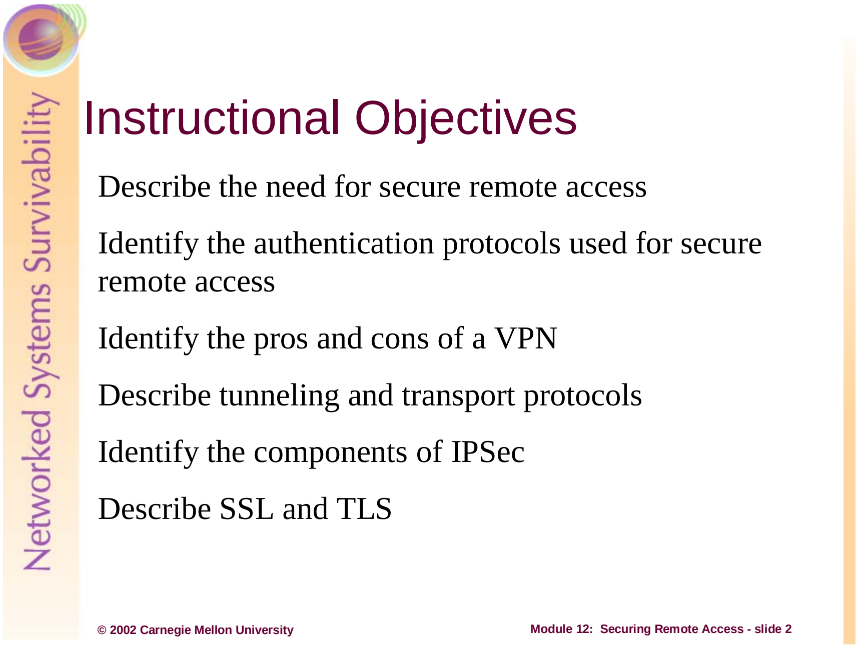
Information Security for Technical Staff

Module 12:

Securing Remote Access

Networked Systems Survivability
CERT® Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

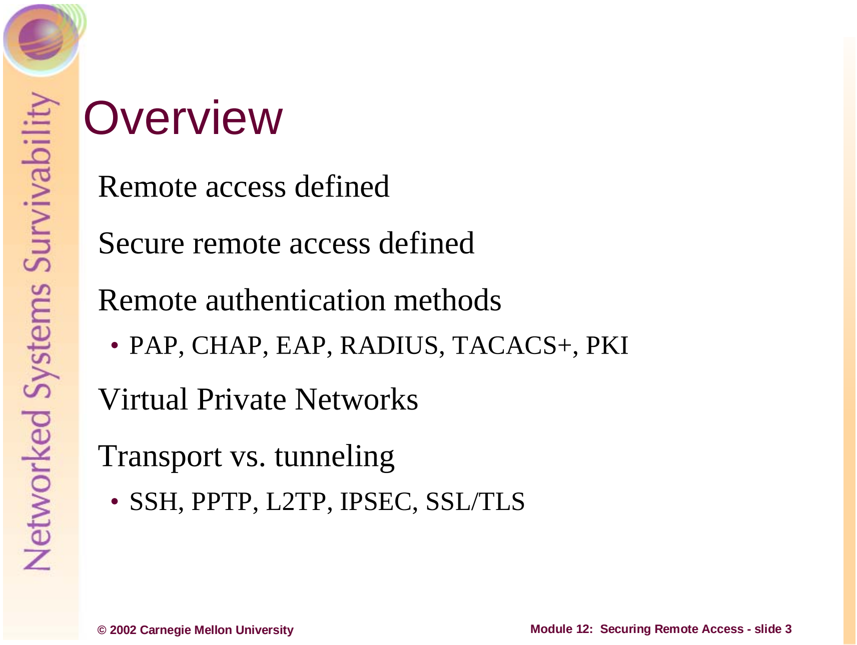

Networked Systems Survivability

Instructional Objectives

- Describe the need for secure remote access
- Identify the authentication protocols used for secure remote access
- Identify the pros and cons of a VPN
- Describe tunneling and transport protocols
- Identify the components of IPSec
- Describe SSL and TLS

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 2



Networked Systems Survivability

Overview

Remote access defined

Secure remote access defined

Remote authentication methods

- PAP, CHAP, EAP, RADIUS, TACACS+, PKI

Virtual Private Networks

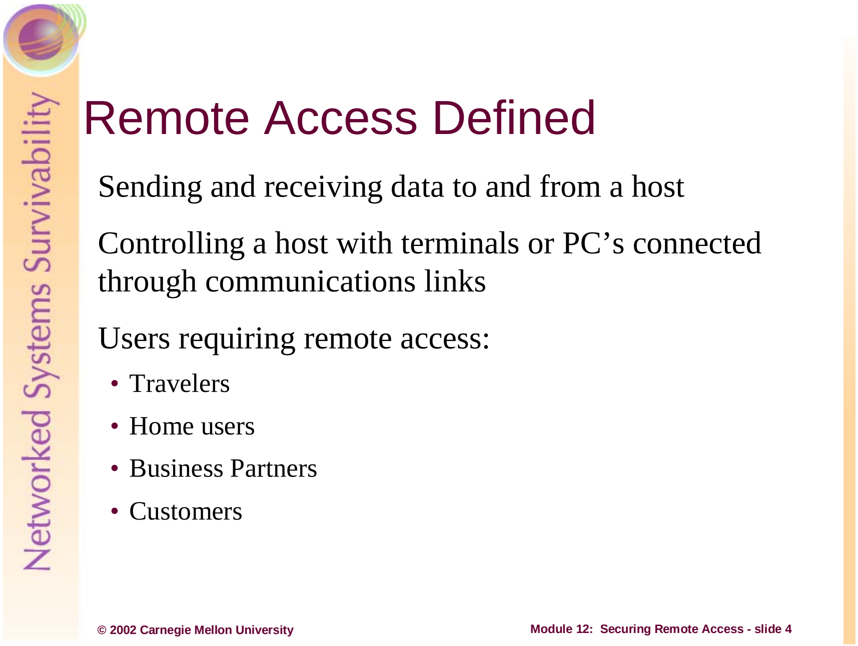
Transport vs. tunneling

- SSH, PPTP, L2TP, IPSEC, SSL/TLS

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 3

In order to understand the requirements and need for secure remote access, we will start by defining remote access, and pointing out some of its inherent vulnerabilities and shortcomings. We'll then discuss the requirements for secure remote access. We will then cover some of the more common means of authentication, and discuss their strengths and weaknesses. We'll then define and describe virtual private networks, and discuss some of the main protocols used to create them.


Networked Systems Survivability

Remote Access Defined

Sending and receiving data to and from a host

Controlling a host with terminals or PC’s connected through communications links

Users requiring remote access:

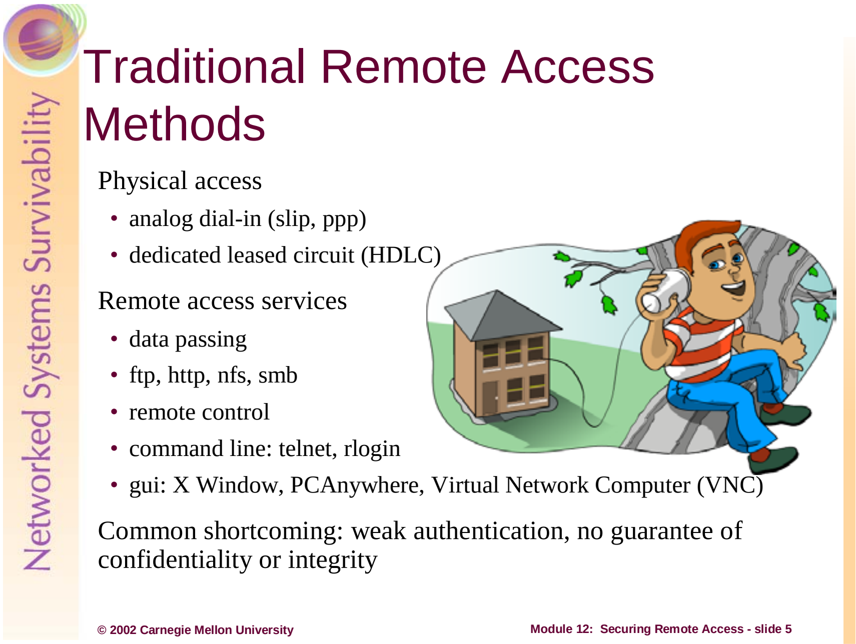
- Travelers
- Home users
- Business Partners
- Customers

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 4

Remote access can define different types of activities that accomplish a similar goal: the ability to access information from a location that is not local. Sending and receiving data to and from a host with access to the information can accomplish this goal. This process of sending and receiving data is normally accomplished through the use of network protocols. Certain network protocols will allow a remote user the ability to control in some fashion this host with access to the information. This control can be accomplished by allowing the manipulation of either a command shell interface or a graphical user interface on this host.

For most every business and network today, there is a need for remote access. It may come in the form of travelers, home users, geographically separated users at remote offices, partners who require access to your network for completion of business transactions, and possibly customers placing orders or checking their account information. So, in virtually every network today, there is need for remote access.



Networked Systems Survivability

Traditional Remote Access Methods

Physical access

- analog dial-in (slip, ppp)
- dedicated leased circuit (HDLC)

Remote access services

- data passing
- ftp, http, nfs, smb
- remote control
- command line: telnet, rlogin
- gui: X Window, PCAnywhere, Virtual Network Computer (VNC)

Common shortcoming: weak authentication, no guarantee of confidentiality or integrity

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 5

Historically, remote access was limited to ‘direct’ physical access. This physical access, which combined authentication protocols (software) with the facilities of a service provider (telephone company), was generally limited to dial-up or leased line services. The connection protocols which were most prevalent for dial-up service, and which are still in use for most dial-up applications, are SLIP (Serial Line Internet Protocol) and PPP (Point to Point Protocol).

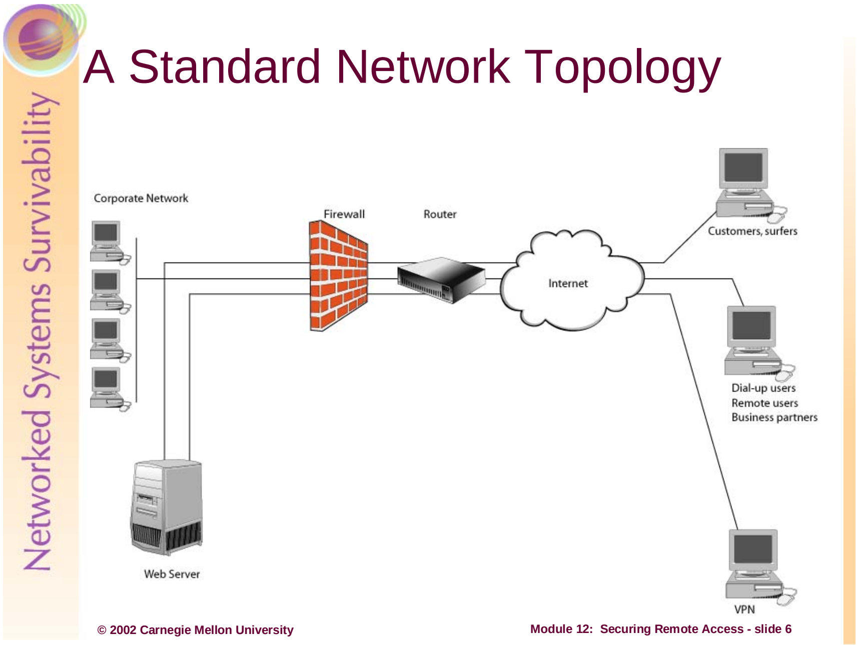
Serial Line IP (SLIP), documented in RFC 1055, was the first protocol for relaying IP packets over dial-up lines. It defines an encapsulation mechanism, but little else. There is no support for dynamic address assignment, authentication, link testing, or multiplexing different protocols over a single link. SLIP has been largely supplanted by PPP.

PPP provides Layer 2 service. Essentially, it packages your computer's TCP/IP packets in a PPP header and forwards those packets to the PPP server. The PPP server removes the PPP header and forwards the TCP/IP packet to the appropriate place (for ISPs, that is the Internet; for corporate environments, it is likely the corporate network). Additionally, PPP passes authentication passwords only in the clear, which, of course, is a security issue.

Most dedicated leased lines use High Level Data Link Control (HDLC) as the authentication and control protocol. HDLC is a protocol that organizes data into a unit (called a *frame*) and sends it across a network to a destination that verifies its successful arrival. The HDLC protocol also manages the flow or pacing at which data is sent. HDLC is one of the most commonly used layer 2 protocols. Variations of HDLC are also used for the public networks that use the X.25 communications protocol and for frame relay, a protocol used in both and Wide Area Network, public and private. Authentication is very weak, as the facilities (run by the phone company) are configured to forward packets through the network of the service provider only between the source to the destination addresses.

There are many services that are used by remote access users. They include sharing/sending data by many means (FTP, SMB, NFS, etc.), command line or shell access (via rlogin, telnet, etc.), remote control of a host, or using graphical interface tools for any of a number of purposes (through tools such as PCAnywhere or an X Windows session).

Remote access measures and applications of the past may have met many functionality requirements, but they all generally have one common shortcoming: poor (or non-existent) authentication of remote users. In addition, with some easily procured tools today in the hands of attackers, remote access methods of the past often fall short of keeping traffic confidential and unaltered. This has resulted in recent efforts to strengthen and secure remote access capabilities.



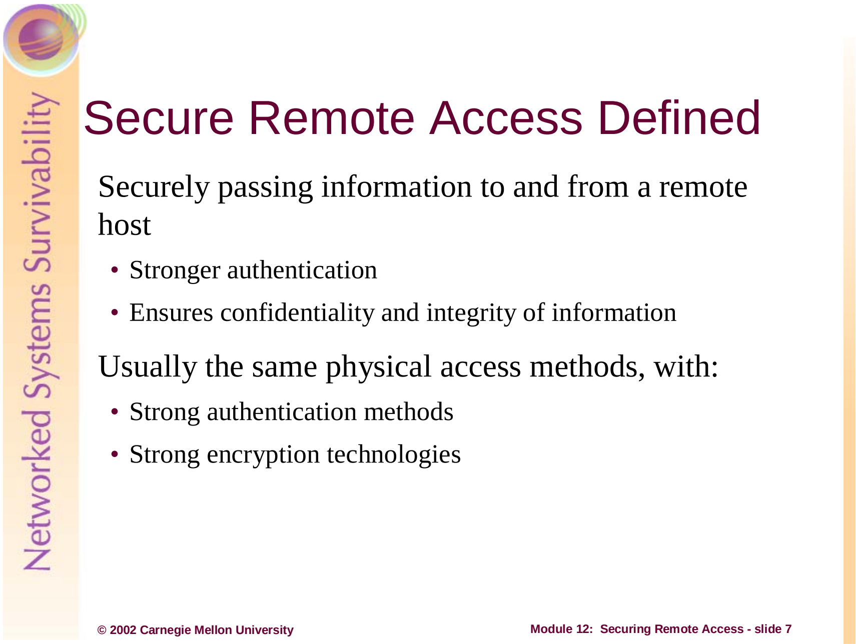
A Standard Network Topology

Networked Systems Survivability

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 6

As the enterprise has expanded and included more and more remote users, the network topology of the enterprise has, of course, also changed. Included now in the remote access scheme for the typical enterprise are a number of remote users. Customers and web surfers, who connect for various reasons which may or may not need to be secured. Dial-up users, remote workers, travelers, and business partners may connect through dedicated infrastructure (i.e. leased lines, frame relay, etc.) or through dial up POTS lines. Additionally, there may be other remote users or partners who will benefit from virtual private network connections – using the public infrastructure to connect to the enterprise. All of these conditions use the principles of secure remote access to keep the connections confidential.



Networked Systems Survivability

Secure Remote Access Defined

Securely passing information to and from a remote host

- Stronger authentication
- Ensures confidentiality and integrity of information

Usually the same physical access methods, with:

- Strong authentication methods
- Strong encryption technologies

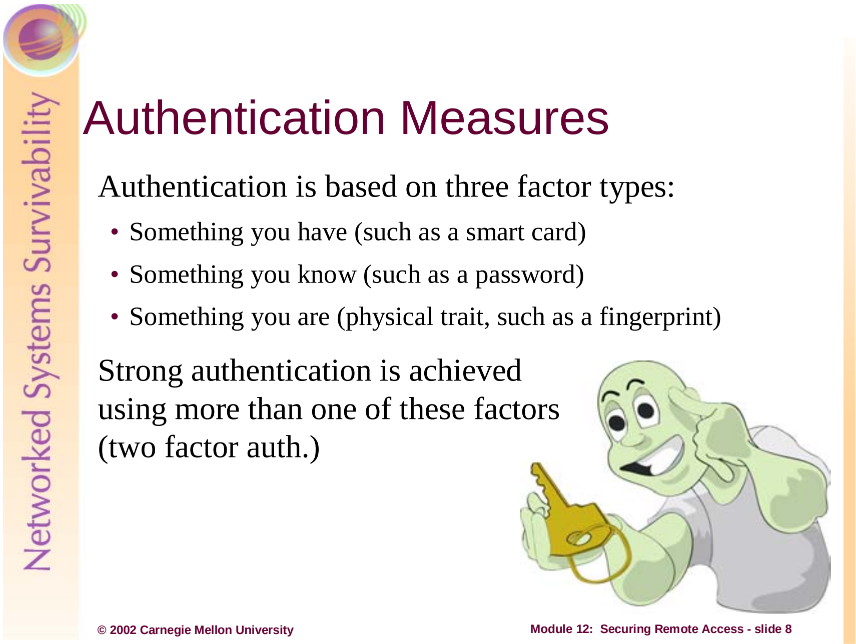
© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 7

There are various protocols and technology that are used to secure remote access for many users of varied needs. This includes provisions for confidentiality (so that the information being passed is free from being seen by unauthorized individuals), and authentication of remote users (to provide positive identification of the remote users). It also includes the ability of implementing strong encryption methods so that it is more difficult for attackers to compromise the data transmission. The basic data transmission methods may be the same as traditional remote access (dial-up, leased line, etc.), or may include the Internet.

In today's increasingly complex and distributed enterprise, it is imperative to consider every 'remote' connection, whether WAN, extranet, or remote access user, as a potentially dangerous and exploitable portal into the network. Therefore, it is critical to understand the range of options available to network administrators for securing these remote connections – including hardware, software, and protocols.

Nearly every enterprise is now dealing with increased remote access requirements. These come in many forms – telecommuters, business partners, customers, and other potential stakeholders. The required access controls that will be implemented aside, there is still a great deal of work to do just to secure the transmission systems and traffic of these remote users. Whether they are customers submitting credit card or social security numbers or they are telecommuters accessing sensitive corporate information from their home, it is negligent to allow transmissions that can be intercepted and potentially exploited by attackers. The threat of such interception and potential exploit will be sufficient to drive many potential customers away, and may cost valuable productivity from remote workers. Therefore, to meet the goals and missions of the organization, we must be prepared to proactively secure these connections. This securing of connections ensures the survivability of our organization's mission and data.



Networked Systems Survivability

Authentication Measures

Authentication is based on three factor types:

- Something you have (such as a smart card)
- Something you know (such as a password)
- Something you are (physical trait, such as a fingerprint)

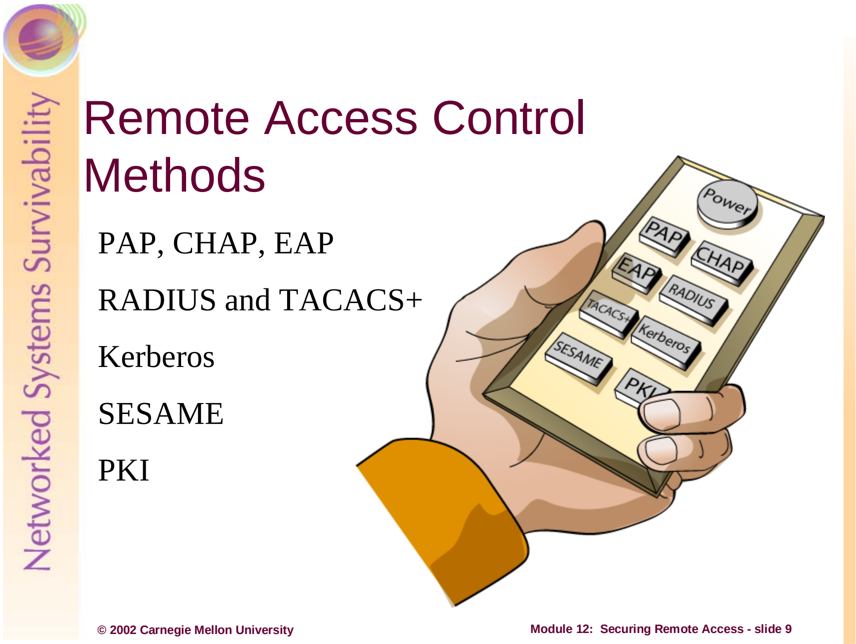
Strong authentication is achieved using more than one of these factors (two factor auth.)

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 8

Authentication is a very important part of information security, and it is a critical part of secure remote access as well. Consider for a moment what is happening as we allow remote users access to our network and systems. We give them the same levels of permissions as they would have if they were local to the network. We have opened our network up to users whom we cannot see, and are trusting that they are who they say they are. There is only one-way to be sure they are who they say they are: strong authentication.

As we discussed in previous modules, authentication can be based on any number of the following factors: something you have (such as a smart card, an ATM card, a SecurID token, etc.), something you know (password, pass phrase, PIN, etc.), or something you are (biometrics – retina, fingerprint, face geometry, etc.). It is best to use two of those factors when authenticating a user for access into the network. Because of the lack of physical presence for a remote user who is accessing the network, it is extremely important to have these strong authentication methods in place. The ability to replay authentication information reinforces the need to have secure remote access control processes.



Networked Systems Survivability

Remote Access Control Methods

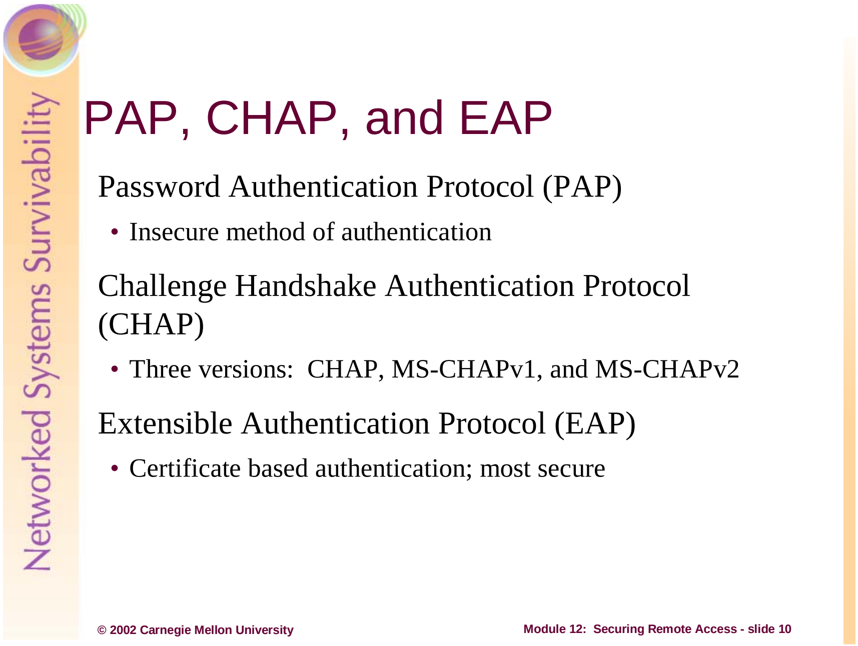
- PAP, CHAP, EAP
- RADIUS and TACACS+
- Kerberos
- SESAME
- PKI

Power
PAP
CHAP
EAP
RADIUS
TACACS+
Kerberos
SESAME
PKI

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 9

There are many protocols that control the remote access session. They include Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), Extensible Authentication Protocol (EAP), Remote Authentication Dial In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), Kerberos, Secure European System in A Multi-vendor Environment (SESAME), and Public Key Infrastructure (PKI).



Networked Systems Survivability

PAP, CHAP, and EAP

Password Authentication Protocol (PAP)

- Insecure method of authentication

Challenge Handshake Authentication Protocol (CHAP)

- Three versions: CHAP, MS-CHAPv1, and MS-CHAPv2

Extensible Authentication Protocol (EAP)

- Certificate based authentication; most secure

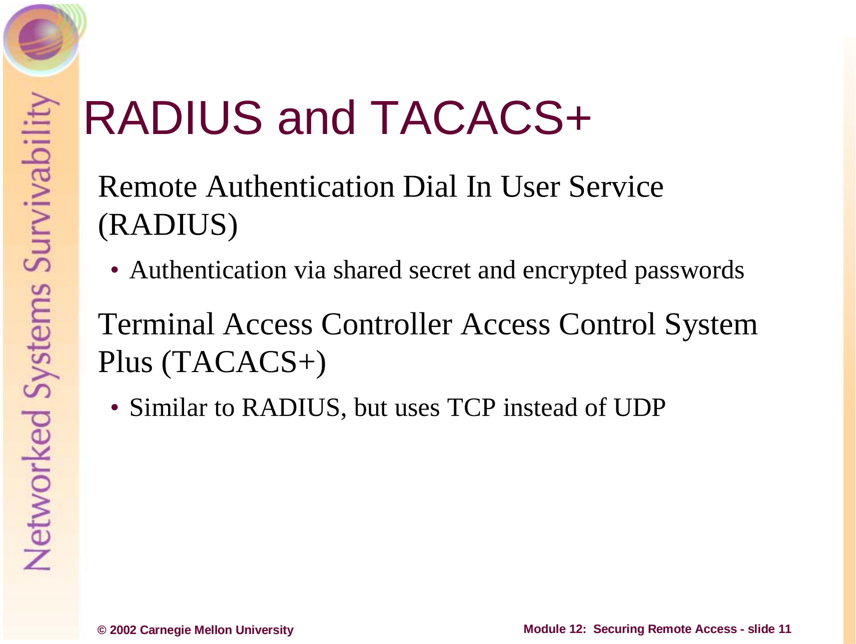
© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 10

PAP provides a simple method for the peer to establish its identity using a 2-way handshake. This is done only upon initial link establishment. An ID/Password pair is repeatedly sent by the peer to the authenticator until authentication is acknowledged or the connection is terminated. PAP is not a strong authentication method. Passwords are sent over the circuit "in the clear", and there is no protection from playback or repeated trial and error attacks. The peer is in control of the frequency and timing of the attempts. Therefore, PAP is not suitable for secure remote access.

CHAP is used to periodically verify the identity of the peer using a 3-way handshake. This is done upon initial link establishment, and MAY be repeated anytime after the link has been established. After the link is established, the authenticator sends a "challenge" message to the peer. The peer responds with a value calculated using the challenge and a "one-way hash" of a shared secret password. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise the connection SHOULD be terminated. CHAP provides protection against playback attack through the use of an incrementally changing identifier and a variable challenge value. The use of repeated challenges is intended to limit the time of exposure to any single attack. The authenticator is in control of the frequency and timing of the challenges. This authentication method depends upon a "secret" known only to the authenticator and that peer. The secret is not sent over the link. This method is most likely used where the same secret is easily accessed from both ends of the link.

EAP is a general authentication protocol which supports multiple authentication mechanisms. EAP does not select a specific authentication mechanism until later in the authentication process. This allows the authenticator to request more information before determining the specific authentication mechanism. This also permits the use of a "back-end" server which actually implements the various mechanisms while the PPP authenticator merely passes through the authentication exchange. EAP also can use the following for authentication: one-time passwords, digital certificates, generic token cards, and many others.


Networked Systems Survivability

RADIUS and TACACS+

Remote Authentication Dial In User Service (RADIUS)

- Authentication via shared secret and encrypted passwords

Terminal Access Controller Access Control System Plus (TACACS+)

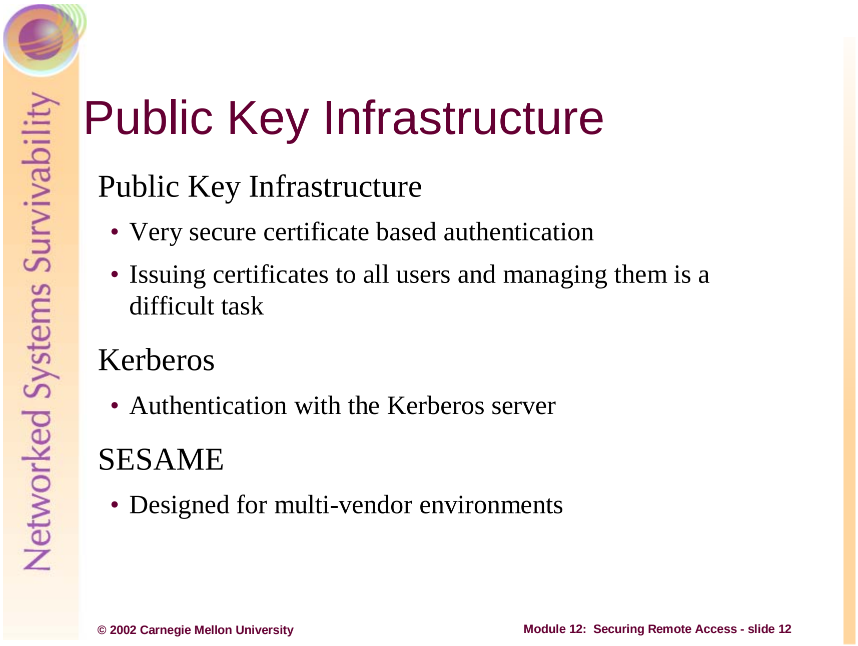
- Similar to RADIUS, but uses TCP instead of UDP

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 11

Transactions between the client and RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and RADIUS server, to eliminate the possibility that someone snooping on an unsecure network could determine a user's password.

TACACS+ is a protocol that is similar in many ways to RADIUS. However, TACACS+ runs over TCP instead of UDP as does RADIUS, and as such the transport is more reliable and less sensitive to disruption of the lower layers. TACACS+ also separates authentication and authorization, whereas RADIUS provides a user profile with the authentication that defines all the user-specific parameters. The separation allows you, for example, to use Kerberos authentication together with TACACS+ authorization.



Networked Systems Survivability

Public Key Infrastructure

Public Key Infrastructure

- Very secure certificate based authentication
- Issuing certificates to all users and managing them is a difficult task

Kerberos

- Authentication with the Kerberos server

SESAME

- Designed for multi-vendor environments

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 12

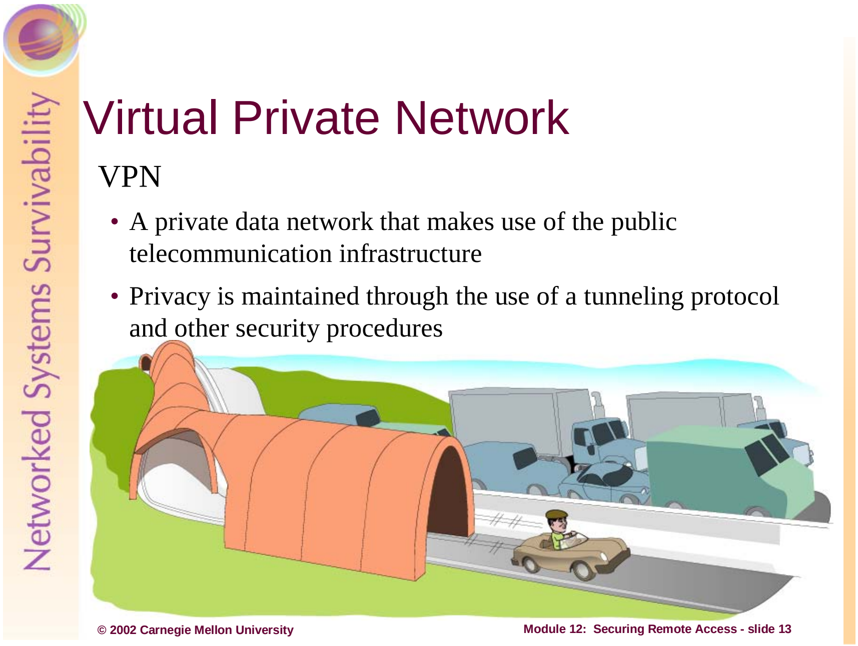
Use of Public Key Infrastructure (PKI) to authenticate remote users has some distinct advantages. It relies on ‘something you have’ as well as the ‘something you know.’ The something you have is your private key or digital certificate. The something you know is a pass phrase or PIN. The requirements for strong PKI authentication include:

- Every remote user must have a smart card or other media that contains their Private key.
- Every remote user must know the Public Key of its home authentication server.
- Every remote user must have a unique Identification that should be created and distributed by the authentication server.
- Every Public Key must be registered with the authentication server and must only be retrieved by providing a mean for its authentication.
- The authentication server may act as a broker for the retrieval of Public Keys of nodes that belong to different domains.
- The keys must be refreshed periodically to reinforce the security of the system.
- Every remote host MUST have the capability of performing cryptographic calculations such as encryption.
- In order to communicate with the network, the remote user must provide information to prove its identity to the authentication server.
- Every mobile entity, including the corresponding node, MUST support at least the SHA-1 or MD5 algorithms.

Kerberos, which has been covered extensively in another module, can be used as a means of authentication. See the module on Securing Network Infrastructure for a description of the Kerberos authentication process.

Secure European System in A Multi-vendor Environment (SESAME) is a protocol in development to handle authentication issues for the common multi-vendor enterprise. The authentication within SESAME is based upon the Kerberos system but adds several extra features, including: logon using digital signatures preventing off line dictionary attacks; mutual authentication between clients and servers; and cross-realm authentication using public key cryptography. There is a central database of the access control information associated with each user. This database is kept on and managed by a Privilege

Attribute Server (PAS). Access control information is conveyed to application servers in the form of ECMA 219 Privilege Attribute Certificates (PACs). PACs are carried inside a Kerberos protocol extension. The Kerberos rules for handling extension fields ensure that servers that do not know how to process PACs can simply ignore them.



Networked Systems Survivability

Virtual Private Network

VPN

- A private data network that makes use of the public telecommunication infrastructure
- Privacy is maintained through the use of a tunneling protocol and other security procedures

© 2002 Carnegie Mellon University

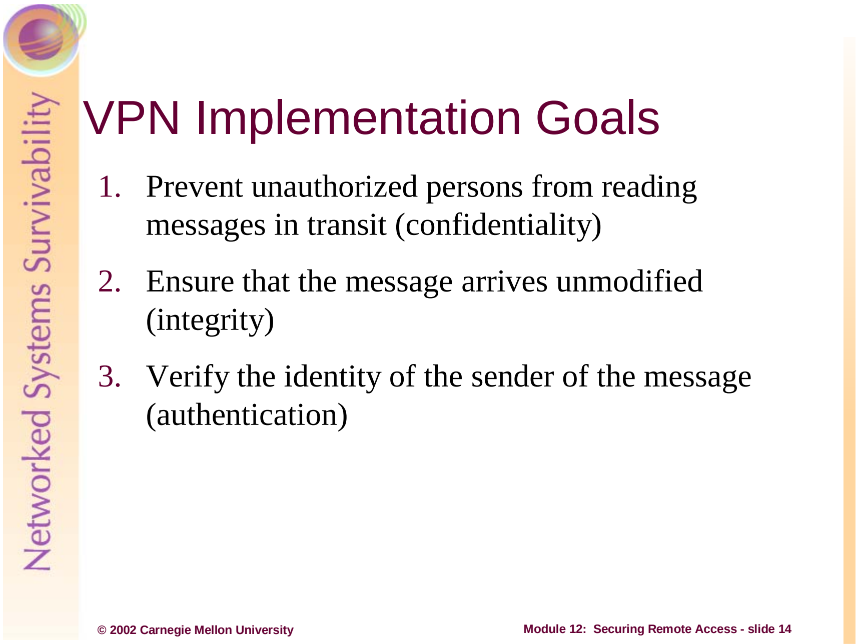
Module 12: Securing Remote Access - slide 13

Defining ‘Private Network’

A private network can be defined as a ‘closed network.’ This implies a level of privacy and ‘boundedness.’ Therefore, not everyone will have access. In the traditional sense, private networks have been deployed in the corporate setting – one must be a member of the corporation (or a partner or other trusted outsider) to be able to gain access to resources on the network. This has historically been done via physical means – through LAN connectivity combined with dedicated, leased lines for WAN connections. If there were remote users, they often dialed directly in to the network (via modem banks), and were authenticated. Therefore, as remote users, everyone used all corporate infrastructure assets for network access (except the POTS and leased lines themselves, but you get the point).

Defining ‘Virtual Private Network’

The term ‘virtual’ is an antonym of ‘actual.’ Therefore, when we discuss virtual private networks, they are by definition not ‘actual private networks.’ So, they must contain some non-private, public, pieces. Traditionally today, this is accomplished by allowing remote users to connect to the private network via an Internet connection. Without proper planning as well as proper implementations, this would leave the transmissions between the remote users and the private network vulnerable to interception and modification. A properly configured and administered VPN, on the other hand, can prove to be a safe, cost effective alternative to standard private networking infrastructure and connections.


Networked Systems Survivability

VPN Implementation Goals

1. Prevent unauthorized persons from reading messages in transit (confidentiality)
2. Ensure that the message arrives unmodified (integrity)
3. Verify the identity of the sender of the message (authentication)

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 14

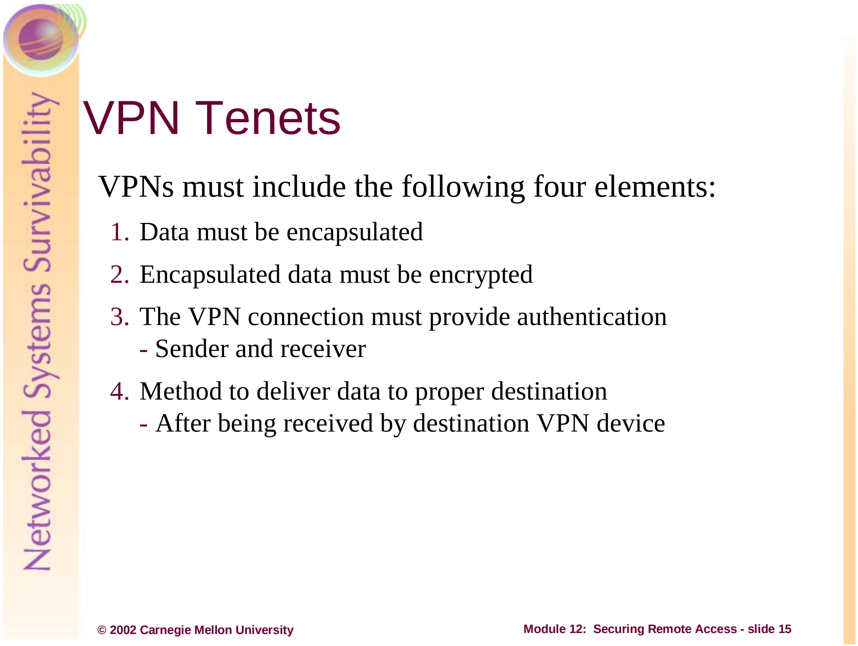
VPNs, to be effective, must accomplish all of the following:

Confidentiality – The VPN must encrypt the data being transmitted across the public network such that unauthorized persons may not read the messages. This is important because, for a private network connection to be established across the public Internet, the same expectations of confidentiality that are afforded to users on the actual private network must be afforded to the users on the virtual private network.

Integrity – The VPN must make provisions for the verification that the data has arrived unmodified. If data can be modified in transit, than the private nature of the network has been compromised, and thus the private network is not private anymore.

Authentication – The VPN must be able to verify the identity of the sender of the data. If unauthorized users were able to access the private network and send data posing as someone else, than the private network has been breached and thus is no longer private.

As you can see, it is important to structure the VPN such that all security considerations and options that are available for the actual private network are extended to the virtual private network.


Networked Systems Survivability

VPN Tenets

VPNs must include the following four elements:

1. Data must be encapsulated
2. Encapsulated data must be encrypted
3. The VPN connection must provide authentication
   - Sender and receiver
4. Method to deliver data to proper destination
   - After being received by destination VPN device

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 15

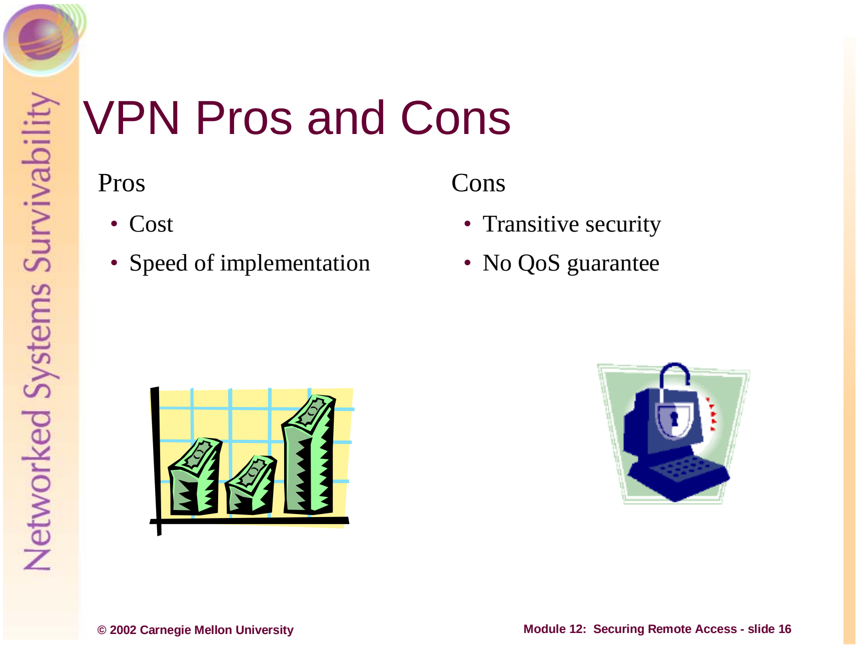
To accomplish the goals, the following four 'tenants' must be applied to a VPN connection:

1. *Data must be encapsulated*. Essentially, a tunnel must be established between VPN end points. When packets are sent across this tunnel, they are first embedded in a tunneling protocol (which will be discussed later). Afterwards, they have the appropriate TCP/IP headers placed on them for transmission across the Internet.

2. *Encapsulated data must be encrypted.* For the confidentiality goal to be met, the data must be encrypted. This is best accomplished after the initial tunneling protocol encapsulation has taken place, and before the TCP/IP headers have been placed onto the packet.

3. *The VPN connection must provide authentication*. There must be a way for the VPN infrastructure to be able to establish that a remote user is a) who they say they are; and b) should have access to the private network via the VPN.

4. *There must be a method to deliver data to its proper destination on each end of the VPN*. After the tunneling protocol's header is stripped off the packet by the VPN device, it must know how to send the packet on it's way on the local network.



Networked Systems Survivability

VPN Pros and Cons

Pros

- Cost
- Speed of implementation

Cons

- Transitive security
- No QoS guarantee

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 16

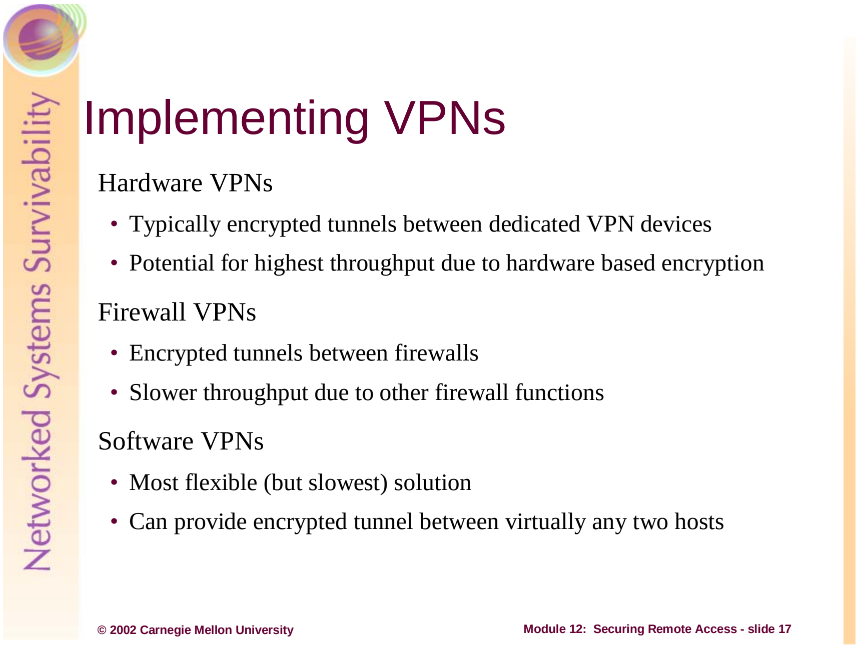
So why introduce public infrastructure into your private network? First and foremost is cost. The cost to implement a T-1 from Chicago to California is approximately $6000 per month. A similar Frame Relay circuit would cost approximately $3000 per month. An Internet connection of similar speed (symmetric digital subscriber line (SDSL) at 1.5 Mbps) would be approximately $400 on each side of the connection, or approximately $800 per month. This would equate to a savings of over $62,000 per year over the leased line and $26,000 per year on the frame relay circuit.

In addition to lower cost, the time to turn a T-1 order of that length around would likely be 6-8 weeks, while the installation of the SDSL connections would be 2-3 weeks. Therefore, you could have similar speeds quicker and cheaper with the VPN than you could with the dedicated leased lines.

VPNs appear to be great from a cost and speed of implementation standpoint. What could possibly make a VPN unattractive? Two main things . . .

There is certainly a security consideration when implementing a VPN. Consider that you will be opening up the ability to connect to the private network through your Internet connection. This will introduce a potential vulnerability – one which must be managed and administered correctly so that it is not exploited. And, if you have remote users who have other connections – whether to the Internet or other private networks, the VPN may allow unauthorized access or bring into the picture other possible vulnerabilities (i.e. additional means for viruses to be introduced).

Another drawback is that there is not any guarantee for quality of service for VPNs. With dedicated services, there is bandwidth allocated for the connection. Even if there is a high amount of traffic on the carrier's network, there are guarantees in place which ensure the delivery of packets within certain acceptable delays. This is known as Quality of Service. On the public Internet, there are no such guarantees. This means that there is not bandwidth set aside for the VPN connection on the public Internet, which means that the VPN traffic will be in contention with the rest of the traffic. It is a problem if there is congestion on the Internet – as any bottlenecks may cause delays in VPN packets, and may render the VPN unusable. For time sensitive traffic, this may be an issue which causes a VPN to be an unacceptable answer for connectivity.



Networked Systems Survivability

Implementing VPNs

Hardware VPNs

- Typically encrypted tunnels between dedicated VPN devices
- Potential for highest throughput due to hardware based encryption

Firewall VPNs

- Encrypted tunnels between firewalls
- Slower throughput due to other firewall functions

Software VPNs

- Most flexible (but slowest) solution
- Can provide encrypted tunnel between virtually any two hosts

© 2002 Carnegie Mellon University

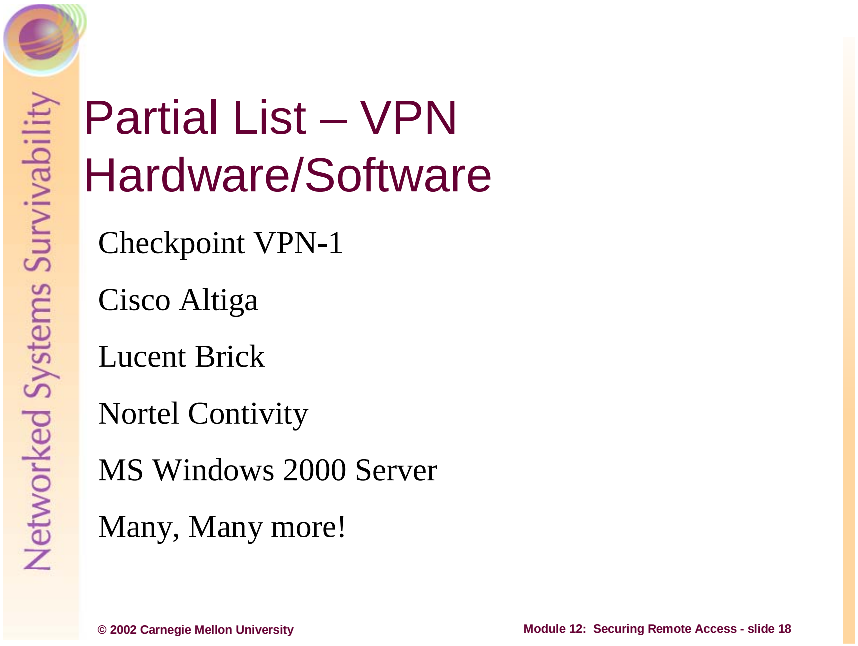
Module 12: Securing Remote Access - slide 17

There are three main categories of VPNs:

Hardware VPNs are typically dedicated devices that offer hardware/software to terminate encrypted tunnels between them. There is very little processing occuring, other than encrypting the packet and attaching the TCP/IP headers, and thus they are very fast. However, they require specially designed hardware to implement, and thus are usually only used for creating a VPN from one corporate network to another corporate (or business partner) network. They are seldom used for home users, as the cost of the infrastructure to support such a VPN is high for such use.

Firewall VPNs create the encrypted tunnel between firewall devices. Because of the processing load that is inherent on the firewall, these implementations are slower than hardware VPNs. Because of the nature of firewall devices, however, this implementation is generally considered to be more secure and easier to administer than the hardware VPNs.

Software VPNs offer a great deal of flexibility. Any host that can install and run the client VPN software can terminate a VPN. This opens up the feasibility for home users and for very small offices that may not have routers or firewall devices present to terminate their VPN connection. Properly configured and administered software VPNs can be very secure. However, their flexibility comes at the expense of speed, as these implementations are typically the slowest.



Networked Systems Survivability

Partial List – VPN Hardware/Software

Checkpoint VPN-1

Cisco Altiga

Lucent Brick

Nortel Contivity

MS Windows 2000 Server

Many, Many more!

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 18

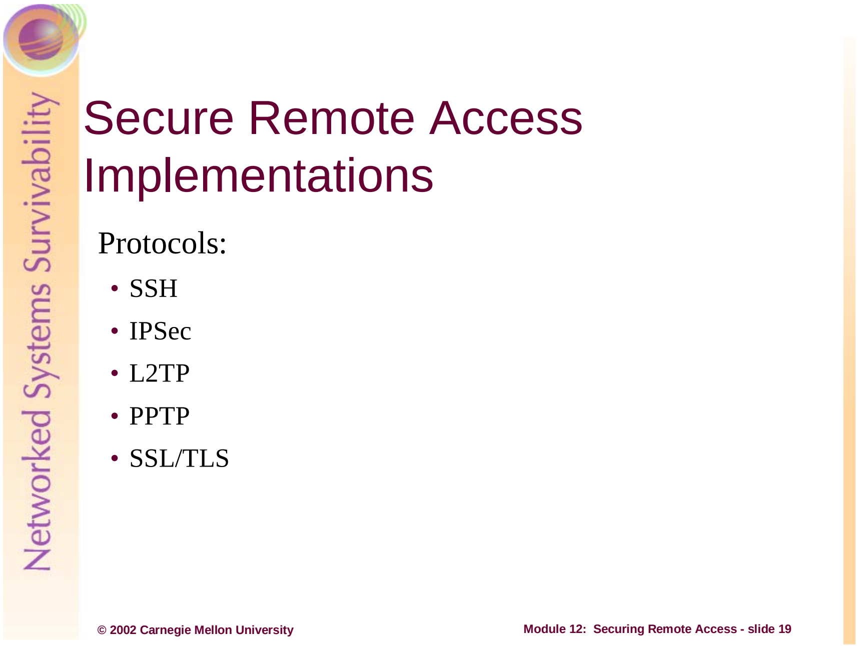
There is a host of VPN hardware and software available, both commercially and through the GPL. Some interesting information regarding GPL VPNs, and other VPN info, can be found at:

FreeS/WAN - http://www.freeswan.org/. Linux FreeS/WAN is an implementation of IPSEC & IKE for Linux.

VPN Consortium - http://www.vpnc.org/. The primary purposes of the VPNC are:

- Promote the products of its members to the press and to potential customers
- Increase interoperability between members by showing where the products interoperate
- Serve as the forum for the VPN manufacturers and service provider throughout the world
- Help the press and potential customers understand VPN technologies and standards
- Provide publicity and support for interoperability testing events

SecurePoint – http://www.securepoint.com/. Securepoint firewall & VPN server offers a full-featured suite of firewall tools designed for enterprisewide deployment. Not only can it protect an internal network from outside attacks, it also helps segregate parts of your internal network and define custom protection rules for each. Securepoint lets you create and manage VPN tunnels for remote users and define traffic filters, reports, and alerts for your entire network.



Networked Systems Survivability

Secure Remote Access Implementations

Protocols:

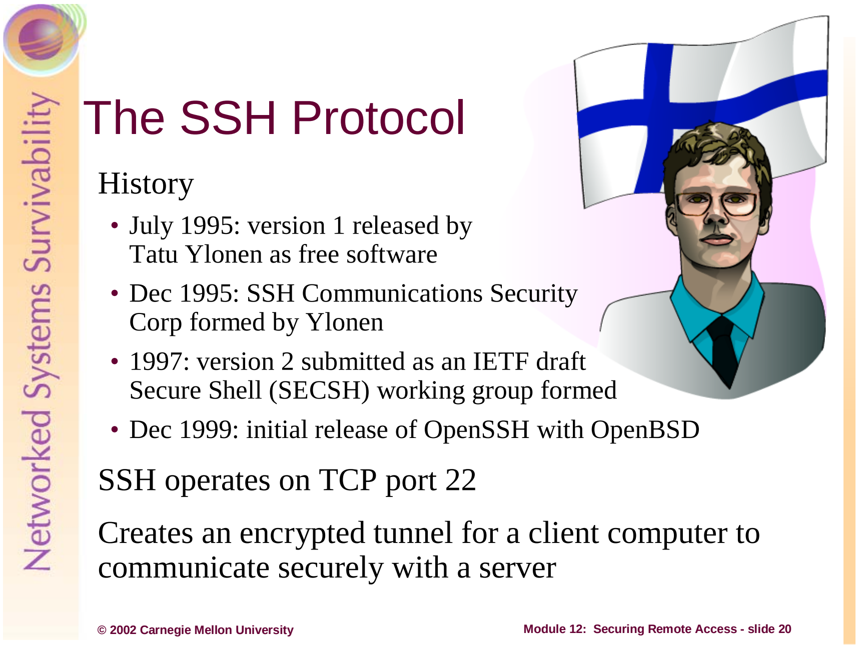
- SSH
- IPSec
- L2TP
- PPTP
- SSL/TLS

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 19

Some of the protocols and implementations will be best suited for a certain type of connection and certain type of situation. For example, it may be appropriate for a remote worker to use a tunneling protocol for securing their connection into the enterprise. This could be classified as an ‘expected’ and established connection – where both of the end-points are predictable and known. Protocols such as ssh, IPSec, PPTP, and L2TP will suit this purpose very well.

Other types of connections where we want to allow secure connections to be made from any location, however, will require different implementations and protocols. For example, an e-commerce web site customer checking their balance or looking up their personal information may use SSL or TLS to secure their transaction.



Networked Systems Survivability

The SSH Protocol

History

- July 1995: version 1 released by Tatu Ylonen as free software
- Dec 1995: SSH Communications Security Corp formed by Ylonen
- 1997: version 2 submitted as an IETF draft Secure Shell (SECSH) working group formed
- Dec 1999: initial release of OpenSSH with OpenBSD

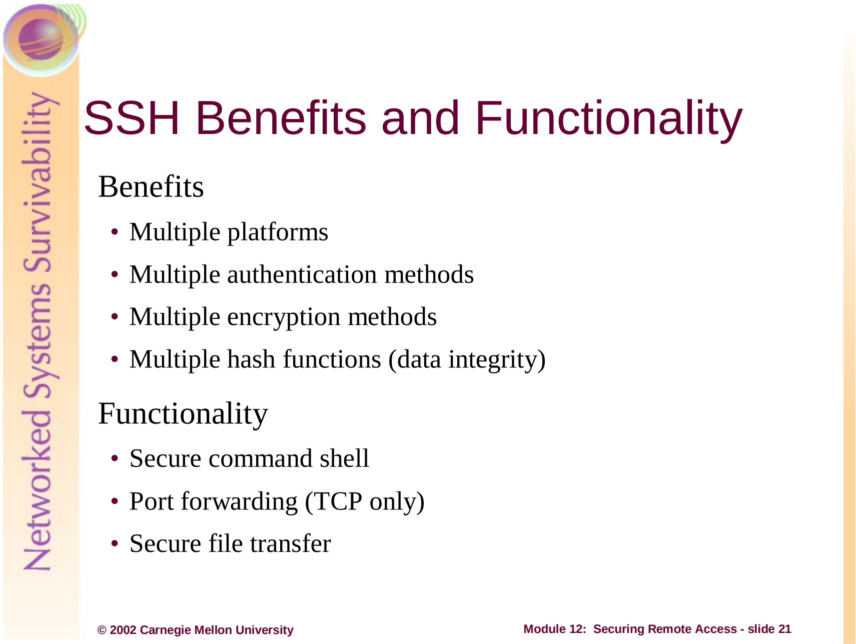
SSH operates on TCP port 22

Creates an encrypted tunnel for a client computer to communicate securely with a server

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 20

The need for secure tunnels that can be used for many applications was met in 1995 when Tatu Ylonen developed the ssh protocol. This allowed a TCP based tunnel to be created between client and server. The client and server have many options regarding the use of the tunnel – through encryption, port forwarding, and shell capabilities.



Networked Systems Survivability

SSH Benefits and Functionality

Benefits

- Multiple platforms
- Multiple authentication methods
- Multiple encryption methods
- Multiple hash functions (data integrity)

Functionality

- Secure command shell
- Port forwarding (TCP only)
- Secure file transfer

Module 12: Securing Remote Access - slide 21

Multiple Platforms

ssh is available for many different platforms, including: Windows (virtually all versions), most versions of Linux (i.e. Redhat, SuSe, Mandrake), Mac (OS X), most versions of UNIX (Solaris, HP-UX, AIX, FreeBSD, etc.).

Multiple Authentication Methods

1. Password - Password authentication uses the /etc/passwd or /etc/shadow file on your UNIX system, depending on how your passwords are set up.
2. Public key - the recommended replacement for password authentication.
3. Hostbased - For Unix only, hostbased provides a non-interactive authentication method. It can be used for automating backups and file transfers using scripts.
4. Kerberos5 - a popular network authentication protocol developed by MIT. It can be used to provide strong authentication for client/server systems for Unix platforms.
5. PAM - the de facto standard for single sign-on authentication services for Linux and Solaris platforms. It allows system administrators to maintain a flexible security policy management.
6. SecurID - offers a comprehensive range of authenticator options including hardware tokens, smart cards and software tokens.

Multiple Encryption Methods

Various ciphers are integrated into ssh – including: 3DES, CAST, Arcfour, Blowfish, Twofish, and DSA

Multiple Hash Functions

MD5 and SHA

Secure command shell

ssh offers secure command shell for access to systems which offer a command shell.

Port Forwarding

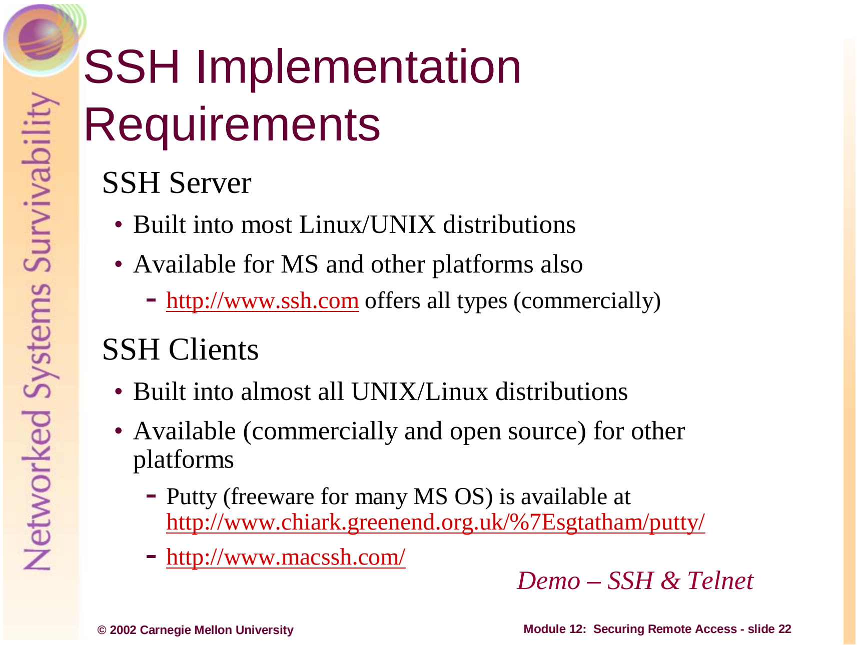
X11, TCP, and authentication agent

Secure file transfer

Ability to securely transfer files. Some operating systems and implementations can be configured to support *only* secure file transfers with ssh – and no other ssh capabilities.

Data compression

ssh can improve some performance on slow connections by compressing the data.


Networked Systems Survivability

SSH Implementation Requirements

SSH Server

- Built into most Linux/UNIX distributions
- Available for MS and other platforms also
  - http://www.ssh.com offers all types (commercially)

SSH Clients

- Built into almost all UNIX/Linux distributions
- Available (commercially and open source) for other platforms
  - Putty (freeware for many MS OS) is available at http://www.chiark.greenend.org.uk/%7Esgtatham/putty/
  - http://www.macssh.com/

Demo – SSH & Telnet

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 22

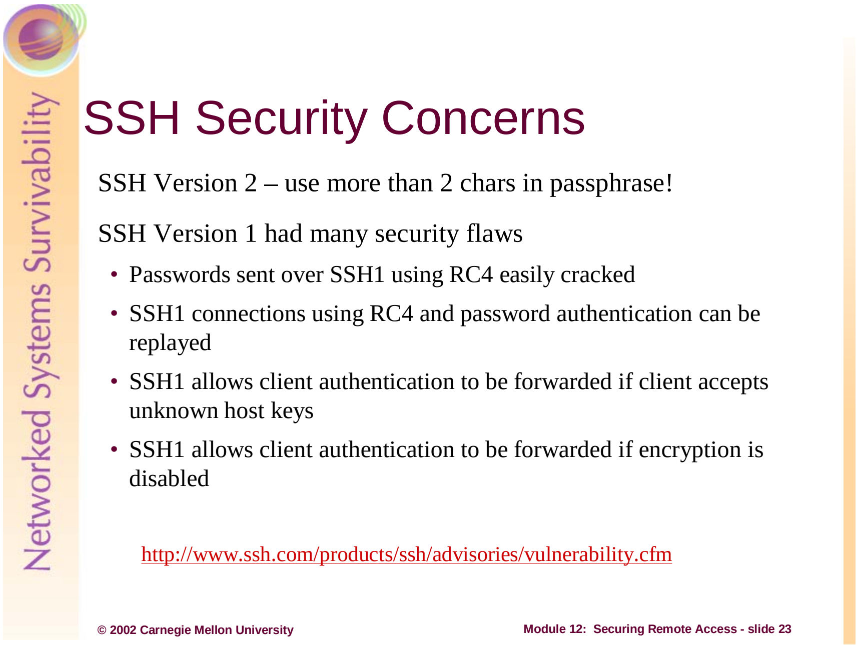
The authentication process for SSH takes the following steps:

1. User invokes SSH Client
2. SSH client connects to SSHD Server running on remote computer.
3. SSH client and server exchange keys and establish an encrypted tunnel for communication
4. SSH client forwards login/password/commands over encrypted tunnel
5. SSHD server authenticates user information and allows remote command to be executed
6. User is then free to use the system exactly as if he had issued a “telnet” command, but all data passed between the client and the server is now encrypted.

To set up and deploy SSH, all that is needed is an SSH server and an SSH client. There are many available versions of the SSH server, some which come included in UNIX and Linux distributions, and some which are commercially available.

OpenSSH – http://www.openssh.org/. OpenSSH is a free version of the SSH server which is available for most UNIX and Linux operating systems.

SSH Communications Security – http://www.ssh.com/. Many versions, including Windows SSH server and clients, are available from SSH Communications Security.


Networked Systems Survivability

SSH Security Concerns

SSH Version 2 – use more than 2 chars in passphrase!

SSH Version 1 had many security flaws

- Passwords sent over SSH1 using RC4 easily cracked
- SSH1 connections using RC4 and password authentication can be replayed
- SSH1 allows client authentication to be forwarded if client accepts unknown host keys
- SSH1 allows client authentication to be forwarded if encryption is disabled

http://www.ssh.com/products/ssh/advisories/vulnerability.cfm

© 2002 Carnegie Mellon University

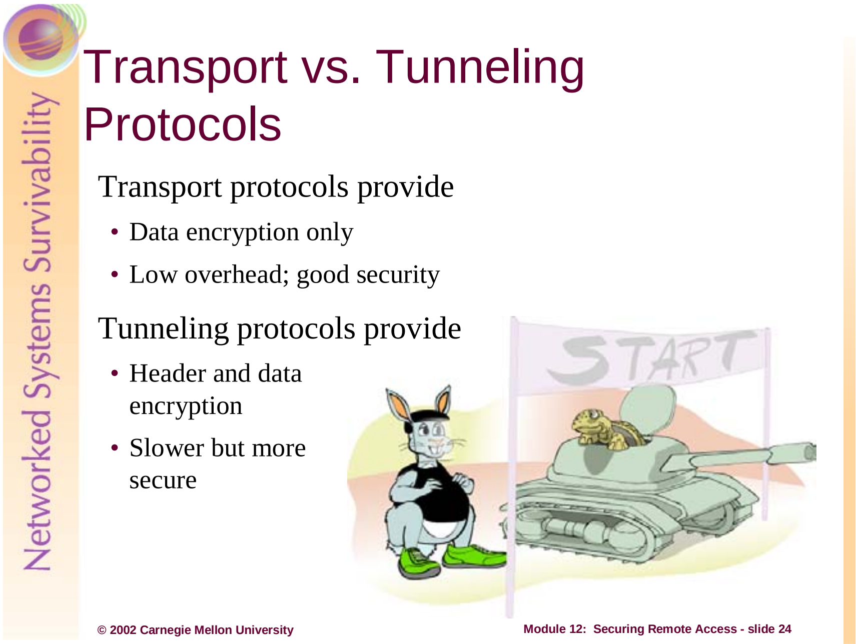
Module 12: Securing Remote Access - slide 23

Much has been written, debated, and argued over SSH’s vulnerability to man-in-the-middle (MITM) attacks. Like an implementation of any encryption system, it is possible, with a certain set of circumstances, for a MITM attack to be carried out successfully against SSH.

Nearly all implementations of SSH support what is known as "Password Authentication", in which the client sends its password directly to the server, hoping that the encrypted connection will protect it in transit. The security of this method is highly dependent on the particular method of server authentication being used. Keep in mind that an attacker who successfully bypasses server authentication in this case gets the *plaintext* passwords of any users who subsequently attempts to log in, and can do so undetectably.

One commonly used method of server authentication is "ad-hoc" distribution of server public keys. The server sends the client its (non-certified) public host key, and the client uses this to encrypt a session key and send it to the server. The actual protocol is slightly more complicated, but what matters is that the client has no way of knowing if the host public key it received was in fact the right one, which makes this protocol susceptible to MITM attacks in practice. Although the client tries to keep track of previously-received host keys, there is no way for it to know if a change in host keys is legitimate or the beginning of an attack, so users will in most cases either ignore the warning that SSH spits out or inundate the help desk (if there is one) every time a host key expires or the network configuration changes for any reason.

In addition to the MITM attack, several vulnerabilities have been identified (and patched) in older versions of SSH.



Networked Systems Survivability

Transport vs. Tunneling Protocols

Transport protocols provide

- Data encryption only
- Low overhead; good security

Tunneling protocols provide

- Header and data encryption
- Slower but more secure

© 2002 Carnegie Mellon University

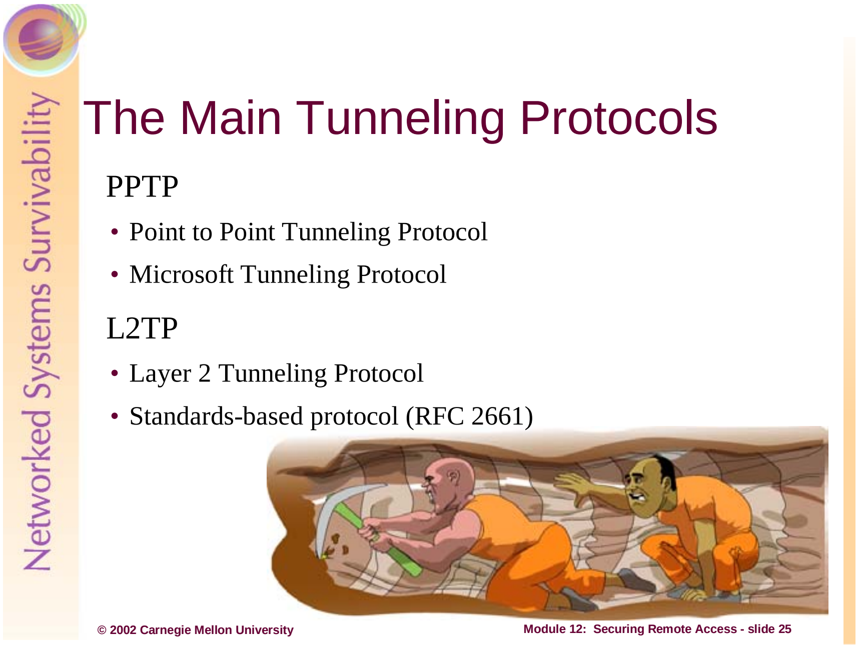
Module 12: Securing Remote Access - slide 24

Transport protocols provide encryption of the payload of the original packet only. They are relatively fast as they have a lower workload than do the tunneling protocols. All of the original headers from the native network packet are unchanged – only the data within the packet is changed.

The goal of the tunneling protocols is to encapsulate the packet in its own header. Why?

For a couple of reasons…

> First, the data inside of the original packet (which was encapsulated) can now be effectively encrypted and kept confidential. This is particularly useful when a remote worker is accessing information from the enterprise network. This can be the cornerstone of a true VPN.
>
> Second, it may be important to obfuscate the origin of the packet (for non-nefarious reasons). This hiding of the true source of a packet is a result of the tunneling protocols ability to support network address translation (NAT).



Networked Systems Survivability

The Main Tunneling Protocols

PPTP

- Point to Point Tunneling Protocol
- Microsoft Tunneling Protocol

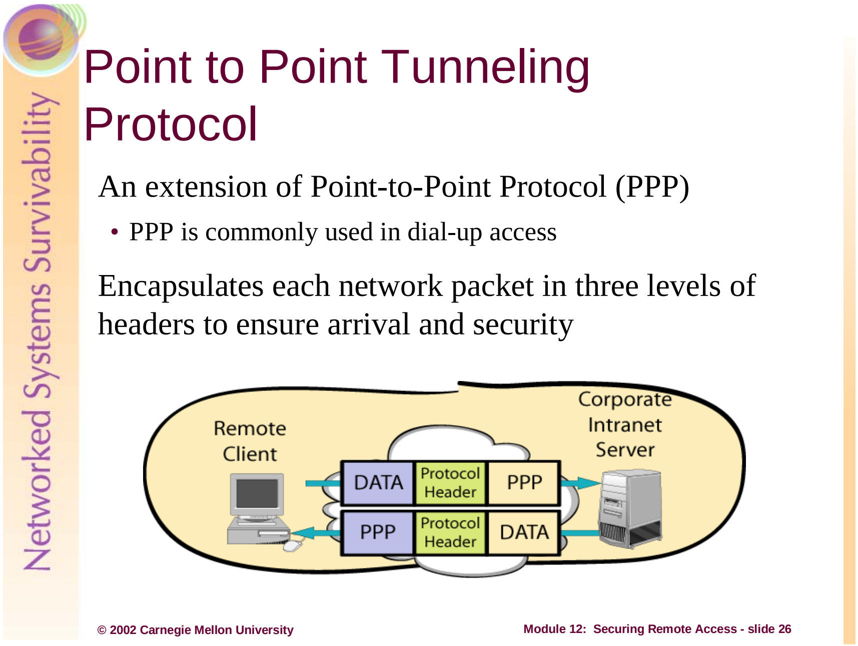
L2TP

- Layer 2 Tunneling Protocol
- Standards-based protocol (RFC 2661)

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 25

The main two tunneling protocols are PPTP and L2TP. Microsoft, in conjunction with other companies such as 3Com, established PPTP as their proprietary protocol for tunneling. Cisco established a protocol called Layer 2 Forwarding (L2F) as their attempt at a proprietary tunneling protocol. Then the IETF (Internet Engineering Task Force) basically took the best of both protocols and created L2TP. L2F is less common than the others – and all of the protocols basically perform the same functions.



Networked Systems Survivability

Point to Point Tunneling Protocol

An extension of Point-to-Point Protocol (PPP)

- PPP is commonly used in dial-up access

Encapsulates each network packet in three levels of headers to ensure arrival and security

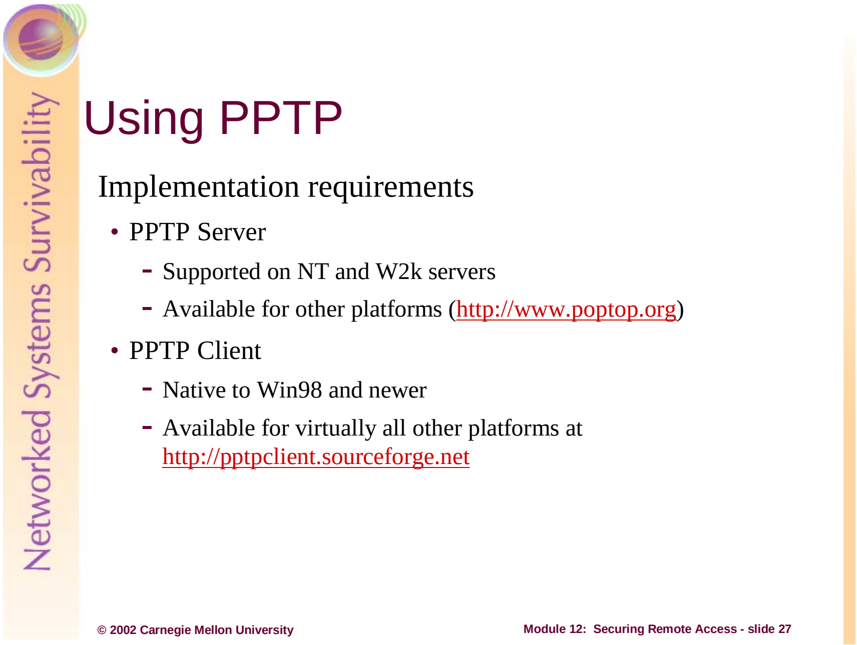
Remote Client
Corporate Intranet Server
DATA
Protocol Header
PPP
PPP
Protocol Header
DATA

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 26

Point-to-Point Tunneling Protocol (PPTP) is a network protocol that enables the secure transfer of data from a remote client to a private enterprise server by creating a virtual private network (VPN) across TCP/IP-based data networks. PPTP supports on-demand, multi-protocol, virtual private networking over public networks such as the Internet.

The networking technology of PPTP is an extension of the remote access Point-to-Point protocol defined in RFC 1171 (a.k.a. The Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams over Point-to-Point Links). PPTP is a network protocol that encapsulates PPP packets into IP datagrams for transmission over the Internet or other public TCP/IP-based networks. This then allows the PPP packets (encapsulated in PPTP headers) to traverse the Internet – something that they could not do in the past. PPTP can also be used in private LAN-to-LAN networking across a public network.



Networked Systems Survivability

Using PPTP

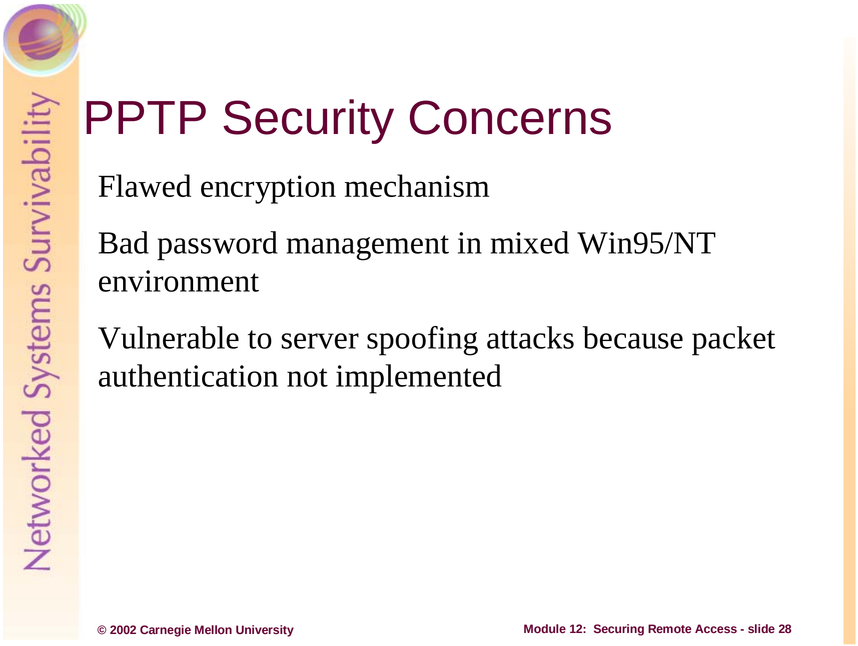
Implementation requirements

- PPTP Server
  - Supported on NT and W2k servers
  - Available for other platforms (http://www.poptop.org)
- PPTP Client
  - Native to Win98 and newer
  - Available for virtually all other platforms at http://pptpclient.sourceforge.net

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 27

Much like other remote access solutions, PPTP requires a server and client to communicate for a PPTP connection to be established. Windows NT and newer server operating systems come with PPTP support built in, as do Windows 98 and newer Microsoft clients.



Networked Systems Survivability

PPTP Security Concerns

- Flawed encryption mechanism
- Bad password management in mixed Win95/NT environment
- Vulnerable to server spoofing attacks because packet authentication not implemented

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 28

While there are many benefits of implementing PPTP, there are a few security concerns. Some of these can be overcome with proper administration, others with patches. But these are well known PPTP issues:

Flawed encryption mechanism

- non-random keys, session keys weak hash of user password, key lengths too short (non-configurable)

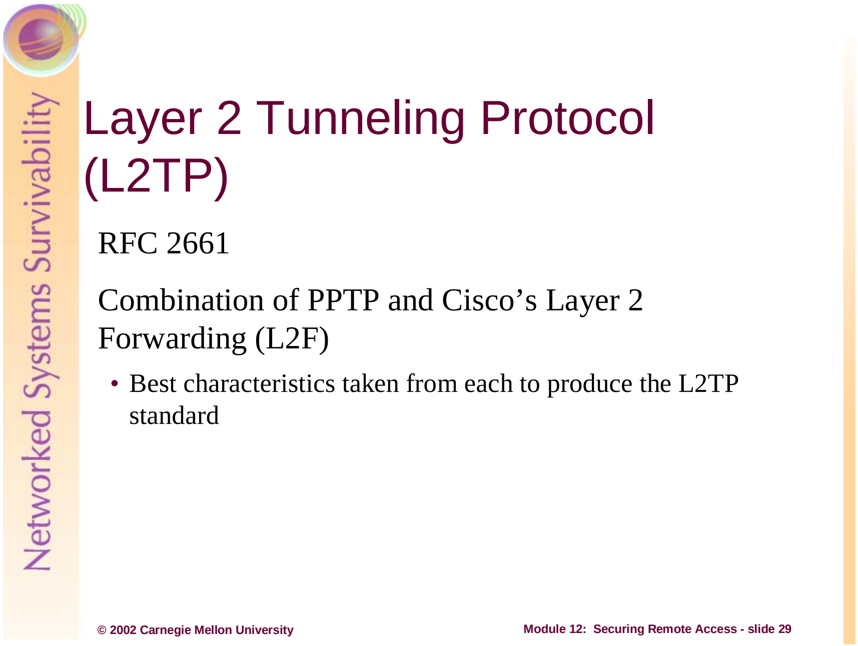
Bad password management in mixed Win95/NT environment

- static passwords easily compromised

Vulnerable to server spoofing attacks because packet authentication not implemented

- easy denial-of-service attacks even inside firewalls

(Microsoft claims cryptographic weaknesses have not yet been exploited)



Networked Systems Survivability

Layer 2 Tunneling Protocol (L2TP)

RFC 2661

Combination of PPTP and Cisco’s Layer 2 Forwarding (L2F)

- Best characteristics taken from each to produce the L2TP standard

© 2002 Carnegie Mellon University

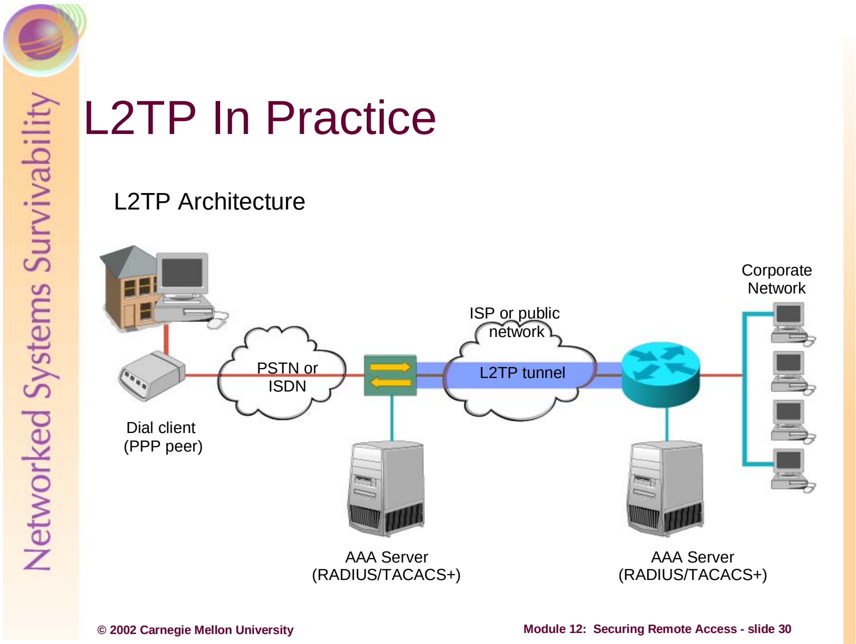
Module 12: Securing Remote Access - slide 29

L2TP is an extension of the PPP protocol. L2TP allows VPNs to be established through Internet connections offered by Internet Service Providers (ISPs), as opposed to requiring direct access (via leased lines or direct dial-up connections).

PPP [RFC1661] defines an encapsulation mechanism for transporting multiprotocol packets across layer 2 (L2) point-to-point links. Typically, a user obtains a L2 connection to a Network Access Server (NAS) using one of a number of techniques (e.g., dialup POTS, ISDN, ADSL, etc.) and then runs PPP over that connection. In such a configuration, the L2 termination point and PPP session endpoint reside on the same physical device (i.e., the NAS).

L2TP extends the PPP model by allowing the L2 and PPP endpoints to reside on different devices interconnected by a packet-switched network. With L2TP, a user has an L2 connection to an access concentrator (e.g., modem bank, ADSL DSLAM, etc.), and the concentrator then tunnels individual PPP frames to the NAS. This allows the actual processing of PPP packets to be divorced from the termination of the L2 circuit.

One obvious benefit of such a separation is that instead of requiring the L2 connection terminate at the NAS (which may require a long-distance toll charge), the connection may terminate at a (local) circuit concentrator, which then extends the logical PPP session over a shared infrastructure such as frame relay circuit or the Internet. From the user's perspective, there is no functional difference between having the L2 circuit terminate in a NAS directly and using L2TP [From RFC 2661].



Networked Systems Survivability

L2TP In Practice

L2TP Architecture

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 30

The parts of L2TP

L2TP Access Concentrator (LAC)

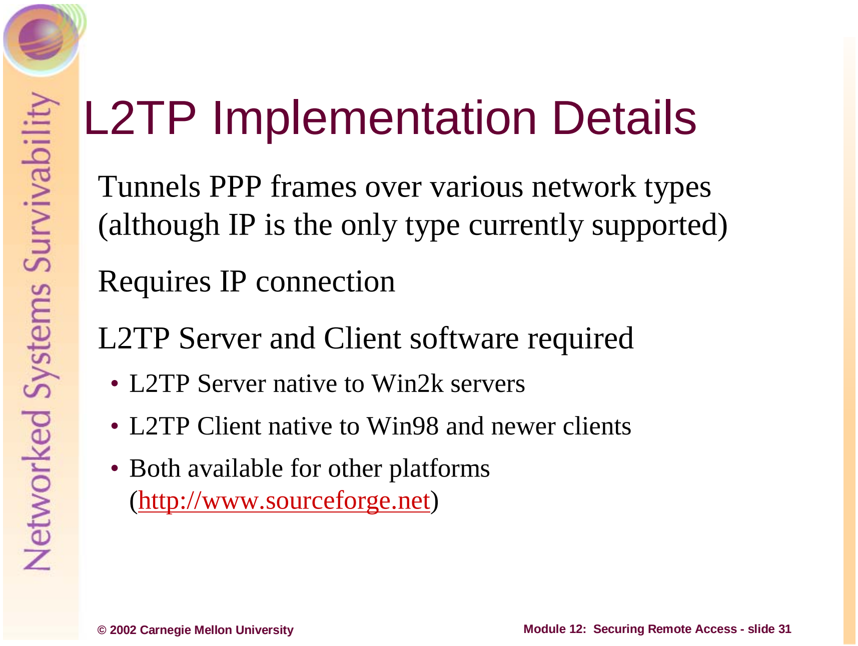
A node that acts as one side of an L2TP tunnel endpoint and is a peer to the L2TP Network Server (LNS). The LAC sits between the LNS and a remote system and forwards packets to and from each. Packets sent from the LAC to the LNS require tunneling with the L2TP protocol as defined in this document. The connection from the LAC to the remote system is either local or a PPP link.

L2TP Network Server (LNS)

A node that acts as one side of an L2TP tunnel endpoint and is a peer to the L2TP Access Concentrator (LAC). The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the LAC.

Session

L2TP is connection-oriented. The LNS and LAC maintain state for each connection that is initiated or answered by an LAC. An L2TP Session is created between the LAC and LNS when an end-to-end PPP connection is established between a Remote System and the LNS. Datagrams related to the PPP connection are sent over the Tunnel between the LAC and LNS. There is a one to one relationship between established L2TP Sessions and their associated connections.



Networked Systems Survivability

L2TP Implementation Details

Tunnels PPP frames over various network types (although IP is the only type currently supported)

Requires IP connection

L2TP Server and Client software required

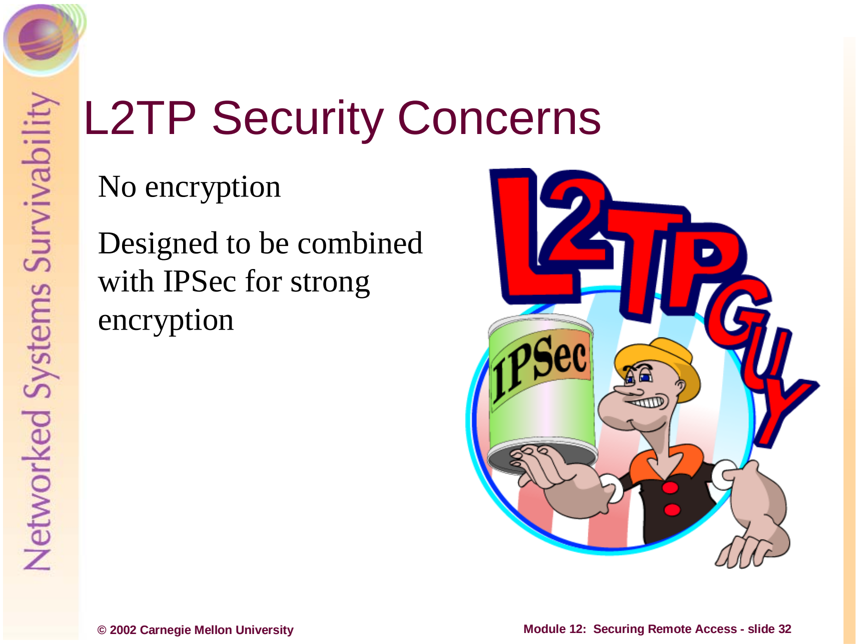
- L2TP Server native to Win2k servers
- L2TP Client native to Win98 and newer clients
- Both available for other platforms (http://www.sourceforge.net)

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 31

Historically, when a dial-up customer would connect to an ISP, there would be a PPP connection established between the customer and the network access server (NAS). On the same systems (the customer's PC and the NAS), there would be a layer 2 connection as well. This is effective for a true Internet dial-up connection, or when a remote user dials directly into an enterprise's modem pool. However, it is not possible to establish VPN if the termination of the PPP connection is at this point. To maintain security, the PPP sessions must initiate at the remote PC and terminate within the enterprise network.

And that is exactly what L2TP allows. L2TP allows for the separation of the PPP termination point and the layer 2 tunnel termination point. Therefore, we can now terminate the L2 session in the Internet Service Provider's network and still extend the PPP session to the enterprise to which we are establishing a VPN.



Networked Systems Survivability

L2TP Security Concerns

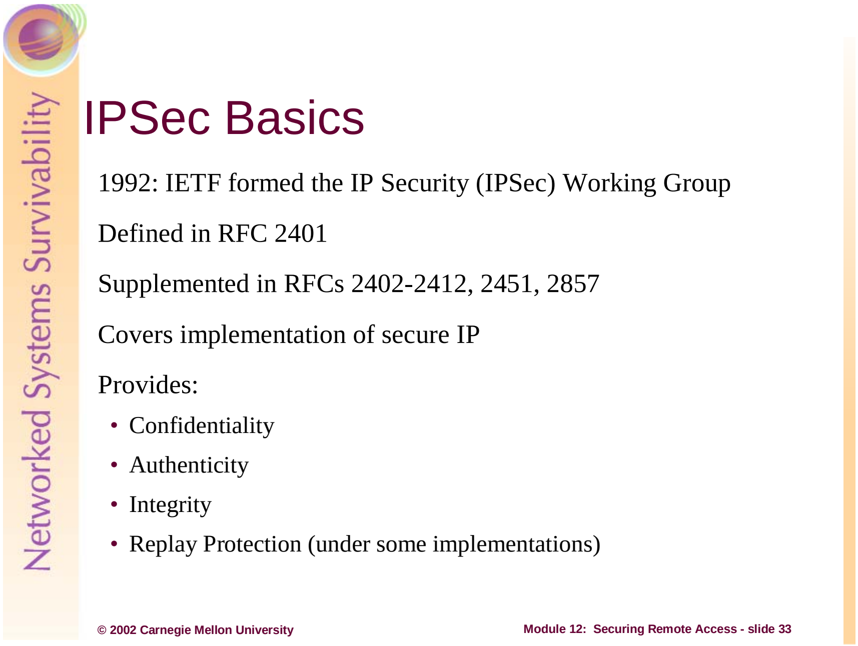
No encryption

Designed to be combined with IPSec for strong encryption

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 32

L2TP alone is not secure, as it does not offer strong encryption for its packets. Therefore, it is not practical to use it alone as the means to establish a VPN. It does offer flexibility in the establishment of the VPN, but requires additional encryption methods to secure it. Therefore, when implemented in tandem with IPSec (for strong encryption), and administered correctly, the pair is virtually impenetrable.



Networked Systems Survivability

IPSec Basics

1992: IETF formed the IP Security (IPSec) Working Group

Defined in RFC 2401

Supplemented in RFCs 2402-2412, 2451, 2857

Covers implementation of secure IP

Provides:

- Confidentiality
- Authenticity
- Integrity
- Replay Protection (under some implementations)

© 2002 Carnegie Mellon University

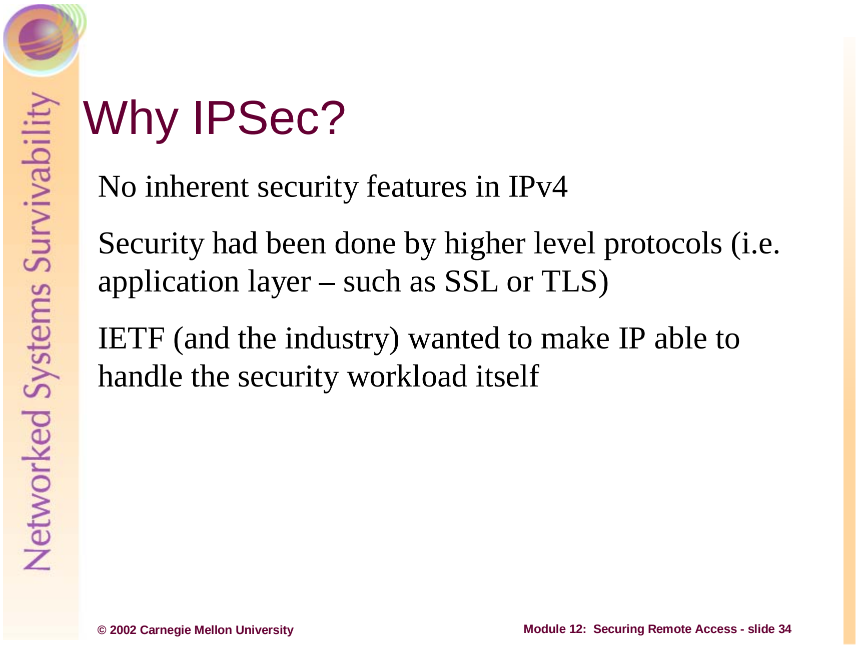
Module 12: Securing Remote Access - slide 33

IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6. The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays (a form of partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols.

These objectives are met through the use of two traffic security protocols, the Authentication Header (AH) and the Encapsulating Security Payload (ESP), and through the use of cryptographic key management procedures and protocols. The set of IPsec protocols employed in any context, and the ways in which they are employed, will be determined by the security and system requirements of users, applications, and/or sites/organizations.

When these mechanisms are correctly implemented and deployed, they ought not to adversely affect users, hosts, and other Internet components that do not employ these security mechanisms for protection of their traffic. These mechanisms also are designed to be algorithm-independent. This modularity permits selection of different sets of algorithms without affecting the other parts of the implementation. For example, different user communities may select different sets of algorithms (creating cliques) if required.

A standard set of default algorithms is specified to facilitate interoperability in the global Internet. The use of these algorithms, in conjunction with IPsec traffic protection and key management protocols, is intended to permit system and application developers to deploy high quality, Internet layer, cryptographic security technology [RFC 2401].



Networked Systems Survivability

Why IPSec?

No inherent security features in IPv4

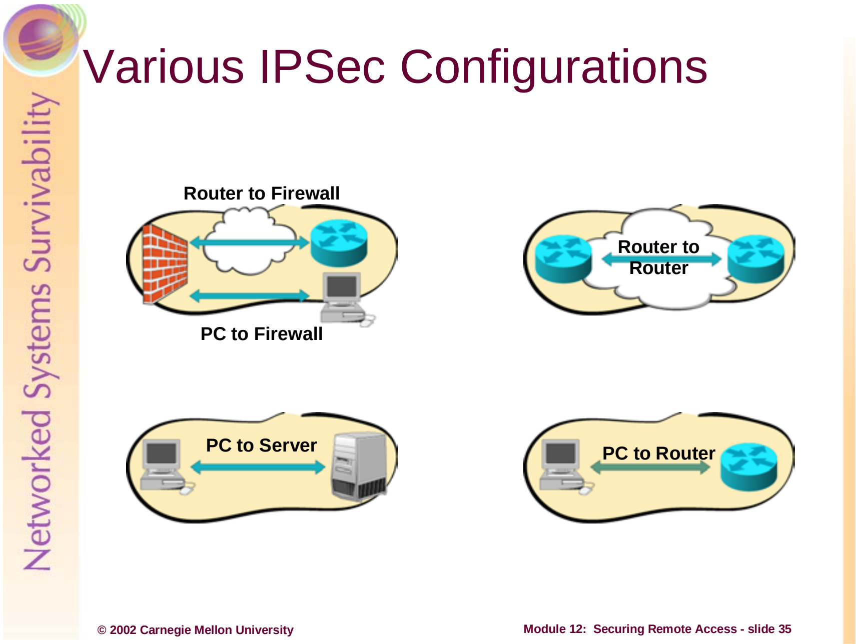
Security had been done by higher level protocols (i.e. application layer – such as SSL or TLS)

IETF (and the industry) wanted to make IP able to handle the security workload itself

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 34

Since no security features were built into IPv4, IP has been reliant on other (application layer) protocols to introduce security to its transmissions. The industry and the IETF, though, wanted to add security features (and thus confidentiality, non-repudiation, and authenticity) to IP itself.



Networked Systems Survivability

Various IPSec Configurations

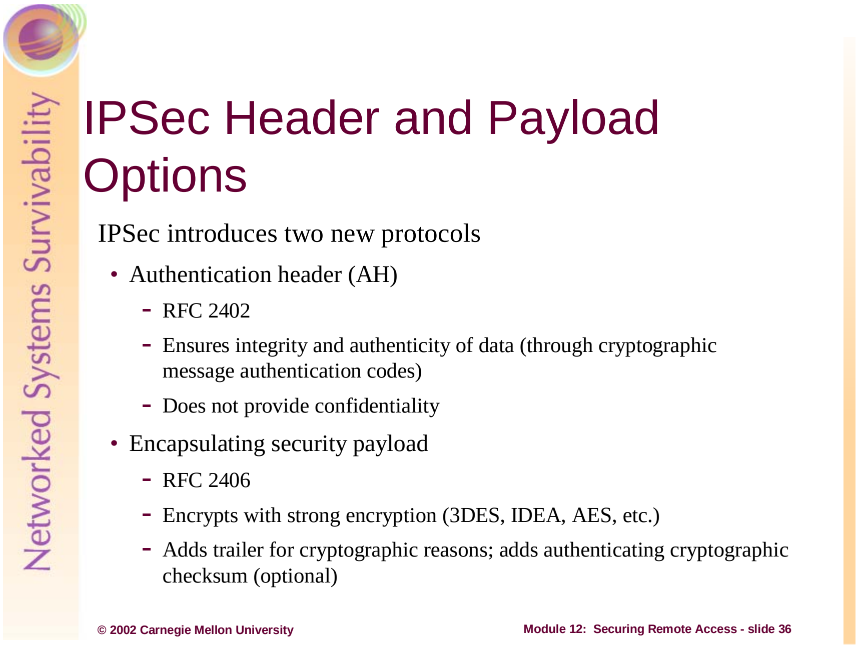
© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 35

IPSec can be implemented in various manners – and many networking systems can terminate IPSec sessions. These include:

- Firewalls
- PCs
- Routers
- Servers

To do so, they must be configured with IPSec software, which is available from many sources for virtually all operating systems.


Networked Systems Survivability

IPSec Header and Payload Options

IPSec introduces two new protocols

- Authentication header (AH)
  - RFC 2402
  - Ensures integrity and authenticity of data (through cryptographic message authentication codes)
  - Does not provide confidentiality
- Encapsulating security payload
  - RFC 2406
  - Encrypts with strong encryption (3DES, IDEA, AES, etc.)
  - Adds trailer for cryptographic reasons; adds authenticating cryptographic checksum (optional)

© 2002 Carnegie Mellon University

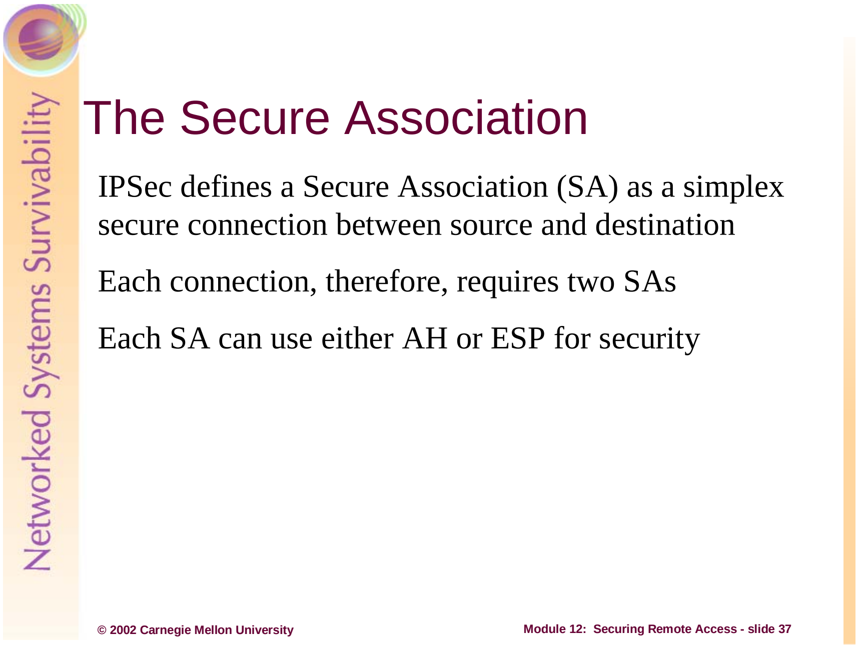
Module 12: Securing Remote Access - slide 36

IPSec uses two protocols to provide traffic security -- Authentication Header (AH) and Encapsulating Security Payload (ESP). Both protocols are described in more detail in their respective RFCs.

- The IP Authentication Header (AH) provides connectionless integrity, data origin authentication, and an optional anti-replay service.
- The Encapsulating Security Payload (ESP) protocol may provide confidentiality (encryption), and limited traffic flow confidentiality. It also may provide connectionless integrity, data origin authentication, and an anti-replay service. (One or the other set of these security services must be applied whenever ESP is invoked.)
- Both AH and ESP are vehicles for access control, based on the distribution of cryptographic keys and the management of traffic flows relative to these security protocols.

These protocols may be applied alone or in combination with each other to provide a desired set of security services in IPv4 and IPv6. Each protocol supports two modes of use: transport mode and tunnel mode. In transport mode the protocols provide protection primarily for upper layer protocols; in tunnel mode, the protocols are applied to tunneled IP packets [RFC 2401].

In addition to the security services of the IP Authentication Header protocol, the IP Encapsulating Security Payload protocol provides security services related to the confidentiality or privacy of the data being transmitted within the IP packet. Specifically, ESP provides encryption as a security service to protect the data content of the IP packet. For compliance with the IPsec standard, an implementation must include, at least, the DES encryption algorithm for use in providing ESP security services.


Networked Systems Survivability

The Secure Association

IPSec defines a Secure Association (SA) as a simplex secure connection between source and destination

Each connection, therefore, requires two SAs

Each SA can use either AH or ESP for security

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 37

There are two modes of IPSec. The differences in these come in how they do the encapsulation and encryption of the IPSec encapsulated packet. The term "security gateway" is used throughout the IPsec documents to refer to an intermediate system that implements IPsec protocols. For example, a router or a firewall implementing IPsec is a security gateway.

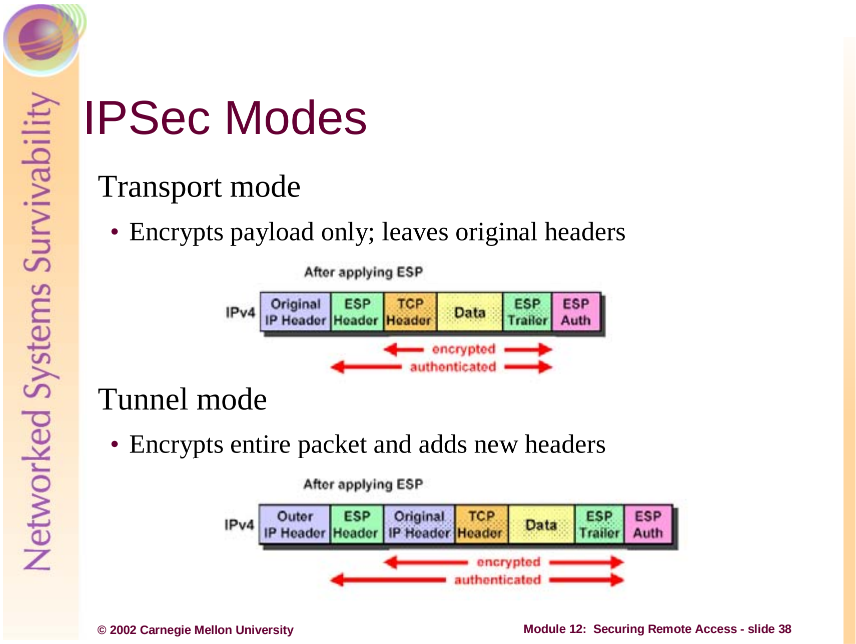
Transport mode [RFC 2401]

- A transport mode SA is a security association between two hosts. In IPv4, a transport mode security protocol header appears immediately after the IP header and any options, and before any higher layer protocols (e.g., TCP or UDP). In IPv6, the security protocol header appears after the base IP header and extensions, but may appear before or after destination options, and before higher layer protocols. In the case of ESP, a transport mode SA provides security services only for these higher layer protocols, not for the IP header or any extension headers preceding the ESP header. In the case of AH, the protection is also extended to selected portions of the IP header, selected portions of extension headers, and selected options (contained in the IPv4 header, IPv6 Hop-by-Hop extension header, or IPv6 Destination extension headers).

Tunnel mode [RFC 2401]

- A tunnel mode SA is essentially an SA applied to an IP tunnel. Whenever either end of a security association is a security gateway, the SA MUST be tunnel mode. Thus an SA between two security gateways is always a tunnel mode SA, as is an SA between a host and a security gateway. Note that for the case where traffic is destined for a security gateway, e.g., SNMP commands, the security gateway is acting as a host and transport mode is allowed. But in that case, the security gateway is not acting as a gateway, i.e., not transiting traffic. Two hosts MAY establish a tunnel mode SA between themselves. The requirement for any (transit traffic) SA involving a security gateway to be a tunnel SA arises due to the need to avoid potential problems with regard to fragmentation and reassembly of IPsec packets, and in circumstances where multiple paths (e.g., via different security gateways) exist to the same destination behind the security gateways.
- For a tunnel mode SA, there is an "outer" IP header that specifies the IPsec processing destination, plus an "inner" IP header that specifies the (apparently) ultimate destination for the packet. The security protocol header appears after the outer IP header, and before the inner IP header. If AH is employed in tunnel mode, portions of the outer IP header are afforded protection (as above), as well as all of the tunneled IP packet (i.e., all of the inner IP header is protected, as well as higher layer

protocols). If ESP is employed, the protection is afforded only to the tunneled packet, not to the outer header.



IPSec Modes

Transport mode

- Encrypts payload only; leaves original headers

Tunnel mode

- Encrypts entire packet and adds new headers

Module 12: Securing Remote Access - slide 38

Transport Mode

A transport packet has the following characteristics:

- Original IP header. Since only the payload is encrypted and there is no tunnel established, the original IP header remains in place.
- Authenticated ESP Header
- Encrypted TCP packet (includes the TCP header, and the data)
- Encrypted ESP trailer
- ESP authentication trailer

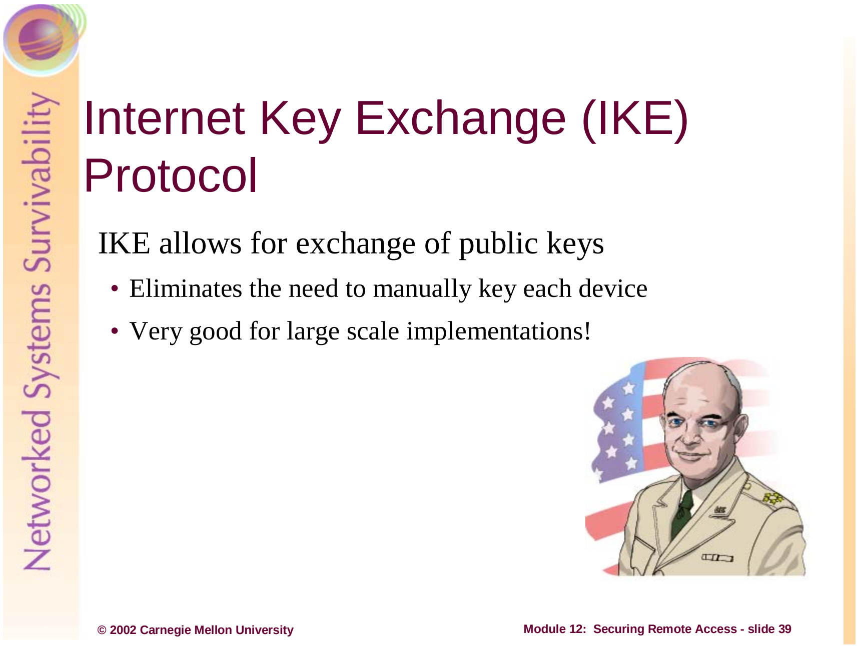
The original IP header is not encrypted, and thus can be read while traversing the Internet. This is practical for non-tunneling applications. The encryption of the data, along with the checksum provided in the ESP authentication trailer, provide confidentiality of the data, and integrity of the packet (except the original IP header).

Tunnel Mode

A completely constructed IPSec tunnel packet, when created using ESP, contains the following information:

- The original data packet, which has been encrypted
- The TCP header, which has been encrypted
- The original IP header, which has been encrypted
- The ESP header and trailer
- The ESP authentication trailer (for checksumming)
- And the new outer IP header (for routing across the Internet)

In examining the packet, you will see that only the outer IP header, the ESP header, and the ESP checksum are not encrypted, while only the outer IP header is left out of the checksum function. This provides strong proof that the packet has not been altered or read in transport.



Networked Systems Survivability

Internet Key Exchange (IKE) Protocol

IKE allows for exchange of public keys

- Eliminates the need to manually key each device
- Very good for large scale implementations!

© 2002 Carnegie Mellon University

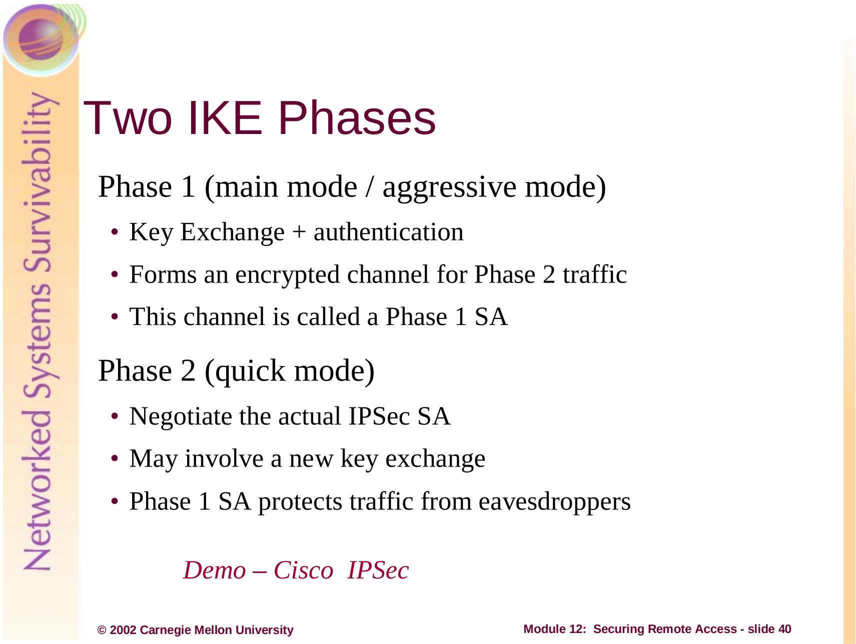
Module 12: Securing Remote Access - slide 39

The Internet Key Exchange is IPSec’s method for agreeing upon and using cryptographic keys. It draws on the foundations of public key cryptography. It is critical in large-scale implementations, because manually keying each device is cumbersome and may lead to ‘fat fingering’ issues.

The following cryptographic elements are used and present within IKE:

- Encryption algorithms (DES, 3DES)
- Hashing algorithms (MD5 and SHA-1)
- Authentication via pre-shared keys

IKE, like public key algorithms such as RSA and KEA, uses asymmetric encryption to secretly agree on a session key, which is used for a symmetric algorithm, to encrypt the traffic (due to the speed advantages that symmetric cryptography enjoys over asymmetric cryptography).



Networked Systems Survivability

Two IKE Phases

Phase 1 (main mode / aggressive mode)

- Key Exchange + authentication
- Forms an encrypted channel for Phase 2 traffic
- This channel is called a Phase 1 SA

Phase 2 (quick mode)

- Negotiate the actual IPSec SA
- May involve a new key exchange
- Phase 1 SA protects traffic from eavesdroppers

Demo – Cisco IPSec

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 40

There are two Phases of IKE, and three Modes. The Phases are numbered 1 and 2, and the modes are labeled Main mode, Aggressive mode, and Quick mode.

IPsec works hand-in-hand with IKE, or Internet Key Exchange. IKE provides a key exchange mechanism, when used in conjunction with IPsec you can encrypt data, create security associations (SA), and operate VPNs. IKE is further explained in RFC 2409. IKE provides the auto-management of cryptographic key exchange between security endpoints. Without IKE, you would have to manually key each device. This solution may be acceptable for small environments with few users, but it does not scale well. If you plan to have more than just a handful of systems passing IPsec traffic, then Internet Key Exchange is a very useful protocol to have in place. Also, it is important to note that when using manual-keyed IPsec, no replay protection is provided.

IKE uses a two-phase process for establishing the IPSec parameters between two IPSec nodes [RFC 2409].

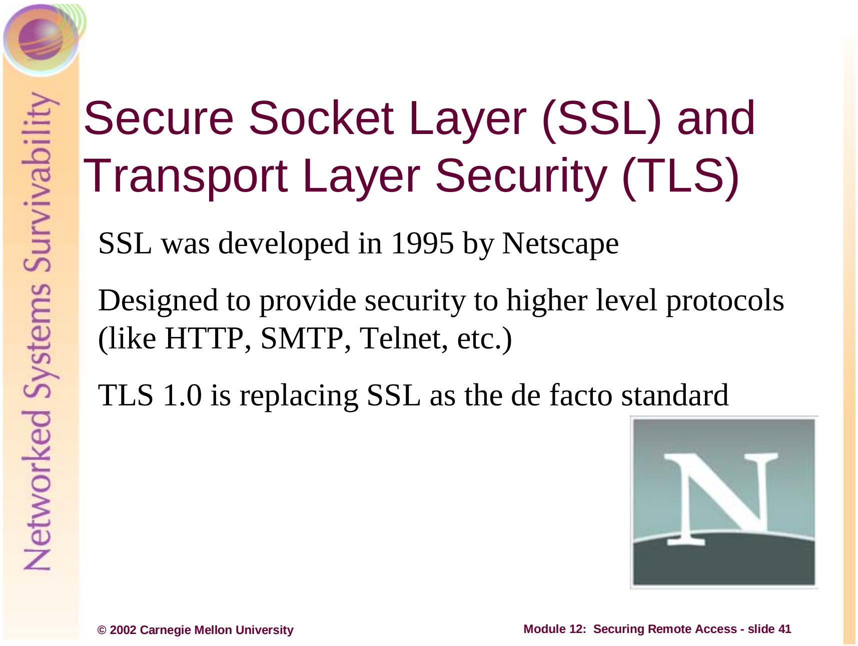
- Phase 1 is where the two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP Security Association (SA). "Main Mode" and "Aggressive Mode" each accomplish a phase 1 exchange. "Main Mode" and "Aggressive Mode" MUST ONLY be used in phase 1.
- Phase 2 is where Security Associations are negotiated on behalf of services such as IPsec or any other service which needs key material and/or parameter negotiation. "Quick Mode" accomplishes a phase 2 exchange. "Quick Mode" MUST ONLY be used in phase 2.

In Phase 1, Main Mode allows for the key exchange with identity protection, as the IKE SA is negotiated over a sequence. Aggressive Mode does not provide identity protection since all authentication data is sent at once. As a result, it should only be used when bandwidth is scarce, and security not completely crucial. During Phase 1, both IPSec nodes establish a connection where they authenticate each other. This is usually done using a pre-shared secret, or a X.509 digital certificate from a well-known, mutually trusted certificate authority. In Phase 2, IPsec creates the actual tunnels between IPsec hosts that will be used. Quick Mode can be used in Phase 2 since the SA was created during Phase 1, and it’s not necessary to repeat full authentication. Finally, the main purpose of Phase 2 is to exchange cryptographic keys, and get the IPsec VPN up and running.

Quick Mode is not a complete exchange itself (in that it is bound to a phase 1 exchange), but is used as part of the SA negotiation process (phase 2) to derive keying material and negotiate shared policy for non-ISAKMP SAs. The information exchanged along with Quick Mode MUST be protected by the ISAKMP SA-- i.e. all payloads except the ISAKMP header are encrypted. In Quick Mode, a HASH payload MUST immediately follow the ISAKMP header and a SA payload MUST immediately follow the HASH. This HASH authenticates the message and also provides liveliness proofs.

The message ID in the ISAKMP header identifies a Quick Mode in progress for a particular ISAKMP SA which itself is identified by the cookies in the ISAKMP header. Since each instance of a Quick Mode uses a unique initialization vector (see Appendix B) it is possible to have multiple simultaneous Quick Modes, based off a single ISAKMP SA, in progress at any one time.

Quick Mode is essentially a SA negotiation and an exchange of nonces that provides replay protection. The nonces are used to generate fresh key material and prevent replay attacks from generating bogus security associations. An optional Key Exchange payload can be exchanged to allow for an additional Diffie-Hellman exchange and exponentiation per Quick Mode. While use of the key exchange payload with Quick Mode is optional it MUST be supported [RFC 2409].



Networked Systems Survivability

Secure Socket Layer (SSL) and Transport Layer Security (TLS)

SSL was developed in 1995 by Netscape

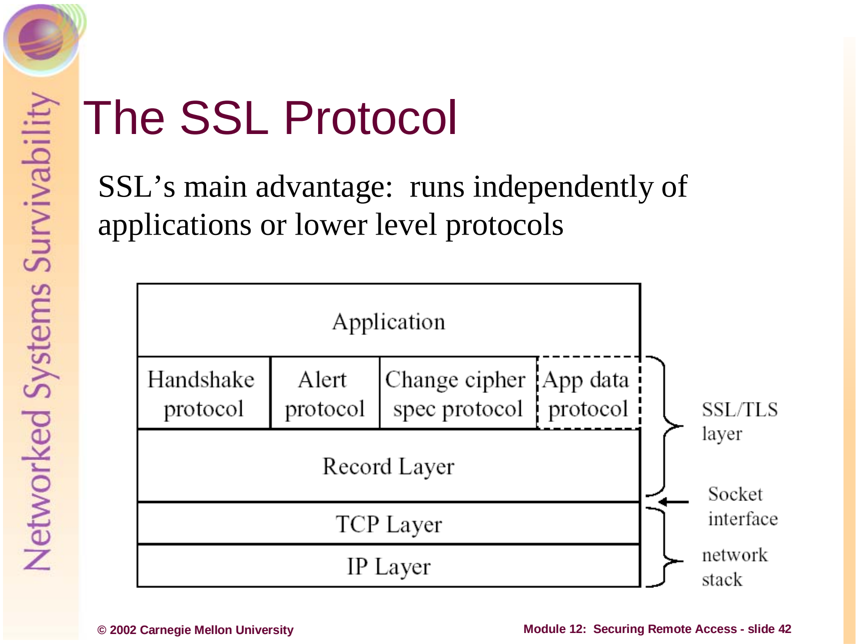
Designed to provide security to higher level protocols (like HTTP, SMTP, Telnet, etc.)

TLS 1.0 is replacing SSL as the de facto standard

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 41

For the needs of the Internet community to accomplish secure transmissions across the public (unsecure) network, Netscape released a protocol in 1995 called Secure Socket Layer (SSL). SSL was designed to sit below the application layer in the protocol stack, and to offer encrypted data transfers independent of the lower layers. It can support encryption of practically any higher-level protocol (such as SMTP, HTTP, etc.). While SSL was not without its security issues (vis-à-vis the faulty random seed for the 128 bit secret key), it was a nice step forward for the Internet commerce community. Netscape has released various versions with 3.0 being the most recent. Microsoft and other browser developers have implemented SSL capabilities in their browsers as well. While SSL has never been implemented as an IETF standard, it has been the de facto standard since its inception. Recently, however, the Transport Security Layer (TLS) has begun to unseat SSL as the industry 'standard' – and may in fact be adopted by the IETF as a standard at some point in the future.



Networked Systems Survivability

The SSL Protocol

SSL’s main advantage: runs independently of applications or lower level protocols

Application

Handshake protocol | Alert protocol | Change cipher spec protocol | App data protocol

Record Layer

TCP Layer

IP Layer

SSL/TLS layer

Socket interface

network stack

© 2002 Carnegie Mellon University

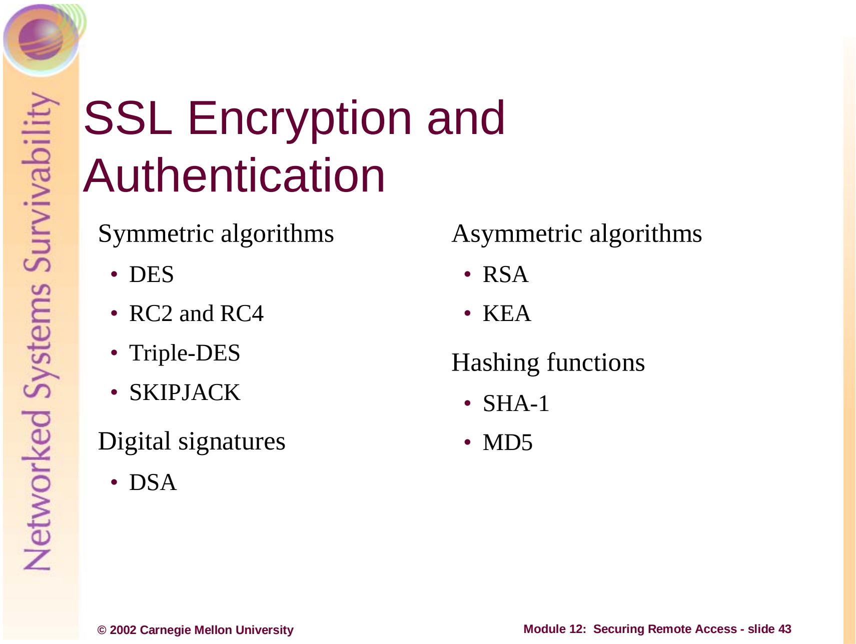
Module 12: Securing Remote Access - slide 42

From the Transport Layer Security Working Group draft on SSL 3.0:

The primary goal of the SSL Protocol is to provide privacy and reliability between two communicating applications. The protocol is composed of two layers. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the SSL Record Protocol. The SSL Record Protocol is used for encapsulation of various higher-level protocols. One such encapsulated protocol, the SSL Handshake Protocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. One advantage of SSL is that it is application protocol independent. A higher-level protocol can layer on top of the SSL Protocol transparently. The SSL protocol provides connection security that has three basic properties:

- The connection is private. Asymmetric encryption is used after an initial handshake to define a secret key. Symmetric cryptography is used for data encryption (e.g., DES, RC4, etc.)
- The peer's identity can be authenticated using asymmetric, or public key, cryptography (e.g., RSA, DSS, etc.).
- The connection is reliable. Message transport includes a message integrity check using a keyed Message Authentication Code (MAC). Secure hash functions (e.g., SHA, MD5, etc.) are used for MAC computations.

As discussed in the draft from the Transport Layer Security Group, SSL’s two layers sit between the application that is using SSL to communicate and the lower layer network protocols.


Networked Systems Survivability

SSL Encryption and Authentication

Symmetric algorithms

- DES
- RC2 and RC4
- Triple-DES
- SKIPJACK

Digital signatures

- DSA

Asymmetric algorithms

- RSA
- KEA

Hashing functions

- SHA-1
- MD5

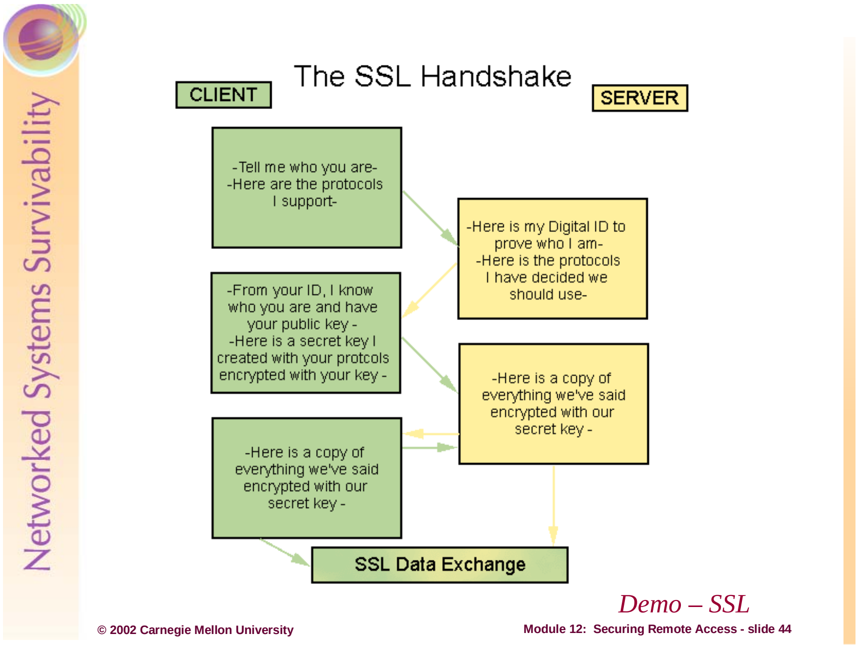
© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 43

The four cryptographic operations are designated: digitally-signed, stream-ciphered, block-ciphered, and public-key-encrypted, respectively. A field's cryptographic processing is specified by prepending an appropriate key word designation before the field's type specification. Cryptographic keys are implied by the current session state.

SSL incorporates many common symmetric (DES, Triple-DES, Skipjack, RC2 and RC4) and public key (RSA, and KEA) cryptographic algorithms into the protocol, as well as common cryptographic hashing functions (MD5 and SHA-1) and digital signature capabilities (DSA).

The KEA, like Diffie-Hellman, is a key exchange algorithm. All calculations for the KEA require a 1024-bit prime modulus. The modulus and related values are generated per the Digital Signature Standard (DSS) specification. The KEA is based on Diffie-Hellman, and uses SKIPJACK to reduce the final values to an 80-bit key.



The SSL Handshake

-Tell me who you are-
-Here are the protocols I support-

-Here is my Digital ID to prove who I am-
-Here is the protocols I have decided we should use-

-From your ID, I know who you are and have your public key -
-Here is a secret key I created with your protcols encrypted with your key -

-Here is a copy of everything we've said encrypted with our secret key -

-Here is a copy of everything we've said encrypted with our secret key -

SSL Data Exchange

Demo – SSL

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 44

5.5 Handshake protocol overview

The cryptographic parameters of the session state are produced by the SSL Handshake Protocol, which operates on top of the SSL Record Layer. When a SSL client and server first start communicating, they agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public-key encryption techniques to generate shared secrets. These processes are performed in the handshake protocol, which can be summarized as follows: The client sends a client hello message to which the server must respond with a server hello message, or else a fatal error will occur and the connection will fail. The client hello and server hello are used to establish security enhancement capabilities between client and server. The client hello and server hello establish the following attributes: Protocol Version, Session ID, Cipher Suite, and Compression Method. Additionally, two random values are generated and exchanged: ClientHello.random and ServerHello.random.

Following the hello messages, the server will send its certificate, if it is to be authenticated. Additionally, a server key exchange message may be sent, if it is required (e.g. if their server has no certificate, or if its certificate is for signing only). If the server is authenticated, it may request a certificate from the client, if that is appropriate to the cipher suite selected. Now the server will send the server hello done message, indicating that the hello-message phase of the handshake is complete. The server will then wait for a client response. If the server has sent a certificate request Message, the client must send either the certificate message or a no_certificate alert. The client key exchange message is now sent, and the content of that message will depend on the public key algorithm selected between the client hello and the server hello. If the client has sent a certificate with signing ability, a digitally-signed certificate verify message is sent to explicitly verify the certificate.

At this point, a change cipher spec message is sent by the client, and the client copies the pending Cipher Spec into the current Cipher Spec. The client then immediately sends the finished message under the new algorithms, keys, and secrets. In response, the server will send its own change cipher spec message, transfer the pending to the current Cipher Spec, and send its finished message under the new Cipher Spec.

At this point, the handshake is complete and the client and server may begin to exchange application layer data.

```
  Client                                                Server

  ClientHello                   -------->
                                                   ServerHello
                                                  Certificate*
                                            ServerKeyExchange*
                                           CertificateRequest*
                                <--------      ServerHelloDone
  Certificate*
  ClientKeyExchange
  CertificateVerify*
  [ChangeCipherSpec]
  Finished                      -------->
                                            [ChangeCipherSpec]
                                <--------             Finished
  Application Data              <------->     Application Data
```

* Indicates optional or situation-dependent messages that are not always sent.

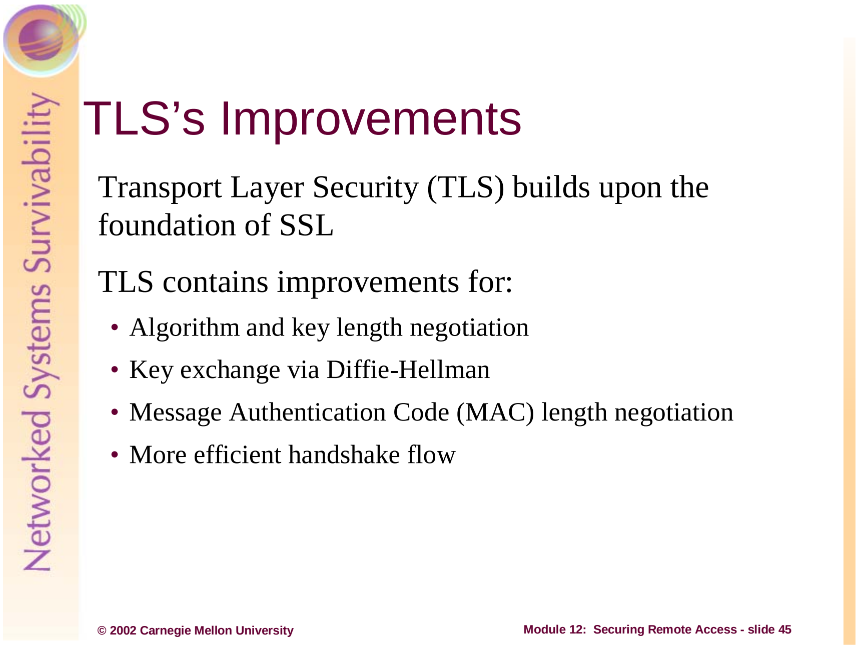
Note: To help avoid pipeline stalls, ChangeCipherSpec is an independent SSL Protocol content type, and is not actually an SSL handshake message.

When the client and server decide to resume a previous session or duplicate an existing session (instead of negotiating new security parameters) the message flow is as follows:

The client sends a ClientHello using the Session ID of the session to be resumed. The server then checks its session cache for a match. If a match is found, and the server is willing to re-establish the connection under the specified session state, it will send a ServerHello with the same Session ID value. At this point, both client and server must send change cipher spec messages and proceed directly to finished messages. Once the re-establishment is complete, the client and server may begin to exchange application layer data. (See flow chart below.) If a Session ID match is not found, the server generates a new session ID and the SSL client and server perform a full handshake.

```
Client                                                Server

  ClientHello                   -------->
                                                   ServerHello
                                          [change cipher spec]
                                <--------             Finished
  change cipher spec
  Finished                      -------->
 Application Data          <------->     Application Data
```



Networked Systems Survivability

TLS's Improvements

Transport Layer Security (TLS) builds upon the foundation of SSL

TLS contains improvements for:

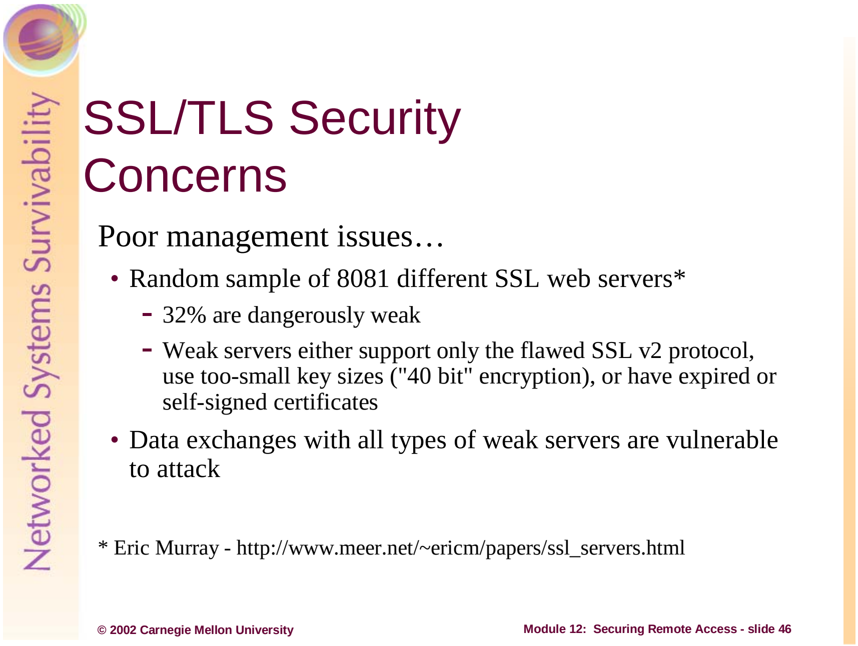
- Algorithm and key length negotiation
- Key exchange via Diffie-Hellman
- Message Authentication Code (MAC) length negotiation
- More efficient handshake flow

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 45

Transport Layer Security (TLS) has made some improvements over the SSL protocol. The main components of the protocols are identical, but TLS adds the following attributes:

- Algorithm negotiation. This will allow TLS users to negotiate the encryption protocol with the TLS server – and to negotiate the length of the secret key for the symmetric session.
- Diffie-Hellman key exchange support has been added.
- Message Authentication. Various algorithms are supported for Message Authentication Codes (MACs) – which can now be of different lengths. Negotiation of the MAC length is another added feature.
- And finally, a more efficient handshake flow has been established – to streamline the connection process.


Networked Systems Survivability

SSL/TLS Security Concerns

Poor management issues…

- Random sample of 8081 different SSL web servers*
  - 32% are dangerously weak
  - Weak servers either support only the flawed SSL v2 protocol, use too-small key sizes ("40 bit" encryption), or have expired or self-signed certificates
- Data exchanges with all types of weak servers are vulnerable to attack

* Eric Murray - http://www.meer.net/~ericm/papers/ssl_servers.html

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 46

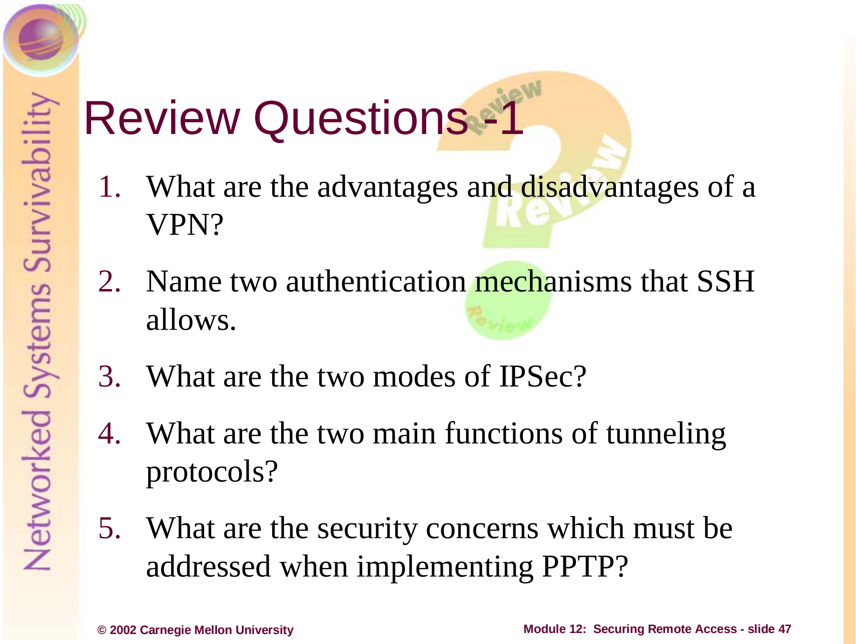
SSL/TLS, like any protocol which includes cryptography, must be implemented and administered correctly to avoid weak keys and weak encryption.

Take a look at Kurt Seifried's article on dsniff 2.3 and its man-in-the-middle attack possibilities against SSL and SSH at: http://www.seifried.org/security/cryptography/20011108-end-of-ssl-ssh.html

The main flaws in SSL come from poor implementations, as evidenced by the following CERT/CC advisories:

http://www.cert.org/advisories/CA-2000-05.html

http://www.cert.org/advisories/CA-2002-23.html



Networked Systems Survivability

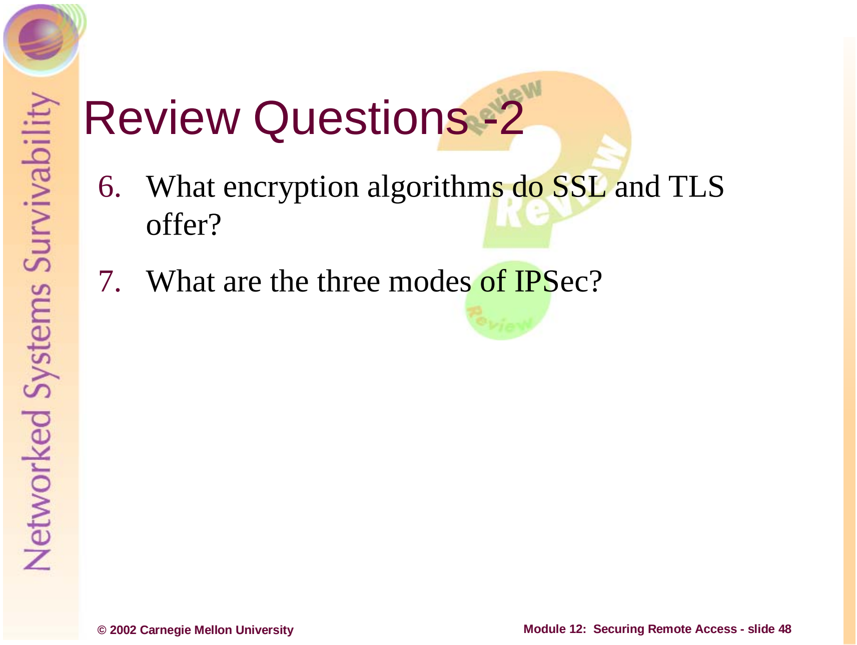
Review Questions -1

1. What are the advantages and disadvantages of a VPN?
2. Name two authentication mechanisms that SSH allows.
3. What are the two modes of IPSec?
4. What are the two main functions of tunneling protocols?
5. What are the security concerns which must be addressed when implementing PPTP?

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 47

1. Advantages: Cost, speed of implementation. Disadvantages: QoS, Security impacts
2. Password, Public key, Hostbased, Kerberos5, SecurID
3. Transport mode and tunnel mode
4. To encapsulate the packet in its own header, and to provide address translation (when needed)
5. In a mixed environment, there is bad password management. Also, there are flaws in the encryption algorithm, and it can be susceptible to replay attacks.



Networked Systems Survivability

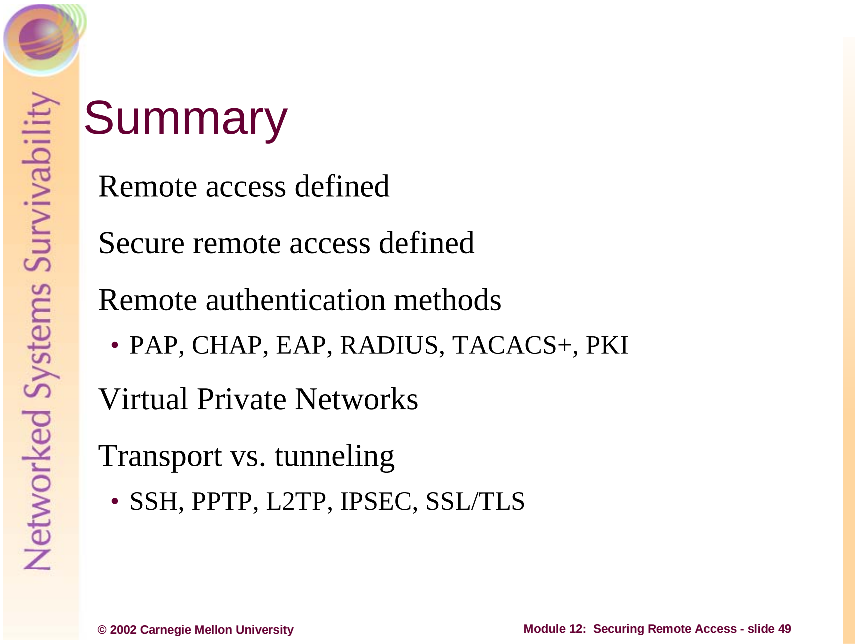
Review Questions -2

6. What encryption algorithms do SSL and TLS offer?
7. What are the three modes of IPSec?

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 48

6. Symmetric Algorithms (DES, RC2, RC4, Triple-DES, SKIPJACK); Asymmetric Algorithms (RSA, KEA)
7. Main mode; aggressive mode; quick mode



Networked Systems Survivability

Summary

Remote access defined

Secure remote access defined

Remote authentication methods

- PAP, CHAP, EAP, RADIUS, TACACS+, PKI

Virtual Private Networks

Transport vs. tunneling

- SSH, PPTP, L2TP, IPSEC, SSL/TLS

© 2002 Carnegie Mellon University

Module 12: Securing Remote Access - slide 49

We’ve seen that there is a need for remote access within the enterprise today. In order to keep our data secure when we allow remote access into our network, we can use a variety of protocols and authentication methods. Strong authentication (as is offered by EAP and PKI) along with strong encryption (as is offered by SSL and IPSec) can keep our data secure.

References:

Kevin Downes et. al., *Internetworking Technologies Handbook.* Cisco Press, 1998.

Harkins & Carrel. "The Internet Key Exchange". RFC 2409. November 1998. URL: http://www.ietf.org/rfc/rfc2409.txt

Fraser, Barbara and Ts’o, Theodore. IETF IPsec Working Group Charter. 31 July 2001. URL: http://www.ietf.org/html.charters/ipsec-charter.html (15 September 2001)

Microsoft Corporation. *Designing Microsoft Windows 2000 Network Security*. Microsoft Press, 2001.

Wenstrom, Michael. *Managing Cisco Network Security.* Cisco Press, 2001.

Paquet, Catherine. *Building Cisco Remote Access Networks.* Cisco Press, 2002.