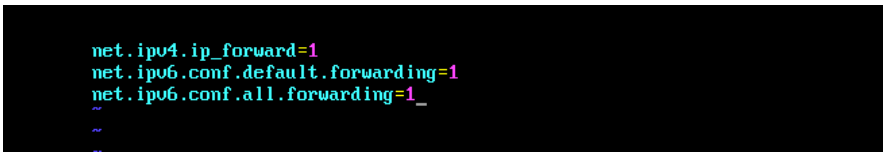
How we enabled Ip forwarding permanently

1. Created a newfile:
    - nvim /etc/sysctl.d/30-ipforward.conf
2. Added the following lines
    - 

3. Reboot the system
4. Now each time we log in to check if ipforwarding is enabled:
    - sysctl net.ipv4.ip_forward
    - The output should be equal to 1

Part 1

Linux user: root

Password : Tkn#Ror23@

Linux user: bayo

password : 1exAJhTD31%@bl

Windows User: Administrator

password: Cl#jK0ud23@pen

Windows User: bayo

Password: duDu%9kaXz1p@

Reason: completely random, has length of atleast 8 characters and with a mixture of different characters

How to change password of a user on Linux Machine:

- Simply type : passwd username

How to add a user on Linux Machine:

- Simply type: sudo useradd -m username
- To be able to login we need to create a password for this user:

    passwd username

How to change password of a user on windows

- Sign in as the user
- Click home
- Click settings
- Click accounts

- Click sign in options
- Click change under password

How to add a user on windows

- Sign in as admin
- Go to accounts by following the steps above
- Click other people
- Click add someone else to this pc
- Click users under name
- Click more actions under Users under Actions
- Click new user
- Enter the details in pop up menu

Part 2

FOR LINUX

Home directory for ftp : /etc

Home directory for http: /etc/httpd

Each ftp service should allow only anonymous access

- Edited the vsftpd.conf file: nvim /etc/vsftpd.conf
- Uncommented and changed the following lines;

  anonymous_enable = YES
- Added the following lines:

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
#
# No password is required for an anonymous login
no_anon_password=YES
```

Test:

- Log in as root user
- type: ftp localhost
- When it prompts you to log in if you type in any other name asides “anonymous” log in will fail
-

```
ftp localhost
ftp: connect to address ::1: Connection refused
ftp: Trying 127.0.0.1 ...
Connected to localhost.
220 (vsFTPd 3.0.3)
Name (localhost:root): root
530 This FTP server is anonymous only.
ftp: Login failed.
```

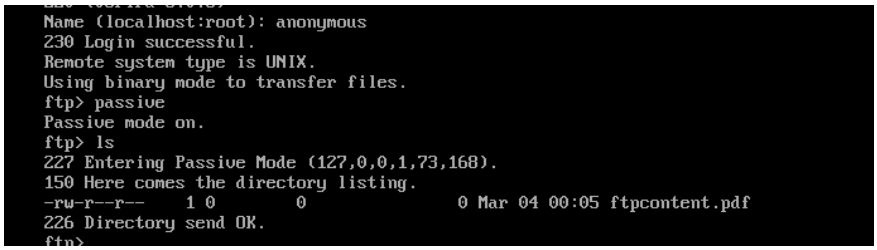
```
Name (localhost:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Each ftp server should have “ftpcontent.pdf”

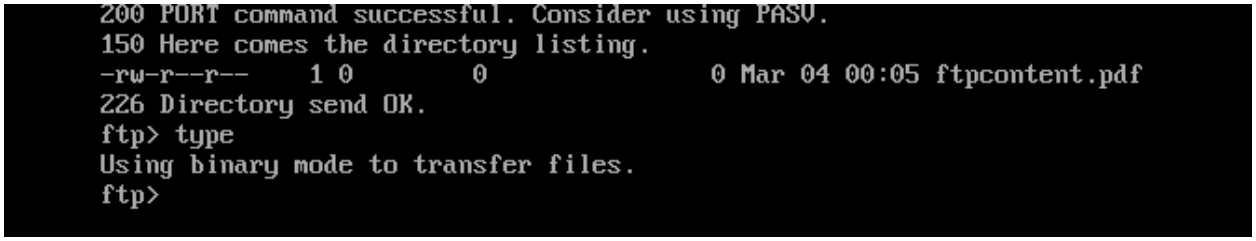
- Created a directory in /var/ called ftp : mkdir ftp
- Created a file in this directory called ftpcontent.pdf : nvim ftpcontent.pdf
- Set the owner of the directory to nobody and group to nobody: chown nobody:nobody /var/ftp
- Added the following lines to vsftpd.conf file:
  - anon_root = /var/ftp
  - seccomp_sandbox=NO

  Test:
  - After logging into ftp as anonymous user
  - Type ls and you should see the file that file listed

Test for passive ftp mode



Test for active ftp mode



Each http server should only allow access to user specified

- I installed mod-auth-external by cloning the repo:
  - https://github.com/phokz/mod-auth-external/tree/master/mod_authnz_external.
- Followed the installation instructions(installed it dynamically)

- I installed pwauth by cloning the repo: https://github.com/phokz/pwauth/tree/master/pwauth. Followed the installation instructions.
    - Changed the part of the config.h file that specified Server Uid to 33
- Then in my httpd.conf located at /etc/http/conf/httpd.conf i added these lines:
    -
```
<ifModule mod_authnz_external.c>
   DefineExternalAuth pwauth pipe /usr/sbin/pwauth
</ifModule>

   <Directory "/srv/http">
      Authtype Basic
      Authname "Secure Area!"
      AuthBasicProvider external
      AuthExternal pwauth
      Require user bayo_
   </Directory>
```

By using pwauth(external authenticator) this ensures that the user must be a valid user. I tested this by changing the password for the user (passwd bayo), when i tried to log in with the old password it failed, when i used the new password it successfully authenticated it. And showed the "/" directory.

Each http server should have "webcontent.html"

- The http server will serve files in the path specified by DocumentRoot in the httpd.conf file.
- To find this path: nvim /etc/httpd/conf/httpd.conf
- In this file I found the DocumentRoot to be specified as /srv/http.
- I went to this directory: cd /srv/http
- then added the file "webcontent.html" : nvim webcontent.html

  Test:
    - Sign into windows machine
    - Go to internet explorer
    - Type the address: http://10.299.1.1
    - After successful authentication, you should see the webcontent.html file

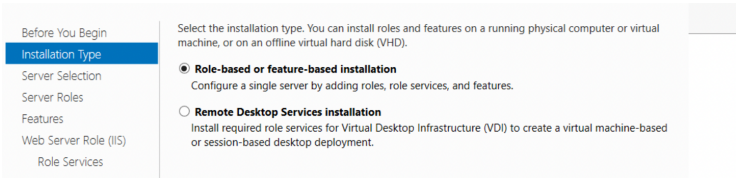
      Index of /

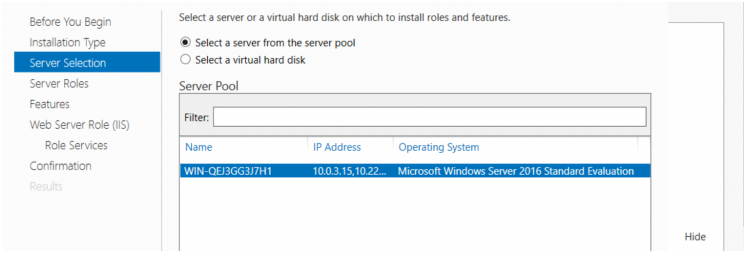
Name	Last modified	Size	Description
webcontent.html	2021-03-03 20:18	0	

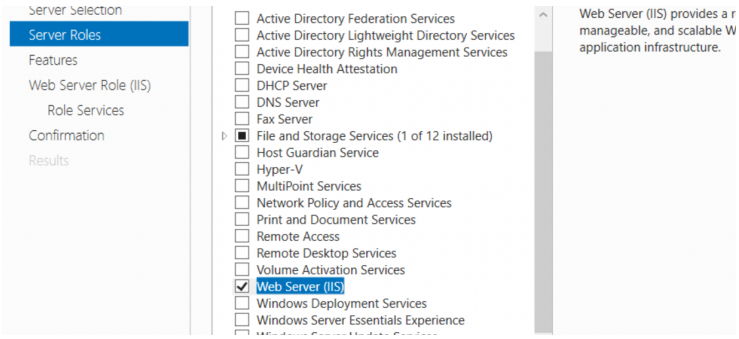
    -

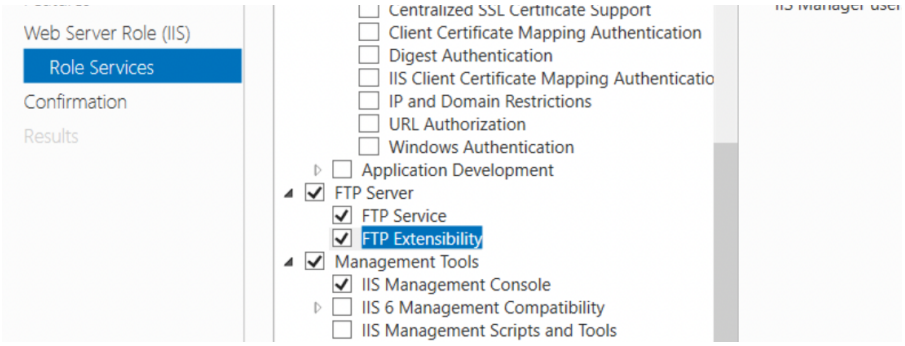
For Windows

First we download the ftp server

- Open the app Server Manager
- Click the manage menu
- Click Add Roles and Features
- 

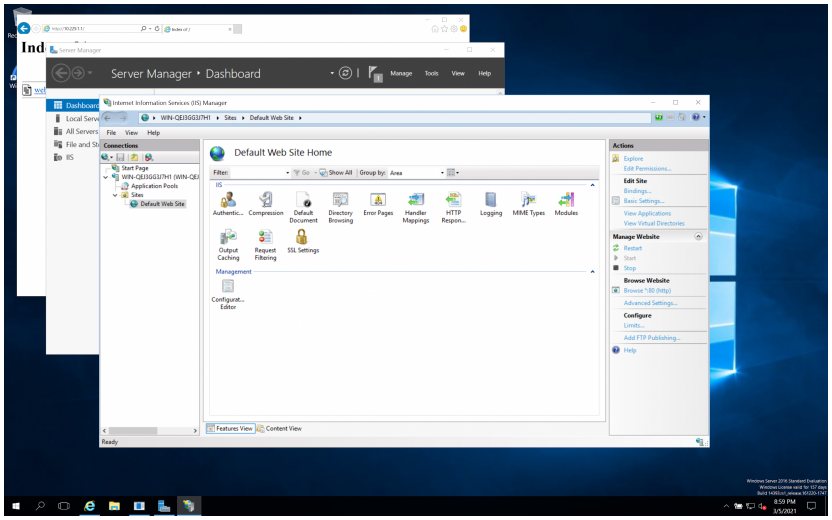
- 

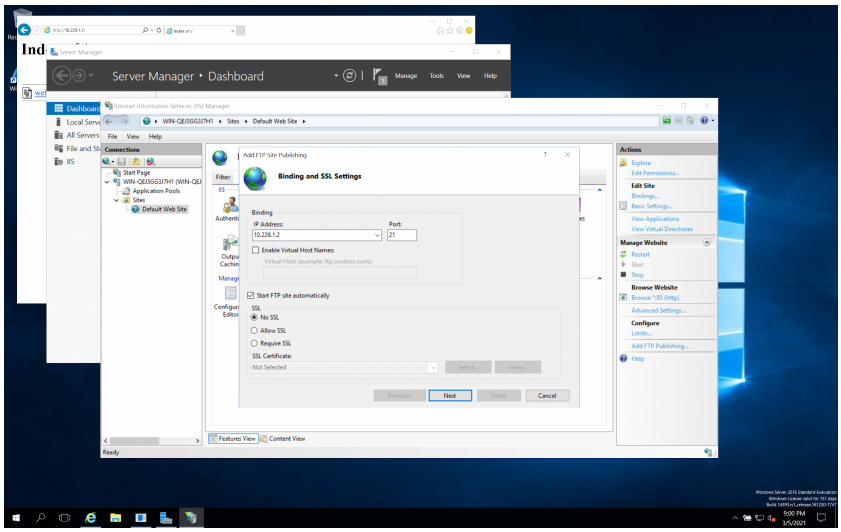
- 

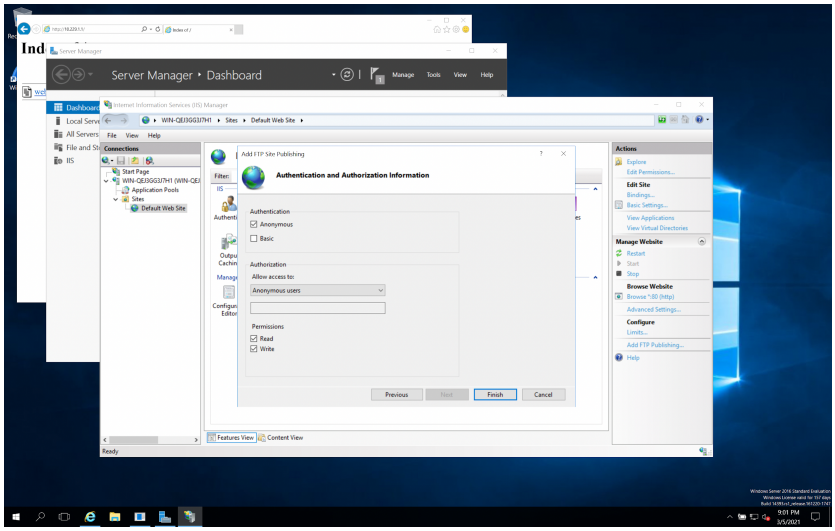
- 

Each ftp service should allow only anonymous access

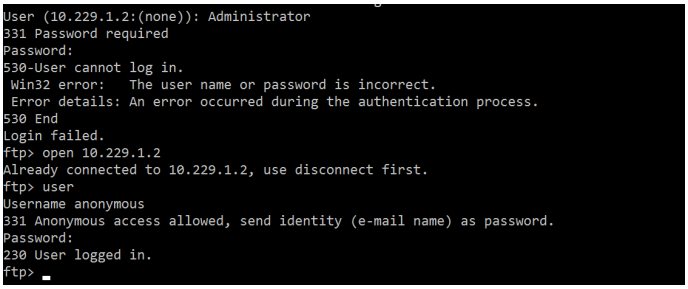
- Open server manager
- Click tools
- Click Internet Information services(IIS) manager
- Click add ftp publishing in the bottom left

- 

- Fill in the options below and click next
- 

- Fill in the option below then click finish
- 

Test:

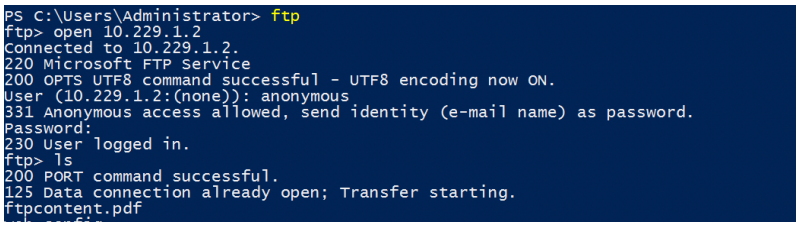
- Open cmd
- Type ftp 10.229.1.2
- 

- It will only accept “anonymous” as user all other username fails

Each ftp server should have “ftpcontent.pdf”

- Open server manager
- Click tools
- Click Internet Information services(IIS) manager
- Click “Default Web site” on the left
- Click “content View” at the bottom
- Click “explore” at the top left
- Right click and add new file named “ftpcontent.pdf”

Test

- Open cmd
- Type ftp, and do the following;
- 

Test for passive ftp

- Open control panel
- Click Network and internet
- Click internet options
- Go to the advanced tab
- Scroll and check Use passive ftp

Each http server should only allow access to user specified

- First we download Windows Authentication
    - Open server manager
    - Click the manage menu
    - Click add roles and feature
    - Click next
    - Select role based installation and next
    - Select the server and click next
    - On the Server Roles page, expand Web Server (IIS), expand Web Server, expand Security, and then select Windows Authentication.Click Next.
    - Click Next till you get to confirmation
    - Click install
- Enable Windows Authentication
    - Open server manager
    - Click tools
    - Click Internet Information services(IIS) manager
    - Click “Default Web site” on the left
    - Click on authentication in the iis tab
    - Right Click on Windows authentication and click enable
    - Go back the default website page
    - Click on directory browsing and enable(so we don’t get a 403 error)
    - Click apply on the right side
    - Click restart on the right side

Test to show the user is properly authenticated when password is changed

- On the linux machine do the following
- 

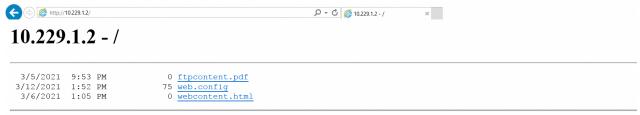
- When i logged into the user “bayo” and i changed the password (used passwd user) and tried again
- Then i tried with the new password i set

- 

Each http server should have “webcontent.html”

- Open server manager
- Click tools
- Click Internet Information services(IIS) manager
- Click “Default Web site” on the left
- Click “content View” at the bottom
- Click “explore” at the top left
- Right click and add new file named “webcontent.html”

Test

- Open internet explorer
- type : http://10.229.1.2
- You would see the added file
- 

Differences in authenticating user account to allow http access

- On the linux machine i had to directly edit the configurations files, but on the windows this is done using GUI.
- On the linux machine we make use of an external authenticator(pwauth) which i had to download but on windows it has an inbuilt authenticator

Differences in ensuring when a user account changes password it automatically reflects when authenticating

- We use an external authenticator (pwauth) which i had to download on the linux machine, on the windows it has its own authenticator no need download anything

PART 3

- We enabled ICMP ping and sshd request on my windows VM in windows firewall with Advanced Security
- We also allowed inbound connections on the windows host in windows firewall with Advanced Security
- We installed an Openssh server from a git repository (https://github.com/PowerShell/Win32-OpenSSH/wiki/Install-Win32-OpenSSH)
- I had to run “chmod u + r” on my sh file because i got a zsh: permission denied error
- We loaded the *nf_conntrack_ftp* module inorder to properly handle ftp connections

ACTIVE FTP MODE

ADVANTAGES

- We have better security(less attack) on the server side as only port 21 needs to be open inbound
- Easier to set up on the server side

DISADVANTAGES

- FTP clients are often using NAT behind firewalls, so if they are using a movable device(like a laptop); this will have an external ip address that is always changing. So this means that ip address sent to the server, using the PORT command will have to be reconfigured each time the external ip address changes
- Wherever the ftp client is at the time of initiating an ftp session. It will need to ensure that the dynamic data port has been allowed in any firewalls between the server and the client

PASSIVE FTP MODE

ADVANTAGES

- Server side is responsible for configuration, which is less likely to be changing or mobile
- The client requires no inbound firewall requirements

DISADVANTAGES

- On the server side, a range of dynamic ports need to be opened for the data channel

Testing of IP Tables

We tested various rules by using tcp dump to view different isolated traffic and check if the rules are doing what they are supposed to. Also used wireshark on the windows VM as well.

Enp0s3 handles communication between the windows and other vms

Enp0s8 handles communication between OurLinux and the other two VMS

- Allow ICMP echo messages(pings) from any network

    - We ran “tcpdump -n -i enp0s3 icmp” to isolate only the icmp traffic that this interface sends to the windows machine. I saw ping requests/replies from the appropriate hosts.(also checked on enp0s8)
- Allow any network to connect to ssh of our hosts
    - We ran “tcpdump -n - i enp0s3 port 22” to isolate only ssh traffic and checked to see if traffic is actually being sent from any host to our windows and linux hosts (also checked on enp0s8)

- Allow all host on 1.0/24 network to connect to all services of our hosts
    - We ran “tcpdump -n -i enp0s3 src net 10.229.1.0/24” and saw that the traffic coming any host on this network had access all the services of our groups host (also checked on enp0s8)
- Allow any host on the network 2.0/24 have access to http service on windows host except for host on 3.0/24
    - We ran “tcpdump -n -i enp0s3 port 80” and saw that only packets from this network were forwarded to the windows host (10.229.1.2)

- Allow any host on the network 3.0/24 have access to http service on linux host except for host on 2.0/24
    - We ran “tcpdump -n -i enp0s8 port 80” and saw only packets from this network were allowed into to the linux host (10.229.1.1)
- Allow any host on the network 3.0/24 have access to ftp service on linux host except for host on 2.0/24
    - We ran “tcpdump -n -i enp0s8 tcp ‘port 20 or port 21 or portrange 1024-65535 and not (port 20 or port 80)’” and saw that only packets from this network were able to connect to to linux ftp services in both active and passive mode.
- Allow any host on the network 2.0/24 have access to ftp service on Windows host except for host on 3.0/24
    - We ran “tcpdump -n -i enp0s3 tcp ‘port 20 or port 21 or portrange 1024-65535 and not (port 20 or port 80)’” and saw

that only packets from this network were able to connect to to windows ftp services in both active and passive mode.

- Prohibiting windows from initiating any service
    - We run ““tcpdump -n -i enp0s3 ‘src 10.229.1.2 and dst net 10.229.2.0/24’” and see that the only packets sent are replies to existing connections.

Testing of logging rules

- To test logging for input, output and forwarded traffic, we first added prefixes to each of the log chains when writing the rules.
    - INPUT: "INPUT traffic logged: "
    - OUTPUT: "OUTPUT traffic logged: "
    - FORWARD: "FORWARD traffic logged: "
    - This allowed us to more clearly see what was being logged when testing.
- To do the actual test we used the command “ **journalctl -k | grep "IN=.*OUT=.*" | less** ” to display the log messages.
    - When looking at the log messages we looked specifically at the source and destination addresses and confirmed that what is logged is indeed intended to be dropped.

Assumption

It was assumed that a violation of the outbound restriction is traffic which was dropped due to the rule

Handling UDP services

- I didn't write any rules to explicitly handle UDP traffic because the default policy on the chains seems to be enough in my case.