



**UNIVERSITY of  
GREENWICH**

**Module: System Administration and Security**

**Module Code: COMP-1475-M01-2020-21**

**Coordinator: Dr Mariusz Pelc**

**Student Name: Redwan Ahmed**

**Student ID: 001147921**

**Group Number: 24**

**Group members:**

**Prince Elvis Gomes**

**Md Sumsush Zaman**

## CONTENTS

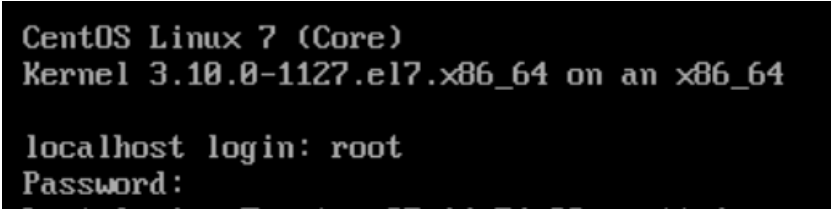
Log Book: .....	3
Centos Version:.....	3
Apache Installation:.....	3
MySQL Installation:.....	8
PHP Installation: .....	13
Introduction: .....	15
Literature Review:.....	15
Lamp Architecture: .....	16
Part 1.....	16
LAMP Stack Installation: .....	16
MySQL Installation:.....	16
Part 2:.....	18
Vulnerability report: .....	18
Risk Assessment: .....	18
Audit Report: .....	19
Conclusion:.....	19
References: .....	20

## LOG BOOK:

LAMP Stack is a bundle of open-source software's including Linux, Apache, MySQL/MariaDB, PHP/Python, together they provide high-performance websites and web app also connecting the clients and the servers. The architecture which Lamp Stack was designed goes like this, the very bottom level is Linux then Apache alone with MariaDB and finally on the top of it is the PHP.

The Installation of lamp stack is as given below –

### CENTOS VERSION:



```
CentOS Linux 7 (Core)
Kernel 3.10.0-1127.el7.x86_64 on an x86_64

localhost login: root
Password:
```

Figure 1: CentOS version

### APACHE INSTALLATION:

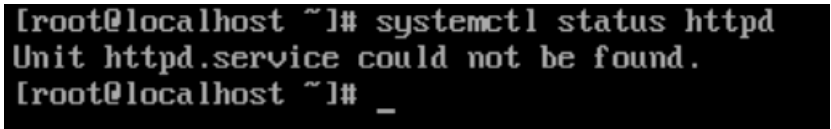
Step 1 – Install Apache,

Currently, one of the most popular web server is Apache web server.

In order to check apache running first command is -

**systemctl status httpd,**

systemctl is the system control and if it's not installed "Unit httpd.service could not be found" will show which indicates the server can't be found and installed.



```
[root@localhost ~]# systemctl status httpd
Unit httpd.service could not be found.
[root@localhost ~]# _
```

Figure 2: Check Apache running or not

If we like to check netstat, to get the netstat installed we need to check the package 1<sup>st</sup> and the command for that is - **yum provides \*/netstat**. To see the package it should be net-tools, we can use the command **yum**

-y  
install  
net-  
tools.  
Now if  
we can  
see  
netstat  
by  
using

```
[root@localhost ~]# yum -y install net-tools
Loaded plugins: fastestmirror
Determining fastest mirrors
base                                                    | 3.0 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package net-tools.x86_64 0:2.0-0.25.20131004git.e17 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version                               Repository      Size
=====
Installing:
net-tools               x86_64           2.0-0.25.20131004git.e17            base            306 k
Transaction Summary
=====
Install 1 Package

Total download size: 306 k
Installed size: 917 k
Downloading packages:
net-tools-2.0-0.25.20131004git.e17.x86_64.rpm          | 306 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : net-tools-2.0-0.25.20131004git.e17.x86_64      1/1
  Verifying  : net-tools-2.0-0.25.20131004git.e17.x86_64      1/1

Installed:
  net-tools.x86_64 0:2.0-0.25.20131004git.e17

Complete!
[root@localhost ~]#
```

Figure 3: Installed httpd packages

command **netstat -a | less**, this shows some open parts “ssh, smtp, bootpc, ipv6-icmp” open. By doing this, it is indicating that httpd is not installed in here.

Active Internet connections (servers and established)						
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	
tcp	0	0	0.0.0.0:ssh	0.0.0.0:*	LISTEN	
tcp	0	0	localhost:smtp	0.0.0.0:*	LISTEN	
tcp6	0	0	:::ssh	:::*	LISTEN	
tcp6	0	0	localhost:smtp	:::*	LISTEN	
udp	0	0	localhost:323	0.0.0.0:*		
udp	0	0	0.0.0.0:bootpc	0.0.0.0:*		
udp6	0	0	localhost:323	:::*		
raw6	13824	0	:::ipv6-icmp	:::*	7	
Active UNIX domain sockets (servers and established)						
Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ ACC ]	STREAM	LISTENING	12320	/run/systemd/private
unix	3	[ ]	DGRAM		7480	/run/systemd/notify
unix	2	[ ]	DGRAM		7482	/run/systemd/cgroups-agent
unix	2	[ ACC ]	STREAM	LISTENING	19231	public/pickup
unix	2	[ ACC ]	STREAM	LISTENING	19235	public/cleanup
unix	2	[ ACC ]	STREAM	LISTENING	19238	public/qmgr
unix	2	[ ACC ]	STREAM	LISTENING	19275	public/showq
unix	2	[ ACC ]	STREAM	LISTENING	7496	/run/systemd/journal/stdout
unix	2	[ ACC ]	STREAM	LISTENING	17739	/var/run/NetworkManager/private-dhcp
unix	5	[ ]	DGRAM		7499	/run/systemd/journal/socket
unix	13	[ ]	DGRAM		7501	/dev/log
unix	2	[ ACC ]	STREAM	LISTENING	19260	public/flush
unix	2	[ ACC ]	STREAM	LISTENING	12404	/run/lvm/lvm2d.socket
unix	2	[ ]	DGRAM		12700	/run/systemd/shutdown
unix	2	[ ACC ]	STREAM	LISTENING	12706	/run/lvm/lvmpolld.socket
unix	2	[ ACC ]	SEQPACKET	LISTENING	12708	/run/udev/control
unix	2	[ ]	DGRAM		15792	/var/run/chrony/chronyd.sock
unix	2	[ ACC ]	STREAM	LISTENING	19242	private/tlsmgr
unix	2	[ ACC ]	STREAM	LISTENING	19245	private/rewrite
unix	2	[ ACC ]	STREAM	LISTENING	19248	private/bounce
unix	2	[ ACC ]	STREAM	LISTENING	19251	private/defer
unix	2	[ ACC ]	STREAM	LISTENING	19254	private/trace
unix	2	[ ACC ]	STREAM	LISTENING	19257	private/verify
unix	2	[ ACC ]	STREAM	LISTENING	19263	private/proxymap

Figure 4: checking netstat

So, in the system no httpd is available in the system, but we want to install httpd daemon to installed we can check the package. The command is **yum provides \*/httpd**.

```

Repo      : base
Matched from:
Filename  : /usr/lib64/httpd
Filename  : /usr/sbin/httpd
Filename  : /var/log/httpd
Filename  : /etc/httpd
Filename  : /run/httpd
Filename  : /etc/logrotate.d/httpd
Filename  : /var/cache/httpd
Filename  : /usr/libexec/initscripts/legacy-actions/httpd
Filename  : /etc/sysconfig/httpd
Filename  : /usr/share/httpd

httpd-devel-2.4.6-93.el7.centos.x86_64 : Development interfaces for the Apache HTTP server
Repo      : base
Matched from:
Filename  : /usr/include/httpd

python-custodia-0.3.1-4.el7.noarch : Sub-package with python2 custodia modules
Repo      : base
Matched from:
Filename  : /usr/lib/python2.7/site-packages/custodia/httpd

qt3-devel-docs-3.3.8b-51.el7.x86_64 : Documentation for the Qt 3 GUI toolkit
Repo      : base
Matched from:
Filename  : /usr/share/doc/qt3-devel-docs-3.3.8b/examples/network/httpd

```

Figure 5; httpd package checking

Installation of Apache is a simple task and we can do that using yum manager and yum manager is in CentOS packet manager. “YUM Repositories are warehouses of Linux software (RPM package files)”. In order to install **httpd** the command **yum -y install httpd** is needed. Now we can see the status of the server, **systemctl status httpd**, we can see that httpd is ‘dead’ here, which means it is not yet running. We can check if it’s running by connecting with the localhost.

```

Install 1 Package (+4 Dependent packages)

Total download size: 3.0 M
Installed size: 10 M
Downloading packages:
(1/5): apr-util-1.5.2-6.el7.x86_64.rpm           | 92 kB  00:00:00
(2/5): apr-1.4.8-5.el7.x86_64.rpm               | 103 kB 00:00:00
(3/5): httpd-tools-2.4.6-93.el7.centos.x86_64.rpm | 92 kB  00:00:00
(4/5): mailcap-2.1.41-2.el7.noarch.rpm           | 31 kB  00:00:00
(5/5): httpd-2.4.6-93.el7.centos.x86_64.rpm      | 2.7 MB 00:00:00
-----
Total                                           13 MB/s | 3.0 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.4.8-5.el7.x86_64                1/5
  Installing : apr-util-1.5.2-6.el7.x86_64           2/5
  Installing : httpd-tools-2.4.6-93.el7.centos.x86_64 3/5
  Installing : mailcap-2.1.41-2.el7.noarch            4/5
  Installing : httpd-2.4.6-93.el7.centos.x86_64       5/5
  Verifying  : apr-1.4.8-5.el7.x86_64                1/5
  Verifying  : httpd-tools-2.4.6-93.el7.centos.x86_64 2/5
  Verifying  : mailcap-2.1.41-2.el7.noarch            3/5
  Verifying  : httpd-2.4.6-93.el7.centos.x86_64       4/5
  Verifying  : apr-util-1.5.2-6.el7.x86_64           5/5

Installed:
  httpd.x86_64 0:2.4.6-93.el7.centos

Dependency Installed:
  apr.x86_64 0:1.4.8-5.el7                apr-util.x86_64 0:1.5.2-6.el7
  httpd-tools.x86_64 0:2.4.6-93.el7.centos mailcap.noarch 0:2.1.41-2.el7

Complete!
[root@localhost ~]#

```

Figure 6: Installing httpd

We can also check turning the localhost. So, the programmes which can be used to connect lynx.

In order to install Lynx the command is **yum -y install lynx**.

```
=====
Package Arch Version Repository Size
=====
Installing:
lynx x86_64 2.8.8-0.3.dev15.el7 base 1.4 M
Installing for dependencies:
centos-indexhtml noarch 7-9.el7.centos base 92 k
Transaction Summary
=====
Install 1 Package (+1 Dependent package)

Total download size: 1.5 M
Installed size: 5.4 M
Downloading packages:
(1/2): centos-indexhtml-7-9.el7.centos.noarch.rpm | 92 kB 00:00:00
(2/2): lynx-2.8.8-0.3.dev15.el7.x86_64.rpm | 1.4 MB 00:00:00
-----
Total 11 MB/s | 1.5 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : centos-indexhtml-7-9.el7.centos.noarch 1/2
Installing : lynx-2.8.8-0.3.dev15.el7.x86_64 2/2
Verifying : lynx-2.8.8-0.3.dev15.el7.x86_64 1/2
Verifying : centos-indexhtml-7-9.el7.centos.noarch 2/2

Installed:
lynx.x86_64 0:2.8.8-0.3.dev15.el7

Dependency Installed:
centos-indexhtml.noarch 0:7-9.el7.centos

Complete!
[root@localhost ~]#
```

Figure 7: Installing lynx

Then can try lynx <http://localhost>, It shows alert “Unable to connect to remote host”, it means that the service is installed, but not running.

```
Looking up localhost
Making HTTP connection to localhost
Alert!: Unable to connect to remote host.

lynx: Can't access startfile http://localhost/
[root@localhost ~]# _
```

Figure 8: Apache is installed but not showing

After installation, we can start Apache virtual private server by using this command **systemctl start httpd.service**.

If we give command **systemctl start httpd**, it is running and if we again check lynx localhost, we get “Test webpage” which means we successfully connected to the server.

```
Testing 123..

This page is used to test the proper operation of the Apache HTTP server after it has been
installed. If you can read this page it means that this site is working properly. This
server is powered by CentOS.

Just visiting?

The website you just visited is either experiencing problems or is undergoing routine
maintenance.

If you would like to let the administrators of this website know that you've seen this page
instead of the page you expected, you should send them e-mail. In general, mail sent to the
name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send
e-mail to "webmaster@example.com".

Are you the Administrator?

You should add your website content to the directory /var/www/html/.

To prevent this page from ever being used, follow the instructions in the file
/etc/httpd/conf.d/welcome.conf.

Promoting Apache and CentOS

You are free to use the images below on Apache and CentOS Linux powered HTTP servers.
Thanks for using Apache and CentOS!

[ Powered by Apache ] [ Powered by CentOS Linux ]

Important note:
-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H?elp O?ptions P?rint G?o M?ain screen Q?uit /?search [delete]=history list
```

Figure 9: Apache is running

To enable Apache we use the command **systemctl status httpd.service**.

```
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# systemctl status httpd
■ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-04-23 09:34:27 BST; 3s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 1437 (httpd)
    Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           └─1437 /usr/sbin/httpd -DFOREGROUND
             └─1438 /usr/sbin/httpd -DFOREGROUND
               └─1439 /usr/sbin/httpd -DFOREGROUND
                 └─1440 /usr/sbin/httpd -DFOREGROUND
                   └─1441 /usr/sbin/httpd -DFOREGROUND
                     └─1442 /usr/sbin/httpd -DFOREGROUND

Apr 23 09:34:26 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Apr 23 09:34:27 localhost.localdomain httpd[1437]: AH00558: httpd: Could not reliably determine...ge
Apr 23 09:34:27 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

Figure 10: Apache is successfully running

## MYSQL INSTALLATION:

### Step 2 – Install MySQL

Since the Apache server is up and running properly, the next step is to install now MariaDB.

Again, we can check that mysql is running or not by using this command **systemctl status mysql**. So, it is not found.



```
[root@localhost ~]# systemctl status mysql
Unit mysql.service could not be found.
[root@localhost ~]# _
```

Figure 11: MySQL is not installed

We can check then the specific package that contains mysql using the command **yum provides \*/mysqld**. We can see mariadb server in the Figure 12.

```
[root@localhost ~]# yum provides */mysqld
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
1:mariadb-server-5.5.65-1.el7.x86_64 : The MariaDB server and related files
Repo      : base
Matched from:
Filename  : /usr/libexec/mysqld
```

Figure 12: Checking mysqld packages

We also need to check mysql packages by using the command **yum provides \*/mysql**.

```
python-sqlalchemy-0.9.8-2.el7.x86_64 : Modular and flexible ORM library for python
Repo      : base
Matched from:
Filename  : /usr/lib64/python2.7/site-packages/sqlalchemy/dialects/mysql

resource-agents-4.1.1-46.el7.x86_64 : Open Source HA Reusable Cluster Resource Scripts
Repo      : base
Matched from:
Filename  : /usr/lib/ocf/resource.d/heartbeat/mysql
Filename  : /usr/share/resource-agents/ocft/configs/mysql

selinux-policy-minimum-3.13.1-266.el7.noarch : SELinux minimum base policy
Repo      : base
Matched from:
Filename  : /etc/selinux/minimum/active/modules/100/mysql

selinux-policy-mls-3.13.1-266.el7.noarch : SELinux mls base policy
Repo      : base
Matched from:
Filename  : /etc/selinux/mls/active/modules/100/mysql

selinux-policy-targeted-3.13.1-266.el7.noarch : SELinux targeted base policy
Repo      : base
Matched from:
Filename  : /etc/selinux/targeted/active/modules/100/mysql
```

Figure 13: Checking mysql package

Mariadb needed to be installed because that contains mysql. The installation command is **yum -y install mariadb**.

```
Verifying : perl-Time-Local-1.2300-2.el7.noarch 14/28
Verifying : 4:perl-macros-5.16.3-295.el7.x86_64 15/28
Verifying : 4:perl-5.16.3-295.el7.x86_64 16/28
Verifying : perl-Carp-1.26-244.el7.noarch 17/28
Verifying : 4:perl-Time-HiRes-1.9725-3.el7.x86_64 18/28
Verifying : perl-Scalar-List-Utils-1.27-248.el7.x86_64 19/28
Verifying : perl-Pod-Usage-1.63-3.el7.noarch 20/28
Verifying : perl-Encode-2.51-7.el7.x86_64 21/28
Verifying : perl-Pod-Perldoc-3.20-4.el7.noarch 22/28
Verifying : perl-podlators-2.5.1-3.el7.noarch 23/28
Verifying : perl-File-Path-2.09-2.el7.noarch 24/28
Verifying : perl-threads-1.87-4.el7.x86_64 25/28
Verifying : perl-Filter-1.49-3.el7.x86_64 26/28
Verifying : perl-Getopt-Long-2.40-3.el7.noarch 27/28
Verifying : perl-Text-ParseWords-3.29-4.el7.noarch 28/28

Installed:
  mariadb.x86_64 1:5.5.65-1.el7

Dependency Installed:
  perl.x86_64 4:5.16.3-295.el7
  perl-Encode.x86_64 0:2.51-7.el7
  perl-File-Path.noarch 0:2.09-2.el7
  perl-Filter.x86_64 0:1.49-3.el7
  perl-HTTP-Tiny.noarch 0:0.033-3.el7
  perl-Pod-Escapes.noarch 1:1.04-295.el7
  perl-Pod-Simple.noarch 1:3.20-4.el7
  perl-Scalar-List-Utils.x86_64 0:1.27-248.el7
  perl-Storable.x86_64 0:2.45-3.el7
  perl-Time-HiRes.x86_64 4:1.9725-3.el7
  perl-constant.noarch 0:1.27-2.el7
  perl-macros.x86_64 4:5.16.3-295.el7
  perl-podlators.noarch 0:2.5.1-3.el7
  perl-threads-shared.x86_64 0:1.43-6.el7
  perl-Carp.noarch 0:1.26-244.el7
  perl-Exporter.noarch 0:5.68-3.el7
  perl-File-Temp.noarch 0:0.23.01-3.el7
  perl-Getopt-Long.noarch 0:2.40-3.el7
  perl-PathTools.x86_64 0:3.40-5.el7
  perl-Pod-Perldoc.noarch 0:3.20-4.el7
  perl-Pod-Usage.noarch 0:1.63-3.el7
  perl-Socket.x86_64 0:2.010-5.el7
  perl-Text-ParseWords.noarch 0:3.29-4.el7
  perl-Time-Local.noarch 0:1.2300-2.el7
  perl-libs.x86_64 4:5.16.3-295.el7
  perl-parent.noarch 1:0.225-244.el7
  perl-threads.x86_64 0:1.87-4.el7

Complete!
[root@localhost ~]#
```

Figure 14: MariaDB installed

To install the server of mariadb, **yum -y install mariadb-server**. To start the server the command follows **systemctl start mariadb.service**. Our database server is running. If we give command **mysql** then, we can enter the MariaDB monitor.

```
Installing : perl-IO-Compress-2.061-2.el7.noarch 4/9
Installing : perl-Net-Daemon-0.48-5.el7.noarch 5/9
Installing : perl-PIRPC-0.2020-14.el7.noarch 6/9
Installing : perl-DBI-1.627-4.el7.x86_64 7/9
Installing : perl-DBD-MySQL-4.023-6.el7.x86_64 8/9
Installing : 1:mariadb-server-5.5.65-1.el7.x86_64 9/9
Verifying : 1:mariadb-server-5.5.65-1.el7.x86_64 1/9
Verifying : perl-Net-Daemon-0.48-5.el7.noarch 2/9
Verifying : perl-Data-Dumper-2.145-3.el7.x86_64 3/9
Verifying : perl-DBD-MySQL-4.023-6.el7.x86_64 4/9
Verifying : perl-IO-Compress-2.061-2.el7.noarch 5/9
Verifying : 1:perl-Compress-Raw-Zlib-2.061-4.el7.x86_64 6/9
Verifying : perl-DBI-1.627-4.el7.x86_64 7/9
Verifying : perl-Compress-Raw-Bzip2-2.061-3.el7.x86_64 8/9
Verifying : perl-PIRPC-0.2020-14.el7.noarch 9/9

Installed:
  mariadb-server.x86_64 1:5.5.65-1.el7

Dependency Installed:
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.el7      perl-Compress-Raw-Zlib.x86_64 1:2.061-4.el7
  perl-DBD-MySQL.x86_64 0:4.023-6.el7              perl-DBI.x86_64 0:1.627-4.el7
  perl-Data-Dumper.x86_64 0:2.145-3.el7            perl-IO-Compress.noarch 0:2.061-2.el7
  perl-Net-Daemon.noarch 0:0.48-5.el7              perl-PIRPC.noarch 0:0.2020-14.el7

Complete!
[root@localhost ~]# systemctl start mariadb.service
[root@localhost ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.65-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Figure 15: MariaDB service installed

We can update the password using SET command, **UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root'**; here, MyNewPass is the new password. When anyone want to enter the mariadb, they need to enter this password.

To reload, the grant table, we can use this command **flush PRIVILEGES**;

```

root@localhost ~]# UPDATE mysql.user SET authentication_string=PASSWORD('MyNewPass') WHERE User='root';
-bash: syntax error near unexpected token '('
root@localhost ~]# UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root';
-bash: syntax error near unexpected token '('
root@localhost ~]# UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE USER='root';
-bash: syntax error near unexpected token '('
root@localhost ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 5.5.65-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> UPDATE mysql.user SET Password=PASSWORD('MyNewPass') Where User='root';
Query OK, 4 rows affected (0.01 sec)
Rows matched: 4  Changed: 4  Warnings: 0

MariaDB [(none)]> flush PRIVILEGES
-> mysql -p
-> CREATE USER 'prince' @'localhost' IDENTIFIED BY 'password';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MariaDB server version for the right syntax to use near 'mysql -p
CREATE USER 'prince' @'localhost' IDENTIFIED BY 'password'' at line 2
MariaDB [(none)]> mysql -p
-> MyNewPass
-> CREATE USER 'prince' @'localhost' IDENTIFIED BY 'password';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MariaDB server version for the right syntax to use near 'mysql -p
MyNewPass
CREATE USER 'prince' @'localhost' IDENTIFIED BY 'password'' at line 1
MariaDB [(none)]> CREATE USER 'prince'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> _

```

Figure 16: Updating password

Then we need to create user by using command “**CREATE USER ‘prince’@‘localhost’ IDENTIFIED BY ‘password’**”; localhost is specify local host which mysql treat this localhost specifically. Here ‘prince’ has no authorization to do anything in the database. It is try to login, it will not able to connect to mysql shell.

Next step is to create a database and giving a name of the database. The command is, **CREATE DATABASE lamp**.

Use lamp

To create a table in database we can use this command, **CREATE TABLE person(Name VARCHAR(64), Age INT, address VARCHAR(128))**; here varchar is a data type which store text. It can store 64 characters in Name, and 128 characters in address.

We can add data by using INSERT command, **insert into person(Name, Age, address) values ('Prince', 26, 'Bangladesh');**

To get one or more privileges, we can use the grant command. This can available on both local ans remote users and the command is **grant SELECT on person to 'prince'@'localhost';**

After creating the database we can exit from this by giving **exit** command.

```

MariaDB [(none)]> CREATE USER 'prince'@'localhost' IDENTIFIED BY 'password';
ERROR 1396 (HY000): Operation CREATE USER failed for 'prince'@'localhost'
MariaDB [(none)]> CREATE USER 'elvis'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE DATABASE lamp
-> ;
ERROR 1007 (HY000): Can't create database 'lamp'; database exists
MariaDB [(none)]> use lamp
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [lamp]> CREATE TABLE person(Name VARCHAR(64), Age INT, address VARCHAR(128));
ERROR 1050 (42S01): Table 'person' already exists
MariaDB [lamp]> insert into person(Name, Age, address) values ('Elvis Gomes', 27, 'Bangladesh');
Query OK, 1 row affected (0.00 sec)

MariaDB [lamp]> grant SELECT on person to 'elvis'@'localhost';
Query OK, 0 rows affected (0.00 sec)

MariaDB [lamp]> exit
Bye
[root@localhost ~]#

```

Figure 17: Creating table

## PHP INSTALLATION:

### Step 3 – Install PHP

To install PHP 7.3 version we can use this command, **yum install php php-common php-opcache php-mcrypt php-cli php-gd php-curl php-mysqlnd**

```
--> Running transaction check
--> Package libX11-common.noarch 0:1.6.7-2.el7 will be installed
--> Package libxcb.x86_64 0:1.13-1.el7 will be installed
--> Processing Dependency: libXau.so.6()(64bit) for package: libxcb-1.13-1.el7.x86_64
--> Running transaction check
--> Package libXau.x86_64 0:1.0.8-2.1.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                        Arch                Version              Repository            Size
=====
Installing:
php                            x86_64              5.4.16-48.el7        base                  1.4 M
php-cli                        x86_64              5.4.16-48.el7        base                  2.7 M
php-common                    x86_64              5.4.16-48.el7        base                  565 k
php-gd                         x86_64              5.4.16-48.el7        base                  128 k
php-mysqld                    x86_64              5.4.16-48.el7        base                  174 k
Installing for dependencies:
libX11                         x86_64              1.6.7-2.el7          base                  607 k
libX11-common                 noarch              1.6.7-2.el7          base                  164 k
libXau                        x86_64              1.0.8-2.1.el7        base                   29 k
libXpm                        x86_64              3.5.12-1.el7         base                   55 k
libjpeg-turbo                x86_64              1.2.90-8.el7         base                  135 k
libxcb                        x86_64              1.13-1.el7           base                   214 k
libzip                        x86_64              0.10.1-8.el7         base                   48 k
php-pdo                       x86_64              5.4.16-48.el7        base                   99 k
t1lib                         x86_64              5.1.2-14.el7         base                   166 k

Transaction Summary
=====
Install 5 Packages (+9 Dependent packages)

Total download size: 6.4 M
Installed size: 22 M
Is this ok [y/d/N]:
```

To secure the copy files, we can use the command `scp student@192.168.3.97:/home/student/* /var/www/html`

```
Installing : php-pdo-5.4.16-48.el7.x86_64 5/14
Installing : libXau-1.0.8-2.1.el7.x86_64 6/14
Installing : libxcb-1.13-1.el7.x86_64 7/14
Installing : libX11-1.6.7-2.el7.x86_64 8/14
Installing : libXpm-3.5.12-1.el7.x86_64 9/14
Installing : t1lib-5.1.2-14.el7.x86_64 10/14
Installing : libjpeg-turbo-1.2.90-8.el7.x86_64 11/14
Installing : php-gd-5.4.16-48.el7.x86_64 12/14
Installing : php-mysqld-5.4.16-48.el7.x86_64 13/14
Installing : php-5.4.16-48.el7.x86_64 14/14
Verifying : libXpm-3.5.12-1.el7.x86_64 1/14
Verifying : libjpeg-turbo-1.2.90-8.el7.x86_64 2/14
Verifying : php-mysqld-5.4.16-48.el7.x86_64 3/14
Verifying : t1lib-5.1.2-14.el7.x86_64 4/14
Verifying : libXau-1.0.8-2.1.el7.x86_64 5/14
Verifying : php-cli-5.4.16-48.el7.x86_64 6/14
Verifying : php-pdo-5.4.16-48.el7.x86_64 7/14
Verifying : php-gd-5.4.16-48.el7.x86_64 8/14
Verifying : libzip-0.10.1-8.el7.x86_64 9/14
Verifying : libX11-common-1.6.7-2.el7.noarch 10/14
Verifying : libxcb-1.13-1.el7.x86_64 11/14
Verifying : php-common-5.4.16-48.el7.x86_64 12/14
Verifying : php-5.4.16-48.el7.x86_64 13/14
Verifying : libX11-1.6.7-2.el7.x86_64 14/14

Installed:
php.x86_64 0:5.4.16-48.el7 php-cli.x86_64 0:5.4.16-48.el7
php-common.x86_64 0:5.4.16-48.el7 php-gd.x86_64 0:5.4.16-48.el7
php-mysqld.x86_64 0:5.4.16-48.el7

Dependency Installed:
libX11.x86_64 0:1.6.7-2.el7 libX11-common.noarch 0:1.6.7-2.el7 libXau.x86_64 0:1.0.8-2.1.el7
libXpm.x86_64 0:3.5.12-1.el7 libjpeg-turbo.x86_64 0:1.2.90-8.el7 libxcb.x86_64 0:1.13-1.el7
libzip.x86_64 0:0.10.1-8.el7 php-pdo.x86_64 0:5.4.16-48.el7 t1lib.x86_64 0:5.1.2-14.el7

Complete!
[root@localhost ~]#
```

Now, we need to check the table we created on the mysql. So, we need to restart the httpd, using the command, **systemctl restart httpd.service**

Lynx is already installed, but we can still give the command **yum -y install lynx**

By giving this command, **lynx** <http://localhost/>, we can see the table we created on database.

## INTRODUCTION:

The necessity of a single platform for fast deployable web applications became necessity not to mention to connecting clients to the servers, Lamp Stack was introduced which included Linux, Apache, MariaDB and PHP. The word Stack comes from the term “solution stack” which is a collection of software for a platform. Michalel Kunze in 1998 introduced the term “LAMP Stack”. In this report the full installation process in how to install and configure Lamp Stack is describes also a step by step command process, furthermore two different variant of Lamp Stack Enos and Mean Stack was discussed. A honourable mentions goes to different versions of lamp stack for different platform such as WIMP and WAMP. A brief discussion on vulnerability, threat / risk assessment and audit is also included in this report. This report also includes the limitations and the potential security risks on the Lamp Stack version used here.

## LITERATURE REVIEW:

In order to perform operations on lamp stack first one must install lamp stack and in (B. Brennen, 2016) show how to install Linux, Apache, MySQL, PHP (LAMP). There are countless experiments on Lamp Stack which measures its performance and compatibility, in (U.V.Ramana, 2005) they have performed several tests to determine the performance of PHP and MySQL in different platforms such as WIMP and WAMP platforms.

Similar to lamp stack Enos Stack is specially designed to reproduce scientific experiments. *“EnosStack, an open source software stack we specially designed to assist experimenters. Deploying real experiments that are repeatable and reproducible remains challenging, and we argued that experimenters should benefit from a LAMP-like stack to help them in running their experiments”*(R.A. Cherrueau, 2018).

There is alos a more advanced version of Lamp stack which is MEAN which runs on extension of JavaScript (AmgulerJS). *“Developing a Web-driven application (either mobile or browser-based) typically requires the provisioning of some server-side infrastructure as well as the development of some code to run on it. Such code will often consume APIs. But occasionally, it provide them as well. For many years, the go-to infrastructure in such situations was affectionately referred to as the LAMP stack and it primarily involved Linux, Apache, MySQL and PHP, Perl or Python. But, thanks in part to Javascript’s applicability to both client and server-side scripting, there’s a another stack that’s now widely considered as an alternative to LAMP; the MEAN stack”* (M. Rajput, 2015). The main difference between Mean and Lamp Stack are on server and language, further more (M. Rajput, 2015) mentioned that in near future Lamp Stack might become obsolete.

For the Mean Stack the main purpose was to deploy a common language for the client and the server. *“As web applications evolved into more single page applications, many development teams needed server-side*

*developers using a language such as PHP, ASP, or Java, and a front-end developer that used JavaScript in the browser client. A common language for both client and server side code would provide an opportunity for quicker development of web applications by smaller teams of developers” (J. Hightower, 2016).*

## LAMP ARCHITECTURE:

In Lamp architecture the first step is the Linux also known as the 1<sup>st</sup> layer, it's the core as all the other elements runs on it. The next layer is the Apache, Apache web server serves more than half of the websites on the internet. It works using HTTP to process and transmit requests over the internet. The 3<sup>rd</sup> layer is MySQL database. It's an database management system that stores application data. The last and the final layer is PHP, which is a programming language and its role is to combine all the elements of Lamp Stack and run web application efficiently.

The process begins when Apache web server receives requests for the webpage from the user's browser. After that PHP uses the code in file and the data from the database. The browser needs this to create the HTML that browsers requires to display web pages. Soon after running the code, PHP then passes the data back to the web server, than it's sent to the browser. While this process is occurring it stores new data in MySQL.

## PART 1

### LAMP STACK INSTALLATION:

1. `systemctl status httpd`
2. `yum -y install net-tools`
3. `netstat -a | less`
4. `yum provides */httpd`
5. `yum -y install httpd`
6. `yum -y install lynx`
7. `lynx httpd://localhost`
8. `systemctl start httpd`
9. `systemctl status httpd`

### MYSQL INSTALLATION:

10. `systemctl status mysql`



11. yum provides \*/mysqld
12. yum provides \*/mysql
13. yum -y install mariadb
14. yum -y install mariadb-server
15. systemctl start mariadb.service
16. mysql
17. UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root';
18. flush PRIVILEGES;
19. CREATE USER 'mariusz'@'localhost' IDENTIFIED BY 'password';
20. CREATE Database lamp;
21. Use lamp
22. CREATE TABLE person(Name VARCHAR(64), Age INT, address VARCHAR(128));
23. insert into person(Name, Age, address) values ('Mariusz Pelc', 15, 'Poland');
24. grant SELECT on person to 'mariusz'@'localhost';
25. exit
26. yum install php php-common php-opcache php-mcrypt php-cli php-gd php-curl php-mysqlnd
27. scp student@192.168.3.97:/home/student/\* /var/www/html
28. systemctl restart httpd.service
29. yum -y install lynx
30. lynx <http://localhost/>

## PART 2:

### VULNERABILITY REPORT:

Web servers are the most used thing in the internet and the applications are running in the web servers such as PHP, JAVA and Python. Because of this web servers are flexible. Hackers are trying to infiltrate the web servers frequently, sometimes the web pages becomes unavailable causing instructions to face financial losses, failing to provide the valuable service that provide. So in order to make the server more secure system admins needs to install or add additional security for the web servers. This is where Apache comes in and its available in Linux and windows.

The very 1<sup>st</sup> thing we need to look when installing Apache is the version one must install the latest version of Apache alone with keeping an eye out for any additional updates. If the version is not properly updated attackers can exploit it and damage the system.

```
[root@localhost ~]# httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   Apr  2 2020 13:13:23
```

There is one other key vulnerabilities of the Apache and that is visibility of the version number. Since attackers are looking every chance to harm the system.

The flexibility of the HTTPD is also one of its core weakness. Since older technologies can be easily replaced with newer technologies which makes it harder to maintain its security. The reason is the Apache has to change the service configuration all over again in order to enable this updates. It's like a double edged sword, either sacrifice security for flexibility or the other way around.

There should be only one directory for all the websites and if the admin fails to do that any unauthorized person can get aces to the root files.

Another important vulnerability of the Lamp Stack is SQL injection (S.C. Wu, 2008). In (Z. Su, 2006) all the aspects of SQL injection is briefly explained. “*SQL injection [2] is a security loophole in the database layer of an application. It occurs when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed*” (S.C. Wu, 2008). After the installation of MySQL the admin needs to change the password and preventing unauthorized individuals to get access to the database and resulting data theft.

### RISK ASSESSMENT:

The very first task starts when installing Apache the system administrator needs to check the version of Apache since different version have different kinds of flaws and those can be exploited easily. If the version is out-dated the system admin needs to do the following steps.

1<sup>st</sup> upgrade the server and by doing so the commonly known exploits can be prevented. There are one main and two sub version of the Apache since any kind of attempt of breaching the server is done on a specific version other sub version won't be affected by this.

Installing an Apache server from a source, not changing the information option or the redistribution of the package is only to install for HTTP protocol but if anyone want to install the SSL that person need to install yum from the installation package.

There is also another core risk of the HTTPS and which is flexibility the of HTTPS if it becomes flexible it has to compromise its security or the other way if its more secure it won't be flexible.

*“Buffer overflow [1] is a programming error which may result in abnormal program behavior such as memory access exception, program termination, or a possible breakdown of system security” (S.C. Wu, 2008).* Buffer appears when the data is overflowed and crossed the boundary and as a result data from other source gets over written.

## AUDIT REPORT:

Exploring the vulnerabilities and solving them is the most appropriate ways to make a system secure.

To begin with a secure password for the system and web server one should make the password as complex as possible.

The Second task should be to change the .htaccess hidden files to root files to prevent hacker's from changing the system level.

Thirdly in the PHP section admin show hide the PHP version and later disabling the index listing form PHP. In the Apache we need to enable the Cross-site-scripting projection in the system. We can then enable the firewall to input extra layer of security.

To prevent SQL injection SQL statements should not and must not be embedded directly and let the user input escape or parameterized statements can be used as well. However there is a flaw in it as well since web applications are programmed by different programmers these steps such as use of escapes and parameterized SQL statements can't be guaranteed.

## CONCLUSION:

The main objective of the course work is to understand LAMP Stack how it works and how install and configure it. There is also discussions about different variant of Lamp, not to mention the possible risks and the vulnerability with the intention of how to fix them. Even though there many updated versions of Lamp and the variants such as Mean Stack is much more refined compare to Lamp which is outdated. But in the given scenario Lamp seems to be much suited if the possible risk and lacking can be fixed which is mentioned in the report and can provide a better Lamp Stack all together.

## REFERENCES:

Brennen Bearnes, 2016. How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04. Digital Ocean 1–30.

Cherrueau, R.A., Simonin, M., Van Kempen, A., 2018. EnosStack: A LAMP-like stack for the experimenter, in: INFOCOM 2018 - IEEE Conference on Computer Communications Workshops. Institute of Electrical and Electronics Engineers Inc., pp. 336–341. doi:10.1109/INFCOMW.2018.8407024\

Hightower, J., 2016. Mean: a full JavaScript stack for web development: conference tutorial. Journal of Computing Sciences in Colleges 31, 109–110.

Kuperman, B.A., Brodley, C.E., ... Jalote, A., 2005. Detection and prevention of stack buffer overflow attacks. Communications of the ACM. doi:10.1145/1096000.1096004

Programmableweb (2021) “What is Mean Stack and why it is better” Available at: <https://www.programmableweb.com/news/what-mean-stack-and-why-it-better-lamp/analysis/2015/12/22>

Rajput, M., 2015. What is the MEAN Stack and Why is it Better than LAMP? | ProgrammableWeb [WWW Document]. Programmable Web. URL <http://www.programmableweb.com/news/what-mean-stack-and-why-it-better-lamp/analysis/2015/12/22>

Ramana, U.V., Prabhakar, T.V., 2005. Some experiments with the performance of LAMP architecture, in: Proceedings - Fifth International Conference on Computer and Information Technology, CIT 2005. pp. 916–920. doi:10.1109/CIT.2005.169

Su, Z., Wassermann, G., 2006. The essence of command injection attacks in web applications. ACM SIGPLAN Notices 41, 372–382. doi:10.1145/1111320.1111070

Stackify (2021) “what is Lamp Stack” Available at: <https://stackify.com/what-is-lamp-stack/>

Wu, S.C., Ku, Y.C., Lee, T.L., 2008. Zero-configuration security module for LAMP application system, in: Proceedings of the 3rd IEEE Asia-Pacific Services Computing Conference, APSCC 2008. IEEE Computer Society, pp. 113–116. doi:10.1109/APSCC.2008.242