

15.11.3 VPN Access and Authentication Facts

This lesson covers the following topics:

- VPN access and authentication
- Secure Sockets Layer (SSL)
- IPSec
- Datagram Transport Layer Security (DTLS)

VPN Access and Authentication

A virtual private network or VPN is a type of network that uses encryption to allow IP traffic to travel securely over a TCP/IP network and is used primarily to support secure communications over an untrusted network. In simpler terms, a VPN is an encrypted connection between two or more remote computers.

VPNs are most commonly used by employees who are working remotely, yet require access to their company's network resources, and by companies that want to maintain a secure connection between remote sites.

The software required to connect your Linux client to a VPN, may vary depending on the type of VPN to which you are connecting. After contacting your VPN provider, you can check your local system to see if the correct software is installed. If not go to the software installer application and search for the NetworkManager package which works with your VPN and install it. Once installed, you will need to create a new VPN network, configured to the specifications of your VPN to which you are connecting.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL), is the standard technology used for keeping an internet connection secure. It does this by encrypting the data sent between systems and using digital certificates to ensure that only the intended recipients can view and use the data sent. For example, SSL is often used to secure your communications between your web browser and a web server.

An SSL VPN is a type of virtual private network that uses the SSL protocol or the Transport Layer Security (TLS) protocol to provide secure, remote-access VPN capability. However, in most cases, TLS is most often used since the Internet Engineering Task Force deprecated SSL. Although TLS is most often used, most still refer to them together as SSL/TLS.

The Transport Layer Security protocol is an improved version of SSL. It ensures that messages

being transmitted on the Internet are private and tamper proof.

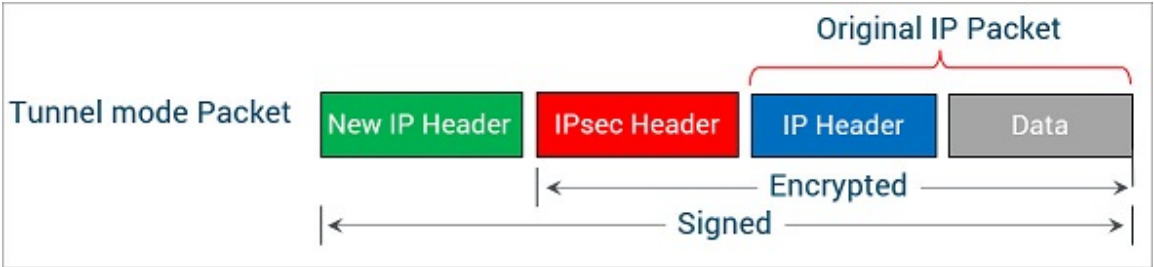
To set up your SSL/TLS installation, you must define your channels used to connect your systems to use SSL or TLS. You'll also have to create and manage your digital certificates. Once everything is setup, you can test your SSL or TLS conditions using self-signed certificates. Be aware, however, that self-signed certificates cannot be revoked, which could allow an attacker to spoof an identity after a private key has been compromised so only use these types of certificates for testing.

OpenSSL is a command-line tool that is often installed on many Linux distribution by default or it can be downloaded and installed as needed. With OpenSSL, you can create and view certificates, and test your SSL/TLS connections.

IPsec

Internet Protocol Security (IPsec), is an extension to the IP protocol and like SSL also secures sessions between computers by validating and encrypting the packets of data that are sent across a network.

When setting up and using IPsec, it's important to understand that IPsec has two modes: tunnel and transport.

IPsec Mode	Description
Tunnel	<p>The default mode. The entire original IP packet is protected by IPsec, meaning that IPsec wraps and encrypts the original packet and then adds a new IP header which is then sent on to the other side of the VPN tunnel.</p> <div><p>The diagram illustrates the structure of a Tunnel mode IPsec packet. It shows a sequence of four components: a green box labeled 'New IP Header', a red box labeled 'IPsec Header', a blue box labeled 'IP Header', and a grey box labeled 'Data'. A red bracket above the 'IP Header' and 'Data' boxes is labeled 'Original IP Packet'. Below the boxes, a double-headed arrow spans the 'IPsec Header', 'IP Header', and 'Data' sections, labeled 'Signed'. Another double-headed arrow spans the 'IP Header' and 'Data' sections, labeled 'Encrypted'.</p></div> <p>When using VPN tunnel mode and NAT, the NAT information is stripped off and the original IP packet is sent on preserving the digital signature.</p>
Transport	<p>Only the payload of the IP packet is encrypted, and the original IP headers are left intact. Although this lessens the number of additional bytes required to use VPN it also allows devices on the public network to see the final source and destination of the packet. By passing the IP</p>

12/9/22, 12:38 AM

TestOut LabSim

header as unencrypted data, the transport mode allows an attacker to perform some traffic analysis. The transport mode is often used for end-to-end communications, such as when connecting a client and a server.

The diagram illustrates the structure of an IP packet in transport mode. It shows two packets: an 'Original IP Packet' and a 'Transport mode Packet'. The 'Original IP Packet' has an 'IP Header' (blue box) and 'Data' (grey box, labeled 'Payload'). The 'Transport mode Packet' has an 'IP Header' (blue box), an 'IPsec Header' (red box), and 'Data' (grey box). A bracket labeled 'Signed' spans the 'IPsec Header' and the 'Data' section. Another bracket labeled 'Encrypted' spans the 'Data' section. An arrow points from the 'Original IP Packet' to the 'Transport mode Packet'.

Datagram Transport Layer Security (DTLS)

DTLS is based on the TLS protocol and provides security for datagram-based applications by allowing them to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. The main difference between DTSL and TLS is that DTLS uses UDP and TLS uses TCP.

TLS relies on TCP to guarantee that the packet was delivered correctly in the event that the packet was fragmented, reordered or loss. Getting rid of any one of those TCP features would normally break the TLS crypto logic, but DTLS has created its own solution or workaround for each as shown in the following table:

TCP Feature	DTLS with UDP Solution
Message fragmentation	Fragmentation occurs when a packet datagram is too large to fit within the maximum transmission unit (MTU). DTLS provides fragmentation offset and length values in the DTLS message. This ensure that both ends of the communication are provided fragmentation information regardless of the underlying transport.
Message reordering	If a packet arrives out of order and is not reassembly correctly, the packet cannot be decrypted properly. DTLS adds sequence numbering to the application, allowing it to not be dependent on the underlying transport technology.
Packet loss	DTLS adds a simple retransmission timer allowing it to retransmit packets without relying on the transport protocol.

One disadvantage of using DTLS, is that the above built-in recovery functions requires additional memory.

Copyright © 2022 TestOut Corporation All rights reserved.