

# **Configuration and Illustration of Network Security Principles on a Cloud Testbed**

**By**

**OJUAWO O.O**

**DEPARTMENT OF COMPUTER SCIENCE, FEDERAL POLYTECHNIC, P.M.B. 50,  
ILARO, OGUN STATE, NIGERIA**

**E-MAIL: [olutayo.ojuawo@federalpolyilaro.edu.ng](mailto:olutayo.ojuawo@federalpolyilaro.edu.ng)**

**PHONE NO: +23408038417080**

## **ABSTRACT**

*Over the years, a lot of internet users had been victims of telemarketing, identity theft, transaction fraud, piracy and hacking due to the unsecured website and little or no knowledge about cybercrime, the vulnerability of the system and unsecured websites. Cybercrime has been a tool for privacy violation whereby an individual or internet user loses his private information to attackers through stalking, fraud, identity theft, or scam. Network security is a vital viewpoint in system networking and a major instrument in curbing cybercrimes most especially online transaction fraud. A secured website must have its information encrypted so that information cannot be accessed by unauthorized users and authentication must be ensured to improve computer user's trust. Most universities do find it difficult in teaching and illustrating the process of setting up a Secure Socket Layer or how to generate a self-signed certificate to the students in the laboratory. Educators find it hard in setting up a network security workshop for students to have knowledge about the real-life network and information attacks and implementation of network security principles. This paper, however, discusses the overview of network security and the public key infrastructure (PKI) and illustrates the process or steps taken in setting up a Secure Socket Layer (SSL) certificate for a localhost website, running on Apache web server by demonstrating this process on a large scale network and cloud testbed called Deterlab*

*Keywords: Network security, PKI, Deterlab, cybersecurity, SSL,*

## 1.0 INTRODUCTION

It is very difficult to actually understand how security is administered and it is more difficult for a student to understand and try it themselves because it requires so many protocols. So in this project, the various steps of managing and configuring some security protocols in a virtual environment called Deterlab will be illustrated and review the technical solution and show practical steps with the example of Deterlab.

A secured website has so many benefits ranging from the owner's point of view to the user's point of view. From the website owner's point of view, a secure website builds website authenticity. A secure communication is established between the server and browsers. A secure website also increases the website owner's return of investment. For a website to be secure, it has to be protected from malware and must be less vulnerable as some SSL certificate performs vulnerability and malware scanning which is a great benefit to the owner of the website. A website with an RSA encryption of 2048 bits and 256 bits algorithm is a well-secured website which makes sensitive information secured and unavailable to the third party. On the internet user's point of view, a secure website ensures that information of the user is protected also phishing websites are avoided. The user's information is also secured with top security.

In regards to network security, a website with SSL certificate is assured of information encryption. It is impossible for attackers to decipher without the right encryption key due to SSL certificate installation on the website which prevents attackers from having access to user's information. In an online trading and online business, encrypted information prevents identity theft and loss of credit card information of the user.

Network security is an imperative perspective in system networking. Network security is a strategy taken by an organization to ensure that its assets and all networks are secured. Network security manages all perspectives identified with the protection of sensitive information resources existing on the network. Different mechanisms are covered by network security in the provision of efficient security services when transferring data across networks. Network security utilizes security hardware and software. In an institute of learning, the teaching of network and information security method is essential but difficult due to the facilities required in performing practical examples of cyber security practices. Hence, the introduction of deterlab ensures that cyber security technology can be tested at this facility [3]. Deterlab is a testbed that can be used by the public to study the effect of network security and cyber security controls [8]. The aim of this project is to comprehend the general network practices by illustrating and configuring the Secure Socket Layer (SSL) on a localhost Apache web server and actualizing the security activity in a pedagogical tool called Deterlab virtual environment.

## 2.0 LITERATURE REVIEW

### 2.1 Network Security

Security has been a major concern for protecting communication between terminals in a hostile domain [7]. The fast rate of development internet and information technology gave rise to the government, schools, military, companies and individuals in joining the internet which also gave rise to the increase in illegal users to attack and decimate the network by utilizing a fake website, introduction of virus, Trojan horse and a lot more [1]. A communication channel should not be vulnerable to attack when transmitting data because an attacker can target the communication channel to obtain, decrypt and alter data [6]. A secure network denotes computer security and data security.

Network Security comprises of technologies and procedures that are developed to form protection over internal networks and external threats. The main goal of network security is to give controls along the network border which permit access to the internal network and allow traffic pass if the traffic is valid, authorized and of limited risk [1]. The characteristics of Network security ranges from data integrity, data availability, data confidentiality, and data controllability while the threats to network security are human error, unauthorized access, malicious attacks and loopholes of networking software [1].

In network and information security system, the requirements that users requests for are Confidentiality, Integrity, and Availability, which forms foundations of any well-composed network security. This is shown in figure 1 below.

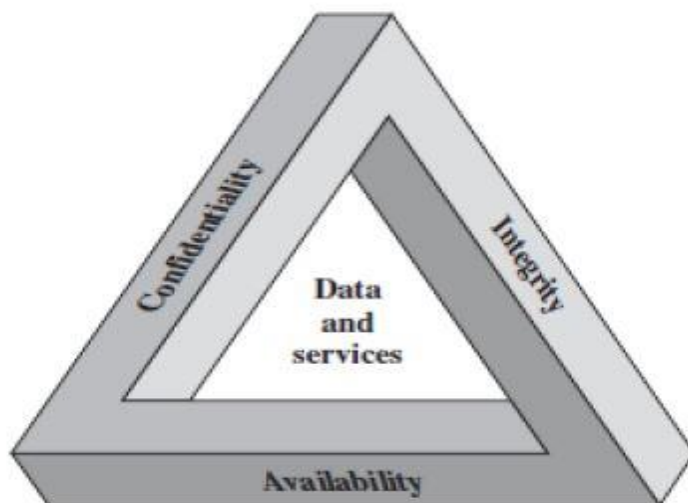


Fig 1: Security Goals

The three components are described below [10]:

- **Confidentiality:** This is the provision of restrictions on accessing and disclosing information, implying means ensuring personal privacy and restrictive information are protected. Confidential information can only be accessed by authorized user. When confidentiality is breached, information can be disclosed to an unauthorized user. Confidentiality are of two concepts ; data confidentiality, which guarantees that private or secret data is most certainly not made accessible or revealed to unapproved people, and privacy, which assures that the kind of information related to an individual is controlled or influenced by them and may be received and stored by who the information is disclosed.
- **Integrity:** This guard against improper alteration or destruction of information by ensuring authenticity and non-repudiation of information. A loss of integrity is when unauthorized personnel modifies or destroy of information. Integrity covers three concepts: data integrity, which ensures that information are altered only in an authorized and predefined manner of an unauthorized user is not modified by unauthorized personnel; non-repudiation, which ensures that a sent message cannot be denied by the sender, and origin integrity, which ensures that the origin of a message can be verified.
- **Availability:** This ensures that authorized personnel can have legitimate access to information at the right time. A loss of availability depicts the disruption in information access or use of information. For example, students must have access to information on their university website at any time.

## **2.2 PUBLIC KEY INFRASTRUCTURE (PKI)**

The term Public Key Infrastructure (PKI) is utilized to depict the procedures, advancements, and practices that are required to give a secure infrastructure [2]. PKI is a system that comprises of software, hardware, procedures, and policies required in management, creation, storage and distribution of keys and digital certificates. PKI additionally gives methodology in generation, distribution, and utilization of certificates and keys. The PKI gives a mechanism in the publication of public keys that are a piece of public key cryptography. It depicts the standards software and policies that are utilized to manage certificates private and public keys. PKI is a system that comprises of security policies, encryption mechanisms, and applications that produce, store, and oversee keys [4]. PKI gives the way in setting up of establishing trust by creating a bond between public keys and identities, giving a sensible affirmation that secure communication is established when communicating with the user you think you are [12]. A PKI is expected to provide the following [2]:

- Confidentiality: This is described as a secure transmission of information across networks guaranteeing that it is inaccessible to unauthorized individuals. PKI utilizes encryption algorithms which ensure confidentiality.
- Authentication: This can be described as distinguishing proof offered by PKI through digital certificates. Authentication can also be called Origin Integrity.
- Access Control: This is to guarantee that exclusive individuals with the required security benefits are permitted access to data or information. PKI guarantees access control through the pair of Public and private keys.
- Non-Repudiation: The premise of this is that the sender can't deny sending any data or information at a later time. Non-repudiation guarantees that there is a dependable method for guaranteeing responsibility for the sent electronic information. PKI ensures this through digital signature.
- Data Integrity: This concept describes how Data ought not to be changed or altered at all while being sent across the network. Data integrity is guaranteed by message hashing.

The main goal of a PKI is to enable a secure communication among users that are unknown to each other [5]. The PKI relies on a trusted third party (TTP) called Certification authority (also known as Certificate Authority) who is in charge of certificate issuance that is trusted by users [9]. PKI requires various functions to perform keeping in mind the end goal to provide security and trust to electronic communication [4]. The functions are:

- Public and private key pair generation for digital signature authentication and creation.
- Controlling access to the private key by providing authentication.
- Creation and issuance of certificates to achieve users' authentication.
- Registration of new users to validate them.
- Maintenance of key history for future references.
- Revoking invalid certificates.
- The need to update and recover keys if key compromise should arise.

## **2.1 Components Of A PKI**

As discussed above, PKI is a structure that comprises of policies, hardware, techniques, and software for keys and certificates management. For the functionality of the framework, various components are required. Each component has a particular role to perform. The components required are [4]:

- Certification Authority
- Registration Authority

- Public Key Infrastructure (PKI) clients
- Digital certificates
- Repository or Certificate Distribution System (CDS).

## **3.0 DESIGN AND IMPLEMENTATION OF DETERLAB EXPERIMENT**

### **3.1 Self-Signed SSL Installation**

The goal of this paper is to set up an SSL certificate on apache on a web server installed in Deterlab testbed environment. The basic steps were taken from creating an experiment in Deterlab, starting an experiment in Deterlab, accessing Deterlab virtual node, installing apache on Deterlab through the console, and setting SSL certificate on Apache installed in Deterlab. An additional exercise, Cross-Site Scripting (XSS) vulnerability exploitation, was demonstrated.

The network simulator is discrete event packet-level simulator that covers a substantial number of application of various protocols of various types of a network comprising of various network components and activity models [11]. Network simulator is a bundle of tools that simulates the network behavior such as creating network topologies, analyze events and comprehend the network.

In Deterlab, the essential platforms for running Network Simulator (NS) are Windows 95/98/NT/2000/XP, Unix and Unix-like systems, Free BSD, Linux (Use Fedora or Ubuntu versions) and SunOS/Solaris. A Network Simulator (NS) file is needed to design the topology of the node. The Network simulator file designed for this experiment is the Linux (Ubuntu Version) having one node.

#### **3.1.1 Creating an Experiment**

To create an experiment, log in with your username and password on the Deterlab log in page.

1. Go to <http://isi.deterlab.net> and log in using your username and password. This should take you to the “My DETERLAB” page.
2. Download the file `deterlab-linux-intro.ns` from study space. This is an “input file” which specifies virtual host machines on the DeterLab cluster. Once implemented you will be able to access these as though they were real computers.
3. On the top panel of the My DETERLab page, select “Experimentation → Begin an Experiment”. An input form box should appear:
  - a. For “Select Project” select NISVLTE.

- b. For “Name” add your own name (e.g. “Olutayo”) or any other name
- c. For “Description” add a short description (e.g. “ Security Experiment”).
- d. For “NS file”, upload the deterlab-linux-intro.ns file.
- e. Set “Idle-Swap” to 2 hours.
- f. Set “Max. Duration” to 8 hours.
- g. Click “Submit”.

After the submission, Deterlab will process the details you submitted and this can take up to ten minutes maximum. An email will be sent to you indicating the successful creation of the experiment. The message identifies the number of virtual host in the experiment, the operating system to be used and the hostname.

### 3.1.2 Starting the Experiment

The created experiment is yet to be swapped in, which makes it inactive. To establish this:

1. Click on My Deterlab and find the table “*Current Experiment*”. The table has one entry with the PID (Project ID), and EID (Experiment ID) having NISVLTE, Your Experiment Name and Swapped respectively. The Nodes has “1” entry while Description has “Security Experiment”
2. To enter the experiment’s web page, click on your name in the EID field. On the left, see a list of “Experiment Options” will be displayed. Select “Swap Experiment In” at the left corner of the page. This is shown in figure 2 below.

161 Free PCs, 6 reloading									
bpc2800	10	dl380q3	2	bpc2133	44	sm	0	pc2133n	9
pc3000	3	bpc3000	25	smX10	0	pc3060	2	bpc3060	21
pc2133	35	MicroCloud	0	bvx2200	20	dl360q8-6p	0	pc2133x	1

Fig.2 List of Experiment Options.

3. When prompted, select “Confirm”. This process can take between two to ten minutes. After a while you will see a message telling you that the experiment has been “swapped in”, meaning that it is now active. You will receive an email notifying you that the experiment has to be swapped in.

4. Click on “My DETERlab” tab to verify that the status of the experiment has turned to “Active”.

### 3.1.3 Accessing the Virtual Node

The experimental node can be accessed over the internet as if it is a real machine. This makes it virtual.

To access the node, a lightweight SSH client for windows called PuTTY will be used.

PuTTY is used to create SSH connection to the virtual node. To establish this, the following steps should be taken

1. Open PuTTY, and type “users.isi.deterlab.net” into the Host Name field and click open. (This connects you to the DeterLab control server.). This is shown in figure 3 below.

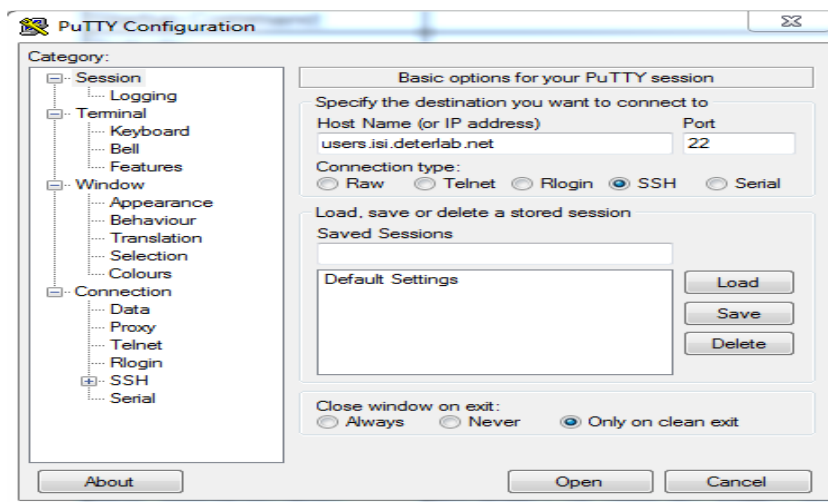


Fig. 3: Putty with the Host Name input

2. A web interface will open where the user will be required to enter the username and password. The console or terminal is where all files will be created using Ubuntu commands. This is shown in figure 4 below.

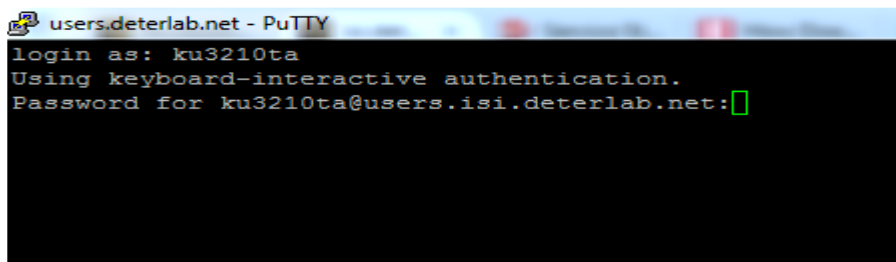
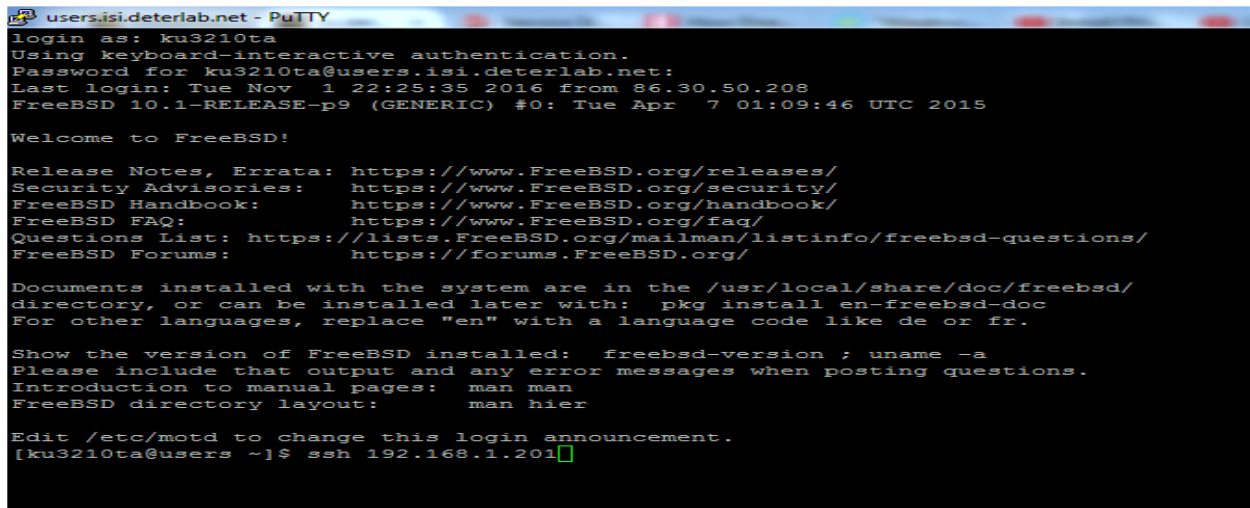


Fig .4: PuTTY Log in



3. After typing the login details, press enter. An SSH connection from the control server to the experimental node is required. To do this, type ssh and the IP address of the node as shown in figure 5 below.



```
users.isi.deterlab.net - PuTTY
login as: ku3210ta
Using keyboard-interactive authentication.
Password for ku3210ta@users.isi.deterlab.net:
Last login: Tue Nov  1 22:25:35 2016 from 86.30.50.208
FreeBSD 10.1-RELEASE-p9 (GENERIC) #0: Tue Apr  7 01:09:46 UTC 2015

Welcome to FreeBSD!

Release Notes, Errata:  https://www.FreeBSD.org/releases/
Security Advisories:   https://www.FreeBSD.org/security/
FreeBSD Handbook:      https://www.FreeBSD.org/handbook/
FreeBSD FAQ:           https://www.FreeBSD.org/faq/
Questions List:         https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:         https://forums.FreeBSD.org/

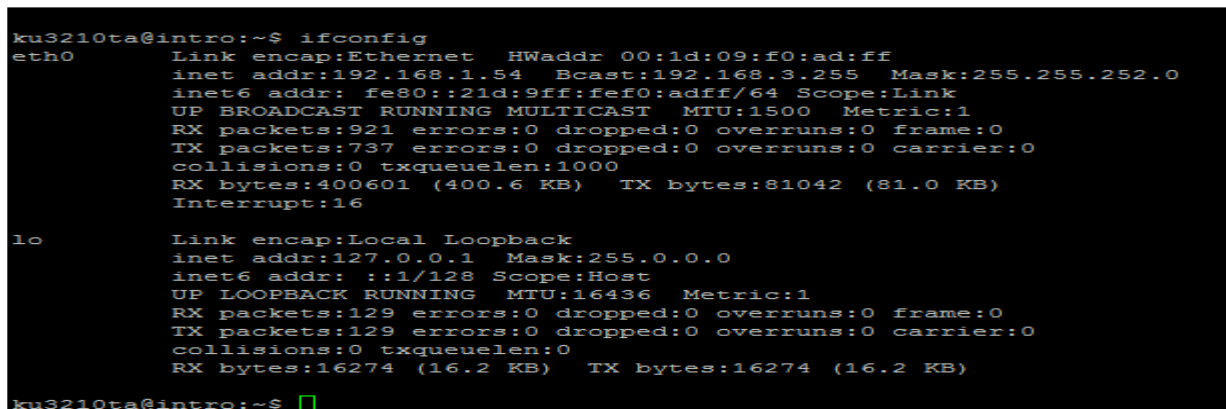
Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

Edit /etc/motd to change this login announcement.
[ku3210ta@users ~]$ ssh 192.168.1.201
```

Fig. 5: SSH connection using the IP address of the Node.

4. The IP address of the experimental node can be viewed by clicking on the EID, then click on the Node ID on the Reserve Nodes” table. The Node ID signifies the PC ID you are using. After clicking on the Node ID, you will see the node details. The IP Address of the node is the “Control Net IP”. Alternatively, the IP address can be viewed by typing “ifconfig” on the Ubuntu console. This will display the network configuration information of the node stating the broadcast address (or netmask address), IP address and so on. Figure 6 shows the IP address in the network configuration information.



```
ku3210ta@intro:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1d:09:f0:ad:ff
          inet addr:192.168.1.54  Bcast:192.168.3.255  Mask:255.255.252.0
          inet6 addr: fe80::21d:9ff:fef0:adff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:921 errors:0 dropped:0 overruns:0 frame:0
          TX packets:737 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:400601 (400.6 KB)  TX bytes:81042 (81.0 KB)
          Interrupt:16

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:129 errors:0 dropped:0 overruns:0 frame:0
          TX packets:129 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16274 (16.2 KB)  TX bytes:16274 (16.2 KB)

ku3210ta@intro:~$
```

Fig. 6: Network Configuration Information.

5. Alternatively, you can initiate an SSH connection using your qualified name. This can be viewed in the email you received when the experiment was swapped in. this is shown in figure 7 below.

```
users.deterlab.net - PuTTY
login as: ku3210ta
Using keyboard-interactive authentication.
Password for ku3210ta@users.isi.deterlab.net:
Last login: Tue Nov  1 23:33:47 2016 from 86.30.50.208
FreeBSD 10.1-RELEASE-p9 (GENERIC) #0: Tue Apr  7 01:09:46 UTC 2015

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:  https://www.FreeBSD.org/security/
FreeBSD Handbook:     https://www.FreeBSD.org/handbook/
FreeBSD FAQ:          https://www.FreeBSD.org/faq/
Questions List:       https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:       https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:     man hier

Edit /etc/motd to change this login announcement.
[ku3210ta@users ~]$ ssh intro.OLUTAYO.NISVLTE.isi.deterlab.net
```

Fig. 7: SSH connection using the qualified name.

### 3.1.4 Installing Apache

Apache is an open source and most widely used web server software that run on sixty-seven percent of all web servers worldwide. It is a reliable, secure and fast web server that can be customized to address the issues of different environments by utilizing modules and extension. To set up SSL, apache must first be installed. The following steps are required to install Apache web server through Ubuntu console.

1. From the Ubuntu console, type the command “sudo nano /etc/apt/sources.list”. This command is used to edit the directory path. The default Ubuntu “Main repository” and the “Update repository” must be edited for apache to be installed. This is necessary because apache installation will be denied. This is shown in figure 8 below.

```
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://scratch/ubuntu/ precise main restricted
deb-src http://scratch/ubuntu/ precise main

## Major bug fix updates produced after the final release of the
## distribution.
deb http://scratch/ubuntu/ precise-security main restricted
deb-src http://scratch/ubuntu/ precise-security main
```

Fig. 8: Ubuntu repository edit.

2. From the above, the line “deb-src http://scratch/ubuntu/ precise main restricted” was updated to “deb-src http://scratch/ubuntu/ precise main” while the lines “deb http://scratch/ubuntu/ precise-updates main restricted” and “deb-src http://scratch/ubuntu/ precise-updates main restricted” was updated to “deb http://scratch/ubuntu/ precise-security main restricted” and “deb-src http://scratch/ubuntu/ precise-security main” respectively.

3. After the update, press **ctrl+x** to go back to the console. The user will be asked to save the change, choose yes by pressing **y**, then press enter to save it in the same location.
4. Ubuntu needs to be updated after the repository in the source list has been edited. To ensure this, type **sudo apt-get update** or **sudo aptitude update** command on the console. This is to update the package index.
5. After the update, the next step is to install apache by typing **sudo apt-get install apache2** or **sudo aptitude install apache2** command on the console then press enter.
6. After apache installation, it is required to verify if it is installed or not. To verify, type **aptitude show apache2** command on the terminal. This displays the details of the apache package stating its status as shown in figure 9 below.

```
ku3210ta@intro:~$ aptitude show apache2
Package: apache2
State: installed
Automatically installed: no
Version: 2.2.22-1ubuntu1.11
Priority: optional
Section: web
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Uncompressed Size: 29.7 k
```

Fig. 9: Apache installation details.

After installation, the next step is to enable ssl. To do this type **sudo a2enmod ssl**. This is shown in figure 10 below.

```
ku3210ta@intro:~$ sudo a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
create self-signed certificates.
To activate the new configuration, you need to run:
    service apache2 restart
ku3210ta@intro:~$
```

Fig. 10: Enable SSL

8. Type **sudo /etc/init.d/apache2 restart** or **sudo service apache2 restart** to restart apache.

Apache can also be verified on the local machine web browser. Since the installation is done on a virtual node in deterlab, It will be impossible to run the localhost on the local machine web browser directly unlike installing apache on a local machine. To ensure this, an SSH port forwarding is required using PuTTY. Port forwarding is a network address translation (NAT) application that diverts a request from the combination of an address and port number to another when packets are navigating a network gateway such as firewall. To do this:

9. Open a new PuTTY, then type **users.isi.deterlab.net** into the hostname field, do not click open. Before making the connection, the configuration is required by setting up the tunnel parameters. On the list of category, expand the **SSH**, then click on **Tunnels** for configuration.
10. On the **Local Port** field, type **12345**. On the **Destination** field, type **your Node ID:80** e.g pc54:80. This means the local port 12345 is forwarded to a remote desktop protocol server port 80. Figure 11 shows the tunnel configuration with Node ID pc201.

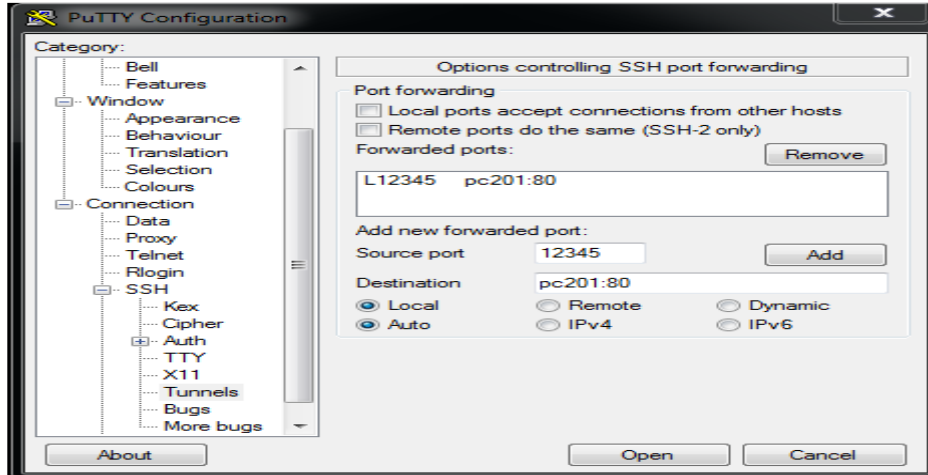


Fig. 11: Tunnel Configuration.

11. Click on **Open** to make a configuration. After that, log in to the web console with your Deterlab username and password.
12. Open a web browser on your local machine, then type **localhost:12345** or **127.0.0.1:12345** on the URL. A page confirming that apache web server was installed stating “It works!” is displayed as shown in figure 12 below.

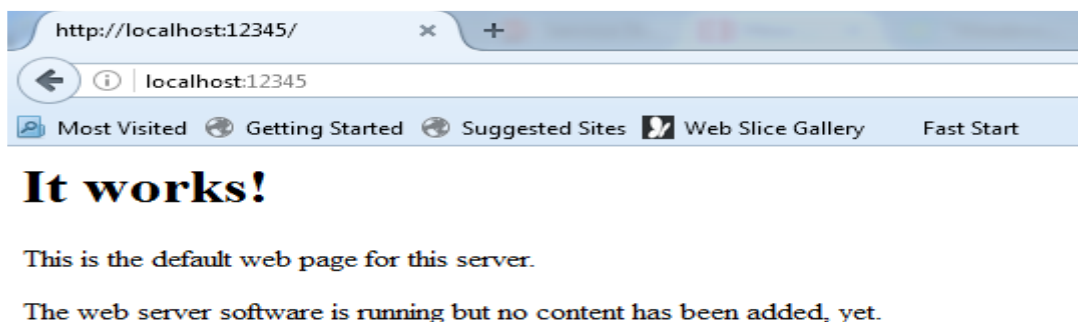
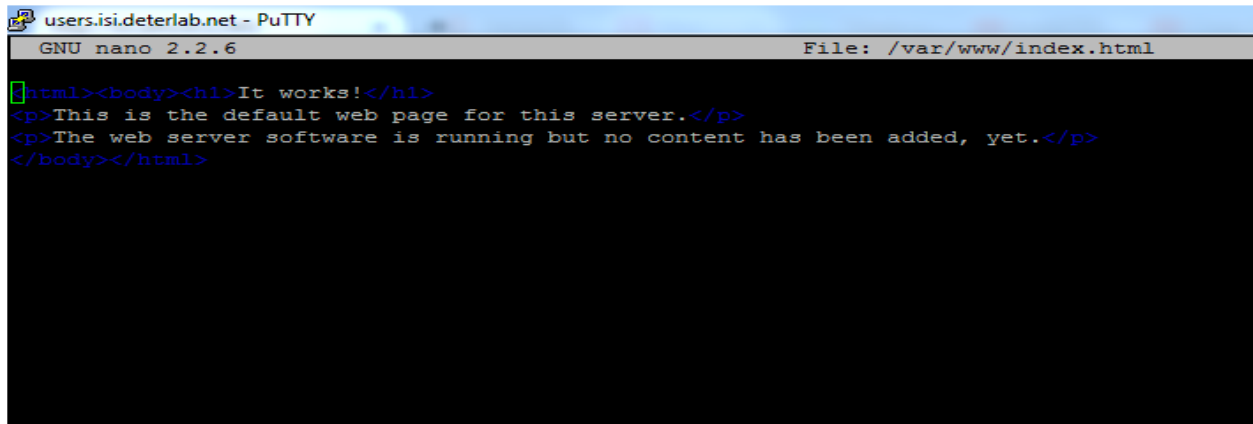


Fig. 12: web page confirming apache is installed.

The web page in figure 12 above can be altered. To do this, the index.html file located inside apache should be edited. The index.html page is the default page shown on a website when there is no other page

specified. To edit index.html file content, type **sudo nano/var/www/index.html**, then press enter. The index.html file will be displayed as shown in figure 13 below.

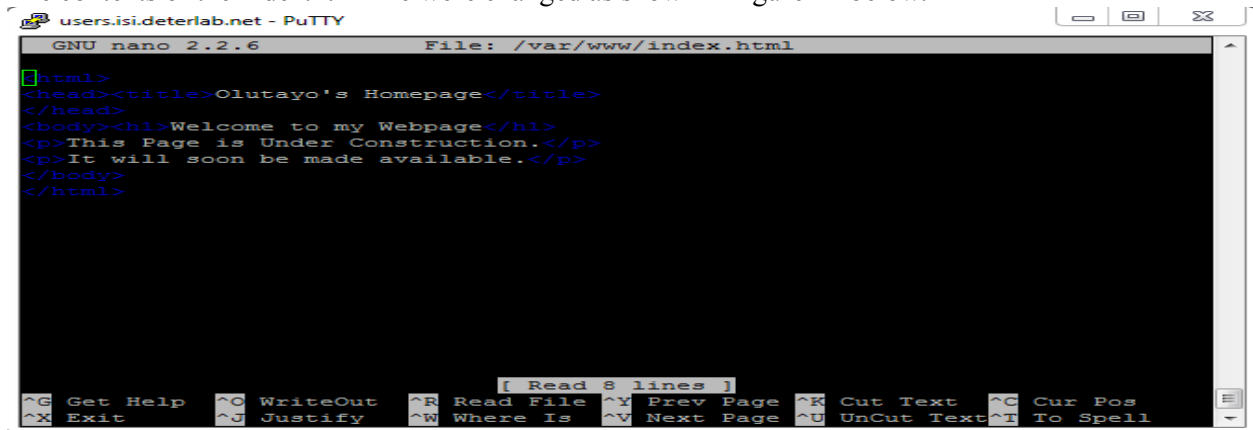


```
users.isi.deterlab.net - PuTTY
GNU nano 2.2.6 File: /var/www/index.html

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
```

Fig. 13: index.html content.

The contents of the index.html file were changed as shown in figure 14 below.



```
users.isi.deterlab.net - PuTTY
GNU nano 2.2.6 File: /var/www/index.html

<html>
<head><title>Olutayo's Homepage</title>
</head>
<body><h1>Welcome to my Webpage</h1>
<p>This Page is Under Construction.</p>
<p>It will soon be made available.</p>
</body>
</html>
```

Fig. 14: index.html file edit.

As displayed in figure 14 above, save the new content and type **localhost:12345** on the web page URL. A new content will be displayed on the web page as shown in figure 15 below.

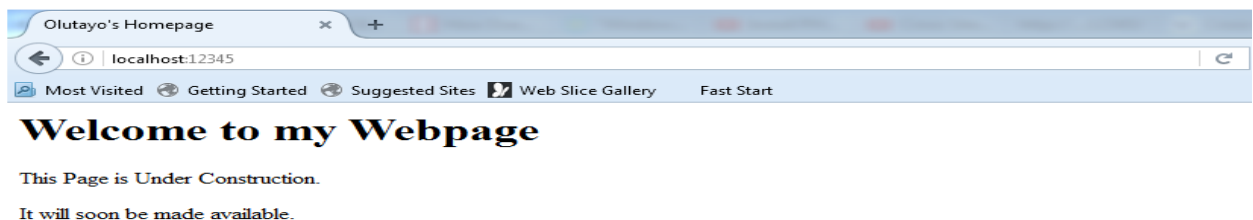


Fig. 15: Edited web page.

### 3.1.5 Creating a Self-Signed SSL Certificate on Apache

To install SSL certificate on apache, an open source project, called OpenSSL, is required to be installed. OpenSSL is an all-purpose cryptography library that offers implementation of open-source in Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. The cryptography library gives the most prominent algorithms for symmetric key and asymmetric key cryptography, message digests and hash functions. To install OpenSSL, the following steps are taken:

1. On the console, type **sudo apt-get install openssl** or **sudo aptitude install openssl**. This will take some time to install.
2. To verify if OpenSSL is installed, type **aptitude show openssl**. The result is shown in figure 16 below.

```
ku3210ta@intro:~$ aptitude show openssl
Package: openssl
State: installed
Automatically installed: no
Version: 1.0.1-4ubuntu5.38
Priority: standard
Section: utils
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Uncompressed Size: 928 k
Depends: libc6 (>= 2.15), libssl1.0.0 (>= 1.0.1)
Suggests: ca-certificates
Conflicts: openssl
Description: Secure Socket Layer (SSL) binary and related cryptographic tools
 This package contains the openssl binary and related tools.

It is part of the OpenSSL implementation of SSL.

You need it to perform certain cryptographic actions like:
* Creation of RSA, DH and DSA key parameters;
* Creation of X.509 certificates, CSRs and CRLs;
* Calculation of message digests;
* Encryption and decryption with ciphers;
* SSL/TLS client and server tests;
* Handling of S/MIME signed or encrypted mail.

ku3210ta@intro:~$ █
```

Fig. 16: OpenSSL installation details

After the verification, the next step is to create an SSL certificate. The following steps are taken to create the certificate.

3. The first step is to create a new directory where the server key and the certificate will be stored. To do this, type **sudo mkdir /etc/apache2/ssl**.
4. The next step is to generate the keys for the Certificate Signing Request (CSR). To do this, type **sudo openssl genrsa -out ca.key 2048** command on the console. This is displayed in figure 17 below.

```
users.isi.deterlab.net - PuTTY
ku3210ta@intro:~$ sudo openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
ku3210ta@intro:~$
```

Fig. 17: RSA Private Key generation.

In figure 17 above, the private key, ca.key, with the length of 2048 bits modulus was generated.

The next step is the request for a certificate. Here, the information that will be incorporated into the certificate request will be entered. To request for certificate:

5. Type **sudo openssl req -nodes -new -key ca.key -out ca.csr** on the Ubuntu console and press enter. Some set of questions such as Country Name (a two letter code), State or Province Name (full name), Locality Name (e.g city), Organization Name (e.g company), Organizational Unit Name (e.g section), Common Name (e.g server FQDN or YOUR name), and Email Address will be requested to be filled. An extra attribute is also requested which will be sent with the certificate request which is a challenged password and an optional company name. Figure 18 shows the details of the certificate request.

```
ku3210ta@intro:~$ sudo openssl req -nodes -new -key ca.key -out ca.csr
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]:UK
State or Province Name (full name) [Some-State]:London
Locality Name (eg, city) []:Kingston
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Kingston Uni
Organizational Unit Name (eg, section) []:NIS
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
ku3210ta@intro:~$
```

Fig. 18: Certificate Request.

6. The next step is to create a self-signed certificate. To do this, type **sudo openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt** on the console and press enter. This is shown in figure 19 below.

```
users.isi.deterlab.net - PuTTY
ku3210ta@intro:~$ sudo openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
Signature ok
subject=/C=UK/ST=London/L=Kingston/O=Kingston Uni/OU=NIS/CN=Olutayo
Getting Private key
ku3210ta@intro:~$
```

Fig. 19: Self-Signed Certificate creation.



7. To verify if the private key and SSL certificate, type **ls** on the command line. The **ca.csr**, **ca.crt** and **ca.key** will be displayed as shown in figure 20 below.

```
ku3210ta@intro:~$ ls
ca.crt  ca.csr  ca.key  topsecret
ku3210ta@intro:~$
```

Fig. 20: Private Key and SSL certificate generated.

8. The next step is to move the certificate files into the created folder. Type **sudo mv ca.crt ca.key ca.csr /etc/apache2/ssl/**

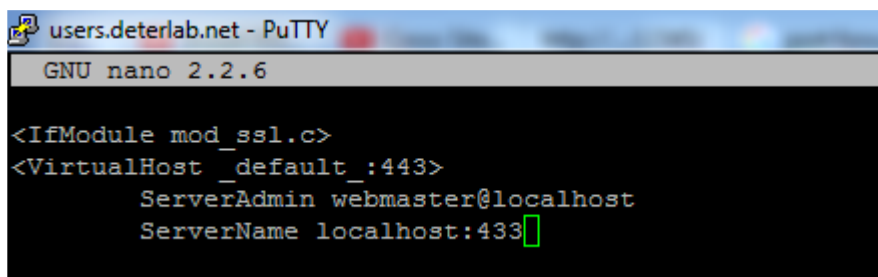
9. Type **sudo a2ensite default-ssl** to enable the site using SSL.

10. Type **sudo service apache2 reload** to reload apache.

11. Type **cd /etc/apache2/sites-enabled/** to change directory to the location of apache configuration file.

12. Type **sudo nano default-ssl** on the console to edit the SSL configuration file for apache.

13. On the fourth line after the **ServerAdmin webmaster@localhost**, type **ServerName localhost:443** as shown in figure 21 below.



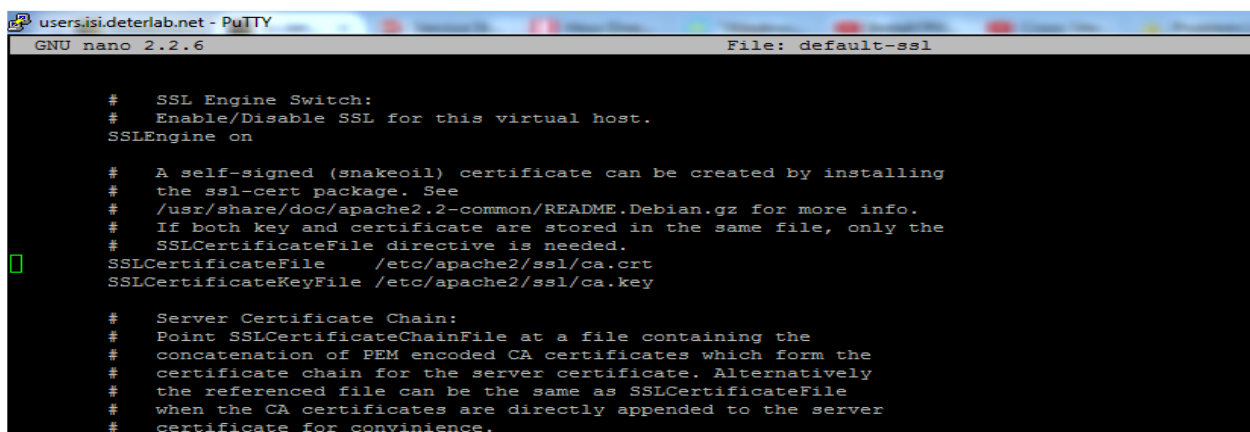
```
users.deterlab.net - PuTTY
GNU nano 2.2.6

<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    ServerName localhost:443
```

Fig. 21: Addition of server name to Apache config file.

The server name is the common name filled in the information for certificate request in figure 24 above.

14. The next step is to edit the next two lines after the comment “# SSLCertificateFile directive is needed”. Edit **SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem** and **SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key** to **SSLCertificateFile /etc/apache2/ssl/ca.crt** and **SSLCertificateKeyFile /etc/apache2/ssl/ca.key** respectively. This is shown in figure 22 below.



```
users.isi.deterlab.net - PuTTY
GNU nano 2.2.6                               File: default-ssl

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/apache2/ssl/ca.crt
SSLCertificateKeyFile /etc/apache2/ssl/ca.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
```

Fig.22: SSL certificate file Edit.



15. After editing the apache configuration file, type **Ctrl+X** to close, save the modified file by pressing “y”, and press “enter” to save to the same folder.

The virtual host needs to be enabled before the website on port 443 can be activated. To set up the virtual host to display the new certificate, the following steps must be taken:

To view the SSL certificate created, localhost should be run on the local machine web browser. To achieve this, the following steps should be taken:

16. Open a new PuTTY to port forward. Type **user.isi.deterlab.net** as the hostname. Do not click on “open”.

17. Expand “Connection”, expand “SSH”, and then click on “Tunnels”.

18. Type **443** as the source port while the destination port is your **Node ID:443**. The node ID assigned to this experiment is **pc201**.

19. Click on “ADD”. The update will be displayed as shown in figure 23 below.

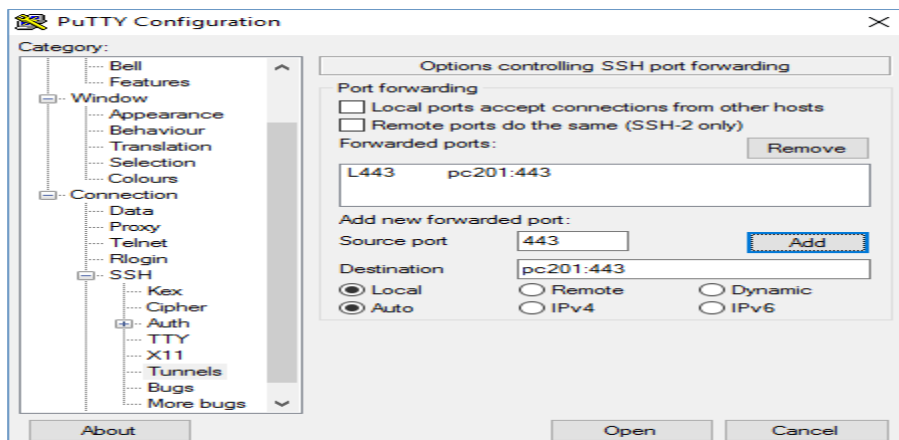


Fig. 23: Port Forwarding.

Note that Deterlab allocates different node ID anytime an experiment is being run.

20. Click on **Open** to initiate the connection.

21. Type in your Deterlab username and password.

22. Open any local web browser and type <https://localhost:443> or <https://localhost>. The result will be displayed as shown in figure 24 below.

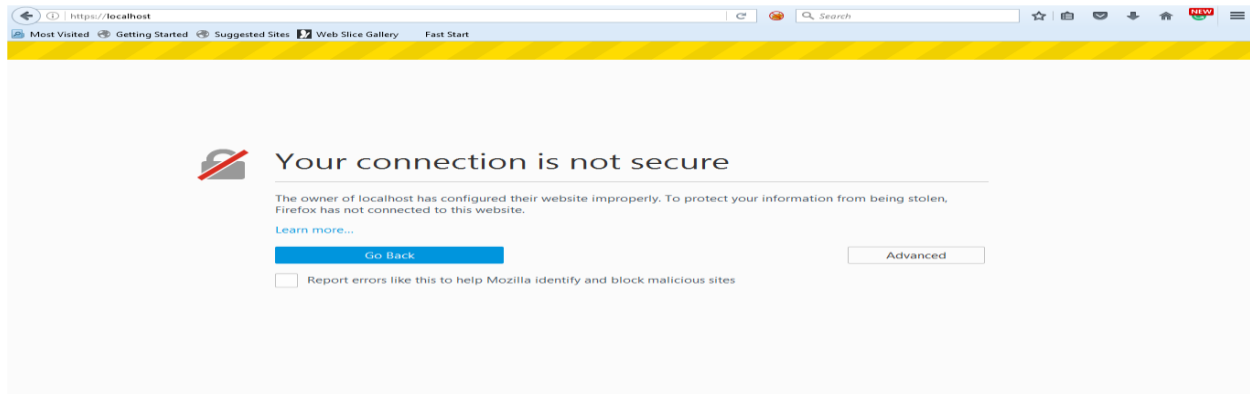


Fig. 24: Running https on local machine web browser.

23. Click on **Advanced**, then click on **Add Exceptions**. An **Add security Exception** page will pop up as shown in figure 25 below.

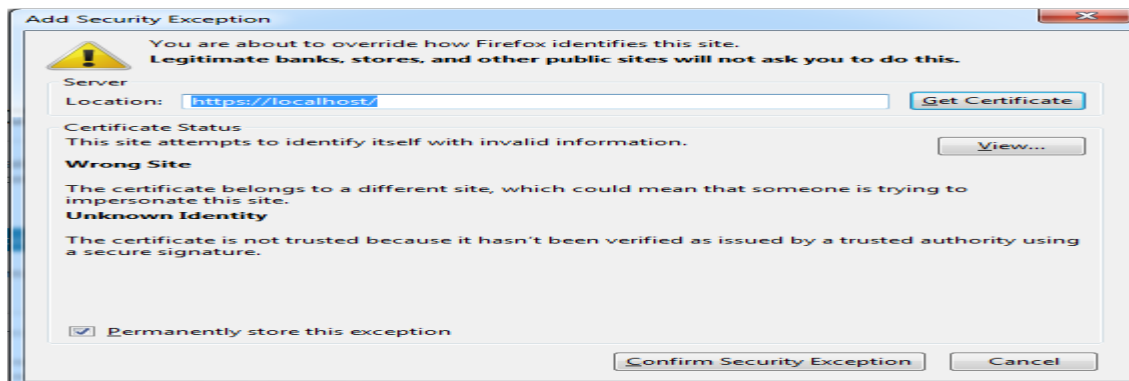


Fig. 25: Add Security Exception Page.

24. Click on **Confirm Security Exception**, and the localhost is secured. This is displayed in figure 26 below.

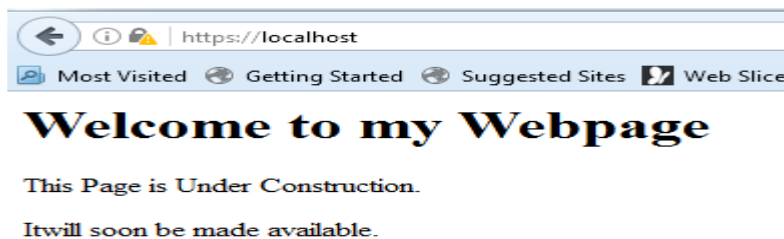


Fig. 26: Secure web server.

To view the self-signed SSL certificate, click on the padlock displayed on the URL, click on the arrow in front of the localhost to expand, and click on “More Information”. A page displaying the **Page info** – https://localhost/ pops up as shown in figure 27.

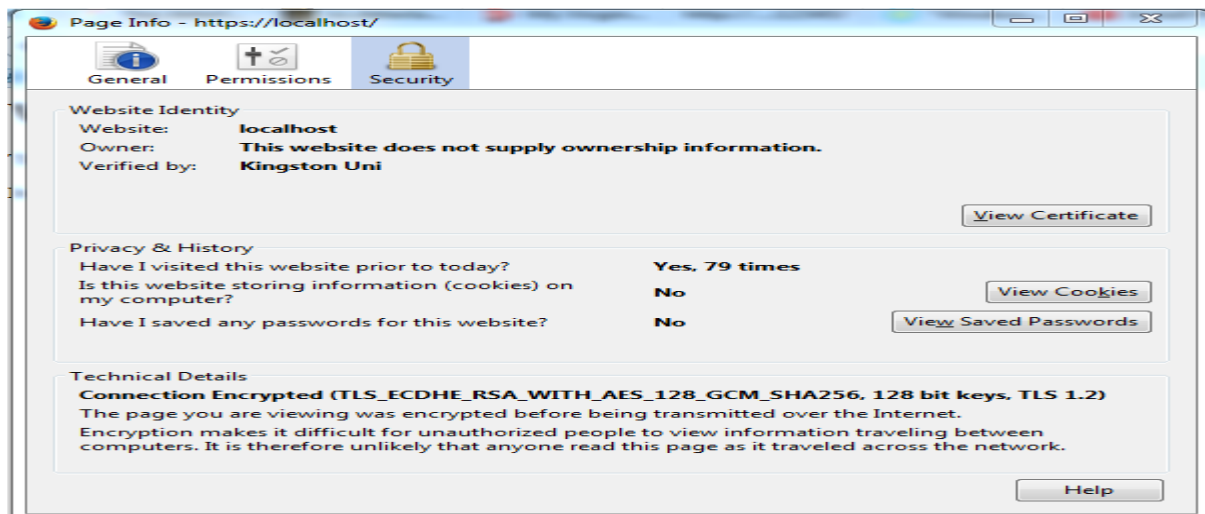


Fig. 27: Page Info.

25. The certificate can be viewed by clicking on **View Certificate** and the certificate information will be displayed as shown in figure 28 below.

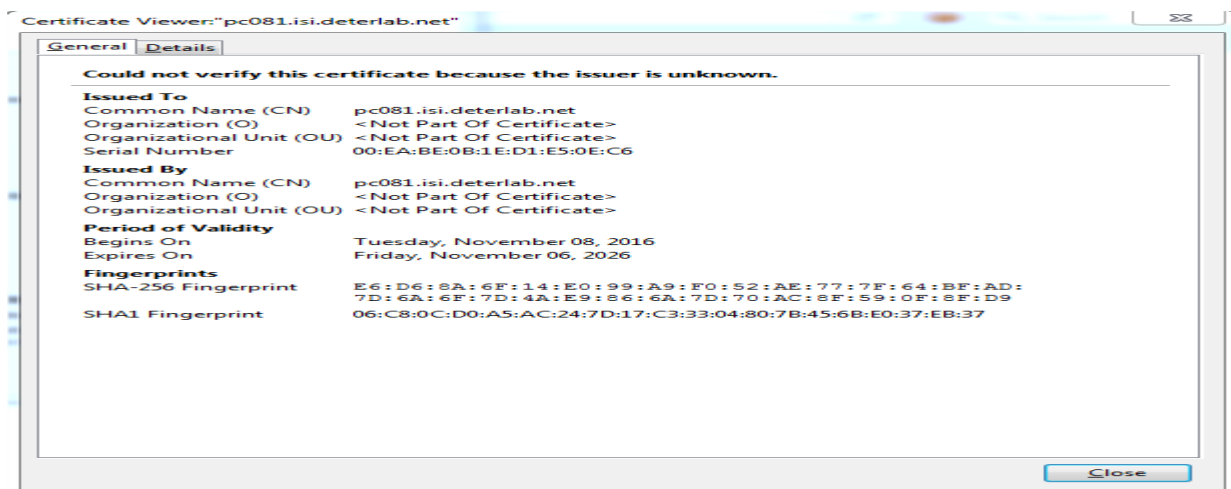


Fig. 28: Certificate info.

Note that when a self-signed certificate is already installed on the local machine, it will not work. For a local machine with an already installed self-signed certificate, to run the SSL content from Deterlab on a local machine web browser requires, the TCP listening port must be checked.

To check the TCP listening ports, type **sudo netstat -plnt** on the command line. A list of listening ports apart from port 443 will be listed.

From the list, source port 80 was chosen for port forwarding. This is displayed in figure 29 below.

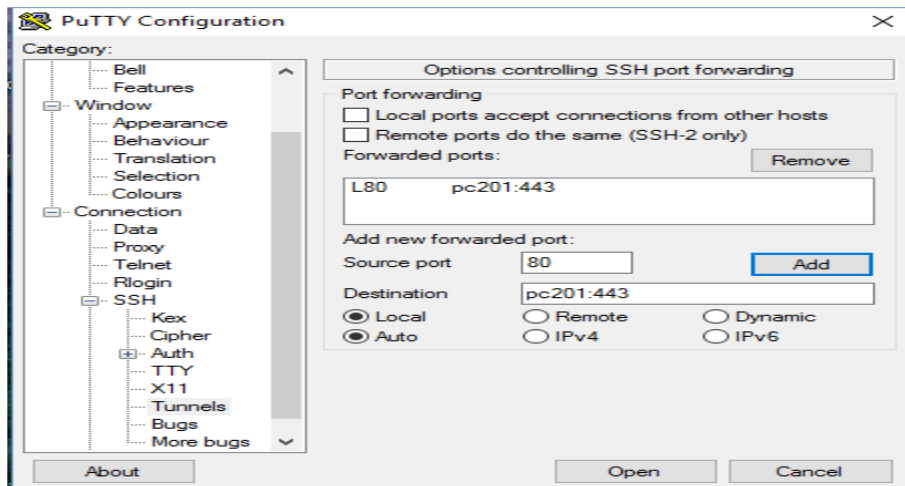


Fig. 29: Port forwarding with source port 80.

## 4.0 RESULTS AND CONCLUSION

This paper address issues of the unsecured website, demonstration of how to set up a Secure Socket Layer (SSL) on a localhost and how it can also be implemented in Deterlab. The different components of public key infrastructure were discussed and the various functions of a certificate authority were described. An experiment was also implemented to set up a self-signed SSL certificate.

The paper can serve as a benefit to students on the illustration on how to secure a web server, how to generate a self-signed Secure Socket Layer (SSL) certificates for a web server and how to make use of Deterlab to run the experiment. Lecturers or educators can also benefit on this new secure testbed by understanding the procedure for registration and steps involved in running experiments in Deterlab virtual environment.

Deterlab offers various potential favorable benefits over the utilization of local university laboratories for exercises on security. Benefits like sharing and reuse of instructive materials. Exercises on Deterlab are effectively reusable over different locales and institutions since they are altogether created inside a typical, broadly accessible environment.

Since the configuration of exercises can be filed and reused, educationalists find it is easy to impart activities to others and utilize Deterlab's activities for reiteration of courses. Deterlab offers huge benefits to group individuals that can't bolster nearby labs all alone [3]. To understand these favorable circumstances, Deterlab created and furnished educators with a portal for the public to have access to shared activities and to contribute new material to the bigger instructive community.

Deterlab additionally offers effective useful and advantageous preferences to instructors or educators by giving extensible hardware resources and can bolster activities and exercises of noteworthy, sensible size and difficulty. Deterlab facility was designed unequivocally to cover risk from exercises that involve live malware and other unsafe practices, permitting these points to be incorporated into the instructive material. The competencies of Deterlab's remote access permit understudies to perform lab exercises by themselves. This can be done from their homes, dormitories, or labs.

In the instructive utilization of Deterlab, the Deter project has completed exercises in two unmistakable measurements: advancement of education-specific specialized abilities and improvement of openly available and shared educational curriculum materials. Deterlab testbed contains some mechanism which are Access control suited to the use in classroom and authoritative consent delegation, the capacity to create in batch and to delete in batch students in view of information from external sources, and mechanisms for scheduling and new allocation for resource sharing and management over educational activities at various granularities e.g class, each student, or group, without trading off the research users of facility.

## 5.0 REFERENCES

- [1] Amandeep K, Harmandeep S. Network Security: A Literature Review. International Journal of Emerging Research in Management & Technology 2014;3(10):32-39.
- [2] An Idiots Guide to Public Key Infrastructure 2002 <https://www.giac.org/paper/gsec/2171/idiots-guide-public-key-infrastructure/103692>
- [3] Benzel T, Branden R, Kim D, Joseph AD, Numan BC, Ostrega R, Schwab S, Sklwer K. Design, Deployment, and Use of the DETER Testbed. DETER; 2007.
- [4] Choudhury S, Bhatnagar K, Haque W. Public key infrastructure implementation and design: John Wiley & Sons, Inc.; 2002.
- [5] Cooper D. Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile. 2008.
- [6] Daya B. Network security: History, importance, and future. University of Florida Department of Electrical and Computer Engineering 2013.
- [7] Goyal P, Batra S, Singh A. A literature review of security attack in mobile ad-hoc networks. International Journal of Computer Applications 2010;9(12):11-15.
- [8] Mirkovic J, Benzel T. Deterlab testbed for cybersecurity research and education. Journal of Computing Sciences in Colleges 2013;28 (4):163-163.
- [9] Ristic I. Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications. : Feisty Duck; 2013.
- [10] Stallings W. Network security essentials: applications and standards. : Pearson Education India; 2007.
- [11] The network simulator ns-2 <http://www.isi.edu/nsnam/ns/>
- [12] Viega J, Messier M, Chandra P. Network Security with OpenSSL: Cryptography for Secure Communications. : " O'Reilly Media, Inc."; 2002