An Introduction to DHCP

Jan 19, 2006 By Dean Wilson In Linux Journal

DHCP stands for dynamic host configuration protocol. What it does is dynamically assign network settings from a server. In other words, instead of having to configure the parameters related to how your computer communicates with a network, it happens automatically.

Assigning an IP address dynamically is the most basic piece but there is a lot more to DHCP. This includes the netmask, host name, domain name, gateway and name servers. In addition, DHCP can supply other information such as a time server.

Many people are anti-DHCP, because they see it as a way that an ISP offers you an IP address that changes. This, of course, makes it difficult to advertise a server. On the other hand, DHCP can save you a lot of ongoing configuration work within your company or organization.

Besides the ISP-provided DHCP servers, they commonly exist in inexpensive router boxes. Netgear, Linksys and other vendors offer these systems with multiple LAN ports, an 802.11b wireless interface or both. The Netgear RP114 is an example of a wired LAN, while the Linksys WAP11 is an 802.11b type. Many other product choices are available. When you use one, the router box becomes the system that the ISP knows about, and all of your real computers hide behind this box.

Hide? Effectively, yes. What is visible to the public Internet is the router. The LAN has private IP addresses and uses network address translation (NAT) to handle connections from the internal systems to the Internet. Although this isn't really a firewall, NAT offers a basic level of protection.

Most routers in this class allow you to:

- Clone the MAC (hardware) address of one of your computers. This allows you to make the ISP think it is talking to a computer system you previously identified rather than to a router with possibly multiple machines connected to it.
- Handle static IP addresses. This means you could pick a local network address (192.168.1.*x*, for example) and assign specific addresses in this range.
- Dynamically assign IP addresses from a specified range. For example, the router could be configured to offer DHCP for 20 different addresses, say 192.168.1.100 thru 192.168.1.119.

That is the basics of "DHCP for Beginners". If you simply are trying to decide between using DHCP or a static IP address, this may be enough information. On the other hand, you could decide to run a DHCP server on a Linux system. In that case, there are more options.

Source:

http://www.linuxjournal.com/article/8820

Benefits of DHCP

Deploying DHCP on your enterprise network provides the following benefits:

- **Safe and reliable configuration.** DHCP minimizes configuration errors caused by manual IP address configuration, such as typographical errors, as well as address conflicts caused by a currently assigned IP address accidentally being reissued to another computer.
- **Reduced network administration.**
    - TCP/IP configuration is centralized and automated.
    - Network administrators can centrally define global and subnet-specific TCP/IP configurations.
    - Clients can be automatically assigned a full range of additional TCP/IP configuration values by using DHCP options.
    - Address changes for client configurations that must be updated frequently, such as remote access clients that move around constantly, can be made efficiently and automatically when the client restarts in its new location.
    - Most routers can forward DHCP configuration requests, eliminating the requirement of setting up a DHCP server on every subnet, unless there is another reason to do so.

DHCP is based on a client/server model, as illustrated in Figure 4.1.

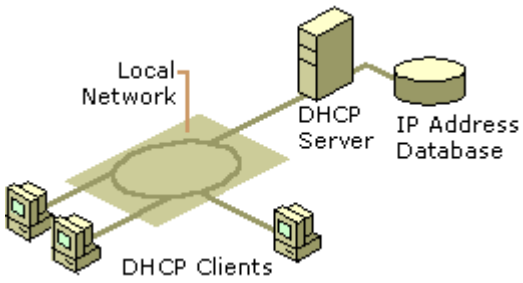


Figure 4.1 The Basic DHCP Model

The network administrator establishes one or more DHCP servers that maintain TCP/IP configuration information and provide address configuration to DHCP-enabled clients in the form of a lease offer. The DHCP server stores the configuration information in a database, which includes:

- Valid TCP/IP configuration parameters for all clients on the network.
- Valid IP addresses, maintained in a pool for assignment to clients, as well as reserved addresses for manual assignment.
- Duration of the lease offered by the server—the length of time for which the IP address can be used before a lease renewal is required.

A DHCP-enabled client, upon acceptance of a lease offer, receives:

- A valid IP address for the network it is joining.
- Additional TCP/IP configuration parameters, referred to as DHCP options.

Source:

http://technet.microsoft.com/en-us/library/cc961399.aspx

Introduction to WINS

The Windows Internet Naming Service (WINS) provides a dynamic database to register NetBIOS names and resolve them to IP addresses. Clients can dynamically register their NetBIOS names with a WINS server, and query the WINS server when they need to resolve a NetBIOS name to an IP address.

WINS solves the problem of registering and resolving NetBIOS names in a routed environment. In a nonrouted environment, NetBIOS names can be registered and resolved using local broadcasts. However, in a routed environment this poses a problem because routers are not normally configured to forward broadcasts between subnets.

By using WINS, name registration and renewal requests can be directed to a WINS server, thereby allowing name registration and renewal across subnets.

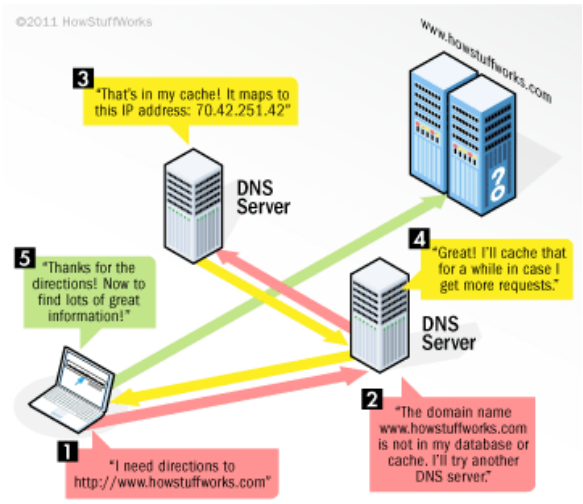
There are a number of other benefits to implementing a WINS server on the network including the following:

- It provides a dynamic database for registering NetBIOS names and resolving them to an IP address.
- It centralizes the management of NetBIOS names to IP addresses and eliminates the need for LMHOSTS files.
- It reduces the amount of broadcast traffic on the network. Clients can directly query the WINS server for name registration and resolution instead of performing a broadcast.
- It allows pre–Windows Server 2003 clients to locate domain controllers that are not on their local subnet.

Source:

http://www.informit.com/articles/article.aspx?p=600983&seqNum=2

How Domain Name Servers Work



When you enter a URL into your Web browser, your DNS server uses its resources to resolve the name into the IP address for the appropriate Web server.

Introduction to How DNS

If you've ever used the Internet, it's a good bet that you've used the **Domain Name System**, or **DNS**, even

without realizing it. DNS is a protocol within the set of standards for how computers exchange data on the Internet and on many private networks, known as the TCP/IP protocol suite. Its basic job is to turn a user-friendly **domain name** like "howstuffworks.com" into an Internet Protocol (IP) address like 70.42.251.42 that computers use to identify each other on the network. It's like your computer's GPS for the Internet.

Computers and other network devices on the Internet use an IP address to route your request to the site you're trying to reach. This is similar to dialing a phone number to connect to the person you're trying to call. Thanks to DNS, though, you don't have to keep your own address book of IP addresses. Instead, you just connect through a **domain name server**, also called a **DNS server** or **name server**, which manages a massive database that maps domain names to IP addresses.

Whether you're accessing a Web site or sending e-mail, your computer uses a DNS server to look up the domain name you're trying to access. The proper term for this process is **DNS name resolution**, and you would say that the DNS server resolves the domain name to the IP address. For example, when you enter "http://www.howstuffworks.com" in your browser, part of the network connection includes resolving the domain name "howstuffworks.com" into an IP address, like 70.42.251.42, for HowStuffWorks' Web servers.

You can always bypass a DNS lookup by entering 70.42.251.42 directly in your browser (give it a try). However, you're probably more likely to remember "howstuffworks.com" when you want to return later. In addition, a Web site's IP address can change over time, and some sites associate multiple IP addresses with a single domain name.

Without DNS servers, the Internet would shut down very quickly. But how does your computer know what DNS server to use? Typically, when you connect to your home network, Internet service provider (ISP) or WiFi network, the modem or router that assigns your computer's network address also sends some important network configuration information to your computer or mobile device. That configuration includes one or more DNS servers that the device should use when translating DNS names to IP address.

So far, you've read about some important DNS basics. The rest of this article dives deeper into domain name servers and name resolution. It even includes an introduction to managing your own DNS server. Let's start by looking at how IP addresses are structured and how that's important to the name resolution process.

DNS Servers and IP Addresses

You just learned that the primary job of a domain name server, or DNS server, is to resolve (translate) a domain name into an IP address. That sounds like a simple task, and it would be, except for the following points:

- There are billions of IP addresses currently in use, and most machines have a human-readable name as well.
- DNS servers (cumulatively) are processing billions of requests across the Internet at any given time.
- Millions of people are adding and changing domain names and IP addresses each day.

With so much to handle, DNS servers rely on network efficiency and Internet protocols. Part of the IP's effectiveness is that each machine on a network has a unique IP address in both the IPV4 and IPV6 standards managed by the Internet Assigned Numbers Authority (IANA). Here are some ways to recognize an IP address:

- An IP address in the IPV4 standard has four numbers separated by three decimals, as in: 70.74.251.42
- An IP address in the IPV6 standard has eight hexadecimal numbers (base-16) separated by colons, as in 2001:0cb8:85a3:0000:0000:8a2e:0370:7334. Because IPV6 is still a very new standard, we'll concentrate on the more common IPV4 for this article.
- Each number in an IPV4 number is called an "octet" because it's a base-10 equivalent of an 8-digit base-2 (binary) number used in routing network traffic. For example, the octet written as 42 stands for 00101010. Each digit in the binary number is the placeholder for a certain power of two from 2^0 to 2^7, reading from right to left. That means that in 00101010, you have one each of 2^1, 2^3 and 2^5. So, to get the base-10 equivalent, just add $2^1 + 2^3 + 2^5 = 2 + 8 + 32 = 42$. For more about how IP addresses are constructed, see our article "What is an IP address?"
- There are only 256 possibilities for the value of each octect: the numbers 0 through 255.
- Certain addresses and ranges are designated by the IANA as **reserved IP addresses**, which means they have a specific job in IP. For example, the IP address 127.0.0.1 is reserved to identify the computer you're currently using. So, talking to 127.0.0.1 is just talking to yourself!

Web servers and other computers that need a consistent point of contact use **static IP addresses**. This

means that the same IP address is always assigned to that system's network interface when it's online. To make sure that interface always gets the same IP address, IP associates the address with the Media Access Control (MAC) address for that network interface. Every network interface, both wired and wireless, has a unique MAC address embedded in it by the manufacturer.

What's in a name? For your domain, the name could make a difference in whether people remember or want to visit your Web site.

Domain Names

If we had to remember the IP addresses of all our favorite Web sites, we'd probably go nuts! Human beings are just not that good at remembering strings of numbers. We are good at remembering words, however, and that is where domain names come in. You probably have hundreds of domain names stored in your head, such as:

- google.com -- one of the most used domain names in the world
- mit.edu -- a popular EDU name
- bbc.co.uk -- a three-part domain name using the country code UK

You'll recognize domain names as having strings of characters separated by dots (periods). The last word in a domain name represents a **top-level domain**. These top-level domains are controlled by the IANA in what's called the Root Zone Database, which we'll examine more closely later. The following are some common top-level domains:

- COM -- commercial Web sites, though open to everyone
- NET -- network Web sites, though open to everyone
- ORG -- non-profit organization Web sites, though open to everyone
- EDU -- restricted to schools and educational organizations
- MIL -- restricted to the U.S. military
- GOV -- restricted to the U.S. government
- US, UK, RU and other two-letter country codes -- each is assigned to a domain name authority in the respective country

In a domain name, each word and dot combination you add before a top-level domain indicates a level in the domain structure. Each level refers to a server or a group of servers that manage that domain level. For example, "howstuffworks" in our domain name is a second-level domain off the COM top-level domain. An organization may have a hierarchy of **sub-domains** further organizing its Internet presence, like "bbc.co.uk" which is the BBC's domain under CO, an additional level created by the domain name authority responsible for the UK country code.

The left-most word in the domain name, such as www or mail, is a **host name**. It specifies the name of a specific machine (with a specific IP address) in a domain, typically dedicated to a specific purpose. A given domain can potentially contain millions of host names as long as they're all unique to that domain.

Because all of the names in a given domain need to be unique, there has to be some way to control the list and makes sure no duplicates arise. That's where registrars come in. A registrar is an authority that can assign domain names directly under one or more top-level domains and register them with InterNIC, a service of ICANN, which enforces uniqueness of domain names across the Internet. Each domain registration becomes part of a central domain registration database known as the whois database. Network Solutions, Inc. (NSI) was one of the first registrars, and today companies like GoDaddy.com offer domain registration in addition to many other Web site and domain management services. [source: InterNIC]

Domain name servers connect to each other across the Internet and cache lookup information to make name resolution more efficient.

The Distributed System

Every domain has a domain name server handling its requests, and there is a person or IT team maintaining the records in that DNS server's database. No other database on the planet gets as many requests as DNS servers, and they handle all those queries while also processing data updates from millions of people every day. That's one of the most amazing parts of DNS -- it is completely distributed throughout the world on millions of machines, managed by millions of people, and yet it behaves like a single, integrated database!

Because managing DNS seems like such a big job, most people tend to leave it to the IT professionals. However, by learning a little bit about how DNS works and how DNS servers are distributed across the Internet, you can manage DNS with confidence. The first thing to know is what the purpose of a DNS server is on the network where it resides. A DNS server will have one of the following as its primary task:

- Maintain a small database of domain names and IP addresses most often used on its own network, and delegate name resolution for all other names to other DNS servers on the Internet.
- Pair IP addresses with all hosts and sub-domains for which that DNS server has authority.

DNS servers that perform the first task are normally managed by your Internet service provider (ISP). As mentioned earlier, the ISP's DNS server is part of the network configuration you get from DHCP as soon as you go online. These servers reside in your ISP's data centers, and they handle requests as follows:

- If it has the domain name and IP address in its database, it resolves the name itself.
- If it doesn't have the domain name and IP address in its database, it contacts another DNS server on the Internet. It may have to do this multiple times.
- If it has to contact another DNS server, it caches the lookup results for a limited time so it can quickly resolve subsequent requests to the same domain name.
- If it has no luck finding the domain name after a reasonable search, it returns an error indicating that the name is invalid or doesn't exist.

The second category of DNS servers mentioned above is typically associated with Web, mail and other Internet domain hosting services. Though some hardcore IT gurus set up and manage their own DNS servers, hosting services have made DNS management much easier for the less technical audience. A DNS server that manages a specific domain is called the **start of authority** (**SOA**) for that domain. Over time, the results from looking up hosts at the SOA will propagate to other DNS servers, which in turn propagate to other DNS servers, and so on across the Internet.

This propagation is a result of each DNS server caching the lookup result for a limited time, known as its Time To Live (TTL), ranging from a few minutes to a few days. People managing a DNS server can configure its TTL, so TTL values will vary across the Internet. So, each time you look up "www.howstuffworks.com," it's possible that the DNS server for your ISP will find the lookup results "70.42.251.42" in its own cache if you or someone else using that server looked for it before within the server's TTL.

This great web of DNS servers includes the **root name servers**, which start at the top of the domain hierarchy for a given top-level domain. There are hundreds of root name servers to choose from for each top-level domain. Though DNS lookups don't have to start at a root name server, they can contact a root name server as a last resort to help track down the SOA for a domain.

Now that you know how DNS servers are interconnected to improve the name resolution process, let's look at how you can configure a DNS server to be the authority for your domain.

To increase visibility and prevent misdirected customers, many businesses create the same second-level domain name under several top-level domains worldwide.

Source:

http://www.howstuffworks.com/dns.htm/printable

The Routing and Remote Access service (RRAS)

RRAS is named for the two primary networking services that it provides.

Routing

A router is a device that manages the flow of data between network segments, or subnets. A router directs incoming and outgoing packets based on the information about the state of its own network interfaces and a list of possible sources and destinations for network traffic. By projecting network traffic and routing needs based on the number and types of hardware devices and applications used in your environment, you can better decide whether to use a dedicated hardware router, a software-based router, or a combination of both. Generally, dedicated hardware routers handle heavier routing demands best, and less expensive software-based routers handle lighter routing loads.

A software-based routing solution, such as RRAS in this version of Windows, can be ideal on a small, segmented network with relatively light traffic between subnets. Enterprise network environments that have a large number of network segments and a wide range of performance requirements might need a variety of hardware-based routers to perform different roles throughout the network.

Remote access

By configuring RRAS to act as a remote access server, you can connect remote or mobile workers to your organization's networks. Remote users can work as if their computers are directly connected to the network.

All services typically available to a directly connected user (including file and printer sharing, Web server access, and messaging) are enabled by means of the remote access connection. For example, on an RRAS server, clients can use Windows Explorer to make drive connections and to connect to printers. Because drive letters and universal naming convention (UNC) names are fully supported by remote access, most commercial and custom applications work without modification.

An RRAS server provides two different types of remote access connectivity:

- **Virtual private networking**. A virtual private network (VPN) is a secured, point-to-point connection across a public network, such as the Internet. A VPN client uses special TCP/IP-based protocols called tunneling protocols to make a connection to a port on a remote VPN server. The VPN server accepts the connection, authenticates the connecting user and computer, and then transfers data between the VPN client and the corporate network. Because the data traverses a public network, you must encrypt data sent over the connection to ensure privacy.

- **Dial-up networking**. In dial-up networking, a remote access client makes a dial-up telephone connection to a physical port on a remote access server by using the service of a telecommunications provider, such as analog telephone or ISDN. Dial-up networking over an analog phone or ISDN is a direct physical connection between the dial-up networking client and the dial-up networking server. You can encrypt data sent over the connection, but it is not required because the phone line is typically considered secure.

RRAS Features

- **Unicast IP routing**, where a router forwards data packets between a two-way, point-to-point connection.

- **IP Multicast**, which allows the sending, receiving and forwarding of IP multicast traffic, used for online multimedia presentations. Multicast traffic is sent to a single host but is processed by multiple hosts and is used for delivering real-time data to multiple users.

- **IPX router functionality,** through **RIP for IPX**, the primary routing protocol used in IPX internetworks; **Novell NetWare SAP for IPX**, a protocol used for the collection and distribution of service names and addresses; and **NetBIOS over IPX** broadcast forwarding.

- **AppleTalk router functionality** to forward AppleTalk packets and support the use of the Routing Table Maintenance Protocol (RTMP). Windows Server 2003 supports an AppleTalk protocol stack and AppleTalk routing software so that a server can connect to and provide routing for AppleTalk-based Macintosh networks.

- **Demand-Dial Routing,** which is the routing of packets over point-to-point links and allows you to connect to the Internet, to branch offices, or to implement router-to-router VPN connections. With demand-dial routing. IP and IPX traffic can be forwarded over persistent or on-demand WAN links.

- **Remote Access**, which allows the Windows Server 2003 computer to accept remote access, or dial-in connections from remote access clients.

- **VPN Server**, which supports PPTP and L2TP over IPSec and accepts remote access and router-to-router or demand-dial, VPN connections from remote access clients and calling routers.

- **RADIUS Client-Server**, which performs centralised authentication, authorisation, auditing and accounting (AAAA) of connections for dial-up and VPN remote access and demand-dial connections. It can be used along with RRAS and enables the use of a single or multiple vendor network of remote access or VPN equipment.

- **Simple Network Management Protocol (SNMP)** agent functionality with support for Internet MIB II.

- **Point-to-Point Protocol over Ethernet (PPPoE),** which provides a method of connecting individual client computers to a

service provider over a single broadband connection.

- **Integrated Firewall**, which can be enabled when the server is configured as an Internet gateway.
- **Mixed Media Bridging**, which allows Windows Server 2003 or Windows XP Professional to bridge between disparate segments, merging them into a single interface with a single IP address.
- **Integrated 802.11x wireless security,** through **Extensible Authentication Protocol (EAP)** or **Protected EAP (PEAP)**.
- **IPv6support,** via the **netsh** utility.
- **IAS Proxy**, which permits the Internet Authentication Service (IAS) to forward Remote Access Dial-In User Services (RADIUS) requests to another IAS or RADIUS server. This feature allows you to use RADIUS for dial-in, VPN and 802.11x wireless authentication.

Source:

http://www.sqa.org.uk/e-learning/NetInf104CD/page_03.htm

Virtual Private Networking

This topic has not yet been rated Rate this topic

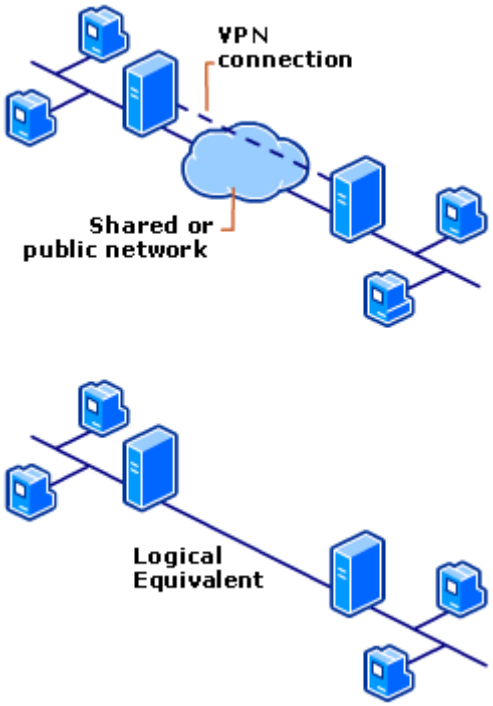
Updated: February 13, 2009

A virtual private network (VPN) is a point-to-point connection across a private or public network, such as the Internet. A VPN client uses special TCP/IP-based protocols called *tunneling protocols* that establish a secure channel between two computers through which they can send data. From the perspective of the two participating computers, there is a dedicated point-to-point link between them, though in reality the data is routed through the Internet as would be any other packet. In a typical VPN deployment, a client initiates a virtual point-to-point connection to a remote access server over the Internet. The remote access server answers the call, authenticates the caller, and transfers data between the VPN client and the organization's private network.

To emulate a point-to-point link, data is encapsulated, or wrapped, with a header. The header provides routing information that enables the data to traverse the shared or public network to reach its endpoint. To emulate a private link, the data being sent is encrypted for confidentiality. For more information about the tunneling protocols supported in this version of Windows, see VPN Tunneling Protocols.

For installation requirements, see Requirements for Installing RRAS as a VPN Server.

VPN connection



There are two types of VPN connections:

Remote access VPN

A remote access VPN connection enables a user working at home or on the road to access a server on a private network by using the infrastructure provided by a public network, such as the Internet. From the user's perspective, the VPN is a point-to-point connection between the client computer and the organization's server. The infrastructure of the shared or public network is irrelevant because it appears logically as if the data is sent over a dedicated private link.

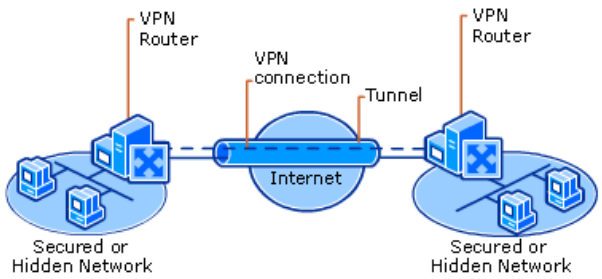
Site-to-site VPN

A site-to-site VPN connection (sometimes called a router-to-router VPN connection) enables an organization to have routed connections between separate offices or with other organizations over a public network while helping to maintain secure communications. When networks are connected over

the Internet, as shown in the following figure, a VPN-enabled router forwards packets to another VPN-enabled router across a VPN connection. To the routers, the VPN connection appears logically as a dedicated, data-link layer link.

A site-to-site VPN connection connects two private networks. The VPN server provides a routed connection to the network to which the VPN server is attached. The calling router authenticates itself to the answering router, and, for mutual authentication, the answering router authenticates itself to the calling router. In a site-to site VPN connection, the packets sent from either router across the VPN connection typically do not originate at the routers.

VPN connecting two remote sites across the Internet



Properties of VPN connections

- **Encapsulation**. Private data is encapsulated with a header that contains routing information that allows the data to traverse the transit network. For examples of encapsulation, see VPN Tunneling Protocols.

- **Authentication**. Authentication for VPN connections takes three different forms:

    - **User-level authentication by using Point-to-Point Protocol (PPP) authentication**. To establish the VPN connection, the VPN server authenticates the VPN client that is attempting the connection by using a PPP user-level authentication method and verifies that the VPN client has the appropriate authorization. If mutual authentication is used, the VPN client also authenticates the VPN server, which provides protection against computers that are masquerading as VPN servers.

    - **Computer-level authentication by using Internet Key Exchange (IKE)**. To establish an Internet Protocol security (IPsec) security association (SA), the VPN client and the VPN server use the IKE protocol to exchange either computer certificates or a preshared key. In either case, the VPN client and server authenticate each other at the computer level. Computer certificate authentication is is a much stronger authentication method and is therefore highly recommended. Computer-level authentication is used by Layer Two Tunneling Protocol (L2TP)/IPsec or IKE version 2 connections.

    - **Data origin authentication and data integrity**. To verify that the data sent on the VPN connection originated at the other end of the connection and was not modified in transit, the data contains a cryptographic checksum based on an encryption key known only to the sender and the receiver. Data origin authentication and data integrity are available for L2TP/IPsec and IKE version 2 connections.

- **Data encryption**. To ensure confidentiality of the data as it traverses the shared or public transit network, the data is encrypted by the sender and decrypted by the receiver. The encryption and decryption processes depend on both the sender and the receiver using a common encryption key.

  Intercepted packets sent along the VPN connection in the transit network are unintelligible to anyone who does not have the common encryption key. The length of the encryption key is an important security parameter. You can use computational techniques to determine the encryption key. However, such techniques require more computing power and computational time as the encryption keys get larger. Therefore, it is important to use the largest possible key size to ensure data confidentiality.

Source:

http://technet.microsoft.com/en-us/library/dd469653%28v=ws.10%29.aspx

VPN Tunneling Protocols

Updated: February 13, 2009

Tunneling enables the encapsulation of a packet from one type of protocol within the datagram of a different protocol. For example, VPN uses Point-to-Point Tunneling Protocol (PPTP) to encapsulate IP packets over a public network, such as the Internet. You can configure a VPN solution based on PPTP, Layer Two Tunneling Protocol (L2TP), Secure Socket Tunneling Protocol (SSTP), or Internet Protocol security (IPsec) using Internet Key Exchange version 2 (IKEv2).

PPTP, L2TP, and SSTP depend heavily on the features originally specified for Point-to-Point Protocol (PPP). PPP was designed to send data across dial-up or dedicated point-to-point connections. For IP, PPP encapsulates IP packets within PPP frames and then transmits the encapsulated PPP packets across a point-to-point link. PPP was originally defined as the protocol to use between a dial-up client and a network access server. Unlike the other tunnel types, IKEv2 does not run on top of PPP.

PPTP

PPTP allows multiprotocol traffic to be encrypted and then encapsulated in an IP header to be sent across an IP network or a public IP network, such as the Internet. PPTP can be used for remote access and site-to-site VPN connections. When using the Internet as the public network for VPN, the PPTP server is a PPTP-enabled VPN server with one interface on the Internet and a second interface on the intranet.

L2TP/IPsec

L2TP/IPsec allows multiprotocol traffic to be encrypted and then sent over any medium that supports point-to-point datagram delivery, such as IP or Asynchronous Transfer Mode (ATM). L2TP is a combination of PPTP and Layer 2 Forwarding (L2F), a technology developed by Cisco Systems, Inc. L2TP represents the best features of PPTP and L2F.

Unlike PPTP, the Microsoft implementation of L2TP does not use MPPE to encrypt PPP datagrams. L2TP uses IPsec in Transport Mode for encryption services. The combination of L2TP and IPsec is known as L2TP/IPsec.

Both L2TP and IPsec must be supported by both the VPN client and the VPN server. Client support for L2TP is built in to the Windows remote access clients, and VPN server support for L2TP is built in to the Windows Server operating system.

SSTP

Secure Socket Tunneling Protocol (SSTP) is a tunneling protocol that uses the HTTPS protocol over TCP port 443 to pass traffic through firewalls and Web proxies that might block PPTP and L2TP/IPsec traffic. SSTP provides a mechanism to encapsulate PPP traffic over the Secure Sockets Layer (SSL) channel of the HTTPS protocol. The use of PPP allows support for strong authentication methods, such as EAP-TLS. SSL provides transport-level security with enhanced key negotiation, encryption, and integrity checking.

When a client tries to establish a SSTP-based VPN connection, SSTP first establishes a bidirectional HTTPS layer with the SSTP server. Over this HTTPS layer, the protocol packets flow as the data payload.

Authentication Protocols

Extensible Authentication Protocol (EAP) (Source: RFC3748)

- an authentication framework which supports multiple authentication methods.

- It typically runs directly over data link layers such as Point-to-Point (PPP) or IEEE802, without requireing IP.

- It provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees.

- It may be used on dedicated links, as well as switched circuits, and wired as well as wireless links.

- It has been implemented with honts and routers that connect via switched circuits or dial-up lines using PPP.

- It has also been implemented with switches and access points ussing IEEE 802.

- One of the advantages of the EAP architecture is its flexibility.

- It is used to select a specific authentication mechanism, typically after the authenticator requests more information in order to determine the specific authentication method, with the authenticator acting as a pass-through for some or all methods and peers.

- Within this document, authenticator requirements apply regardless of whether the authenticator is

operating as a pass-through or not. where the requirement is meant to apply to either the authenticator or backend authentication server, depending on where the EAP authentication is terminated, the ther "EAP Server" will be used.

Applicability

EAP was designed for use in network access authentication, where IP layer connectivity may not be available. Since EAP does not require IP connectivity, it provides just enough support for the reliable transport fo authentication protocols, and no more.

EAP is a lock-step protocol which only supports a single packet in flight. As a result, EAP cann't efficiently transport bulk data, unlike transport protocols like TCP.

While EAP provides support for retransmission, it assumes ordering guarantees provided by the lower layer, so out of order reception is not supported.

Since EAP does not support fragmentation and reassembly, EAP authentication methods generating payloads larger than the minimum EAP MTU need to provide fragmentation support.

While authentication methods such as EAP-TLS provides support for fragmentation and reassembly basic EAP doesn't.

EAP authentication is initiated by the server(authenticator), whereas many authentication protocols are initiated by the client. As a result it may be necessary for an authentication algorithm to add one or two additional messages in ordeer to run over EAP.

Where Certification-based authentication is supported, the number of additional roundtrips may be much larger due to fragmentation of certificate chains.

Where EAP runs over a lower layer in which significant packet loss is expected, or where the connection between the authenticator and authentication server server experiences significant packet loss, EAP methods requireing many round-trips can experience difficulties. In these situations, use of EAP methodswith fewer roundtrips is advisable.

The EAP authentication exchange proceeds as follows:

[1] The authenticator sends a Request to authenticate the peer. The Request has a Type field to indicate what is being requested. Examples of Request Types include Identity, MD5-challenge, etc. The MD5-challenge Type corresponds closely to the CHAP authentication protocol [RFC1994]. Typically, the authenticator will send an initial Identity Request; however, an initial Identity Request is not required, and MAY be bypassed. For example, the identity may not be required where it is determined by the port to which the peer has connected (leased lines, dedicated switch or dial-up ports), or where the identity is obtained in another fashion (via calling station identity or MAC address, in the Name field of the MD5-Challenge Response, etc.).

[2] The peer sends a Response packet in reply to a valid Request. As with the Request packet, the Response packet contains a Type field, which corresponds to the Type field of the Request.

[3] The authenticator sends an additional Request packet, and the peer replies with a Response. The sequence of Requests and Responses continues as long as needed. EAP is a 'lock step' protocol, so that other than the initial Request, a new Request cannot be sent prior to receiving a valid Response. The authenticator is responsible for retransmitting requests as described in Section 4.1. After a suitable number of retransmissions, the authenticator SHOULD end the EAP conversation. The authenticator MUST NOT send a Success or Failure packet when retransmitting or when it fails to get a response from the peer.

[4] The conversation continues until the authenticator cannot authenticate the peer (unacceptable Responses to one or more Requests), in which case the authenticator implementation MUST transmit an EAP Failure (Code 4). Alternatively, the authentication conversation can continue until the authenticator determines that successful authentication has occurred, in which case the authenticator MUST transmit an EAP Success (Code 3).

Advantages:

o The EAP protocol can support multiple authentication mechanisms without having to pre-negotiate a particular one.

o Network Access Server (NAS) devices (e.g., a switch or access point) do not have to understand each authentication method and MAY act as a pass-through agent for a backend authentication server. Support for pass-through is optional. An authenticator MAY authenticate local peers, while at the same time acting as a pass-through for non-local peers and authentication methods it does not implement locally.

o Separation of the authenticator from the backend authentication server simplifies credentials management and policy decision making.

Disadvantages:

o For use in PPP, EAP requires the addition of a new authentication Type to PPP LCP and thus PPP implementations will need to be modified to use it. It also strays from the previous PPP authentication model of negotiating a specific authentication mechanism during LCP. Similarly, switch or access point implementations need to support [IEEE-802.1X] in order to use EAP.

o Where the authenticator is separate from the backend authentication server, this complicates the security analysis and, if needed, key distribution.

Password Authentication Protocol (PAP)

It is a simple authentication procedure with a two-step process;

1. The user who wants to access a system sends an authentication identification (usually the user name) and a password.
2. The System checks the validity of the identification and password and either accepts or denies connection.

Figure below shows the three types of packets used by PPP and how they are actually exchanged. The three PAP packets are authenticate-request, authenticate-ack, and authenticate-nak. The first packet is used by the user to send the user name and password. The second is used by the sytem to allow access. The third is used by the system to deny access.

Challenge Handshake Authentication Protocol (CHAP)

It is a three way hand-shakeing authentication protocol that provides greater security than PAP In this method, the password is kept secret; it is never sent online.

1. The system sends the user a challenge packet containing a challenge value, usually a few bytes.
2. The user applies a predefined function that takes the challenge value and the user's own password and creates a result. The user sends the result in the response packet to the system.
3. The system does the same. It applies the same function to the password of the user (known to the system) and the challenge value to create a result. If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied. CHAP is ore secure than PAP, especially if the system continuously changes the challenge value. Even if the intruder learns the challenge value and the result, the password is still secret. Fig below shows the packets and how they are used.

Shiva Password Authentication Protocol (SPAP)

Shiva Password Authentication Protocol (SPAP) is a simple encrypted password authentication protocol supported by Shiva remote access servers. With SPAP, the remote access client sends an encrypted password to the remote access server. SPAP uses a two-way encryption algorithm. The remote access server decrypts the password and uses the plaintext form to authenticate the remote access client.