**LAB MANUAL**

**CLOUD SECURITY**

**Lab1:**Install Oracle Virtual box and create two VMs on your laptop/Desktop.

Aim: To install Oracle VirtualBox on a laptop/desktop and create two virtual machines (VMs).

**Prerequisites:**

1. A laptop or desktop with at least **8GB RAM**, **50GB free disk space**, and **a multi-core processor**.
2. A stable internet connection.
3. Downloaded installation files for **Oracle VirtualBox** and an operating system (OS) ISO file (such as Ubuntu or Windows 10)

**Step 1: Install Oracle VirtualBox**

1. **Download VirtualBox:**
   * Visit the official website: <https://www.virtualbox.org/>
   * Download the appropriate version for your OS (Windows, macOS, Linux).
2. **Install VirtualBox:**
   * Open the downloaded installer and follow the installation wizard.
   * Select the default settings and click **Next**.
   * Allow VirtualBox to install network drivers when prompted.
   * Click **Install** and wait for the process to complete.
   * Once done, launch VirtualBox.

**Step 2: Create a Virtual Machine (VM)**

1. **Open VirtualBox** and click on **New**.
2. **Enter VM Name** and select OS type:
   * Example: Name: **Ubuntu\_VM**
   * Type: **Linux** (or **Windows** for Windows VMs)
   * Version: **Ubuntu (64-bit)**
3. **Allocate Memory (RAM):**
   * Recommended: At least **2GB (2048 MB)** for Linux and **4GB (4096 MB)** for Windows.
   * Click **Next**.
4. **Create Virtual Hard Disk:**
   * Select **Create a virtual hard disk now**.
   * Choose **VDI (VirtualBox Disk Image)** and click **Next**.
   * Select **Dynamically allocated** and set at least **20GB** storage.
   * Click **Create**.
5. **Attach OS ISO File:**
   * Select the created VM and click **Settings**.
   * Go to **Storage** > **Controller: IDE** > Click **Empty**.
   * Click **Choose a disk file** and select the downloaded OS ISO.
   * Click **OK**.

**Step 3: Install the OS on the VM**

1. Select the VM and click **Start**.
2. The OS installation wizard will appear; follow the steps to install it.
3. Configure user settings, passwords, and system preferences.
4. Once installation is complete, remove the ISO from the VM’s settings.
5. Restart the VM and ensure it boots into the installed OS.

**Step 4: Create a Second Virtual Machine**

1. Repeat **Step 2** using a different OS or another instance of the same OS.
2. Follow **Step 3** to install the OS on the second VM.

**Troubleshooting Tips:**

* **VM won’t start?** Ensure virtualization is enabled in BIOS.
* **Low performance?** Increase RAM or CPU cores from VM settings.
* **Networking issues?** Set network adapter to **Bridged** or **NAT**.

**Lab 2:** Test ping command to test the communication between the guest OS and Host OS .

**Aim:** To test and verify network communication between the Guest and Host OS using the ping command.

**Theory:**

The ping command is used to test network connectivity by sending ICMP Echo Request packets to the target machine and measuring the response time. This helps in diagnosing network connectivity between the Host OS and Guest OS.

**Procedure:**

**Step 1: Identify IP Addresses**

1. Open a **Command Prompt (Windows)** or **Terminal (Linux)** on the Guest OS.
2. Run the following command to find the Guest OS IP address:
   * Windows: ipconfig
   * Linux: ifconfig or ip a
3. Note down the IP address assigned to the Guest OS.
4. On the Host OS, open a **Command Prompt (Windows)** or **Terminal (Linux)**.
5. Run the following command to find the Host OS IP address:
   * Windows: ipconfig
   * Linux: ifconfig or ip a
6. Note down the IP address assigned to the Host OS.

**Step 2: Perform Ping Test**

**From Guest OS to Host OS:**

1. Open a **Terminal** or **Command Prompt** in the Guest OS.
2. Run the command:

ping <Host\_OS\_IP>

1. If successful, you will see replies with response times.
2. If it fails, troubleshoot firewall settings and network adapter configurations.

**From Host OS to Guest OS:**

1. Open a **Terminal** or **Command Prompt** in the Host OS.
2. Run the command:

ping <Guest\_OS\_IP>

1. Check for successful replies.

**Troubleshooting:**

1. **Firewall Issues:**
   * Ensure that ICMP (ping) is allowed in the firewall settings of both Host OS and Guest OS.
   * On Windows, allow File and Printer Sharing (Echo Request - ICMPv4-In) in the firewall settings.
2. **Network Adapter Issues:**
   * Ensure that the correct network adapter is selected in VM settings.
   * If using **NAT mode**, Guest OS might not be able to ping the Host OS.
   * If using **Host-Only Adapter**, ensure the adapter is enabled.
3. **Check Connectivity with Other Tools:**
   * Use tracert <IP> (Windows) or traceroute <IP> (Linux) to check the route of network packets.

**Viva Questions:**

1. What is the purpose of the ping command?
2. What are the differences between NAT, Bridged, and Host-Only networking modes?
3. What could be the possible reasons if the ping command fails?
4. How can we enable ICMP requests in Windows Firewall?
5. What alternative command can be used to check network connectivity?

**Lab 3:** Find a procedure to transfer the files from one virtual machine to another virtual machine.

**Aim:** To transfer files from one virtual machine to another virtual machine using different methods.

**Procedure:**

**Method 1: Using Shared Folders**

1. Enable shared folders in the virtualization settings of the virtual machines.
2. Mount the shared folder in both VMs.
3. Copy the required files into the shared folder.
4. Access the files from the other VM.

**Method 2: Using SCP (Secure Copy Protocol)**

1. Ensure SSH is enabled on both virtual machines.
2. Open a terminal on the source VM and run:

scp filename user@destination\_VM\_IP:/destination/path/

1. Enter the password for authentication.
2. Verify the file transfer by checking the destination folder on the target VM.

**Method 3: Using Network File Sharing**

1. Configure a network file sharing service:
   * **For Linux:** Use NFS.
   * **For Windows:** Use SMB (Samba).
2. Mount the shared folder in the destination VM.
3. Copy files to the shared folder and access them from the other VM.

**Method 4: Using USB Drive Emulation**

1. Attach a virtual USB drive to the source VM.
2. Copy the files to the USB drive.
3. Disconnect the USB from the source VM and attach it to the destination VM.
4. Access the copied files from the USB storage.

**Viva Questions:**

1. Which method provides the most secure way to transfer files between VMs?
2. What are the advantages of using SCP over shared folders?
3. How does a network file share improve file transfer efficiency?
4. What challenges might arise when using USB emulation for file transfer?
5. How can you automate file transfers between VMs?

**Troubleshooting**

1. **Permission Denied Error**
   * **Ensure the user has read/write access to the file or directory:**

**chmod +r /path/to/file**

* + **Use sudo if necessary:**

**sudo scp file user@destination\_IP:/root/**

1. **Connection Refused or Timeout**
   * **Verify SSH service is running:**

**sudo systemctl status ssh**

* + **Check firewall settings:**

**sudo ufw allow ssh**

1. **Unknown Host Error**
   * **Add the remote host’s fingerprint manually:**

**ssh-keyscan -H destination\_IP >> ~/.ssh/known\_hosts**

**Lab 4: Setting Up an AWS EC2 Instance**

**Aim:** To establish an AWS account, use the AWS Management Console to launch an EC2 instance, and connect to it.

**Tools and Software Required:**

* Web Browser (Google Chrome, Mozilla Firefox, etc.)
* AWS Free Tier Account
* SSH client (PuTTY for Windows, Terminal for Linux/macOS)

**Procedure:**

**Step 1: Create an AWS Account**

1. Go to the AWS website: <https://aws.amazon.com>
2. Click on **Create an AWS Account**.
3. Enter your email, create a password, and specify an AWS account name.
4. Provide personal details, including billing information (a credit/debit card is required, but AWS Free Tier is available).
5. Complete identity verification via phone and email.
6. Choose a **Basic Plan (Free Tier)** and complete account creation.

**Step 2: Access AWS Management Console**

1. Log in to your AWS account at [AWS Console](https://aws.amazon.com/console/).
2. Navigate to **EC2 Dashboard** by searching "EC2" in the AWS search bar.

**Step 3: Launch an EC2 Instance**

1. Click **Launch Instance**.
2. Choose an Amazon Machine Image (AMI):
   * Select **Amazon Linux 2023 (Free Tier eligible)**.
3. Choose an Instance Type:
   * Select **t2.micro (Free Tier eligible)** and click **Next**.
4. Configure Instance Details:
   * Leave defaults and click **Next**.
5. Add Storage:
   * Keep the default 8 GB (Free Tier eligible) and click **Next**.
6. Add Tags:
   * Click **Add Tag** → Set **Key** as Name, **Value** as MyFirstEC2.
   * Click **Next**.
7. Configure Security Group:
   * Select **Create a New Security Group**.
   * Allow SSH (22/TCP) from **My IP** (recommended for security).
   * Click **Review and Launch**.
8. Review settings and click **Launch**.
9. Create a new **key pair** (e.g., MyEC2Key), download it (.pem file), and click **Launch Instances**.

**Step 4: Connect to the EC2 Instance**

1. Go to **EC2 Dashboard** → **Instances**.
2. Select the newly created instance and note the **Public IPv4 Address**.
3. Use SSH to connect:
   * **On Linux/macOS:**

chmod 400 MyEC2Key.pem

ssh -i MyEC2Key.pem ec2-user@<Public\_IP>

* + **On Windows (PuTTY):**
    - Convert .pem to .ppk using **PuTTYgen**.
    - Open **PuTTY**, enter Public IP under **Host Name**, and load the .ppk file under **SSH → Auth**.
    - Click **Open** to connect.

**Viva Questions:**

1. What is an AMI in AWS?
2. Why is the chmod 400 command used on the key file?
3. What is the difference between Public IP and Private IP in AWS EC2?
4. How can you securely connect to an EC2 instance without using passwords?
5. What are security groups in AWS, and how do they enhance security?