Alumno: Rodrigo Vila

Modulo 3 Unidad 1

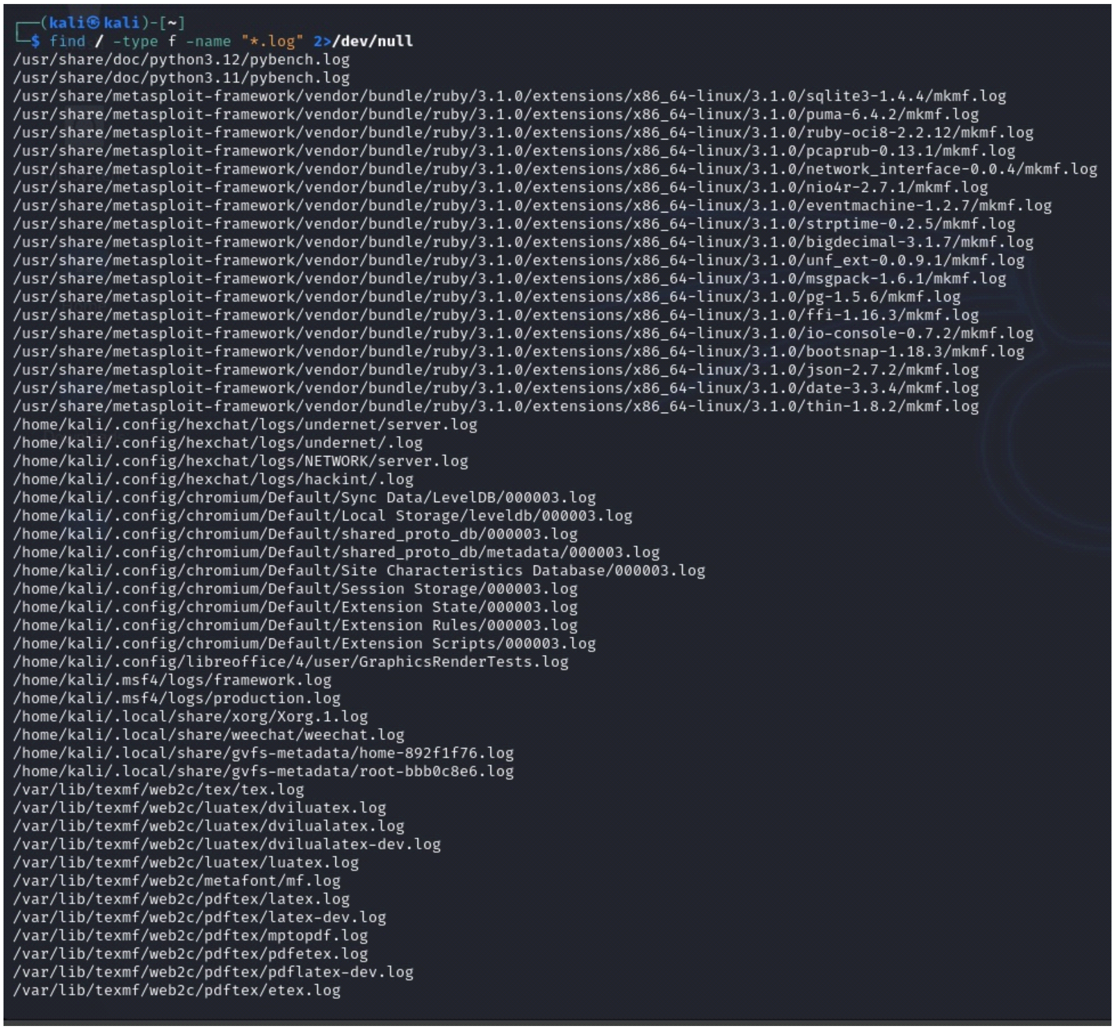
Adicional: Comprensión de los Shells Linux

Ejercicios:

- Encontrar utilizando el comando find todos los archivos que terminen con la extensión .log, guardar su nombre dentro de un archivo llamado logs_del_sistema dentro de nuestro home directory. El listado de archivos debe estar ordenado alfabéticamente:

find / -type f -name ".log" 2>/dev/null*

- /: Especifica el directorio raíz desde donde se iniciará la búsqueda.
- -type f: Limita la búsqueda a archivos regulares (no directorios ni enlaces simbólicos).
- -name "*.log": Especifica el patrón de búsqueda para archivos que terminen con la extensión .log.
- 2>/dev/null: Redirige los mensajes de error a /dev/null para que no se muestren en la salida.

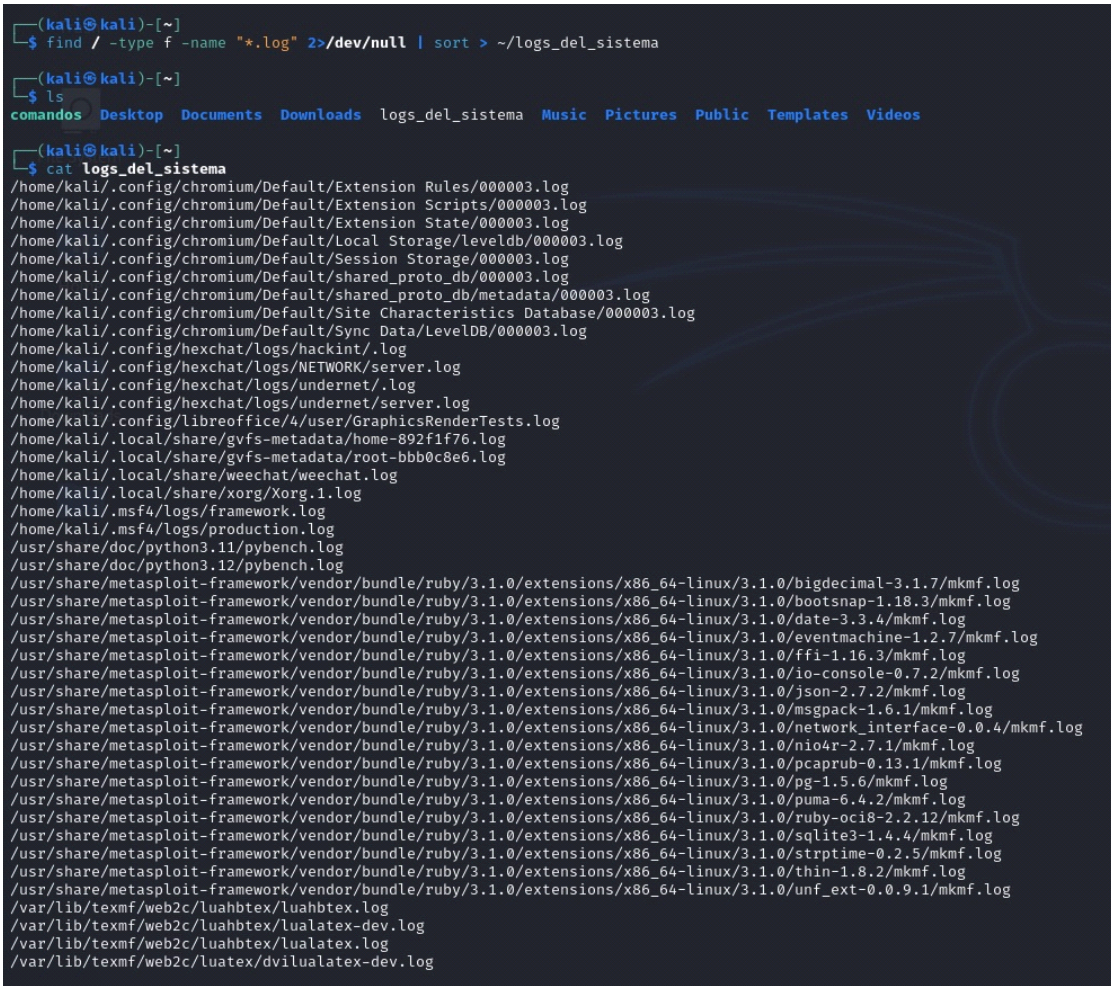


```
┌──(kali㉿kali)-[~]
└─$ find / -type f -name "*.log" 2>/dev/null
/usr/share/doc/python3.12/pybench.log
/usr/share/doc/python3.11/pybench.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/sqlite3-1.4.4/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/puma-6.4.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/ruby-oci8-2.2.12/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/pcaprub-0.13.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/network_interface-0.0.4/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/nio4r-2.7.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/eventmachine-1.2.7/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/strptime-0.2.5/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/bigdecimal-3.1.7/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/unf_ext-0.0.9.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/msgpack-1.6.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/pg-1.5.6/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/ffi-1.16.3/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/io-console-0.7.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/bootsnap-1.18.3/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/json-2.7.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/date-3.3.4/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/thin-1.8.2/mkmf.log
/home/kali/.config/hexchat/logs/undernet/server.log
/home/kali/.config/hexchat/logs/undernet/.log
/home/kali/.config/hexchat/logs/NETWORK/server.log
/home/kali/.config/hexchat/logs/hackint/.log
/home/kali/.config/chromium/Default/Sync Data/LevelDB/000003.log
/home/kali/.config/chromium/Default/Local Storage/leveldb/000003.log
/home/kali/.config/chromium/Default/shared_proto_db/000003.log
/home/kali/.config/chromium/Default/shared_proto_db/metadata/000003.log
/home/kali/.config/chromium/Default/Site Characteristics Database/000003.log
/home/kali/.config/chromium/Default/Session Storage/000003.log
/home/kali/.config/chromium/Default/Extension State/000003.log
/home/kali/.config/chromium/Default/Extension Rules/000003.log
/home/kali/.config/chromium/Default/Extension Scripts/000003.log
/home/kali/.config/libreoffice/4/user/GraphicsRenderTests.log
/home/kali/.msf4/logs/framework.log
/home/kali/.msf4/logs/production.log
/home/kali/.local/share/xorg/Xorg.1.log
/home/kali/.local/share/weechat/weechat.log
/home/kali/.local/share/gvfs-metadata/home-892f1f76.log
/home/kali/.local/share/gvfs-metadata/root-bbb0c8e6.log
/var/lib/texmf/web2c/tex/tex.log
/var/lib/texmf/web2c/luatex/dviluatex.log
/var/lib/texmf/web2c/luatex/dvilualatex.log
/var/lib/texmf/web2c/luatex/dvilualatex-dev.log
/var/lib/texmf/web2c/luatex/luatex.log
/var/lib/texmf/web2c/metafont/mf.log
/var/lib/texmf/web2c/pdftex/latex.log
/var/lib/texmf/web2c/pdftex/latex-dev.log
/var/lib/texmf/web2c/pdftex/mptopdf.log
/var/lib/texmf/web2c/pdftex/pdfetex.log
/var/lib/texmf/web2c/pdftex/pdflatex-dev.log
/var/lib/texmf/web2c/pdftex/etex.log
```

Una vez que estoy seguro de que el comando muestra la lista correcta de archivos .log, redirijo la salida hacia el archivo logs_del_sistema en el /home:

find / -type f -name ".log" 2>/dev/null | sort > ~/logs_del_sistema*

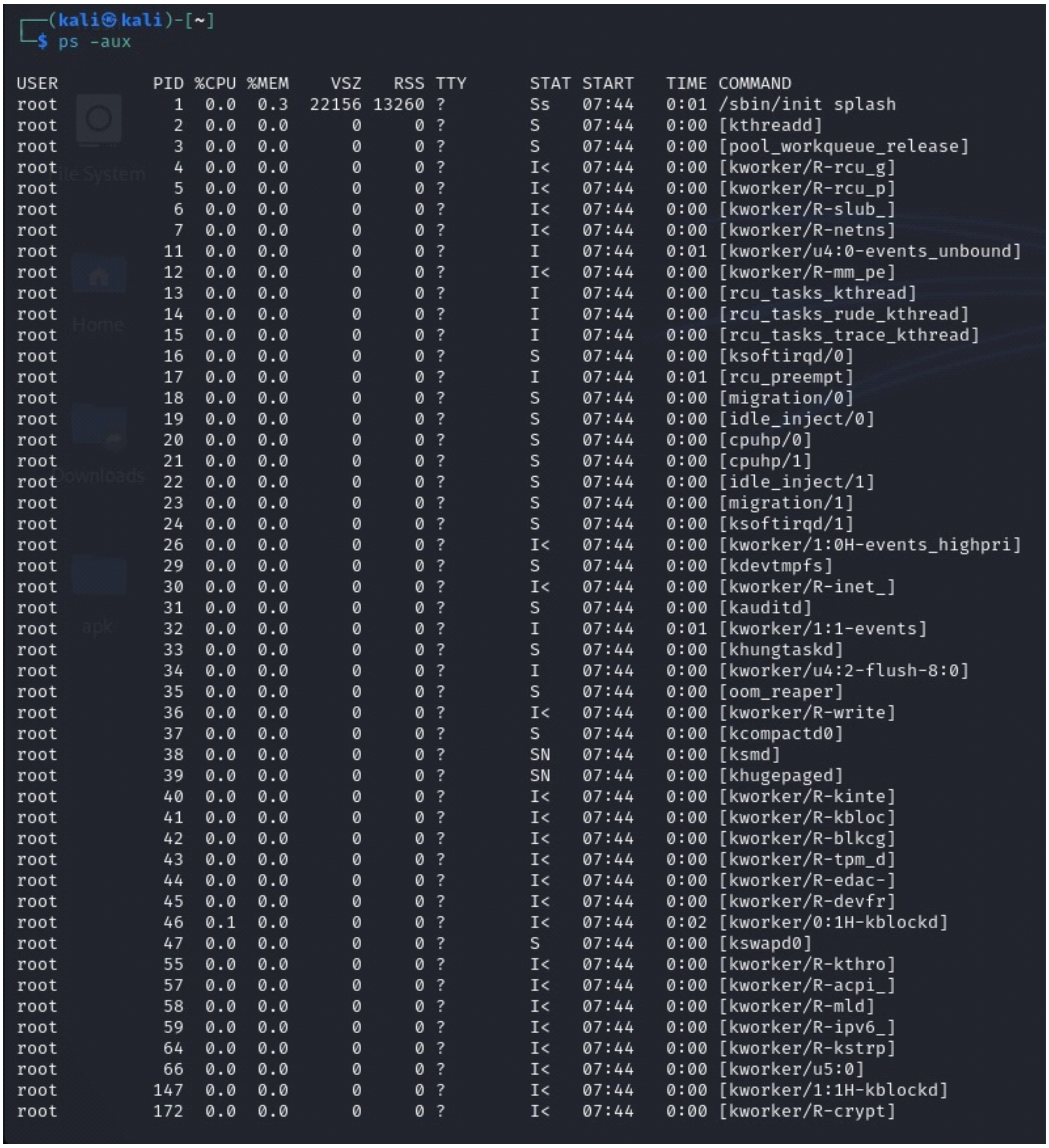
- | sort: Este símbolo de tubería (|) envía la salida del comando find al comando sort, que ordena alfabéticamente la lista de archivos.
- > ~/logs_del_sistema: Redirige la salida ordenada hacia el archivo llamado logs_del_sistema en el directorio home (~/ representa el directorio home).



```
┌──(kali㉿kali)-[~]
└─$ find / -type f -name "*.log" 2>/dev/null | sort > ~/logs_del_sistema

┌──(kali㉿kali)-[~]
└─$ ls
comandos  Desktop  Documents  Downloads  logs_del_sistema  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ cat logs_del_sistema
/home/kali/.config/chromium/Default/Extension Rules/000003.log
/home/kali/.config/chromium/Default/Extension Scripts/000003.log
/home/kali/.config/chromium/Default/Extension State/000003.log
/home/kali/.config/chromium/Default/Local Storage/leveldb/000003.log
/home/kali/.config/chromium/Default/Session Storage/000003.log
/home/kali/.config/chromium/Default/shared_proto_db/000003.log
/home/kali/.config/chromium/Default/shared_proto_db/metadata/000003.log
/home/kali/.config/chromium/Default/Site Characteristics Database/000003.log
/home/kali/.config/chromium/Default/Sync Data/LevelDB/000003.log
/home/kali/.config/hexchat/logs/hackint/.log
/home/kali/.config/hexchat/logs/NETWORK/server.log
/home/kali/.config/hexchat/logs/undernet/.log
/home/kali/.config/hexchat/logs/undernet/server.log
/home/kali/.config/libreoffice/4/user/GraphicsRenderTests.log
/home/kali/.local/share/gvfs-metadata/home-892f1f76.log
/home/kali/.local/share/gvfs-metadata/root-bbb0c8e6.log
/home/kali/.local/share/weechat/weechat.log
/home/kali/.local/share/xorg/Xorg.1.log
/home/kali/.msf4/logs/framework.log
/home/kali/.msf4/logs/production.log
/usr/share/doc/python3.11/pybench.log
/usr/share/doc/python3.12/pybench.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/bigdecimal-3.1.7/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/bootsnap-1.18.3/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/date-3.3.4/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/eventmachine-1.2.7/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/ffi-1.16.3/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/io-console-0.7.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/json-2.7.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/msgpack-1.6.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/network_interface-0.0.4/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/nio4r-2.7.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/pcaprub-0.13.1/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/pg-1.5.6/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/puma-6.4.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/ruby-oci8-2.2.12/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/sqlite3-1.4.4/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/strptime-0.2.5/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/thin-1.8.2/mkmf.log
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/extensions/x86_64-linux/3.1.0/unf_ext-0.0.9.1/mkmf.log
/var/lib/texmf/web2c/luahbtex/luahbtex.log
/var/lib/texmf/web2c/luahbtex/lualatex-dev.log
/var/lib/texmf/web2c/luahbtex/lualatex.log
/var/lib/texmf/web2c/luatex/dvilualatex-dev.log
```

- Imprimir dentro de un archivo llamado procesos_root en nuestro home directory el STDOUT del comando “ps –aux”, pero solamente la última columna. Pista: investigar el comando cut y sus flags:

Ejecuto el comando ps -aux para listar todos los procesos en el sistema:

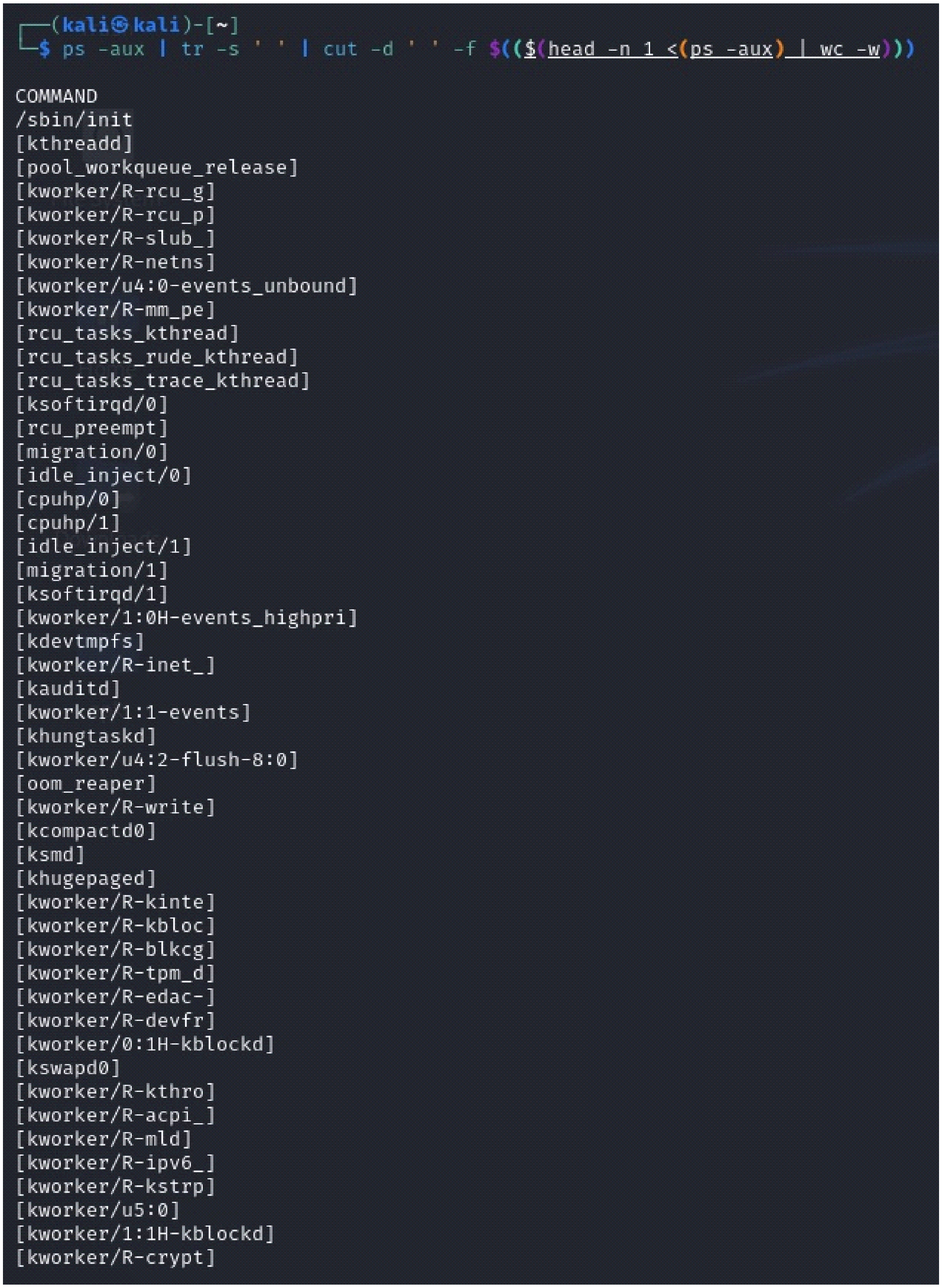


```
┌──(kali㉿kali)-[~]
└─$ ps -aux

USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.3  22156 13260 ?        Ss   07:44   0:01 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    07:44   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    07:44   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-netns]
root          11  0.0  0.0      0     0 ?        I    07:44   0:01 [kworker/u4:0-events_unbound]
root          12  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-mm_pe]
root          13  0.0  0.0      0     0 ?        I    07:44   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    07:44   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    07:44   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    07:44   0:00 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    07:44   0:01 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    07:44   0:00 [migration/0]
root          19  0.0  0.0      0     0 ?        S    07:44   0:00 [idle_inject/0]
root          20  0.0  0.0      0     0 ?        S    07:44   0:00 [cpuhp/0]
root          21  0.0  0.0      0     0 ?        S    07:44   0:00 [cpuhp/1]
root          22  0.0  0.0      0     0 ?        S    07:44   0:00 [idle_inject/1]
root          23  0.0  0.0      0     0 ?        S    07:44   0:00 [migration/1]
root          24  0.0  0.0      0     0 ?        S    07:44   0:00 [ksoftirqd/1]
root          26  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/1:0H-events_highpri]
root          29  0.0  0.0      0     0 ?        S    07:44   0:00 [kdevtmpfs]
root          30  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-inet_]
root          31  0.0  0.0      0     0 ?        S    07:44   0:00 [kauditd]
root          32  0.0  0.0      0     0 ?        I    07:44   0:01 [kworker/1:1-events]
root          33  0.0  0.0      0     0 ?        S    07:44   0:00 [khungtaskd]
root          34  0.0  0.0      0     0 ?        I    07:44   0:00 [kworker/u4:2-flush-8:0]
root          35  0.0  0.0      0     0 ?        S    07:44   0:00 [oom_reaper]
root          36  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-write]
root          37  0.0  0.0      0     0 ?        S    07:44   0:00 [kcompactd0]
root          38  0.0  0.0      0     0 ?        SN   07:44   0:00 [ksmd]
root          39  0.0  0.0      0     0 ?        SN   07:44   0:00 [khugepaged]
root          40  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-kinte]
root          41  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-kbloc]
root          42  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-blkcg]
root          43  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-tpm_d]
root          44  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-edac-]
root          45  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-devfr]
root          46  0.1  0.0      0     0 ?        I<   07:44   0:02 [kworker/0:1H-kblockd]
root          47  0.0  0.0      0     0 ?        S    07:44   0:00 [kswapd0]
root          55  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-kthro]
root          57  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-acpi_]
root          58  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-mld]
root          59  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-ipv6_]
root          64  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-kstrp]
root          66  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/u5:0]
root         147  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/1:1H-kblockd]
root         172  0.0  0.0      0     0 ?        I<   07:44   0:00 [kworker/R-crypt]
```

Utilizo cut para extraer la última columna de la salida de ps -aux:

ps -aux | tr -s ' ' | cut -d ' ' -f $(($(head -n 1 <(ps -aux) | wc -w)))

- tr -s ' ': Este comando elimina cualquier espacio adicional entre las columnas para garantizar que solo haya un espacio entre cada columna.
- cut -d ' ' -f $(($(head -n 1 <(ps -aux) | wc -w))): Esto extraerá la última columna de la salida de ps -aux.

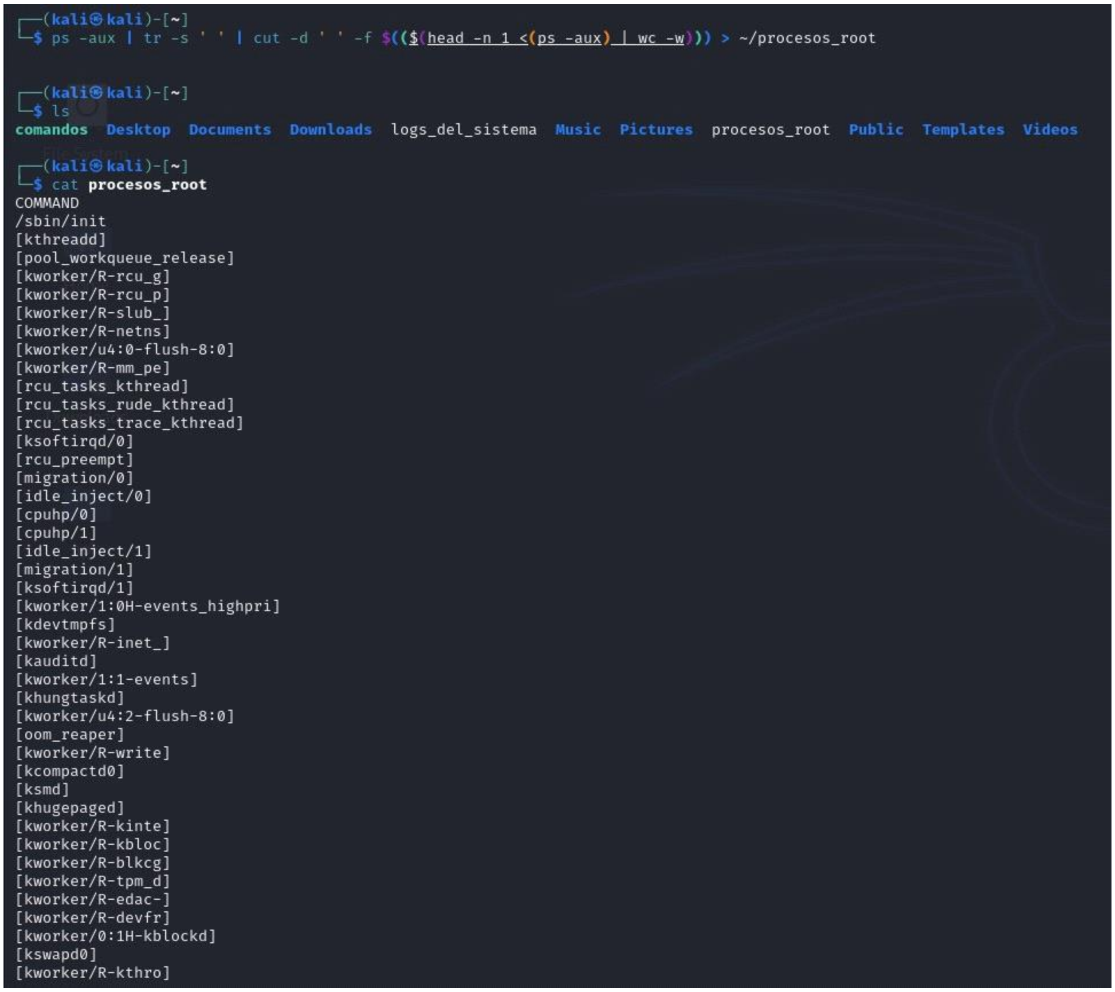


```
┌──(kali㉿kali)-[~]
└─$ ps -aux | tr -s ' ' | cut -d ' ' -f $(($(head -n 1 <(ps -aux) | wc -w)))

COMMAND
/sbin/init
[kthreadd]
[pool_workqueue_release]
[kworker/R-rcu_g]
[kworker/R-rcu_p]
[kworker/R-slub_]
[kworker/R-netns]
[kworker/u4:0-events_unbound]
[kworker/R-mm_pe]
[rcu_tasks_kthread]
[rcu_tasks_rude_kthread]
[rcu_tasks_trace_kthread]
[ksoftirqd/0]
[rcu_preempt]
[migration/0]
[idle_inject/0]
[cpuhp/0]
[cpuhp/1]
[idle_inject/1]
[migration/1]
[ksoftirqd/1]
[kworker/1:0H-events_highpri]
[kdevtmpfs]
[kworker/R-inet_]
[kauditd]
[kworker/1:1-events]
[khungtaskd]
[kworker/u4:2-flush-8:0]
[oom_reaper]
[kworker/R-write]
[kcompactd0]
[ksmd]
[khugepaged]
[kworker/R-kinte]
[kworker/R-kbloc]
[kworker/R-blkcg]
[kworker/R-tpm_d]
[kworker/R-edac-]
[kworker/R-devfr]
[kworker/0:1H-kblockd]
[kswapd0]
[kworker/R-kthro]
[kworker/R-acpi_]
[kworker/R-mld]
[kworker/R-ipv6_]
[kworker/R-kstrp]
[kworker/u5:0]
[kworker/1:1H-kblockd]
[kworker/R-crypt]
```

Ahora, redirijo esta salida hacia un archivo llamado procesos_root en el home directory:

ps -aux | tr -s ' ' | cut -d ' ' -f $(($(head -n 1 <(ps -aux) | wc -w))) > ~/procesos_root

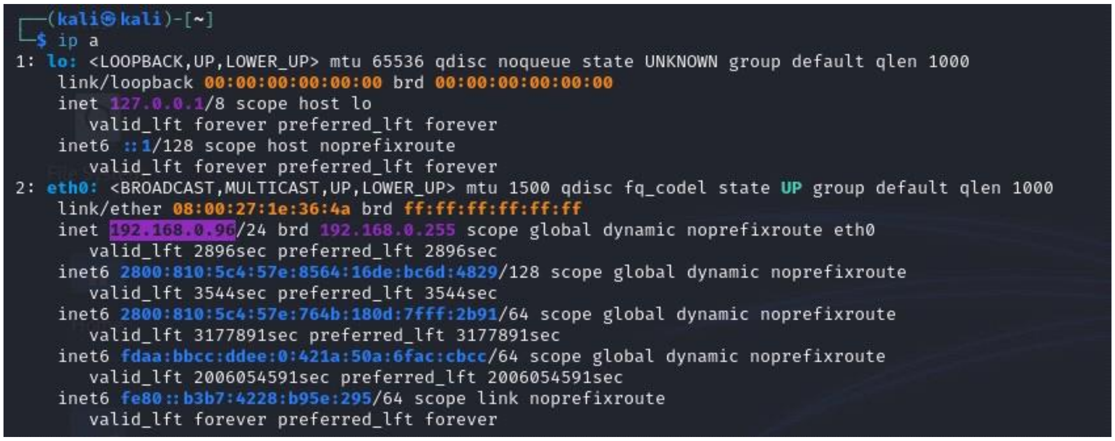
y compruebo que el contenido sea el que busqué extraer:



```
┌──(kali㉿kali)-[~]
└─$ ps -aux | tr -s ' ' | cut -d ' ' -f $(($(head -n 1 <(ps -aux) | wc -w))) > ~/procesos_root

┌──(kali㉿kali)-[~]
└─$ ls
comandos  Desktop  Documents  Downloads  logs_del_sistema  Music  Pictures  procesos_root  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ cat procesos_root
COMMAND
/sbin/init
[kthreadd]
[pool_workqueue_release]
[kworker/R-rcu_g]
[kworker/R-rcu_p]
[kworker/R-slub_]
[kworker/R-netns]
[kworker/u4:0-flush-8:0]
[kworker/R-mm_pe]
[rcu_tasks_kthread]
[rcu_tasks_rude_kthread]
[rcu_tasks_trace_kthread]
[ksoftirqd/0]
[rcu_preempt]
[migration/0]
[idle_inject/0]
[cpuhp/0]
[cpuhp/1]
[idle_inject/1]
[migration/1]
[ksoftirqd/1]
[kworker/1:0H-events_highpri]
[kdevtmpfs]
[kworker/R-inet_]
[kauditd]
[kworker/1:1-events]
[khungtaskd]
[kworker/u4:2-flush-8:0]
[oom_reaper]
[kworker/R-write]
[kcompactd0]
[ksmd]
[khugepaged]
[kworker/R-kinte]
[kworker/R-kbloc]
[kworker/R-blkcg]
[kworker/R-tpm_d]
[kworker/R-edac-]
[kworker/R-devfr]
[kworker/0:1H-kblockd]
[kswapd0]
[kworker/R-kthro]
```

SSH (Secure Shell) & Putty:

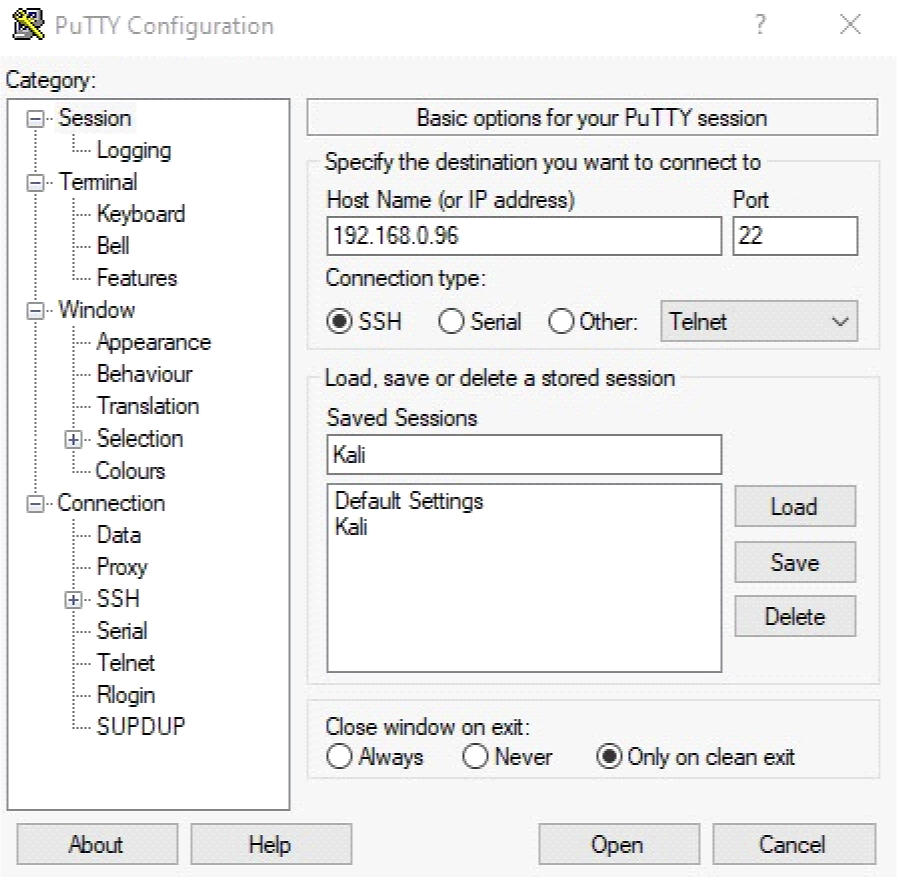
ip a



```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1e:36:4a brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.96/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
       valid_lft 2896sec preferred_lft 2896sec
    inet6 2800:810:5c4:57e:8564:16de:bc6d:4829/128 scope global dynamic noprefixroute
       valid_lft 3544sec preferred_lft 3544sec
    inet6 2800:810:5c4:57e:764b:180d:7fff:2b91/64 scope global dynamic noprefixroute
       valid_lft 3177891sec preferred_lft 3177891sec
    inet6 fdaa:bbcc:ddee:0:421a:50a:6fac:cbcc/64 scope global dynamic noprefixroute
       valid_lft 2006054591sec preferred_lft 2006054591sec
    inet6 fe80::b3b7:4228:b95e:295/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

IP: 192.168.0.96

Saving the session:



Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.96

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

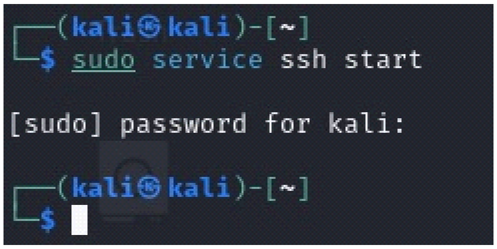
Saved Sessions: Kali

Default Settings
Kali

Close window on exit: ○ Always ○ Never ● Only on clean exit

Levantando servidor SSH:

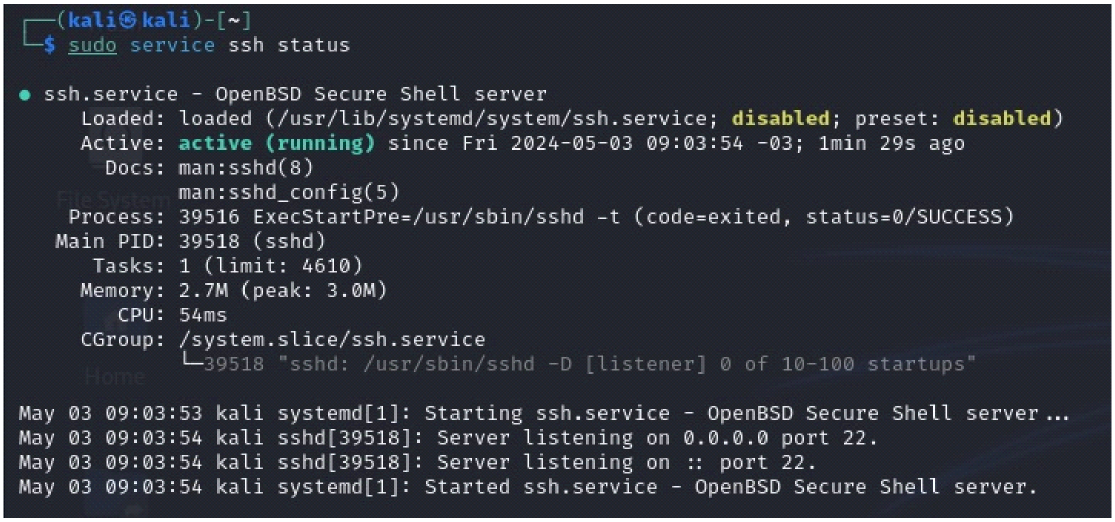
sudo service ssh start



```
┌──(kali㉿kali)-[~]
└─$ sudo service ssh start

[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ 
```

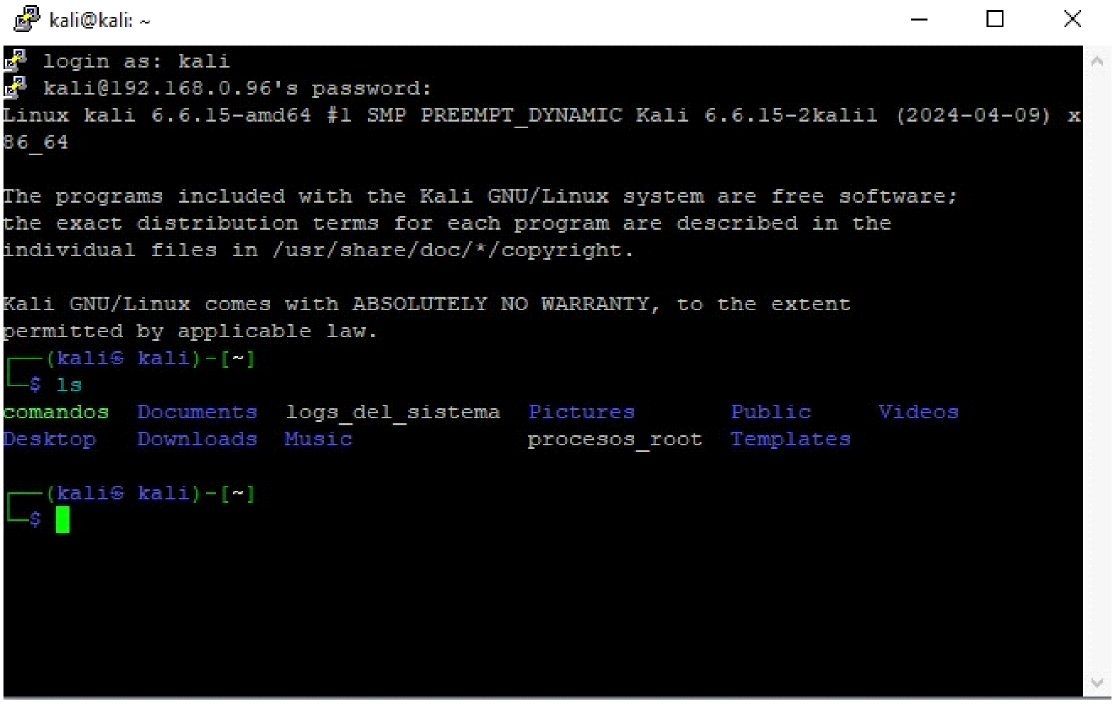
Verificando estado del servidor:

sudo service ssh status



```
┌──(kali㉿kali)-[~]
└─$ sudo service ssh status

● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
     Active: active (running) since Fri 2024-05-03 09:03:54 -03; 1min 29s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 39516 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 39518 (sshd)
      Tasks: 1 (limit: 4610)
     Memory: 2.7M (peak: 3.0M)
        CPU: 54ms
     CGroup: /system.slice/ssh.service
             └─39518 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

May 03 09:03:53 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 03 09:03:54 kali sshd[39518]: Server listening on 0.0.0.0 port 22.
May 03 09:03:54 kali sshd[39518]: Server listening on :: port 22.
May 03 09:03:54 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

Conectando al servidor mediante SSH:

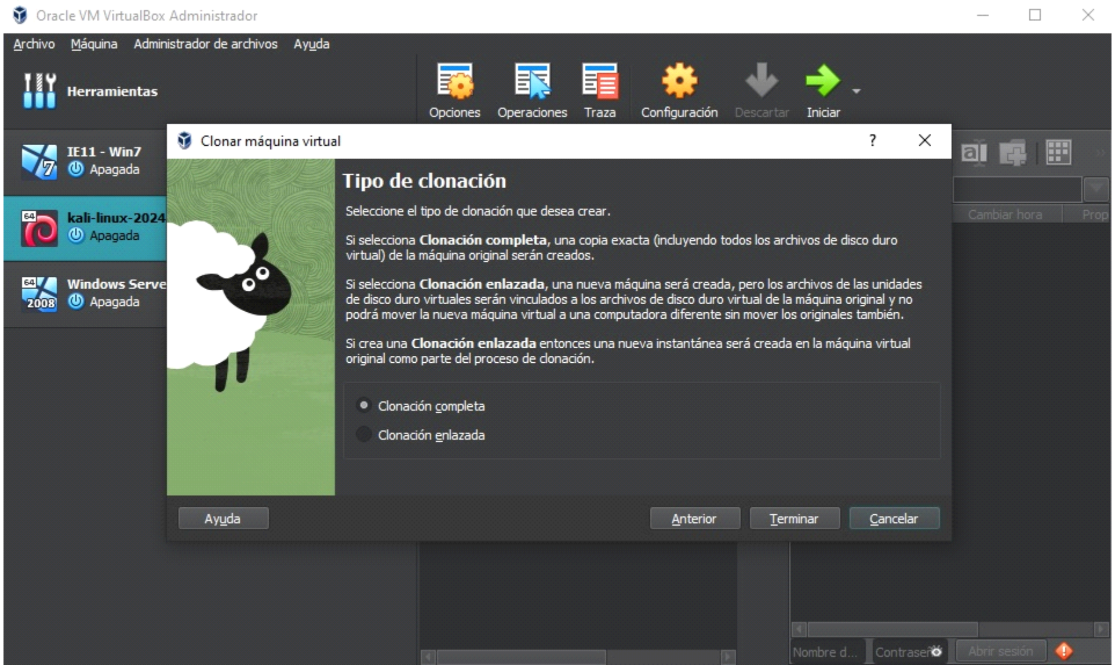


```
login as: kali
kali@192.168.0.96's password:
Linux kali 6.6.15-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-04-09) x
86_64

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)-[~]
└─$ ls
comandos  Documents  logs_del_sistema  Pictures       Public     Videos
Desktop   Downloads  Music             procesos_root  Templates

┌──(kali㉿kali)-[~]
└─$ 
```

Dato personal: Aunque éste no era un ejercicio en sí, quise hacer la práctica porque me trajo muchos recuerdos cuando utilizabamos este método para conectar y alojar nuestros bots Eggdrop para IRC en tiempos de cybercafé. Disclaimer: Con fines éticos de investigacion en servidores propios.

Ejercicio: Crear una máquina virtual ya sea clonando la existente o instalando desde cero una nueva, obtener la dirección ip de la misma y realizar una conexión ssh desde una virtual hasta la otra.

Clonando VM:



Clonar máquina virtual

Tipo de clonación

Seleccione el tipo de clonación que desea crear.

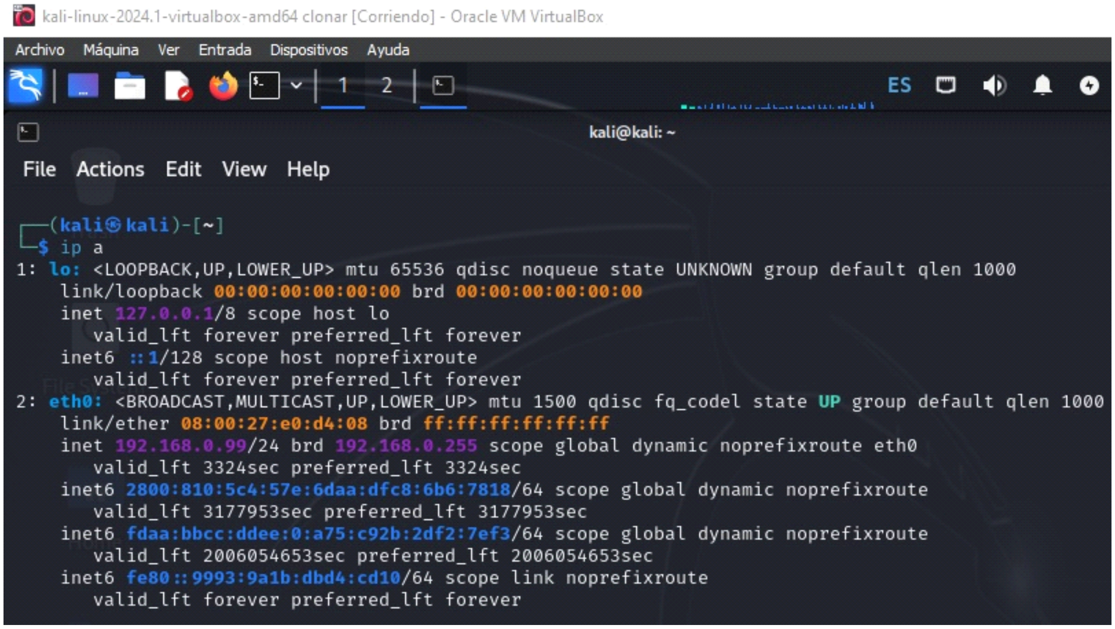
Si selecciona **Clonación completa**, una copia exacta (incluyendo todos los archivos de disco duro virtual) de la máquina original serán creados.

Si selecciona **Clonación enlazada**, una nueva máquina será creada, pero los archivos de las unidades de disco duro virtuales serán vinculados a los archivos de disco duro virtual de la máquina original y no podrá mover la nueva máquina virtual a una computadora diferente sin mover los originales también.

Si crea una **Clonación enlazada** entonces una nueva instantánea será creada en la máquina virtual original como parte del proceso de clonación.

● Clonación completa
○ Clonación enlazada

Obteniendo direccion IP:

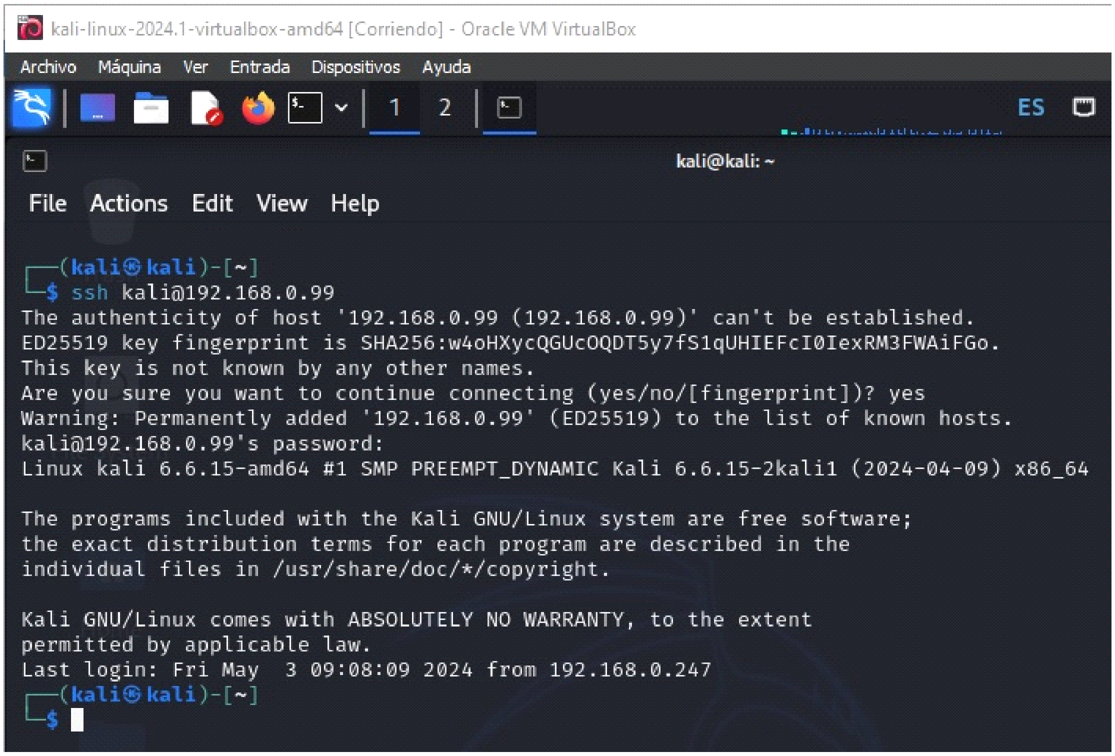

kali-linux-2024.1-virtualbox-amd64 clonar [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
ES
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e0:d4:08 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.99/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
       valid_lft 3324sec preferred_lft 3324sec
    inet6 2800:810:5c4:57e:6daa:dfc8:6b6:7818/64 scope global dynamic noprefixroute
       valid_lft 3177953sec preferred_lft 3177953sec
    inet6 fdaa:bbcc:ddee:0:a75:c92b:2df2:7ef3/64 scope global dynamic noprefixroute
       valid_lft 2006054653sec preferred_lft 2006054653sec
    inet6 fe80::9993:9a1b:dbd4:cd10/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

IP Cliente: 192.168.0.96

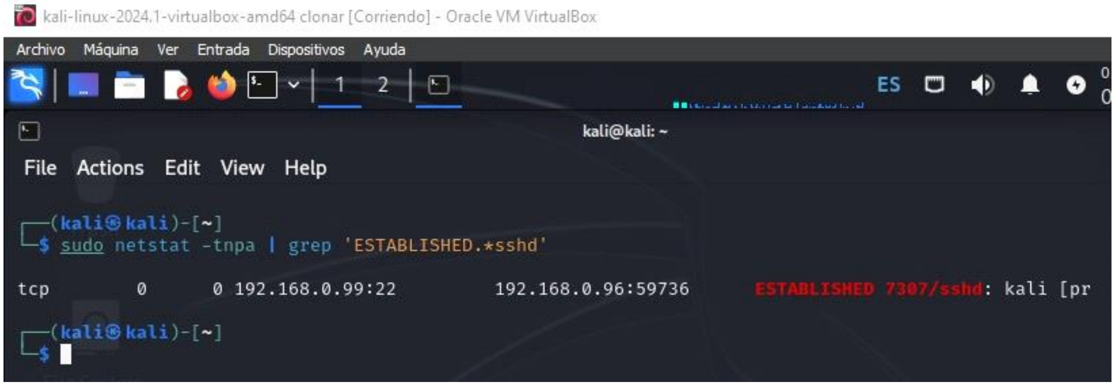
IP Servidor: 192.168.0.99

Estableciendo conexión SSH:


kali-linux-2024.1-virtualbox-amd64 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
1
2
ES
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ssh kali@192.168.0.99
The authenticity of host '192.168.0.99 (192.168.0.99)' can't be established.
ED25519 key fingerprint is SHA256:w4oHXycQGUcOQDT5y7fS1qUHIEFcI0IexRM3FWAiFGo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.99' (ED25519) to the list of known hosts.
kali@192.168.0.99's password:
Linux kali 6.6.15-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-04-09) 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.
Last login: Fri May  3 09:08:09 2024 from 192.168.0.247
┌──(kali㉿kali)-[~]
└─$ 
```

Comprobando conexión SSH establecida desde el servidor:


kali-linux-2024.1-virtualbox-amd64 clonar [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
1
2
ES
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ sudo netstat -tnpa | grep 'ESTABLISHED.*sshd'

tcp        0      0 192.168.0.99:22         192.168.0.96:59736      ESTABLISHED 7307/sshd: kali [pr

┌──(kali㉿kali)-[~]
└─$ 
```

Explicando el comando netstat:

- sudo: Se utiliza para ejecutar el comando con privilegios de superusuario, ya que algunas opciones de netstat requieren permisos especiales para mostrar cierta información.
- netstat -tnpa: Muestra una lista de todas las conexiones de red y los puertos que están escuchando, junto con los procesos asociados.
- -t: Muestra solo las conexiones TCP.
- -n: Muestra las direcciones IP y los números de puerto en formato numérico en lugar de nombres.
- -p: Muestra los identificadores de proceso (PID) y los nombres de los programas que

están utilizando los sockets.

- -a: Muestra todas las conexiones y puertos, tanto escuchando como establecidos.
- grep 'ESTABLISHED.*sshd': Filtra la salida de netstat para mostrar solo las conexiones SSH establecidas.
- 'ESTABLISHED.*sshd': Esto busca las conexiones que están en estado "ESTABLISHED" y están asociadas con el proceso SSH (sshd).

Dato personal: Mi experiencia con el comando netstat se remonta a épocas de cybercafé cuando mientras enviabamos un archivo por MSN se utilizaba el comando netstat -n para ver las conexiones de red activas.

Rodrigo Vila.-