

TP SSH

2.1 Faire une connexion ftp et remarquer la présence des mots de passe en clair.

Nous commençons par installer un serveur ftp sur la machine serveur :

```
sudo apt install vsftpd
```

Puis nous éditons sa configuration :

```
nano /etc/vsftpd.conf
```



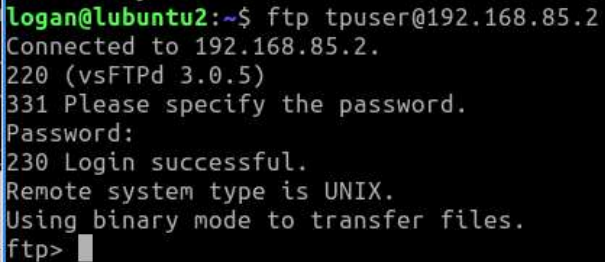
```
#  
# Allow anonymous FTP? (Disabled by default).  
anonymous_enable=NO  
#  
# Uncomment this to allow local users to log in.  
local_enable=YES  
#  
# Uncomment this to enable any form of FTP write command.  
write_enable=YES  
#
```

On utilise les utilisateurs unix locaux et pas l'anonymus

Enfin, on démarre le daemon ftp :

```
sudo systemctl restart vsftpd.service
```

Il est maintenant possible de se connecter en ftp sur le client : `ftp tpuser@192.168.85.2`



```
logan@lubuntu2:~$ ftp tpuser@192.168.85.2  
Connected to 192.168.85.2.  
220 (vsFTPd 3.0.5)  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

Le répertoire affiché est le /home/ de l'utilisateur courant

Capturons les paquets échangés lors de cette connexion. Pour cela, nous utilisons la commande `tcpdump` en indiquant l'interface utilisée, et le port 21 qui correspond au protocole ftp par défaut, avec un filtre GREP pour l'occurrence « FTP » :

```
sudo tcpdump -s0 -i eth0 tcp and port 21 | grep FTP
```

```

logan@lubuntu1:~$ sudo tcpdump -s0 -l ens37 tcp and port 21 | grep FTP
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens37, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:54:50.219270 IP lubuntu1.ftp > lubuntu2.44820: Flags [P.], seq 1:21, ack 1
, win 510, options [nop,nop,TS val 2311632480 ecr 4238467476], length 20: FTP
: 220 (vsFTPD 3.0.5)
09:54:50.220875 IP lubuntu2.44820 > lubuntu1.ftp: Flags [P.], seq 1:14, ack 2
1, win 16384, options [nop,nop,TS val 4238467498 ecr 2311632480], length 13:
FTP: USER tpuser
09:54:50.221629 IP lubuntu1.ftp > lubuntu2.44820: Flags [P.], seq 21:55, ack
14, win 510, options [nop,nop,TS val 2311632482 ecr 4238467498], length 34: F
TP: 331 Please specify the password.
09:54:53.710202 IP lubuntu2.44820 > lubuntu1.ftp: Flags [P.], seq 14:27, ack
55, win 16384, options [nop,nop,TS val 4238470987 ecr 2311632482], length 13:
FTP: PASS secret
09:54:53.963706 IP lubuntu1.ftp > lubuntu2.44820: Flags [P.], seq 55:78, ack
27, win 510, options [nop,nop,TS val 2311636224 ecr 4238470987], length 23: F
TP: 230 Login successful.
09:54:53.968884 IP lubuntu2.44820 > lubuntu1.ftp: Flags [P.], seq 27:33, ack

```

Le paquet FTP est en clair, le mot de passe est « secret »

2.2 Être assuré de l'identité du serveur lors de la première connexion client/serveur ssh.

D'abord, nous essayons de nous connecter au serveur en ssh :

```
ssh server
```

Il nous est demandé de vérifier l'empreinte du serveur afin d'établir son authenticité :

```

thibaut@Ubuntu:~$ ssh server
The authenticity of host 'server (192.168.46.2)' can't be established.
ED25519 key fingerprint is SHA256:XYdRbWhCA80Mq6Ph223jQGZr8UZeNLjCHQIKlQITz3g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?

```

Afin de récupérer l'empreinte du serveur, nous nous connectons dessus puis extrayons l'empreinte de la clé publique ED25519 (nous voyons dans la capture d'écran précédente qu'il s'agit de la clé utilisée, ED25519 est l'algorithme utilisé). Cela peut se faire grâce à la commande

```
ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub
```

Où :

- `ssh-keygen` est l'utilitaire de gestion des clés SSH,
- `-l` permet d'extraire l'empreinte digitale d'une clé publique
- `-f` permet de spécifier le fichier contenant la clé publique

```

thibaut@server:~$ ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub
256 SHA256:XYdRbWhCA80Mq6Ph223jQGZr8UZeNLjCHQIKlQITz3g root@Ubuntu (ED25519)

```

Nous remarquons que les empreintes correspondent : nous sommes donc assurés de l'identité du serveur. Nous pouvons copier cette clé et la renseigner au client (ou simplement entrer « yes »). Ainsi, la signature du serveur sera ajoutée dans `~/ssh/known_hosts` et l'identité du serveur ne sera plus redemandée par la suite.

Après avoir entré le mot de passe correspondant à l'utilisateur courant sur le serveur, nous y sommes bien connectés en ssh :

```
thibaut@Ubuntu:~$ ssh server
The authenticity of host 'server (192.168.46.2)' can't be established.
ED25519 key fingerprint is SHA256:XYdRbWhCA8OMq6Ph223jQGZr8UZeNLjCHQIKlQITz3g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? SHA256:XYdRbWhCA8OMq6Ph223jQGZr8UZeNLjCHQIKlQITz3g
Warning: Permanently added 'server' (ED25519) to the list of known hosts.
thibaut@server's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

La maintenance de sécurité étendue pour Applications n'est pas activée.

8 mises à jour peuvent être appliquées immédiatement.
7 de ces mises à jour sont des mises à jour de sécurité.
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgradable

Activez ESM Apps pour recevoir des futures mises à jour de sécurité supplémentaires.
Visitez https://ubuntu.com/esm ou exécutez : sudo pro status

1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Thu May  2 11:44:43 2024
thibaut@server:~$
```

2.3 Permettre une connexion client/serveur ssh sans phase d'authentification.

Pour autoriser une connexion ssh au serveur (en tant qu'un utilisateur donné) sans phase d'identification, nous devons ajouter sur le serveur la clé publique du client, dans le fichier `~/.ssh/authorized_keys`.

Générons d'abord une trousseau de clés asymétrique sur le client :

```
ssh-keygen -t ed25519
```

- `-t` permet d'indiquer le type de clé. Nous utilisons ici des clés ed25519 (dans la continuité du début du TP)

```
thibaut@Ubuntu:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/thibaut/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/thibaut/.ssh/id_ed25519
Your public key has been saved in /home/thibaut/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:sM/iNyXJrL/7F6qMfVm/seOHTQ4BCbCdLrrusQGuIW8 thibaut@Ubuntu
The key's randomart image is:
+--[ED25519 256]--+
|
|      . . . . .
|      o .o
|      . . o .
|      o . .
|      . ..S.. .
|      . . ++. . + .
|      . . =.oo. + X
|      oEo . *+= .+ o.B
|      .o  oB+o.=+...++
+-----[SHA256]-----+
```

La clé a bien été générée :

```
thibaut@Ubuntu:~$ ls -l ~/.ssh | grep id_ed25519
-rw----- 1 thibaut thibaut 411 mai 23 10:17 id_ed25519
-rw-r--r-- 1 thibaut thibaut 96 mai 23 10:17 id_ed25519.pub
```

Maintenant, il ne reste qu'à la copier la clé publique sur le serveur, dans ~/.ssh/authorized_keys, à l'aide de la commande ssh-copy-id :

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub user@server
```

- `-i` permet d'indiquer que nous souhaitons seulement ajouter l'identité correspondant à la clé `id_ed25519.pub`.

```
thibaut@Ubuntu:~$ ssh-copy-id -i ~/.ssh/id_ed25519.pub thibaut@server
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/thibaut/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install
the new keys
thibaut@server's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'thibaut@server'"
and check to make sure that only the key(s) you wanted were added.
```

Maintenant, nous pouvons nous connecter au serveur sans phase d'authentification :


```

thibaut@Ubuntu:~$ ssh server
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

La maintenance de sécurité étendue pour Applications n'est pas activée.

8 mises à jour peuvent être appliquées immédiatement.
7 de ces mises à jour sont des mises à jour de sécurité.
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgradable

Activez ESM Apps pour recevoir des futures mises à jour de sécurité supplémentaires.
Visitez https://ubuntu.com/esm ou exécutez : sudo pro status

1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Thu May 23 09:52:34 2024 from 192.168.46.3
thibaut@server:~$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIASCGIu/9+j/xUdV+d9b9Dofdw+vPjK29vKd5EC1yyPl thibaut@Ubuntu
thibaut@server:~$

```

2.4 Avoir la possibilité de télécharger le fichier readme via wget et protocole ftp

Pour télécharger un document avec wget via ftp, on utilise le protocole ftp sous la forme d'URL
 user:mdp@serveur/path

```

logan@lubuntu2:~$ wget ftp://tpuser:secret@192.168.85.2/bonjour.txt
--2024-05-16 12:21:12--  ftp://tpuser:*password*@192.168.85.2/bonjour.txt
=> 'bonjour.txt'
Connexion à 192.168.85.2:21... connecté.
Ouverture de session en tant que tpuser... Session établie.
==> SYST ... terminé.      ==> PWD ... terminé.
==> TYPE I ... terminé.    ==> CWD n'est pas nécessaire.
==> SIZE bonjour.txt ... 8
==> PASV ... terminé.      ==> RETR bonjour.txt ... terminé.
Taille : 8 (non certifiée)

bonjour.txt          100%[=====]          8  ---KB/s      ds 0,002s
2024-05-16 12:21:13 (3,18 KB/s) - 'bonjour.txt' enregistré [8]

logan@lubuntu2:~$ cat bonjour.txt
bonjour
logan@lubuntu2:~$

```

Le fichier bonjour.txt téléchargé via ftp

2.5 Facultatif : avoir la possibilité de télécharger le fichier index.html via wget et protocole http (l'utilisation d'un navigateur web comme firefox est également possible)

L'utilisation de wget aussi simple que de mettre l'URL http en tant que premier argument.

```

logan@ubuntu2:~$ wget http://http.kagescan.fr/index.html
--2024-05-16 12:23:20-- http://http.kagescan.fr/index.html
Résolution de http.kagescan.fr (http.kagescan.fr): 45.140.165.37
Connexion à http.kagescan.fr (http.kagescan.fr)[45.140.165.37]:80_ connecté.
requête HTTP transmise, en attente de la réponse_ 200 OK
Taille : 26 [text/html]
Enregistre : 'index.html'

Index.html      100%[=====]      26  --.-KB/s   ds 0s

2024-05-16 12:23:21 (433 KB/s) - 'index.html' enregistré [26/26]

logan@ubuntu2:~$ cat index.html
un site en http sans le s
logan@ubuntu2:~$

```

Le fichier index.html est téléchargé

2.6 Faire une connexion sftp et remarquer la présence des mot-de-passes encryptés.

Avant de se connecter en sftp, nous retirons l'empreinte de la clé publique du client sur le serveur (nous voulons avoir à entrer le mot de passe).

Nous faisons la connexion sftp simplement avec la commande suivante :

```
sftp server
```

```

thibaut@Ubuntu:~$ sftp server
thibaut@server's password:
Connected to server.
sftp>

```

En écoutant grâce à tcpdump sur le port 22 (sftp utilise ssh, donc ce port), nous remarquons que la connexion est totalement encryptée, il nous est impossible de lire en clair le mot de passe :

```

0x0000: 177a c9c4 2b34 cfe8 aaaa 018e 7ca6 9992 72..+4.....
0x0000: 448e 0b8a bf36 ..C.8.....E.
11:02:11.069721 IP server.ssh > Ubuntu.57378: Flags [P.], seq 2874:2958, ack 3822, win 501, options [nop,nop,TS val 540543785 ecr 2446569659], length 76
0x0000: 0000 2743 b438 0800 27fd c1a1 0800 4508 ..C.8.....E.
0x0010: 0000 0d1e 4000 4000 e9fb c8a8 2e02 c0a0 ..P.8.0.....E.
0x0020: 2e03 0016 e022 30ac e03b 0b84 72d5 8018 ....0.....f....
0x0030: 01f5 37f9 0000 0101 080a 2038 0ad9 91d3 ..7.....8.....
0x0040: 00bb 9770 51d7 cf43 5b12 c9df d32d 0c9d ...VQ..C[....I.
0x0050: 0175 fb8b 1324 0738 0702 03f4 bb40 7cc7 ..u...$.X..H|.
0x0060: 38ca 328c cf0c 0b0a b800 adc3 70bd f2d3 K.2...f...p...
0x0070: b90d 8252 50db c73e 0b07 9074 00eb 96d1 ...RP...g.t....
0x0080: b507 4d13 bace b90d 2b2a d5bc 083f ..R.....+*...7
11:02:11.071871 IP Ubuntu.57378 > server.ssh: Flags [P.], seq 3022:3074, ack 2958, win 501, options [nop,nop,TS val 2446569663 ecr 540543785], length 52
0x0000: 0000 27fd c1a1 0800 2743 b438 0800 4508 ..C.8.....E.
0x0010: 0000 d0de 4000 4000 0031 c0a8 2e03 c0a8 ..h..8.0.....8
0x0020: 2e02 e022 0016 0b84 72d1 30ac e0b7 8018 ....f.0.....8
0x0030: 01f5 d0b0 0000 0101 080a 91d3 b0bf 2038 ..7.....8.....
0x0040: 8ad9 7h10 0e4c 300e da37 ac0b 40fe 645c ....10..TL.H.3|
0x0050: e8b0 78aa b0ad d892 89c7 a7c2 a072 1ffa ..k..a.....f...
0x0060: e0c2 5db2 5016 79dc 3075 f674 ca70 c2c8 ..V.y.0u.t.z...
0x0070: 99d7 fdb2 39ca ....0.....
11:02:11.072689 IP server.ssh > Ubuntu.57378: Flags [P.], seq 2958:3042, ack 3074, win 501, options [nop,nop,TS val 540543786 ecr 2446569663], length 92
0x0000: 0000 2743 b438 0800 27fd c1a1 0800 4508 ..C.8.....E.
0x0010: 0000 0d1f 4000 4000 e9fa c8a8 2e02 c0a0 ..P.8.0.....E.
0x0020: 2e03 0016 e022 30ac e03b 0b84 72d5 8018 ....0.....f....
0x0030: 01f5 d184 0000 0101 080a 2038 0ad9 91d3 ..7.....8.....
0x0040: b0bf 420b 0e4c 80b0 80b7 c39a 0d13 e0c9 ..B..f.....
0x0050: 1350 960b e08e c20a 005c 407d 52e6 d7b1 ..P.k.....}R...
0x0060: 0130 ff1c bd51 ecf0 e0d4 90fe ab9d 8ae8 ..8...Q.....
0x0070: 230b fad0 d572 17dd ad93 0097 baac 9511 #....f.....
0x0080: 74b6 ba2b 6e4f 307d a889 040a b09f 86da T..v00.....]...
0x0090: 0982 d48a 0e30 d916 b9f8 d5c2 c92c .....0.....
11:02:11.115248 IP Ubuntu.57378 > server.ssh: Flags [P.], seq 3042, win 501, options [nop,nop,TS val 2446569706 ecr 540543786], length 8
0x0000: 0000 27fd c1a1 0800 2743 b438 0800 4508 ..C.8.....E.

```

2.7 Facultatif: faire une connexion http et remarquer la présence d'un transfert en clair.

Filtrons les paquets http utiles (filtre port 80 avec un contenu supérieur à 200 octets) :

```
sudo tcpdump -A -s0 -i ens33 tcp and port 80 and greater 200
```

```
logan@lubuntu1:~$ sudo tcpdump -A -s0 -i ens33 tcp and port 80 and greater 200
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
10:15:14.773847 IP 45.140.165.37.http > lubuntu1.57244: Flags [P.], seq 17551
96626:1755196961, ack 3492262733, win 64240, length 335: HTTP: HTTP/1.1 200 O
K
E..w.....8...%...P..h,...'.MP...h...HTTP/1.1 200 OK
Date: Thu, 23 May 2024 08:14:36 GMT
Server: Apache/2.4.41 (Ubuntu)
Upgrade: h2,h2c
Connection: Upgrade, Keep-Alive
Last-Modified: Mon, 22 May 2023 07:01:47 GMT
ETag: "1a-5fc42d8178f73"
Accept-Ranges: bytes
Content-Length: 26
Keep-Alive: timeout=5, max=100
Content-Type: text/html

un site en http sans le s
```

On voit ici la réponse en clair !

2.8 Facultatif: faire un tunnel ssh/http et remarquer la présence du même transfert encrypté.

Créons un tunnel SSH pour faire en sorte que le port 80 de la machine distante soit forwardé vers le port 8080 de la machine cible :

```
Ssh -L 8080:localhost:80 server
```

Lançons une écoute active TCP sur le NAT de notre VM (ens33) :

```
tcpdump -XX -s0 -i
```

Après avoir fait la requête, on constate que la réponse est illisible :

```

10:20:27.643495 IP 45.140.165.37.31322 > lubuntu1.50630: Flags [P.], seq 45:6
25, ack 248, win 64240, length 580
 0x0000: 000c 29b0 34a3 0050 56e9 bb52 0800 4500 ..).4..PV..R..E.
 0x0010: 026c 01ce 0000 8006 24df 2d8c a525 c0a8 .l.....$.-%..
 0x0020: 7e85 7a5a c5c6 5fba 8d5b 7c40 b969 5018 ~.zZ.._..[|@.iP.
 0x0030: faf0 acdd 0000 bb99 217b f513 2eba 7602 .....!{....v.
 0x0040: 8ee6 475e 08af f5e1 2f44 abf3 406a d2ae ..G^.... /D..@j..
 0x0050: a088 c47f c5d3 356e cea4 e127 bf42 e953 .....5n...'.B.S
 0x0060: bc5b ffcd 547a 8f1a 3b4d 9b58 633d 2419 .[..Tz...;M.Xc=$.
 0x0070: c92d bc09 002c 952c 6f64 1621 d436 997c .-....,od!.6.|
 0x0080: 486c d069 183a d838 3a37 c2a7 e791 a77b Hl.i.:8:7.....{
 0x0090: 3196 65b7 dd7a 9b8d 5761 9470 948c e5a3 1.e...z...Wa.p....
 0x00a0: 8467 2514 9745 ed03 2913 74ee 71a9 d430 .g%..E..).t.q..0
 0x00b0: 9ca9 3565 827f ed2d a8e7 8c3a 0100 4c51 ..5e...-....LQ
 0x00c0: cfd9 9003 7274 81cb 8f6d d0e1 1f2e d704 ....rt...m.....
 0x00d0: ac26 c2b4 7017 a296 a25e 5319 88fc 9b11 .&..p....^S.....
 0x00e0: 7e02 eb95 d99d f6b4 8fec d835 ba82 11b1 ~.....5....
 0x00f0: d83d e2e6 3e20 14ec 21c5 2c59 174f f2e8 .=>...!.,Y.O..
 0x0100: 8130 3b41 dacc 2d4e 7848 c2d1 b240 aedc .0;A...-NxH...@..
 0x0110: acd2 69e4 0dbd e7c9 2c69 24fc 9cdd fa2e ..i.....,i$.....
 0x0120: dae9 41d5 fe61 e47b db65 faf1 7b0c 3e1a ..A..a.{.e..{.>.
 0x0130: cfec a117 dccc 93db 1a0c 2711 5c2d 94c0 .....'\-..
 0x0140: 6fe9 8a20 ae41 b605 4423 f6c5 0ea7 f679 o....A..D#.....y

```

Résultat écoute du port : on voit que la réponse est illisible !