**RHCE in RHEL-7 DEMO QUESTION & SOLUTION**

**=======================================**

**01. SELinux must be running in enforcing mode.**

#vim /etc/selinux/config or vim /etc/sysconfig/selinux

SELINUX=enforcing

:wq!

#reboot

#getenforce

**02. Create a repository by using the following URL http://classroom.example.com/pub/rhel7/dvd/**

#vim /etc/yum.repos.d/yum.repo

[yum]

name=yum.repo

baseurl=http://classroom.example.com/pub/rhel7/dvd/

enabled=1

gpgcheck=0

:wq!

#yum clean all

#yum repolist all

**03. SSH Configuration.**

**-Configure ssh access on your virtual hosts as follows**

**-Client within hacker.org should NOT have access to ssh on your system**

[By default it may configure ssh server, IF NOT ]

#yum install openssh-server -y

#systemctl start sshd.service

#systemctl enable sshd.service

#firewall-cmd --permanent --add-service=ssh

#firewall-cmd --reload

#firewall-cmd --list-all

#firewall-config

Will open Firewall GUI mode then Select public go to Configuration & select Permanent-->Rich Rule-->Add

Family:ipv4

Element:service ssh

Action: reject

Source: 172.16.0.0/24 (Already provided this IP for hacker.org)

Press OK & OK

Option--> Reload Firewalld

or File--> Quit & #firewall-cmd–reload

#firewall-cmd --list-all

**04. Configure serverX to forward traffic incoming on port 5243/tcp from source network 172.25.x.0/255.255.255.0 to port on 80/tcp**

#firewall-config

Will open Firewall GUI mode then Select public go to Configuration & select Permanent-->Rich Rule-->Add

Family:ipv4

Element:service forward-port

Protocol:tcp, Port / Port Range:5243 select Local Forwarding

Port/Port Range:80

OK

Source: 172.16.10.10/24 (where X=Station Number)

Ok

Option--> Reload Firewalld

or File--> Quit & #firewall-cmd–reload

**05. Customize user environment**

**-Create a command called qstat on both serverX & desktopX**

**-It should be able to execute the following command**

**-(ps-eo,pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,com)**

**#vim /etc/bashrc** or **#vim /etc/profile**

qstat ()

{

ps-eo,pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,com

}

:wq!

**#source /etc/bashrc** or **#source /etc/profile**

#qstat[For Check]

**06. Configure ipv6 network**

**-Configure eth0 with a static ipv6 address as follows**

**-Configure a static IP address in serverX as fddb:fe2a:ab1e::c0a8:64/64**

**-Configure a static IP address in desktopX as fddb:fe2a:ab1e::c0a8:02/64**

**-Both machine can be communicates with fddb:fe2a:ab1e/64**

#nmcli connection show

#nmcli connection modify "System eth0" ipv6.addresses fddb:fe2a:ab1e::c0a8:64/64 ipv6.method static connection.autoconnect yes **[By default connection.autoconnect yes]**

#nmcli connection down "System eth0" ;nmcli connection up "System eth0"

#ifconfig -a [For Check]

#ping6 fddb:fe2a:ab1e::c0a8:02 [For Check]

**N.B: Same way in desktopX machine also.**

**07. Link aggregation**

**-Configure your systems which watches for link changes & selects an active port for data transfers.**

**-Systems should have the address 192.168.0.10/255.255.255.0**

#lab teambridge setup [Only for lab environment]

Or

#lab sa3-review setup

#ifconfig -a [shows how many LAN port’s are available in system]

#nmcli connection show [For Check]

#systemctl enable NetworkManager

#systemctl restart NetworkManager

#nmcli connection add con-name team0 type team ifname team0 config '{"runner": {"name": "activebackup"}}'

#nmcli connection add con-name team-port1 type team-slave ifname slave1 master team0

#nmcli connection add con-name team-port2 type team-slave ifname slave2 master team0

#systemctl restart NetworkManager

#teamdctl team0 state

[For Check: To know the state of **team-port1,** **team-port2 & teamdctl team-port1 state**]

#nmcli connection show [For Check]

#nmcli connection modify team0 ipv4.addresses 192.168.10.10/24 ipv4.method static

#nmcli restart NetworkManager [To get this ip192.168.10.10/24 live]

#ifconfig -a [For Check]

#teamdctl team0 state [For Check]

**N.B: Same way in desktopX machine also.**

**08. Your ServerX system should accept new email Message over SMTP from 172.25.X.0/24 subnet. All Message not addressed to @serverX.example.com or @localhost[.localdomain] should be forwarded to the SMTP smarthost running on desktoX.example.com**

#yum install postfix –y

#firewall-cmd –permanent –add-service=smtp

#firewall-cmd –reload

#firewall-cmd –list-all

#systemctl start postfix

#systemctl enable postfix

#systemctl restart postfix

#cd /etc/postfix/

#cp main.cf main.cf.ori

#vim main.cf

myhostname= server7.example.com

mydomain = example.com

myorigin = $myhostname

or

myorigin = $mydomain

inet\_interfaces = all

inet\_protocols = all

mydestination = $myhostname, localhost.$mydomain, localhost, mydomain,

mynetworks = 192.168.44.132/24 127.0.0.0/8

relayhost = desktop7.example.com

wq!

Or

#vim main.cf

inet\_interfaces = all

mynetworks = 172.25.7.0/24 127.0.0.0/8

mydestination =server7.example.com, localhost, localhost.localdomain

relayhost = desktop7.example.com

wq!

#systemctl restart postfix

[Test Mail Server}

#mail -v student(another systemuser)

Subject:.....

Mail Body:......

.

#su student (systemuser)

#tail -f /var/spool/mail/student

**09. Configure the SMTP mail service on desktopX which relay the mail only from local system through classroom.example.com, all outgoing mail has their sender domain as example.com. Ensure that mail should not store locally.**

**- Verify the mail server is working by sending mail to a natasha user.**

**- Check the mail on desktopX with the below URL** [**http://content.example.com/system2**](http://content.example.com/system2)

#yum install postfix –y

#firewall-cmd –permanent –add-service=postfix

#firewall-cmd –reload

#firewall-cmd –list-all

#systemctl start postfix

#systemctl enable postfix

#systemctl restart postfix

#cd /etc/postfix/

#cp main.cf main.cf.ori

#vim main.cf

Intet\_interfaces = Loopback-only

Mynetworks = 127.0.0.0/8, [::1]/128

Myorigin = example.com

Mydestination =

Relayhost = [classroom.exaple.com]

Local\_transport = error: local delivery disabled

wq!

[Test Mail Server}

#mail -v natasha@example.com(another domainuser)

Subject:.....

Mail Body:......

.

#mailq

Same configuration in desktop

**10. Implement a webserver for the site** [**http://serverX.example.com**](http://serverX.example.com) **Download the webpage from** [**http://content.example.com/pub/rhce/rhce.html**](http://content.example.com/pub/rhce/rhce.html)**. Rename the downloaded file into index.html copy the fileinto your webserver document root. Don not make any modification with the content of the index.html. Clients within hacker.org should NOT access the webserver on your systems.**

#yum install httpd –y

#firewall-cmd –permanent –add-service=http

#firewall-cmd –reload

#firewall-cmd –list-all

#systemctl start httpd [no need to run this]

#systemctl enable httpd

#systemctl restart httpd

#cd /etc/httpd/conf/

#cp httpd.conf httpd.conf.ori

#vim httpd.conf

ServerName server7.example.com:80

#cd /var/www/html

#wget<http://content.example.com/pub/rhce/rhce.html>

#mv rhce.html index.html

#systemctl restart httpd

#curl server7.example.com [for check]

#firewall-config

Will open Firewall GUI mode then Select public go to Configuration & select Permanent-->Rich Rule-->Add

Family:ipv4

Element:service http

Action: reject

Source: 172.16.0.0/24

Press OK & OK

Option--> Reload Firewalld

or File--> Quit & #firewall-cmd –reload

#firewall-cmd --list-all

**11. Configure the website** [**https://serverX.example.com**](https://serverX.example.com) **with TLS SSLCertificate file http://classroom.example.com /pub/tls/certs/server.crt**

**SSLCertificatekeyfile** [**http://classroom.example.com/pub/tls/private/serverX.key**](http://classroom.example.com/pub/tls/private/serverX.key)

**SSL CA certificate file http://classroom.example.com/pub/example-ca.crt**

#yum install mod\_ssl -y

#cd /etc/httpd/conf.d

#cp ssl.conf ssl.conf.ori

#vim ssl.conf

Find: /**SSLCertificateFile** text and copy the link location of the **SSLCertificateFile** then **wget** the provided link content on the provided location. After that rename the downloaded file same with provided link text to the station number of the **serverX.crt**

Find: **/SSLCertificateKeyFile**text and copy the link location of the **SSLCertificateKeyFile** then **wget** the provided link content on the provided location. After that rename the downloaded file same with provided link text to the station number of the **serverX.key**

Then uncomment **SSLCertificateChainFile** or **SSLCACertificateFile** and similarly copy the link location of the **SSLCertificateChainFile** or **SSLCACertificateFile** then **wget** the provided link content on the provided location. After that change the link text same with the downloaded file to station number of the **server-chain.crt** or **ca-bundle.crt**

#firewall-cmd –permanent –add-port=443/tcp

#firewall-cmd –permanent –add-service=https

#firewall-cmd –reload

#firewall-cmd –list-all

#systemctl start httpd

#systemctl enable httpd

#systemctl restart httpd

[Check the webpage link started with https://serverX.example.com]

**12. Create a folder topsecret in your web server Document Root. Download http://content.example.com/pub/thce/restrict.html Rename the file into index.html. The content of topsecret should be visible to everyone browsing from your localsystem but should not accessible from other location.**

#mkdir /var/www/html/topsecret

#cd /var/www/html/topsecret

#wgethttp://content.example.com/pub/thce/restrict.html

#mv restrict.html index.html

#vim /etc/httpd/conf.d/vhosts.conf

<VirtualHost serverX.example.com:80>

ServerName serverX.example.com

DocumentRoot /var/www/html

</VirtualHost>

<Directory /var/www/html/topsecret>

Require host serverX.example.com

</Directory>

:wq!

#systemctl start httpd

#systemctl enable httpd

#systemctl restart httpd

[Check from local system firefox serverX.example.com/topsecret]

**13. Setup a virtual host with an alternative document root. Extend your web to include a virtual for the site http://wwwX.example.com. Set the document root as /virtual. Download http://content.example/pub/rhce/www.html and Place this to document root of the virtual host. Rename the file as index.html. Note: The other websites configures for your server must still accessible. wwwX.example.com is already provided by the name server on example.com**

#mkdir /virtual

#cd /virtual

#wget http://content.example/pub/rhce/www.html

#mv www.html index.html

#chcon -t httpd\_sys\_content\_t /virtual -R

Or

#chcon --reference /var/www/html /virtual -R

#ls -ldZ /virtual [for check SE context]

#vim /etc/httpd/conf.d/vhosts.conf

<VirtualHost serverX.example.com:80>

ServerName serverX.example.com

DocumentRoot /var/www/html

</VirtualHost>

<Directory "/var/www/html/topsecret">

Require host serverX.example.com

</Directory>

**<VirtualHost wwwX.example.com:80>**

**ServerName wwwX.example.com**

**DocumentRoot /virtual**

**</VirtualHost>**

**<Directory /virtual>**

**Require all granted**

**</Directory>**

:wq!

#systemctl restart httpd

#curl wwwX.example.com

**14. Configure website http://webappX.example.com:9189/myapp on serverX with the document root /dynamic. Site should executes appweb.wsgi. Page is already provided on http://content.example.com/pub/appweb.wsgi. Content of the script should not be modified.**

#yum install mod\_wsgi -y

#mkdir /dynamic

#cd /daynamic

#wget http://content.example.com/pub/appweb.wsgi

#chcon -t httpd\_sys\_content\_t /dynamic -R

Or

#chcon --reference /var/www/html /dynamic -R

#ls -ldZ / dynamic [for check SE context]

#vim /etc/httpd/conf.d/vhosts.conf

<VirtualHost serverX.example.com:80>

ServerName serverX.example.com

DocumentRoot /var/www/html

</VirtualHost>

<Directory "/var/www/html/topsecret">

Require host serverX.example.com

</Directory>

<VirtualHost wwwX.example.com:80>

ServerName wwwX.example.com

DocumentRoot /virtual

</VirtualHost>

<Directory /virtual>

Require all granted

</Directory>

**<VirtualHost webappX.example.com:9189>**

**ServerName webappX.example.com**

**WSGIScriptAlias /myapp /dynamic/appweb.wsgi**

**</VirtualHost>**

**<Directory /dynamic>**

**Require all granted**

**</Directory>**

**Listen 9189**

:wq!

#semanage port -l | grep http

#semanage port -a -t http\_port\_t -p tcp 9189

#semanage port -l | grep http

#firewall-cmd --permanent --add-port=9189

#firewall-cmd --reload

#firewall-cmd --list-all

#systemctl restart httpd

[check from foundation machine firefox **webappX.example.com:9189/myapp**]

**15. Restore a database on serverX from the backup file http://content.example.com/new\_inventory.dump. The database name should be Contacts. It should be access only within the localhost. Set a password for root user as "Postroll". Other than the root user, the user andrew able to read the query from the above mentioned database. The user should be authenticated with the password as "Postroll".**

#yum groupinstall mariadb mariadb-client -y

#firewall-cmd –permanent –add-service=mysql

#firewall-cmd –reload

#firewall-cmd –list-all

#systemctl start maraiadb

#systemctl enable mariadb

#systemctl restart mariadb

#mysql\_secure\_installation

[Set root password as "Postroll", Disallow root login from remote and Reload privileges for mariadb]

#mysql -u root -p

Password: Postroll

Or

#mysql -u root -pPostroll

<none>create database Contacts;

<none>show databases;

Exit

#wget http://content.example.com/new\_inventory.dump

#mysql -u root -pPostroll Contacts < new\_inventory.dump

#mysql -u root -pPostroll

<none>show databases;

<none>use mysql;

<mysql>create user andrew@localhost identified by 'Postroll';

<mysql>grant select on Contacts.\* to andrew@localhost identified by 'Postroll';

<mysql>show grants for andrew@localhost;

Exit

**16. Contacts database queries using a database on the system serverX and use the appropriate SQL query to answer the following questions: 1. Find the first\_name of the person who's password is lrsji . Create a file in /root called fname.ext and write the name there. 2. Find the number of persons who live in Dhaka. Create a file in /root called number.txt and write the number.**

#mysql -u root -pPostroll

<none>show databases;

<none>use Contacts;

<Contacts> show tables;

<Contacts> select \* from user\_info where password='lrsji';

<Contacts> select count(\*) from user\_info where address='Dhaka';

Exit

#echo result\_of\_1st\_query > /root/fname.txt

#echo result\_of\_2nd\_query > /root/number.txt

**17. Create a script on serverX called /root/random with following details.**

**When run as /root/random postconf, should bring the output as "postroll"**

**When run as /root/random postroll, should bring the output as "postconf"**

**When run with any other argument or without argument, should bring the stdrr as "/root/random postconf | postroll"**

#cd /root

#vim random.sh

#!/bin/bash

if [ $# -eq 1 ]

then if [ $1 == postconf ]

then echo "postroll"

elif [ $1 == postroll ]

then echo "postconf"

else echo "/root/random postconf | postroll"

fi

else echo "/root/random postconf | postroll"

fi

wq!

#chmod +x /root/random.sh

#sh /root/random.sh postroll (check)

#sh /root/random.sh postconf

#sh /root/random.sh post

#sh /root/random.sh

or

#./root/random.sh postroll

#./root/random.sh postconf

#./root/random.sh post

**18. Create a script on serverX called /root/createusers. When this script is with the testfile argument, it should add all the users from the file download the file from http://content.example.com/pub/testfile**

**All users should have the login shell as /bin/false, password not required. When this script is called with anyother argument, it should print the message as "Input File Not Found". When this scriptis run without argument, it should display "Usage: /root/createusers" NOTE: If the users are added no need to delete.**

#cd /root

#wget http://content.example.com/pub/testfile

#vim createuser.sh

#!/bin/bash

if [ $# -eq 1 ]

then if [ $1 == testfile ]

then for user in $(cat $1)

do useradd -s /bin/false $user

done

else echo "Input Not Found"

fi

else echo "usage:/root/createusers"

fi

wq!

#chmod +x /root/createuser.sh

# sh /root/createusers.sh testfile

# ./createusers.sh testfile

#cat /etc/passwd

**FOR ANYFILE AS ARGUMENT**

#cd /root

#vim createuser.sh

#!/bin/bash

if [ $# -eq 1 ] ; then

if [ -f $1 ] ; then

test=$1

for user in $ ( cat $test )

do

useradd –s /bin/false $user

echo “$user”

done

else

echo “Input not found”

fi

else

echo “Usage:/root/createusers”

fi

wq!

#chmod +x /root/createuser.sh

# sh /root/createusers.sh testfile

# ./createusers.sh testfile

#cat /etc/passwd

**19. Create a new 1GB target on your serverX.example.com.**

* **The block device name should be data\_block.**
* **The server should export an iscsi disk called iqn.2014-10.com.example:serverX.**

#yum install targetcli –y

#firewall-cmd –permanent –add-port=3260/tcp; firewall-cmd –reload; firewall-cmd –list-all

#fdisk –l

#fdisk /dev/vdb

n

+1G

t

8e

w

#partprobe

#fdisk –l

#systemctl enable target

#systemctl start target

#targetcli

cd ->backstore->block

create data\_block /dev/vdb1

cd ->iscsi->

create iqn.2014-10.com.example:server7

cd -> iscsi -> acls

create iqn.2014-10.com.example:desktop7

cd ->iscsi -> luns

create /block\_data /dev/vdb1

cd ->iscsi -> portals

create 172.25.7.11

cd /

/>saveconfig

Exit

#systemctl restart target

**20. The serverX.example.com provides an iscsi port (3260). Connect the disk with desktopX.example.com and configure filesystem with the following requirements.**

* **Create 800MB partition on ISCSI block device and assign the filesystem as xfs.**
* **Mount the volume under /mnt/initiatior at the system boot time.**
* **The file system should contain the copy of** [**http://sontent.example.com/pub/iscsi.txt**](http://sontent.example.com/pub/iscsi.txt)**.**
* **The file should be owned by root with 0644 permission.**
* **NOTE: content of the file should not be modified.**

#yum install iscsi-initiator-utils –y

#vim /etc/iscsi/initiator.iscsi

InitiatorName: iqn.2014-10.com.example:desktop7

wq!

#iscsiadm –m discovery –t st –p server7

#iscsiadm –m node –T iqn.2014-10.com.example:server7 –l

#fdisk –l

#fdisk /dev/sda

n

+800M

t

8e

w

#partprobe

#fdisk –l

#mkfs.xfs /dev/sda1

#mkdir /mnt/initiator

#blkid

#vim /etc/fstab

UUID=” SSKFJ24455J5HH5K5K5H3H3H3K4” /mnt/initiator xfs defaults,\_netdev 0 0

wq!

#mount –a

#reboot [rht-vmctl poweroff desktop]

#df –hT

#cd /mnt/initiator

#wget [**http://sontent.example.com/pub/iscsi.txt**](http://sontent.example.com/pub/iscsi.txt)

#chmod 0644 iscsi.txt

**21. Share the /sambadir directory via SMB on server**

**-Your SMB server must be a member of the TESTGROUP workgroup.**

**-The share’s name must be data.**

**-The data share must be available to example.com domain clients only.**

**-The data share must be browse able.**

**-Susan must have read access to the share, authentication with the same password password if necessary.**

#yum install samba samba-client –y

#systemctl enable smb nmb

#systemctl start smb nmb

#systemctl restart smb nmb

#firewall-cmd –permanent –add-service=samba

#firewall-cmd –reload

#firewall-cmd –list –all

#cd /etc/samba

#cp smb.conf smb.conf.ori

#vim smb.com

workgroup=TESTGROUP

interfaces=lo etho

hosts allow=127. 172.25.3.

[data]

path = /sambadir

browseable = yes

writable = no

valid users = susan

[custer]

path = /opstack

browseable = yes

writable = no

valid users = martin,frankenstein

write list = Frankenstein

wq!

#testparm

#chcon –t samba\_share\_t /opstake –R

#chcon –t samba\_share\_t /sambadir –R

#chmod o=rwx /opstack

#useradd susan

#useradd martin

#useradd frankenstein

#smbpasswd –a susan

#smbpasswd –a martin

#smbpasswd –a frankenstein

#pdbedit –L

#systemctl restart smb nmb

#smbclient //server3.example.com/cluster –U martin

Password:SaniTago

**22. Configure the serverX to share /opstack with SMB share name must be cluster.**

**The user frankenstin readable, writeable and accessible to the /opstack SMB share.**

**The user martin has read access to the /opstack SMB share.**

**Both users should have SMB password ‘SaniTago’.**

#yum install samba-client cifs-utils –y

#smbclient //server3.example.com/cluster –U martin

Password:SaniTago

Exit

#mkdir /mnt/smbspace

#vim /root/password.txt

username=frankenstein

password=SaniTago

wq!

#vim /etc/fstab

//server3.example.com/cluster /mnt/smbspace cifs defaults,credentials=/root/password.txt,multiuser,sec=ntlmssp 0 0

wq!

#mount –a

#useradd frankenstain

#su – Frankenstein

#cifscredes add server3

#cd /mnt/smbspace

#mkdir hello

#ls

#exit

**23. Configure serverXwith the following requirements:**

**Share the /nfsshare directory within the example.com domain clients only, share must be writeable.**

**Share the /nfssecure, enable krb5p security to secure access to the NFS share from URL** [**http://classroom.example.com/pub/keytabs**](http://classroom.example.com/pub/keytabs)**/serverX.keytab. Create a directory named as protected under /nfssecure The exported directory should have read/write access from all sub-domains of the example.com domain. Ensure the directory /nfssecure/protected should be owned by the user harry with read/write permission.**

#yum install nfs-utils –y

#firewall-cmd –permanent –add-service =nfs;firewall-cmd –reload

#mkdir /nfsshare

#mkdir /nfssecure/protected –p

#chcon –t public\_content\_t /nfssecure/protected

#chown nfsnobody /nfsshare

#chown nfsnobody /nfssecure

#chown harry:harry /nfssecure/protected

#wget –O /etc/krb5.keytab <http://classroom.example.com/pub/keytabs/serverX.keytab>

#systemctl enable nfs-server nfs-secure-server

#systemctl restart nfs-server nfs-secure-server

#vim /etc/exports

/nfsshare \*.example.com(rw)

/nfssecure \*.example.com(rw,sec=krb5p)

:wq!

#exportfs –r

#systemctl restart nfs-secure-server

**24. Mount /nfsshare directory on deskto.example.compXunder /public directory persistently at system boot time.**

**Mount /nfssecure/protected with krb5 secured share on desktop beneath /secure/protected with keytab** [**http://classroom.example.com/pub/keytabs/desktopX.keytab**](http://classroom.example.com/pub/keytabs/desktopX.keytab)

**The user harry able to write files on /secure directory.**

#mkdir /public

#mkdir /secure/protected –p

#wget –O /etc/krb5.keytab <http://classroom.example.com/pub/keytabs/desktopX.keytab>

#systemctl enable nfs-secure

#systemcl restart nfs-secure

#vim /etc/fstab

serverX.example.com:/nfsshare /public nfs defaults 0 0

server.example.com:/nfssecure/protected /secure/protected nfs defaults,sec=krb5p 0 0

:wq!

#mount -a