Lab 7: BACKUP

Objective: Student will learn how to setup a simple backup server.

Scenario:

- **Domain: coyote365.net**
- **Systems administrator: Davd McKay (dmckay)**
- **Servers and workstations:**

Server Name:	IP Address:	Function:
hadrian	192.168.1.1	Gateway
ns	192.168.1.2	DNS
ns2	192.168.1.3	DNS
www	192.168.1.4	Web server
beehive	192.168.1.5	NFS
chango	192.168.1.6	Backup
lin-0 ~ lin-99	192.168.1.100 ~ 192.168.1.199	Linux Workstations
win-0 ~ lin-54	192.168.1.200 ~ 192.168.1.254	Windows Workstations

Preparing for Backup Server: Create a virtual machines
NAME: change, beehive **CPU: 1**
MEM: 256MB **HDD: 8GB**
NETWORK: Local (Internal network 192.168.1.0/24)

[1] Firewall (iptables) configuration:

[root@chango /root]# **setup**

Select "Firewall configuration"

```
---------- Choose a Tool ----------
|       Authentication configuration      |
|       Firewall configuration            |
|       Network configuration             |
|       System services                   |
|       [ Run Tool ]    [ Quit ]          |
-----------------------------------------
```

Select "Enabled" and SELinux "Disabled"

```
---------- Firewall Configuration ----------
| A firewall protects against unauthorized  network intrusions.  |
| Enabling a firewall blocks all incoming connections.           |
| Disabling a firewall allows all connections and is not recommended |
| allows all connections and is not recommended.                 |
|       Security Level: (*) Enabled ( ) Disabled                 |
|       SELinux:  Enforcing                                      |
|                 Permissive                                     |
|                 Disabled                                       |
|                 [ OK ]   [ Customize ]   [ Cancel ]            |
-----------------------------------------------------------------
```

Select "OK"

```
---------- Firewall Configuration - Customize ----------
| You can customize your firewall in two ways. First, you can select to allow all traffic from |
| certain network interfaces. Second, you can allow certain protocols explicitly through the firewall. |
| Specify additional ports in the form 'service:protocol', such as 'imap:tcp'.                |
|       Trusted Devices:          [ ] eth2 [ ] eth1 [ ] eth0                                  |
|       MASQUERADE Devices:       [ ] eth2 [ ] eth1 [ ] eth0                                  |
|                          [*] SSH          [ ] Telnet  [ ] FTP                               |
|       Allow incoming:    [ ] WWW (HTTP)   [ ] Samba   [ ] Mail (SMTP)                       |
|                          [ ] Secure WWW (HTTPS)        [ ] NFS4                            |
|                          Other ports _____________                                         |
|                          [ OK ]                                                            |
-----------------------------------------------------------------------------------------------
```

Display current rule set with line numbers:

[root@chango /root]# **iptables - -line-numbers –L**

```
Chain INPUT (policy ACCEPT)
num target    prot opt source                destination
1   RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
num target    prot opt source                destination
1   RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
num target    prot opt source                destination

Chain RH-Firewall-1-INPUT (2 references)
num target    prot opt source                destination
1   ACCEPT    all  --  anywhere             anywhere
2   ACCEPT    icmp --  anywhere             anywhere            icmp any
3   ACCEPT    esp  --  anywhere             anywhere
4   ACCEPT    ah   --  anywhere             anywhere
5   ACCEPT    udp  --  anywhere             224.0.0.251         udp dpt:mdns
6   ACCEPT    udp  --  anywhere             anywhere            udp dpt:ipp
7   ACCEPT    tcp  --  anywhere             anywhere            tcp dpt:ipp
8   ACCEPT    all  --  anywhere             anywhere            state RELATED,ESTABLISHED
9   ACCEPT    tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
10  REJECT    all  --  anywhere             anywhere            reject-with icmp-host-prohibited
```

Delete unnecessary rules:

Delete rule number 3:

[root@chango /root]# **iptables –D RH-Firewall-1-INPUT 3**

Delete rule number 4: the rule #4 became #3 after delete rule #3.

[root@chango /root]# **iptables –D RH-Firewall-1-INPUT 3**

Delete rule number 6: the rule #6 became #4 after delete rule #3 and #4.

[root@chango /root]# **iptables –D RH-Firewall-1-INPUT 4**

Delete rule number 7: the rule #7 became #4 after delete rule #3, #4, and #6.

[root@chango /root]# **iptables –D RH-Firewall-1-INPUT 4**

Check the current rule set:

[root@chango /root]# **iptables - -line-numbers –L**

```
Chain INPUT (policy ACCEPT)
num target    prot opt source                destination
1   RH-Firewall-1-INPUT  all  --             anywhere            anywhere

Chain FORWARD (policy ACCEPT)
num target    prot opt source                destination
1   RH-Firewall-1-INPUT  all  --             anywhere            anywhere

Chain OUTPUT (policy ACCEPT)
num target    prot opt source                destination

Chain RH-Firewall-1-INPUT (2 references)
num target    prot opt source                destination
1   ACCEPT    all  --  anywhere             anywhere
2   ACCEPT    icmp --  anywhere             anywhere            icmp any
3   ACCEPT    udp  --  anywhere             224.0.0.251         udp dpt:mdns
4   ACCEPT    all  --  anywhere             anywhere            state RELATED,ESTABLISHED
5   ACCEPT    tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
6   REJECT    all  --  anywhere             anywhere            reject-with icmp-host-prohibited
```

Add following rule and replace the rule for SSH.

[root@chango /root]# **iptables –I RH-Firewall-1-INPUT 5 –s 192.168.1.0/24 –j ACCEPT**
[root@chango /root]# **iptables –R RH-Firewall-1-INPUT 6 –s 192.168.1.0/24 –p tcp --dport 22 –j** ACCEPT
[root@chango /root]# **iptables - -line-numbers –L**

```
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     icmp --  anywhere             anywhere            icmp any
3    ACCEPT     udp  --  anywhere             224.0.0.251         udp dpt:mdns
4    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
5    ACCEPT     all  --  192.168.1.0/24       anywhere
6    ACCEPT     tcp  --  192.168.1.0/24       anywhere            tcp dpt:ssh
7    REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited
```

Save current rule set:

[root@chango /root]# **iptables-save > /etc/sysconfig/iptables**
[root@chango /root]# **service iptables restart**

[2] DNS settings:

[root@chango /root]# **vi /etc/resolv.conf**

```
search coyote.net
nameserver      192.168.1.2
nameserver      139.182.2.6
```

Add following entries into DNS server ns and replicate DNS data into ns2:

beehive	**192.168.1.5**	**NFS**
chango	**192.168.1.6**	**Backup**

Make sure that your name server resolve following names

[root@chango /root]# **nslookup beehive**
[root@chango /root]# **nslookup chango**
[root@chango /root]# **nslookup ns**
[root@chango /root]# **nslookup ns2**

If these servers not registered or registered incorrectly, please fix the problem.

[3] NFS Server Setup:

NFS Server Installation:

[root@beehive ~]# **yum –y install portmap nfs-utils nfs-utils-lib**

NFS Server Configuration:

Create directories to share:

[root@beehive ~]# **mkdir /pool /share**

Edit: /etc/exports

```
/pool           192.168.1.0/255.255.255.0(rw)
/share          192.168.1.0/255.255.255.0(ro)
```

Start NFS services:

Server:

```
[ root@nfs ~]# service portmap start
Starting portmap:                                          [ OK ]
[ root@nfs ~]# service nfs start
Starting NFS services:                                     [ OK ]
Starting NFS quotas:                                       [ OK ]
Starting NFS daemon:                                       [ OK ]
Starting NFS mountd:                                       [ OK ]
```

Clients:

```
[ root@chango ~]# service portmap start; service nfslock start
Starting portmap:                                          [ OK ]
Starting NFS statd:                                        [ OK ]
```

Testing NFS Server:

```
[ root@beehive ~]# showmount -e
Export list for jb356-s0.csci.csusb.edu:
/pool  192.168.1.0/255.255.255.0
/share 192.168.1.0/255.255.255.0
[root@beehvie ~]# rpcinfo –p
program vers proto   port
  100000  2  tcp   111  portmapper
  100000  2  udp   111  portmapper
  100011  1  udp   808  rquotad
  100011  2  udp   808  rquotad
  100011  1  tcp   811  rquotad
  100011  2  tcp   811  rquotad
  100003  2  udp  2049  nfs
  100003  3  udp  2049  nfs
  100003  4  udp  2049  nfs
  100021  1  udp  1092  nlockmgr
  100021  3  udp  1092  nlockmgr
  100021  4  udp  1092  nlockmgr
  100003  2  tcp  2049  nfs
  100003  3  tcp  2049  nfs
  100003  4  tcp  2049  nfs
  100021  1  tcp  3811  nlockmgr
  100021  3  tcp  3811  nlockmgr
  100021  4  tcp  3811  nlockmgr
  100005  1  udp   823  mountd
  100005  1  tcp   826  mountd
  100005  2  udp   823  mountd
  100005  2  tcp   826  mountd
  100005  3  udp   823  mountd
  100005  3  tcp   826  mountd
```

NFS Client setup: *(do this on ns, ns2, hadrian, www)*

```
[root@chango /root]# mkdir /pool
[root@chango /root]# vi /etc/fstab
/dev/VolGroup00/LogVol00 /                       ext3     defaults                1 1
LABEL=/boot              /boot                   ext3     defaults                1 2
tmpfs                    /dev/shm                tmpfs    defaults                0 0
devpts                   /dev/pts                devpts   gid=5,mode=620  0 0
sysfs                    /sys                    sysfs    defaults                0 0
proc                     /proc                   proc     defaults                0 0
/dev/VolGroup00/LogVol01 swap                    swap     defaults                0 0
beehive:/pool            /pool                   nfs      defaults                1 1
[root@chango /root]# chkconfig portmap on;service portmap restart
[root@chango /root]# chkconfig nfslock on;service nfslock restart
[root@chango /root]# chkconfig netfs on;service netfs restart
[root@chango /root]# df
/dev/mapper/VolGroup00-LogVol00
                      18314824  1028832  16340636   6% /
/dev/hda1               101086    11732     84135  13% /boot
tmpfs                   127572        0    127572   0% /dev/shm
beehive:/pool         18314824  1028832  16340636   6% /pool
```

[4] SSH configuration: *Please change following items. (do this on ns, ns2, hadrian, www)*
[root@chango /root]# **vi /etc/ssh/sshd_config**
PermitRootLogin no
StrictModes yes
MaxAuthTries 3
X11Forwarding no
PermitTunnel no
AllowUsers ***dmckay***
Banner /etc/ssh/banner

[root@chango /root]# **vi /etc/ssh/banner**
This is a coyote.net computer system and is the property of the coyote.net Inc.
It is for authorized use only. This computer system, including all related equipment is for authorized use only. Unauthorized or improper use of this system may result in administrative disciplinary action and civil and criminal penalties.
Coyote365.net Inc.
[root@chango /root]# **service sshd restart**

[5] Setup TIME: *Backup server time has to be correct in order run crontab on time.*
[root@chango ~] $ **yum –y install ntp**
[root@chango ~] $ **date**
[root@chango ~] $ **ntpdate pool.ntp.org**
[root@chango ~] $ **date**
[root@chango ~] $ **hwclock –r**
[root@chango ~] $ **hwclock –w**
[root@chango ~] $ **hwclock –r**
[root@chango ~] $ **chkconfig ntpd on**
[root@chango ~] $ **service ntpd start**

[6] Add *dmckay* to sudo user: *(do this on ns, ns2, hadrian, www)*
(You need to do this each serer that you want to connect w/o password)
[root@www ~]# **useradd -c"David Mckay" –G wheel dmckay**
[root@www ~]# **passwd dmckay**
[root@www ~]# **visudo**
%wheel ALL=(ALL) NOPASSWD: ALL

[7] Setup SSH No password between the backup server and target servers:
You need to do following command only once on each server.
[dmckay@backup ~]# **ssh-keygen –t rsa**
Enter passphrase (empty for no passphrase): **Enter**
Enter same passphrase again: **Enter**

Prepare NFS(beehive) server:
You need to do this each serer that you want to connect w/o password.
Login as *dmckay* on backup server
[dmckay@backup ~]# **mkdir .ssh**
[dmckay@backup ~]$ **chmod 700 .ssh**
[dmckay@backup ~]$ **cp .ssh/id_rsa.pub .ssh/authorized_keys**
[dmckay@backup ~]$ **chmod 600 .ssh/authorized_keys**
[dmckay@backup ~]$ **cd .ssh**
[dmckay@backup ~]$ **ln -s authorized_keys authorized_keys2**
[dmckay@backup ~]$ **exit**
[dmckay@backup ~]$

[8] Testing: ssh to acme without password
[dmckay@backup ~] $ **ssh beehive**
[dmckay@beehive ~] $ **exit**
[dmckay@backup ~] $ **ssh beehive hostname**
beehive

[9] Backup on beehive
Create backup directory: (on beehive)
[dmckay@backup ~] $ **ssh beehive**
[dmckay@beehive ~] $ **sudo su –**
[root@beehive ~] $ **mkdir /backup**
[root@beehive ~] $ **tar –zcvf /backup/beehive.tgz /pool**
[root@beehive ~] $ **crontab -e**
01 4 * * * /bin/tar –zcvf /backup/beehive.tgz /pool

[10] Backup on backup
Create backup directory:
[dmckay@backup ~] $ **mkdir -p /backup/beehive**

Run rsync command to backup beehive on backup server:
[dmckay@backup ~] $ **rsync –auvz –e ssh dmckay@beehive:/backup/* /backup/beehive/**
[dmckay@backup ~] $ **ls -ha /backup/beehive/**

[dmckay@backup ~] $ **ssh beehive**
[dmckay@beehive ~] $ **mkdir –p CSE365/{HW,LAB}/{1,2,3,4,5,6,7,8,9,10}**
[dmckay@beehive ~] $ **tree $HOME**

[dmckay@beehive ~] $ **sudo tar –zcvf /backup/beehive.tgz /pool**
[dmckay@beehive ~] $ **exit**
[dmckay@backup ~] $ **rsync –auvz –e ssh dmckay@beehive:/backup/* /backup/beehive/**
[dmckay@backup ~] $ **ls -ha /backup/beehive/**
*** Please backup MySQL databases from www too.**

[11] Activate email:
[dmckay@backup ~] $ **sudo /sbin/chkconfig sendmail on**
[dmckay@backup ~] $ **sudo /sbin/service sendmail start**

[12] Write a backup script:
[dmckay@backup ~] $ **mkdir bin**
[dmckay@backup ~] $ **cd bin**
[dmckay@backup bin] $ **touch backup.bash**
[dmckay@backup bin] $ **chmod 700 backup.bash**
[dmckay@backup bin] $ **vi backup.bash**

#!/bin/bash
rsync –auvz –e ssh dmckay@beehive:/backup/* /backup/beehive/
echo `date` > mesg
/bin/df -h /backup/beehive >> mesg
echo >> mesg
echo "BEEHIVE:" >> mesg
ls -lh /backup/ >> mesg
cat mesg | /bin/mail -s "BEEHIVE BACKUP IS FINISHED" dmckay@coyote365.net #*(Use \ your real email address to test instead of dmckay@coyote365.net.)*
~
~
:wq

[13] Automate the backup task using crontab:
[dmckay@backup bin] $ **crontab –e**
0 4 * * 0 /pool/it/dmckay/bin/FullBackup.bash
0 4 * * 1-6 /pool/it/dmckay/bin/IncrBackup.bash

[14] Please backup hadrian on backup server:
[15] Please backup DNS servers on backup server:
[16] Please backup Web-server on backup server:

Lab 7 Report:

[1] Why do we create group on the each server?
[2] What will happen if DNS could not resolve the target server's IP address?
[3] Why you make user *dmckay* as sudo user?
[4] Why do you remove the password when *dmckay* login to others servers from backup server?
[5] What will happen if system time is not correct on backup server?
[6] Why do you make symbolic link as following in this laboratory?
[dmckay@beehive ~] $ **ln -s authorized_keys authorized_keys2**
[7] What is the difference between (1) and (2)?
(1) %wheel ALL=(ALL) NOPASSWD: ALL
(2) %wheel ALL=(ALL) ALL
[8] What following commands do?
[dmckay@backup ~] $ **rsync –auvz –e ssh dmckay@beehive:/backup/* /backup/beehive/**
[dmckay@backup ~] $ **ls -na /backup/beehive/**
[dmckay@backup ~] $ **ssh beehive**
[dmckay@beehive ~] $ **mkdir –p CSE365/{HW,LAB}/{1,2,3,4,5,6,7,8,9,10}**
[dmckay@beehive ~] $ **tree $HOME**
[dmckay@beehive ~] $ **sudo tar –zcvf /backup/beehive.tgz /pool**
[dmckay@beehive ~] $ **exit**
[dmckay@backup ~] $ **rsync –auvz –e ssh dmckay@beehive:/backup/* /backup/beehive/**
[dmckay@backup ~] $ **ls -ha /backup/beehive/**
[9] Explain following script.

```
#!/bin/bash
rsync –auvz –e ssh dmckay@beehive:/backup/* /backup/beehive/
echo `date` > mesg
/bin/df -h /backup/beehive >> mesg; echo >> mesg
echo "BEEHIVE:" >> mesg
ls -lh /backup/ >> mesg
cat mesg | /bin/mail -s "BEEHIVE BACKUP IS FINISHED" dmckay@coyote.net
```

[10] How to make backup.bash to run on every Sunday?
[dmckay@backup bin] $ **crontab –e**
01 4 * * * bin/backup.bash
[11] How to make backup.bash to run on every 15 minutes?
[dmckay@backup bin] $ **crontab –e**
01 4 * * * bin/backup.bash

[12] What did you learn from this lab?