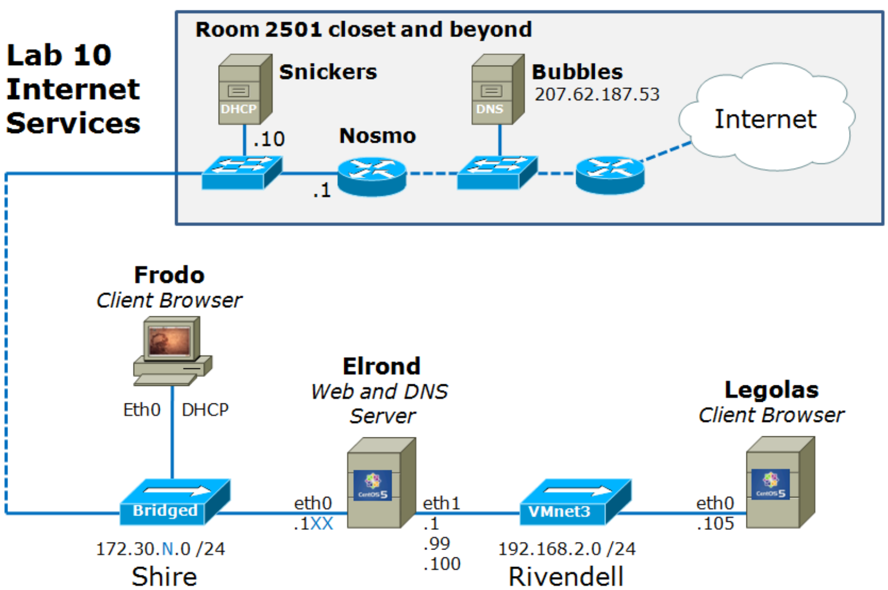
Lab 10: Internet Services

The goal of this lab is to configure a Linux system as a web server capable of hosting multiple web sites. You will do this in three different ways:

1. Multiple websites will be supported through user accounts on the system hosting the web service. These accounts will be accessible through Apache's support of home directories via the *servername/~username* construct.
2. A second approach involves virtual domains, so that our websites appear to be independent of each other. These virtual domains will be based on multiple IP addresses assigned to the web server, one per virtual domain.
3. The third method will use name-based virtual domains. Obviating the need for multiple IP addresses, this approach requires clients to support HTTP 1.1



.1XX is based on your station number and the IP Table in the Appendix
N=1 for the classroom and N=4 for the CIS lab or CTC

Supplies

- VMWare Server 1.08 or higher
- 192 VMs shown above

Preconfiguration

- Original versions of all VMs. Note, this will set the network configurations back to down or DHCP settings.
- You will need access to a DHCP server to assign addresses for the 172.30.N.0/24 network. This is already configured if the lab is done using the CIS VMware Stations in the CIS Lab (room 2504) or the CTC. If you plan to do this lab at home see: http://simms-teach.com/howtos/202-working-at-home-nat.pdf

Forum

Use the forum to ask and answer questions, collaborate, and report any equipment issues. Post tips and any lessons learned when you have finished. Forum is at: http://opus.cabrillo.edu/forum/viewforum.php?f=5

Background

Virtual domain web hosting relies on DNS to provide address records and aliases for the host machine. Therefore you will have to set up DNS, as we did in a previous lab, to support methods two and three described above.

The services and programs we will be using for this lab are:

- *named (Bind)*
- *httpd (Apache)*
- *mozilla or firefox*

The installation of apache and configuring the home directories will require root access.

Setup

1. Revert Frodo, Elrond and Legolas to their snapshots.
2. Cable and configure Frodo, Elrond, and Legolas as shown in the map above.
3. Use **ifconfig eth1:1 192.168.2.99/24** and **eth1:2 192.168.2.100/24** to add the IP aliases on Elrond. Refer to Lesson 5 for setting the permanent alias IP addresses.

Part I

In this step, you will turn Elrond into a Linux Web server using Apache. Each user account will be enabled to publish files from their own *public_html* directory. The URL to access a user's site is http://*servername*/~*username*. This involves configuring the *httpd.conf* file, creating some users, adding web file content, setting file permissions and SELinux contexts, and starting the httpd service.

1. Log in as root on Elrond and verify the httpd package installed:
   rpm –qa | grep httpd
   Note: it's also useful to have the httpd manual pages which can be installed with:
   yum install httpd-manual
2. Edit */etc/httpd/conf/httpd.conf* and:

- Uncomment and modify the **ServerName** directive with your server's full domain name (elrond.rivendell). Be sure to append the port number :80 to the name.
- Search for the **UserDir** option. By default, this option is disabled, thereby not allowing users on the system to have websites within their home directories. To allow this feature, comment out the following line:
  UserDir disable to **#UserDir disable**
  and uncomment the line below it:
  #UserDir public_html to **UserDir public_html**
  public_html is the directory name that serves as the DocumentRoot for each user's web site.

3. Create four new user accounts: legolas, elrond, celebrian, and arwen:
   useradd -g users *username*
   passwd *username*
   Make sure the permissions on the home directories are set to 751
   chmod 751 /home/*username*
4. Make sure that each user’s home directory has a subdirectory called *public_html* with the same permissions as the home directory.
5. Populate each of the four *public_html* directories with a unique *index.html* web page. Feel free to use and modify the sample *index.html* file and **.jpg* images in the */home/cis192/depot* directory on Opus.
6. Verify and modify if necessary permissions as follows:
   - home and *public_html* directories are set to 751
   - *index.html* and **.jpg* files are set to 644
7. Check that SELinux is set to enforcing with **getenforce** and home directories are enabled with **getsebool httpd_enable_homedirs**
8. Modify the SElinux context for all *public_html* directories with
   chcon -vR -t httpd_sys_content_t /home/*/public_html
9. Start the httpd services:
   service httpd start
10. Log in on another terminal as **cis192** and start a web browser in a graphical session. Visit the default Apache home page by entering the following line in the browser's address text box:
    http://elrond
    

11. Visit each of the user's home pages at the following addresses:
    http://elrond/~*username*
    

12. Open TCP port 80 in your firewall (refer to Lesson 14 if needed) and try browsing from a neighboring machine, (make sure that your computer's host name is resolved to an IP address in the clients */etc/hosts* file).

There may be permission, SELinux or firewall settings that result in "Forbidden", "Failed to Connect" and other errors. See the troubleshooting section in the appendix to resolve.

Apache includes the *mod_userdir* module by default, so user's home directories are supported for having web publishing capabilities.

Part II

In this procedure we will create virtual domains using multiple IP addresses to distinguish each website. We will need to configure DNS and add additional A records to the zone file. These websites will be kept the /www directory rather than in user's home directories.

1. Install the DNS service packages on Elrond with:
   yum install bind caching-nameserver
2. Log in as root and change directory to */var/named*.
3. Verify that the DNS packages are installed:
   rpm –qa | grep bind
   rpm –qa | grep caching
4. Create the two zone database files, *db.rivendell* and *db.2.168.192* in your */var/named* directory. You may use the samples in the Appendix for these two files.
5. Change directory to */etc* and create the *named.conf* file for these two zones. See the Appendix for a sample file.
6. Now edit the *resolv.conf* file to reflect the fact that your server is the DNS server for this domain:
   search rivendell
   nameserver *server-ip-address*
7. Also check the **hosts** line in the */etc/nsswitch.conf* file to make sure that **dns** is in front of the **files** keyword. This will insure we're getting resolution from DNS and not the */etc/hosts* file.
8. Open port 53 on your server. Optionally you may wish to provide NAT services for Rivendell clients and open up the FORWARD chain with:
   iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
   iptables -D FORWARD 1
9. Now you are ready to start the DNS service:
   service named start
10. Do not proceed until the named server starts successfully.
11. Test your DNS service with the host command:
    host elrond
    host legolas
12. The next step is to configure the HTTP server. Do this by editing Section 3 of the *httpd.conf* file, which is for virtual hosting.
13. Add a VirtualHost directive to the end of Section 3 just below the commented out example of a virtual host (at the end of the file):
    <VirtualHost 192.168.2.99>
    ServerName remus-farm.rivendell
    DocumentRoot /www/remus-farm
    </VirtualHost>

    <VirtualHost 192.168.2.100>
    ServerName holy-grail.rivendell
    DocumentRoot /www/holy-grail
    </VirtualHost>
    Notice that both DocumentRoots don't exist yet!

14. From the root directory (/), create the Document Root directories:
    cd /
    mkdir -p www/remus-farm www/holy-grail
    chmod -R 751 www
15. Now populate these directories with unique index.html files. You can download a template as well as additional image files from the */home/cis192/depot* file on Opus.
16. Our final preparatory step is to add the virtual domain names to our forward lookup zone file:
    cd /var/named
    vim db.rivendell
    Add the following two A records at the bottom of the address section:
    remus-farm IN A 192.168.2.99
    holy-grail IN A 192.168.2.100
    Save your changes and exit vi.
17. We are now ready to try this out. Reload and restart the *named* and *httpd* services:
    rndc reload
    service httpd restart
    Note: use **chcon -R –v -t httpd_sys_content_t /www** so SELinux won't block Apache serving from a DocumentRoot other than /var/www/html/.
18. If you want to test this from a different machine on the network, you'll have to point that client's resolver to your machine by editing the */etc/resolv.conf* file appropriately.
19. Visit your virtual domains:
    http://remus-farm
    

    http://holy-grail.rivendell
    

    Also try **http://192.168.2.99, http://elrond, http://elrond/~arwen**
20. What happens when you try to access a virtual host that doesn't exist? e.g.
    http://arwen.rivendell
    How about: **http://www.elrond**

Part III

We will now configure our web server to use name-based virtual domain hosting. The different websites are distinguished by the name of the server used in the URL. CNAME records are added to the zone file so the server can have additional names (aliases).

1. Remove the IP aliases we made earlier:
    ifconfig eth1:1 down
    ifconfig eth1:2 down
2. Edit the *httpd.conf* file:

- Uncomment the line: #NameVirtualHost and replace it with:
  NameVirtualHost *your-machine-ip-address*
- Replace each of the two *aliased* ip addresses in the <VirtualHost> directives with the ip address of your machine, (the hosting system).

3. Change directory to */var/named* and remove the two address records you added in the last procedure. At the bottom of the zone file, add the following two CNAME records:
   remus-farm **IN CNAME** *your-servername*
   holy-grail **IN CNAME** *your-servername*
4. That is all there is to it. Reload and restart the *named* and *httpd* services:
   rndc reload
   service httpd restart
   and browse to **http://remus-farm.rivendell** and **http://holy-grail.rivendell**

5. Additional directives can be specified in these VirtualHosts sections. They will override the global settings in Sections 1 and 2. For instance, try logging transactions and errors separately for each virtual domain:
   TransferLog /www/*domain*/transfer_log
   ErrorLog /www/*domain*/error_log
6. Test your web sites as you did in procedure two. Do you notice any different behavior? What is the default behavior when you make a request for a virtual domain that resolves to an IP address, but isn't supported as a virtual host? The answer to this question is important for you to discover :-) Compare:
   http://remus-farm with **http://holy-grail**

These are the three most popular ways of hosting multiple websites on a single server. The first technique is what is being used by the Cabrillo web server for the faculty and staff websites. The second method is used by companies that want to support browsers still using HTTP 1.0 protocol, for they cannot handle name-based virtual domains. The third method is the preferred method for modern web-hosting.

To turn in

Your *lab10* **text** file should contain the following sections.

- Standard boilerplate information:
  - CIS 192 Lab *XX*
  - *Name*
  - *Date*
  - TBA hours*: X.X*
  - Station number: CIS-Lab-*XX*
- **sed -e '/^#/d' /etc/httpd/conf/httpd.conf** output
- db.rivendell file
- **iptables -nL** output
- **ls -lZR /www /home** (permissions and SELinux contexts)
- **tail -n 20 /www/*/*log** (all four log files)
- Example command/configuration summary

The command/configuration summary should be a concise set of documented examples that can be used as a resource for repeated operations in future labs.

Check your work for completeness then submit as many times as you wish up until the due date deadline. Remember, **late work is not accepted**, so start early, plan ahead for things to go wrong and use the forum to ask questions.

[p]scp lab10 cis192@opus.cabrillo.edu:lab10.*lastname*

Grading rubric (30 points)

2 points for correct submittal, professional appearance and quality
5 points for correct httpd.conf file
5 points for correct db.rivendell zone file
5 points for correct firewall settings (open for DNS and HTTP)
5 points for correct SELinux settings
5 points for correct transfer and error logs (four logs total)
3 points for complete and concise command summary

Appendix - Static IP address table by station number:

Station	IP	Static 1	Static 2
CIS-Lab-01	172.30.4.101	172.30.4.121	172.30.4.122
CIS-Lab-02	172.30.4.102	172.30.4.123	172.30.4.124
CIS-Lab-03	172.30.4.103	172.30.4.125	172.30.4.126
CIS-Lab-04	172.30.4.104	172.30.4.127	172.30.4.128
CIS-Lab-05	172.30.4.105	172.30.4.129	172.30.4.130
CIS-Lab-06	172.30.4.106	172.30.4.131	172.30.4.132
CIS-Lab-07	172.30.4.107	172.30.4.133	172.30.4.134
CIS-Lab-08	172.30.4.108	172.30.4.135	172.30.4.136
CIS-Lab-09	172.30.4.109	172.30.4.137	172.30.4.138
CIS-Lab-10	172.30.4.110	172.30.4.139	172.30.4.140
CIS-Lab-11	172.30.4.111	172.30.4.141	172.30.4.142
CIS-Lab-12	172.30.4.112	172.30.4.143	172.30.4.144
Pod 1		172.30.4.113	172.30.4.145
Pod 2		172.30.4.114	172.30.4.146
Pod 3		172.30.4.115	172.30.4.147
Pod 4		172.30.4.116	172.30.4.148

Appendix – Web Server Troubleshooting

- **403 Forbidden**
  - Insure 751 permissions on home and public_html directories
  - Insure 644 permissions on index.html and *jpg files
  - SELinux changes
    - To show SELinux context use **ls –lZR** on files and directories
    - httpd_enable_homedirs must be on (**setsebool –P httpd_enable_homedirs=1**)
    - To change context on published files and directory use:
      chcon -R -v user_u:object_r:httpd_sys_content_t /home/*/public_html
      or **chcon -R –v -t httpd_sys_content_t /www**

- **Failed to connect**
  - Open HTTP port in firewall with:
    iptables -I RH-Firewall-1-INPUT 9 -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
    (all on one line)
  - To make the firewall changes permanent use:
    iptables-save > /etc/sysconfig/iptables

- **Address not found**
  - Might be a typo in the URL or /etc/hosts
  - The firewall on Elrond may not be opened to allow DNS requests. Open with:
    iptables -I RH-Firewall-1-INPUT 9 -m state --state NEW -m udp -p udp --dport 53 -j ACCEPT

Appendix – DNS configuration files

```
[root@elrond bin]# cat /etc/named.conf
options {
        directory "/var/named";
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below. Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        query-source address * port 53;
};

//
// a caching only nameserver config
//
controls {
        inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

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

zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
        allow-update { none; };
};

zone "rivendell" IN {
        type master;
        file "db.rivendell";
        allow-update { none; };
};

zone "2.168.192.in-addr.arpa" IN {
        type master;
        file "db.2.168.192";
        allow-update { none; };
};

// A key file needs to be referenced for use by rndc
include "/etc/rndc.key";

[root@elrond bin]#
```

```
[root@elrond ~]# cat /var/named/db.rivendell
$TTL 604800
; Rivendell Zone Definition
;
;
Rivendell.      IN SOA elrond.rivendell. root.rivendell. (
                2010050500      ; serial number
                8H              ; refresh rate
                2H              ; retry
                1W              ; expire
                1D)             ; minimum
;
;Name Server Records
Rivendell.      IN NS elrond.rivendell.

;
;Address Records
localhost       IN A 127.0.0.1
elrond          IN A 192.168.2.1
legolas         IN A 192.168.2.105


[root@elrond ~]# cat /var/named/db.2.168.192
$TTL    86400
;192.168.2.* Reverse Zone Definition
;
2.168.192.in-addr.arpa. IN SOA  elrond.rivendell. root.rivendell.  (
                           2010050500      ; serial number
                           8H              ; refresh rate
                           2H              ; retry
                           1W              ; expire
                           1D)             ; minimum

;
;Name Server Records
;
2.168.192.in-addr.arpa. IN NS elrond.rivendell.
;
;Address Records
1                       IN PTR  elrond.rivendell.
105                     IN PTR  legolas.rivendell.
```