

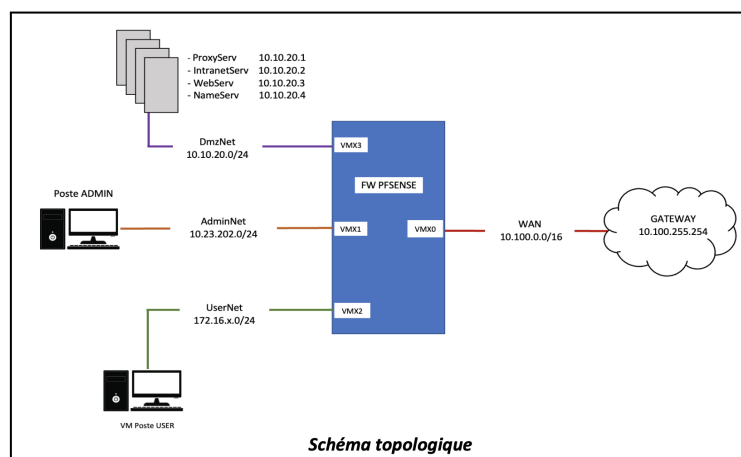
# IF03 – Compte Rendu TP

<b>1 - Découverte d'un Firewall IPSec</b>	<b>1</b>
1.1 - Objectifs du TP	1
1.2 - Mise en place prise en main du Firewall	2
1.3 - Translation d'adresses par une règle de type PAT	2
1.4 - Redirection de ports	3
1.5 - Activation des fonctions de sécurité & politique de filtrage	3
<b>2 - Initiation à l'analyse Forensique par un gendarme</b>	<b>4</b>
2.1 But du TP et outils à disposition	4
2.2 Nos apprentissages sur l'analyse Forensique	4
2.3 Découvertes notables sur le disque dur	5
<b>3 - Capture de drapeaux sur ROOT-ME</b>	<b>5</b>
3.1 Thème : Web serveur	5
3.2 Thème : Stéganographie	6
3.3 Thème : WEB-CLIENT	7
3.4 Thème : Réseau	8

## 1 – Découverte d'un Firewall IPSec

### 1.1 – Objectifs du TP

Dans le cadre de ce TP, il nous a été proposé de travailler sur l'administration d'un firewall virtuel Pfsense. Ce dernier permet de visualiser le fonctionnement des paquets qui transitent dans le réseau afin de procéder à la configuration du pare-feu, à l'établissement de règles de flux et de translation d'adresses. Afin de mieux comprendre son fonctionnement, nous disposons d'une architecture de réseau virtuelle schématisée ci-dessous.



Conformément à l'illustration notre architecture comprend trois réseaux internes connectés à Internet via le pare-feu FW Pfsense :

- **AdminNet (10.23.202.0)** : Réseau spécifique à l'administration du système d'information.
- **DmzNet (10.10.20.0)** : Réseau spécifique aux serveurs.
- **UserNet (172.16.x.0)** : Réseau spécifique aux utilisateurs.

## 1.2 – Mise en place prise en main du Firewall

Dans le cadre de cette configuration du pare-feu, nous suivons plusieurs étapes pour assurer son bon fonctionnement et sa connectivité avec notre réseau interne, voici celles que nous avons réalisées :

### Configuration du pare-feu :

- On lance et on effectue la configuration initiale du pare-feu.
- On lie les interfaces du pare-feu aux interfaces du réseau interne.
- On attribue une adresse IP au PC Admin pour accéder au panneau de configuration du pare-feu.

### Configuration des interfaces :

- On configure depuis le poste admin différentes interfaces (WAN, LAN, OPT1, OPT2).
- On ajoute les adresses IP correspondantes à chaque interface en fonction du sous-réseau auquel elle est destinée.

### Vérification de la configuration :

- On vérifie que la passerelle par défaut est le routeur du FAI.
- On utilise le menu "Diagnostic – Ping" pour effectuer des pings vers les différents réseaux connectés au pare-feu et vérifier leur connectivité.

## 1.3 – Translation d'adresses par une règle de type PAT

Dans le cadre de la configuration de la translation d'adresses (NAT – Network Address Translation), nous mettons en place des règles qui permettent d'associer les adresses IP publiques des paquets entrants aux adresses IP privées de notre architecture. Dans notre cas, nous disposons d'une seule adresse IP publique et nous utilisons une règle de type PAT (Port Address Translation) pour faire correspondre plusieurs adresses IP privées à cette unique adresse IP publique. Voici les différentes étapes de la configuration :

### Configuration de la translation d'adresses :

- On ajoute une nouvelle règle manuellement pour la translation d'adresses depuis le menu "Firewall – NAT – Outbound"
- On configure la règle de manière à remplacer l'adresse IP source des paquets émis par UserNet par l'adresse IP externe du pare-feu.
- Cette configuration permet au poste UserNet d'avoir accès à Internet grâce à l'adresse IP publique du pare-feu.

### Vérification de la configuration :

- On utilise la commande ping depuis le poste Admin et le poste User pour tester la connectivité vers l'adresse IP 8.8.8.8 : le poste Admin ne fonctionne pas contrairement au poste User.

## 1.4 – Redirection de ports

Actuellement, les adresses IP privées de notre architecture sont masquées à l'extérieur car notre architecture utilise une seule adresse IP visible depuis Internet. Pour permettre l'accès à un service

spécifique de notre réseau privé, tel que notre serveur web, depuis l'extérieur, nous allons créer une règle de translation d'adresses (NAT) qui redirigera le trafic entrant de l'extérieur vers la machine appropriée.

#### Configuration de la redirection de port :

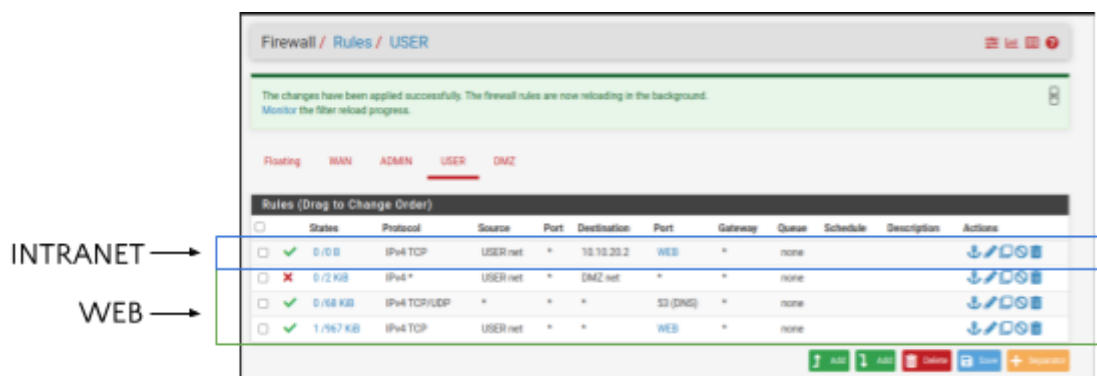
- On accède au menu "NAT – Port Forward" dans le panneau de configuration du pare-feu.
- On crée une nouvelle règle spécifiant que toutes les requêtes provenant de l'interface d'entrée WAN doivent être redirigées vers une seule machine utilisant le protocole TCP.
- On teste cette règle en utilisant notre poste User et en accédant à l'adresse de notre architecture. Nous constatons que nous sommes redirigés avec succès vers notre serveur web.

## 1.5 – Activation des fonctions de sécurité & politique de filtrage

Nous configurons les filtres et les règles nécessaires pour sécuriser notre architecture en bloquant le trafic non autorisé en suivant cet ordre :

- On ajoute des règles de filtrage dans le pare-feu en utilisant une table de règles et en rejetant par défaut le trafic qui ne correspond pas.
- On configure les règles spécifiques pour les postes du réseau "UserNet" dans le menu "Firewall – Rules – Usernet" (en autorisant les navigations web (http et https)).
- On ajoute une règle de redirection de port au-dessus de la règle de blocage pour le serveur DMZ.

Pour finir, **nous créons en autonomie des règles** autorisant uniquement "User" à naviguer sur le web et sur l'intranet (voir figure). Pour cela nous avons créé un alias WEB pour les ports 443 et 80, http et https. Une règle similaire ne permettra pas au poste admin de naviguer car les règles ne passent pas par le firewall directement mais par un routeur.



## 2 – Initiation à l'analyse Forensique par un gendarme

### 2.1 But du TP et outils à disposition

Le TP vise à simuler une expertise judiciaire concernant la création de faux diplômes universitaires. Le but est d'analyser une image .e01 de disque dur d'un ordinateur utilisé par la personne mise en examen. Nous devons déterminer s'il existe des traces ou des éléments démontrant ou niant l'implication du mis en examen.

#### Les outils que nous avons utilisés :

- Le logiciel **Autopsy** (Open-Source) nous permet d'extraire les preuves du disque dur
- Le site **SQLite Online Compiler** nous permet de visualiser les données des navigateurs
- Le logiciel **Volatility** nous permet d'analyser la RAM
- Différents convertisseurs (binaire, hexadécimal)

## 2.2 Nos apprentissages sur l'analyse Forensique

### Apprentissage général

- Pour **incriminer une personne** mise en examen il faut 3 éléments distincts :
  - ◆ Un élément légal (infraction prévue dans un texte)
  - ◆ Un élément matériel (le comportement a pour effet de produire le résultat visé par la loi)
  - ◆ Un élément moral (l'infraction doit être intentionnelle)
- La **base de registre** donne des informations sur les utilisateurs, elle est persistante, donne toutes les informations sur la vie d'une machine et nous indique où sont stockées les données.
- Les **partitions** sont des subdivisions logiques d'un disque dur permettant de stocker des données de manière optimisée. Dans notre cas il y en a 12 et nous avons utilisé les n°8 et n°12 car elles nous ont été indiquées par la base de registre.
- Les **bases de données de navigateurs** (SQLite) permettent de récupérer des informations comme l'historique, les téléchargements, les cookies. Elles retracent l'activité de l'utilisateur sur internet.
- La **RAM** d'un ordinateur permet d'obtenir des informations en temps réel, de récupérer des données effacées et de reconstituer l'activité récente d'un système. Elle ne doit pas être sous-estimée en analyse Forensique.
- Un **hash est une valeur numérique unique** générée à partir de d'un ensemble de données. Il permet de vérifier leur intégrité et leur authenticité.
- De manière générale, nous avons appris que **tout est toujours enregistré sur une machine** et qu'il est difficile d'y cacher un élément. La seule variable est le temps que l'on prend à découvrir un ce que l'on cherche.

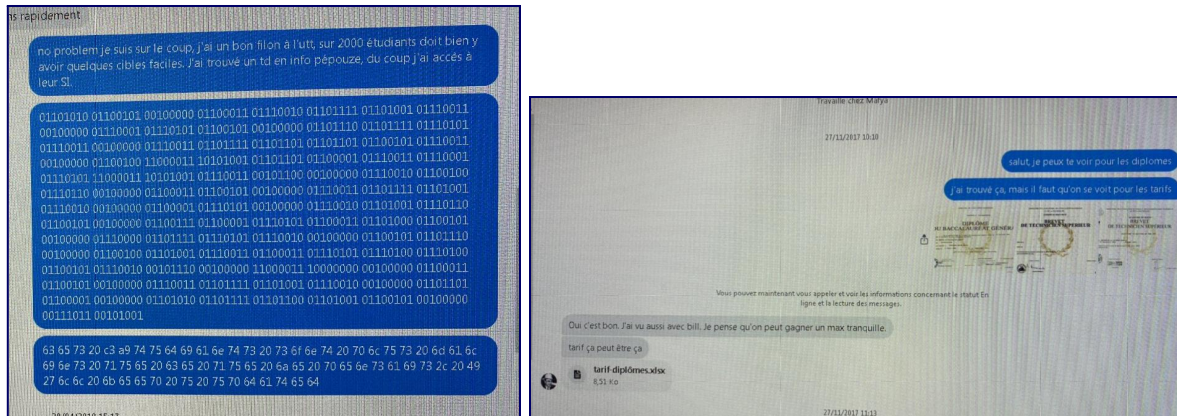
### Apprentissage technique

- Le **fichier msuser.dat** correspond à la base de données des utilisateurs
- Les partitions commençant par eb52 sont des **partitions windows**.
- Le **fichier places.sqlite** est une base de données de navigateur Firefox (resp. history.sqlite pour Google). Fichier accessible depuis :  
C:\Users\UTT3MSFOR\AppData\Local\Google\Chrome\Userdata\Default\History
- Un **fichier JPG** commence toujours par FFD8.
- Les **métadonnées d'une image** permettent notamment de voir si elles ont été modifiées par un logiciel.

## 2.3 Découvertes notables sur le disque dur

- D'après le fichier `User/UTT3MSFOR/APPDATA/ROAMING/MICROSOFT/WINDOWS/RECENT`, les derniers fichiers accédés portent le nom de **mdp.txt**, **diplome.eml**, **Faire un diplôme avec photoshop.mp4**.
- La base de registre nous indique qu'un fichier mdp.txt a été supprimé de la corbeille. Nous l'avons récupéré mais nous ne savons plus comment.
- Grâce au mot de passe, on **accède aux diplômes trafiqués** contenus dans un dossier compressé nommé image.7z.
- Les **images de diplôme** ont été scannées depuis une imprimante Canon MP550 Series. Il sera intéressant de vérifier lors d'une perquisition s'il s'agit bien de son imprimante personnelle.

- Les métadonnées nous indiquent que **les images de diplôme ont bien été modifiées** avec le logiciel Paint.
- Le fichier mdp.txt nous a permis d'**accéder aux réseaux sociaux** de l'individu. En réalité, il faut être en perquisition pour utiliser des identifiants en ligne. Nous y avons découvert des conversations avec des clients et des complices (voir photos ci-dessous).



La traduction du texte en binaire permet d'attester qu'il s'agit du complice de l'individu et le fichier tarif-diplomes.xlsx nous renseigne sur les prix proposés à des clients.

## Conclusion

Nous retrouvons bien à travers cette analyse les éléments permettant de dire qu'il s'agit d'une infraction :

- **Légal** : l'article 441-1 du code pénal réprime la création de faux documents.
- **Matériel** : nous avons trouvé les faux diplômes.
- **Moral** : les recherches de l'individu attestent de sa volonté de créer des faux documents.

## 3 - Capture de drapeaux sur ROOT-ME

### 3.1 Thème : Web serveur

**Challenge** : HTTP - Directory indexing

<https://www.root-me.org/fr/Challenges/Web-Serveur/HTTP-Directory-indexing>

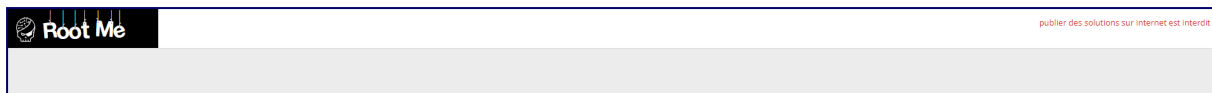
#### Description :

Le challenge "Directory Indexing" sur la plateforme Root-Me met en évidence une vulnérabilité courante dans la configuration des serveurs web, connue sous le nom d'**indexation des répertoires**.

L'indexation des répertoires est une fonctionnalité des serveurs web qui permet d'afficher le contenu des répertoires qui ne contiennent pas de fichier d'index, tels que index.html ou index.php. Lorsque cette fonctionnalité est activée, les utilisateurs **peuvent accéder à une liste des fichiers et des dossiers** présents dans le répertoire cible, ce qui peut potentiellement exposer des informations sensibles ou fournir des indices pour progresser dans le challenge.

Le but de ce challenge est de trouver un répertoire sur le serveur web cible où la fonctionnalité d'indexation des répertoires est activée. L'url initial du challenge est la suivante:

- <https://challenge01.root-me.org/web-serveur/ch4/>



Cet Url affiche une page vide qui contient que le header.

On a commencé par testé le dossier root mais sans résultat. On a ensuite essayé avec le dossier admin et voici ce qu'on obtient:

On accède au dossier car l'indexation du répertoire est activée. On clique sur pass.html mais il n'y a rien d'intéressant. On suit ensuite le dossier backup puis admin.txt et on y trouve le mot de passe: LINUX

/web-serveur/ch4/admin/		
File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
backup/	-	2021-Dec-10 22:35
pass.html	346 B	2021-Dec-10 22:35

**Niveau :** 2

**Problèmes rencontrés :** Trouver le dossier accessible, Comprendre le principe de l'activation d'un répertoire.

**Outils utilisés :** Google

**Solution :** LINUX

## 3.2 Thème : Stéganographie

**Challenge :** TXT – George et Alfred

<https://www.root-me.org/fr/Challenges/Steganographie/TXT-George-et-Alfred>

**Description :**

Le challenge "TXT – George et Alfred" sur la plateforme RootMe est un défi axé sur l'analyse de fichiers texte pour extraire des informations cachées.

Le challenge présente un fichier texte nommé "George\_et\_Alfred.txt" contenant en apparence un simple message. Cependant, ce message peut dissimuler des indices, des codes ou des informations essentielles pour résoudre le challenge.

L'objectif du challenge "TXT – George et Alfred" est de parcourir attentivement le contenu du fichier texte pour trouver des éléments cachés. Cela peut nécessiter l'application de techniques d'analyse des fichiers texte, telles que :

- **Analyse des caractères spéciaux :** Certains caractères spéciaux peuvent être utilisés pour cacher des informations. L'examen minutieux de ces caractères peut révéler des indices importants.
- **Analyse du formatage :** Le formatage du texte, tel que l'utilisation de sauts de ligne, d'espaces ou de tabulations, peut être significatif. Il est important de prendre en compte ces éléments lors de l'analyse du fichier.
- **Analyse des mots et des phrases :** Certains mots ou phrases peuvent sembler normaux à première vue, mais ils peuvent contenir des indices cachés lorsqu'ils sont examinés attentivement.
- **Analyse des codes ou des chiffres :** Le fichier texte peut contenir des codes ou des chiffres qui doivent être décodés ou interprétés d'une manière spécifique pour obtenir les informations nécessaires.

Dans notre cas, c'est l'analyse des mots et des phrases qui nous a permis d'obtenir un résultat. En effet, on remarque que dans la réponse d'Alfred de Musset, si on prend à chaque fois le premier mot de chaque ligne, on obtient la phrase suivante:

→ **Quand Voulez-vous Que Je Couche Avec Vous**

Si on réitère la même opération sur la réponse de George Sand, on obtient le flag pour le challenge qui est:

→ **Cette Nuit**

**Niveau : 2**

**Problèmes rencontrés :** Trouver quelle analyse du texte était la plus pertinente

**Outils utilisés :** Google et notre tête

**Solution :** Cette Nuit

### 3.3 Thème : WEB-CLIENT

**Challenge :** Javascript – Obfuscation 3

<https://www.root-me.org/fr/Challenges/Web-Client/Javascript-Obfuscation-3>

**Description :**

Dans le challenge "Javascript – Obfuscation 3", notre objectif est de désobfusquer le code. Comme son nom l'indique, cette tâche implique la révélation du code caché.

Nous commençons par étudier la fonction déchiffre. Nous remarquons alors que le résultat de cette fonction ne dépend pas de la valeur de son paramètre. En réalité, elle ne sert qu'à générer le message "FAUX PASSWORD HAHA".

Peu importe le mot de passe saisi, les deux dernières lignes du script donneront toujours le même résultat. Par conséquent, nous portons notre attention sur l'autre partie du code. On remarque rapidement que la clé réside dans la chaîne de caractères passée en paramètre, qui doit être décodée en deux étapes:

→ **Première étape:** Convertir les codes hexadécimaux en caractères en utilisant le code ASCII et un convertisseur automatique. Par exemple, "\x35" correspond à 5 en décimal. Ainsi, "\x35\x35\x2c" devient "55.". En appliquant le même principe au paramètre d'entrée, la chaîne de caractères obtenue est donc la suivante : 55,56,54,79,115,69,114,116,107,49,50.

→ **Deuxième étape :** Utiliser la version désobfusquée de la fonction "dechiffre" en remplaçant la valeur de la chaîne passée par celle que nous avons découverte. En effectuant le calcul suivant : `String.fromCharCode(55,56,54,79,115,69,114,116,107,49,50)`, nous obtenons le mot de passe "786OsErk12", qui est la solution du challenge.

**Niveau : 3**

**Problèmes rencontrés :** Difficulté de compréhension du code, Comprendre l'importance du paramètre d'entrée

**Outils utilisés :** Google

**Solution :** 786OsErk12

### 3.4 Thème : Réseau

**Challenge :** Authentification twitter

<https://www.root-me.org/fr/Challenges/Reseau/Authentification-twitter>



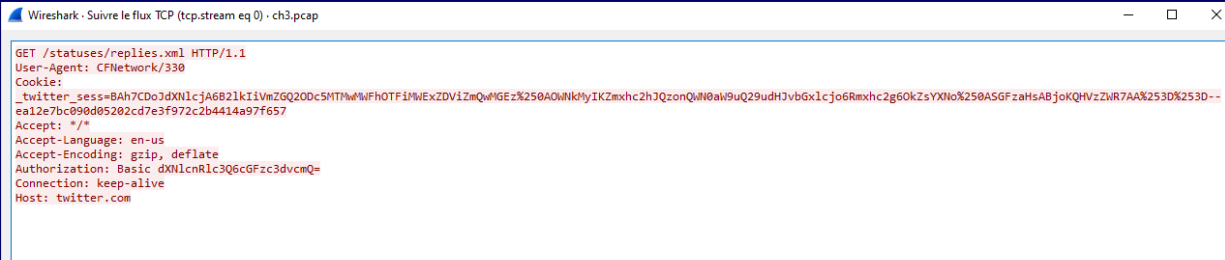
**Description :**

Le challenge "Authentification Twitter" sur Root Me est une épreuve qui met à l'épreuve nos compétences en matière de déchiffrement et de compréhension des protocoles réseau. Pour résoudre ce défi, nous avons utilisé l'outil Wireshark, un analyseur de protocole réseau. En **capturant le flux TCP entre notre machine et le serveur Twitter**, nous avons pu examiner les paquets échangés et extraire des informations pertinentes.

En analysant attentivement le flux TCP, nous avons identifié les paquets contenant les échanges d'informations d'authentification entre notre machine et le serveur Twitter. Cela peut inclure des paquets contenant des requêtes HTTP, des en-têtes de demande ou de réponse, ainsi que **des données potentiellement chiffrées**.

Voici le fichier qu'on obtient:

On remarque que dans le champ Authorization, on a le code suivant:



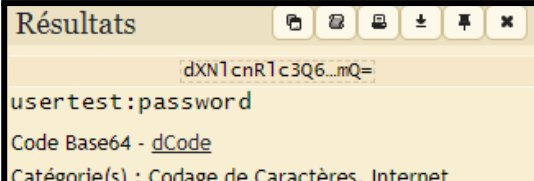
```

GET /statuses/replies.xml HTTP/1.1
User-Agent: CFNetwork/330
Cookie:
  _twitter_sess=BAh7CDoJdXNlcnRlc3Q6cGFzc3dvcmQ6MTpWfH0TF1hWExZDV1ZmQwMGEz%250AOWNkYlYkZmxhc2h3QzonQmN0aW9uQ29udHJvbGx1cjo6Rmxhc2g6OkZsYXNo%250ASGFzaHsABj0KQHVzZWR7AA%253D%253D--
  Es12e7bc890d85202cd7e3f972c2b4414a97f657
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Authorization: Basic dXNlcnRlc3Q6cGFzc3dvcmQ=
Connection: keep-alive
Host: twitter.com
  
```

→ **dXNlcnRlc3Q6cGFzc3dvcmQ=**

On a donc cherché à trouver un moyen pour décoder ce code et on a essayé un décodeur de base 64.

On obtient le résultat suivant :



```

Résultats
  dXNlcnRlc3Q6cGFzc3dvcmQ=
usertest:password
Code Base64 - dCode
Catégorie(s) : Codage de Caractères, Internet
  
```

**Niveau : 2**

**Problèmes rencontrés :** Prise en main de wireshark, compréhension du problème, analyse des flux TCP

**Outils utilisés :** Google, Wireshark

**Solution :** password