🔐 SSH (Secure Shell) – Technical Writeup

📌 Introduction

SSH (Secure Shell) is a cryptographic network protocol used for secure communication over an unsecured network. It enables users to securely access remote systems, execute commands, transfer files, and manage services. SSH replaces older protocols like **Telnet**, **FTP**, and **rlogin**, which transmit data in plaintext.

🔐 SSH Protocol Stack

SSH operates at the **application layer** of the OSI model and comprises three major components:

Layer	Description
Transport Layer Protocol	Provides server authentication, confidentiality, integrity via encryption and MACs.
User Authentication Protocol	Handles client authentication (e.g., password, public key).
Connection Protocol	Multiplexes encrypted tunnel into channels for terminal sessions, file transfers, etc.

🔄 SSH Key Generation

```
# Generate key pair (RSA 4096 bits)
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Files generated:

- `~/.ssh/id_rsa` → Private key
- `~/.ssh/id_rsa.pub` → Public key

To copy public key to server:

```
ssh-copy-id user@hostname
```

⚙️ SSH Configuration

Global: `/etc/ssh/ssh_config` User-specific: `~/.ssh/config`

Example `~/.ssh/config`:

```
Host devserver
    HostName dev.example.com
    User anish
    Port 2222
    IdentityFile ~/.ssh/id_rsa
```

Usage:

```
ssh devserver
```

🔁 SSH Agent and Agent Forwarding

- `ssh-agent` stores decrypted private keys in memory.
- `ssh-add` adds keys to agent.
- `ForwardAgent yes` allows key forwarding across jump hosts.

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_rsa
```

📦 Advanced Tools

Tool	Purpose
`autossh`	Auto-reconnect dropped SSH sessions
`sshuttle`	VPN-like tunneling using SSH
`mosh`	UDP-based remote shell over SSH
`paramiko`	Python SSH automation library

📜 SSH Log Location

- Ubuntu/Debian: `/var/log/auth.log`
- Red Hat/CentOS: `/var/log/secure`

🏁 Conclusion

SSH is a fundamental tool for secure remote administration, scripting, and file transfer in Unix-like systems. Mastering its full capabilities, including tunneling, key management, and secure configurations, significantly enhances your system security posture and operational efficiency.