UNIVERSITÉ DE TECHNOLOGIE D'HAITI

Faculté des Sciences de Genie Civil et d'achitecture
Science informatique

Sujet: Création d'une nouvelle machine virtuelle dans virtualBox

Créer un nouveau dépôt : TD

Preparer par: Djenou LACOMBE

Soumis au prof: Ismael SAINT AMOUR

Dans le cadre du cours de cyber sécurité

16 Fevrier 2025

Étapes réalisée :

Créez un dossier cybersec avec trois sous-dossiers : scan , logs , script

Après avoir installé kali linux sur une machine virtuelle avec virtualBox ou VMware, je commence par mettre à jour le système pour assurer son bon fonctionnement.

Ensuite, je crée une structure de dossiers organisée avec un repertoire cybersec contenant trois (3) sous-dossiers: scan, logs, scripts.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ ls
notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ mv notes.txt ~/Bureau

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ ls
```

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scripts]
└─$ ls
notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scripts]
└─$ rm notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scripts]
└─$ ls

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scripts]
└─$
```

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scripts]
└─$ ls
notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scripts]
└─$
```

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ ls
notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ mv notes.txt ~/Bureau

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ ls
```

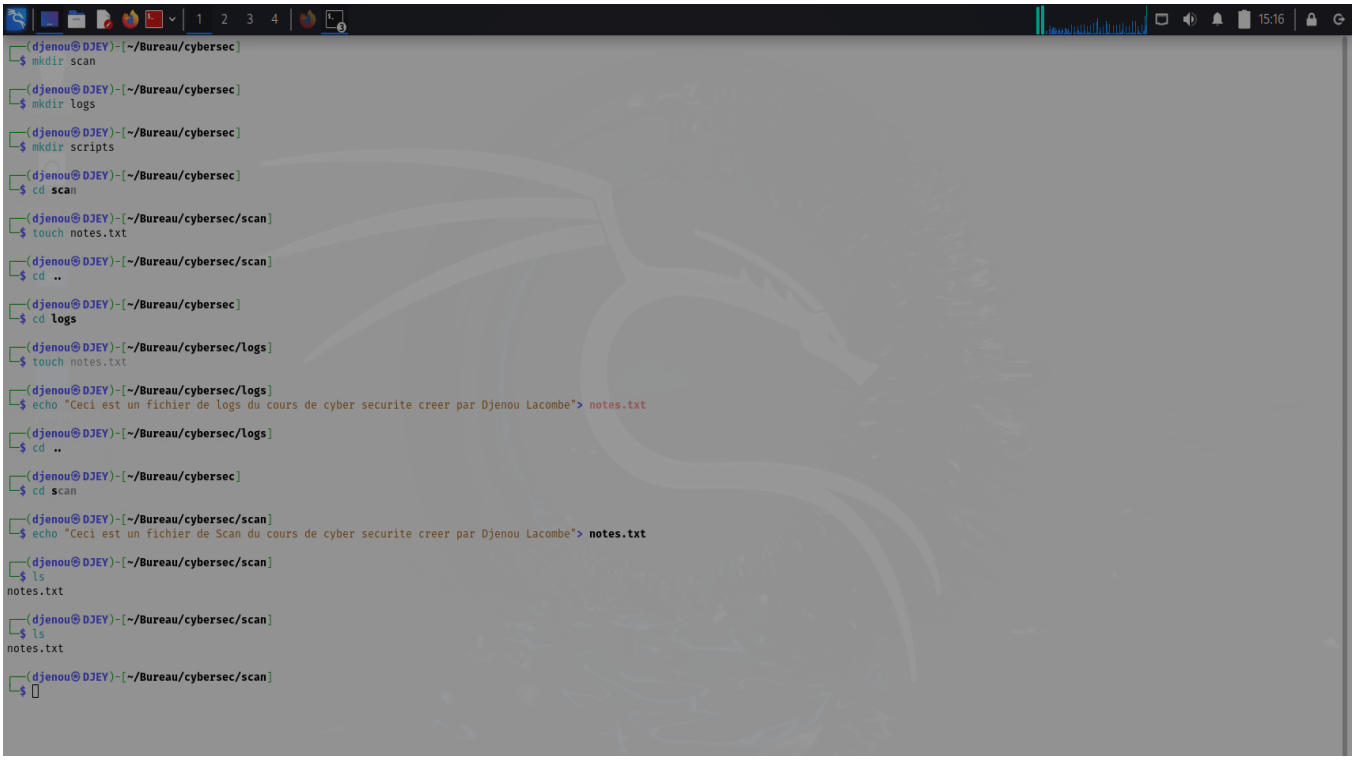
```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ touch secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ touch secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ chmod 400 secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ ls -l
total 4
-rw-rw-r-- 1 djenou djenou 68 14 fév 13:47 README.md
-r-------- 1 djenou djenou  0 15 fév 13:12 secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

1- J'ajoute des fichiers texte, y insère du contenu, puis je les manipule en les copiant, déplacant et supprimant tout en verifiant chaque étape.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ echo "Ceci est un fichier de Scan du cours de cyber securite creer par Djenou Lacombe"> notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ ls
notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ ls
notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ cat notes.txt
Ceci est un fichier de Scan du cours de cyber securite creer par Djenou Lacombe

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$
```

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec/logs]
└─$ cd ..

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ cd scan

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$ echo "Ceci est un fichier de Scan du cours de cyber securite creer par Djenou Lacombe"> notes.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
└─$
```



```
(djenou㉿DJEY)-[~/Bureau/cybersec]
$ mkdir scan

(djenou㉿DJEY)-[~/Bureau/cybersec]
$ mkdir logs

(djenou㉿DJEY)-[~/Bureau/cybersec]
$ mkdir scripts

(djenou㉿DJEY)-[~/Bureau/cybersec]
$ cd scan

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ touch notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ cd ..

(djenou㉿DJEY)-[~/Bureau/cybersec]
$ cd logs

(djenou㉿DJEY)-[~/Bureau/cybersec/logs]
$ touch notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/logs]
$ echo "Ceci est un fichier de logs du cours de cyber securite creer par Djenou Lacombe"> notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/logs]
$ cd ..

(djenou㉿DJEY)-[~/Bureau/cybersec]
$ cd scan

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ echo "Ceci est un fichier de Scan du cours de cyber securite creer par Djenou Lacombe"> notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ ls
notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ ls
notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$
```


```
(djenou㉿DJEY)-[~/Bureau/cybersec]
$ cd scan

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ touch notes.txt

(djenou㉿DJEY)-[~/Bureau/cybersec/scan]
$ cd ..

(djenou㉿DJEY)-[~/Bureau/cybersec]
$ cd logs

(djenou㉿DJEY)-[~/Bureau/cybersec/logs]
$ touch notes.txt
```

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ mkdir scan

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ mkdir logs

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ mkdir scripts

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ touch notes.txt
```

5. Scanner un réseau : ifconfig ou ip a : Affiche les informations réseau. Utilisez nmap pour scanner votre réseau local et identifier les appareils connectés. grep "motif" fichier.txt cat fichier.txt echo "Bonjour" > fichier.txt ps aux kill PID # Remplacez PID par l'ID du processus

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe39:7bb2  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:39:7b:b2  txqueuelen 1000  (Ethernet)
        RX packets 1974  bytes 1212984 (1.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1763  bytes 262576 (256.4 KiB)
        TX errors 0  dropped 9 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Boucle locale)
        RX packets 378  bytes 28451 (27.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 378  bytes 28451 (27.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

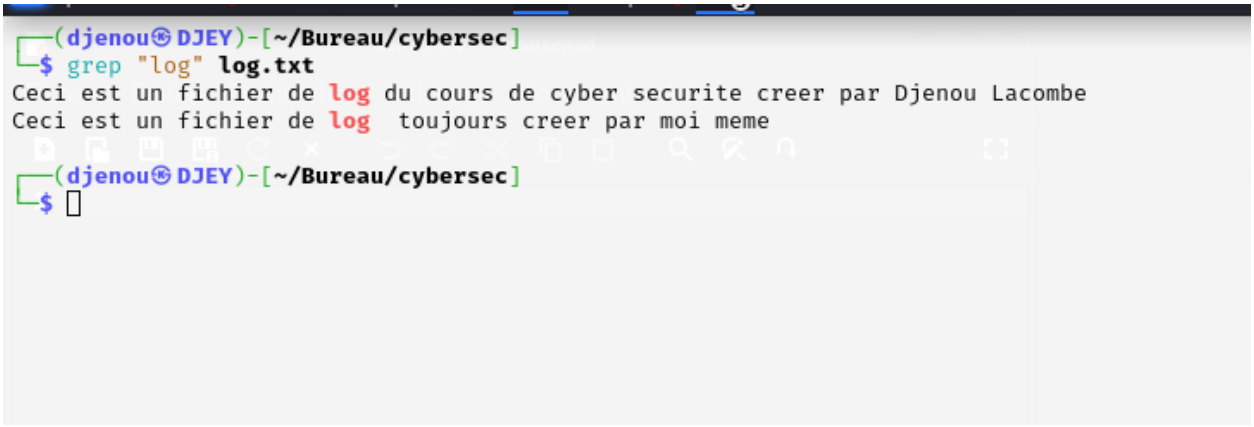
Utilisez nmap pour scanner votre réseau local et identifier les appareils connectés.

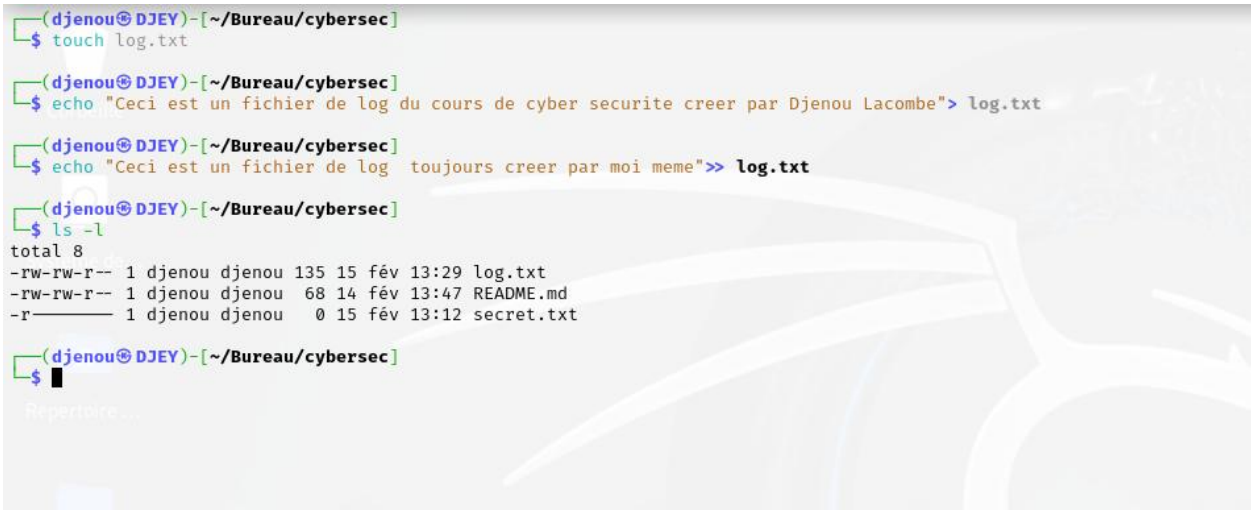
```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ nmap 10.0.2.15/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-15 13:08 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.0.2.2
Host is up (0.011s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
2179/tcp open  vmrdp
3389/tcp open  ms-wbt-server
5560/tcp open  isqlplus
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)

Nmap scan report for 10.0.2.3
Host is up (0.0084s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
2179/tcp open  vmrdp
3389/tcp open  ms-wbt-server
5560/tcp open  isqlplus
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)

Nmap scan report for 10.0.2.4
Host is up (0.0092s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
1151/tcp open  unizensus
2179/tcp open  vmrdp
3389/tcp open  ms-wbt-server
5560/tcp open  isqlplus
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)

Nmap scan report for 10.0.2.15
Host is up (0.0000060s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
```

6. Manipuler les permissions : Créez un fichier secret.txt et changez ses permissions pour qu'il ne soit accessible qu'en lecture par le propriétaire.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ touch secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ chmod 400 secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ ls -l
total 4
-rw-rw-r-- 1 djenou djenou 68 14 fév 13:47 README.md
-r-------- 1 djenou djenou  0 15 fév 13:12 secret.txt

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

7. Utiliser grep : Créez un fichier log.txt avec des lignes de texte, puis utilisez grep pour rechercher un mot spécifique.





j'explore les fonctionnalités de réseau en affichant les informations de connexion et en utilisant nmap pour la scanner mon reseau local et identifier les appareils conectés.

La commande **`df -h`** permet d'afficher l'espace disque disponible et utilisé sur les différentes partitions du système, avec une présentation en **format lisible par l'humain** (h pour "human-readable"

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ df -h
Sys. de fichiers Taille Utilisé Dispo Uti% Monté sur
udev               3,9G       0  3,9G   0% /dev
tmpfs              795M    972K  794M   1% /run
/dev/sda1           38G     16G   20G  44% /
tmpfs              3,9G    4,0K  3,9G   1% /dev/shm
tmpfs              5,0M       0  5,0M   0% /run/lock
tmpfs              1,0M       0  1,0M   0% /run/credentials/systemd-journald.service
tmpfs              3,9G    212K  3,9G   1% /tmp
tmpfs              1,0M       0  1,0M   0% /run/credentials/getty@tty1.service
tmpfs              795M    124K  795M   1% /run/user/1000

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

`du -sh` : Afficher la taille d'un dossier ou fichier

La commande **`du -sh`** permet de connaître **l'espace disque occupé** par un dossier ou un fichier, avec une taille **lisible par l'humain**.

```
tmpfs              795M    124K  795M   1% /run/user/1000

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ du -sh
200K    .

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

`free -h` : Afficher l'utilisation de la mémoire (RAM)

La commande **`free -h`** permet de voir la mémoire **RAM** et l'espace **swap** utilisés et disponibles, en format **lisible par l'humain** (**Go, Mo, Ko**).

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ free -h
               total        utilisé      libre     partagé tamp/cache   disponible
Mem:           7,8Gi        1,9Gi       5,4Gi       104Mi       899Mi       5,9Gi
Échange:       2,1Gi          0B       2,1Gi

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
```

`ps aux` : Afficher tous les processus en cours

La commande **`ps aux`** permet de **lister tous les processus** qui s'exécutent sur le système, avec des détails comme l'utilisateur, l'utilisation de la mémoire et du CPU, et la commande lancée.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.1  23140 14164 ?        Ss   09:16   0:03 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    09:16   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    09:16   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-rcu_gp]
root           5  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-sync_wq]
root           6  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-slub_flushwq]
root           7  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-netns]
root          11  0.0  0.0      0     0 ?        I    09:16   0:00 [kworker/u8:0-ipv6_addrconf]
root          12  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-mm_percpu_wq]
root          13  0.0  0.0      0     0 ?        I    09:16   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    09:16   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    09:16   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    09:16   0:04 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    09:16   0:16 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    09:16   0:00 [rcu_exp_par_gp_kthread_worker/0]
root          19  0.0  0.0      0     0 ?        S    09:16   0:00 [rcu_exp_gp_kthread_worker]
root          20  0.0  0.0      0     0 ?        S    09:16   0:01 [migration/0]
root          21  0.0  0.0      0     0 ?        S    09:16   0:00 [idle_inject/0]
root          22  0.0  0.0      0     0 ?        S    09:16   0:00 [cpuhp/0]
root          23  0.0  0.0      0     0 ?        S    09:16   0:00 [cpuhp/1]
root          24  0.0  0.0      0     0 ?        S    09:16   0:00 [idle_inject/1]
root          25  0.0  0.0      0     0 ?        S    09:16   0:01 [migration/1]
root          26  0.0  0.0      0     0 ?        S    09:16   0:08 [ksoftirqd/1]
root          33  0.0  0.0      0     0 ?        S    09:16   0:00 [kdevtmpfs]
root          34  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-inet_frag_wq]
root          36  0.0  0.0      0     0 ?        S    09:16   0:00 [kauditd]
root          37  0.0  0.0      0     0 ?        S    09:16   0:00 [khungtaskd]
root          38  0.0  0.0      0     0 ?        S    09:16   0:00 [oom_reaper]
root          40  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-writeback]
root          41  0.0  0.0      0     0 ?        S    09:16   0:02 [kcompactd0]
root          42  0.0  0.0      0     0 ?        SN   09:16   0:00 [ksmd]
root          43  0.0  0.0      0     0 ?        SN   09:16   0:03 [khugepaged]
root          44  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-kintegrityd]
root          45  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-kblockd]
root          46  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-blkcg_punt_bio]
root          47  0.0  0.0      0     0 ?        S    09:16   0:00 [irq/9-acpi]
root          49  0.0  0.0      0     0 ?        I<   09:16   0:00 [kworker/R-tpm_dev_wq]
```

`lspci` : Afficher les périphériques PCI du système

La commande **`lspci`** permet de **lister tous les périphériques PCI** (Peripheral Component Interconnect) connectés à l'ordinateur, comme la carte graphique, la carte réseau, le contrôleur USB,

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
```

`sudo apt install traceroute` : Installer l'outil `traceroute` sous Linux

La commande **`sudo apt install traceroute`** permet d'installer **l'outil `traceroute`** sur un système basé sur **Debian** (comme **Kali Linux, Ubuntu, Debian**).

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ sudo apt install traceroute
traceroute est déjà la version la plus récente (1:2.1.6-1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  libbfio1      libconfig++9v5       libfmt9         libglvnd-core-dev          libgtksourceviewmm-3.0-0v5  libpaper1         libtagc0                  openjdk-23-jre
  libc++1-19    libconfig9           libgl1-mesa-dev libglvnd-dev               libhdf5-hl-100t64           libsuperlu6       libunwind-19              openjdk-23-jre-headless
  libc++abi1-19 libdirectfb-1.7-7t64 libgles-dev     libgtksourceview-3.0-1     libjxl0.9                   libtag1v5         libwebrtc-audio-processing1  python3-appdirs
  libcapstone4  libegl-dev           libgles1        libgtksourceview-3.0-common libmbedcrypto7t64          libtag1v5-vanilla libx265-209
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
```

`traceroute google.com` : Analyse du chemin emprunté par les paquets réseau

La commande **`traceroute google.com`** permet de déterminer **l'itinéraire suivi par les paquets** envoyés depuis l'ordinateur local jusqu'au serveur **Google**. Elle identifie **chaque routeur intermédiaire** traversé, ainsi que **le temps nécessaire** pour atteindre chaque point du réseau.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ traceroute google.comclear
google.comclear: Nom ou service inconnu
Cannot handle "host" cmdline arg `google.comclear' on position 1 (argc 1)

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

`netstat -tuln` : Affichage des ports ouverts et des services en écoute

La commande **`netstat -tuln`** permet d'afficher **les ports réseau en écoute** sur un système ainsi que les services associés. Elle est particulièrement utile pour **analyser la sécurité d'un serveur** et vérifier les connexions entrantes potentielles.

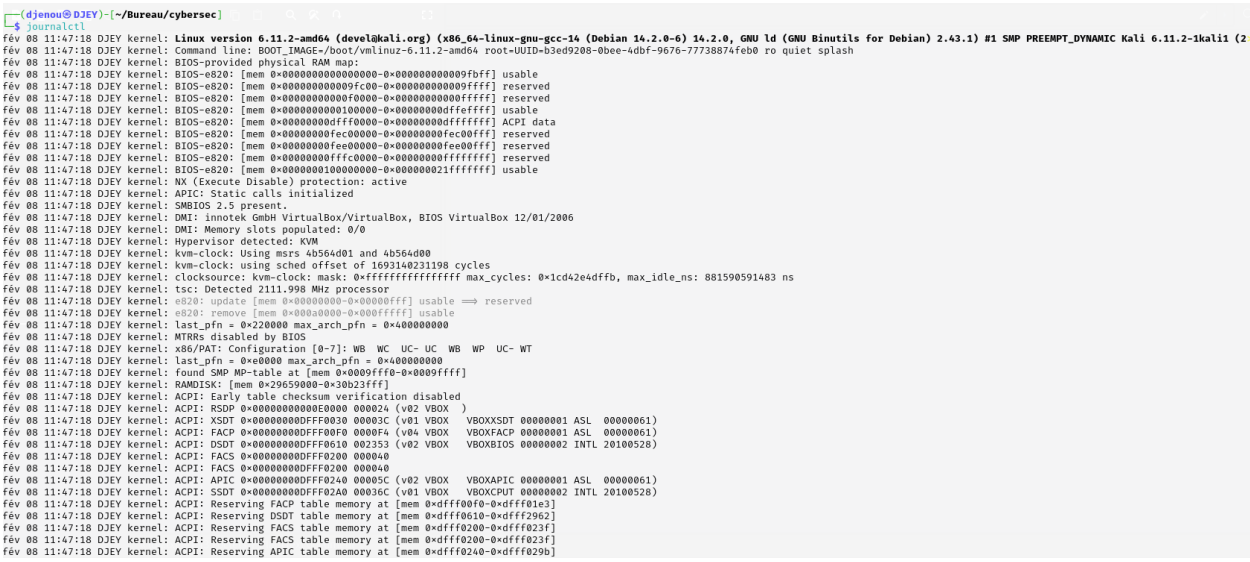
```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ netstat -tuln
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale           Adresse distante         Etat
udp        0      0 10.0.2.15:3702          0.0.0.0:*
udp        0      0 239.255.255.250:3702    0.0.0.0:*
udp        0      0 0.0.0.0:42666           0.0.0.0:*
udp6       0      0 fe80::a00:27ff:fe3:3702 :::*
udp6       0      0 ff02::c:3702            :::*
udp6       0      0 :::38853                :::*

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

`ss -tuln` : Affichage des ports ouverts et des services en écoute

La commande **`ss -tuln`** est une alternative moderne à `netstat -tuln`. Elle permet d'afficher **les ports en écoute** sur un système ainsi que les services qui les utilisent. Elle est **plus rapide et plus efficace** que `netstat`, car elle interroge directement les structures de données du noyau Linux.

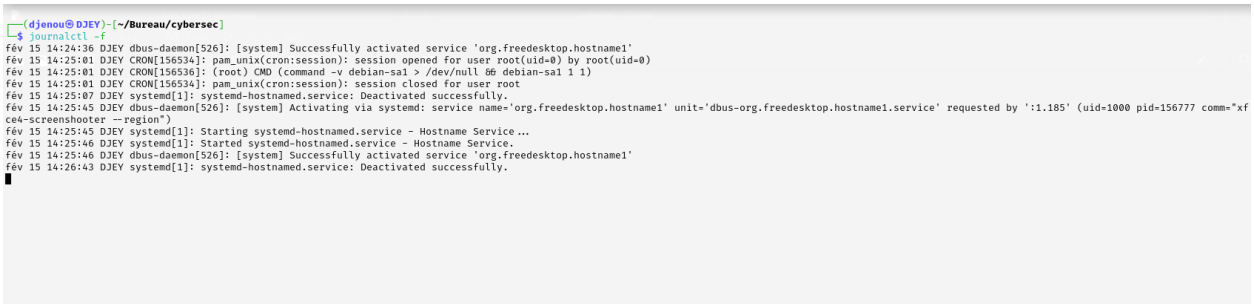
`journalctl` : Afficher les journaux système sous Linux

La commande **`journalctl`** permet d'afficher les **journaux du système** sur les distributions Linux qui utilisent **`systemd`** comme système d'init. Elle permet de consulter, filtrer et analyser les **logs système**, les événements du noyau, les messages des services et des applications.



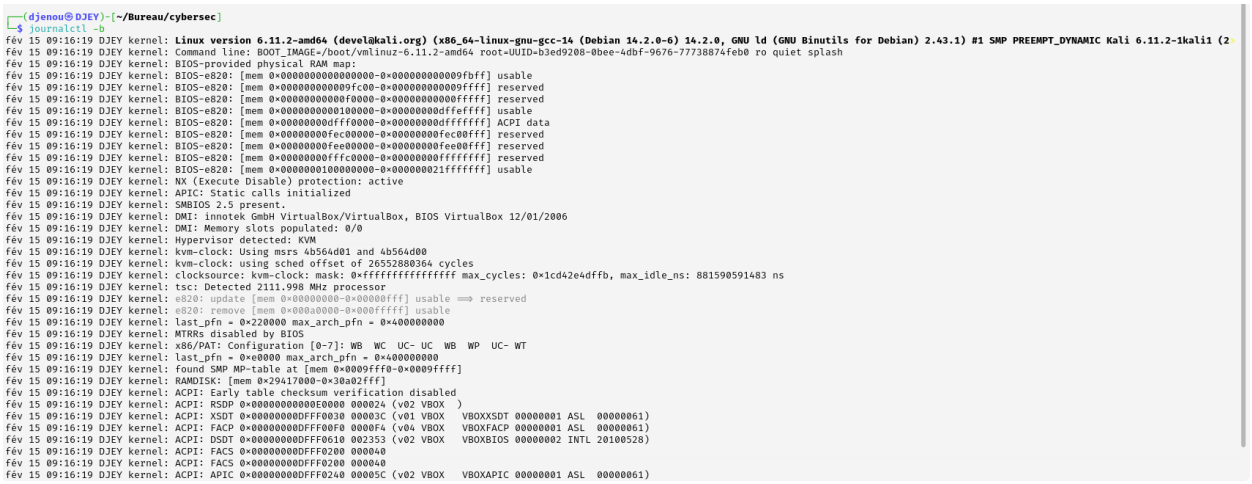
`journalctl -f` : Suivi en temps réel des journaux système

La commande **`journalctl -f`** permet de suivre les journaux système en **temps réel**. Cela équivaut à utiliser la commande **`tail -f`** sur un fichier de logs, mais elle s'applique directement aux logs collectés par **`systemd`**. Cette commande est particulièrement utile pour **observer l'activité** du système pendant qu'il se déroule.



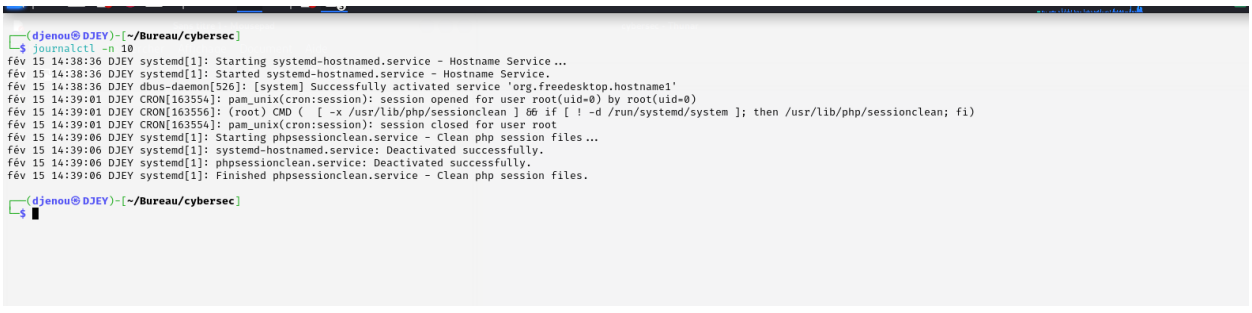
`journalctl -b` : Afficher les journaux du démarrage actuel

La commande **`journalctl -b`** permet d'afficher les journaux du système **depuis le dernier démarrage**. Cela inclut tous les messages générés par **`systemd`**, les services, le noyau et d'autres applications depuis le dernier démarrage de la machine.



`journalctl -n 10` : Afficher les 10 derniers journaux

La commande **`journalctl -n 10`** permet d'afficher les **10 derniers journaux** enregistrés par **`systemd`**. Elle est utile pour obtenir rapidement un aperçu des dernières entrées dans les journaux système sans avoir à faire défiler tout l'historique des logs.



`date` : Afficher ou définir la date et l'heure du système

La commande `date` permet d'afficher la **date et l'heure actuelles** du système. Elle peut également être utilisée pour **modifier la date et l'heure** du système si l'utilisateur dispose des privilèges nécessaires.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ date
sam 15 fév 2025 14:40:54 EST

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

`timedatectl` : Gérer la date, l'heure et le fuseau horaire du système

La commande `timedatectl` est utilisée pour afficher et modifier la **date, l'heure**, et le **fuseau horaire** sur les systèmes Linux qui utilisent `systemd`. Elle permet de configurer le temps système de manière centralisée

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ timedatectl
               Local time: sam 2025-02-15 15:20:00 EST
           Universal time: sam 2025-02-15 20:20:00 UTC
                 RTC time: sam 2025-02-15 20:19:59
                Time zone: America/Port-au-Prince (EST, -0500)
System clock synchronized: no
              NTP service: inactive
          RTC in local TZ: no

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

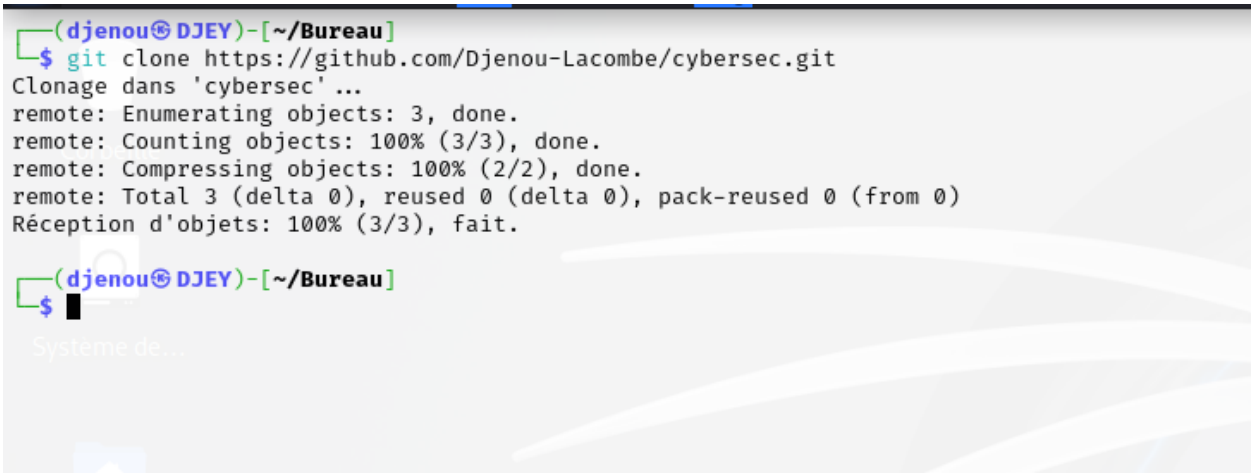
`hostnamectl` : Gérer le nom d'hôte du système

La commande `hostnamectl` est utilisée pour afficher et modifier le **nom d'hôte** (hostname) d'un système Linux. Elle permet également de gérer certains paramètres liés au système, comme le **système d'exploitation**, le **type de machine**, et d'autres informations liées au réseau.

```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ hostnamectl
 Static hostname: DJEY
       Icon name: computer-vm
         Chassis: vm 
      Machine ID: 501665a5087d4773bbf5e421718fd357
         Boot ID: 1742768498a345f7957568820deecbb9
  Virtualization: oracle
Operating System: Kali GNU/Linux Rolling
          Kernel: Linux 6.11.2-amd64
    Architecture: x86-64
 Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
Firmware Version: VirtualBox
   Firmware Date: Fri 2006-12-01
    Firmware Age: 18y 2month 2w 2d

┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$
```

Pour changer le nom d'hôte, vous pouvez utiliser la commande suivante sudo hostnamectl set-hostname [nouveau_nom]


```
┌──(djenou㉿DJEY)-[~/Bureau/cybersec]
└─$ hostnamectl
 Static hostname: DJEY
       Icon name: computer-vm
         Chassis: vm 
      Machine ID: 501665a5087d4773bbf5e421718fd357
         Boot ID: 1742768498a345f7957568820deecbb9
  Virtualization: oracle
Operating System: Kali GNU/Linux Rolling
          Kernel: Linux 6.11.2-amd64
    Architecture: x86-64
 Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
Firmware Version: VirtualBox
   Firmware Date: Fri 2006-12-01
    Firmware Age: 18y 2month 2w 2d
```



Ce travail me permet de me famliariser avec les comandes de gestion des fichier sous linux, ainsi qu'avec les outils de base en cybersécurité, essentiel pour les comprend l'arninistration systéme et annalyse réseau.

Au terme de cette série de tâches, plusieurs étapes clés ont été réalisées pour configurer et gérer un environnement Kali Linux. Voici les principaux points à retenir :

1. **Installation et configuration initiale de Kali Linux** :
    - Kali Linux a été installé avec succès, et le système a été mis à jour pour garantir une version à jour et fonctionnelle.
2. **Gestion des fichiers et des répertoires** :
    - Une structure de répertoires a été créée avec des sous-dossiers et des fichiers texte. Cela a permis de tester les commandes de gestion des fichiers, telles que la copie, le déplacement et la suppression.
    - L'organisation des fichiers a été suivie, et des vérifications ont été effectuées pour confirmer que les actions ont été réalisées correctement.
3. **Exploration des commandes réseau** :
    - Des commandes comme **`nmap`**, **`ifconfig`**, et **`traceroute`** ont été utilisées pour explorer le réseau, détecter les appareils connectés, et diagnostiquer la connectivité réseau.
    - Cela a permis de mieux comprendre la gestion réseau sous Linux, ce qui est essentiel dans un environnement de cybersécurité.
4. **Gestion du temps et du fuseau horaire** :
    - Des outils comme **`timedatectl`** et **`hostnamectl`** ont été utilisés pour configurer l'heure système et le nom d'hôte de la machine. Cela est crucial pour assurer la synchronisation correcte des systèmes et une gestion optimale des ressources.
5. **Apprentissage pratique et développement des compétences** :
    - La manipulation des commandes en ligne de commande sous Linux a renforcé la compréhension de la gestion des systèmes, notamment dans un environnement sécurisé et contrôlé comme Kali Linux.
    - L'exécution de ces tâches a permis de mieux appréhender les outils de gestion du système et des réseaux, essentiels pour toute personne travaillant dans le domaine de la cybersécurité.

En somme, cette tâche a permis non seulement de se familiariser avec l'environnement Kali Linux, mais aussi d'acquérir des compétences pratiques dans la gestion des fichiers, la configuration du système et l'analyse réseau. Ces compétences sont fondamentales pour tout travail en cybersécurité et en administration système.