**Day 12**

**Exploitation Analyst**

**SSH Protocol:**

YouTube:

<https://youtu.be/QPBhfdXhfXw?si=jTlRkLqFNRG4QBdO>

**What is SSH?**

SSH stands for Secure Shell. It's a protocol to securely connect to remote computers over a network.

**How it works?**

1. Client generates an asymmetric key pair:

* Public key
* Private key

1. Public key is copied to the server, saved in ~/.ssh/authorized\_keys.
2. When the client connects:

* Server generates a random session key (this is your “new key”).
* Server encrypts this session key with the client's public key.

1. Server sends this encrypted session key to the client.
2. Client decrypts it using its private key.
3. Now both sides share the same session key (symmetric), used to encrypt the actual SSH session.

**How this connection differs from that of SSL?**

| Step | SSL/TLS (Web/HTTPS) | SSH (Remote Login) |
| --- | --- | --- |
| Who generates keys? | Server generates key pair | Client generates key pair |
| Who holds private key? | Server | Client |
| Public key stored where? | In a certificate, signed by CA, sent to clients | On server (authorized\_keys file) |
| Who verifies identity? | Client checks CA’s signature to trust server | Server trusts client via matching public key |
| How is session key exchanged? | Client generates session key, encrypts with server’s public key | Server generates it, encrypts with client’s public key |
| Authentication | Server is authenticated via CA certificate | Client is authenticated via key possession |
| Use case | Secure websites (e.g., HTTPS) | Secure remote shell access (e.g., ssh user@host) |

**Advantages of SSH:**

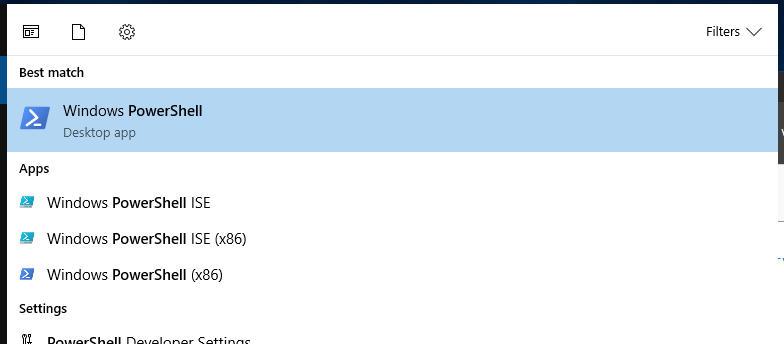
1. Secure Communication: Encrypted data protects against eavesdropping.
2. Authentication Options: Supports passwords and key-based login.
3. Remote Access: Safely control servers over networks.
4. Port Forwarding: Securely tunnel other services (like databases).
5. File Transfer: With SCP or SFTP, send files securely.
6. Integrity & Confidentiality: Prevents tampering and spoofing.

**Disadvantages of SSH:**

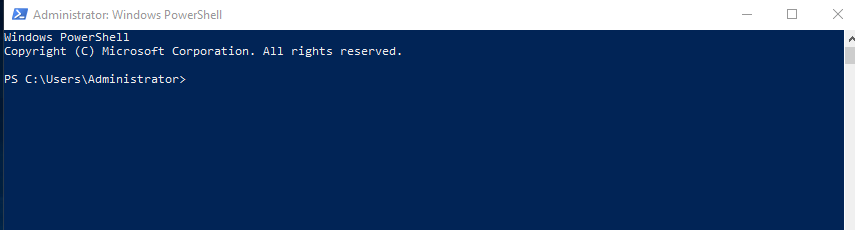
1. Setup Complexity: Key management can be tricky for beginners.
2. Risk if Keys/Passwords Leaked: If private key or password is stolen, access is compromised.
3. No Built-in GUI: It's command-line based (hard for non-tech users).
4. Firewall Restrictions: Port 22 might be blocked in some networks.
5. Brute Force Attacks: Needs strong passwords or keys to stay safe.

**Connecting to SSH of Windows 2019 server using id name and password:**

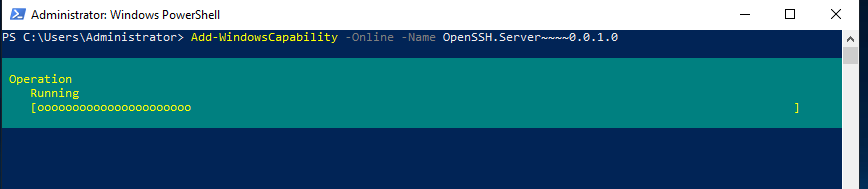
Open “PowerShell” in Windows 2019 server in admin mode:



Following screen will appear:

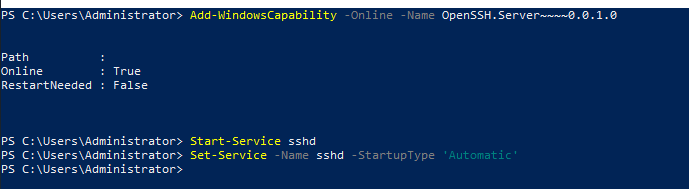


Type the following command:

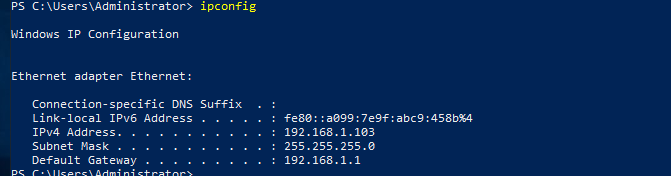


This command installs the OpenSSH Server feature on your Windows Server 2019 system. It allows the server to accept SSH connections, just like a Linux machine. The -Online flag means it’s applied to the running OS. Without this, your server won’t understand or respond to SSH requests.

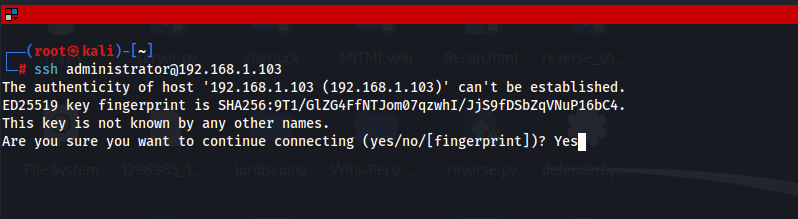
When the above state changes, type the following:



These two commands start the SSH server immediately and ensure it starts every time the server boots up. Without this, you'd have to manually start the SSH service every time you restart the server, which is not ideal for remote access.

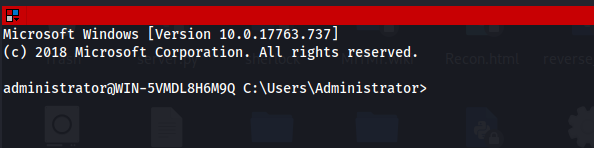
Note the IP of the Windows 2019: it is 192.168.1.103  


Now, in kali, use below command with the id name of it and IP of the windows 2019:

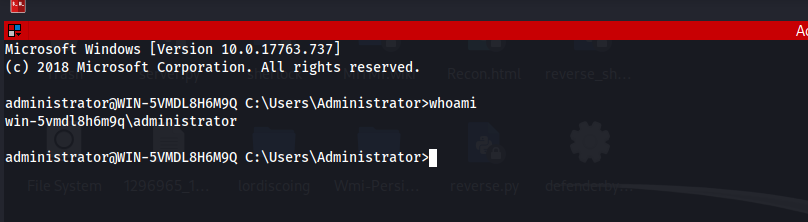


This command is run from Kali to initiate an SSH connection to the Windows Server. You replace username with the actual Windows account name. Once executed, it tries to connect to the server over the network and gives you a secure remote terminal if authentication succeeds.

Once, you write ‘yes’ (as shown above) and click enter, it will ask for password. Once you entered that: following screen will appear.



Proving my identity:



**Why this method is not secure?**

1. Brute Force Attacks

* Attackers can try millions of username-password combos using automated tools.
* Common usernames like Administrator, root, etc. are easily guessed.

1. Password Reuse
   * Many users reuse passwords across systems. If one is leaked, others are exposed.
2. Keyloggers & Phishing
   * Passwords can be stolen via malware or phishing emails.
3. No Length or Complexity Enforcement by SSH
   * SSH does not force strong password policies by default.
4. No Two-Factor (by default)
   * Password login lacks an extra layer like OTP or hardware key (unless manually added).

**Can SSH username & password be sniffed using Wireshark?**

No — not in plaintext. When you use SSH, the entire connection is encrypted.

--The End--