How To Scan a Network With Hping3

Hping3

Hping3 is a command-line oriented TCP/IP packet assembler and analyser and works like Nmap.

The application is able to send customizes TCP/IP packets and display the reply as ICMP echo packets, even more Hping3 supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features like DDOS flooding attacks.

Hping3 can be used to perform:

- OS fingerprinting
- ICMP pings
- Traceroute
- Port scanning

- Firewall testing
- Test IDSes
- Network testing and auditing
- MTU discovery
- Exploit and vulnerabilities discovery
- DDOS and ICMP flooding

Hping3 comes pre-installed with Kali Linux but and can also be installed on most Linux distros, also you need to run the commands with sudo privileges. Visit the official documentation at to learn more on how you can use Hping3
http://www.hping.org/documentation.php

Useful Options

-h	Show this help
-v	Show version
-c	Packet count
-i	–interval
	–flood
-V	Verbose mode
-D	Debugging
-f	Fragment packets
-Q	Display sequence number

-0	RAW IP mode
-1	ICMP mode
-2	UDP mode
-8	SCAN mode
-9	listen mode

-F	Set the FIN flag
-S	Set the SYN flag
-P	Set the PUSH flag

-A	Set the ACK flag
-U	Set the URG flag

Commands

Send a ACK packet to a target

```
hping3 –A 192.168.100.11
```

```
HPING 192.168.100.11 (eth0 192.168.100.11): A set, 40 headers + 0 data bytes
len=46 ip=192.168.100.11 ttl=128 id=29627 sport=0 flags=R seq=0 win=32767 rtt=4.0
ms
len=46 ip=192.168.100.11 ttl=128 id=29628 sport=0 flags=R seq=1 win=32767 rtt=2.0
ms
len=46 ip=192.168.100.11 ttl=128 id=29629 sport=0 flags=R seq=2 win=32767 rtt=2.0
ms
len=46 ip=192.168.100.11 ttl=128 id=29632 sport=0 flags=R seq=3 win=32767 rtt=2.0
ms
len=46 ip=192.168.100.11 ttl=128 id=29633 sport=0 flags=R seq=4 win=32767 rtt=0.6
ms
len=46 ip=192.168.100.11 ttl=128 id=29634 sport=0 flags=R seq=5 win=32767 rtt=8.0
ms
len=46 ip=192.168.100.11 ttl=128 id=29635 sport=0 flags=R seq=6 win=32767 rtt=7.1
ms
len=46 ip=192.168.100.11 ttl=128 id=29636 sport=0 flags=R seq=7 win=32767 rtt=7.0
ms
len=46 ip=192.168.100.11 ttl=128 id=29637 sport=0 flags=R seq=8 win=32767 rtt=5.0
ms
```

Use the -c option to decide on how many packets to send, in this example i am setting the count option to 5.

```
hping3 -A -c 5 192.168.100.11
```

```
HPING 192.168.100.11 (eth0 192.168.100.11): A set, 40 headers + 0 data bytes
len=46 ip=192.168.100.11 ttl=128 id=30010 sport=0 flags=R seq=0 win=32767 rtt=7.9
ms
len=46 ip=192.168.100.11 ttl=128 id=30011 sport=0 flags=R seq=1 win=32767 rtt=7.0
ms
len=46 ip=192.168.100.11 ttl=128 id=30012 sport=0 flags=R seq=2 win=32767 rtt=7.6
ms
len=46 ip=192.168.100.11 ttl=128 id=30013 sport=0 flags=R seq=3 win=32767 rtt=5.1
ms
len=46 ip=192.168.100.11 ttl=128 id=30014 sport=0 flags=R seq=4 win=32767 rtt=4.0
ms

--- 192.168.100.11 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 4.0/6.3/7.9 ms
```

Create a SYN packet and use the scan mode to scan port 1-1000 on a target.

```
hping3 -S -8 1-1000 192.168.100.11
```

```
Scanning 192.168.100.11 (192.168.100.11), port 1-1000
1000 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   53 domain     : .S..A... 128 55677 64240    46
   88 kerberos   : .S..A... 128 55933 64240    46
  135 epmap      : .S..A... 128 56189 64240    46
  139 netbios-ssn: .S..A... 128 56445 64240    46
  389 ldap       : .S..A... 128 56701 64240    46
  445 microsoft-d: .S..A... 128 56957 64240    46
  464 kpasswd    : .S..A... 128 57213 64240    46
  593            : .S..A... 128 52863 64240    46
  636 ldaps      : .S..A... 128 53375 64240    46
All replies received. Done.
Not responding ports: (199 smux) (202 at-nbp) (203 ) (204 at-echo) (299 ) (300 )
(301 ) (306 ) (307 ) (308 ) (309 ) (312 ) (313 ) (407 ) (500 isakmp) (514 shell)
(723 ) (729 ) (743 ) (761 ) (763 ) (764 ) (766 ) (767 ) (768 ) (769 ) (772 ) (782 )
(783 spamd) (784 ) (790 ) (791 ) (793 ) (794 ) (798 ) (799 ) (802 ) (803 ) (804 )
```

```
(805 ) (808 omirr) (809 ) (810 ) (811 ) (812 ) (813 ) (817 ) (818 ) (819 ) (820 )
(821 ) (822 ) (823 ) (824 ) (825 ) (827 ) (828 ) (829 ) (831 ) (832 ) (833 ) (834 )
(836 ) (837 ) (838 ) (839 ) (840 ) (841 ) (842 ) (843 ) (844 ) (845 ) (846 ) (847 )
(848 ) (849 ) (854 ) (855 ) (858 ) (878 ) (879 ) (880 ) (881 ) (911 ) (912 ) (913 )
(918 )
root@iPhone:~#
```

Send a UDP scan mode to send UDP request on port 80 to a target, if the UDP port is open then you will get a respond back, great to use when the target have blocked ICMP ping.

```
hping3 -2 192.168.100.17 -c 2 -p 80
```

Create a ping packet and use the ICMP mode.

```
hping3 -1 -c 4 192.168.100.11
```

```
HPING 192.168.100.11 (eth0 192.168.100.11): icmp mode set, 28 headers + 0 data
bytes
len=46 ip=192.168.100.11 ttl=128 id=34163 icmp_seq=0 rtt=8.1 ms
len=46 ip=192.168.100.11 ttl=128 id=34164 icmp_seq=1 rtt=5.9 ms
len=46 ip=192.168.100.11 ttl=128 id=34167 icmp_seq=2 rtt=4.0 ms
len=46 ip=192.168.100.11 ttl=128 id=34168 icmp_seq=3 rtt=3.0 ms

--- 192.168.100.11 hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3.0/5.2/8.1 ms
root@iPhone:~#
```

Traceroute to a target using ICM mode and show verbose.

```
hping3 --traceroute -V -1 192.168.100.11
```

```
using eth0, addr: 172.168.200.110, MTU: 1500
HPING google.com (eth0 216.58.211.142): icmp mode set, 28 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=172.168.200.2 name=_gateway
hop=1 hoprtt=3.9 ms
hop=2 TTL 0 during transit from ip=192.168.10.1 name=UNKNOWN
hop=2 hoprtt=2.0 ms
hop=3 TTL 0 during transit from ip=10.33.221.74 name=UNKNOWN
hop=3 hoprtt=8.9 ms
hop=4 TTL 0 during transit from ip=88.129.174.18 name=gbg1.dr8.a3network.se
hop=4 hoprtt=8.9 ms
hop=5 TTL 0 during transit from ip=88.129.128.62 name=gbg1.a7network.se
hop=5 hoprtt=8.0 ms
hop=6 TTL 0 during transit from ip=85.8.9.16 name=gbg1.cr1.a3network.se
hop=6 hoprtt=6.9 ms
hop=7 TTL 0 during transit from ip=85.8.10.20 name=sto2.cr1.a3network.se
```

Traceroute to determined if port 443 is open, set that local traffic is generated from source port 8080

```
hping3 --traceroute -V -S -p 443 -s 8080 google.com
```

```
using eth0, addr: 172.168.200.110, MTU: 1500
HPING google.com (eth0 216.58.211.142): S set, 40 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=172.168.200.2 name=_gateway
hop=1 hoprtt=8.9 ms
len=46 ip=216.58.211.142 ttl=128 id=34374 tos=0 iplen=44
sport=443 flags=SA seq=8 win=64240 rtt=13.8 ms
seq=905581660 ack=1390210946 sum=3cce urp=0

len=46 ip=216.58.211.142 ttl=128 id=34376 tos=0 iplen=44
sport=443 flags=SA seq=9 win=64240 rtt=13.9 ms
seq=277232268 ack=486133387 sum=5a24 urp=0

len=46 ip=216.58.211.142 ttl=128 id=34377 tos=0 iplen=44
```

```
sport=443 flags=SA seq=10 win=64240 rtt=13.0 ms
seq=1939483389 ack=2029365982 sum=8498 urp=0

len=46 ip=216.58.211.142 ttl=128 id=34378 tos=0 iplen=44
sport=443 flags=SA seq=11 win=64240 rtt=12.9 ms
seq=90127368 ack=1561834414 sum=c208 urp=0
```

Use the TTL in tracerout to check load balancing devices IP address.

```
hping3 -S 192.168.100.100 -p 80 -T --ttl 13 --tr-keep-ttl -n
```

Ping a subnet and don’t scan in order, instead randomize the scan. Use the –rand-dest and the interface -I eth0 operators.

```
hping3 -1 192.168.100.x --rand-dest -I eth0
```

Send a ICMP packet to request a timestamp from a target, if the target have the ICMP responses blocked it wont respond to ICMP packets however it might allow response to timestamp request.

```
hping3 -1 192.168.100.17 --icmp-ts -c 3
```

Malicious Commands

DISCLAIMER: This software/tutorial is for educational purposes only. It should not be used for illegal activity. The author is not responsible for its use or the users action, always perform the attacks on your own lab system.

Common used parameters.

- The –flood parameter, activates the fastest packet sending mode
- The -p “destport” parameter, specifies the destination port
- The –spoof parameter, specifies which IP address to be spoofed
- The -rand-source parameter, activates a random source address
- The –interface parameter, used to specify interface

Main attack flags.

- The -S parameter sets the SYN flag
- The -A parameter sets the ACK flag
- The -F parameter sets the FIN flag
- The -R parameter sets the RESET flag
- The -P parameter sets the PUSH flag
- The -U parameter sets the URGENT flag

To start a SYN flood attack run the command bellow

NOTE: When running the commands hping3 will ***not*** show any output, it is working in the background.

```
hping3 --flood -p [DST_PORT] [VICTIM_IP] -S
```

Use hping3 to run a SYN flood attack with a inactive spoofed IP address from the network.

```
hping3 --flood -p [DST_PORT] [VICTIM_IP] -S --spoof [INACTIVE_IP]
```

SYN flood attack with with random source IP address.

```
hping3 --flood -p [DST_PORT] [VICTIM_IP] -S --rand-source
```

ACK flood attack.

```
hping3 --flood -p [DST_PORT] [VICTIM_IP] -A
```

FIN flood attack.

```
hping3 --flood -p [DST_PORT] [VICTIM_IP] -F
```

Conclusion

In this lab we have covered the basic commands you can do in hping3, we assembled TCP and UDP packets and used them to scan networks and discovered devices, as always when doing this kind of scans make sore you are authorized to scan the network and devices you are scanning.

How To Install Apache2 (LAMP) Ubuntu 18.04

How to install apache2 (LAMP) server

LAMP is a acronym of the names in the applications stack, Linux OS, Apache server, MySQL database and PHP programming language.

Together they build a framework to run web applications and host sites like WordPress , all applications in the stack are open source and released on most Linux distributions.

Requirements

1. Ubuntu 18.04 LTS
2. SSH access to the server (Setup SSH)
3. A non root user with sudo privileges (Add sudo user)
4. Enabled firewall (Setup ufw)

5. Configured hostname (Setup hostname)
6. DNS entries

Step 1: Install Apache2

1.1 Lets start by updating the repository's and software packages.

```
sudo apt update
sudo apt upgrade -y
```

1.2 Install the apache2 package.

```
sudo apt install apache2 -y
```

1.3 Confirm installation.

```
sudo systemctl status apache2
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2019-06-19 19:08:58 UTC; 3min 21s ago
 Main PID: 7685 (apache2)
    Tasks: 55 (limit: 2213)
   CGroup: /system.slice/apache2.service
           ├─7685 /usr/sbin/apache2 -k start
           ├─7878 /usr/sbin/apache2 -k start
           └─7879 /usr/sbin/apache2 -k start
```

```
Jun 19 19:08:48 iphone systemd[1]: Starting The Apache HTTP Server...
Jun 19 19:08:58 iphone apachectl[7660]: AH00558: apache2: Could not reliably det
Jun 19 19:08:58 iphone systemd[1]: Started The Apache HTTP Server.

toor@iphone:~$
```

Step 2: Configure Firewall

2.1 Add firewall rules for Apache.

```
sudo ufw allow in "Apache Full"
```

```
Rule added
Rule added (v6)
toor@iphone:~$
```

2.2 Display firewall rules and confirm that the firewall is configured

```
sudo ufw status
```

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
```

```
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)

toor@iphone:~$
```

2.3 Confirm that you can browse to the site

```
http://your_server_ip
```

Step 3: Create the Directory Structure

3.1 Virtual host enables us to have multiple websites on one server, each website can have its own home folder “document root” and a unique SSL certificate ,we can have different security policies for each site, and much more.

Create the directory structure.

```
/var/www/
├── Domain-1.local
│ └── html
├── Domain-2.local
│ └── html
```

Before we create the directory for the site, make sure to configure hostname and hosts file.

I will create a website for my local lab domain ceh.local, the /etc/hots file should the the fallowing entry in it.

```
YOUR-IP-ADDRESS ceh.local
```

In this example i am creating a virtual hosts directory called “ceh.local” and i am using the -p flag to create parent directories.

```
sudo mkdir -p /var/www/ceh.local/html
```

3.2 Assign ownership of the directory to current user

```
sudo chown -R $USER:$USER /var/www/ceh.local/html
```

3.3 Set directory permissions

```
sudo chmod -R 755 /var/www/ceh.local/html
```

3.4 Create a sample index.html file using your favorite editor and add it to the root directory.

```
sudo nano /var/www/ceh.local/html/index.html
```

Add the html code bellow

```
<!DOCTYPE html>
```

```
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Welcome to ceh.local</title>
  </head>
  <body>
    <h1>Success! ceh.local home page</h1>
  </body>
</html>
```

Exit & Save

Step 4: Configure Virtual Hosts File

Apache virtual hosts configuration files are stored in.

- /etc/apache2/sites-enabled
- /etc/apache2/sites-available

Lets add a Virtual Hosts configuration file for domain1.local

```
sudo nano /etc/apache2/sites-available/ceh.local.conf
```

Add the fallowing lines and modify them to your site.

```
<VirtualHost *:80>
    ServerName ceh.local
    ServerAlias www.ceh.local
    ServerAdmin admin@ceh.local
    DocumentRoot /var/www/ceh.local/html

    <Directory /var/www/ceh.local/html>
        Options -Indexes +FollowSymLinks
```

```
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/ceh.local-error.log
    CustomLog ${APACHE_LOG_DIR}/ceh.local-access.log combined
</VirtualHost>
```

Exit & Save

4.2 Enable the Virtual Hosts configuration file file with a2ensite

```
sudo a2ensite ceh.local.conf
```

```
Enabling site domain1.local.
To activate the new configuration, you need to run:
  systemctl reload apache2
toor@iphone:~$
```

4.3 Disable the default site defined in 000-default.conf

```
sudo a2dissite 000-default.conf
```

```
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
toor@iphone:~$
```

4.4 Test the configuration file for any syntax errors

```
sudo apache2ctl configtest
```

```
Syntax OK
toor@iphone:/var/www/domain1.local/html$
```

4.5 Restart the Apache service for the changes to take effect

```
sudo systemctl restart apache2
```

4.6 Confirm that the service have started

```
sudo systemctl status apache2
```

4.7 launch a web browser and start browsing ceh.local

Step 5: Install MySQL

5.1 To install MySQL run

```
sudo apt install mysql-server -y
```

5.2 Verify that MySQL service is running

```
sudo systemctl status mysql
```

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Fri 2019-06-21 11:59:29 UTC; 8s ago
 Main PID: 17807 (mysqld)
    Tasks: 27 (limit: 2322)
   CGroup: /system.slice/mysql.service
           └─17807 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi

Jun 21 11:59:28 srv6 systemd[1]: Starting MySQL Community Server...
Jun 21 11:59:29 srv6 systemd[1]: Started MySQL Community Server.
lines 1-10/10 (END)
toor@srv6:~$
```

5.3 The default MySQL user “root” have a blank password., we need to secure the MySQL server and remove the default database.

```
sudo mysql_secure_installation
```

Then enter the following security questions

- VALIDATE PASSWORD plugin = NO
- Set root password and confirm
- Remove anonymous users? = YES
- Disallow root login remotely? = NO
- Remove test database and access to it? = YES
- Reload privilege tables now? = YES

5.4 Start from MySQL Server 5.7, if you do not provide a password to root user during the installation, it will use auth_socket plugin for authentication.

If we want to configure a password authentication, we need to run the following commands.

```
sudo mysql
```

5.5 Display current configuration

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

5.6 Alter authentication_string for the root user

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'changeme';
```

5.7 Flush the privileges and update the changes

```
FLUSH PRIVILEGES;
```

5.8 Display current configuration

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

5.9 Exit from the mysql prompt:

```
exit
```

Step 6: Install PHP

6.1 PHP is a server side scripting language used to generate dynamic content on websites and applications.

Install PHP (default version is PHP 7.2) and some of the basic modules for web deployments.

```
sudo apt install php php-common php-mysql php-gd php-cli -y
```

6.2 Create info.php file in the Apache root document folder.

Usually, the apache2 root document folder will be /var/www/html/ or /var/www/ in most Debian based Linux distributions.

If you have followed the guide then the the file should be in /var/www/ceh.local/html/

```
sudo nano /var/www/ceh.local/html/info.php
```

Add the following lines

```
<?php
phpinfo();
?>
```

Exit and save

6.3 Restart Apache

```
sudo systemctl restart apache2
```

6.4 Test PHP page, open a web browser and enter “http://ceh.local/info.php”

Step 7: Install PhpMyAdmin

7.1 With phpMyAdmin we can administrating MySQL from a web browser, start by adding the needed repository.

```
sudo add-apt-repository universe
```

7.2 Install phpmyadmin

```
sudo apt install phpmyadmin -y
```

Go through the package installation process, select Apache2 and configure a password for the phpmyadmin database.

7.3 Restart Apache

```
sudo systemctl restart apache2
```

Conclusion

We have installed installing Apache2, MySQL, PHP and Virtual Hosts on a Ubuntu server and secured it.

For administration of the website we have installed phpmyadmin.

How To Install a FTP Server On Ubuntu Server 18.04

VsFTPD ” Very Secure FTP Daemon”

VsFTPD “very secure FTP daemon” is an open source FTP server for Linux systems, in this quick guide we will install VsFTPD on a Ubuntu server and secure the FTP server with SSL/TLS. Please visit the official website of VsFTPD if you need more information about the application.

Requirements

- Ubuntu Server 18.04
- User with sudo privileges.
- Static IP address
- Configured firewall
- Server connected to internet

For more information on how to create a sudo and configure a static IP please see the quick guides Create Sudo User , Set Static IP address and Configure Ubuntu Firewall.

Install VsFTPD

Vsftpd is available in Ubuntu 18.04 default repository and do not require any extra pre configuration.

Run the following command to install Vsftpd

```
sudo apt-get install vsftpd -y
```

Wait for the application to finish installing, start the Vsftpd service and enable it to start on boot.

```
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
```

Verify that the VsFTPD is up and running.

```
sudo systemctl status vsftpd
```

```
root@iphone:~# sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset:
enabled)
   Active: active (running) since Sat 2019-06-08 18:17:39 UTC; 2min 54s ago
 Main PID: 2311 (vsftpd)
    Tasks: 1 (limit: 2214)
   CGroup: /system.slice/vsftpd.service

Jun 08 18:17:39 iphone systemd[1]: Starting vsftpd FTP server...
Jun 08 18:17:39 iphone systemd[1]: Started vsftpd FTP server.
```

Configure The Firewall

We need to open port 20 and 21 for active FTP and ports 40000-50000 for passive FTP.

```
sudo ufw allow 20/tcp
```

```
sudo ufw allow 21/tcp
```

```
sudo ufw allow 40000:50000/tcp
```

Display the firewall rules.

```
sudo ufw status
```

```
root@iphone:~# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)

root@iphone:~#
```

Create FTP User

Create a low privileges user that can be used to access the FTP server.

When prompted enter password and user information for the user.

```
sudo adduser ftpuser
```

```
root@iphone:~# sudo adduser ftpuser
Adding user `ftpuser' ...
Adding new group `ftpuser' (1001) ...
Adding new user `ftpuser' (1001) with group `ftpuser' ...
Creating home directory `/home/ftpuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ftpuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@iphone:~#
```

Create a FTP Directory For The FTP User

First we want to create a FTP folder for the ftpuser.

```
sudo mkdir /home/ftpuser/ftp
```

Next we want to set the folder ownership.

```
sudo chown nobody:nogroup /home/ftpuser/ftp
```

Remove write permissions to the ftp folder.

```
sudo chmod a-w /home/ftpuser/ftp
```

Verify FTP folder permissions.

```
sudo ls -la /home/ftpuser/ftp
```

```
root@iphone:/home/ftpuser# sudo ls -la /home/ftpuser/ftp
total 8
dr-xr-xr-x 2 nobody  nogroup 4096 Jun  8 19:01 .
drwxr-xr-x 3 ftpuser ftpuser 4096 Jun  8 19:02 ..
root@iphone:/home/ftpuser#
```

Create a directory for file uploads and assign ownership to ftpuser.

```
sudo mkdir /home/ftpuser/ftp/files
sudo chown ftpuser:ftpuser /home/ftpuser/ftp/files
```

Verify the new folder permission.

```
sudo ls -la /home/ftpuser/ftp
```

```
root@iphone:/home/ftpuser/ftp/files# sudo ls -la /home/ftpuser/ftp
total 12
dr-xr-xr-x 3 nobody  nogroup 4096 Jun  8 19:08 .
drwxr-xr-x 3 ftpuser ftpuser 4096 Jun  8 19:02 ..
drwxr-xr-x 2 ftpuser ftpuser 4096 Jun  8 19:08 files
```

```
root@iphone:/home/ftpuser/ftp/files#
```

Create and add txt file to the files folder we created in the step above.

```
echo "Test create txt file" | sudo tee /home/ftpuser/ftp/files/txt01.txt
```

Configuring VsFTPD

Edit the VsFTPD configuration file vsftpd.conf

```
cd etc/
sudo nano vsftpd.conf
```

```
##
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
##
```

Enable uploading to the FTP server by uncomment the write_enable parameter.

```
##
write_enable=YES
##
```

Prevent the FTP users from accessing files or to run commands outside there directory by uncomment the chroot_local_user=YES parameter.

```
##
chroot_local_user=YES
##
```

Scroll down to the bottom and add the the port range for passive FTP.

```
pasv_min_port=40000
pasv_max_port=50000
```

Previously we created a ftp/file directory and folder for the ftpsuser, now we need to configure VsFTPD to log the ftpuser to home ftp directory we created.

Add the line bellow.

```
user_sub_token=$USER
local_root=/home/$USER/ftp
```

Restart the daemon.

```
sudo systemctl restart vsftpd
```

Testing The FTP Access

You can use a ftp client like FileZilla or the command line to confirm that you can access the ftp server and that you can see the txt file you created in the ftpuser ftp directory.

I am using the command line on the FTP server in this example to confirm that i can access the FTP and that i can download the txt01.txt.

```
root@iphone:/# ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 3.0.3)
Name (127.0.0.1:toor): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Lets confirm that we can change to the “files” directory.

```
ls
cd files
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Jun 08 19:17 files
226 Directory send OK.
ftp> cd files
250 Directory successfully changed.
ftp>
```

List the directory and use the get command to transfer the test file.

```
ls
get txt01.txt
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0              21 Jun 08 19:16 txt01.txt
226 Directory send OK.
ftp> get txt01.txt
local: txt01.txt remote: txt01.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for txt01.txt (21 bytes).
226 Transfer complete.
21 bytes received in 0.00 secs (259.5926 kB/s)
ftp>
```

Upload the file with a new name to test users write permissions. To upload a file we use the put command.

```
put txt01.txt txt01-upload.txt
```

```
ftp> put txt01.txt txt01-upload.txt
local: txt01.txt remote: txt01-upload.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
21 bytes sent in 0.00 secs (1.0541 MB/s)
ftp>
```

Listing the files directory should show two files now.

```
ls
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-------    1 1001     1001           21 Jun 08 21:00 txt01-upload.txt
-rw-r--r--    1 0        0              21 Jun 08 19:16 txt01.txt
226 Directory send OK.
ftp>
```

(Optional) Secure The FTP Server With TLS

Lets start adding the firewall rule for TLS traffic, add port 990 to the firewall access list.

```
sudo ufw allow 990/tcp
```

```
root@iphone:/# sudo ufw allow 990/tcp
Rule added
Rule added (v6)
root@iphone:/#
```

Confirm firewall status

```
sudo ufw status
```

```
root@iphone:/# sudo ufw status
```

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)

root@iphone:/#
```

Create a OpenSSL certificate

Create a OpenSSL certificate for TLS/SSL encryption, first make a directory where you can save the certificate.

```
sudo mkdir /etc/ftpcert
```

Now we will create a new certificate, use the -days flag to make it valid for two years, 730 days. Next set the bit value of the RSA key, i am running with a 2048-bit RSA key.

Type in the -keyout and the -out flag, the flags will set the key values for the private key and the certificate.

NOTE: Setting both flags with the same value will create both the private key and the certificate in the same file.

You will be asked to enter details like country, state, etc. You don’t have to fill in the information. Just keep pressing ENTER for defaults.

```
sudo openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout
/etc/ftpcert/vsftpd.pem -out /etc/ftpcert/vsftpd.pem
```

Confirm that the private key and the certificate is the ftpscert directory.

```
root@iphone:/# cd /etc/ftpcert/
root@iphone:/etc/ftpcert# ls
vsftpd.pem
root@iphone:/etc/ftpcert#
```

Next we need to configure vsftpd to allow TLS/SSL traffic and point out the directory of the private key and the certificate , open the vsftpd configuration file with a editor.

```
cd etc/
sudo nano vsftpd.conf
```

Scroll down until you find the rsa parameters, Comment them out and replace them with new lines that points out the privet key and the certificate we created.

```
##
# rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
# rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
##

rsa_cert_file=/etc/ftpcert/vsftpd.pem
rsa_private_key_file=/etc/ftpcert/vsftpd.pem
```

Configure FTP connections to use use SSL/TLS, change the ssl_enable=NO parameter to YES.

```
##
ssl_enable=YES
##
```

Now add the following lines to deny anonymous connections over SSL and to require SSL for logging and transferring data.

```
##
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
##
```

Configure the server to use the TLS protocol

```
##
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
##
```

Last configure SSL reuse parameter to NO due that it can have conflicts with FTP clients, next we need to use high encryption cipher suite, which means that the key lengths is equal to or greater than 128 bits.

Paste thee lines below.

```
##
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

```
##
```

The configuration should have the below entry’s configured.

```
#
rsa_cert_file=/etc/ftpcert/vsftpd.pem
rsa_private_key_file=/etc/ftpcert/vsftpd.pem
#
#
ssl_enable=YES
#
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
##
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
#
require_ssl_reuse=NO
ssl_ciphers=HIGH
#
```

Restart the VsFTPD to load the new configuration.

```
sudo systemctl restart vsftpd
```

Confirm FTP TLS Configuration

Download a FTP client like FileZilla, you grab the FileZilla client from the official site https://filezilla-project.org/

Run and install the FTP client, when connecting to the FTP server use “Require explicit

FTP over TLS”. If everything is configured correct you should be grated with a pop up windows that displays the server certificate we created.

If you try to connect to the FTP server with just plain FTP protocol, you will get an error and you wont be able to connect to the server.

```
Status: Connection established, waiting for welcome message...
Response:       220 (vsFTPd 3.0.3)
Command:        USER ftpuser
Response:       530 Non-anonymous sessions must use encryption.
Error:  Could not connect to server
```

Conclusion

In this quick guide we have installed a FTP server on Ubuntu18.04.02, generated a certificate with OpenSSL and secured the server connectivity with TLS.

How To Scan a Network With Nmap

How To Scan With Nmap

Nmap is a great tool to learn, the application have the ability to scan and map networks and much more, it is a great tool for everybody that works in IT.

It is the first tool i use when i want troubleshot, we can do regular ping or a ping sweeps that scans a range of the subnet or the whole subnet.

The application also offers host discovery, port discovery, operating system version discovery, MAC address, services, exploit and vulnerability detection.

Another great tool to use while learning nmap is Wireshark, It is highly recommended to run Wireshark wile using nmap, following the flow of network traffic will help you analyze and visuals the scans.

We will try some of the popular scanning method that can be used with nmap.

This guide is just meant to give you high level understanding on how to use the different scanning techniques.

Please don’t scan networks or host you are not authorized to do. The networks and hosts scanned in the guide is my home lab.

If you want a more in-depth explanation on how you can use nmap and the switches, i recommend that you read “The Official Nmap Project Guide to Network Discovery and Security Scanning”.

Save Output To Txt/Xml File

Description	Command	Example
Save output to file	nmap -oN [file.txt] [Target]	nmap -oN file.txt 192.168.100.11
Save output as XML	nmap -oX [file.xml] [Target]	nmap -oX file.xml192.168.100.11
Save in all formats	nmap -oA [file] [Target]	nmap -oA file 192.168.100.11

Basic Scanning

Description	Command	Example
Scan a single host	nmap [Target]	nmap 192.168.100.100
Scan multiple targets	nmap [Target1, Target2]	nmap 192.168.100.10,192.168.100.100
Scan a range of IP address	nmap [IP Range]	nmap 192.168.100.10-99
Scan a Class C subnet	nmap [IP/CDIR]	nmap 192.168.100.0/24
Resolve FQDN	nmap [FQDN]	nmap www.eaxmple.com

Quick Scans

Description	Command	Example
Ping scan	nmap -sP [Target]	nmap -sP 192.168.100.11
Ping Scan – disable port scanining	nmap -sn [Target]	nmap -sn 192.168.100.0/24

-sP switch can be used when you want to make a quick ping, the host or hosts will replay to ICMP ping packets.

```
nmap -sP 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-26 21:05 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0010s latency).
Nmap done: 1 IP address (1 host up) scanned in 5.84 seconds
```

The **-sn** switch is used to to sweep a network without doing any port scans.

```
nmap -sn 192.168.100.0/24

Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-02 00:02 W. Europe Daylight Time
Nmap scan report for 192.168.100.1
Host is up (0.0010s latency).
Nmap scan report for srv1.online-it.nu (192.168.100.11)
Host is up (0.0020s latency).
Nmap scan report for 192.168.100.13
Host is up (0.0010s latency).
Nmap scan report for srv7.home.local (192.168.100.17)
Host is up (0.0011s latency).
Nmap scan report for 192.168.100.100
Host is up (0.0013s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 10.82 seconds
```

Banner Grabbing & Service Detection

Description	Command	Example
Detect OS	nmap -O [Target]	nmap -O 192.168.100.11
Detect OS & Services	nmap -A [Target]	nmap -A 192.168.100.11
Detect Services	nmap -sV [Target]	nmap -sV 192.168.100.11

The **-O** switch scans for operating system details. This type of scan can be used to identify the operating system of the scanned host and the services the host is running.

```
nmap -O 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-26 21:12 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.00032s latency).
Not shown: 988 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
Device type: general purpose
Running: Microsoft Windows 2016
OS CPE: cpe:/o:microsoft:windows_server_2016
OS details: Microsoft Windows Server 2016 build 10586 - 14393
Network Distance: 2 hops

OS detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.96 seconds
```

Port Scans Types

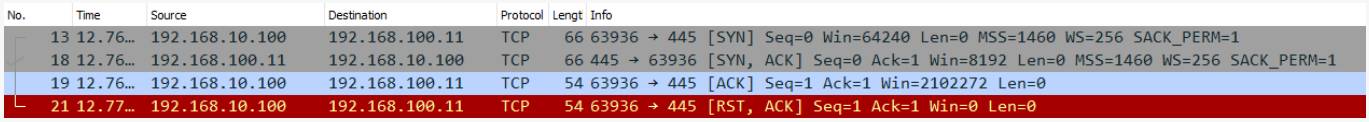
Description	Command	Example
Scan a single Port	nmap -p [Port] [Target]	nmap -p 80 192.168.100.11
Scan a range of ports	nmap -p [Port-Port] [Target]	nmap -p 20-99 192.168.100.11
Scan the first 100 ports	nmap -F [Port] [Target]	nmap -F 192.168.100.11
Scan using TCP Handshake	nmap -sT [Target]	nmap -sT 192.168.100.11
Scan using TCP SYN (Stealth)	nmap -sS [Target]	nmap -sS 192.168.100.11
Scan UDP port	nmap -sU [Target]	nmap -sU 192.168.100.11

The **-sT** switch creates a full TCP handshake with the target. This is considered more accurate than SYN scan but is slower and can be easy detected by firewalls and IDS.

```
nmap -sT 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-26 21:18 W. Europe Daylight Time

Nmap scan report for 192.168.100.11
Host is up (1.0s latency).
Not shown: 986 closed ports
PORT     STATE    SERVICE
25/tcp   filtered smtp
53/tcp   open     domain
88/tcp   open     kerberos-sec
110/tcp  filtered pop3
135/tcp  open     msrpc
139/tcp  open     netbios-ssn
389/tcp  open     ldap
445/tcp  open     microsoft-ds
464/tcp  open     kpasswd5
593/tcp  open     http-rpc-epmap
636/tcp  open     ldapssl
3268/tcp open     globalcatLDAP
3269/tcp open     globalcatLDAPssl
3389/tcp open     ms-wbt-server
```

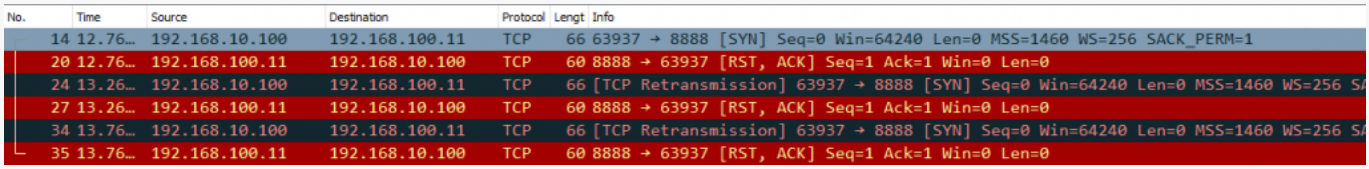
```
Nmap done: 1 IP address (1 host up) scanned in 219.83 seconds
```

Analysing the scan in wireshark we can see that the open port is responding to the handshake.



No.	Time	Source	Destination	Protocol	Lengt	Info
13	12.76…	192.168.10.100	192.168.100.11	TCP	66	63936 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
18	12.76…	192.168.100.11	192.168.10.100	TCP	66	445 → 63936 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
19	12.76…	192.168.10.100	192.168.100.11	TCP	54	63936 → 445 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
21	12.77…	192.168.10.100	192.168.100.11	TCP	54	63936 → 445 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

If the port is closed on the host, then the target host will respond with a RST+ACK packets.



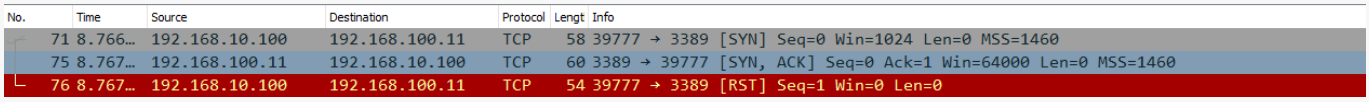
No.	Time	Source	Destination	Protocol	Lengt	Info
14	12.76…	192.168.10.100	192.168.100.11	TCP	66	63937 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
20	12.76…	192.168.100.11	192.168.10.100	TCP	60	8888 → 63937 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24	13.26…	192.168.10.100	192.168.100.11	TCP	66	[TCP Retransmission] 63937 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SA
27	13.26…	192.168.100.11	192.168.10.100	TCP	60	8888 → 63937 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
34	13.76…	192.168.10.100	192.168.100.11	TCP	66	[TCP Retransmission] 63937 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SA
35	13.76…	192.168.100.11	192.168.10.100	TCP	60	8888 → 63937 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

The **-sS** switch sends only a TCP SYN packet and waits for a TCP ACK. If it receives an ACK on the specific probed port then it response with a RST packet, in this way the scan can be undetected by the firewall. If the scanned port is closed on the target host, then target will only respond with a RST packet.

```
nmap -sS 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-26 21:24 W. Europe Daylight Time
Nmap scan report for 192.168.100.11

Host is up (0.0013s latency).
Not shown: 988 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
```

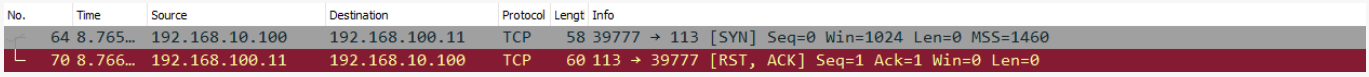
```
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 6.31 seconds
```

Analysing the packets in wireshark we can see that we first send a SYN packet to the scanned port on the target host, if it port is opened the target will response wit a SYN+ACK packet and we respond back with a RST packet.



No.	Time	Source	Destination	Protocol	Lengt	Info
71	8.766…	192.168.10.100	192.168.100.11	TCP	58	39777 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
75	8.767…	192.168.100.11	192.168.10.100	TCP	60	3389 → 39777 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460
76	8.767…	192.168.10.100	192.168.100.11	TCP	54	39777 → 3389 [RST] Seq=1 Win=0 Len=0

If the port is closed on the scanned target the we will get a RST+ACK back.



No.	Time	Source	Destination	Protocol	Lengt	Info
64	8.765…	192.168.10.100	192.168.100.11	TCP	58	39777 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
70	8.766…	192.168.100.11	192.168.10.100	TCP	60	113 → 39777 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

The **-sU** switch will scan after UDP ports, UDP is a connectionless protocol, UDP packets dose not have any ACK flag set, the UDP protocol don’t require the reviser to confirm that he revised a UDP packet.

If the there is a firewall enabled on the host or on the network you will get a response back “open|filtered ports” and a list of ports that are blocked by the firewall.

```
nmap -sU 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 19:58 W. Europe Daylight Time

Nmap scan report for 192.168.100.11
Host is up (0.0016s latency).
Not shown: 997 open|filtered ports
PORT    STATE SERVICE
53/udp  open  domain
123/udp open  ntp
389/udp open  ldap
```

```
Nmap done: 1 IP address (1 host up) scanned in 17.27 seconds
```

If the firewall is disabled then they will be no response back.

Inverse Scans

Description	Command	Example
Xmas scan	nmap -sX [Target]	nmap -sX 192.168.100.11
FIN scan	nmap -sF [Target]	nmap -sF 192.168.100.11
TCP Null scan	nmap -sN [Target]	nmap -sN 192.168.100.11
ACK scan	nmap -sA [Target]	nmap -sA 192.168.100.11

The **-sX** switch is called a Xmas Scan, when you scan a network or a target host with Xmax scan, the xmas scan sends a packet that contains multiple flags, the packet contains the URG, PSH & FIN flags. If the host have closed ports, it will respond with a single RST packet. If the ports are open on the host, then the host will respond as an open ports. Modern operating systems, firewalls and IDS drops this kind of packets if they are properly configured.

We will run the xmax scan against a windows server with firewall enabled.

```
nmap -sX 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 17:07 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0010s latency).
All 1000 scanned ports on 192.168.100.11 are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 27.62 seconds
```

Observe the line “All 1000 scanned ports on 192.168.100.11 are open|filtered” the output is showing that all scanned ports are “open|filtered”. This means that the firewall are enabled on the target host.

Lets try the same scan but this time we will disable the firewall on our target host.

```
nmap -sX 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 17:13 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0012s latency).
All 1000 scanned ports on 192.168.100.11 are closed

Nmap done: 1 IP address (1 host up) scanned in 6.78 seconds
```

Now we get “All 1000 scanned ports on 192.168.100.11 are closed” this indicates that the firewall disabled.

The **-sF** switch scans the the host with a FIN scan, a FIN scan sends a packet with only the FIN flag set, this allows the packet to pass the firewall. If the port is open you will not get any respond, if the port is closed the target will respond with a RST packet.

When the firewall is enabled on the target the output will have a “open|filtered” response.

```
nmap -sF 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 17:51 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0010s latency).
All 1000 scanned ports on 192.168.100.11 are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 27.19 seconds
```

If the firewall is disabled on the target the output will have a “are closed” response.

```
nmap -sF 192.168.100.11
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 18:06 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0019s latency).
All 1000 scanned ports on 192.168.100.11 are closed

Nmap done: 1 IP address (1 host up) scanned in 6.29 seconds
```

The **-sN** switch will scan the target with a NULL scan, the scan sends a packet without any flags set. if the NULL packet is sent to an open port, the will be no response back. If the NULL packet is sent to a close port, it will respond with a RST packet. This type of scan is easy to detect due that there are no reason to send a TCP packet without a flag.

When using the NULL scan the target will respond similar to the FIN and Xmaz scans.

The **-sA** switch send a packet with the ACK flag set when scanning a host, when the target receive the ACK packet it will replay with a RST packet. if the port is closed and the firewall is enabled the firewall will block the target host response and there will be no response back.

Observe the output in namp when the firewall is enabled.

```
nmap -sA 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 19:36 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0010s latency).
All 1000 scanned ports on 192.168.100.11 are filtered

Nmap done: 1 IP address (1 host up) scanned in 27.58 seconds
```

If the firewall is enabled the “All 1000 scanned ports on 192.168.100.11 are filtered” line will comeback with the “**filtered**” value. The “filtered” response shows that a firewall is enabled in the system.

Running the same command against a target with a disabled firewall, the output will have a different value.

```
nmap -sA 192.168.100.11

Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 19:39 W. Europe Daylight Time
Nmap scan report for 192.168.100.11
Host is up (0.0013s latency).
All 1000 scanned ports on 192.168.100.11 are unfiltered

Nmap done: 1 IP address (1 host up) scanned in 7.23 seconds
```

The response back on the “All 1000 scanned ports on 192.168.100.11 are unfiltered” is coming back with the “unfiltered” value. The response back means that there are no firewall enabled on the target.

Firewall Evasion

Description	Command
Idle zombie scan	nmap -sI [zombie] [target]
Use a decoy	nmap -D RND: [number] [target]
Fragment packets	nmap -f [target]
Specify MTU	nmap —mtu [MTU] [target]
Randomize scan order	nmap —randomize-hosts [target]
Send bad checksums	nmap —badsum [target]
Specify source port	nmap —source-port [port] [target]
Spoof MAC Address	nmap —spoof-mac [MAC|0|vendor] [target]

The **-sI** is called a Idle scan or a zombie scan is a stealth technique, when using the a zombie scan packets revised on the scanned host cant be traced back the sender, all network traffic to the target host are going trough a second host on the network called “zombie”.

For a more detail explanation on how the idle scan work i recommend to read the official nmap documentation at https://nmap.org/book/idlescan.html

The **-f** switch is used to fragment probes into 8-byte packets, the scan will split the TCP header up to several packet, it is a very effective way to hide thee and make it harder for intrusion detection systems to the detect the scans.

The **-D** switch is used to hide port scans by using one or more decoys IP address,the network traffic on the scanned host will appear coming from the decoys IP address.

The **–source-port** switch is used to manually specify the source port number of a probe.

The **–-randomize-hosts** switch is used to randomize the scanning order of the specified ping sweap or a range scan.

Script Engines

Description	Command
Run script	nmap –script [script.nse] [target]
Run scripts	nmap –script [expression] [target
Run scripts by category	nmap –script [cat] [target]
Run multiple scripts categories	nmap –script [cat1,cat2,cat3] [target]
Update script database	nmap –script-updatedb
Script categories	all
	discovery
	default
	auth
	external
	malware

Description	Command
	vuln
	intrusive
	safe

Useful scans

Find Information about IP address

```
nmap --script=asn-query,whois,ip-geolocation-maxmind [target]
```

Detect Heart bleed SSL vulnerability

```
nmap -sV -p 443 --script=ssl-heartbleed [target]
```

Scan for DDOS reflection UDP services

```
nmap –sU –A –PN –n –pU:19,53,123,161 –script=ntp-monlist,dns-recursion,snmp-
sysdescr [target]
```

Scan HTTP Service

Get page titles

```
nmap --script=http-title [target]
```

Get HTTP headers

```
nmap --script=http-headers [target]
```

Recommended sites

https://highon.coffee/blog/nmap-cheat-sheet/

Conclusion

We have looked into some of the scanning techniques we can use with nmap.

Check out the Ethical Hacking notes for more Kali Linux quick guides.