Day 14

Exploitation Analyst

SSH Protocol:

Hacking SSH protocol:

What are the various key generated in the SSH key pair connection?

In an SSH key-based authentication setup, a key pair is generated consisting of two keys: a private key and a public key. The private key (e.g., id_rsa) remains securely on the client machine and must be protected; it should never be shared. The public key (e.g., id_rsa.pub) is copied to the remote server's ~/.ssh/authorized_keys file. During connection, the server challenges the client, which proves its identity using the private key. The server uses the matching public key to verify this proof. These keys are typically generated using RSA, ECDSA, or Ed25519 algorithms and form the core of secure, passwordless SSH authentication.

What if the key which is shared get sniffed?

If the public key (the one you copy to the server) gets sniffed or intercepted during transfer, it's not a security risk. Public keys are meant to be shared — anyone can have a copy. The security of SSH lies in the private key, which must stay secret and protected on your machine.

How Private Key Theft Happens:

Steps: (assuming hacker already has the remote access to our computer)

1. Victim generates an SSH key pair

- User runs ssh-keygen on their system (e.g., Kali).
- This creates:
  - A **private key** (~/.ssh/id_rsa)
  - A **public key** (~/.ssh/id_rsa.pub)
- The public key is copied to the server's authorized_keys, enabling key-based login.

2. Attacker gains access to victim's system

This can happen in several ways:

- Through a **phishing attack** or **malicious email attachment**.
- By exploiting a **vulnerability** in software or services.
- Via **physical access** or a misconfigured shared directory.

3. Private key file is located and stolen

- Attacker searches typical locations like ~/.ssh/id_rsa.
- If file permissions are weak (e.g., world-readable), it can be copied easily.
- The attacker **downloads the private key** to their own machine.

4. Private key is used to access the server

- Attacker uses the stolen private key:
- ssh -i id_rsa username@target_ip
- If:
    - The **private key has no passphrase** or
    - The attacker can **crack the passphrase**,

then they gain **full access** to the remote server without needing a password.

5. Attacker maintains persistence or escalates

- Installs their own public key for future access.
- Adds backdoors or malware to maintain control.
- May disable logs or alerts to avoid detection.

How to Protect Against This:

1. Always encrypt private keys with a strong passphrase.
2. Use file permissions: *chmod 600 ~/.ssh/id_rsa*
3. Never store keys on shared or public folders.
4. Use hardware tokens (like YubiKey) for secure key storage.
5. Monitor servers for new or unauthorized public keys.
6. Revoke access immediately if key theft is suspected.