Implementing a Secure SSH Server

(CompTIA Security + SY – 601)

Objectives:

- To implement secure network designs
- To use the appropriate tool to assess organizational security

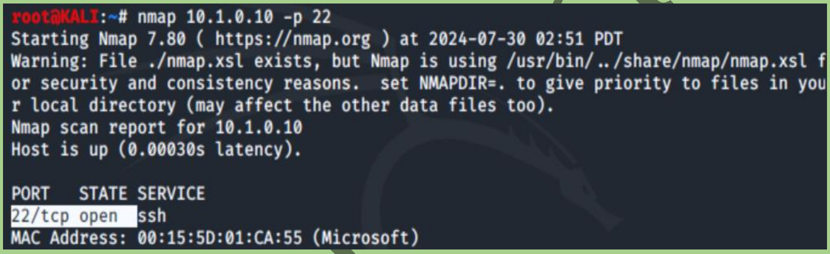
Resources:

- Kali Virtual Machine (PT1-Kali)
- LX SSH server
- Centos Server

Instructions:

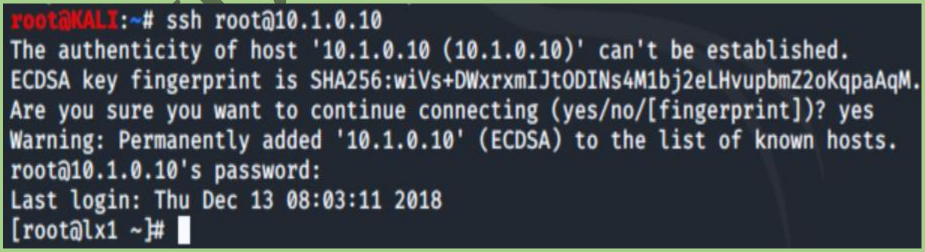
Remotely connect by using SSH with basic Password Authentication

- Log-in in to the **PT1-Kali** VM
- From the top bar, select the **Terminal** Emulator icon
- Run the following command to check the status of port 22 on the remote LX1 SSH server **nmap 10.1.0.10 -p 22**



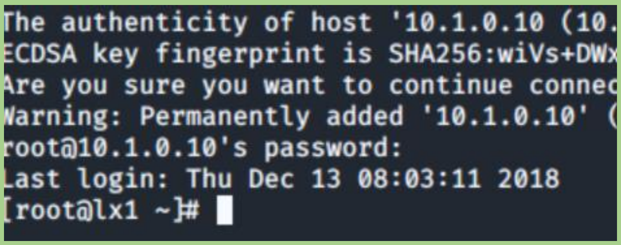
```
root@KALI:~# nmap 10.1.0.10 -p 22
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-30 02:51 PDT
Warning: File ./nmap.xsl exists, but Nmap is using /usr/bin/../share/nmap/nmap.xsl f
or security and consistency reasons.  set NMAPDIR=. to give priority to files in you
r local directory (may affect the other data files too).
Nmap scan report for 10.1.0.10
Host is up (0.00030s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:15:5D:01:CA:55 (Microsoft)
```

- Run the following command to test SSH connectivity to the CentOS SSH server: **ssh root@10.1.0.10**



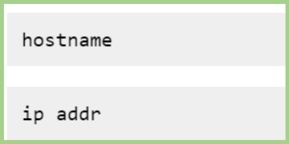
```
root@KALI:~# ssh root@10.1.0.10
The authenticity of host '10.1.0.10 (10.1.0.10)' can't be established.
ECDSA key fingerprint is SHA256:wiVs+DWxrxmIJtODINs4M1bj2eLHvupbmZ2oKqpaAqM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.1.0.10' (ECDSA) to the list of known hosts.
root@10.1.0.10's password:
Last login: Thu Dec 13 08:03:11 2018
[root@lx1 ~]# █
```

- Enter Yes when prompted to confirm the connection. This is the host key, which validates the identity of the server
- Enter **Pa$$w0rd** when prompted for the password. This is the password for a user account with local logon that is authenticated by the SSH server



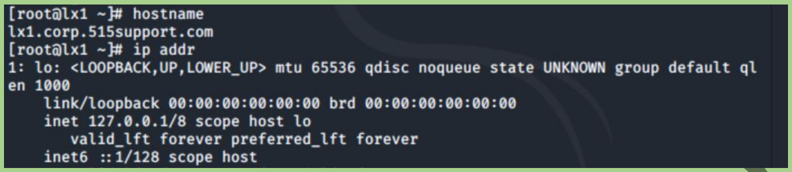
```
The authenticity of host '10.1.0.10 (10.
ECDSA key fingerprint is SHA256:wiVs+DWx
Are you sure you want to continue connec
Warning: Permanently added '10.1.0.10' (
root@10.1.0.10's password:
Last login: Thu Dec 13 08:03:11 2018
[root@lx1 ~]# 
```

- Run the following commands to verify that you are connected to the remote SSH server



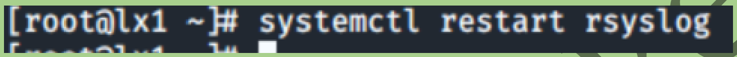
```
hostname
```

```
ip addr
```



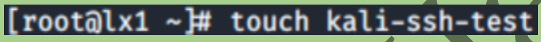
```
[root@lx1 ~]# hostname
lx1.corp.515support.com
[root@lx1 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
en 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
```

- Run the following command to prove you have administrative privileges on the remote server: **systemctl restart ryslog**



```
[root@lx1 ~]# systemctl restart rsyslog
[root@lx1 ~]# 
```

- Use **touch** command to create a new file named **kali-ssh-test** in the root user home directory on the remote CentOS server

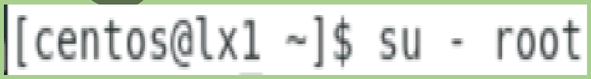


```
[root@lx1 ~]# touch kali-ssh-test
```

- Type exit to disconnect from the remote CentOS server

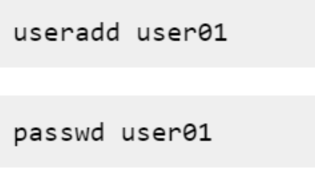
Configure the Centos SSH server

- Switch to the LX1 SSH Server
- The VMs privacy screen is probably enabled-click anywhere on the screen with the mouse and press **ENTER**. A login prompt should appear
- Sign-in with the preconfigured creator account and a Password of **Pa$$w0rd**
- Right-click the desktop and select **Open Terminal**
- Run the following command to elevate your credentials to root: **su- root**

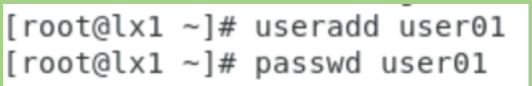


```
[centos@lx1 ~]$ su - root
```

- Type **Pa$$w0rd** when prompted
- Run the following commands to create a new user and set a password:

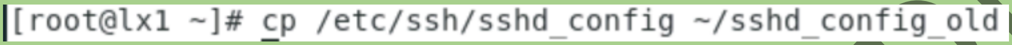


```
useradd user01

passwd user01
```



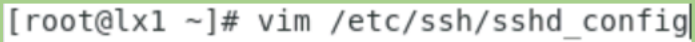
```
[root@lx1 ~]# useradd user01
[root@lx1 ~]# passwd user01
```

- ➢ Set the password as Pa$$word when prompted. You may ignore any warnings about the password quality
- ➢ Run the following command to back up the current configuration file: **cp /etc/ssh/sshd_config ~/sshd_config_old**



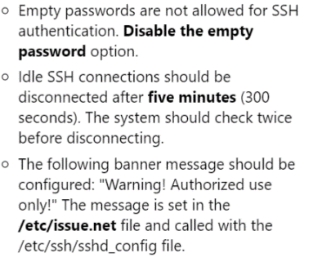
```
[root@lx1 ~]# cp /etc/ssh/sshd_config ~/sshd_config_old
```

- ➢ Run the following command to open the SSH configuration file with the vim text editor: **vim /etc/ssh/sshd_config**



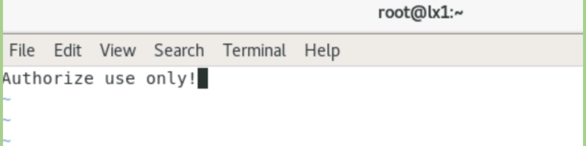
```
[root@lx1 ~]# vim /etc/ssh/sshd_config
```

- ➢ Review the following list of the required SSH configurations defined by your company's written security policy:



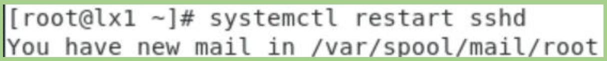
- ○ Empty passwords are not allowed for SSH authentication. **Disable the empty password** option.
- ○ Idle SSH connections should be disconnected after **five minutes** (300 seconds). The system should check twice before disconnecting.
- ○ The following banner message should be configured: "Warning! Authorized use only!" The message is set in the **/etc/issue.net** file and called with the /etc/ssh/sshd_config file.

- ➢ In the test editor, edit the **/etc/ssh/sshd_config** file to match the required settings. You will need to uncomment some lines and then edit them. Other lines you may simply edit
- ➢ In the Vim, save your changes by pressing **ESC** and then typing: **wq**
- ➢ Use vim to edit the **/etc/issue.net** file. Remove all of the existing content in the file and then add the following warning: **Authorized use only!**



```
Authorize use only!
~
~
~
```

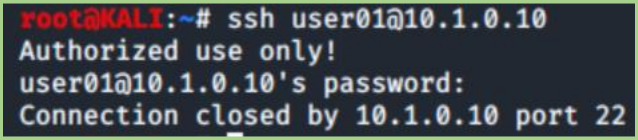
- ➢ To save changes and quit vim, press ESC and then type **:wq**
- ➢ In the command prompt window run the following command to restart the SSH service: **systemctl restart sshd**



```
[root@lx1 ~]# systemctl restart sshd
You have new mail in /var/spool/mail/root
```

Test the new security configurations

- Switch to the **PT1-Kali SSH client** VM
- Attempt the SSH connection by using the user01 credentials. Use **Pa$$w0rd** as password

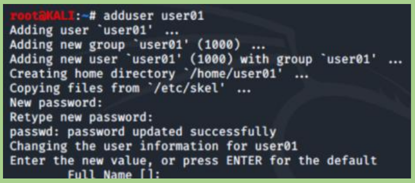


```
root@KALI:~# ssh user01@10.1.0.10
Authorized use only!
user01@10.1.0.10's password:
Connection closed by 10.1.0.10 port 22
```

- Type exit to disconnect from the remote **LX1 SSH** server

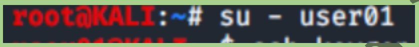
SSH Key-based authentication

- On the PT1-Kali VM confirm the command prompt is root@KALI to ensure that you do not currently have an SSH connection to the remote LX1 SSH server
- Run the following commands to create new user named user01 with Pa$$w0rd as the password: **adduser01**



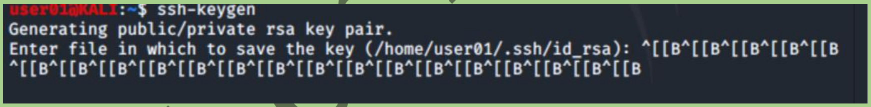
```
root@KALI:~# adduser user01
Adding user `user01' ...
Adding new group `user01' (1000) ...
Adding new user `user01' (1000) with group `user01' ...
Creating home directory `/home/user01' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user01
Enter the new value, or press ENTER for the default
        Full Name []:
```

- Run the following command to switch to the new user01 user



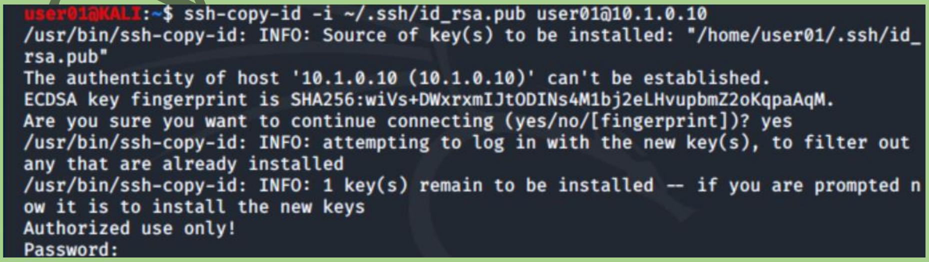
```
root@KALI:~# su - user01
```

- Run the following command to generate an SSH keypair that defines user01: **ssh-keygen**



```
user01@KALI:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user01/.ssh/id_rsa): ^[[B^[[B^[[B^[[B^[[B
^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B
```

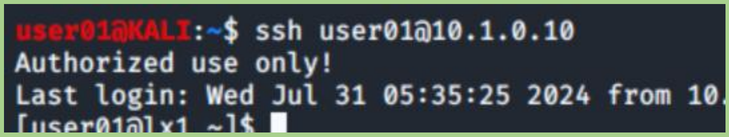
- Press **ENTER** three times to accept the default settings
- Run the following command to copy the public key to the SSH server



```
user01@KALI:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub user01@10.1.0.10
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user01/.ssh/id_
rsa.pub"
The authenticity of host '10.1.0.10 (10.1.0.10)' can't be established.
ECDSA key fingerprint is SHA256:wiVs+DWxrxmIJtODINs4M1bj2eLHvupbmZ2oKqpaAqM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted n
ow it is to install the new keys
Authorized use only!
Password:
```

- Enter Yes when prompted. Enter Pa$$word when prompted

- Run the following command to connect to the SSH server and confirm that the key-based authentication works**: ssh** [user01@10.1.0.10]()

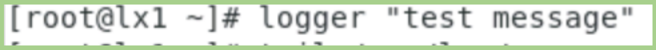


```
user01@KALI:~$ ssh user01@10.1.0.10
Authorized use only!
Last login: Wed Jul 31 05:35:25 2024 from 10.
[user01@lx1 ~]$ 
```

- Type exit to terminate the SSH connection

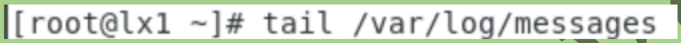
Review SSH log files

- Switch to the **LX1** VM
- If a terminal window is not already open, right-click on the desktop and select **Open Terminal**
- Run the following command to generate an entry in the /var/log/messages log file: **logger “test message”**



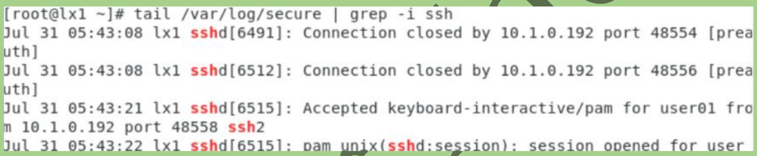
```
[root@lx1 ~]# logger "test message"
```

- Run the following commands to view the lines at the bottom of the tail **/var/log/messages**



```
[root@lx1 ~]# tail /var/log/messages
```

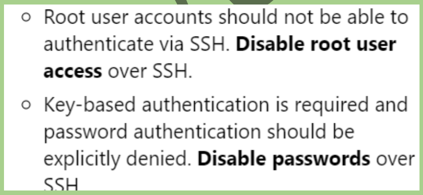
- Run the following command to view SSH events in the log file: **/var/log/secure | grep -I ssh**



```
[root@lx1 ~]# tail /var/log/secure | grep -i ssh
Jul 31 05:43:08 lx1 sshd[6491]: Connection closed by 10.1.0.192 port 48554 [prea
uth]
Jul 31 05:43:08 lx1 sshd[6512]: Connection closed by 10.1.0.192 port 48556 [prea
uth]
Jul 31 05:43:21 lx1 sshd[6515]: Accepted keyboard-interactive/pam for user01 fro
m 10.1.0.192 port 48558 ssh2
Jul 31 05:43:22 lx1 sshd[6515]: pam_unix(sshd:session): session opened for user
```

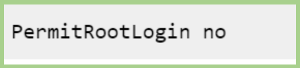
Prevent root from authenticating over SSH

- Review the following list of the required SSH configurations defined by your company’s written policy



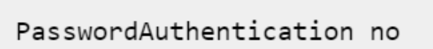
- Root user accounts should not be able to authenticate via SSH. **Disable root user access** over SSH.
- Key-based authentication is required and password authentication should be explicitly denied. **Disable passwords** over SSH.

- On the LX1 VM use the Vim test editor to open the **/etc/ssh/sshd_config file**
- Uncomment the following line by deleting the # character and then change the entry from **“yes”** to **“no”** to prevent the user from signing-in via SSH:



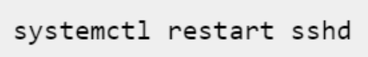
```
PermitRootLogin no
```

- Change the entry on the following line from **“yes”** to **“no”** to prevent password-based authentication



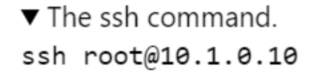
```
PasswordAuthentication no
```

- Select **Esc** and then type: **wq** to save you changes exit vim
- Run the following command to restart the SSH service causing it to re-read the configuration file and apply the new settings: **systemctl restart sshd**



```
systemctl restart sshd
```

- Switch to the **PT1-Kali** VM
- Attempt the SSH connections by using the **root** credentials



▼ The ssh command.

```
ssh root@10.1.0.10
```

- Attempt the SSH connection by using the **user01** credentials

Observations:

- The SSH server on the remote LX1 was successfully configured and tested with basic password authentication.
- Administrative privileges were confirmed on the remote server by successfully restarting the rsyslog service.
- Key-based authentication was successfully implemented and verified for the user user01.
- SSH log files were reviewed, and relevant SSH events were identified and analyzed

Results:

- **Objective 1: Implement secure network designs**

- Successfully configured SSH server with secure settings as per the company's security policy.

- **Objective 2: Use the appropriate tool to assess organizational security**

- Used `nmap` and SSH commands to test and verify the SSH server's configuration and connectivity.

Conclusion:

The lab successfully demonstrated the implementation and configuration of a secure SSH server, achieving both secure network design and organizational security assessment objectives. Future work will focus on enhancing security measures, regular auditing, and user education to maintain and improve SSH server security.

Future Work:

- Implement two-factor authentication (2FA) for SSH access to further enhance security.
- Automate regular audits of SSH configurations and access logs to detect and respond to any anomalies or unauthorized access attempts promptly.
- Regularly update and patch the SSH server to mitigate any vulnerabilities and ensure compliance with the latest security standards.
- Develop and implement a comprehensive user training program on secure SSH practices and password management to minimize human errors.