M.Sc (Computer Science)

II year – III Semester

COMPUTER NETWORK SECURITY AND CRYPTOGRAPHY

Handled by

Dr.K.Shanmugavadivu

Internet Protocol (IP)

IP Addressing

IP uses a numeric routing system to identify subnets and specific addresses. Each IP address contains 4 bytes. For example, the IP address is 10.1.3.100.

Subnet Classes:

The three main categories of subnets are class-A, class-B, and class-C. A*class-A* subnet contains 1 bit to designate a class-A style address, 7 bits for the network,and the remaining 3 bytes as unique identifiers (*B.C.D*).

Class	High-Order Bits	Subnet Range	Unique Nodes per Subnet
A	0	0.*x.x.x* – 127.*x.x.x*	16,777,216
B	10	128.*n.x.x* – 191.*n.x.x*	65,536
C	110	192.*n.n.x* – 223.*n.n.x*	256
D	1110	224.*n.n.n* – 239.*n.n.n*	special; multicast Packets[RFC3171]
E	11110	240.*n.n.n* – 247.*n.n.n*	reserved [RFC3330]
F	111110	248.*n.n.n* – 251.*n.n.n*	special;extended addressing[RFC1365]
others	111111	252.*n.n.n* – 255.*n.n.n*	reserved

n identifies part of the subnet. *x* identifies a byte for the unique identifier.

The *class-B* subnet is defined by having the first 2 bits "10." The next 14 bitsidentify the subnet, and the remaining 2 bytes define the unique identifier. Thereare 16,384 unique class-B subnets. Class-B subnets are commonly described by thesubnet bytes *A.B*, with the unique host denoted by *C.D*.
Class-C subnets begin with the 3 bits "110" and use the next 22 bits for theunique subnet. Although there are 4,194,304 unique class-C subnets, each onlycontains 256 unique addresses. Class-C subnets are commonly described by thefirst 3 bytes. For example, the host 192.168.15.2 is in the 192.168.15 subnet.

Network Masks

A subnet can be identified based on a combination of the network address and a*network mask*. A network mask, or *netmask*, is a set of 4 bytes that are combinedusing a bitwise-AND to define the

subnet (shown in below). If two addresses, whencombined with a netmask, yield the same value, then both addresses are on thesame subnet.

	IP Address	**Binary Format**
Address:	**10.1.5.100**	**00001010.00000001.00000101.01100100**
Netmask:	**255.0.0.0**	**11111111.00000000.00000000.00000000**
Combine Address with Netmask Using a Bitwise-AND Subnet:	**10.0.0.0**	**00001010.00000000.00000000.00000000**

Broadcast Addresses

The IP network protocol supports broadcast addressing. Packets with a destinationaddress of 255.255.255.255 are treated as local broadcast packets

Routable Broadcast Addresses

Along with the 255.255.255.255 broadcast address, each subnet contains a local broadcast address. The last IP address in a subnet is typically used as a broadcast address. For example, the subnet 192.168.0.0/16 has the broadcast address192.168.255.255.

Determine Subnet Size:

Thesubnets may appear as class-A, class-B, or class-C, the actual size of the subnet may vary. An attacker can use broadcast pings to determine the size of a remote subnet. By increasing the netmask by 1 bit and determining the relative broadcast address, a ping scan can determine the likely size of the subnet.

To prevent this type of scan from revealing all systems on a network, mostrouters respond to the echo request but do not forward the packet to the entire subnet.

Routing:

Each system in an IP network maintains a routing table that associates a network addressand netmask with a network interface. When a packet is being transmitted, the destination IP address is compared with the subnet associated with each interface. When a match is found, the data is transmitted through the associated interface.

The gateway is a router that will relay packets to the desired subnet. Gatewaysare used when a remote network is not directly reachable from thetransmitting host.Each routing table contains a *default gateway*.

Metrics:

Routing tables do not require uniqueness. Two different table entries may route tothe same subnet. This overlap permits fail-over redundancy—when one route isunavailable, the next one is attempted.

When a packet is ready for transmission, the route with the lowest metric is attempted first. If there is a transmission failure, then the next path is attempted.When two paths have identical metrics, the decision may be resolved throughround robin, load balancing, first listed, or default route selection.

TTL:

IP networks may contain very long routes between two subnets. IP networks may also contain loops. Long routes may be desirable, but loops can quickly consume all network bandwidth. To avoid packets being indefinitely forwarded through loops, IP uses a *time-to-live* (TTL) counter.

The TTL is decremented each time the packet passes through a router. If the TTL reaches zero, then the destination is considered to be unreachable and the packet is discarded. For IP, the TTL is 1 byte, with a maximum value of 255. Subnets connected by more than 254 routers are unreachable.

Nonroutable Addresses:

RFC1918 defines three sets of reserved network addresses: 10.0.0.0/8, 172.16.0.0/12, and192.168.0.0/16. These addresses are intended for private networks. Other nonroutable addresses include extended addresses, such as 250.0.0.0/8, and loopback addresses (Net127), such as 127.0.0.1.

ICMP:(Internet Control Message Protocol)

ICMP providessupport for testing, flow control, and error handling.

Type	Name	Service	Purpose
8/0	Echo Request/Reply	Testing	Determine connectivity
11	Time Exceeded	Testing/Error	Determine if the TTL expired
3	Destination Unreachable	Error	Identify failed delivery
4	Source Quench	Service	Basic flow control
5	Redirect	Service	Indicate more desirable route
17/18	Mask Solicitation	Service	Identify subnets
9/10	Router Advertisement	Service	Identify viable routers

ICMP Support

Each ICMP packet contains three parts. The first part indicates the type of ICMPpacket (e.g., ICMP_UNREACH, 0x03), numeric code for the packet type, and checksumfor the ICMP header. The second part contains code type specific data. The finalpart contains packet data.

For example, a host may support echo requests but not support router advertisement. Other examples include the following:

ICMP Checksums: RFC791 defines the ICMP checksum (Listing 12.2), but does not specify the behavior if the checksum is invalid.

Undefined Codes: Each type of ICMP packet may include codes for definingsubtypes.

Broadcast Addresses: Some hosts handle broadcast addresses differently. Anecho request sent to the 255.255.255.255 broadcast address may receive a differentset of replies than a request sent to a specific subnet (e.g., 10.255.255.255in the 10.0.0.0/8 subnet).

Echo Request and Reply:

A host will transmit an echo request (*ping*) to a remote system. The remote systemreceives the request and generates an echo reply (*pong*) back to the callingsystem. If the transmitting system does not receive a pong, then the remote systemmay be offline or unavailable. Similarly, the delay between the ping and pong can beused to measure network latency.

Ping Scans:

Ping scans, or *ping sweeps*, can be used to search subnets for systems that are online.

An alternate form of a ping sweep uses an ICMP echo request sent to a broadcastnetwork address for the destination. Rather than scanning an entire subnet, asingle ICMP echo request is broadcasted to the network, and every host on the networkreplies.

Ping of Death:

The echo request includes a payload for testing network capacity and MTU. Thispermits a network administrator to identify whether network issues are related topacket size. For example, small packets may be delivered, but large packets may fail.

A vulnerability was discovered where an ICMP payload could be larger than the expected MTU. Dubbed the *Ping of Death*, the oversized packet would cause systems to overflownetwork stack space, which resulted in a system crash or hang.

Smurf Attack:

A *Smurf attack* specifies a victim's IP address as thedestination and then sprays thousands of systems with ICMP echo requests. Eachsystem that receives the echo request sends an echo reply to the victim. Even thougha single echo reply is a small packet, thousands can easily overwhelm a

network. Becausethe true source of the attack is not associated with the packet, the victim isoverwhelmed by a DoS from an anonymous attacker**.**

Time Exceeded

ICMP provides error-handling support for IP. One such error concerns undeliverablepackets. If the TTL of a packet reaches zero before delivery, the router is supposedto drop the packet and transmit an ICMP error message to the source. Thereare two types of TTL expiration messages: *TTL count exceeded* and *TTL reassemblyexpired*. A TTL count exceeded indicates that the TTL counter decremented to zerobefore being delivered. The TTL reassembly expiration error indicates that fragmentsbegan to expire before all parts were received.

Destination Unreachable

Each unreachable packet includes a code describing the reason for the error. The Table lists the defined return codes.

Code	Purpose
UNREACH_NET	Network unreachable
UNREACH_HOST	Host unreachable; implies network is reachable
UNREACH_PROTOCOL	OSI transport layer protocol unavailable
UNREACH_PORT	OSI transport layer port unavailable
UNREACH_NEEDFRAG	Fragmentation required, but the "don't fragment"flag was set
UNREACH_SRCFAIL	Source routing failed
UNREACH_NET_UNKNOWN	Network is unknown
UNREACH_HOST_UNKNOWN	Host is unknown
UNREACH_ISOLATED	Network or host is isolated
UNREACH_NET_PROHIB	Access to the network is prohibited
UNREACH_TOSNET	Type of service (TOS) is unavailable for the network
UNREACH_TOSHOST	TOS is unavailable for the host
UNREACH_FILTER_PROHIB	Prohibited filtering
UNREACH_HOST_PRECEDENCE	Unreachable due to host precedence; indicates low priority host
UNREACH_PRECEDENCE_CUTOFF	Unreachable due to precedence cut-off; indicates low priority traffic

Unreachable Packet Format

Each ICMP packet contains three parts. The first part indicates the type of ICMP packet (e.g., ICMP_UNREACH, 0x03), the numeric code for the packet type, and the checksum for the ICMP header. The second part contains code type specific data. For unreachable packets, this is an unused field. The final part contains information that relates to the packet. For unreachable packets, this is the header from the failed packet.

Unreachable Attacks

Attackers that monitor IP traffic can create ICMP unreachable packets and transmit them to the acket source.

Unknown Unreachable

ICMP destination unreachable packets are intended to inform a host when a packet cannot be delivered. Yet, there are several situations

TTL Exceeded: Different hosts use different default TTL values. The TTL onthe ICMP packet may expire before being delivered.

Firewalls: Many firewalls and routers do not forward ICMP traffic. The errorreply may become filtered.

Undetected Failure: The packet may become lost in transit due to a transmissionerror or routing failure.

Source Quench

If the transmitting system is sending packets too quickly, the receiving system may send a *source quench* request. This request informs the sender to "send slower." Source quench has a number of limitations:

No Specified Rate: The source quench request does not specify a transmission rate.

No Specified Duration: The source quench does not include a duration for the request

No Send Faster: Although source quench is used to request a slower transmission rate, there is no support for increasing the transmission rate.

No Authentication: An attacker can flood an established connection with forged source quench requests. The result is an extremely slow network connection.

Redirect

When a network contains multiple routers, one route may be more desirable than another. Routers have the option of providing ICMP redirect messages to a host. The four types of supported redirection are for a network, host, type of service(TOS) for a network, and TOS for a host.

Mask Solicitation

ICMP provides a service for one host to request the network mask from anotherhost. Defined in RFC950, a new host on the network may transmit an ICMP masksolicitation.

Router Advertisement

Router discovery provides a way for nodes to discover neighboring routers. Describedin RFC1256, each ICMP router advertisement lists the networks reachablefrom the sender. Hosts may passively receive router advertisements or may activelysend an address request. An address request asks all routers to reply.

SECURITY OPTIONS:

There are a few steps that can mitigate risk exposure.These include disabling unnecessary features, using nonroutable IP addresses,filtering IP traffic, and employing security-oriented protocols when possible.

Disable ICMP

RFC791 describes ICMP as providing testing, flow control, and error handling, none of the functions provided by ICMP are essential for network routing.

Nonroutable Addresses

Hosts that do not need to be directly connected to the Internet can use nonroutable network addresses. RFC1597 defines a set of subnets that are not routable. The subnets are designated for use on private networks.

Nonroutable Subnets

Address Range	Scope
10.0.0.0 - **10.255.255.255**	Class-A
172.16.0.0 - **172.31.255.255** 16	Class-B
192.168.0.0 - **192.168.255.255** 256	Class-C

Network Address Translation (NAT)

Network address translation (NAT) is an OSI layer 4 (transport layer) service that relays packets from an isolated network through a common gateway [RFC1631, RFC2663]. The NAT gateway is a dual-homed system bridging the Internet (outside) with a private network (inside).

For each outbound packet, the NAT server stores a mapping in an internal relay table. The table contains (1) the source's IP address and transport layer port, (2) the destination's IP address and transport layer port, and (3) a port number for the NAT server's external interface. The packet is then relayed to the outside network using the NAT server's external IP address and port number. The

external destination views the packet as coming from the NAT server and not from the internal system.

When the NAT server receives a packet on the outside network interface, it compares the packet's destination with the internal relay table. The packet is then forwarded to the internal source.

Reverse-NAT (RNAT)

Reverse-NAT (RNAT) is an OSI layer 4 protocol that provides a static mapping from an external port on the NAT server to an internal port and IP address on the private network. Using RNAT, an external connection to port 80 (HTTP) on the NAT server can be routed to a Web server on the private subnet.

IP Filtering

Filter rules can restrict packets based on specific hosts, subnets, or type ofservice (e.g., TCP, UDP, or ICMP). This filtering can apply to the source or destinationaddress to provide maximum control over IP support.

Egress Filtering

Egress filtering applies firewall rules to outbound traffic. The simplest egress filtersoperate at the network layer by preventing packets with forged (or misconfigured)sources addresses from exiting the network. More complicated egressfiltering can be performed at higher OSI layers

Sample Egress Filtering by OSI Layer

Layer	**Type of Filtering**
7 (Application)	Web URL restrictions (e.g.,anti-porn filters);email/spamfiltering; virus canning
6 (Presentation)	VPN and SSL authentication
5 (Session)	Restrictive DNS and LDAP access
4 (Transport)	Port access restrictions (e.g., forbid IRC or chatroom software)
3 (Network)	Host and subnet restrictions based on network address;inbound and egress filtering
2 (Data Link)	Host based on source hardware address
1 (Physical)	None

IPsec

The main security issues around IP concern authentication, validation, and privacy:

No Authentication: Any host can transmit an IP packet and make the packet appear to come from another host; there is no authentication.

No Validation: IP packets use a simple checksum to validate the content's integrity. If corruption occurs at the network, data link, or physical layers, the checksum will likely identify the error. But the checksum does not provide protection against an intentional modification. An attacker can intercept a packet, modify the contents, and apply a valid checksum.

No Privacy: All IP data is transmitted without any encoding.

For example,RFC2402 describes an authentication header for IP packets. RFC2409 defines a keyexchange system. *IPsec* is a collection of security standards for the network layer. Bydesigning IPsec as a collection of standards, new security options can be added, and
less-secure solutions can be removed.

IPv6

IPv6 contains a different (and optimized) header format aswell as some significant improvements.

Address Space: IPv6 increases the address space from 32 bits to 128 bits. Thismitigates the impact from poor IPv4 subnet allocations.

Broadcast Routing: For IPv4, multicast packets and remote network broadcastingwere afterthoughts. As such, most IPv4 routers do not support relayingIPv4 multicast packets. In contrast, IPv6 natively supports routing multicastand broadcast packets.

Integrated Security: IPv6 includes functionality for authenticating and encryptingconnections at the network layer.

Integrated VPN: IPv6 natively supports encapsulation, which allows networktunneling. Together with encryption and authentication, IPv6 provides a fullVPN solution.

Transport Layer

Common Protocols

The two most common transport protocols are *Transmission Control Protocol*(TCP) and *User Datagram Protocol* (UDP).

TCP and UDP are almost universally supported. The main distinction comes fromtheir connection management: TCP is a connection-oriented protocol, whereasUDP is for connection-less communications.

UDP is connection-less [RFC768]. UDP transmits data,but does not validate that the data was received. Because UDP does not wait for acknowledgements,it generally transmits data much faster than TCP; however, UDPdoes not detect lost packets. UDP is preferred by many application layer protocols that provide audio and video, such as VoIP and Microsoft NetMeeting.

SCTP

the *Stream Control Transmission Protocol* (SCTP) addresses these shortcomings [RFC2960, RFC3286,RFC3309]. For example, SCTP uses a sliding window for packet acknowledgements. Rather than periodically stopping transmission and waiting for an acknowledgement,an acknowledgement window is provided. Large data streams cancontinue transmitting while receiving periodic acknowledgements. This providesconnection-oriented functionality with a lower overhead than TCP. SCTP includes support for packet authentication and encryption.

NetBIOS-DG

File-sharing protocols, such as NFS, use UDP for data transfers. The *NetBIOS Datagram Service* (NetBIOS-DG) provides similarfunctionality to UDP. NetBIOS-DG is used for transporting data between Windows systems.NetBIOS-DG is an open protocol defined by RFC1001 and RFC1002

AppleTalk

Apple's operating systems incorporate the AppleTalkprotocol. AppleTalk includes a variety of transport protocols that are optimizedfor specific tasks. For example, the AppleTalk Data Stream Protocol (ADSP)is optimized for streaming data, whereas the Name Binding Protocol (NBP) is used for hostname resolution. Other AppleTalk transport protocols include the AppleTalk Transaction Protocol (ATP) and Echo Protocol (AEP).

Transport Layer Functions.

This includes connection management, packet assembly, and serviceidentification. The two core elements that make these functions possible are transportlayer ports and sequencing.

Ports and Sockets

An *active socket* indicates an established networkconnection, either on a client or server. In contrast, a *passive socket* is one thatis not in use. Servers use passive sockets while awaiting and listening for networkconnections.

The transport layer creates *ports*. Each port contains a unique identifier for aspecific higher-layer service. A single socket may host many ports, but a port requiresa socket. Ports may be associated with specific higher-layer protocols or dynamicallyallocated. For example, the Web (HTTP) uses a service on port 80 of theTCP protocol—denoted as 80/tcp. Email uses 25/tcp, and DNS uses both 53/udpand 53/tcp.

Sequencing

The transport layer receives a block of data from the upper layers and separates thedata into *packets*. Each packet is assigned a unique *sequence identifier* that is used totrack the packet. The transport layer maintains a list of sequence numbers that have been used on a port—new sequence numbers are

not used until the entire sequenceis used. This prevents a packet with a duplicate sequence number frombeing confused with a previously seen packet.

Sequence Hijacking

An attacker who observes a transport layer packet must identify the sequence to insertor hijack the connection. If the attacker sends forged packets that are out of sequence,then they will likely be ignored by the target system; however, forgedpackets with valid sequence numbers can appear authentic and acceptable to thetarget system.

Random Sequence Numbers

This deters an attacker from guessing the first packet; however, randomsequence numbers within an established connected-oriented transport connectionare uncommon.

Blind Sequence Attacks

Sequence numbers are commonly started with aninitial random value but increment throughout the course of the session. An observercan easily identify the sequence pattern and hijack the session, but a blind attackercannot identify the initial sequence number.

Sequence Reconnaissance Attacks

An attackercan establish many transport connections and determine the pattern for initial sequence numbers. This weakness allows a blind attacker to potentiallycompromise transport layer connections.

Hijacking Mitigation Options

Most services that require securityrely on higher OSI layer protocols to authenticate connections. For example,SSH connections can be hijacked through TCP, but the hijacker is unlikely to beable to authenticate or respond correctly to SSH packets.

Connection Management

transport layer protocols may be connection-oriented orconnection-less. In a *connection-oriented* protocol, the recipient acknowledges eachdata transfer; *connection-less* protocols do not.

A single transport layer packet may be split into multiple networklayer packets, resulting in multiple network-layer acknowledgements for eachtransport layer packet.

The four types of connectionorientedconfirmations are per packet, per set, sliding window, and implicit. Eachof these approaches balances tradeoffs between performance and reliability.

Per-Packet Confirmation

The simplest type of connection-oriented service uses *per-packet confirmation*. Eachtransmitted packet receives an acknowledgement.

Per-Set Confirmation

In *per-set confirmation*, a group of packets may be transmitted with one acknowledgementaccounting for the entire group.

Sliding Window Confirmation

The *sliding window confirmation* approach extends the concept of per-set confirmation.Rather than transmitting a block of data and then waiting for a reply, the slidingwindow continually transmits data. Acknowledgements are sent periodically andconfirm a set of transmitted packets.

Implicit Confirmation

Not every connection-oriented protocol is explicitly connection-oriented. Someprotocols, such as those that use cryptography, may be designated as connectionlessbut actually provide connection-oriented acknowledgements.

Packet Ordering

The transport layer may divide data into multiple packets, combine small datablocks into larger packets, and multiplex packets through multiple connections. the network layer routes the packets to remote systems but does notguarantee the order of the packet's arrival. Different routes may result in differentpacket ordering. To ensure that the upper OSI layers receive data as it was sent, thetransport layer uses the packet sequence numbers to ensure data ordering.

Keep-Alive

Network and lower OSI layers may connect and reconnect as needed, without impactingthe transport layer. Network devices such as NAT routers, however, may decide a transport layer connection is terminated when no traffic isseen after a few minutes. NAT devices and firewalls normally store a table of addresses,ports, and protocols that are permitted to enter the network. After a timeoutfrom lack of use, table entries are dropped.

GATEWAYS

Gateways are transport layer devices that can pass data between different network layer protocols. For example, a gateway may link traffic from an IPv6 network to anX.25 network.

TCP

CONNECTION-ORIENTED PROTOCOL

TCP is a connection-oriented protocol and offers two guarantees for higher layerprotocols: reliable data delivery and sequential data ordering. As a transport layerprotocol, TCP also provides facilities for flow control and multiple connectionsTCP uses a segmented header to enclose transmission data.

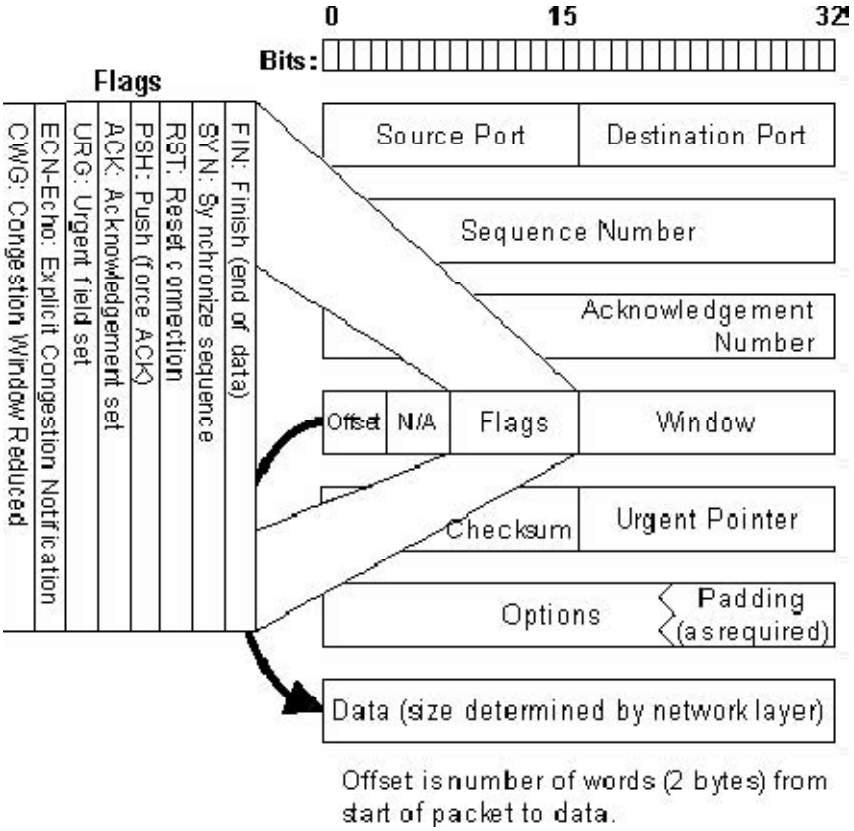

Fig:A basic TCP header

Besidesthe basic 20-byte header, TCP also may include optional header information. The smallest TCP packet (containing no data and no options) is 20 bytes, whereasthe largest can exceed 64 KB. Usually a TCP packet with data is only a few kilobytesin size, which ensures that one TCP packet can fit within one data link layer frame.

Reliable Acknowledgements:

TCP guarantees reliable data delivery. Each byte sent by a TCP packet is confirmed with an acknowledgement. If no acknowledgement is received, then it assumes that the data transfer failed.

1 Checksums

TCP encases the application data within a TCP header. The TCP header includes a simple checksum value to detect transmission corruptions. The checksum functionis a 16-bit CRC function.

2 **Sequence Numbers:**

TCP guarantees the order of application data delivery. Although lower layer protocolsmay transmit and receive packets out of order, TCP will reconstruct the ordering before passing the data up the stack to an application. The reconstruction is based on sequence numbers. Each TCP packet is

assigned a sequence number, and subsequent packets increase the sequence number by the amount of data transported.

3 **Flow Control:**

TCP manages data flow using control flags and windows. These two methods combine to provide transport state, acknowledgements, priority, and optimized buffering.

Control Flags:

Each TCP packet header contains 12 bits for use as control flags. One flag denotes an acknowledgement, another identifies a reset request, and a third defines urgent data. In total, there are six primary control flags (Table 15.1). The flags do not need to be used symmetrically.

Name	Bit	Purpose
FIN	0x001	Finish; denotes the end of a connection
SYN	0x002	Synchronize; denotes the start of a connection
RST	0x004	Reset; requests a connection reset
PSH	0x008	Push; require recipient to passed all data up the stack and send an acknowledgement
ACK	0x010	Acknowledge; sender acknowledges receipt of data
URG	0x020	Urgent; data in this packet takes precedence over regular data transfers and handling
ECE	0x040	Explicit Congestion Notification – Echo; allows notification when packets are dropped
CWR	0x080	Congestion Window Reduced; notifies recipient when less window space is available
Remainder	0xF00	Reserved

Window Size

In a basic connection-oriented system, each transmission is acknowledged. Theacknowledgement confirms data delivery, but creates significant overhead and delays.For example, if the transport layer needs to pass 100 KB of data, it may segmentthe data into 100 packets containing 1 KB each. Each of the 100 packets wouldrequire a confirmation—another 100 packets. To reduce the amount of bidirectionalcommunication, TCP allows each packet header to define the amount of datathat can be transferred before requiring an ACK. If the client specifies a window size

of 8,192 bytes, then the server can send up to 8,192 bytes of data before requiringan ACK. For large transfers, larger window sizes can be used;

Multiple Connections and Ports

The source and destination port numbers are combined with the source and destinationnetwork addresses to define each TCP connection. Data from one set ofports and addresses (one connection) is not intermixed with data from another setof ports and addresses. This allows the host system to manage many connections atonce.

A TCP server binds to a specific port. For a client to connect to the server, itmust first acquire its own local port number. Then it connects to a server and establishesa unique transport session defined by the network client and server addressesas well as the transport client and server ports.

TCP CONNECTIONS

TCP maintains three basic states: initialization, data exchange, and disconnect.

TCP Connection Initialization

Initialization uses a *three-way handshake*: SYN, SYN-ACK, and ACK.The client initially sends a TCP packet containing the SYN flag to a specific port on the server. The packet contains the initial sequence number that will be used by the client. The server replies with both SYN and ACK flags set (SYN-ACK). The server sets its initial sequence number and acknowledges the client's sequence number.

TCP Data Exchange

After establishing a TCP connection with the three-way handshake, data packetsmay be transmitted. Each data packet includes an ACK flag and the data. When theclient sends data to the server, the server increments the client's sequence number(a counter) and acknowledges the data with an ACK packet. The server may includeresponse data along with the data acknowledgement.

TCP Disconnect

TCP identifies the end of a connection when one side transmits a packet containing a f*inish flag* (FIN). The closure is completed when the recipient transmits an ACK for the FIN. As with the SYN packets, the FIN packets are not expected to transmit data. If any data is transmitted after the FIN, then the recipient responds with a reset (RST).

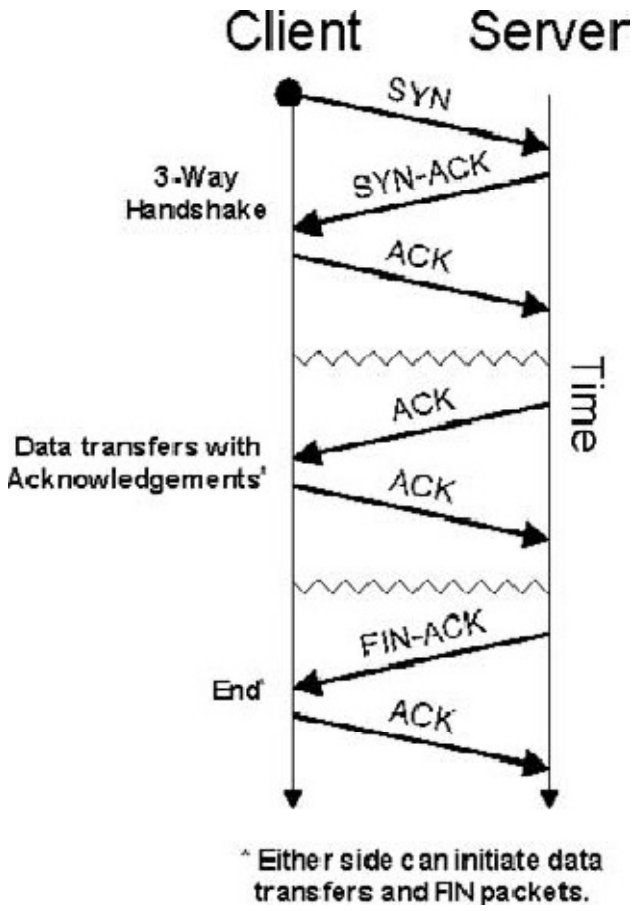

FIGURE: TCP three-way initialization handshake, data transfer, and connection termination.

UDP

The *User Datagram Protocol* (UDP) is a simplified transport protocol commonly used for connection-less services. UDP packets have a simplified header. The header only includes a source port, destination port, data length, and checksum.

UDP transmissions do not generate acknowledgements. UDP requires significantly less network bandwidth than TCP, it is more prone to attack. UDP attacks are usually based on unvalidated packets and impersonation.

Unvalidated Inbound Sources

UDP servers perform no initial handshake. Any host can connect to a UDP server,and the connection is not authenticated.

UDP Hijacking

UDP servers receive packets from any hosts without authentication. As a result, any client can send a packet to any UDP server. UDP is very vulnerable to hijacking. An attacker can easily forge the correct networkaddresses and UDP ports to insert data into the recipient.

UDP Keep-Alive Attack

UDP has no clear indicators for open or closed connections.As a result, most firewalls open UDP ports when the first outbound connectionis seen but do not close the port until after a period of inactivity. Attackers canuse this weakness to hold open UDP ports.

UDP Smurf Attack

UDP Smurf Attack, a forged packet is sent to a UDP server. The attackerforges the victim's network address as the packer sender. The server responds bysending one or more UDP packets to the victim.

UDP Reconnaissance

UDP offers few options for system profiling and reconnaissance. Because UDP has no window size, sequence number, or header options, these cannot be used to profile the system. UDP port scans rely on ICMP and packet replies.

If no UDP service exists on a scanned port, then an ICMP "Destination unreachable" packet is returned. The only way to defeat this type of port scan is to not return any ICMP packets. This makes no reply indistinguishable from no service.

Session Layer:

SESSION STATE MACHINE

Each state within the session layer corresponds with a functional primitive. These well-defined *primitives* determine the available functionality. Adding to the complexity of the finite state machine, primitives are commonly referenced by cryptic abbreviations.

Not every session layer protocol implements every state associated with the session layer. Rather, only states used by a protocol are required. For example, DNS uses very short-duration sessions, so major and minor synchronization requests are not needed.

Connection States

The OSI session layer defines three basic connection states: connect, release, andabort.

Connect: A new session layer connection is established. A single session layermay control multiple session.

Release: This terminates an established session.

Abort: At any time, the finite state machine may enter an invalid state. Anabort message abruptly terminates an established session.

Session Network Usage

A single session layer connection may span multiple transport layer (and lowerlayer) connections. Sessions may change transport protocols, network addresses, orroutes without interfering with the session.

Sequential Network Connections: Each session may choose to end (*drop*) atransport layer connection at any time. A connection is establishedwithin the timeframe required by the session layer, lower layers maydynamically connect and disconnect as needed. Dropping and re-establishing connections may permit lower layers to optimize routing performance.

ParallelNetwork Connections:

Parallelconnections may provide faster performance than a single network connection.

Communication States

After establishing a session between two hosts, data may be transferred and states may be modified. Any data transfer or state change can be accepted or rejected. To prevent a total transaction loss, the session layer provides two levels of synchronization: major and minor

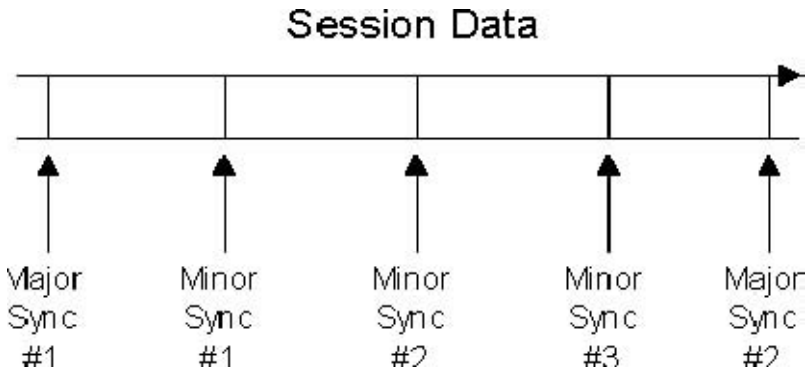

FIGURE Major and minor synchronization points within an established session dialog.

Major Synchronization: Major synchronization permits confirmation of largesession transactions within a single dialog. All transactions within the majorsynchronization are independent of transactions before and after the synchronizationpoint. Each major point must be explicitly confirmed.

Minor Synchronization: Minor synchronization permits confirmation of actionsubsets within a major synchronization block.

Resynchronization: Resynchronization permits a session to abandon or restart a dialog

segment.

Flow Control Functionality

The session layer provides two different methods for controlling data flow.

The first method focuses on data transfers and processing rates.

The second approach manages bidirectional communications.

The session layer defines two types of data transfer: normal and expedited. Anexpedited data transfer permits user data or state management to bypass normal routing. Expedited data sent after normal data may be processed first.

Quality-of-Service Functionality

The session layer includes many functional primitives for providing a desirable quality-of-service level.

Timing Delays:

permitted connection, reconnection, transit, and release settings.

Throughput: The session layer provides a facility for flow control by regulatingnetwork throughput.

Priority: Different transport layer connections may be assigned different prioritylevels. This can lead to optimal data transfers.

Timer: The timer can be set to ensure that a confirmation occurs within a specified timeframe.

Session Layer Functionality:

The session layer extends these concepts across multiple network connections.It is capable of providing flow control and full-duplex/half-duplex supportregardless of the lower-layer capabilities.

SESSIONS AND STACKS:

The session layer is uniquely associated with the ISO OSI network stack.

Sessions and TCP/IP

stack predates the OSI model and does not separate out sessionorientedfunctionality. Every network application based on the TCP/IP stack mustmanage sessions independently, and few applications manage sessions the sameway. For example, Web sites use a variety of methods for maintaining sessions, includingthe following:

Cookies: Cookies are strings generates by a Web server and passed to abrowser. The browser then returns the cookie with each subsequent Webrequest. If the server sees the same cookie, then it knows it is the same session.

Basic Authentication: Web pages can employ simple authentication schemes,such as Basic Authentication, where a username and password is transmittedwith each request.

Complexity: Each application must define and manage its own session information.

This increases the complexity of the applications and results in moreopportunities for exploitation.

Vulnerability: If there is an implementational vulnerability in a lower-levelprotocol, such as IP or UDP, then a single patch fixes the problem.

Sessions and OSI

The OSI model's session layer permits consistency and reuse between different network services.

Name Services

The text-based hostnames are associatedwith network addresses. This allows hosts to be easily remembered. Because IP usesIP addresses and not hostnames, a system must exist to map hostnames to IP addresses

Remote Procedure Calls

A *remote procedure call* (RPC) is a way to use a networked host as an extension ofthe local system. Procedures and functions are executed on remote hosts ratherthan locally.

Remote File Systems

A remote server can *share* a file, directory, or partition to hosts onthe network. This establishes an accessible remote file system. The remote file systemcan then be *mounted* by a networked host and accessed.

Remote file systems have many uses

Limited Disk Space: Remote file systems permit storing data when the localsystem has limited disk space.

Collaboration Space: In work environments, data is commonly shared betweengroups or departments. Remote file systems permit many users to shareaccess to common data.

Random Access: Not everyone (nor every application) needs all data in a commondirectory. Rather than attempting to identify the needed files, remote filesystems permit listing files and transferring their contents on an as-needed basis.

Authentication Systems

The session layer is ideal for user authentication. Each established session indicatesauthenticated communication. The session layer can also be used to encrypt dataand provide session-based privacy.

Other session layer authentication systems include Kerberos and DCE. Both of these systems use cryptographic-based services to provideauthentication tickets for service access. Each *ticket* corresponds to an authenticatedsession.

SSL FUNCTIONALITY

SSL provides the means to negotiate algorithms. SSL operates in threedistinct phases: negotiation, key exchange, and data transfer.

SSL Connection

The first phase of the SSL protocol establishes network connectivity. Because SSLoperates at the presentation layer, a network socket must be established through thelower layers. To track the connection state, SSL maintains a *context*. The context is an abstractdata structure that stores session, negotiation, and data transfer information.

SSL Negotiation

the SSL client and server negotiate cryptographicalgorithms. The client transfers a list of supported cipher combinations.

The list is transmitted in priority order—the first cipher is more desirable than thesecond cipher, the second is more desirable than the third, and so on. The cipher combinations specify four cryptographic elements: SSL version, keyexchange, encryption cipher, and validation algorithm.

SSL Version

There are many versions of the SSL protocol. Although SSLv1 (version 1) is seldomused, SSLv2 and SSLv3 are common.

SSL Key Exchange and Authentication Algorithm

Many key exchange algorithms are supportedby SSL. The most common are the following:

Diffie-Hellman key exchange (DH)

RSA (named after its authors: Rivest, Shamir, and Adleman)

Digital Signature Algorithm (DSA) from the Digital Signature Standard (DSS)

Encryption Cipher

The negotiated cipherset specifies an encryption cipher. The selection includes the cipher (DES,3DES, RC2, RC4, and AES) and a choice of bit sizes and block chaining algorithms.

Null Cipher

The *null cipher* is another example of an insecure option. This cipher performsno authentication and no encryption—data is transmitted unencoded. The null cipheris usually used for testing and debugging.

Validation Cipher

For explicit validation of data integrity, SSL supports MD5 and SHA1digests.

SSL Key Exchange

The *key exchange* permits authentication between the client and server. Thismitigates the threat from a MitM attack. When using certificates or preshared keys,the exchange also validates the identity of the client and server.

SSL Data Transfer

the algorithm negotiation and key exchange, both ends may transfer applicationdata.

SSL Communication Flow

Step 1: The client sends a hello record that includes the version of SSL beingUsed

Step 2: The server responds with the selected cipher set and either a server certificate (*server-side certificate*) or part of a key exchange.

Step 3: If the server provides a certificate, then the client contacts a certificate authority (CA) and validates the certificate.

Step 4: The client responds to the server, initiating a key exchange. If a clientside certificate is available, then the client may provide it.

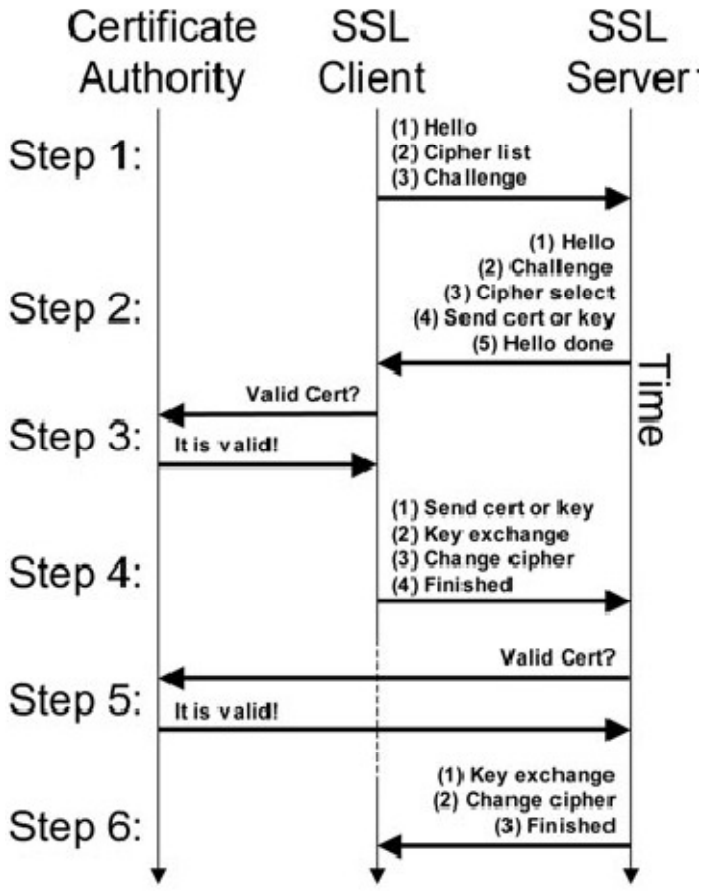

FIGURE SSL initialization flow between client and server

Step 5: If the client provides a certificate to the server, then the server may contacta CA to authenticate the client.

Step 6: The server sends an encrypted handshake to complete the validationprocess.

SSL includes a few out-of-band signals for managing the encrypted tunnel.

These include the hello record for initiating the handshake, a cipher change signalto begin using symmetrical encryption, and a finished signal to denote the end ofan SSL connection.

CERTIFICATES in SSL

CERTIFICATES

SSL's strength comes from endpoint validation. Through the use of digitalcertificates, the client can authenticate the server and vice versa.

SSL supports two types of certificates.

Server-side certificates are stored on the server

Client-side certificates are stored on client

X.509 Certificates

X.509 certificate follows a well-defined format and provides information foridentification and validation [RFC2459]. Each X.509 certificate defines three components:identification, public key, and private key.

X.509 Identification

The X.509 identification contains field-value pairs that provide information aboutthe certificate. The data is stored in the ASN.1 format. Some of the certificate fieldsform a *distinguished name* (DN) that identifies the certificate.

Some of the commonDN attributes : CN -Common name O- Organization OU Organization unit

X.509 Public and Private Keys

SSL uses the X.509 certificate as a container for transferring keys in a *Public Key Infrastructure* (PKI). The PKI defines a set of public and private keys.

The two main certificate validation algorithms are RSA and DSS.

Server Certificates

Server-side certificates are used to validate the server to the client. The server sendsa public key (or in the case of DSS, a public key digest) to the client. It is up to theclient to validate the certificate.

A trusted *certificate authority* (CA) is usually not the samehost as the system being validated, so it can validate the certification.

Client Certificates

Client-side certificates are used to authenticate the client to the server. Very securesystems use these to ensure that only authorized clients access the OSI applicationlayer.

Certificate Authority (CA)

A CA is a system that can validate a certificate. Although the CA server may behosted by the certificate's domain (or on the same server as the certificate), they areusually hosted on separate systems or in remote domains.

To prevent a hijacker from impersonating a CA server, most servers presharetheir own public keys. A client validating a server-side certificate will encode the certificate with the CA server's public key.

A trusted CA server may perform any of three actions:

Accept: The certificate is found to be valid. The CA server responds with anacceptance.

Reject: The certificate is invalid, and the submitter is asked to reject the certificate.

This usually happens for expired certificates, but invalid certificatescan also lead to rejections.

Revoke: In the event that a certificate is compromised,

Certificate Generation

To validate a certificate, the CA server must know about the private certificate thatcontains the private key. This is a multistep process. First, the server generates apublic and private key pair. After generating system keys, a *certificate-signing request* (CSR) is generatedand passed to the CA server.

SSH AND SECURITY

The SSH protocol addresses each of the basic security concepts

Confidentiality: Each SSH connection is encrypted, preventing an eavesdropperfrom viewing information. For added security, SSH periodically reexchangeskeys to ensure that a compromise of one set of keys does notcompromise the entire session.

Authentication: Before establishing a connection, the client must authenticatethe server *and* the server must authenticate the client. Client authentication canbe any combination of certificates (keys), passwords, or digital tokens. AlthoughSSH is usually used with one-part authentication (a password or a key),it can support two- and three-part authentication systems. The server only usesa certificate to authenticate with the client.

Authorization: SSH limits the traffic that can enter the tunnel and can restricthow data exits the tunnel. For remote login access, SSH restricts authorization of the user's login privileges

Integrity: SSH uses encryption and cryptographic checksums to validate eachpacket.

Non-repudiation: Each packet is cryptographically signed using an HMAC,ensuring that the data actually came from the sender.

SSH has three primary uses: establish a secure network tunnel, provide a VPNwith port-forwarding characteristics, and supply application-layer login functionality.

SSH PROTOCOL

The SSH protocol operates in five phases: algorithm negotiation, key exchange,server authentication, client authentication, and data transfer.

Phase 1: Algorithm Negotiation

SSH supports a variety of cryptographic algorithms. Upon the initialconnection, the systems negotiate the cryptographic algorithms that will be used. Different SSH versions and platforms support different algorithms.

The SSH algorithm negotiation is very similar to SSL. The client offers a list ofsupported cipher combinations, and the server responds with its selection. Thereare some significant differences between the negotiations for SSL and SSH:

Combinations

Weak Ciphers

Bidirectional Negotiation

Phase 2: Key Negotiation

The algorithm negotiation is immediately followed with a key exchange using theDiffie-Hellman (DH) algorithm. SSH performs key exchanges periodically—usuallyevery hour.

Phase 3: Server Authentication

SSH clients associate server keys with network addresses. Unfortunately,servers are occasionally reinstalled or assigned new addresses. The user must resolve theissue, and the resolution varies with different SSH clients:

Open SSH: User's manually edit their host-key cache files to remove the mismatched Server entry.

PuTTY: PuTTY stores the host-key cache in the Registry under HKEY_CURRENT_USER **Mocha Telnet:** Mocha Telnet does not

cache server keys. As a result, all servers are always accepted.

Phase 4: Client Authentication

The client may transmitany combination of a password, a challenge signed by the client key or a digitalidentifier for authentication. If the authentication works, then the client successfullyconnects. If the authentication fails, however, the client is disconnected.

Phase 5: Data Transfer

The SSH VPN permitsmultiple channels—similar to the transport layer's ports. Through concurrentchannels, SSH supports many simultaneous connections.

STMP

EMAIL GOALS:

SMTP was designed for the timely, but not immediate, delivery of text messages.This is different and distinct from other communication protocols.

SMTP was designed to be extendable. This flexible protocol can be easily extended to support most data formats, encryption systems, and some authentication.

SMTP Data Format

SMTP uses a simple file transfer protocol to deliver email. Each email message is afile, and each file contains three components: meta header, blank line, and content.The meta header consists of field: value pairs. These are used to identify recipients,subject, and routing information. These header entries are used for tracingemails and identifying delivery information.

SMTP Command Structure

SMTP defines a command and response environment for communicating betweena *mail user agent* (MUA—mail client) and *message transport agent* (MTA—mailserver). The MUA connects to the MTA and issues a set of commands based on theinformation in the data's email headers.

MTA Return Codes

MUA sends a command to the server, and the server responds with an acknowledgement.

The return codes follow a format originally defined for FTP. Each code has a three-digit identifier. The first digit tells the class ofcode. The second identifies the response category and the remainingdigit tells the exact code type.

Sending Email

To send email, the MUA must connect to the MTA and transmit a series of commands.

After each command, the MTA provides a status reply.

Command Exploitation

Many of the MTA commands can leak information, and attackers can use this information to identify potential weaknesses. Some commands are required, and others are optional. Other optional commands, such as VRFY and EXPN, are used to verify and expandemail addresses.

COMMON SERVERS

There are many MTA options, most mail servers on the Internet either use Sendmail, Microsoft Exchange, Qmail, or Postfix. Each of these mailer options has tradeoffs with regards to performance, features, and security.

Sendmail

The Sendmail MTA (*http://www.sendmail.org/*) is the single most common mailtransport agent. It is open source and provided as the default MTA distribution formost Unix systems. Sendmail is a master email daemon; it receives, processes, forwards,and delivers email. Sendmail is readily identifiable by the SMTP welcome message, HELP file identifier,and version information placed in the email header.

Microsoft Exchange

Similar to Sendmail, the Microsoft Exchange server is a monolithic server designedas an alternative to SMTP. Within the MTA, the system uses the proprietary *Exchange*data format, but it also supports communicating with standard SMTP MTAs.

Qmail

It is intended as a replacement for Sendmail. It was designedfor performance, reliability, implicity, and modularity. As a result, Qmail isvery fast and relatively easy to configure and maintain.

Postfix

It began as an alternative to Sendmail. In 1998, Webbasedapplication servers were modularly designed to handle high loads and limit theimpact from security exploits.Postfix employs dozens of special-purpose applications but maintains the samecommand-line options as Sendmail.

Postfix does not have the same security issuesas Sendmail. Instead, it implements many security-oriented concepts:

Least privileged: Each module runs with the minimum permissions necessary.

Insulation: Each module performs one task independently from other modules.

Controlled environment: No modules operate as other users.

No SUID: No Postfix module runs with “set user id” access. In Unix, SUID isused to change access privileges.

Trust: The concept of trust between modules is well defined and constantlyvalidated.

Large Inputs: Application memory is properly handled to mitigate risks fromoverflows.

HTTP GOALS:

HTTP was designed as a flexible, real-time data transfer protocol. The originalspecification for HTTP version 1.0 was defined in RFC1945.

HTTP Command-Reply

A command-reply protocol that uses meta information to augmentrequests and responses. In HTTP 1.0, each TCP connection contained up to four elements:

Command: A keyword command, such as GET or POST, followed by a resourceIdentifier.

Meta Information: A series of field: value pairs for augmenting the requestinformation.

Blank Line: A single blank line indicates the end of the HTTP meta header.

Optional Data: Most HTTP replies contain data provided after the blank line.

The data transfer is completed when the TCP connection is closed.

HTTP Command Structure

HTTP defines many different commands for managing the resource identifier andmeta information. The most common commands are used with the Web:

GET: The GET command indicates data retrieval from the server.

HEAD: The HEAD command is similar to GET, but it only returns the associatedmeta header, not the actual data.

POST:POST provides upload functionality.

other general purpose commands

PUT: specifies that the data be placed in the location specified by the resource identifier.

DELETE: Requests the removal of the resource identifier from theserver.

LINK: Create a relationship between a resource identifierand some other source.

UNLINK: removes a relationship established by the LINKcommand.

Return Codes

HTTP server replies with a status code. The statuscode follows a three-digit structure, similar to SMTP. The first digit identifies thetype of reply, and the remaining two digits denote the specific reply. Some HTTP attacks, the returned status code can be used for reconnaissance.

HTTP Extensions

HTTP is constantly evolving. HTTP 1.0 was releasedin 1996. In 1999, HTTP 1.1 was solidified as RFC2616. During this time, additionalmeta headers and commands were added to the protocol.

Maintaining State

HTTP request operates independently; there is no relationship betweensubsequent HTTP requests. Cookies raise encoding, privacy, and misuse issues.

HTTP 1.1 Extensions

HTTP 1.1 introduced many extensions to HTTP 1.0. HTTP 1.1 permits multiple commandsto be sent over a connection. HTTP 1.1 also provided data chunking. Data chunking allows the server to return parts of the reply without having the client expire theconnection.
Beyond GET, HEAD, and POST, HTTP 1.1 introduced commands for connectionmanagement. **CONNECT, TRACE and OPTIONS**

Basic Authentication:

HTTP 1.0 does define a minor amount of authentication. Called *Basic Authentication*, the server provides a *realm* (string) to the client. The client must return ausername and password combination for access to the realm. Other authentication methods, such as Digest authentication, useshared secret information and a cryptographic hash function to authenticate.

SSL Security

To provide security, HTTP is commonly tunneled through an SSL connection. HTTPS is usually implemented with server-side certificates but not client-sidecertificates.

Related Protocols

HTTP is commonly associated with the Web, it has also been extended toother protocols.

URL:

HTTP uses a *uniform resource locator* (URL) as a shorthand notation for specifyingthe type of query. This notation not only allows the identification of remoteservices and files but also leads to exploitable attack vectors.

URL Format:

URL defines a protocol, account information, server information, and a *uniform resource indicator* (URI) that specifies the exact instruction.
The actual format of a URL is
service://username:password@server:port/URI?options#anchor, but many of these fields are optional.

Service: This specifies the application layer protocol. HTTP, HTTPS, and FTP,

Server Information: After // indicate the starts. Only HTTP and HTTPS do require it.

Username:Password@: Used for login information.

Server: The server's address is specified for protocols such as HTTP and FTP.

Port: specified port numbers default port numbers for HTTP is 80/tcp.

URI: A slash separates the server information from the query information.The URL contains a specific directive for the service on the specifiedhost.

Query: The URI may end with an optional query, denoted by a questionmark (?). (&) to separate parameters.

Anchor: Under HTTP, the hash character (#) denotes an anchor withinan HTML document.

Dynamic Submission Methods

There are three common approaches for submitting information to a server.

The first approach uses the URL query field to pass variables and parameter informationto the HTTP server's CGI application.

A second approach uses the URI's path to indicate parameters to a CGI application.

The third approach places query information in the submitted data.