15.11.1 VPN Access and Authentication

Click one of the buttons to take you to that part of the video.

VPN Access and Authentication 0:00-0:36

In this lesson, we're going to talk about Linux VPNs and the protocols used to support VPNs.

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 supports 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 two groups: remote employees that require access to their company's network resources and companies that want to maintain a secure connection between remote sites.

Secure Sockets Layer 0:37-2:17

To understand how VPNs work, we need to talk about SSL.

Secure Sockets Layer, or SSL, is the standard technology used to keep 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 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 protocol to provide secure remote-access VPN capability. However, in most cases, Transport Layer Security, or TLS, is used since the Internet Engineering Task Force deprecated SSL. Although TLS is most often used, most people 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 the channels that connect your systems to use SSL or TLS. You'll also have to create and manage your digital certificates. Once everything is set up, 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's installed on many Linux distribution by default; and, of course, you can also download and install it yourself.

With OpenSSL, you can create and view certificates and test your SSL/TLS connections.

IPSec Transport Mode 2:18-3:18

Internet Protocol Security, or IPSec, is an extension to the IP protocol. Like SSL, it 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, transport mode and tunnel mode.

In transport mode, only the payload of the IP packet is encrypted, and the original IP headers are left intact.

This mode has the advantage of adding only a few bytes to each packet. But it also allows devices on the public network to see the final source and destination of the packet. However, by passing the IP header as unencrypted data, transport mode allows an attacker to perform some traffic analysis.

This mode is often used for end-to-end communications, such as communication between a client and a server or a workstation and a gateway acting as a host.

For example, this mode is suited to use with an encrypted Telnet session from a workstation to a router in which the router is the actual destination.

IPSec Tunnel Mode 3:19-4:13

However, when working with VPNs, IPSec is normally used in tunnel mode, which is the default mode.

In tunnel mode, the entire original IP packet is protected by IPSec. This means 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.

You can really see one of tunnel mode's benefits while using NAT routers.

A NAT router lets users or a company using private IP address send their data across the internet using a public IP address generated by the NAT router.

The problem with NAT and IPSec is that NAT modifies the contents of the IP packet. As a result, these packets are rejected by the receiving target because the signature is wrong.

If you use tunnel mode and NAT, this concern disappears because once the NAT information is stripped off, the original IP packet is sent on, and the signatures will still match.

Datagram Transport Layer Security (DTLS) 4:14-6:17

The last VPN protocol we need to discuss is the Datagram Transport Layer Security protocol, or DTLS.

DTLS is based on the TLS protocol and provides security for datagram-based applications by allowing them to communicate using a method 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 lost.

First, we have message fragmentation.

Fragmentation occurs when a packet datagram is too large to fit within the maximum transmission unit (MTU).

Although TCP has a way to fix this issue; UDP does not. However, DTSL resolves this issue by introducing its own fragmentation offset and length value in the DTLS message itself. This ensure that both ends of the communication are provided fragmentation information, regardless of the underlying transport.

Next, we have message reordering. With TCP, if a packet arrives out of order, it uses sequence numbering to ensure that the original data is reassembled properly. If the reassembly is not performed correctly, then the packet can't be properly decrypted.

DTLS solves this problem by adding its own sequence numbering to the application, allowing it to be independent of the underlying transport technology.

And, finally, we have the issue of lost packets. With TCP, if a packet is lost, it tells the sender to resend the lost packet. UTP can't do this. DTLS fixes this problem by adding a simple retransmission timer to its application logic, allowing it to retransmit packets without relying on the transport protocol.

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

Summary 6:18-6:33

That's it for this lesson.

In this lesson, we introduced VPN as a client and then explained the role SSL/TLS plays in a VPN connection.

We also explained how an IPSec VPN works, including its transport mode and tunnel mode.

And we ended this lesson by discussing how using DTLS VPNs with DTLS overcomes the weaknesses of TCP features.
