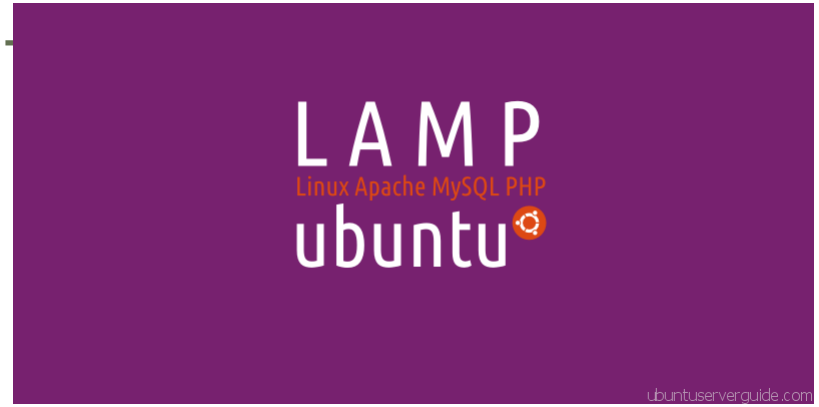
UEEN 3113 / 3413

SERVER CONFIGURATION AND MANAGEMENT



SSH (Secure Shell)

- A suite of network communication tools that allows users to connect to a remote server with encrypted session.
- There are 2 packages for SSH in Ubuntu (other distributions might use different name for the packages):
  - openssh-server
  - openssh-client (usually installed by default)
- We will install openssh-server in the server.
- Any client that needs to connect to server using SSH must install openssh-client.

SSH (Secure Shell)

- To check if the SSH server is running / installed:
  - service ssh status *OR*
  - systemctl status ssh *OR*
  - which sshd

- If SSH is disabled, we can enable it by:
  - systemctl enable ssh

SSH (Secure Shell)

- To check the listening port (default port is 22):
  - netstat -tulpn | grep ssh

```
user@u-server:~$ sudo netstat  -tulpn | grep ssh
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1548/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      1548/sshd
user@u-server:~$ _
```

- To connect to server / other machine, using currently logged in username:
  - ssh host_name *OR* ssh ip_address
  - Example: ssh u-server

SSH (Secure Shell)

```
user@u-server2:~$ ssh u-server
user@u-server's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

90 packages can be updated.
41 updates are security updates.


Last login: Wed Mar  7 10:18:42 2018 from 192.168.30.102
user@u-server:~$ _
```

- To connect with other username:
  - ssh username@host_name *OR* ssh username@ip_address

SSH (Secure Shell)

- If the ssh is listening to a non-standard port (other than 22), we need to specify the port number with the -p option.
  - ssh -p port_number username@hostname
  - Example, ssh –p 13300 user2@u-server
- To disconnect: enter the **exit** command or press Ctrl+D.
- If we started background processes on target machine via ssh, use Ctrl+D to end the session, otherwise the processes will be terminated.

SSH (Secure Shell)

- Notice that we are asked for password when we connect to server / machine, which means the password will be transmitted during the connecting process. (Is this risky?)
- It's better to disable **password authentication** and implement **Public Key Authentication**.

SSH (Secure Shell)

- First, we nee to generate a public/private key pair .
  - ssh-keygen

```
user@u-server2:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:0SVi6sF5XyqUjIUuMq8TS00Mj6X4dpGf6RxDygItiRA user@u-server2
The key's randomart image is:
+---[RSA 2048]----+
|E.     .+ . .    |
|. . ...B + o     |
|o+ B o* * . .    |
|* * B.o+ o o     |
| + B *.oS o      |
|  * * *  .       |
| o * o o         |
|  +   o          |
|   .             |
+----[SHA256]-----+
user@u-server2:~$
```

SSH (Secure Shell)

- The passphrase can be empty but it's recommended to provide one, which should be different from system password.
- 2 files will be generated and stored in the default directory (/home/user_name/.ssh)
  - id_rsa (private key)
  - id_rsa.pub (public key)
- Private key (id_rsa) should never leave the machine, be given to someone or stored on external storage media.

SSH (Secure Shell)

- The public key is to be copied to target servers / machines that we wish to connect to using ssh.
  - ssh-copy-id -i ~/.ssh/id_rsa.pub host_name
  - Example: ssh-copy-id -i ~/.ssh/id_rsa.pub u-server

```
user@u-server2:~$ ssh-copy-id -i .ssh/id_rsa.pub u-server
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
user@u-server's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'u-server'"
and check to make sure that only the key(s) you wanted were added.

user@u-server2:~$
```

SSH (Secure Shell)

- In the target server / machine, .ssh directory will be created to store the public key in a file named **authorized_keys**
- If we connect from multiple machines (a key is generated by each machine), additional keys will be appended to the bottom of this file, one per line.

SSH (Secure Shell)

- To simplify ssh connection, we can create a **config** file in .ssh directory to store all necessary information for ssh connection.
  - nano ~/.ssh/config
- Content of config file:

  host *host_name* (*host_name* here serves as shortcut)

  Hostname actual_host_name *OR* ip_address

  Port port_number

  User user_name

SSH (Secure Shell)

- Example of *config* file:

```
host server1
    Hostname u-server
    Port 22
    User user
```

```
  GNU nano 2.5.3                                  File: config

host server1
        Hostname u-server
        Port 22
        User user
```

- To connect to u-server, with *config* file created:

```
ssh server1
```

SSH (Secure Shell)

- Configuration file for ssh daemon is /etc/ssh/sshd_config
- It’s safer to create a backup copy of the configuration file before we change the configuration.
- Some suggestions to secure the ssh
  - Change number that's ab ther service).

```
  GNU nano 2.5.3                    File: /etc/ssh/sshd_config

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
#Port 22
Port 12233
```

SSH (Secure Shell)

- Some suggestions to secure the ssh
  - Change the port number (preferably a high number that's above 10000 and not in use by any other service).
  - Make sure that ssh is listening to **Protocol 2** (should be default in newer server)
  - Allow only specified users / groups to connect via ssh (by default every user created is allowed) by adding the following sections in config
    - AllowUsers users_separated_by_a_space
    - AllowGroups groups_separated_by_a_space

SSH (Secure Shell)

- Some suggestions to secure the ssh
  - Examples for AllowUsers and AllowGroups
    - AllowUsers user user2 john
    - AllowGroups admins sshuser
  - Turn off PermitRoorLogin
    - Default setting is *prohibit-password*, means key authentication is allowed for root but passwords for root aren't accepted
    - To turn off, replace *prohibit-password* with *no*

```
#PermitRootLogin prohibit-pa
PermitRootLogin no
```

SSH (Secure Shell)

- Some suggestions to secure the ssh
  - Disable password authentication after key authentication has been setup successfully

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no
```

- Restart the ssh service whenever we change the configuration.
  - systemctl restart ssh

SSH (Secure Shell)

- If firewall is enable, make sure that we enable traffic via SSH.
- To check if firewall is enabled and running in Ubuntu:
  - sudo ufw status
- To allow traffic from specific machine to SSH port (TCP and UDP):
  - ufw allow from *ip_address* to *any* port *port_number*
- To allow traffic from a subnet:
  - ufw allow from *subnet* to *any* port *port_number*


OK,
SO WHAT'S
NEXT?