Lab 8 - Remote Management

Remote Management:

Remote management in the context of Linux refers to the ability to administer and control a Linux system from a remote location, often over a network. This is particularly important for servers, where physical access might be limited or impractical. Remote management allows system administrators to perform various tasks, monitor the system, and troubleshoot issues without being physically present at the machine.

In order to understand Remote Management, let’s firstly go through some of the fundamentals of networking

Fundamentals of Networking

Networking is a really vast concept and in the coming semesters, you’ll get to know about everything in more detail, for now, let’s just focus on the following few things

1. Protocols
2. IP Addresses
3. Ports
4. Schemas

Understanding of the above 4 concepts is a must for remote management.

1. Protocols:

Think of protocols as sets of rules that devices use to communicate with each other. Imagine you and a friend are talking on the phone. To make sure you understand each other, you might have some agreed-upon rules, like saying "hello" to start the conversation or "goodbye" to end it. In the same way, computers and devices follow protocols to communicate effectively.

Normally, there are 2 (and one another) protocols that are most commonly used:

1. Transmission Control Protocol [TCP]

   TCP is a simple protocol, this is used where we want to ensure that data is transferred without being messed up/changed.

2. User Datagram Protocol [UDP]

   UDP protocol is used where we need to make sure that data is sent, but we don’t care about whether it is sent properly or not.

3. HyperText Transmission Protocol [HTTP]

   HTTP protocol is used by web applications in order to establish communication between the client and the server.

2. IP Addresses:

An IP address is like a home address for your computer on the internet. When you want to send or receive information, it needs to find its way to the right place. Just as your home has a unique address, your computer has a unique IP address that helps data find its way to and from your device on the internet. To make it simpler, there are two kind of IP Addresses:

1. Public
2. Private

Public IP Addresses are the IP Addresses that can be used to identify you over the internet meaning that over the world, you’ll be known by this ip address. However, a private address is what your IP Address is on the local network. Normally, the Private IP Addresses begin with: `10.x.x.x`, `172.x.x.x` or `192.x.x.x`. There are classes to IP Addresses but are out of the scope of this course.

3. Ports:

Think of ports like different doors in a building. Imagine your computer is a big office building, and each application or service running on your computer is like an office with its own door. Ports are the doors that allow data to come in or go out. When you use the internet, different applications use different doors (ports) to send and receive information, making sure everything goes to the right place.

Each specific application use a different port to `listen` on. We will talk about ports later.

4. Schemas:

Schemas are like templates or blueprints that help computers understand how to organize and interpret information. Imagine you're reading a recipe. The ingredients, steps, and cooking times are all organized in a specific way so you can follow them

easily. Similarly, computers use schemas to understand the structure of data, making it easier for them to process and use the information correctly.

Normally, the schemas are as follows:

```
http://
tcp://
udp://
file://
ws:// # ws stands for WebSockets
```

Each schemas is: `protocol` followed by `://`

Softwares used for Remote Management

There are certain softwares that are used in order to manage servers remotely, these are:

1. SSH
2. SCP
3. FTP
4. SAMBA/SMB
5. RDP

1. SSH

SSH stands for Secure SHell. It is like a secure tunnel that allows you to connect and communicate with another computer (usually a server) over the internet. It's like a secret passage that ensures your conversations and data are protected from prying eyes. It is most commonly used to connect to remote servers in order to perform a certain task. SSH uses `TCP` protocol to connect to hosts.

In order to connect to a server, `OpenSSH/sshd` which is the server which handles SSH traffic must be running on the remote server.

`ssh` command is used to connect to a remote server.

→ Usage:

```
ssh <username>@<ip/host> [-p PORT] [-i CERT]
```

By default, in order to connect to a server, you need to know the following things:

1. username
2. IP/Hostname

Now, to authenticate with the server, there are two ways:

1. Using a password

   This way, you will simply use a password (which is not echoed back to you) in order to connect to a remote instance

2. Using a certificate

   Instead of a password, think of a certificate like a special key. If your computer has the right key (certificate), the server lets it in. It's a more secure way to prove identity than just a password. This is done by providing `-i` option to the SSH command along with the path to where the certificate is.

→ Quick Exercise:

Connect to the following remote instance:

```
ssh user@139.59.77.99 -p 1523
# use password: user
```

Then, once you are connected, run the following command:

```
cat flag.txt
```

Well, a quick `flag.txt` ;)

One more thing, by default, SSH uses port `22`. But it can be changed by modifying `/etc/ssh/sshd_config` file.

2. SCP

SCP is like a secure delivery service for files. It allows you to securely copy files between your computer and a remote server over the internet, ensuring that your files are safe during the transfer. The Syntax of SCP command is the same as SSH command, but the only difference is:

```
# In order to transfer a file from your system to remote:
# echo "THIS_IS_A_FILE" > my-file
```

```
scp [-P PORT] [-i CERT] <local-file> <username>@<ip/host>:<path/on/remote>

## Example:
scp -P1523 my-file user@139.59.77.99:/tmp/my-file
### Now, when you will ssh again, you will see that /tmp/my-file exists.
```

Now, SCP can also be used to download files from the remote server as well:

```
scp [-P PORT] [-i CERT] <username>@<ip/host>:<path/on/remote> <local-file>

## Example:
scp -P 1523 user@139.59.77.99:/home/user/check.txt ./check.txt
## This will download the file check.txt from remote to local.
```

3. FTP

FTP is like a traditional courier service for files. It provides a way to transfer files between your computer and a remote server over the internet. However, unlike SCP, FTP is not inherently secure, and additional measures may be needed to protect data during transfer. FTP is used in scenarios where file transfer is required, but it's important to note that the basic FTP protocol doesn't provide encryption for the data being transferred. It's often used in less security-sensitive environments. For example:

- Updating a website by uploading files to a web server.
- Sharing large files between computers.

There are two authentication modes in FTP:

1. Connecting using a username and password

   This is the generic way of connecting to any server, you provide username and password; it validates and authenticates you.

2. Anonymous Login

   In Anonymous Login, no credentials are required, the username is `anonymous` and the password is nothing. This is by-default disabled (Duh!, a big security risk.)

In order to connect to FTP server, there are several ways. Most commonly:

1. FileZilla (A GUI based client)
2. FTP CLI

Since we are l33t, we'll be using the FTP CLI. In order to connect to a server, we use the following syntax:

```
ftp <server-name>
## It will then ask you for your username and password:

ftp 139.59.77.99
# username: myuser
# password: mypass
```

Once you have successfully accessed ftp, you can simply write the following command to download a file:

```
dir # This will list the files
get <file-name>
## Example: get myfile.txt
```

Along with downloading, you can also upload a file using `put`

```
put <file-name>
## Example: put test.txt
## test.txt must be present in the current directory.
```

4. SMB

SMB stands for Server Message Block. SMB is like a shared workspace where different computers in a network can collaborate and share files with each other. It's a protocol that allows devices to communicate and share resources, like files and printers, on a local network. SMB is commonly used in local network environments, such as offices or homes, where multiple devices need to share files and resources. For example:

- Sharing files between computers in an office.
- Accessing shared folders on a network-attached storage (NAS) device.

Similar to FTP, we can connect either with valid credentials or with anonymous credentials. However, in SMB, anonymous login is called `NULL Authentication`, meaning we don't pass any credentials.

In order to interact with an SMB Server, we use `smbclient` on linux and simple `Network` on Windows.

```
## Usage of smbclient:
smbclient [-L] //<host>/ [-U USER]
```

In SMB, we need to firstly list available shares, we can do that by using:

```
smbclient -L //139.59.77.99/
```

This will list down all available shares. There's another tool called `smbmap` which tells whether the share is readable/writable.

After running the command, we will get a somewhat similar output:

```
Sharename       Type      Comment
---------       ----      -------
share           Disk      Public
secure-share    Disk      Authenticated Share.
IPC$            IPC       IPC Service (Samba Server)
```

Here, we can see that we have two `shares`. A share is just a like folder that multiple people can access over a network and make reflective changes to.

We have a `share`, which has a comment `Public` and the other `secure-share`, which has a comment `Authenticated Share`. In order to connect to `share`, we run the following command:

```
smbclient //139.59.77.99/share
```

When you're prompted to enter password, just press enter. After this, you will get a prompt similar to that of FTP, go ahead and read `README.txt`.

Now, in order to connect to `secure-share`, we will use the following credentials:

```
user:pass
```

To connect:

```
smbclient //139.59.77.99/secure-share -U user
# -U option represents user
```

Once again, a similar prompt will be received like FTP and you will have to get the `README.txt`.

5. RDP

RDP stands for Remote Desktop Protocol. RDP is like having a magic mirror that allows you to see and control another computer as if you were sitting in front of it. It's a protocol that enables you to connect to a remote computer and interact with its desktop as if it were your own. RDP is commonly used in scenarios where you need to access and control a computer remotely. For example:

- IT support personnel remotely assisting users with technical issues.
- Accessing your work computer from home.

By default, RDP is used when interacting with Windows Servers. In order to connect to a server using RDP, we use different clients. These include (but are not limited to):

→ CLI:

- xfreerdp
- rdp
- rdesktop

→ GUI:

- Remmina
- TigerVNC
