CHAPTER 6

NETWORK SECURITY AND PUBLIC KEY INFRASTRUCTURE

CHAPTER OUTLINE

After studying this chapter, the students will be able to understand the

- Overview of Network Security
- Digital Certificates and X.509 certificates, Certificate Life Cycle Management
- PKI trust models, PKIX
- Email Security: Pretty Good Privacy (PGP)
- Secure Socket Layer (SSL) and Transport Layer Security (TLS)
- IP Security (IPSec)
- Firewalls and their types

OVERVIEW OF NETWORK SECURITY

Network security is a broad term that covers a multitude of technologies, devices and processes. In its simplest term, it is a set of rules and configurations designed to protect the integrity, confidentiality and accessibility of computer networks and data using both software and hardware technologies. Every organization, regardless of size, industry or infrastructure, requires a degree of network security solutions in place to protect it from the ever-growing landscape of cyber threats in the wild today.

Today's network architecture is complex and is faced with a threat environment that is always changing and attackers that are always trying to find and exploit vulnerabilities. These vulnerabilities can exist in a broad number of areas, including devices, data, applications, users and locations. For this reason, there are many network security management tools and applications in use today that address individual threats and exploits and also regulatory non-compliance. When just a few minutes of downtime can cause widespread disruption and massive damage to an organization's bottom line and reputation, it is essential that these protection measures are in place.

There are many layers to consider when addressing network security across an organization. Attacks can happen at any layer in the network security layers model, so your network security hardware, software and policies must be designed to address each area. Network security typically consists of three different controls: **physical, technical and administrative.** Here is a brief description of the different types of network security and how each control works.

- **Physical Network Security:** Physical security controls are designed to prevent unauthorized personnel from gaining physical access to network components such as routers, cabling cupboards and so on. Controlled access, such as locks, biometric authentication and other devices, is essential in any organization.
- **Technical Network Security:** Technical security controls protect data that is stored on the network or which is in transit across, into or out of the network. Protection is twofold; it needs to protect data and systems from unauthorized personnel, and it also needs to protect against malicious activities from employees.
- **Administrative Network Security:** Administrative security controls consist of security policies and processes that control user behavior, including how users are authenticated, their level of access and also how IT staff members implement changes to the infrastructure.

DIGITAL CERTIFICATES AND X.509 CERTIFICATES, CERTIFICATE LIFECYCLE MANAGEMENT

Digital certificates are electronic credentials issued by a trusted third party (Certificate Authority). A digital certificate is a digitally signed message used to attest to the validity of the public key of a communicating element. It not only verifies the identity of the owner, but also verifies that the owner owns the public key. It is based on trust or chain of trust. The trusted third party is a certificate authority, an entity that issues digital certificates. It verifies the digital signature is truly signed by the claimed signer.

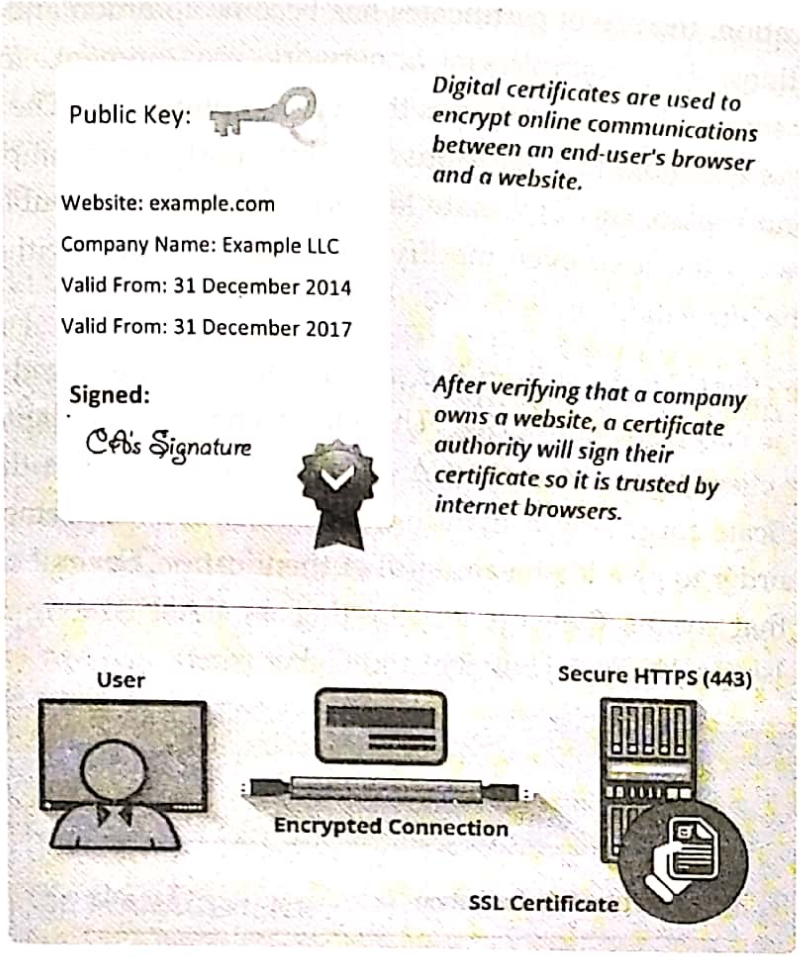


Figure 6.1: Digital certificate example

As we pointed out, digital certificates must adhere to a format. Most digital certificates follow the International Telecommunication Union (ITU-T) X.509 standard. According to RFC 1422, the X.509 digital certificate has the following fields as shown in table 6.1.

Table 6.1 The ITU-T X.509 digital certificate format

Field	Purpose
Version number	Most certificates use X.509 version 3.
Serial number	Unique number set by a CA
Issuer	Name of the CA
Subject issued certificate	Name of a receiver of the certificate
Validity period	Period in which certificate will valid
Public-key algorithm information of the subject of the certificate	Algorithm used to sign the certificate with digital signature
Digital signature of the issuing authority	Digital signature of the certificate signed by CA
Public key	Public key of the subject
Extension	Optional Extensions (e.g. Key usage)

In modern communication, the use of certificates has become common and vital to the security of such communications. For example, in a network environment, in order to encrypt transmissions to your server, the client requires the server's public key. The integrity of that key is vital to the security of the subsequent sessions. If a third party, for example, were to intercept the communication and replace the legitimate key with his/her own public key, that man-in-the-middle could view all traffic or even modify the data in transit. Neither the client nor the server would detect the intrusion.

So to prevent this, the client demands from the server, and the server sends the public key in a certificate signed by a certificate authority. The client checks that digital signature. If the signature is valid, the client knows that the CA has certified that this is the server's authentic certificate, not a certificate forged by a man-in-the-middle. It is important that the CA be a trusted third party in order to provide meaningful authentication. Several companies now offer digital certificates – that means they are functioning as CAs. Among those are VeriSign, American Express, Netscape, US Postal Service, and Cyber trust.

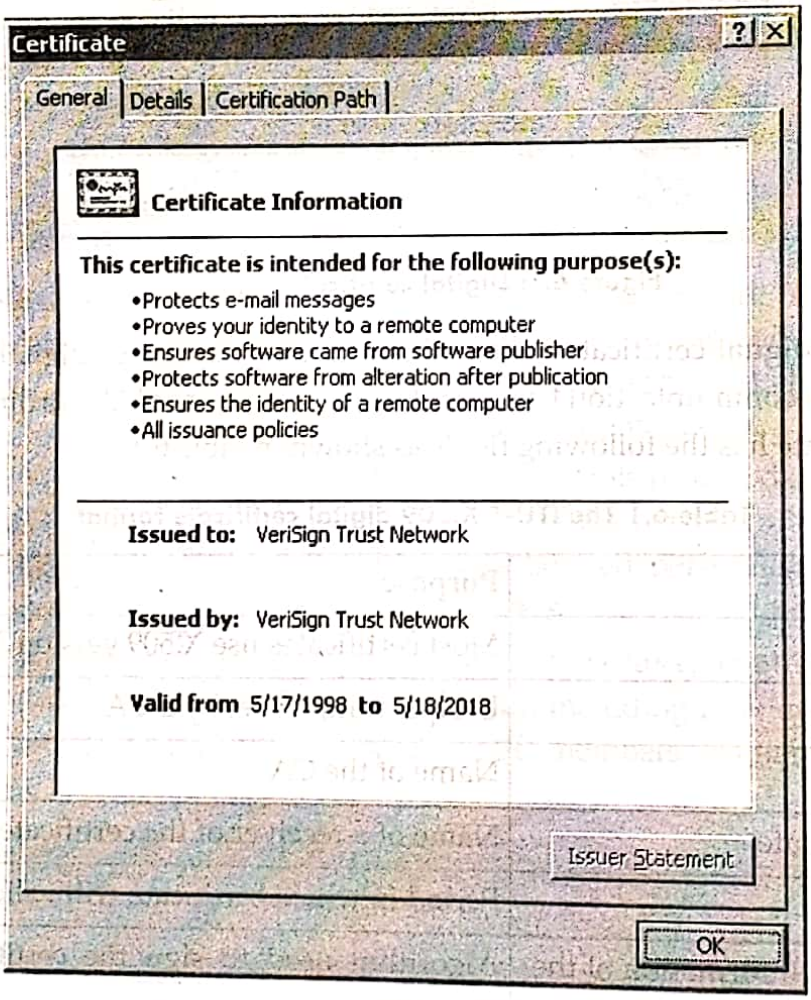


Figure 6.2: X.509 v3 Certificate issued by VeriSign Trust Network

X.509 VERSION 3

The X.509 version 2 formats does not convey all of the information that recent design and implementation experience has shown to be needed. List of the requirements not satisfied by version 2 are as follows:

1. The Subject field is inadequate to convey the identity of a key owner to a public key user. X.509 names may be relatively short and lacking in obvious identification details that may be needed by the user.
2. The Subject field is also inadequate for many applications, which typically recognize entities by an Internet e-mail address, a URL, or some other Internet-related identification.
3. There is a need to indicate security policy information. This enables a security application or function, such as IPSec, to relate an X.509 certificate to a given policy.
4. There is a need to limit the damage that can result from a faulty or malicious CA by setting constraints on the applicability of a particular certificate.
5. It is important to be able to identify different keys used by the same owner at different times. This feature supports key life cycle management, in particular the ability to update key pairs for users and CAs on a regular basis or under exceptional circumstances.

Rather than continue to add fields to a fixed format, standards developers felt that a more flexible approach was needed. Thus, version 3 includes a number of optional extensions that may be added to the version 2 format. Each extension consists of an extension identifier, a criticality indicator, and an extension value. The criticality indicator indicates whether an extension can be safely ignored. If the indicator has a value of TRUE and an implementation does not recognize the extension, it must treat the certificate as invalid.

CERTIFICATE LIFECYCLE MANAGEMENT

As network environments continue to expand, securing digital communications between components increases in complexity. Digital certificates establish the credentials of network components and ensure the secure transmission of sensitive information between client and server through identity authentication and data encryption.

A large deployment of digital certificates and private keys must be managed and doing so taxes an organization's time and resources. Managing multiple certificates with differing expiration dates issued by different vendors challenges even the most sophisticated enterprise.

Many organizations institute a Managed Public Key Infrastructure (MPKI) initiative to alleviate the strain. However, much of an MPKI initiative involves tedious, resource-intensive tasks. Failing to complete those tasks can result in the expiration of certificates that can put your network at risk, knock your organization out of compliance, make servers and other network assets unavailable and damage your brand due to network downtime.

PKI TRUST MODELS, PKIX

In large networks with varying communication topologies where network communicating elements cannot belong to the same KDC, key distribution becomes a real problem. These problems are solved when a Public Key Infrastructure (PKI) is used instead of KDCs to provide trusted and efficient key and certificate management. What then is this PKI? Merike Kaeo, quoting the Internet X.509 Public Key Infrastructure PKIX defines public key infrastructure (PKI) as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke certificates based on public key cryptography. PKI automate all these activities. PKI

works best when there is a large mass of users. Under such circumstances, it creates and distributes digital certificates widely to many users in a trusted manner. It is made up of four major pieces: the certificates that represent the authentication token; the CA that holds the ultimate decision on subject authentication; the registration authority (RA) that accepts and processes certificate signing requests on behalf of end users; and the Lightweight Directory Access Protocol (LDAP) directories that hold publicly available certificate information.

- **Certificates:** We defined certificates in previous section as the cryptographic proof that the public key they contain is indeed the one that corresponds to the identity stamped on the same certificate. The validation of the identity of the public key on the certificate is made by the CA that signs the certificate before it is issued to the user. Let us note here for emphasis that public keys are distributed through digital certificates. The X.509 v3 certificate format, as we noted in previous section has nine fields. The first seven make up the body of the certificate. Any change in these fields may cause the certificate to become invalid. If a certificate becomes invalid, the CA must revoke it. The CA then keeps and periodically updates the certificate revocation list (CRL). End-users are, therefore, required to frequently check on the CRL.
- **Certificate Authority (CA):** CAs are vital in PKI technology to authoritatively associate a public key signature with an alleged identity by signing certificates that support the PKI. Although the CAs play an important role in the PKI technology, they must be kept offline and used only to issue certificates to a select number of smaller certification entities. These entities perform most of the day-to-day certificate creation and signature verification.

  Since the CAs are offline and given their role in the PKI technology, there must be adequate security for the system on which they are stored so that their integrity is maintained. In addition, the medium containing the CA's secret key itself should be kept separate from the CA host in a highly secure location. Finally, all procedures that involve the handling of the CA private key should be performed by two or more operators to ensure accountability in the event of a discrepancy.
- **Registration Authority (RA):** The RAs accept and process certificate signing requests from users. Thus, they create the binding among public keys, certificate holders, and other attributes.
- **Lightweight Directory Access Protocols (LDAP):** These are repositories that store and make available certificates and Certificate Revocation Lists (CRLs). Developed at the University of Michigan, the LDAP was meant to make the access to X.509 directories easier. Other ways of distributing digital certificates are by FTP and HTTP.

PKIX MANAGEMENT FUNCTIONS

PKIX identifies a number of management functions that potentially need to be supported by management protocols. These include the following:

- **Registration:** This is the process whereby a user first makes itself known to a CA (directly, or through an RA), prior to that CA issuing a certificate or certificates for that user. Registration begins the process of enrolling in a PKI. Registration usually involves

some off-line or online procedure for mutual authentication. Typically, the end entity is issued one or more shared secret keys used for subsequent authentication.

- **Initialization:** Before a client system can operate securely, it is necessary to install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure. For example, the client needs to be securely initialized with the public key and other assured information of the trusted CA(s) to be used in validating certificate paths.
- **Certification:** This is the process in which a CA issues a certificate for a user's public key and returns that certificate to the user's client system and/or posts that certificate in a repository.
- **Key pair recovery:** Key pairs can be used to support digital signature creation and verification, encryption and decryption, or both. When a key pair is used for encryption/decryption, it is important to provide a mechanism to recover the necessary decryption keys when normal access to the keying material is no longer possible, otherwise it will not be possible to recover the encrypted data. Loss of access to the decryption key can result from forgotten passwords/PINs, corrupted disk drives, damage to hardware tokens, and so on. Key pair recovery allows end entities to restore their encryption/decryption key pair from an authorized key backup facility (typically, the CA that issued the end entity's certificate).
- **Key pair update:** All key pairs need to be updated regularly (i.e., replaced with a new key pair) and new certificates issued. Update is required when the certificate lifetime expires and as a result of certificate revocation.
- **Revocation request:** An authorized person advises a CA of an abnormal situation requiring certificate revocation. Reasons for revocation include private key compromise, change in affiliation, and name change.
- **Cross certification:** Two CAs exchange information used in establishing a cross-certificate. A cross-certificate is a certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates.

PKIX MANAGEMENT PROTOCOLS

The PKIX working group has defines two alternative management protocols between PKIX entities that support the management functions listed in the preceding subsection. RFC 2510 defines the certificate management protocols (CMP). Within CMP, each of the management functions is explicitly identified by specific protocol exchanges. CMP is designed to be a flexible protocol able to accommodate a variety of technical, operational, and business models.

RFC 2797 defines certificate management messages over CMS (CMC), where CMS refers to RFC 2630, and cryptographic message syntax. CMC is built on earlier work and is intended to leverage existing implementations. Although all of the PKIX functions are supported, the functions do not all map into specific protocol exchange.

PKI VERSES KERBEROS

The main secure architectures that can be implemented within any organization to secure the network interactions are PKI or Kerberos. The following table 6.2 illustrates the key difference between PKI and Kerberos:

Table 6.2: Comparison between PKI and Kerberos

PKI	Kerberos
Represents Asymmetric Cryptography.	Represents Symmetric Cryptography
With such architecture, each user has a pair of key, private key and public key. Where public key is published to users, the private key is kept secret. Private key is used to generate a digital signature, while the public key is used to verify such signature.	Tickets are used to authentication users, and the tickets are issues via online Key Distribution Center (KDC)
Private key is used to authenticate users. The private key is stored on disk, and maintain by users.	Password is required to authenticate users.
Pre-registration is not required in this case.	The key Distribution Center (KDC) must register every user to able to have access to the network.

EMAIL SECURITY: PRETTY GOOD PRIVACY (PGP)

The importance of sensitive communication cannot be underestimated. Sensitive information, whether in motion in communication channels or in storage, must be protected as much as possible. The best way, so far, to protect such information is to encrypt it. In fact, the security that the old snail mail offered was based on a seemingly protective mechanism similar to encryption when messages were wrapped and enclosed in envelopes. There was, therefore, more security during the days of snail mail because it took more time and effort for someone to open somebody's mail. First, one had to get access to it, which was no small task. Then one had to steam the envelope in order to open it and seal it later so that it looks unopened after. There were more chances of being caught doing so. Well, electronic communication has made it easy to intercept and read messages in the clear. So encryption of e-mails and any other forms of communication is vital for the security, confidentiality, and privacy of everyone. This is where PGP comes in and this is why PGP is so popular today. In fact, currently PGP is one of the popular encryption and digital signatures schemes in personal communication.

Pretty Good Privacy (PGP), developed by Phil Zimmermann, is a public-key cryptosystem. In public key encryption, one key is kept secret and the other key is made public. Secure communication with the receiving party (with a secret key) is achieved by encrypting the message to be sent using the recipient's public key. This message then can be decrypted only using the recipient's secret key.

PGP works by creating a circle of trust among its users. In the circle of trust, users, starting with two, form a key ring of public key/name pairs kept by each user. Joining this "trust club" means trusting and using the keys on somebody's key ring. Unlike the standard PKI infrastructure, this circle of trust has a built-in weakness that can be penetrated by an intruder. However, since PGP can be used to sign messages, the presence of its digital signature is used to verify the authenticity of a document or file. This goes a long way in ensuring that an e-mail message or file just downloaded from the Internet is both secure and un-tampered with.

PGP is regarded as hard encryption, that which is impossible to crack in the foreseeable future. Its strength is based on algorithms that have survived extensive public review and are already considered by many to be secure. Among these algorithms are RSA which PGP uses for encryption, DSS, and Diffie-Hellman for public key encryption; CAST-128, IDEA, and 3DES for conventional encryption; and SHA-1 for hashing. The actual operation of PGP is based on five services: authentication, confidentiality, compression, e-mail compatibility, and segmentation.

AUTHENTICATION

PGP provides authentication via a digital signature scheme. The hash code (MAC) is created using a combination of SHA-1 and RSA to provide an effective digital signature. It can also create an alternative signature using DSS and SHA-1. The signatures are then attached to the message or file before sending. PGP, in addition, supports unattached digital signatures. In this case, the signature may be sent separately from the message.

CONFIDENTIALITY

PGP provides confidentiality by encrypting messages before transmission. PGP encrypts messages for transmission and storage using conventional encryption schemes such as CAST-128, IDEA, and 3DES. In each case, a 64-bit cipher feedback mode is used. As in all cases of encryption, there is always a problem of key distribution; so PGP uses a conventional key once. This means for each message to be sent, the sender mints a brand new 128-bit session key for the message. The session key is encrypted with RSA or Diffie-Hallman using the recipient's public key; the message is encrypted using CAST-128 or IDEA or 3DES together with the session key. The combo is transmitted to the recipient. Upon receipt, the receiver uses RSA with his or her private key to encrypt and recover the session key which is used to recover the message.

COMPRESSION

PGP compresses the message after applying the signature and before encryption. The idea is to save space.

E-MAIL COMPATIBILITY

As we have seen above, PGP encrypts a message together with the signature (if not sent separately) resulting into a stream of arbitrary 8-bit octets. But since many e-mail systems permit only use of blocks consisting of ASCII text, PGP accommodates this by converting the raw 8-bit binary streams into streams of printable ASCII characters using a radix-64 conversion

scheme. On receipt, the block is converted back from radix-64 format to binary. If the message is encrypted, then a session key is recovered and used to decrypt the message. The result is then decompressed. If there is a signature, it has to be recovered by recovering the transmitted hash code and comparing it to the receiver's calculated hash before acceptance.

Segmentation

To accommodate e-mail size restrictions, PGP automatically segments email messages that are too long. However, the segmentation is done after all the housekeeping is done on the message, just before transmitting it. So the session key and signature appear only once at the beginning of the first segment transmitted. At receipt, the receiving PGP strip off all e-mail headers and re-assembles the original mail. PGP's popularity and use has so far turned out to be less than anticipated because of two reasons: first, its development and commercial distribution after Zimmermann sold it to Network Associates, which later sold it to another company did not do well; second, its open source cousin, the OpenPGP, encountered market problems including the problem of ease of use. Both OpenPGP and commercial PGP are difficult to use because it is not built into many e-mail clients. This implies that any two communicating users who want to encrypt their e-mail using PGP have to manually download and install PGP, a challenge and an inconvenience to many users.

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

Netscape originated SSL. Version 3 of the protocol was designed with public review and input from industry and was published as an Internet draft document. Subsequently, when a consensus was reached to submit the protocol for Internet standardization, the TLS working group was formed within IETF to develop a common standard. This first published version of TLS can be viewed as essentially an SSLv3.1 and is very close to and backward compatible with SSLv3.

This section is devoted to a discussion of SSLv3. In the next section, the principal differences between SSLv3 and TLS are described.

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 6.3. The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides 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.

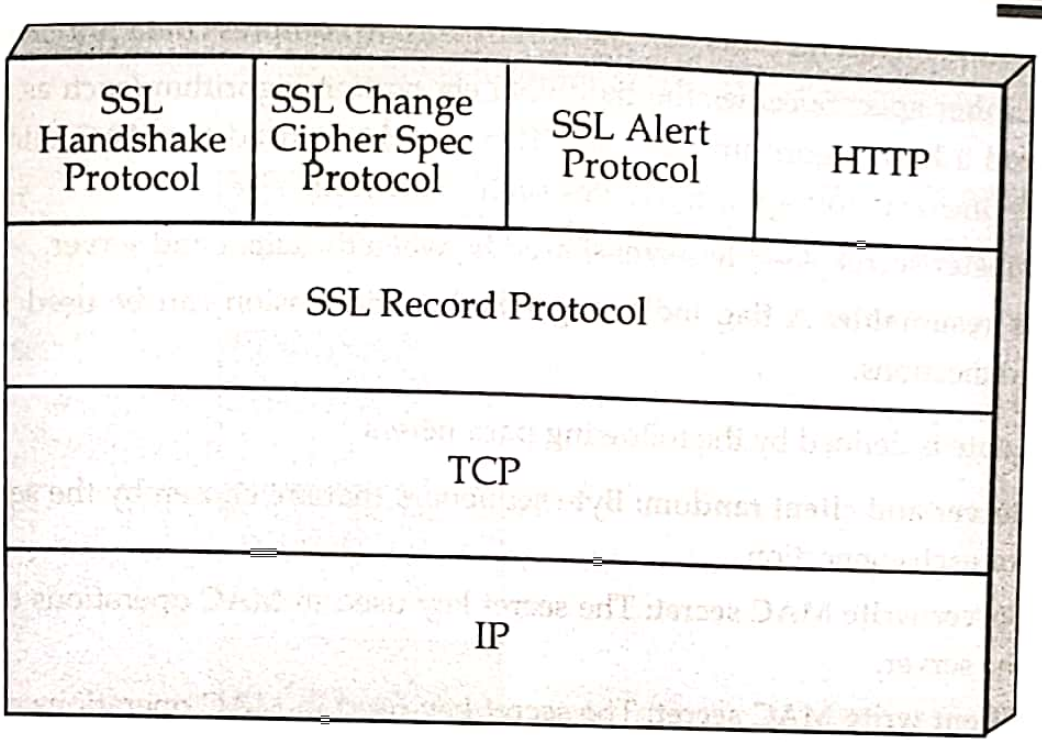


Figure 6.3: SSL Protocol Stack

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 writes 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.

A connection state is defined by the following parameters.

- **Server and client random:** Byte sequences that are chosen by the server and client for each connection.
- **Server write MAC secret:** The secret key used in MAC operations on data sent by the server.
- **Client write MAC secret:** The secret key used in MAC operations on data sent by the client.
- **Server write key**: The secret encryption key for data encrypted by the server and decrypted by the client.
- **Client write key:** The symmetric encryption key for data encrypted by the client and decrypted by the server.
- **Initialization vectors:** When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record.
- **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed $2^{64}-1$.

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 6.4 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.

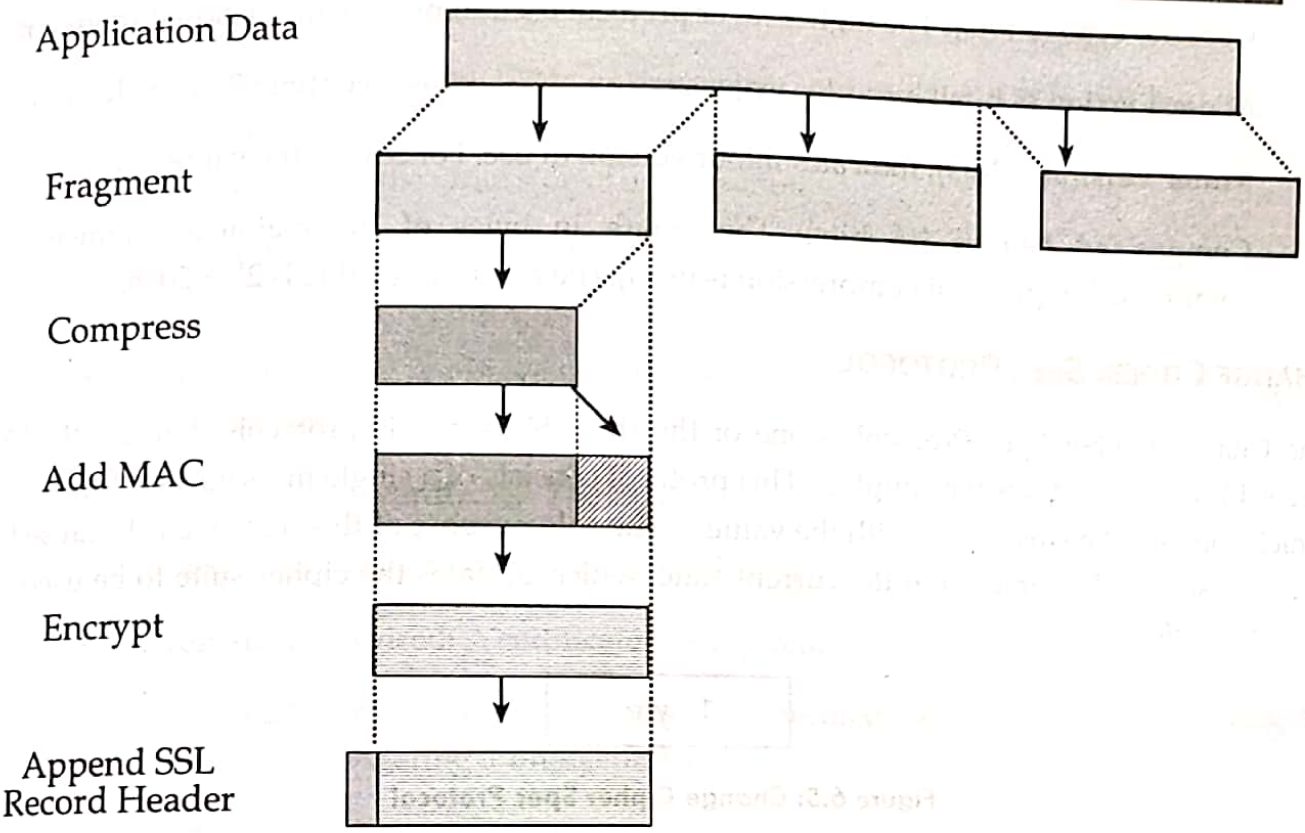


Figure 6.4: SSL Record Protocol Operation

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. The next step in processing is to compute a **message authentication code** over the compressed data. For this purpose, a shared secret key is used. This is very similar to **HMAC** algorithm. Next, the compressed message plus the **MAC** are **encrypted** using symmetric encryption. Encryption may not increase the content length by more than 1024 bytes, so the total length may not exceed 2^{14} + 2048. The following encryption algorithms are permitted:

Block Cipher		Stream Cipher	
Algorithm	**Key Size**	**Algorithm**	**Key Size**
AES	128,256	RC4-40	40
IDEA	128	RC4-128	140
RC2-40	40		
DES-40	40		
DES	56		
3DES	168		
Fortezza	80		

The final step of **SSL Record Protocol** processing is to prepare a header consisting of the following fields:

- **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. For SSLv3, the value is 3.
- **Minor Version (8 bits):** Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits):** The length in bytes of the plaintext fragment (or compressed fragment if compression is used).The maximum value is $2^{14} + 2048$.

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 (see figure 6.5), 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.

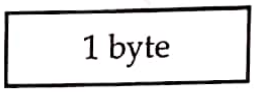


Figure 6.5: Change Cipher Spec Protocol

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.

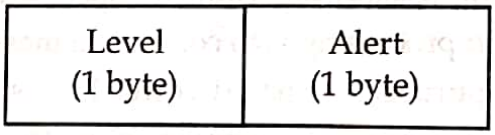


Figure 6.6: Alert Protocol

Each message in this protocol consists of two bytes (see figure 6.6). The first byte takes the value warning (1) or fatal (2) to convey the severity of the message. The second byte contains a code that indicates the specific alert. The first listed group is the fatal alerts, the others are warnings.

- **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.

The remaining alerts are the following.

- **close_notify:** Notifies the recipient that the sender will not send any more messages on this connection. Each party is required to send a **close_notify** alert before closing the write side of a connection.
- **no_certificate:** May be sent in response to a certificate request if no appropriate certificate is available.
- **bad_certificate:** A received certificate was corrupt (e.g., contained a signature that did not verify).
- **unsupported_certificate:** The type of the received certificate is not supported.
- **certificate_revoked:** A certificate has been revoked by its signer.
- **certificate_expired:** A certificate has expired.
- **certificate_unknown:** Some other unspecified issue arose in processing the certificate, rendering it unacceptable.

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 6.7.

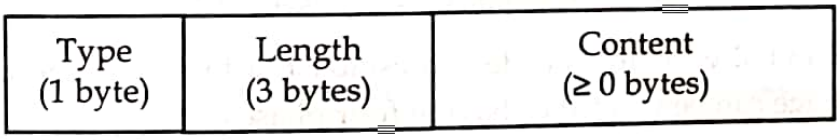


Figure 6.7: Handshake Protocol

Table 6.3: 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	parameters, signature
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

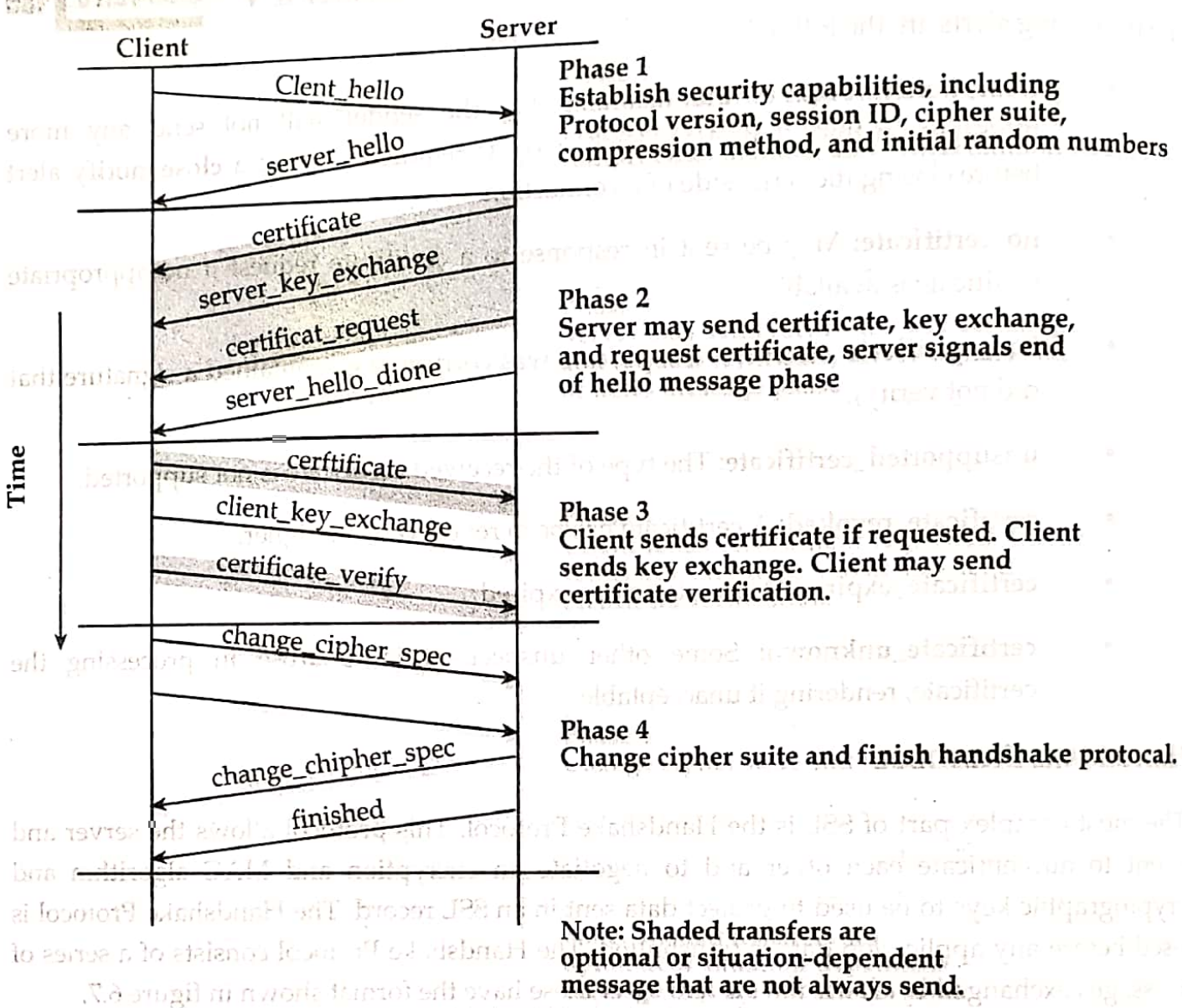


Figure 6.8: Handshake Protocol Action

Figure 6.8 shows the initial exchange needed to establish a logical connection between client and server. The exchange can be viewed as having four phases.

Phase 1, Establish Security Capabilities: This phase is used by the client to initiate a logical connection and to establish the security capabilities that will be associated with it.

- This phase comprises of exchange of two messages – **client_hello** and **server_hello**.
- **client_hello** contains of list of cryptographic algorithms supported by the client, in decreasing order of preference.
- **server_hello** contains the selected Cipher Specification (CipherSpec) and a new session_id.
- The CipherSpec contains fields like:
  - Cipher Algorithm (DES, 3DES, RC2, and RC4)
  - MAC Algorithm (based on MD5, SHA-1)
  - Public-key algorithm (RSA)
  - Both messages have "nonce" to prevent replay attack.

Phase 2, Server Authentication and Key Exchange: The server begins this phase by sending its certificate if it needs to be authenticated.

- Server sends chosen cipher suite.
- Server may request client certificate. Usually it is not done.
- Server indicates end of **Server_hello**.

Phase 3, Client Authentication and Key Exchange: The client should verify that the server provided a valid certificate if required and check that the **server_hello** parameters are acceptable.

- Client sends certificate, only if requested by the server.
- It also sends the Pre-master Secret (PMS) encrypted with the server's public key.
- Client also sends **Certificate_verify** message if certificate is sent by him to prove he has the private key associated with this certificate. Basically, the client signs a hash of the previous messages.

Phase 4, Finish: This phase completes the setting up of a secure connection. The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec

TRANSPORT LAYER SECURITY

Transport Layer Security (TLS) is the result of the 1996 Internet Engineering Task Force (IETF) attempt at standardization of a secure method to communicate over the Web. The 1999 outcome of that attempt was released as RFC 2246 spelling out a new protocol – the Transport Layer Security or TLS. TLS was charged with providing security and data integrity at the transport layer between two applications. TLS version 1.0 was an evolved SSL 3.0. So, as we pointed out earlier, TLS is the successor to SSL 3.0. Frequently, the new standard is referred to as SSL/TLS.

Since then, however, the following additional features have been added:

- **Interoperability:** Ability to exchange TLS parameters by either party, with no need for one party to know the other's TLS implementation details.
- **Expandability:** To plan for future expansions and accommodation of new protocols.

IP SECURITY (IPSEC)

The IP security (IPSec) is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted and authenticated packets. Although it was designed to run in the new version of the Internet Protocol, IP Version 6 (IPv6), it has also successfully run in the older IPv4 as well. Some of the **benefits** of IPSec:

- When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPSec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPSec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router. Even if IPSec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPSec can provide security for individual users if needed. This is useful for offsite workers and for setting up a secure virtual sub network within an organization for sensitive applications.

In addition to supporting end users and protecting premises systems and networks, IPSec can play a vital role in the routing architecture required for internetworking. IPSec can assure that

- A router advertisement (a new router advertises its presence) comes from an authorized router.
- A neighbor advertisement (a router seeks to establish or maintain a neighbor relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial IP packet was sent.
- A routing update is not forged.

Without such security measures, an opponent can disrupt communications or divert some traffic. Routing protocols such as Open Shortest Path First (OSPF) should be run on top of security associations between routers that are defined by IPSec.

IPSec sets out to offer protection by providing the following services at the network layer:

- **Access control:** to prevent an unauthorized access to the resource.
- **Connectionless integrity:** to give an assurance that the traffic received has not been modified in any way.
- **Confidentiality:** to ensure that Internet traffic is not examined by non-authorized parties. This requires all IP datagrams to have their data field, TCP, UDP, ICMP, or any other datagram data field segment, encrypted.
- **Authentication:** particularly source authentication so that when a destination host receives an IP datagram, with a particular IP source address, it is possible to be sure that the IP datagram was indeed generated by the host with the source IP address. This prevents spoofed IP addresses.

- **Replay protection:** to guarantee that each packet exchanged between two parties is different.

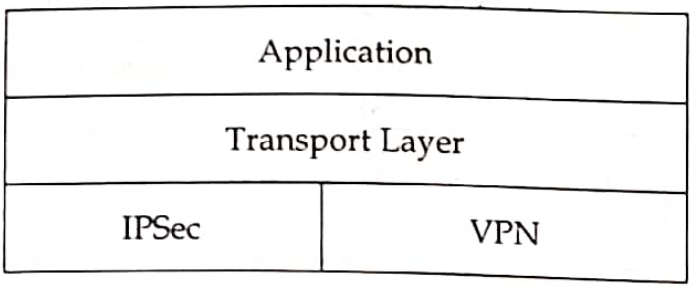


Figure 6.9: Network Layer Security Protocols and Standards

IPSec protocol achieves these two objectives by dividing the protocol suite into two main protocols: Authentication Header (AH) protocol and the Encapsulation Security Payload (ESP) protocol. The AH protocol provides source authentication and data integrity but no confidentiality. The ESP protocol provides authentication, data integrity, and confidentiality. Any datagram from a source must be secured with either AH or ESP.

AUTHENTICATION HEADER (AH)

AH protocol provides source authentication and data integrity but not confidentiality. This is done by a source that wants to send a datagram first establishing an SA, through which the source can send the datagram. A source datagram includes an AH inserted between the original IP datagram data and the IP header to shield the data field which is now encapsulated as a standard IP datagram. Upon receipt of the IP datagram, the destination host notices the AH and processes it using the AH protocol. Intermediate hosts such as routers, however, do their usual job of examining every datagram for the destination IP address and then forwarding it on.

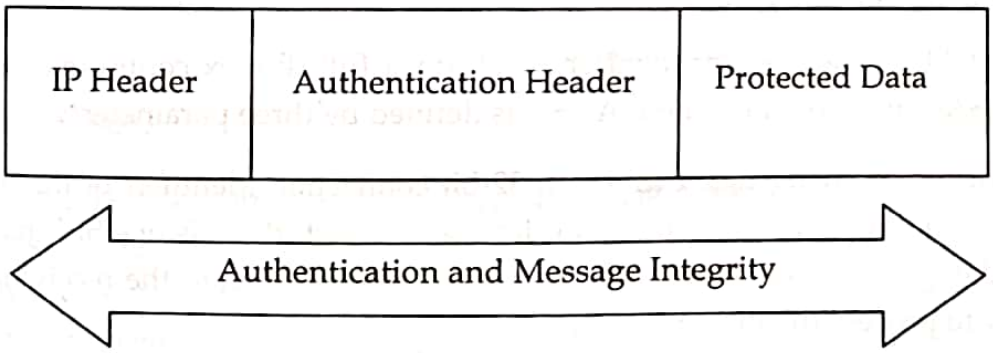


Figure 6.10: IPSec's AH Protocol Protection

ENCAPSULATING SECURITY PAYLOAD (ESP)

Unlike the AH protocol, ESP protocol provides source authentication, data integrity, and confidentiality. This has made ESP the most commonly used IPSec header. Similar to AH, ESP begins with the source host establishing an AS which it uses to send secure datagrams to the destination. Datagram are secured by ESP by surrounding their original IP datagrams with a new header and trailer fields all encapsulated into a new IP datagram. Confidentiality is

provided by DES_CBC encryption. Next to the ESP trailer field on the datagram is the ESP Authentication Data field.

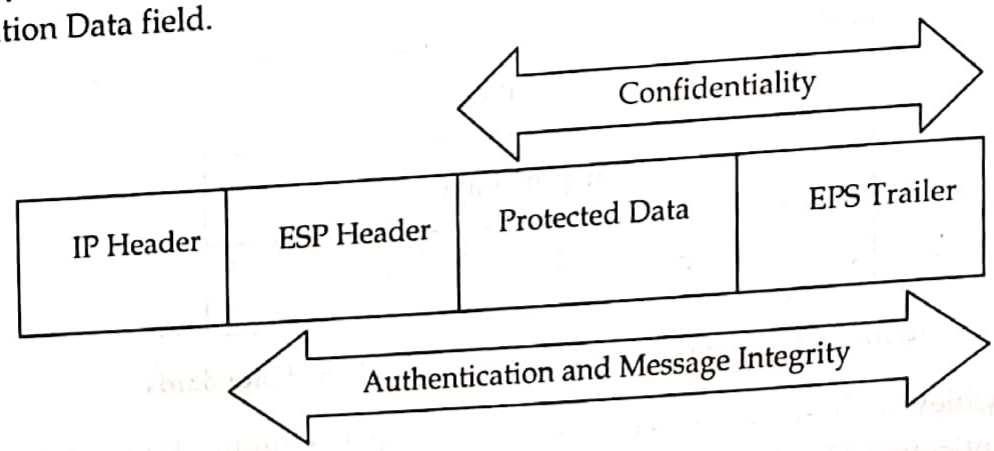


Figure 6.11: IPSec's ESP Protocol Protection

SECURITY ASSOCIATIONS

In order to perform the security services that IPSec provides, IPSec must first get as much information as possible on the security arrangement of the two communicating hosts. Such security arrangements are called security associations (CAs). A security association is a unidirectional security arrangement defining a set of items and procedures that must be shared between the two communicating entities in order to protect the communication process.

Recall from OSI Layer that in the usual network IP connections, the network layer IP is connectionless. However, with security associations, IPSec creates logical connection-oriented channels at the network layer. This logical connection- oriented channel is created by a security agreement established between the two hosts stating specific algorithms to be used by the sending party to ensure confidentiality (with ESP), authentication, message integrity, and anti-replay protection.

Since each AS establishes a unidirectional channel, for a full duplex communication between two parties, two SAs must be established. An SA is defined by three parameters.

- Security Parameter Index (SPI) – a 32-bit connection identifier of the SA. For each association between a source and destination host, there is one SPI that is used by all datagrams in the connection to provide information to the receiving device on how to process the incoming traffic.
- IP Destination Address – address of a destination host.
- A Security Protocol (AH or ESP) to be used and specifying if traffic is to be provided with integrity and secrecy. The protocol also defines the key size, key lifetime, and the cryptographic algorithms.
- Secret key – which defines the keys to be used.
- Encapsulation mode – defining how encapsulation headers are created and which parts of the header and user traffic are protected during the communication process.

TRANSPORT AND TUNNEL MODES

The security associations discussed above are implemented in two modes: transport and tunnel. This means that IPSec is operating in two modes. Let us look at these.

Transport mode provides host-to-host protection to higher layer protocols in the communication between two hosts in both IPv4 and IPv6. In IPv4, this area is the area beyond the IP address as shown in figure 6.12. In IPv6, the new extension to IPv4, the protection includes the upper protocols, the IP address, and any IPv6 header extensions as shown in figure 6.10. The IP addresses of the two IPSec hosts are in the clear because they are needed in routing the datagram through the network.

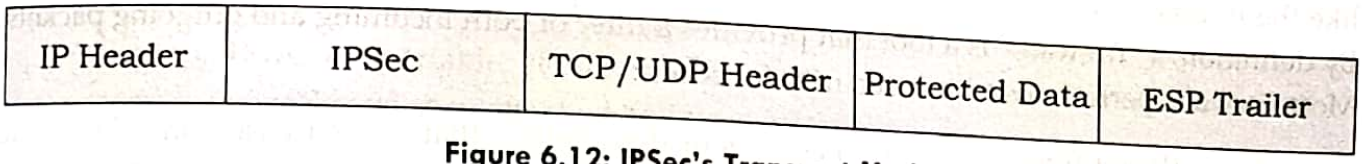


Figure 6.12: IPSec's Transport Mode

Tunnel mode offers protection to the entire IP datagram both in AH and ESP between two IPSec gateways. This is possible because of the added new IP header in both IPv4 and IPv6 as shown in figure 6.13. Between the two gateways, the datagram is secure and the original IP address is also secure. However, beyond the gateways, the datagram may not be secure. Such protection is created when the first IPSec gateway encapsulate the datagram including its IP address into a new shield datagram with a new IP address of the receiving IPSec gateway. At the receiving gateway, the new datagram is unwrapped and brought back to the original datagram. This datagram, based on its original IP address, can be passed on further by the receiving gateway, but from this point on unprotected.

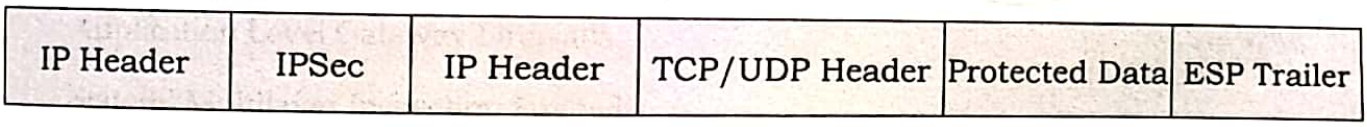


Figure 6.13: IPSec's Transport Mode

OTHER IPSEC ISSUES

Any IPSec compliant system must support single-DES, MD5, and SHA-1 as an absolute minimum; this ensures that a basic level of inter-working is possible with two IPSec compliant units at each end of the link. Since IPSec sits between the Network and Transport layers, the best place for its implementation is mainly in hardware.

FIREWALLS AND THEIR TYPES

The rapid growth of the Internet has led to a corresponding growth of both users and activities in cyberspace. No one can deny the fact that the dynamic rise of the Internet has brought the world closer. But at the same time, it has left us with different kinds of security threats. Therefore, there has been a need to protect company systems, and now individual PCs, keeping them out of access from the "bad users" out on the "bad Internet." As companies build private networks and decide to connect them onto the Internet, network security becomes one of the

most important concerns network system administrators face. In fact, these network administrators are facing threats from two fronts: the external Internet and the internal users within the company network. So network system administrators must be able to find ways to restrict access to the company network or sections of the network from both the "bad Internet" outside and from unscrupulous inside users.

Such security mechanisms are based on a **firewall**. A firewall is hardware, software, or a combination of both that monitors and filters traffic packets that attempt to either enter or leave the protected private network. It is a tool that separates a protected network or part of a network, and now increasingly a user PC, from an unprotected network – the "bad network" like the Internet. In many cases the "bad network" may even be part of the company network. By definition, a "firewall," is a tool that provides a filter of both incoming and outgoing packets. Most firewalls perform two basic security functions:

- Packet filtering based on accepts or deny policy that is itself based on rules of the security policy.
- Application proxy gateways that provide services to the inside users and at the same time protect each individual host from the "bad" outside users.

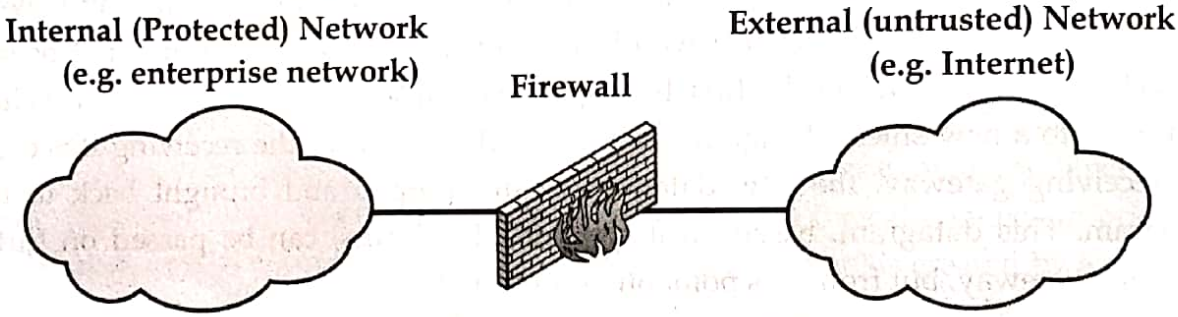


Figure 6.14: General model of firewall

FIREWALL CHARACTERISTICS

Before proceeding to the details of firewall types and configurations, it is best to summarize what one can expect from a firewall. The following capabilities are within the scope of a firewall:

1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks.
2. A firewall provides a location for monitoring security-related events
3. A firewall is a convenient platform for several Internet functions that are not security related, such as NAT and Internet usage audits or logs.
4. A firewall can serve as the platform for IPSec to implement virtual private networks.

Firewalls have their limitations, including the following:

1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.
2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.
3. An improperly secured wireless LAN may be accessed from outside the organization. An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall.
4. A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally.

TYPES OF FIREWALL

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 pattern generated by a sequence of packets. In this section, we look at the principal types of firewalls and they are:

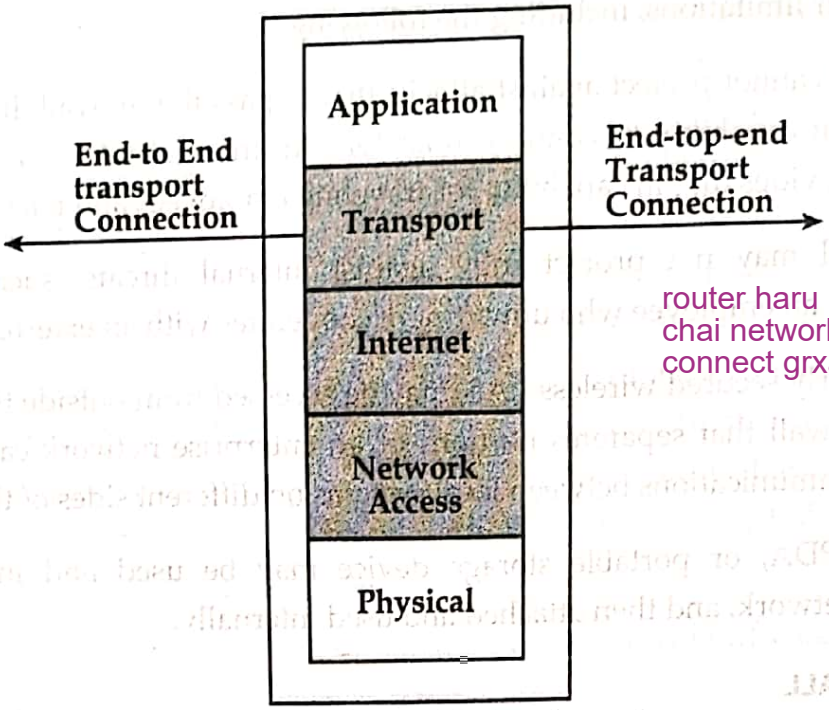
1. Packet Filtering Firewalls
2. Circuit Level Gateway Firewalls
3. Application Level Gateway Firewalls
4. Stateful Multilayer Inspection Firewalls

jun layer ma vyeni diagram ma transport batae dekhaxa

1. Packet Filtering Firewall

network layer ma hunxa

Packet filtering firewalls are normally deployed on the routers which connect the internal network to internet. It can only to be implemented on the network layer of OSI model. Packet filtering firewalls work on the basis of rules defines by Access Control Lists (ACL). They check all the packets and screen them against the rules defined by the network administrator as per the ACLs. If in case, any packet does not meet the criteria then that packet is dropped and logs are updated about this information. Administrators can create their ACLs on the basis of address, protocols and packet attributes. The biggest advantage of packet filtering firewalls is cost and lower resource usage. It is best suited for smaller networks. The disadvantage of packet filtering firewalls is it can work only on the network layer and these firewalls do not support complex rule based models. Also, it is vulnerable to spoofing in some cases.



router haru ma hunxa jsle chai network lai internet snga connect grxa so esto hola

Figure 6.15: Packet filtering firewall

2. Circuit Level Gateway Firewalls

sessin layer ma hunxa

Circuit level gateways are deployed at the session layer of the OSI model and they monitor sessions like TCP three way handshake to see whether a requested connection is legitimate or not. Major screening happens before the connection is established. Information sent to a computer outside the network through a circuit level gateway appears to have originated from the gateway. This helps in creating a stealth cover for the private network from outsiders. One advantage of circuit level gateway is it is comparatively inexpensive and provides anonymity to the private network. One disadvantage of circuit level gateway is it does not filter individual packets. After establishing a connection, an attacker may take advantage of this.

duiwota computer bich ma TCP/IP handshake haru use gaererw connection establish hune raexa

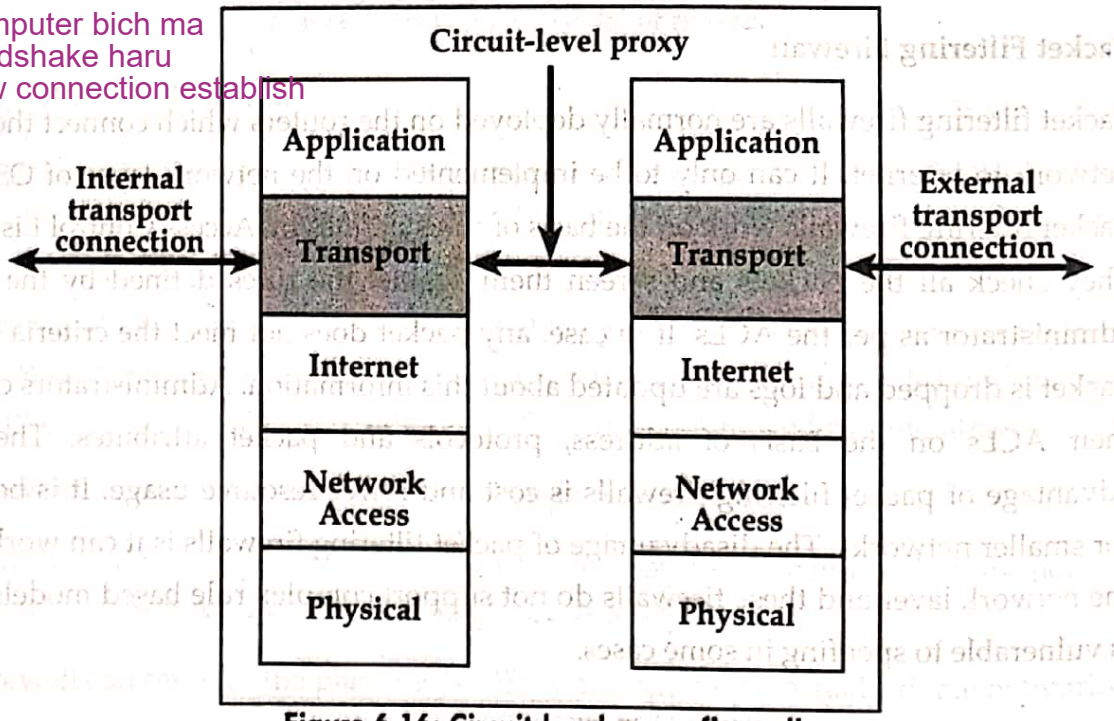


Figure 6.16: Circuit-level proxy firewall

3. Application Level Gateway Firewalls

Application level gateways work on the application layer of the OSI model and provide protection for a specific application layer protocol. Proxy server is the best example of application level gateways firewalls. Application level gateway would work only for the protocols which are configured. For example, if we install a web proxy based firewall than it will only allow HTTP protocol data. They are supposed to understand application specific commands such as HTTP:GET and HTTP:POST as they are deployed on the application layer, for a specific protocol. Application level firewalls can also be configured as caching servers which in turn increase the network performance and makes it easier to log traffic.

main kaam vneko
evaluate packets
if webproxy firewall ho vne
http packet chai allow hunxa
ani else command ni http:get rw
http:post bujxa

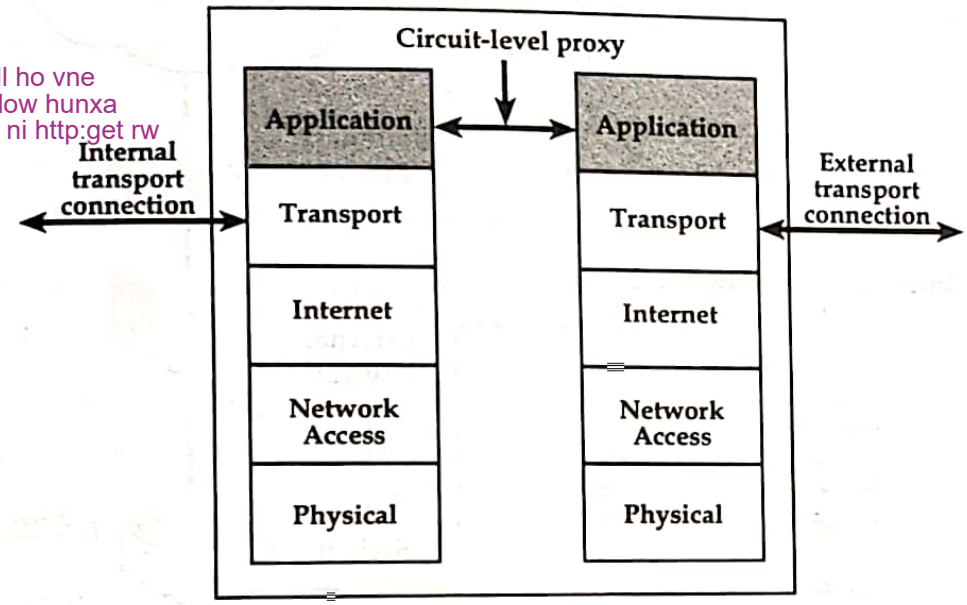


Figure 6.17: Application proxy firewall

4. Stateful Multilayer Inspection Firewalls

Stateful multilayer inspection firewall is a combination of all the firewalls that we have studied till now. They can filter packets at network layer using ACLs, check for legitimate sessions on the session layers and they also evaluate packets on the application layer (ALG). Stateful multilayer inspection firewall can work on a transparent mode allowing direct connections between the client and the server which was earlier not possible. Stateful multilayer inspection firewall can also implement algorithms and complex security models which are protocol specific, making the connections and data transfer more secure.

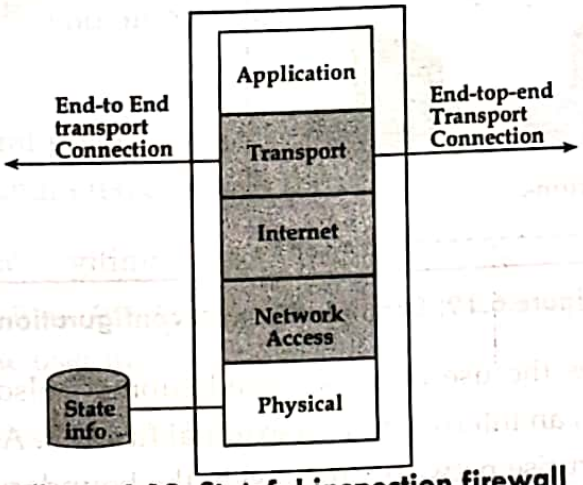


Figure 6.18: Stateful inspection firewall

DEMILITARIZED ZONE (DMZ) NETWORKS

Portion of network separating purely internal network from external network which allows control of accesses to some trusted systems inside the corporate perimeter is called DMZ network. If DMZ systems breached, internal systems will be still safe and can perform different types of checks at boundary of internal, DMZ networks and DMZ, Internet network.

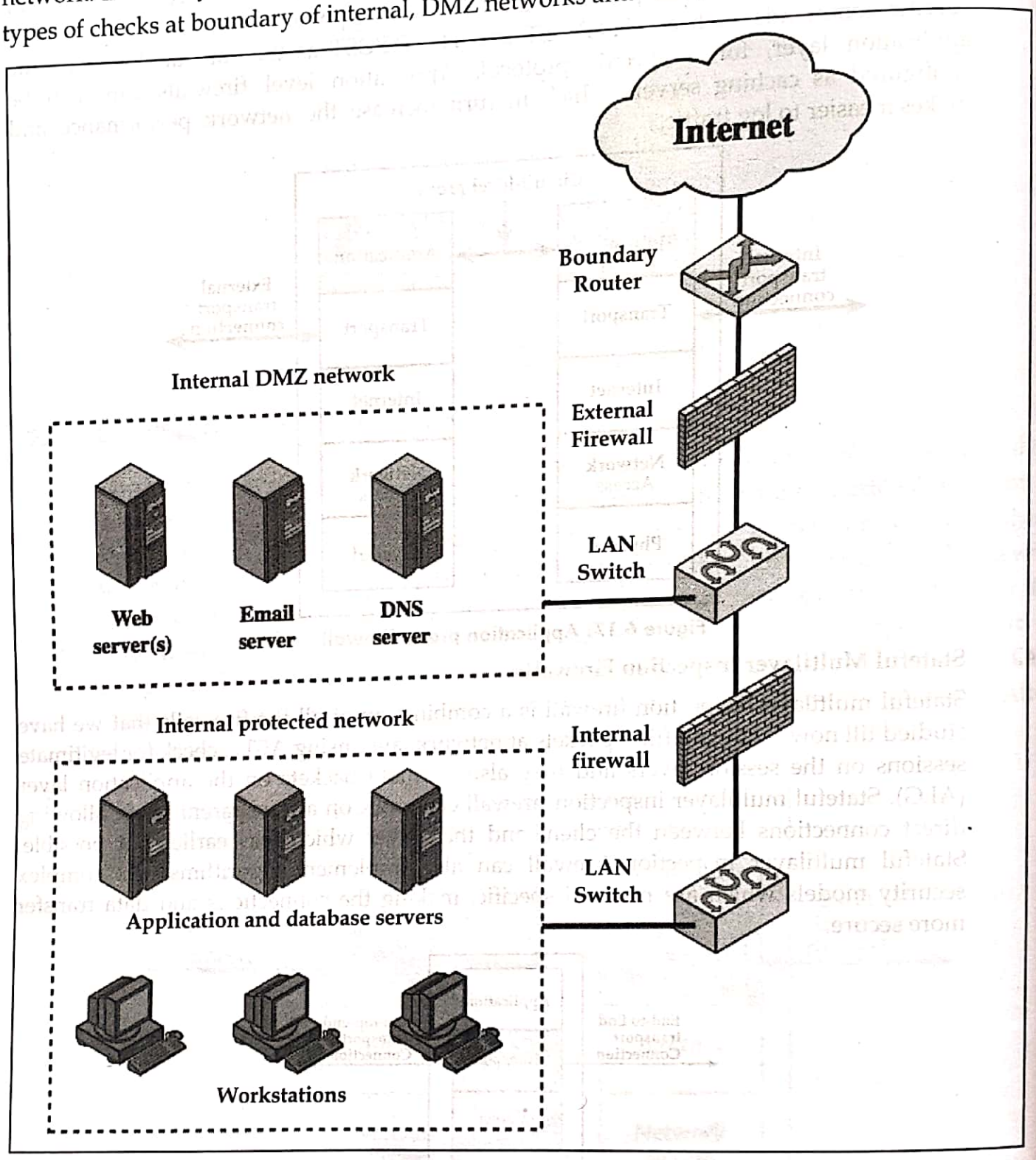


Figure 6.19: Example firewall configuration

Figure 6.19 further illustrates the use of a "screened subnet", also known as a demilitarized zone (DMZ), located between an internal and an external firewall. An external firewall is placed at the edge of a local or enterprise network, just inside the boundary router that connects to the

Internet or some wide area network (WAN). One or more internal firewalls protect the bulk of the enterprise network. Systems that are externally accessible but need some protections are usually located on DMZ networks. Typically, the systems in the DMZ require or foster external connectivity, such as a corporate Web site, an e-mail server, or a DNS (domain name system) server. The external firewall provides a measure of access control and protection for the DMZ systems consistent with their need for external connectivity. The external firewall also provides a basic level of protection for the remainder of the enterprise network. In this type of configuration, internal firewalls serve three purposes:

1. The internal firewall adds more stringent filtering capability, vs the external firewall, to protect enterprise servers and workstations from external attack.
2. The internal firewall provides two-way protection with respect to the DMZ, as it protects the remainder of the network from attacks launched from DMZ systems, and protects DMZ systems from attack by internal hosts.
3. Multiple internal firewalls can be used to protect portions of the internal network from each other.

A common practice is to place the DMZ on a different network interface on the external firewall from that used to access the internal networks.

ADVANTAGES OF FIREWALL

Firewalls have a number of advantages:

- They can stop incoming requests to inherently insecure services, e.g. you can disallow rlogin, or RPC services such as NFS.
- They can control access to services.
- They are more cost effective than securing each host on the corporate network since there are often only one or a few firewall systems to concentrate on.

DISADVANTAGES OF FIREWALL

Firewalls are not the be all and end all of network security. They do have some disadvantages, such as:

- They are a central point for attack, and if an intruder breaks through the firewall they may have unlimited access to the corporate network.
- They may restrict legitimate users from accessing valuable services, for example, corporate users may not be let out onto the Web, or when working away from home a corporate user may not have full access to the organization's network.
- They do not protect against back door attacks.

DISCUSSION EXERCISE

1. Define network security. Describe the different types of network security in brief.
2. Discuss the differences between digital certificates and digital signatures in authentication.
3. Why is PKI so vital in modern communications? Explain.
4. What are the core components of a PKI? Briefly describe each component.
5. Discuss the PKIX management function.
6. Differentiate between Kerberos and PKI.
7. What does PGP stand for? What is it used primarily for? And what are the five services provided by the PGP protocol?
8. How is the sender authentication carried out in PGP?
9. PGP has been a very successful communication protocol. Why is this so? What features brought it that success?
10. We say that SSL/TLS is not really a single protocol, but a stack of protocols. Explain. What are the different protocols in the SSL/TLS stack?
11. What is the role of the SSL record protocol in SSL/TLS? Explain.
12. Differentiate between SSL Session and SSL Connection.
13. How SSL record protocol provides confidentiality and message integrity?
14. Discuss five benefits of IPSec as a security protocol.
15. What services are provided by IPSec?
16. Differentiate between the transport mode and tunnel mode of IPSec. Is one mode better than the other? Under what conditions would you use one over the other?
17. List the major security services provided by AH and ESP, respectively.
18. Where does IPSec reside in a protocol stack?
19. Define firewall. List three design goals for a firewall.
20. Discuss the types of firewall.
21. What is the difference between a packet filtering firewall and a stateful inspection firewall?
22. What are limitations of the firewall?
23. What is DMZ network and what types of system would you expect to find on such networks?
24. Write the pros and cons of firewall.

❑❑❑