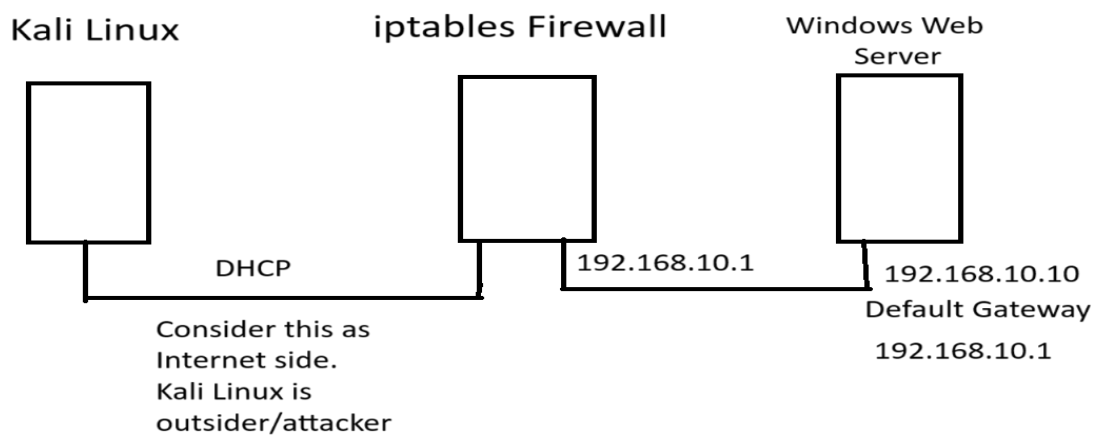


Configure port forwarding in iptables and protecting against Network Threats

This lab requires 3 virtual machines as shown below.



Put all virtual machine network cards in NAT mode. Make sure VMWare DHCP server is working.

For firewall VM Keep the first network card on DHCP. This network card should be connected to Internet connection. Assign a fixed IP address to the second network card. Here we assign 192.168.10.1 IP address with default subnet mask. This IP address will be given as the default gateway to the clients. Set a hostname for this machine.

The Windows virtual machine is given the IP address 192.168.10.10. The default gateway given is 192.168.10.1. DNS address for Windows should be given as per your network.

The Kali Linux virtual machine will be on DHCP. Thus no configuration is required.

Install iptables

First configure the Firewall virtual machine.

The default firewall in RedHat based systems is FirewallD.

You need to install iptables.

But first you need to stop FirewallD and disable it.

```
sudo systemctl stop firewalld
```

```
sudo systemctl disable firewalld
```

```
sudo systemctl mask firewalld
```

Now Install iptables.

```
sudo yum install iptables-services iptables-utils iptables-dev -y
```

Now start and enable the iptables service.

```
Sudo systemctl start iptables
```

verify with

```
sudo systemctl status iptables
```

Now enable the service for Auto Start

```
sudo systemctl enable iptables
```

Enable IP forwarding.

By default the Linux kernel will not forward the IP packets received on one network interface to other network interface.

But the network firewall will need to take packets from the client and forward then to Internet. Thus we need to enable IP forwarding.

Check if IP forwarding is enabled by default.

```
sudo cat /proc/sys/net/ipv4/ip_forward
```

OR

```
sudo sysctl net.ipv4.ip_forward
```

if the value is 0 then IP forwarding is disabled and if it is 1 then IP forwarding is enabled.

To enable IP forwarding immediately, use any one of the following command.

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

OR

```
sudo sysctl net.ipv4.ip_forward=1
```

Check with

```
sudo cat /proc/sys/net/ipv4/ip_forward
```

OR

```
sudo sysctl net.ipv4.ip_forward
```

To permanently enable IP forwarding (even after the system restarts), edit the sysctl.conf file.

```
sudo vim /etc/sysctl.conf
```

And add following line.

```
net.ipv4.ip_forward = 1
```

Save the file.

Enable NAT rule

Now you need to enable the NAT rule so that all LAN clients will go out with the public IP. This is necessary to provide Internet to LAN clients as they are using private IP addresses.

First find out the name of the first network adapter.

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:36:ff:d0 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.234.132/24 brd 192.168.234.255 scope global dynamic noprefixroute ens160
        valid_lft 1755sec preferred_lft 1755sec
    inet6 fe80::20c:29ff:fe36:ffd0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:36:ff:da brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.220.143/24 brd 192.168.220.255 scope global dynamic noprefixroute ens192
        valid_lft 1755sec preferred_lft 1755sec
    inet6 fe80::fbf5:7ee9:d013:b5ee/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost ~]#
```

Here the name of the first network interface is ens160.

The LAN side network address is 192.168.220.143. This is the address of the second network interface. Thus the LAN side network is 192.168.220.0/24.

Now add the NAT rule as below.

```
sudo iptables -t nat -A POSTROUTING -s 192.168.220.0/24 -o ens160 -j MASQUERADE
```

Check with

```
sudo iptables -t nat -L
```

```
[root@localhost ~]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  192.168.220.0/24      anywhere
[root@localhost ~]#
```

Allow IP Forwarding in iptables.

If you check with

```
sudo iptables -L
```

You will find a default rule in the FORWARD chain of the filter table which blocks everything. Thus we need to add an allow rule ahead of it for our network or delete it.

```
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@localhost ~]#
```

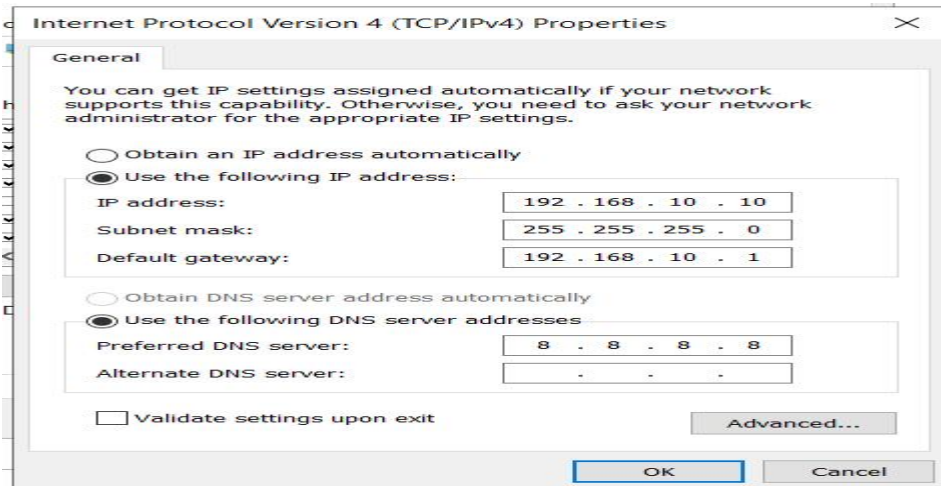
Here we delete this rule.

```
sudo iptables -D FORWARD 1
```

Now this system is configured as a network router and all clients will be able to access Internet.

Configure Windows machine

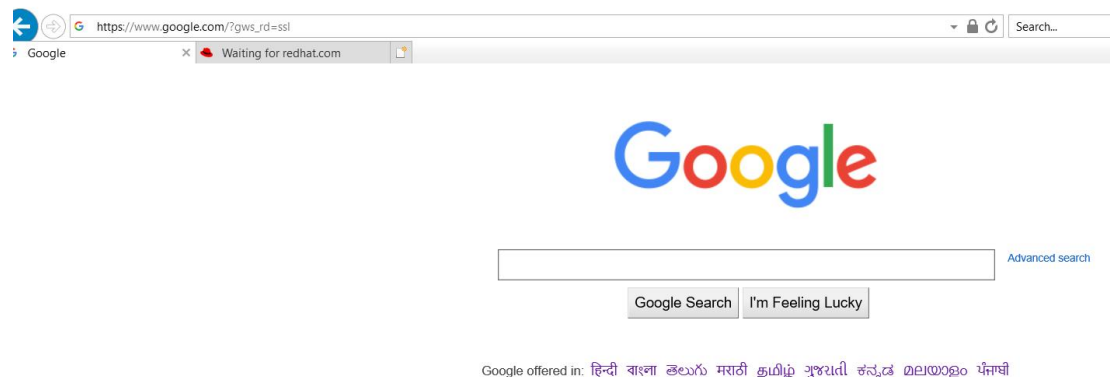
Login as administrator to Windows machine. First configure the IP address as shown in the initial diagram.



Make sure the IE Enhanced Security Configuration is turned off.

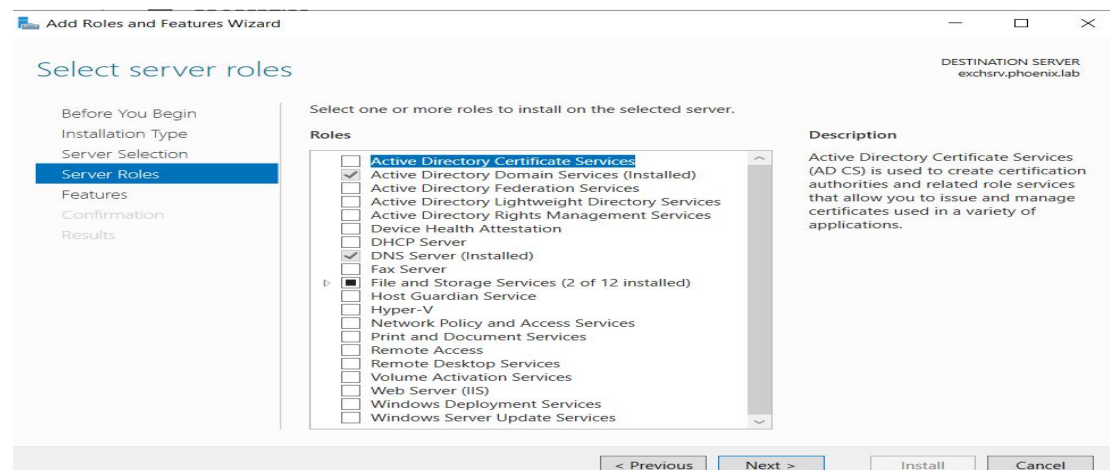
IE Enhanced Security Configuration Off

Open the browser and check if you are able to get the Internet.

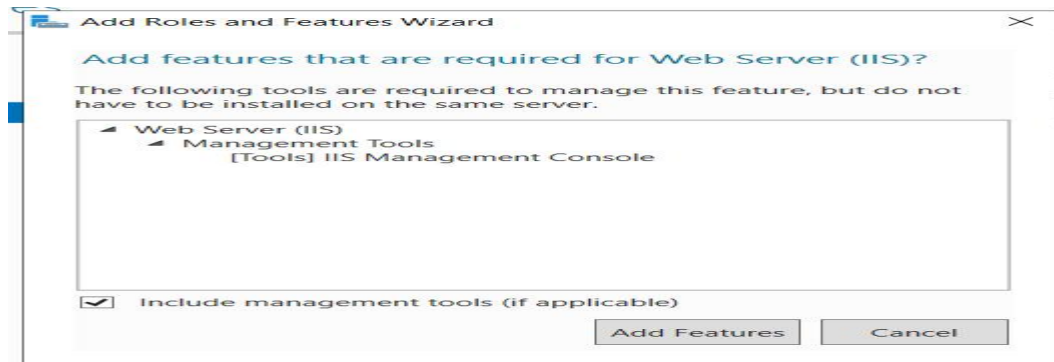


Install IIS web server on Windows.

Open Server Manager. Click Manage. Click Add Roles and Features. Click Next till you get to the following screen.



Scroll down and select Web Server (IIS) checkbox.



Click Add Features on the above window displayed.

Click Next on all screens to keep default options. Finally click Install button to Install IIS Web Server.

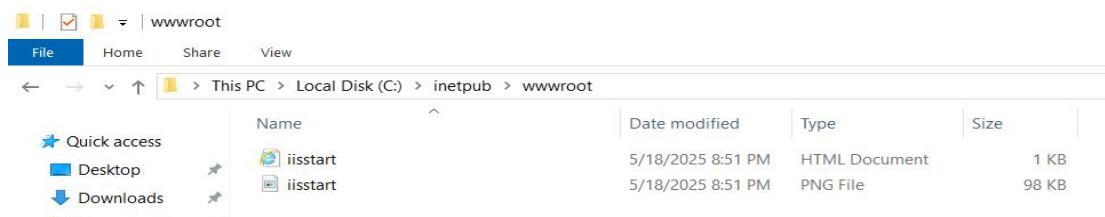
You will get following message, once the installation completes.



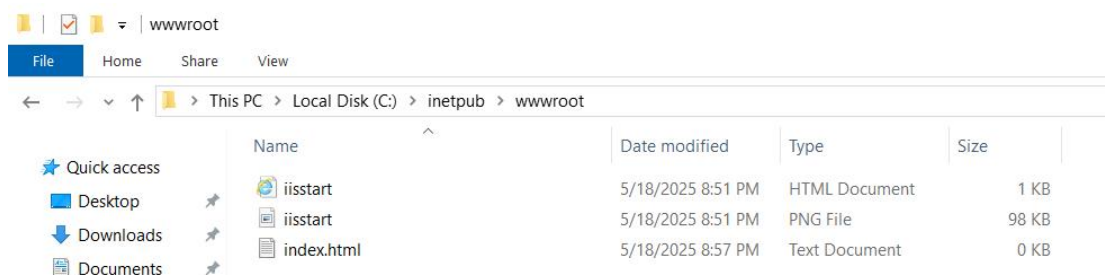
Click Close to close the window.

Now create your own index.html page. By default IIS uses the c:\inetpub\wwwroot directory. Thus you have to create the index.html page inside this directory.

Go to the c:\inetpub\wwwroot directory.



Right click below. Select New. Select Text Document. Give the name as index.html.



Double click on the file and type the message that you want to display on the website.

index.html - Notepad

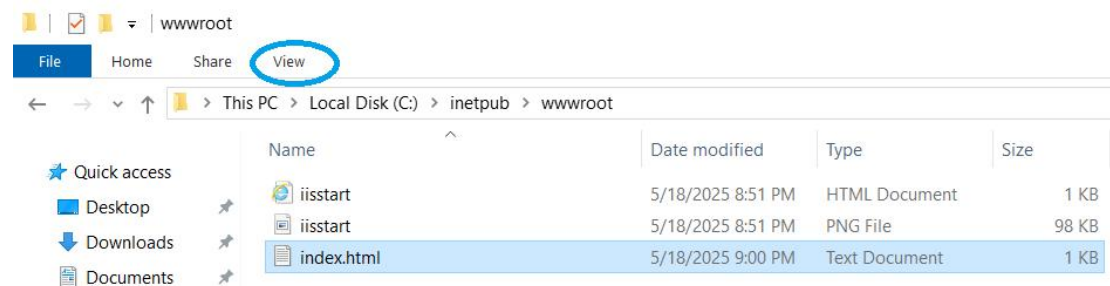
File Edit Format View Help

This website is hosted on 192.168.10.1 on a Windows Server.

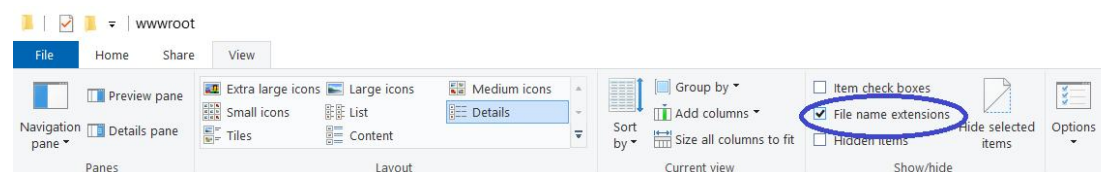
Save the file.

This file name is actually index.html.txt. Windows by default will not display the file extension. You need to rename this to index.html.

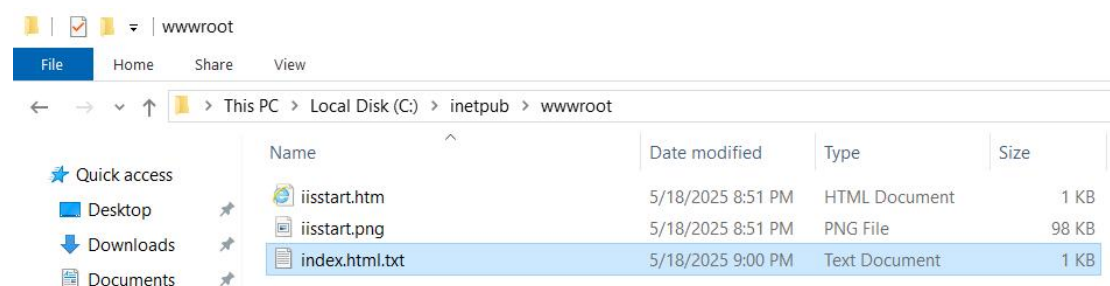
For this click the View menu.



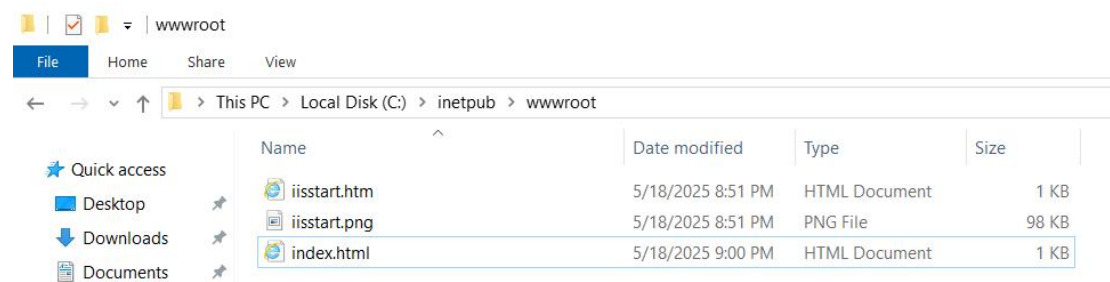
In the View menu, select the checkbox of File name extensions as shown below.



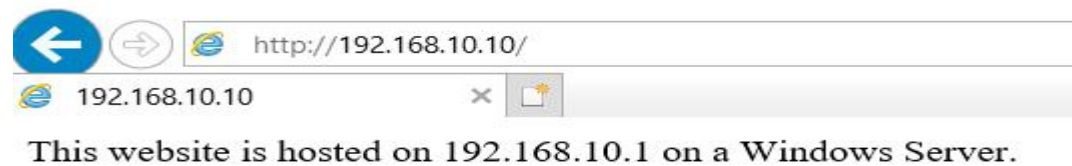
This will display the file extensions as shown below.



Now rename the file. Remove the txt extension and just keep it as index.html. The file should look like as below.



Now open the browser in Windows and type <http://192.168.10.10>. Check if your website is visible.



This website is available on the local network. But we have to open it for the Internet users. Thus we need to configure port forwarding on the iptables firewall. So that this website will open on the Firewall Internet side IP address.

Enable Port Forwarding on the iptables firewall

Go to the iptables firewall machine. Login with a user with a sudo permission.

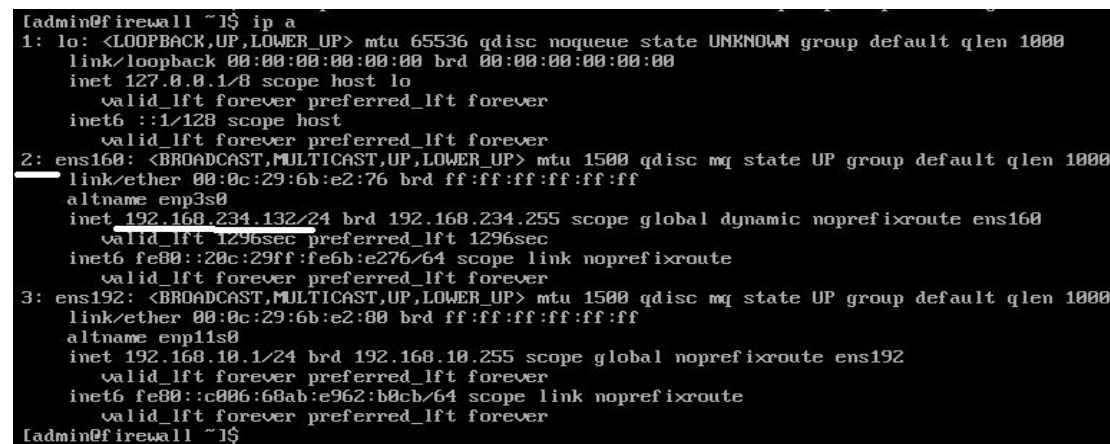
Give following command to enable port forwarding to open the website running on Windows server using IP address 192.168.10.10.

```
sudo iptables -t nat -A PREROUTING -i ens160 -p tcp --dport 80 -j DNAT --to-destination 192.168.10.10:80
```



In the above command make sure the Internet side interface (First Ethernet Interface) name is ens160. If it is different then change it to match to your interface name.

Now find out the IP address of this interface.



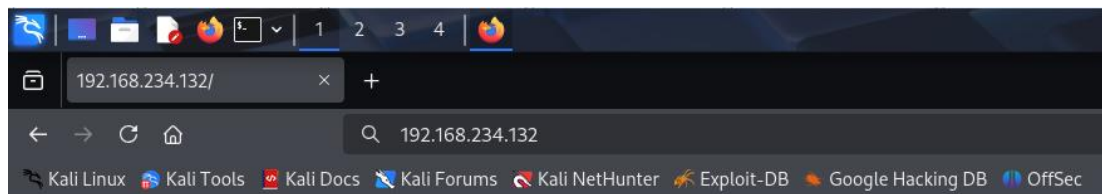
Here the interface marked with the white line in the Internet side interface. The IP address of this interface is like a public IP. The website should be accessible on this IP address.

Test if Port forwarding is working

Go to the Kali Linux machine. Login with user. Open the browser. And type the IP address of the iptables firewall first interface. Thus in the above scenario it will be like

<http://192.168.234.132>

If everything is configured correctly, the website should open as shown below.



This website is hosted on 192.168.10.1 on a Windows Server.

This is how you have successfully opened an internal website to outside world using port forwarding in iptables.

Network Attacks and iptables countermeasures

TCP SYN Flood Attack - Denial of Service

On the Windows Server press Ctrl+Alt+Del (In case of Virtual machine press Ctrl+Alt+Insert) and start the task manager. Click Performance. Check the current CPU and memory usage.

From Kali Linux first perform a SYN flood attack on the Windows web server. For this you use hping3 utility. Type following command. Change the IP address to the outside IP address of the iptables firewall machine.

```
sudo hping3 --count 150000 --data 800 --win 64 --syn -p 80 --flood --rand-source 192.168.234.132
```

```
(hacker@kali)-[~/Desktop]
$ sudo hping3 --count 150000 --data 800 --win 64 --syn -p 80 --flood --rand-source 192.168.234.132
```

Execute the command. It will show the output as below.

```
(hacker@kali)-[~/Desktop]
$ sudo hping3 --count 150000 --data 800 --win 64 --syn -p 80 --flood --rand-source 192.168.234.132
HPING 192.168.234.132 (eth0 192.168.234.132): S set, 40 headers + 800 data bytes
hping in flood mode, no replies will be shown
```

Go to the Windows Server and check the CPU and Memory usage.

The Windows server may not respond at all. Press Ctrl+C to stop the command.

Now go to the iptables firewall. Add following rule to prevent this attack.

```
sudo iptables -A FORWARD -p tcp --dport 80 -m limit --limit 10/s --limit-burst 100 -j ACCEPT
```

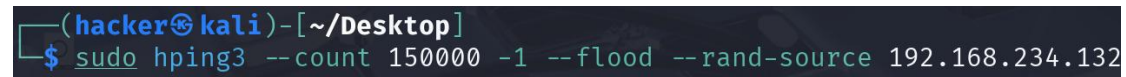
```
[admin@firewall ~]$ sudo iptables -A FORWARD -p tcp --dport 80 -m limit --limit 10/s --limit-burst 100 -j ACCEPT_
```

Now perform the attack again. Check the CPU load on the Windows Server.

Ping Flood Attack - Denial of Service

From Kali Linux machine give the following command. This command will perform a Ping flood attack using ICMP on the outside IP address of the firewall machine.

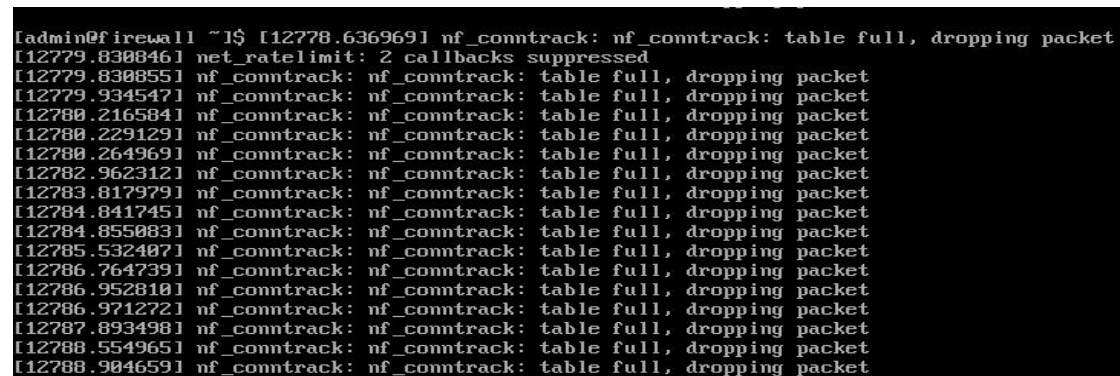
```
sudo hping3 --count 150000 -1 --flood --rand-source 192.168.234.132
```



```
(hacker@kali)-[~/Desktop]
$ sudo hping3 --count 150000 -1 --flood --rand-source 192.168.234.132
```

In the above command please change the IP address to match to your iptables firewall machine IP address. The -1 options enables ICMP protocol. Default is TCP.

After sometime you may get following error messages on the iptables firewall machine console.



```
[admin@firewall ~]$ [12778.636969] nf_conntrack: nf_conntrack: table full, dropping packet
[12779.830846] net_ratelimit: 2 callbacks suppressed
[12779.830855] nf_conntrack: nf_conntrack: table full, dropping packet
[12779.934547] nf_conntrack: nf_conntrack: table full, dropping packet
[12780.216584] nf_conntrack: nf_conntrack: table full, dropping packet
[12780.229129] nf_conntrack: nf_conntrack: table full, dropping packet
[12780.264969] nf_conntrack: nf_conntrack: table full, dropping packet
[12782.962312] nf_conntrack: nf_conntrack: table full, dropping packet
[12783.817979] nf_conntrack: nf_conntrack: table full, dropping packet
[12784.841745] nf_conntrack: nf_conntrack: table full, dropping packet
[12784.855083] nf_conntrack: nf_conntrack: table full, dropping packet
[12785.532407] nf_conntrack: nf_conntrack: table full, dropping packet
[12786.764739] nf_conntrack: nf_conntrack: table full, dropping packet
[12786.952810] nf_conntrack: nf_conntrack: table full, dropping packet
[12786.971272] nf_conntrack: nf_conntrack: table full, dropping packet
[12787.893498] nf_conntrack: nf_conntrack: table full, dropping packet
[12788.554965] nf_conntrack: nf_conntrack: table full, dropping packet
[12788.904659] nf_conntrack: nf_conntrack: table full, dropping packet
```

To prevent this, there are 2 methods.

First one is to block ICMP Ping which is the ICMP type echo_request. The rule for this is.

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

But by default in iptables in Rocky/CentOS/RedHat there is a rule to accept ICM traffic from anywhere. This rule is at number 2 in the INPUT chain. Thus you need to delete this rule and add above rule there. For this give following commands.

```
sudo iptables -D INPUT 2
sudo iptables -I INPUT 2 -p icmp --icmp-type echo-request -j DROP
```

Now again perform the attack and check if you get above error.

The second method is to limit the number of packets accepted. This is useful in case if you can not block ICMP ping.

```
sudo iptables -D INPUT 2
sudo iptables -I INPUT 2 -p icmp --icmp-type echo-request -m limit 5/s -j ACCEPT
```

Now again perform the attack and check if you get above error

Slowloris - Denial of Service Attack

Slowloris is a tool that performs a denial of service attack and allows a single machine to take down a web server with minimal efforts. Slowloris tries to keep many connections to the target web server open and hold them open as long as possible.

Slowloris installation on Kali Linux.

Slowloris is a python script available as open source on the GitHub. You just need to clone the repository. Open Terminal in Kali Linux.

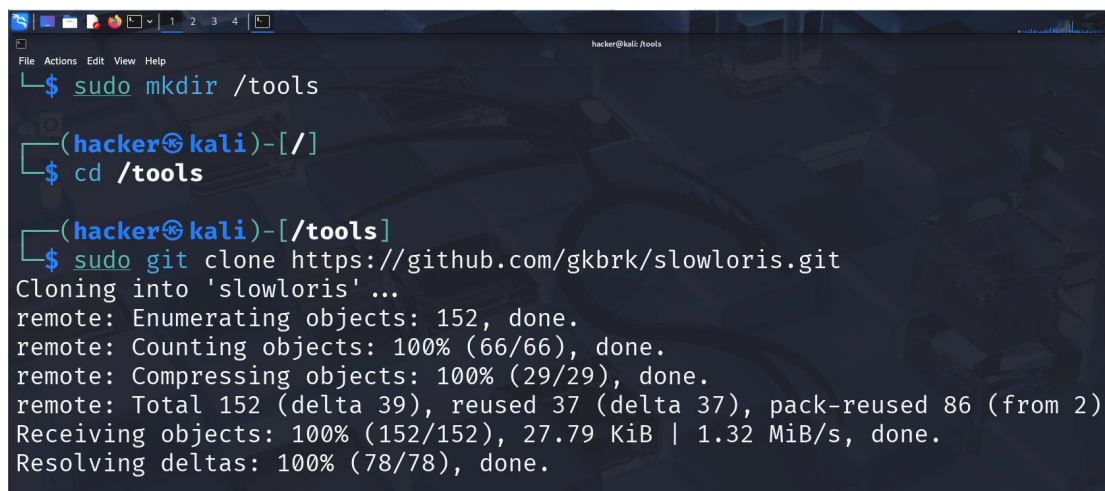
```
sudo mkdir /tools
```

```
cd /tools
```

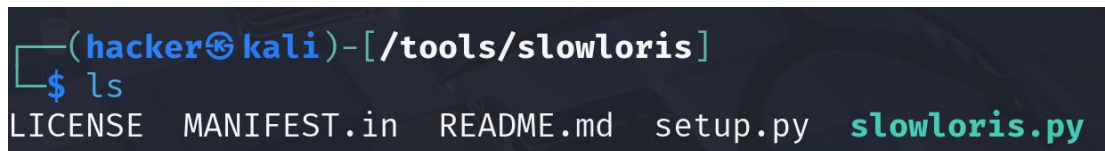
```
sudo git clone https://github.com/gkbrk/slowloris.git
```

```
cd slowloris
```

This is as shown below.

A terminal window on Kali Linux showing the installation of Slowloris. The user runs 'sudo mkdir /tools', then 'cd /tools', and finally 'sudo git clone https://github.com/gkbrk/slowloris.git'. The output shows the cloning process: 'Cloning into 'slowloris'...', 'remote: Enumerating objects: 152, done.', 'remote: Counting objects: 100% (66/66), done.', 'remote: Compressing objects: 100% (29/29), done.', 'remote: Total 152 (delta 39), reused 37 (delta 37), pack-reused 86 (from 2)', 'Receiving objects: 100% (152/152), 27.79 KiB | 1.32 MiB/s, done.', and 'Resolving deltas: 100% (78/78), done.'

```
File Actions Edit View Help
(hacker@kali)~$ sudo mkdir /tools
(hacker@kali)~$ cd /tools
(hacker@kali)~/tools$ sudo git clone https://github.com/gkbrk/slowloris.git
Cloning into 'slowloris'...
remote: Enumerating objects: 152, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 152 (delta 39), reused 37 (delta 37), pack-reused 86 (from 2)
Receiving objects: 100% (152/152), 27.79 KiB | 1.32 MiB/s, done.
Resolving deltas: 100% (78/78), done.
```

A terminal window on Kali Linux showing the contents of the Slowloris directory. The user runs 'ls' in the directory '/tools/slowloris'. The output shows: 'LICENSE MANIFEST.in README.md setup.py slowloris.py'.

```
(hacker@kali)~/tools/slowloris$ ls
LICENSE  MANIFEST.in  README.md  setup.py  slowloris.py
```

Perform the attack using the slowloris.py script.

Go to the slowloris directory as shown above. Give the following command.

```
Sudo python3 slowloris.py -p 80 -s 5000 <Your-web-server-IP-address>
```

In the above command -p defines the port number. Here you are attacking the web server thus the port number is 80. The -s options defines the number of sockets to be created. Make sure to change the IP address of the web server as per your environment.

Now go to the Windows machine. Open the command prompt. Give the following command.

```
netstat -an
```

You will find a lot of connections created from the Kali Linux machine IP address. This will be as shown below.

```

TCP      192.168.10.10:80      192.168.234.1:53825    ESTABLISHED
Can not  obtain ownership      information             192.168.234.1:53826    ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33332  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33342  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33344  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33356  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33370  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33372  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33374  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33384  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33392  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33402  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33404  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33410  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33414  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33420  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33430  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33436  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33438  ESTABLISHED
Can not  obtain ownership      information             192.168.234.140:33444  ESTABLISHED
TCP      192.168.10.10:80      192.168.234.140:33444  ESTABLISHED

```

To prevent this use following rule in the iptables.

```
iptables -I INPUT -p tcp --dport 80 -m connlimit --connlimit-above 20 --connlimit-mask 32 -j DROP
```

This rule limits one host to 20 connections to port 80, which should not affect non-malicious user, but would render slowloris unusable from one host.

Now perform the attack from the Kali Linux. The output of the slowloris will display following.

```

[19-05-2025 12:46:32] Creating 4980 new sockets ...
[19-05-2025 12:46:51] Sending keep-alive headers ...
[19-05-2025 12:46:51] Socket count: 20
[19-05-2025 12:46:51] Creating 4980 new sockets ...
[19-05-2025 12:47:10] Sending keep-alive headers ...
[19-05-2025 12:47:10] Socket count: 20

```

Here you can see that it is allowed to create only 20 connections.

Go to Windows Server and check with netstat -an command. This time you will find less number of connections from the Kali Linux machine.

Limit SSH connections using recent module

```
iptables -I INPUT 1 -p tcp --dport 22 -m state --state NEW -m recent --set --name ssh-list
```

```
iptables -I INPUT 2 -p tcp --dport 22 -m state --state NEW -m recent --update --name ssh-list
--seconds 60 --hitcount 4 -j DROP
```

If you want to automatically remove banned IP addresses from the list use --reap option in the second command as shown below.

```
iptables -I INPUT 2 -p tcp --dport 22 -m state --state NEW -m recent --update --name ssh-list
--seconds 60 --reap --hitcount 4 -j DROP
```