UNIT - V: Network and Internet Security

IP Security: IP Security Overview, IPSec documents, IPSec Services, IPSec AH header, and Encapsulating Security Payload.

Transport Level Security: Web Security Requirements, Secure Socket Layer (SSL) and Transport Layer Security (TLS),

System Security: Firewalls- Characteristics, Types of Firewalls, Intruders.

IP Security Overview : The Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636). The report identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IPv4 and the future IPv6.

Applications of IPsec IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

• **Secure branch office connectivity over the Internet:** A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead. • **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network.This reduces the cost of toll charges for traveling employees and telecommuters. • **Establishing extranet and intranet connectivity with partners:** IPsec can be used to secure communication with other organizations, ensuring authentica tion and confidentiality and providing a key exchange mechanism.

• **Enhancing electronic commerce security:** Even though some Web and electronic commerce applications have built-in security protocols, the use of IPsec enhances that security. IPsec guarantees that all traffic designated by the network adminis trator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

IPSec documents :

IPsec encompasses three functional areas: authentication, confidentiality, and key management. The totality of the IPsec specification is scattered across dozens of RFCs and draft IETF documents, making this the most complex and difficult to grasp of all IETF specifications. The best way to grasp the scope of IPsec is to consult the latest version of the

IPsec document roadmap, which as of this writing is [FRAN09]. The documents can be categorized into the following groups.

• **Architecture:** Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology. The current specification is RFC 4301, Security Architecture for the Internet Protocol.

• **Authentication Header (AH):** AH is an extension header to provide message authentication. The current specification is RFC 4302, IP Authentication Header. Because message authentication is provided by ESP, the use of AH is deprecated. It is included in IPsecv3 for backward compatibility but should not be used in new applications. We do not discuss AH in this chapter.

• **Encapsulating Security Payload (ESP):** ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/authentication. The current specification is RFC 4303,IP Encapsulating Security Payload (ESP).

• **Internet Key Exchange (IKE):** This is a collection of documents describing the key management schemes for use with IPsec. The main specification is RFC 4306,Internet Key Exchange (IKEv2) Protocol, but there are a number of related RFCs.

• **Cryptographic algorithms:** This category encompasses a large set of docu ments that define and describe cryptographic algorithms for encryption, mes sage authentication, pseudorandom functions (PRFs), and cryptographic key exchange.

• **Other:** There are a variety of other IPsec-related RFCs, including those deal ing with security policy and management information base (MIB) content.

IPsec Services IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services. Two protocols are used to provide security: an authentication protocol designated by the header of the protocol, Authentication Header (AH); and a combined encryption/ authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP). RFC 4301 lists the following services: • Access control

• Connectionless integrity

• Data origin authentication

• Rejection of replayed packets (a form of partial sequence integrity)

• Confidentiality (encryption)

• Limited traffic flow confidentiality

Table 19.1 Tunnel Mode and Transport Mode Functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

IPSec AH (Authentication Header)

IPSec (Internet Protocol Security) is a suite of protocols designed to ensure the security of communications over IP networks. One of its core components is the **Authentication Header (AH)**, which provides data integrity, data origin authentication, and protection against replay attacks.

Key Features of AH

1. **Data Integrity:** Ensures that the data has not been altered during transmission.
2. **Data Origin Authentication:** Verifies the source of the data, ensuring that it comes from a legitimate sender.
3. **Replay Protection:** Protects against replay attacks by using sequence numbers.

AH Header Format

The AH header is inserted into the IP packet, either between the IP header and the payload (transport mode) or before the encapsulated IP header (tunnel mode). The AH header format is as follows:

1. **Next Header (8 bits):** Indicates the type of the next payload after the AH header (e.g., TCP, UDP, ICMP).
2. **Payload Length (8 bits):** Specifies the length of the AH header in 32-bit words minus two (this value is fixed for IPv4).
3. **Reserved (16 bits):** Reserved for future use and set to zero.
4. **Security Parameters Index (SPI) (32 bits):** Identifies the security association (SA) for this packet. The SA contains the parameters required to process the packet.
5. **Sequence Number (32 bits):** A monotonically increasing counter value, providing anti-replay protection.
6. **Authentication Data (variable length):** Contains the Integrity Check Value (ICV), which is the result of applying the authentication algorithm to the packet. The ICV ensures data integrity and authenticity.

AH in Transport and Tunnel Modes

- **Transport Mode:**
  - The AH header is inserted between the IP header and the transport layer (TCP, UDP) header.
  - Used for end-to-end communications.
- **Tunnel Mode:**
  - The entire original IP packet is encapsulated within a new IP packet with the AH header inserted before the new IP header.
  - Used for network-to-network or host-to-network communications, such as in VPNs.

Advantages and Disadvantages

Advantages:

- Provides strong data integrity and authentication.
- Simple and efficient for integrity and authenticity checks.

Disadvantages:

- Does not provide data confidentiality (encryption).
- Increases packet size, leading to additional overhead.
- Some fields in the IP header (e.g., TTL) can change during transit, complicating authentication.

Example

Consider a scenario where Host A wants to send a secure message to Host B using IPSec AH:

1. Host A and Host B establish an SA.
2. Host A constructs the AH header, calculates the ICV, and inserts the AH header into the IP packet.
3. Host A sends the packet to Host B.
4. Host B receives the packet, retrieves the SA, and verifies the ICV.
5. If the ICV matches, Host B processes the packet; otherwise, it discards it.

By providing data integrity, origin authentication, and replay protection, AH plays a crucial role in securing IP communications in various network environments.

Encapsulating Security Payload :

ESP can be used to provide confidentiality, data origin authentication, connection less integrity, an anti-replay service (a form of partial sequence integrity), and (lim ited) traffic flow confidentiality. The set of services provided depends on options selected at the time of Security Association (SA) establishment and on the location of the implementation in a network topology. ESP can work with a variety of encryption and authentication algorithms, including authenticated encryption algorithms such as GCM.

ESP Format Figure 19.5a shows the top-level format of an ESP packet. It contains the following fields.

• **Security Parameters Index (32 bits):** Identifies a security association.

• **Sequence Number (32 bits):** A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.

• **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.

• **Padding (0 – 255 bytes):** The purpose of this field is discussed later.

• **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.

• **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP).

• **Integrity Check Value (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

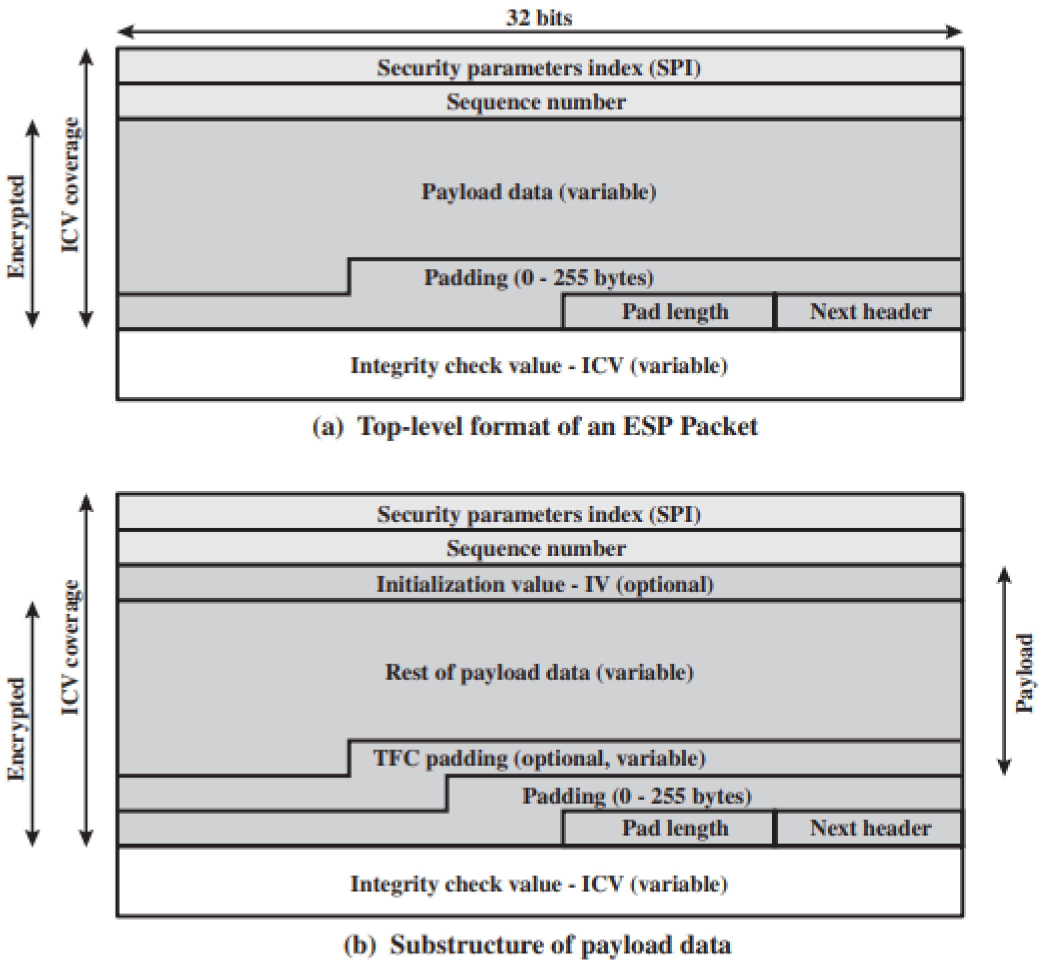

Figure 19.5 ESP Packet Format

When any combined mode algorithm is employed, the algorithm itself is expected to return both decrypted plaintext and a pass/fail indication for the integrity check. For combined mode algorithms, the ICV that would normally appear at the end of the ESP packet (when integrity is selected) may be omitted. When the ICV is omitted and integrity is selected, it is the responsibility of the combined mode algorithm to encode within the Payload Data an ICV-equivalent means of verifying the integrity of the packet. Two additional fields may be present in the payload (Figure 19.5b). An initialization value (IV), or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for ESP. If tunnel mode is being used, then the IPsec implementation may add traffic flow confidentiality (TFC) padding after the Payload Data and before the Padding field, as explained subsequently.

Web Security Requirements :

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets. The Internet is two-way. Unlike traditional publishing environments even electronic publishing systems involving teletext, voice response, or fax-back the Web is vulnerable to attacks on the Web servers over the Internet

• The Web is increasingly serving as a highly visible outlet for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.

• Although Web browsers are very easy to use, Web servers are relatively easy to configure and manage, and Web content is increasingly easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws. The short history of the Web is filled with examples of new and upgraded systems, properly installed, that are vulnerable to a variety of security attacks.

• A Web server can be exploited as a launching pad into the corporation’s or agency’s entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data and systems not part of the Web itself but connected to the server at the local site.

• Casual and untrained (in security matters) users are common clients for Web-based services. Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures

Secure Socket Layer (SSL) :

SSL Architecture

SSL is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols, as illustrated in Figure 16.2. The SSL Record Protocol provides basic security services to various higher layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which pro vides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges and are examined later in this section. Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows.

• **Connection:** A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

• **Session:** An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice.

There are a number of states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states. A session state is defined by the following parameters.

• **Session identifier**: An arbitrary byte sequence chosen by the server to identify an active or resumable session state.

• **Peer certificate**: An X509.v3 certificate of the peer. This element of the state may be null.

• **Compression method**: The algorithm used to compress data prior to encryption.

• **Cipher spec:** Specifies the bulk data encryption algorithm (such as null, AES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash_size.

• **Master secret:** 48-byte secret shared between the client and server.

• **Is resumable**: A flag indicating whether the session can be used to initiate new connections.

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

• **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

• **Message Integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC). Figure 16.3 indicates the overall operation of the SSL Record Protocol. The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled before being delivered to higher-level users.

The first step is fragmentation. Each upper-layer message is fragmented into blocks of 214 bytes (16384 bytes) or less. Next, compression is optionally applied. Compression must be lossless and may not increase the content length by more than 1024 bytes.1 In SSLv3 (as well as the current version of TLS), no compression algorithm is specified, so the default compression algorithm is null.

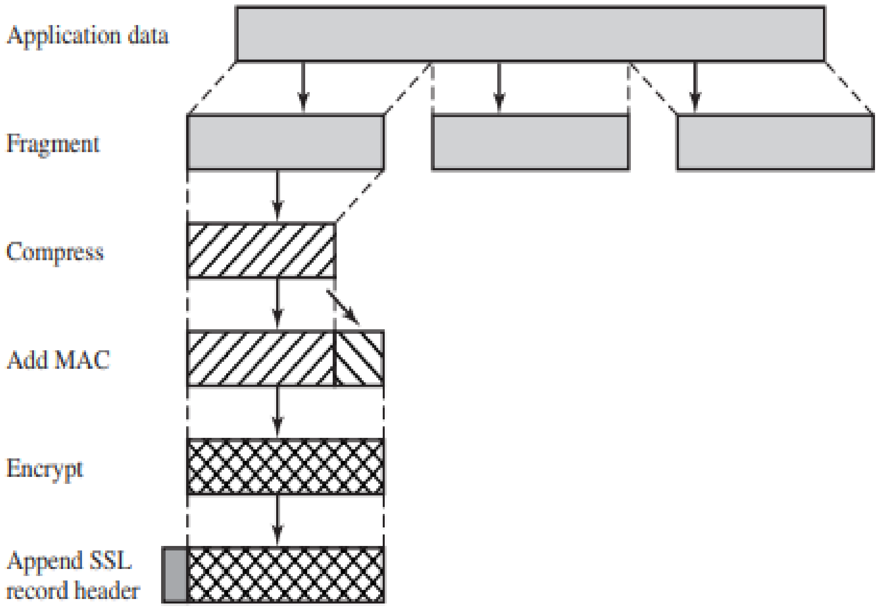

Figure 16.3 SSL Record Protocol Operation

Change Cipher Spec Protocol :

The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a single message (Figure 16.5a), which consists of a single byte with the value 1.The sole pur pose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

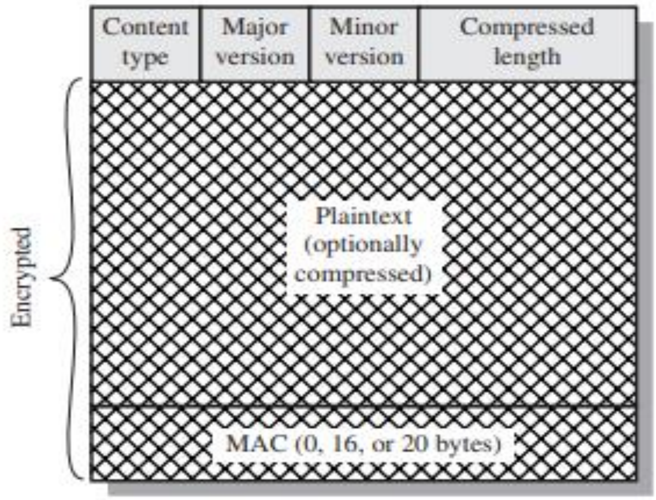

Figure 16.4 SSL Record Format

Alert Protocol The Alert Protocol is used to convey SSL-related alerts to the peer entity. As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state.

Each message in this protocol consists of two bytes (Figure 16.5b). The first byte takes the value warning (1) or fatal (2) to convey the severity of the message. If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert. First, we list those alerts that are always fatal (definitions from the SSL specification).

• **unexpected_message**: An inappropriate message was received.

• **bad_record_mac**: An incorrect MAC was received.

• **decompression_failure:** The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).

• **handshake_failure**: Sender was unable to negotiate an acceptable set of security parameters given the options available.

• **illegal_parameter**: A field in a handshake message was out of range or inconsistent with other fields.

Handshake Protocol

The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted.

The Handshake Protocol consists of a series of messages exchanged by client and server. All of these have the format shown in Figure 16.5c. Each message has three fields:

• Type (1 byte): Indicates one of 10 messages. Table 16.2 lists the defined message types.

• Length (3 bytes): The length of the message in bytes.

• Content (bytes): The parameters associated with this message; these are listed in Table 16.2.

Table 16.2 SSL Handshake Protocol Message Types

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Transport Layer Security (TLS) :

TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL. TLS is defined as a Proposed Internet Standard in RFC 5246. RFC 5246 is very similar to SSLv3.

Version Number

The TLS Record Format is the same as that of the SSL Record Format (Figure 16.4), and the fields in the header have the same meanings. The one differ ence is in version values. For the current version of TLS, the major version is 3 and the minor version is 3.

Message Authentication Code

There are two differences between the SSLv3 and TLS MAC schemes: the actual algorithm and the scope of the MAC calculation. TLS makes use of the HMAC algorithm defined in RFC 2104. Recall from Chapter 12 that HMAC is defined as

$$\mathrm{HMAC}_K(M) = \mathrm{H}[(K^+ \oplus \mathrm{opad}) \| \mathrm{H}[(K^+ \oplus \mathrm{ipad}) \| M]]$$

where

- H = embedded hash function (for TLS, either MD5 or SHA-1)
- M = message input to HMAC
- K^+ = secret key padded with zeros on the left so that the result is equal to the block length of the hash code (for MD5 and SHA-1, block length = 512 bits)
- ipad = 00110110 (36 in hexadecimal) repeated 64 times (512 bits)
- opad = 01011100 (5C in hexadecimal) repeated 64 times (512 bits)

SSLv3 uses the same algorithm, except that the padding bytes are concatenated with the secret key rather than being XORed with the secret key padded to the block length. The level of security should be about the same in both cases.

For TLS, the MAC calculation encompasses the fields indicated in the following expression:

```
MAC(MAC_write_secret,seq_num || TLSCompressed.type ||
    TLSCompressed.version || TLSCompressed.length ||
    TLSCompressed.fragment)
```

The MAC calculation covers all of the fields covered by the SSLv3 calculation, plus the field TLSCompressed.version, which is the version of the protocol being employed.

Pseudorandom Function

TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation. The objective is to make use of a relatively small shared secret value but to generate longer blocks of data in a way that is secure from the kinds of attacks made on hash functions and MACs. The PRF is based on the data expansion function (Figure 16.7) given as

```
P_hash(secret, seed) = HMAC_hash(secret,A(1) || seed) ||
                       HMAC_hash(secret, A(2) || seed) ||
                       HMAC_hash(secret, A(3) || seed) || . . .
```

where A() is defined as

$$A(0) = \text{seed}$$
$$A(i) = \text{HMAC\_hash}(\text{secret}, A(i-1))$$

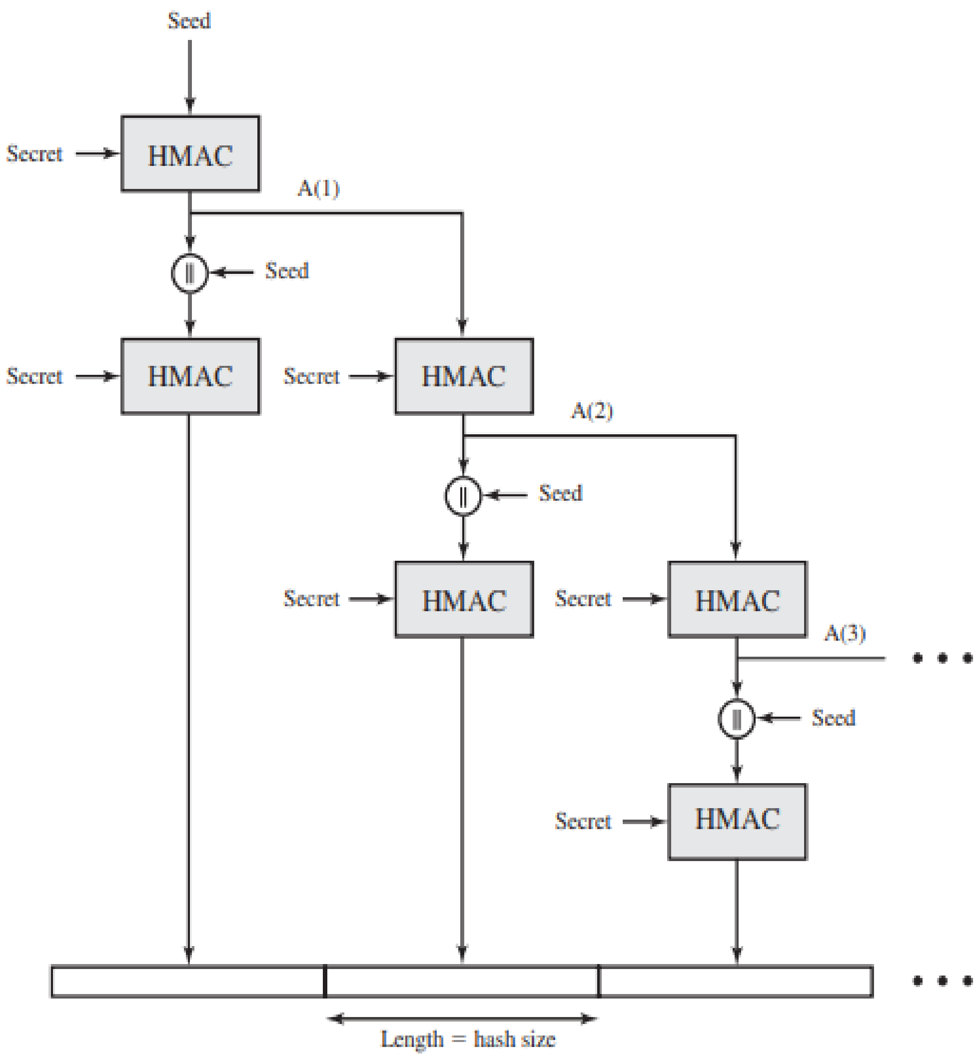

Figure 16.7 TLS Function P_hash(secret, seed)

Alert Codes

TLS supports all of the alert codes defined in SSLv3 with the exception of no_certificate. A number of additional codes are defined in TLS; of these, the following are always fatal.

• **record_overflow**: A TLS record was received with a payload (ciphertext) whose length exceeds bytes, or the ciphertext decrypted to a length of greater than bytes.

• **unknown_ca:** A valid certificate chain or partial chain was received, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA.

• **access_denied**: A valid certificate was received, but when access control was applied, the sender decided not to proceed with the negotiation.

• **decode_error:** A message could not be decoded, because either a field was out of its specified range or the length of the message was incorrect.

• **protocol_version**: The protocol version the client attempted to negotiate is recognized but not supported.

• **insufficient_security**: Returned instead of handshake_failure when a negotiation has failed specifically because the server requires ciphers more secure than those supported by the client.

• **unsupported_extension**: Sent by clients that receive an extended server hello containing an extension not in the corresponding client hello.

• **internal_error:** An internal error unrelated to the peer or the correctness of the protocol makes it impossible to continue.

• **decrypt_error**: A handshake cryptographic operation failed, including being unable to verify a signature, decrypt a key exchange, or validate a finished message.

Firewalls Characteristics :

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later in this chapter.

2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies, as explained later in this chapter.

3. The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications.

Firewalls focused primarily on service control, but they have since evolved to provide all four:

• Service control: Determines the types of Internet services that can be accessed, inbound or outbound. The firewall may filter traffic on the basis of IP address, protocol, or port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as a Web or mail service.

• Direction control: Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall.

- User control: Controls access to a service according to which user is attempt ing to access it. This feature is typically applied to users inside the firewall perimeter (local users). It may also be applied to incoming traffic from external users; the latter requires some form of secure authentication technology, such as is provided in IPsec .

• Behavior control: Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

Types of Firewalls :

A firewall may act as a packet filter. It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria. Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pat tern generated by a sequence of packets. we look at the principal types of firewalls.

Packet Filtering Firewall

A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet (Figure 22.1b). The firewall is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet:

• **Source IP address:** The IP address of the system that originated the IP packet (e.g., 192.178.1.1)

• **Destination IP address:** The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2)

• **Source and destination transport-level address:** The transport-level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET

• **IP protocol field**: Defines the transport protocol

• **Interface:** For a firewall with three or more ports, which interface of the fire wall the packet came from or which interface of the firewall the packet is destined for

Stateful Inspection Firewalls

A traditional packet filter makes filtering decisions on an individual packet basis and does not take into consideration any higher layer context. To understand what is meant by context and why a traditional packet filter is limited with regard to con text, a little background is needed. Most standardized applications that run on top of TCP follow a client/server model. For example, for the Simple Mail Transfer Protocol (SMTP), e-mail is transmitted from a client system to a server system. The client system generates new e-mail messages, typically from user input. The server system accepts incoming e-mail messages and places them in the appropriate user mailboxes. SMTP operates by setting up a TCP connection between client and server, in which the TCP server port number, which identifies the SMTP server application, is 25. The TCP port number for the SMTP client is a number between 1024 and 65535 that is generated by the SMTP client. In general, when an application that uses TCP creates a session with a remote host, it creates a TCP connection in which the TCP port number for the remote (server) application is a number less than 1024 and the TCP port number for the local (client) application is a number between 1024 and 65535.The numbers less than 1024 are the “well-known” port numbers and are assigned permanently to particular applications (e.g., 25 for server SMTP). The numbers between 1024 and 65535 are generated dynamically and have temporary significance only for the lifetime of a TCP connection. A simple packet filtering firewall must permit inbound network traffic on all these high-numbered ports for TCP-based traffic to occur. This creates a vulnerability that can be exploited by unauthorized users. A stateful inspection packet firewall tightens up the rules for TCP traffic by creating a directory of outbound TCP connections, as shown in Table 22.2. There is an entry for each currently established connection. The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory. A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections (Figure 22.1c). Some stateful firewalls also keep track of TCP sequence numbers to prevent attacks that depend on the sequence number, such as session hijacking. Some even inspect limited amounts of application data for some well-known protocols like FTP, IM and SIPS commands, in order to identify and track related connections.

Application-Level Gateway

An application-level gateway, also called an application proxy, acts as a relay of application-level traffic (Figure 22.1d). The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features. Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level. A prime disadvantage of this type of gateway is the additional processing overhead on each connection. In effect, there are two spliced connections between the end users, with the gateway at the splice point, and the gateway must examine and forward all traffic in both directions.

Circuit-Level Gateway

A fourth type of firewall is the circuit-level gateway or circuit-level proxy (Figure 22.1e). This can be a stand-alone system or it can be a specialized func tion performed by an application-level gateway for certain applications. As with an application gateway, a circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed. A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections. In this configuration, the gateway can incur the processing overhead of examining incoming application data for forbidden functions but does not incur that overhead on outgoing data.

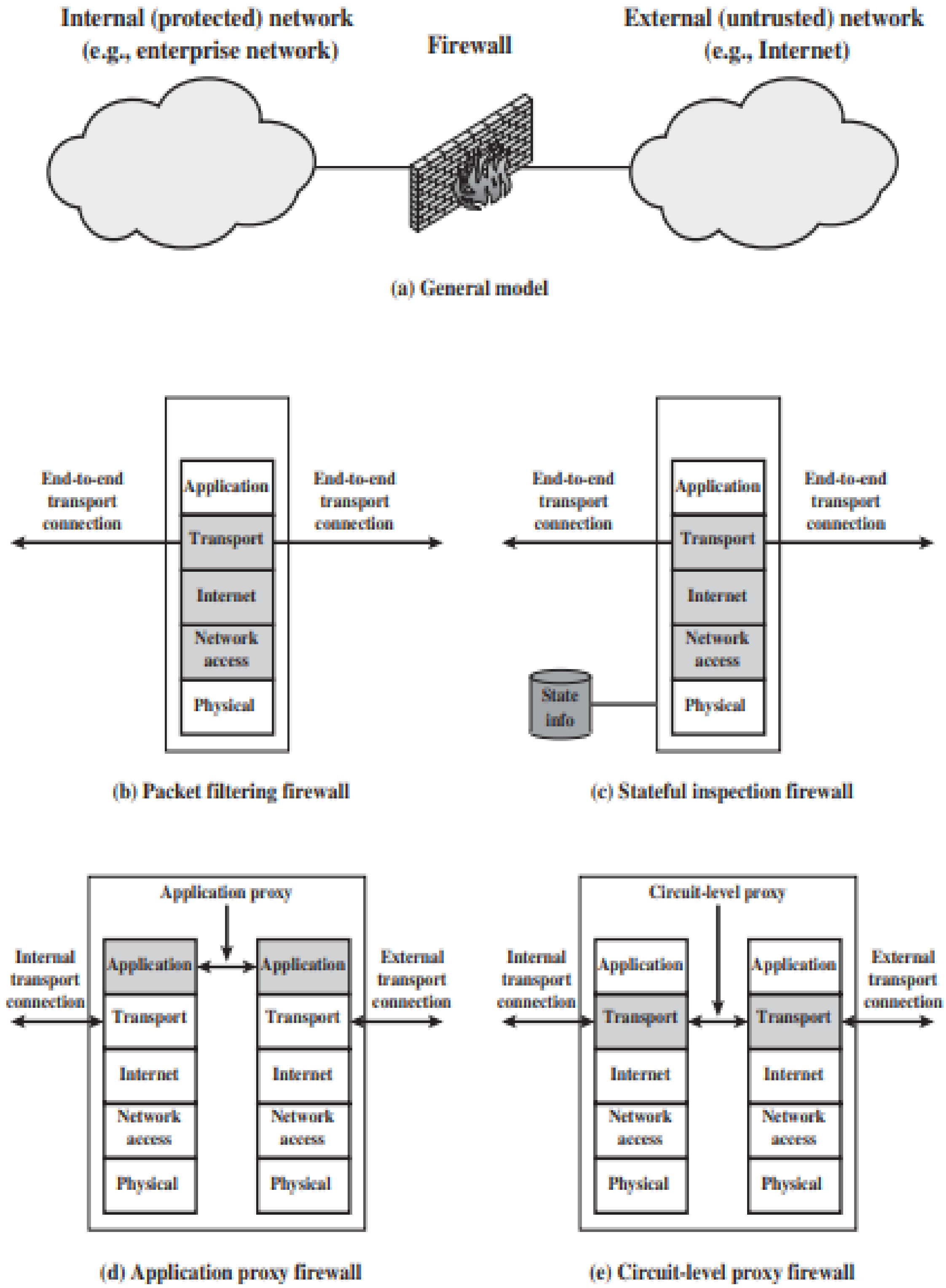

Figure 22.1 Types of Firewalls

Intruders :

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. In an important early study of intrusion, Anderson [ANDE80] identified three classes of intruders:

• **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account

• **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges

• **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider. Intruder attacks range from the benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

The following examples of intrusion:

• Performing a remote root compromise of an e-mail server

• Defacing a Web server • Guessing and cracking passwords

• Copying a database containing credit card numbers

• Viewing sensitive data, including payroll records and medical information, without authorization

• Running a packet sniffer on a workstation to capture usernames and pass□words

• Using a permission error on an anonymous FTP server to distribute pirated software and music files

• Dialing into an unsecured modem and gaining internal network access

• Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password

• Using an unattended, logged-in workstation without permission.

Intrusion Techniques :

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a back door into the system. Alternatively, the intruder attempts to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user. Typically, a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it and learn passwords. The password file can be protected in one of two ways:

• **One-way function:** The system stores only the value of a function based on the user's password. When the user presents a password, the system transforms that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the one-way function and in which a fixed-length output is produced.

• **Access control:** Access to the password file is limited to one or a very few accounts.

If one or both of these countermeasures are in place, some effort is needed for a potential intruder to learn passwords. On the basis of a survey of the literature and interviews with a number of password crackers, the following techniques for learning passwords:

1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.

2. Exhaustively try all short passwords.

3. Try words in the system's online dictionary or a list of likely passwords. Examples of the latter are readily available on hacker bulletin boards.

4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.

5. Try users' phone numbers, Social Security numbers, and room numbers.

6. Try all legitimate license plate numbers for this state.

7. Use a Trojan horse to bypass restrictions on access.

8. Tap the line between a remote user and the host system.

Assignment cum Tutorial Questions

A. Objective Questions

1.Which of the following is not a service provided by IPSec?

A) Data Confidentiality B) Data Integrity C) Authentication D) Data Compression

2. The Security Parameters Index (SPI) in IPSec is used to:

A) Identify the security protocol (AH or ESP) B) Identify the encryption algorithm used

C) Identify the security association D) Identify the sequence number of the packet

3. In IPSec, the AH protocol primarily provides which of the following?

A) Encryption B) Data Integrity and Authentication

C) Traffic Flow Confidentiality D) Data Compression

4. Which RFC document describes the Encapsulating Security Payload (ESP) in IPSec?

A) RFC 2401 B) RFC 2402 C) RFC 2406 D) RFC 2403

5. Which of the following protocols is the predecessor of TLS?

A) IPSec B) SSH C) SSL D) HTTP

6. The process of establishing a secure connection in TLS begins with which of the following?

A) Data Encryption B) Key Exchange C) Handshake Protocol D) Packet Filtering

7. Which of the following web security requirements is not directly addressed by TLS?

A) Confidentiality B) Integrity C) Authentication D) High Availability

8. Which of the following firewall types examines packets at the application layer?

A) Packet-Filtering Firewall B) Stateful Inspection Firewall

C) Proxy Firewall D) Next-Generation Firewall (NGFW)

9. In the context of network security, a "Masquerader" is best described as:

A) A legitimate user misusing their privileges

B) An unauthorized user posing as an authorized user

C) A user who bypasses security mechanisms to take control of a system

D) A program that infiltrates a system undetected

10. Which type of firewall maintains a state table to track active connections and make decisions based on the state of the connection?

A) Packet-Filtering Firewall B) Stateful Inspection Firewall

C) Proxy Firewall D) Application Firewall

B. Descriptive Questions

1. Explain the architecture of IPSec and its key components. How does IPSec ensure secure communication over IP networks?
2. Discuss the roles of the Authentication Header (AH) and Encapsulating Security Payload (ESP) in IPSec. Compare and contrast the services provided by AH and ESP.
3. What is a Security Association (SA) in IPSec? Describe how SAs are established and managed. Explain the importance of the Security Parameters Index (SPI) in this context.
4. Outline the steps involved in the IPSec protocol during the establishment of a secure connection. How do the AH and ESP headers ensure data integrity, authenticity, and confidentiality?
5. Examine the concept of anti-replay protection in IPSec. How do sequence numbers contribute to preventing replay attacks in both AH and ESP?
6. Describe the process of the TLS handshake. How does TLS ensure the security of data transmission between a client and a server during this process?
7. Compare and contrast the Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocols. Highlight the improvements introduced in TLS over SSL.
8. What are the key security requirements for web communication? Explain how TLS addresses these requirements, including confidentiality, integrity, and authentication.
9. Discuss the role of certificates and public key infrastructure (PKI) in the TLS protocol. How does certificate-based authentication work in TLS?
10. Explain the potential vulnerabilities and attacks against TLS. How can these vulnerabilities be mitigated to ensure secure communication?
11. Define a firewall and explain its importance in network security. Discuss the key characteristics and functions of firewalls in protecting networks.
12. Differentiate between packet-filtering firewalls, stateful inspection firewalls, and proxy firewalls. Provide examples of scenarios where each type would be most appropriate.
13. Discuss the concept of a Next-Generation Firewall (NGFW). What advanced features do NGFWs provide that go beyond traditional firewall capabilities?
14. Who are intruders in the context of system security? Describe the different types of intruders, including masqueraders, misfeasors, and clandestine users, and the threats they pose.
15. Examine the role of intrusion detection systems (IDS) and intrusion prevention systems (IPS) in enhancing system security. How do these systems work in conjunction with firewalls?