UNIT V
SECURITY PRACTICE AND SYSTEM SECURITY

Electronic Mail Security-PGP, S/MIME-IP Security, Web Security-SYSTEM SECURITY: Intruders-Malicious Software-Viruses-Firewalls

SECURITY SERVICES FOR ELECTRONIC MAIL

1. **Privacy**—the ability to keep anyone but the intended recipient from reading the message.
2. **Authentication**—reassurance to the recipient of the identity of the sender.
3. **Integrity**—reassurance to the recipient that the message has not been altered since it was transmitted by the sender.
4. **Non-repudiation**—the ability of the recipient to prove to a third party that the sender really did send the message. This feature is also sometimes called **third party authentication**. The term non-repudiation means that the sender cannot later deny sending the message.
5. **Proof of submission**—verification given to the sender that the message was handed to the mail delivery system.
6. **Proof of delivery**—verification that the recipient received the message. Postal mail has a similar feature (return receipt requested), but again it only verifies that something was delivered on a particular date to the recipient. With electronic mail it is possible to verify the contents, as we mentioned under proof of submission.
7. **Message flow confidentiality**—an extension of privacy such that Carol not only cannot know the content of the message Alice sent Bob, but cannot even determine whether Alice sent Bob a message.
8. **Anonymity**—the ability to send a message so that the recipient can't find out the identity of the sender.
9. **Containment**—the ability of the network to keep certain security levels of information from leaking out of a particular region.
10. **Audit**—the ability of the network to record events that might have some security relevance, such as that Alice sent a message to Bob on a particular date. This would be fairly straightforward to implement, but is not mentioned in any of the secure mail standards, so we don't have a section on it.
11. **Accounting**—the ability of the mail system to maintain system usage statistics. In addition to providing clues for system resource management, this information allows the mail system to charge its clients according to their usage.
12. **Self-destruct**—an option allowing a sender to specify that a message should be destroyed after delivery to the recipient. This allows Alice to send a message to Bob that Bob cannot forward or store. The mail system will decrypt and display the message, but then delete it.
13. **Message sequence integrity**—reassurance that an entire sequence of messages arrived in the order transmitted, without any loss.

PRETTY GOOD PRIVACY

PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications. In essence, Zimmermann has done the following:

1. Selected the best available cryptographic algorithms as building blocks.
2. Integrated these algorithms into a general-purpose application that is independent of operating system and processor.
3. Made the package and its documentation, including the source code, freely available via the Internet.
4. Entered into an agreement with a company to provide a fully compatible, low-cost commercial version of PGP.

PGP has grown explosively and is now widely used. A number of reasons can be cited for this growth:

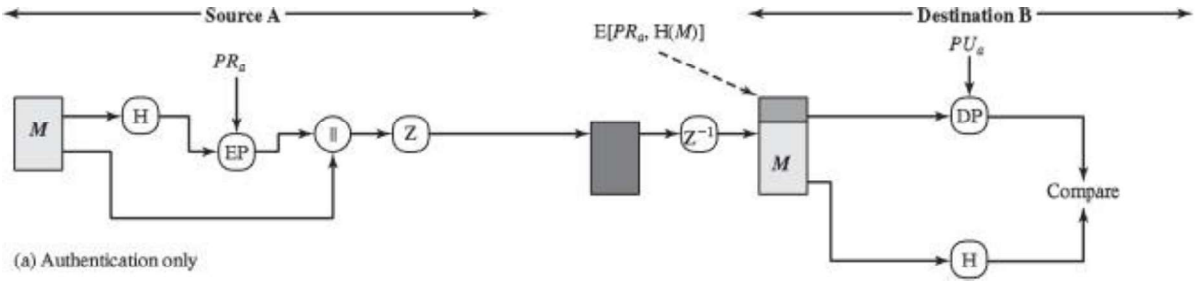
1. It is available free worldwide in versions that run on a variety of platforms.
2. It is based on algorithms that are extremely secure.
3. It has a wide range of applicability.
4. It was not developed by, nor is it controlled by, any governmental or standards organization
5. PGP is now on an Internet standards track.

Operational Description

(i) Authentication
(ii) Confidentiality
(iii) Compression
(iv) E-Mail compatibility
(v) Segmentation

(i) Authentication

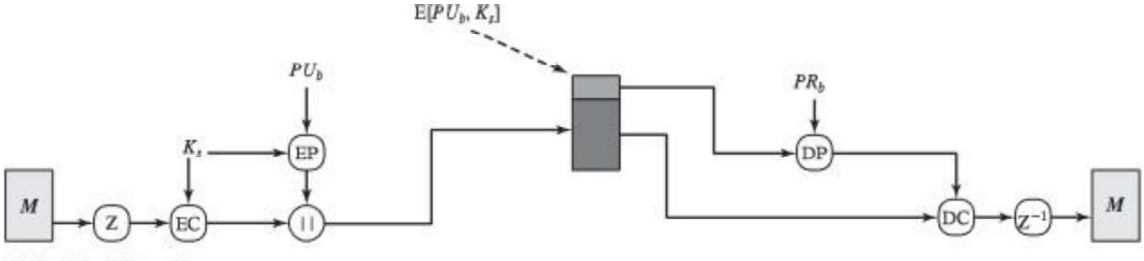
- The sender creates a message.
- SHA-1 is used to generate a 160-bit hash code of the message.
- The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.
- The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
- The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.



(a) Authentication only

(ii) Confidentiality

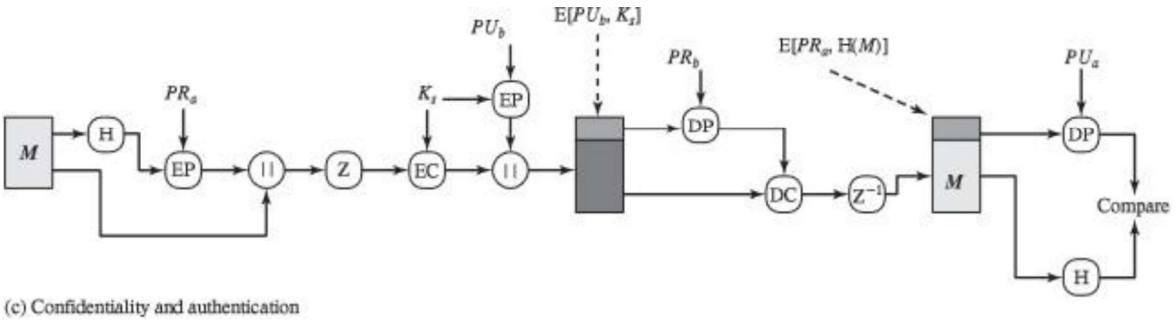
- The sender generates a message and a random 128-bit number to be used as a session key for this message only.
- The message is encrypted, using CAST-128 (or IDEA or 3DES) with the session key.
- The session key is encrypted with RSA, using the recipient's public key, and is prepended to the message.
- The receiver uses RSA with its private key to decrypt and recover the session key.
- The session key is used to decrypt the message.



(b) Confidentiality only

(iii) Confidentiality and Authentication

When both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and then encrypts the session key with the recipient's public key.



(c) Confidentiality and authentication

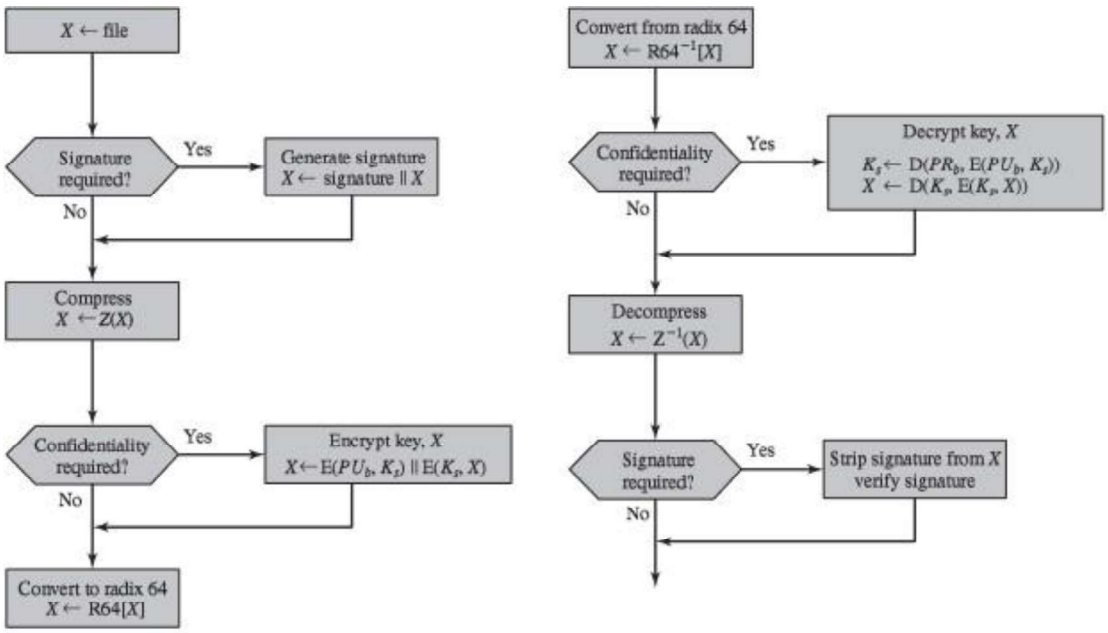
(iv) Compression

PGP compresses the message after applying the signature but before encryption. The compression algorithm is indicated by Z for compression and Z^{-1} for decompression. The signature is generated before compression for two reasons:

a) It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.

b) Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic.

c) Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.



(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

Transmission and Reception of PGP Messages

(v) E-mail Compatibility

When PGP is used, at least part of the block to be transmitted is encrypted. PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII. For this PGP using radix-654 conversion. Each group of 3 octets of binary data is mapped in to 4 ASCII characters. The uses of radix 64 expands a message by 33%

(vi) Segmentation and reassembly

Any message longer than that must be broken up into smaller segments, each of which is mailed separately. At the receiving end, the PGP must strip off all e-mail header and retrieve the essential block.

The header is separated from the body by a blank line, a header line consists of a keyword, followed by a colon followed by keyword's arguments.

Date:
From:
To:
Subject:

S/MIME

S/MIME is a security enhancement to MIME. S/MIME will emerge as the industry standard for commercial and organizational use.

To understand the S/MIME, we need first to have a general understanding of the e-mail format RFC822.

RFC822

RFC 822 defines a format for text messages that are sent using e-mail. InRFC 822 messages are said to have an envelope and contents.

Envelop: Information needed for transmission and delivery is present.

Content: It contains the object to be delivered to the receiver.

Each line in the header consists of a keyword such as ***From, To, Subject, Date*.** The following are the limitations of SMTP/RFC 82 scheme.

- SMTP cannot transmit executable or other binary data.
- SMTP cannot transmit text data that includes natural language characters.
- SMTP server may reject mail message over a certain type

Overview of MIME

1. Five new message header fields are defined, which may be included in an RFC 822 header.
2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.
3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

The five header fields defined in MIME are as follows

MIME-Version: This field must have a parameters value of 1.

Content-Type: This deals with the definition of variety of content types. In general content type specifies the type of data.

Content-Transfer-Encoding: Indicates the type of transformation that has been used to represent the body of the message.

Content-ID: Used to identify MIME entities.

Content-Description: A text description of the object within the body.

Content-Type

There are seven different major types of content and total of 15 subtypes.

Type	Subtype	Description
Text	Plain	Unformatted text;
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.

	Parallel	The multiple parts can be presented in parallel. For example, picture or text part could be accompanied by voice message.
	Alternative	The different representation of the same information.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/ rfc822.
Message	rfc822	The body is itself an encapsulated message.
	Partial	Used to allow fragmentation of large mail items.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN.
Application	PostScript	Adobe Postscript.
	octet-stream	General binary data consisting of 8-bit bytes.

MIME Transfer Encodings

The objective is to provide reliable delivery across a largest range of environments.

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters.
binary	The lines are not necessarily short enough and non-ASCII characters.
quoted-printable	Data being encoded are mostly ASCII text.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

S/MIME functionality

S/MIME provides the following functions

Enveloped Data: This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.

Signed Data: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base 64 encoding.

Clear-signed Data: The digital signature is encoded using base64. As a result recipients without S/MIME capability can view the message content, although they cannot verify the signature.

Signed and enveloped data: Signed only and encrypted only entities may be nested, so that encrypted data may be signed.

S/MIME messages

S/MIME secures a MIME entity with a signature, encryption or both.

Type	**Subtype**	**smime Parameter**	**Description**
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs 7-mime	signedData	A signed S/MIME entity.
	pkcs 7-mime	envelopedData	An encrypted S/MIME entity
	pkcs 7-mime	degenerate ignedData	s An entity containing only public- key certificates.
	pkcs 7-mime	CompressedData	A compressed S/MIME
	pkcs 7- signature	signedData	The content type of the signature subpart of a multipart/signed message.

Content Type of S/MIME

Enveloped Data

The steps for preparing an enveloped Data are as follows

1. Generate a pseudo-random session key for a particular symmetric encryption algorithm.
2. For each recipient, encrypt the session key with the recipient's public RSA key.
3. For each recipient, prepare a block known as RecipientInfo that contains the sender's public-key certificate, an identifier for the algorithm used to encrypt the session key, and the encrypted session key.
4. Encrypt the message content with the session key.

SignedData

The steps for preparing a signedData MIME entity are as follows:

1. Select a message digest algorithm.
2. Compute the message digest, or hash function, of the content to be signed.
3. Encrypt the message digest with the signer's private key.
4. Prepare a block known as SignerInfo that contains the signer's public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest.

S/MIME Certificate Processing

S/MIME user has several key-management functions to perform:

Key generation: Must be capable of generating separate Diffie-Hellman and DSS key pairs and should be capable of generating RSA key pairs. Each key pair must be generated from a good source of nondeterministic random input and be protected in a secure fashion.

Registration: A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate.

Certificate storage and retrieval: The list of certificates could be maintained by the user or by some local administrative entity on behalf of a number of users.

VeriSign Certificates

VeriSign provides a service that is intended to be compatible with S/MIME and a variety of other applications. VeriSign issues X.509 certificates with the product name VeriSign Digital ID. Each digital ID contains

- Owner's public key
- Owner's name or alias
- Expiration date of the Digital ID
- Serial number of the Digital ID
- Name of the certification authority that issued the Digital ID

Enhanced Security Services

Signed Recepits
Security Labels
Secure Mailing Lists

IP Sec

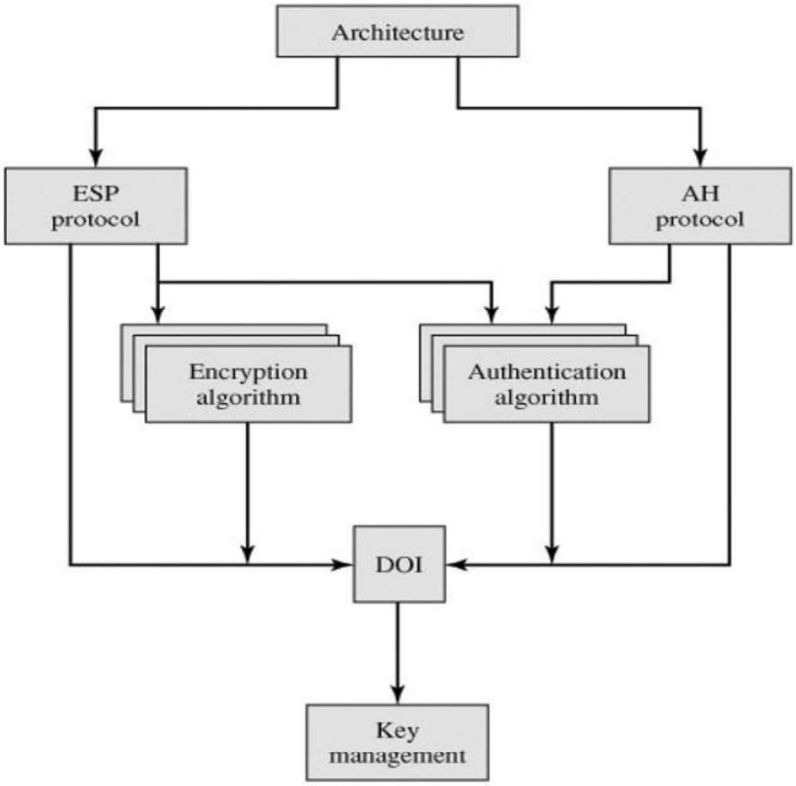
IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples are

- Secure branch office connectivity over the Internet
- Secure remote access over the Internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security

IP Security Architecture

The IPSec specification consists of numerous documents. The most important of these, issued in November of 1998, are

- RFC 2401: An overview of a security architecture
- RFC 2402: Description of a packet authentication extension to IPv4 and IPv6
- RFC 2406: Description of a packet encryption extension to IPv4 and IPv6
- RFC 2408: Specification of key management capabilities



IPSec Document Overview

Architecture: Covers the general concepts, security requirements, definitions, and mechanisms defining IPSec technology.
Encapsulating Security Payload (ESP): Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.
Authentication Header (AH): Covers the packet format and general issues related to the use of AH for packet authentication.
Encryption Algorithm: A set of documents that describe how various encryption algorithms are used for ESP.
Authentication Algorithm: A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.
Key Management: Documents that describe key management schemes.
Domain of Interpretation (DOI): This document contains values needed for other documents to relate to each other.

IPSec Services

IPSec provides security services at the IP layer by enabling a system to select required security protocols. Two protocols are used to provide security:

- Authentication protocol
- Encryption/authentication protocol (ESP). The services are

The services are

- Access control
- Connectionless integrity

- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Security Associations

An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If security exchange is needed in both directions then two-way security association is needed. A security association is uniquely identified by three parameters:

Security Parameters Index (SPI): A bit string assigned to this SA.

IP Destination Address: Only unicast addresses are allowed.

Security Protocol Identifier: This indicates whether the association is an AH or ESP security association

SA Parameters

A security association is normally defined by the following parameters:

Sequence Number Counter: A 32-bit value used to generate the Sequence Number field in AH or ESP headers.

Sequence Counter Overflow: A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA.

Anti-Replay Window: Used to determine whether an inbound AH or ESP packet is a replay or not.

AH Information: Specifies an authentication related parameters like authentication algorithm, authentication key, and key lifetimes.

ESP Information: Specifies the encryption and authentication algorithm, keys, initialization values, key lifetimes

Lifetime of this SA: This is the time interval after which SA must be replaced with a new SA.

IPSec Protocol Mode: This parameter specifies the mode of transfer.

Path MTU: Specifies the maximum transmission unit.

SA Selectors

IPSec provides the user with flexibility in the way in which IPSec services are applied to IP traffic. The means by which IP traffic is related to specific SAs is the nominal Security Policy Database (SPD). SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic. Each SPD entry is defined by a set of IP and upper-layer protocol field values, called selectors. These selectors are used to filter outgoing traffic in order to map it into a particular SA.

The following selectors determine an SPD entry:

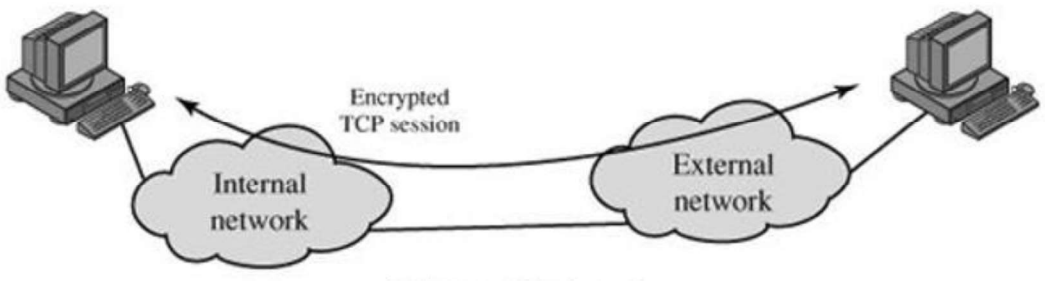
- Destination IP Address
- Source IP Address
- UserID
- Data Sensitivity Level
- Transport Layer Protocol
- Source and Destination Ports

Modes of Transfer

- Transport Mode
- Tunnel Mode

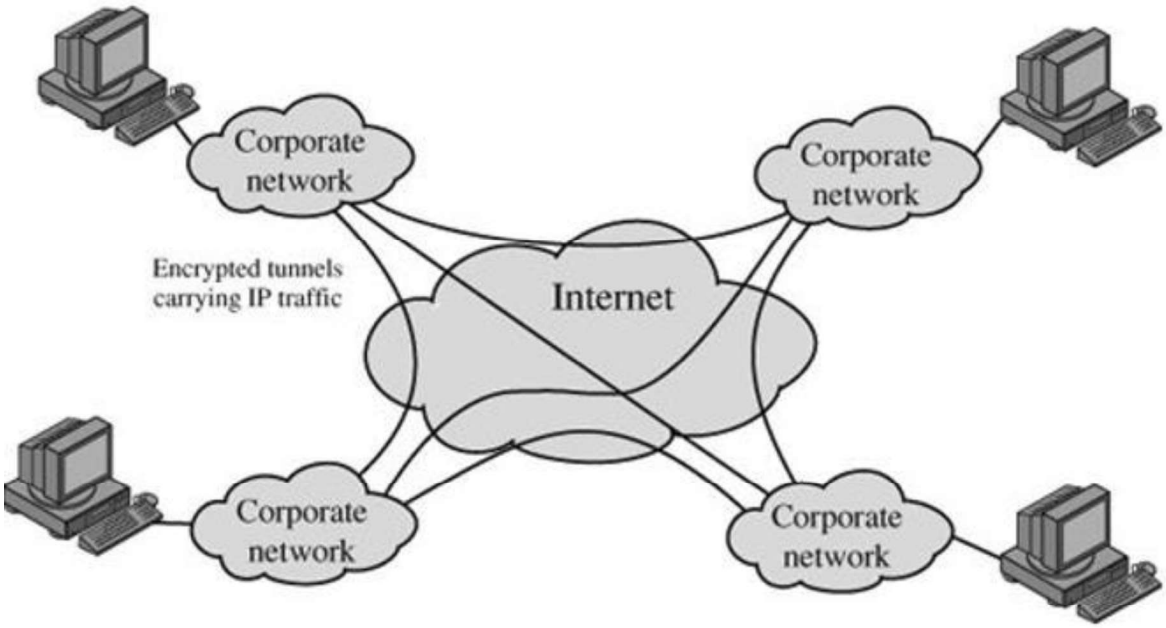
Transport Mode

Transport mode provides protection primarily for upper-layer protocols. The transport mode protection extends to the payload of an IP packet. Transport mode is used for end to end connections.



Tunnel Mode

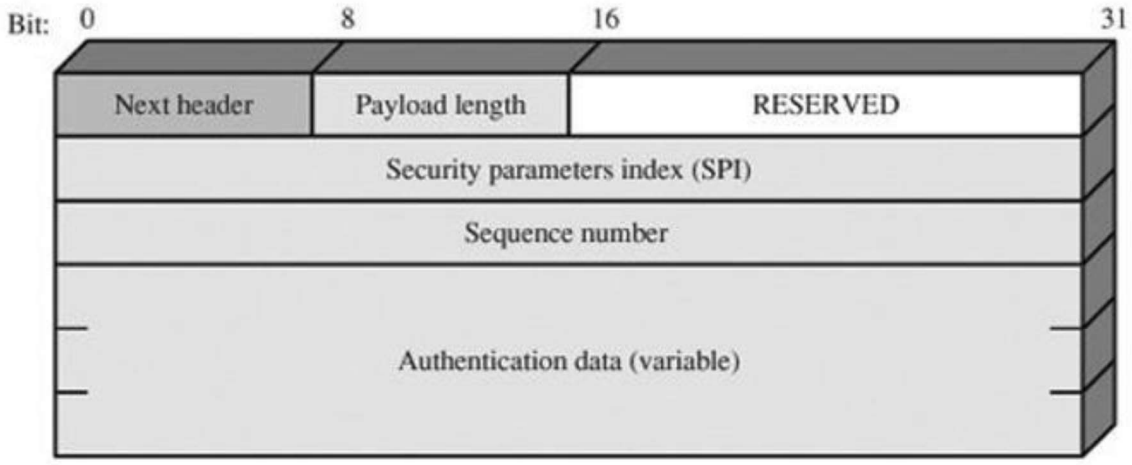
Tunnel mode provides protection to the entire IP packet. Tunnel mode authenticates the entire inner IP and selected portion of outer IP header, IP V6 extension header.



(b) A virtual private network via tunnel mode

AUTHENTICATION HEADER

The Authentication Header provides data integrity and authentication of IP packets. The data integrity assures that modification during transit is not possible. The authentication enables the system to authenticate the user and prevents the address spoofing attacks.



The Authentication Header consists of the following fields

Next Header (8 bits): Identifies the type of header immediately following this header.
Payload Length (8 bits): Length of Authentication Header in 32-bit words, minus 2.
Reserved (16 bits): For future use.
Security Parameters Index (32 bits): Identifies a security association.
Sequence Number (32 bits): A monotonically increasing counter value.
Authentication Data (variable): A variable-length which contains the Integrity Check Value.

Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The Sequence Number field is designed to overcome such attacks.

Generation of sequence number by the sender

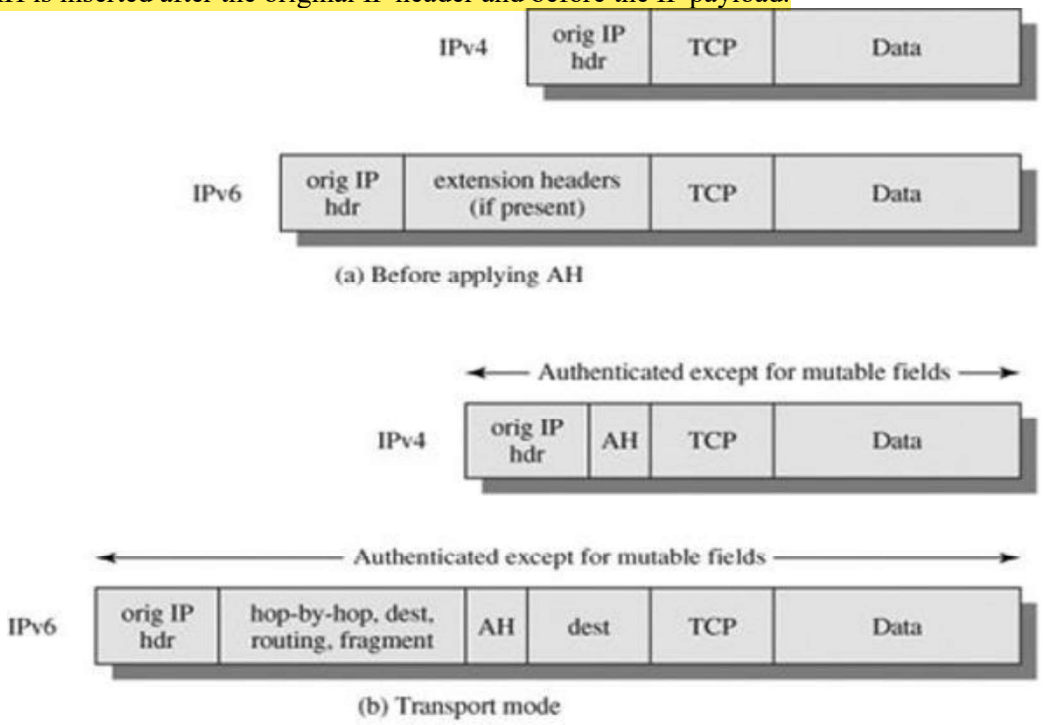
When a new SA is established, the sender initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled the sender must not allow the sequence number field to come back to 0, after cycle past 2^{32} -1. If the limit of 2^{32}-1 is reached, the sender should terminate this SA and negotiate a new SA with a new key.

For any incoming packet, the processing

1. If the received packet falls within the window and is new, the MAC is generated.
2. If the packet is authenticated, the corresponding slot in the window is marked as received.
3. If the received packet is to the left of the window, or if authentication fails, the packet is discarded.

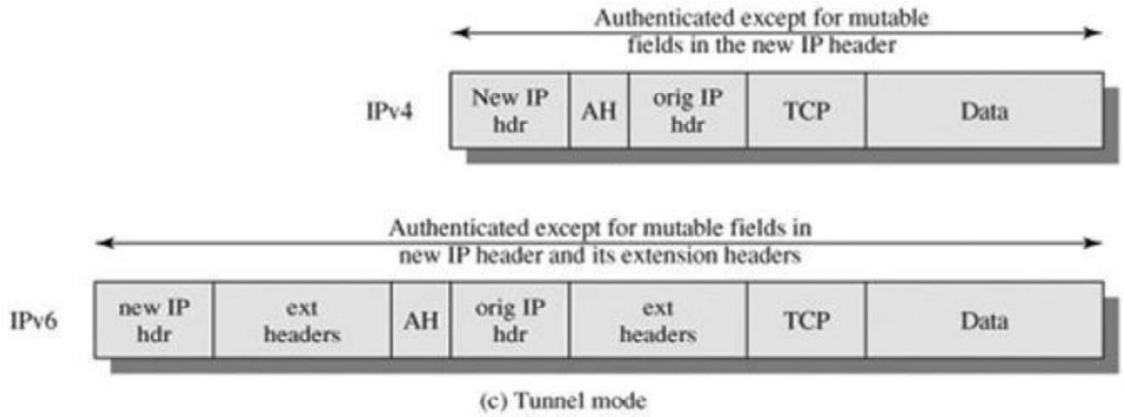
Transport mode AH

The AH is inserted after the original IP header and before the IP payload.



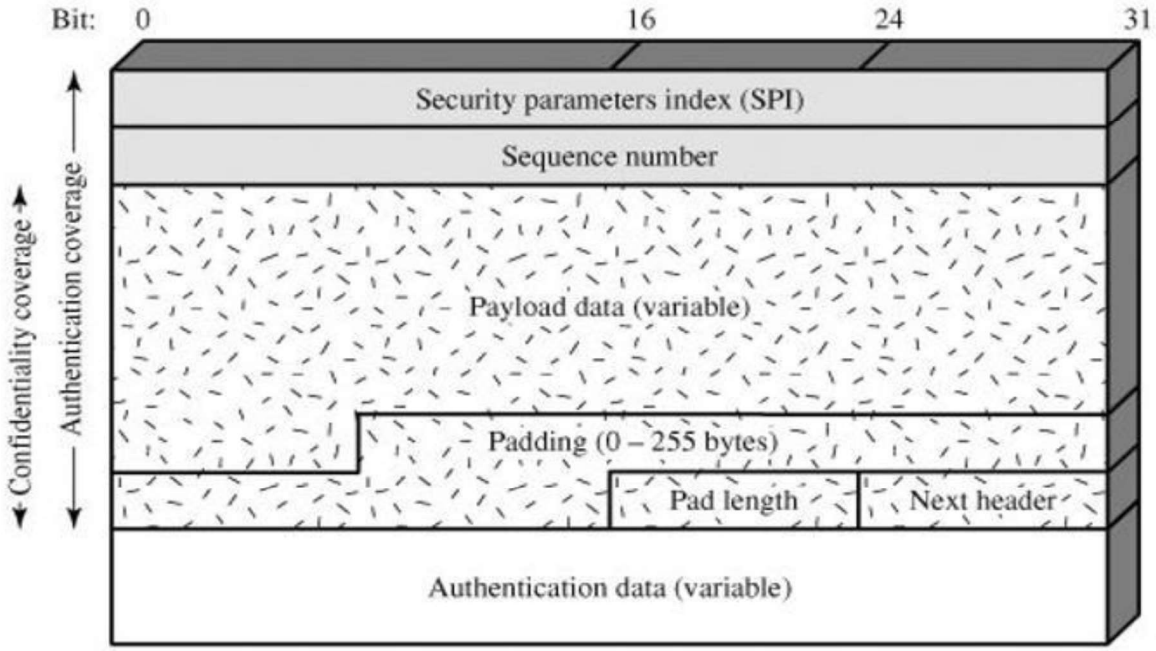
Tunnel Mode of AH

The entire original IP packet is authenticated and the AH is inserted between the orginal IP header and new IP header.



ENCAPSULATING SECURITY PAYLOAD

The Encapsulating Security Payload provides confidentiality of message contents and limited for traffic flow confidentiality.



Security Parameters Index (32 bits): Identifies a security association.
Sequence Number (32 bits): A monotonically increasing counter value; this provides an anti-replay function.
Payload Data (variable): This is a transport-level segment or IP packet 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.
Authentication Data (variable): A variable-length field that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

Encryption and Authentication Algorithms

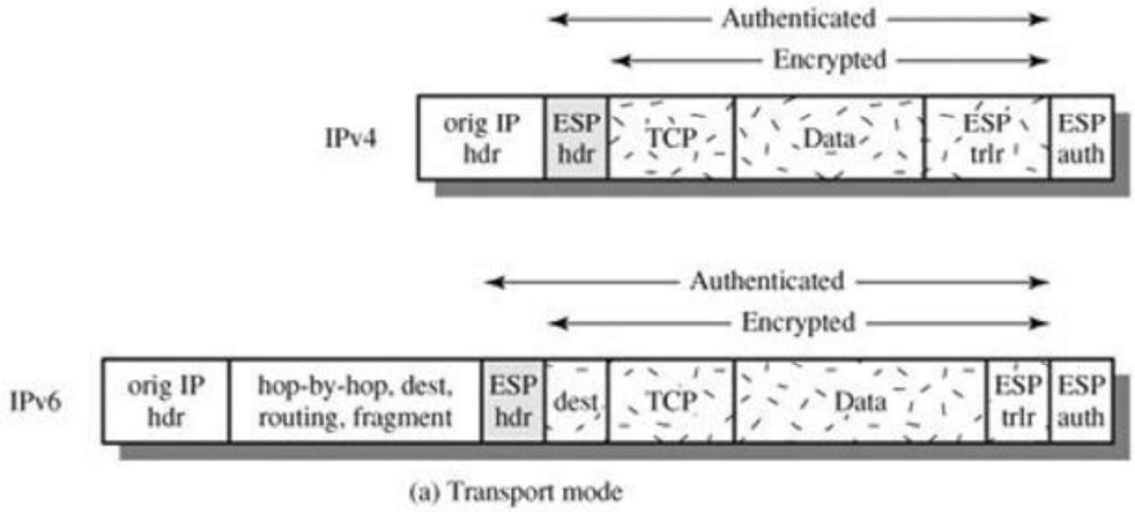
The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. The algorithms used for encryption are

- Triple DES
- RC5
- IDEA
- Three-key triple IDEA
- CAST
- Blowfish

The uses of Padding are

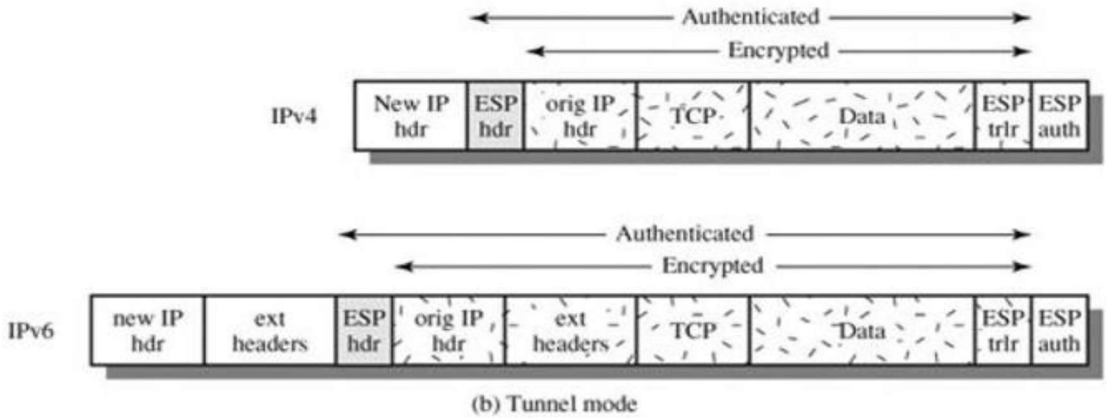
(i) If an encryption algorithm needs the plaintext to be a multiple of some number of bytes, the Padding is needed.
(ii) The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32bit word.
(iii) The padding conceals the actual length of payload thus brings confidentiality.

Transport Mode ESP
Transport mode ESP is used for encryption and authentication of data.



(a) Transport mode

Tunnel Mode ESP
Tunnel mode ESP is used to encrypt an entire IP packet. The tunnel mode is useful for firewall or security gateway that protects the trusted network from external networks.



(b) Tunnel mode

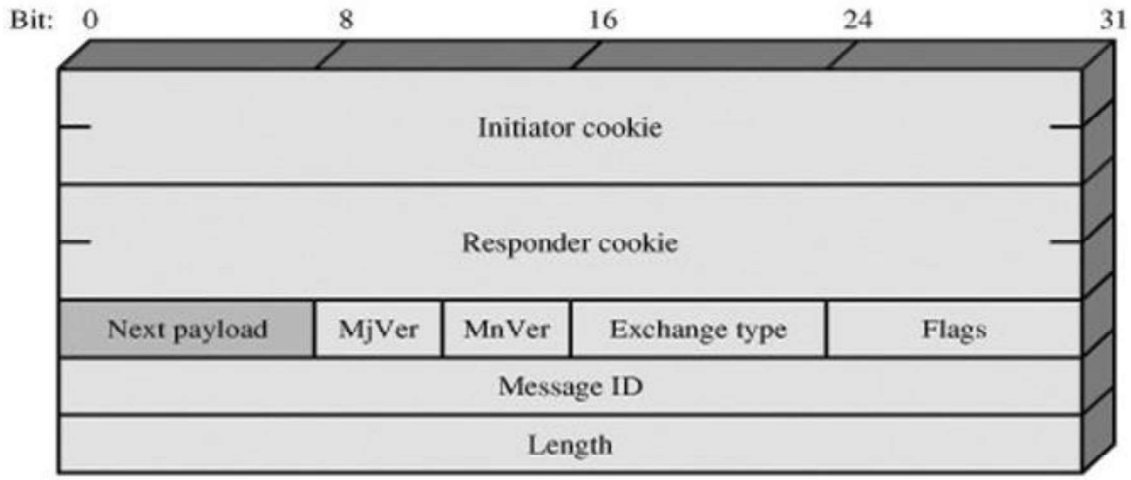
KEY MANAGEMENT

The key management portion of IPSec involves the determination and distribution of secret keys. The IPSec support two types of key management:
Manual: A system administrator manually configures each system with its own keys and with the keys of other communicating systems.
Automated: This system supports on-demand creation of keys for SAs and supports use of large system. The automated key management protocol for IPsec is called ISAKMP/Oakly.

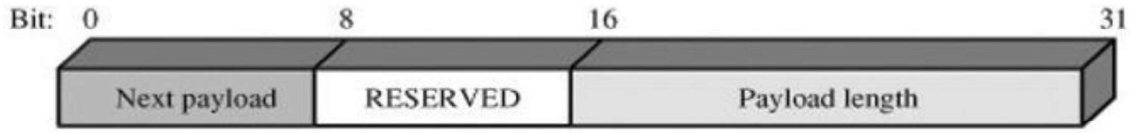
ISAKMP

ISAKMP defines procedures and formats to establish, negotiate, modify, and delete security associations. ISAKMP defines payload for exchanging key generation and authentication data.
Initiator Cookie (64 bits): Cookie of entity that initiated SA establishment, SA notification, or SA deletion.
Responder Cookie (64 bits): Cookie of responding entity; null in first message from initiator.
Next Payload (8 bits): Indicates the type of the first payload in the message.
Major Version (4 bits): Indicates major version of ISAKMP in use.
Minor Version (4 bits): Indicates minor version in use.
Exchange Type (8 bits): Indicates the type of exchange.
Flags (8 bits): Indicates specific options set for this ISAKMP exchange.

Message ID (32 bits): Unique ID for this message.
Length (32 bits): Length of total message (header plus all payloads) in octets.



(a) ISAKMP header



(b) Generic payload header

ISAKMP Payload Types

Type	Description
Security Association (SA)	Used to negotiate security attributes and indicate the DOI and Situation under which negotiation is taking place.
Proposal (P)	Used during SA negotiation; indicates protocol to be used and number of transforms.
Transform (T)	Used during SA negotiation; indicates transform and related SA attributes.
Key Exchange (KE)	Supports a variety of key exchange techniques.
Identification (ID)	Used to exchange identification information
Certificate (CERT)	Used to transport certificates and other certificate- related information.
Certificate Request (CR)	Used to request certificates; indicates the types of certificates requested and the acceptable certificate authorities.
Hash (HASH)	Contains data generated by a hash function.
Signature (SIG)	Contains data generated by a digital signature function.
Nonce (NONCE)	Contains a nonce.
Notification (N)	Used to transmit notification data, such as an error condition.
Delete (D)	Indicates an SA that is no longer valid.

ISAKMP messages
Responder-Lifetime: Communicates the SA lifetime chosen by the responder.
Replay-Status: Used for positive confirmation of the responder's election of whether or not the responder will perform anti-replay detection.
Initial-Contact: Informs the other side that this is the first SA being established with the remote system.
The Delete payload: Indicates that the sender has deleted the SA from its database.

ISAKMP Exchanges

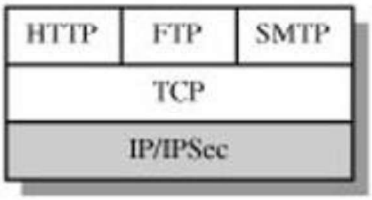
The Base Exchange: Allows key exchange and authentication to be transmitted together.
The Identity Protection Exchange: Expands the base to protect the users' identities.
The Authentication Only Exchange: Used to perform mutual authentication, without a key exchange.
The Aggressive Exchange: Provides identity protection.
The Informational Exchange: Used for one-way transmittal of information for SA management.

WEB SECURITY

The web security threats can be found in web security, web browser and the network traffic between the browser and server.

Web Traffic Security Approaches

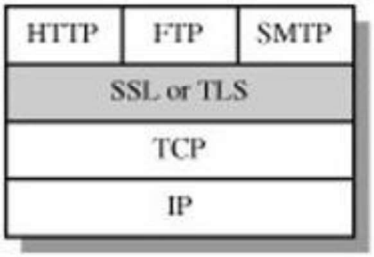
Approach1



(a) Network level

This approach is transparent to end users and provides a general purpose solution. IPSec data filtering, so that only selected traffic can incur IPSec processing.

Approach 2

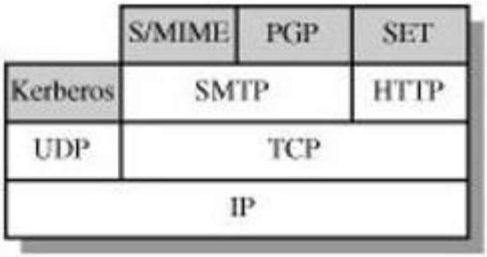


(b) Transport level

Implement the security above TCP. SSL can be embedded with SSL to Netscape Navigator, Microsoft explorer.

Approach 3

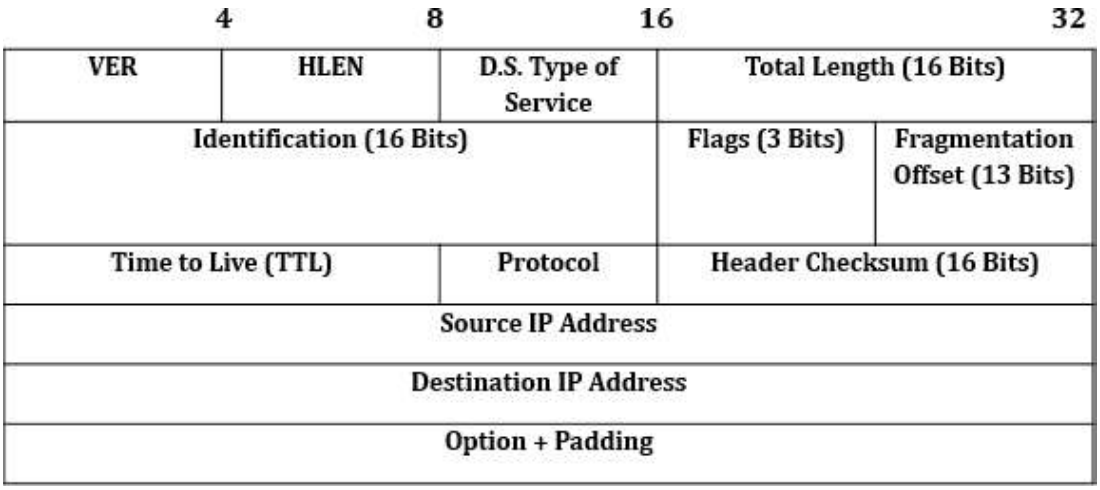
Security services are embedded in a particular application. SET is an example of this approach.



(c) Application level

IPV4

Internet protocol being a layer-3 protocol takes data segments from layer-4 and divides in to packets. IP packets encapsulates data unit received from above layer and add to its own header information.

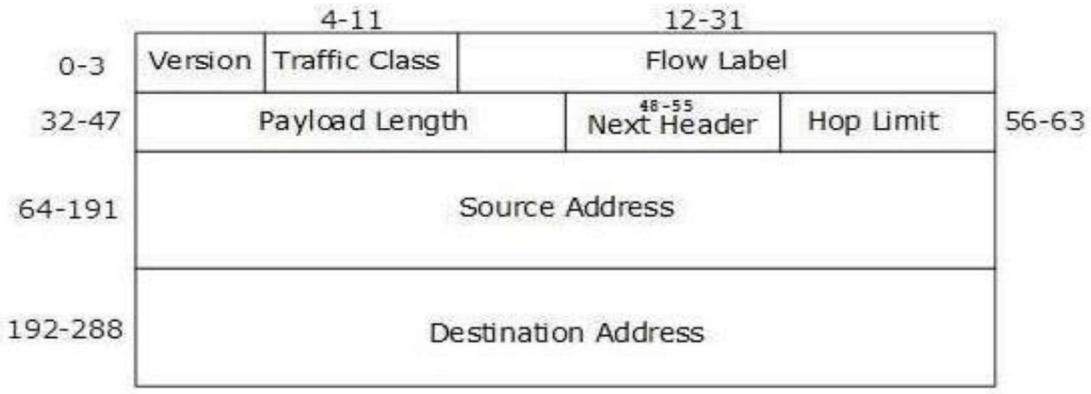


Version: This Field defines the version of IP. It is Static 4 bit value.
Header Length: This Field defines the length of the entire IP header. It is 4 bit value.
Type of Service: It is 8 bit value. It is used tell the network how to treat the IP packet. These bits are generally used to indicate the Quality of Service (QoS) for the IP Packet.
Packet Length: Length of the entire IP packet.
Identification: Used to identify the original IP packet.
Flags: It is 3 bits value. Flag tells the whether the IP packet is fragmented or not.
Fragment offset: 13 bit value tells the exact position of the fragment in the original IP packet.
Time to Live: 8 bit value. To avoid looping on the network every packet is sent with some TTL value set, which tell the network how many routers this packet can cross.
Protocol: 8 bit value used to indicate the type of protocol being used.
Header checksum: It is 16 bit value. It is used to indicate errors in the header only.
Source address: 32 bit value representing the IP address of the sender of the IP packet.
Destination address: 32 bit value representing the IP address of the packets final destination.
Options: Options are not required for every datagram. They are used for network testing and debugging.
Padding: Variable size bit field. These bits are used to ensure a 32 bit boundary for the header is achieved.

IPv6

IPV6 is the most recent version of the Internet Protocol. It is a communication protocol that provides an identification and location for computers on networks and routes traffic across the network. IPV6 is developed by IETF – Internet Engineering Task Force.

Version (4-bits): It represents the version of Internet Protocol.
Traffic Class (8-bits): The most significant 6 bits are used for Type of Service to let the Router Known. The least significant 2 bits are used for Explicit Congestion Notification (ECN).
Flow Label (20-bits): This label is used to maintain the sequential flow of the packets
Payload Length (16-bits): Indicates how much information a particular packet.
Next Header (8-bits): Used to indicate either the type of Extension Header.
Hop Limit (8-bits): This field is used to stop packet to loop in the network infinitely.
Source Address (128-bits): This field indicates the address of originator of the packet.
Destination Address (128-bits): This field provides the address of intended recipient of the packet.



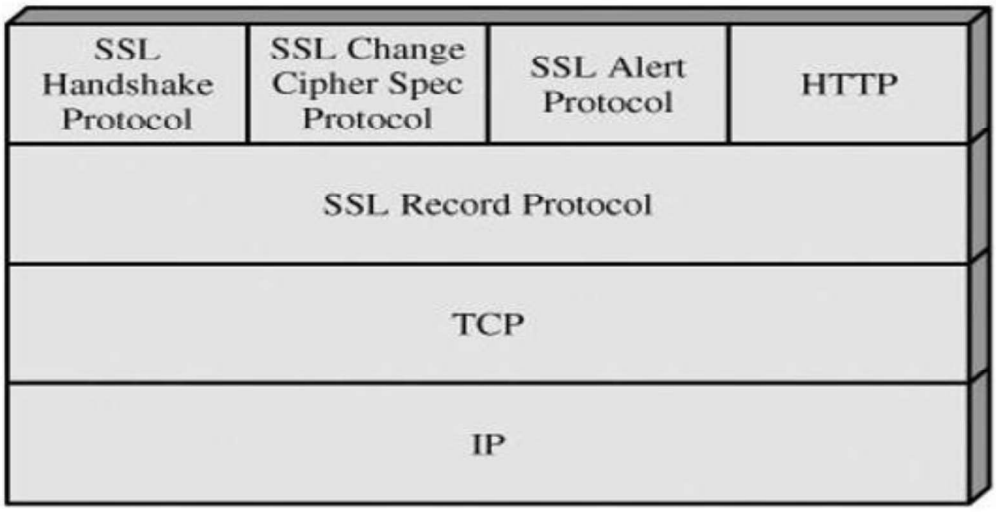
Extension Header

- Routing – Extended routing, such as IPv4 loose source route
- Fragmentation – Fragmentation and reassembly
- Authentication – Integrity and authentication, and security
- Encapsulating Security Payload – Confidentiality
- Hop-by-Hop options – Special options that require hop-by-hop processing
- Destination options – Optional information to be examined by the destination node

SECURE SOCKET LAYER

SSL was originated by Netscape.

SSL Architecture



The SSL Record Protocol provides basic security services to various higher-layer protocols. Three higher layer protocols are part of SSL viz

- Handshake Protocol,
- Change Cipher Spec Protocol,
- Alert Protocol

SSL concepts

Two important SSL concepts are

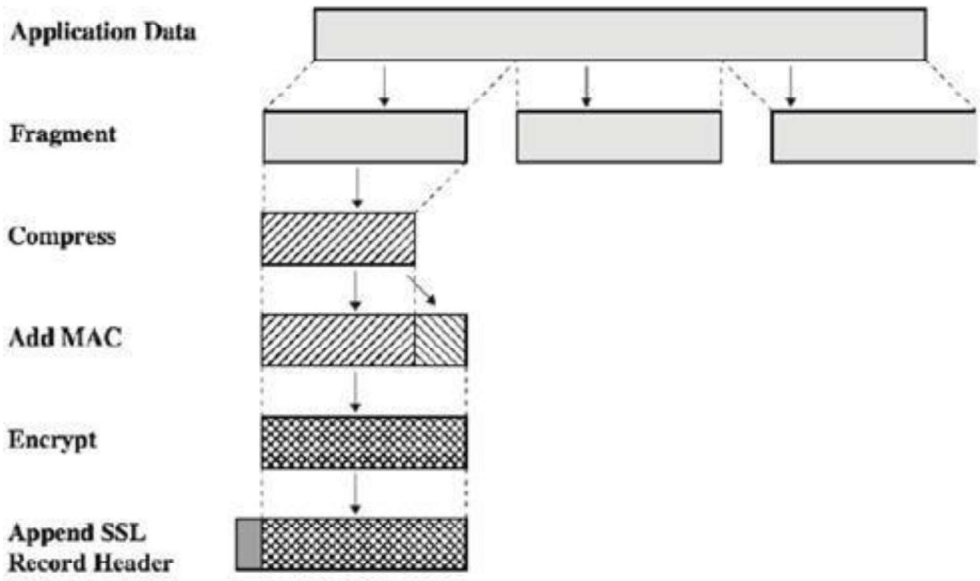
- SSL session
- SSL connection

Parameters of a session state
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.
Compression method: The algorithm used to compress data prior to encryption.
Cipher spec: defines cryptographic attributes such as the hash size.
Master secret: 48-byte secret shared between the client and server.
Isresumable: A flag indicating whether the session can be used to initiate new connections

Parameters that define the connection state

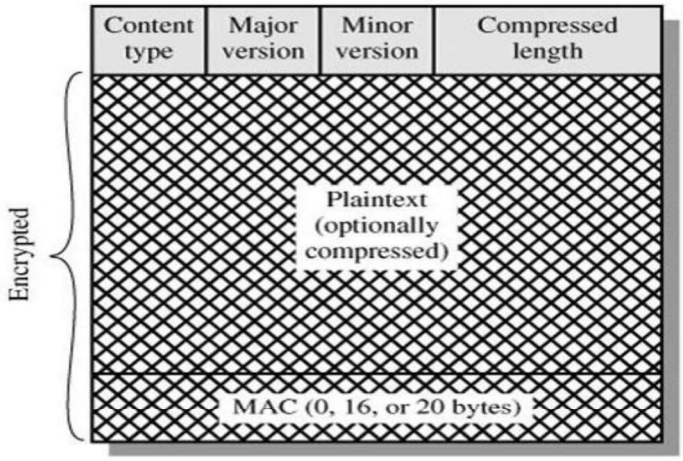
- Server and client random
- Server write MAC secret:
- Client write MAC secret
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

SSL protocol stack
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).
SSL record protocol operations



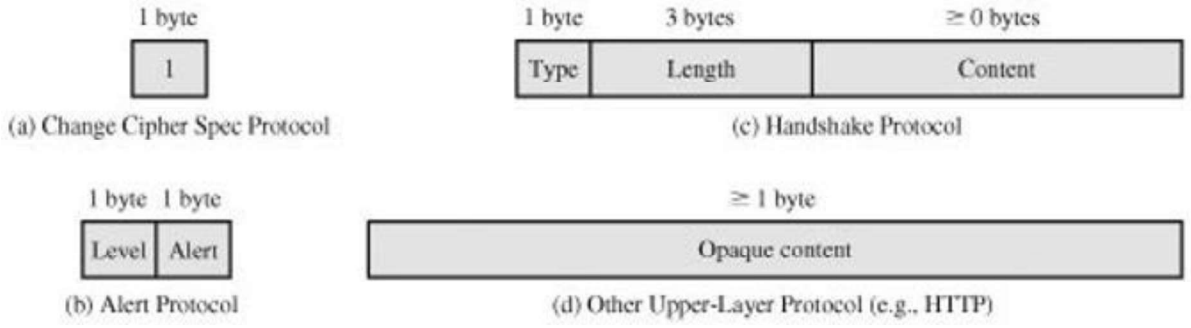
The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled and then delivered to higher-level users.

SSL record format
Content Type (8 bits): The higher layer protocol used to process the enclosed fragment.
Major Version (8 bits): Indicates major version of SSL in use which is 3.
Minor Version (8 bits): Indicates minor version in use which is 0.
Compressed Length (16 bits): The length in bytes of the plaintext fragment, maximum value is 2^{14} + 2048.



Change Cipher Spec Protocol

This protocol consists of a single message which consists of a single byte with the value 1. The sole purpose 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.



Alert Protocol

The Alert Protocol is used to convey SSL-related alerts to the peer entity. The SSL messages are compressed and encrypted, as specified by the current state. The alerts are

unexpected_message: An inappropriate message was received.

bad_record_mac: An incorrect MAC was received.

decompression_failure: The decompression function received improper input **handshake_failure**: Sender was unable to negotiate an acceptable set of security parameters given the options available.

Handshake Protocol

The Handshake Protocol consists of a series of messages exchanged by client and server

Phase 1: Establish Security Capabilities

This phase is used to initiate a logical connection and to establish the security capabilities that will be associated with it. The exchange is initiated by the client by sending the 'client hello' message. It contains the

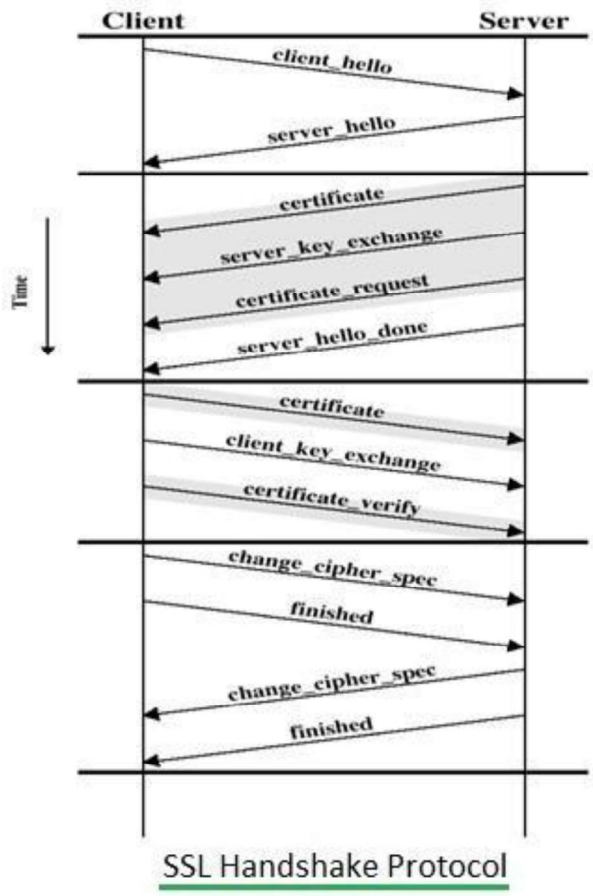
- Version
- Random
- Session id
- Cipher Suite
- Compression Method

The server also sends a 'server hello' message to the client.

Phase 2: Server Authentication and Key Exchange

- The server sends its certificate, if it needs to be authenticated.

- Server key exchange message is optional
- Server request a certificate from the client
- 'Server done' is sent by the server to indicate the end of server hello and associated messages. After this message the server will not wait for the client response.



SSL Handshake Protocol

Phase 3. Client Authentication and Key Exchange

The client should verify that the server provided a valid. If all are satisfied the client sends one or more messages back to the server.

Phase 4. Finish

The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec. The client then immediately sends the finished message under the new algorithms, keys, and secrets. The finished message verifies that the key exchange and authentication processes were successful.

In response to these two messages, the server sends its own change_cipher_spec message, transfers the pending to the current CipherSpec, and sends its finished message. Now, the handshake is complete and the client and server exchange data.

FIREWALL

A firewall is a device or group of devices that controls access between networks. A firewall generally consists of filters and gateway(s), varying from firewall to firewall. It is a security gateway that controls access between the public Internet and an intranet and is a secure computer system placed between a trusted network and an untrusted Internet.

A firewall is an agent that screens network traffic in some way, blocking traffic it believes to be inappropriate, dangerous, or both. Firewalls act as an intermediate server in handling SMTP and HTTP connections in either direction.

Roles of Firewalls

- The firewall imposes restrictions on packets entering or leaving the private network.
- All traffic from inside to outside, and vice versa, must pass through the firewall, but only authorized traffic will be allowed to pass.
- Packets are not allowed through unless they conform to a filtering specification or unless there is negotiation involving some sort of authentication.
- The firewall itself must be immune to penetration.
- Firewalls create checkpoints between an internal private network and an untrusted Internet. Once the choke points established, the device can monitor, filter, and verify all inbound and outbound traffic.
- The firewall may filter on the basis of IP source and destination addresses and TCP port number.
- The firewall also enforces logging and provides alarm capacities as well.
- The firewall provides protection from various kinds of IP spoofing and routing attacks.
- A firewall can limit network exposure by hiding the internal network systems and information from the public Internet.

Four general techniques that firewalls use to control access and enforce the site's security policy

1. **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 and TCP port number;
2. **Direction control:** Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall.
3. **User control:** Controls access to a service according to which user is attempting to access it. This feature is typically applied to users inside the firewall perimeter (local users).
4. **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.

FIREWALL-RELATED TERMINOLOGY

1. Bastion Host

A bastion host is a publicly accessible device for the network's security, which has a direct connection to a public network such as the Internet. Bastion hosts must check all incoming and outgoing traffic and enforce the rules specified in the security policy.

- **Single-homed bastion host:** This is a device with only one network interface, normally used for an application-level gateway. The external router is configured to send all incoming data to the bastion host, and all internal clients are configured to send all outgoing data to the host.
- **Dual-homed bastion host:** This is a firewall device with at least two network interfaces. The advantage of using such hosts is that they create a complete break between the external network and the internal network.
- **Multi-homed bastion host:** Used to allow the user to enforce strict security mechanisms. When the security policy requires all inbound and outbound traffic to be sent through a proxy server, a new proxy server should be created for the new streaming application.

2. Proxy Server

Proxy servers are used to communicate with external servers on behalf of internal clients. Application proxies forward packets only when a connection has been established using some known protocol. When the connection closes, a firewall using application proxies rejects individual packets. Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection. Each proxy is independent of other proxies on the bastion host. If there is a problem with the operation of any proxy, or if future vulnerability is discovered, it is easy to replace the proxy without affecting the operation of the proxy's applications.

3. SOCKS

The SOCKS protocol version 4 provides for unsecured firewall traversal for TCP-based client/server applications, including HTTP, TELNET, and FTP. SOCKS define how to establish authenticated connections, but currently, it does not provide a clear-cut solution to the problem of encrypting the data traffic.

4. Choke Point

A choke point is the point at which a public Internet can access the internal network. The most comprehensive and extensive monitoring tools should be configured on the choke points. All traffic is flowing through the firewalls, security administrators, as a firewall strategy, need to create choke points to limit external access to their networks. Once these choke points have been clearly established, the firewall devices can monitor, filter, and verify all inbound and outbound traffic.

5. Demilitarized Zone (DMZ)

In terms of a firewall, the DMZ is a network that lies between an internal private network and the external public network. DMZ networks are sometimes called *perimeter networks*. A DMZ is used as an additional buffer to further separate the public network from the internal network.

6. Logging and Alarms

Logging is usually implemented at every device in the firewall. Since a choke point is installed at the firewall, a prospective hacker will go through the choke point. If so, the comprehensive logging devices will probably capture all hacker activities, including all user activities as well. The user can then tell exactly what a hacker is doing and have such information available for audit. The audit log is an essential tool for detecting and terminating intruder attacks.

7. VPN

VPNs are appropriate for any organization requiring secure external access to internal resources. All data transmitted over a VPN is usually encrypted because an opponent with access to the Internet could eavesdrop on the data as it travels over the public network. The VPN encapsulates all the encrypted data within an IP packet. Authentication, message integrity, and encryption are very important fundamentals for implementing a VPN.

TYPES OF FIREWALLS

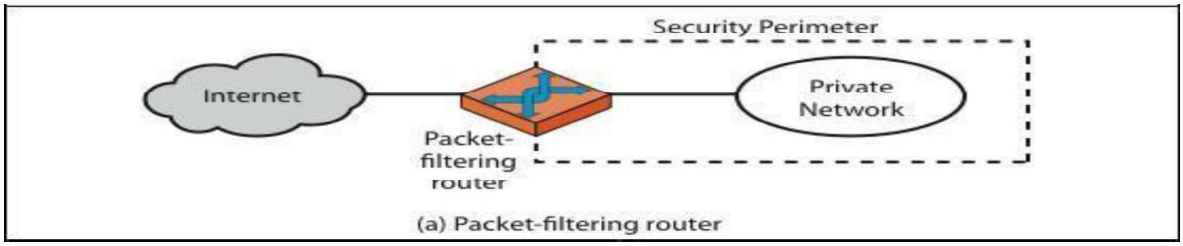
Firewalls are classified into three common types:

1. **Packet filtering Router**
2. **Circuit-level gateways**
3. **Application-level gateways**

1. Packet filtering Router

A packet filtering router applies a set of rules to each incoming IP packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions. Filtering rules are based on the information contained in a network packet:

Source IP address – IP address of the system that originated the IP packet.
Destination IP address – IP address of the system, the IP is trying to reach.
Source and destination transport level address – transport level port number.
Interface – for a router with three or more ports, which interface of the router the packet come from or which interface of the router the packet is destined for.



The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there is no match to any rule, then a default action is taken.
Two default policies are possible:

Default = discard: That which is not expressly permitted is prohibited.
Default = forward: That which is not expressly prohibited is permitted.

TELNET packet filtering

TELNET is a simple remote terminal access that allows a user to log onto a computer across an Internet. TELNET establishes a TCP connection and then passes keystrokes from the user's keyboard directly to the remote computer. TELNET also carries output from the remote machine back to the user's screen.

Rule number	Action	Source IP	Source port	Destination IP	Destination port	Protocol
1	Discard	*	23	*	*	TCP
2	Discard	*	*	*	23	TCP

If a packet is passed through the filter and has a source port of 23, it will immediately be discarded. If a packet with a destination port of 23 is passed through this filter, it is discarded only after rule 2 has been applied. All other packets will be discarded.

FTP packet filtering

If the FTP service is to apply the same basic rule as applied to TELNET, the packet filter to allow or block FTP would look like following Table. The FTP service is typically associated with using TCP ports 20 and 21.

Rule number	Action	Source IP	Source port	Destination IP	Destination port	Protocol
1	Allow	192.168.10.0	*	*	21	TCP
2	Block	*	20	192.168.10.0	< 1024	TCP
3	Allow	*	20	192.168.10.0	*	TCP ACK = 1

Advantages of packet filter router

- Simple
- Transparent to users
- Very fast

Weakness of packet filter firewalls

- Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application specific vulnerabilities or functions.
- Because of the limited information available to the firewall, the logging functionality present in packet filter firewall is limited.
- It does not support advanced user authentication schemes.
- They are generally vulnerable to attacks such as layer address spoofing.

Attacks

IP address spoofing – The intruders transmit packets from the outside with a source IP address field containing an address of an internal host. To overcome this attack, discard the packet with an inside source address if the packet arrives on an external interface.

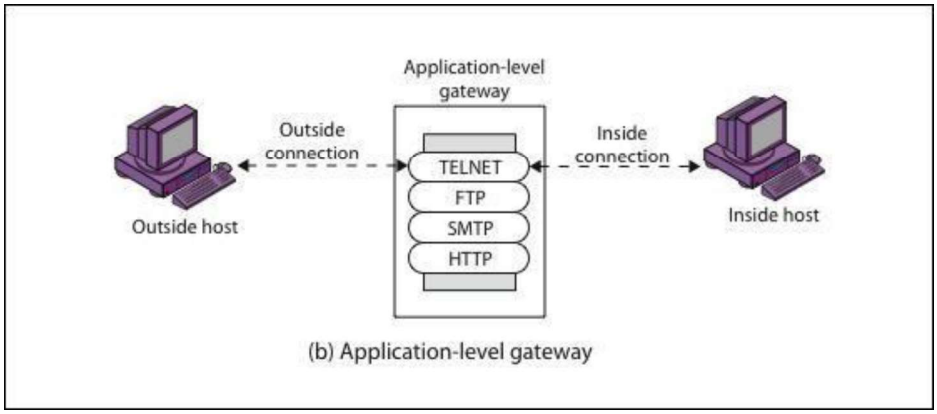
Source routing attacks – The source station specifies the route that a packet should take as it crosses the internet; i.e., it will bypass the firewall. The countermeasure is to discard all packets that uses this option.

Tiny fragment attacks – The intruder create extremely small fragments and force the TCP header information into a separate packet fragment. The attacker hopes that only the first fragment is examined and the remaining fragments are passed through. The countermeasure is to discard all packets where the protocol type is TCP and the IP fragment offset is equal to 1.

2. Application level gateway

An Application level gateway, also called a proxy server, acts as a relay of application level traffic. 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.

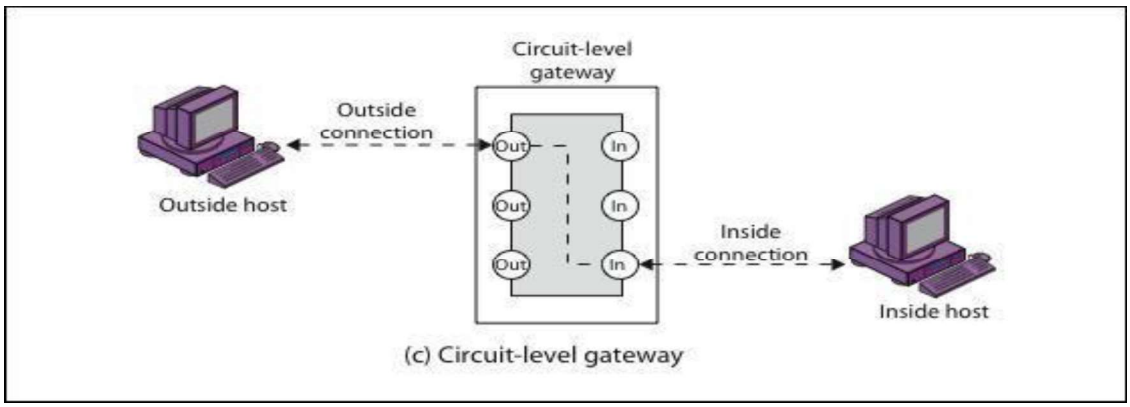
Application level gateways tend to be more secure than packet filters. It is easy to log and audit all incoming traffic at the application level. A prime disadvantage is the additional processing overhead on each connection.



(b) Application-level gateway

3. Circuit-Level Gateway

Circuit level gateway can be a stand-alone system or it can be a specified function performed by an application level gateway for certain applications. 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 outer 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.



(c) Circuit-level gateway

FIREWALL DESIGNS

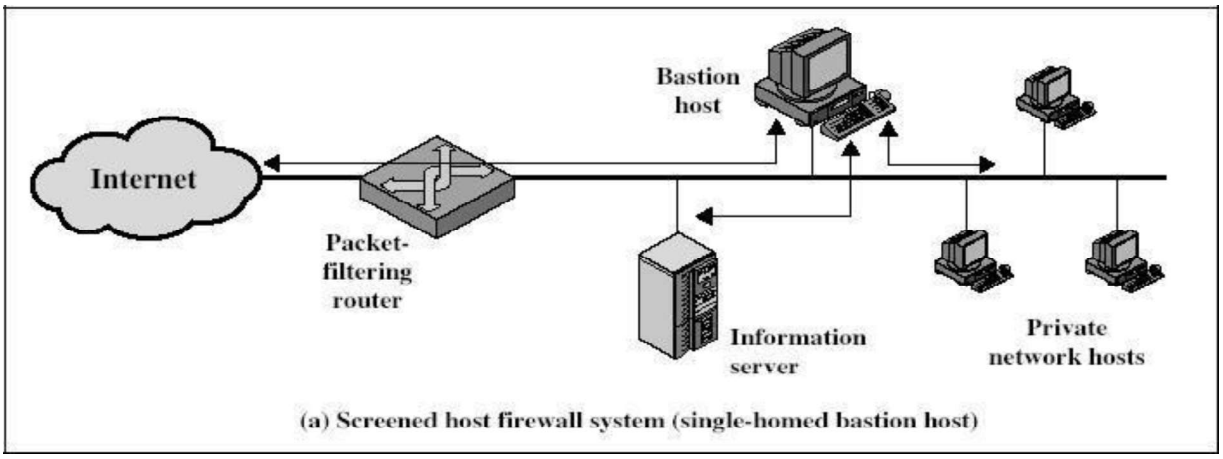
The primary step in designing a secure firewall is obviously to prevent the firewall devices from being compromised by threats. To provide a certain level of security, the three basic firewall designs are considered:

1. **Single-homed bastion host**
2. **Dual-homed bastion host**
3. **Screened subnet firewall**

1. Screened Host Firewall (Single-Homed Bastion Host)

The first type of firewall is a screened host which uses a single-homed bastion host plus a packet-filtering router. Single-homed bastion hosts can be configured as either circuit-level or application-level gateways. When using either of these two gateways, each of which is called

a *proxy server*, the bastion host can hide the configuration of the internal network.



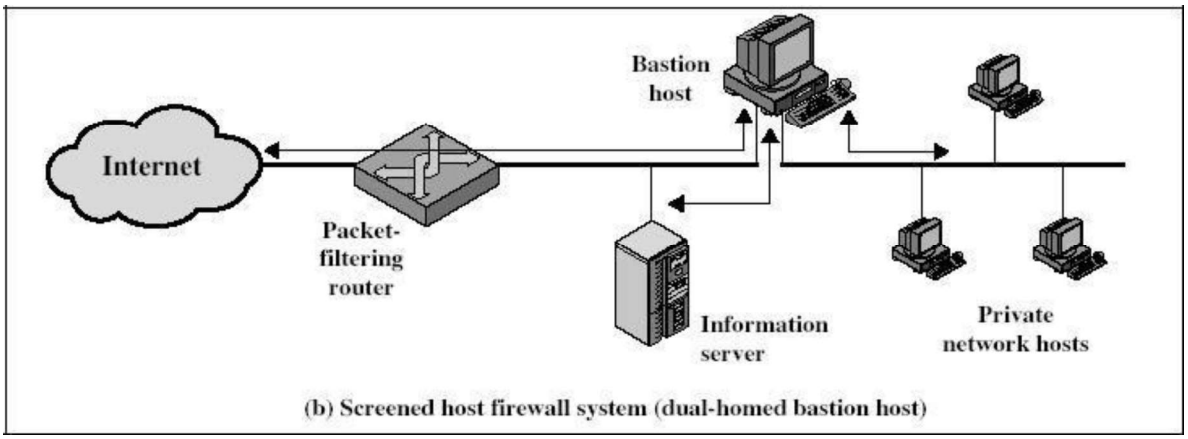
(a) Screened host firewall system (single-homed bastion host)

In this configuration, the firewall consists of two systems: a packet filtering router and a bastion host. Typically, the router is configured so that

- For traffic from the internet, only IP packets destined for the bastion host are allowed in. For traffic
- from the internal network, only IP packets from the bastion host are allowed out. The bastion host
- performs authentication and proxy functions. This configuration has greater security than simply a packet filtering router or an application level.

2. Dual-Homed Bastion Host

The configuration of the screened host firewall using a dual-homed bastion host adds significant security, compared with a single-homed bastion host. Dual-homed bastion host has two network interfaces. This firewall implementation is secure due to the fact that it creates a complete break between the internal network and the external Internet.



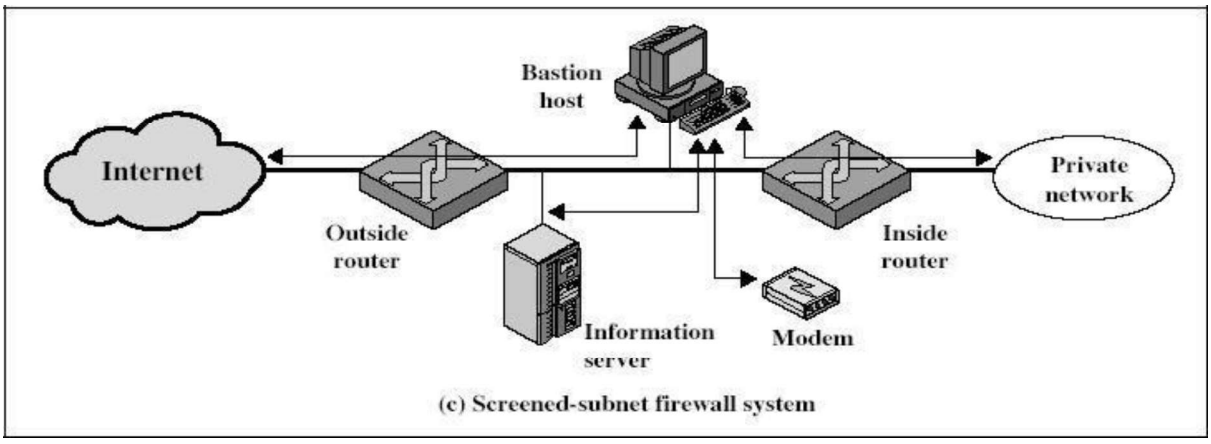
(b) Screened host firewall system (dual-homed bastion host)

3. Screened Subnet Firewall

The third implementation of a firewall is the screened subnet, which is also known as a *DMZ*. This firewall is the most secure one among the three implementations, simply because it uses a bastion host to support both circuit- and application-level gateways. All publicly accessible devices, including modem and server, are placed inside the DMZ. The screened subnet firewall contains external and internal screening routers. Each is configured such that its traffic flows only to or from the bastion host. This router also uses filters to prevent attacks such as IP spoofing and source routing.

This configuration offers several advantages:

- There are now three levels of defense to thwart intruders.
- The outside router advertises only the existence of the screened subnet to the internet; therefore the internal network is invisible to the internet.
- Similarly, the inside router advertises only the existence of the screened subnet to the internal network; therefore the systems on the internal network cannot construct direct routes to the internet.



(c) Screened-subnet firewall system

INTRUDER

One of the most publicized attacks to security is the intruder, generally referred to as hacker or cracker. Three classes of intruders are as follows:

1. **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.
2. **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 misuse his or her privileges.
3. **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.

Intrusion Techniques

The objective of the intruders is to gain access to a system or to increase the range of privileges accessible on a system. The password files can be protected in one of the two ways: **One way encryption –** The system stores only an encrypted form of user's password.
Access control – Access to the password file is limited to one or a very few accounts.

The following techniques are used 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.
4. Collect information about users such as their full names, the name of their spouse and children, pictures in their office and books in their office that are related to hobbies.
5. Try user's phone number, social security numbers and room numbers.
6. Try all legitimate license plate numbers.
7. Use a torjan horse to bypass restriction on access.
8. Tap the line between a remote user and the host system.

Two principle countermeasures:

1. **Detection –** concerned with learning of an attack, either before or after its success.
2. **Prevention –** challenging security goal

legitimate user in ways that can be quantified. Although the typical behaviour of an intruder differs from the typical behaviour of an authorized user, there is an overlap in these behaviours. Thus, a loose interpretation of intruder behaviour, which will catch more intruders,will also lead to a number of "false positives," or authorized users identified as intruder.

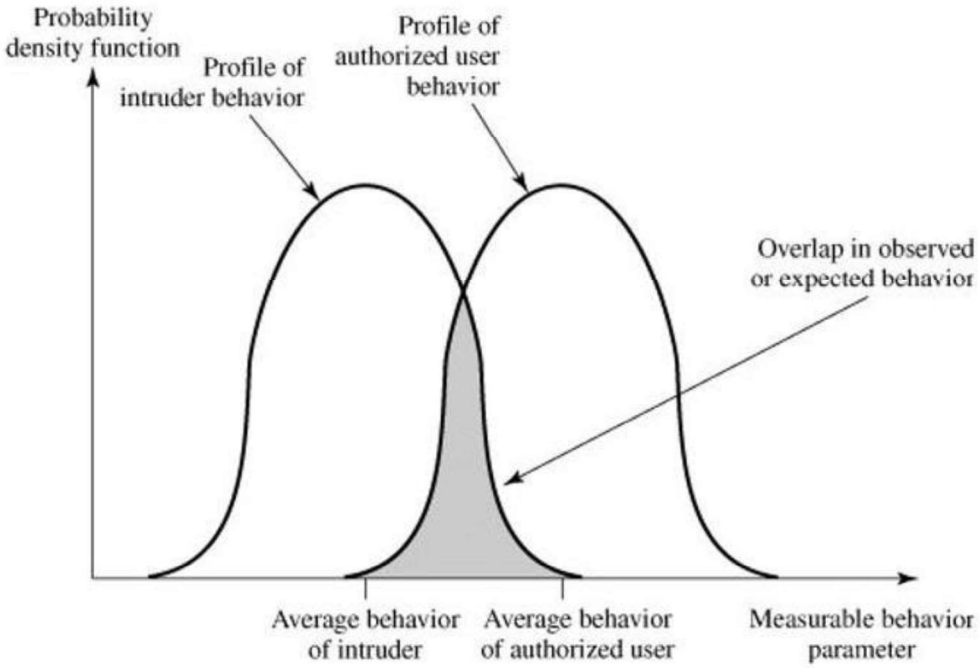


Figure: Profiles of Behavior of Intruders and Authorized Users Approaches to intrusion detection

1. **Statistical anomaly detection**
2. **Rule-based detection:**
3. **Distributed Intrusion Detection**
4. **Honeypot**

1. Statistical anomaly detection:

Involves the collection of data relating to the behaviour of legitimate users over a period of time. Then statistical tests are applied to observed behaviour to determine with a high level of confidence whether that behaviour is not legitimate user behaviour.

a) **Threshold detection:** This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.
b) **Profile based:** A profile of the activity of each user is developed and used to detect changes in the behaviour of individual accounts.

2. Rule-based detection:

Involves an attempt to define a set of rules that can be used to decide that a given behaviour is that of an intruder.

a) **Anomaly detection:** Rules are developed to detect deviation from previous usage patterns.
b) **Penetration identification:** An expert system approach that searches for suspicious behaviour.

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

1. **Native audit records:** Virtually all multiuser operating systems include accounting software that collects information on user activity. The advantage of using this

it in a convenient form.

2. **Detection-specific audit records:** A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems. The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine.

Each audit record contains the following fields:

- **Subject:** Initiators of actions. A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users.
- **Action:** Operation performed by the subject on or with an object; for example, login, read, perform I/O, execute.
- **Object:** Receptors of actions. Examples include files, programs, messages, records, terminals, printers, and user- or program-created structures
- **Exception-Condition:** Denotes which, if any, exception condition is raised on return.
- **Resource-Usage:** A list of quantitative elements in which each element gives the amount used of some resource
- **Time-Stamp:** Unique time-and-date stamp identifying when the action took place.

3. Distributed Intrusion Detection

Traditional focus is on single systems. But typically have networked systems. More effective defense has these working together to detect intrusions

- Dealing with varying audit record formats
- Integrity & confidentiality of networked data
- Centralized or decentralized architecture

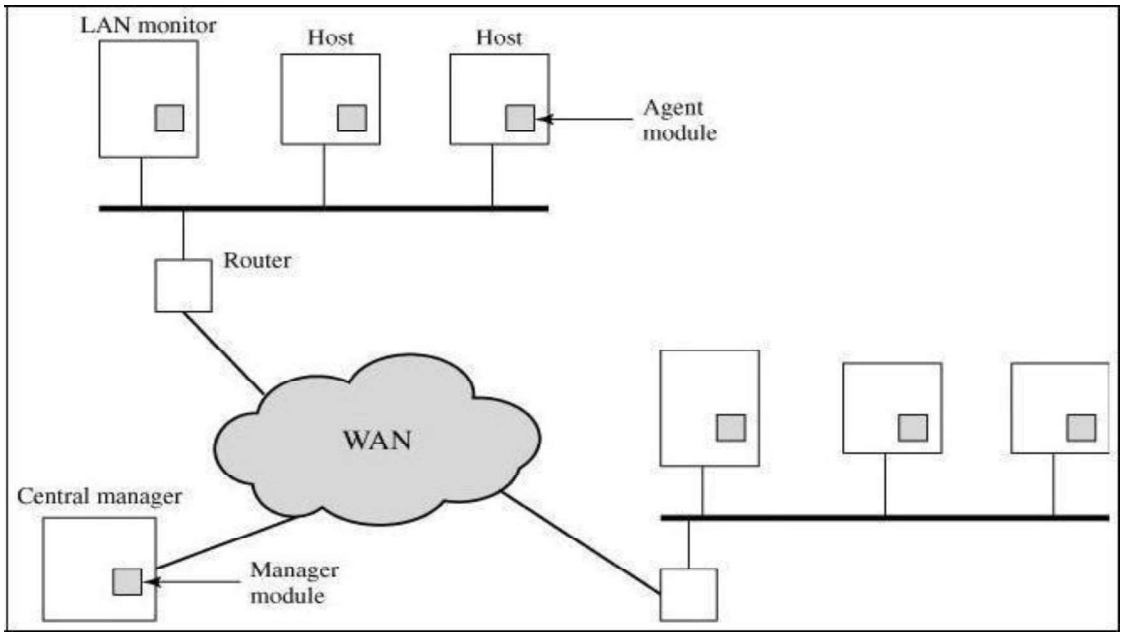


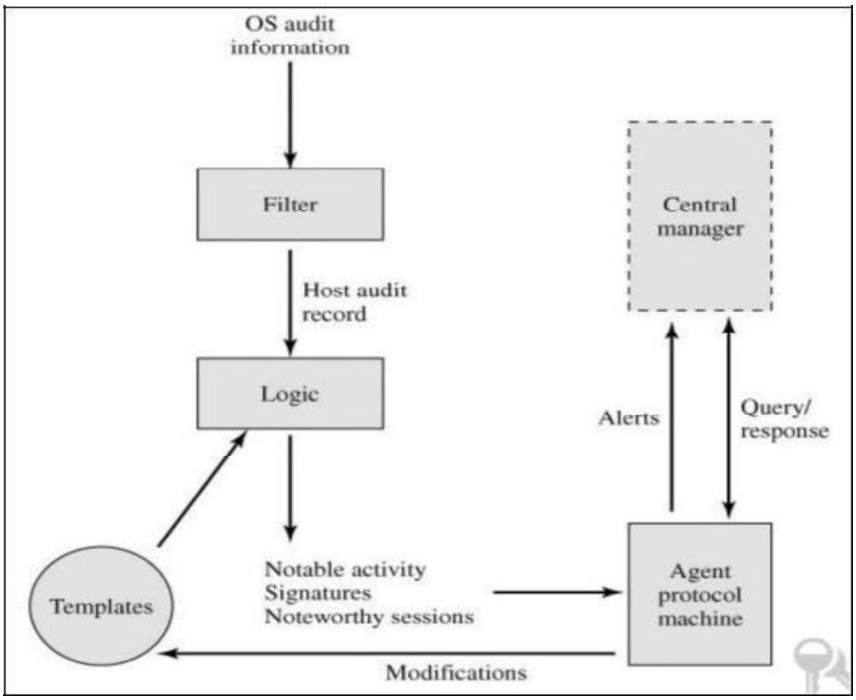
Figure. Architecture for Distributed Intrusion

Detection Three main components

1. **Host agent module:** An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security-related events on the host and transmit these to the central manager.
2. **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyses LAN traffic and reports the results to the central manager.
3. **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.

The agent captures each audit record produced by the native audit collection system.

1. Filter is applied that retains only those records that are of security interest.
2. These records are then reformatted into a standardized format referred to as the host audit record (HAR).
3. Next, a template-driven logic module analyzes the records for suspicious activity.
4. At the lowest level, the agent scans for notable events that are of interest independent of any past events.
5. At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures).
6. Finally, the agent looks for anomalous behaviour of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like.
7. When suspicious activity is detected, an alert is sent to the central manager.
8. The central manager includes an expert system that can draw inferences from received data.
9. The manager may also query individual systems for copies of HARs to correlate with those from other agents.
10. The LAN monitor agent also supplies information to the central manager.
11. The LAN monitor agent audits host-host connections, services used, and volume of traffic.
12. It searches for significant events, such as sudden changes in network load, the use of security-related services, and network activities such as rlogin.



4. Honeypots

Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems. Honeypots are designed to

- Divert an attacker from accessing critical systems
- Collect information about the attacker's activity
- Encourage the attacker to stay on the system long enough for administrators to respond

- These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system wouldn't access.
- The system is instrumented with sensitive monitors and event loggers that detect these accesses and collect information about the attacker's activities.
- Because any attack against the honeypot is made to seem successful, administrators have time to mobilize and log and track the attacker without ever exposing productive systems.

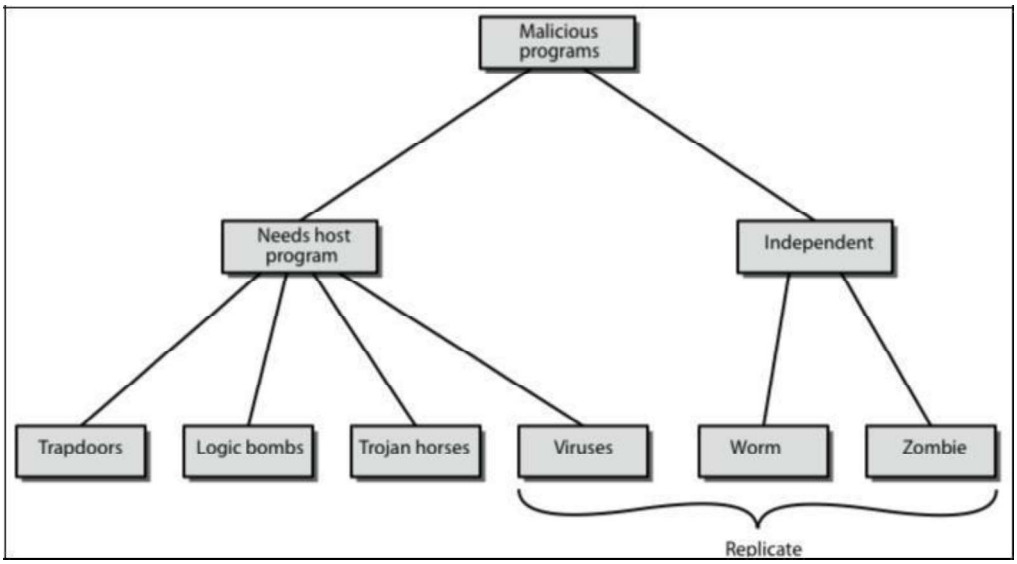
MALICIOUS SOFTWARE

Malicious software is software that is intentionally included or inserted in a system for a harmful purpose.

VIRUS AND RELATED THREATS

Malicious Programs

Name	Description
Virus	Attaches itself to a program and propagates copies of itself to other programs
Worm	Program that propagates copies of itself to other computers
Logic bomb	Triggers action when condition occurs
Trojan horse	Program that contains unexpected additional functionality
Backdoor (trapdoor)	Program modification that allows unauthorized access to functionality
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access
Zombie	Program activated on an infected machine that is activated to launch attacks on other machines



Malicious software can be divided into two categories:

Host program: Programs that cannot exist independently of some actual application program, utility, or system program.

Example: logic bombs, backdoors, and zombie programs.

Independent: A program fragment or an independent program that, when executed, may produce one or more copies of itself to be activated later on the same system or some other system.

Example: Viruses and worms

Backdoor or Trapdoor

A backdoor, also known as a trapdoor, is a secret entry point into a program that allows someone that is aware of the backdoor to gain access without going through the usual security access procedures. Programmers have used backdoors legitimately for many years to debug and test programs.

Logic Bomb

One of the oldest types of program threat. The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met. Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application. Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.

Trojan Horse

A Trojan horse is a useful program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function. Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly

Zombie

A zombie is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the zombie's creator. Zombies are used in denial of-service attacks, typically against targeted Web sites.

The Nature of Viruses

A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.

A typical virus goes through the following four phases:

- **Dormant phase**: The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit.
- **Propagation phase:** The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.
- **Triggering phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- **Execution phase:** The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Virus Structure

A virus can be prepended or postpened to an executable program, or it can be embedded in some other fashion. The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

An infected program begins with the virus code and works as follows.

- The first line of code is a jump to the main virus program. The second line is a special marker
- That is used by the virus to determine whether or not a potential victim program has already been infected with this virus.
- When the program is invoked, control is immediately transferred to the main virus program. The virus program first seeks out uninfected executable files and infects them.
- This action could be performed every time the program is invoked.
- Finally, the virus transfers control to the original program.

We assume that program P_1 is infected with the virus CV. When this program is invoked, control passes to its virus, which performs the following steps

1. For each uninfected file P_2 that is found, the virus first compresses that file to produce P'_2, which is shorter than the original program by the size of the virus.
2. A copy of the virus is prepended to the compressed program.
3. The compressed version of the original infected program, P'_1, is uncompressed.
4. The uncompressed original program is executed.

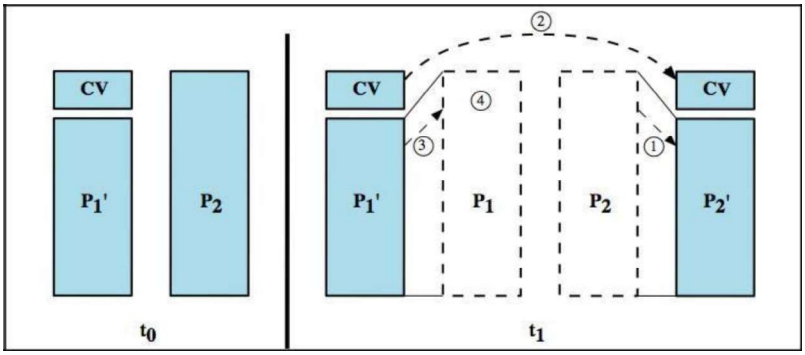


Figure: A Compression Virus

Types of Viruses

1. **Parasitic virus:** The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.
2. **Memory-resident virus:** Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.
3. **Boot sector virus:** Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.
4. **Stealth virus:** A form of virus explicitly designed to hide itself from detection by antivirus software.
5. **Polymorphic virus:** A virus that mutates with every infection, making detection by the "signature" of the virus impossible.
6. **Metamorphic virus:** As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at

each iteration, increasing the difficulty of detection. Metamorphic viruses my change their behavior as well as their appearance.

7. **Macro Viruses**
   - A macro virus is platform independent. Virtually all of the macro viruses infect Microsoft Word documents.
8. **E-mail Viruses**

   The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment. If the recipient opens the e-mail attachment, the Word macro is activated. Then the e-mail virus sends itself to everyone on the mailing list in the user's e-mail package.

VIRUS COUNTERMEASURES

Antivirus Approaches

The ideal solution to the threat of viruses is prevention: The next best approach is to be able to do the following:

- **Detection:** Once the infection has occurred, determine that it has occurred and locate the virus.
- **Identification:** Once detection has been achieved, identify the specific virus that has infected a program.
- **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the disease cannot spread further.

If detection succeeds but either identification or removal is not possible, then the alternative is to discard the infected program and reload a clean backup version.
There are four generations of antivirus software:

First-generation - simple scanners
- Scanner uses virus signature to identify virus
- Or change in length of programs

Second-generation - heuristic scanners
- Uses heuristic rules to spot viral infection
- Or uses crypto hash of program to spot changes

Third-generation - activity traps
- Memory-resident programs identify virus by actions

Fourth-generation - full-featured protection
- Packages with a variety of antivirus techniques
- Eg scanning & activity traps, access-controls

Advanced Antivirus Techniques

1. Generic Decryption

Generic decryption (GD) technology enables the antivirus program to easily detect even the most complex viruses. In order to detect such a structure, executable files are run through a GD scanner, which contains the following elements:

- **CPU emulator:** A software-based virtual computer. Instructions in an executable file are interpreted by the emulator rather than executed on the underlying processor. **Virus signature scanner:** A module that scans the target code looking for known virus signatures.
- **Emulation control module:** Controls the execution of the target code.

2. Digital Immune System

- General purpose emulation & virus detection
- Any virus entering organization is captured, analyzed, detection/shielding created for it, removed

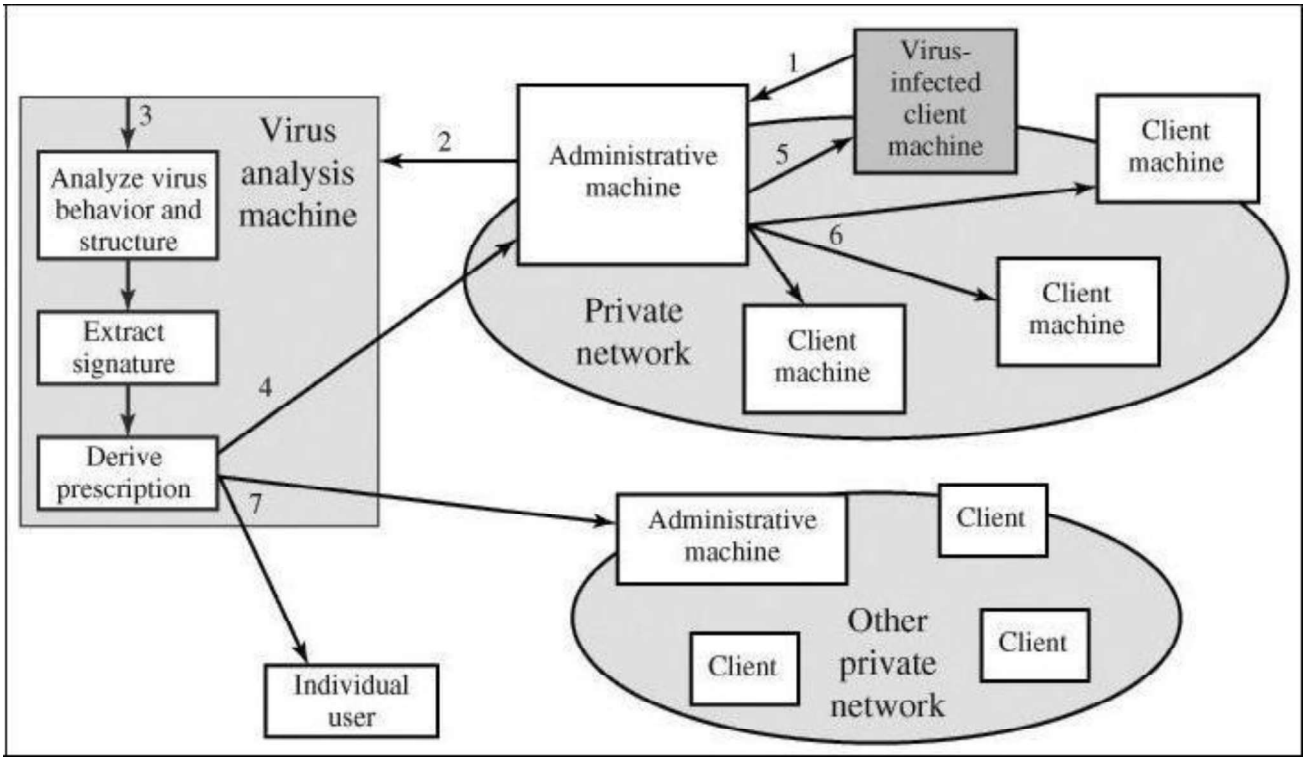


Figure illustrates the typical steps in digital immune system operation

1. The monitoring program forwards a copy of any program thought to be infected to an administrative machine within the organization.
2. The administrative machine encrypts the sample and sends it to a central virus analysis machine.
3. This machine creates an environment in which the infected program can be safely run for analysis. The virus analysis machine then produces a prescription for identifying and removing the virus.
4. The resulting prescription is sent back to the administrative machine.
5. The administrative machine forwards the prescription to the infected client.
6. The prescription is also forwarded to other clients in the organization.
7. Subscribers around the world receive regular antivirus updates that protect them from the new virus.

3. Behavior-Blocking Software

The behavior blocking software then blocks potentially malicious actions before they have a chance to affect the system. Monitored behaviors can include the following:

- Attempts to open, view, delete, and/or modify files;
- Attempts to format disk drives and other unrecoverable disk operations;
- Modifications to the logic of executable files or macros;
- Modification of critical system settings, such as start-up settings;

TRUSTED SYSTEMS

One way to enhance the ability of a system to defend against intruders and malicious programs is to implement trusted system technology.

1. Data access control

A general model of access control as exercised by a file or database management system is that of an access matrix. The basic elements of the model are as follows:

Subject: An entity capable of accessing objects. Generally, the concept of subject equates with that of process.

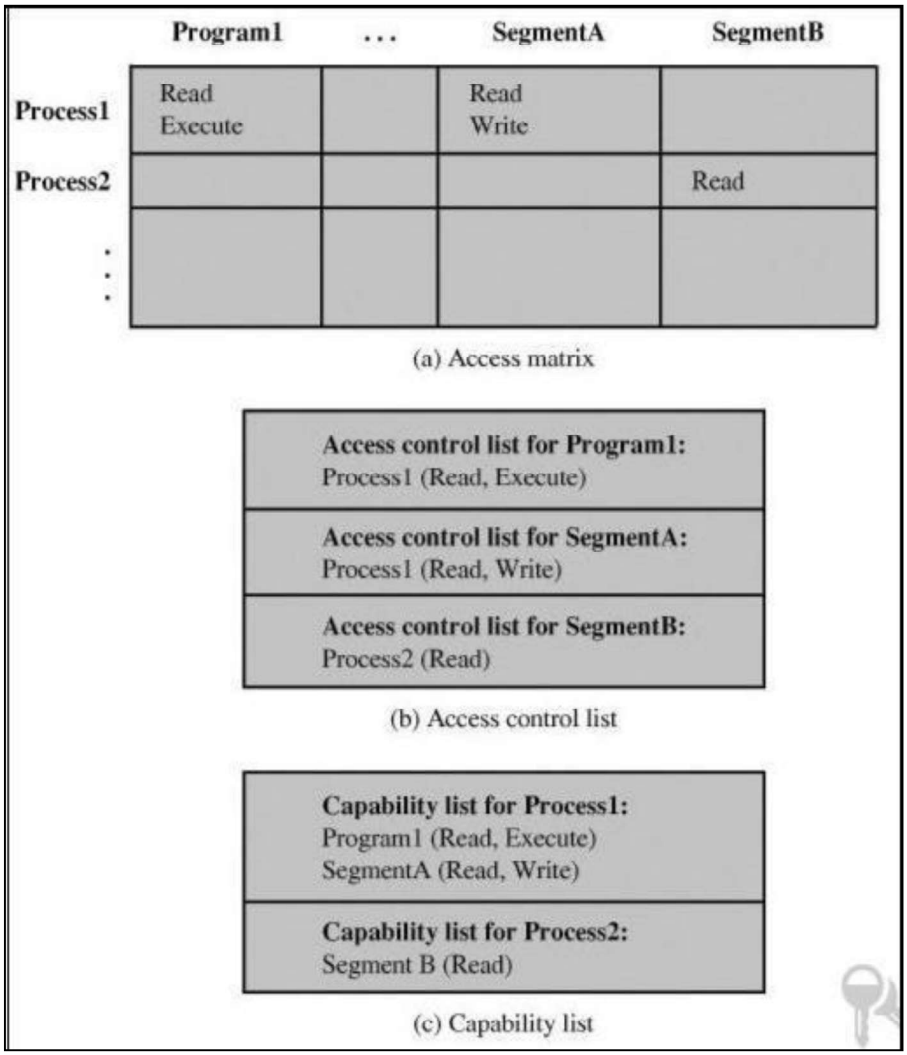
Object: Anything to which access is controlled. Examples include files, portion of files, programs, and segments of memory.

Access right: The way in which the object is accessed by a subject. Examples are read, write and execute

An access matrix is usually sparse and is implemented by decomposition in one of two ways.

1. **Access control lists**

   For each object, an access control list lists users and their permitted access rights. The access control list may contain a default, or public, entry



The Concept of Trusted Systems

When multiple categories or levels of data are defined, the requirement is referred to as **multilevel security**. A multilevel secure system must enforce the following:

- **No read up:** A subject can only read an object of less or equal security level. This is referred to in the literature as the **Simple Security Property**.
- **No write down:** A subject can only write into an object of greater or equal security level. This is referred to in the literature as the ***-Property**

These two rules, if properly enforced, provide multilevel security. For a data processing system, the approach that has been taken, and has been the object of much research and development, is based on the **reference monitor concept**

2. Reference Monitor concept

The reference monitor is a controlling element in the hardware and operating system of a computer that regulates the access of subjects to objects on the basis of security parameters of the subject and object. The reference monitor enforces the security rules (no read up, no write down) and has the following properties:

- **Complete mediation:** The security rules are enforced on every access
- **Isolation:** The reference monitor and database are protected from unauthorized modification.
- **Verifiability:** The reference monitor's correctness must be provable