Havoc C2: First look

By Nee

Published: 2022-10-05 · Archived: 2026-04-05 21:27:27 UTC

Havoc is a modern and malleable post-exploitation command and control framework, created by @C5pider. I first came into contact with Havoc C2 in April 2022 when it was still a private tool under development. C5pider went on Flangvik's stream to discuss about development in general and demoed this awesome tool. Back in May it was announced that Havoc would be released in about 3-5 months and here we are!

I'm gona deploying this into my infra and playing around with it in this post! Been wanting to test out the Sleep Obfuscation implementation on the Demon for a while now.

Sidenote: You'll notice a lot of similarities between Havoc and Cobalt Strike and that's not necessarily a downside IMO!

Prerequisites

- Debian-Based Host (C2 Server)
- Debian-Based Host (C2 Client)
- Target Host (Windows 7/10/11)

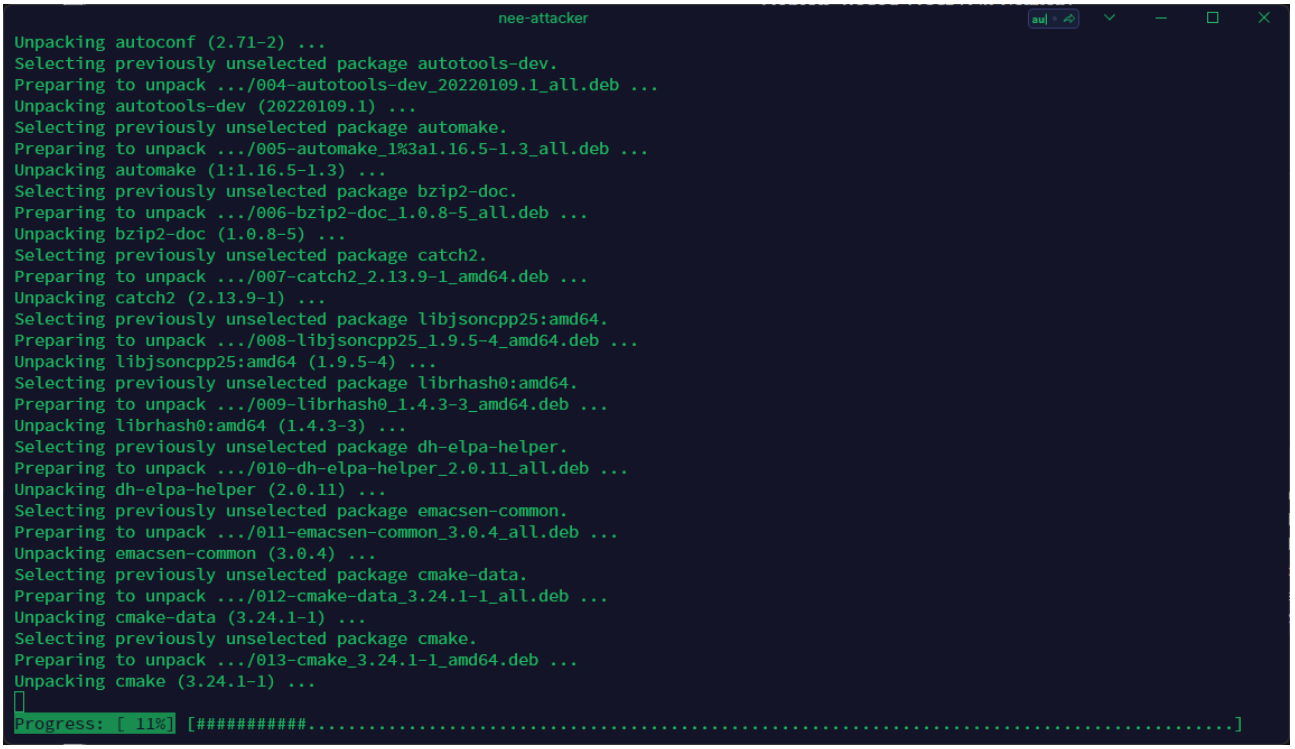
Setup & Installation

(C2 Server)

Installation

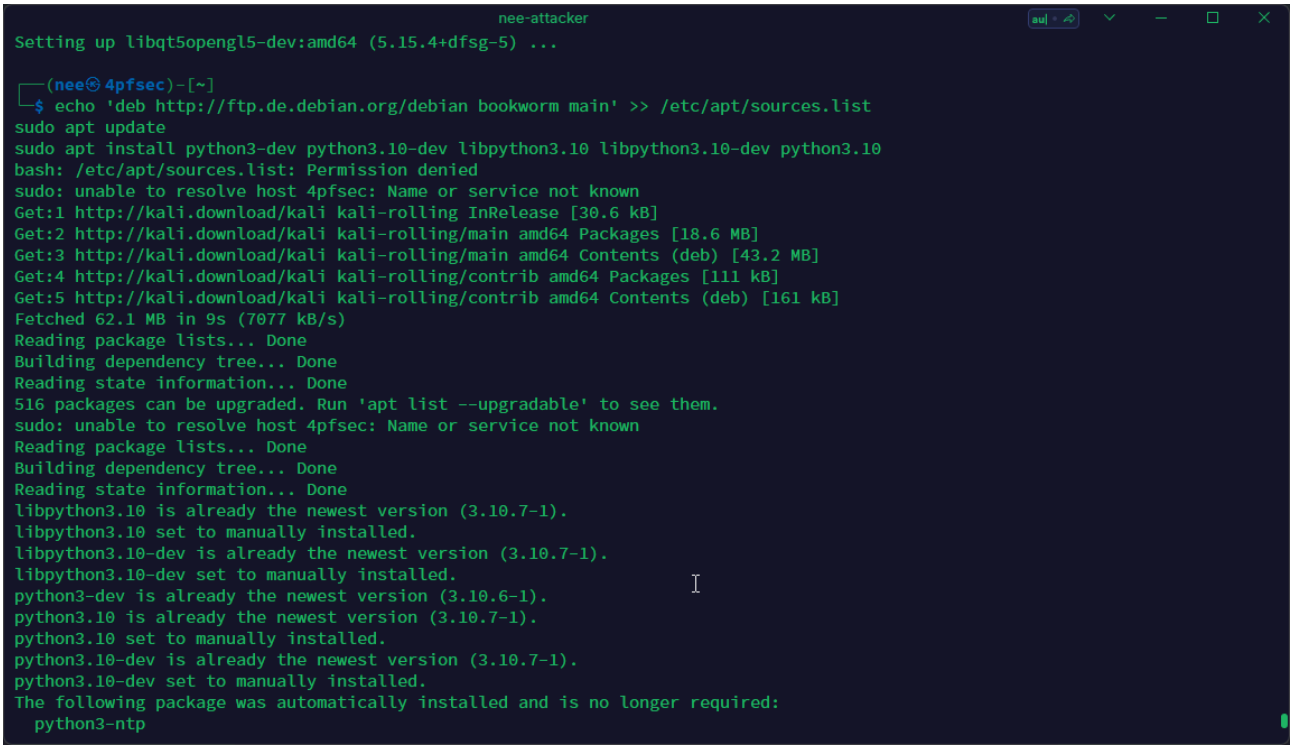
Prerequisites Packages

```
┌──(nee㉿4pfsec)-[~]
└─$ sudo apt install -y git build-essential apt-utils cmake libfontconfig1 libglu1-mesa-dev libgtest-dev libsp
```



Setting up the `bookworm` repo for Python 3.10.

```
┌──(nee㉿4pfsec)-[~]
└─$ echo 'deb http://ftp.de.debian.org/debian bookworm main' >> /etc/apt/sources.list sudo apt update sudo apt
```



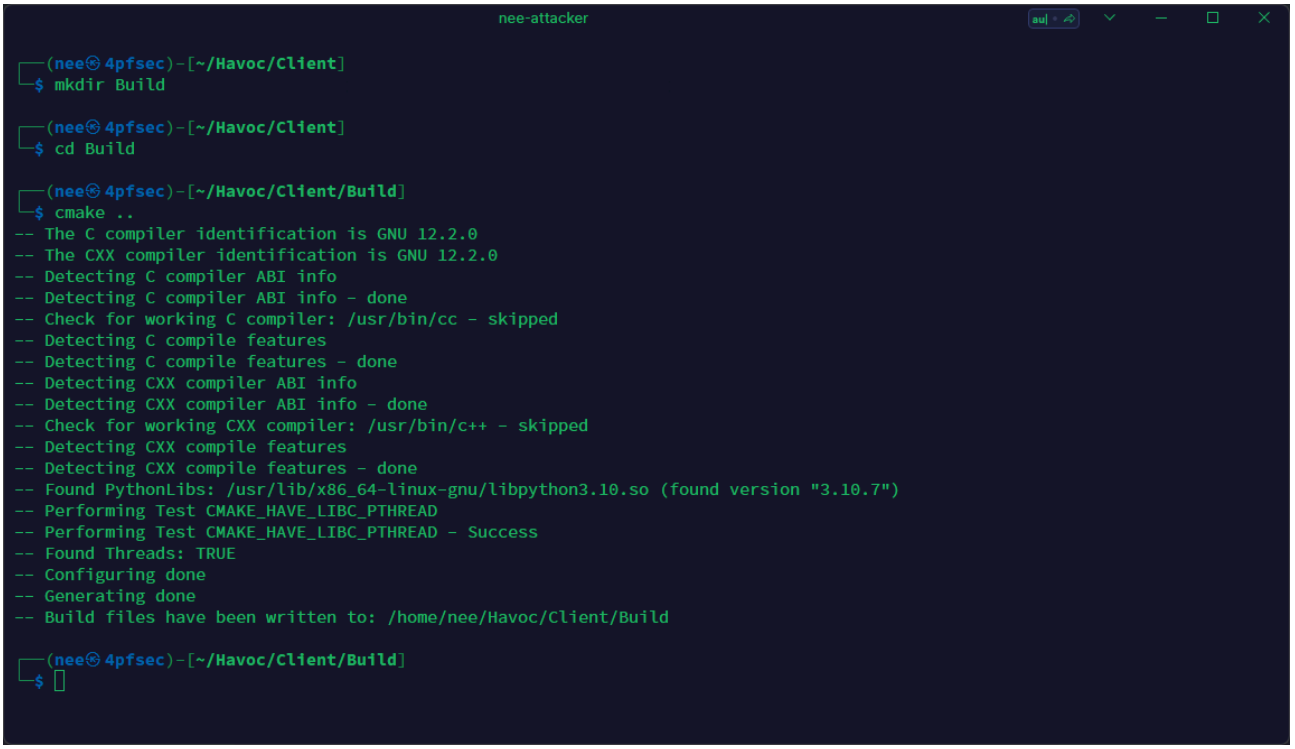
Setup

Git Clone

```
┌──(nee㉿4pfsec)-[~]
└─$ git clone https://github.com/HavocFramework/Havoc.git
```

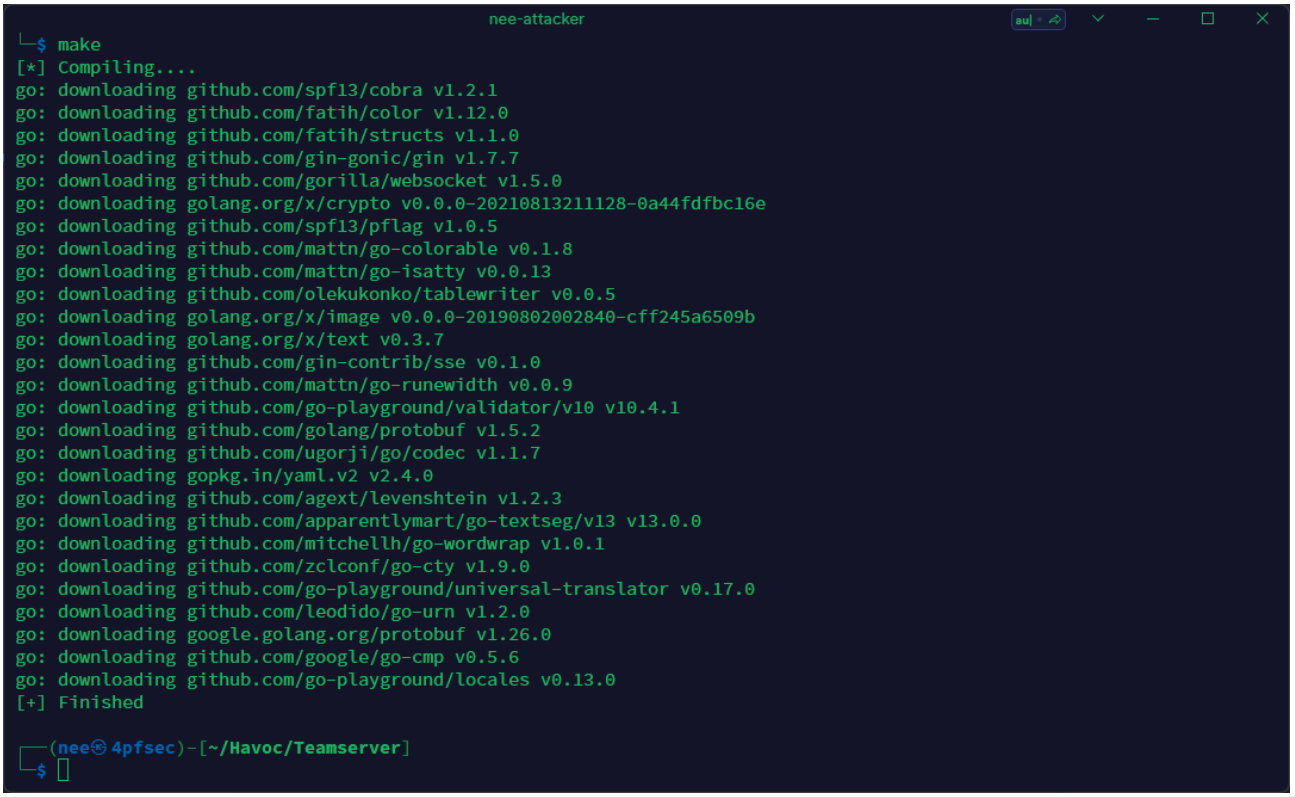
Building the Client

cd Havoc/Client mkdir Build cd Build cmake .. cd .. ./Install.sh

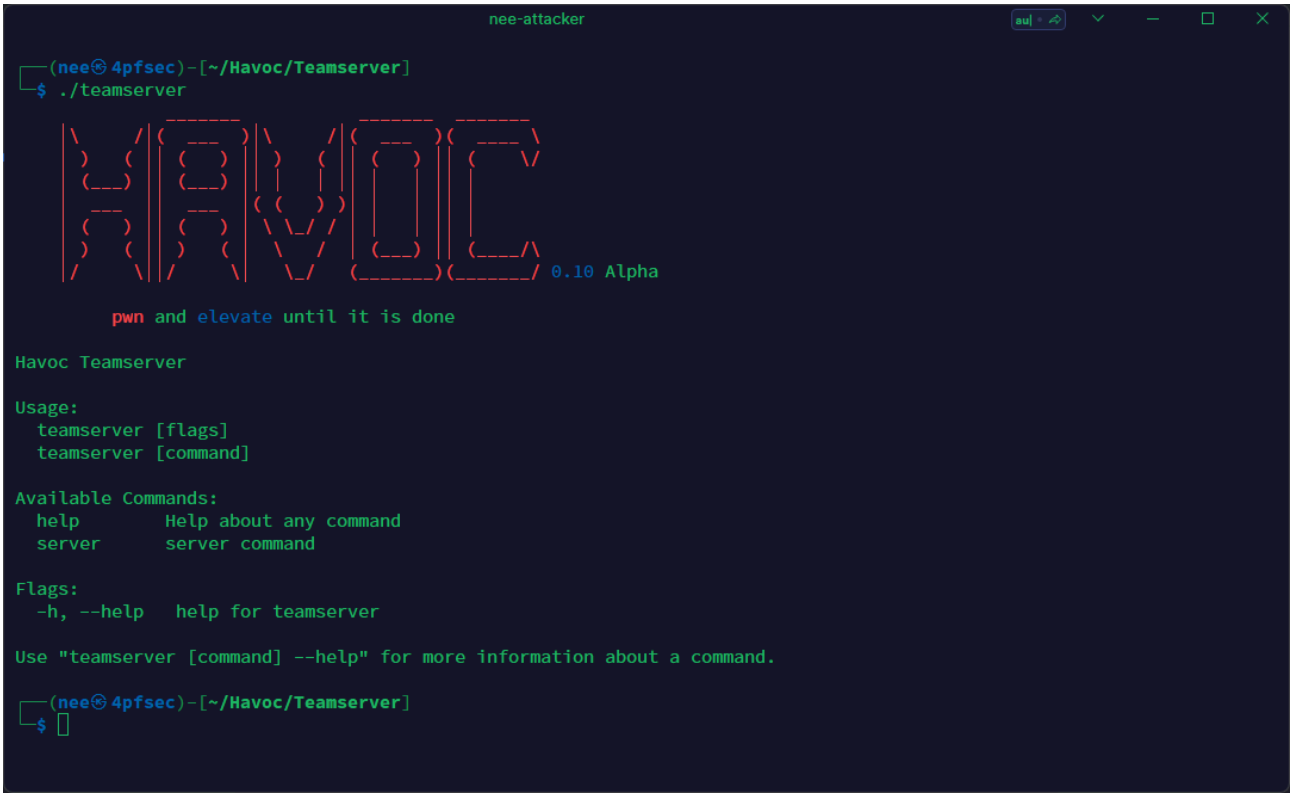


Building the Teamserver

cd Havoc/Teamserver go mod download golang.org/x/sys
go mod download github.com/ugorji/go



┌──(nee㉿4pfsec)-[~/Havoc/Teamserver] └─$ ./teamserver



With that, Havoc is installed and ready to go!

Havoc Framework

The C2 consists of 2 main parts. The client and the team server. Let's start off with the Teamserver.

Teamserver

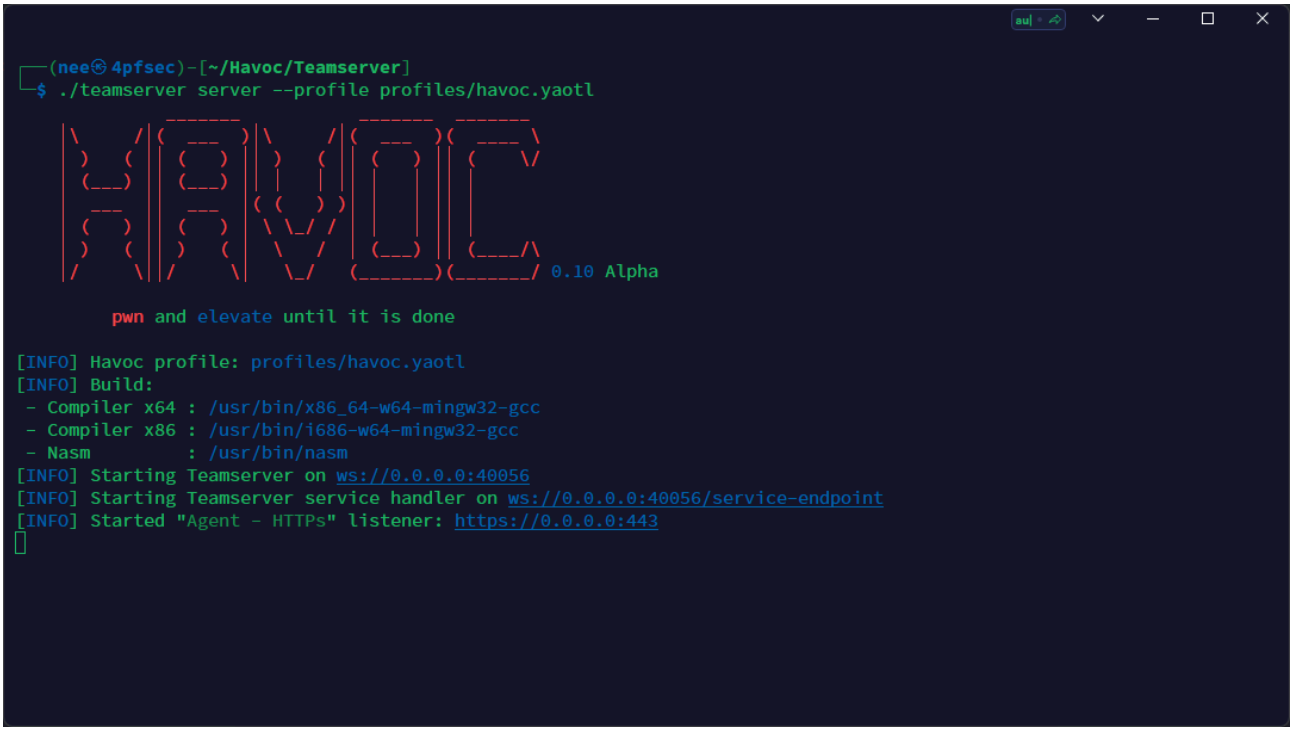
The teamserver allows us to specify a profile or use the default one. The profile allows us to edit configs of the following domains:

- Teamserver
- Operator
- Listener
- Service
- Payload

The default profile is located at `Havoc/Teamserver/profiles`

Running the teamserver with a profile

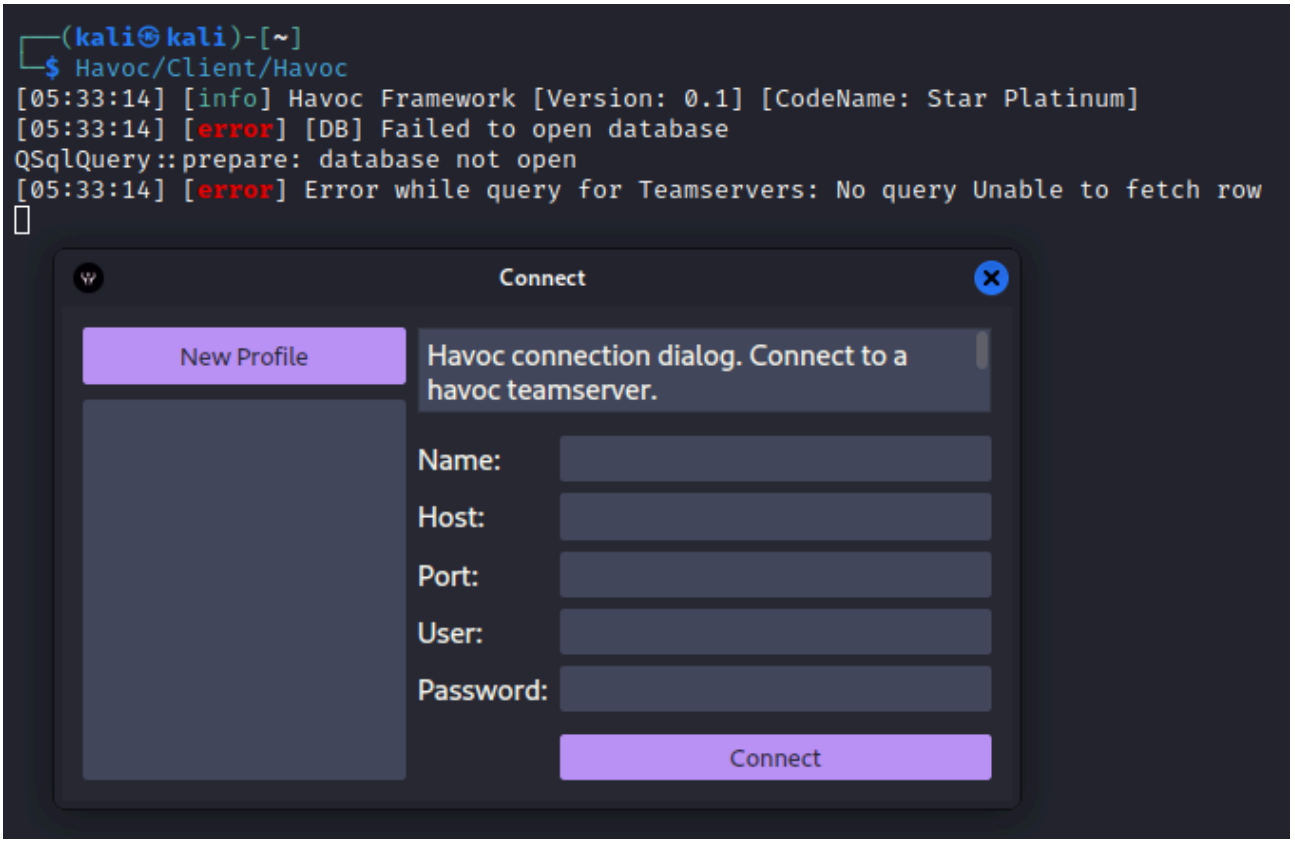
```
┌──(nee㉿4pfsec)-[~/Havoc/Teamserver]
└─$ ./teamserver server --profile profiles/havoc.yaotl
```



Client

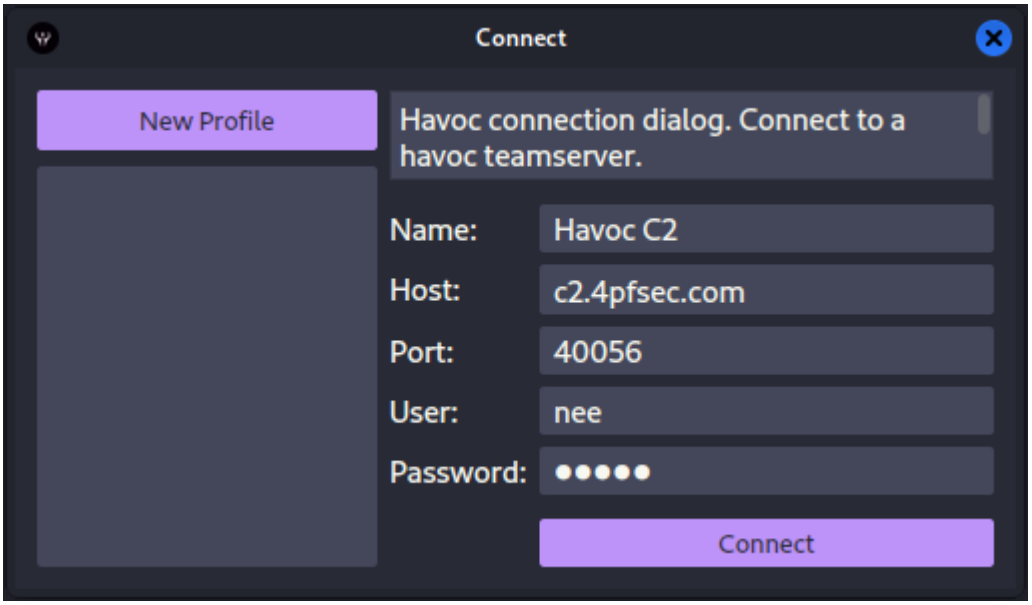
Running the Client

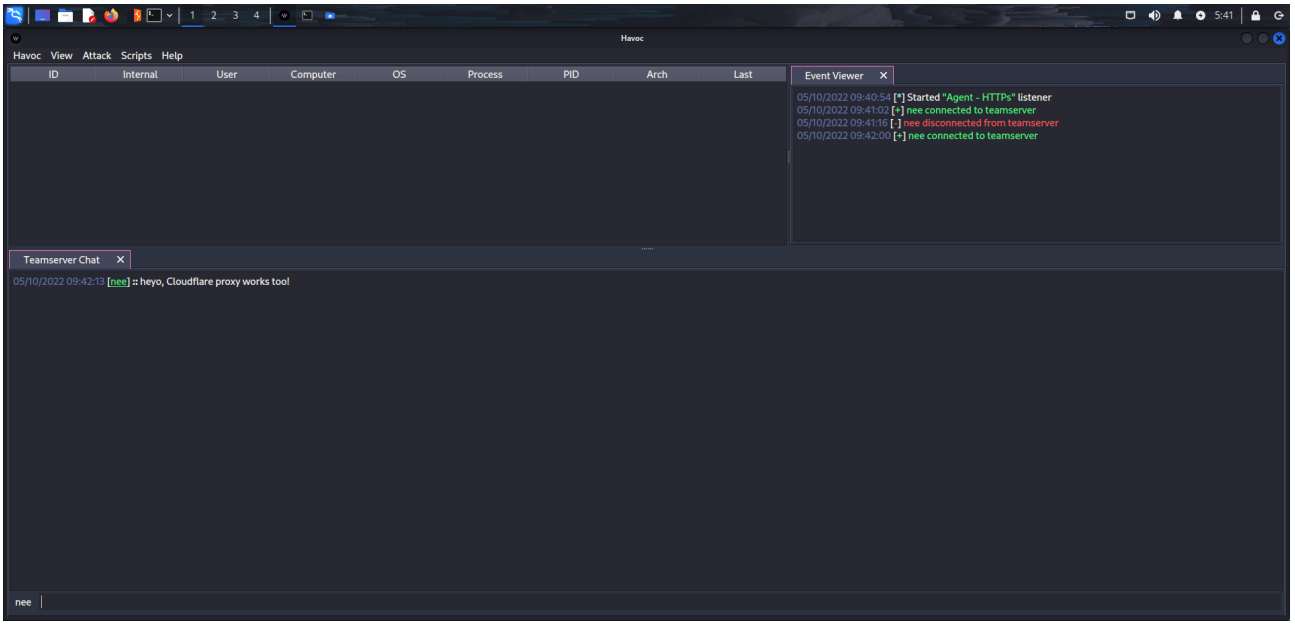
```
┌──(kali㉿kali)-[~]
└─$ Havoc/Client/Havoc
```



Connecting to the teamserver

- Name
- C2 Host
- C2 port
- C2 User:Password

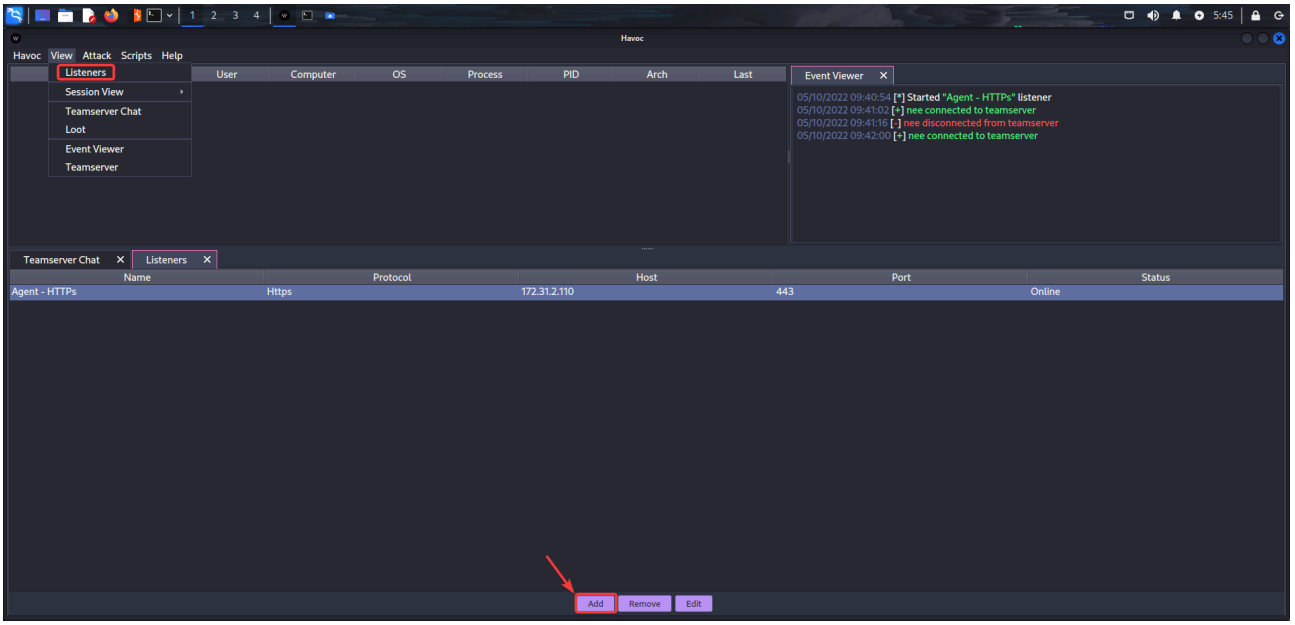


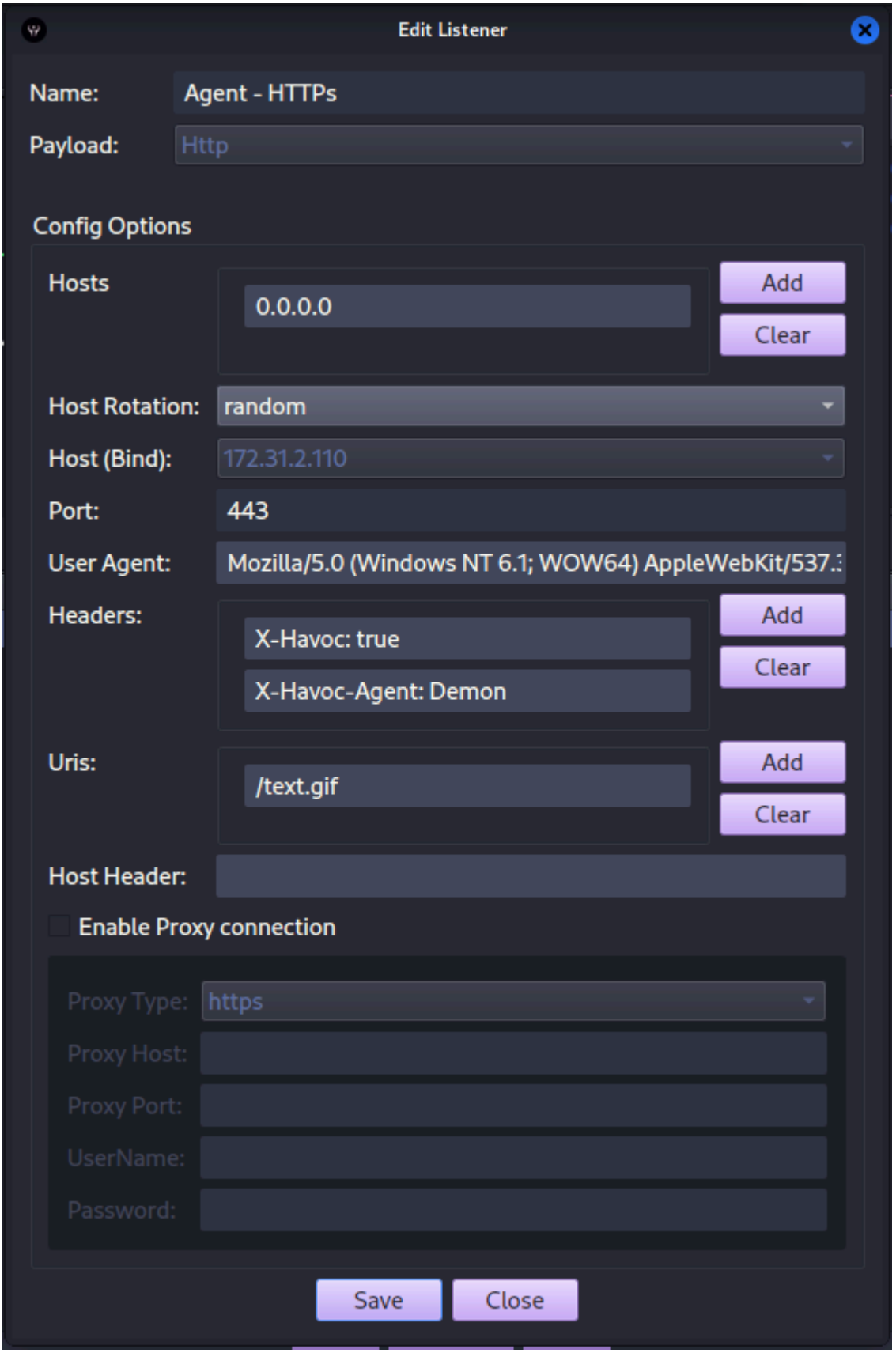


And we're in! The Dracula theme on the client looks really good. Let's check out some of the functionalities!

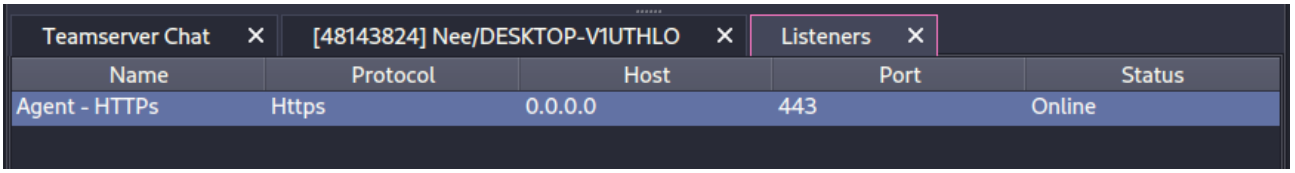
Configuring Listeners

View->Listeners->Add


Havoc
Havoc View Attack Scripts Help
Listeners
Session View
Teamserver Chat
Loot
Event Viewer
Teamserver
User Computer OS Process PID Arch Last
Event Viewer
05/10/2022 09:40:54 [*] Started "Agent - HTTPs" listener
05/10/2022 09:41:02 [+] nee connected to teamserver
05/10/2022 09:41:16 [-] nee disconnected from teamserver
05/10/2022 09:42:00 [+] nee connected to teamserver
Teamserver Chat
Listeners
Name Protocol Host Port Status
Agent - HTTPs Https 172.31.2.110 443 Online
Add
Remove
Edit



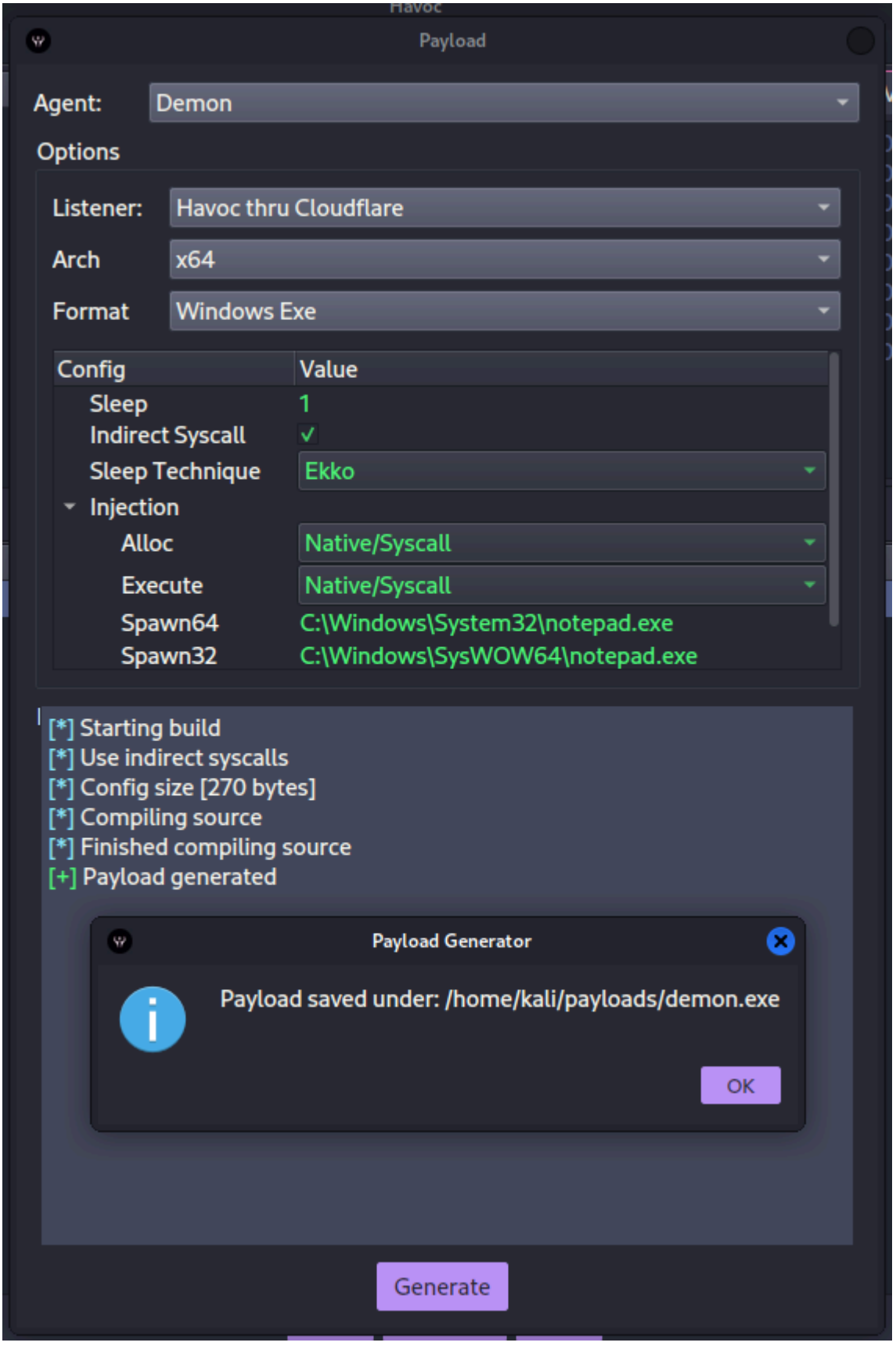
Let's configure our listener and point the host to `c2.4pfsec.com`. This is the domain proxied through Cloudflare.



Generating Payload (UNDETECTED BY Windows Defender)

As of writing, the payload is not detected by Microsoft Defender. (05/10/22)

Attack->Payload->Generate



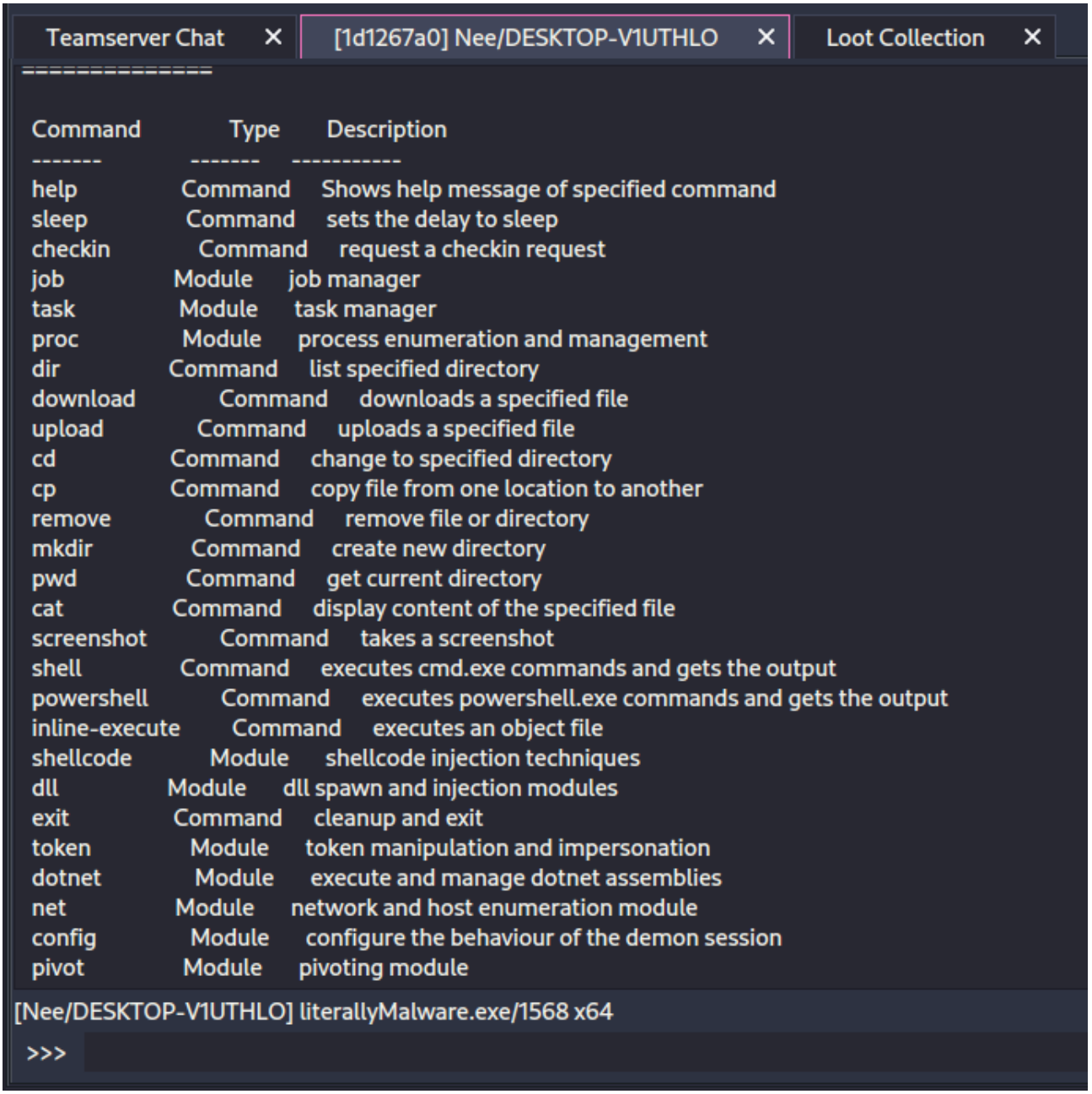
Callback to C2 (UNDETECTED BY Windows Defender)

As of writing, the callback method is not picked up by Microsoft Defender. (05/10/22)

Now that we have our payload, lets deliver and execute it. [You're free to use any delivery method]

I simply hosted an SMB share and transferred the payload to the target. As shown in the demo below, I was able to get a call back from a fully patched Windows 11 Pro Machine using the generated payload.

Interacting with Target

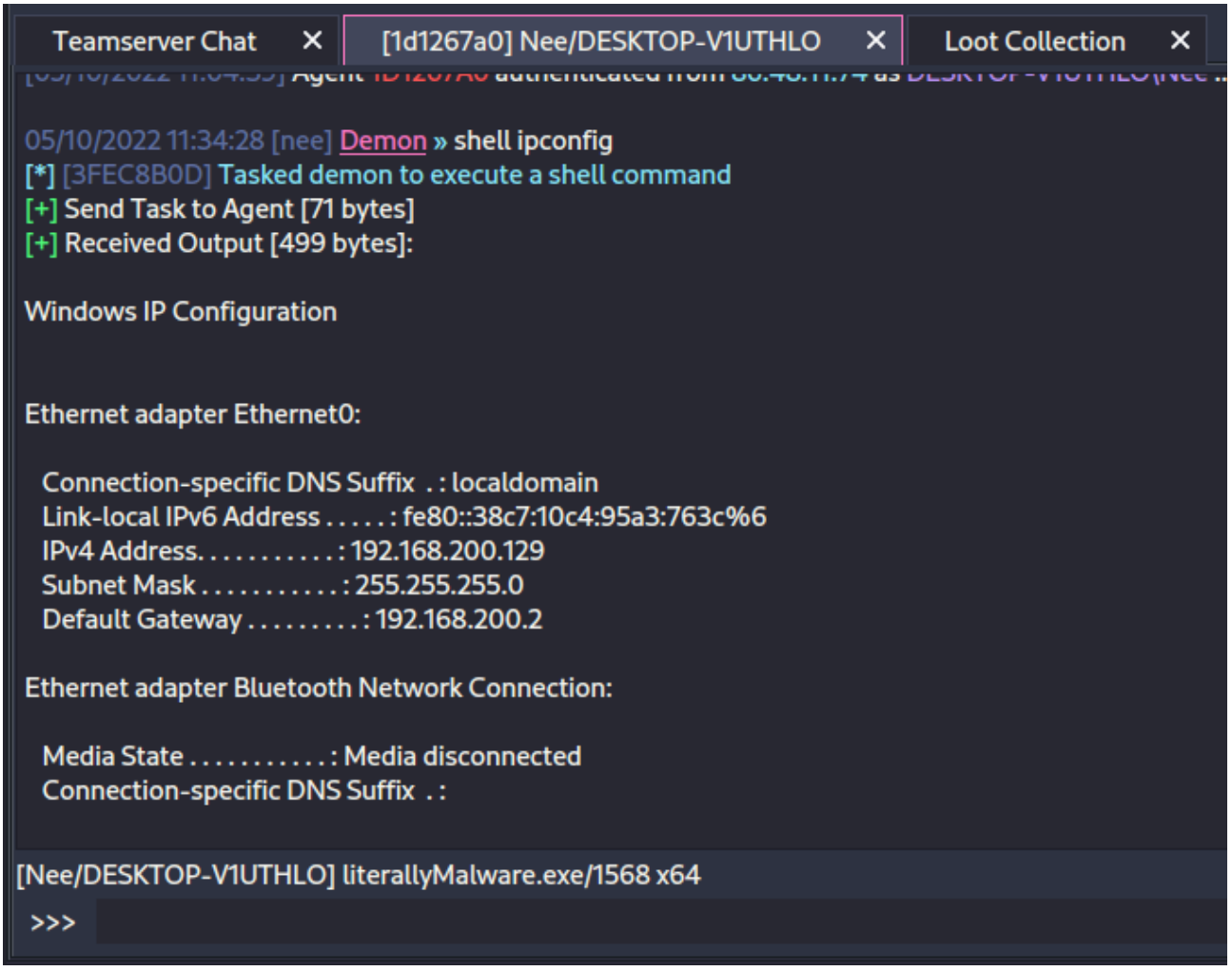


There's a whole list of commands that you're able to run on the target once it calls back to your C2. The target will fetch the C2 for jobs based on the given sleep duration during payload generation.

shell

You're able to run shell commands directly on the target with the help of Havoc

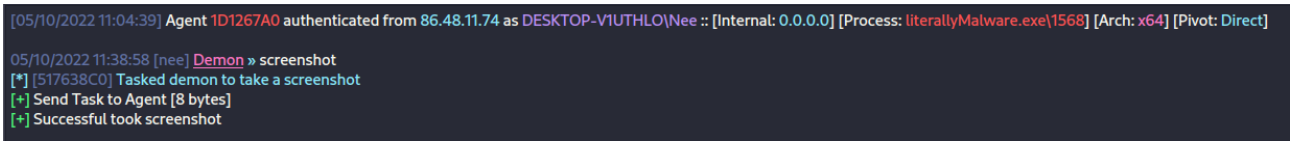
\>>> shell [command]



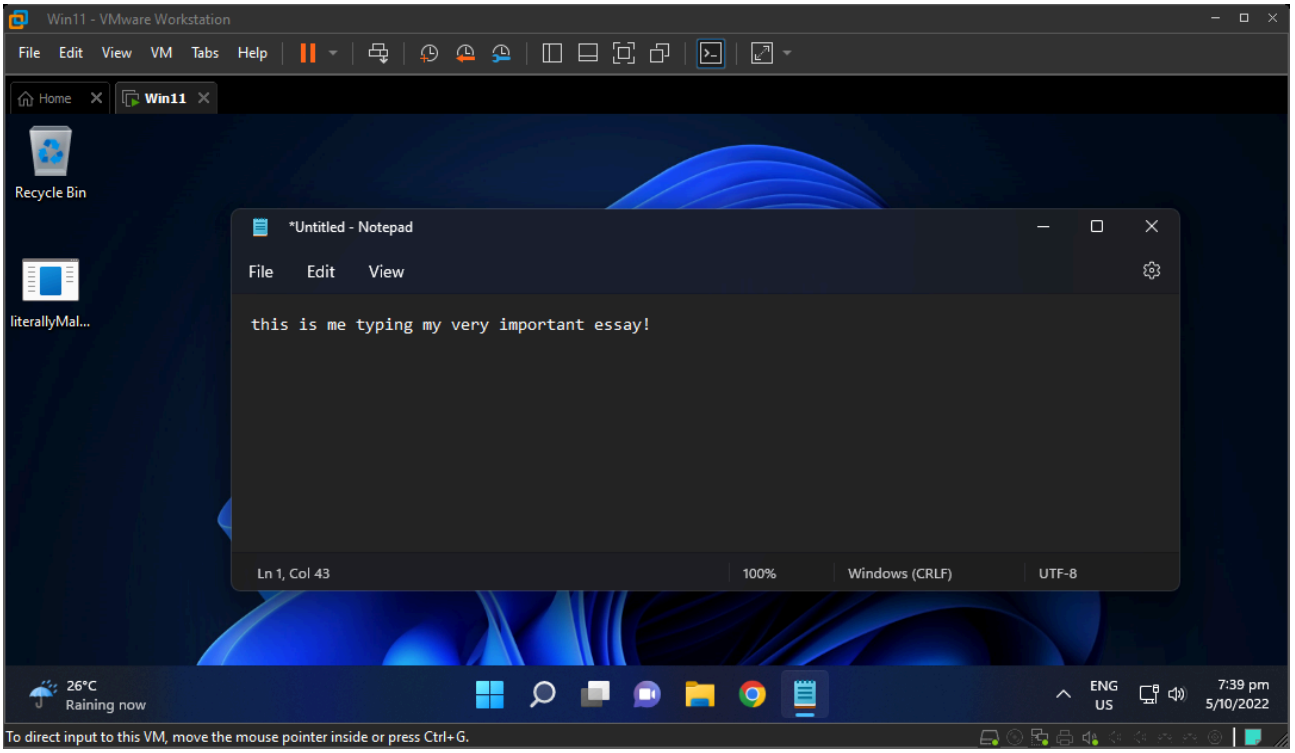
Screenshot

The screenshot command takes a snapshot of the target's desktop and send it back to the C2.

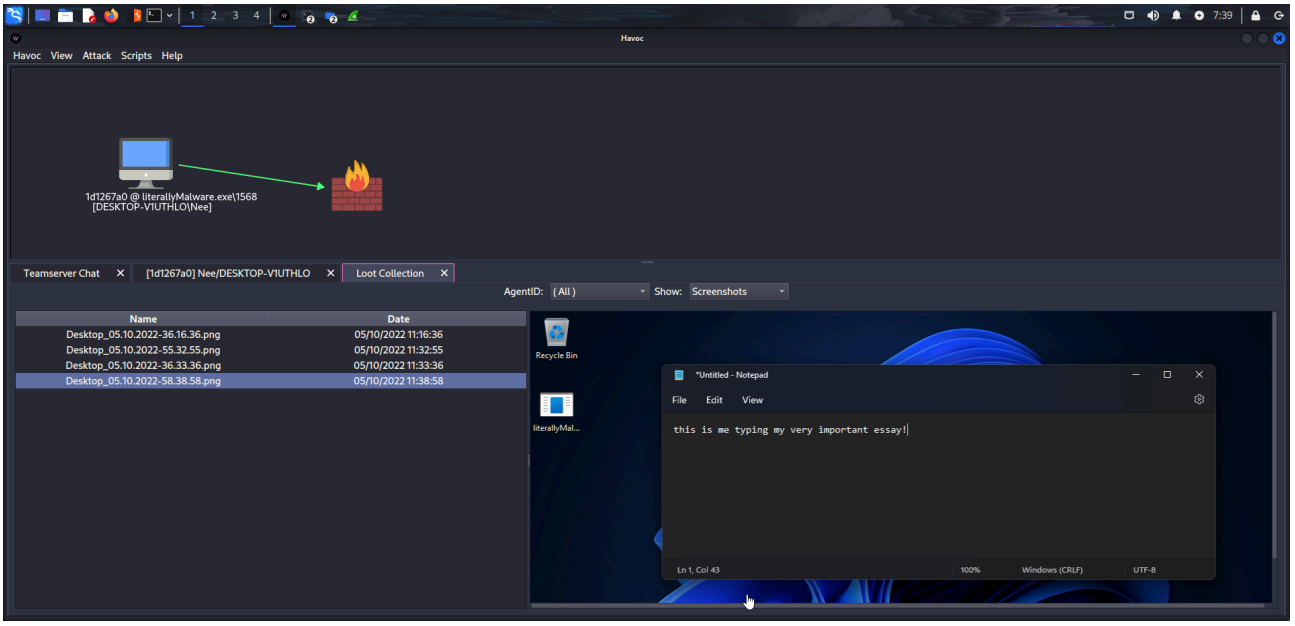
\>>> screenshot



Seen on Host



Seen on C2



These were just some of Post exploitation offered by Havoc.

Havoc looks to have great potential and I hope to continue this series by exploring the C2 in-depth real soon!

Source: https://4pfsec.com/havoc-c2-first-look/