



CY-302 Programación Avanzada

Entrega 2: Herramientas base

Proyecto: Red Team vs Blue Team en Azure

Docente: Andrés Vargas

Estudiantes:

- Eduardo Jiménez Bonilla
- Andrés Alonso Obando Fallas
  - Daniel Vargas
  - Victoria Arguedas Chacón
  - Fabricio Calderón Medrano
  - Jose Castillo

Fecha: 19 de noviembre de 2025

Repositorio: [https://github.com/fxbricm/proyecto\\_ciberseguridad/tree/main](https://github.com/fxbricm/proyecto_ciberseguridad/tree/main)

## Contenido

Roles por equipo (Blue / Red) .....	3
Blue team script .....	3
Capturas de evidencia Blue_team: .....	3
Red team script .....	4
Capturas de Evidencia .....	5

## Roles por equipo (Blue / Red)

- Red\_team/scanner.py (Nmap con flags básicos: -sS, -sV, -Pn) y salida guardada.
- Blue\_team/os\_audit.py (usuarios, puertos abiertos, servicios) + blue\_team/log\_events.txt.

## Blue team script

Para ejecutar el script del Blue Team, primero se debe acceder a la máquina virtual en Azure mediante SSH y dirigirse al directorio donde se encuentra la carpeta blue\_team. Una vez dentro, se confirma que el archivo os\_audit.py esté presente y, se le otorgan permisos de ejecución con `chmod +x os_audit.py`. Luego, el script se ejecuta utilizando el comando `python3 os_audit.py`, lo que generará automáticamente el archivo `log_events.txt` dentro del mismo directorio. Finalmente, se revisa el contenido de ese archivo con `cat log_events.txt`.

## Capturas de evidencia Blue\_team:

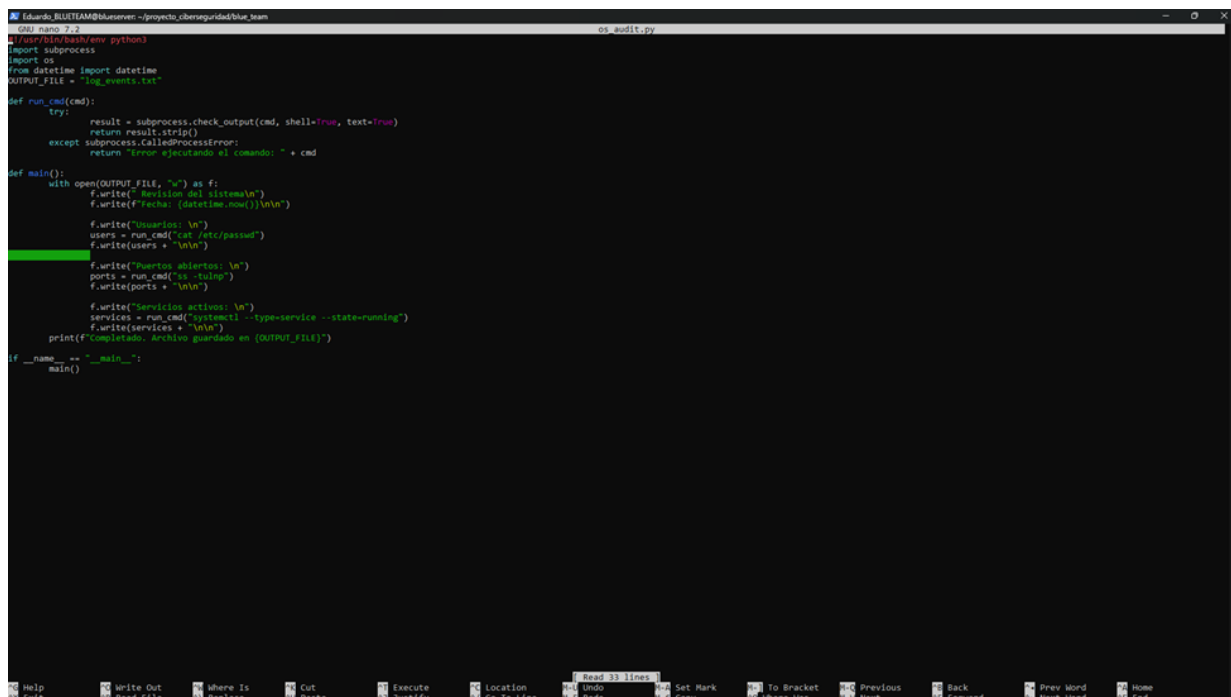
```
(venv) Eduardo_BLUETEAM@blueserver:~/proyecto_ciberseguridad/blue_team$ python3 os_audit.py
Completado. Archivo guardado en log_events.txt
(venv) Eduardo_BLUETEAM@blueserver:~/proyecto_ciberseguridad/blue_team$ _
```

```
GNU nano 2.7.2 log_events.txt
Revision del sistema
Fecha: 2025-11-20 03:26:15, 968172

Usuarios:
root:x:0:0:root:/root:/usr/sbin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail list Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
lxc:x:42:65534:/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/usr/sbin/nologin
systemd-timesync:x:999:999:systemd Time Synchronization:/usr/sbin/nologin
dhcpd:x:100:65534:DHCP Client Daemon,,/usr/lib/dhcpd/bin/false
messagebus:x:101:101:/nonexistent:/usr/sbin/nologin
syslog:x:102:102:/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver:/usr/sbin/nologin
cups:x:103:103:/run/cups:/usr/sbin/nologin
sshd:x:104:104:TPM Software Stack,,/usr/lib/tpm/bin/false
ssh:x:105:65534:/run/ssh:/usr/sbin/nologin
pollinate:x:106:11:/var/cache/pollinate/bin/false
tcpdump:x:107:100:/nonexistent:/usr/sbin/nologin
landscape:x:108:109:/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:998:999:firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
polkitd:x:989:989:user for polkitd:/usr/sbin/nologin
chrony:x:109:113:Chrony daemon,,/var/lib/chrony:/usr/sbin/nologin
Eduardo_BLUETEAM:x:1000:1000:/home/Eduardo_BLUETEAM:/bin/bash

Puertos abiertos:
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 0 127.0.0.53:53 0.0.0.0:*
udp UNCONN 0 0 172.16.0.42eth0:68 0.0.0.0:*
udp UNCONN 0 0 127.0.0.1:323 0.0.0.0:*
udp UNCONN 0 0 [::]:323 [::]:*
tcp LISTEN 0 4096 127.0.0.53:53 0.0.0.0:*
tcp LISTEN 0 4096 0.0.0.0:22 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.24:53 0.0.0.0:*
tcp LISTEN 0 4096 [::]:22 [::]:*

Servicios activos:
UNIT LOAD ACTIVE SUB DESCRIPTION
chrony.service loaded active running chrony, an NTP client/server
cron.service loaded active running Regular background program processing daemon
dbus.service loaded active running D-Bus System Message Bus
fwupd.service loaded active running Firmware update daemon
getty@tty1.service loaded active running Getty on tty1
hv-kvp-daemon.service loaded active running Hyper-V KVP Protocol Daemon
modemmanager.service loaded active running Modem Manager
multipathd.service loaded active running Device-Mapper Multipath Device Controller
networkd-dispatcher.service loaded active running Dispatcher daemon for systemd-networkd
```



```
idardo@BLUSTIAM@bluesteam: ~/projecto_ciberseguridad/blue_team
GNU nano 7.2 os_audit.py
#!/usr/bin/bash/env python3
import subprocess
import os
from datetime import datetime
OUTPUT_FILE = "log_events.txt"

def run_cmd(cmd):
    try:
        result = subprocess.check_output(cmd, shell=True, text=True)
        return result.strip()
    except subprocess.CalledProcessError:
        return "Error ejecutando el comando: " + cmd

def main():
    with open(OUTPUT_FILE, "w") as f:
        f.write("Version del sistema\n")
        f.write(f"Fecha: {datetime.now()}\n\n")
        f.write("Usuarios: \n")
        users = run_cmd("cat /etc/passwd")
        f.write(users + "\n\n")
        f.write("Puertos abiertos: \n")
        ports = run_cmd("ss -tulnp")
        f.write(ports + "\n\n")
        f.write("Servicios activos: \n")
        services = run_cmd("systemctl --type=service --state=running")
        f.write(services + "\n\n")
        print(f"Finalizado. Archivo guardado en {OUTPUT_FILE}")

if __name__ == "__main__":
    main()
```

## Red team script

Para ejecutar el script del Red Team, es necesario tener instalado Python y Nmap. Para empezar, se abre la terminal del dispositivo en el que se encuentra,. Luego, se dirige al directorio donde esta el documento scanner.py y se ejecuta lo siguiente:

1. En Windows: sudo python scanner.py
  2. En MacOS/Linux: sudo python3 scanner.py
- Ambos van a solicitar la contraseña de tu dispositivo, debido a que se esta ejecutando con permisos sudo.

Esto generará una carpeta llamada "results" en la que se encuentran todos los archivos .txt de los escaneos con la fecha y hora incluidas en el nombre del archivo.

## Capturas de Evidencia

```
red_team — nmap ◀ sudo — 80x24
Last login: Wed Nov 19 16:53:27 on ttys000
[fabriciocalderon@fxbris-MacBook-Air ~ % cd /Users/fabriciocalderon/U/Progra\ Ava]
nizada/proyecto_ciberseguridad/red_team
[fabriciocalderon@fxbris-MacBook-Air red_team % sudo python3 scanner.py
>Password:
=== Scanner Red Team ===
Ingresa las IPs que deseas escanear (separadas por comas): 51.57.65.187

Guardando resultados en: red_team/results/scan_results_2025-11-19_18-49-14.txt

Iniciando escaneo...
█
```

```
scan_results_2025-11-19_18-49-14.txt
===== RESULTADOS DEL ESCANEO (RED TEAM) =====
Fecha: 2025-11-19_18-49-14
Flags usados: -sS -sV -Pn

--- Escaneando 51.57.65.187 ---
Starting Nmap 7.98 ( https://nmap.org ) at 2025-11-19 18:49 -0600
Nmap scan report for 51.57.65.187
Host is up (0.098s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.73 seconds
```