Part II : Network Linux Services
Linux Services Organization : Linux DNS Linux Server

Domain Name Servers DNS, or nameserver, maps devices hostnames with their respective IP addresses. DNS is normally implemented using a central server/s that is authoritative for a domain and refer to other DNS servers for other domains. There are four DNS server configuration types :

Master
It has the authoritative zone records for the domain that act as DNS server. Answers directly queries about the authoritative domain and forwards other domain queries to other DNS servers.

Slave
Slave DNS server acts as an authoritative DNS server getting the zone records from the DNS master server.

Caching-only
Caching-only DNS server is not authoritative for any zone, all queries are forwarded to other DNS servers if they are not stored in the DNS-cache zone. Answers for all queries are cached in DNS-cache zone for a time.

Forwarding
As caching-only DNS server, forwarding DNS server is not authoritative for any zone, all queries are forwarded to a specific list of nameservers.
A nameserver can be master for some zones, slave for others and offer forwarding to others.

Packages

On RHEL6 DNS is based on the **named** daemon which is installed on the **bind** package developed through the Internet Source Consortium and some additional packages:

bind-chroot
Provides a isolated 'chroot-jail' which limit the access if DNS is compromised.

bind-devel
Includes development libraries from bind.

bind-libbind-devel
Contains the libbind resolve library.

bind-libs
Adds library files used by the bind and bind-utils packages.

bind-sdb
Supports alternative databases for bind.

bind-utils
Includes tools such dig that provides DNS information about an internet device.

DNS client

/etc/nsswitch.conf

When a linux computer looks for another computer IP it looks for the information in two files : **/etc/hosts** and **/etc/resolv.conf**. The order in which the files are consulted is configured on **/etc/nsswitch.conf**:

```
$ cat /etc/nsswitch.conf

hosts: files,dns
```

Search first on files (/etc/hosts) and then on dns (/etc/resolv.conf).

/etc/hosts

This file is a simple database that relates a numeric IP with a hostname. It can be edited as a normal file with 'vi' command in order to add more information.

```
# cat /etc/hosts
```

```
127.0.0.1 localhost.localdomain localhost
192.168.1.1 server.info.net server
```

The first line maps the 127.0.0.1 IP to the hostnames localhost, short hostname, and localhost.localdomain, FQHN hostname. The second line maps the 192.168.1.1 IP to server and server.info.net hostname.

/etc/resolv.conf

In order to configure a linux computer as a DNS client the file /etc/resolv.conf must be used.

```
# cat /etc/resolv.conf
```

```
search info.net
nameserver 192.168.1.1
```

In this case all DNS queries launched from the computer will be addressed to the nameserver on 192.168.1.1. If a short hostname is provided it will be complemented automatically with 'info.net' domain.

Note: By default if a DNS query is done and can be answered from /etc/hosts the nameserver configured on /etc/resolv.conf is not consulted. Only the information obtained from /etc/hosts is taken as valid.

DNS server with bind

In order to install bind nameserver service, the bind package must be installed :

```
# yum install bind
```

The bind nameserver root directory is on /var/named and the configuration files are stored on /etc as usual :

/etc/sysconfig/named
Configuration file that set up how is executed on the system the bind daemon.

/etc/named.conf
Main DNS configuration file that includes data from other files.

/etc/named.rfc1912.zones
It contains appropriate zones for localhost names and addresses.

/var/named/my.internal.zone.db
Zone file for the local network.

/var/named/slaves/my.slave.internal.zone.db
Zone file for a slave nameserver.

/var/named/localdomain.zone

Zone file for localhost domain.

/var/named/localhost.zone
Zone file for localhost computer.

/var/named/named.broadcast
Broadcast record for localhost.

/var/named/named.ca
List of the root DNS servers for the Internet consulted by nameserver when a nameserver resolution can not be done by the nameserver.

/var/named/named.ip6.local
IPv6 version of named.local.

/var/named/named.zero
Defaults to the broadcast record for the localhost.

/var/named/data/named.stats.txt
Nameserver statistics.

Templates of these files can be found on **/usr/share/doc/bind*/sample**. If the package **bind-chroot** is installed the nameserver root directory will be in **/var/named/chroot/var/named** and the configuration directory will be in **/var/named/chroot/etc**.

Caching-only nameserver

When a DNS query is performed against a cache-only name server the query will be forwarded to another nameserver if the answer for this query is not located on the dns caching-only name server cache. When the external nameserver answers to the DNS query the caching-only name server puts the answer on his cache and forwards the answer to the client who has made the query. If somebody repeats the same query against the caching-only name server it will be answered directly (faster) from the caching-only nameserver because the answer for this query will be on the cache.

In order to configure a caching-only name server for all your LAN the file /etc/named.conf installed by default by bind rpm must be changed in just two lines:

listen-on port 53 {127.0.0.1; }; --> **listen-on port 53 {127.0.0.1; 192.168.1.10; };**

allow-query {localhost; }; --> **allow-query{localhost; 192.168.1.0/24; };**

The first line will put the bind service listening on the localhost (127.0.0.1) and LAN (192.168.1.10) network interfaces. The second line will give access to all hosts on 192.168.1.0/24 LAN to use this server as nameserver, once the service will be started and the firewall open.

The configuration file for a caching-only nameserver is as follows :

```
# cat /etc/named.conf

options {
listen-on port 53 { 127.0.0.1; 192.168.1.10; };
listen-on-v6 port 53 { ::1; };
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
memstatistics-file "/var/named/data/named_mem_stats.txt";
allow-query { localhost; 192.168.1.0/24; };
recursion yes;

dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;

/* Path to ISC DLV key */
bindkeys-file "/etc/named.iscdlv.key";
};

logging {
channel default_debug {
file "data/named.run";
severity dynamic;
};
};

zone "." IN {
type hint;
file "named.ca";
};

include "/etc/named.rfc1912.zones";
```

Once this is done the next step is start bind daemon and make sure it will start on boot. Make sure that security considerations described on 'DNS security' are also applied. :

```
# /etc/init.d/named start
# chkconfig named on
```

With this configuration a cache-only nameserver will be serving name resolution on 192.168.1.10 for all 192.168.1.0/24 LAN. Have a look on lab1...

Forward nameserver

This kind of nameserver only needs a simple configuration option on /etc/named.conf file that configures the nameservers that the forwarding nameserver will forward all DNS queries. The /etc/named.conf from caching-only nameserver that is installed by default with bind RPM can be used to generate the forward nameserver configuration file just adding the following lines at 'option' section :

```
# Set nameserver type to forward
forward only;

# Configures the nameservers IPs where DNS queries will be forwarded
forwarders {192.168.1.20; 192.168.1.30; }
```

The configuration file for a forwarding-only nameserver is as follows :

```
# cat /etc/named.conf

options {
listen-on port 53 { 127.0.0.1; 192.168.1.10; };
listen-on-v6 port 53 { ::1; };
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
memstatistics-file "/var/named/data/named_mem_stats.txt";
allow-query { localhost; 192.168.1.0/24; };
recursion yes;

forward only;
forwarders {192.168.1.20; 192.168.1.30; };


dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;

/* Path to ISC DLV key */
bindkeys-file "/etc/named.iscdlv.key";
};

logging {
channel default_debug {
file "data/named.run";
severity dynamic;
};
};

zone "." IN {
type hint;
file "named.ca";
};

include "/etc/named.rfc1912.zones";
```

With this configuration the nameserver will be configured to forward all DNS queries that are not cached only to 192.168.1.20 , 192.168.1.30 DNS servers. The nameserver is caching + forward.

Once this is done the next step is start bind daemon and make sure it will start on boot. Make sure that security considerations described on 'DNS security' are also applied. :

```
# /etc/init.d/named start
# chkconfig named on
```

Master nameserver

When the nameserver is configured to serve name resolution for an specified domain (local domain or Internet domain) that server has the authoritative DNS records for that domain. This nameserver is consulted by other nameservers when a resolution for the domain where it is authoritative is performed on others servers.

In order to configure a nameserver as master nameserver for a domain the bind RPM must be installed. Next step is copy the file /usr/share/doc/bind*/sample/etc/named.conf to /etc/named.conf file (or /var/named/chroot/etc/named.conf if the package bind-chroot also has been installed) and perform the following changes :

```
# Make named daemon to listen on the nameserver IPv4 network IP (192.168.1.10 in this case) plus the localhost IP (127.0.0.1)
listen-on port 53 { 127.0.0.1; 192.168.1.10 };

# Allow only query from clients on your LAN plus localhost

allow-query { localhost; 192.168.1.0/24; };
allow-query-cache { localhost; 192.168.1.0/24; };

# Introduce on the 'view localhost_resolver' and 'view internal' the direct and reverse zone file. The direct file maps your domain hostnames with the numerical IP address, the reverse zone file maps the numerical IP values with their correspondents hostnames
```

```
view localhost_resolver {
...
# Direct authoritative zone file for our domain 'info.net'
zone "info.net" {
type master;
file "info.net.zone";
};

# Reverse authoritative zone file for our domain 'info.net'
zone "1.168.192.in-addr.arpa" IN {
type master;
file "info.net.rr.zone";
allow-update { none; };
};
};

# The same for 'view internal'. On this view substitute 'zone "my.internal.zone"' by 'zone "info.net"' and create the reverse zone 'view localhost_resolver'.

# On 'view internal' comment all 'zone "my.ddns.internal.zone', this nameserver is not going to be updated dynamically...

# In order to prevent unauthorized access to the named daemon, BIND uses a shared secret key authentication method to grant privileges to hosts. This means an identical key must be present in both /etc/named.conf and the rndc configuration file, /etc/rndc.conf.

key ddns_key {
algorithm hmac-md5;
secret "N7ypFzAWQrEo2nzwigHPKA==";
};

# Remove 'view external' section, in this case this nameserver is not going to allow DNS queries from clients outside the LAN.
```

After all this changes the file /etc/named.conf will be as follows :

```
# cat /etc/named.conf

options
{
// Put files that named is allowed to write in the data/ directory:
directory "/var/named"; // "Working" directory
dump-file "data/cache_dump.db";
statistics-file "data/named_stats.txt";
memstatistics-file "data/named_mem_stats.txt";


listen-on port 53 { 127.0.0.1; 192.168.1.10; };
//listen-on port 53 { 127.0.0.1; };

listen-on-v6 port 53 { any; };
//listen-on-v6 port 53 { ::1; };


allow-query { localhost; 192.168.1.0/24; };
allow-query-cache { localhost; 192.168.1.0/24; };

// Enable/disable recursion - recursion yes/no;
recursion yes;

/* DNSSEC related options. See information about keys ("Trusted keys", bellow) */
/* Enable serving of DNSSEC related data - enable on both authoritative
and recursive servers DNSSEC aware servers */
dnssec-enable yes;

/* Enable DNSSEC validation on recursive servers */
dnssec-validation yes;

/* Enable DLV by default, use built-in ISC DLV key. */
dnssec-lookaside auto;
};

logging
{
channel default_debug {
file "data/named.run";
severity dynamic;
};
};


view "localhost_resolver"
{
/* This view sets up named to be a localhost resolver ( caching only nameserver ).
* If all you want is a caching-only nameserver, then you need only define this view:
*/
match-clients { localhost; };
recursion yes;

# all views must contain the root hints zone:
zone "." IN {
type hint;
file "/var/named/named.ca";
};

/* these are zones that contain definitions for all the localhost
* names and addresses, as recommended in RFC1912 - these names should
* not leak to the other nameservers:
*/
include "/etc/named.rfc1912.zones";

zone "info.net" {
type master;
file "info.net.zone";
};

zone "1.168.192.in-addr.arpa" IN {
type master;
file "info.net.rr.zone";
allow-update { none; };
};
};
view "internal"
{
/* This view will contain zones you want to serve only to "internal" clients
that connect via your directly attached LAN interfaces - "localnets" .
*/
match-clients { localnets; };
recursion yes;

zone "." IN {
type hint;
file "/var/named/named.ca";
};

/* these are zones that contain definitions for all the localhost
* names and addresses, as recommended in RFC1912 - these names should
* not leak to the other nameservers:
*/
include "/etc/named.rfc1912.zones";

// These are your "authoritative" internal zones, and would probably
// also be included in the "localhost_resolver" view above :

//zone "mynternal.zone".internal.zone" {
// type master;
// file "my.internal.zone.db";
//};
zone "my.slave.internal.zone" {
type slave;
file "slaves/my.slave.internal.zone.db";
masters { /* put master nameserver IPs here */ 127.0.0.1; } ;
// put slave zones in the slaves/ directory so named can update them
};
//zone "my.ddns.internal.zone" {
// type master;
// allow-update { key ddns_key; };
// file "dynamic/my.ddns.internal.zone.db";
// // put dynamically updateable zones in the slaves/ directory so named can update them
//};

zone "info.net" {
type master;
file "info.net.zone";
};

zone "1.168.192.in-addr.arpa" IN {
type master;
file "info.net.rr.zone";
allow-update { none; };
};

};

key ddns_key
{
algorithm hmac-md5;
secret "N7ypFzAWQrEo2nzwigHPKA==";
};
...
```

Now is time to create the direct and reverse zone files on **/var/named** directory for our domain 'info.net' :

```
# cat /var/named/info.net.zone

$TTL 86400
@ IN SOA rhel6.info.net. root.rhel6.info.net. (
42 ; serial (d. adams)
3H ; refresh
15M ; retry
1W ; expiry
1D ) ; minimum
IN NS rhel6
IN MX 10 rhel6
node01 IN A 192.168.1.101
```

```
node02 IN A 192.168.1.102
rhel6 IN A 192.168.1.10
server IN CNAME rhel6
www IN CNAME rhel6
ftp IN CNAME rhel6
```

```
# cat /var/named/info.net.rr.zone
```

```
@ IN SOA rhel6.info.net. root.rhel6.info.net. (
1997022700 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS rhel6.info.net.
101 IN PTR node01.info.net.
102 IN PTR node02.info.net.
10 IN PTR rhel6.info.net.
```

Once the direct and reverse zone files has been created the ownership must be changed to named user and restart the named service :

```
# chown named:named info.net.zone info.net.rr.zone
# /etc/init.d/named restart
```

Now from node01, configuring rhel6 server as nameserver for info.net domain , we can test direct and reverse name resolution against info.net domain:

```
node01> cat /etc/resolv.conf
```

```
search info.net
nameserver 192.168.1.10
```

# Direct www.info.net name resolution

```
node01> dig www.info.net
...
;; ANSWER SECTION:
www.info.net. 86400 IN CNAME rhel6.info.net.
rhel6.info.net. 86400 IN A 192.168.1.10
...
;; SERVER: 192.168.1.10#53(192.168.1.10)
```

# Reverse 192.168.1.102 resolution

```
node01> dig -x 192.168.1.102
...
;; ANSWER SECTION:
102.1.168.192.in-addr.arpa. 86400 IN PTR node02.info.net.
...
;; SERVER: 192.168.1.10#53(192.168.1.10)
```

Slave nameserver

In order to configure a nameserver as slave nameserver for a domain, the following configuration must be added on the internal view :

```
zone "example.org" IN {
type slave;
file "slaves/example.com.org";
masters {
```

```
192.168.1.50
};
};
```

The task for a slave server is easier; it periodically checks with the master DNS server (in this case 192.168.1.50.) When it does, it automatically reads the master DNS server data and creates the slave.example.com. zone file in the /var/named/slaves directory.

The dig command can be used to force a zone transfer from the master nameserver to the slave nameserver. From slave nameserver execute the following command:

```
# dig @master_nameserver_ip example.org axfr
```

DNS security

As said before the bind nameserver service can be secured running it into an isolated chroot-jail on /var/named/chroot just installing the bind-chroot package. All the files described in this session will be valid from bind-chroot service, but they will be placed on the chroot-jail on /var/named/chroot :

```
# yum install bind-chroot
```

Nameserver service can be secured via firewall. The port 53 TCP/IP and UDP must be accessible to the clients :

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT
```

It can also be secured using SElinux, if it is active the nameserver service will be executed in a confined environment similar to the chroot jail installed by bind-chroot RPM. If this is the case the following SElinux parameter must be configured in order to get running nameserver service with SElinux :

```
# setsebool -P named_write_master_zones 1
```

DNS and Internet

DNS is an Internet-wide database that maps domain names and IP addresses. The information that goes into the database must be up to date and properly formatted. Many network problems are caused from poorly administered DNS servers :

***Time**
Changes made on your DNS connected to Internet can take more than 48h to be propagated on all Internet.

* **Increment the serial number**
Every time a change on the zone file is done the serial number parameter must be increased in order to advise other DNS servers of the change before the service restart. If the serial number is not increased the change is NOT propagated on Internet.

Questions

1.- DNS on rhel6 is implemented by bind daemon (true/false).
2.- In order to run DNS services on a chroot-jail the RPM named-chroot must be installed (true/false).
3.- One nameserver can be configured as caching-only nameserver and forward all unknown DNS queries to the same nameserver (true/false).
4.- One nameserver can be configured as caching-only nameserver and forward all unknown DNS queries to the nameservers on named.ca file (true/false).
5.- One nameserver can be configured as master for a domain and slave for another domain (true/false).
6.- Which command must be used in order to force a zone transfer for domain 'example.com' from nameserver 192.168.1.1 ?.

7.- Which command must be used in order ask for node01.info.net resolution to server 192.168.1.1 ?.
8.- Which command must be used in order ask for 192.168.1.102 resolution to server 192.168.1.10 ?.
9.- Which of the following files sets up the order in which the nameserver resolution is performed on a DNS client ?.
A - /etc/hosts
B - /etc/nsswitch.conf
C - /etc/resolv.conf
D - /etc/named.conf
9.- Which of the following commands must be used in order list all info.net DNS entries ?
A - dig info.net
B - host -l info.net
C - Both of them
D - None of them

Labs

1.- Configure a cache-only nameserver on your rhel6 server (192.168.1.10) for your LAN 192.168.1.0/24 . Configure node01 (192.168.1.101) as client for it and verify that rhel6 server is performing as expected.
2.- Configure a forward nameserver on your rhel6 server (192.168.1.10) for your LAN 192.168.1.0/24. Use as forwarders nameservers the google public DNS servers 8.8.8.8 and 8.8.4.4. Configure node01 (192.168.1.101) as client for it and verify that rhel6 server is performing as expected.
3.- Configure a 'chroot' master nameserver on your rhel6 server (192.168.1.10) for LAN (info.net) 192.168.1.0/24 and for 192.168.2.0/24 LAN (info.org). Configure node01 (192.168.1.101) as generic caching-only nameserver with a slave zone for LAN (info.org) 192.168.2.0/24 where the master nameserver is rhel6. Test direct/reverse resolutions and info.org zone transfer from rhel6 to node01.

1.- False. It is named daemon.
2.- False. It is bind-chroot RPM.
3.- True.
4.- True.
5.- True.
6.- dig @192.168.1.1 example.com axfr
7.- dig @192.168.1.1 node01.info.net
8.- dig @192.168.1.10 -x 192.168.1.102
9.- B
10.- B

Lab 1

* Login as root on rhel6 (192.168.1.10) and install bind rpm. Make sure rhel6 is connected to Internet and from there you can resolve 'www.google.com' to the numerical IP.

```
# yum install bind
```

```
# ping www.google.com
...get replies...
```

* Edit /etc/named.conf and change the necessary parameters in order to configure rhel6 as caching only server.

listen-on port 53 {127.0.0.1; }; --> **listen-on port 53 {127.0.0.1; 192.168.1.10; };**

allow-query {localhost; }; --> **allow-query{localhost; 192.168.1.0/24; };**

* Configure SElinux to allow bind to be running without problems

```
# setsebool -P named_write_master_zones 1
```

* Start named service. Make sure that it will started at boot time.

```
# /etc/init.d/named start
# chkconfig named on
```

* Check /var/log/messages to verify that bind service (named daemon) has started correctly.

```
# /etc/init.d/named status
```
(It will also report the daemon status)

* Configure the firewall to allow connections to port 53 TCP and UDP from any IP. Add the following lines on /etc/sysconfig/iptables just before deny all configuration :

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT
```

```
# /etc/init.d/iptables restart
```

* Login as root on node01 (192.168.1.101) and configure it to use rhel6 as DNS.

```
nameserver 192.168.1.10 --> /etc/resolv.conf
```

* From node01 resolve the www.linuxsv.org IP. Take note of the query time used to get the numerical resolution. See **';; Query time:'**

```
# dig www.linuxsv.org
...
;; ANSWER SECTION
linuxsv.org 60 IN A 95.60.149.133
...
;; Query time: 1659 msec
;; SERVER 192.168.1.10#53
```

From the answer we can see that www.linuxsv.org has the ip 95.60.149.133, the query has expended 1659 msec and the reply has come from our cache nameserver on 192.168.1.10.

* Repeat the name resolution.

```
# dig www.linuxsv.org
...
;; ANSWER SECTION
linuxsv.org 60 IN A 95.60.149.133
...
;; Query time: 189 msec
;; SERVER 192.168.1.10#53
```

Of course the numerical IP has not changed but have a look on the query time : 189 msec...**10 times faster than previous resolution!!!**. This is because the first query www.linuxsv.org had not been found on rhel6 nameserver cache and the server has forwarded the query to other nameserver on the Internet. On the second query www.linuxsv.org entry had been located on the rhel6 nameserver cache because of the first query, so rhel6 has answered directly to us using less time than in the first query.

Lab 2

* Login as root on rhel6 (192.168.1.10) and install bind rpm. Make sure rhel6 is connected to Internet and from there you can resolve 'www.google.com' to the numerical IP.

```
# yum install bind
```

```
# ping www.google.com
...get replies...
```

* Edit /etc/named.conf and add on 'options' sections the necessary configuration parameters in order to configure rhel6 as forward nameserver using google public DNS.

```
forward only;
forwarders{8.8.8.8; 8.8.4.4;};
```

* Make sure that /etc/named.conf is also configured to act as nameserver to 192.168.1.0/24 LAN.

```
listen-on port 53 {127.0.0.1; 192.168.1.10; };
allow-query{localhost; 192.168.1.0/24; };
```

* Configure SElinux to allow bind to be running without problems

```
# setsebool -P named_write_master_zones 1
```

* Start named service. Make sure that it will started at boot time.

```
# /etc/init.d/named start
# chkconfig named on
```

* Check /var/log/messages to verify that bind service (named daemon) has started correctly.

# /etc/init.d/named status (It will also report the daemon status)

* Configure the firewall to allow connections to port 53 TCP and UDP from any IP. Add the following lines on /etc/sysconfig/iptables just before deny all configuration :

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT
```

```
# /etc/init.d/iptables restart
```

* Login as root on node01 (192.168.1.101) and configure it to use rhel6 as DNS.

```
nameserver 192.168.1.10 --> /etc/resolv.conf
```

* From node01 resolve the www.yy.com IP. At the same time open on rhel6 an tcpdump session '**tcpdump -i eth1 port 53**' where eth1 is the network interface used to connect to Internet and verify that all not-cached DNS queries are forwarded to public DNS servers :

```
node01> dig www.yy.com
```

```
...
;; ANSWER SECTION
www.yy.com ... 61.158.248.3
...
;; SERVER 192.168.1.10#53
```

* At the same time on rhel6

```
rhel6> tcpdump -i eth1 port 53
...
IP 192.168.199.148.23910 > google-public-dns-a.google.com.domain: 44345+% [1au] A? www.yy.com. (39)
...
```

Lab 3

* Login as root on rhel6 (192.168.1.10) and install bind and bind-chroot rpms.

```
# yum install bind bind-chroot
```

* Generate /var/named/chroot/etc/named.conf from template file on /usr/share/doc/bind*/sample/etc/named.conf

```
--> listen-on port 53 { 127.0.0.1; 192.168.1.10 };
--> allow-query { localhost; 192.168.1.0/24; };
--> allow-query-cache { localhost; 192.168.1.0/24; };
```

Add direct/reverse 'info.net' and 'info.org' zones on views localhost_resolver and internal.

```
zone "info.net" {
type master;
file "info.net.zone";
};

zone "1.168.192.in-addr.arpa" IN {
type master;
file "info.net.rr.zone";
allow-update { none; };
};

zone "info.org" {
type master;
file "info.org.zone";
};

zone "2.168.192.in-addr.arpa" IN {
type master;
file "info.org.rr.zone";
allow-update { none; };
};
```

Add the dns security key from file /etc/rndc.conf

```
key ddns_key {
algorithm hmac-md5;
secret "N7ypFzAWQrEo2nzwigHPKA==";
};
```

* Remove 'view external' section

* Change the ownership of /var/named/chroot/etc/named.conf to root:named .

```
# chown root:named /var/named/chroot/etc/named.conf
```

* Create the direct/reverse zone file for 'info.net' and 'info.org' domains on directory /var/named/chroot/var/named. See 'Master DNS' section.

* Make sure that all this files have root:named ownership

* Restart named service and makes sure that the services is running on a 'chroot' jail.

```
# /etc/init.d/named restart
# ps auxwww | grep named

named 9795 0.0 2.2 49772 11512 ? Ssl 22:17 0:00 /usr/sbin/named -u named -t /var/named/chroot
```

* Login as root on node01 and install bind RPM.

```
# yum install bind
```

* Configure named.conf as caching-only + slave for 'info.org' from rhel6. Add the slave zone 'info.org'.

```
zone "info.org" IN {
type slave;
file "slaves/info.org.db";
masters { 192.168.1.10; };
};
```

* Apply all DNS security considerations and start named service on node01.

```
node01> /etc/init.d/named start
node01> chkconfig named on
```

* From traces on /var/log/messages you can see that a zone transfer from rhel6 to node01 has been done and the file /var/named/slaves/info.org.db has been created with all DNS information of 'info.org'. The zone transfer can be force manually with dig command :

```
node01> dig @192.168.1.10 info.org axfr

; <<>> DiG 9.3.4-P1 <<>> @192.168.1.10 info.org axfr
; (1 server found)
;; global options: printcmd
info.org. 86400 IN SOA server.info.org. root.server.info.org. 42 10800 900 604800 86400
info.org. 86400 IN NS server.info.org.
info.org. 86400 IN MX 10 server.info.org.
ftp.info.org. 86400 IN CNAME server.info.org.
node01.info.org. 86400 IN A 192.168.2.101
node02.info.org. 86400 IN A 192.168.2.102
server.info.org. 86400 IN A 192.168.2.10
www.info.org. 86400 IN CNAME server.info.org.
info.org. 86400 IN SOA server.info.org. root.server.info.org. 42 10800 900 604800 86400
;; Query time: 6 msec
;; SERVER: 192.168.1.10#53(192.168.1.10)
;; WHEN: Fri Feb 4 20:45:24 2011
;; XFR size: 9 records (messages 1)
```

* Of course direct and reverse resolutions on rhel6 master server for 'info.net' and 'info.org' zones must be working.

```
node01> dig @192.168.1.10 node02.info.net
...
node01> dig @192.168.1.10 node01.info.org
...
node01> dig @192.168.1.10 -x 192.168.1.101
...
node01> dig @192.168.1.10 -x 192.168.2.102
```

Linux Services Organization : Linux DHCP Linux Server

DHCP Dynamic Host Configuration Protocol allows network settings configuration for all clients from a central dhcp server. The dhcp clients request an IP address and other network settings to all dhcp servers listening on the local LAN where the client is connected. The dhcp server leases to the client one IP address based on the client MAC or just from a IP range, then the client accepts the configuration served by the dhcp server and notify it to the dhcp server.

DHCP server

In order to configure a server as dhcp server, the dhcp RPM package must be installed.

```
# yum install dhcp
```

The copy the sample configuration file from shared/doc to /etc/dhcpd/dhcpd.conf file.

```
# cp /usr/share/doc/dhcp*/dhcpd.conf.sample /etc/dhcpd/dhcpd.conf
```

Edit the sample config file with your LAN (192.168.1.0/24 in this case) network parameters configuration.

```
# cat /etc/dhcpd.conf

default-lease-time 600;
max-lease-time 7200;

# Use this to enble / disable dynamic dns updates globally.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.100 192.168.1.110;
option domain-name-servers 192.168.1.1;
option domain-name "192.168.1.1";
option routers 192.168.1.1;
option broadcast-address 192.168.1.255;
default-lease-time 600;
max-lease-time 7200;
}

# Hosts which require special configuration options can be listed in
# host statements.

host fantasia {
hardware ethernet 08:00:07:26:c0:a5;
fixed-address 192.168.1.200;
}
```

From this file can be seen that the dhcp server will serve the network configuration for 192.168.1.0/24 LAN providing IPs from the range 192.168.1.100-192.168.1.110. It also will configure the DNS server 192.168.1.1 on /etc/resolv.conf and default gateway on 192.168.1.1 for all clients. It will also reserve the IP 192.168.1.200 to the node with MAC 08:00:07:26:c0:a5 and it will call it fantasia.

Once the dhcp server has been configured the next step is start the service and make sure that it will be started on boot. It will start the dhcp service on the port 67/UDP.

```
# /etc/init.d/dhcpd start
# chkconfig dhcpd on
```

DHCP Security

In order to allow dhcp service through a firewall the port 67/UDP must be open on the dhcp server.

-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 67 -j ACCEPT

And the port 68/UDP must be open on the dhcp client.

-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 68 -j ACCEPT

If SElinux is interfering on the dhcpd service on the server, the dhcpd service will be disabled from SElinux protection.

# setsebool -P dhcpd_disable_trans 1

DHCP Client

The dhcp client configuration can be configured at the network device configuration file, /etc/sysconfig/network-script/ifcfg-eth0 for eth0. The following parameters must be used :

BOOTPROTO='dhcp'

The dhcp configuration for a network device as /dev/eth0 can be executed by hand with the dhclient command.

# dhclient eth0

In both cases using ifcfg-eth0 file or through 'dhclient' command the client node broadcast the LAN searching for dhcp configuration from a dhcp server.

Questions

1.- The dhcpd service can be started selectively on each interface using '/etc/init.d/dhcpd start ethx' where ethx is the network interface (true/false).
2.- The dhcpd service can server different network parameters for different LANs (true/false).
3.- The ip leased by the dhcp server are written on /var/lib/dhcpd/dhcpd.leases (true/false).
4.- Which command can be used in order to get an IPv6 address for the eth1 network interface.
5.- Which parameter must be used on the ifcfg-ethx file in order to not actualize the DNS on /etc/resolv.conf on the dhcp client.

Labs

1.- Configure rhel6 as dhcp server for your LAN 192.168.1.0/24 using the IP Range 192.168.1.20-192.168.1.29 . Configure rhel6 (192.168.1.10) as default gateway and DNS server and reserve the IP 192.168.1.30 for node01 with MAC 00:0C:29:E9:F1:75. Configure node01 the network interface with MAC 00:0C:29:E9:F1:75 to obtain the IP via dhcp at boot, verify the booked ip 192.168.1.30 is assigned from rhel6 dhcp server. Obtain a IP from dhcp range for interface eth1.

1.- True
2.- True.
3.- True.
4.- dhclient -6 eth1
5.- PEERDNS=no

Lab 1

* Login as root on rhel6 (192.168.1.10) and install dhcp rpm.

# yum install dhcp

* Copy the dhcpd.conf.sample from /usr/share/doc on /etc/dhcpd/dhcpd.conf file.

```
# cp /usr/share/doc/dhcp*/dhcpd.conf.sample /etc/dhcpd/dhcpd.conf
```

Edit the sample config file with your LAN (192.168.1.0/24 in this case) network parameters configuration.

```
# cat /etc/dhcpd.conf

default-lease-time 600;
max-lease-time 7200;

# Use this to enble / disable dynamic dns updates globally.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.20 192.168.1.29;
option domain-name-servers 192.168.1.10;
option domain-name "192.168.1.10";
option routers 192.168.1.10;
option broadcast-address 192.168.1.255;
default-lease-time 600;
max-lease-time 7200;
}

# Hosts which require special configuration options can be listed in
# host statements.

host node01 {
hardware ethernet 00:0C:29:E9:F1:75;
fixed-address 192.168.1.30;
}
```

* Login as root on node01 and configure eth0 to get the IP through dhcp.

* Edit /etc/sysconfig/network-script/ifcfg-eth0 file

```
DEVICE="eth0"
HWADDR="00:0C:29:E9:F1:75"
NM_CONTROLLED="no"
ONBOOT="yes"
BOOTPROTO="dhcp"
```

* Execute 'tail -f /var/log/messages &' command and restart the network service.

```
# tail -f /var/log/messages &
# /etc/init.d/network restart
```

* From traces can be seen that the reserved IP 192.168.1.30 has been assigned to eth0. Also the 192.168.1.10 DNS has been configured on /etc/resolv.conf

* Obtain a IP from dhcp server to eth1 interface from 'dhclient' command.

# dhclient eth1
...

The first IP available on the dhcp server IP Range, 192.168.1.20 in this case, is assigned to eth1.

Linux Services Organization : Linux NTP Linux Server

NTP (Network Time Protocol) allows a system to sync its time clock with a time server. Time synchronization on IT infrastructures is critical, for example if time system of a node in a cluster is too different from the rest of the nodes the cluster software will think that this node is not responding and automatically will be removed from the cluster.

Configuring a local NTP server

To keep sync the time clocks of all nodes in a LAN, a local NTP server can be configured. All nodes on the LAN will keep time clock sync with the local NTP server using the NTP protocol, and the local NTP server will be in sync with other NTP servers on the Internet.

To configure a server as local NTP server, the ntp RPM must be installed.

# yum install ntp

Edit /etc/ntp.conf file and uncomment the line that allows access to all nodes on your LAN (192.168.1.0/24)

restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap--> /etc/ntp.conf

Note: local NTP server will be synchronized with other public NTP servers on Internet listed on '**server**' directive on /etc/named.conf. Examples **0.rhel.poll.ntp.org, 1.rhel.poll.ntp.org, ...**

Restart ntpd service with the new configuration and make sure it will started on boot.

# /etc/init.d/ntpd restart
# chkconfig ntpd on

NTP server security

Some security considerations have to be taken in order to run ntpd service secure. The first one is open the NTP server firewall to allow connections from/to other ntp client/server.

-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 123 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 123 -j ACCEPT

Note: NTP clients must keep open the 123/UDP port on the firewall.

By default the ntpd services is protected by SElinux. In order to disable this protection, just in case of causing problems :

# setsebool -P ntpd_disable_trans 1

NTP clients

In order to configure the ntpd service on a Linux node as client for your LAN NTP server just add the local NTP server IP on 'server' directive in /etc/ntp.conf . Make sure that port 123/UDP is open on your firewall and the service ntpd is up and running.

# /etc/init.d/ntpd restart
# chkconfig ntpd on

In order to force a NTP synchronization the command ntpdate can be used. For example if we want to sync our time clock from NTP server 192.168.1.10 :

```
ntpdate -u 192.168.1.10
```

Note : if the NTP server used has not terminated the calculation of its drift time (/var/lib/ntp/driftime file), the NTP server will be not ready to be used and the message "**no server suitable for synchronization found**" will be displayed on the ntp client. Just be patient and wait, sometimes takes hours ...

Questions

1.- A node can be configured as ntp server and client at the same time (true/false).
2.- NTP client must have the port 123/tcp (true/false).
3.- NTP server must have the port 123/tcp open in order to be operational (true/false).
4.- Which command forces time sync from ntp client using ntp server on ntpserver.info.net.
5.- Which command shows on ntp client the ntp servers where the ntp client is in sync.

Labs

1.- Configure rhel6 as NTP server for your LAN 192.168.1.0/24. Configure node01 (192.168.1.101) as client NTP for it and verify the time sync.

1.- True.
2.- False, 123/UDP.
3.- True.
4.- ntpdate -u ntpserver.info.net
5.- ntpq -p

Lab 1

* Login as root on rhel6 (192.168.1.10) and install ntp rpm.

```
# yum install ntp
```

* Edit /etc/ntp.conf file and uncomment the line that allows access to all nodes on your LAN (192.168.1.0/24)

restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap--> /etc/ntp.conf

Note: local NTP server will be synchronized with other public NTP servers on Internet listed on '**server**' directive on /etc/named.conf so make sure that rhel6 has access to the Internet. Examples **0.rhel.poll.ntp.org, 1.rhel.poll.ntp.org, ...**

Restart ntpd service with the new configuration and make sure it will started on boot.

```
# /etc/init.d/ntpd restart
# chkconfig ntpd on
```

* Open port 123/TCP and 123/UDP on firewall.

```
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 123 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 123 -j ACCEPT
```

* Make sure that SElinux will not interfere with ntp.

```
# setsebool -P ntpd_disable_trans 1
```

* Execute the command 'ntpq -p' in order to verify the NTP server on rhel6 with external NTP servers.

* Login as root on node01 (192.168.1.101) and install ntp rpm.

```
node01> yum install ntp
```

* Edit /etc/ntp.conf file and add 192.168.1.10 as NTP server for this NTP client

```
server 192.168.1.10 --> /etc/ntp.conf
```

* Open port 123/UDP on firewall.

```
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 123 -j ACCEPT
```

* Make sure that SElinux will not interfere with ntp.

```
node01> setsebool -P ntpd_disable_trans 1
```

* Execute a ntp query against rhel6 NTP server to verify the rhel6 sync status.

```
node01> ntpdate -q 192.168.1.10

server 192.168.1.10, stratum 16, offset 16742.980064, delay 0.02586
15 Feb 10:44:03 ntpdate[1685]: no server suitable for synchronization found
```

As can be seen the NTP server on rhel6 is not ready to server as NTP server, the stratum is 16 and for be operational the stratum must be under 16. ... one hour later ...

```
node01> ntpdate -q 192.168.1.10

server 192.168.1.10, stratum 3, offset -0.030818, delay 0.02585
15 Feb 16:15:42 ntpdate[1854]: adjust time server 192.168.1.10 offset -0.030818 sec
```

stratum 3 --> NTP server ready on rhel6

* NTP server on rhel6 is ready so change the clock time on node01 to be two minutes less than rhel6 clock time.

```
node01> date -s "16:08"
```

* Restart ntpd service on node01 and verify that clock time on node01 is in sync with time clock on rhel6, NTP server.

```
node01> /etc/init.d/ntpd restart
node01> chkconfig ntpd on

rhel6> date
Tue Feb 15 16:13:55 CET 2011

node01>date
Tue Feb 15 16:13:55 CET 2011
```

Linux Services Organization : Linux NFS Linux Server

NFS Network File System is a server-client protocol used for sharing files on Unix, Linux systems. It allows sharing files from a central server allowing several users to access and modify the same files from different clients making all the changes on the files visible on all clients. The NFS server/client can be running on different S.O. without any problem.

NFS Server

In order to configure a node as NFS server the packages 'nfs-utils' and 'rpcbind' must be installed and running :

```
# yum install rpcbind nfs-utils
# /etc/init.d/nfs start
# /etc/init.d/nfslock start
# /etc/init.d/rpcbind start
# chkconfig nfs on
# chkconfig nfslock on
# chkconfig rpcbind on
```

NFS uses portmap (rpcbind) services to export files/directories to other nodes so rpcbind service must be running. The nfs service starts the following processes :

*** rpc.mountd**
Handles mount requests.

*** nfsd**
Starts an nfsd kernel process for each shared directory.

*** rpc.rquotad**
Reports disk quota statistics to clients.

*** rpc.svcgssd**
Gives a security api to provide security for protocols using rpc (in particular, nfs).

If any of these processes is not running, NFS won't work properly. In order to check if any of these processes are running the command 'rpcinfo -p' can be used :

```
# rpcinfo -p
```

/etc/exports

This is the most important file on NFS server configuration. It lists the directories to be exported from the NFS server to clients, the hosts to which it will be exported, and the options that apply to this export. The line format is the following :

(directory) (host1)(options1) (hostN)(options2)

Where **directory** is a local directory on the NFS server to be exported to **host1** with options **options1** and to **host2** with**options2**. Lets se some examples :

```
# cat /etc/exports

/pub (ro,insecure,sync) node01.info.net(rw,insecure,sync)
/home *.info.net(rw,insecure,sync)
/tftpboot diskless.info.net(rw,insecure,no_root_squash,sync)
/tmp 192.168.1.0/24(ro,sync,no_wdelay)
```

The first line exports the /pub directory in read-only mode to everybody less to node01.info.net that is exported in read-write mode. The second line the /home directory is exported in read-write mode to all computers on info.net domain. On the third line the directory /tftpboot is exported in read-write mode to all users (included root) to the computer diskless.info.net. Finally on last line the directory /tmp is exported in read-only mode to all nodes on LAN 192.168.1.0/24.

The sync flag requires all changes to be written to disk before a command such as a file copy is complete. The insecure flag allows access on ports above 1024.

Once configured /etc/exports file the command 'exportfs -a' must be executed to notify nfs daemon for the new exports :

```
# exportfs -a
```

NFS Security

Firewall

NFS uses portmap in order to execute some of its network services and by default portmap uses random UDP/TCP ports. This can be a problem when NFS must be configured in order to offer service through a firewall because this ports are randomly assigned. Solution: fix the ports used by NFS-portmap services and then open this ports on the firewall. This action can be done configuring/fixing the NFS-portmap ports on **/etc/sysconfig/nfs** file.

```
# cat /etc/sysconfig/nfs

...
RQUOTAD_PORT=60001
...
LOCKD_TCPPORT=60002
...
LOCKD_UDPPORT=60002
...
MOUNTD_PORT=60003
...
STATD_PORT=60005
...
STATD_OUTGOING_PORT=60004
...
```

In order to apply this changes correctly the best way is **reboot** the node and verify that the NFS-portmap port used are fixed.

```
# netstat -putan | grep rpc

tcp 0 0 0.0.0.0:60001 0.0.0.0:* LISTEN 1785/rpc.rquotad
tcp 0 0 0.0.0.0:60003 0.0.0.0:* LISTEN 1801/rpc.mountd
tcp 0 0 0.0.0.0:60005 0.0.0.0:* LISTEN 1258/rpc.statd
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 1233/rpcbind
tcp 0 0 :::60005 :::* LISTEN 1258/rpc.statd
tcp 0 0 :::111 :::* LISTEN 1233/rpcbind
udp 0 0 0.0.0.0:984 0.0.0.0:* 1233/rpcbind
udp 0 0 0.0.0.0:60001 0.0.0.0:* 1785/rpc.rquotad
udp 0 0 0.0.0.0:60003 0.0.0.0:* 1801/rpc.mountd
udp 0 0 0.0.0.0:60005 0.0.0.0:* 1258/rpc.statd
udp 0 0 0.0.0.0:111 0.0.0.0:* 1233/rpcbind
udp 0 0 0.0.0.0:1010 0.0.0.0:* 1258/rpc.statd
udp 0 0 :::984 :::* 1233/rpcbind
udp 0 0 :::60005 :::* 1258/rpc.statd
udp 0 0 :::111 :::* 1233/rpcbind
```

Now the ports has been fixed and is time to open the ports on the firewall.

```
# cat /etc/syscopnfig/iptables
...
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60001 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 60001 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60003 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 60003 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60004 -j ACCEPT
```

```
-A INPUT -m state --state NEW -m udp -p udp --dport 60005 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 60005 -j ACCEPT
```

Port 2049 TCP/UDP is from the NFS service itself, the port 111 TCP/UDP is portmap (rpcbind) and the rest are the NFS-portmap ports fixed on /etc/sysconfig/nfs file.

```
# /etc/init.d/iptables restart
```

SElinux

If your server is protected by SElinux some SElinux-NFS booleans can be configured in order to allow NFS process to be executed without any SElinux interference. By default all this values are enabled.

```
# setsebool -P allow_gssd_read_tmp 1
```

Supports the reading of temporary directories by the General Security Services daemon, gssd, which helps protect NFS using Kerberos

```
# setsebool -P allow_nfsd_anon_write 1
```

Supports NFS servers when they modify files on public file transfer services.

```
# setsebool -P nfs_export_all_ro 1
```

Supports read-only access to shared NFS directories.

```
# setsebool -P nfs_export_all_rw 1
```

Supports read-write access to shared NFS directories.

```
# setsebool -P use_nfs_home_dirs 1
```

Allow the export of home directories.

If a new directory (/newdir) has been created on the NFS server, in order to export it in read-write mode it must be labelled with the 'public_content_rw_t' SElinux label if we want write on it assuming that the mentioned SElinux-NFS booleans has been configured.

```
chcon -R -t public_content_rw_t /newdir
```

Security Considerations

NFS includes a number of security problems and should never be used in hostile environments such as the Internet. The main security problems can be listed as :

* Portmap (rpcbind) services used on NFS Server/Client transactions are not secure.

* Authentication relies on the host to report user and group IDs. This can expose your files if root users on other computers access your NFS shares as root. Data that is accessible via NFS to any user can potentially be accessed by any other user. By default, NFS is set up to **root_squash**, which prevents root users on an NFS client from gaining root access on the NFS server.

* Be careful with any symbolic links on an exported directory. The client interprets a symbolically linked file with respect to its own local filesystem.

NFS Client

When you start NFS-Client command such as 'mount' the following processes are executed :

rpc.statd
Control the status of NFS servers from the client.

rpc.lockd

Manages file locking on client side. It allows simultaneous access to the same file from different NFS-Clients.

In order to list the NFS shares accessible from a client the command 'showmount' can be used :

```
# showmount -e rhel6
```

```
Export list for rhel6:
/tmp 192.168.1.0/24
```

As we can see the /tmp NFS share is exported from rhel6 NFS server to all nodes on LAN192.168.1.0/24. In order to mount that NFS share on a local directory the command 'mount can be used' :

```
# mount -t nfs -o defaults,soft,timeo=100 rhel6:/tmp /mnt
```

With this command the NFS share /tmp exported from rhel6 server has been mounted locally on /mnt directory in 'soft,timeo=100' mode that makes that the NFS client does not crash in case of connectivity problems with the NFS server. Just use the 'mount' command to verify it :

```
# mount
...
rhel6:/tmp on /mnt type nfs (rw,soft,timeo=100,vers=4,addr=192.168.1.10,clientaddr=192.168.1.101)
```

As can be seen /tmp NFS share from rhel6 (192.168.1.10) has been mounted on /mnt of the NFS client (192.168.1.101) in read-write mode.

Lets try to write on it.

```
# touch /mnt/file
```

```
# ls -lrt /mnt/file
```

```
-rw-r--r--. 1 nfsnobody nfsnobody 0 Feb 17 2011 /mnt/file
```

Note that the file created belongs to 'nfsnobody' user and not to 'root'.

The file '/etc/fstab' can be used in order to mount NFS shares on boot time. For the case exposed before just add the corresponding line on /etc/fstab.

```
# echo "rhel6:/tmp /mnt nfs defaults,soft,timeo=100 0 0" >> /etc/fstab
```

NFS over UDP

NFS server runs over TCP by default but shares can be mounted on NFS clients using udp protocol :

```
# mount -o udp rhel6.info.net:/tmp /mnt
```

This action increases the performance on read-write operations on the NFS share but it decreases the fault tolerance because of TCP is connection oriented and UDP is not. You can read/write faster on the NFS share but the probability of get/create corrupted data higher than using TCP protocol.

NFS Limitations

Some limitations derived from NFS architecture has to be considered :

* The stateless nature of the NFS protocol makes that the NFS client wait if the NFS server ever has to be rebooted. The client waits, and waits, and waits ... Using the '**soft**' option when mounting NFS filesystems, when an NFS server fails, a soft-mounted NFS filesystem will fail rather than hang.

* NFS relies heavily on DNS direct and reverse name resolution. If NFS can't find a host name, rpc.mountd will deny

access to that client. For security reasons, it also adds a "request from unknown host" entry in /var/log/messages.

Questions

1.- When new NFS shares are added on NFS server, the server will export it automatically (true/false).
2.- NFS protocol can be used only in Linux S.O. (true/false).
3.- Mounting NFS shares using udp protocol increases the fault tolerance on read/write operations over the share (true/false).
4.- Which configuration parameter is used to force writes to NFS shares immediately.
5.- Which configuration parameter is used in order to allow root user on the NFS client be root user on the NFS server.
6.- Which configuration parameters must be used when mounting a NFS share in order to avoid crashes on the NFS client in case of connectivity problems with the NFS server.
7.- Which command can be used in order to reload the information about new shares added on NFS server?.
8.- Which command can be used in order to show the NFS shares exported from server.info.net?.
9.- Which of the following configurations will export the /home directory in read/write mode only to 192.168.1.101 node ?.

A - /home 192.168.1.101/24(ro)
B - /home 192.168.1.101/24 (rw)
C - Both of them
D - None of them
10.- Which of the following files specifies the list of NFS shares exported through NFS?.

A - /etc/exportfs
B - /etc/exports
C - /etc/export
D - /etc/fstab

Labs

1.- Configure rhel6 as NFS server and export /var/log/ directory in read-only mode to all nodes on 192.168.1.0/24 LAN. Configure node01 (192.168.1.101) to mount it at boot on /mnt/rhle6log directory. Make sure the NFS server on rhel6 node works correctly through a firewall and SElinux.
2.- Export /home from rhel6 NFS server in read/write mode to all nodes on 192.168.1.0/24. Configure node01 (192.168.1.101) to mount it on boot on local /home. Create user john (uid=1001) on rhel6 and node01 with home directory on /home/john. Make sure that john can write on his home directory /home/john on node01, exported from rhel6.
3.- As user john on node01 measure the time expended by user john to create a 100MB file /home/john/file. Remount /home on node01 and mount it using UDP protocol. Again measure the time expended by user john to create the same 200MB file on /home/john/file. Explain the time difference.
1.- False. NFS service restart or 'exportfs -a' command is required.
2.- False. It can be used in all *nix S.O.
3.- False. UDP protocol is connectionless so fault tolerance is lower than TCP protocol.
4.- 'sync'
5.- 'no_root_squash'
6.- 'mount -o defaults,soft,timeo=100 ...'
7.- 'exportfs -a'
8.- 'showmount -e server.info.net'
9.- D. B has an error, an space between the LAN ip and NFS options --> result /home is exported to everybody !!!
10.- B

Lab 1

* Login as root on rhel6 (192.168.1.10) and make sure that rpmdind and nfs-utils RPMS are installed.

* Export the share /var/log/ in read-only mode to 192.168.1.0/24 LAN through /etc/exports file.

```
# echo "/var/log 192.168.1.0/24(ro,sync)" >> /etc/exports
# exportfs -a
```

* Fix the rpmbind ports used by nfs and enable all the NFS ports on rhel6 server.

```
cat /etc/sysconfig/nfs

...
RQUOTAD_PORT=60001
...
LOCKD_TCPPORT=60002
...
LOCKD_UDPPORT=60002
...
MOUNTD_PORT=60003
...
STATD_PORT=60005
...
STATD_OUTGOING_PORT=60004
```

* Open all NFS related ports on the firewall.

```
# cat /etc/syscopnfig/iptables
...
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60001 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 60001 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60003 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 60003 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60004 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 60005 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 60005 -j ACCEPT
```

* Reboot rhel6 in order to take effect.

```
# reboot
```

* Connect as root on node01, create the directory /mnt/rhel6log and modify /etc/fstab to mount /var/log share from rhel6 on it at boot.

```
node01> mkdir -p /mnt/rhel6log
node01> echo "rhel6:/var/log /mnt/rhel6log nfs defaults,soft,timeo=100 0 0" >> /etc/fstab
```

* Verify that share rhel6:/var/log is exported read-only from rhel6 and mount it on node01 as specified on /etc/fstab.

```
node01> showmount -e rhel6

Export list for rhel6:
/var/log 192.168.1.0/24

node01> mount -a

node01> mount | grep rhel6log

rhel6:/var/log on /mnt/rhel6log type nfs
(rw,soft,timeo=100,vers=4,addr=192.168.1.10,clientaddr=192.168.1.101)
```

The share has been mounted on read-write mode but it has been exported on read-only mode from rhel6 so the net result is the share mounted on read-only mode.

```
node01> cat /mnt/rhel6log/dmesg
...
node01> vi /mnt/rhel6log/dmesg
Permission denied.
```

Lab 2

* Connect on rhel6 as root, export /home from rhel6 NFS server in read/write mode to all nodes on 192.168.1.0/24.

```
# echo "/home 192.168.1.0/24(rw,sync)" >> /etc/exports
# exportfs -a
```

* Make sure that SElinux will allow homes dir exports from rhel6. It is enabled by default.

```
# setsebool -P use_nfs_home_dirs 1
```

* Create user john (uid=1001)

```
# useradd -u 1001 john
```

* Connect as root on node01 and configure /etc/fstab to mount /home from rhel6 on local /home at boot.

```
node01> echo "rhel6:/home /home nfs defaults,soft,timeo=100 0 0" >> /etc/fstab
```

* Verify that share rhel6:/home is exported from rhel6 and mount it on node01 as specified on /etc/fstab.

```
node01> showmount -e rhel6
Export list for rhel6:
/home 192.168.1.0/24
/var/log 192.168.1.0/24

node01> mount -a

node01> mount |grep /home
rhel6:/home on /home type nfs (rw,soft,timeo=100,vers=4,addr=192.168.1.10,clientaddr=192.168.1.101)
```

* Login as user john on node01 and verify that he has read/write access to his home dir /home/john, exported from rhel6.

```
node01> su - john

john> echo file > /home/john/file
john> ls -lrt /home/john/file
-rw-rw-r--. 1 john john 5 Feb 20 2011 /home/john/file
```

Lab 3

* Connect on node01 as john and measure the time expended for creating a 100MB file /home/john/file. (/home is still exported from rhel6 as in previuos labs)

```
john> dd if=/dev/zero of=/home/john/file bs=1024 count=100000
100000+0 records in
100000+0 records out
102400000 bytes (102 MB) copied, 12.8351 s, 8.0 MB/s
```

* Logout as john, login as root on node01, umount /home and mount it using udp protocol.

```
# umount /home
# mount -o udp rhel6:/home /home
```

* Login again as john and create the same 100MB file.

```
# su - john

john> dd if=/dev/zero of=/home/john/file bs=1024 count=100000
100000+0 records in
100000+0 records out
102400000 bytes (102 MB) copied, 8.34394 s, 12.3 MB/s
```

Using NFS over UDP has taken **8.3 seconds** is create /home/john/file in comparation with the **12.8 seconds** taken to create the same file with NFS over TCP. NFS over UDP is faster than NFS over TCP BUT is less fault tolerant.

Linux Services Organization : Linux FTP Linux Server

FTP File Transfer Protocol allows file transfer between hosts on a network without having to login on a standard shell directly on the remote host. The file transfer is done using a standard set of simple commands without encryption, so it must be used only in a not hostile environment. Features like 'scp' that uses ssh protocol for encrypted file transfer can be used to file transfers on a hostile environment like Internet.

FTP Server

In order to configure a host as a FTP server the package vsftp must be installed, configured through /etc/vsftpd/vsftpd.conf and configured to be started at boot.

```
# yum install vsftpd
# chkconfig vsftpd on
# /etc/init.d/vsftpd start
```

/etc/vsftpd/vsftpd.conf

This is the main configuration file and specifies the way that the FTP server runs. The most important parameters that can be configured are the following :

anonymous_enable=YES
It allows FTP transfer using the anonymous user with password anonymous.

local_enable=YES
Local accounts are valid FTP accounts.

write_enable=YES
Enables write operations on FTP.

#anon_upload_enable=YES
It allows anonymous user to upload files. By default this line is commented so the anonymous user by default con not upload files to the FTP server.

#chroot_list_enable=YES
With chroot_local_user=YES you can configure users who are logged on FTP server to be confined in to their home directory on the FTP server. Disabled by default.

pam_service_name=vsftpd
Configures Pluggable Authentication Modules (PAM) security for FTP.

userlist_enable=YES
Keeps users such as root and system user listed on /etc/vsftpd/user_list from logging into the FTP server. It must be

activated always !!!

tcp_wrappers=YES
Supports the use of security commands in /etc/hosts.allow and /etc/hosts.deny through tcpwrappers

FTP Security

Firewall

The FTP server listen on port 21 TCP so it must be open on the firewall .

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
```

In the case of FTP server is also required to load the nat iptable module that keep track all FTP connections and allows it. This configuration is applied on **/etc/sysconfig/iptables-config** file :

```
IPTABLES_MODULES="nf_conntrack_ftp"-->/etc/sysconfig/iptables-config
```

```
# /etc/init.d/iptables restart
```

SElinux

There are five directives associated with making FTP server work with SELinux in targeted mode:

```
# setsebool -P allow_ftpd_full_access 1
```
If this parameter is enabled ftpd will run on a SElinux context without any restriction.

```
# setsebool -P allow_ftpd_anon_write 1
```
Supports the writing of files to directories configured with the public_content_rw_t SELinux setting.

```
# setsebool -P allow_ftpd_use_cifs 1
```
Allows the use of files shared via CIFS on an FTP server.

```
# setsebool -P allow_ftpd_use_nfs 1
```
Allows the use of files shared via NFS on an FTP server.

```
# setsebool -P ftp_home_directory 1
```
Supports FTP read/write access to user home directories.

In addition any directory that is going to be used on read-write FTP operations it must be labelled as 'public_content_rw_t' SElinux attribute in order to work correctly in SElinux targered mode .

```
# chcon -R -t public_content_rw_t /var/pub/ftp
```

FTP anonymous server

In this section we are going to configure a FTP server on rhel6 server and only allow anonymous login. Only downloading data from FTP server must be allowed files, uploading must be forbidden.

```
# cat /etc/vsftpd/vsftp.conf | grep -v ^#
```

```
anonymous_enable=YES
local_enable=NO
write_enable=NO
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
```

```
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

Configure the firewall as defined on '**FTP Security**'. If SElinux is running on targered mode the easy way applied in this case is give full access to the ftpd daemon on SElinux context :

```
# setsebool -P allow_ftpd_full_access 1
```

Lets create a file on the root of the ftp directory **/var/ftp/pub**. This file will be downloaded by anonymous user.

```
# dd if=/dev/null of=/var/ftp/pub/file bs=1024 count=1000
```

And finally restart the ftp service. Make sure the service starts correctly watching logs on /var/log/messages.

```
# /etc/init.d/vsftpd restart
```

From another host login to the FTP server on rhel6 as anonymous user using the FTP client 'lftp'. Download file created previously and verify that uploading is forbidden.

```
node01> lftp 192.168.1.10
lftp 192.168.1.10:~> cd pub
cd ok, cwd=/pub
lftp 192.168.1.10:/pub> ls
-rw-r--r-- 1 0 0 10240000 Feb 22 20:36 file
lftp 192.168.1.10:/pub> get file
10240000 bytes transferred
```

By default the FTP client 'lftp' login as anonymous. From there file has been downloaded correctly. Lets try to download a file :

```
lftp 192.168.1.10:/pub> put anaconda-ks.cfg
put: Access failed: 550 Permission denied. (anaconda-ks.cfg)
```

Uploads are not allowed.

```
lftp 192.168.1.10:/> cd /var
cd: Access failed: 550 Failed to change directory. (/var)
```

Navigate outside the FTP server is not allowed.

Try to login as other user as anonymous and verify that only anonymous logins are permitted.

```
node01> lftp -u john
Password:
lftp john@:~> ls
ls: Not connected
```

The same is we try as root and other users ... only anonymous logins are allowed.

FTP non-anonymous server

In this case we are going to configure an FTP server on rhel6 that must only allow logins to all system users less the listed on /etc/vsftpd/user_list . Download/upload must be allowed for these users.

```
cat /etc/vsftpd/vsftpd.conf | grep -v ^#
```

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

Configure the firewall as defined on '**FTP Security**'. In this case we are going to configure ftpd to run on SElinux environment. This is not the easy way as in previous example :

```
# setsebool -P allow_ftpd_full_access 0
# setsebool -P allow_ftpd_anon_write 1
# setsebool -P allow_ftpd_use_cifs 1
# setsebool -P allow_ftpd_use_nfs 1
# setsebool -P ftp_home_dir 1
```

Lets create a file on 'john' /home dir of the ftp directory **/home/john**. This file will be downloaded by user john.

```
# cp /var/ftp/pub/file /home/john
# chown john:john /home/john/file
```

And finally restart the ftp service. Make sure the service starts correctly watching logs on /var/log/messages.

```
# /etc/init.d/vsftpd restart
```

From another host login to the FTP server on rhel6 as 'john' user using the FTP client 'lftp'. Download file created previously and verify that uploading is allowed on john /home.

```
node01> lftp -u john 192.168.1.10
Password:
lftp john@192.168.1.10:~> ls
-rw-r--r-- 1 1001 1001 10240000 Feb 22 22:08 file
lftp john@192.168.1.10:~> get file
10240000 bytes transferred
lftp john@192.168.1.10:~> put install.log
21820 bytes transferred
lftp john@192.168.1.10:~> ls
-rw-r--r-- 1 1001 1001 10240000 Feb 22 22:08 file
-rw-r--r-- 1 1001 1001 21820 Feb 23 20:06 install.log
lftp john@192.168.1.10:~> cd /var
lftp john@192.168.1.10:/var>
```

As can be seen 'john' user can download/upload files on /home/john through FTP. But 'john' still has access to directories outside his home, on **Lab1** we will configure the FTP server to chroot users onto his home directory.

Users listed on /etc/vsftpd/user_list are not allowed to login on FTP server :

```
node01> lftp -u root 192.168.1.10
Password: lftp root@192.168.1.10:~> dir
`ls' at 0 [Delaying before reconnect: 20]
...
```

FTP Client

As has been seen in previous sections the **lftp** RPM can be used as FTP Client.

```
# yum install lftp
```

In order to login as user 'john' on FTP server 192.168.1.10 :

```
lftp -u john 192.168.1.10
Password:
lftp john@192.168.1.10:~>
```

If no user is specified the FTP login is done using the anonymous user.

In order to execute a remote command on the FTP server as 'ls' :

```
lftp john@192.168.1.10:~> ls
-rw-r--r-- 1 1001 1001 10240000 Feb 22 22:08 file
-rw-r--r-- 1 1001 1001 21820 Feb 23 20:06 install.log
```

In order to execute a local command on the FTP client as 'ls' :

```
lftp john@192.168.1.10:~> ! ls
file install.log install.log.syslog test
```

To download a file from FTP server use 'get' command :

```
lftp john@192.168.1.10:~> get file
10240000 bytes transferred
```

To upload a file from FTP client to the FTP server use 'put' command :

```
lftp john@192.168.1.10:~> put install.log
21820 bytes transferred
```

More info on '**man lftp**'.

Questions

1.- By default FTP data transfer is encrypted (true/false).
2.- FTP protocol can be used to transfer files between Linux, Unix and Microsoft Windows S.O. (true/false).
3.- In order to get working an FTP server through a firewall the only action required is open 21/TCP port (true/false).
4.- Which configuration parameter on file /etc/vsftpd/vsftpd.conf must be configured in order to allow anonymous login on the FTP server?.
5.- Which configuration parameter on file /etc/vsftpd/vsftpd.conf must be configured in order disable local logins on the FTP server?.
6.- Which configuration parameter on file /etc/vsftpd/vsftpd.conf must be configured in order disable logins from users listed in /etc/vsftpd/vsftpd.conf on the FTP server?.
7.- Which command can be used in order to disable SElinux protection to the ftpd service?.
8.- Which command can be used in order to give SElinux access to users logged through FTP client on their home directories on the FTP server?.
9.- Which command can be used in order to connect to the FTP server using anonymous account?.

A - lftp 192.168.1.10
B - lftp -u anonymous 192.168.1.10
C - Both of them
D - None of them
10.- Which configuration parameter on /etc/vsftpd/vsftpd.conf makes that the only users listed on /etc/vsftpd/user_list are allowed to connect to the FTP server ?.

A - userlist_deny=YES
B - userlist_deny=NO

C - /etc/export
D - /etc/fstab

Labs

1.- Configure a FTP server on rhel6 server and only allow anonynous login. Downloading and uploading must be allowed.
2.- Configure an FTP server on rhel6 and only allow logins to all system users less the listed on /etc/vsftpd/user_list . Download/upload must be allowed for these users and the users must be chrooted on their home directory.

1.- False.
2.- True.
3.- False, open 21/TCP port on firewall and load the iptables module 'IPTABLES_MODULES="nf_conntrack_ftp"'. .
4.- 'anonymous_enable=YES'
5.- 'local_enable=NO'
6.- 'userlist_enable=YES'
7.- 'setsebool -P allow_ftpd_full_access 1'
8.- 'setsebool -P ftp_home_directory 1'
9.- C
10.- B

Lab 1

* Login as root on rhel6 (192.168.1.10) and configure a FTP server as on section 'FTP anonymous server only' with the following vsftd.conf :

```
# cat /etc/vsftpd/vsftpd.conf | grep -v ^#

anonymous_enable=YES
local_enable=NO
write_enable=YES
local_umask=022
anon_upload_enable=YES
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

Have a look on the directive '**anon_upload_enable=YES**', it allows uploading for the anonymous user. But just before connect as anonymous user the permissions on the ftp home directory /var/ftp/pub must be fixed in order to allow to the anonymous user write into it (ftp user on the FTP server) :

```
# cd /var/ftp
# chmod -R root.ftp pub
# chmod -R 775 pub
```

Apply security settings as described on 'FTP anonymous server only' section and restart the vsftpd service.

```
# /etc/init.d/vsftpd restart
```

Connect to rhel6 FTP server from node01 as anonymous and download/upload a file. Verify that only anonymous user can login on the FTP server.

```
node01> lftp 192.168.1.10
lftp 192.168.1.10:~> cd pub
cd ok, cwd=/pub
```

```
lftp 192.168.1.10:/pub> get file
10240000 bytes transferred
lftp 192.168.1.10:/pub> put install.log
21820 bytes transferred

node01> lftp -u john 192.168.1.10
Password:
lftp john@192.168.1.10:~> cd pub
cd `pub' [Delaying before reconnect: 3]
```

Lab 2

* Login as root on rhel6 (192.168.1.10) and configure a FTP server as on section 'FTP non-anonymous server' with the following vsftd.conf :

```
# cat /etc/vsftpd/vsftpd.conf | grep -v ^#

anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
chroot_local_user=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

Have a look on the directive '**chroot_local_user=YES**', it will chroot the FTP users on their home directory.

Apply security settings as described on 'FTP anonymous server only' section and restart the vsftpd service.

```
# /etc/init.d/vsftpd restart
```

From node01 connect to the FTP server on rhel6 using john account, verify that john can upload/download files on his home directory and that can not navigate outside /home/john.

```
node01> ftp -u john 192.168.1.10
Password:
lftp john@192.168.1.10:~> put install.log
21820 bytes transferred
lftp john@192.168.1.10:/> get file
10240000 bytes transferred
lftp john@192.168.1.10:/> cd /var
cd: Access failed: 550 Failed to change directory. (/var)
```

Verify that anonymous user can not login to FTP server.

```
node01> lftp 192.168.1.10
lftp 192.168.1.10:~> cd pub
cd `pub' [Delaying before reconnect: 13]
```

Linux Services Organization : Linux SAMBA Linux Server

Samba provides a stable and highly compatible file and print sharing service that allows a Linux node to act as a client, a member server, or even a Primary Domain Controller (PDC) or a member of an Active Directory (AD) service on Microsoft-based networks. Samba interacts with Microsoft's CIFS built on the Server Message Block (SMB) protocol.

Samba is installed through the samba rpms :

```
# yum install samba*
```

Samba Server

Samba is build on two daemons (smbd, nmbd) and one service (smb) which control the daemons.

smbd
The smbd server daemon provides file sharing and printing services to Windows/Linux clients. It is also responsible for user authentication, resource locking, and data sharing through the SMB protocol. The ports on which the server listens for SMB traffic are TCP ports 139 and 445. It is controlled by the smb service.

nmbd
The nmbd server daemon understands and replies to NetBIOS name service requests such as those produced by SMB/CIFS in Windows systems. It also participates in the browsing protocols that make up the Windows Network Neighbourhood view. The port that the server listens to for NMB traffic is UDP port 137. The nmbd daemon is controlled by the smb service.

/etc/samba/smb.conf

This is the main configuration file and is plenty of comments that explain every option. The following is a basic samba server configuration that just exports the printers and /home dir to all Windows/Linux neighbours.

```
# cat /etc/samba/smb.conf

[global]
# Set the workgroup name (samba domain) to RHEL6-WG.
workgroup = RHEL6-WG
server string = Samba Server Version %v

# Samba name for this server, is the name controlled by nmbd daemon
netbios name = rhel6

; interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
; hosts allow = 127. 192.168.12. 192.168.13.

# --------------------------- Logging Options -----------------------------
# logs split per machine
log file = /var/log/samba/log.%m
# max 50KB per log file, then rotate
max log size = 50

# ----------------------- Standalone Server Options ------------------------
# Use local system accounts for authentication. To create the samba user 'john'
# use the command 'smbpasswd -a john' an set the same password as on the system.
# To remove john account on samba server 'smbpasswd -x john'

security = user
passdb backend = tdbsam

# --------------------------- Printing Options -----------------------------
# Use CUPs for printing

load printers = yes
cups options = raw
```

```
; printcap name = /etc/printcap
#obtain list of printers automatically on SystemV
; printcap name = lpstat
; printing = cups

#============================ Share Definitions ==============================
# Export /home and printers

[homes]
comment = Home Directories
browseable = no
writable = yes
; valid users = %S
; valid users = MYDOMAIN\%S

[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
guest ok = no
writable = no
printable = yes
```

There is a tool that can be used to verify the smb.conf configuration : '**testparam**'.

```
# testparm /etc/samba/smb.conf

Load smb config files from /etc/samba/smb.conf
rlimit_max: rlimit_max (1024) below minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
workgroup = RHEL6-WG
server string = Samba Server Version %v
log file = /var/log/samba/log.%m
max log size = 50
cups options = raw

[homes]
comment = Home Directories
read only = No
browseable = No

[printers]
comment = All Printers
path = /var/spool/samba
printable = Yes
browseable = No
```

Now samba is ready to be started.

```
# /etc/init.d/smb restart
# chkconfig smb on
```

Server Security

Firewall

In order to allow samba server to work through a firewall the following ports must be open .

```
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 137 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 138 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 139 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 445 -j ACCEPT
```

SElinux

In case that SElinux has been configured as 'enforcing' in targered mode, the following SElinux parameters must be configured to allow samba server to be executed on SElinux environment.

```
# setsebool -P samba_run_unconfined 1
```

Disables SElinux restrictions to samba.

```
# setsebool -P samba_enable_home_dirs 1
```

Allows samba to share users' home directories.

```
# setsebool -P samba_share_nfs 1
```

Allows Samba to share directories already shared via NFS.

```
# setsebool -P use_samba_home_dirs 1
```

Supports remote access to local home directories using Samba.

```
# chcon -R -t samba_share_t /home/share
```

It labels /home/share to be exported rw mode through samba on a SElinux environment. The label public_content_rw_t is also valid.

Samba Client

The following is a list of the samba client utility than can be used. For this section consider the node rhel6 (192.168.1.10) configured as the samba server defined on 'Samba Server' section and the samba client utilities are launched from node01 (192.168.1.101) against samba server on rhel6.

smbclient

It displays the samba shares exported from a Samba server.

```
node01> smbclient -L 192.168.1.10 -U john

Enter john's password:
Domain=[RHEL6-WG] OS=[Unix] Server=[Samba 3.5.4-68.el6]

Sharename Type Comment
--------- ---- -------
IPC$ IPC IPC Service (Samba Server Version 3.5.4-68.el6)
john Disk Home Directories

Domain=[RHEL6-WG] OS=[Unix] Server=[Samba 3.5.4-68.el6]

Server Comment
```

```
--------- -------

Workgroup Master
--------- -------
```

The samba server account 'john' generated before with the command 'smbpasswd -a john' has been used to list the samba shares. For 'john' user the share 'john' that corresponds to /home/john on rhel6 server is available.

mount

Standard mount command can be used in order to mount remote samba shares on a client using the option '-t cifs'.

```
node01> mount -t cifs //192.168.1.10/john /mnt -o username=john

Password:
node01> ls -lrt /mnt
total 10024
-rw-r--r--. 1 john john 10240000 Feb 22 23:08 file
-rw-r--r--. 1 john john 21820 Feb 26 13:47 install.log
```

Windows client

Of course a Windows node connected to the same LAN as the samba server can access to the samba server as it was a Windows node ...

Questions

1.- Through samba Windows shares can be mounted on a Linux node (true/false).
2.- Nautilius file browser can be used to access to remote samba shares (true/false).
3.- By default samba is not protected by SElinux in 'enforcing' mode with 'targered' policy (true/false).
4.- Which 'smbclient' command option can be used in order to access as user 'john' to 'public' samba share on 192.168.1.10 as an FTP environment?.
5.- Which command can be used in order to mount as user 'kate' a remote samba share //192.168.1.10/share on /mnt directory?.
6.- Which command can be used in order to list the samba shares exported by the samba server 192.168.1.10?.
7.- Which command can be used in order test the samba server configuration ?.
8.- Which SElinux boolean can be configured in order to deactivate SElinux protection to samba server?.
9.- Which samba configuration parameter must be applied on a share in order to export that share in read-write mode only to users on group 'group'?.

A - write list = .group
B - write list = @group
C - Both of them
D - None of them
10.- Which samba configuration parameter must be applied on the samba server configuration file in order to allow only access to any share only from 192.168.1.10/24 LAN ?.

A - hosts allow=192.168.1.0/255.255.255.0
B - hosts allow=192.168.1.
C - Both of them
D - None of them

Labs

1.- Configure samba server rhel6 (192.168.1.10) in domain REDHAT to share homes directories in rw . Create user 'smith' and mount the /home/smith from rhel6 on node01 (192.168.1.10) on /mnt directory.
2.- Share with samba on server rhel6 (192.168.1.10) the share /storage rw to user 'cash' and ro to other valid users.

The share will be available only to node01, 192.168.1.101.

1.- True.
2.- True, smb://
3.- False.
4.- 'smbclient //192.168.1.10/public -U john'
5.- 'mount -t cifs //192.168.1.10/share /mnt -o username=kate'
6.- 'smbclient -L 192.168.1.10 -U john'
7.- 'testparam /etc/samba/smb.conf'
8.- 'setsebool -P samba_run_unconfined 1'
9.- B
10.- C

Lab 1

* Login as root on rhel6 (192.168.1.10) and configure samba server on domain REDHAT to export all /home directories in read-writer mode.

```
cat /etc/samba/smb.conf

[global]
workgroup = REDHAT
server string = Samba Server Version %v
netbios name = rhel6

log file = /var/log/samba/log.%m
max log size = 50

security = user
passdb backend = tdbsam

load printers = yes
cups options = raw

[homes]
comment = Home Directories
browseable = no
writable = yes
```

* Configure the firewall to allow samba connections.

```
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 137 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 138 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 139 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 445 -j ACCEPT
```

* Configure SElinux to allow samba server to export home directories in read-write mode.

```
# setsebool -P samba_enable_home_dirs 1
```

* Start samba server.

```
# /etc/init.d/smb restart
# chkconfig smb on
```

* Create local user account 'smith' password 'smith'. Create also the same account on the samba server.

```
# useradd smith
# passwd smith
# smbpasswd -a smith
```

* From node01 (192.168.1.101) using 'smith' samba account list the shares exported by the samba server rhel6 (192.168.1.10).

```
node01> smbclient -L 192.168.1.10 -U smith

Enter smith's password:
Domain=[REDHAT] OS=[Unix] Server=[Samba 3.5.4-68.el6]

Sharename Type Comment
--------- ---- -------
IPC$ IPC IPC Service (Samba Server Version 3.5.4-68.el6)
smith Disk Home Directories
```

* As can be seen samba server on 192.168.1.10 domain REDHAT exports the share 'smith' that corresponds to /home/smith when authenticated as user smith. Lets mount it on /mnt and verify that we have read-write access.

```
node01> mount -t cifs //192.168.1.10/smith /mnt -o username=smith

Password:
node01> cd /mnt
node01> ls -lrta
-rw-r--r--. 1 1003 1003 124 Jun 22 2010 .bashrc
-rw-r--r--. 1 1003 1003 176 Jun 22 2010 .bash_profile
node01> echo 123 > file
node01> cat file
123
```

Note uid=1003 corresponds to local user account 'smith' on rhel6 samba server. Files are created as user 'smith' on /home/smith directory on rhel6 node.

Lab 2

* Using the configuration done on lab1, add the share storage on rhel6 samba server.

```
[storage]
comment = Storage dir
path = /storage
writable = no
write list = cash
hosts allow = 192.168.1.101
```

* Create the directory /storage and set up the corresponding permissions.

```
# mkdir -p /storage
# chmod 1777 /storage
# chcon -R -t samba_share_t /storage
```

* Restart samba service.

```
#/etc/init.d/smb restart
```

* Add the user 'cash' passwd 'cash' to rhel6 server and samba server accounts.

```
# useradd cash
# passwd cash
# smbpasswd -a cash
```

* Connect from node01 (192.168.1.101) to share storage as user cash and verify that can write to it.

```
node01> mount -t cifs //192.168.1.10/storage /mnt -o username=cash
```

```
Password:
node01> cd /mnt
node01> echo 123 > filecash
```

* Connect from node01 (192.168.1.101) to share storage as user smith and verify that has read-only access.

```
node01> cd /tmp; umount /mnt
node01> mount -t cifs //192.168.1.10/storage /mnt -o username=smith

Password:
node01> cd /mnt
node01> echo 456 > filesmith
-bash: filesmith: Permission denied
```

* Connect from another node that node01 and verify that the access is denied.

```
node02> mount -t cifs //192.168.1.10/storage /mnt -o username=cash

Password:
mount error(13): Permission denied
```

Linux Services Organization : Linux HTTPD Linux Server

Apache is the most popular Web server used in the Internet. Based on the HTTP daemon (httpd) provides a secure access to all types of content using the regular HTTP protocol as well as its secure version HTTPS which encrypts the HTTP traffic. Apache is a robust open source Web server developed by the Apache Software Foundation http://www.apache.org.

Apache Web Server

In order to install Apache httpd server, the httpd rpm must be installed.

```
# yum install httpd
```

It is also possible install the secure version of Apache (https) with the mod_ssl rpm.

```
# yum install mod_ssl
```

Apache Basic Configuration

The two configuration key files are **/etc/httpd/conf/httpd.conf** for http web server and **/etc/httpd/conf.d/ssl.conf** for https web server. The default version of these file creates a generic and functional web server.

```
# cat /etc/httpd/conf/httpd.conf

### Section 1: Global Environment

#It limits what readers see about your Web server when you browse to a nonexistent page. With this option you are not showing which subcomponents are running the httpd server.
ServerTokens OS

#Root location of configuration and log files is determined by the ServerRoot directive.
ServerRoot "/etc/httpd"

#The number of seconds before receives and sends time out when no http activity is generated.
Timeout 120

#TCP/IP port where the httpd server listen, by default 80.
```

```
Listen 80

#Load config files from the config directory "/etc/httpd/conf.d".
Include conf.d/*.conf

#The name (or #number) of the user/group to run httpd as.
User apache
Group apache

### Section 2: 'Main' server configuration

#Admin email address, server problems will be emailed automatically to this address.
ServerAdmin root@info.net

#The directory out of which you will serve your documents. By default, all requests are taken from this directory, but symbolic links and aliases may be used to point to other locations.
DocumentRoot "/var/www/html"

#First configure the "default" web directories to be a very restrictive set of features. FollowSymLinks line supports the use of symbolic links for Web pages. AllowOverride None line disables any .htaccess files can allow others users to administer your server.
‹ Directory / ›
Options FollowSymLinks
AllowOverride None
‹ /Directory ›

#The next limits access to /var/www/html the default DocumentRoot directive. Indexes setting allows readers to see a list of files if no index.html file is present on DocumentRoot. The Order and Allow lines allow all users to access the Web pages on DocumentRoot.
‹ Directory /var/www/html ›
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
‹ /Directory ›

# LogLevel: Control the number of messages logged to the error_log. Possible values include: debug, info, notice, warn, error, crit, alert, emerg. By default log files are located on /var/log/httpd directory.
LogLevel warn

# Explained in more detail in next sections.
### Section 3: Virtual Hosts
...
```

Note: if 'mod_ssl' rpm has been installed a secure Apache web server (https) will be running by default on port 443 TCP/IP. The configuration file for this sercure web server is on **/etc/httpd/conf.d/ssl.conf**.

Note: Apache logs are located on **/var/log/httpd** directory.

Apache Security

Firewall

In order to run an Apache web server through a firewall, the ports 80 (http) and 443 (https) TCP/IP must be open.

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```

SElinux

By default Apache web server is protected by SElinux in targered mode. In order to allow Apache to be executed through SElinux the following parameters can be configured.

```
# setsebool -P httpd_enable_cgi 1
```

Allow cgi scripts to be executed through the web server.

```
# setsebool -P httpd_enable_home_dirs 1
```

By default SElinux does not allow to access to users home directories through the web server. This directive makes it possible, but first the home directories must be labelled as web server files :'**chcon -R -t httpd_sys_content_t ~user/public_html**'.

In addition if some system directory is going to be accessed through the web server, first it must be labelled as SElinux http file. For example if the directory /secwww is going to be used as a web server DocumentRoot.

```
# chcon -R -u system_u /secwww
# chcon -R -t httpd_sys_content_t /secwww
```

Host access security

In httpd.conf file the directives **allow** and **deny** can regulate access to different areas of the web server based on host names or IP addresses.

```
‹ Directory /var/www/html/web ›
...
Order deny,allow
Deny from all
Allow from .info.net
Allow from 192.168.30.0/24
...
‹ /Directory ›
```

The line '**Order deny,allow**' means that first deny directives are applied and then allow directives. In this case access is denied to all less hosts on .info.net domain or in 192.168.30.0/24 LAN.

User access security

Access to different areas of the web server can be regulated through username and password.

```
‹ Directory "/var/www/html/marketing" ›
...
AuthType Basic
AuthName "Password Protected Marketing"
AuthUserFile /etc/httpd/markpass
Require user john kate
...
‹ /Directory ›
```

This configuration will allow access only to users john and kate to marketing web area. When a connection is made against marketing web area the web server asks for a username/password that will be authenticated against the password file on /etc/httpd/markpass. In order to create john and kate accounts the command htpasswd can be used.

```
# htpasswd -c /etc/httpd/markpass john
Password:

# htpasswd /etc/httpd/markpass kate
Password:
```

Note that '-c' option on htpasswd must be used if the authentication file does not exists (if is the first user that we are creating).

Executable files in Apache

The ScriptAlias directive can be used to enable web directories with executable CGI files. The following ScriptAlias directive links the default cgi-bin directory to /var/www/cgi-bin.

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin"

‹ Directory /var/www/cgi-bin ›
AllowOverride None
Options None
Order allow,deny
Allow from all
‹ /Directory ›
```

Remember to change the SElinux context for this directory to allow SElinux to execute the scripts through Apache.

```
# chcon -t httpd_sys_script_exec_t /var/www/cgi-bin
```

Makes sure the Apache can execute cgi scripts through SElinux.

```
# setsebool -P httpd_enable_cgi 1
```

Limiting resources and rejecting DoS attacks

There are some configuration parameters that can be used to limit the system resources that Apache can take from the system in order to minimize the impact of a DoS Attack .

```
# StartServers: number of server processes to start when httpd is started. More httpd process are started if required until reach 'MaxClients' limit.
StartServers 8

# ServerLimit: maximum value for MaxClients for the lifetime of the server. Is limits the maximum number of simultaneous clients that can connect to the web server.
ServerLimit 256

# MaxClients: maximum number of server processes allowed to start. It limits the maximum number of simultaneous httpd process on the web server. The MaxClients directive sets the limit of simultaneous requests that can be served, if there are requests past the Maxclients they will be queued.
MaxClients 256
```

Apache Virtual Hosts

An useful feature of Apache is its ability to manage different web sites using a single IP address creating multiple virtual hosts on the same web server on the file /etc/httpd/conf/httpd.conf. The final result is that multiple domain names such as www.info.net and www.example.net can be served on the same web server using the same IP address. It is also possible configure virtual hosts on the secure web server configuring it on file /etc/httpd/conf.d/ssl.conf .

```
# cat /etc/http/conf/httpd.conf
...
### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
```

```
NameVirtualHost *:80
...
‹ VirtualHost *:80 ›
ServerAdmin webmaster@info.net
DocumentRoot /var/www/info.net
ServerName www.info.net
ErrorLog logs/www.info.net-error_log
CustomLog logs/www.info.net-access_log common
‹ /VirtualHost ›

‹ VirtualHost *:80 ›
ServerAdmin webmaster@example.net
DocumentRoot /var/www/example.net
ServerName www.example.net
ErrorLog logs/www.example.net-error_log
CustomLog logs/www.example.net-access_log common
‹ /VirtualHost ›
```

In this case two web sites has been configured on the same web server : www.info.net (DocumentRoot on /var/www/info.net and logs on www.info.net*log) and www.example.net (DocumentRoot on /var/www/example.net and logs on www.example.net*log) . Lets create a index.html for each web server and label it as web server files for SElinux.

```
# mkdir -p /var/www/info.net
# echo "info net web page" > /var/www/info.net/index.html
# mkdir -p /var/www/example.net
# echo "example net web page" > /var/www/example.net/index.html
```

```
# chcon -R -u system_u /var/www/info.net
# chcon -R -t httpd_sys_content_t /var/www/info.net
# chcon -R -u system_u /var/www/example.net
# chcon -R -t httpd_sys_content_t /var/www/example.net
```

And finally restart the Apache web server with the new configuration.

```
# /etc/init.d/httpd restart
```

In order to test the result add the following lines on /etc/hosts files just to resolve the hostnames www.info.net and www.example.com to the Apache web server IP 192.168.1.10.

```
# echo "192.168.1.10 www.info.net" >> /etc/hosts
# echo "192.168.1.10 www.example.net" >> /etc/hosts
```

Lets check the httpd.conf content with the command 'httpd'

```
# httpd -t
Syntax OK
```

```
# httpd -D DUMP_VHOSTS

VirtualHost configuration:
wildcard NameVirtualHosts and _default_ servers:
*:80 is a NameVirtualHost
default server www.info.net (/etc/httpd/conf/httpd.conf:1013)
port 80 namevhost www.info.net (/etc/httpd/conf/httpd.conf:1013)
port 80 namevhost www.example.net (/etc/httpd/conf/httpd.conf:1021)
Syntax OK
```

And finally access to www.info.net and www.example.net and verify that the corresponding index.html page is displayed.

```
# elinks http://www.info.net
--> info net web page

# elinks http://www.example.net
--> example net web page
```

Questions

1.- The Apache web server daemon httpd is executed by default as root user (true/false).
2.- In order to install a secure Apache web server the rpm 'https' must be installed (true/false).
3.- With Apache is possible customize the error code messages (true/false).
4.- Which httpd.conf configuration parameter is used to configure the system directory that contains the file/directories that form the web server?.
5.- Which httpd.conf configuration parameter is used to allow to follow symbolic links on a web sever directory?.
6.- Which command can be used in order to test the httpd.conf content?.
7.- Which command can be used in order test the the httpd.conf virtual host configuration?.
8.- Which SElinux command must be used in order to label a system directory to be used as web server directory?.
9.- Which httpd.conf configuration parameter must be used in order to configure where the Apache web server configuration is stored?.

A - ServerTokens
B - ServerAdmin
C - DocumentRoot
D - ServerRoot
10.- Which httpd.conf configuration parameter must used to set the file that is displayed when the directory that contains that file is requested on a web server?.

A - DirectoryIndex
B - DirectoryRoot
C - Both of them
D - None of them

Labs

1.- Create on rhel6 (192.168.1.10) a security dir /var/security accessible on https://server.example.com/secure and allow only access from 192.168.1.0/24 to authenticated users listed in /etc/httpd/secure. User donna/donna and mike/mike must have access. (http://sever.example.com must be accessible from everywhere):.
2.- On web server configured on the previous example(http://sever.example.com) allow all users share their /home/user/public_html ro via Apache. Make sure that works with SElinux.:
3.- Configure vhosts on rhel6 (192.168.1.10) in https web server for host1.example.com (/host1) and host2.example.com (/host2).

1.- False, httpd is executed as apache user.
2.- False, mod_ssl rpm.
3.- True.
4.- 'DocumentRoot'
5.- 'FollowSymLinks'
6.- 'httpd -t'
7.- 'httpd -D DUMP_VHOSTS'
8.- 'chcon -R -t httpd_sys_content_t directory_path' '
9.- D
10.- A

Lab 1

* Login as root on rhel6 (192.168.1.10) and install 'httpd' and 'mod_ssl rpms'.

```
# yum install httpd
# yum install mod_ssl
```

* Create the index.html file for the default web server on the default DocumentRoot, /var/www/html.

```
# echo "Default http web page" > /var/www/html/index.html
```

* Start http server and make sure it will start on boot.

```
# /etc/init.d/httpd start
# chkconfig httpd on
```

* Create a line on /etc/hosts in order to resolve server.example.com to the IP 192.168.1.10.

```
# echo "192.168.1.10 server.example.com" >> /etc/hosts
```

* Open the http and https ports on the system firewall.

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```

```
# /etc/init.d/iptables restart
```

* Verify the access to the default web server configured on rhel6.

```
# elinks http://server.example.com
-->Default http web page
```

* Create the secure web directory on '/var/security', label it as SElinux http directory and create a default web page.

```
# mkdir -p /var/security
# echo "Secure https web page" > /var/security/index.html
# chcon -R -u system_u /var/security
# chcon -R -t httpd_sys_content_t /var/security
```

* On /etc/httpd/conf.d/ssl.conf create an Alias directive that points from 'secure' to '/var/security' directory. Create de directory tag for '/var/security' and add the corresponding ACLs to that directory.

```
Alias /secure "/var/security/"
‹ Directory "/var/security/" ›
Options Indexes MultiViews FollowSymLinks
AllowOverride None
Order deny,allow
Deny from all
Allow from 192.168.1.0/24
AuthType Basic
AuthName "Password Protected Secure Area"
AuthUserFile /etc/httpd/secure
Require user donna mike
‹ /Directory ›
```

* Create donna and mike users accounts on /etc/httpd/secure.

```
# htpasswd -c /etc/httpd/secure donna
# htpasswd /etc/httpd/secure mike
```

* Restart httpd service.

```
# /etc/init.d/httpd restart
```

* From node01 (192.168.1.101) run 'firefox' browser and try to access to https://server.example.com/secure. Makes sure through /etc/hosts file that server.example.com resolves to IP 192.168.1.10.

```
node01> firefox
```

* Accept the ssl certificate and use user donna/donna web account to access to the secure web. Also verify that the default web page on http://server.example.com is still available.

```
node01> firefox http://server.example.com
```
-->Default http web page

```
node01> firefox https://server.example.com/secure
```
--> Authenticating as donna --> Secure https web page

Lab 2

* Using the web server configured on lab1 accessible on http://server.example.com, enable SElinux boolean to allow users to access their home directories in read-only mode through Apache web server.

```
# setsebool -P httpd_enable_homedirs 1
```

* Configure /etc/httpd/conf/httpd.conf to enable this feature.

```
# cat /etc/httpd/conf/httpd.conf
...
UserDir public_html
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
‹ Directory /home/*/public_html ›
AllowOverride FileInfo AuthConfig Limit
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
‹ Limit GET POST OPTIONS ›
Order allow,deny
Allow from all
‹ /Limit ›
‹ LimitExcept GET POST OPTIONS ›
Order deny,allow
Deny from all
‹ /LimitExcept ›
‹ /Directory ›
...
```

* Restart Apache web service.

```
# /etc/init.d/httpd restart
```

* Create 'john' system account, if it does not already exists.

```
# useradd john
```

* Create john default web page on /home/john/public_html/index.html and set the appropriate permissions.

```
# su - john
john> mkdir -p /home/john/public_html
john> echo "John web page" > /home/john/public_html/index.html
john> chmod 701 /home/john
john> chmod 705 /home/john/public_html
john> setsebool -P httpd.enable.homedirs 1
john> chcon -R -t httpd_sys_content_t /home/john/public_html
```

* From node01 (192.168.1.101) verify that files on /home/john/public_html can be accessed through the web server.

```
node01> elinks http://server.example.com/~john/
--> John web page
```

Note that the access in this way is public, everybody can access to public_html files on /home/john through the web server. If you want more security configure more restrictive ACLs on it.

Lab 3

* Using the same web server than in previous labs, create host1.example.com (/host1) and host2.example.com (/host2) virtual hosts on ssl.conf

```
# cat /etc/httpd/conf.d/ssl.conf
...

NameVirtualHost *:443

‹ VirtualHost *:443 ›

DocumentRoot "/host1"
ServerName host1.example.com:443
ErrorLog logs/host1_ssl_error_log
TransferLog logs/host1_ssl_access_log
LogLevel warn

SSLEngine on
SSLProtocol all -SSLv2
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

‹ Files ~ "\.(cgi|shtml|phtml|php3?)$" ›
SSLOptions +StdEnvVars
‹ /Files ›

SetEnvIf User-Agent ".*MSIE.*" \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0

CustomLog logs/host1_ssl_request_log \
"%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

‹ /VirtualHost ›

‹ VirtualHost *:443 ›

DocumentRoot "/host2"
ServerName host2.example.com:443
ErrorLog logs/host2_ssl_error_log
TransferLog logs/host2_ssl_access_log
LogLevel warn

SSLEngine on
SSLProtocol all -SSLv2
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

‹ Files ~ "\.(cgi|shtml|phtml|php3?)$" ›
SSLOptions +StdEnvVars
‹ /Files ›
```

```
SetEnvIf User-Agent ".*MSIE.*" \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0

CustomLog logs/host2_ssl_request_log \
"%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

‹ /VirtualHost ›
```

* Create the DocumentRoot for each virtual host and configure the appropriate permissions.

```
# mkdir -p /host1
# mkdir -p /host2
# echo "host1 https web page" > /host1/index.html
# echo "host2 https web page" > /host2/index.html
# chcon -R -t httpd_sys_content_t /host1
# chcon -R -t httpd_sys_content_t /host2
```

* Restart httpd service and verify the vhost configuration.

```
# /etc/init.d/httpd restart

# httpd -t
Syntax OK

# httpd -D DUMP_VHOSTS
VirtualHost configuration:
wildcard NameVirtualHosts and _default_ servers:
*:443 is a NameVirtualHost
default server 10.0.0.1 (/etc/httpd/conf.d/ssl.conf:74)
port 443 namevhost 10.0.0.1 (/etc/httpd/conf.d/ssl.conf:74)
port 443 namevhost host1.example.com (/etc/httpd/conf.d/ssl.conf:239)
port 443 namevhost host2.example.com (/etc/httpd/conf.d/ssl.conf:267)
Syntax OK
```

* On node01 (192.168.1.101) configure /etc/hosts file to resolve host1.example.com,host2.example.com to rhel6 web server ip 192.168.1.10.

```
node01> echo "192.168.1.10 host1.example.com"
node01> echo "192.168.1.10 host1.example.com"
```

* Using firefox access to the secure web pages. Remember to accepts the SSL certificate.

```
node01> firefox https://host1.example.com
```
--> host1 https web page

```
node01> firefox https://host2.example.com
```
--> host2 https web page

Linux Services Organization : Linux Proxy Linux Server

Squid Web proxy cache is a HTTP and FTP caching proxy server. It stores data from Web pages and files accessed through it in order to offer to his clients the data they need without having to look to the Internet, if that data is cached on the proxy server. By default Squid runs as a caching proxy server on TCP port 3128 and can control who can use the proxy server based on host or user authentication. It also allows to filter the access to Internet based on destinations ports, destination URLs, etc ...

With Squid web proxy cache you can control and monitor who is accessing where on the Internet.

Squid proxy Server

In order to configure a server as Squid proxy server the 'squid' rpm must be installed.

```
# yum install squid
# chkconfig squid on
```

Squid runs as a 'squid' system daemon storing cached data on **/var/spool/squid** directory. It is configured through the files on **/etc/squid** directory and stores the logs on **/var/log/squid**.

/etc/squid/squid.conf

This is the main configuration file that sets the way the proxy cache server is executed. In this case nodes on 192.168.1.0/24 LAN and localhost are allowed to connect to Internet to the safe ports listed on 'Safe_ports' directive through the proxy.

```
# cat /etc/squid/squid.conf

acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl localhost src ::1/128
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl to_localhost dst ::1/128

# Rule allowing access from your local networks.
acl localnet src 192.168.1.0/24

# This acl directives specify ports through which traffic is cached and are considered as safe ports.
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

# Recommended minimum Access Permission configuration:
# Only allow cachemgr access from localhost.
http_access allow manager localhost
http_access deny manager

# Deny requests to certain unsafe ports.
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports.
http_access deny CONNECT !SSL_ports

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user.
http_access deny to_localhost

# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed.
http_access allow localnet
http_access allow localhost

# And finally deny all other access to this proxy.
http_access deny all

# Squid normally listens to port 3128.
http_port 3128

# We recommend you to use at least the following line.
hierarchy_stoplist cgi-bin ?

# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 100 16 256

# Leave coredumps in the first cache dir.
coredump_dir /var/spool/squid

# Add any of your own refresh_pattern entries above these. The refresh_pattern directive specifies when data from a specified server is considered "fresh" and there is not need to refresh it into proxy cache.
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

Note: there is a full documented squid.conf sample file on **/usr/share/doc/squid-3.1.4/squid.conf.documented**.

Once the configuration file has been configured the squid daemon must be started and configured to start at boot.

```
# /etc/init.d/squid start
# chkconfig squid on
```

Squid Security

Firewall

Because of squid listen on 3128 TCP/IP by default, that port must be open on the firewall.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3128 -j ACCEPT
```

Squid can be configured in transparent mode where the client does not known that is connecting to Internet through a proxy server. In this case Squid server must be running on the LAN default gateway and all traffic that pass through it with destination port 80,446,... must be redirected to the Squid port. With this configuration the client is accessing to the Internet using the Squid proxy cache without having to connect directly to it.

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-ports 3128
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT --to-ports 3128
```

SElinux

In order to allow Squid service to run without any SElinux interference the following parameter must be activated.

```
# setsebool -P squid_connect_any 1
```

Host Based Security

There are two methods that can be used on **/etc/squid/squid.conf** in order to allow/deny client access to the Squid proxy web cache server : using the IP or the MAC client address. For example if we want allow access to the 192.168.1.101 address :

```
# cat /etc/squid/squid.conf
...
acl node01 src 192.168.1.101/32
...
http_access allow node01
...
http_access deny all
```

Instead of IP address we can use the MAC address.

```
# cat /etc/squid/squid.conf
...
acl node01mac arp 00:0C:29:78:97:8C
...
http_access allow node01mac
...
http_access deny all
```

User Based Security

If we want to control the access to Squid web proxy cache to certain users in order to allow to access to Internet to that users, the module '**ncsa**' can be used.

```
# cat /etc/squid/squid.conf
...
acl localnet src 192.168.1.0/24
...
# NCSA proxy authentication configuration.
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd
# Users ACL definition.
acl ncsa_users proxy_auth REQUIRED
# Allow access users only from localnet.
http_access allow ncsa_users localnet
# Deny the rest.
http_access deny all
```

Finally the password file used to authenticate users through ncsa must be created using the '**htpasswd**' command as in the case of http user authentication.

```
# htpasswd -c /etc/squid/passwd john
# chown root:squid /etc/squid/passwd
# chmod 640 /etc/squid/passwd

# /etc/init.d/squid restart
```

URL Filter

Squid can also filter the sites that his client is trying to access. For example if you want to deny the access to all webs that has in their URL the word 'adult' you can use the '**url_regex**' directive on **/etc/squid/squid.conf** file.

```
# cat /etc/squid/squid.conf
...
```

```
acl filterurl url_regex adult
...
http_access deny filterurl
...
```

Squid Client

One way to configure a client to use a Squid proxy to connect to Internet is configuring the web browser (Firefox) to use the Proxy server in order to connect to internet. For example is you are using Firefox and you want to use the Squid Proxy running on 192.168.1.10 port 3128 :

Edit --> Preferences --> Network --> Connection --> Settings
Manual Proxy Configuration
Proxy HTTP 192.168.1.10 Port 3128

In case of using test web browser as 'elinks' the way to configure the browser to use a Proxy cache is through '**http_proxy**' environment variable.

```
# export http_proxy=http://192.168.1.10:3128
# elinks http://www.linuxsv.org
```

Questions

1.- By default Squid web proxy cache runs on 8080 TCP/IP port (true/false).
2.- NSCA authentication module is used to control the access to the Squid web proxy cache to authenticated users (true/false).
3.- Squid web proxy cache server can control the access to Internet based on the type of the client web browser used (true/false).
4.- Which command must be executed in order to configure access to a web proxy cache (192.168.1.10:3128) on text web browser?.
5.- Which SElinux boolean must be configured in order to allow Squid to run without any restriction on a SElinux targered environment?.

Labs

1.- Configure the squid proxy server on port 3128 on rhel6 server (192.168.1.10) allowing internet access to 192.168.1.0/24 but blocking msn.com site to access. Test it connecting from node01 (192.168.1.101).
2.- Use elinks text web browser to connect from node01 to http://www.linuxsv.org through the squid proxy configured on 192.168.1.10:3128..
3.- Configure user access to the squid web proxy cache configured on Lab1. Only users 'john' with password 'john' and 'charles' with password 'charles' must be allowed to connect to Internet through squid service.

1.- False, the default port used by squid is 3128 TCP/IP.
2.- False, it is NCSA module not NSCA.
3.- True, 'acl aclname browser ' can be used on squid.conf.
4.- 'export http_proxy=http://192.168.1.10:3128'
5.- 'setsebool -P squid_connect_any 1'

Lab 1

* Login as root on rhel6 (192.168.1.10) and install 'squid' rpm. It is supposed that rhel6 has internet access.

```
# yum install squid
```

* Edit /etc/squid/squid.conf file in order to fit the requirements of the exercise.

```
# cat /etc/squid/squid.conf
```

```
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl localhost src ::1/128
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl to_localhost dst ::1/128

acl localnet src 192.168.1.0/24
acl filterurl url_regex msn.com

acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
```

# In squid.conf file the order is important : first match found is taken and no more process is done. First we deny msn.com and later we allow access to 192.168.1.0/24 LAN. If we put first the allow localnet the mns.com filter will never be applied to 192.168.10/24 !!!.

```
http_access deny filterurl
http_access allow localnet
http_access allow localhost
http_access deny all

http_port 3128

hierarchy_stoplist cgi-bin ?

cache_dir ufs /var/spool/squid 100 16 256

coredump_dir /var/spool/squid

refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

* Start squid service and verify that logs are generated on /var/log/squid directory,

```
# /etc/init.d/squid restart
```

* Open port 3128 TCP/IP on the firewall and make sure that squid service will run without any SElinux interference.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3128 -j ACCEPT --> /etc/sysconfig/iptables
# /etc/init.d/iptables restart
# setsebool -P squid_connect_any 1
```

* Connect on node01 and configure firefox web browser to use squid web proxy cache server configured on rhel6.

```
node01> firefox
```

Edit --> Preferences --> Network --> Connection --> Settings
Manual Proxy Configuration
Proxy HTTP 192.168.1.10 Port 3128

* With firefox browser try to access to www.google.com. Verify that logs on /var/log/squid register the action .

node01> firefox --> www.google.com (as usual)

* With firefox browser try to access to www.msn.com. Verify that logs on /var/log/squid register the action .

node01> firefox --> www.msn.com --> '**Access Denied.**'
/var/log/squid/access.log --> '1300712175.270 3 192.168.1.2 **TCP_DENIED/403** 4227 GET http://www.msn.com/ - NONE/- text/html'

* Also verify that data cached on Squid web proxy cache is stored on /var/spool/squid.

```
# ls -lrt /var/spool/squid
```

New files/directories has been created ...

Lab 2

* On node01 configure the environment variable 'http_proxy' in order to access to the web http://www.linuxsv.org through squid proxy server on 192.168.1.10:3128.

```
node01> export http_proxy=http://192.168.1.10:3128
```

* Using elinks connect to http://www.linuxsv.org. Verify that on rhel6 server on /var/log/squid directory has been created squid logs.

```
node01> elinks http://www.linuxsv.org
--> Connected
```

On rhel6:/var/log/squid/access.log -->
1300731486.845 3206 192.168.1.2 TCP_MISS/200 4689 **GET http://www.linuxsv.org/ - DIRECT/95.60.149.133 text/html**

Lab 3

* On Squid web proxy cache server configured on Lab1 in rhel6 node add the ncsa configuration module just before http allow/deny directives.

```
# cat /etc/squid/squid.conf
...
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd
acl ncsa_users proxy_auth REQUIRED
...
http_access deny filterurl
http_access allow ncsa_users localnet
...
http_access deny all
...
```

* Create the user database and add 'john' and 'charles' accounts.

```
# htpasswd -c /etc/squid/passwd john
New password:
Re-type new password:

# htpasswd /etc/squid/passwd charles
# chown root:squid /etc/squid/passwd
# chmod 640 /etc/squid/passwd
```

* Restart squid service.

```
# /etc/init.d/squid restart
```

* From node01 using the firefox web browser configured to use squid service on rhel6 configured on Lab1 try to access to www.google.com.

```
node01>firefox www.google.com
```

Username : john
Password : john

```
# /var/log/squid/access.log --> 1300726567.591 930 192.168.1.2 TCP_MISS/200 6478 GET http://www.google.es/
john DIRECT/209.85.227.104 text/html.
```

* Close all firefox windows and try to access with an unauthorized user.

```
node01>firefox www.google.com
```

Username : root
Password : root

"Cache Access Denied." Authentication is Required.

```
# /var/log/squid/access.log --> 1300726998.640 1 192.168.1.2 TCP_DENIED/407 4120 GET http://www.google.com/
root NONE/- text/html
```

* As user 'charles' verify that you still can not connect to www.msn.com web.

```
node01>firefox www.msn.com
```

Username : charles
Password : charles

"Access Denied."

Linux Services Organization : Linux SSH Linux Server

OpenSSH is the open source version of SSH secure shell protocol. It provides secure (encrypted) communication between systems using a client/server architecture. It allows users to log into remote systems or execute remote commands in a safe way because of all data transferred between ssh client and server is encrypted. It also allows secure (encrypted) data transfers between systems using **scp** or **sftp** the encrypted secure FTP version.

By default SSH tools are installed on RHEL6 systems, the **openssh*** rpms are included on the default installation. The server daemon **sshd** listens on port 22 TCP/IP and the configuration files are located in the **/etc/ssh** directory.

Introduction to Encryption

Encryption with SSH requires a private key and a public key, generated by '**ssh-keygen**' command. In order to establish an SSH encrypted communication between you and others the first step is send your public key to others keeping always your private key private. When others want to send data to you through SSH, their messages are encrypted with your public key that you have send previously. Your computer can decrypt the message with your private key. As can be seen the public and private keys are related with not easy to guess mathematical algorithms.

Private keys

Private key must be secure and used only by you to decrypt messages encrypted with you public key. Secure SSH encrypted communications are based on keeping the private key secure.

Public keys

Public key is publicly available. The recipient of your messages will encrypt the data with your public key that previously you have send. Only you using your private key will be able to decrypt that message.

SSH Tools

These are the most basic SSH tools than a Linux user must be aware.

sshd

The daemon service that implements the ssh server. By default it must be listening on port 22 TCP/IP.

ssh

The Secure Shell command ssh is a secure way to log and execute commands in to a remote machine using the private/public key encryption method replacing the insecure tools traditionally used for it: telnet, rlogin, rexec, rsh, etc.

scp

The Secure Copy command is a secure way to transfer files between computers using the private/public key encryption method replacing the insecure tool traditionally used for it: ftp.

ssh-keygen

This utility is used to create the public/private keys with the command 'ssh-keygen -t keytype' where keytype can be DSA (Digital Secure Algorithm) or RSA1 (RSA Security). Later in this lesson will be shown how to use it.

ssh-agent

This utility holds private keys used for RSA authentication. The idea is that the ssh-agent command is started in the beginning of an X session or a login session, and all other windows or programs are started as clients to the ssh-agent program. In this way all clients of the ssh-agent can remember through the use of environment variables the public/private keys used when ssh-agent was started, so the user will not be ask for this in all these client sessions.

ssh-add

Adds RSA identities to the authentication agent ssh-agent.

SSH Server

The SSH server configuration file is **/etc/ssh/sshd_config**. This file is well commented so just having a look on it one can understand the meaning of the main directives.

```
# cat /etc/ssh/sshd_config
# This directive configures SSH version 2, which is more secure that version 1
Protocol 2

# The following sends all logging attempts to the appropriate log file /var/log/secure
SyslogFacility AUTHPRIV

# This directive authorizes authentication based on local user passwords
PasswordAuthentication yes

# Set this to 'yes' to enable PAM authentication, account processing, and session processing
UsePAM yes

# The following directive allows to open remote GUI tools executed through SSH using the local X Server
X11Forwarding yes

# This directive supports the use of SSH encryption for secure FTP file transfers
Subsystem sftp /usr/libexec/openssh/sftp-server
```

Once the configuration file has been set lets start the ssh server and make sure it will start at system boot.

```
# /etc/init.d/sshd restart
# chkconfig sshd on
```

SSH client

The SSH client standard configuration file for all system is **/etc/ssh/ssh_config**. Each user can have custom SSH client configurations in their ~/.ssh/config files.

Some examples of SSH client tools can be :

ssh

Allows to login and execute shell commands on remote systems.

```
node01> ssh rhel6 -l john
```

It will login as john on rhel6 system.

```
node01> ssh rhel6 "ls -lrt /home/john".
```

It will execute the command 'ls -lrt /home/john' as user john on rhel6 system. The command output is displayed on node01 the SSH client from where are launched the connection

scp

Used to transfer data between computer systems using SSH.

```
node01> scp /tmp/file.txt john@rhel6:/tmp/file.txt
```

This command will transfer file /tmp/file.txt from SSH client node01 to SSH server rhel6 on /tmp directory using 'john' account.

```
node01> scp -r john@rhel6:/tmp/dir /tmp/
```

This command will transfer from SSH client rhel6 the directory /tmp/dir to the SSH server node01 on /tmp dir using 'john' account. In this case node01 receives the data so node01 is the SSH server, sshd daemon must be running on node01.

SSH Security

Firewall

As has been commented the sshd server listen on port 22 TCP/IP so this port must be open in order to allow ssh server service through a firewall.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
```

User and Host Based Security

Some additional directives can be added to **/etc/sshd/sshd_config** file in order to make the access to ssh server more restrictive.

```
# Do not allow empty passwords
PermitEmptyPasswords no

# The following directive will not allow to root user to log on the system using ssh. (Do not allow remote root logins)
PermitRootLogin no

# Limit the users allowed to access a system via SSH. In this case only users 'john' and 'charles' are allowed to login on the system using SSH
AllowUsers john charles

# Or even more restrictive, only allow login through SSH users 'john' and 'charles' from 192.168.1.101 node.
AllowUsers john@192.168.1.101 charles@192.168.1.101

# In addition you can restrict the access to users. In this case all users less 'john' are allowed to connect to the SSH server.
DenyUsers john
```

SSH using only public/private keys

If the system where SSH server is running is directly connected to the Internet it will be a good idea to disable password authentication on the SSH server and allow only public/private keys authentication. This will reduce dramatically the chance that a cracker has login on your system because the probability that he has to guess the pair user/private_key is much lower that user/password pair. In order to accomplish this the following directives must be changed/added to **/etc/ssh/sshd_config** file.

```
# cat /etc/ssh/sshd_config
...
# Do not allow password authentication
PasswordAuthentication no

# Allow public/private key authentication
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
...
```

Next step is create the public/private key pair on the ssh client node01 from where are going to connect to the SSH server (rhel6).

```
node01> su - john
john-$>ssh-keygen -t dsa
```

(It will ask for a passphrase in order to protect your private key on the local node)

This command will create john private key on **/home/john/.ssh/id_dsa** (permissions 600) and john public key on **/home/john/.ssh/id_dsa.pub** (permissions 644)

Change de .ssh directory permissions to 755.

```
john-$> chmod 755 .ssh
```

Copy the content of /home/john/.ssh/id_dsa.pub (john public key) generated on node01 (the node from we want to login as john on SSH server) to /home/john/.ssh/authorized_keys on SSH server. If necessary create /home/john/.ssh directory with permission 755 on SSH server.

```
john-$> cat /home/john/.ssh/id_dsa.pub --> >> SSH server(rhel6):/home/john/.ssh/authorized_keys
```

On SSH server (rhel6) change the permissions of /home/john/.ssh/authorized_keys to 644.

```
# chmod 644 /home/john/.ssh/authorized_keys
```

The final step is restart the ssh server and verify that you can connect from SSH client (node01) to SSH server (rhel6) only using public/private key and not using the user password. Have a look on Lab2.

```
# /etc/init.d/sshd reload
```

Note: In order to use the private key on SSH client to connect to SSH server the passphrase introduced when the private key has been created with 'ssh-keygen' is asked. If you have left this passphrase empty you will be able to login to SSH server directly without passphrase BUT using your public/private keys. We do not recommend to left this passphrase empty but in any case this method is more secure that using standard password because in this case the cracker must guess the public/private keys that normaly are random strings with at least 512K of size !!!

Using ssh-agent

When we are running a graphical environment on SSH client as gnome or kde we can use the ssh-add utility in order to do not have to enter the passphrase every time we try to connect to the SSH server.

```
john-$> exec /usr/bin/ssh-agent $SHELL
john-$> ssh-add
```

(--> Enter john passphrase)

The john passphrase now is stored in the environment variables for 'john' graphical session, so john must not be to retype his passphrase any time that try to login to the SSH server from this graphical environment on SSH client.

SSH Port Forwarding

SSH can secure insecure TCP/IP protocols via port forwarding, SSH server becomes an encrypted conduit to the SSH client. Port forwarding maps a local port on the SSH client to a remote port on the SSH server.

```
client> ssh -l john -L 2525:server.info.net:25 server.info.net
```

Once the user john has been logged on server.info.net through this ssh connection an SSH encrypted Tunnel has been established between port 25 TCP/IP on server.info.net and port 2525 TCP/IP on client.info.com. In this way if you execute the command 'telnet localhost 2525' on client.info.com you are making the telnet directly to port 25 TCP/IP on server.info.net.

```
client> telnet localhost 2525
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 server.info.net ESMTP Sendmail 8.13.8/8.13.8; Fri, 25 Mar 2011 13:18:29 +0100
```

!!! IT IS MAGIC !!!

If you want forward a port from a machine that is not running an SSH server, but another machine on the same network is, SSH can still be used to secure a SSH tunnel.

```
client> ssh -l john -L 1100:pop.info.net:110 server.info.net
```

With this command you are making a ssh tunnel from pop.info.net:110 (that is not running an SSH server) to your local machine client.info.com:1100 connecting as user 'john' on server.info.net that is in the same LAN as pop.info.net. As POP service does not encrypt the data itself, with the SSH tunnel the data is encrypted by SSH, so you are making more secure the connection to your pop service.

Note: SSH Tunnels can be used to skip firewalls. Imagine that there is a firewall that blocks the connection between your local machine client.info.com and your POP service on pop.info.net port 110 TCP/IP. If the firewall is not blocking access to the SSH server on pop.info.net:22 (or a machine in the same LAN running SSH server) you can establish an SSH tunnel from client.info.com:110 and pop.info.net:1100 and skip the firewall. **!!! In reality you can forward any port and skip the firewall if you can connect through ssh!!!**

Maybe for security reasons you want to disable port forwarding through your SSH server. In this case the following directive must be configured on the SSH server configuration file **/etc/ssh/sshd_config** and then reload the SSH server service.

```
AllowTcpForwarding no
```

Questions

1.- By default SSH server runs on port 23 TCP/IP (true/false).

2.- OpenSSH can be used to encrypt remote X Windows applications (true/false).

3.- OpenSSH can be used to encrypt the traffic generated by any network application (true/false).

4.- Which command must be used in order to log as user 'kate' on node mark.info.net using ssh?.

5.- Which command must be used in order to copy the local file /root/script.sh on node admin.info.net on /root as root using scp?.

6.- Which command must be used in order to generate user charles RSA public/private keys used by SSH?.

7.- Which configuration parameter must be configured on SSH Server configuration file in order to not allow root logins through ssh?.

8.- Which configuration parameter must be configured on SSH Server configuration file in order to allow only user 'kate' login through ssh?.

9.- Which of the following is the SSH server configuration file?.

A - /etc/ssh_config
B - /etc/ssh/ssh_config
C - /etc/sshd_config
D - /etc/ssh/sshd_config

10.- Which of the following commands will open an SSH Tunnel between remote.info.net:80 and localhost:8080. (SSH Server is running on remote.info.net)?.

A - ssh -l root -L 8080:remote.info.net:80 remote.info.net
B - ssh -l root -L 80:remote.info.net:8080 remote.info.net
C - Both of them
D - None of them

Labs

1.- Configure a SSH server on rhel6 (192.168.1.10). Do not allow root and john users to login to it and allow the rest of users. Verify that SSH data transfers are encrypted.

2.- Re-configure SSH server on rhel6 (192.168.1.10) to allow logins only using public/private keys. Generate the keys for user 'charles' on node01 (192.168.1.101) with an empty passphrase and configure 'charles' account on rhel6 in order to allow 'charles' ssh login from node01.

3.- Using SSH Tunnel (Port forwarding) redirect rhel6 port 23 TCP/IP (telnet) to node01 port 2323 TCP/IP. Verify that you are able to loggin on rhel6 from node01 through telnet skipping the firewall on rhel6.

1.- False.
2.- True.
3.- True, using SSH Tunnel (port forwarding).
4.- 'ssh -l kate mark.info.net' or 'ssh kate@mark.info.net'
5.- 'scp /root/script.sh root@admin.info.net:/root/'
6.- 'su - charles; ssh-keygen -t rsa'
7.- PermitRootLogin no
8.- AllowUsers kate
9.- D
10.- A

Lab 1

* Login as root on rhel6 (192.168.1.10) and configure SSH server to meet the requirements specified.

```
# cat /etc/ssh/sshd_config

...
Protocol 2
..
SyslogFacility AUTHPRIV
...
PasswordAuthentication yes
```

```
...
PermitRootLogin no
DenyUsers john
```

* Reload SSH server service.

```
# /etc/init.d/sshd reload
```

* Open port 22 TCP/IP.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT -> /etc/sysconfig/iptables
# /etc/init.d/iptables restart
```

* Create users john/john, charles/charles on rhel6 if necessary with command 'useradd'.

* From node01 try lo logging to rhel6 through ssh as user john and root and verify that the action is denied by SSH server.

```
node01> ssh -l john 192.168.1.10
```

--> Login Failed after 3 attempts.

```
node01> ssh -l root 192.168.1.10
```

--> Login Failed after 3 attempts.

* From node01 try lo logging to rhel6 through ssh as user charles and verify that the acction is allowed by SSH server. Also run tcpdump command to verify that data transferred in this ssh transaction is encrypted.

```
node01> tcpdump -v -XX port 22
```

```
node01> ssh -l charles 192.168.1.10
```

--> Login successful.
--> Tcpdump does not show clear text data. All data is encrypted.

Lab 2

* On rhel6 modify Lab1 SSH server configuration file to meet the new requirements.

```
# cat /etc/ssh/sshd_config
```

```
...
Protocol 2
...
SyslogFacility AUTHPRIV
...
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
...
PasswordAuthentication no
...
PermitRootLogin no
DenyUsers john
```

* Reload SSH server service.

```
# /etc/init.d/sshd reload
```

* Connect to node01, create charles account and create DSA public/private keys for user charles with an empty passphrase.

```
node01> useradd charles
node01> su - charles
charles-$> ssh-keygen -t dsa
```

--> Enter an empty passphrase

```
charles-$> chmod 755 .ssh
```

* Open a connection to rhel6 and add 'charles' public key (/home/charles/.ssh/id_dsa.pub) to the file rhel6:/home/charles/.ssh/authorized_keys . Create this file if necessary and apply the correct permissions.

```
rhel6-charles# mkdir .ssh
rhel6-charles# chmod 755 .ssh
node01:/home/charles/.ssh/id_dsa.pub --> >> SSH server(rhel6):/home/charles/.ssh/authorized_keys
rhel6# chmod 644 /home/charles/.ssh/authorized_keys
```

* From node01 try to logging on rhel6 server as user charles.

```
node01> ssh -l charles 192.168.1.10
```

--> Login successful with an empty passphrase.

* From node01 try to login through ssh as user john on rhel6.

```
node01> ssh 192.168.1.10 -l john
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

This message means that john has not public/private keys configured on node01/rhel6 so as this is the unique valid method for ssh authentication on rhel6 the logging is DENIED before the directive 'DenyUsers john' has taken effect. The same happens to root account.

Lab 3

* Login as root on rhel6 and install telnet-server rpm. Start it and make sure that it will start at boot.

```
# yum install telnet-server
# chkconfig telnet on
# /etc/init.d/xinetd start
# chkconfig xinetd on
```

* From node01 try to open a telnet connection to rhel6 server. This connection will be blocked by the firewall on rhel6.

```
node01> telnet 192.168.1.10 23

Trying 192.168.1.10...
telnet: connect to address 192.168.1.10: No route to host
telnet: Unable to connect to remote host: No route to host
```

* Using a SSH Tunnel forward rhel6:23 to node01:2323 as user 'charles'.

```
node01>su - charles
charles-$> ssh -l charles -L 2323:192.168.1.10:23 192.168.1.10
```

* Verify now that you can telnet to rhel6 just making 'telnet localhost 2323' on node01.

```
node01> telnet localhost 2323

Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Red Hat Enterprise Linux Server release 6.0 (Santiago)
Kernel 2.6.32-71.el6.i686 on an i686
login: charles
```

```
Password:
Last login: Sun Mar 27 22:38:43 from 192.168.1.2

charles@rhel6> hostname
rhel6.info.net
```

* Magnific, we have skipped the firewall and at the end we have logged through telnet on rhel6 from node01.

Linux Services Organization : Linux MAIL Linux Server

Email messages transaction is done using a client/server topology. An email message is created using a mail client program that sends the message to an email server using **SMTP** protocol. The server then forwards the message to the recipient's SMTP email server, where the message is then supplied to the recipient's email using **POP**/**IMAP** protocols.

SMTP Simple Mail Transfer Protocol is a set of rules for transferring email data used by various mail transfer agents in order to transport emails messages from the source where the email is created to the destination recipient. As many services on the Internet SMTP depends on DNS resolutions and routing in order to delivery the email to recipient SMTP email server. Once the email have reached the SMTP email server, the email is dropped to the final user email client using POP or IMAP protocols.

SMTP Server

As said before the purpose of SMTP server is to transfer email between mail servers. To send email, the client sends the message to an outgoing mail server, which in turn contacts the destination mail server for delivery.

SMTP protocol does not require authentication. It allows anyone on the Internet to send email to anyone else or even to large groups of people. Imposing relay restrictions limits any users on the Internet from sending email through your SMTP server, to other servers on the Internet. Servers that do not impose such restrictions are called **open relay** servers and are labelled as SPAM SMTP server.

On RHEL6 the default SMTP email server is 'postfix' installed by postfix rpm. The 'postfix' service listen on port **25 TCP/IP**, it is configured on **/etc/postfix** directory files and logs on **/var/log/maillog**.

```
# yum install postfix
```

/etc/postfix/main.conf

The main postfix SMTP server configuration file is **/etc/postfix/main.conf**. The following are the main directives that can be configured.

```
# cat /etc/postfix/main.conf

...
# This directive configures from which domain the postfix server is going to be the SMTP server.
mydomain = info.net

...
# It complements the email address with 'mydomain' domain. For example a mail for user 'john' -> 'john@info.net'
myorigin = $mydomain

# In which server interfaces the SMTP server port 25 TCP/IP must be listening. In this case it will be listening on all
system interfaces.
inet_interfaces = all

...
# The mydestination parameter specifies the list of domains that this machine considers itself the final destination for.
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain, server.$mydomain,
mail.$mydomain

...

# The mynetworks parameter specifies the list of "trusted" SMTP clients that have more privileges than "strangers".
mynetworks = 192.168.01.0/24, 127.0.0.0/8

...
# The home_mailbox parameter specifies the pathname of a mailbox file relative to a user's home directory where the mailbox will be stored
home_mailbox = Maildir/
...
```

Once configured the postfix service just start it and make sure that it will be started at boot.

```
# /etc/init.d/postfix restart
# chkconfig postfix on
```

SMTP Server Security

Firewall

As said before SMTP server listen on port 25 TCP/IP. It also uses port 25 UDP for data transactions so both ports must be open in order to allow SMTP service through a firewall.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 25 -j ACCEPT
```

SElinux

The unique SElinux parameter that can be configured is the ability to write on /var/spool/mail by postfix service. In this directory is where are stored the email received by the SMTP server and is enabled by default.

```
# setsebool -P allow_postfix_local_write_mail_spool 1
```

Host Based Restrictions

Using the following configuration parameters on **/etc/postfix/main.cf** and **/etc/postfix/access** files is possible restrict the access to the SMTP server based on host/IP address.

```
smtpd_client_restrictions=hash:/etc/postfix/access --> /etc/postfix/main.cf
```

```
# echo "192.168.2.0/24 OK" >> /etc/postfix/access
# echo "192.168.2.10 REJECT" >> /etc/postfix/access
# postmap /etc/postfix/access
# /etc/init.d/postfix reload
```

With this configuration the SMTP server must allow connections from clients on 192.168.2.0/24 LAN except from 192.168.2.10 IP.

Email Domain Forwarding

Sometimes is necessary to forward any incoming email for a secondary domain that our SMTP server recognises as virtual domain to a secondary SMTP.

```
virtual_alias_domains = example.net --> /etc/postfix/main.cf
virtual_alias_maps = hash:/etc/postfix/virtual --> /etc/postfix/main.cf
```

```
# echo "@example.net infonetaccount@example.com" >> /etc/postfix/virtual

# postmap /etc/postfix/virtual
# /etc/init.d/postfix reload
```

With this configuration any incoming email with '@example.net' address destination will be forwarded to 'infonetaccount@external.com' account on external.com SMTP server.

Email Forwarding : /etc/aliases

For one-to-one email forwarding is much easier the use of the /etc/aliases file.

```
# echo "root: john" >> /etc/aliases
# echo "sales: charles,john,mike" >> /etc/aliases
# echo "charles: charles@gmail.com" >> /etc/aliases
# newaliases
```

With this configuration any email coming to root@info.net will be forwarded to john@info.net without leaving our info.net SMTP server. We have also created the email group called 'sales@info.net', any email directed to this address will be forwarded to john, charles and mike email addresses. Also any email coming to charles@info.net will be forwarded to charles@gmail.com on gmail.com SMTP server.

SMTP and DNS

When a SMTP server has to send an email to an external SMTP server it relies on DNS name resolution to send the email to the correct SMTP server IP. For example if our info.net SMTP server has to send a message to charles@gmail.com account, our SMTP server will look for the domain gmail.com **MX** registry using the local DNS configured on /etc/resolv.conf as said on DNS lesson.

```
# dig gmail.com mx
...
gmail.com. 345 IN MX 10 alt1.gmail-smtp-in.l.google.com.
...
```

So the email to charles@gmail.com will be forwarded by our SMTP server to alt1.gmail-smtp-in.l.google.com port 25 TCP/IP where the gmail.com SMTP server is running.

Taking into account the strong relation between SMTP and DNS, in order to make your SMTP server public on the Internet and receive emails from others servers, your domain DNS server must have the MX registry pointing to the SMTP server IP. Of course your DNS must be also public to the Internet.

When name resolution is not working, postfix doesn't know where to send your outbound e-mail. These messages are placed in a queue that tries to resend your e-mail at regular intervals. Messages like following are written to /var/log/maillog in this situation. .

```
550 5.1.2 mike@gmail.com ... Host unknown
```

All mails queued on the SMTP server can be displayed with the command mailq. Some info about the reason of why they are queued is displayed also.

```
# mailq
```

SMTP Open Relay

An Open Relay SMTP server is configured in a way that processes a mail message from any client on the Internet (Open) where neither the sender or the recipient is a local user (Relay) . An Open Relay SMTP server can be used by spammers in order to send SPAM emails to anywhere to the Internet, making the SMTP server labelled as SPAM source and then all emails coming from it will be labelled as SPAM.

By default postfix SMTP server on RHEL6 systems is NOT configured as an Open Relay SMTP server. It only allows RELAY from clients on the internal network specified on 'mynetwork' configuration parameter on /etc/postfix/main.cf.

```
mynetworks = 192.168.1.0/24, 127.0.0.0/8
```

*** On Lab1 can be seen the procedure to test if a SMTP server is configured as Open Relay. ***

POP/IMAP Server

POP Post Office Protocol and IMAP Internet Message Access Protocol are two protocols used by email client applications to retrieve email from mail servers. While POP downloads all e-mail to the client, an IMAP server maintains all mail messages on the server. IMAP is commonly used by businesses that service users who log in from different locations. It's also the most common mail delivery protocol for Web-based mail services.

On RHEL6 systems both protocols are handled by '**dovecot**' service installed by dovecot rpm.

```
# yum install dovecot
```

The **/etc/dovecot/dovecot.conf** file is used to configure POP (port 111 TCP/UDP) , IMAP (port 143 TCP/UDP) services and his secure versions POPs (port 995 TCP/UDP) and IMAPs (port 993 TCP/UDP) protocols.

```
# cat /etc/dovecot/dovecot.conf

...
protocols = imap pop3
...
mail_location = maildir:~/Maildir
...

# /etc/init.d/dovecot restart
# chkconfig dovecot on
```

Firewall

Open the corresponding TCP/UDP ports on the firewall to allow POP/IMAP dovecot services to run through the system firewall.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 995 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 995 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 993 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 993 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 143 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 143 -j ACCEPT
```

Email clients

In order to use an email client that uses the SMTP server to send and receive emails '**evolution**' GUI email client tools can be used installing evolution rpm.

```
# yum install evolution
```

The client email must be configured to use the SMTP server for outgoing email and dovecot (POP/IMAP) to retrieve the email from the SMTP server. It is also possible the use of command line email clients as '**mail**' command installed by default on the RHEL6 server installation by '**mailx**' rpm.

```
# echo "Test message" | mail -s "Test subject" root@info.net
```

It uses the system SMTP server to send and receive emails. It can also be used to read emails just typing 'mail' command. It will open the local mailbox for the user that has executed the mail command.

Questions

1.- Sendmail is a valid alternative SMTP server to postfix (true/false).
2.- Dovecot is a valid alternative SMTP server to postfix (true/false).(true/false).
3.- Dovecot provides POP and IMAP server services (true/false).
4.- Which SElinux command must be executed in order to allow postfix to write emails on /var/spool/mail?.
5.- Which command can be used in order to display the emails queued on a SMTP server?.
6.- Which command can be used in order to send an email to local root user ?.
7.- Which command must be executed just before updating an email alias on /etc/aliases and make the alias available to the postfix SMTP server?.
8.- Which type of DNS registry is used to locate the domain SMTP server hostname?.
9.- Which of the following is the standard dovecot configuration file ?.

A - /etc/dovecot/dovecot.conf
B - /etc/dovecot.conf
C - /etc/mail/dovecot.conf
D - /etc/postfix/dovecot.conf
10.- Which of the following is the standard postfix configuration file ?.

A - /etc/mail/postfix.conf
B - /etc/postfix/master.conf
C - /etc/mail/master.conf
D - /etc/postfix/main.conf

Labs

1.- Configure a SMTP email server on rhel6 (192.168.1.10) for the 192.168.1.0/24 LAN, domain info.net. Verify from node external01 10.0.0.101 that the SMTP server is not an Open Relay SMTP server.
2.- Install and configure dovecot on rhel6 in order to retrieve emails from the SMTP server configured on Lab1 using just IMAPs protocol.
3.- On node01 install and configure Mozilla Thunderbird email client. Configure 'john@info.net' account using rhel6 as SMTP server and IMAPs server. Configure /etc/aliases to forward any email send to root@info.net to be forwarded to john@info.net and test it using Evolution email client..
1.- True.
2.- False.
3.- True.
4.- 'setsebool -P allow_postfix_local_write_mail_spool 1'
5.- 'mailq'
6.- 'echo 123 | mail -s "123" root'
7.- 'newaliases'
8.- DNS MX registry
9.- A
10.- D

Lab 1

* Login as root on rhel6 (192.168.1.10) and install postfix SMTP server.

```
# yum install postfix
```

* Configure postfix on /etc/postfix/main.conf to act as SMTP server for info.net domain, 192.168.1.0/24 LAN.

```
# cat /etc/postfix/main.conf
...
```

```
mydomain = info.net
...
myorigin = $mydomain
...
inet_interfaces = all
...
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain, rhle6, rhel6.$mydomain
...
mynetworks = 192.168.1.0/24, 127.0.0.0/8
...
home_mailbox = Maildir/
...
```

* Start postfix service and make sure it will start at boot.

```
# /etc/init.d/postfix restart
# chkconfig postfix on
```

* Open ports 25 TCP/UDP on the system firewall.

```
# -A INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT --> /etc/sysconfig/iptables
# -A INPUT -m state --state NEW -m udp -p udp --dport 25 -j ACCEPT --> /etc/sysconfig/iptables
# /etc/init.d/iptables restart
```

* Make sure that the DNS that is using rhel6 server has configured the MX registry for domain info.net pointing to rhel6 server IP. If not configure MX registry on the DNS server following the instructions on 'lesson 19 Linux DNS server'

```
# dig info.net mx
...
;; ANSWER SECTION:
info.net. 86400 IN MX 10 rhel6.info.net.

...
```

* From node01 (192.168..101) test the SMTP server with telnet commands and send an email from root@info.net to john@info.net.

```
node01>telnet 192.168.1.10 25

Trying 192.168.1.10...
Connected to laptop (192.168.1.10).
Escape character is '^]'.
220 rhel6.info.net ESMTP Postfix
helo rhel6
250 rhel6.info.net
MAIL FROM: root@info.net
250 2.1.0 Ok
RCPT TO: john@info.net
250 2.1.5 Ok
data
354 End data with .
john email test
.
250 2.0.0 Ok: queued as 181F010CB9
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Postfix SMTP server runninig on rhel6:25 has accepted the email to john@info.net, on /var/log/maillog the transaction has been logged

```
# cat /var/log/maillog

rhel6 postfix/local[4676]: 181F010CB9: to=, relay=local, delay=30, delays=30/0.02/0/0.01, dsn=2.0.0,
status=sent (delivered to mailbox)

# cat /home/john/Maildir/cur/1301853043.Vfd00I10cc5M533105.rhel6.info.net\:2\,

From root@info.net Sun Apr 3 15:54:36 2011
Return-Path:
X-Original-To: john@info.net
Delivered-To: john@info.net
Received: from rhel6 (unknown [192.168.1.101])
by rhel6.info.net (Postfix) with SMTP id 181F010CB9
for ; Sun, 3 Apr 2011 15:54:06 +0200 (CEST)

john email test
```

The email sent by used root from node01 has been received correctly by user john on rhel6 SMTP server.

* Verify that SMTP Relay is allowed from node01 (192.168.1.101).

```
node01> telnet 192.168.1.10 25

Trying 192.168.1.10...
Connected to laptop (192.168.1.10).
Escape character is '^]'.
220 rhel6.info.net ESMTP Postfix
helo rhel6
250 rhel6.info.net
MAIL FROM: pp@pp.net
250 2.1.0 Ok
RCPT TO: kk@kk.net
250 2.1.5 Ok
data
354 End data with .
test
.
250 2.0.0 Ok: queued as B388A10CBA
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

As expected RELAY (from pp@pp.net -> to kk@kk.net) from node01 (192.168.1.101) is allowed because of node01 is in the internal network 192.168.1.0/24 configured on 'mynetwork' parameter on /etc/postfix/main.conf.

* Verify that this SMTP server is not an Open Relay SMTP server repeating the Relay test from a client outside 'mynetwork', external01 (10.0.0.101). Of course there is connectivity between rhel6 and external01.

```
external01> telnet 192.168.1.10 25

Trying 192.168.1.10...
Connected to 192.168.1.10.
Escape character is '^]'.
220 rhel6.info.net ESMTP Postfix
helo rhel6
250 rhel6.info.net
MAIL FROM: pp@pp.net
250 2.1.0 Ok
RCPT TO: kk@kk.net
554 5.7.1 : Relay access denied
```

As can be seen (**Relay access denied !!!**) in this case RELAY is denied because external01 is not in any LAN configured as 'mynetwork' so this SMTP server **IS NOT AN OPEN RELAY**.

Lab 2

* Login as root on rhel6 (192.168.1.10) and install dovecot. Configure dovecot in order to retrieve email via IMAPs protocol.

```
# yum install dovecot
protocols = imap --> /etc/dovecot/dovecot.conf
mail_location = maildir:~Maildir--> /etc/dovecot/dovecot.conf
```

* Start dovecot service and make sure it will start at boot.

```
# /etc/init.d/dovecot restart
# chkconfig dovecot on
```

* Open 993 TCP/UDP IMAPs ports on rhel6 system firewall in order to allow IMAPs transactions through the firewall.

```
# -A INPUT -m state --state NEW -m tcp -p tcp --dport 993 -j ACCEPT --> /etc/sysconfig/iptables
# -A INPUT -m state --state NEW -m udp -p udp --dport 993 -j ACCEPT --> /etc/sysconfig/iptables
# /etc/init.d/iptables restart
```

* From node01 verify that there is access to IMAPs service on rhel6.

```
node01> telnet 192.168.1.10 993

Trying 192.168.1.10...
Connected to laptop (192.168.1.10).
Escape character is '^]'.
```

Lab 3

* Login as root on node01 (192.168.1.101) and install Evolution email client. Execute Evolution and configure john@info.net account using rhel6 server as SMTP and IMAPs server.

```
node01> yum install thunderbird
node01> thunderbird
```

Edit-->Account Settings-->Email Account-->Next-->Name: john Email Address: john@info.net-->Next--> IMAP incoming server : 192.168.1.10, SMTP outgoing server: 192.168.1.10 -->Incoming user name: john-->Next-->Account name: john@info.net-->Next-->Finish-->Server Settings--> Click on SSL-->OK-->Click on john@info.net and type john password when asked (passwd: john)--> Now you are connected to john@info.net email account.

* From newly created account write a new message to john@info.net, send it and verify that you get the message on the john@info.net account.

thunderbird --> Write--> TO : john@info.net Subject: Test ... --> Send

Automatically the email with subject Test appears on john@info.net account .

* Configure on rhel6 SMTP server /etc/aliases to forward any email with root@info.net address destination to john@info.net.

```
# echo "root: john" >> /etc/aliases
# newaliases
```

* From thunderbird john@info.net account on node01 send an email to root@info.net. That email should appear on john@info.net mailbox.