

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
```

```

nee-attacker
Unpacking autoconf (2.71-2) ...
Selecting previously unselected package autotools-dev.
Preparing to unpack .../004-autotools-dev_20220109.1_all.deb ...
Unpacking autotools-dev (20220109.1) ...
Selecting previously unselected package automake.
Preparing to unpack .../005-automake_1%3a1.16.5-1.3_all.deb ...
Unpacking automake (1:1.16.5-1.3) ...
Selecting previously unselected package bzip2-doc.
Preparing to unpack .../006-bzip2-doc_1.0.8-5_all.deb ...
Unpacking bzip2-doc (1.0.8-5) ...
Selecting previously unselected package catch2.
Preparing to unpack .../007-catch2_2.13.9-1_amd64.deb ...
Unpacking catch2 (2.13.9-1) ...
Selecting previously unselected package libjsoncpp25:amd64.
Preparing to unpack .../008-libjsoncpp25_1.9.5-4_amd64.deb ...
Unpacking libjsoncpp25:amd64 (1.9.5-4) ...
Selecting previously unselected package librhash0:amd64.
Preparing to unpack .../009-librhash0_1.4.3-3_amd64.deb ...
Unpacking librhash0:amd64 (1.4.3-3) ...
Selecting previously unselected package dh-elpa-helper.
Preparing to unpack .../010-dh-elpa-helper_2.0.11_all.deb ...
Unpacking dh-elpa-helper (2.0.11) ...
Selecting previously unselected package emacs-common.
Preparing to unpack .../011-emacs-common_3.0.4_all.deb ...
Unpacking emacs-common (3.0.4) ...
Selecting previously unselected package cmake-data.
Preparing to unpack .../012-cmake-data_3.24.1-1_all.deb ...
Unpacking cmake-data (3.24.1-1) ...
Selecting previously unselected package cmake.
Preparing to unpack .../013-cmake_3.24.1-1_amd64.deb ...
Unpacking cmake (3.24.1-1) ...
Progress: [ 11%] [#####.....]

```

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
```

```

nee-attacker
Setting up libqt5opengl5-dev:amd64 (5.15.4+dfsg-5) ...

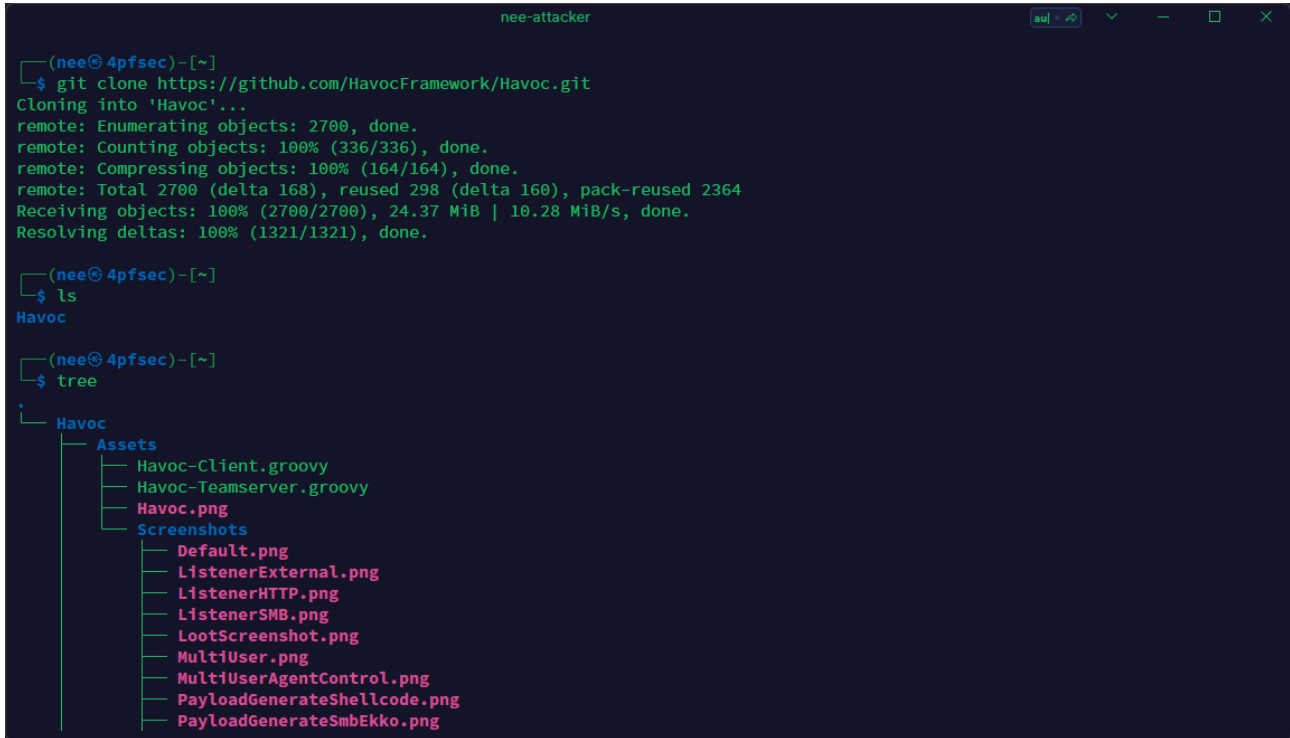
(nee@4pfsec)-[~]
$ echo 'deb http://ftp.de.debian.org/debian bookworm main' >> /etc/apt/sources.list
sudo apt update
sudo apt install python3-dev python3.10-dev libpython3.10 libpython3.10-dev python3.10
bash: /etc/apt/sources.list: Permission denied
sudo: unable to resolve host 4pfsec: Name or service not known
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [18.6 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [43.2 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [111 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [161 kB]
Fetched 62.1 MB in 9s (7077 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
516 packages can be upgraded. Run 'apt list --upgradable' to see them.
sudo: unable to resolve host 4pfsec: Name or service not known
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libpython3.10 is already the newest version (3.10.7-1).
libpython3.10 set to manually installed.
libpython3.10-dev is already the newest version (3.10.7-1).
libpython3.10-dev set to manually installed.
python3-dev is already the newest version (3.10.6-1).
python3.10 is already the newest version (3.10.7-1).
python3.10 set to manually installed.
python3.10-dev is already the newest version (3.10.7-1).
python3.10-dev set to manually installed.
The following package was automatically installed and is no longer required:
python3-ntp

```

Setup

Git Clone

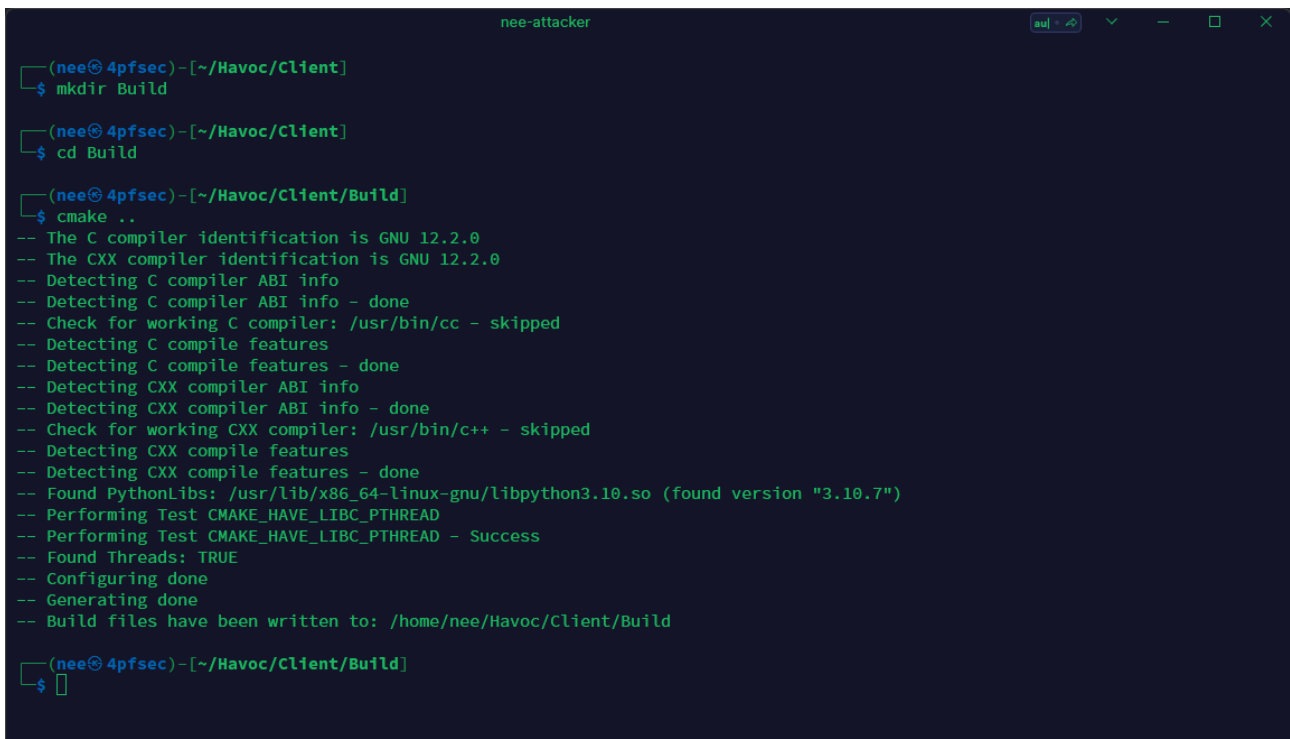
```
(nee@4pfsec)-[~]  
└─$ git clone https://github.com/HavocFramework/Havoc.git
```



```
nee-attacker  
  
(nee@4pfsec)-[~]  
└─$ git clone https://github.com/HavocFramework/Havoc.git  
Cloning into 'Havoc'...  
remote: Enumerating objects: 2700, done.  
remote: Counting objects: 100% (336/336), done.  
remote: Compressing objects: 100% (164/164), done.  
remote: Total 2700 (delta 168), reused 298 (delta 160), pack-reused 2364  
Receiving objects: 100% (2700/2700), 24.37 MiB | 10.28 MiB/s, done.  
Resolving deltas: 100% (1321/1321), done.  
  
(nee@4pfsec)-[~]  
└─$ ls  
Havoc  
  
(nee@4pfsec)-[~]  
└─$ tree  
.  
├── Havoc  
│   ├── Assets  
│   │   ├── Havoc-Client.groovy  
│   │   ├── Havoc-Teamserver.groovy  
│   │   ├── Havoc.png  
│   │   └── Screenshots  
│   │       ├── Default.png  
│   │       ├── ListenerExternal.png  
│   │       ├── ListenerHTTP.png  
│   │       ├── ListenerSMB.png  
│   │       ├── LootScreenshot.png  
│   │       ├── MultiUser.png  
│   │       ├── MultiUserAgentControl.png  
│   │       ├── PayloadGenerateShellcode.png  
│   │       └── PayloadGenerateSmbEkko.png
```

Building the Client

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



```
nee-attacker  
  
(nee@4pfsec)-[~/Havoc/Client]  
└─$ mkdir Build  
  
(nee@4pfsec)-[~/Havoc/Client]  
└─$ cd Build  
  
(nee@4pfsec)-[~/Havoc/Client/Build]  
└─$ cmake ..  
-- The C compiler identification is GNU 12.2.0  
-- The CXX compiler identification is GNU 12.2.0  
-- Detecting C compiler ABI info  
-- Detecting C compiler ABI info - done  
-- Check for working C compiler: /usr/bin/cc - skipped  
-- Detecting C compile features  
-- Detecting C compile features - done  
-- Detecting CXX compiler ABI info  
-- Detecting CXX compiler ABI info - done  
-- Check for working CXX compiler: /usr/bin/c++ - skipped  
-- Detecting CXX compile features  
-- Detecting CXX compile features - done  
-- Found PythonLibs: /usr/lib/x86_64-linux-gnu/libpython3.10.so (found version "3.10.7")  
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD  
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success  
-- Found Threads: TRUE  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /home/nee/Havoc/Client/Build  
  
(nee@4pfsec)-[~/Havoc/Client/Build]  
└─$
```

Building the Teamserver

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

```
nee-attacker  
└─$ make  
[*] Compiling...  
go: downloading github.com/spf13/cobra v1.2.1  
go: downloading github.com/fatih/color v1.12.0  
go: downloading github.com/fatih/structs v1.1.0  
go: downloading github.com/gin-gonic/gin v1.7.7  
go: downloading github.com/gorilla/websocket v1.5.0  
go: downloading golang.org/x/crypto v0.0.0-20210813211128-0a44dfbc16e  
go: downloading github.com/spf13/pflag v1.0.5  
go: downloading github.com/mattn/go-colorable v0.1.8  
go: downloading github.com/mattn/go-isatty v0.0.13  
go: downloading github.com/olekukonko/tablewriter v0.0.5  
go: downloading golang.org/x/image v0.0.0-20190802002840-cff245a6509b  
go: downloading golang.org/x/text v0.3.7  
go: downloading github.com/gin-contrib/sse v0.1.0  
go: downloading github.com/mattn/go-runewidth v0.0.9  
go: downloading github.com/go-playground/validator/v10 v10.4.1  
go: downloading github.com/golang/protobuf v1.5.2  
go: downloading github.com/ugorji/go/codec v1.1.7  
go: downloading gopkg.in/yaml.v2 v2.4.0  
go: downloading github.com/agext/levenshtein v1.2.3  
go: downloading github.com/apparentlymart/go-textseg/v13 v13.0.0  
go: downloading github.com/mitchellh/go-wordwrap v1.0.1  
go: downloading github.com/zclconf/go-cty v1.9.0  
go: downloading github.com/go-playground/universal-translator v0.17.0  
go: downloading github.com/leodido/go-urn v1.2.0  
go: downloading google.golang.org/protobuf v1.26.0  
go: downloading github.com/google/go-cmp v0.5.6  
go: downloading github.com/go-playground/locales v0.13.0  
[+] Finished  
  
└─(nee@4pfsec)-[~/Havoc/Teamserver]  
└─$
```

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

```
nee-attacker  
└─(nee@4pfsec)-[~/Havoc/Teamserver]  
└─$ ./teamserver  
  
  HAVOC 0.10 Alpha  
  
  pwn and elevate until it is done  
  
Havoc Teamserver  
  
Usage:  
  teamserver [flags]  
  teamserver [command]  
  
Available Commands:  
  help      Help about any command  
  server    server command  
  
Flags:  
  -h, --help  help for teamserver  
  
Use "teamserver [command] --help" for more information about a command.  
  
└─(nee@4pfsec)-[~/Havoc/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

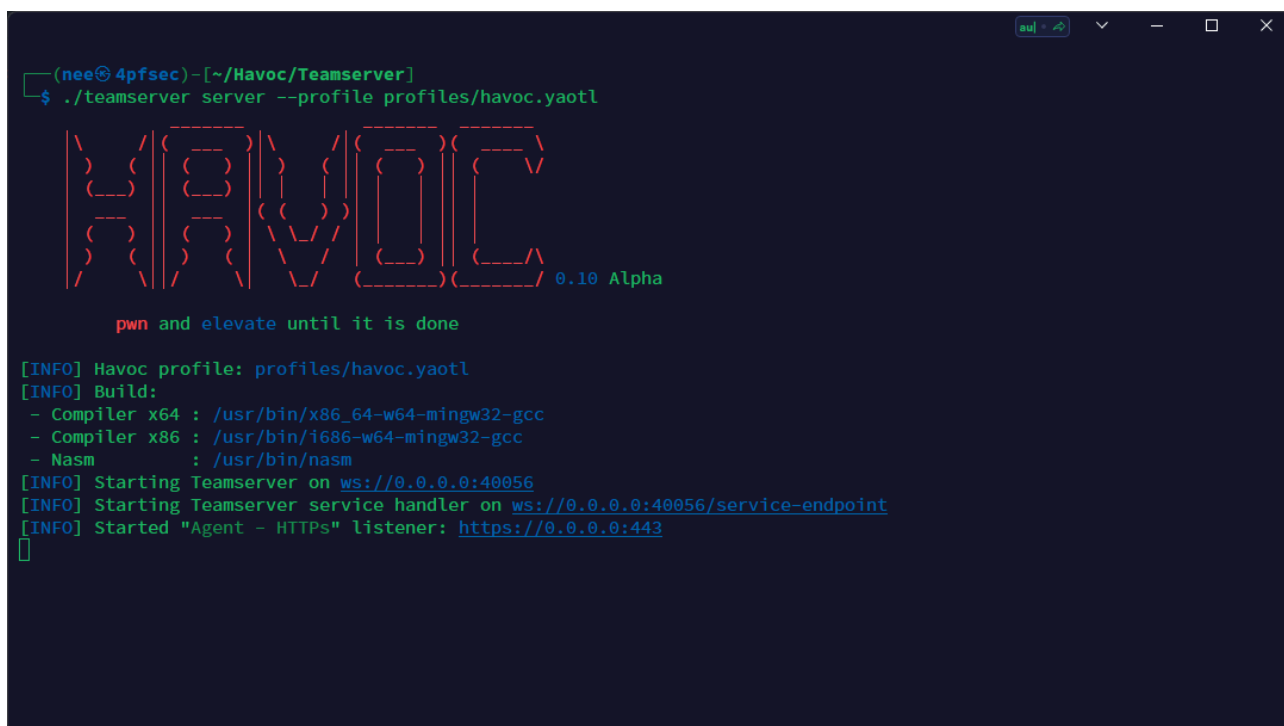
The teamserver allows us to specify a profile or use the default one. The profile allows us to edit configs of the following domains:

- 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
```



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

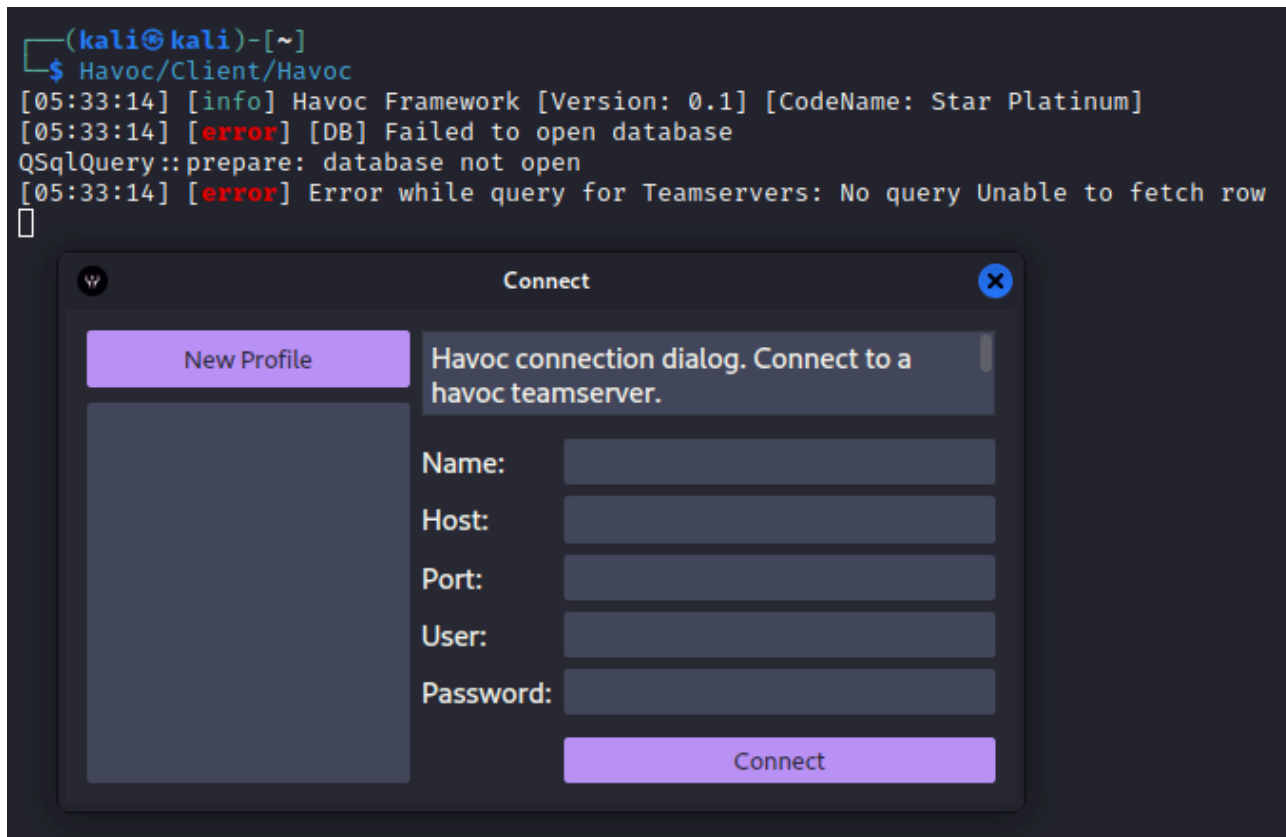
  HAVOC 0.10 Alpha
  pwn and elevate until it is done

[INFO] Havoc profile: profiles/havoc.yaotl
[INFO] Build:
- Compiler x64 : /usr/bin/x86_64-w64-mingw32-gcc
- Compiler x86 : /usr/bin/i686-w64-mingw32-gcc
- Nasm       : /usr/bin/nasm
[INFO] Starting Teamserver on ws://0.0.0.0:40056
[INFO] Starting Teamserver service handler on ws://0.0.0.0:40056/service-endpoint
[INFO] Started "Agent - HTTPs" listener: https://0.0.0.0:443
█
```

Client

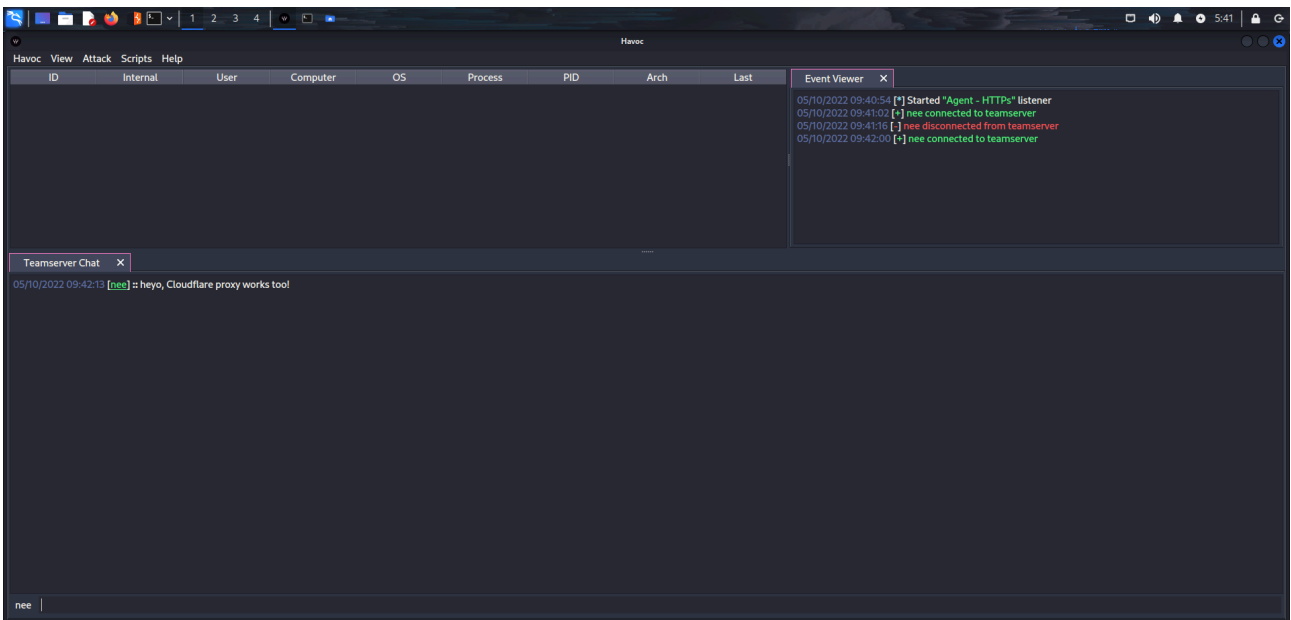
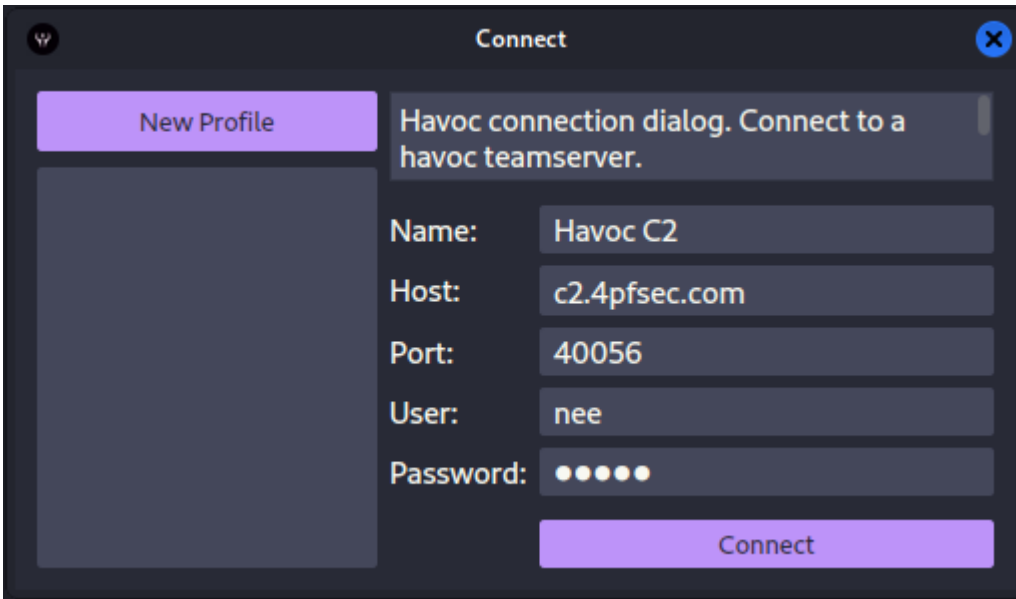
Running the Client

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



Connecting to the teamserver

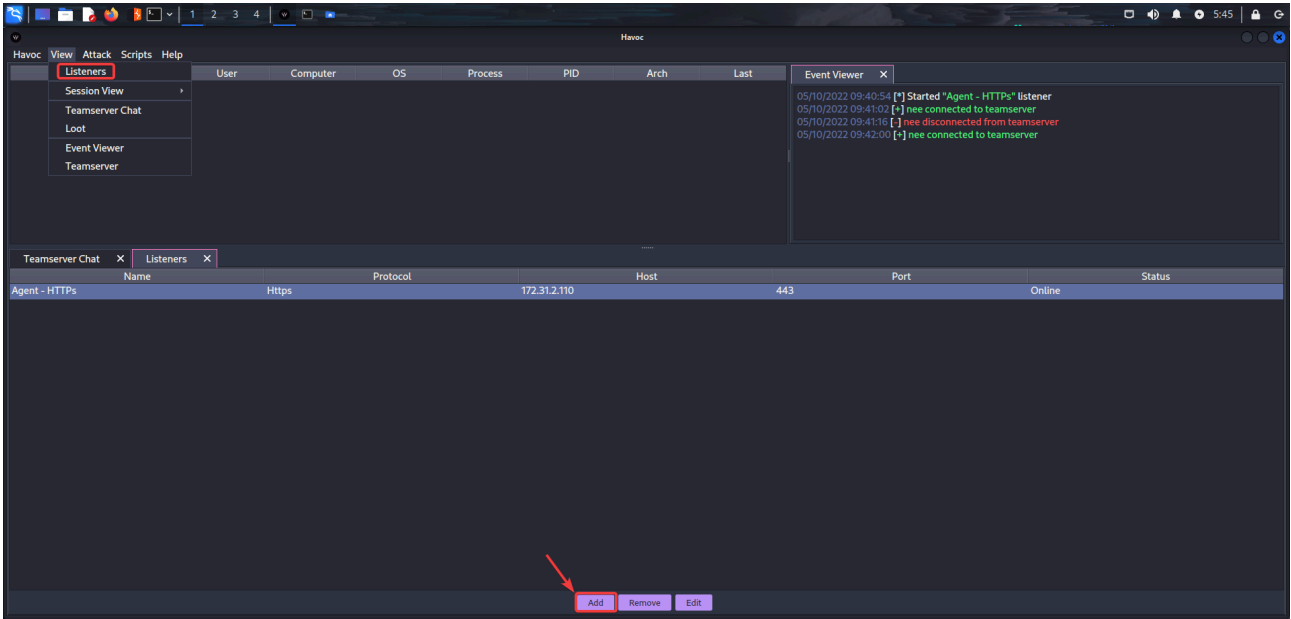
- Name
- C2 Host
- C2 port
- C2 User:Password



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



Edit Listener

Name: Agent - HTTPs

Payload: Http

Config Options

Hosts: 0.0.0.0 [Add] [Clear]

Host Rotation: random

Host (Bind): 172.31.2.110

Port: 443

User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537...

Headers: X-Havoc: true [Add] [Clear]
X-Havoc-Agent: Demon

Uris: /text.gif [Add] [Clear]

Host Header:

Enable Proxy connection

Proxy Type: https

Proxy Host:

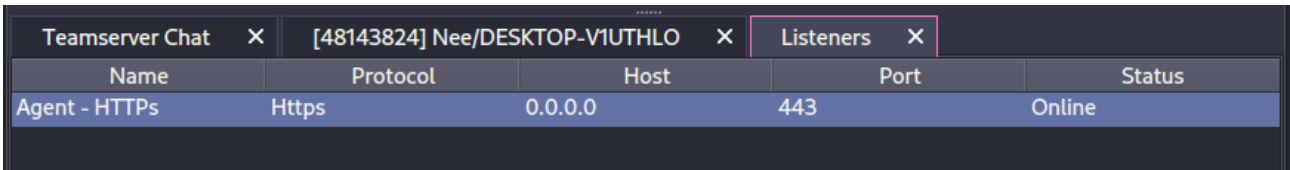
Proxy Port:

UserName:

Password:

[Save] [Close]

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



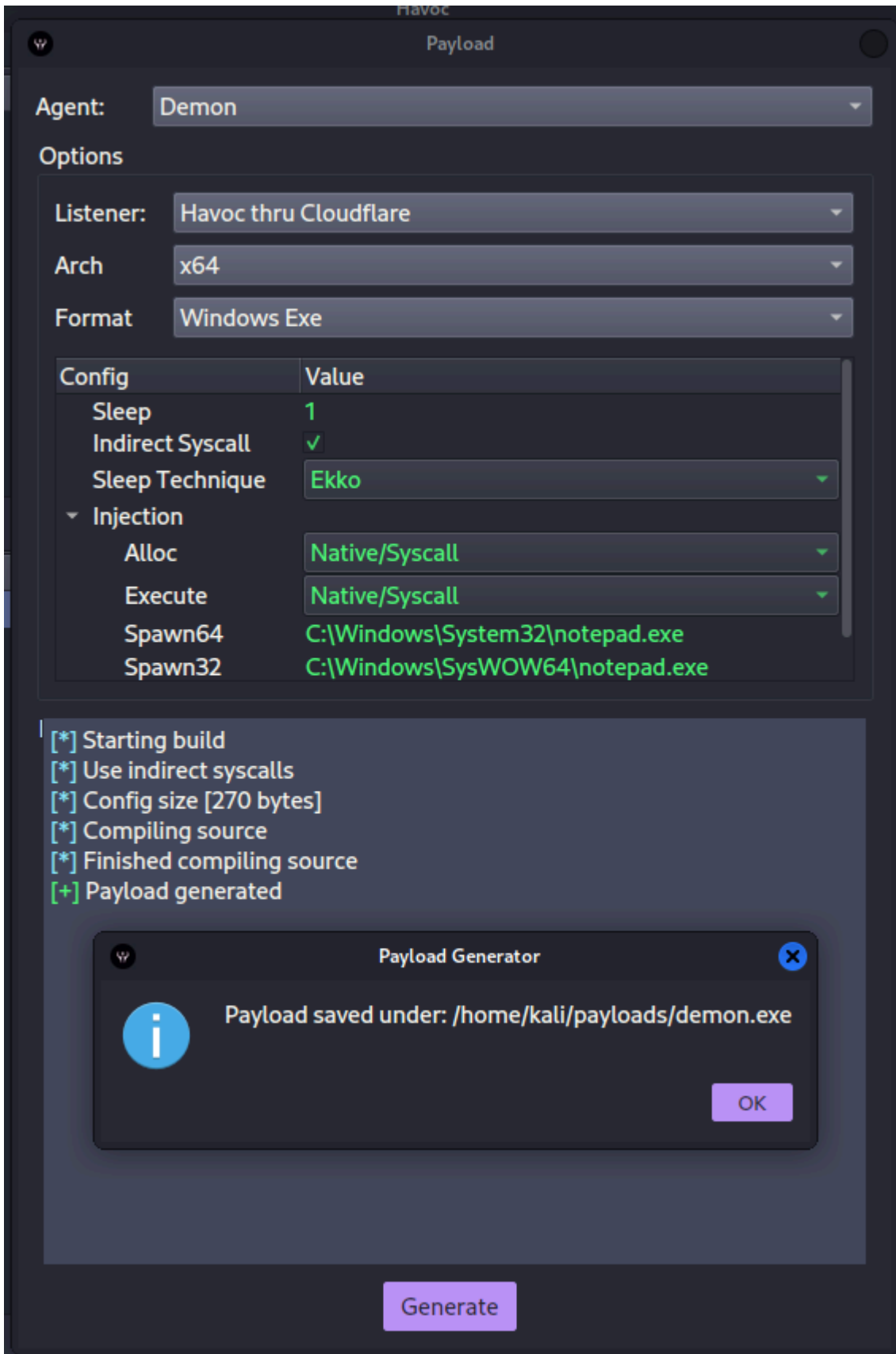
The screenshot shows a window titled 'Listeners' with a table of active connections. The table has five columns: Name, Protocol, Host, Port, and Status. One entry is visible: 'Agent - HTTPs' with Protocol 'Https', Host '0.0.0.0', Port '443', and Status 'Online'.

Name	Protocol	Host	Port	Status
Agent - HTTPs	Https	0.0.0.0	443	Online

Generating Payload (UNDETECTED BY Windows Defender)

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

Attack->Payload->Generate

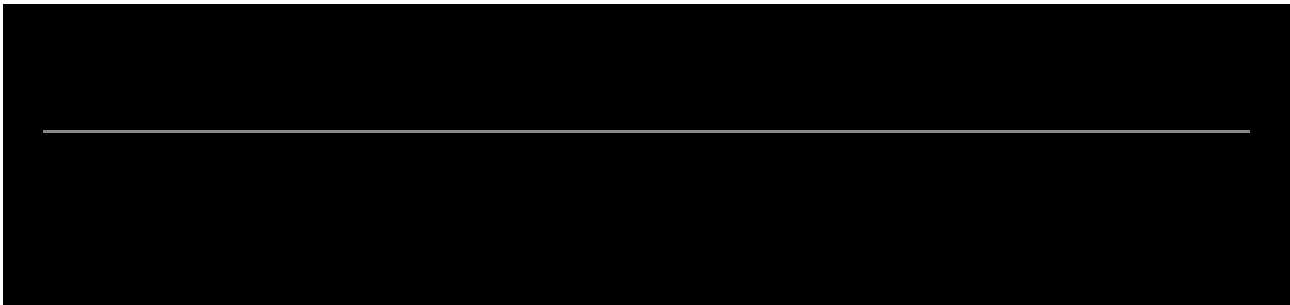
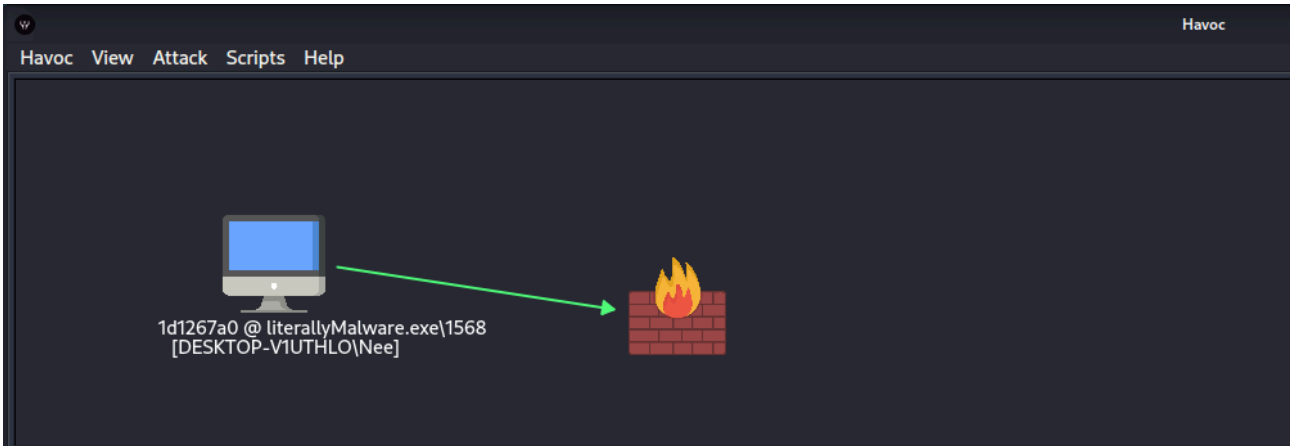


Callback to C2 (UNDETECTED BY Windows Defender)

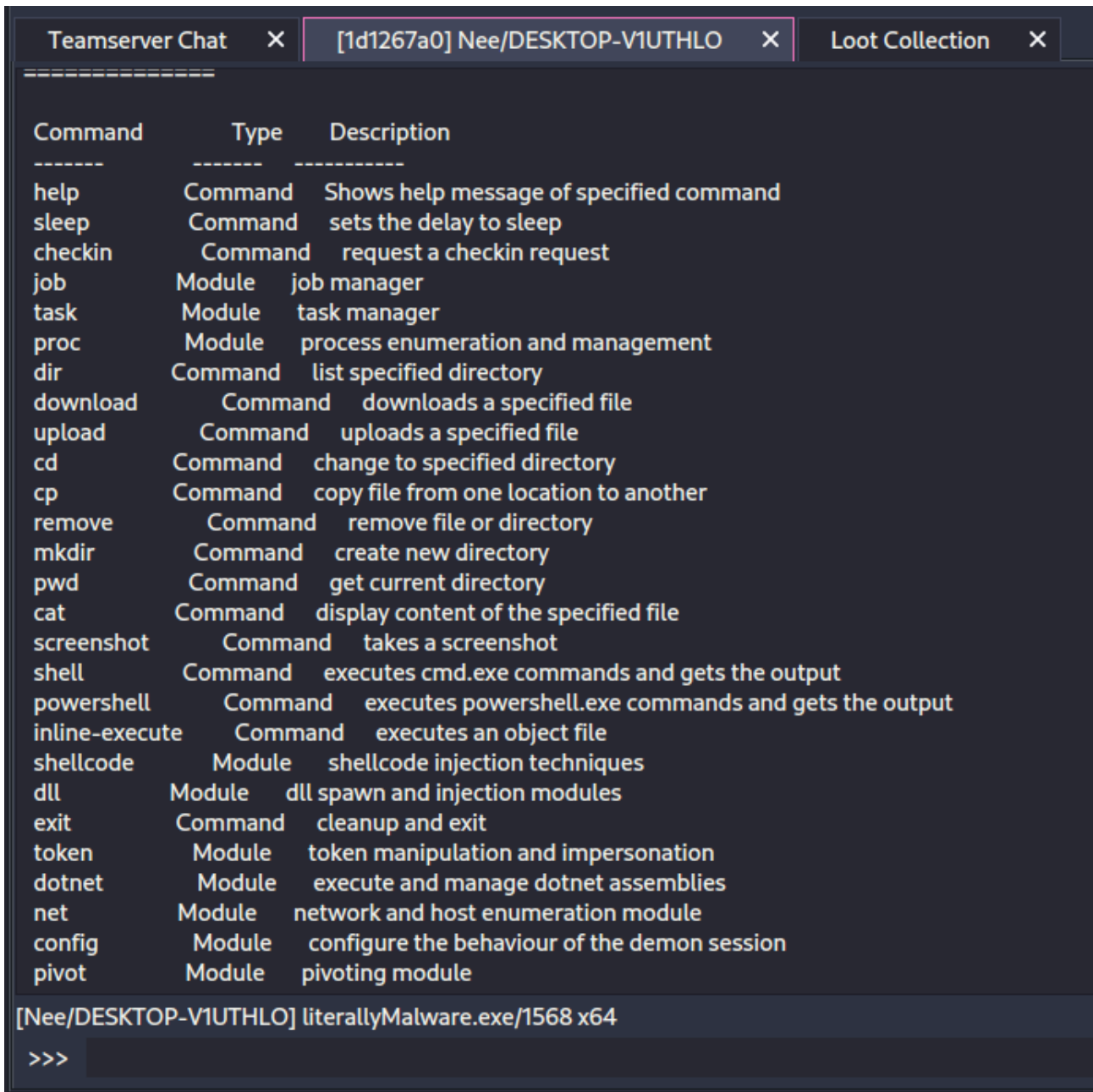
As of writing, the callback method is not picked up by Microsoft Defender. (05/10/22)

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

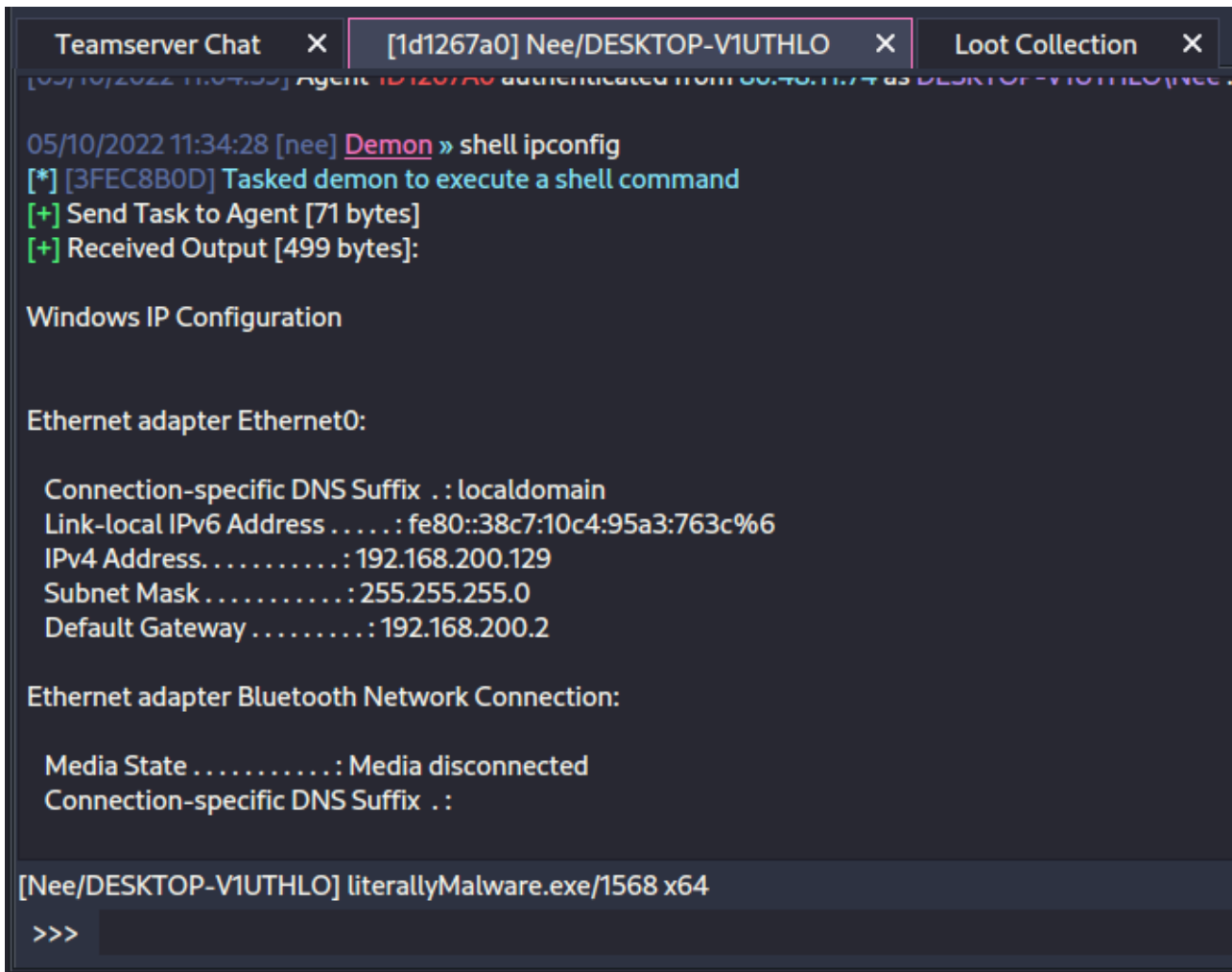


There's a whole list of commands that you're able to run on the target once it calls back to your C2. The target will fetch the C2 for jobs based on the given sleep duration during payload generation.

shell

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

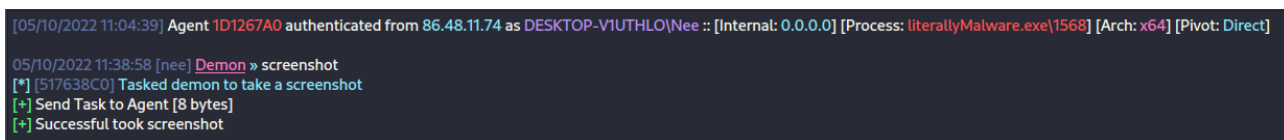
\>>> shell [command]



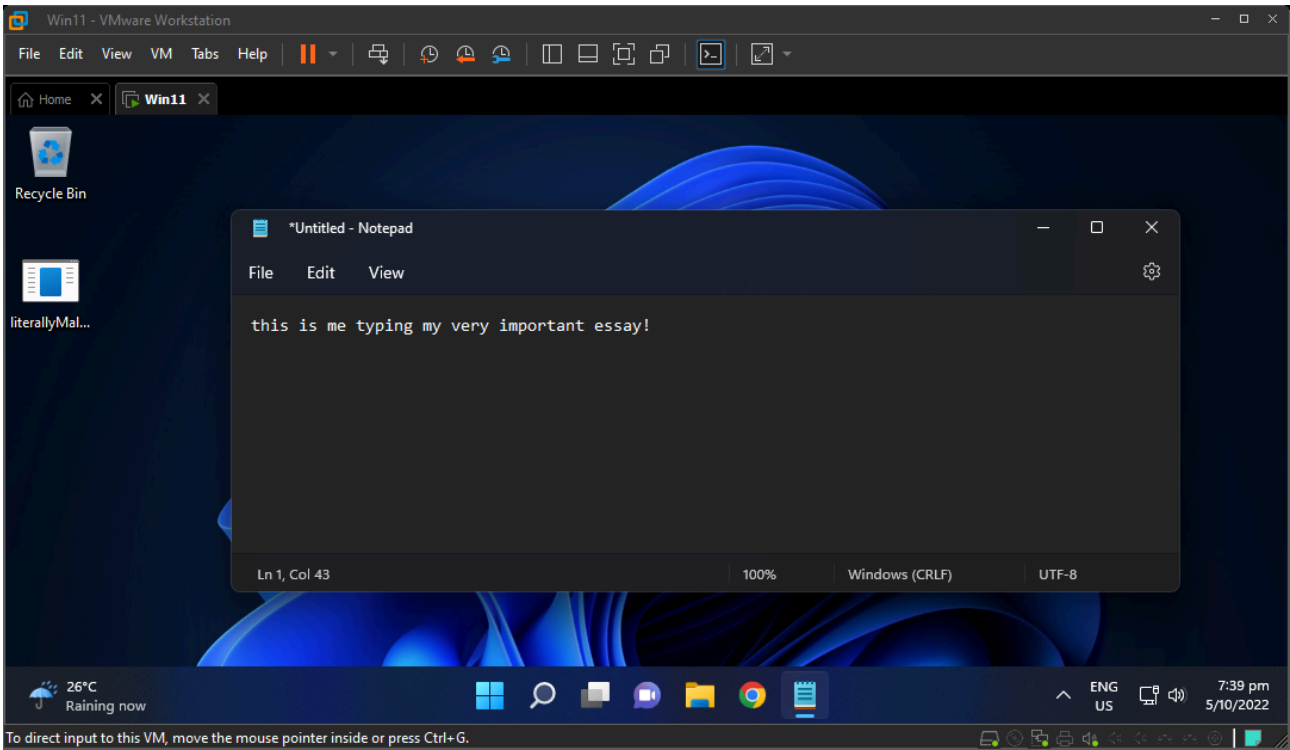
Screenshot

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

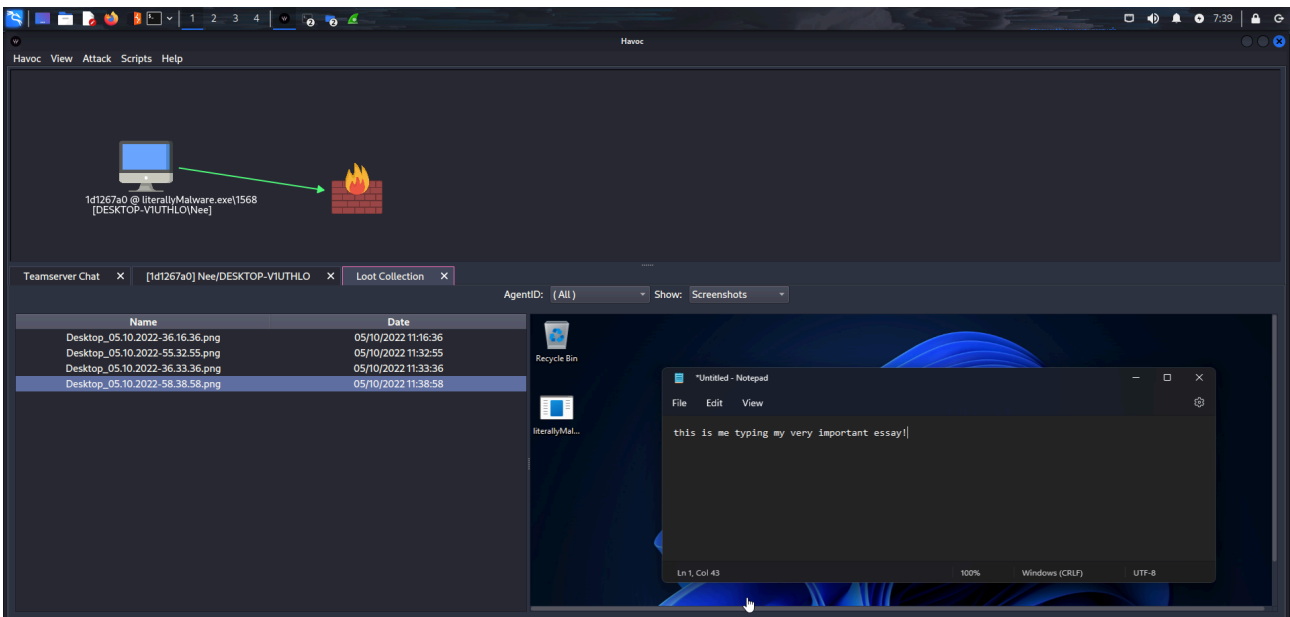
\>>> screenshot



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/>