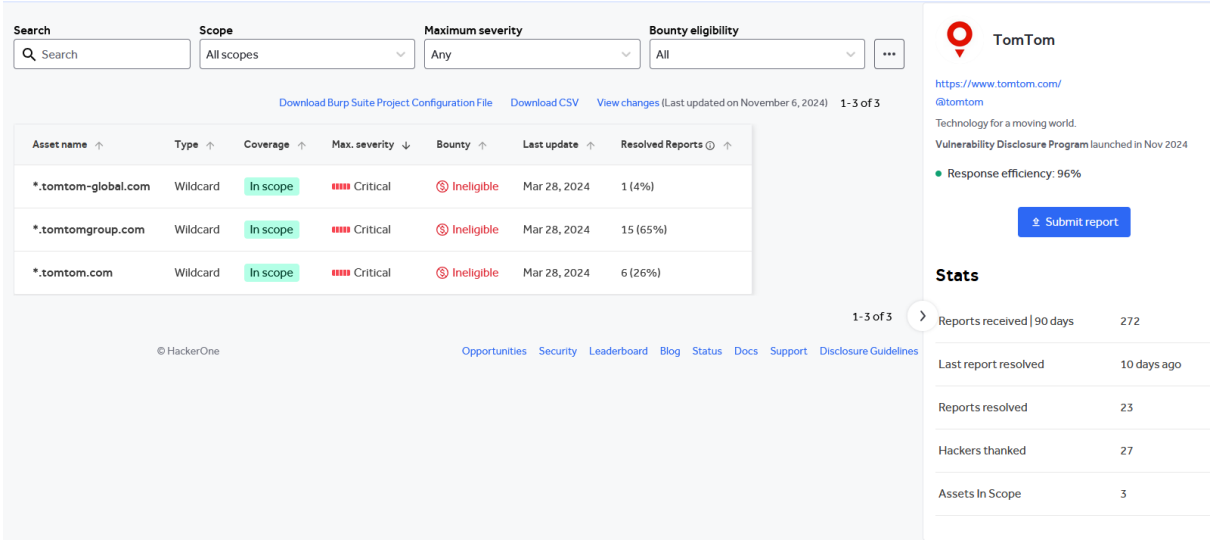
Ejercicio final Red Team

Ejercicio 1: Planificación y reconocimiento de una organización

Escogida empresa objetivo desde HackerOne para tener en consideración el scope:

https://hackerone.com/mintel/policy_scopes

*.tomtom.com



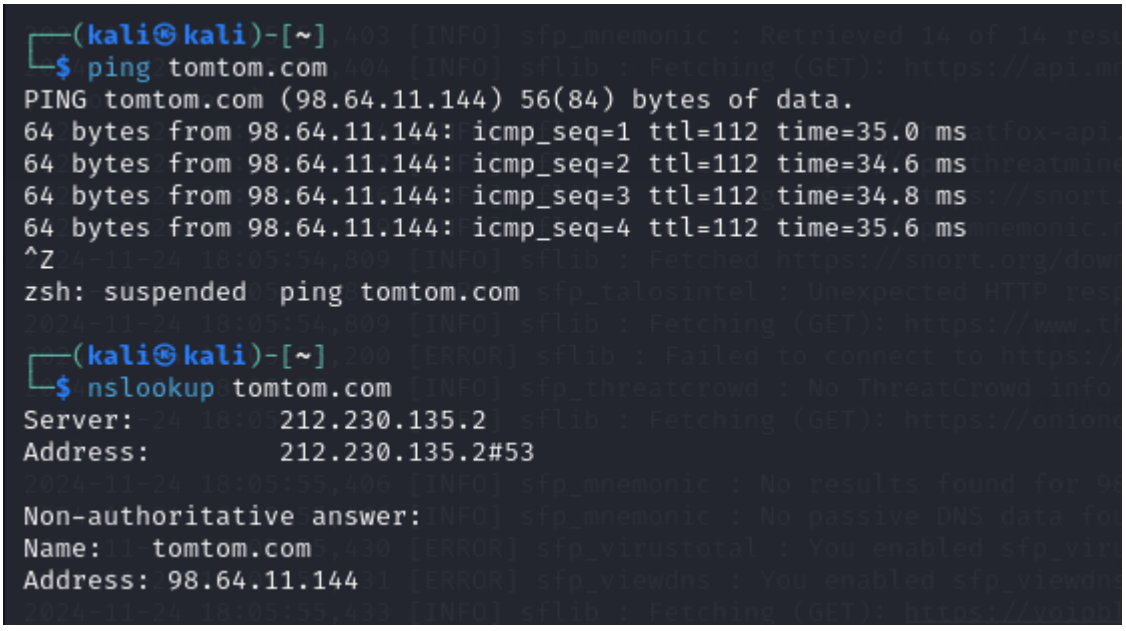
1. Nombres / Empresas incluidas para la empresa matriz

TomTom NV es una compañía global con sede en Ámsterdam, Países Bajos. La empresa está dividida en dos segmentos principales:

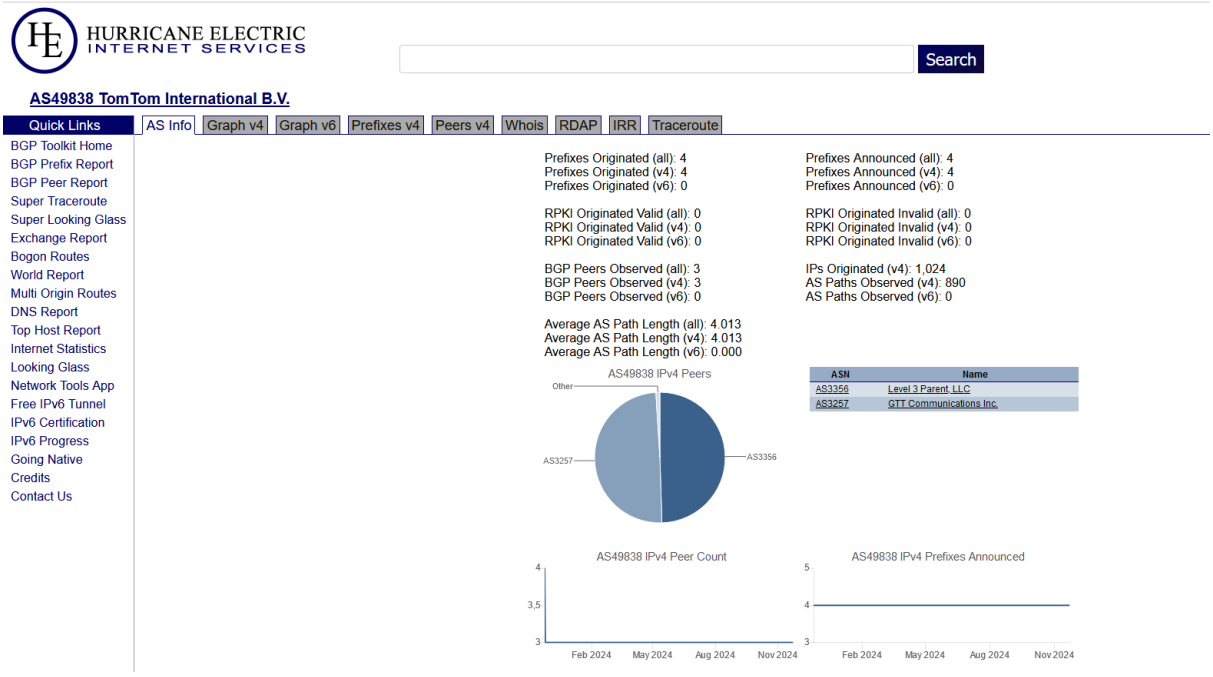
```
Location Technology: Incluye mapas, software y servicios pa
ra empresas y el sector automotriz.
Consumer: Productos de navegación portátiles y aplicaciones
móviles
```

2. Sistemas autónomos

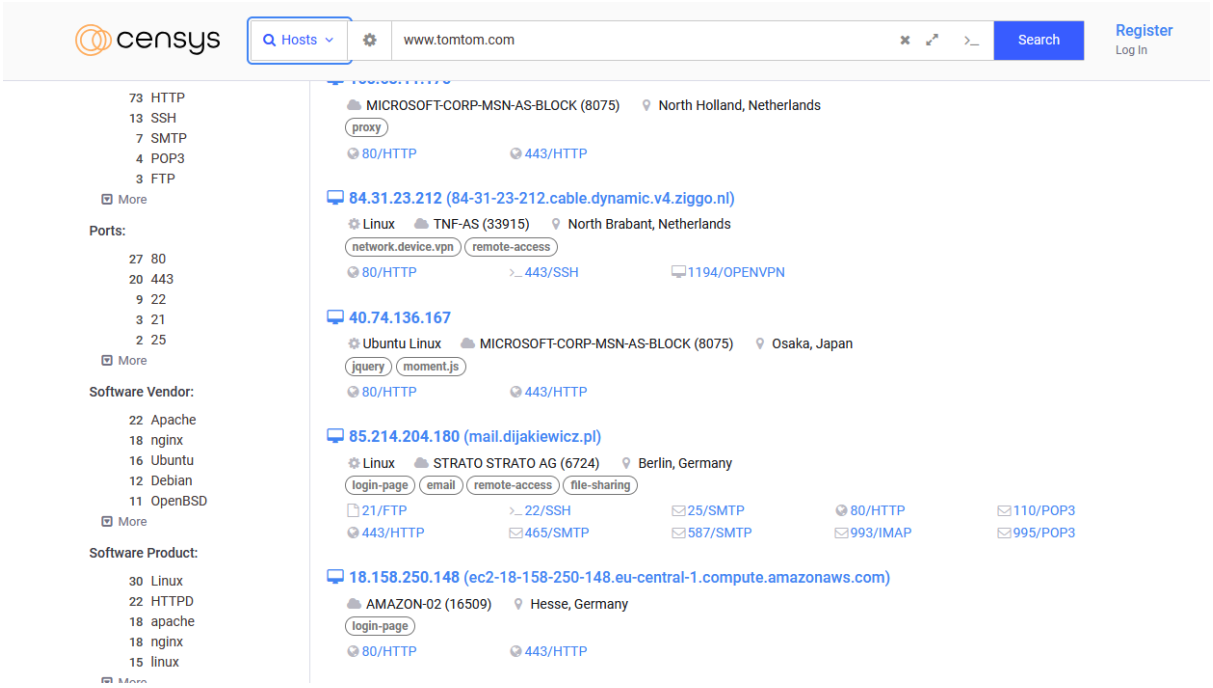
Haciendo `ping` y lanzando un `nslookup` a **tomtom.com** nos devuelve la ip de la url y del servidor.



Si buscamos la empresa en HURRICANE ELECTRIC averiguamos que su AS es **AS49838.**

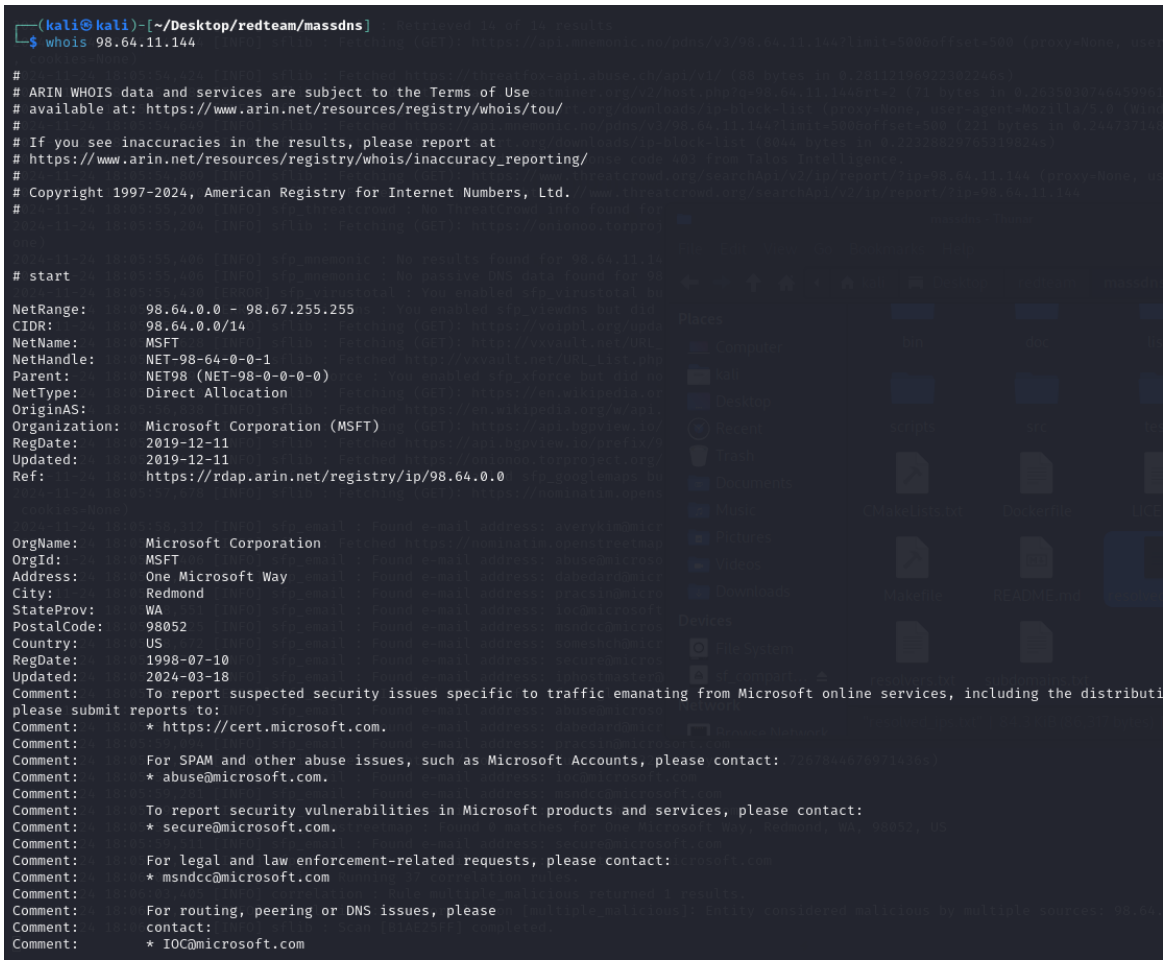


Al consultar en Censys vemos que existen varios equipos de la compañía con el puerto `ssh` abierto lo que podría ser una vía para un posible pentest.

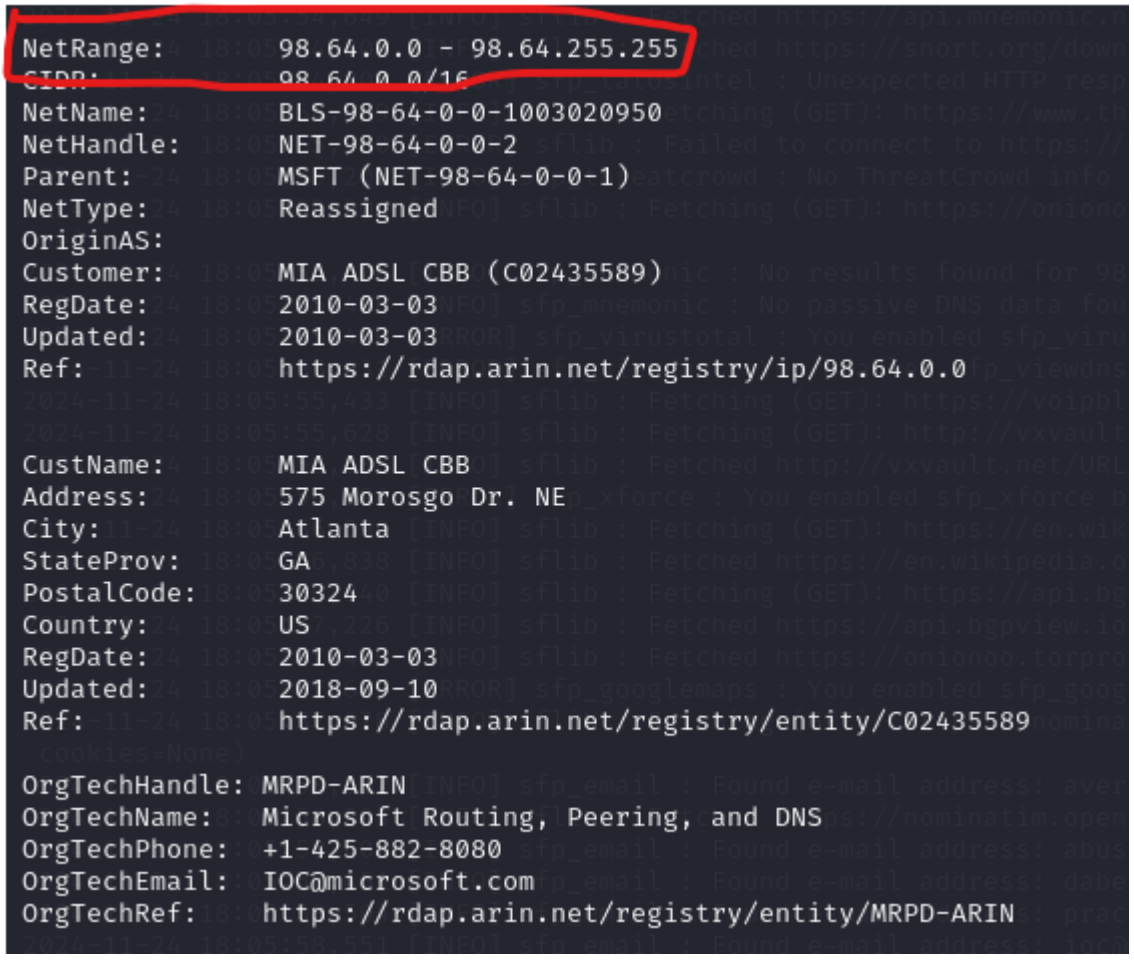


3. Rangos de red

Lanzamos un `whois` para saber el rango de red:



```
┌──(kali㉿kali)-[~/Desktop/redteam/massdns]
└─$ whois 98.64.11.144

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2024, American Registry for Internet Numbers, Ltd.
#

# start

NetRange:       98.64.0.0 - 98.67.255.255
CIDR:           98.64.0.0/14
NetName:        MSFT
NetHandle:      NET-98-64-0-0-1
Parent:         NET98 (NET-98-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Microsoft Corporation (MSFT)
RegDate:        2019-12-11
Updated:        2019-12-11
Ref:            https://rdap.arin.net/registry/ip/98.64.0.0


OrgName:        Microsoft Corporation
OrgId:          MSFT
Address:        One Microsoft Way
City:           Redmond
StateProv:      WA
PostalCode:     98052
Country:        US
RegDate:        1998-07-10
Updated:        2024-03-18
Comment:        To report suspected security issues specific to traffic emanating from Microsoft online services, including the distributi
please submit reports to:
Comment:        * https://cert.microsoft.com.
Comment:
Comment:        For SPAM and other abuse issues, such as Microsoft Accounts, please contact:
Comment:        * abuse@microsoft.com.
Comment:
Comment:        To report security vulnerabilities in Microsoft products and services, please contact:
Comment:        * secure@microsoft.com.
Comment:
Comment:        For legal and law enforcement-related requests, please contact:
Comment:        * msndcc@microsoft.com
Comment:
Comment:        For routing, peering or DNS issues, please
Comment:        contact:
Comment:        * IOC@microsoft.com
```



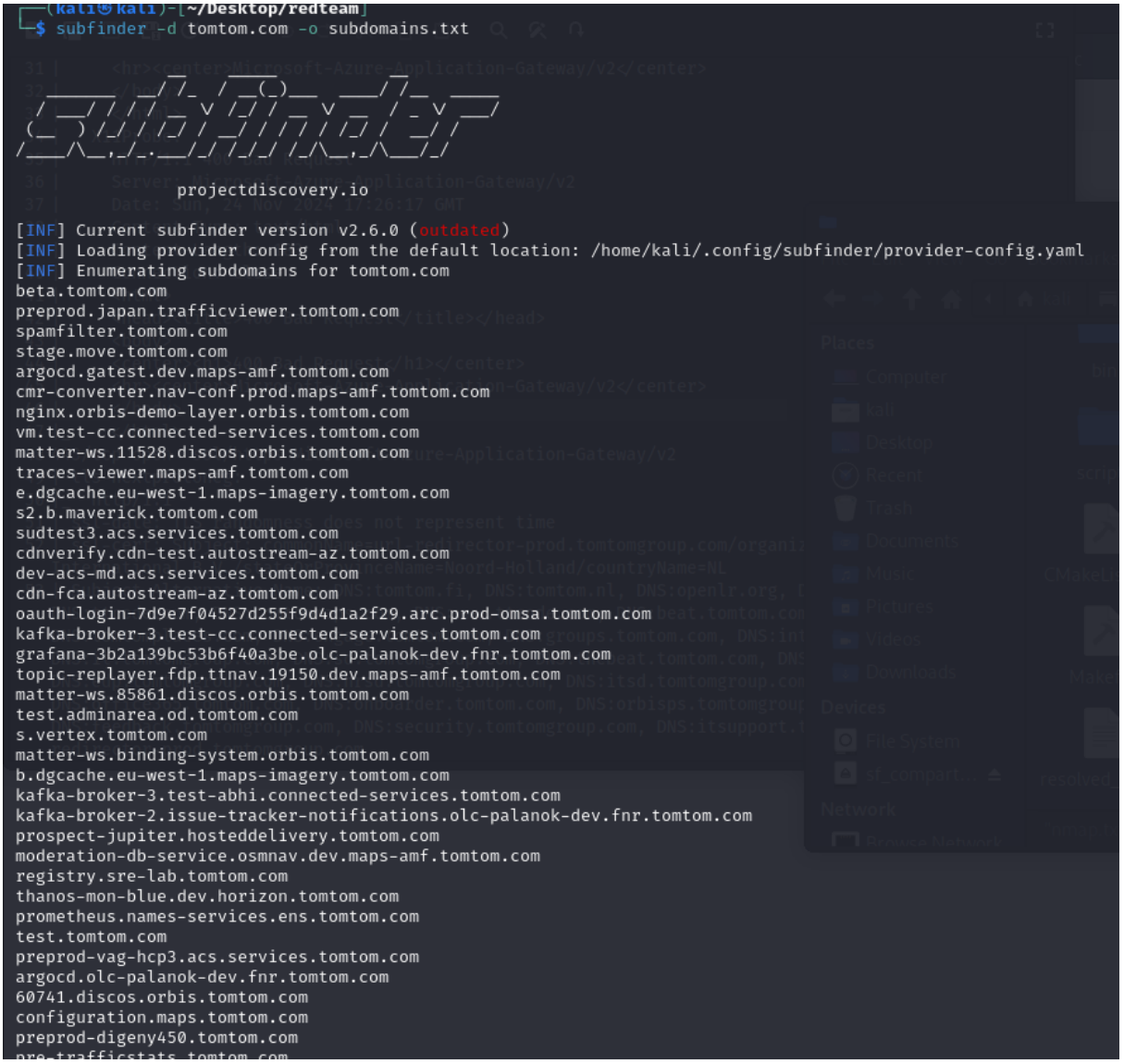
4. Dominios principales

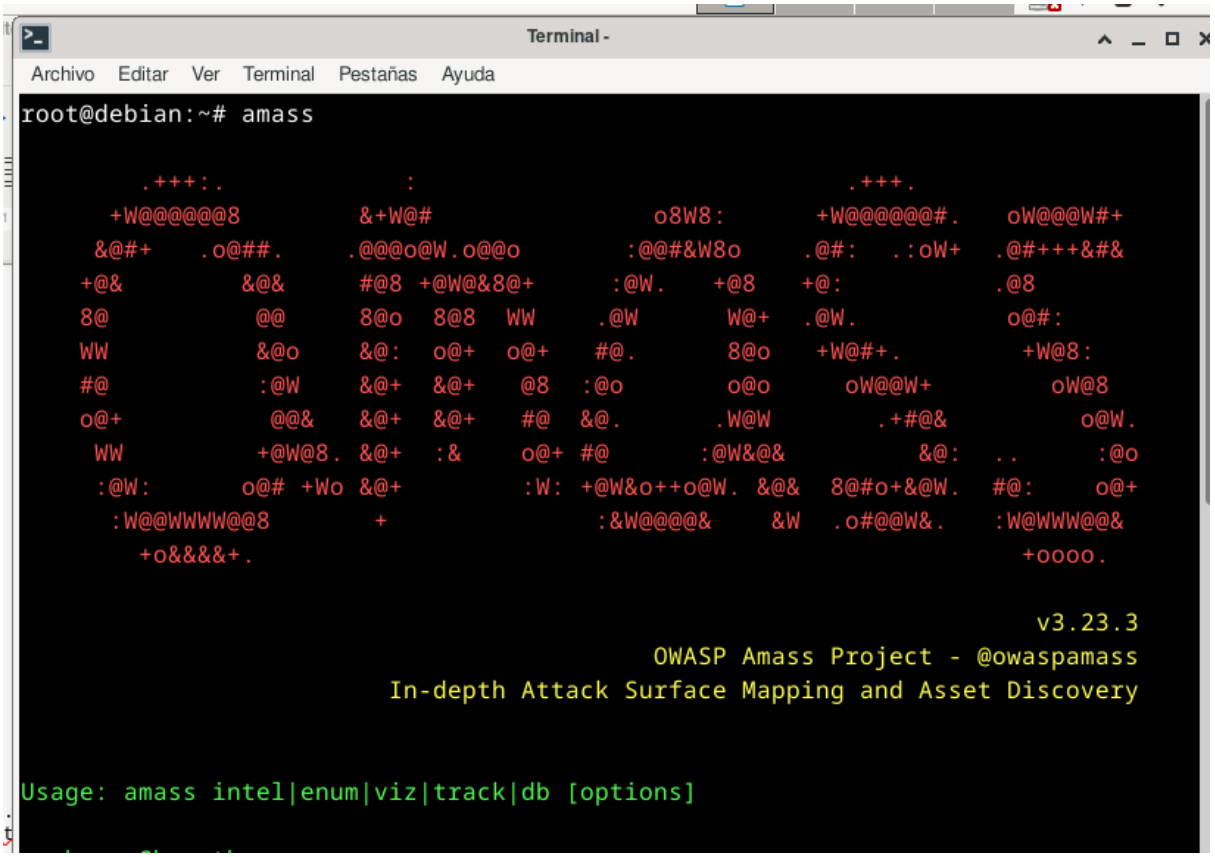
El dominio oficial de TomTom es **www.tomtom.com**, que actúa como el dominio raíz para varios servicios y subdominios.

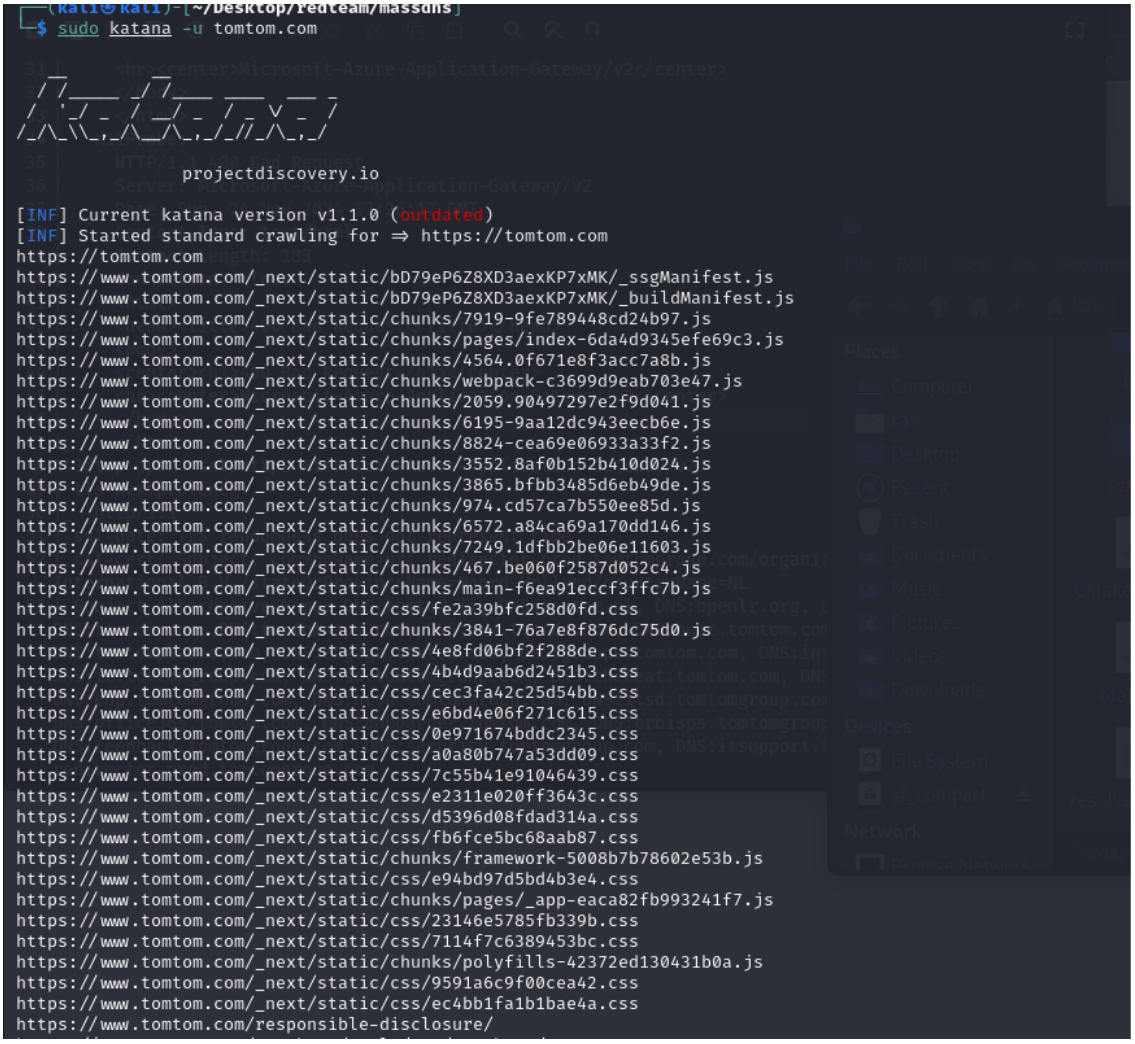
5. Subdominios identificados

Para enumerar subdominios se han utilizado las herramientas: `Amass, Katana y Subfinder.`

Se adjunta a la practica archivos .txt con los subdominios encontrados con cada herramienta.

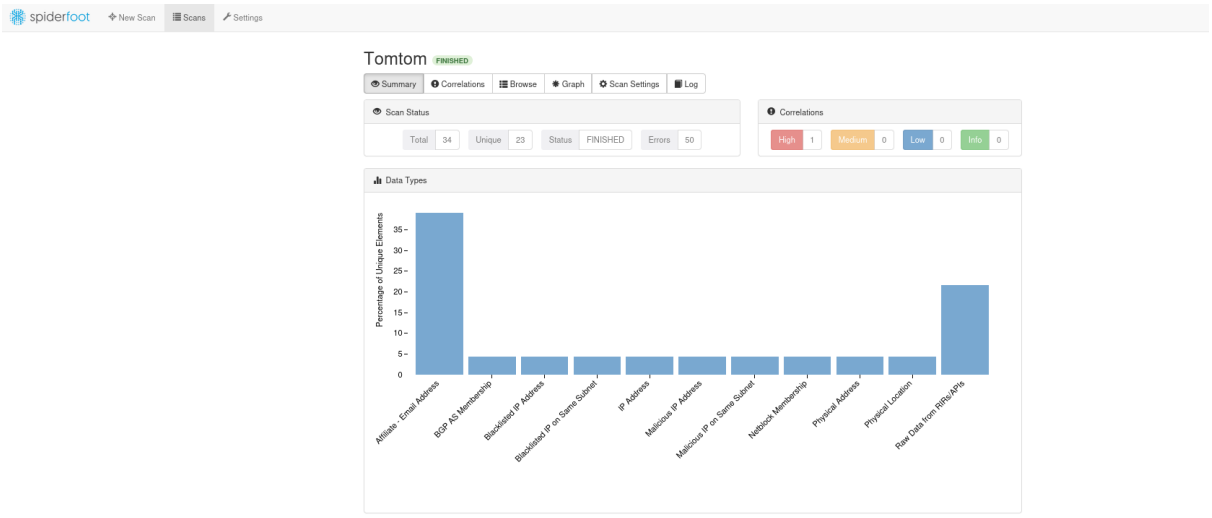

┌──(kali㉿kali)-[~/Desktop/redteam]
└─$ subfinder -d tomtom.com -o subdomains.txt
projectdiscovery.io
[INF] Current subfinder version v2.6.0 (outdated)
[INF] Loading provider config from the default location: /home/kali/.config/subfinder/provider-config.yaml
[INF] Enumerating subdomains for tomtom.com
beta.tomtom.com
preprod.japan.trafficviewer.tomtom.com
spamfilter.tomtom.com
stage.move.tomtom.com
argocd.gatest.dev.maps-amf.tomtom.com
cmr-converter.nav-conf.prod.maps-amf.tomtom.com
nginx.orbis-demo-layer.orbis.tomtom.com
vm.test-cc.connected-services.tomtom.com
matter-ws.11528.discos.orbis.tomtom.com
traces-viewer.maps-amf.tomtom.com
e.dgcache.eu-west-1.maps-imagery.tomtom.com
s2.b.maverick.tomtom.com
sudtest3.acs.services.tomtom.com
cdnverify.cdn-test.autostream-az.tomtom.com
dev-acs-md.acs.services.tomtom.com
cdn-fca.autostream-az.tomtom.com
oauth-login-7d9e7f04527d255f9d4d1a2f29.arc.prod-omsa.tomtom.com
kafka-broker-3.test-cc.connected-services.tomtom.com
grafana-3b2a139bc53b6f40a3be.olc-palanok-dev.fnr.tomtom.com
topic-replayer.fdp.ttnav.19150.dev.maps-amf.tomtom.com
matter-ws.85861.discos.orbis.tomtom.com
test.adminarea.od.tomtom.com
s.vertex.tomtom.com
matter-ws.binding-system.orbis.tomtom.com
b.dgcache.eu-west-1.maps-imagery.tomtom.com
kafka-broker-3.test-abhi.connected-services.tomtom.com
kafka-broker-2.issue-tracker-notifications.olc-palanok-dev.fnr.tomtom.com
prospect-jupiter.hosteddelivery.tomtom.com
moderation-db-service.osmnav.dev.maps-amf.tomtom.com
registry.sre-lab.tomtom.com
thanos-mon-blue.dev.horizon.tomtom.com
prometheus.names-services.ens.tomtom.com
test.tomtom.com
preprod-vag-hcp3.acs.services.tomtom.com
argocd.olc-palanok-dev.fnr.tomtom.com
60741.discos.orbis.tomtom.com
configuration.maps.tomtom.com
preprod-digeny450.tomtom.com

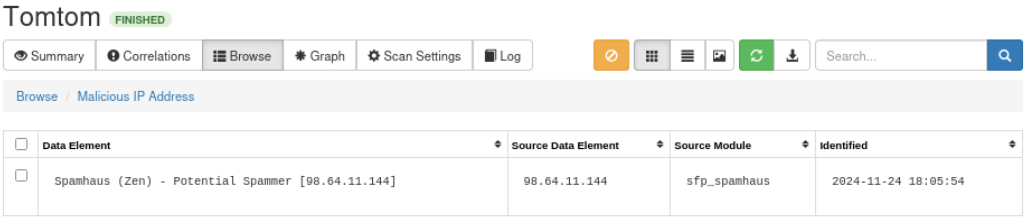

Terminal -
Archivo Editar Ver Terminal Pestañas Ayuda
root@debian:~# amass
v3.23.3
OWASP Amass Project - @owaspamass
In-depth Attack Surface Mapping and Asset Discovery
Usage: amass intel|enum|viz|track|db [options]



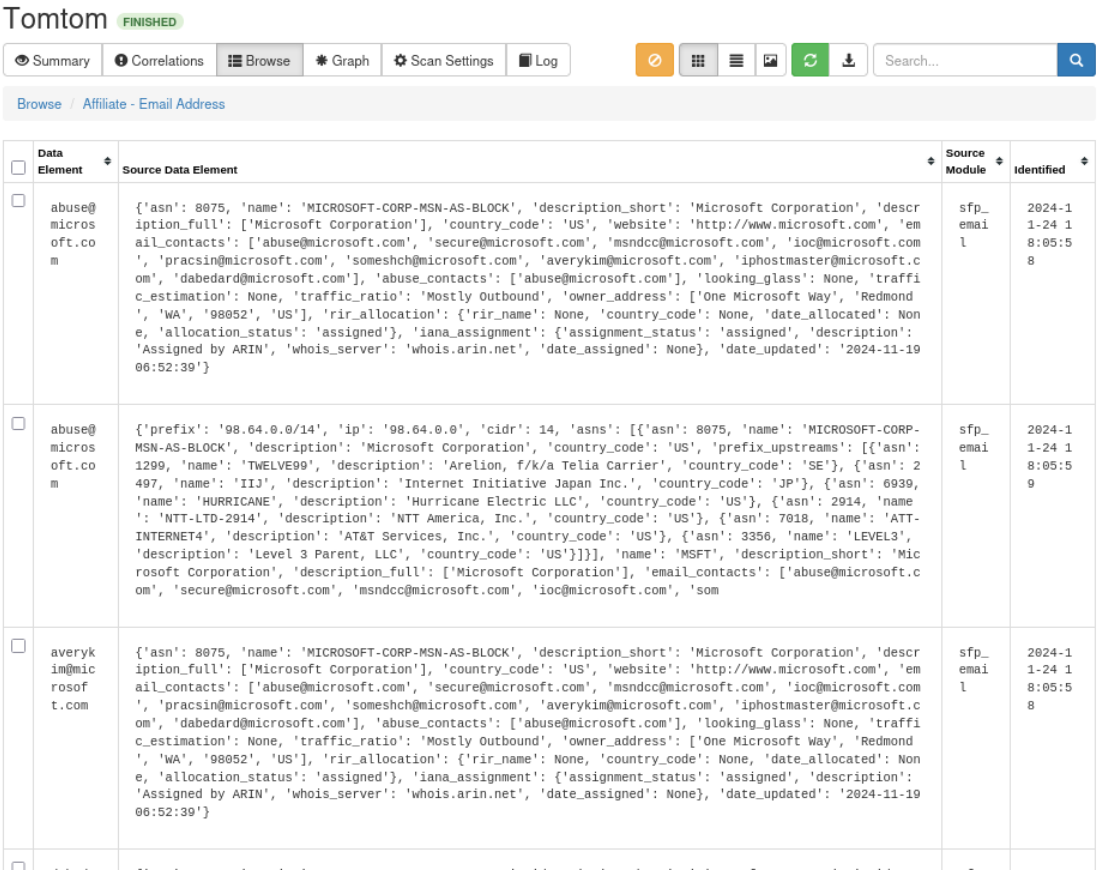
6. Posibles vectores de acceso:

Utilizamos la herramienta spiderfoot en busca de todo tipo de información: emails, direcciones de ip, direcciones físicas...

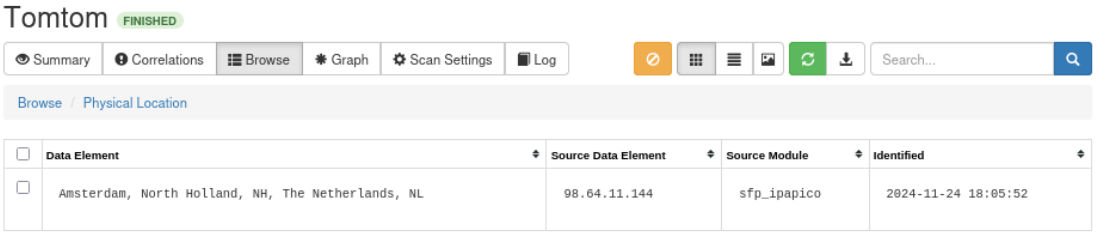




Como podemos ver esta ip tiene indicios de ser un Spammer pontencial.

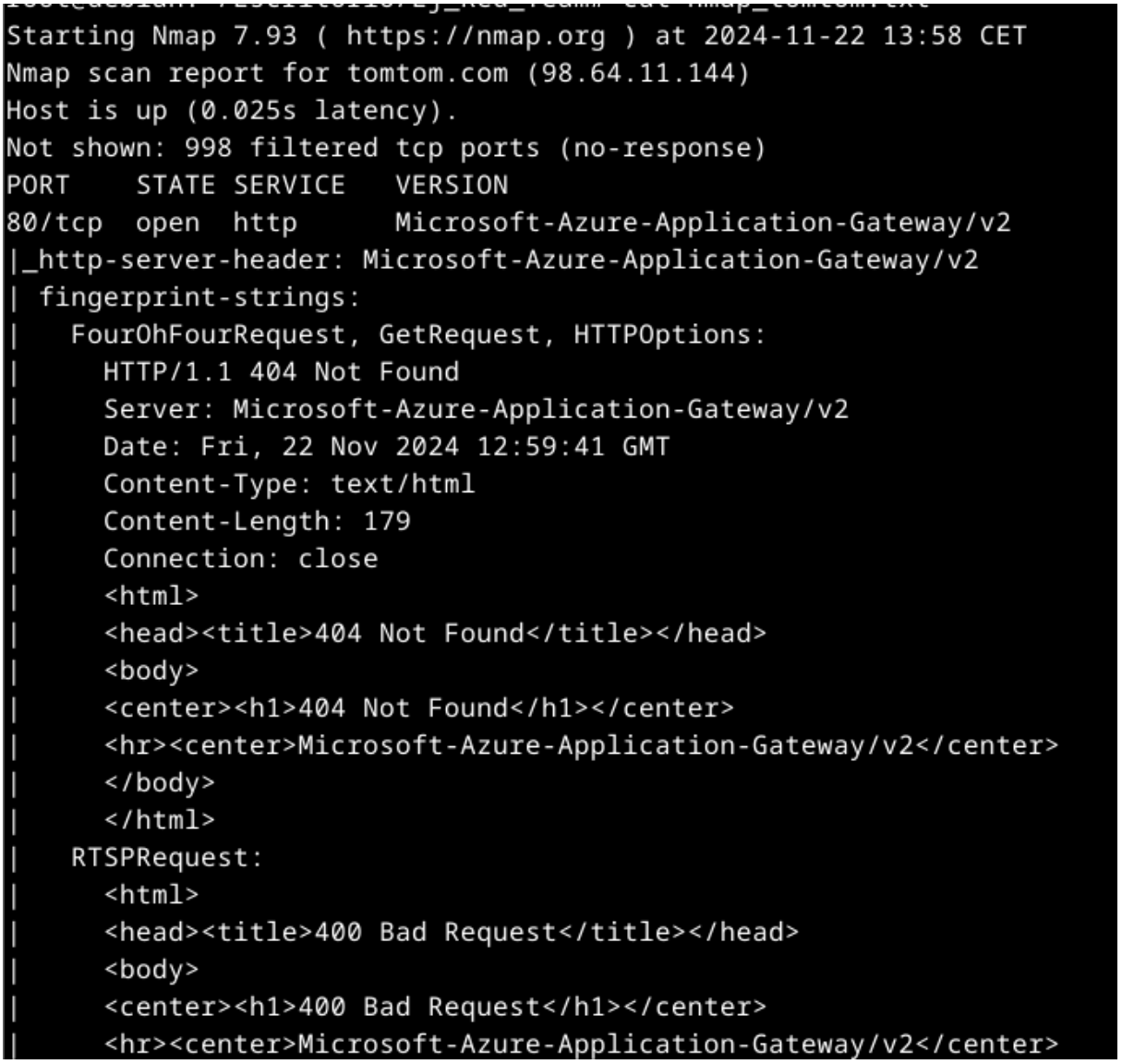


También hemos conseguido una serie de emails relacionados con Tomtom los cuales podriamos utilizar para hacer phishing.



Gracias a esta herramienta también sabemos la localización física del servidor.

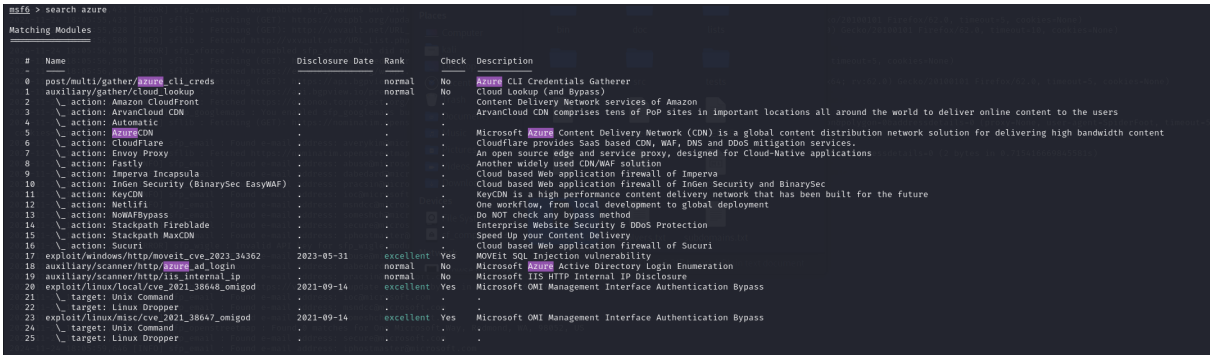
También hacemos un escaneo de tomtom.com con Nmap para saber que puertos estan abiertos y son posibles vias de acceso.



Ahora sabemos que tanto el puerto 80 como el 443 están abiertos.

Vemos que es un servicio web que se encuentra en un servidor de Microsoft Azure.

Abrimos msfconsole y buscamos azure:



```
msf6 > search azure

Matching Modules
================

   #   Name                                                Disclosure Date  Rank       Check  Description
   -   ----                                                ---------------  ----       -----  -----------
   0   post/multi/gather/azure_cli_creds                   .                normal     No     Azure CLI Credentials Gatherer
   1   auxiliary/gather/cloud_lookup                       .                normal     No     Cloud Lookup (and Bypass)
   2     \_ action: Amazon CloudFront                      .                .          .      Content Delivery Network services of Amazon
   3     \_ action: ArvanCloud CDN                         .                .          .      ArvanCloud CDN comprises tens of PoP sites in important locations all around the world to deliver online content to the users
   4     \_ action: Automatic                              .                .          .      .
   5     \_ action: AzureCDN                               .                .          .      Microsoft Azure Content Delivery Network (CDN) is a global content distribution network solution for delivering high bandwidth content
   6     \_ action: CloudFlare                             .                .          .      Cloudflare provides SaaS based CDN, WAF, DNS and DDoS mitigation services.
   7     \_ action: Envoy Proxy                            .                .          .      An open source edge and service proxy, designed for Cloud-Native applications
   8     \_ action: Fastly                                 .                .          .      Another widely used CDN/WAF solution
   9     \_ action: Imperva Incapsula                      .                .          .      Cloud based Web application firewall of Imperva
   10    \_ action: InGen Security (BinarySec EasyWAF)     .                .          .      Cloud based Web application firewall of InGen Security and BinarySec
   11    \_ action: KeyCDN                                 .                .          .      KeyCDN is a high performance content delivery network that has been built for the future
   12    \_ action: Netlify                                .                .          .      One workflow, from local development to global deployment
   13    \_ action: NoWAFBypass                            .                .          .      Do NOT check any bypass method
   14    \_ action: Stackpath Fireblade                    .                .          .      Enterprise Website Security & DDoS Protection
   15    \_ action: Stackpath MaxCDN                       .                .          .      Speed Up your Content Delivery
   16    \_ action: Sucuri                                 .                .          .      Cloud based Web application firewall of Sucuri
   17  exploit/windows/http/moveit_cve_2023_34362          2023-05-31       excellent  Yes    MOVEit SQL Injection vulnerability
   18  auxiliary/scanner/http/azure_ad_login               .                normal     No     Microsoft Azure Active Directory Login Enumeration
   19  auxiliary/scanner/http/iis_internal_ip              .                normal     No     Microsoft IIS HTTP Internal IP Disclosure
   20  exploit/linux/local/cve_2021_38648_omigod           2021-09-14       excellent  Yes    Microsoft OMI Management Interface Authentication Bypass
   21    \_ target: Unix Command                           .                .          .      .
   22    \_ target: Linux Dropper                          .                .          .      .
   23  exploit/linux/misc/cve_2021_38647_omigod            2021-09-14       excellent  Yes    Microsoft OMI Management Interface Authentication Bypass
   24    \_ target: Unix Command                           .                .          .      .
   25    \_ target: Linux Dropper                          .                .          .      .
```

Vemos que existen varios tipos de exploits con los que podriamos vulnerar las seguridades de Tomtom.

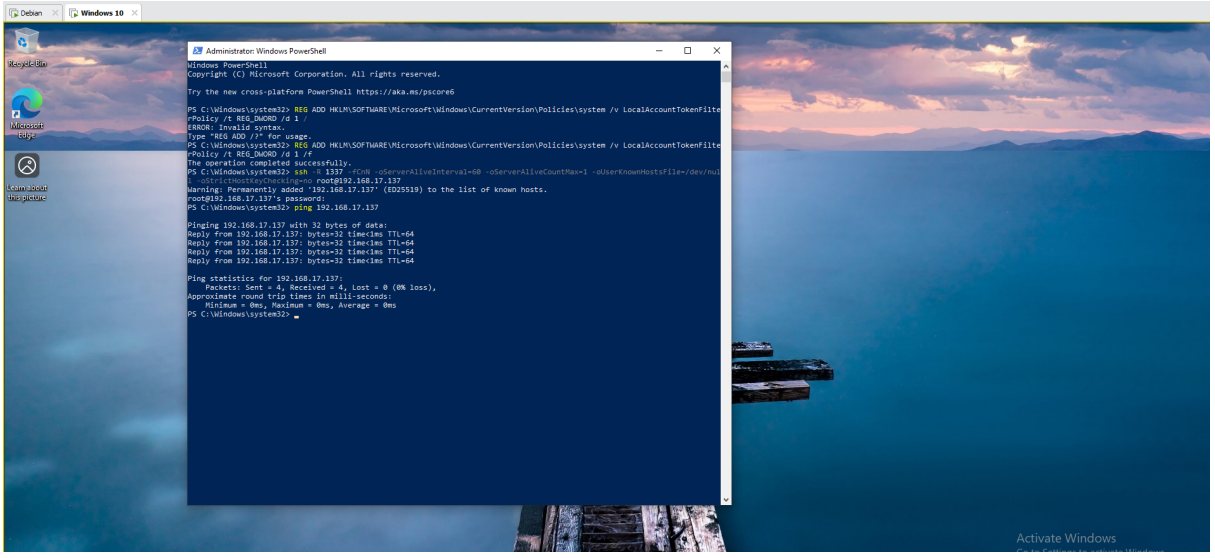
Ejercicio 2: Ejercicio de Red Team

Para realizar este ejercicio de Red Team, vamos a describir paso a paso cómo construir un laboratorio con las dos máquinas (Windows 10 y Linux), instalar un sistema de Command and Control (C&C), infectar la máquina Windows 10.

Una vez tenemos la dos máquinas instaladas y la máquina Windows totalmente actualizada hacemos un ping para comprobar la visibilidad entre ellas.

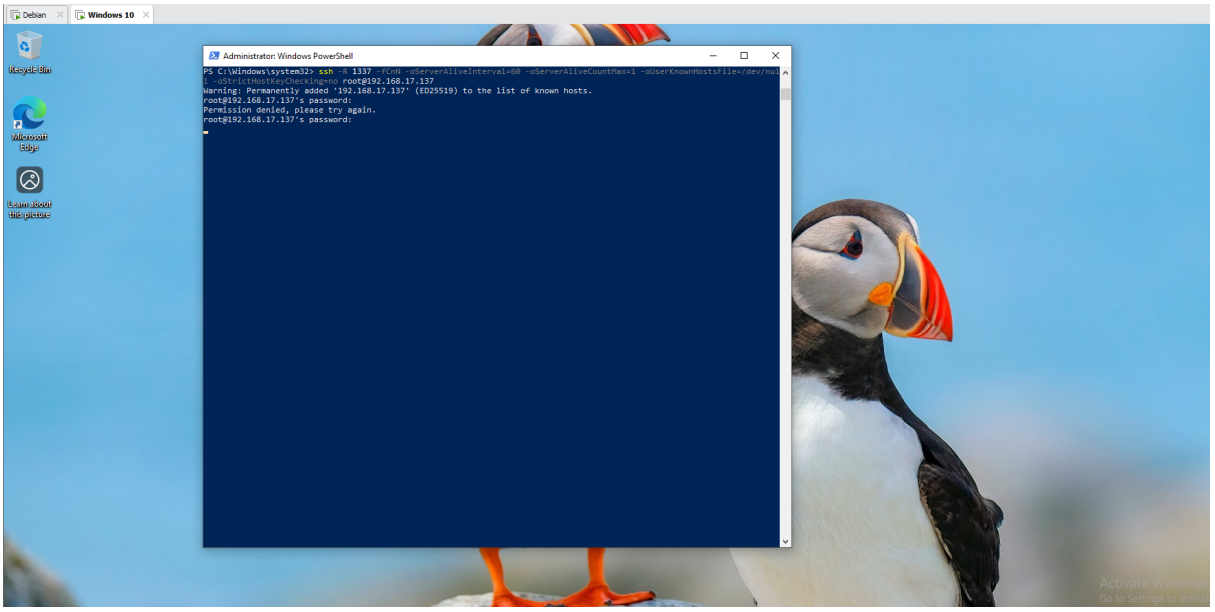
El ping se realiza desde la máquina Windows ya que si lo hacemos desde la Debian el firewall nos bloqueará la petición y no obtendremos respuesta.

```
root@debian:~/Escritorio# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.17.137  netmask 255.255.255.0  broadcast 192.168.17.255
        inet6 fe80::20c:29ff:fef4:20b6  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:f4:20:b6  txqueuelen 1000  (Ethernet)
        RX packets 301  bytes 35081 (34.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 9007 (8.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17295  bytes 3848412 (3.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17295  bytes 3848412 (3.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@debian:~/Escritorio#
```

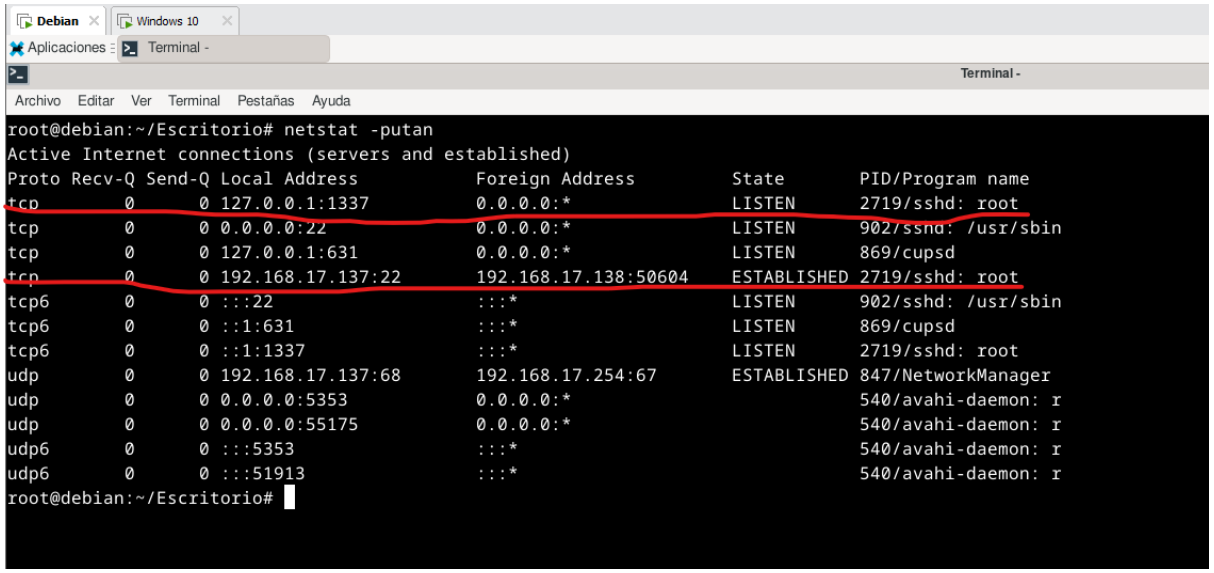


Antes de nada editaremos el archivo sshd_config, permitiremos el **PermintRootLogin yes**, habilitaremos la directiva **AllowTcpForwarding yes** y reiniciamos el servicio **sshd.**

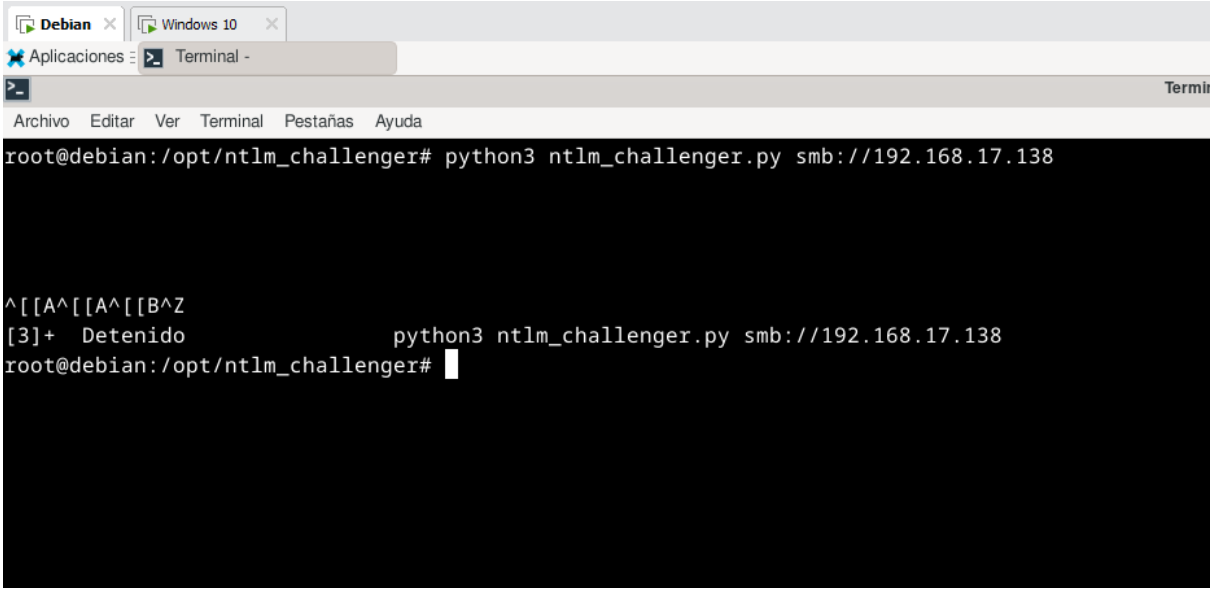
Primero abriremos un túnel ssh desde la máquina victima (Windows) hacia nosotros (Debian).



Haciendo un netstat -putan podemos ver que tenemos el túnel abierto con el puerto 1337 en escucha. También podemos ver que tenemos una conexión establecida con la máquina victima (192.168.17.138).

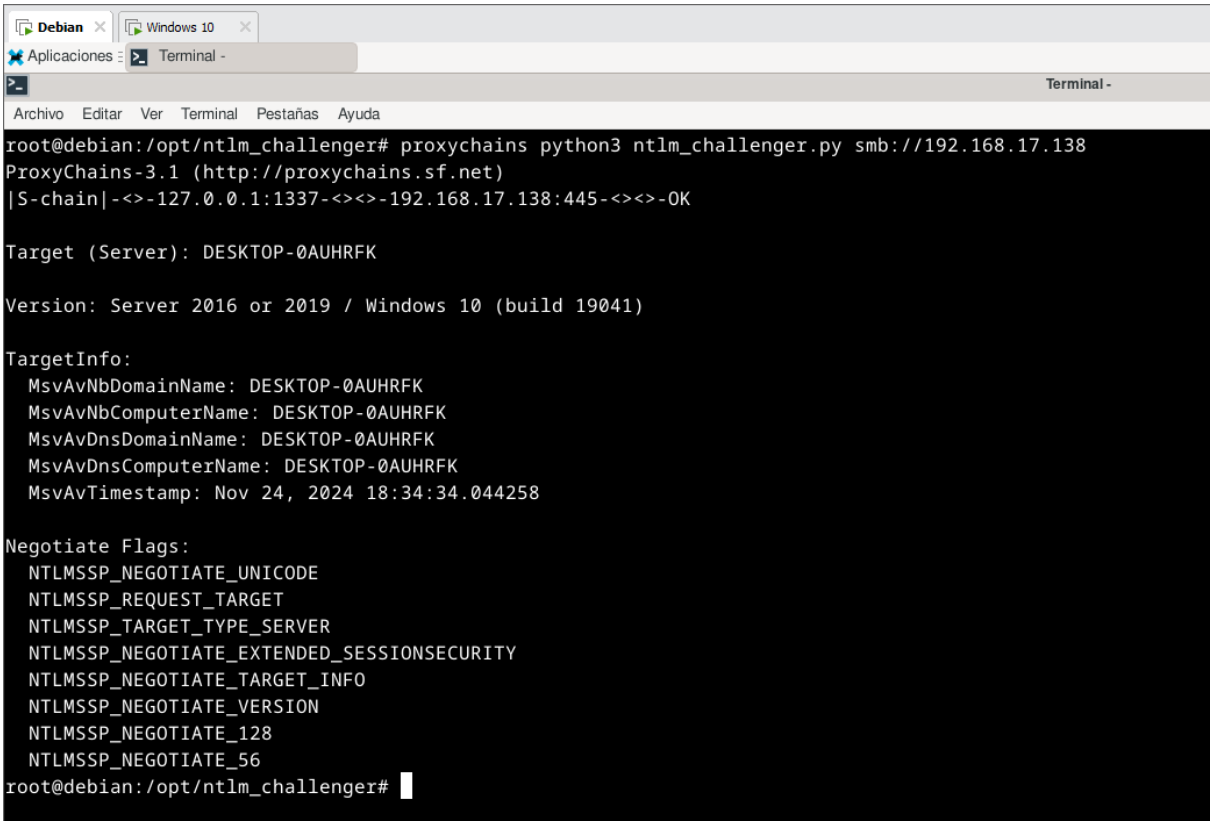


Ahora intentamos conectarnos con `ntlm_challenger` pero vemos que no funciona ya que el `firewall` nos bloquea:

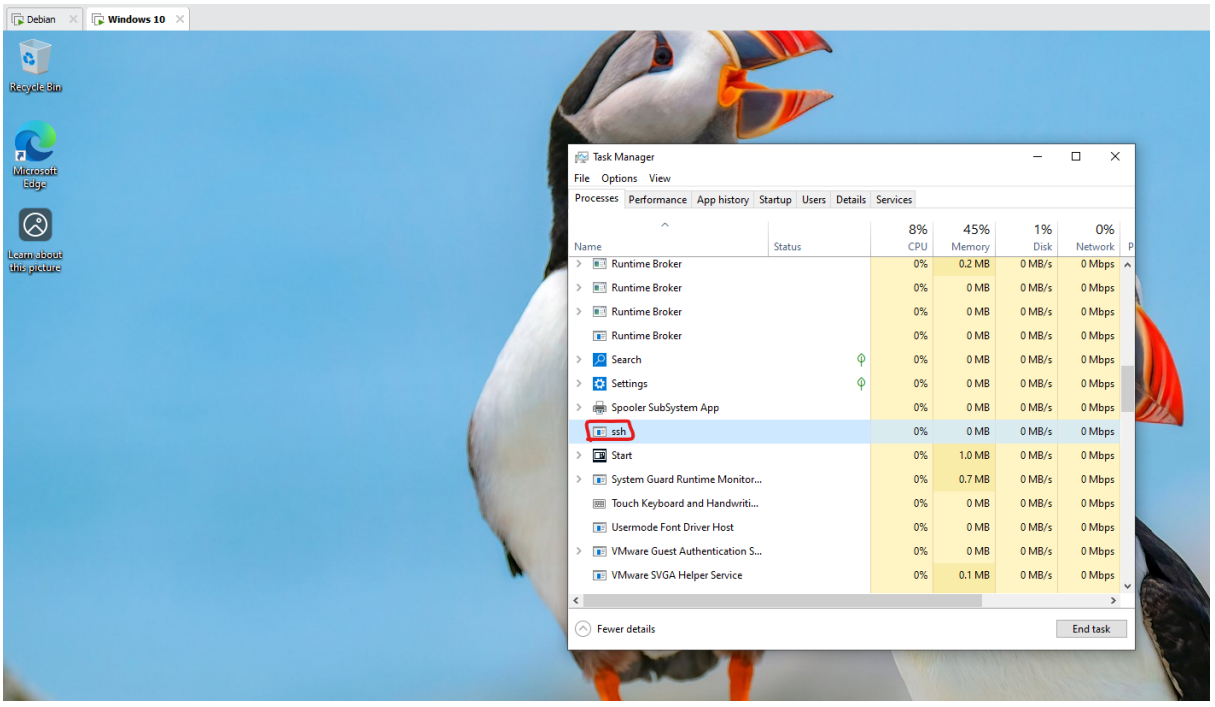


Para evitar esto utilizamos `proxychains` , así hacemos bypass al firewall de Windows.

Y ahora funciona a la perfección:

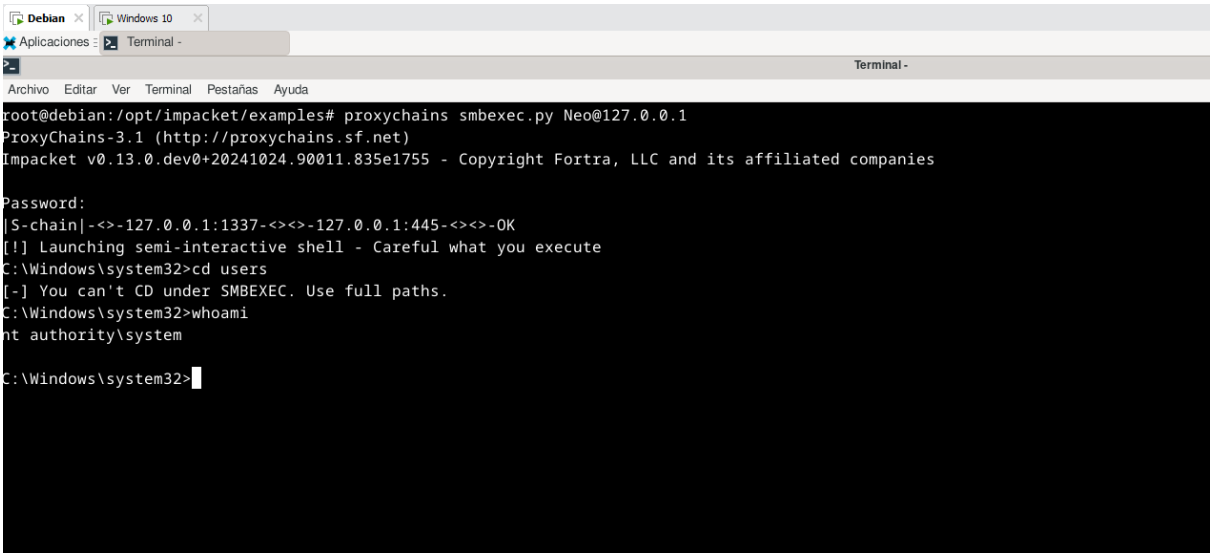


Si volvemos a Windows podemos ver que aunque esté el Powershell cerrado se queda el proceso ssh abierto en segundo plano.

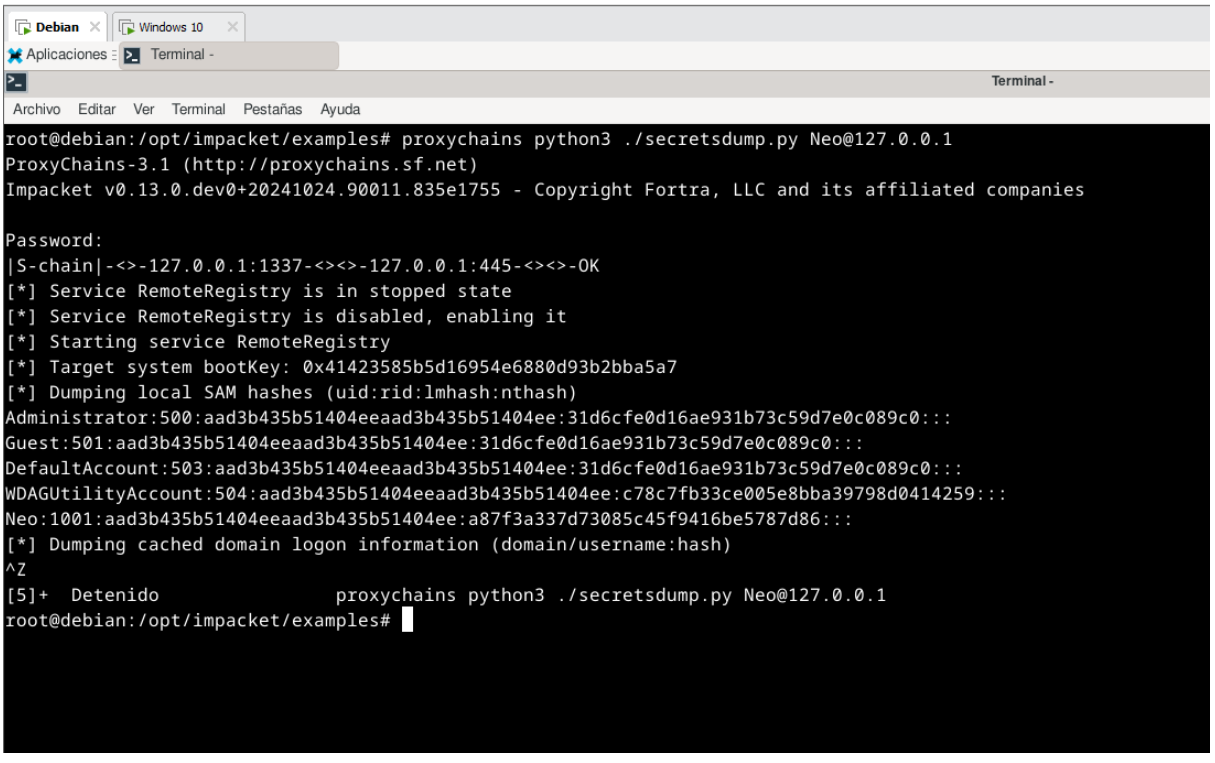


Ahora utilizando este mismo túnel y script de python de impacket smbexec.py nos conecta una shell directa al disco C de Windows.

Al hacer whoami vemos que no solo somos administrador sino que somos nt authority/system que es el usuario con más privilegios que existe.



Si utilizamos el script secretsdump.py podemos encontrar los hashes de las contraseñas de todos los usuarios (incluido administrador) de la máquina.



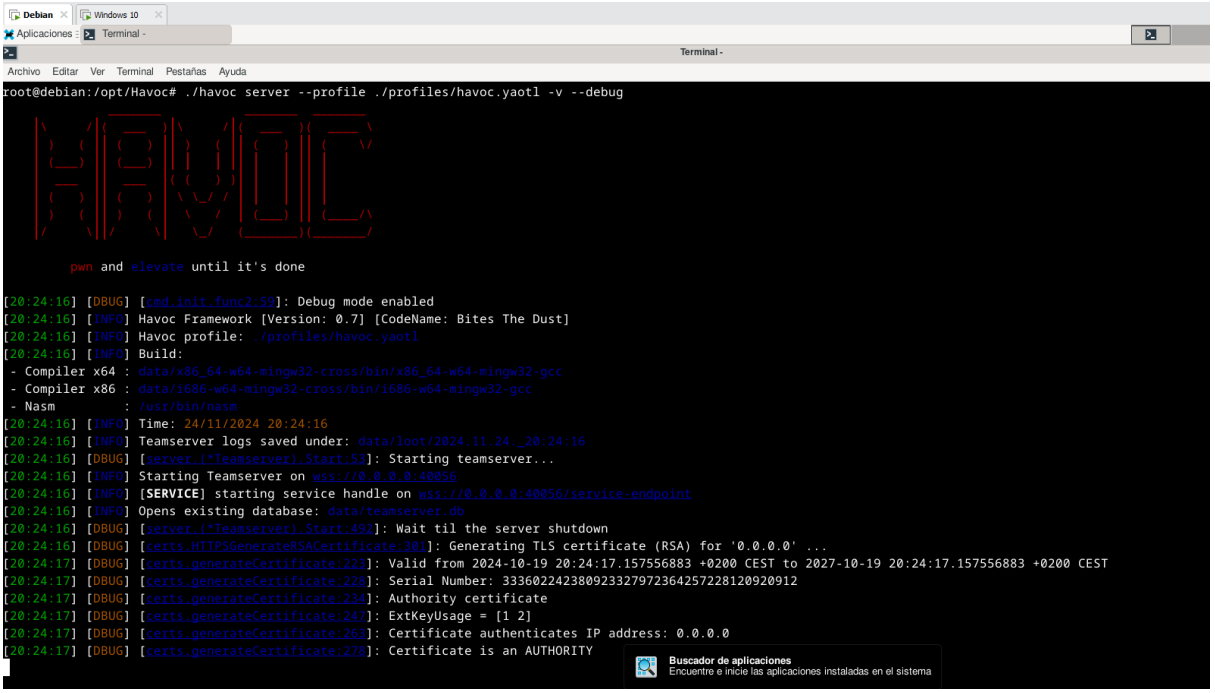
Ahora utilizamos Havoc:

Havoc es una herramienta de **Command and Control (C&C)** que se utiliza en pruebas de penetración y simulaciones de ataques de **Red Team**. Havoc es especialmente conocida por su capacidad para ejecutar comandos en máquinas comprometidas de manera muy eficiente, permitiendo a los pentesters y a los atacantes éticos ejecutar diversas actividades maliciosas o simuladas de forma controlada.

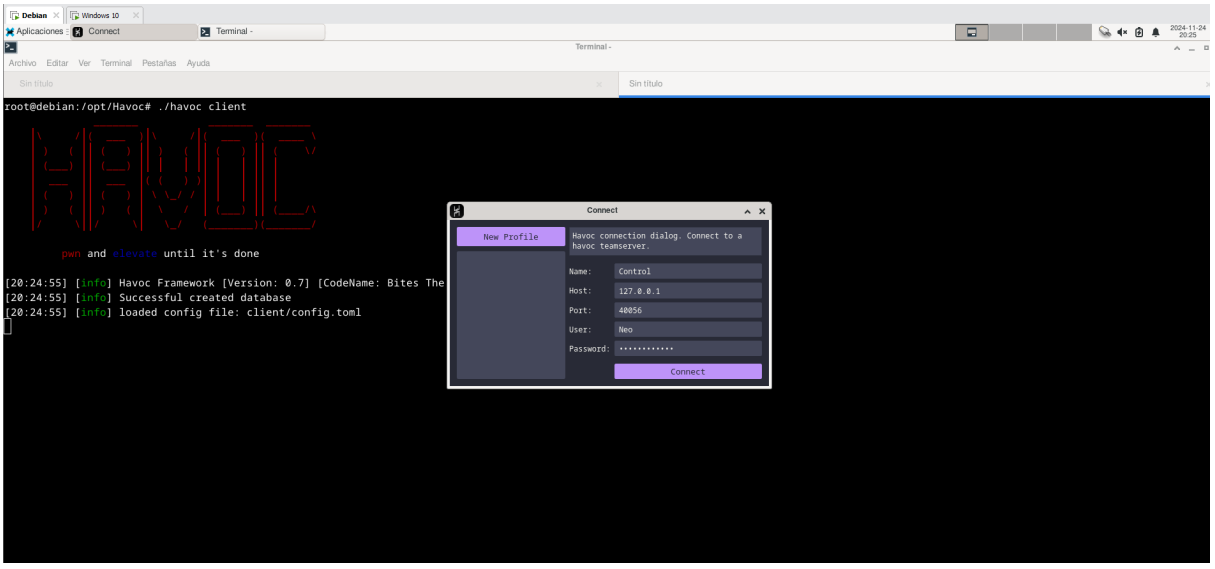
En un escenario de prueba de penetración, Havoc funciona típicamente de la siguiente manera:

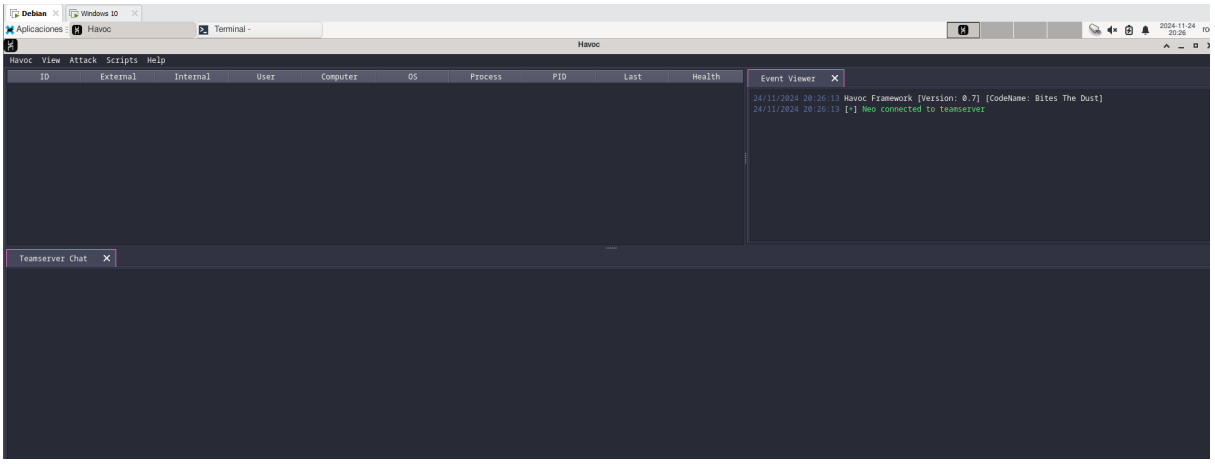
1. **Instalación del agente**:
   Un "payload" o agente malicioso se instala en la máquina objetivo. Esto podría hacerse de diversas maneras, como a través de ingeniería social o explotación de vulnerabilidades.
2. **Conexión al servidor C&C**:
   Una vez que el payload está instalado, establece una conexión con el servidor de comando y control (C&C). Este servidor es donde el atacante puede emitir comandos que serán ejecutados en la máquina comprometida.
3. **Ejecución de comandos**:
   A través de la interfaz de Havoc, el atacante puede enviar comandos a la máquina comprometida. Estos pueden incluir comandos para robar datos, ejecutar scripts, acceder a recursos de la red, etc.
4. **Módulos adicionales**:
   Como mencioné, la herramienta es modular, por lo que el atacante puede agregar o modificar funcionalidades específicas según sus necesidades.

Primero iniciamos el servidor de Havoc:

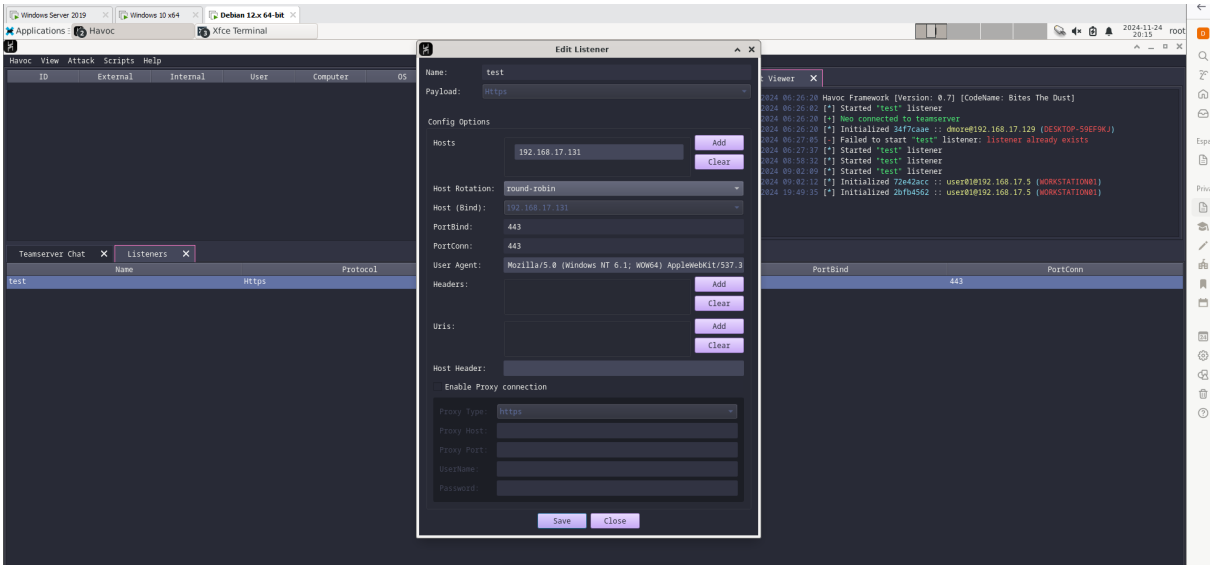


Y nos conectamos a él:

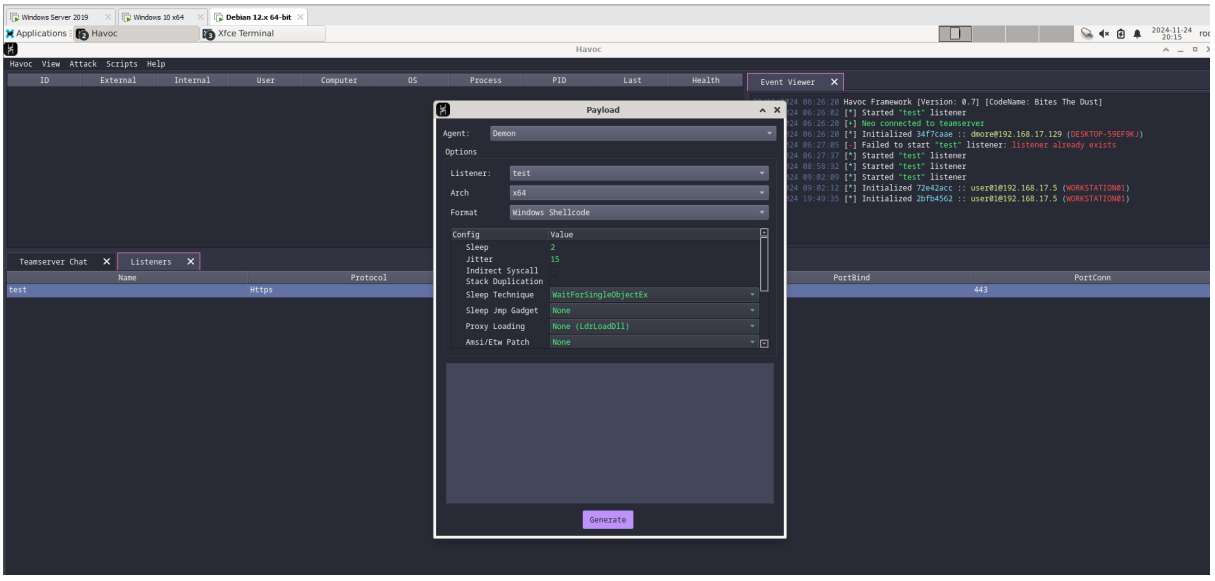




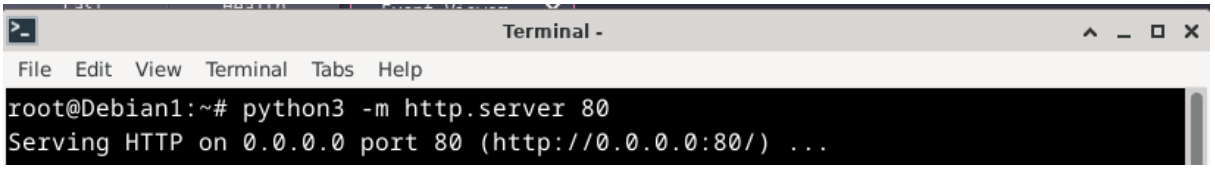
Creamos un listener en el puerto 443:



Ahora compilamos un payload que se ejecutará en la máquina objetivo y nos dará una especie de reverse shell:

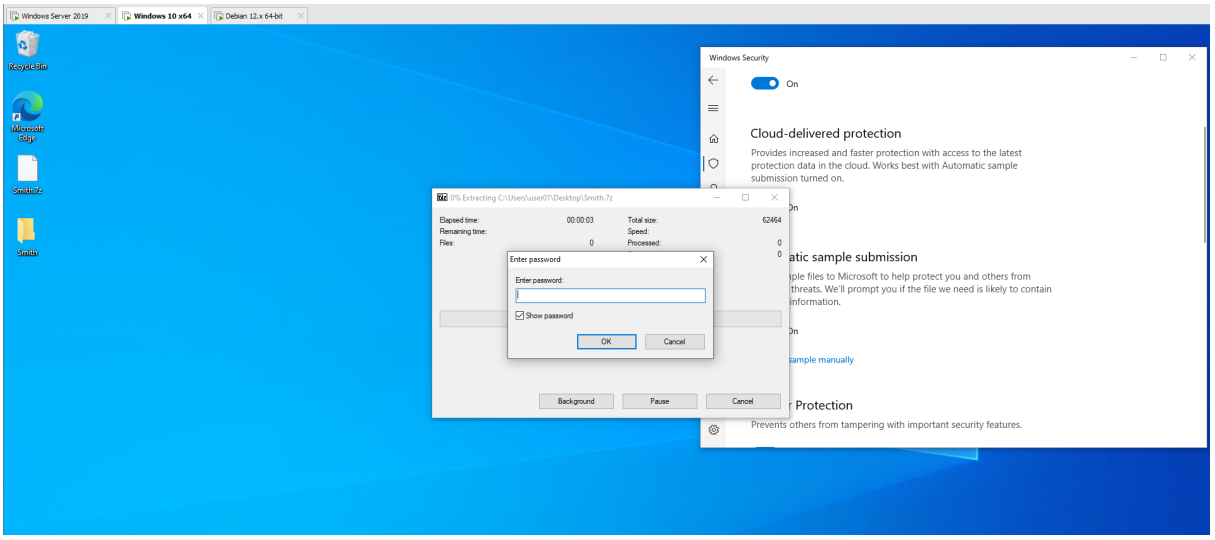


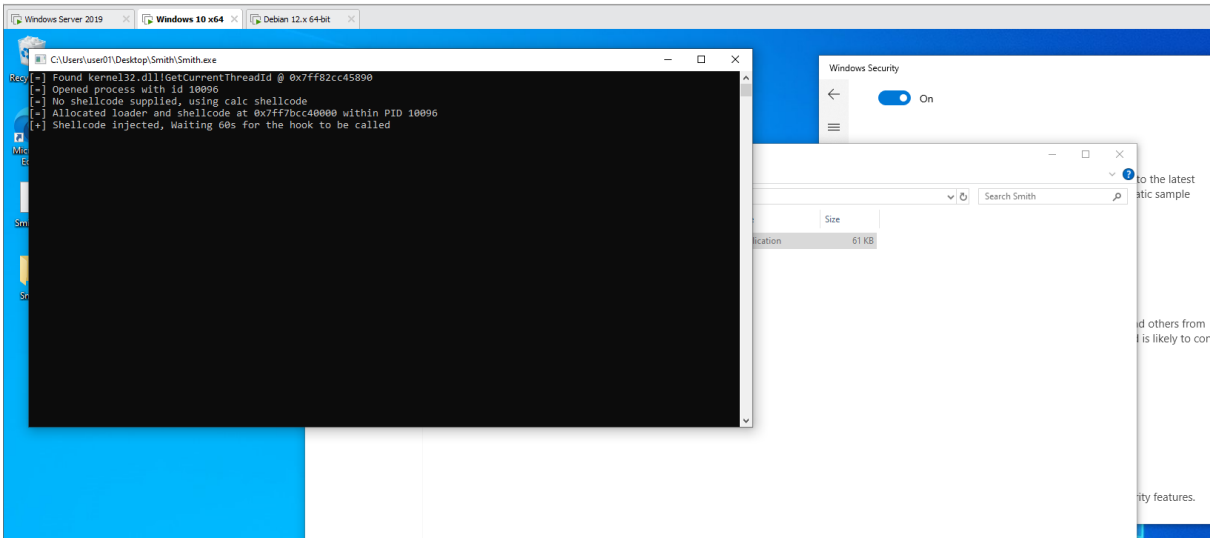
Iniciamos un servidor http desde el que se podrá descargar el payload.



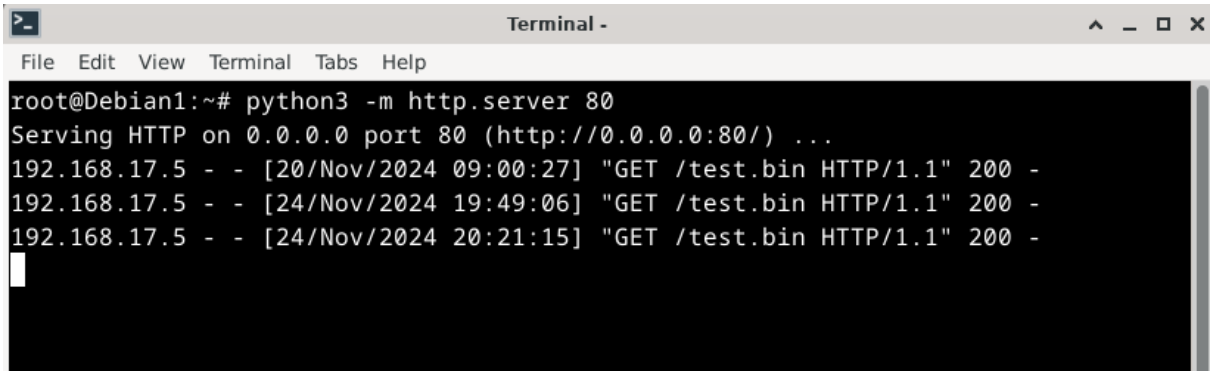
Mandamos el AgenteSmith a la máquina objetivo ya se mediante phishing u otros metodos.

Este script una vez alguien lo ejecute descargará el payload desde el servidor que hemos montado en Debian.





Una vez ejecutado vemos la petición de descarga en la consola.



Cuando termine de descargar el payload la ventana se cerrará y quedará conectado a nuestro Havoc.

Desde la consola de Havoc ahora podemos hacer peticiones desde la shell y ya estamos dentro de la máquina sin que el antivirus nos detecte.