Computer Networks, Web

COMPASS CTF

June 30, 2022

Overview

What would happen when we connect to the Internet?

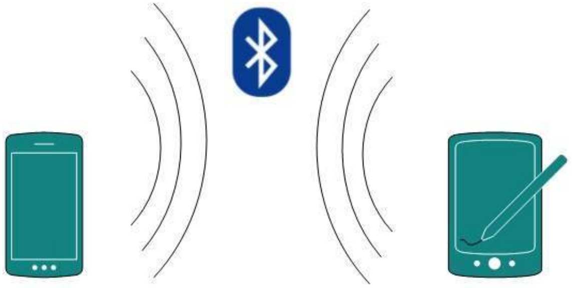
Have you thought about how our computer communicates with other computers?

When we open a website, how does HTTPS secure our device?

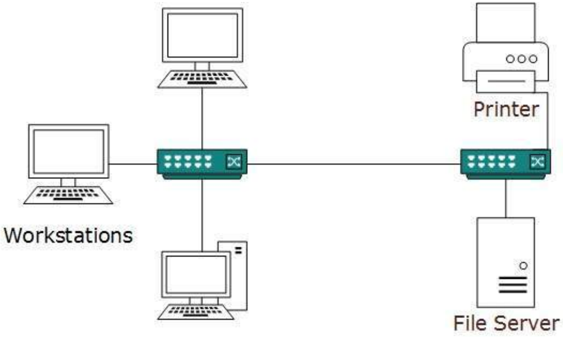
This is the topic we would cover today - Computer Networks and Websites.

Overview

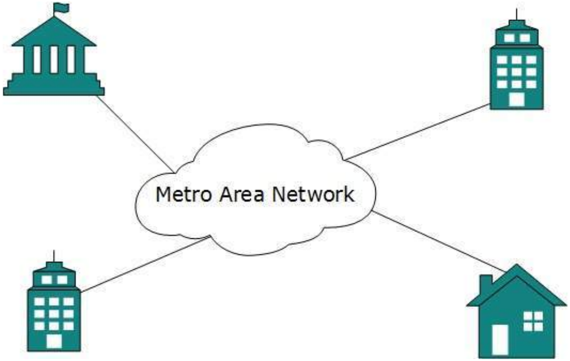
Let's have a view of those web types:



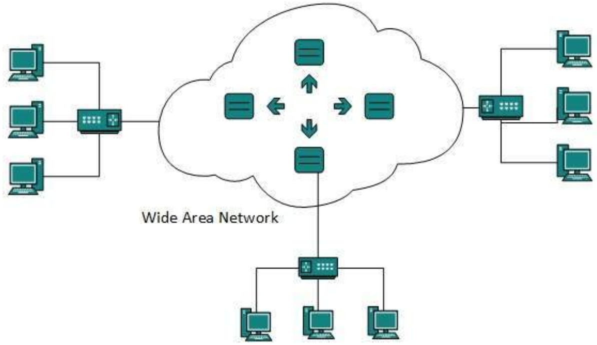
Overview



Overview



Overview



Overview

All those things are using network:

- Web sites
- E-mail
- Instant Messaging
- Blogging
- Social Media
- Marketing
- Networking
- Resource Sharing
- Audio and Video Streaming

Overview

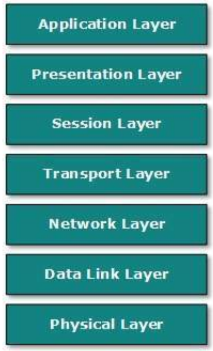
Things we would cover include the structure and the protocols of the web. From the computer networks to the high-level Internet web pages. This is a bottom-to-top tutorial.

Contents:

- Computer Networks
- Application Protocol: HTTP and Web
- Database: MySQL Overview
- Vulnerabilities in the Web
- Solving Web Challenges (basics)

OSI Model

Open System Interconnect is an open standard for all communication systems. OSI model is established by International Standard Organization (ISO). This model has seven layers:



Computer Networks

In the OSI Model, there are 7 layers:

- Application Layer: This layer is responsible for providing an interface to the application user. This layer encompasses protocols that directly interact with the user.
- Presentation Layer: This layer defines how data in the native format of the remote host should be presented in the native format of the host.
- Session Layer: This layer maintains sessions between remote hosts. For example, once user/password authentication is done, the remote host maintains this session for a while and does not ask for authentication again during that time.
- Transport Layer: This layer is responsible for end-to-end delivery between hosts.
- Network Layer: This layer is responsible for address assignment and uniquely addressing hosts in a network.
- Data Link Layer: This layer is responsible for reading and writing data from and onto the line. Link errors are detected at this layer.
- Physical Layer: This layer defines the hardware, cabling wiring, power output, pulse rate, etc.

Computer Networks

In fact, in the real network, the Session layer, the Presentation layer, and the Application layer are combined into one layer.

Some examples of the each layer:

- Application Layer: HTTP Protocl, FTP, NFS.
- Transport Layer: TCP, UDP.
- Network Layer: IP, ARP, ICMP.
- Data Link Layer: IEEE 802.2, PPP.
- Physical Layer: Ethernet (IEEE 802.3), RS-232.

Computer Networks

Computer Networks Security

During the initial days of the internet, its use was limited to military and universities for research and development purposes. Later when all networks merged and formed the internet, the data was used to travel through a public transit network. Common people may send data that can be highly sensitive such as their bank credentials, username, and passwords, personal documents, online shopping details, or confidential documents.

All security threats are intentional i.e. they occur only if intentionally triggered. Security threats can be divided into the following categories:

Interruption

Interruption is a security threat in which the availability of resources is attacked. For example, a user is unable to access its web-server or the web-server is hijacked.

Privacy-Breach

In this threat, the privacy of a user is compromised. Someone, who is not the authorized person is accessing or intercepting data sent or received by the originally authenticated user.

Integrity

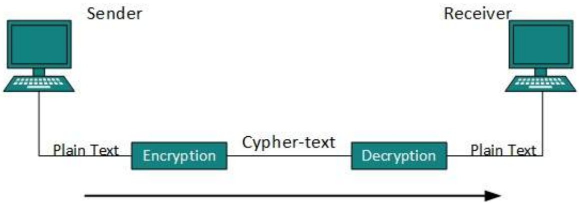
This type of threat includes any alteration or modification in the original context of communication. The attacker intercepts and receives the data sent by the sender and the attacker then either modifies or generates false data and sends it to the receiver. The receiver receives the data assuming that it is being sent by the original Sender.

Authenticity

This threat occurs when an attacker or a security violator, poses as a genuine person and accesses the resources or communicates with other genuine users.

Computer Networks

No technique in the present world can provide 100% security. But steps can be taken to secure data while it travels in an unsecured network or internet. The most widely used technique is Cryptography.



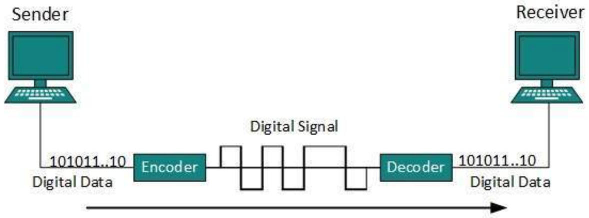
Cryptography is a technique to encrypt the plain-text data which makes it difficult to understand and interpret. There are several cryptographic algorithms available present day as described below:

- Secret Key
- Public Key
- Message Digest

Physical Layer - Digital Transmission

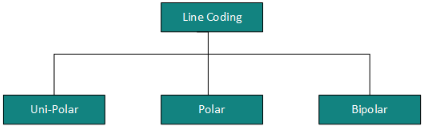
Line Coding

The process for converting digital data into a digital signal is said to be Line Coding. Digital data is found in binary format. It is represented (stored) internally as a series of 1s and 0s.



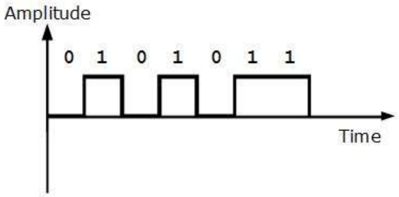
Computer Networks

Digital signal is denoted by discreet signal, which represents digital data.There are three types of line coding schemes available:



Uni-polar Encoding

Unipolar encoding schemes use a single voltage level to represent data. In this case, to represent binary 1, high voltage is transmitted, and to represent 0, no voltage is transmitted. It is also called Unipolar-Non-return-to-zero, because there is no rest condition i.e. it either represents 1 or 0.



Polar Encoding

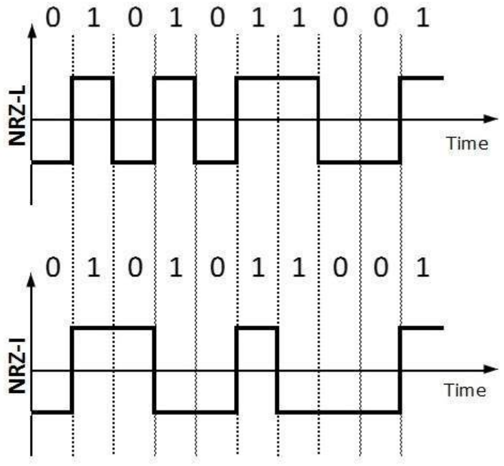
The polar encoding scheme uses multiple voltage levels to represent binary values. Polar encodings are available in four types:

Polar Non-Return to Zero (Polar NRZ)

It uses two different voltage levels to represent binary values. Generally, a positive voltage represents 1 and a negative value represents 0. It is also NRZ because there is no rest condition.

Computer Networks

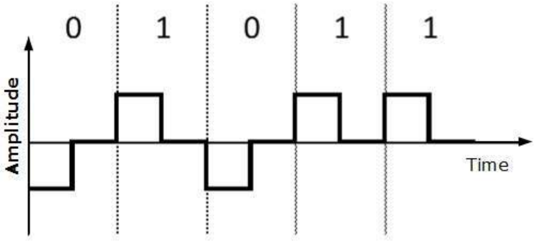
NRZ scheme has two variants: **NRZ-L** and **NRZ-I**.



NRZ-L changes voltage level when a different bit is encountered whereas NRZ-I changes voltage when a 1 is encountered.

Return to Zero (RZ)

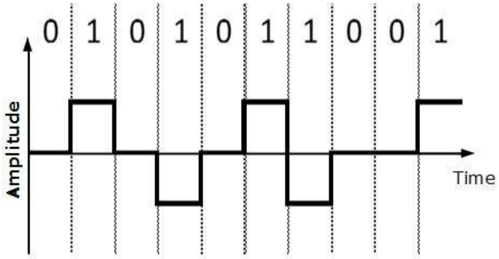
The problem with NRZ is that the receiver cannot conclude when a bit ended and when the next bit is started, in the case when the sender and receiver's clock are not synchronized.



RZ uses three voltage levels, positive voltage to represent 1, negative voltage to represent 0, and zero voltage for none. Signals change during bits not between bits.

Bipolar Encoding

The bipolar encoding uses three voltage levels, positive, negative, and zero. Zero voltage represents binary 0 and bit 1 is represented by altering positive and negative voltages.



Physical Layer - Analog Transmission

To send the digital data over an analog media, it needs to be converted into analog signal.There can be two cases according to data formatting.

Bandpass:The filters are used to filter and pass frequencies of interest. A bandpass is a band of frequencies which can pass the filter.

Low-pass: Low-pass is a filter that passes low frequencies signals.

When digital data is converted into a bandpass analog signal, it is called digital-to-analog conversion. When low-pass analog signal is converted into bandpass analog signal, it is called analog-to-analog conversion.

Digital-to-Analog Conversion

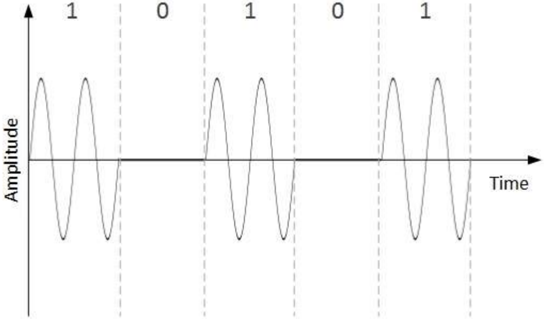
When data from one computer is sent to another via some analog carrier, it is first converted into analog signals. Analog signals are modified to reflect digital data.

An analog signal is characterized by its amplitude, frequency, and phase. There are three kinds of digital-to-analog conversions:

Computer Networks

Amplitude Shift Keying

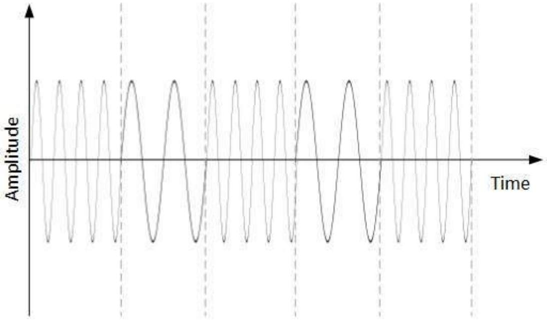
In this conversion technique, the amplitude of analog carrier signal is modified to reflect binary data.



When binary data represents digit 1, the amplitude is held; otherwise it is set to 0. Both frequency and phase remain same as in the original carrier signal.

Frequency Shift Keying

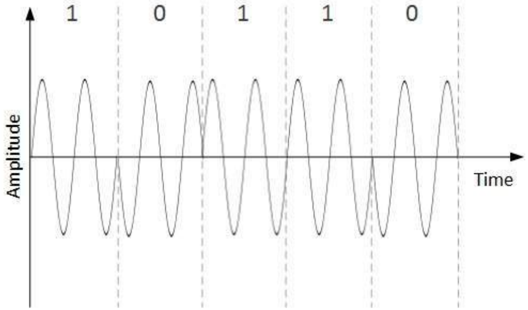
In this conversion technique, the frequency of the analog carrier signal is modified to reflect binary data.



This technique uses two frequencies, f1 and f2. One of them, for example f1, is chosen to represent binary digit 1 and the other one is used to represent binary digit 0. Both amplitude and phase of the carrier wave are kept intact.

Phase Shift Keying

In this conversion scheme, the phase of the original carrier signal is altered to reflect the binary data.



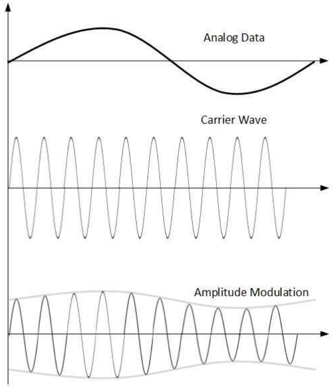
When a new binary symbol is encountered, the phase of the signal is altered. Amplitude and frequency of the original carrier signal is kept intact.

Analog-to-Analog Conversion

Analog signals are modified to represent analog data. This conversion is also known as Analog Modulation. Analog modulation is required when bandpass is used. Analog to analog conversion can be done in three ways:

Amplitude Modulation

In this modulation, the amplitude of the carrier signal is modified to reflect the analog data.

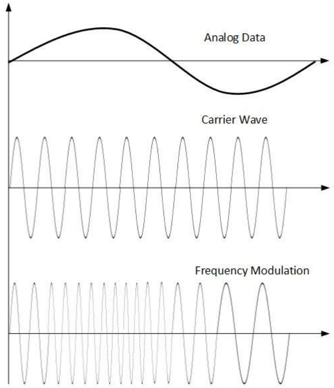


Amplitude modulation is implemented by means of a multiplier. The amplitude of modulating signal (analog data) is multiplied by the amplitude of carrier frequency, which then reflects analog data.

The frequency and phase of carrier signal remain unchanged.

Frequency Modulation

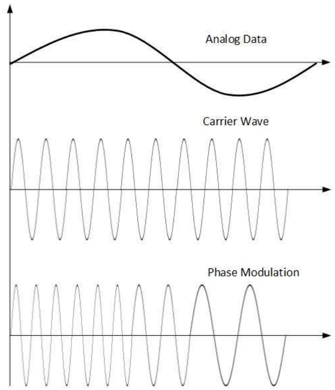
In this modulation technique, the frequency of the carrier signal is modified to reflect the change in the voltage levels of the modulating signal (analog data).



The amplitude and phase of the carrier signal are not altered.

Phase Modulation

In the modulation technique, the phase of carrier signal is modulated in order to reflect the change in voltage (amplitude) of analog data signal.



Phase modulation is practically similar to Frequency Modulation, but in Phase modulation frequency of the carrier signal is not increased. Frequency of carrier is signal is changed (made dense and sparse) to reflect voltage change in the amplitude of modulating signal.

Computer Networks

Network Switching

Switching is process to forward packets coming in from one port to a port leading towards the destination. When data comes on a port it is called ingress, and when data leaves a port or goes out it is called egress. A communication system may include number of switches and nodes. At broad level, switching can be divided into two major categories:

Connectionless: The data is forwarded on behalf of forwarding tables. No previous handshaking is required and acknowledgements are optional.

Connection Oriented: Before switching data to be forwarded to destination, there is a need to pre-establish circuit along the path between both endpoints. Data is then forwarded on that circuit. After the transfer is completed, circuits can be kept for future use or can be turned down immediately.

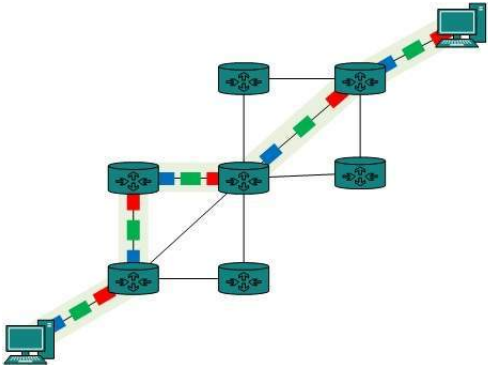
Circuit Switching

When two nodes communicate with each other over a dedicated communication path, it is called circuit switching.There 'is a need of pre-specified route from which data will travels and no other data is permitted.In circuit switching, to transfer the data, circuit must be established so that the data transfer can take place.

Circuits can be permanent or temporary. Applications which use circuit switching may have to go through three phases:

- Establish a circuit
- Transfer the data
- Disconnect the circuit

Computer Networks



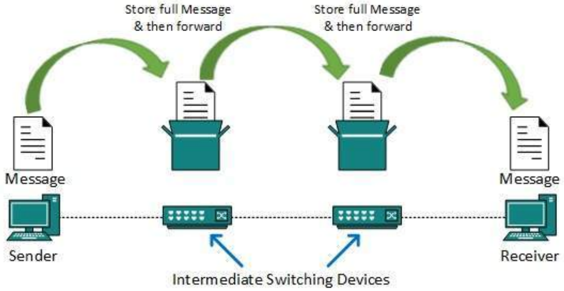
Circuit switching was designed for voice applications. Telephone is the best suitable example of circuit switching. Before a user can make a call, a virtual path between caller and callee is established over the network.

Message Switching

This technique was somewhere in middle of circuit switching and packet switching. In message switching, the whole message is treated as a data unit and is switching / transferred in its entirety.

A switch working on message switching, first receives the whole message and buffers it until there are resources available to transfer it to the next hop. If the next hop is not having enough resource to accommodate large size message, the message is stored and switch waits.

Computer Networks



Store full Message & then forward
Store full Message & then forward
Message
Message
Sender
Receiver
Intermediate Switching Devices

This technique was considered substitute to circuit switching. As in circuit switching the whole path is blocked for two entities only. Message switching is replaced by packet switching. Message switching has the following drawbacks:

Every switch in transit path needs enough storage to accommodate entire message.

Because of store-and-forward technique and waits included until resources are available, message switching is very slow.

Message switching was not a solution for streaming media and real-time applications.

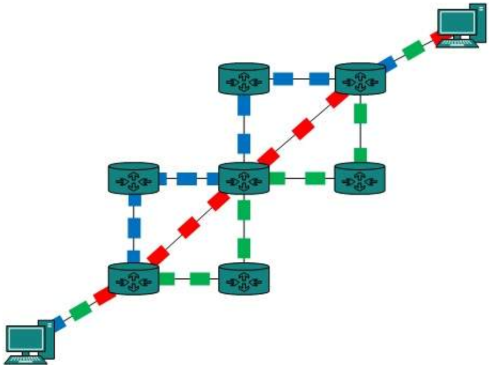
Computer Networks

Packet Switching

Shortcomings of message switching gave birth to an idea of packet switching. The entire message is broken down into smaller chunks called packets. The switching information is added in the header of each packet and transmitted independently.

It is easier for intermediate networking devices to store small size packets and they do not take much resources either on carrier path or in the internal memory of switches.

Computer Networks



Packet switching enhances line efficiency as packets from multiple applications can be multiplexed over the carrier. The internet uses packet switching technique. Packet switching enables the user to differentiate data streams based on priorities. Packets are stored and forwarded according to their priority to provide quality of service.

Data-link Layer

Data Link Layer is second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layer hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.

Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast. Systems on broadcast network are said to be on same link. The work of data link layer tends to get more complex when it is dealing with multiple hosts on single collision domain.

Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer.

Data link layer has two sub-layers:

- Logical Link Control: It deals with protocols, flow-control, and error control
- Media Access Control: It deals with actual control of media

Functionality of Data-link Layer

Data link layer does many tasks on behalf of upper layer. These are:

Framing

Data-link layer takes packets from Network Layer and encapsulates them into Frames.Then, it sends each frame bit-by-bit on the hardware. At receiver' end, data link layer picks up signals from hardware and assembles them into frames.

Addressing

Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

Synchronization

When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

Computer Networks

Error Control

Sometimes signals may have encountered problem in transition and the bits are flipped.These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

Flow Control

Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machine to exchange data on same speed.

Multi-Access

When host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides mechanism such as CSMA/CD to equip capability of accessing a shared media among multiple Systems.

Data-link Control and Protocols

Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

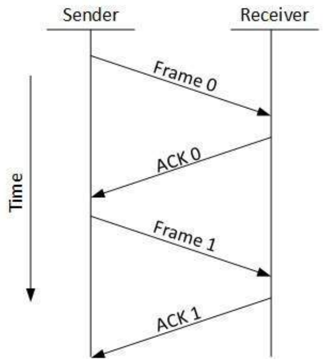
Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

Stop and Wait

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



Computer Networks

Sliding Window

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Computer Networks

Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

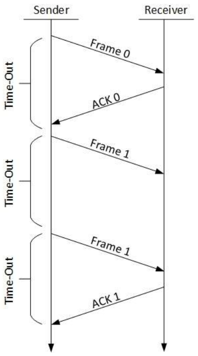
Requirements for error control mechanism:

- Error detection - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
- Positive ACK - When the receiver receives a correct frame, it should acknowledge it.
- Negative ACK - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
- Retransmission: The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

Computer Networks

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

Stop-and-wait ARQ



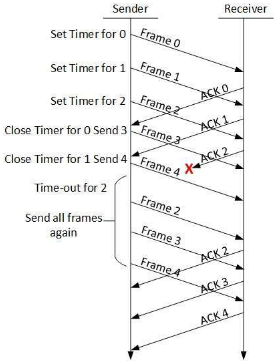
Computer Networks

The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

Go-Back-N ARQ

Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.

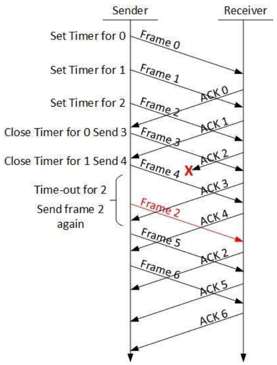


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

Computer Networks

Selective Repeat ARQ



In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

Network Layer

Layer-3 in the OSI model is called Network layer. Network layer manages options pertaining to host and network addressing, managing sub-networks, and internetworking.

Network layer takes the responsibility for routing packets from source to destination within or outside a subnet. Two different subnet may have different addressing schemes or non-compatible addressing types. Same with protocols, two different subnet may be operating on different protocols which are not compatible with each other. Network layer has the responsibility to route the packets from source to destination, mapping different addressing schemes and protocols.

Layer-3 Functionalities

Devices which work on Network Layer mainly focus on routing. Routing may include various tasks aimed to achieve a single goal. These can be:

- Addressing devices and networks.
- Populating routing tables or static routes.
- Queuing incoming and outgoing data and then forwarding them according to quality of service constraints set for those packets.
- Internetworking between two different subnets.
- Delivering packets to destination with best efforts.
- Provides connection oriented and connection less mechanism.

Network Layer Features

With its standard functionalities, Layer 3 can provide various features as:

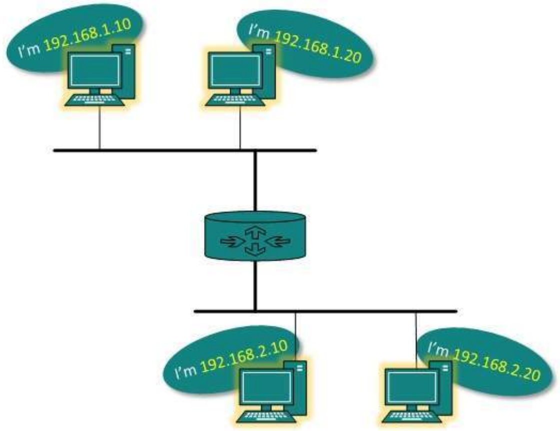
- Quality of service management
- Load balancing and link management
- Security
- Interrelation of different protocols and subnets with different schema.
- Different logical network design over the physical network design.
- L3 VPN and tunnels can be used to provide end to end dedicated connectivity.

Internet protocol is widely respected and deployed Network Layer protocol which helps to communicate end to end devices over the internet. It comes in two flavors. IPv4 which has ruled the world for decades but now is running out of address space. IPv6 is created to replace IPv4 and hopefully mitigates limitations of IPv4 too.

Computer Networks

Network Addressing

We are discussing IP here as it is the only one we use in practice these days.



Network Layer Routing

When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing. Routing is done by special network devices called routers or it can be done by means of software processes.The software based routers have limited functionality and limited scope.

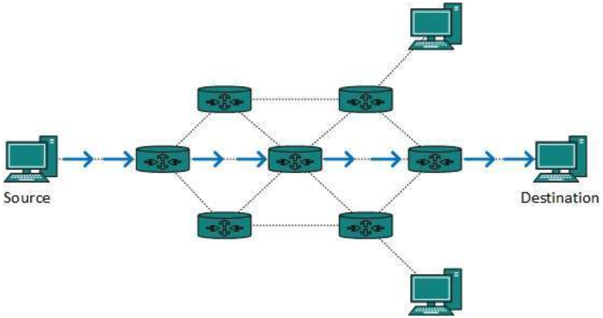
A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple path existing to reach the same destination, router can make decision based on the following information:

- Hop Count
- Bandwidth
- Metric
- Prefix-length
- Delay

Routes can be statically configured or dynamically learnt. One route can be configured to be preferred over others.

Unicast routing

Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination. Routing unicast data over the internet is called unicast routing. It is the simplest form of routing because the destination is already known. Hence the router just has to look up the routing table and forward the packet to next hop.



Broadcast routing

By default, the broadcast packets are not routed and forwarded by the routers on any network. Routers create broadcast domains. But it can be configured to forward broadcasts in some special cases. A broadcast message is destined to all network devices.

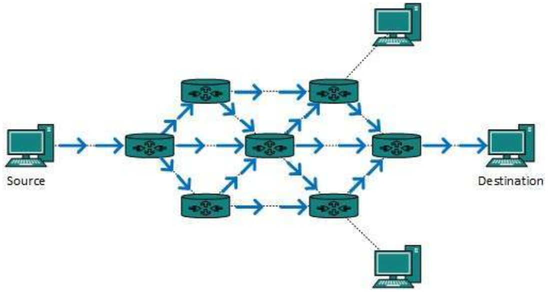
Broadcast routing can be done in two ways (algorithm):

A router creates a data packet and then sends it to each host one by one. In this case, the router creates multiple copies of single data packet with different destination addresses. All packets are sent as unicast but because they are sent to all, it simulates as if router is broadcasting.

This method consumes lots of bandwidth and router must destination address of each node.

Secondly, when router receives a packet that is to be broadcasted, it simply floods those packets out of all interfaces. All routers are configured in the same way.

Computer Networks

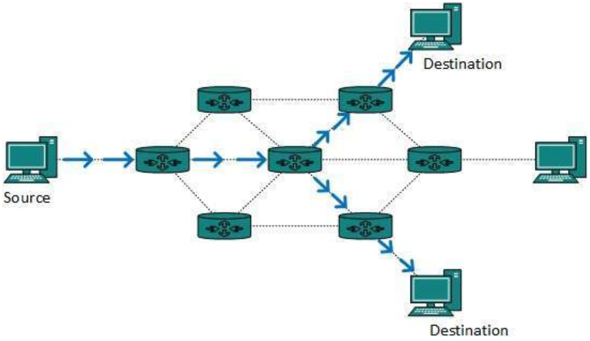


This method is easy on router's CPU but may cause the problem of duplicate packets received from peer routers.

Reverse path forwarding is a technique, in which router knows in advance about its predecessor from where it should receive broadcast. This technique is used to detect and discard duplicates.

Multicast Routing

Multicast routing is special case of broadcast routing with significance difference and challenges. In broadcast routing, packets are sent to all nodes even if they do not want it. But in Multicast routing, the data is sent to only nodes which wants to receive the packets.



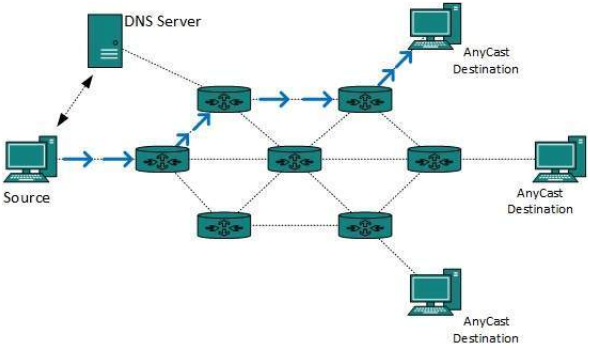
Computer Networks

The router must know that there are nodes, which wish to receive multicast packets (or stream) then only it should forward. Multicast routing works spanning tree protocol to avoid looping.

Multicast routing also uses reverse path Forwarding technique, to detect and discard duplicates and loops.

Anycast Routing

Anycast packet forwarding is a mechanism where multiple hosts can have same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in routing topology.



Anycast routing is done with help of DNS server. Whenever an Anycast packet is received it is enquired with DNS to where to send it. DNS provides the IP address which is the nearest IP configured on it.

Network Layer Protocols

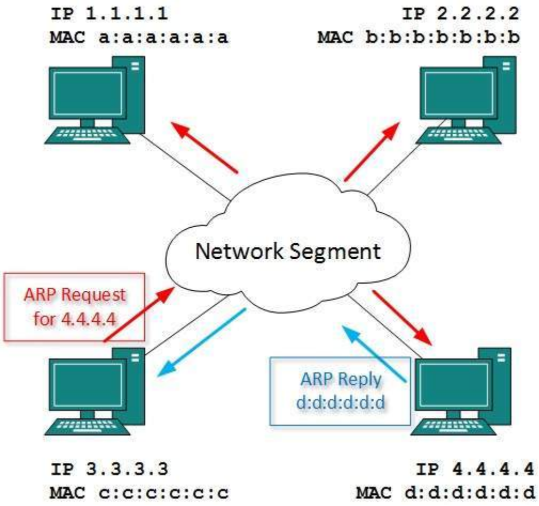
Every computer in a network has an IP address by which it can be uniquely identified and addressed. An IP address is Layer-3 (Network Layer) logical address. This address may change every time a computer restarts. A computer can have one IP at one instance of time and another IP at some different time.

Address Resolution Protocol(ARP)

While communicating, a host needs Layer-2 (MAC) address of the destination machine which belongs to the same broadcast domain or network. A MAC address is physically burnt into the Network Interface Card (NIC) of a machine and it never changes.

On the other hand, IP address on the public domain is rarely changed. If the NIC is changed in case of some fault, the MAC address also changes. This way, for Layer-2 communication to take place, a mapping between the two is required.

Computer Networks



Computer Networks

To know the MAC address of remote host on a broadcast domain, a computer wishing to initiate communication sends out an ARP broadcast message asking, “Who has this IP address?” Because it is a broadcast, all hosts on the network segment (broadcast domain) receive this packet and process it. ARP packet contains the IP address of destination host, the sending host wishes to talk to. When a host receives an ARP packet destined to it, it replies back with its own MAC address.

Once the host gets destination MAC address, it can communicate with remote host using Layer-2 link protocol. This MAC to IP mapping is saved into ARP cache of both sending and receiving hosts. Next time, if they require to communicate, they can directly refer to their respective ARP cache.

Reverse ARP is a mechanism where host knows the MAC address of remote host but requires to know IP address to communicate.

Internet Control Message Protocol (ICMP)

ICMP is network diagnostic and error reporting protocol. ICMP belongs to IP protocol suite and uses IP as carrier protocol. After constructing ICMP packet, it is encapsulated in IP packet. Because IP itself is a best-effort non-reliable protocol, so is ICMP.

Any feedback about network is sent back to the originating host. If some error in the network occurs, it is reported by means of ICMP. ICMP contains dozens of diagnostic and error reporting messages.

ICMP-echo and ICMP-echo-reply are the most commonly used ICMP messages to check the reachability of end-to-end hosts. When a host receives an ICMP-echo request, it is bound to send back an ICMP-echo-reply. If there is any problem in the transit network, the ICMP will report that problem.

Computer Networks

Internet Protocol Version 4 (IPv4) IPv4 is 32-bit addressing scheme used as TCP/IP host addressing mechanism. IP addressing enables every host on the TCP/IP network to be uniquely identifiable.

IPv4 provides hierarchical addressing scheme which enables it to divide the network into sub-networks, each with well-defined number of hosts. IP addresses are divided into many categories:

- Class A - it uses first octet for network addresses and last three octets for host addressing
- Class B - it uses first two octets for network addresses and last two for host addressing
- Class C - it uses first three octets for network addresses and last one for host addressing
- Class D - it provides flat IP addressing scheme in contrast to hierarchical structure for above three.
- Class E - It is used as experimental.

Internet Protocol Version 6 (IPv6)

Exhaustion of IPv4 addresses gave birth to a next generation Internet Protocol version 6. IPv6 addresses its nodes with 128-bit wide address providing plenty of address space for future to be used on entire planet or beyond.

IPv6 has introduced Anycast addressing but has removed the concept of broadcasting. IPv6 enables devices to self-acquire an IPv6 address and communicate within that subnet. This auto-configuration removes the dependability of Dynamic Host Configuration Protocol (DHCP) servers. This way, even if the DHCP server on that subnet is down, the hosts can communicate with each other.

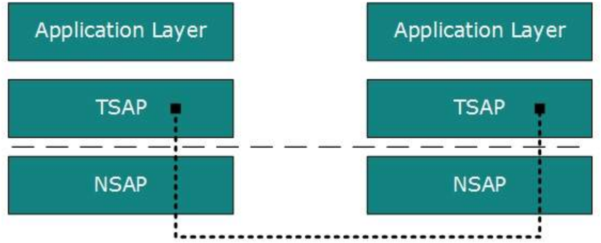
IPv6 provides new feature of IPv6 mobility. Mobile IPv6 equipped machines can roam around without the need of changing their IP addresses.

Computer Networks

Transport Layer

End-to-End Communication

A process on one host identifies its peer host on remote network by means of TSAPs, also known as Port numbers. TSAPs are very well defined and a process which is trying to communicate with its peer knows this in advance.



Computer Networks

For example, when a DHCP client wants to communicate with remote DHCP server, it always requests on port number 67. When a DNS client wants to communicate with remote DNS server, it always requests on port number 53 (UDP).

The two main Transport layer protocols are:

Transmission Control Protocol

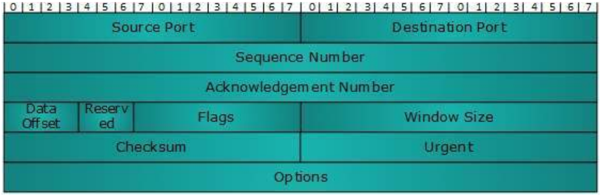
It provides reliable communication between two hosts.

User Datagram Protocol

It provides unreliable communication between two hosts.

Transmission Control Protocol

The length of TCP header is minimum 20 bytes long and maximum 60 bytes.



Computer Networks

Source Port (16-bits) - It identifies source port of the application process on the sending device.

Destination Port (16-bits) - It identifies destination port of the application process on the receiving device.

Sequence Number (32-bits) - Sequence number of data bytes of a segment in a session.

Acknowledgement Number (32-bits) - When ACK flag is set, this number contains the next sequence number of the data byte expected and works as acknowledgement of the previous data received.

Data Offset (4-bits) - This field implies both, the size of TCP header (32-bit words) and the offset of data in current packet in the whole TCP segment.

Reserved (3-bits) - Reserved for future use and all are set zero by default.

Flags (1-bit each)

Windows Size - This field is used for flow control between two stations and indicates the amount of buffer (in bytes) the receiver has allocated for a segment, i.e. how much data is the receiver expecting.

Checksum - This field contains the checksum of Header, Data and Pseudo Headers.

Urgent Pointer - It points to the urgent data byte if URG flag is set to 1.

Options - It facilitates additional options which are not covered by the regular header. Option field is always described in 32-bit words. If this field contains data less than 32-bit, padding is used to cover the remaining bits to reach 32-bit boundary.

Addressing

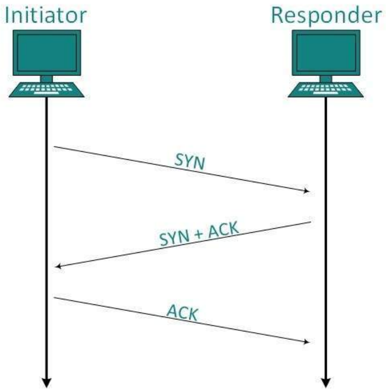
TCP communication between two remote hosts is done by means of port numbers (TSAPs). Ports numbers can range from 0 – 65535 which are divided as:

- System Ports (0 – 1023)
- User Ports (1024 – 49151)
- Private/Dynamic Ports (49152 – 65535)

Computer Networks

Connection Management

TCP communication works in Server/Client model. The client initiates the connection and the server either accepts or rejects it. Three-way handshaking is used for connection management.



Establishment

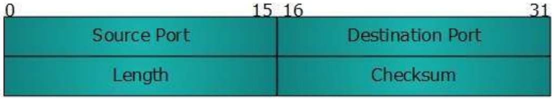
Client initiates the connection and sends the segment with a Sequence number. Server acknowledges it back with its own Sequence number and ACK of client's segment which is one more than client's Sequence number. Client after receiving ACK of its segment sends an acknowledgement of Server's response.

Release

Either of server and client can send TCP segment with FIN flag set to 1. When the receiving end responds it back by ACKnowledging FIN, that direction of TCP communication is closed and connection is released.

User Datagram Protocol

UDP header is as simple as its function.



Source Port - This 16 bits information is used to identify the source port of the packet.

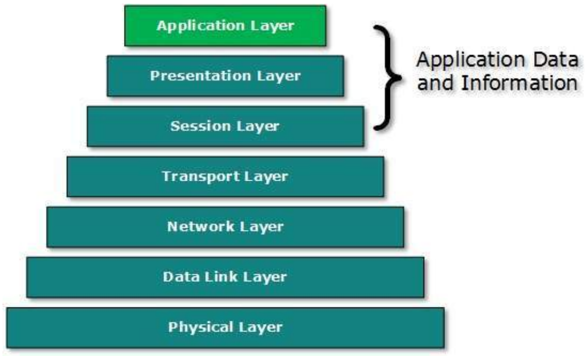
Destination Port - This 16 bits information, is used identify application level service on destination machine.

Length - Length field specifies the entire length of UDP packet (including header). It is 16-bits field and minimum value is 8-byte, i.e. the size of UDP header itself.

Checksum - This field stores the checksum value generated by the sender before sending. IPv4 has this field as optional so when checksum field does not contain any value it is made 0 and all its bits are set to zero.

Computer Networks

Application Layer



Computer Networks

Client Server Model

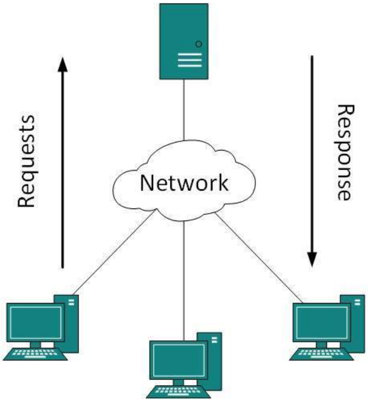
Two remote application processes can communicate mainly in two different fashions:

Peer-to-peer: Both remote processes are executing at same level and they exchange data using some shared resource.

Client-Server: One remote process acts as a Client and requests some resource from another application process acting as Server.

In client-server model, any process can act as Server or Client. It is not the type of machine, size of the machine, or its computing power which makes it server; it is the ability of serving request that makes a machine a server.

Computer Networks



A system can act as Server and Client simultaneously. That is, one process is acting as Server and another is acting as a client. This may also happen that both client and server processes reside on the same machine.

Computer Networks

Communication

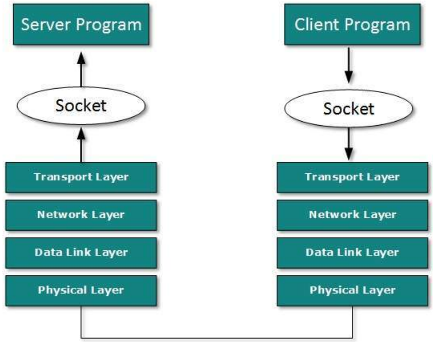
Two processes in client-server model can interact in various ways:

- Sockets
- Remote Procedure Calls (RPC)

Computer Networks

Sockets

In this paradigm, the process acting as Server opens a socket using a well-known (or known by client) port and waits until some client request comes. The second process acting as a Client also opens a socket but instead of waiting for an incoming request, the client processes 'requests first'.



When the request is reached to server, it is served. It can either be an information sharing or resource request.

Remote Procedure Call

This is a mechanism where one process interacts with another by means of procedure calls. One process (client) calls the procedure lying on remote host. The process on remote host is said to be Server. Both processes are allocated stubs. This communication happens in the following way:

- The client process calls the client stub. It passes all the parameters pertaining to program local to it.
- All parameters are then packed (marshalled) and a system call is made to send them to other side of the network.
- Kernel sends the data over the network and the other end receives it.
- The remote host passes data to the server stub where it is unmarshalled.
- The parameters are passed to the procedure and the procedure is then executed.
- The result is sent back to the client in the same manner.

HTTP and Web

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (ie. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extension of its request methods, error codes and headers.

Basically, HTTP is an TCP/IP based communication protocol, which is used to deliver data (HTML files, image files, query results etc) on the World Wide Web. The default port is TCP 80, but other ports can be used. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients request data will be constructed and sent to the serve, and how servers respond to these requests.

HTTP and Web

Basic Features

There are following three basic features which makes HTTP a simple but powerful protocol:

HTTP is connectionless: The HTTP client ie. browser initiates an HTTP request and after a request is made, the client disconnects from the server and waits for a response. The server process the request and re-establish the connection with the client to send response back.

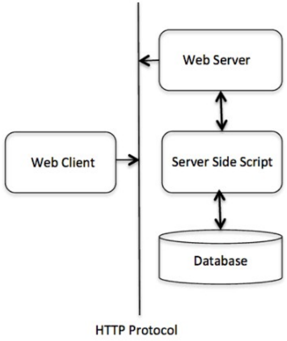
HTTP is media independent: This means, any type of data can be sent by HTTP as long as both the client and server know how to handle the data content. This is required for client as well as server to specify the content type using appropriate MIME-type.

HTTP is stateless: As mentioned above, HTTP is a connectionless and this is a direct result that HTTP is a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different request across the web pages.

HTTP and Web

Basic Architecture

Following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



HTTP and Web

The HTTP protocol is a request/response protocol based on client/server based architecture where web browser, robots and search engines, etc. act like HTTP clients and Web server acts as server.

Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity metainformation, and possible entity-body content.

HTTP and Web

HTTP - Requests

An HTTP client sends an HTTP request to a server in the form of a request message which includes following format:

- A Request-line
- Zero or more header (General|Request|Entity) fields followed by CRLF
- An empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields
- Optionally a message-body

HTTP and Web

Message Request-Line

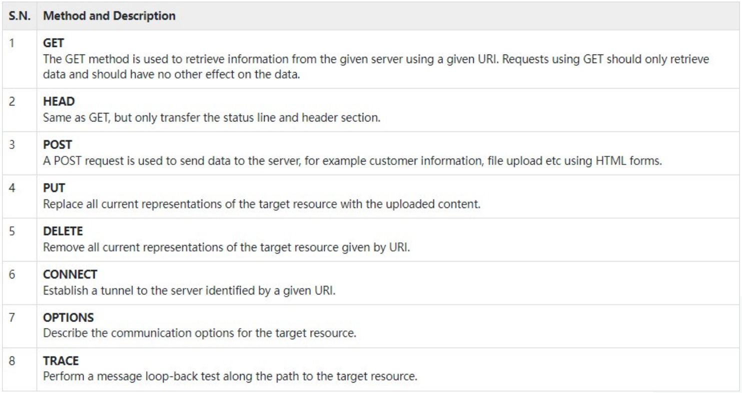
The Request-Line begins with a method token, followed by the Request-URI and the protocol version, and ending with CRLF. The elements are separated by space SP characters.

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

HTTP and Web

Request Method

The request Method indicates the method to be performed on the resource identified by the given Request-URI. The method is case-sensitive ans should always be mentioned uppercase. Following are supported methods in HTTP/1.1



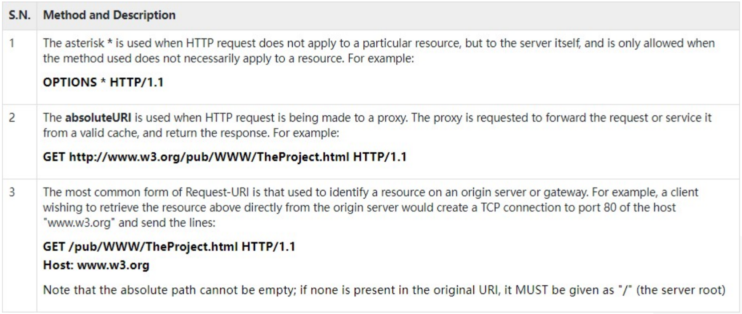
S.N.	Method and Description
1	**GET** The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
2	**HEAD** Same as GET, but only transfer the status line and header section.
3	**POST** A POST request is used to send data to the server, for example customer information, file upload etc using HTML forms.
4	**PUT** Replace all current representations of the target resource with the uploaded content.
5	**DELETE** Remove all current representations of the target resource given by URI.
6	**CONNECT** Establish a tunnel to the server identified by a given URI.
7	**OPTIONS** Describe the communication options for the target resource.
8	**TRACE** Perform a message loop-back test along the path to the target resource.

HTTP and Web

Request-URI

The Request-URI is a Uniform Resource Identifier and identifies the resource upon which to apply the request. Following are the most commonly used forms to specify an URI:

Request-URI = "*" | absoluteURI | abs_path | authority

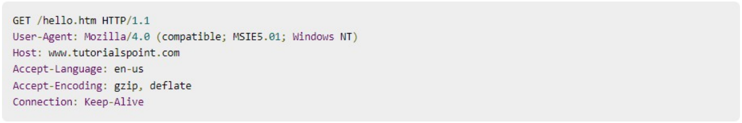


S.N.	Method and Description
1	The asterisk * is used when HTTP request does not apply to a particular resource, but to the server itself, and is only allowed when the method used does not necessarily apply to a resource. For example: **OPTIONS * HTTP/1.1**
2	The **absoluteURI** is used when HTTP request is being made to a proxy. The proxy is requested to forward the request or service it from a valid cache, and return the response. For example: **GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1**
3	The most common form of Request-URI is that used to identify a resource on an origin server or gateway. For example, a client wishing to retrieve the resource above directly from the origin server would create a TCP connection to port 80 of the host "www.w3.org" and send the lines: **GET /pub/WWW/TheProject.html HTTP/1.1** **Host: www.w3.org** Note that the absolute path cannot be empty; if none is present in the original URI, it MUST be given as "/" (the server root)

HTTP and Web

Request Message Examples

Now let's put it all together to form an HTTP request to fetch hello.htm page from the web server running on tutorialspoint.com



```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

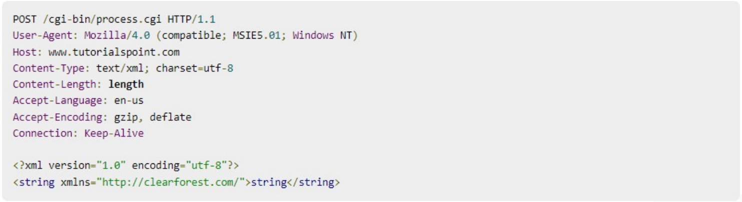
HTTP and Web

Here we are not sending any request data to the server because we are fetching a plan HTML page from the server. Connection is a general-header used here and rest of the headers are request headers. Following is one more example where we send form data to the server using request message body:

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

licenseID=string&content=string&/paramsXML=string
```

HTTP and Web

Here given URL /cgi-bin/process.cgi will be used to process the passed data and accordingly a response will be retuned. Here content-type tells the server that passed data is simple web form data and length will be actual length of the data put in the message body. Following example shows how you can pass plan XML to your web server:



```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```

HTTP and Web

HTTP - Responses

After receiving and interpreting a request message, a server responds with an HTTP response message:

- A Status-line
- Zero or more header (General|Response|Entity) fields followed by CRLF
- An empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields
- Optionally a message-body

HTTP and Web

Message Status-Line

The Status-Line consisting of the protocol version followed by a numeric status code and its associated textual phrase. The elements are separated by space SP characters.

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

HTTP Version

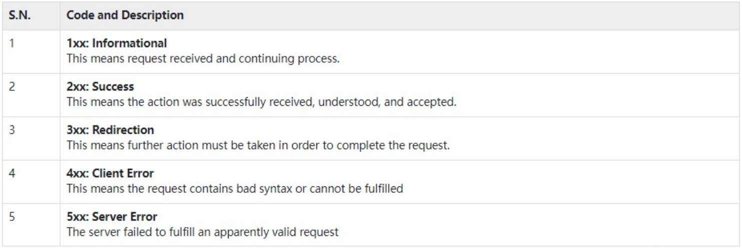
A server supporting HTTP version 1.1 will return following version information:

HTTP-Version = HTTP/1.1

HTTP and Web

Status Code

The Status-Code element is a 3-digit integer where first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. There are 5 values for the first digit:



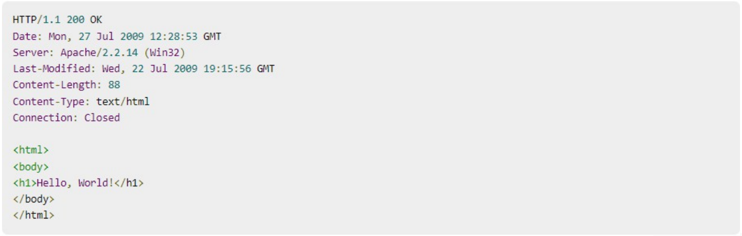
S.N.	Code and Description
1	**1xx: Informational** This means request received and continuing process.
2	**2xx: Success** This means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** This means further action must be taken in order to complete the request.
4	**4xx: Client Error** This means the request contains bad syntax or cannot be fulfilled
5	**5xx: Server Error** The server failed to fulfill an apparently valid request

HTTP status codes are extensible and HTTP applications are not required to understand the meaning of all registered status codes. A list of all the status code has been given in a separate chapter for you reference.

HTTP and Web

Response Message Examples

Now let’s put it all together to form an HTTP response for a request to fetch hello.htm page from the web server running on tutorialspoint.com



```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed

<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

HTTP and Web

Following is an example of HTTP response message showing error condition when web server could not find requested page:

```
HTTP/1.1 404 Not Found
Date: Sun, 18 Oct 2012 10:36:20 GMT
Server: Apache/2.2.14 (Win32)
Content-Length: 230
Connection: Closed
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
   <title>404 Not Found</title>
</head>
<body>
   <h1>Not Found</h1>
   <p>The requested URL /t.html was not found on this server.</p>
</body>
</html>
```

HTTP and Web

Following is an example of HTTP response message showing error condition when web server encountered a wrong HTTP version in given HTTP request:

```
HTTP/1.1 400 Bad Request
Date: Sun, 18 Oct 2012 10:36:20 GMT
Server: Apache/2.2.14 (Win32)
Content-Length: 230
Content-Type: text/html; charset=iso-8859-1
Connection: Closed

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
   <title>400 Bad Request</title>
</head>
<body>
   <h1>Bad Request</h1>
   <p>Your browser sent a request that this server could not understand.<p>
   <p>The request line contained invalid characters following the protocol string.<p>
</body>
</html>
```

Database - MySQL

What is a Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys.

Database - MySQL

A Relational DataBase Management System (RDBMS) is a software that

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

Database - MySQL

RDBMS Terminology

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- Database A database is a collection of tables, with related data.
- Table A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- Column One column (data element) contains data of one and the same kind, for example the column postcode.
- Row A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- Redundancy Storing data twice, redundantly to make the system faster.
- Primary Key A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- Foreign Key A foreign key is the linking pin between two tables.
- Compound Key A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- Index An index in a database resembles an index at the back of a book.
- Referential Integrity Referential Integrity makes sure that a foreign key value always points to an existing row.

Database - MySQL

Browsing

- SHOW DATABASES;
- SHOW TABLES;
- SHOW FIELDS FROM table / DESCRIBE table;
- SHOW CREATE TABLE table;
- SHOW PROCESSLIST;
- KILL process_number;

Database - MySQL

Select

- SELECT * FROM table;
- SELECT * FROM table1, table2;
- SELECT field1, field2 FROM table1, table2;
- SELECT ... FROM ... WHERE condition
- SELECT ... FROM ... WHERE condition GROUP BY field;
- SELECT ... FROM ... WHERE condition GROUP BY field HAVING condition2;
- SELECT ... FROM ... WHERE condition ORDER BY field1, field2;
- SELECT ... FROM ... WHERE condition ORDER BY field1, field2 DESC;
- SELECT ... FROM ... WHERE condition LIMIT 10;
- SELECT DISTINCT field1 FROM ...
- SELECT DISTINCT field1, field2 FROM ...

Database - MySQL

Create / Open / Delete Database

- CREATE DATABASE DatabaseName;
- CREATE DATABASE DatabaseName CHARACTER SET utf8;
- USE DatabaseName;
- DROP DATABASE DatabaseName;
- ALTER DATABASE DatabaseName CHARACTER SET utf8;

Database - MySQL

Conditions

- field1 = value1
- field1 <> value1
- field1 LIKE 'value _ %'
- field1 IS NULL
- field1 IS NOT NULL
- field1 IS IN (value1, value2)
- field1 IS NOT IN (value1, value2)
- condition1 AND condition2
- condition1 OR condition2

Database - MySQL

Insert

- INSERT INTO table1 (field1, field2) VALUES (value1, value2);

Database - MySQL

Update

- UPDATE table1 SET field1=new_value1 WHERE condition;
- UPDATE table1, table2 SET field1=new_value1, field2=new_value2, ... WHERE table1.id1 = table2.id2 AND condition;

Database - MySQL

Delete

- DELETE FROM table1 / TRUNCATE table1
- DELETE FROM table1 WHERE condition
- DELETE FROM table1, table2 FROM table1, table2 WHERE table1.id1 = table2.id2 AND condition

Database - MySQL

Create / Delete / Modify Table

- CREATE TABLE table (field1 type1, field2 type2);
- CREATE TABLE table (field1 type1, field2 type2, INDEX (field));
- CREATE TABLE table (field1 type1, field2 type2, PRIMARY KEY (field1));
- CREATE TABLE table (field1 type1, field2 type2, PRIMARY KEY (field1,field2));

Database - MySQL

Create / Delete / Modify Table

- DROP TABLE table;
- DROP TABLE IF EXISTS table;
- DROP TABLE table1, table2, ...

Database - MySQL

Create / Delete / Modify Table

- ALTER TABLE table MODIFY field1 type1
- ALTER TABLE table MODIFY field1 type1 NOT NULL ...
- ALTER TABLE table CHANGE old_name_field1 new_name_field1 type1
- ALTER TABLE table CHANGE old_name_field1 new_name_field1 type1 NOT NULL ...
- ALTER TABLE table ALTER field1 SET DEFAULT ...
- ALTER TABLE table ALTER field1 DROP DEFAULT
- ALTER TABLE table ADD new_name_field1 type1
- ALTER TABLE table ADD new_name_field1 type1 FIRST
- ALTER TABLE table ADD new_name_field1 type1 AFTER another_field
- ALTER TABLE table DROP field1
- ALTER TABLE table ADD INDEX (field);

Database - MySQL

MySQL - and SQL Injection

If you take user input through a webpage and insert it into a MySQL database, there's a chance that you have left yourself wide open for a security issue known as SQL Injection. This chapter will teach you how to help prevent this from happening and help you secure your scripts and MySQL statements.

The SQL Injection usually occurs when you ask a user for input, like their name and instead of a name they give you a MySQL statement that you will unknowingly run on your database.

Never trust the data provided by a user, process this data only after validation; as a rule, this is done by pattern matching. In the following example, the username is restricted to alphanumerical characters plus underscore and to a length between 8 and 20 characters – modify these rules as needed.

```
if (preg_match("/^\w{8,20}$/", $_GET['username'], $matches)) {
   $result = mysql_query("SELECT * FROM users WHERE username = $matches[0]");
} else  {
   echo "username not accepted";
}
```

Database - MySQL

To demonstrate this problem, consider the following excerpt.

```
// supposed input
$name = "Qadir'; DELETE FROM users;";
mysql_query("SELECT * FROM users WHERE name = '{$name}'");
```

The function call is supposed to retrieve a record from the users table, where the name column matches the name specified by the user. Under normal circumstances, $name would only contain alphanumeric characters and perhaps spaces. But here, by appending an entirely new query to $name, the call to the database turns into a disaster. The injected DELETE query removes all the records from users.

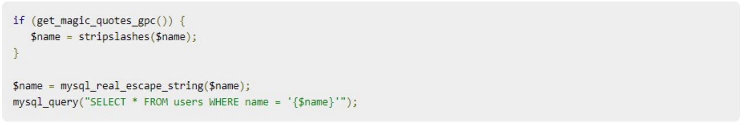
Fortunately, if you use MySQL, the mysql_query() function does not permit query stacking or executing multiple queries in a single function call. If you try to stack queries, the call fails.

However, other PHP database extensions, such as SQLite and PostgreSQL, happily perform stacked queries, executing all the queries provided in one string and creating a serious security problem.

Database - MySQL

Preventing SQL Injection

You can handle all escape characters smartly in scripting languages like PERL and PHP. The MySQL extension for PHP provides the function mysql_real_escape_string() to escape input characters that are special to MySQL.



```
if (get_magic_quotes_gpc()) {
   $name = stripslashes($name);
}

$name = mysql_real_escape_string($name);
mysql_query("SELECT * FROM users WHERE name = '{$name}'");
```

Database - MySQL

The LIKE Quandary

To address the LIKE quandary, a custom escaping mechanism must convert user-supplied % and _ characters to literals. Use addcslashes(), a function that lets you specify a character range to escape.

```
$sub = addcslashes(mysql_real_escape_string("%something_"), "%_");
// $sub == \%something\_
mysql_query("SELECT * FROM messages WHERE subject LIKE '{$sub}%'");
```

Vulnerabilities

Web Exploitation

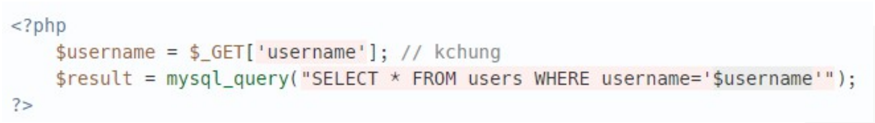
Websites all around the world are programmed using various programming languages. While there are specific vulnerabilities in each programming langage that the developer should be aware of, there are issues fundamental to the internet that can show up regardless of the chosen language or framework.

These vulnerabilities often show up in CTFs as web security challenges where the user needs to exploit a bug to gain some kind of higher level privelege.

Vulnerabilities

SQL Injection

SQL Injection is a vulnerability where an application takes input from a user and doesn't vaildate that the user's input doesn't contain additional SQL.



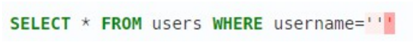
```
<?php
    $username = $_GET['username']; // kchung
    $result = mysql_query("SELECT * FROM users WHERE username='$username'");
?>
```

Vulnerabilities

If we look at the $username variable, under normal operation we might expect the username parameter to be a real username (e.g. kchung).

But a malicious user might submit different kind of data. For example, consider if the input was '?

The application would crash because the resulting SQL query is incorrect.

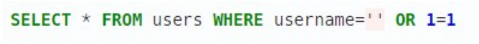


```
SELECT * FROM users WHERE username='''
```

Vulnerabilities

With the knowledge that a single quote will cause an error in the application we can expand a little more on SQL Injection.

What if our input was ' OR 1=1?



```
SELECT * FROM users WHERE username='' OR 1=1
```

Vulnerabilities

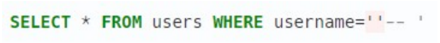
1 is indeed equal to 1. This equates to true in SQL. If we reinterpret this the SQL statement is really saying

```
SELECT * FROM users WHERE username='' OR true
```

Vulnerabilities

This will return every row in the table because each row that exists must be true.

We can also inject comments and termination characters like – or /* or ;. This allows you to terminate SQL queries after your injected statements. For example '– is a common SQL injection payload.



```
SELECT * FROM users WHERE username=''-- '
```

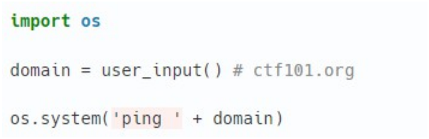
This payload sets the username parameter to an empty string to break out of the query and then adds a comment (–) that effectively hides the second single quote.

Using this technique of adding SQL statements to an existing query we can force databases to return data that it was not meant to return.

Vulnerabilities

Command Injection

Command Injection is a vulnerability that allows an attacker to submit system commands to a computer running a website. This happens when the application fails to encode user input that goes into a system shell. It is very common to see this vulnerability when a developer uses the system() command or its equivalent in the programming language of the application.

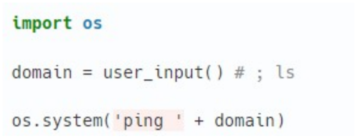


```
import os

domain = user_input() # ctf101.org

os.system('ping ' + domain)
```

Vulnerabilities

The above code when used normally will ping the ctf101.org domain.

But consider what would happen if the user_input() function returned different data?



```
import os

domain = user_input() # ; ls

os.system('ping ' + domain)
```

Vulnerabilities

Because of the additional semicolon, the os.system() function is instructed to run two commands.

It looks to the program as:

```
ping ; ls
```

Because the ping command is being terminated and the ls command is being added on, the ls command will be run in addition to the empty ping command!

This is the core concept behind command injection. The ls command could of course be switched with another command (e.g. wget, curl, bash, etc.)

Command injection is a very common means of privelege escalation within web applications and applications that interface with system commands. Many kinds of home routers take user input and directly append it to a system command. For this reason, many of those home router models are vulnerable to command injection.

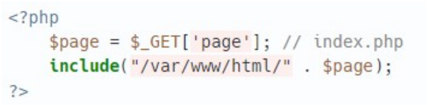
Vulnerabilities

Directory Traversal

Directory Traversal is a vulnerability where an application takes in user input and uses it in a directory path.

Any kind of path controlled by user input that isn't properly sanitized or properly sandboxed could be vulnerable to directory traversal.

For example, consider an application that allows the user to choose what page to load from a GET parameter.



```php
<?php
    $page = $_GET['page']; // index.php
    include("/var/www/html/" . $page);
?>
```

Vulnerabilities

Under normal operation the page would be index.php. But what if a malicious user gave in something different?

```
<?php
    $page = $_GET['page']; // ../../../../../../../../etc/passwd
    include("/var/www/html/" . $page);
?>
```

Here the user is submitting ../../../../../../../../etc/passwd.

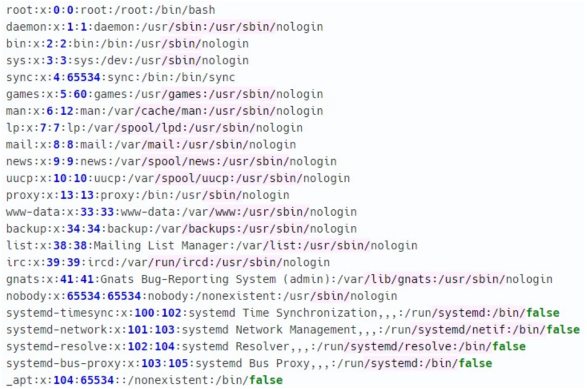
This will result in the PHP interpreter leaving the directory that it is coded to look in ('/var/www/html') and instead be forced up to the root folder.

```
include("/var/www/html/../../../../../../../../etc/passwd");
```

Vulnerabilities

Ultimately this will become /etc/passwd because the computer will not go a directory above its top directory.

Thus the application will load the /etc/passwd file and emit it to the user like so:



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
```

Vulnerabilities

Cross Site Scripting (XSS)

Cross Site Scripting or XSS is a vulnerability where on user of an application can send JavaScript that is executed by the browser of another user of the same application.

This is a vulnerability because JavaScript has a high degree of control over a user's web browser.

For example JavaScript has the ability to:

- Modify the page (called the DOM)
- Send more HTTP requests
- Access cookies

By combining all of these abilities, XSS can maliciously use JavaScript to extract user's cookies and send them to an attacker controlled server. XSS can also modify the DOM to phish users for their passwords. This only scratches the surface of what XSS can be used to do.

XSS is typically broken down into three categories:

- Reflected XSS
- Stored XSS
- DOM XSS

Vulnerabilities

Reflected XSS

Reflected XSS is when an XSS exploit is provided through a URL paramater.

For example:

```
https://ctf101.org?data=<script>alert(1)</script>
```

Vulnerabilities

You can see the XSS exploit provided in the data GET parameter. If the application is vulnerable to reflected XSS, the application will take this data parameter value and inject it into the DOM.

For example:

```
<html>
    <body>
        <script>alert(1)</script>
    </body>
</html>
```

Vulnerabilities

Depending on where the exploit gets injected, it may need to be constructed differently.

Also, the exploit payload can change to fit whatever the attacker needs it to do. Whether that is to extract cookies and submit it to an external server, or to simply modify the page to deface it.

One of the deficiencies of reflected XSS however is that it requires the victim to access the vulnerable page from an attacker controlled resource. Notice that if the data paramter, wasn't provided the exploit wouldn't work.

In many situations, reflected XSS is detected by the browser because it is very simple for a browser to detect malicous XSS payloads in URLs.

Vulnerabilities

Stored XSS

Stored XSS is different from reflected XSS in one key way. In reflected XSS, the exploit is provided through a GET parameter. But in stored XSS, the exploit is provided from the website itself.

Imagine a website that allows users to post comments. If a user can submit an XSS payload as a comment, and then have others view that malicious comment, it would be an example of stored XSS.

The reason being that the web site itself is serving up the XSS payload to other users. This makes it very difficult to detect from the browser's perspective and no browser is capable of generically preventing stored XSS from exploiting a user.

Vulnerabilities

DOM XSS

DOM XSS is XSS that is due to the browser itself injecting an XSS payload into the DOM. While the server itself may properly prevent XSS, it's possible that the client side scripts may accidentally take a payload and insert it into the DOM and cause the payload to trigger.

The server itself is not to blame, but the client side JavaScript files are causing the issue.

Tools

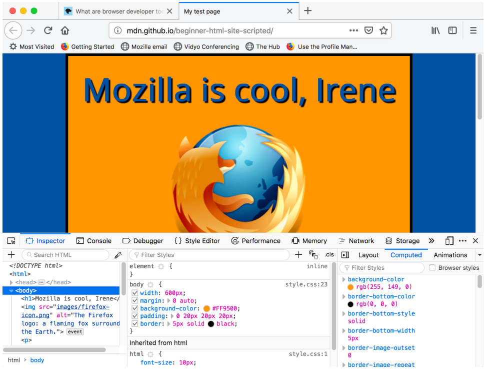
What are browser developer tools?

Every modern web browser includes a powerful suite of developer tools. These tools do a range of things, from inspecting currently-loaded HTML, CSS and JavaScript to showing which assets the page has requested and how long they took to load. This article explains how to use the basic functions of your browser's devtools.

Tools

How to open the devtools in your browser

The devtools live inside your browser in a subwindow that looks roughly like this, depending on what browser you are using:



```
<!DOCTYPE html>
<html>
<head>…</head>
<body>
  <h1>Mozilla is cool, Irene</
  <img src="images/firefox-
  icon.png" alt="The Firefox
  logo: a flaming fox surround
  the Earth."> event
  <p>
```

```
element {                      inline
}
body {                         style.css:23
  width: 600px;
  margin: 0 auto;
  background-color: #FF9500;
  padding: 0 20px 20px 20px;
  border: 5px solid black;
}
Inherited from html
html {                         style.css:1
  font-size: 10px;
```

```
background-color
  rgb(255, 149, 0)
border-bottom-color
  rgb(0, 0, 0)
border-bottom-style
  solid
border-bottom-width
  5px
border-image-outset
  0
border-image-repeat
```

Tools

How do you pull it up? Three ways:

Keyboard: Ctrl + Shift + I, except

Internet Explorer and Edge: F12

Menu bar:

Firefox: Menu > Web Developer > Toggle Tools, or Tools > Web Developer > Toggle Tools

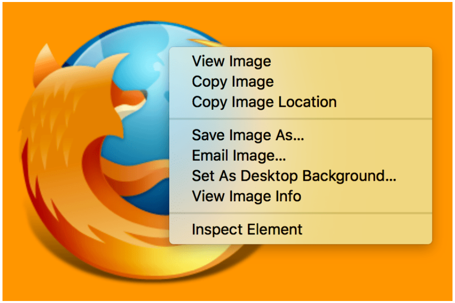
Chrome: More tools > Developer tools

Safari: Develop > Show Web Inspector. If you can't see the Develop menu, go to Safari > Preferences > Advanced, and check the Show Develop menu in menu bar checkbox.

Opera: Developer > Developer tools

Tools

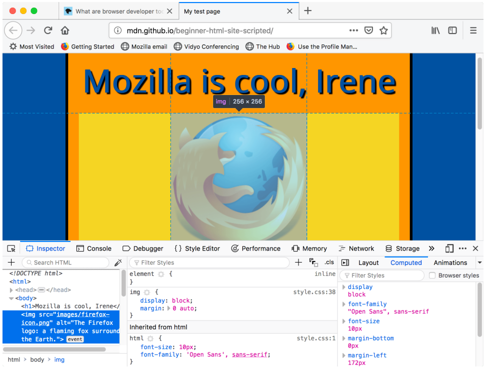
Context menu: Press-and-hold/right-click an item on a webpage (Ctrl-click on the Mac), and choose Inspect Element from the context menu that appears. (An added bonus: this method straight-away highlights the code of the element you right-clicked.)



Tools

The Inspector: DOM explorer and CSS editor

The developer tools usually open by default to the inspector, which looks something like the following screenshot. This tool shows what the HTML on your page looks like at runtime, as well as what CSS is applied to each element on the page. It also allows you to instantly modify the HTML and CSS and see the results of your changes reflected live in the browser viewport.



```
<!DOCTYPE html>
<html>
  <head>…</head>
  <body>
    <h1>Mozilla is cool, Irene</
    <img src="images/firefox-
    icon.png" alt="The Firefox
    logo: a flaming fox surround
    the Earth."> event
```

```
element {                          inline
}
img {                         style.css:38
  display: block;
  margin: 0 auto;
}
Inherited from html
html {                         style.css:1
  font-size: 10px;
  font-family: 'Open Sans', sans-serif;
}
```

```
display
block
font-family
"Open Sans", sans-serif
font-size
10px
margin-bottom
0px
margin-left
172px
```

Tools

If you don’t see the inspector,

Tap/click the Inspector tab.

In Internet Explorer, tap/click DOM Explorer, or press Ctrl + 1 .

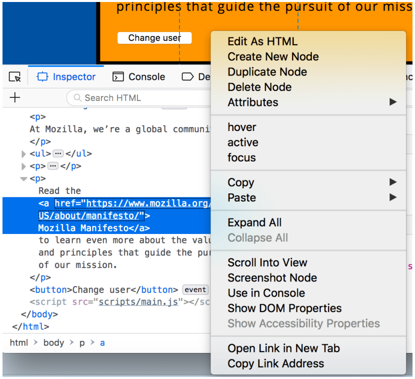
In Microsoft Edge, or Opera, tap/click Elements.

In Safari, the controls are not so clearly presented, but you should see the HTML if you haven’t selected something else to appear in the window. Press the Style button to see the CSS.

Tools

Exploring the DOM inspector

For a start, right-click (Ctrl-click) an HTML element in the DOM inspector and look at the context menu. The available menu options vary among browsers, but the important ones are mostly the same:



```
    <p>
    At Mozilla, we're a global communi
    </p>
  ▸ <ul>…</ul>
  ▸ <p>…</p>
  ▾ <p>
      Read the
      <a href="https://www.mozilla.org/
      US/about/manifesto/">
      Mozilla Manifesto</a>
      to learn even more about the valu
      and principles that guide the pur
      of our mission.
    </p>
    <button>Change user</button> event
    <script src="scripts/main.js"></sc
  </body>
</html>
```

Tools

Delete Node (sometimes Delete Element). Deletes the current element.

Edit as HTML (sometimes Add attribute/Edit text). Lets you change the HTML and see the results on the fly. Very useful for debugging and testing.

:hover/:active/:focus. Forces element states to be toggled on, so you can see what their styling would look like.

Copy/Copy as HTML. Copy the currently selected HTML.

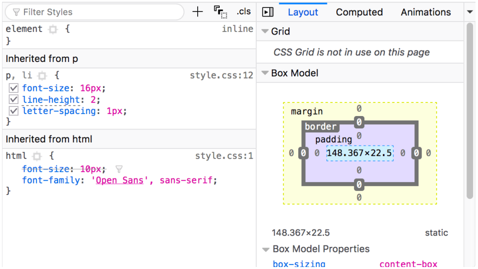
Some browsers also have Copy CSS Path and Copy XPath available, to allow you to copy the CSS selector or XPath expression that would select the current HTML element.

Try editing some of your DOM now. Double-click an element, or right-click it and choose Edit as HTML from the context menu. You can make any changes you'd like, but you cannot save your changes.

Tools

Exploring the CSS editor

By default, the CSS editor displays the CSS rules applied to the currently selected element:



```
element {                                  inline
}
Inherited from p
p, li {                              style.css:12
  ✓ font-size: 16px;
  ✓ line-height: 2;
  ✓ letter-spacing: 1px;
}
Inherited from html
html {                                style.css:1
    ~~font-size: 10px;~~
    font-family: 'Open Sans', sans-serif;
}
```

```
148.367×22.5                static
box-sizing              content-box
```

Tools

These features are especially handy:

The rules applied to the current element are shown in order of most-to-least-specific.

Click the checkboxes next to each declaration to see what would happen if you removed the declaration.

Click the little arrow next to each shorthand property to show the property's longhand equivalents.

Click a property name or value to bring up a text box, where you can key in a new value to get a live preview of a style change.

Next to each rule is the file name and line number the rule is defined in. Clicking that rule causes the dev tools to jump to show it in its own view, where it can generally be edited and saved.

You can also click the closing curly brace of any rule to bring up a text box on a new line, where you can write a completely new declaration for your page.

Tools

You'll notice a number of clickable tabs at the top of the CSS Viewer:

Computed: This shows the computed styles for the currently selected element (the final, normalized values that the browser applies).

Layout: In Firefox, this area includes two sections:

Box Model: represents visually the current element's box model, so you can see at a glance what padding, border and margin is applied to it, and how big its content is.

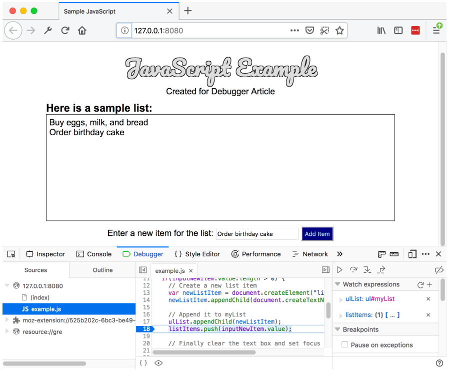
Grid: If the page you are inspecting uses CSS Grid, this section allows you to view the grid details.

Fonts: In Firefox, the Fonts tab shows the fonts applied to the current element.

Tools

The JavaScript debugger

The JavaScript debugger allows you to watch the value of variables and set breakpoints, places in your code that you want to pause execution and identify the problems that prevent your code from executing properly.



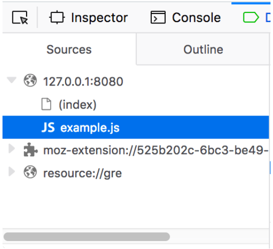
```
    // Create a new list item
    var newListItem = document.createElement("li
    newListItem.appendChild(document.createTextN

    // Append it to myList
    ulList.appendChild(newListItem);
    listItems.push(inputNewItem.value);

    // Finally clear the text box and set focus
```

Tools

Exploring the debugger

File list

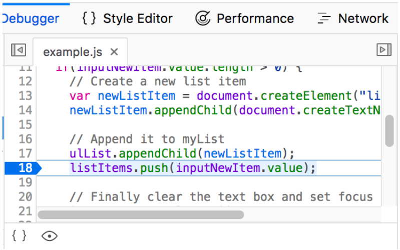
The first pane on the left contains the list of files associated with the page you are debugging. Select the file you want to work with from this list. Click on a file to select it and view its contents in the center pane of the Debugger.



Tools

Source code

Set breakpoints where you want to pause execution. In the following image, the highlight on the number 18 shows that the line has a breakpoint set.



```
11      if(inputNewItem.value.length > 0) {
12        // Create a new list item
13        var newListItem = document.createElement("li
14        newListItem.appendChild(document.createTextN
15
16        // Append it to myList
17        ulList.appendChild(newListItem);
18        listItems.push(inputNewItem.value);
19
20        // Finally clear the text box and set focus
21
```

Tools

Watch expressions and breakpoints

The right-hand pane shows a list of the watch expressions you have added and breakpoints you have set.

In the image, the first section, Watch expressions, shows that the listItems variable has been added. You can expand the list to view the values in the array.

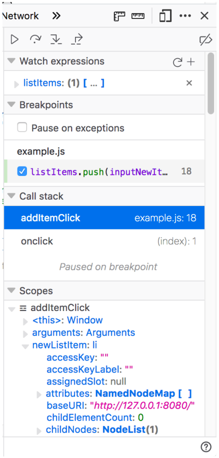
The next section, Breakpoints, lists the breakpoints set on the page. In example.js, a breakpoint has been set on the statement listItems.push(inputNewItem.value);

The final two sections only appear when the code is running.

The Call stack section shows you what code was executed to get to the current line. You can see that the code is in the function that handles a mouse click, and that the code is currently paused on the breakpoint.

Tools

The final section, Scopes, shows what values are visible from various points within your code. For example, in the image below, you can see the objects available to the code in the addItemClick function.



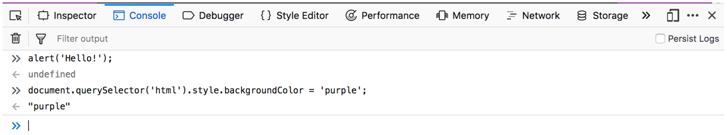
```
listItems: (1) [ … ]
Pause on exceptions
example.js
listItems.push(inputNewIt…   18
addItemClick    example.js: 18
onclick         (index): 1
Paused on breakpoint
addItemClick
  <this>: Window
  arguments: Arguments
  newListItem: li
    accessKey: ""
    accessKeyLabel: ""
    assignedSlot: null
    attributes: NamedNodeMap [  ]
    baseURI: "http://127.0.0.1:8080/"
    childElementCount: 0
    childNodes: NodeList(1)
```

Tools

The JavaScript console

The JavaScript console is an incredibly useful tool for debugging JavaScript that isn't working as expected. It allows you to run lines of JavaScript against the page currently loaded in the browser, and reports the errors encountered as the browser tries to execute your code.

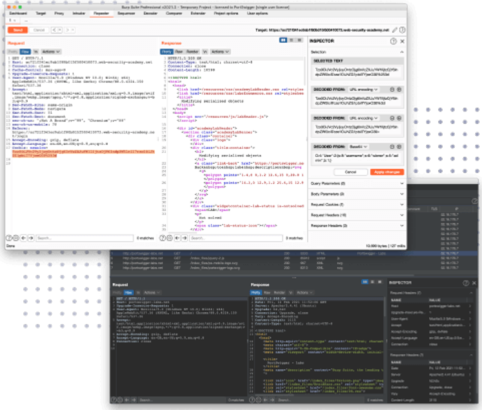
This will give you a window like the following:



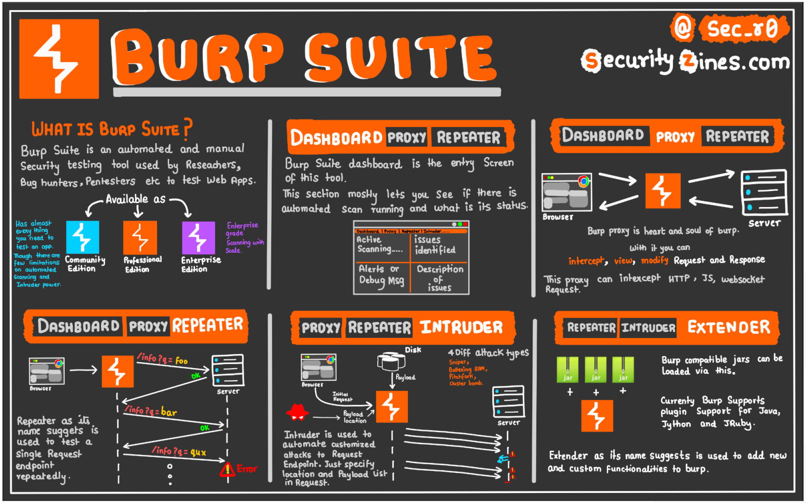
```
>> alert('Hello!');
<- undefined
>> document.querySelector('html').style.backgroundColor = 'purple';
<- "purple"
>> |
```

Tools

Burp Suite



Tools



BURP SUITE

@Sec_r0
SecurityZines.com

WHAT IS BURP SUITE?

Burp Suite is an automated and manual Security testing tool used by Reseachers, Bug hunters, Pentesters etc to test Web Apps.

Available as

Community Edition: Has almost every thing you need to test an app. Though there are few limitations on automated scanning and Intruder power.

Professional Edition

Enterprise Edition: Enterprise grade Scanning with Scale.

DASHBOARD PROXY REPEATER

Burp Suite dashboard is the entry Screen of this tool.

This section mostly lets you See if there is automated scan running and what is its status.

DASHBOARD PROXY REPEATER

Burp proxy is heart and soul of burp.

with it you can intercept, view, modify Request and Response

This proxy can intercept HTTP, JS, websocket Request.

DASHBOARD PROXY REPEATER

Repeater as its name suggets is used to test a single Request endpoint repeatedly.

PROXY REPEATER INTRUDER

4 Diff attack types: Sniper, Battering RAM, Pitchfork, cluster bomb.

Intruder is used to automate customized attacks to Request Endpoint. Just specify location and Payload list in Request.

REPEATER INTRUDER EXTENDER

Burp compatible jars can be loaded via this.

Currenty Burp Supports plugin Support for Java, Jython and JRuby.

Extender as its name suggests is used to add new and custom functionalities to burp.

Summary

Contents:

- Computer Networks
- Application Protocol: HTTP and Web
- Database: MySQL Overview
- Vulnerabilities in the Web
- Solving Web Challenges (basics)

Hope you enjoy today's lecture.