ICSI MODULE 6:

Introduction to VPN

6.1 Introduction to VPN

Virtual Private Networks (VPNs) are a common way to connect remotely to a network in a secure fashion. A VPN creates a private network connection over the Internet to connect remote sites or users together. Instead of using a dedicated connection, a VPN uses virtual connections routed through the Internet from the remote site or user to the private network. Security is accomplished by encrypting all the transmissions.

A VPN allows a remote user to have network access just as if were local to the private network. This means not only connecting the user to the network as if the user were local but also making the connection secure. Because most organisations have many employees traveling and working from home, remote network access has become an important security concern. Users want access, and administrators want security. The VPN is the current standard for providing both.

To accomplish its purpose, the VPN must emulate a direct network connection. This means it must provide both the same level of access and the same level of security as a direct connection. To emulate a dedicated point-to-point link, data is encapsulated, or wrapped, with a header that provides routing information allowing it to transmit across the Internet to reach its destination. This creates a virtual network connection between the two points. The data being sent is also encrypted, thus making that virtual network private.

A VPN does not require separate technology or direct cabling. It is a virtual private network, which means it can use existing connections to provide a secure connection. In most cases it is used over normal Internet connections.

A variety of methods are available for connecting one computer to another. At one time dialling up to an ISP via a phone modem was common. Now cable modems, cellular devices, and other mechanisms are more common. All of these methods have something in common: they are not inherently secure. All data being sent back and forth is unencrypted, and anyone can use a packet sniffer to intercept and view the data. Furthermore, neither end is authenticated. This means you cannot be completely certain who you are really sending data to or receiving data from. The VPN provides an answer to these issues.

This sort of arrangement is generally acceptable for an ISP. The customers connecting simply want a channel to the Internet and do not need to connect directly or securely to a specific network. However, this setup is inadequate for remote users attempting to connect to an organisation's network. In such cases the private and secure connection a VPN provides is critical.

Individual remote users are not the only users of VPN technology. Many larger organisations have offices in various locations. Achieving reliable and secure site-to-site connectivity for such organisations is an important issue. The various branch offices must be connected to the central corporate network through tunnels that transport traffic over the Internet.

Using VPN technology for site-to-site connectivity enables a branch office with multiple links to move away from an expensive, dedicated data line and to simply utilize existing Internet connections.

VPN Protocols

6.2 VPN Protocols

Multiple ways exist to achieve the encryption needs of a VPN. Certain network protocols are frequently used for VPNs. The two most commonly used protocols for this purpose are Point-to-Point Tunnelling Protocol (PPTP) and Layer 2 Tunnelling Protocol (L2TP). The part of the connection in which the data is encapsulated is referred to as the tunnel. L2TP is often combined with IPSec to achieve a high level of security.

6.2.1 PPTP

PPTP is a tunnelling protocol that enables an older connection protocol, PPP (Point-to-Point Protocol), to have its packets encapsulated within Internet Protocol (IP) packets and forwarded over any IP network, including the Internet itself. PPTP is often used to create VPNs. PPTP is an older protocol than L2TP or IPSec. Some experts consider PPTP to be less secure than L2TP or IPSec, but it consumes fewer resources and is supported by almost every VPN implementation. It is basically a secure extension to PPP.

PPTP was originally proposed as a standard in 1996 by the PPTP Forum—a group of companies that included Ascend Communications, ECI Telematics, Microsoft, 3Com, and U.S. Robotics. This group's purpose was to design a protocol that would allow remote users to communicate securely over the Internet.

Although newer VPN protocols are available, PPTP is still widely used, because almost all VPN equipment vendors support PPTP. Another important benefit of PPTP is that it operates at layer 2 of the OSI model (the data link layer), allowing different networking protocols to run over a PPTP tunnel.

When connecting users to a remote system, encrypting the data transmissions is not the only facet of security. You must also authenticate the user. PPTP supports two separate technologies for accomplishing this: Extensible Authentication Protocol (EAP) and Challenge Handshake Authentication Protocol (CHAP).

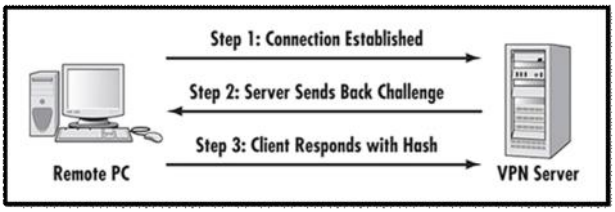
6.2.1.1 Extensible Authentication Protocol (EAP)

EAP was designed specifically with PPTP and is meant to work as part of PPP. EAP works from within PPP's authentication protocol. It provides a framework for several different authentication methods. EAP is meant to supplant proprietary authentication systems and includes a variety of authentication methods to be used, including passwords, challenge-response tokens, and public key infrastructure certificates.

6.2.1.2 Challenge Handshake Authentication Protocol (CHAP)

CHAP is actually a three-part handshaking (a term used to denote authentication processes) procedure. After the link is established,

the server sends a challenge message to the client machine originating the link. The originator responds by sending back a value calculated using a one-way hash function. The server checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise, the connection is usually terminated. This means that the authorization of a client connection has three stages.



What makes CHAP particularly interesting is that it periodically repeats the process. This means that even after a client connection is authenticated, CHAP repeatedly seeks to re-authenticate that client, providing a robust level of security.

6.2.2 L2TP

Layer 2 Tunnelling Protocol is an extension or enhancement of the Point-to-Point Tunnelling Protocol that is often used to operate virtual private networks over the Internet. Essentially, it is a new and improved version of PPTP. As its name suggests, it operates at the data link layer of the OSI model (like PPTP). Both PPTP and L2TP are considered by many experts to be less secure than IPSec. However, seeing IPSec used together with L2TP to create a secure VPN connection is not uncommon.
Like PPTP, L2TP supports EAP and CHAP. However, it also offers support for other authentication methods, for a total of six:

- EAP
- CHAP
- MS-CHAP
- PAP
- SPAP
- Kerberos

6.2.2.1 MS-CHAP

As the name suggests, MS-CHAP is a Microsoft-specific extension to CHAP. Microsoft created MS-CHAP to authenticate remote Windows workstations. The goal is to provide the functionality available on the LAN to remote users while integrating the encryption and hashing algorithms used on Windows networks.
Wherever possible, MS-CHAP is consistent with standard CHAP. However, some basic differences between MS-CHAP and standard CHAP include the following:

- The MS-CHAP response packet is in a format designed for compatibility with Microsoft's Windows networking products.
- The MS-CHAP format does not require the authenticator to store a clear-text or reversibly encrypted password.
- MS-CHAP provides authenticator-controlled authentication retry and password-changing mechanisms. These retry and password-changing mechanisms are compatible with the mechanisms used in Windows networks.
- MS-CHAP defines a set of reason-for-failure codes that are returned in the failure packet's message field if the authentication fails. These are codes that Windows software is able to read and interpret, thus providing the user with the reason for the failed authentication.

6.2.2.2 PAP

Password Authentication Protocol (PAP) is the most basic form of authentication. With PAP, a user's name and password are transmitted over a network and compared to a table of name-password pairs. Typically, the passwords stored in the table are encrypted. However, the transmissions of the passwords are in clear text, unencrypted, the main weakness with PAP. The basic authentication feature built into the HTTP protocol uses PAP. This method is no longer used and is only presented for historical purposes.

6.2.2.3 SPAP

Shiva Password Authentication Protocol (SPAP) is a proprietary version of PAP. Most experts consider SPAP somewhat more secure than PAP because the username and password are both encrypted when they are sent, unlike with PAP.
Because SPAP encrypts passwords, someone capturing authentication packets will not be able to read the SPAP password. However, SPAP is still susceptible to playback attacks (that is, a person records the exchange and plays the message back to gain fraudulent access). Playback attacks are possible because SPAP always uses the same reversible encryption method to send the passwords over the wire.

6.2.2.4 Kerberos

Kerberos is one of the most well-known network authentication protocols. It was developed at MIT and it's named from the mythical three-headed dog that guarded the gates to Hades.
Kerberos works by sending messages back and forth between the client and the server. The actual password (or even a hash of the password) is never sent. That makes it impossible for someone to intercept it. What happens instead is that the username is sent. The server then looks up the stored hash of that password, and uses that as an encryption key to encrypt data and send it back to the client. The client then takes the password the user entered, and uses that as a key to decrypt the data. If the user entered the wrong password, then it will never get decrypted. This is a clever way to verify the password without ever being transmitted. Authentication happens

with UDP (User Datagram Protocol) on port 88.
After the user's username is sent to the authentication service (AS), that AS will use the hash of the user password that is stored as a secret key to encrypt the following two messages that get sent to the client:

- **Message A:** Contains Client/TGS (Ticket Granting Service) session key encrypted with secret key of client
- **Message B:** Contains TGT (Ticket Granting Ticket) that includes client ID, client network address, and validity period

Remember, both of these messages are encrypted using the key the AS generated.
Then the user attempts to decrypt message A with the secret key generated by the client hashing the user's entered password. If that entered password does not match the password the AS found in the database, then the hashes won't match, and the decryption won't work. If it does work, then message A contains the Client/TGS session key that can be used for communication with the TGS. Message B is encrypted with the TGS secret key and cannot be decrypted by the client.
Now the user is authenticated into the system. However, when the user actually requests a service, some more message communication is required. When requesting services, the client sends the following messages to the TGS:

- **Message C:** Composed of the TGT from message B and the ID of the requested service
- **Message D:** Authenticator (which is composed of the client ID and the timestamp), encrypted using the Client/TGS session key

Upon receiving messages C and D, the TGS retrieves message B out of message C. It decrypts message B using the TGS secret key. This gives it the "Client/TGS session key". Using this key, the TGS decrypts message D (Authenticator) and sends the following two messages to the client:

- **Message E:** Client-to-server ticket (which includes the client ID, client network address, validity period, and client/server session key) encrypted using the service's secret key
- **Message F:** Client/server session key encrypted with the Client/TGS session key

Upon receiving messages E and F from TGS, the client has enough information to authenticate itself to the Service Server (SS). The client connects to the SS and sends the following two messages:

- **Message E:** From the previous step (the client-to-server ticket, encrypted using service's secret key)
- **Message G:** A new Authenticator, which includes the client ID and timestamp and is encrypted using the client/server session key

The SS decrypts the ticket (message E) using its own secret key to retrieve the client/server session key. Using the session key, the SS decrypts the Authenticator and sends the following message to the client to confirm its identity and willingness to serve the client:

- **Message H:** The timestamp found in client's Authenticator

The client decrypts the confirmation (message H) using the client/server session key and checks whether the timestamp is correct. If so, then the client can trust the server and can start issuing service requests to the server. The server provides the requested services to the client.
Below are some Kerberos terms to know:

- **Principal:** A server or client that Kerberos can assign tickets to.
- **Authentication Service (AS):** Service that authorizes the principal and connects them to the Ticket Granting Server. Note some books/sources say server rather than service.
- **Ticket Granting Service (TGS):** Provides tickets.
- **Key Distribution Centre (KDC):** A server that provides the initial ticket and handles TGS requests. Often it runs both AS and TGS services.

Realm: A boundary within an organisation. Each realm has its own AS and TGS.

- **Remote Ticket Granting Server (RTGS):** A TGS in a remote realm.
- **Ticket Granting Ticket (TGT):** The ticket that is granted during the authentication process.
- **Ticket:** Used to authenticate to the server. Contains identity of client, session key, timestamp, and checksum. Encrypted with server's key.
- **Session key:** Temporary encryption key.
- **Authenticator:** Proves session key was recently created. Often expires within 5 minutes.

IPSec

6.3 IPSec

Internet Protocol Security (IPSec) is a technology used to create virtual private networks. IPSec is used in addition to the IP protocol that adds security and privacy to TCP/IP communication. IPSec is incorporated with Microsoft operating systems as well as many other operating systems.

For example, the security settings in the Internet Connection Firewall that ships with Windows XP and later versions enables users to turn on IPSec for transmissions. IPSec is a set of protocols developed by the IETF (Internet Engineering Task Force; www.ietf.org) to support secure exchange of packets. IPSec has been deployed widely to implement VPNs.

IPSec has two encryption modes: transport and tunnel. The transport mode works by encrypting the data in each packet but leaves the header unencrypted. This means that the source and destination addresses, as well as other header information, are not encrypted. The tunnel mode encrypts both the header and the data.

This is more secure than transport mode but can work more slowly. At the receiving end, an IPSec-compliant device decrypts each packet. For IPSec to work, the sending and receiving devices must share a key, an indication that IPSec is a single-key encryption technology. IPSec also offers two other protocols beyond the two modes already described:

- **Authentication Header (AH):** The AH protocol provides a mechanism for authentication only. AH provides data integrity, data origin authentication, and an optional replay protection service. Data integrity is ensured by using a message digest that is generated by an algorithm such as HMAC-MD5 or HMAC-SHA. Data origin authentication is ensured by using a shared secret key to create the message digest.
- **Encapsulating Security Payload (ESP):** The ESP protocol provides data confidentiality (encryption) and authentication (data integrity, data origin authentication, and replay protection). ESP can be used with confidentiality only, authentication only, or both confidentiality and authentication.

Either protocol can be used alone to protect an IP packet, or both protocols can be applied together to the same IP packet.

IPSec can also work in two modes. Those modes are transport mode and tunnel mode. Transport mode is the mode where IPSec encrypts the data, but not the packet header. Tunnelling mode does encrypt the header as well as the packet data.

There are other protocols involved in making IPSec work. IKE, or Internet Key Exchange, is used in setting up security associations in IPSec. A security association is formed by the two endpoints of the VPN tunnel, once they decide how they are going to encrypt and authenticate. For example, will they use AES for encrypting packets, what protocol will be used for key exchange, and what protocol will be used for authentication?

All of these issues are negotiated between the two endpoints, and the decisions are stored in a security association (SA). This is accomplished via the IKE protocol. Internet Key Exchange (IKE and IKEv2) is used to set up an SA by handling negotiation of protocols and algorithms and to generate the encryption and authentication keys to be used.

The Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for authentication and key exchange. Once the IKE protocol sets up the SA, then it is time to actually perform the authentication and key exchange.

The first exchange between VPN endpoints establishes the basic security policy; the initiator proposes the encryption and authentication algorithms it is willing to use. The responder chooses the appropriate proposal and sends it to the initiator. The next exchange passes Diffie-Hellman public keys and other data.

Those Diffie-Hellman public keys will be used to encrypt the data being sent between the two endpoints. The third exchange authenticates the ISAKMP session. This process is called main mode. Once the IKE SA is established, IPSec negotiation (Quick Mode) begins.

Quick Mode IPSec negotiation, or Quick Mode, is similar to an Aggressive Mode IKE negotiation, except negotiation must be protected within an IKE SA. Quick Mode negotiates the SA for the data encryption and manages the key exchange for that IPSec SA.

In other words, Quick Mode uses the Diffie-Hellman keys exchanged in main mode, to continue exchanging symmetric keys that will be used for actual encryption in the VPN.

Aggressive Mode squeezes the IKE SA negotiation into three packets, with all data required for the SA passed by the initiator. The responder sends the proposal, key material, and ID, and authenticates the session in the next packet. The initiator replies by authenticating the session. Negotiation is quicker, and the initiator and responder ID pass in the clear.

SSL/TLS

6.4 SSL/TLS

A new type of firewall uses SSL (Secure Sockets Layer) or TLS (Transport Layer Security) to provide VPN access through a web portal. Essentially, TLS and SSL are the protocols used to secure websites. If you see a website beginning with HTTPS, then traffic to and from that website is encrypted using SSL or TLS. Today, we almost always mean TLS when we say SSL. It is just that many people became comfortable to saying SSL, and the phrase stuck. This should be obvious from the brief history of SSL/TLS presented here:

- Unreleased SSL v1 (Netscape).
- Version 2 released in 1995 but had many flaws.
- Version 3 released in 1996 (RFC 6101).
- Standard TLS 1.0, RFC 2246, released in 1999.
- TLS 1.1 defined in RFC 4346 in April 2006.
- TLS 1.2 defined in RFC 5246 in August 2008. It is based on the earlier TLS 1.1 spec.
- As of July 2017, TLS 1.3 is a draft and details have not been fixed yet.

In some VPN solutions the user logs in to a website, one that is secured with SSL or TLS, and is then given access to a virtual private network. However, visiting a website that uses SSL or TLS does not mean you are on a VPN. As a general rule most websites, such as banking websites, give you access only to a very limited set of data, such as your account balances. A VPN gives you access to the network, the same or similar access to what you would have if you were physically on that network.

Whether you are using SSL to connect to an e-commerce website or to establish a VPN, the SSL handshake process is needed to establish the secure/encrypted connection:

1. The client sends the server the client's SSL version number, cipher settings, session-specific data, and other information that the server needs to communicate with the client using SSL.

2. The server sends the client the server's SSL version number, cipher settings, session-specific data, and other information that the client needs to communicate with the server over SSL. The server also sends its own certificate, and if the client is requesting a server resource that requires client authentication, the server requests the client's certificate.

3. The client uses the information sent by the server to authenticate the server—e.g., in the case of a web browser connecting to a web server, the browser checks whether the received certificate's subject name actually matches the name of the server being contacted, whether the issuer of the certificate is a trusted certificate authority, whether the certificate has expired, and, ideally, whether the certificate has been revoked. If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client proceeds to the next step.

4. Using all data generated in the handshake thus far, the client (with the cooperation of the server, depending on the cipher in use) creates the pre-master secret for the session, encrypts it with the server's public key (obtained from the server's certificate, sent in step 2), and then sends the encrypted pre-master secret to the server.

5. If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case, the client sends both the signed data and the client's own certificate to the server along with the encrypted pre-master secret.

6. If the server has requested client authentication, the server attempts to authenticate the client. If the client cannot be authenticated, the session ends. If the client can be successfully authenticated, the server uses its private key to decrypt the pre-master secret, and then performs a series of steps (which the client also performs, starting from the same pre-master secret) to generate the master secret.

7. Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity (that is, to detect any changes in the data between the time it was sent and the time it is received over the SSL connection).

8. The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished.

9. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.

VPN Solutions

6.5 VPN Solutions

Regardless of which protocols you use for VPN, you must implement your choice in some software/hardware configuration. Many operating systems have built-in VPN server and client connections. These are generally fine for small office or home situations. However, they might not be adequate for larger scale operations in which multiple users connect via VPN. For those situations, a dedicated VPN solution might be necessary.

6.5.1 Cisco Solutions

Cisco offers VPN solutions, including a module that can be added to many of their switches and routers to implement VPN services. It also offers client-side hardware that is designed to provide an easy-to-implement yet secure client side for the VPN.

The main advantage of this solution is that it incorporates seamlessly with other Cisco products. Administrators using a Cisco firewall or Cisco router might find this solution to be preferable. However, this solution might not be right for those not using other Cisco products and those who do not have knowledge of Cisco systems. However, many attractive specifications for this product include the following:

- It can use 3DES encryption (an improved version of DES). But AES is preferred and strongly recommended.
- It can handle packets larger than 500 bytes.
- It can create up to 60 new virtual tunnels per second, a good feature if a lot of users might be logging on or off.

6.5.2 Openswan

The Openswan product (www.openswan.org/) is an open source VPN solution available for Linux operating systems. As an open source product, one of its biggest advantages is that it is free. Openswan uses IPSec, making it a highly secure VPN solution. Openswan supports either remote users logging on via VPN, or site-to-site connections. It also supports wireless connections. However, it does not support NAT (network address translation, the new alternative to proxy servers).

QUIZ:

- 1.What does L2TP stand for?
- 2.Which of the following is generally considered the least secure?
- 3.PPTP is based on which protocol?
- 4.The ESP Protocol provides data confidentiality and authentication.
- 5.PPTP is an acronym for which of the following?
- 6.Which of the following is an important security feature in CHAP
- 7.Openswan is a VPN solution provided by CISCO.
- 8.Which authentication protocols are available under PPTP?
- 9.What is the purpose of IKE?
- 10.Which of the following is a weakness in PPTP?