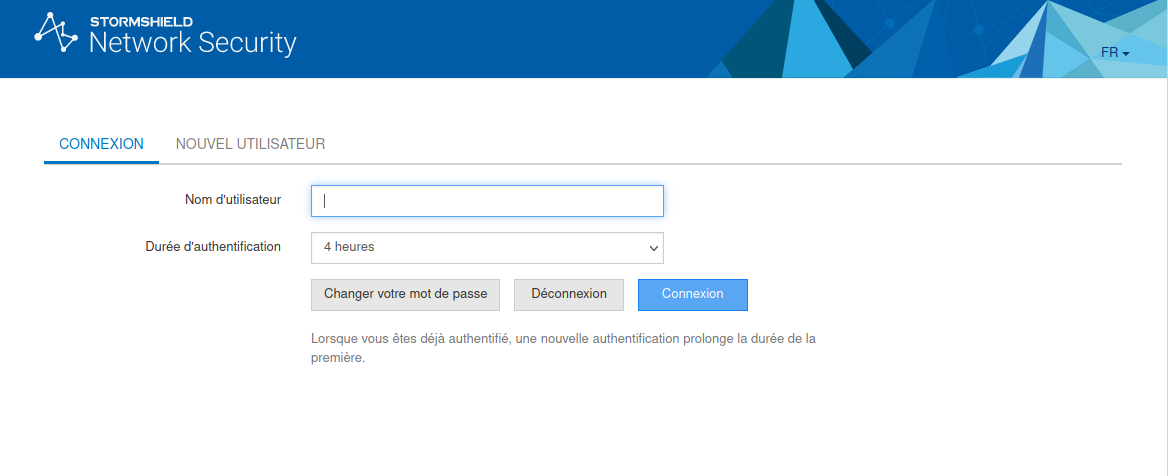
**Étape 5) Installer et paramètrer les clients**

Notre VPN SSL étant prêt et configuré, nous avons mis en place un client Linux avec OpenVPN pour effectuer les tests du VPN sur le client B.

Pour cela, nous avons installé OpenVPN sur le client B.

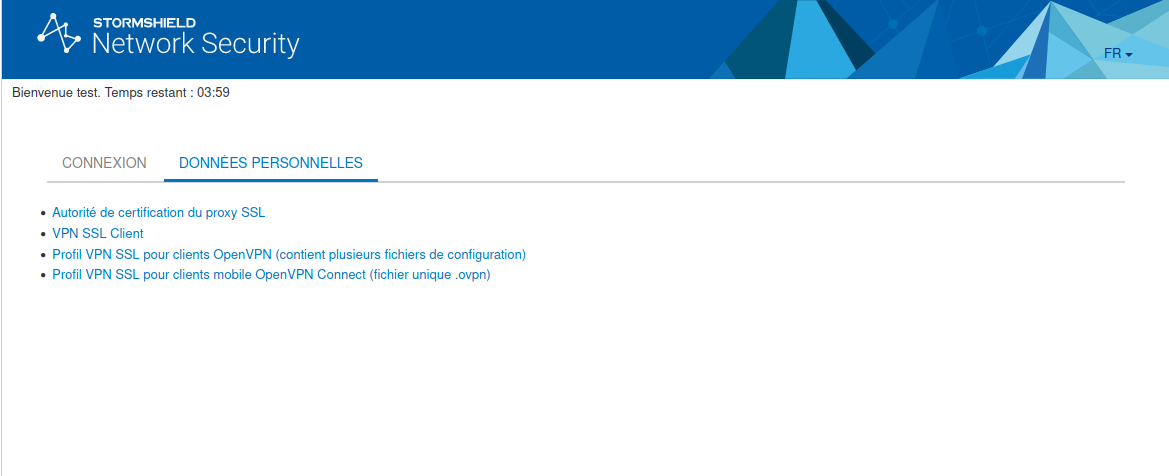
| root@clientB:~# apt install openvpn |
| --- |

Une fois installé, nous avons accèdé à la page du portail captif du stormshield A afin de récupérer notre profil VPN.



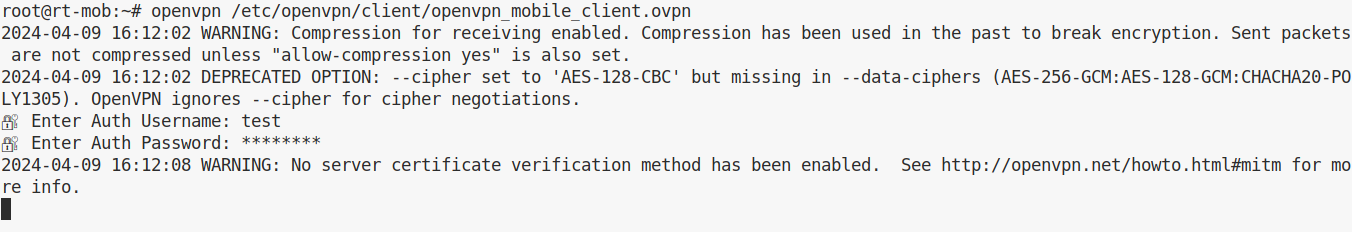
*Illustration 12.6 : Page d’authentification au portail captif*

Nous nous connectons à l'utilisateur “test” comme prévu.

**

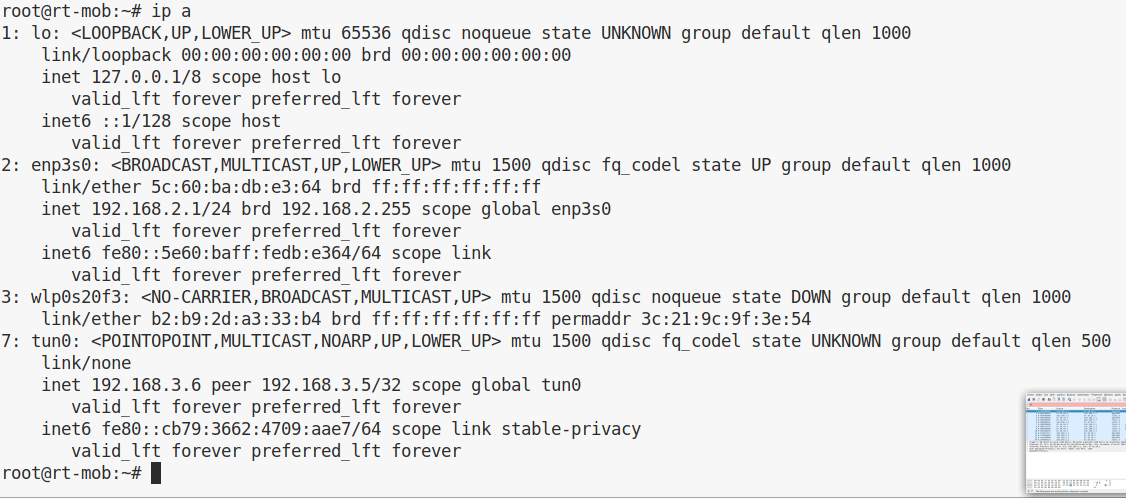
*Illustration 12.7 : Page portail captif une fois connecté*

On récupère donc le profil VPN SSL unique .ovpn et on l'inclut au dossier client de OpenVPN et on lance donc la connexion OpenVPN toujours avec l’utilisateur “test”.

**

*Illustration 12.8 : Connexion avec OpenVPN avec le profil VPN SSL unique récupéré*

Comme nous n’avons pas d’erreur, la connexion VPN est probablement effectué et pour vérifier cela nous avons regardé les IPs du client et nous avons vu un tunnel “tun0” qui correspond à un tunnel VPN SSL et au vue de l’IP nous en concluant que c’est une connexion UDP.

**

*Illustration 12.9 : Affichage IP avec le tunnel VPN SSL*

Pour terminer les tests, nous avons essayé d'accéder à notre page web du serveur Web A à l’aide de son adresse privée et on peut voir que ça fonctionne.

**

*Illustration 12.10 : Accès page web avec le VPN SSL depuis IP Privé du serveur web A*

**Tâche 13 Configuration d’un VPN IPSEC site à site (5,25 points)**

| Liste des personnes impliquées avec pourcentage de répartition |  |
| --- | --- |
| VADAM Julien  WITTIG Antonin | 50%  50% |

Estimation du temps passé sur cette tâche en heure-homme : 3H

**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 |  |
| Testez et faites valider |  |
| Mettez en place un tunnel entre tous vos réseaux en utilisant les VTI |  |
| Testez et faites valider |  |
| Utilisation des certificats pour l’authentification des SNSs |  |
| Testez et faites valider |  |

**Rapport**

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