DevOps Shack

Recovering a Lost Private Key from Linux VMs

Losing the private key to a Linux virtual machine (VM) is a critical issue that can lock you out of your system and disrupt workflows. Private keys are fundamental to secure access, and their recovery requires careful steps to regain control without compromising system integrity. This document provides a detailed step-by-step guide to recovering access to a Linux VM after losing its private key.

Understanding SSH Key Pairs

Before diving into recovery methods, it's essential to understand how SSH key pairs work:

- **Private Key:** Stored securely on the client machine, used to authenticate the user.
- **Public Key:** Placed in the ~/.ssh/authorized_keys file on the server. It verifies incoming connections from corresponding private keys.

When the private key is lost, authentication fails, and you need alternative methods to access the VM.

Recovery Options

The recovery process depends on your VM environment, available backups, and whether you have alternative access methods. Below are detailed recovery approaches:

---

1. Restoring from Backup or Snapshot

Most cloud providers and system administrators maintain regular backups or snapshots of VMs. This is the easiest and most reliable recovery method.

Steps:

1. **Identify Backups:** Check for available backups or snapshots of your VM.
    - Cloud platforms like AWS, Azure, and GCP provide snapshot features.
    - For on-premises VMs, check system-level backups.
2. **Restore the Snapshot:**
    - Restore the VM to a previous state when the private key was intact.
    - Access the restored VM using the recovered private key.
3. **Generate a New Key Pair:**
    - Use ssh-keygen to create a new key pair:

ssh-keygen -t rsa -b 4096 -f ~/.ssh/new_key

    - Update the public key in ~/.ssh/authorized_keys on the restored VM.

4. **Test Access:**
    - Confirm access with the new private key:

ssh -i ~/.ssh/new_key user@VM_IP

---

2. Accessing the VM Through Console or Alternate Login

Scenario:

You may still have alternative methods of accessing the VM, such as:

- Cloud-based console access.
- Another user account with SSH access.

Steps:

1. **Access via Console:**
    - Log in to the cloud provider portal (AWS, Azure, GCP).

- Open the **instance console** or use emergency access tools like Session Manager (AWS).

2. **Generate a New SSH Key Pair:**
   - On your local machine, generate a new key pair:

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/new_key
```

3. **Update the authorized_keys File:**
   - Log in to the VM via the console.
   - Edit the authorized_keys file:

```
nano ~/.ssh/authorized_keys
```

   - Append the new public key (~/.ssh/new_key.pub).

4. **Secure File Permissions:**
   - Set proper permissions for .ssh and authorized_keys:

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

5. **Test Access:**
   - Attempt to connect using the new private key:

```
ssh -i ~/.ssh/new_key user@VM_IP
```

3. Using Backups of the .ssh Directory

If you have a backup of your local .ssh directory, you can restore the private key.

Steps:

1. **Locate Backup:**
   - Search for the backup location of your .ssh directory.
   - Look for files named id_rsa or similar.
2. **Restore Private Key:**
   - Copy the private key file to ~/.ssh/:

```
cp /path/to/backup/id_rsa ~/.ssh/id_rsa
chmod 600 ~/.ssh/id_rsa
```

3. **Test Access:**
   - Use the restored private key to connect:

ssh -i ~/.ssh/id_rsa user@VM_IP

---

4. Resetting the SSH Key Through Cloud Providers

Most cloud providers offer tools to reset SSH keys.

AWS (EC2 Instances):

1. **Use Systems Manager (SSM):**
    - If SSM Agent is enabled, use Session Manager to access the instance.
    - Update the authorized_keys file with a new public key.
2. **Replace the Key via AWS CLI:**
    - Detach the root volume and attach it to another instance.
    - Mount the volume, edit the authorized_keys file, and reattach the volume to the original instance.

Azure (Azure VMs):

1. **Reset SSH Key via Portal:**
    - Navigate to the VM in the Azure portal.
    - Use the **Reset SSH Public Key** option.
    - Upload a new public key.
2. **Reset SSH Key via Azure CLI:**
    - Run the following command:

az vm user update --resource-group <ResourceGroupName> --name <VMName> --username <UserName> --ssh-key-value <NewPublicKey>

Google Cloud (GCP VMs):

1. **Add Public Key to Metadata:**
    - Navigate to the instance in the GCP console.
    - Add the new public key to the instance metadata.

---

5. Booting into Recovery Mode

If no direct access methods work, you can boot the VM into recovery mode.

Steps:

1. **Access Recovery Mode:**
    - Restart the VM and select **Recovery Mode** from the boot menu.
    - If on a cloud platform, use the recovery options provided.

2. **Mount the Root Filesystem:**
    - Identify and mount the root filesystem:

```
mount /dev/sda1 /mnt
```

3. **Edit the authorized_keys File:**
    - Navigate to the user's .ssh directory:

```
nano /mnt/home/user/.ssh/authorized_keys
```

    - Append the new public key.

4. **Exit Recovery Mode:**
    - Unmount the filesystem:

```
umount /mnt
```

    - Reboot the VM into normal mode.

5. **Test Access:**
    - Use the new private key to connect.

---

6. Generating a New Key Pair

If you cannot recover the old private key, generate a new key pair and configure the VM to accept the new public key.

Steps:

1. **Generate New Key Pair:**

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/new_key
```

2. **Add Public Key to VM:**
    - Follow the steps in any of the above methods to update the authorized_keys file on the VM.

3. **Test Access:**
    - Connect using the new private key:

```
ssh -i ~/.ssh/new_key user@VM_IP
```

---

7. Avoiding Cryptographic Attacks

Recovering a private key through cryptographic attacks (e.g., brute force) is computationally infeasible for modern keys (e.g., 2048-bit or higher). These methods are not recommended and should only be considered in exceptional circumstances with professional assistance.

Best Practices to Prevent Future Key Loss

To avoid losing access to VMs in the future, consider these best practices:

1. **Back Up Keys:**
    - Store private keys securely using a password manager or secure storage.
2. **Use Key Management Systems:**
    - Utilize tools like **AWS Secrets Manager**, **Azure Key Vault**, or **HashiCorp Vault** to manage SSH keys.
3. **Enable Secondary Access:**
    - Set up additional user accounts with separate SSH keys.
    - Enable cloud-native access tools like **AWS Systems Manager**, **Azure Bastion**, or **GCP OS Login**.
4. **Document Access Procedures:**
    - Maintain a recovery guide for your VM environment.
5. **Use Passphrases:**
    - Protect private keys with a strong passphrase.
6. **Regularly Rotate Keys:**
    - Update SSH keys periodically and ensure all team members use the updated keys.

Conclusion

Losing a private key is a serious issue, but with the right tools and strategies, you can recover access to your Linux VM. Whether through backups, console access, or recovery modes, following these detailed methods will help you regain control securely. Implementing preventive measures ensures that you are prepared for future incidents and minimizes downtime.