VMS Software

How to Configure DECnet over IP with Stunnel on OpenVMS

VMS Software, Inc. (VSI)

Boston, Massachusetts, USA

Table of contents

1. Procedure 3
   - 1.1 How to Configure DECnet Over IP With Stunnel on OpenVMS 3
   - 1.2 Create the PEM File 4
   - 1.3 Create the Stunnel Configuration Files 5
   - 1.4 Start Stunnel 6
   - 1.5 Verify the Listener Port 7
   - 1.6 Define the Logical 8
   - 1.7 Create a DECnet Over IP Stunnel Connection 9

1. Procedure

1.1 How to Configure DECnet Over IP With Stunnel on OpenVMS

Applies to:

- OpenVMS IA-64 Version 8.4-1H1 and higher
- TCP/IP Services for OpenVMS 5.7 ECO 5, or VSI TCP/IP Services for OpenVMS 6.0 and higher
- Stunnel v5.56B and higher
- DECnet Phase V with DECnet over IP

> **Note**
>
> This procedure assumes that you already have an SSL certificate and private key for your system. For information about certificate generation, see the OpenSSL Release Notes.

Stunnel enables encrypted communications for applications that lack native encryption support. It accepts connections from a local application, encrypts and forwards the traffic to a remote host, where it is decrypted and delivered to the target application. Response traffic follows the reverse path.

To encrypt a connection, select an unused TCP port on both the client and server systems. In the examples in this article, the same port number is used on both sides, although the port numbers do not need to match.

The example configuration forwards DECnet over IP traffic through Stunnel using port `2399`.

1.2 Create the PEM File

Each TLS-enabled daemon needs to present a valid X.509 certificate to the peer.

The `.PEM` file must contain:

- An unencrypted private key
- A signed certificate (not a certificate request)

```
-----BEGIN RSA PRIVATE KEY-----
[encoded key]
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
[encoded certificate]
-----END CERTIFICATE-----
```

Create the `.PEM` file for Stunnel and remove the group and world permissions.

```
$ SET DEF STUNNEL$ROOT:[ETC]
$ COPY/CONCATENATE SERVER.KEY,SERVER.CRT STUNNEL.PEM
$ SET SECURITY/PROTECTION=(G,W) STUNNEL.PEM
```

Verify the file protection:

```
$ DIRECTORY/SECURITY STUNNEL.PEM;1

Directory STUNNEL$ROOT:[000000.etc]

 Stunnel.pem;1        [SYSTEM]                          (RWED,RWED,,)

Total of 1 file.
```

1.3 Create the Stunnel Configuration Files

Create the configuration file for both the client and the server.

More information regarding the configuration file can be found on the Stunnel website.

Sample client configuration file:

```
$ TYPE STUNNEL$ROOT:[ETC]STUNNEL.CONF

debug = 7
output = /stunnel$root/etc/debug.log
cert = /stunnel$root/etc/stunnel.pem
client=yes
;
; Change the "connect" value as appropriate for your environment
[DECNET_IP]
accept  = 2399
; connect address is for the remote/server node
connect = 10.10.10.10:2399
```

Sample server configuration file:

```
$ TYPE STUNNEL$ROOT:[ETC]STUNNEL.CONF

debug = 7
output = /stunnel$root/etc/debug.log
cert = /stunnel$root/etc/stunnel.pem
sslVersion = TLSv1
pid = /stunnel.pid
; socket = l:TCP_NODELAY=1
; socket = r:TCP_NODELAY=1
;
; Change the "accept" port number as appropriate
[DECNET_IP]
accept = 2399
connect = 399
```

1.4 Start Stunnel

Start Stunnel on both the client and server. If Stunnel is already running, shut it down and restart it.

Shutdown:

```
$ @SYS$STARTUP:STUNNEL$SHUTDOWN.COM
```

Startup:

```
$ @SYS$STARTUP:STUNNEL$STARTUP.COM
```

> **Note**
>
> If Stunnel startup is required during system startup, place an entry in `SYS$MANAGER:SYSTARTUP_VMS.COM` (or similar). To ensure a clean shutdown, place the shutdown procedure in `SYS$MANAGER:SYSHUTDOWN.COM` (or similar).

1.5 Verify the Listener Port

Verify that you have a listener on port `2399` on both systems.

Client:

```
$ TCPIP SHOW DEVICE/PORT=2399

                            Port                          Remote
Device_socket  Type     Local  Remote  Service             Host

  bg54319      STREAM    2399       0                    *
```

Server:

```
$ TCPIP SHOW DEVICE/PORT=2399

                            Port                          Remote
Device_socket  Type     Local  Remote  Service             Host

  bg63040      STREAM    2399       0                     *
```

> **Note**
>
> If you do not have a listener port on each system, check the log file for more information. The log file is located at: `STUNNEL$ROOT:[LOGS]*.LOG`.

1.6 Define the Logical

Define the logical used for DECnet over IP port forwarding using:

```
$ DEFINE/USER DECNET_IP_PORT_FORWARD 2399
```

Note

When defining the logical, select the appropriate scope:

- `DEFINE/USER`: Defines the logical for the next command only.
- `DEFINE/PROCESS`: Defines the logical for the current process.
- `DEFINE/SYSTEM`: Defines the logical for the entire system.

If a user needs to utilize Stunnel for the next connection, then `DEFINE/USER` would be appropriate. Subsequent `SET HOST` commands would not have port forwarding configured.

If the user needs to use encryption for every connection, then the logical could be defined with `/PROCESS`.

If all users require the logical, use `/SYSTEM`.

1.7 Create a DECnet Over IP Stunnel Connection

From the client side, issue the following command:

```
$ SET HOST IP$127.0.0.1
```

> **Note**
>
> The port used is the same as specified in the configuration file. Although the connection is made to the localhost address, the response displayed on the screen originates from the server system.
>
> The `IP$address` reference is needed to force DECnet to use DECnet over IP when sending traffic locally.

Example:

```
CLIENT$ DEFINE/USER DECNET_IP_PORT_FORWARD 2399
CLIENT$ SET HOST IP$127.0.0.1

SERVER_HOST (L2) (HP rx2660 (1.67GHz/9.0MB))

               Unauthorized access prohibited.

Username: system
Password:
```

For more information, visit the Stunnel website.