Liste des Sujets à exposer

- **Sujet 1**: Déploiement d'un certificat sur un serveur web
- **Sujet 2**: Crackage de mot de passe d'une machine Windows 7, 8, 10,XP et Linux
- **Sujet 3 :** Mise en place d'un tunnel sécurisé (VPN) avec IPsec
- **Sujet 4 :** Les Injections SQL
- **Sujet 5:** Sécurité de TELNET, SSH, Points d'accès Wifi etc..
- **Sujet 6 :** Supervision de réseaux avec Nagios
- **Sujet 7 :** Sécurité d'une architecture d'un réseau d'entreprise
- **Sujet 8 :** Mise en place d'un portail captif d'un LAN avec PfSense

Exposé 1

Sécurité des réseaux

Sujet : Déploiement d'un certificat sur un serveur web

L'objectif de cet exposé est de configurer ou déployer un certificat SSL avec des paramètres de sécurité (clés existantes) ou avec des paramètres à générer par vous-mêmes.

L'exposé sera guidé par les directives suivantes à suivre pour bien mener les tâches demandées ; mais cela n'empêche pas d'utiliser une méthode, une technique ou une procédure autre que celle recommandée ici.

La procédure décrite ici ne fait qu'office de guide, et il vous reviendra au cas où elle ne marche pas, de chercher une solution et de décrire étape par étape une version finale dans un rapport à rendre avant la date indiquée.

L'exposé sera composé de deux parties :

1ème partie : vous devez faire les configurations nécessaires en vue d'activer le certificat SSL d'un site afin de le faire fonctionner sous https.
2ème partie : vous devez créer des certificats à partir de clés que vous générez vous-mêmes, puis faire les configurations nécessaires pour faire fonctionner le site en mode https.

1ème partie : Configuration d'un site de http à https

Étape 1 : Installation un serveur web Ubuntu

Étape 2 : Sur Ubuntu serveur accessible directement ou via SSH, installez le serveur web Linux Apache, MySql, Php (LAMP) par la commande :
`Apt-get install apache2`
Allez dans le répertoire `/var/www/html` sur lequel pointent deux sites (`default` et `default-ssl`) dont seul `default` est actif et remarquez la présence d'un fichier `index.html`, puis éditer le :

- `vim /var/www/html/index.html`

Vous pourrez par la suite vérifier si le site est accessible en ouvrant le navigateur et en mettant sur la barre d'adresse l'adresse IP du site (serveur Ubuntu). Le site doit normalement s'afficher sur la page d'accueil.

Étape 3 : Clonez ensuite un site (exemple celui de l'ugb.sn) en allant sur https://www.ugb.sn. puis faites `clique droit`, ensuite `afficher le code sources de la page` à `copier` et à `coller` pour remplacer le contenu du fichier `index.html`.

- `sudo vi /var/www/html/index.html`
- `Taper v pour selectionner le texte à remplacer`

Ensuite utilisez successivement la commande `service apache2 restart` pour réactualiser le service ; puis accédez à nouveau sur le site via l'adresse IP.

Étape 4: Usage de Wireshark pour capturer des donner d'authentification d'un site fonctionnant avec http. Il faudra alors cloner ou utiliser un site nécessitant l'envoie d'information sur le serveur.
Installer wireshark sur votre ordinateur, et via l'interface connecté à internet, capturer le trafic

Étape 5: Migration vers https et vérification avec Wireshark du chiffrement des données.
Utiliser les commandes suivantes pour passer de http à https en utilisant un certificat auto-signé.

- `a2enmode ssl` //active le module de sécurité
- `a2ensite default-ssl` //activer le site
- `/etc/init.d/apache2 restart` //réactualiser le service

Tester en accédant au site via https
On peut faire la redirection de `http` vers `https`, i.e rediriger ceux qui tentent d'accéder par `http` vers `https`. Il faudra alors éditer le fichier `/etc/apache2/sites-available/000-default.conf`,
Dans le `virtualhost`, ajoutez la ligne suivante :
`Redirect permanent / https://X.X.X.X` ou X.X.X.X est le nom de domaine ou l'adresse IP
`service apache2 restart`
Réactualiser, puis utiliser Wireshark pour vérifier la confidentialité des données transmises vers le serveur

2ème partie : Création et utilisation d'un certificat auto-signé

Étape 1 : Génération de la paire de clefs RSA

- Générez votre paire de clefs RSA d'une taille de 2048 bits. On suppose nommé `maCle.pem le` fichier contenant votre paire de clefs RSA.

```
$ openssl genrsa -out maCle.pem 2048
```

- Extraire la partie publique de votre clef RSA, on suppose que cette partie est stockée dans un fichier nommé `maClePublique.pem.`

```
$ openssl rsa -in maCle.pem -pubout -out maClePublique.pem
```

Étape 2 : Génération d'un certificat auto-signé
Pour créer un certificat, vous devez d'abord créer une demande de signature de certificat. Vous pouvez envoyer votre demande à une autorité de certification, ou l'utiliser pour créer un certificat auto-signé. Pour faire la création de la requête vous pouvez la faire à partir d'une clé inexistante à priori ou créer d'abord la clé comme `maCle.pem`) .

- **Générer une requête à partir d'une clé existante :**

```
openssl req -new -key maCle.pem -out request.csr
```

Il faudra alors fournir une liste d'informations (dont le FQDN pour lequel le certificat est valide) relative à la création du certificat. Un certificat n'est valide que pour un **FQDN** (**Fully Qualified domain name**) par exemple www.ugb.sn

- **Création du certificat `sscert.cert` à partir de la requête `request.csr`**

```
openssl x509 -req -days 30 -in request.csr -signkey maCle.pem -out
sscert.cert
```

- **Créer un certificat auto-signé de clé privée (inexistante à priori)** de type **rsa** d'une taille de **2048 bits** le fichier de la clé privée est stocké dans le fichier **server.key** alors que le certificat sera contenu dans le fichier **server.crt**. avec une validité de 365 jours à partir de la date du jour.

```
$openssl req -x509 -newkey rsa:2048 -nodes -keyout server.key -out
server.crt -days 30
```

Cela peut se faire aussi de la façon suivante :

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -
out certificate.pem
```

Si ce certificat est déployé sur le navigateur, une notification du genre « site web non sécurisé » est signalée lors de l'accès au site web mentionnant que ce dernier est auto signé.

Étape 3 : Déploiement du certificat

Les communications en **http** sont réalisées sur le port 80 alors que les communications chiffrées sont réalisées sur le port 443. Si vous regardez le fichier **ports.conf** on peut voir que le port 443 sera ouvert si le module **ssl_module** est chargé sinon il n'ouvre pas le port .

```
$ cat /etc/apache2/ports.conf
```

Editer le contenu du dossier `/etc/apache2`

```
$ls /etc/apache2
```

Le fichier **ssl.conf** dans `/etc/apache2/mods-available` contient des configurations par défaut
Dans la suite, vous allez configurer SSL pour le site créé par défaut avec Apach2.

Nous pouvons ainsi éditer le fichier /etc/apache2/sites-available/default-ssl.conf
```
$ sudo vim /etc/apache2/sites-available/default-ssl.conf
```
Dans /etc/apache2/sites-available, créez le fichier /etc/apache2/sites-available/ugb-ssl.conf pour le site www.ugb.sn et y copier le contenu de default-ssl.conf.
Dans le contenu de default-ssl.conf au niveau de la section SSL, on indique respectivement sur les lignes SSLCertificateFile et SSLCertificateKeyFile le chemin des fichiers contenant le certificat et la clé publique.
Pour le site par défaut, créez juste le certificat et écrasez celui existant avec la clé publique associée par simple copie des fichiers créés.
```
<VirtualHost 172.17.0.1:443>
            ------
            ------
           ## SSL section
           SSLEngine on
           SSLCertificateFile "/etc/apache2/ssl/certs/server.crt"
           SSLCertificateKeyFile "/etc/apache2/ssl/private/server.key"
       </VirtualHost>
```
La sécurité du dispositif repose sur clé privée de chiffrement server.key, il est important de la protéger.
```
$  ls -l /etc/apache2/ssl/certs
-rw-r--r-- 1 root root 1704 Apr 21 08:36 server.key
$ sudo chmod o-r /etc/apache2/ssl/certs/server.key
```
Pour un site www.ugb.sn par exemple, on peut créer le certificat avec :
```
$ cd ~
$ mkdir ssl && cd ssl
$ openssl req -x509 -newkey rsa:2048 -nodes -keyout ugb.key -out ugb.crt
$ sudo mkdir /etc/apache2/ssl
$ sudo cp * /etc/apache2/ssl
Dans le fichier /etc/apache2/sites-available/ugb-ssl.conf
```
Vous aurez au niveau de la section SSL :
```
…….
……..
#SSL section
   SSLEngine on
 SSLCertificateFile "/etc/apache2/ssl/ugb.crt"
 SSLCertificateKeyFile "/etc/apache2/ssl/ugb.key"
 </VirtualHost>
```
Validation de la configuration et redémarrage du service.
```
$ cd /etc/apache/sites-enabled
$ sudo ln -s /etc/apache2/sites-available/default-ssl.conf
$ sudo apachectl configtest && sudo /etc/init.d/apache2 restart
        Syntax OK
         * Restarting web server apache2
         ...done.
```
Vous devez pouvoir à présent accéder au site web SSL avec votre navigateur :https://IP-duServeur

Exposé 2

Sécurité des Réseaux

Sujet : Crackage de mot de passe d'une machine Windows 7, 8, 10,XP et Linux

L'objectif de cet exposé est de pouvoir récupérer un mots de passe inconnu sur une machine dont on a accès.

L'exposé sera guidé par les directives suivantes à suivre pour bien mener les tâches demandées ; mais cela n'empêche pas d'utiliser une méthode, une technique ou une procédure autre que celle recommandée ici.

La procédure décrite ici ne fait qu'office de guide, et il vous reviendra au cas où elle ne marche pas, de chercher une solution et de décrire étape par étape la version finale dans un rapport à rendre avant la date indiquée

L'exposé sera composé de deux parties :
1ème partie : Crackage de mot de passe d'une machine Windows 7, 8, 10,XP.
2ème partie : Crackage de mot de passe d'une machine GNU/Linux

Partie 1 : Crackage de mot de passe d'une machine Windows 7, 8, 10,XP.
Avant de pouvoir craquer un mot de passe, on suppose qu'on a accès d'abord à l'ordinateur physique et que l'on a un exploit découvert.
L'oubli de mot de passe peut être une des raisons pour lesquelles on doit se retrouver dans cette situation, mais il y en a encore tant d'autres : vol de mot de passe etc.
Sur le système Windows XP, 7, 8, 10, les mots de passe des utilisateurs sont stockés dans un fichier nommé SAM (Security Account Manager) et se trouvant dans le dossier «C:\Windows\system32\config» et où sont stockés les cryptés des login et mots de passe des utilisateurs. Ces mots de passe ont d'abord été hachés avant d'être chiffrés par une clé qui se trouve dans le fichier SYSTEM se trouvant au même endroit que SAM.
L'objectif de ce TP est :

- d'utiliser d'abord kali Linux pour avoir accès à la base de données SAM.
- d'utiliser samdump2 pour extraire le l'empreinte à partir d'un déchiffrement effectué par le biais de la clé du fichier SYSTEM.
- D'utiliser john the Ripper, pour trouver le mot de passe à partir de l'empreinte.

Avec Kali Linux : Récupération de la base de données SAM

Des outils comme Trinity permettent d'effacer le mot de passe de l'administrateur pour se connecter à sa place, mais ce n'est pas une méthode discrète et réutilisable donc il serait plus intéressant de récupérer le mot de passe administrateur ou d'un utilisateur pour pouvoir se connecter plusieurs fois et paramétrer la machine pour un accès distant plus discret (Backdoor).
Avec Kali Linux en mode déconnecté grâce à une clé USB ou en mode Live. En effet un liveCD utilise un ram disk qui est en fait une partie de la mémoire vive qui est utiliser comme mémoire de masse. Pour avoir accès aux données figurant sur le disque de la machine cible et en particulier à la base SAM, il faut accrocher la partition contenant celle-ci au système des

fichiers de notre LiveCD, on l'appelle cela, « monter une partition » On peut alors vérifier le disque physique sur lequel on souhaite valider notre montage.

Création d'un live CD Kali Linux via Linux

1- télécharger l'image ISO Kali Linux el le mettre dans un dossier VotreNom
2- Lancer le terminal linux et lister les disques de la machine afin de repérer votre clé USB déjà branchée
$ lsblk -S
3- Si votre clé USB est /sd?, et que l'image ISO de Kali Linux se trouve /home/VotreNom/Kali.iso, taper en mode administrateur la commande suivante (? est la lettre assoscée à la clé USB) :
#sudo dd if=/home/VotreNom/Kali.iso of=/dev/sd? bs=512k
4- Dans le BIOS ou l'UEFI, configurer votre ordinateur en mode démarrage avec clé USB puis redémarrer
5- Si le live CD ne se lance pas alors que vous utiliser l'UEFI, entrer à nouveau dans le BIOS (UEFI) et désactiver le « Secure Boot »

Récupération de la base de données SAM

*# sudo su /*pour se connecter en tant que Root dans Kali*/*

 # fdisk -l ou sudo fdisk -l

Device.	*Boot*	*Start*	*End*	*block*	*Id*	*System*
/dev/sda1	*	*2048*	*20687.*	*102400*	*7*	*HPFS/NTFS/extFAT*
/dev/sda2		*20687*	*61435903*	*30614528*	*7*	*HPFS/NTFS/extFAT*

- Montons la partition /dev/sda2 sur le repertoire /mnt/ de Kali
 # mount -t ntfs-3g /dev/sda2. /mnt/
 # mount -t vfat /dev/sda2 /mnt/
- Un ls sur /mnt/ montre bien que nous somme sur le disque monté
- Déplaçons nous dans le repertoire contenant SAM et SYSTEM
 # cd /mnt/Windows/system32/config
- A ce stade, on peut utiliser le programme samdump2, permettant d'extraire les empreintes et les envoyer même dans un fichier (ici hash.txt)
 # samdump2 SYSTEM SAM > /tmp/hash.txt
- A ce stade, on peut utiliser john (s'il est disponible) pour récupérer le mot de passe en clair.
- A ce point on a les hash selon un format donnée on peut même éditer le fichier hash.txt avec
- $ cat hash.txt, puis identifier la ligne de concernant l'utilisateur et mettre le hash dans un autre fichier.
- On peut alors utiliser John ou un autre utilitaire qui permet de trouver à partir de l'empreinte la valeur hachée.
 # john /tmp/hash.txt
ou
 # john –format-NT /tmp/hash.txt
Ou……
John –format=NT /tmp/hash.txt

Partie 2 : Crackage de mot de passe d'une machine GNU/Linux

Avec samdump2

GNU/Linux utilise le salage et les mot de passe sont stockés dans le fichier /etc/shadow et le fichier "/etc/passwd" contient les noms des comptes utilisateurs (logins). Il est nécessaire d'essayer toutes les possibilités jusqu'à trouver le bon mot de passe, technique connue sous le terme d'une attaque par la force brute ou en force brute.
On peut utiliser John the Ripper avec le compte administrateur (root) puisque théoriquement seul ce dernier peut accéder aux fichiers des mots de passe pour des raisons évidentes de sécurité.

- On suppose que John est déjà installé :
#apt-get update
#apt-get install john
- Une attaque par force brute pure s'effectue par la commande :
 # john /etc/shadow

On peut utiliser différentes techniques pour réduire le temps de recherche, par exemple en spécifiant juste un utilisateur et dans ce cas, on va devoir fusionner les 2 fichiers en un seul avec la commande "unshadow".

- Fusionner les deux fichiers

#unshadow /etc/passwd /etc/shadow > hash.txt
#john hash.txt //est identique à john /etc/shadow

- il est possible de spécifier le nom d'un compte à cracker ou bien avec le numéro UID. Exemple pour l'utilisateur faye:
  #john --users=faye hash.txt
  #john --users=1055 hash.txt
  #john –users=faye hash.txt
- Une attaque hybride entre (combinaison de dictionnaire et de force brute) peux réduire significativement le temps de réponse.
  # cat /usr/share/dict/french // pour voir un dictionnaire français qu'on peut utiliser.
  #john –wordlist=/usr/share/dict/french /etc/shadow

Année 2023 Licence 3 en Ingénierie Informatique Cours de Sécurité des Réseaux

Exposé 3

Sécurité des Réseaux

Sujet : Mise en place d'un tunnel sécurisé (VPN) avec IPsec

L'objectif de cet exposé est de mettre en place un tunnel IPsec entre LANs d'une même organisation séparés par Internet.

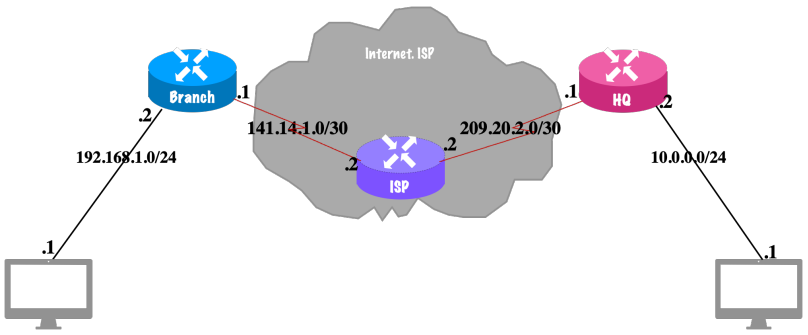
L'exposé sera guidé par les directives suivantes à suivre pour bien mener les tâches demandées ; mais cela n'empêche pas d'utiliser une méthode, une technique ou une procédure autre que celle recommandée ici.

La procédure décrite ici ne fait qu'office de guide, et il vous reviendra au cas où elle ne marche pas, de chercher une solution et de décrire étape par étape votre version finale dans un rapport à rendre avant la date indiquée

L'exposé sera composé de 6 étapes et doit être réalisé dans un environnement réel (3 routeurs cisco au moins) ou sur Packet Tracer:

1. Définir le protocole pour l'échange des clés partagées (**ISAKMP**) : IKE phase 1
2. Définir les paramètres IPSec (Ah et/ou ESP) via **ipsec transform-set :** IKE phase 2
3. Configurer l'ACL pour le trafic du VPN
4. Configurer le **crypto map** qui combine ISAKMP, ipsec transform-set, l'adresse du peer et l'ACL
5. Appliquer le **crypto map** à l'**interface** qui doit créer le VPN.
6. Vérifier la sécurité des paquets entre les routeurs Branch-ISP-HQ

Vous pouvez considérer le schéma suivant:



Configuration du routeur Branch

1. Définir le protocole ISAKMP sur les routeurs Branch et HQ

```
Branch(config)#crypto isakmp policy 1
Branch(config-isakmp)#encryption ae
Branchconfig-isakmp)#encryption aes
Branch(config-isakmp)#authentication pre-share
Branch(config-isakmp)#group 2
Branch(config)#crypto isakmp key gelt2 address 209.20.2.1
Branch(config)#
```

2. Définir les paramètres IPSec (Ah et/ou ESP)

```
Branch(config)#crypto ipsec transform-set vpn-gelt esp-aes esp-sha-hmac
```

3.Configurer l'ACL pour le trafic du VPN

```
Branch(config)#access-list 100 permit ip 192.168.1.0 0.0.0.255 10.0.0.0
0.0.0.255
```

4.Configurer le **crypto map** qui combine ISAKMP, ipsec transform-set, l'adresse du peer et l'ACL

```
Branch(config)#crypto map gelt-map 10 ipsec-isakmp
Branch(config-crypto-map)#set transform-set vpn-gelt
Branch(config-crypto-map)#set peer 209.20.2.1
Branch(config-crypto-map)#match address 100
```

5.Appliquer le **crypto map** à l'**interface** qui doit créer le VPN.

```
Branch(config-if)#crypto map gelt-map
```

Configuration du routeur HQ

1.Définir le protocole ISAKMP sur les routeurs Branch et

```
HQ(config)#crypto isakmp policy 1
HQ(config-isakmp)#encryption ae
HQ(config-isakmp)#encryption aes
HQ(config-isakmp)#authentication pre-share
HQ(config-isakmp)#group 2
HQ(config)#crypto isakmp key gelt2 address 141.14.1.1
HQ(config)#
```

2. Définir les paramètres IPSec (Ah et/ou ESP)

```
(config)#crypto ipsec transform-set vpn-gelt esp-aes esp-sha-hmac
```

3. Configurer l'ACL pour le trafic du VPN

```
HQ(config)#access-list 100 permit ip 10.0.0.0 0.0.0.255
192.168.1.0 0.0.0.255
```

4. Configurer le **crypto map** qui combine ISAKMP, ipsec transform-set, l'adresse du peer et l'ACL

```
HQ (config)#crypto map gelt-map 10 ipsec-isakmp
Branch(config-crypto-map)#set transform-set vpn-gelt
Branch(config-crypto-map)#set peer 141.14.1.1
Branch(config-crypto-map)#match address 100
```

5. Appliquer le **crypto map** à l'**interface** qui doit créer le VPN.

```
Branch(config-if)#crypto map gelt-map
```

6. Test

- Capturer un paquet et vérifier si l'encapsulation ESP est appliquée
- Mettre une autre machine dans le réseau 192.168.1.0/24 et vérifier que quand elle accède à un réseau autre que le 10.0.0.0/24 l'ESP n'est pas appliqué.
- Mettre d'autres réseaux non couverts par l'ACL pour montrer que même si le trafic passe par le tunnel il ne sera pas chiffré si le réseau sources et/ou de destination ne sont pas couverts par l'ACL.

Exposé 4

Sécurité des réseaux

Sujet : Les injections SQL

Les injections SQL (Structured Query Language) font parties des attaques les plus dangereuses et sont classées premières dans le document « The Ten Most Critical Web Application Security Risks» de l'OWASP (Th Open Web Application Security Project).

L'objectif de cet exposé est d'expérimenter les injections SQL. L'exposé sera guidé par les directives suivantes à suivre pour bien mener les tâches demandées ; mais cela n'empêche pas d'utiliser une méthode, une technique ou une procédure autre que celle recommandée ici.

La procédure décrite ici ne fait qu'office de guide, et il vous reviendra au cas où elle ne marche pas, de chercher une solution et de décrire étape par étape une version finale dans un rapport à rendre avant la date indiquée

L'exposé sera composé de 4 étapes :

- **Étape 1 :** Installation du SGBD MySQL
- **Étape 2 : Connection au serveur** Configurer l'ACL pour le trafic du VPN
- **Étape 3 :** Gestion des utilisateurs via phpMyAdmin
- **Étape 4 :** Les injections SQL

Étape 1 : Installation du SGBD MySQL
Installez le serveur `MySQL` et le client avec la commande

```
aptitude install mysl-sever
```

Donnez ensuite le mot de passe de l'administrateur principal de la Base de données. Comme un SGBD ne gère que des données, mais à la fois des droits d'accès à celles-ci. La création des utilisateurs et le réglage des droits sont assez fastidieux en ligne de commande, il convient alors d'installer `php-MyAdmin` ou un outils similaires

```
# aptitude install phpmyadmin
```

Vous pouvez utiliser `Apach2` comme serveur pour le fonctionnement de `phpMyAdmin.`

Étape 2 : Connection au serveur en tant qu'administrateur et création d'une BD et d'une table
MySQL est capable de gérer plusieurs bases de données en même temps. Par défaut il en existe 3: `Information_shema, performance_schema` et `mysql.` La base de données mysql est la plus importante ici dans la mesure où elle contient les droits de tous les utilisateurs.

- Connectez-vous au serveur avec la commande :

```
# mysql -u root -p
```

Mettez le mot de passe

```
mysql>show databases //pour visualiser les bases existantes
```

- Créez votre première base de données `ING2.`

```
mysql>create database ING2
```

- Créez la table `utilisateurs` qui contiendra les futurs utilisateurs de la BD `ING2`, pour cela il faut définir les champs de la table et leur type.

La définition de la façon la plus précise du type du champs est une première étape de sécurité qui empêchera au pirate d'enregistrer tout ce qu'il veut lorsque la table est piratée.

```
mysql>use ING2

mysql>create table utilisateurs (id INT NOT NULL, nom VARCHAR(50), prenom VARCHAR (50), login VARCHAR (20), password VARCHAR(30)
```

- Insérez 4 utilisateurs

```
mysql>insert into utilisateurs values(111,'Yous','FAYE','free', 'aaa') ;

mysql>insert into utilisateurs values(222,'Abdoul','MAJID','orange','bbb') ;

mysql>insert into utilisateurs values(333, 'Jean','MARI','express', 'ccc') ;

mysql>insert into utilisateurs values(444, 'Talla','MBOUP','promo', 'ddd') ;
```

Il faut noter que les mots de passe des utilisateurs sont enregistrés en claire, ce qui altère la sécurité. Les mots de passes peuvent être stockés chiffrés afin que le pirate qui a accès à la base ne puisse s'identifier. En plus, il est facile de chiffrer car MySQL fournit une fonction `password` pour cela. Voici un exemple :

```
mysql>insert into utilisateurs values(555, 'Marie','BA','ipsl',
password('eee')) ;

mysql>select * from utilisateurs ;
```

La fonction `password`, si elle est appliquée (peut ne pas exister sur certaine version de php), rendra illisible le mot de passe. Cependant, si le mot de passe est haché, son empreinte peut être plus long que le champs, dans ce cas on peut mettre à jour le champs en question.

```
mysql>alter table utilisateurs modify password VARCHAR (41) ;

mysql>update utilisateurs set password=password('eee') ;

mysql>update utilisateurs set password=password('eee') ;
```

Pour trouver un utilisateur dont le mot de passe est en claire.

```
mysql>select * from utilisateurs where id='orange' and password='bbb';
```

Pour trouver un utilisateur dont le mot de passe est en haché.

```
mysql>select * from utilisateurs where id='ipsl' and password='eee';
```

Toutes les manipulations effectuées jusque-là l'ont été avec l'utilisateur `root` qui dispose de tous les droits sur toutes les bases. Il serait plus pertinent de créer un utilisateur dont tous les droits seront limités aux actions que nous souhaitons qu'il puisse effectuer. Il peut être plus pratique d'utiliser dans ce cas phpmyadmin.

Etape 3 : Gestion des utilisateurs via phpmyadmin

phpMyAdmin est accessible via le navigateur à l'adresse : http://localhost/phpmyadmin. Il faut ensuite saisir le mot de passe indiquée lors de l'installation de MySQL.

La configuration par défaut (faite ci-dessus) lors de l'installation crée un alias qui donne accès à phpMyAdmin par un url du type : mon-site.fr/phpmyadmin et les robots des hackers le savent. Si votre phpMyAdmin est en ligne, il peut être judicieux de renommer cet alias. Pour le faire, il suffit : d'éditer avec les droits d'administration le fichier :
`/etc/phpmyadmin/apache.conf`
et de remplacer :

```
Alias /phpmyadmin  /usr/share/phpmyadmin
# par quelque chose comme ci-dessous (avec accents ou pas? !)
Alias /nom-accès-personnalisé  /usr/share/phpmyadmin
```

Pour créer un utilisateur avec des droits, aller à `Privilèges` puis `Ajouter un utilisateur.`

- Créez l'utilisateur `Talla,`
- d'où il peut se connecter : choisir `local`
- Mettez le mot de passe de l'utilisateur `Talla`
- Cliquez sur `créer un compte utilisateur` et visualiser après la requête SQL associée à votre action qui peut vous permettra de connaitre les droits de l'utilisateur sur les BD.

- Modifiez ensuite ses droits pour qu'ils ne soient appliqués qu'a la base de données `ING2` en cliquant sur `Changer les privilèges.`
- Donnez les privilèges `SELECT, INSERT, UPDATE` et `DELETE` sur la base `ING2.`
- Ce qui peut générer la requête suivante qu'on aurait pu exécuter en tant que root en se connectant avec la ligne de commande :

```
GRANT SELECT, INSERT, UPDATE et DELETE  ON ING2 * TO 'Talla' '@'localhost';
```

Connectez-vous en ligne de commande avec l'utilisateur `Talla` pour vérifier ses droits.

```
# mysql –u root –p

mysql>use ING2

mysql>show grants ;
```

Essayez d'insérer une ligne dans la table utilisateurs et vérifier si l'opération s'est bien passée.

Etape 4 : Les injections SQL

Une injection SQL consiste à faire exécuter par le serveur des requêtes SQL qui n'étaient pas initialement prévues. Pour l'illustrer, utilisez le script suivant qui a pour fonction de fournir le Nom et le Prénom d'un utilisateur connaissant son identifiant ainsi que son mot de passe.

Identification.php

```
< !DOCTYPE html>
<html>
<head>
<meta charset= « UTF-8 »>
<title> identification </title>
</head>
<body>
<h1>identification</h1>
<p>Saisissez  votre identifiant et votre mot de passe : </p>
< !---formulaire de saisie de l'identifiant et du mot de passe- - >
<form method='POST' name= "identif">
        Identifiant: <input type="text" name="identif" size="30"><br>
        Mot de passe: <input type="password" name="mdp" size="30"><br>
        <input type="hidden" name="ref" value="identification"><br>
        <input type="submit" name="valider" value="valider"><br>
</form>
<hr>
<! - - passage en php pour traitement du formulaire - ->
< ?php
//traitement du formulaire
If($_POST/['ref']=='identification') {
                    //recupération des données du formulaire
                      Echo « Traitement du formulaire<br> » ;
                      $identif=$_POST['identif'] ;
                     $mdp=*_POTS['mdp'] ;
                    //Avec la version 5.1 de PHP, il est conseillé d'utiliser
PDO pour se connecter au BD
                    Try{
                          $con=new PDO('mysql :host=localhost;
                           Dbname=ING2 ',' Talla','several');
                           Echo « connexion au serveur de BD<br> » ;
                           //envoie de la requête
                          $req= « SELECT nom, prenom FROM utilisateurs WHERE
identifiant='".$identif."'
                           AND '".$mdp."';";
                           Echo 3envoie de la requête $req <br> » ;
                       $rep=$con->query($req) ;
                       If($rep)
                          {
                        $lignes=$rep->fetchAll;
```

```
                        If($lignes)
                                    {
                        echo"<hr>";
                       echo. "votre nom est ".$lignes[0]['nom']." et votre
mot de passe est ".$lignes[0]['prenom']."
                                     }
                     Else {echo « Erreur SQL» ;}
                       }
                       else{echo « Erreur BD» ;}
                  $con=null;
                    }
Catch (PDOExeption $e)
                                          {
   print "Erreur de connexion! <br> » ;
   Die() ;
}
} ?>
</body>
< /html>
```

- Enregistrez le fichier dans un endroit accessible par le serveur Apache2 (exemple dans `/home/ING2/www`)
- Tester en utilisant un identifiant comme identifiant `free` et mot de passe `aaa.`
- Vérifiez que le script ne fonctionne pas pour les utilisateurs avec les mot de passe cryptés comme cela n'est pas pris en compte dans la requête envoyée au serveur.
- Si un comportement inattendue est obtenu du serveur, on pourra parler de faille de sécurité. Les principe, sera de fermer la requête en insérant une simple quote puis en la complétant afin d'obtenir une autre réponse.
- Commencez par tester si le formulaire laisse passer une simple quote dans ses champs(identifiant ou mot de passe etc.)
- Testez avec un nom d'utilisateur quelconque et comme mot de passe 'or 1=1# ; et vérifiez si le premier utilisateur va apparaitre.
- Pour trouver l'identifiant et le mot de passe du premier utilisateur, on peut essayer de compléter la requête pour en déclencher une autre en utilisant la commande `UNION` de MySQL avec un nombre de champs identique pour chaque requête. Mettez dans le champs mot de passe '`UNION SELECT login, password FROM utilisateurs #` et ainsi la requête qui sera envoyée au serveur deviendra : `SELECT nom, prenom FROM utilisateurs WHERE login='free' AND password=' ' UNION SELECT login, password FROM utilisateurs # ;`

`

Exposé 5

Sécurité des réseaux

Sujet : Sécurité de TELNET, SSH, Points d'accès Wifi etc..

L'objectif de cet exposé est de sniffer un mot de passe avec Wireshark qui est saisis depuis un ordinateur distant à travers une interface ligne de commande, une interface web ou un Terminal comme Putty ou Teraterm.
Il sera étudié les cas de l'authentification sur un serveur Telnet, SSH (via la CLI ou un Terminal) et sur un point d'accès (à travers une interface web).

La procédure décrite ici ne fait qu'office de guide, et il vous reviendra au cas où elle ne marche pas, de chercher une solution et de décrire étape par étape une version finale dans un rapport à rendre avant la date indiquée

Partie 1: sécurité de TELNET et de SSH

L'objectif de cette partie est de configurer un routeur ou un ordinateur pour l'accès Secure Shell (SSH) et Telnet puis d'analyser une session Telnet et une session SSH avec Wireshark.

, SSH doit remplacer Telnet pour les connexions relatives à la gestion. Telnet utilise des communications non sécurisées en texte clair. SSH assure la sécurité des connexions distantes en fournissant un chiffrement efficace de toutes les données transmises entre les périphériques. Dans cet exercice, vous allez tester la sécurité de l'accès à un commutateur ou un routeur ou un ordinateur distant selon le protocole utilisé (Telnet, SSH, http).

Le principe consiste à configurer un routeur ou commutateur ou l'ordinateur pour qu'il accepte les connexions Telnet ou SSH, puis utiliser Wireshark pour capturer et afficher des sessions Telnet et SSH.

1. Connecter physiquement et logiquement (adressage IP) le PC au routeur/switch/Ordinateur puis tester la communication avec la commande ping.
2. Configurer le routeur pour qu'il accepte les connexions Telnet et SSH sur les lignes VTY
   - Configurez le nom du périphérique.
   - `Router(config)# hostname R1`
   - Configurez le domaine du périphérique.
   - `R1(config)# ip domain-name univ-zig.sn`
   - Configurez la méthode de la clé de chiffrement.
   - `R1(config)#crypto key generate rsa`

```
The name for the keys will be: R1.univ-zig.sn
Choose the size of the key modulus in the range of 360 to 2048 for your   General
Purpose Keys. Choosing a key modulus greater than 512 may take   a few minutes.
 How many bits in the modulus [512]:
```

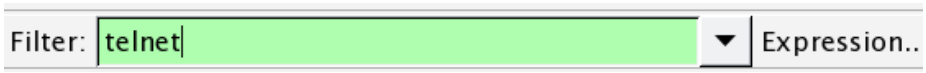
   - Taper 1024 comme taille de la clé

4. Configurez un nom d'utilisateur de base de données locale.
   - `R1(config)# username admin privilege 15 secret adminpass`
   - Remarque: un privilège de niveau 15 offre à l'utilisateur des droits d'administrateur
5. Activez SSH sur les lignes VTY.

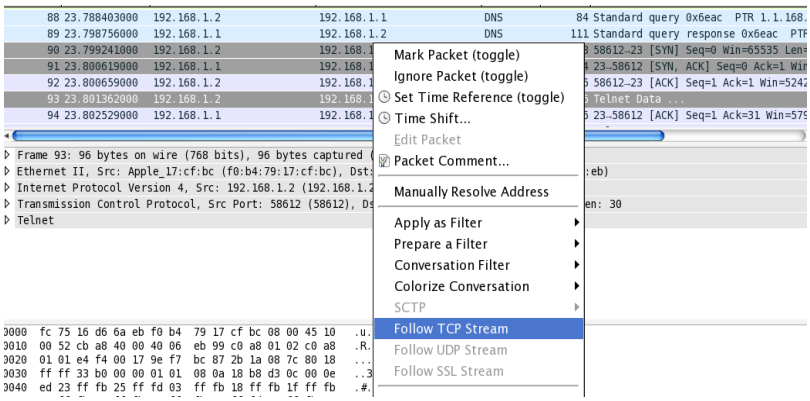
- Activez Telnet et SSH sur les lignes VTY entrantes à l'aide de la commande **`transport input`**
- `R1(config)#line vty 0 4`
- `R1(config-line)#transport input telnet`
- `R1(config-line)#transport input ssh`
- Modifiez la méthode de connexion de façon à ce que la base de données locale soit utilisée pour la vérification de l'utilisateur.
- `R1(config-line)# login local`
- `R1(config-line)# end`
- Pour se connecter au retour ou au switch via la ligne de commande de l'ordinateur Packet Tracer
- `C:\>ssh -l admin AdresseIpRouteur`
- Ou avec votre ordinateur
- `C:\>ssh -l admin@AdresseIpRouteur`

6. Si vous utilisez un ordinateur comme serveur SSH, vous devez installer le serveur SSH sur Ubuntu par exemple avec la commande :
   - `$ sudo apt-get install openssh-server.`

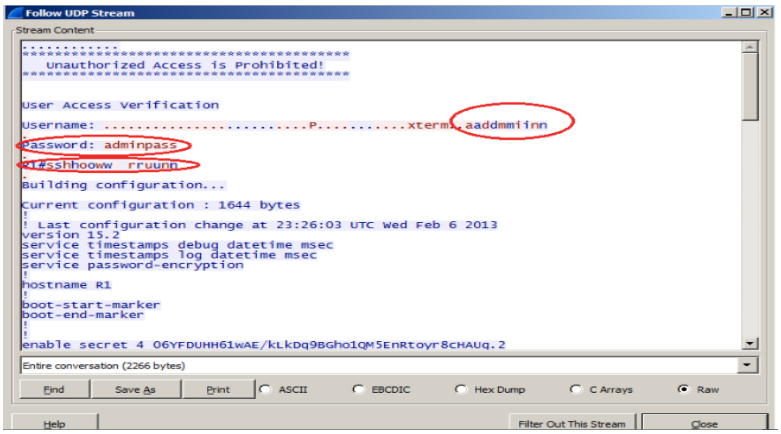
7. Analyser une session Telnet avec Wireshark
   - Installer le logiciel Wireshark,
   - Installer Teraterm ou Putty sur votre ordinateur Windows pour se connecter au routeur
   - Démarrez une session Telnet avec votre terminal Putty ou Teraterm ou la ligne de commande pour accéder à l'interface du serveur Telnet (routeur ou switch ou ordinateur).
   - Si vous utilisez Putty ou Teraterm, sélectionnez la case d'option Service Telnet et dans le champ Hôte, entrez L'ip du serveur Telnet (exemple192.168.1.1)
   - Arrêtez la capture Wireshark, et appliquez un filtre Telnet sur les données de capture Wireshark.



- Utilisez la fonction Follow TCP Stream dans Wireshark pour afficher la session Telnet

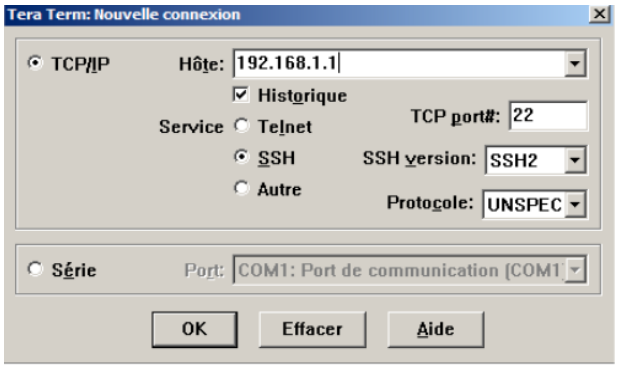


- La fenêtre Follow TCP Stream (Suivre le flux TCP) affiche les données de votre session Telnet avec le routeur.
- La session complète s'affiche en texte clair, y compris votre mot de passe. Notez que le nom de l'utilisateur s'affiche avec des caractères en double. Cela provient du paramètre d'écho dans Telnet qui vous permet d'afficher les caractères que vous tapez à l'écran.

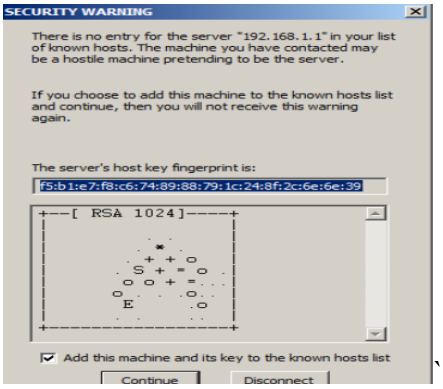


8. Analyser une session SSH avec Wireshark

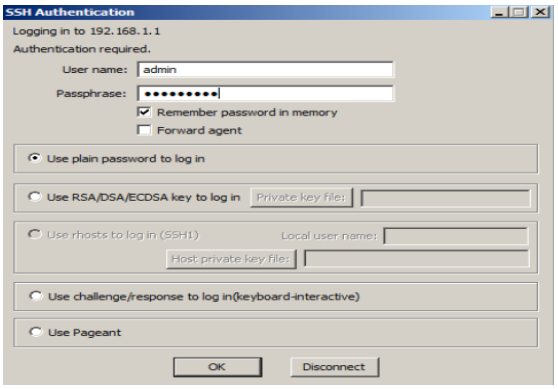
- Ouvrez Wireshark et commencez à capturer des données sur l'interface LAN.
- Démarrez une session SSH sur le routeur via votre terminal et entrez l'adresse IP de l'interface du routeur puis vérifiez que la case d'option SSH est sélectionnée, puis cliquez sur OK pour vous connecter au routeur.



- La première fois que vous avez établi une session SSH à un périphérique, un avertissement de sécurité (SECURITY WARNING) vous informe que vous ne vous êtes pas encore connecté à ce périphérique. Ce message fait partie du processus d'authentification. Lisez l'avertissement de sécurité, puis cliquez sur Continue.



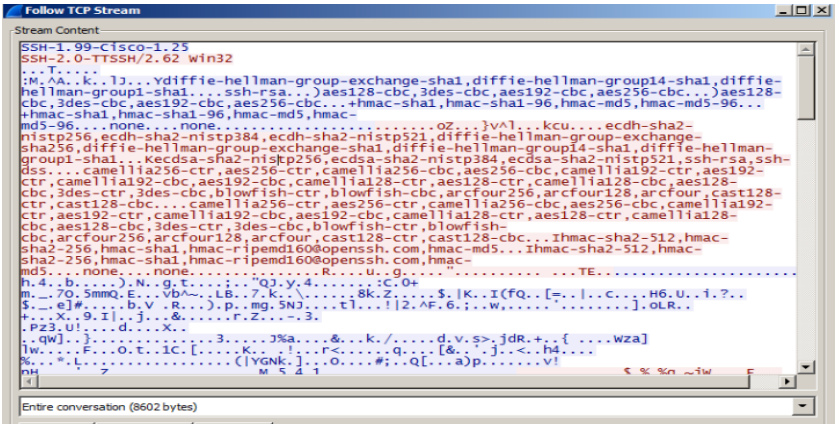
- Dans la fenêtre d'authentification SSH, entrez admin comme nom d'utilisateur et adminpass pour le mot de passe. Cliquez sur OK pour accéder au routeur



- Arrêtez la capture Wireshark et appliquez un filtre SSH sur les données de capture Wireshark.

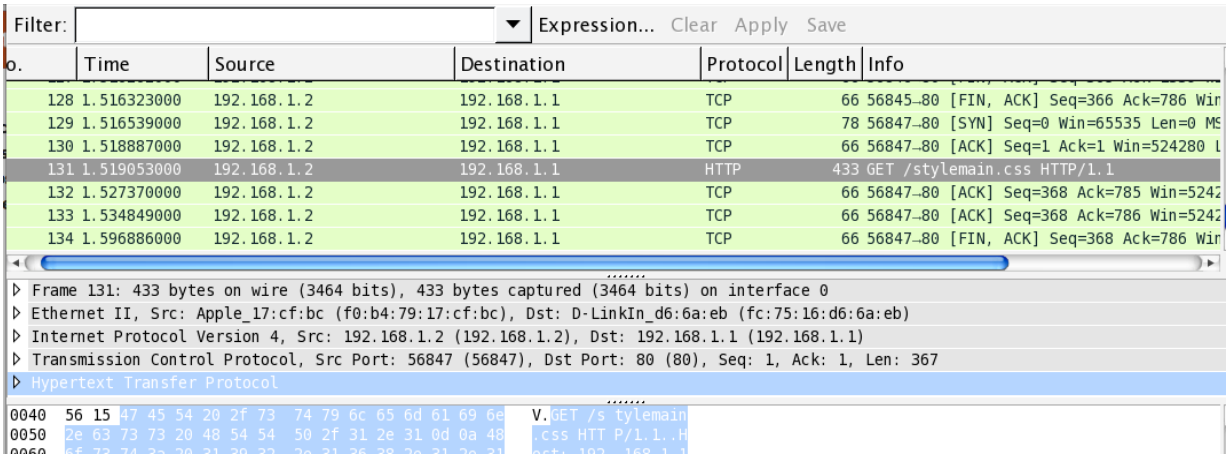
-

- Utilisez la fonction Follow TCP Stream dans Wireshark pour afficher la session SSH.
- Cliquez avec le bouton droit sur l'une des lignes SSHv2 dans la section Packet list (Liste des paquets) de Wireshark et, dans la liste déroulante, sélectionnez l'optionFollow TCP stream (Suivre le flux TCP)
- Examinez la fenêtre Follow TCP Stream (Suivre le flux TCP)de votre session SSH. Les données ont été chiffrées et sont illisibles. Comparez les données de votre session SSH aux données de votre session Telnet



9. Analyser une session http d'un point d'accès avec Wireshark

- Si on suppose que l'interface web du point d'accès est accessible via l'url http://192.168.1.1 par exemple.
- Lancer l'analyseur et faites une capture en ne capturant que les paquets en provenance ou à destination de 192.168.1.1.
- Ouvrir le navigateur et saisir http://192.168.1.1 dans la barre d'adresse.
- Saisir le nom d'utilisateur et le mot de passe pour accéder aux paramètres de configuration du AP
- Dans la fenêtre contenant la liste des trames capturées, sélectionner le protocole http avec la méthode GET.
- Dans la fenêtre d'affichage de la pile des protocoles décodés, cliquer sur hypertex transfer Protocol.



Dans la fenêtre des protocoles décodés, cliquer sur ***Authorization***, dans ***Credentials*** se trouvent les identifiants de l'utilisateur.