ANONYMOUS

LOGINS FOR PENTESTER

WWW.HACKINGARTICLES.IN

Contents

Introduction..3

Setting up Anonymous FTP..3

Attacking Anonymous FTP...8

Setting up Anonymous SMB 10

Attacking Anonymous SMB 13

Conclusion 14

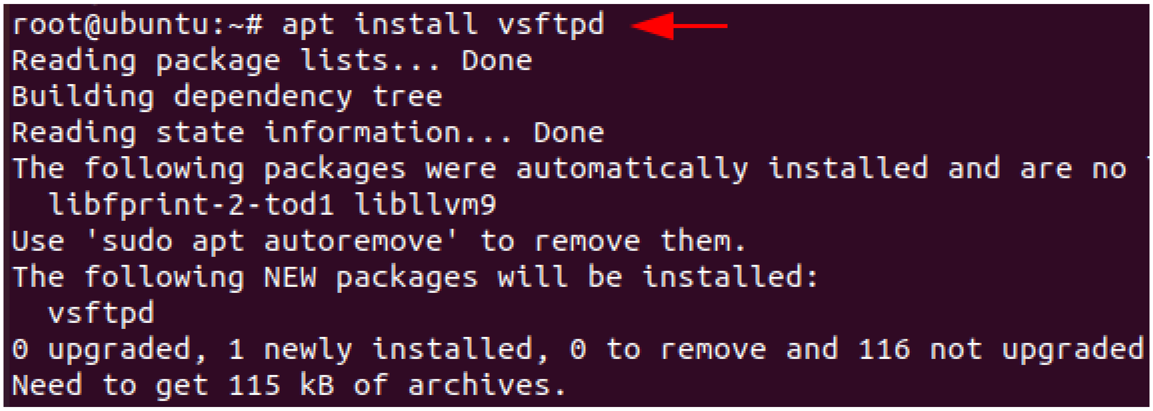
Introduction

Anonymous Logins are a feature that allows the user to set up a service that is accessible by any user. It doesn’t need specific credentials to access that resource. Various servers that wish to host data that should be accessible to a wide range of users via anonymous logins. In real life, while performing network penetration testing, a tester should be able to identify the anonymous service and test it. We will also be looking behind the scenes at how these anonymous services are setup on our local target machine running Ubuntu. We will be learning about the FTP service and the SMB service.

Setting up Anonymous FTP

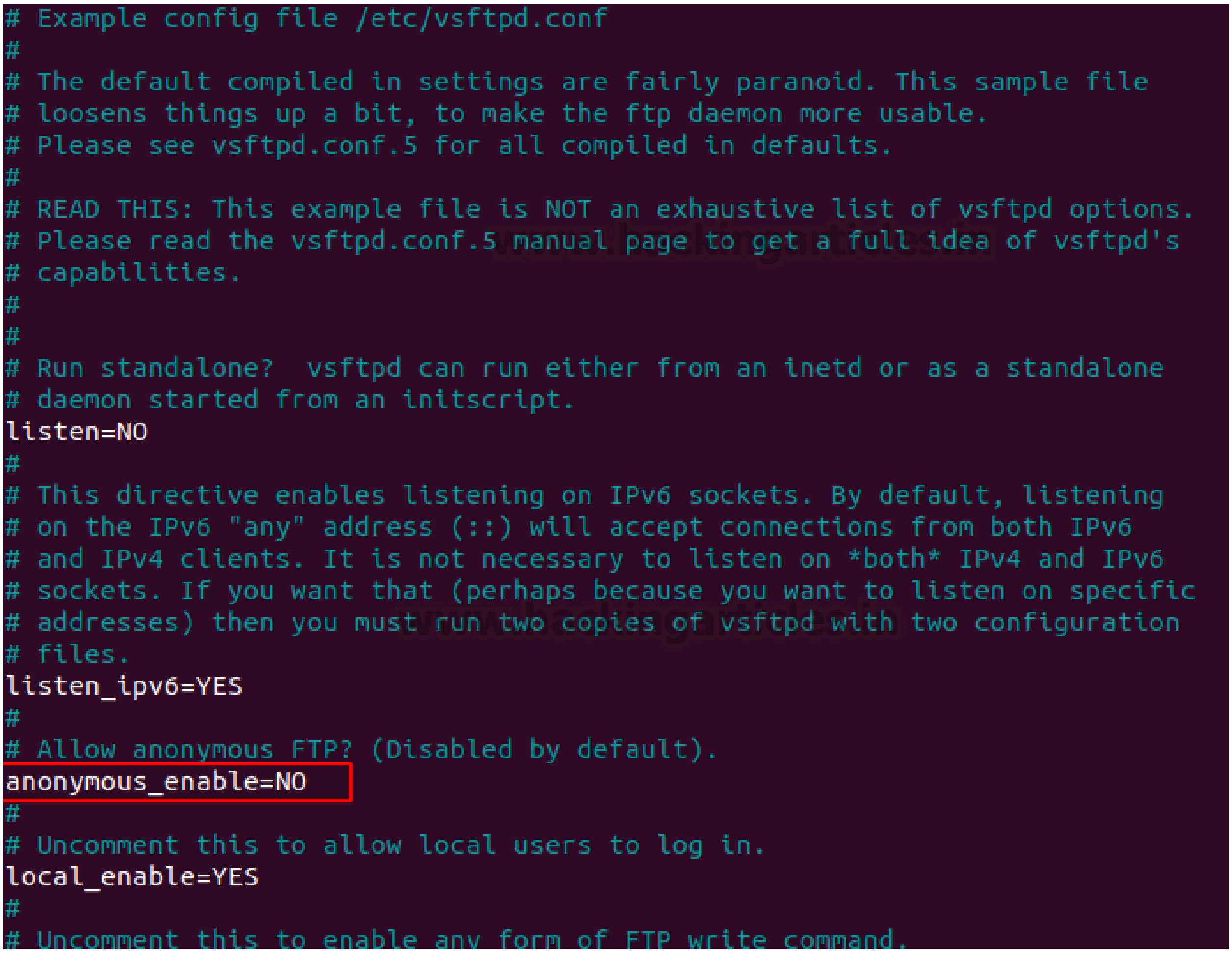
We will begin by demonstrating the process of setting up anonymous access on the FTP service. We have an Ubuntu machine with root access. We install the vsftpd using the apt command.

```
apt install vsftpd
```



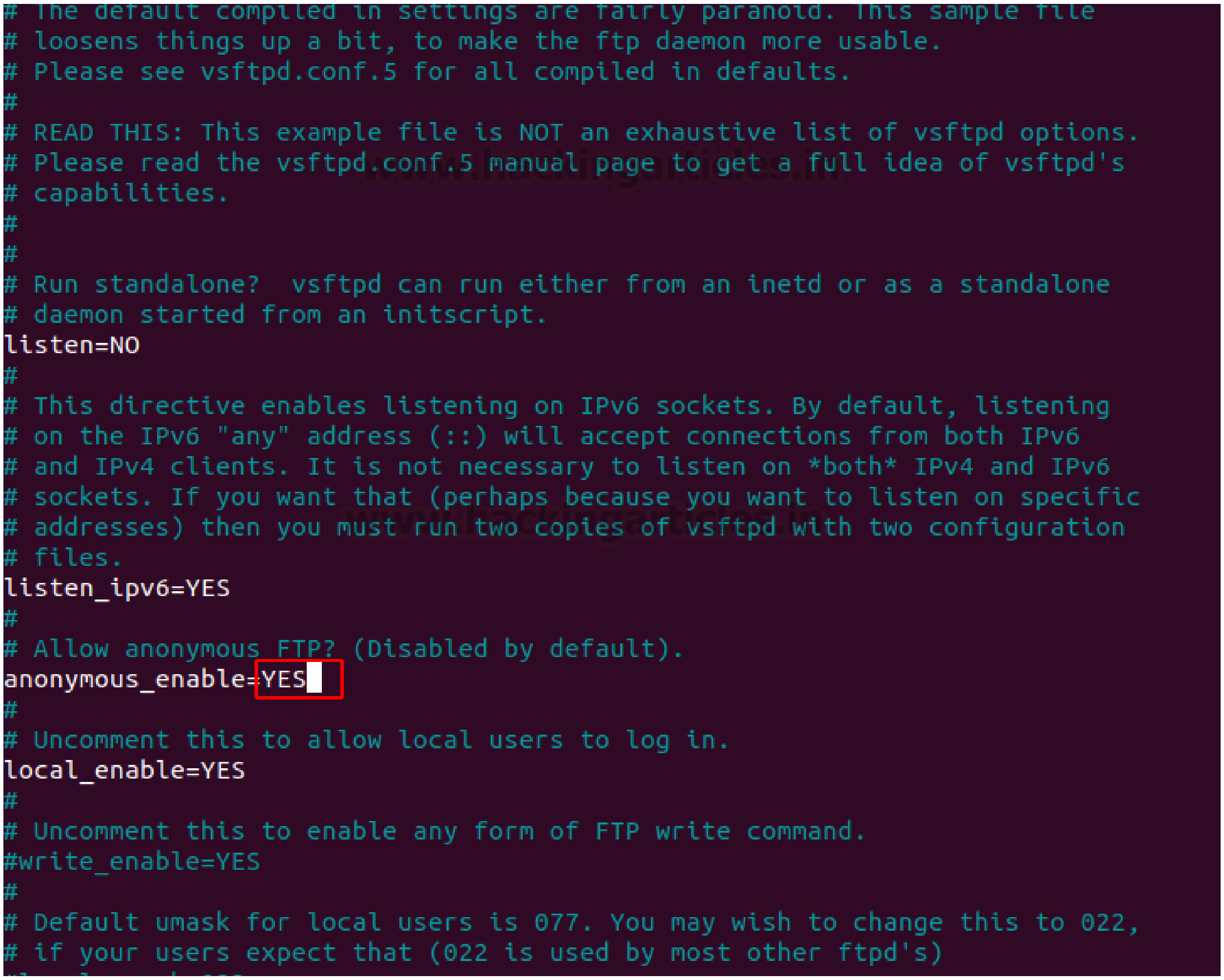
```
root@ubuntu:~# apt install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no
  libfprint-2-tod1 libllvm9
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 116 not upgraded
Need to get 115 kB of archives.
```

Each service that is installed on a Linux machine has a configuration file that can be used to tweak options and settings for that particular service. By default, anonymous login is disabled on the vsftpd. We will need to edit the **/etc/vsftpd.conf** configuration file in order to enable the anonymous login functionality. We edit the configuration file with nano, but you can use any editor of your choice, such as vi or sublime.We go through all of the other options and comments until we get to the "anonymous_enabled=NO" option, which is shown in the image below.



```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
```

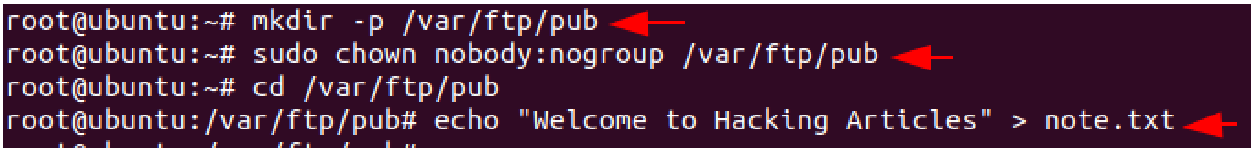
To enable Anonymous Login on the machine, change the "anonymous_enabled=NO" option to "anonymous_enabled=YES."Refer to the screenshot below.



```
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
```

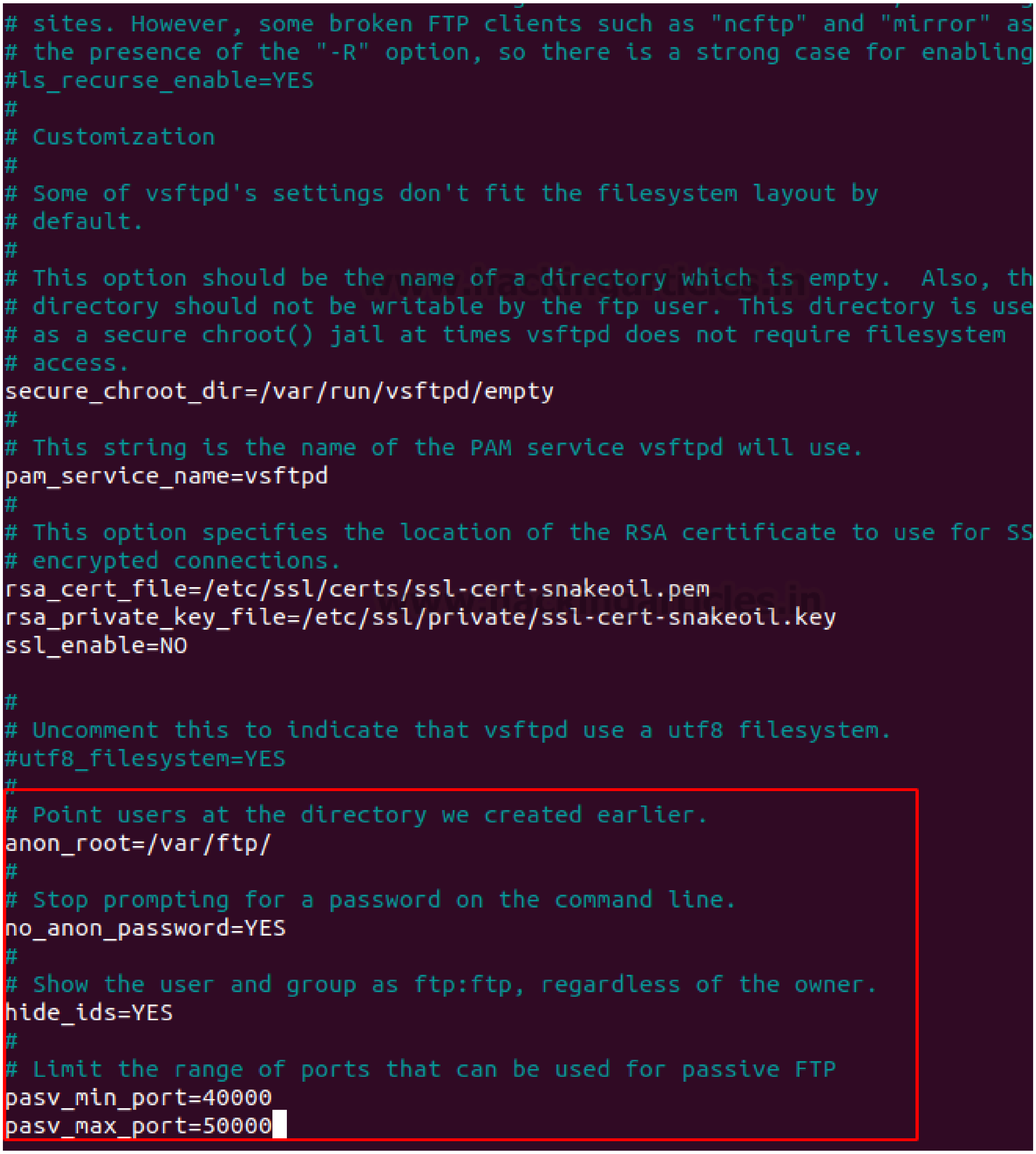
Just enabling the Anonymous login or installing a service is not enough to get it working. We want a fully functional FTP service. To do this, we need to be able to share files using FTP, and since we have enabled anonymous login, we should be able to download the files from the Ubuntu machine using anonymous access. The FTP service requires a directory whose contents can be shared over the network. We create a directory in the /var directory. We named the directory after the pub. We also need to change the ownership of the directory in order to make it suitable for sharing data. After creating and changing ownership, we move into the directory and create a file with the message "Welcome to Hacking Articles" in it. We named the text file note.txt.

```
mkdir -p /var/ftp/pub
sudo chown nobody:nogroup /var/ftp/pub
cd /var/ftp/pub
echo "Welcome to Hacking Articles" > note.txt
```



```
root@ubuntu:~# mkdir -p /var/ftp/pub
root@ubuntu:~# sudo chown nobody:nogroup /var/ftp/pub
root@ubuntu:~# cd /var/ftp/pub
root@ubuntu:/var/ftp/pub# echo "Welcome to Hacking Articles" > note.txt
```

Back to the vsftpd.conf file that we were editing, we need to add a specific configuration to make the anonymous login functional. We add the directory that we just created in the configurations, and then we add the no_anon_password option that will stop prompting for a password. Another option we added is the hide_ids option. When queried, it will revert to the ftp: ftp combination., we need to add the range of ports that can be used for passive FTP.

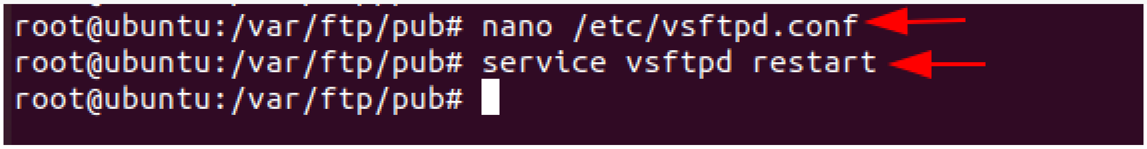


```
# sites. However, some broken FTP clients such as "ncftp" and "mirror" as
# the presence of the "-R" option, so there is a strong case for enabling
#ls_recurse_enable=YES
#
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# This option should be the name of a directory which is empty.  Also, th
# directory should not be writable by the ftp user. This directory is use
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
secure_chroot_dir=/var/run/vsftpd/empty
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SS
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES
#
# Point users at the directory we created earlier.
anon_root=/var/ftp/
#
# Stop prompting for a password on the command line.
no_anon_password=YES
#
# Show the user and group as ftp:ftp, regardless of the owner.
hide_ids=YES
#
# Limit the range of ports that can be used for passive FTP
pasv_min_port=40000
pasv_max_port=50000
```

This completes all the configurations that we require to setup an FTP service with anonymous login enabled on an Ubuntu machine. All that is required is to restart the vsftpd service in order to put the new configurations into effect. Now we will refer to our Kali Linux machine, i.e., the attacker machine.

```
nano /etc/vsftpd.conf
service vsftpd restart
```

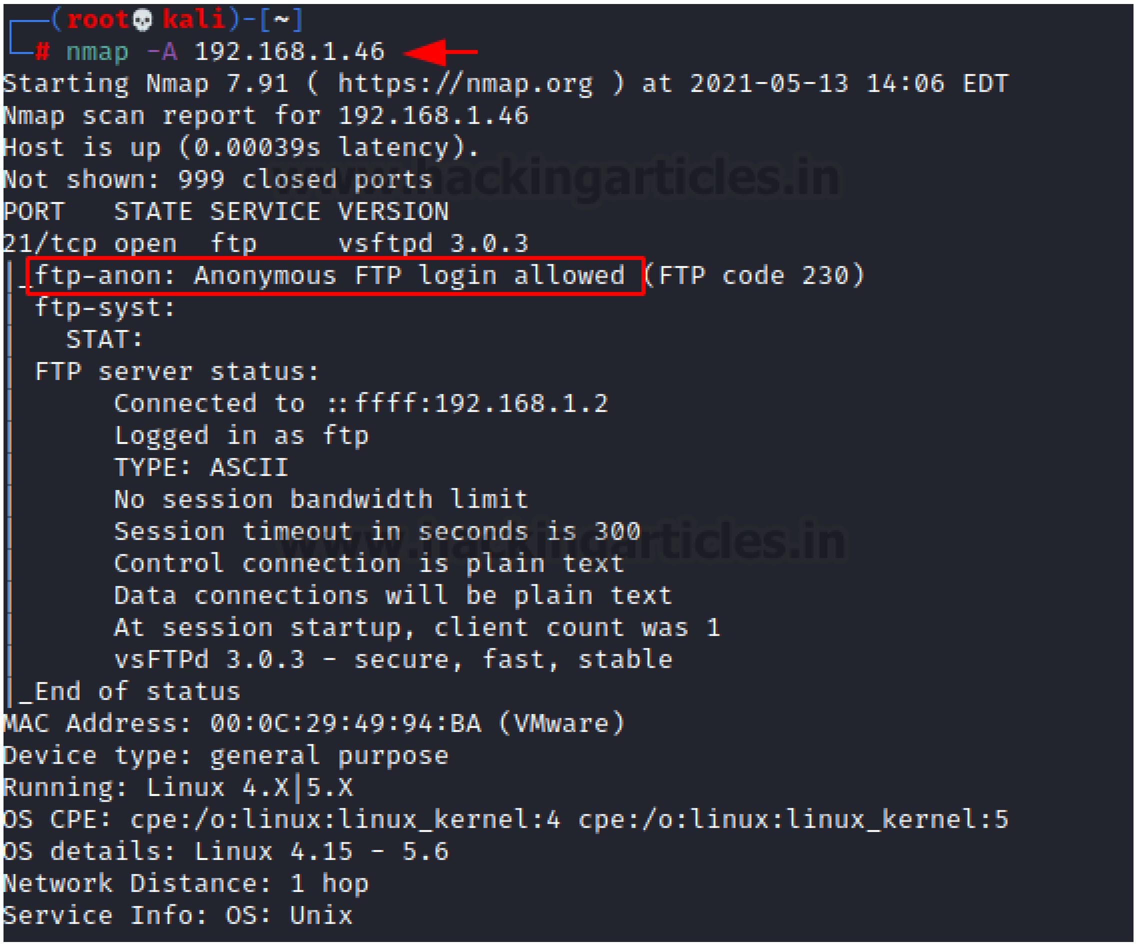


```
root@ubuntu:/var/ftp/pub# nano /etc/vsftpd.conf
root@ubuntu:/var/ftp/pub# service vsftpd restart
root@ubuntu:/var/ftp/pub#
```

Attacking Anonymous FTP

When attacking or targeting a system, one of the initial steps that an attacker takes is to perform a scan of the target. This scan gives the attacker information such as open ports and running services. We used Nmap to scan the Ubuntu machine that we had just configured. We can see that the Nmap was able to identify that the FTP service was functional on the target machine and it also takes another step into enumeration and informs the attacker that the FTP service has Anonymous Login Enabled.

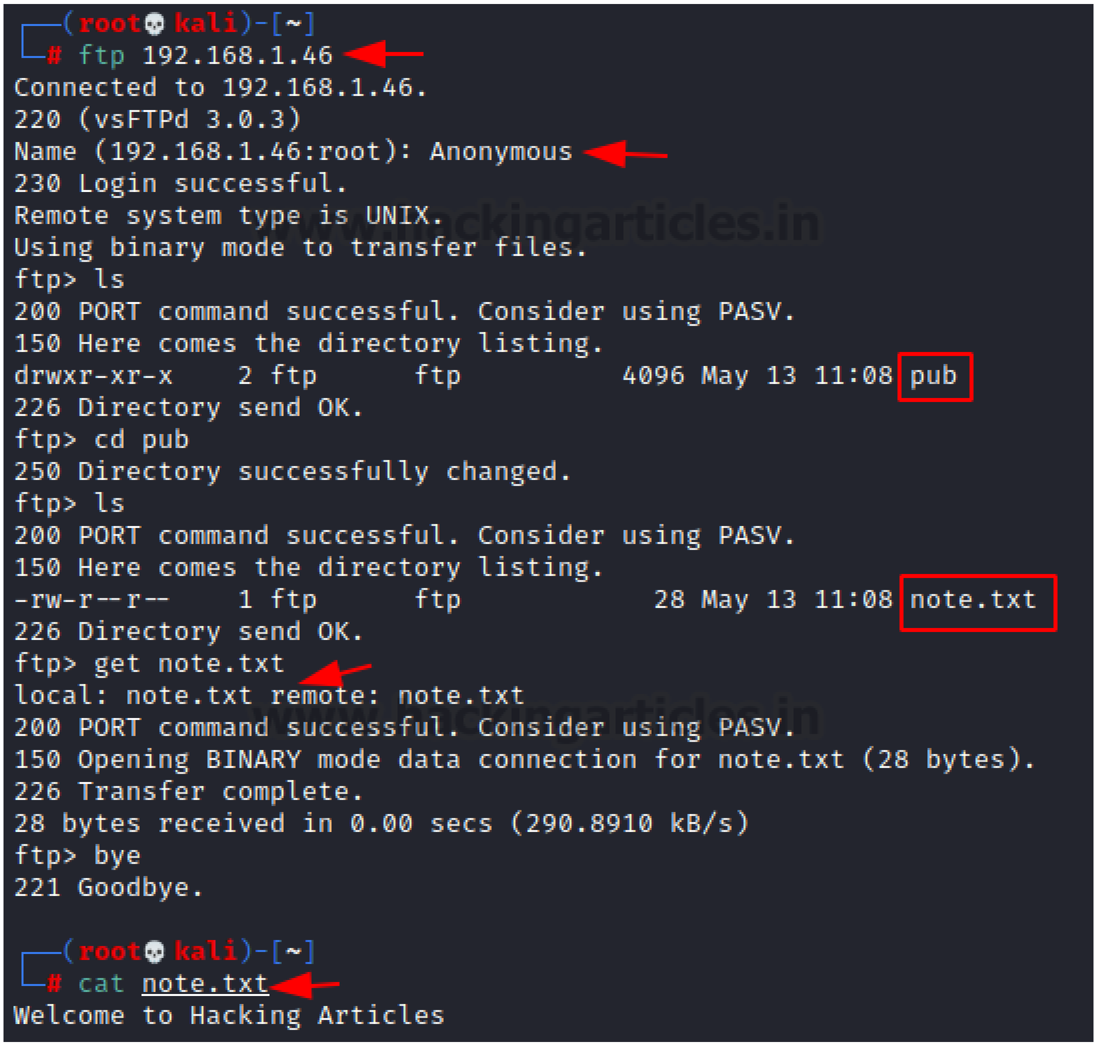
```
nmap -A 192.168.1.46
```



```
┌──(root💀kali)-[~]
└─# nmap -A 192.168.1.46
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-13 14:06 EDT
Nmap scan report for 192.168.1.46
Host is up (0.00039s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.1.2
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
MAC Address: 00:0C:29:49:94:BA (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Unix
```

Now that it has been confirmed that the FTP service is running with Anonymous Login enabled, let's try to access the service. To do this, we will connect to the FTP service by providing the IP address of the machine. Because we don't have any user credentials and anonymous login is enabled, we'll enter "Anonymous" in the Name field and be logged in. We can run the directory listing command ls to find out the files that are shared over FTP. We see that there is a text file by the name of note.txt. We can transfer the text file using the get command as depicted below. After the transfer, we can read the text file to confirm that we have successfully gained the data from the file that was created on the Ubuntu machine.

```
ftp 192.168.1.46
Anonymous
ls
cd pub
ls
get note.txt
bye
cat note.txt
```



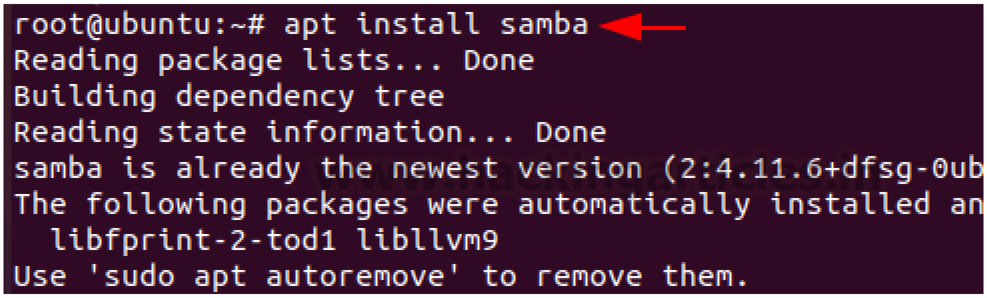
```
┌──(root💀kali)-[~]
└─# ftp 192.168.1.46
Connected to 192.168.1.46.
220 (vsFTPd 3.0.3)
Name (192.168.1.46:root): Anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 May 13 11:08 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp            28 May 13 11:08 note.txt
226 Directory send OK.
ftp> get note.txt
local: note.txt remote: note.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for note.txt (28 bytes).
226 Transfer complete.
28 bytes received in 0.00 secs (290.8910 kB/s)
ftp> bye
221 Goodbye.

┌──(root💀kali)-[~]
└─# cat note.txt
Welcome to Hacking Articles
```

Setting up Anonymous SMB

The next service that can be set up with anonymous access is the SMB service. As it was originally designed for Windows systems, we need to install the samba service on our Ubuntu machine. As with the vsftpd, we used apt to install the samba service, as shown below.

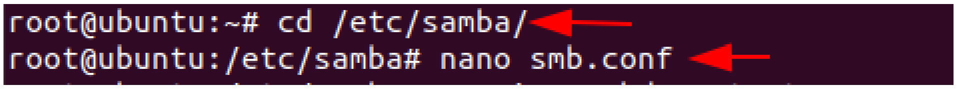
```
apt install samba
```



```
root@ubuntu:~# apt install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
samba is already the newest version (2:4.11.6+dfsg-0ub
The following packages were automatically installed an
  libfprint-2-tod1 libllvm9
Use 'sudo apt autoremove' to remove them.
```

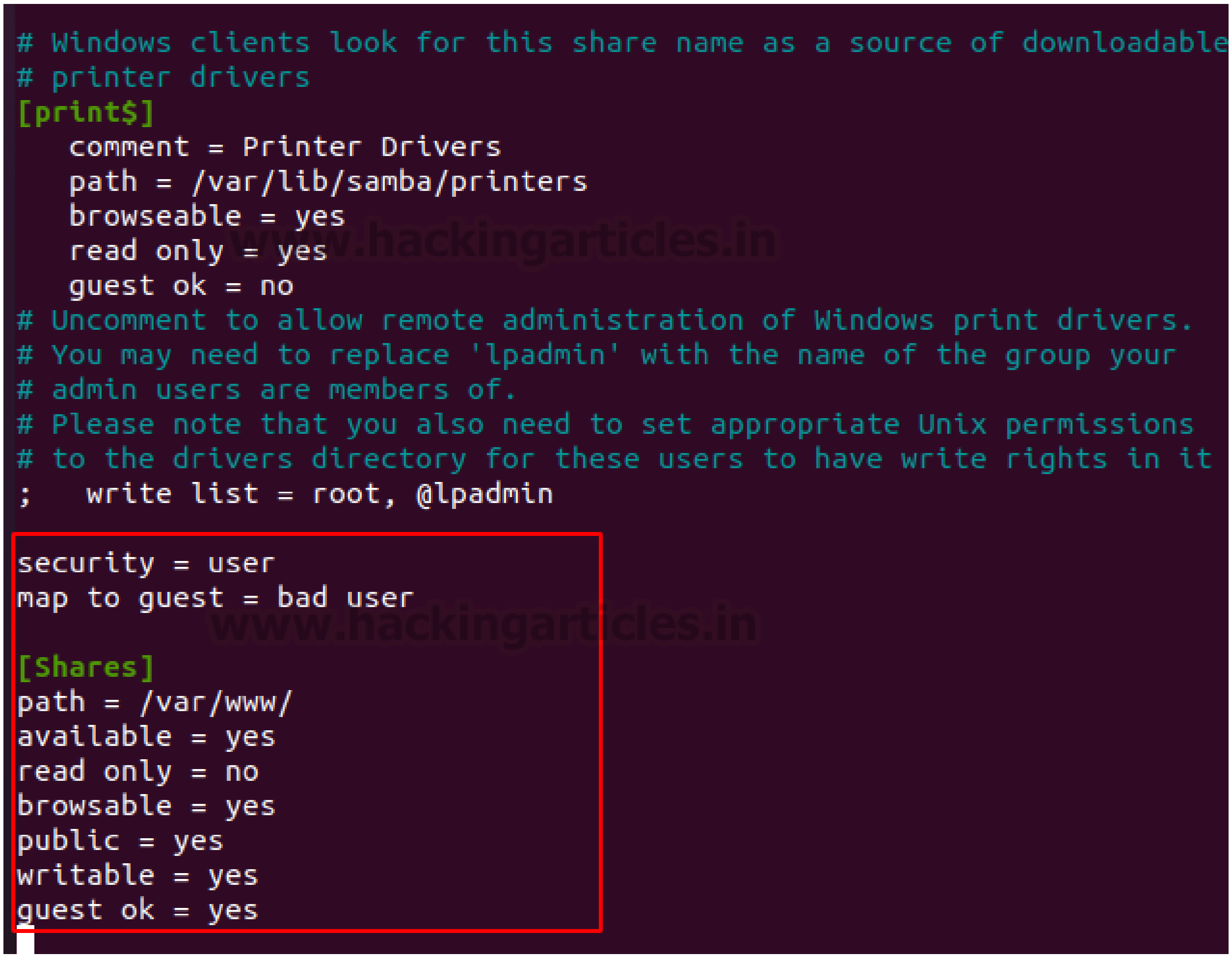
Like all services that are installed on any Linux machine, Samba also has a configuration file that is located inside the /etc directory. Since we are trying to setup the service with anonymous login, we are going to add some additional configurations as compared to the basic installation of samba.

```
cd /etc/samba/
```



```
root@ubuntu:~# cd /etc/samba/
root@ubuntu:/etc/samba# nano smb.conf
```

We are using the nano editor, but you can basically use any editor of your choice. Moving down to the file, we add the following configurations, such as the directory that should be used for sharing the files. We are making the /var/www directory for this purpose. We need to give it proper permissions, such as browsable and public, so that it can be accessed by anonymous login.



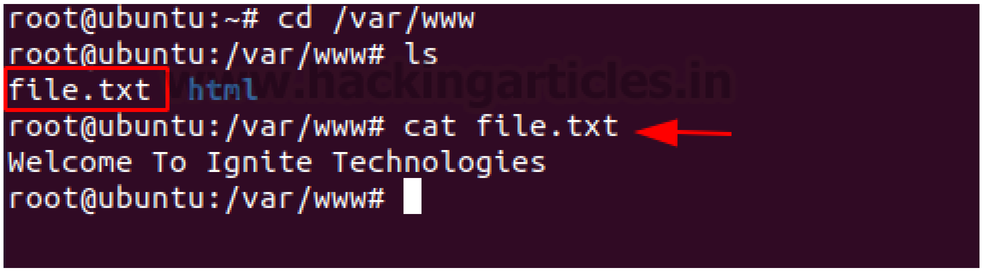
```
# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

security = user
map to guest = bad user

[Shares]
path = /var/www/
available = yes
read only = no
browsable = yes
public = yes
writable = yes
guest ok = yes
```

The next thing that we need to do is create a file that can be used to test the ability of file transfer using SMB. We created a text file called file.txt and filled it with the message "Welcome To Ignite Technologies". You will need to restart the service in order to make the configurations active.

```
cd /var/www
ls
cat file.txt
```

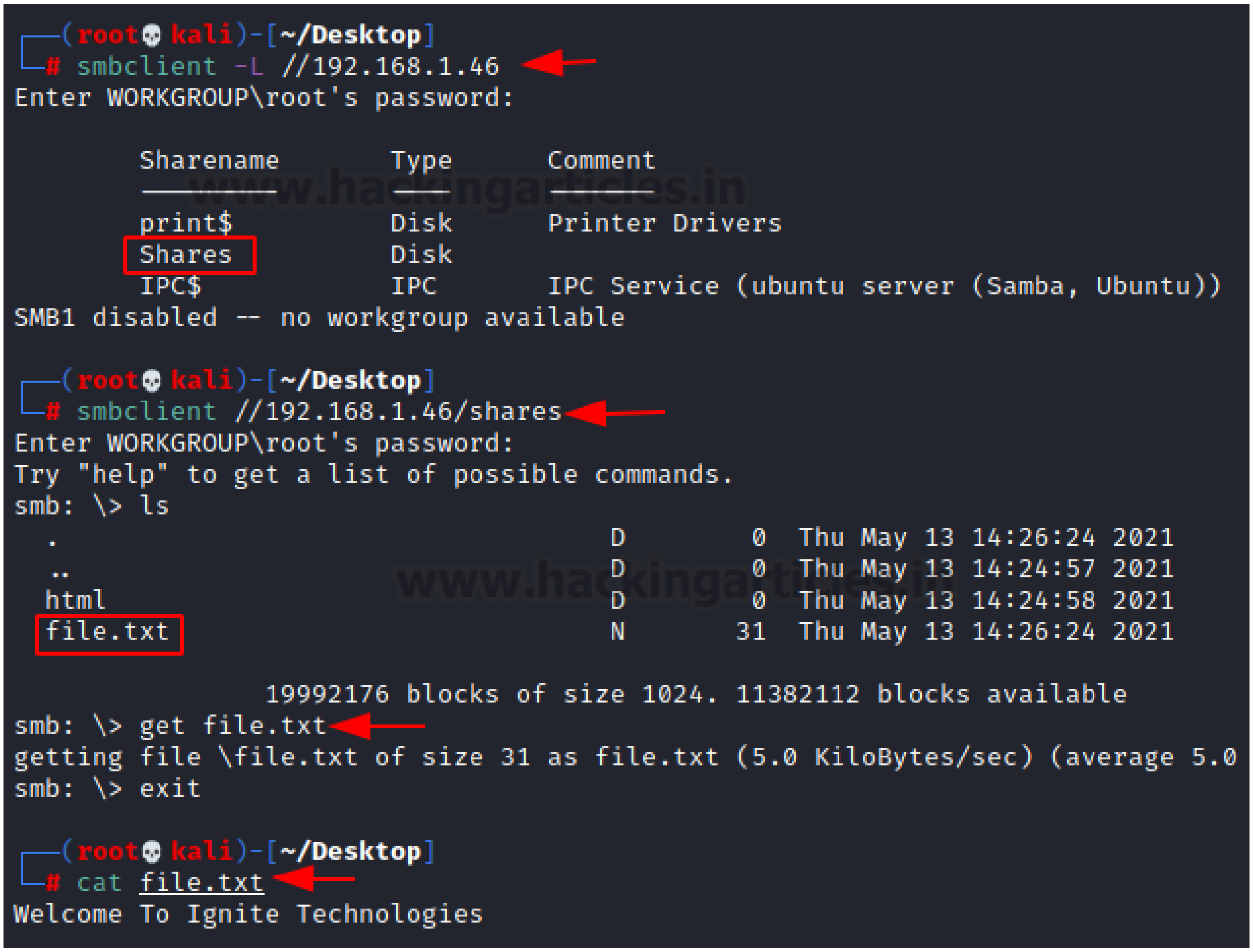


```
root@ubuntu:~# cd /var/www
root@ubuntu:/var/www# ls
file.txt  html
root@ubuntu:/var/www# cat file.txt
Welcome To Ignite Technologies
root@ubuntu:/var/www#
```

Attacking Anonymous SMB

As we did with the FTP service, it is also possible to check if the service is running on the target machine using the nmap scan. Although we are not going to demonstrate it here. We are going to proceed with the assumption that the service is up and running on the target machine. We connect to the service using smbclient. It is quite clear from the image below that we didn't provide a user or password combination to connect to the service since anonymous login is enabled. We then enumerated the shares and found the file.txt shared. We transferred the file to the local Kali Linux machine and confirmed that the SMB Anonymous Login service is active and working.

```
smbclient -L //192.168.1.46
smbclient  //192.168.1.46/shares
cat file.txt
```



```
┌──(root💀kali)-[~/Desktop]
└─# smbclient -L //192.168.1.46
Enter WORKGROUP\root's password: 

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        Shares          Disk      
        IPC$            IPC       IPC Service (ubuntu server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

┌──(root💀kali)-[~/Desktop]
└─# smbclient //192.168.1.46/shares
Enter WORKGROUP\root's password: 
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu May 13 14:26:24 2021
  ..                                  D        0  Thu May 13 14:24:57 2021
  html                                D        0  Thu May 13 14:24:58 2021
  file.txt                            N       31  Thu May 13 14:26:24 2021

                19992176 blocks of size 1024. 11382112 blocks available
smb: \> get file.txt
getting file \file.txt of size 31 as file.txt (5.0 KiloBytes/sec) (average 5.0
smb: \> exit

┌──(root💀kali)-[~/Desktop]
└─# cat file.txt
Welcome To Ignite Technologies
```

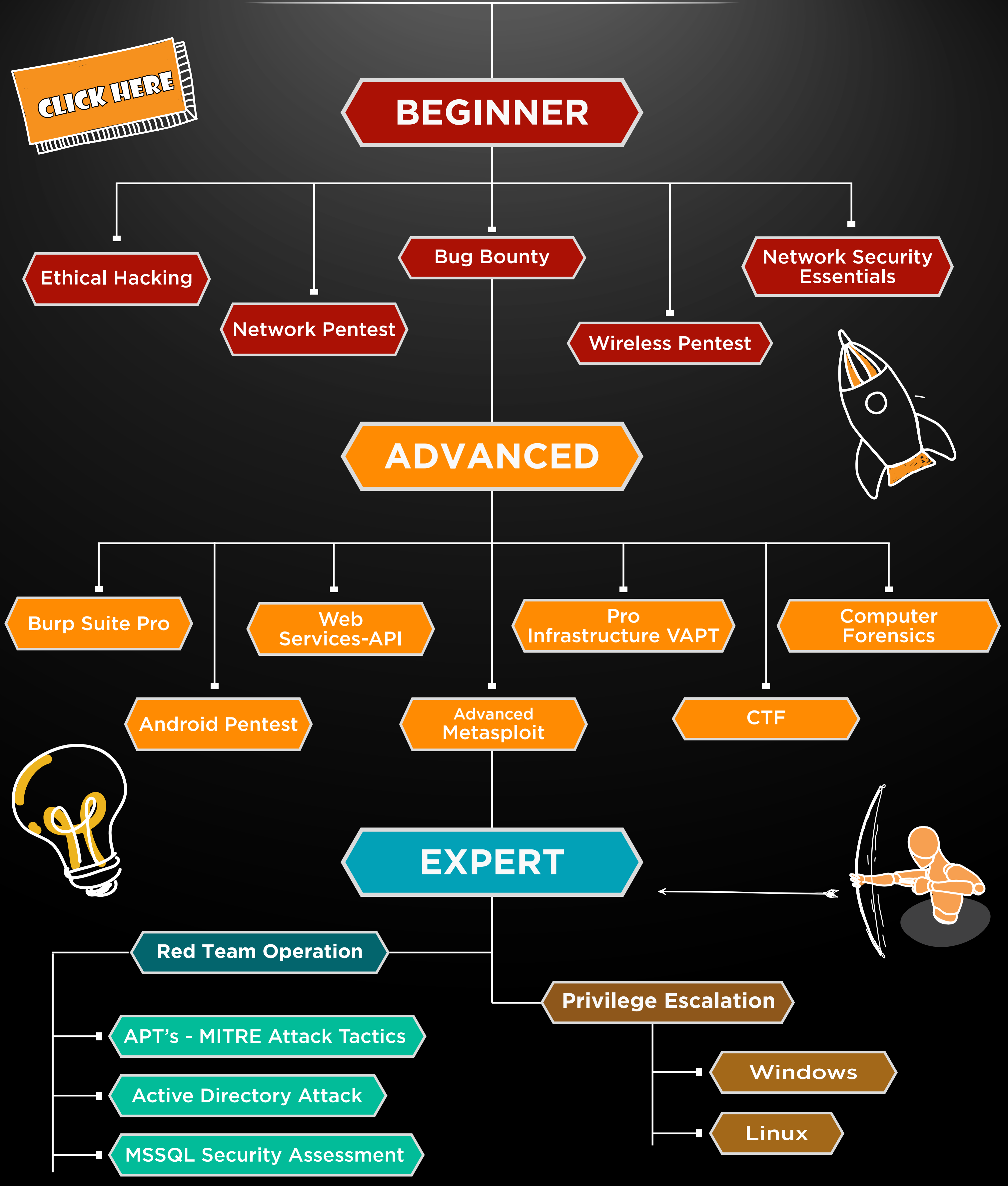
Conclusion

Anonymous logins are quite common in real-life environments and in the Capture the Flags challenges as well. As an attacker, it is important to understand how it works and what kind of setup is required to enable the anonymous login. Most of all, it is important to know how to interact with this kind of access.

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