

Lab 6 - Firewalls and Apache Web Server

Lab 6 - Firewalls and Apache Web Server	1
Lab Description	1
Lab Setup	2
Lab Directions	2
Part 1 - The Firewall	3
Part 2 - Basic Authentication Web Site	5
Part 3 - CAS Authentication	7
Part 4 - HTTPS	8
Extra Credit Questions	9

This lab will be due by 11:59PM April 27 2017. Submit the writeup as a Word document or a PDF in Canvas. As you go through the lab, responses that need to be included in the write up are prefixed with a number. All questions are worth 1 point excepted where noted. Make sure you number your responses accordingly in your submission. There are extra credit questions worth 2 points each at the end of the lab.

Lab Description

In this lab you will be working with a new VM. This VM will have a two interfaces. One interface is bound to a private network upon which only the host system and the VM can communicate. The other interface will acquire an IP from the network that your host is attached to. It will get an IP separate from your host. This will allow your system (or any other system on the network) access to the web server running the VM.

NOTE: your machine must be on a network with DHCP active for the VM to work properly.

This VM has a web server with several virtual host configurations available. It also has an iptables firewall configured. You will need to enable the virtual host configurations, update the firewall accordingly, and then access the web sites with a browser running on your host.

This lab will require you to edit and access files that are owned by root. So most commands you will need to issue with sudo. Alternatively you can use the **sudo su** command to launch a shell with root privileges, though this is slightly more dangerous since all commands typed at the command line will be executed with root privileges.

You can use the command **nano** to edit files. Don't forget about using the commands **cat** or **less** to display the contents of the files. You will need to **cd** to several directories.

Lab Setup

Follow these steps to setup the VM on your system.

1. Create a directory on your system named something like I230_Lab6.
2. Download into this folder the Vagrantfile and the I230_Lab6.box files from the Canvas Lab 6 Materials directory. Make sure the Vagrantfile file is named correctly with no extra extensions from the download process.
3. You will need to add the lab 6 box to vagrant with the command **vagrant box add I230_Lab_6 I230_Lab6.box** from within your lab directory.
4. Bring up the VM with the command **vagrant up**. Vagrant will ask you which host interface to use for the VM's bridged network. Select the option that your host is using to access the network. This will probably be either Wi-Fi or Ethernet and will probably be the first option.
5. Once the VM is launched, login to it with the **vagrant ssh** command.

Lab Directions

These directions assume you have followed the setup instructions above and are logged into the VM. You should also have a browser window open. You may find it convenient to open a couple different tabs in your browser or open a couple windows for the different web sites used in the lab, but it is not necessary.

Part 1 - The Firewall

When you first login to the VM it will display the IP number that is bound to the eth1 interface. This interface will be using a bridged network and communicate directly with the internet just like your host machine. The eth0 interface is bound to an internal network over which only your host and the VM can communicate.

Update your /etc/hosts file on you host system to include the line:

0.0.0.0 www.i230.edu

Replace the 0.0.0.0 above with the actual IP reported to be used by the VM's eth1 interface. Windows users can edit the file with Notepad (launched with administrator privileges) and the file is located in

C:\Windows\System32\drivers\etc\

Mac users can use the command **sudo nano /etc/hosts**.

This website has specific instructions for editing /etc/hosts in different OS's.

<https://www.howtogeek.com/howto/27350/beginner-geek-how-to-edit-your-hosts-file/>

NOTE: if you stop working on the lab and resume in another location you will want to halt your vm and bring it backup. Then you will need to repeat the above step because the IP bound to eth1 will have likely changed. The firewall is reset to the default configuration on each reboot.

1) Why do you need to update your /etc/hosts file in order to access the web server with a URL like <http://www.i230.edu> from your host system's browser? (2 points)

Run the command **iptables -L -v --line-numbers** (don't forget to use it with **sudo** or launch a root shell with **sudo su**.)

2) Looking at the INPUT chain policy which is DROP, is this a blacklist or a whitelist policy? Why?

Now you will enable the the first website on your web server. To enable the website you will need to run the command **a2ensite noauth** and then you will need to tell Apache to reload its configuration with the **service apache2 reload** (again you will need root privileges.)

Examine the configuration file located in */etc/apache2/sites-enabled/noauth.conf* with a command like cat, less, or nano.

3) On which port is the noauth virtualhost listening?

4) From which directory does the virtualhost serve web pages?

Try to connect to your web site with the URL <http://www.i230.edu>. You should find that your browser is not connecting.

5) Why is your browser not connecting? (Hint: look at the iptables rules and think about which interface your web browser is trying to talk to the web server.) **(2 points)**

Use the tail command to look at the last few lines of the `/var/log/kern.log` file. This is where iptables will log packets it drops.

6) Copy and paste the line from the log describing the dropped packet.

7) Looking at that line, from which port on your host did your browser connect to the server?

Insert a rule into the iptables INPUT chain to allow inbound packets to access the noauth virtual host. Use the following command:

`iptables -I INPUT 3 -p tcp -i eth1 --dport 80 -j ACCEPT`

This command is inserting the rule on the INPUT chain at line 3. The rule matches on port 80 with the TCP protocol on the eth1 interface. If a packet matches, it is accepted. Verify the rule was inserted correctly by running:

`iptables -L -v --line-numbers`

8) Copy and paste the output of the INPUT chain.

Try connecting to the web server again with the same URL <http://www.i230.edu>. Your browser should connect this time. Now try the URL <http://www.i230.edu/cgi-bin/env.sh>.

9) Copy and paste the line displaying the port your web browser used to connect to the server (HINT: REMOTE_PORT)

Part 2 - Basic Authentication Web Site

Next you will enable the basicauth website with the command **a2ensite basicauth** and then reload the apache configuration as above with the **service apache2 reload** command.

Examine the configuration file `/etc/apache2/sites-enabled/basicauth.conf`. Apache basic authentication uses a password file with user names and passwords to authenticate users. In addition, it needs an authorization directive to say which of the authenticated users can access a resource. In this case, we are using a group file listing group names with a list of users separated by spaces. Finally we need a directive to specify which group(s) from the file are allowed to access a resource.

10) On which port is this web site listening?

11) What is the path to the passwords file?

12) What is the path to the security groups file?

13) What is the group authorized to access this website?

Again you will need to modify the firewall to allow access to this virtual host port. Use the `iptables` command to insert a rule. This command will be very similar to the one used above and only needs one change.

14) Write the `iptables` command you used to insert the rule for the basicauth virtual host web site.

Verify the rule was inserted correctly with the **`iptables -L -v --line-numbers`** command as above.

Access the website with the URL **`http://www.i230.edu:8080`**. You will be prompted for a userid and password. Use *vagrant* for both. You should be rewarded with a web page indicating you are authenticated.

Now go to the **`http://www.i230.edu:8080/cgi-bin/env.sh`**. This will dump all the values visible to a cgi program executing on the web server.

15) Copy and paste the line displaying the `REMOTE_USER`.

We will allow your favorite superhero to authenticate to the website as well. First you need to add the superhero's name and password to the password file using the **`htpasswd /etc/apache2/security/passwords your_superhero`**

You will be prompted for a password and to verify the password. Use a password of your choosing. (If you should forget it, you can use the same command above to overwrite the old password with a new one.)

Next you need to add your superhero to the security group for this website. Edit the security group file you identified above. You can use the **`nano`** command to edit the file. The usernames are separate by spaces on the line with the group name.

16) Copy and paste the contents of the group file.

Restart your browser and go back to the URL <http://www.i230.edu:8080/cgi-bin/env.sh>.

17) Copy and paste the line displaying the REMOTE_USER.

18) What is the inherent security flaw with this websites implementation? **(2 points)**

Fill in form 1 and click on the *Form 1 Continue* button.

19) Copy and paste the environment variable that contains just the elements and values submitted in the form.

20) What method was used to send this form to the server?

Hit the back link and fill in form 2. Click on the *Form 2 Continue* button.

21) What method was used to send this form to the server?

Part 3 - CAS Authentication

In this part you will enable the CAS_site and use the IU CAS authentication page. Use the **a2ensite CAS_site** command to enable the web site. Then reload the Apache server configuration with the same command as the last website.

Examine the configuration file for the website in */etc/apache2/sites-enabled/CAS_site.conf*.

22) Which port is this virtualhost listening to?

23) What group is authorized to access this website?

24) What directory does the scriptalias directive point to? What are the contents of that directory?

Add a rule to the firewall to open the port for this website. Again this is very similar to the first command you used to change the firewall.

25) Show the command used to open the firewall.

Go to the website **<http://www.i230.edu:8090/>** in your browser.

26) Were you able to access the website? Why not? (2 points)

Edit the group file */etc/apache2/security/groups* and your name to the group authorized to access this website. (Use the **nano** command.)

Reload the page on your browser. Were you successful this time? Now go to the URL **<http://www.i230.edu:8090/cgi-bin/env.sh>**.

27) Copy and paste the REMOTE_USER line from this page.

28) Copy and paste the line indicating the request method used to access this page.

29) Copy and paste the cookies sent with this request.

Part 4 - HTTPS

In this part we will enable a website that uses https protocol. Use the **a2ensite https** to enable the website. You will then need reload the Apache configuration as above.

Examine the https configuration file in `/etc/apache2/sites-enabled/https.conf`.

30) Which port does this website use?

31) What are the directives in this configuration file that enable SSL? (Hint: compare it to the `noauth.conf` and there are 3 directives.)

Open the firewall for this website's port using the `iptables` command like the previous websites.

Go to the URL <http://www.i230.edu:443/>

32) What error did you get? Why? (2 points)

33) What is the correct URL to access this website?

34) What message did you get this time? Why? How could you correct this problem? (2 points)

Your browser should give an option to proceed. Go ahead and accept the option. (For Chrome click on the ADVANCED link.)

Click on the link to show the environment variables.

35) What SSL protocol is the server using?

36) Comparing this website's configuration and the `basicauth` website configuration, what lines would you add to the `basicauth.conf` file to correct the security problem identified earlier?

37) What is the new URL that you would use with the corrected `basicauth` website?

38) Do you need to make a change to the firewall rules to secure the `basicauth` website? Why or why not?

Make the changes to the `/etc/apache2/sites-enabled/basicauth.conf` file. You will need to reload Apache's configuration after making the changes.

Restart close the windows to the lab websites and restart your browser. Go to the `basicauth` website's `env.sh` URL. You will need to modify the URL from the version used previously in Part 2.

39) Copy and paste from the resulting page the `SERVER_SIGNATURE` line along with the line immediately above it and the line immediately below it.

Extra Credit Questions

Answering the following questions can earn you extra points that can help your overall lab score for the class.

Looking at the permissions on the directory */etc/apache2/security* and thinking back to lab 3...

EXC-1) What could be done to improve the security of this directory? What security principle would you be following?

EXC-2) Considering that vagrant boxes usually have a userid of vagrant with the password vagrant, why is it a good idea for the Lab 6 VM to start with its default firewall settings?

EXC-3) What does the **a2ensite** command do to enable the website configurations? (HINT: you can look at man pages for the command and the command **ls -l** on the */etc/apache2/sites-enabled* directory can shed some light on things.)