SECURITY+ V4 LAB SERIES

Lab 15: Implementing Common Protocols and Services for Basic Security Practices

Document Version: 2024-01-29

Material in this Lab Aligns to the Following	
CompTIA Security+ (SY0-601) Exam Objectives	3.1: Given a scenario, implement secure protocols 3.3: Given a scenario, implement secure network designs
All-In-One CompTIA Security+ Sixth Edition ISBN-13: 978-1260464009 Chapters	17: Secure Protocols 19: Secure Network Design

Contents

Introduction

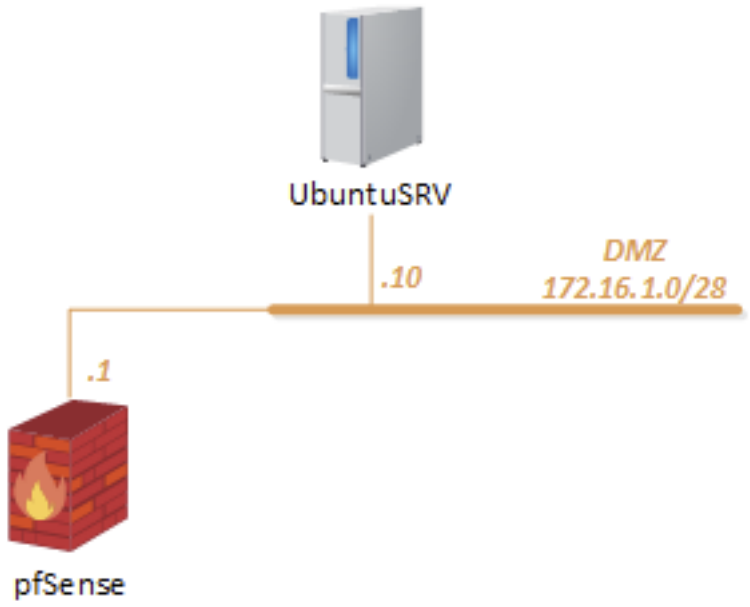
In this lab, you will be conducting network security practices by implementing common protocols.

Objective

In this lab, you will perform the following tasks:

- Configuring a Proxy server
- Configuring and Enabling SSL for HTTP Services

Lab Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

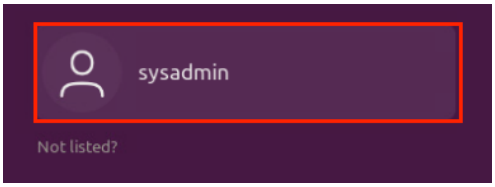
Virtual Machine	IP Address	Account (if needed)	Password (if needed)
pfSense	`192.168.0.1`	`sysadmin`	`NDGlabpass123!`
UbuntuSRV	`172.16.1.10`	`sysadmin`	`NDGlabpass123!`

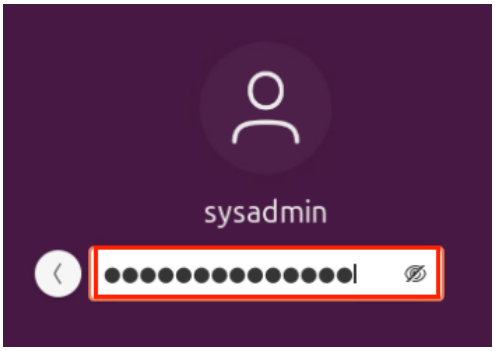
1 Protecting Sensitive Data

1.1 Load Lab Configuration

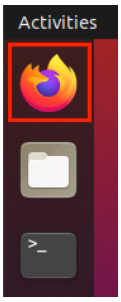
1. Click on the **UbuntuSRV** tab to access the *UbuntuSRV* VM.

2. Log in as username `sysadmin`, password `NDGlabpass123!`.





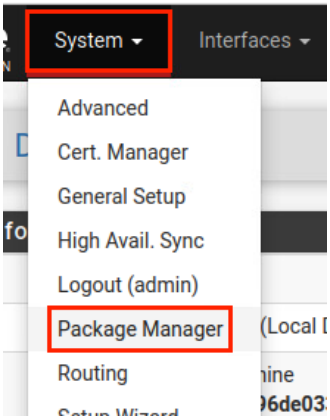
3. Open a web browser by clicking on the **Firefox** icon located in the left menu pane.



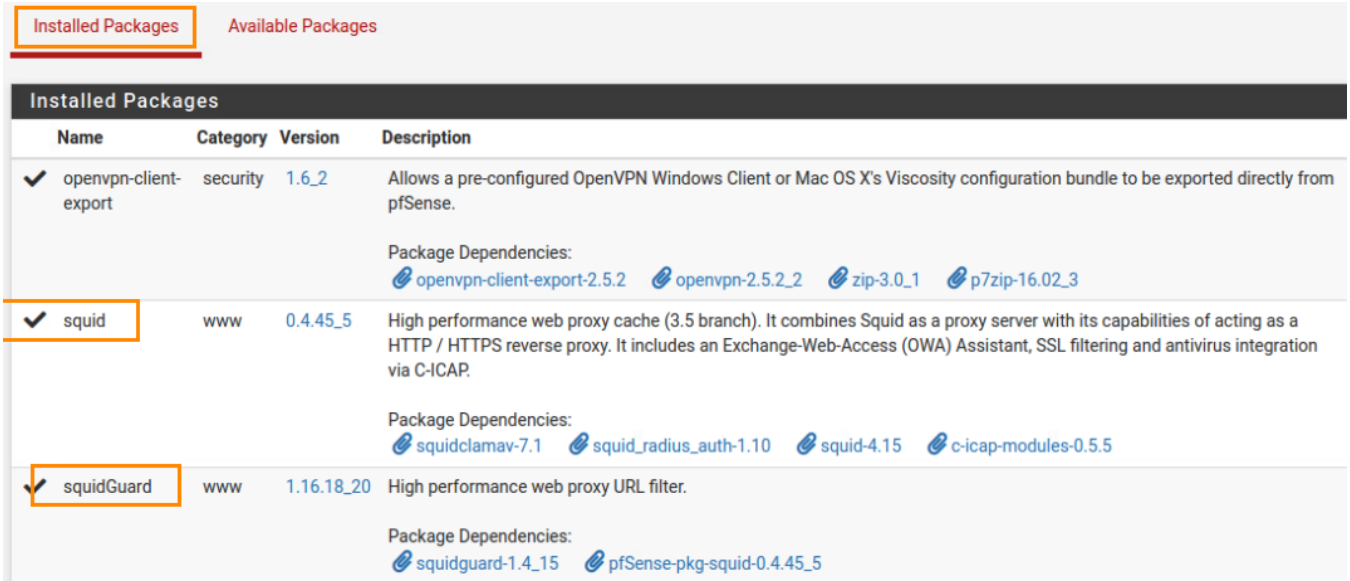
4. Within the *Firefox* web browser, type `172.16.1.1` into the *address bar,* followed by pressing **Enter.**

5. Once presented with the *pfSense* login page, type `sysadmin` as the *username* and `NDGlabpass123!` as the *password*. Click **Login.**

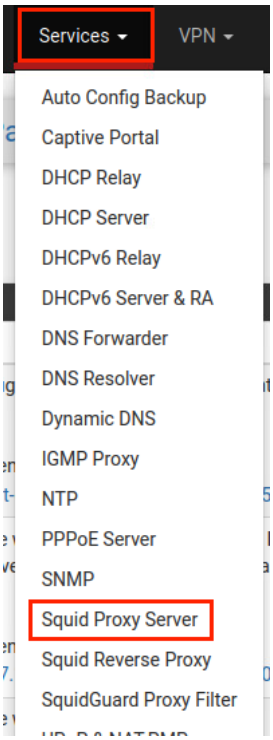
6. Once logged in, focus on the top menu pane and navigate to **System > Package Manager.**



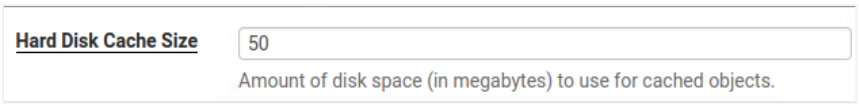
7. Make sure to view the **Installed Packages** tab. Verify that both the *squid* and *squidguard* packages are installed.



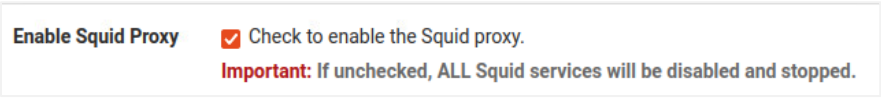
8. Once verified, navigate to **Services > Squid Proxy Server.**



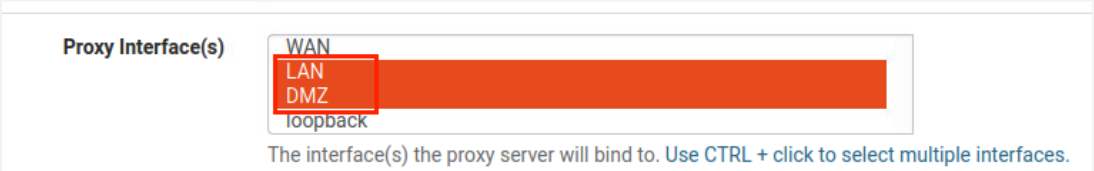
9. First, click on the **Local Cache**, then scroll down to find and set the *Hard Disk Cache Size* to `50`.



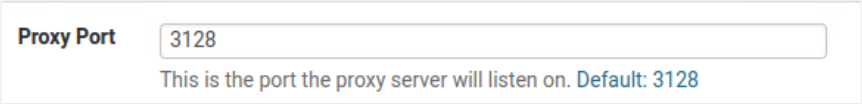
10. Scroll to the bottom of the page and click **Save.** The page will refresh and bring you back to the top after it is saved.
11. Then, click on the **General** tab and check the checkbox to **Enable the Squid Proxy.**



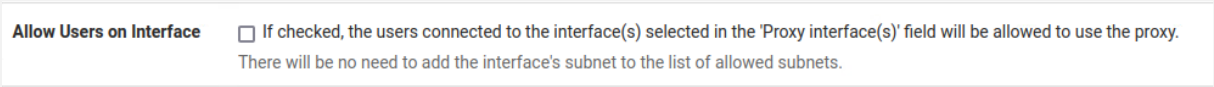
12. Select **LAN** and **DMZ** for the *Proxy interface*. To do so, hold the **CTRL** key and select each entry until both are highlighted.



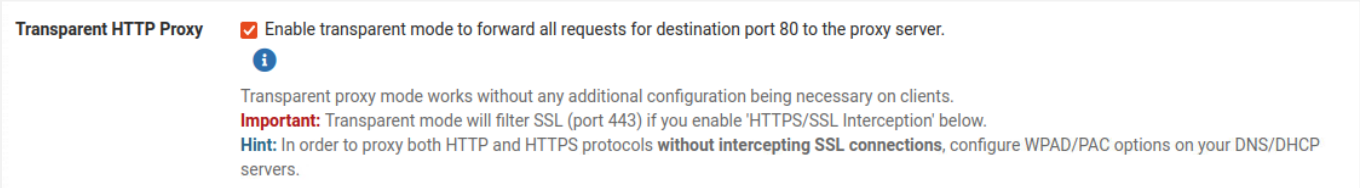
13. Use port number **3128** as the *Proxy port*.



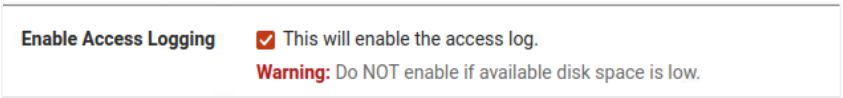
14. Uncheck the checkbox next to **Allow users on interface**.



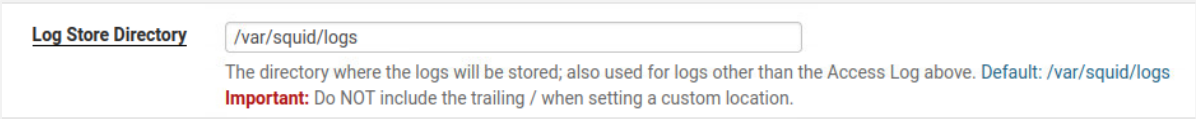
15. Check the checkbox next to **Transparent proxy** to enable this feature.



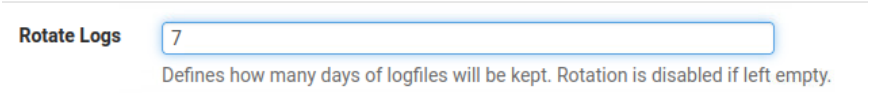
16. Scroll down until you see the **Enabled Access Logging** entry. Check the checkbox to enable.



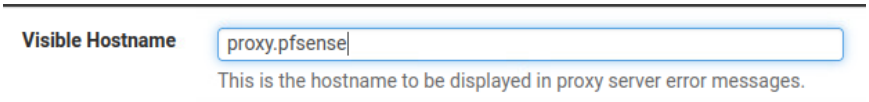
17. Verify that the *Log store directory* is configured to `/var/squid/logs`.



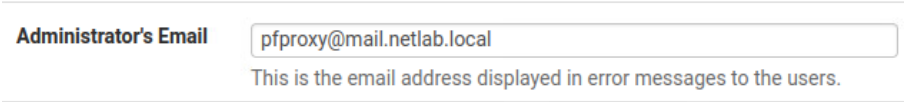
18. Type the number **7** as the value for the *Rotate Logs* field.



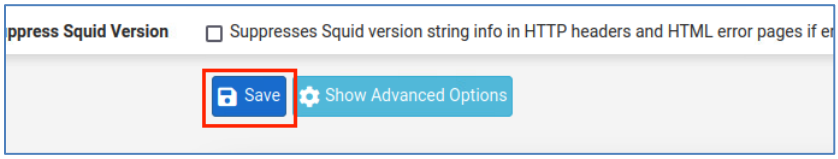
19. For *Visible Hostname*, type `proxy.pfsense`.



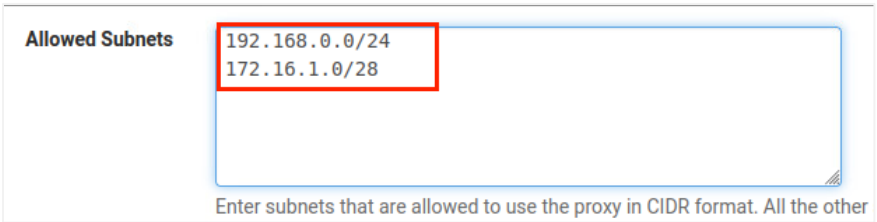
20. For the *Administrator Email*, type `pfproxy@mail.netlab.local`.



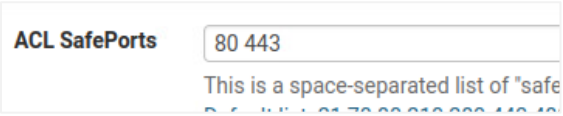
21. Scroll to the bottom of the page and click **Save.**



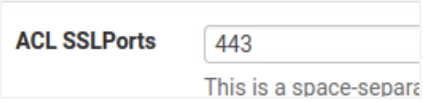
22. Next, click on the **ACLs** tab. Enter the subnets mentioned below into the *Allowed subnets* field.
    a. `192.168.0.0/24`
    b. `172.16.1.0/28`



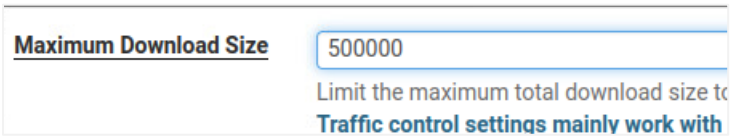
23. Scroll towards the bottom of the page until you see *ACLl Safeports*. Type `80` and `443` into the text field with a space inbetween.



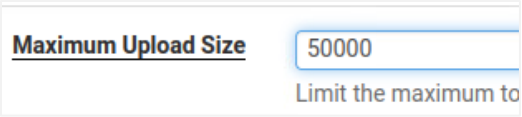
24. Type `443` for *ACL SSLlports.*



25. Click the **Save** button.
26. Click on the **Traffic Mgmt** tab. For *Maximum Download Size*, enter the value `500000` to represent *500MB* as the maximum download file size.



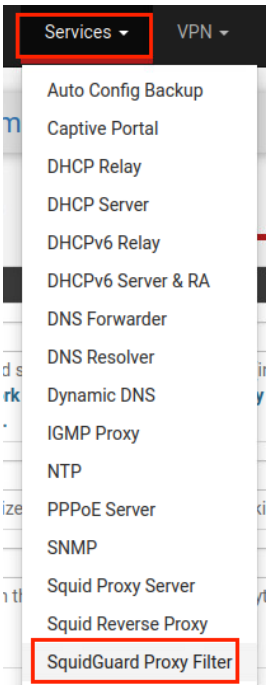
27. For *Maximum Upload Size*, enter the value `50000` to represent *50MB* as the maximum upload file size.



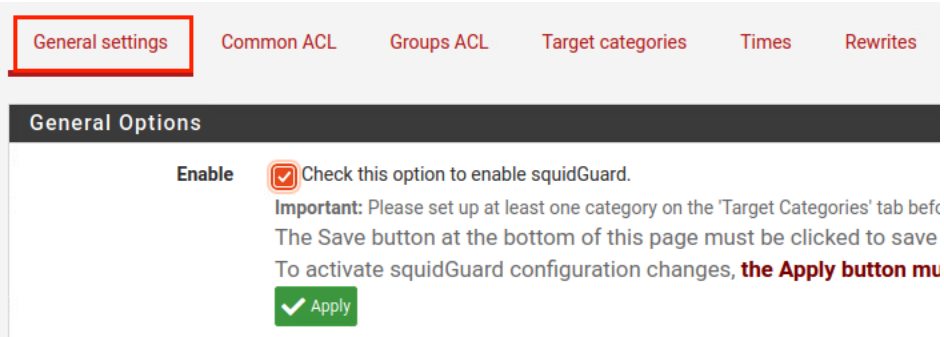
28. Scroll towards the bottom and click the **Save** button.

1.2 Configuring SquidGuard

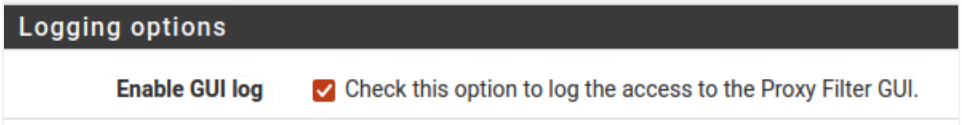
1. While on the *pfSense web configurator*, navigate to **Services > SquidGuard Proxy filter.**



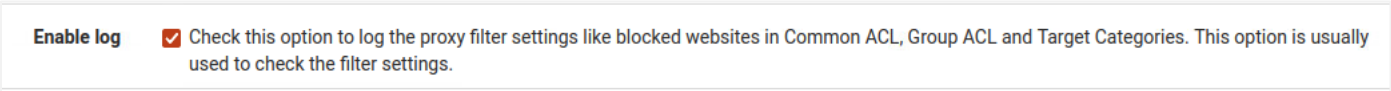
2. On the **General settings** tab, check the checkbox next to **Enable.**



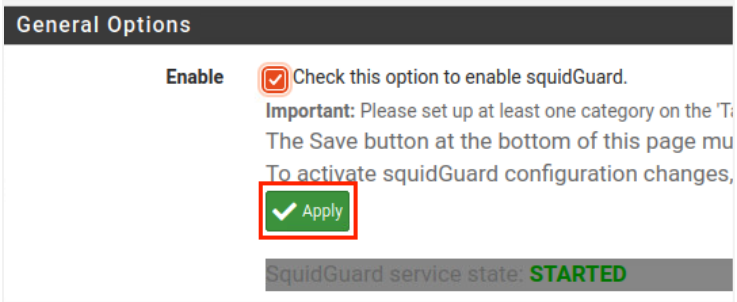
3. Scroll down until you see *Enable GUI log*. Check the checkbox to enable this feature.



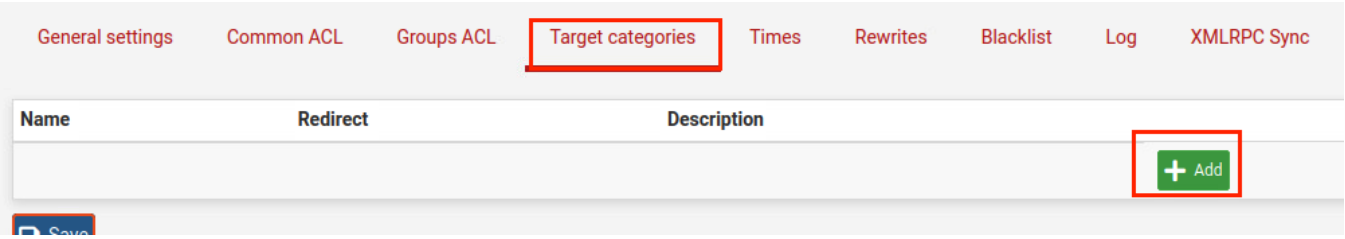
4. Check the box next to **Enable log**.



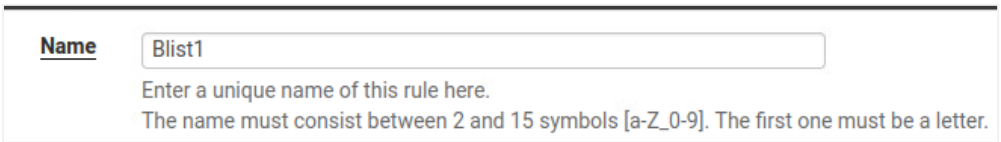
5. Scroll to the bottom of the page and click the **Save** button.
6. Once the page reloads, click the **Apply** button located towards the top of the page.



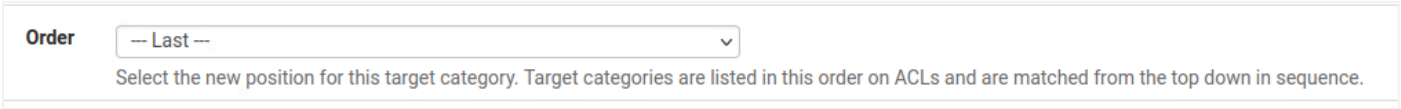
7. Next, click on the **Target categories** tab. Click the **Add a new item** icon.



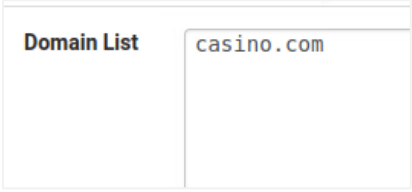
8. For the *Name*, type `Blist1`.



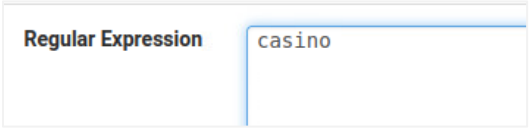
9. For *Order*, select the dropdown box and choose **--- Last ---.**



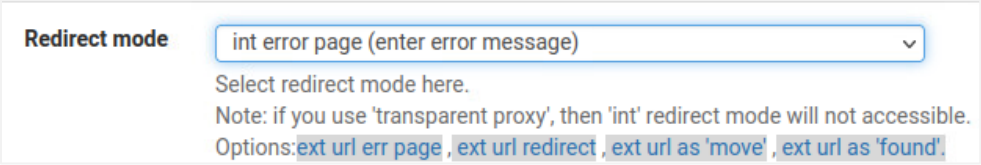
10. Type `casino.com` into the whitespace area for *Domain List.*



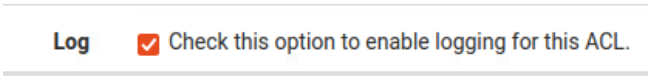
11. Type `casino` into the whitespace area next to *Regular Expression.*



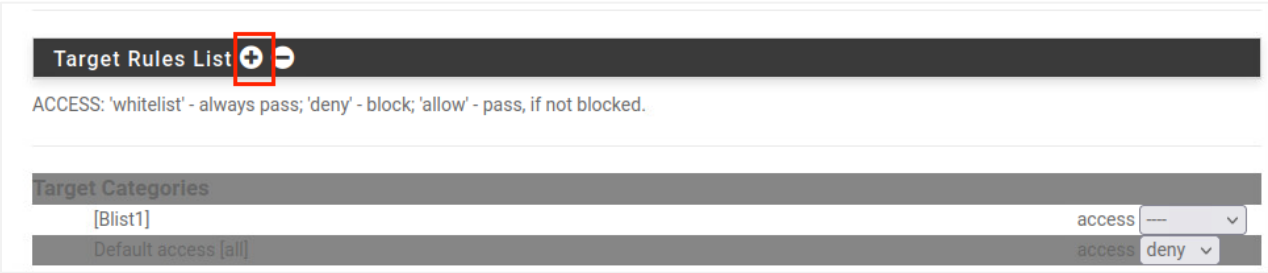
12. Select the dropdown box next to *Redirect mode* and choose **int error page (enter error message)**



13. Check the box next to the **Log** entry to enable logging for the ACL**.**

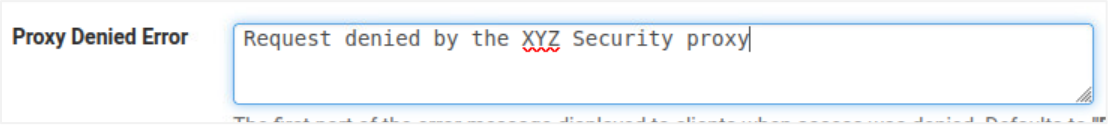


14. Click the **Save** button.
15. Click on the **Common ACL** tab. Click the **Show rules** icon within the *Target Rules List* pane.

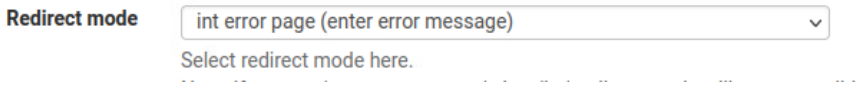


16. Notice *Blist1* is added to the list. For this entry, select the access dropdown box and choose **deny**. Click the dropdown box entry for *Default access [all]* and select **allow**.

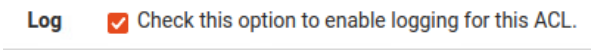
17. Within the whitespace area for the *Proxy Denied Error*, type `Request denied by the XYZ Security proxy.`



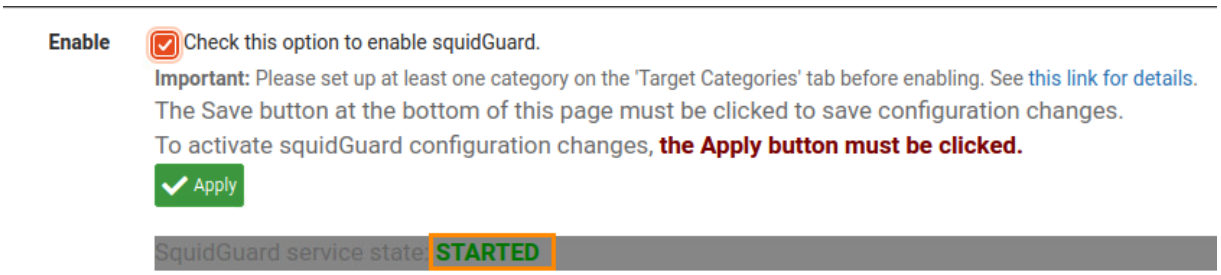
18. Select the dropdown box next to *Redirect mode* and choose **int error page (enter error message)**



19. **Check** the box next to *Log.*

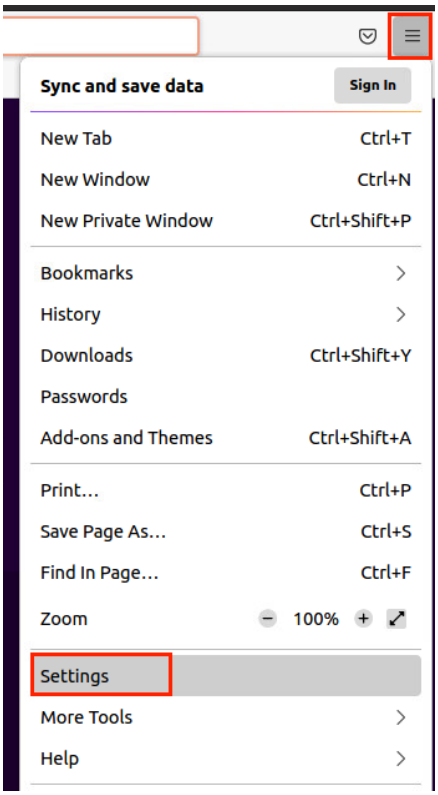


20. Click the **Save** button.
21. Once the page refreshes, click the **General settings** tab.
22. Scroll to the bottom and click the **Save** button.
23. To apply all configurations, click the **Apply** button.
24. Verify that the *SquidGuard* service state is *STARTED*.

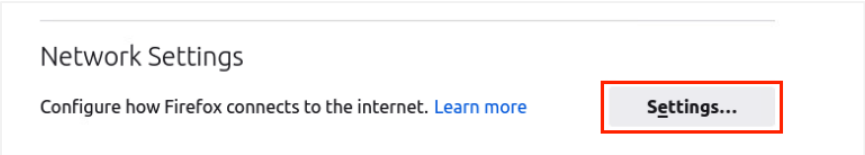


1.3 Configure & Test Firefox Proxy Settings

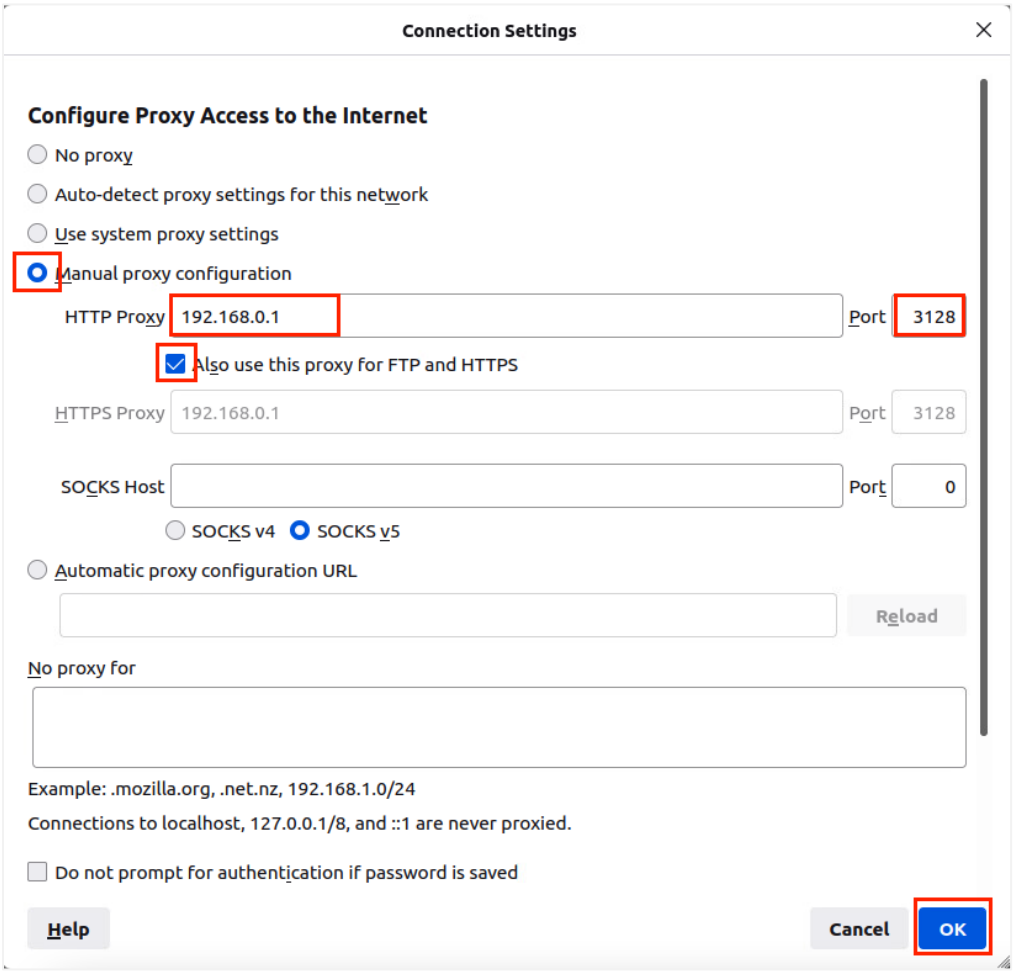
1. While on the *Firefox* web browser, click the **Application Menu** icon located in the top-right corner, followed by clicking on the **Settings** icon.



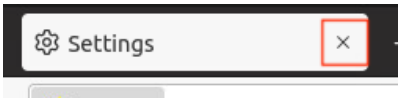
2. Scroll down to the bottom in the **Settings** tab and click on the **Settings…** button.



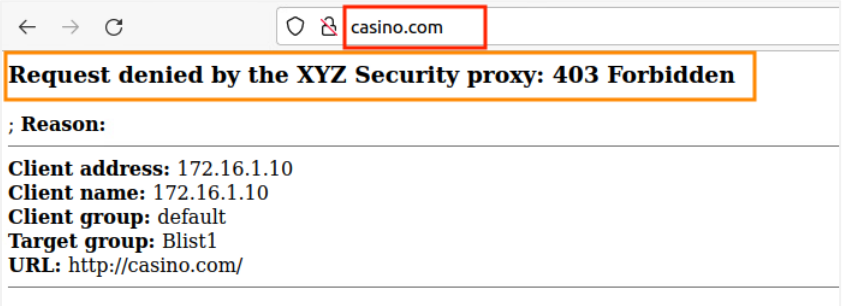
3. A pop-up window appears. Select the radio button for **Manual proxy configuration**. Type `192.168.0.1` as the *HTTP Proxy* and `3128` as the *Port*. Check the checkbox for **Also use this proxy for FTP and HTTPS**



4. Back on the *Firefox Preferences* window, close the **Settings** tab.



5. Open a new tab in Firefox by clicking the "**+**" icon located at the top. Type `casino.com` into the *address field* followed by pressing **Enter.** Notice the traffic will be dropped due to the rule we added.



6. **Close** the web browser
7. Leave the *UbuntuSRV* window open to continue with the next task.

2 Configuring and Enabling SSL for HTTP Services

2.1 Generating a Server Key and Server Certificate

1. While on the *Ubuntu* system, open a new terminal window by clicking on the **terminal** icon located on the left menu pane

2. Create a new directory by typing the command followed by pressing the **Enter** key.

```
sysadmin@ubuntusrv:~$ mkdir sslcerts
```

```
sysadmin@ubuntusrv:~$ mkdir sslcerts
```

3. Change to the newly made directory.

```
sysadmin@ubuntusrv:~$ cd sslcerts
```

4. Verify that *OpenSSL* is installed on the system.

```
sysadmin@ubuntusrv:~/sslcerts$ openssl version
OpenSSL 1.1.1f  31 Mar 2020
```

5. Type the following command to generate an *RSA server key*. When prompted for a *passphrase*, type `NDGlabpass123!` followed by pressing the **Enter** key. When prompted once more, type `NDGlabpass123!` again. Press **Enter**.

```
sysadmin@ubuntusrv:~$ openssl genrsa -des3 -out server.key 2048
```

```
sysadmin@ubuntusrv:~/sslcerts$ openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................................
....................+++++
....................................+++++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
```

6. Verify that the *server.key* has been generated.

```
sysadmin@ubuntusrv:~/sslcerts$ ls -l
total 4
-rw------- 1 sysadmin sysadmin 963 Aug  5 22:45 server.key
```


7. Generate the *Certificate Signing Request (CSR)* with the new **server.key.**

```
sysadmin@ubuntusrv:~$ openssl req -new -key server.key -out server.csr
```

   a. When prompted for the *server.key pass phrase*, type `NDGlabpass123!`. Press **Enter**.
   b. During the signing request process, a series of questions will be asked. Type the information given below for each step, followed by pressing **Enter**.
      i. *Country Name*: `US`
      ii. *State Name*: `TX`
      iii. *Locality Name*: `Austin`
      iv. *Organization Name*: `XYZ Security`
      v. *Organizational Unit Name*: Press **Enter**
      vi. *Common Name*: `ubuntusrv.netlab.local`
      vii. *Email*: Press **Enter**
      viii. *Challenge Password*: Press **Enter**
      ix. *Company Name*: Press **Enter**

```
sysadmin@ubuntusrv:~/sslcerts$ openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:TX
Locality Name (eg, city) []:Austin
Organization Name (eg, company) [Internet Widgits Pty Ltd]:XYZ Security
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:ubuntusrv.netlab.local
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

8. Once completed with the wizard, verify that *server.csr* has been created.

```
sysadmin@ubuntusrv:~/sslcerts$ ls -l
total 8
-rw-rw-r-- 1 sysadmin sysadmin 643 Aug  5 22:50 server.csr
-rw------- 1 sysadmin sysadmin 963 Aug  5 22:45 server.key
```

9. Sign the **server.csr** to create a **server.crt** file. When prompted for the *passphrase*, type `NDGlabpass123!` followed by pressing **Enter.**

```
sysadmin@ubuntusrv:~$ openssl x509 -req -days 365 -in server.csr -signkey
server.key -out server.crt
```

```
sysadmin@ubuntusrv:~/sslcerts$ openssl x509 -req -days 365 -in server.csr -signk
ey server.key -out server.crt
Signature ok
subject=C = US, ST = TX, L = Austin, O = XYZ Security, CN = ubuntusrv.netlab.loc
al
Getting Private key
Enter pass phrase for server.key:
```

10. Verify that the new *server.crt* has been created.

```
sysadmin@ubuntusrv:~/sslcerts$ ls -l
total 12
-rw-rw-r-- 1 sysadmin sysadmin 851 Aug  5 22:53 server.crt
-rw-rw-r-- 1 sysadmin sysadmin 643 Aug  5 22:50 server.csr
-rw------- 1 sysadmin sysadmin 963 Aug  5 22:45 server.key
```


11. View the contents of the newly created **server.crt** certificate.

```
sysadmin@ubuntusrv:~$ openssl x509 -in server.crt -noout -text
```

```
sysadmin@ubuntusrv:~/sslcerts$ openssl x509 -in server.crt -noout -text
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            65:24:d9:d7:a2:af:fc:f0:85:d9:1e:ef:a9:11:4e:b5:9d:25:44:2e
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = US, ST = TX, L = Austion, O = XYZ Seurity, CN = ubuntusrv.r
l
        Validity
            Not Before: Aug 26 19:52:10 2021 GMT
            Not After : Aug 26 19:52:10 2022 GMT
        Subject: C = US, ST = TX, L = Austion, O = XYZ Seurity, CN = ubuntusrv.
al
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:a5:42:cf:5f:43:58:1f:8d:ef:20:55:c9:fd:db:
                    b2:69:33:3a:89:3c:1d:e5:25:a1:44:41:be:8e:91:
                    c8:35:26:57:d0:c9:af:07:be:1c:f4:ad:1e:92:d8:
                    cb:58:5d:e0:06:6b:23:34:4c:b8:3a:87:f6:00:c3:
                    81:6e:d5:66:7a:72:a0:8a:54:0d:db:35:02:b4:ac:
                    75:e4:2d:51:e2:d0:4e:8d:00:6e:5a:c5:2e:13:19:
                    93:89:f0:1b:ed:cc:b7:91:ff:8a:1f:5f:61:20:a5:
                    44:fb:2b:4e:f1:ae:77:8b:54:2e:ca:45:a2:a0:d1:
                    da:b5:53:fa:b3:0c:4d:f7:c0:ee:91:4e:a0:46:57:
                    5b:ba:51:d8:af:92:f9:c2:c9:1e:fa:93:3f:5c:58:
                    51:2a:63:d1:73:d3:d3:4d:c2:07:72:00:82:00:eb:
                    ae:da:6b:0c:a6:9b:b1:91:f0:5e:53:bb:15:d2:86:
                    2a:5b:58:02:cb:00:61:57:67:26:fd:ea:bd:9b:9e:
                    d9:a1:d8:1d:26:9b:55:60:fc:f6:6b:51:f3:8d:e3:
                    0e:9e:44:66:c1:d7:f7:25:0d:8c:4c:bf:a2:fc:2a:
                    a5:2a:ad:24:1e:88:9e:a1:f3:96:96:68:41:2f:e7:
                    f6:75:de:bd:20:49:10:d3:b1:1d:75:0b:99:21:de:
                    07:b7
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
         29:dc:ad:b0:17:2b:e9:59:85:4f:79:0a:2d:23:e1:35:66:d8:
         c2:89:bb:1a:07:33:7c:26:33:82:79:b1:5b:18:99:0c:07:7b:
         41:2e:ea:ba:1e:d6:09:ab:47:d9:33:f9:d9:7d:b0:8d:58:4a:
         fe:71:6a:fa:c2:a7:5c:ed:e8:3a:30:e4:a8:a4:f8:14:fa:b4:
         0e:4c:a3:86:a8:75:d3:6d:8f:28:fe:33:69:dc:64:47:7b:92:
         bc:3b:e2:7d:4e:5d:e5:b4:14:2a:9a:b5:55:bf:fc:3d:2e:c1:
         aa:40:a0:17:f8:80:b8:f0:1d:6b:38:7f:a6:ba:82:8b:85:59:
         4f:03:e5:b6:71:63:9c:4d:c3:be:3a:46:db:b5:2f:1b:0d:72:
         60:66:eb:49:d6:0f:69:34:b5:98:9c:bd:9c:f9:cc:a4:64:3e:
         cc:24:1e:7e:3c:f2:79:50:9d:e0:9f:cd:ad:da:9b:be:41:1e:
         f3:e6:92:17:80:9a:79:e5:28:13:a1:61:77:21:75:04:19:ee:
         81:1f:4c:ad:36:ca:63:01:a1:24:61:83:da:52:29:11:d8:53:
         2b:f2:ab:29:d1:e9:b8:c5:b1:57:69:7c:fe:6e:f2:87:98:c0:
         e9:91:a6:ef:db:f4:a4:29:28:4b:db:2c:f8:fc:19:f1:aa:3b:
         cd:8a:c0:37
```

2.2 Configure Apache to Utilize SSL

1. Create a new directory that will act as a placeholder for the *SSL* objects. If prompted for a password, enter `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo mkdir /etc/apache2/ssl_certs
```

```
sysadmin@ubuntusrv:~/sslcerts$ sudo mkdir /etc/apache2/ssl_certs
[sudo] password for sysadmin:
```

2. While in the */sslcerts* directory, generate the same **server.key** but with no passphrase requirement. When prompted for a password for the *server.key* file, enter `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ openssl rsa -in server.key -out server.key.nopass
```

```
sysadmin@ubuntusrv:~/sslcerts$ openssl rsa -in server.key -out server.key.nopass
Enter pass phrase for server.key:
writing RSA key
```

3. List the current files in the directory. You should now have *four* different files.

```
sysadmin@ubuntusrv:~/sslcerts$ ls -l
total 16
-rw-rw-r-- 1 sysadmin sysadmin 851 Aug  5 22:53 server.crt
-rw-rw-r-- 1 sysadmin sysadmin 643 Aug  5 22:50 server.csr
-rw------- 1 sysadmin sysadmin 963 Aug  5 22:45 server.key
-rw------- 1 sysadmin sysadmin 887 Aug  5 23:12 server.key.nopass
```

4. Copy the **server.key.nopass** to the **/etc/apache2/ssl_certs** directory. If prompted for a password, enter `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo cp server.key.nopass /etc/apache2/ssl_certs
```

```
sysadmin@ubuntusrv:~/sslcerts$ sudo cp server.key.nopass /etc/apache2/ssl_certs/
```

5. Copy the **server.crt** file to the **/etc/apache2/ssl_certs** directory. If prompted for a password, enter `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo cp server.crt /etc/apache2/ssl_certs
```

```
sysadmin@ubuntusrv:~/sslcerts$ sudo cp server.crt /etc/apache2/ssl_certs/
```

6. Change to the **/etc/apache2/ssl_certs** directory.

```
sysadmin@ubuntusrv:~/sslcerts$ cd /etc/apache2/ssl_certs/
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$
```

7. Verify that two files are present in the directory.

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ ls -l
total 8
-rw-r--r-- 1 root root 851 Aug  5 23:16 server.crt
-rw------- 1 root root 887 Aug  5 23:14 server.key.nopass
```

8. Rename the **server.key.nopass** file to **server.key**. If prompted for a password, enter `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo mv server.key.nopass server.key
```

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo mv server.key.nopass server.key
```

9. We already have an nginx server running, but since it is used for regular service, we will use the *Apache* web service for this lab instead. First, type `sudo service nginx stop` to disable the nginx server. If prompted for a password, type `NDGlabpass123!`.

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo service nginx stop
```

10. Then type `sudo service apache2 start` to start the *Apache* service.

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo service apache2 start
```

11. Initiate the *a2enmod* module for *SSL*. Then, restart the apache2 service. If prompted for a password, type `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo a2enmod ssl
sysadmin@ubuntusrv:~$ sudo service apache2 restart
```

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo service apache2 restart
```

12. Create a new symbolic link to the *default-ssl.conf* file. If prompted for a password, type `NDGlabpass123!`. If you receive an error stating that the file already exists, remove the old *000-default.conf* file using the **rm** command and enter the **ln** command below again.

```
sysadmin@ubuntusrv:~$ sudo ln -s /etc/apache2/sites-available/default-ssl.conf
/etc/apache2/sites-enabled/000-default-ssl.conf
```

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo ln -s /etc/apache2/sites-available/
default-ssl.conf /etc/apache2/sites-enabled/000-default-ssl.conf
```

13. Verify that a *Virtual Host* is configured in the default *sites-availabl*e file. Type the command below to open the file with the **nano** text editor. If prompted for a password, type `NDGlabpass123!` .

```
sysadmin@ubuntusrv:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

14. When in the *nano* editor, confirm that the **ServerName** is set to **172.16.1.10:80**. Then check other settings, if you see any missing entries, enter the same as shown in the screenshot below. Once finished, press **CTRL+S** to save, and **CTRL+X** to exit.

```
  GNU nano 4.8        /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
        ServerName 172.16.1.10:80
        ServerAdmin webmaster@ubuntusrv.netlab.local
        DocumentRoot /var/www/html

        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/html>
                Options Indexes FollowSymLinks Multiviews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

15. Next, edit the contents of the *default-ssl* file. Type the command below followed by pressing **Enter**. If prompted for a password, type `NDGlabpass123!`.

```
sysadmin@ubuntusrv:~$ sudo nano /etc/apache2/sites-available/default-ssl.conf
```


16. Add the missing information as you did before. Use the **arrow keys** to position the cursor.

```
  GNU nano 4.8        /etc/apache2/sites-available/default-ssl.conf
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerName 172.16.1.10:443
                ServerAdmin webmaster@ubuntusrv.netlab.local

                DocumentRoot /var/www_ssl

                <Directory />
                        Options FollowSymLinks
                        AllowOverride None
                </Directory>
                <Directory /var/www_ssl>
                        Options Indexes FollowSymLinks Multiviews
                        AllowOverride None
                        Order allow,deny
                        Allow from all
                        Require all granted
                </Directory>

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                SSLEngine on

                SSLCertificateFile      /etc/apache2/ssl_certs/server.crt
                SSLCertificateKeyFile /etc/apache2/ssl_certs/server.key

                <FilesMatch "\.(cgi|shtml|phtml|php)$">
                                SSLOptions +StdEnvVars
                </FilesMatch>
                <Directory /usr/lib/cgi-bin>
                                SSLOptions +StdEnvVars
                </Directory>
        </VirtualHost>
</IfModule>
```

17. Press **CTRL + S** to save, and **CTRL+X** to exit.
18. Create a new directory.

```
sysadmin@ubuntusrv:~$ sudo mkdir /var/www_ssl
```

```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ sudo mkdir /var/www_ssl
```

19. Leave the *Terminal* open for the next section.

2.3 Configuring & Testing HTTPS Test Page

1. While on the *Terminal*, navigate to the **/var/www_ssl** directory.

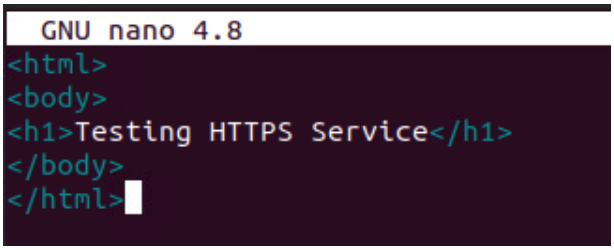
```
sysadmin@ubuntusrv:/etc/apache2/ssl_certs$ cd /var/www_ssl/
```

2. Create a new **index.html** file. If prompted for a password, enter `NDGlabpass123!`

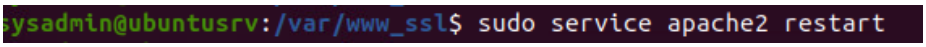
```
sysadmin@ubuntusrv:/var/www_ssl$ sudo nano index.html
```


3. Within the *nano* text editor, type the *HTML* code below.

```
<html>
<body>
<h1>Testing HTTPS Service</h1>
</body>
</html>
```

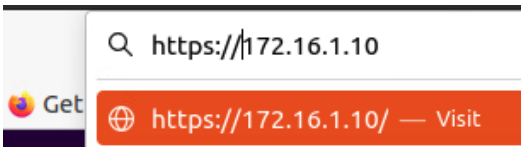


4. Press **CTRL + S** to save, and **CTRL+X** to exit.
5. Restart the *Apache* web service to apply all the configuration changes made. If prompted for a password, enter `NDGlabpass123!`.

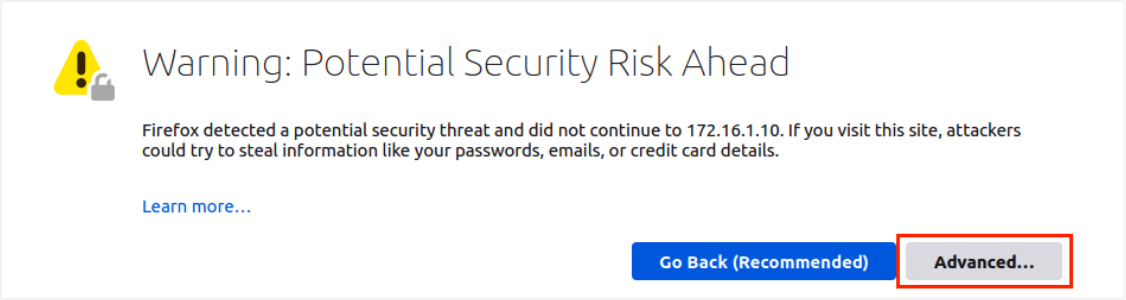


6. Open a new *Firefox* web browser by clicking the **Firefox** icon located on the left menu pane.

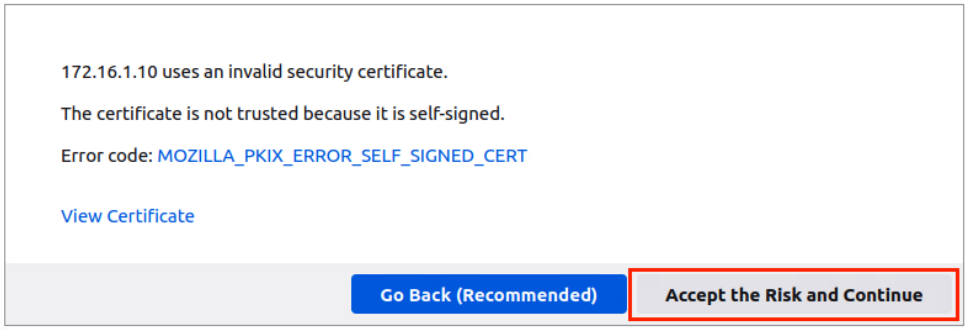
7. Within the *address bar*, type `https://172.16.1.10`. Press **Enter.**



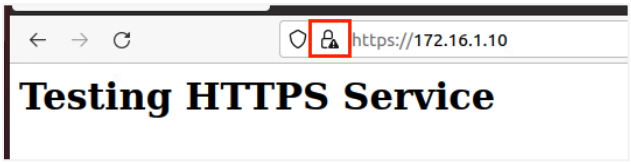
8. When presented with the *Warning* page, click on **Advanced….**



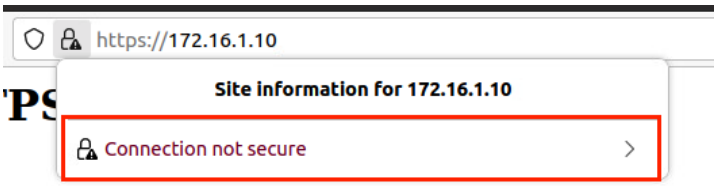
9. When expanded, click on the **Accept the Risk and Continue** button.

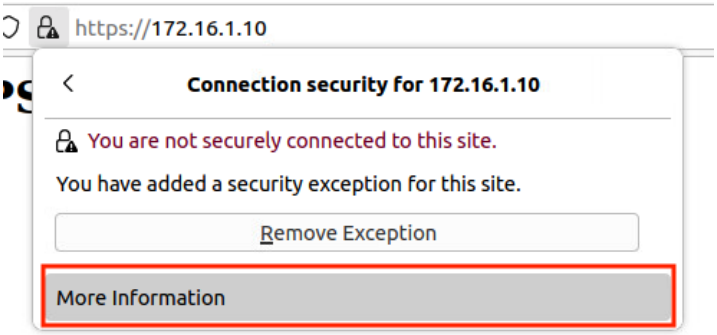


10. Notice the web page with *Testing HTTPS Service* opens. To view the contents of the *server certificate*, click the **lock** icon located to the left of the URL.

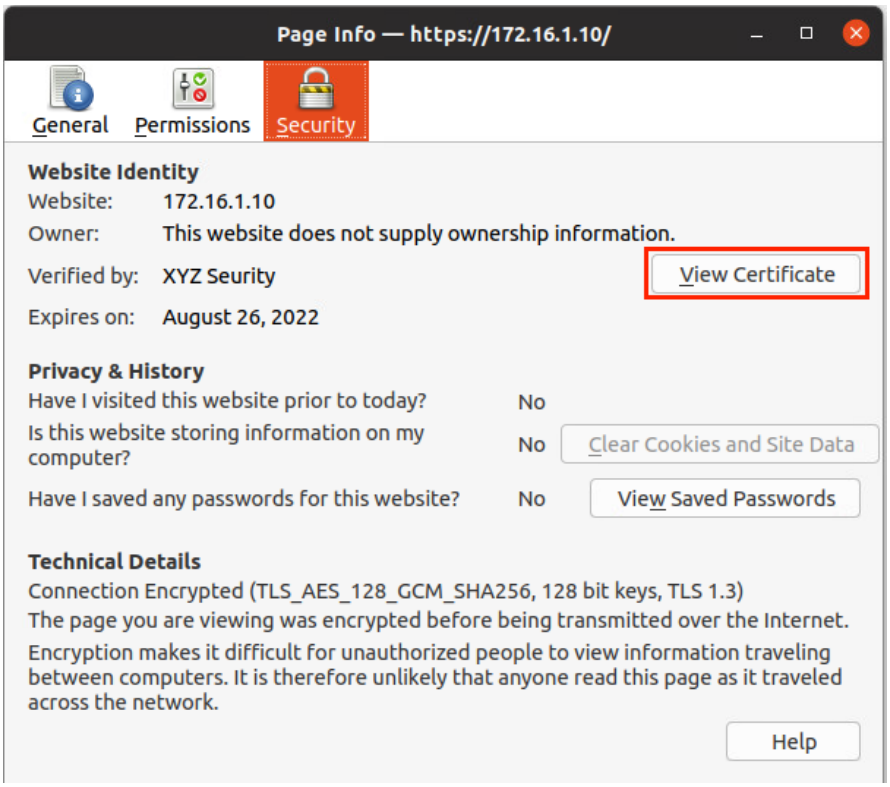


11. A small window will appear. Click on *Connection not secure*, then click *More Information*.

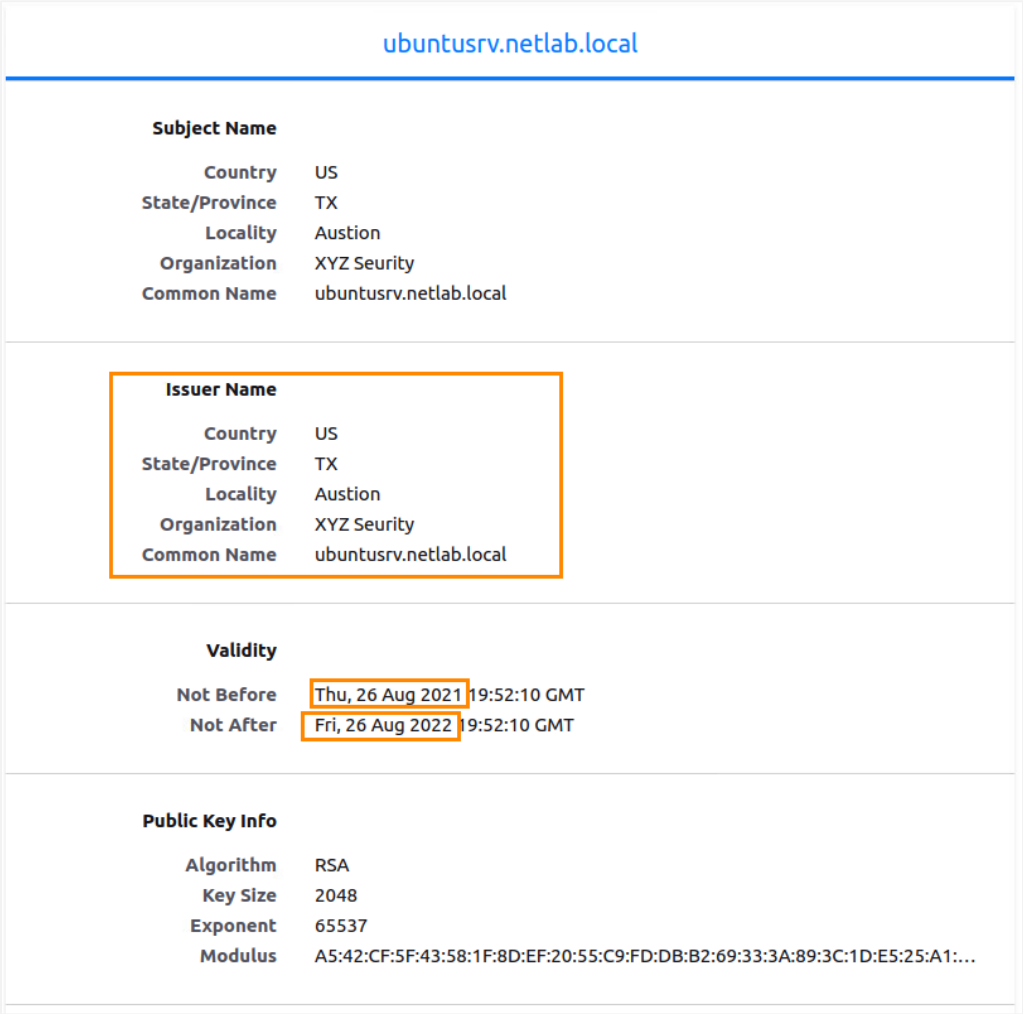




12. On the *Page info* screen, notice the *Website Identity* information. Click the **View Certificate** button.



13. On the *Certificate Viewer* window, notice the entries for *Issuer Name* and the period of *Validity*. All values are reflective of the contingencies set when the self-sign of the certificate took place at the beginning of *Task 2.1.*



14. The lab is now complete; you may end the reservation.

