

How to Configure SSH Host-Based Authentication Between an OpenVMS SSH Client (SSH-CLIENT) and an OpenVMS SSH Server (SSH-SERVER)

Table of contents

1. How to Configure SSH Host-Based Authentication Between an OpenVMS SSH Client (SSH-CLIENT) and an OpenVMS SSH Server (SSH-SERVER)	3
2. Procedure	4
2.1 Enable the SSH Client	4
2.2 Configure the SSH Client	5
2.3 Enable the SSH Server	6
2.4 Configure the SSH Server	7
2.5 Test Host-Based Authentication	8

1. How to Configure SSH Host-Based Authentication Between an OpenVMS SSH Client (SSH-CLIENT) and an OpenVMS SSH Server (SSH-SERVER)

Applies to: TCP/IP Services for OpenVMS V5.7

SSH supports password, public-key, and host-based authentication. Host-based authentication allows trusted systems to authenticate without requiring user credentials.

This article describes how to configure host-based authentication between an OpenVMS SSH client (SSH-CLIENT) and an OpenVMS SSH server (SSH-SERVER).

2. Procedure

2.1 Enable the SSH Client

1. Run the TCP/IP configuration utility:

```
$ @SYS$STARTUP:TCPIP$CONFIG.COM
```

2. From the **TCP/IP Services Configuration Menu**, select:

```
2 - Client components
```

3. From the **Client Components Configuration Menu**, select:

```
7 - SSH Client
```

4. In the **SSH CLIENT configuration options** menu, select:

```
3 - Enable & Start service on this node
```

5. Verify that the client starts successfully with this output:

```
%TCPIP-S-STARTDONE, TCPIP$SSH_CLIENT startup completed
```

2.2 Configure the SSH Client

1. Generate a host key pair on the client system:

```
$ SSH_AGENT
$ SSH_KEYGEN "-P" TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]HOSTKEY
```

This creates:

```
TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]HOSTKEY
TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]HOSTKEY.PUB
```

2. Enable host-based authentication. Edit the file TCPIP\$SSH_DEVICE:
[TCPIP\$SSH.SSH2]SSH2_CONFIG and change:

```
# AllowedAuthentications hostbased,publickey,password
```

to:

```
AllowedAuthentications hostbased
```

2.3 Enable the SSH Server

1. Run the TCP/IP configuration utility:

```
$ @SYS$STARTUP:TCPIP$CONFIG.COM
```

2. From the **TCP/IP Services Configuration Menu**, select:

```
3 - Server components
```

3. From the **Server Components Configuration Menu**, select:

```
19 - SSH
```

4. In the **SSH configuration options** menu, select:

```
3 - Enable & Start service on this node
```

5. When prompted, create the default server host key:

```
* Create a new default server host key? [YES]:
```

Press **Enter** to accept the default value.

6. Verify that the server starts successfully:

```
%TCPIP-S-STARTDONE, TCPIP$SSH startup completed
```

2.4 Configure the SSH Server

1. Enable Host-Based Authentication

Edit:

```
TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSHD2_CONFIG.
```

Change:

```
# AllowedAuthentications hostbased,publickey,password
```

to:

```
AllowedAuthentications hostbased
```

2. Install the Client Host Key

Copy the client host public key to the server's KNOWNHOSTS directory:

```
$ COPY/FTP SSH-CLIENT"username password"::-
TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]HOSTKEY.PUB -
TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]KNOWNHOSTS]-
SSH-CLIENT_SOME_NETWORK_DOMAIN_NET_SSH-DSS.PUB
```

Grant world-read access to the copied key:

```
$ SET FILE/PROTECTION=W:R -
TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]KNOWNHOSTS]-
SSH-CLIENT_SOME_NETWORK_DOMAIN_NET_SSH-DSS.PUB
```

3. Configure Trusted Hosts

Edit the trusted-hosts file:

```
TCPIP$SSH_DEVICE:[TCPIP$SSH]SHOSTS.EQUIV
```

Add the client host name:

```
SSH-CLIENT.SOME.NETWORK.DOMAIN.NET
```

2.5 Test Host-Based Authentication

From the client system, connect to the SSH server and run a command to verify that host-based authentication is working:

```
$ SSH SSH-SERVER SHOW LOGICAL TCPIP$INET_HOST
```

A successful host-based authentication returns output similar to:

```
Authentication successful.
```

```
"TCPIP$INET_HOST" = "ssh-server" (LNM$SYSTEM_TABLE)
```
