

Le protocole Telnet

- ✎ Le **protocole Telnet** (**Telecommunication Network**) est un protocole utilisé sur les réseaux informatiques pour **se connecter à distance à un serveur ou à un équipement réseau**. Une fois connecté à un équipement avec le protocole Telnet, on obtient **un accès à un prompt** afin de pouvoir **saisir et exécuter des commandes**.
- ✎ Le protocole Telnet est **un protocole de type client-serveur**, où les connexions sont effectuées sur **le port 23 en TCP**.
- ✎ Une machine disposant d'un serveur telnet (ex. telnetd sous Linux) permettra à n'importe quelle machine de part le réseau de s'y connecter, au moyen d'un client telnet (peut être représenté par l'ordinateur de l'administrateur système).
- ✎ le serveur Telnet peut être représenté aussi par un équipement réseau que l'on veut administrer à distance.
- ✎ Les clients telnet existent sur la quasi-totalité des plates formes (Windows, Linux, Unix, MacOS...).

37

Le protocole Telnet

- ✎ Il s'agit de l'un des protocoles les plus anciens, puisqu'il **a été créé en 1969** avant d'obtenir sa certification RFC le 1er mai 1973 : la **RFC 495**. Par la suite, deux autres RFC ont été mises en ligne pour mieux décrire le protocole et tenir compte des améliorations : **RFC 854** et **RFC 855**.
- ✎ Aujourd'hui, le protocole Telnet est utilisé pour **se connecter à un commutateur ou un routeur en ligne de commande**, dans le but de l'administrer.

Remarque:

- Le protocole Telnet, qui reste aujourd'hui utilisé pour effectuer de l'administration à distance, bien que ce ne soit pas un protocole **sécurisé**.

38

Faiblesse du protocole Telnet

Telnet, un protocole non sécurisé

- ✎ Le protocole Telnet a été développé à une époque où la sécurité n'était pas une préoccupation. De ce fait, le protocole Telnet n'est pas sécurisé. Pour être plus précis, **toutes les données échangées via Telnet sont transmises en clair sur le réseau**, c'est-à-dire qu'elles ne sont pas chiffrées.
- ✎ Cela signifie que si l'on utilise le protocole Telnet pour se connecter à un équipement réseau ou un serveur, les informations sensibles (**nom d'utilisateur et mot de passe**) seront transmises en clair sur le réseau. Si une personne malveillante parvient à **intercepter le trafic réseau**, elle sera en mesure de **recupérer vos identifiants** et de compromettre votre équipement.
- ✎ L'utilisation du protocole Telnet reste acceptable sur un réseau local, quand il n'y a pas d'autres alternatives, mais sur Internet, c'est à bannir. Dans tous les cas, il est important d'avoir **connaissance de ce risque** et de le prendre en considération.

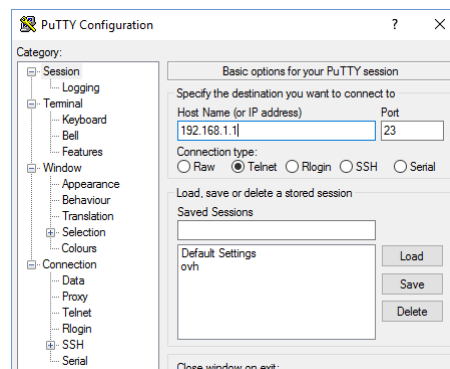
39

Faiblesse du protocole Telnet

Capture des identifiants avec Wireshark

- ✎ Soit l'exemple pratique suivant: on va établir une connexion Telnet depuis un PC Windows vers un équipement. Dans le même temps, une capture du trafic sera réalisée à partir du PC Windows qui initie la connexion Telnet.
- ✎ La connexion est initiée avec l'application PuTTY... Elle est établie, le login et le mot de passe sont saisis. Juste après, la capture Wireshark est arrêtée.

- ✎ Si l'on s'intéresse de plus près aux paquets échangés entre mon PC Windows et mon équipement, autrement dit mon client Telnet et mon serveur Telnet, on peut voir plusieurs paquets "Telnet Data...". Si l'on regarde le détail des paquets, on peut constater "Password:" comme données dans le paquet, ce qui correspond au prompt visible sur la console ci-dessus.



Faiblesse du protocole Telnet

Capture des identifiants avec Wireshark

270	10.216151	192.168.1.149	192.168.1.96	TELNET	64	Telnet Data ...
278	10.260260	192.168.1.96	192.168.1.149	TCP	54	55122 → 23 [ACK] Seq=107 Ack=108 Win=0 Len=0
339	13.277832	192.168.1.96	192.168.1.149	TELNET	70	Telnet Data ...


```

> Frame 270: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_{71B8696D-6E7B-4D...}
> Ethernet II, Src: Synology_e6:ca:10 (00:11:32:e6:ca:10), Dst: AzureWav_ad:68:d7 (90:e8:68:ad:68:d7)
> Internet Protocol Version 4, Src: 192.168.1.149, Dst: 192.168.1.96
  > Transmission Control Protocol, Src Port: 23, Dst Port: 55122, Seq: 98, Ack: 107, Len: 10
    Source Port: 23
    Destination Port: 55122
    [Stream index: 10]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 10]
    Sequence Number: 98 (relative sequence number)
    Sequence Number (raw): 3895658314
    [Next Sequence Number: 108 (relative sequence number)]
    Acknowledgment Number: 107 (relative ack number)
    Acknowledgment number (raw): 351509932
    0101 ... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 229
    [Calculated window size: 29312]
    [Window size scaling factor: 128]
    Checksum: 0xc947 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (10 bytes)
  > Telnet
    Data: Password:
  
```

41

Faiblesse du protocole Telnet

Capture des identifiants avec Wireshark

➤ Puis, dans un autre paquet, émit depuis le client Telnet vers le serveur Telnet, on a cette valeur comme donnée : **"Tuto-Telnet-2023"** ! Il s'agit du **mot de passe de l'utilisateur** ! Au préalable, d'autres paquets ont transité, notamment pour l'identifiant (demo-telnet).

270	10.216151	192.168.1.149	192.168.1.96	TELNET	64	Telnet Data ...
278	10.260260	192.168.1.96	192.168.1.149	TCP	54	55122 → 23 [ACK] Seq=107 Ack=108 Win=0 Len=0
339	13.277832	192.168.1.96	192.168.1.149	TELNET	70	Telnet Data ...


```

> Frame 339: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{71B8696D-6E7B-4D...}
> Ethernet II, Src: AzureWav_ad:68:d7 (90:e8:68:ad:68:d7), Dst: Synology_e6:ca:10 (00:11:32:e6:ca:10)
> Internet Protocol Version 4, Src: 192.168.1.96, Dst: 192.168.1.149
  > Transmission Control Protocol, Src Port: 55122, Dst Port: 23, Seq: 107, Ack: 108, Len: 16
    Source Port: 55122
    Destination Port: 23
    [Stream index: 10]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 16]
    Sequence Number: 107 (relative sequence number)
    Sequence Number (raw): 351509932
    [Next Sequence Number: 123 (relative sequence number)]
    Acknowledgment Number: 108 (relative ack number)
    Acknowledgment number (raw): 3895658324
    0101 ... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 1026
    [Calculated window size: 262656]
    [Window size scaling factor: 256]
    Checksum: 0x0b40 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (16 bytes)
  > Telnet
    Data: Tuto-Telnet-2023
  
```

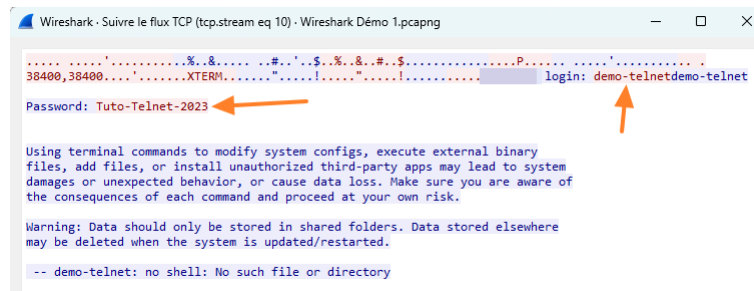
42

Faiblesse du protocole Telnet

Capture des identifiants avec Wireshark

Si l'on utilise la **fonction de suivi de flux TCP de Wireshark**, c'est encore plus flagrant et rapide : toutes les données sont visibles, en clair, dans une fenêtre récapitulative.

On peut récupérer l'identifiant et le mot de passe très facilement.



Cela montre qu'il est **tr s facile de lire les donn es au sein d'une communication entre un client et un serveur** lorsque le protocole Telnet est utilis . Ici, la capture est effectu e depuis le PC qui joue le r le de client Telnet, mais il pourrait s'agir d'une autre machine sur le r seau (qui parvient   se positionner de fa on   intercepter le trafic).

43

Alternatives au protocole Telnet

Pour des raisons de s curit , il est pr f rable d'utiliser d'autres protocoles sur le protocole Telnet, afin d'utiliser un protocole plus moderne et s curiser. La meilleure alternative au protocole Telnet, c'est le protocole **SSH** (Secure SHell). Contrairement au Telnet, le protocole SSH chiffre tous les  changes entre le client et le serveur, offrant ainsi une protection adapt e contre l' coute r seau et l'interception de trafic.

Le protocole SSH, au m me titre que le protocole Telnet, permet de se connecter   distance, en ligne de commande,   un  quipement pour l'administrer. Ceci est vrai pour un  quipement r seau, mais aussi une machine sous Linux ou Windows.

L'exemple ci-dessous montre que les paquets SSH sont chiffr s et le contenu n'est pas lisible.

No.	Time	Source	Destination	Protocol	Length	Info
120	1.853409	192.168.3...	239.255.2...	SSDP	431	NOTIFY * HTTP/1.1
121	1.854056	192.168.3...	239.255.2...	SSDP	494	NOTIFY * HTTP/1.1
428	9.068392	192.168.3...	vps-3c28d...	SSH	210	Client: Encrypted packet (len=64)
431	9.302819	192.168.3...	vps-3c28d...	SSH	118	Client: Encrypted packet (len=64)
444	9.611153	192.168.3...	vps-3c28d...	SSH	118	Client: Encrypted packet (len=64)
447	9.806782	vps-3c28d...	192.168.3...	SSH	118	Server: Encrypted packet (len=64)
450	9.812252	vps-3c28d...	192.168.3...	SSH	118	Server: Encrypted packet (len=64)
457	9.823951	vps-3c28d...	192.168.3...	SSH	118	Server: Encrypted packet (len=64)
459	9.828403	vps-3c28d...	192.168.3...	SSH	262	Server: Encrypted packet (len=208)
480	10.327282	vps-3c28d...	192.168.3...	SSH	228	Server: Encrypted packet (len=224)
483	10.351217	vps-3c28d...	192.168.3...	SSH	118	Server: Encrypted packet (len=64)
487	10.359541	vps-3c28d...	192.168.3...	SSH	166	Server: Encrypted packet (len=112)
852	17.149966	192.168.3...	vps-3c28d...	SSH	134	Client: Encrypted packet (len=80)
858	17.258284	vps-3c28d...	192.168.3...	SSH	166	Server: Encrypted packet (len=112)
650	12.582308	192.168.3...	195.122.1...	SSL	448	Continuation Data
658	12.676560	195.122.1...	192.168.3...	SSL	189	Continuation Data
767	14.645383	192.168.3...	195.122.1...	SSL	464	Continuation Data

44

Alternatives au protocole Telnet

- ✎ Pour des appareils prenant en charge d'autres protocoles, l'administration peut être effectuée via les protocoles HTTP, HTTPS ou RDP, mais l'utilisation dans la pratique sera différente.
- ✎ Le protocole Telnet est à maîtriser, car il fait partie des indispensables protocoles: bien qu'il soit à éviter, vous avez des chances de le croiser alors c'est important d'en savoir un minimum à son principe de fonctionnement.

45

Plan

SSH

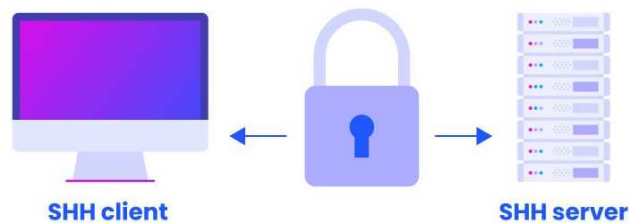
1. Le protocole SSH
2. Les principaux composants de SSH
3. Installation du SSH côté Serveur
4. Configuration du SSH côté Serveur
5. Installation du SSH côté Client
6. Connexion a un Serveur via ssh

46

SSH

SSH (Secure SHell)

- ✎ **Secure Shell** (SSH) est un programme mais aussi un protocole de communication **sécurisé**. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur.
- ✎ Le protocole SSH a été conçu avec l'objectif de **remplacer** les différents protocoles non chiffrés comme rlogin, telnet, rcp et rsh.



47

SSH

SSH (Secure SHell)

Secure Shell SSH est capable de :

- échange de clés de chiffrement
- toutes les trames sont chiffrées
- impossible de lire les trames sur le réseau via un snifer



48

Le Protocole SSH3

En 2023, une alternative à SSH, baptisée SSH3 car elle offre les mêmes services que SSH et s'appuie sur **HTTP/3** et **QUIC** a été proposée.

- **QUIC** est un protocole de transport fiable et sécurisé, en mode connecté, mis au point par Jim Roskind chez Google.
- **HTTP/3** : Une nouvelle version d'HTTP, qui est la troisième et prochaine version majeure du protocole de transfert hypertexte utilisé pour échanger des informations sur le World Wide Web. Celle-ci repose sur le protocole QUIC, développé par Google en 2012.

49

SSH (Secure SHell)

SSH permet de faire, en usage de base :

- ✎ Accès à distance sur la console en ligne commande (shell), ce qui permet, entre autres, d'effectuer la totalité des opérations courantes et/ou d'administration sur la machine distante.
- ✎ Déporter l'affichage graphique de la machine distante.
- ✎ Transferts de fichiers en ligne de commande.
- ✎ Montage ponctuel de répertoire distant, soit en ligne de commande, soit via **Nautilus**, sous Gnome par exemple Montage automatique de répertoires distants.
- ✎ **Remarque:** Nautilus est le gestionnaire de fichiers par défaut d l'environnements GNOME Shell, Il s'agit de l'équivalent de "l'Explorateur Windows" (sur Windows) ou de "Finder" (sur MacOS).

50

SSH

Les principaux composants de SSH

- ✂ sshd : le logiciel serveur, actif sur le port 22, qui ouvre une session à partir d'une connexion d'un client ssh.
- ✂ ssh : le logiciel client qui remplace rsh et rlogin.
- ✂ scp : le logiciel client qui remplace rcp.
- ✂ ssh-keygen : le logiciel qui permet de créer un couple de clés publique/privée

51

SSH

Installation du SSH coté Serveur

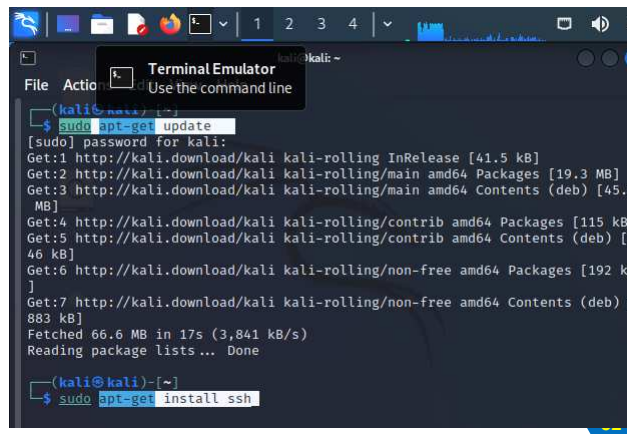
SSH (secure shell) service on Kali Linux

Instructions : Install SSH

From the terminal use apt-get command to install SSH packages:

sudo apt-get update

sudo apt-get install ssh



```
(kali@kali: ~)$ sudo apt-get update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.3 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [45.
MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [
46 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [192 k
]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb)
883 kB]
Fetched 66.6 MB in 17s (3,841 kB/s)
Reading package lists... Done

(kali@kali)~$ sudo apt-get install ssh
```


Installation du SSH coté Serveur

SSH (secure shell) service on Kali Linux

Instructions : Install SSH

```
(kali@kali) ~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 libwireshark15 libwiretap12 libwsutil13
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ssh
0 upgraded, 1 newly installed, 0 to remove and 1917 not upgraded.
Need to get 155 kB of archives.
After this operation, 167 kB of additional disk space will be used.
Get:1 http://kali.download/kali Kali-rolling/main amd64 ssh all 1:9.6p1-4 [155 kB]
Fetched 155 kB in 1s (218 kB/s)
Selecting previously unselected package ssh.
(Reading database ... 300319 files and directories currently installed.)
Preparing to unpack .../ssh_1%3a9.6p1-4_all.deb ...
Unpacking ssh (1:9.6p1-4) ...
Setting up ssh (1:9.6p1-4) ...

(kali@kali) ~$
```

53

Installation du SSH coté Serveur

SSH (secure shell) service on Kali Linux

Instructions : Enable and Start SSH

To make sure that secure shell starts after reboot use systemctl command to enable it: # sudo systemctl enable ssh

To start SSH for a current session execute: # sudo service ssh start

```

(kali@kali) ~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh

(kali@kali) ~$ sudo service ssh start

(kali@kali) ~$
```

54

Installation du SSH coté Serveur

SSH (secure shell) service on Kali Linux

Instructions :Allow SSH Root Access

- By default SSH would not allow you to SSH login as root user, thus the following error message will appear: **Permission denied, please try again.**
- edit or insert the following line within the `sudo nano /etc/ssh/sshd_config` SSH config file:

```

Ond nano 0.2 /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying

```

55

Installation du SSH coté Serveur

SSH (secure shell) service on Kali Linux

Instructions :Allow SSH Root Access

- insert the following line within the `/etc/ssh/sshd_config` SSH config file:
- FROM: `#PermitRootLogin prohibit-password` TO: `PermitRootLogin yes`

```

Ond nano 0.2 /etc/ssh/sshd_config *
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
File Name to Write: /etc/ssh/sshd_config

```

56

SSH

Configuration du SSH coté Serveur

Autoriser/Interdire des utilisateurs

Pour autoriser une liste de certains utilisateurs à se connecter. Modifier ou ajouter cette ligne dans le sshd_config : **AllowUsers user1 user2 user3**

Pour autoriser seulement certains membres de groupes à avoir accès via SSH en modifiant la ligne : **AllowGroups groupe1 groupe2**

Pour refuser la connexion que de certains utilisateurs. Modifier ou ajouter cette ligne dans le sshd_config : **DenyUsers user1 user2 user3**

Modifier le port d'écoute

Par défaut, le serveur openSSH écoute sur le port 22, pour change le port d'écoute du serveur openSSH modifier ou ajouter cette ligne dans le sshd_config : **Port numéro_du_port**

Limiter le nombre de tentative d'authentification

Pour limiter le nombre de de tentative d'authentification par exemple à 4 , modifier la ligne suivante : **MaxAuthTries 4**

57

SSH

Configuration du SSH coté Serveur

Modifier le port d'écoute

Par défaut, le serveur openSSH écoute sur le port 22, pour change le port d'écoute du serveur openSSH modifier ou ajouter cette ligne dans le sshd_config : **Port numéro_du_port**

Autoriser /interdire mot de passe vide :

Pour interdire la connexion au mot de passe vide modifier ou ajouter cette ligne dans le sshd_config : **PermitEmptyPasswords no**

Les valeurs possibles sont donc yes pour autoriser l'accès mot de passe vide, no (par défaut) pour le refuser.

Autoriser / interdire authentification par mot de passe

L'option suivante permet d'autoriser ou non des connexion avec un couple identifiant/mot de passe **PasswordAuthentication yes**

Les valeurs possibles sont donc yes pour autoriser l'authentification par mot de passe , no pour le refuser.

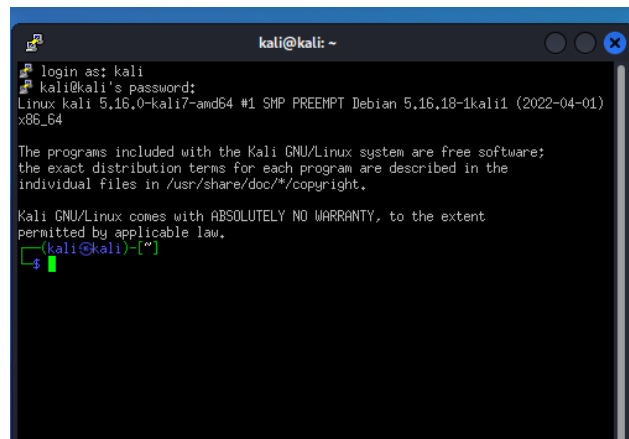
58

Installation du SSH coté Serveur

SSH (secure shell) service on Kali Linux

Instructions :restart ssh service

sudo service ssh restart

A terminal window titled 'kali@kali: ~' showing the Kali Linux login process. The user 'kali' logs in with their password. The terminal displays the Kali GNU/Linux version (5.16.0-kali7-amd64) and the kernel version (5.16.18-1kali1). It also shows the system's warranty information and the user's prompt '(kali@kali)-[~]' with a dollar sign '\$' indicating the root shell.

```
kali@kali: ~  
login as: kali  
kali@kali's password:  
Linux kali 5.16.0-kali7-amd64 #1 SMP PREEMPT Debian 5.16.18-1kali1 (2022-04-01)  
x86_64  
  
The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
[kali@kali]-[~]  
$
```

59

Se connecter par la commande SSH

Authentification par mot de passe

C'est la méthode la plus simple. Depuis la machine cliente,
taper : ssh login@nom du domaine ou adresse IP du serveur

Ensuite, entrez votre mot de passe... et vous verrez apparaître
le prompt, comme si vous vous étiez connecté en local sur la
machine.

En IPV6 ajouter l'option -6

ssh -6 <nom_utilisateur>@<adresse ipv6>

60

Se connecter par la commande SSH

Authentification par clef

Au lieu de s'authentifier par mot de passe, les utilisateurs peuvent s'authentifier grâce à la cryptographie asymétrique et son couple de clefs privée/publique, comme le fait le serveur SSH auprès du client SSH.

Générer la clefs

La création de la paire de clé se fait avec ssh-keygen.

Il existe 2 types de clés : RSA et DSA. Chacune pouvant être de longueur différente : 1024, 2048, 4096 bits (les clés inférieures à 2048 bits sont à proscrire... surtout les RSA). Pour créer une clé DSA de 2048 bits : `ssh-keygen -t dsa -b 2048`. Sans paramètres, les options par défaut sont type RSA en 2048 bits. `$ssh-keygen -t rsa -b 2048`

61

Se connecter par la commande SSH

Se connecter (solution avec ssh-agent)

- La commande est la même que pour une authentification par mot de passe mais sans demander le mot de passe
- Le serveur SSH est maintenant plus sécurisé, mais taper des passphrases à longueur de journée peut se révéler être très pénible surtout si on a choisi une « vraie » passphrase.
- L'agent SSH permet de taper la passphrase une seule fois et de la conserver en mémoire pendant tout son fonctionnement. Les communications SSH fonctionneront donc de façon transparente.
- Il faut lancer l'agent avec un shell (le plus simple étant de le lancer avec la variable \$SHELL qui contient le shell courant).
- Ensuite le programme ssh-add permet de charger les clé présentes dans ~/.ssh/. La passphrase est demandée, toutes les connexions nécessitant les clés chargées par l'agent seront transparentes.

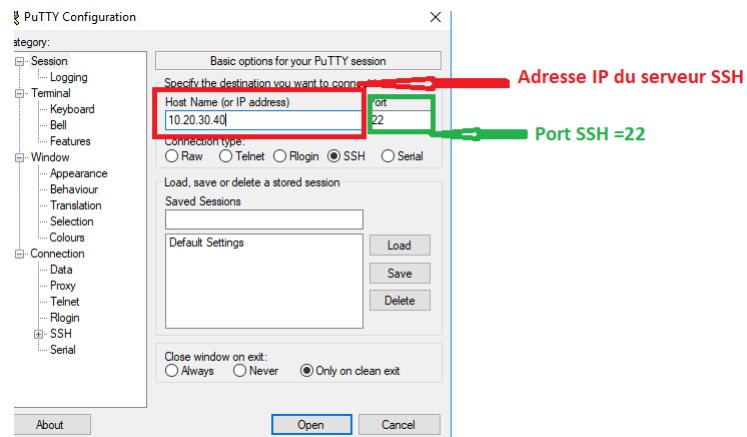
```
$ ssh-agent $SHELL
```

```
$ ssh-add
```

62

Installation du SSH coté Client

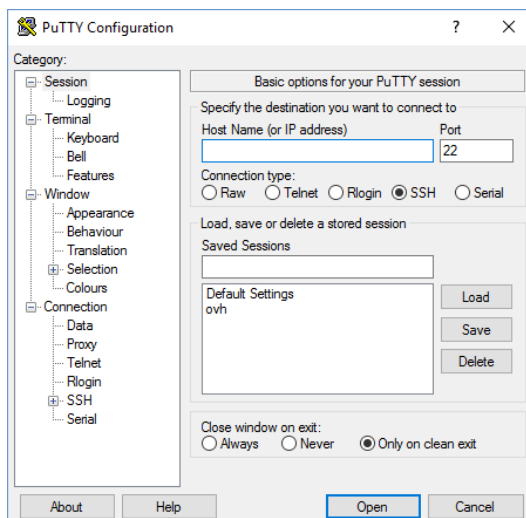
Se connecter par Putty (client Windows)



63

Installation du SSH coté Client

Se connecter par Putty (client Windows)



64