Networking:

DHCP at packet level: [https://medium.com/@bromiley/full-packet-friday-dhcp-abbc6b7b3c77]

- By default the DHCP IP range: 192.168.1.100 - 192.168.1.149
- There are certain ways to dynamically assign an IP address to the networking device.
    1. APIPA
    2. DHCP
- **DHCP process : DORA**
    1. DHCP Discover packet (client-server) (Broadcast):
        - Client will send broadcast DHCP Discover packet in the network to find the DHCP server.
        - Its broadcast packet with IP address 255.255.255.255 [pre-define].
        - Client IP will be 0.0.0.0 [Client doesn't have IP yet.]
        - Its UDP packet.
        - Broadcast packet. (There is one flag inside our DHCP packet.)
        - DHCP ports: [client side = 68 and server side = 67]
        - Server can reach out to client using it's mac address.
        - Transection ID : which maintain the specific DHCP session between client-server, which remains the same during [discover + offer]]the process.
        - In bootstrap section there will be options which contains DHCP message type info.
    2. DHCP Offer packet (server-client) (unicast):
        - DHCP server respond with the DHCP offer to client.
        - Which containers the information about IP that can be used by client and leasing time period.
        - It is DHCP is unicast from server-client.
    3. DHCP Request packet (Client-server) (broadcast)
        - This packet will take the new transaction ID.
        - If client is agreed with the terms and IP address then send the DHCP request packet to the server "Saying that i would like to take this IP address."
        - DHCP request packet still sending with the broadcast IP 255.255.255.255.
        - Also use the UDP
        - DHCP request packet still use the client IP as 0.0.0.0
    4. DHCP Acknowledgement packet (Server-client) (Unicast)
        - After server gets the DHCP request from the client.
        - Finally it will send the DHCP ACK packet to the client. Which will assign IP address and subnet mask to the client along with leasing time and default gateway information.

What is the DHCP relay agent?

A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet.
To configure the DHCP relay agent we can use the command : **dhcp-helper** command

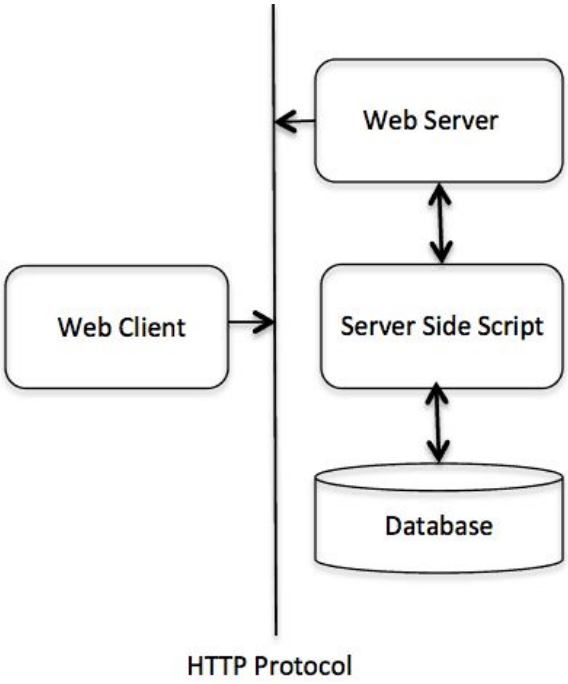
HTTP:

General description:

- ➔ The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, hypermedia information systems.
- ➔ Basically, HTTP is a TCP/IP based communication protocol, that 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 as well. It provides a standardized way for computers to communicate with each other.
- ➔ When client initiate the HTTP request it has some specification and parameter based on which server will resolve the query and send HTTP response back to the client.

Basic features:

1. HTTP is connectionless:
    - Once the HTTP request and HTTP response is completed between the client and server, even the same client initiate the HTTP request to the same server it has to make the new connection has to be established.
2. HTTP is stateless:
    - Suppose once client made HTTP request and that has been resolved by serve then when the same client creating the next HTTP request , this HTTP request doesn't have any information about the previous HTTP session.



URL info:

- It's a Uniform Resource Locator
- Every URL has 4 parameters:
    1. Protocol
    2. Hostname
    3. Port
    4. Path and file name
- In HTTP, server is always listening on the HTTP port 80 for incoming client requests.

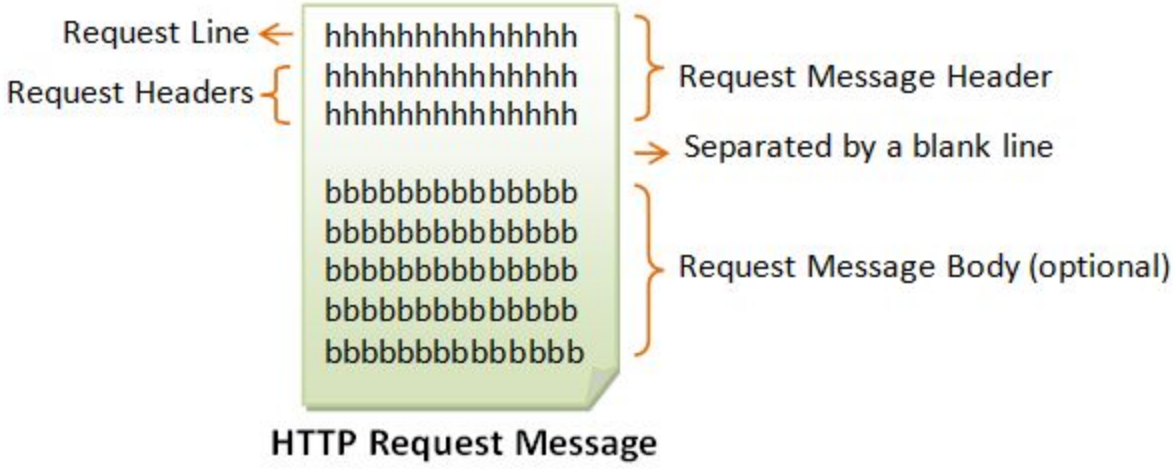
★ Whenever, we write something in browser URL will be created and then the browser will translate the URL in the HTTP request message.

★ Whenever server gets the HTTP request it will perform any of the following three actions.

1. Find the file name path from the HTTP request and locate it into the server and returns the requested file to the client.
2. Map the request to the program kept in the server, execute the program and return the output of the program to the client.
3. Server is unable to resolve the query then, send an error message to the client.

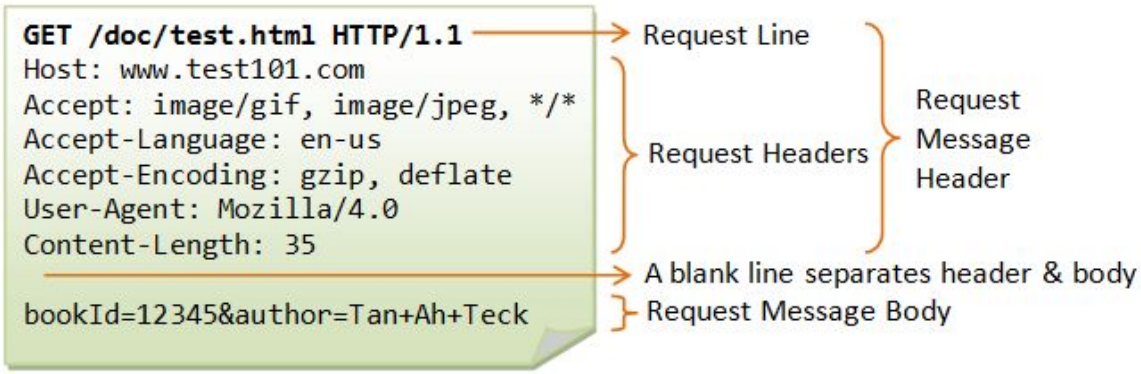
HTTP client and server communicating by sending text messages, client send request message to the server and server reply back with the response message.

<u>HTTP Request Messages:</u>

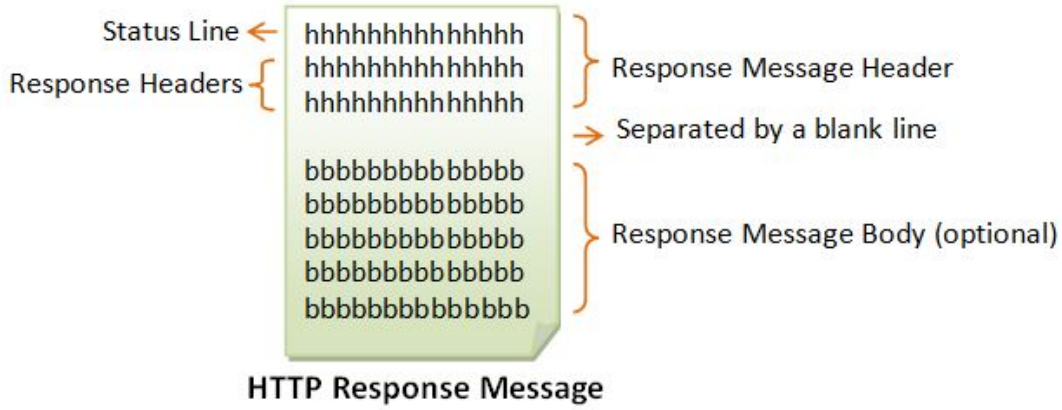


The first line of the request message header is called request line.
Request line: request-method-name request-URI HTTP-version.
Ex. GET /test.html HTTP/1.1

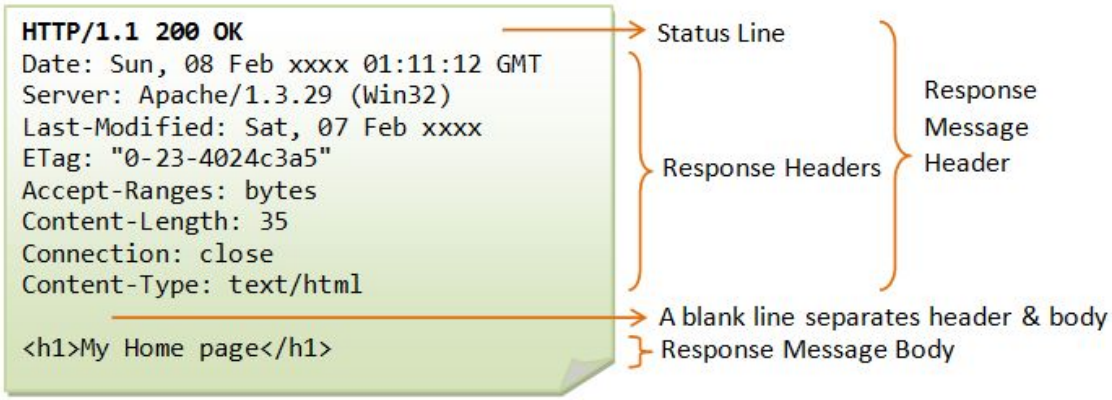
The Request header is key:value pairs. There are lots of parameters in Request header.
Host: www.xyz.com
Connection: Keep-Alive
Accept: image/gif, image/jpeg, */*
Accept-Language: us-en, fr, cn



HTTP Response Message :



First line is called the status line followed by the response headers.
Status line: ***`HTTP-version status-code reason-phrase`***
Ex. **`HTTP/1.1 200 OK`**



HTTP Request Methods:

GET : to get the web resources from the server
HEAD: If the information you need really is metadata about a resource that can be represented nicely in the HTTP headers, or to check if the resource exists or not, HEAD might work nicely. However, if we are using the REST API to check the existence about the resources then GET method will work fine.
PUT: Ask the server to store the data (new one)
POST: Tell the server to update the existing value of data point.
TRACE: Ask the server to send the diagnostic of the action that it takes with the HTTP request
DELETE: Ask the server to delete the data.
OPTIONS: Ask the server to return the list of methods it supports.
CONNECT: This is often used to make the SSL connection through the proxy.

Difference between the HTTP/1.0 and HTTP/1.1 header version?

HTTP/1.0 = In this header version the TCP connection will be terminated after the client request will be resolved by the server. So every HTTP request has to make the new TCP connection individually. By default the connection is not persistent but still if you want to make the tcp connection persistent using the HTTP/1.0 then we can use the optional header [connection: keep-alive]
HTTP/1.1 = Where as in HTTP/1.1 the connection:keep-alive is by default.
HTTP/2.0 =

- HTTP/2 is binary, instead of textual
- It is fully multiplexed, instead of ordered and blocking
- Speed increase reduces additional round trip times (RTT), making your website load faster without any optimization.

Response status code:
The status code is a 3-digit number:

1. 1xx (Informational): Request received, server is continuing the process.
2. 2xx (Success): The request was successfully received, understood, accepted and serviced.
3. 3xx (Redirection): Further action must be taken in order to complete the request.
4. 4xx (Client Error): The request contains bad syntax or cannot be understood.
5. 5xx (Server Error): The server failed to fulfill an apparently valid request.

Some commonly encountered status codes are:
100 Continue: The server received the request and in the process of giving the response.
200 OK: The request is fulfilled.
301 Move Permanently: The resource requested for has been permanently moved to a new location. The URL of the new location is given in the response header called Location. The client should issue a new request to the new location. Application should update all references to this new location.
302 Found & Redirect (or Move Temporarily): Same as 301, but the new location is temporarily in nature. The client should issue a new request, but applications need not update the references.
304 Not Modified: In response to the If-Modified-Since conditional GET request, the server notifies that the resource requested has not been modified.
400 Bad Request: Server could not interpret or understand the request, probably a syntax error in the request message.
401 Authentication Required: The requested resource is protected, and require client's credentials (username/password). The client should re-submit the request with his credentials (username/password).
403 Forbidden: Server refuses to supply the resource, regardless of the identity of client.
404 Not Found: The requested resource cannot be found in the server.

405 Method Not Allowed: The request method used, e.g., POST, PUT, DELETE, is a valid method. However, the server does not allow that method for the resource requested.
408 Request Timeout:
414 Request URI too Large:
500 Internal Server Error: Server is confused, often caused by an error in the server-side program responding to the request.
501 Method Not Implemented: The request method used is invalid (could be caused by a typing error, e.g., "GET" misspell as "Get").
502 Bad Gateway: Proxy or Gateway indicates that it receives a bad response from the upstream server.
503 Service Unavailable: Server cannot response due to overloading or maintenance. The client can try again later.
504 Gateway Timeout: Proxy or Gateway indicates that it receives a timeout from an upstream server.

HTTPS: [https://robertheaton.com/2014/03/27/how-does-https-actually-work/]

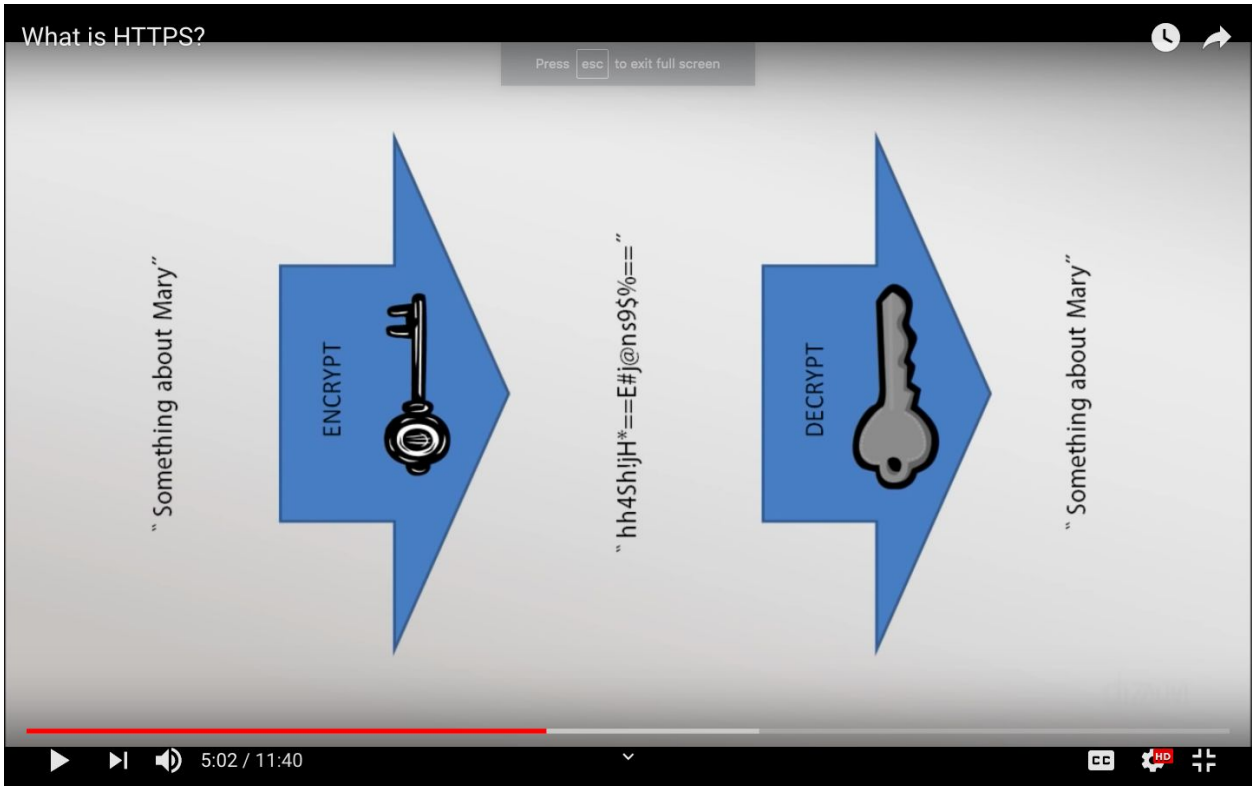
It is a simple HTTP with the extra layer of SSL/TLS encryption as well as decryption.

HTTPS also uses TCP (Transmission Control Protocol) to send and receive data packets, but it does so over port 443, within a connection encrypted by Transport Layer Security (TLS)., SSL.

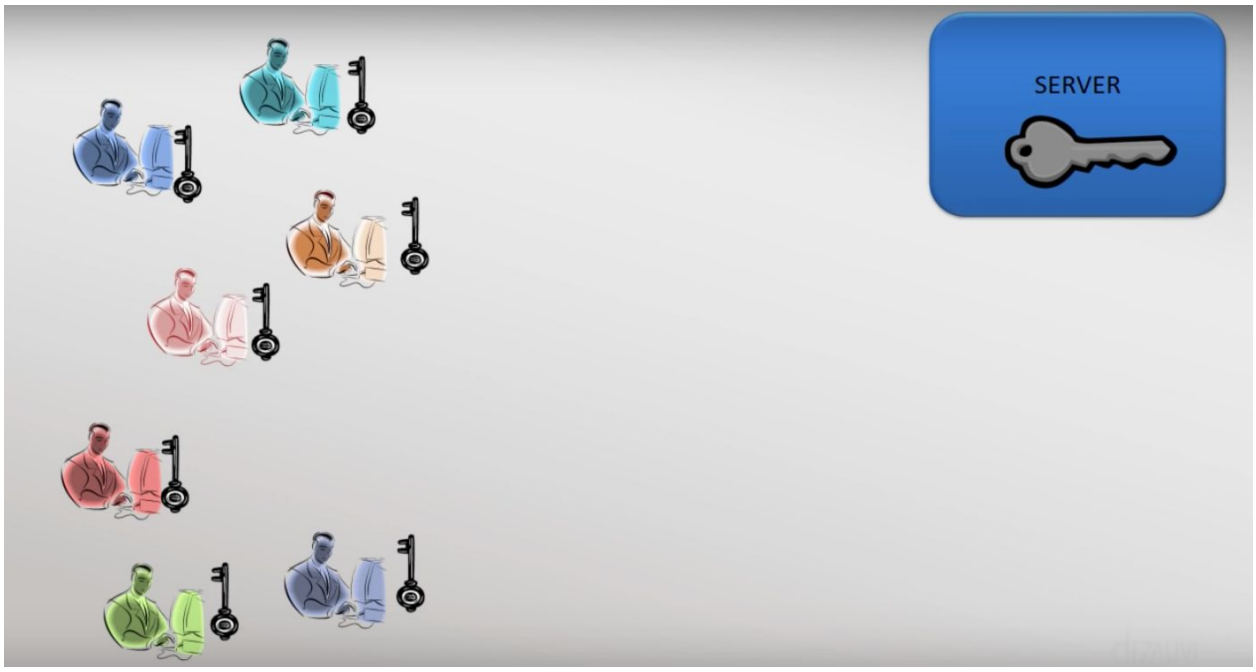
Cryptography:

Encryption is converting the plain text into the cipher that is hard to read and interpret it.
Encryption algorithm is called **cipher.**
Sometime to encrypt the data encryption algorithm use the another string called “key string”.
So then client will send the cipher text to the server and then the server can decrypt the cipher text using the same algorithm that used to encrypt the data. [**plus the *“Key String”* which is used along with the encryption algorithm**]

Symmetric key: When the same key is used to encrypt as well as decrypt the data is called a symmetric key.



Asymmetric key: But there is a way, we can encrypt the data using one “key string” and decrypt the data using another “key string”. This is called the asymmetric key.
Here the key used for the encryption is called the “public key” and the key used for decryption is called the “private key”.



Server has the private key and client has the public key used to encrypt the data by the client.

Let's see it using the SSL protocol.

There are primary two purposes of the SSL:

- Verifying that client is talking to the same server that thinks it talking to
- Ensure that the only server can read what client is sending and only client can able to understand the response from the server.

Now, when we type the URL with https,
Performs the following action.

1. Established the TCP connection with the server on port 443, because of the https.
2. Now the process of SSL handshake starts,
    - **CLIENT HELLO**: first client will send the client hello message to the server which contains the following information
        - ➔ Highest SSL version it support,
        - ➔ Ciphers that client supports
        - ➔ Compression method that client supports.
        - ➔ And some random data that can be later used to create the **symmetric key**
    - **SERVER HELLO:** Now, server will send server hello message to the client which contains the following information. In the server hello message, what are the parameter should be used for the session is defined.
        - ➔ SSL version used for the session
        - ➔ Cipher that will be used during the session
        - ➔ Compression method and the session ID for the session.
        - ➔ Some random data, which will be used during the key generation process.
    - **CERTIFICATE [FROM server to client]:** Now server will send it's digital certificate which is digitally signed by the signing authority like very sign.
      This process has two main purposes.
        1. In this certificate, there will be a public key of the server will be available.
        2. Who issued the certificate and to whom it has been issue and its validity dates so browser can verify the server identity.
    - **SERVER Hello done:** Server will send this message to client as a hint to start the communication of the data from client side.
    - **Certificate verify:** client will send this message to the server saying that the provided certificate is verified.
    - **Change Cipher spec:** client will send this message saying that now onward the HTTPS data will be transferred in encryption using the SSL session from client side.
    - **Client Finished:** client will send this to the server with the DIGEST data, which contains the information about all the messages that has been exchanges with

the server till that time. So server can validate that none of the command or messages is tempered by any outsiders or attacker.

- **Change cipher sec:** Now, server can send this message saying that now onwards all the data from the server side will be encrypted.
- **Server Finished:** same way server will also send the finished message and DIGEST. So Client can validate none of the message or data is tempered.
  #### At this moment the SSL handshake is completied ######

- **Symmetric secret key creation:** Client will create the symmetric key and encrypt it using the public key of the server and only server can able to decrypt it. Symmetric key is lighter for the transmission data.
- **Now all the data transfer between the client and server is encrypted and decrypted using the Symmetric Secret Key.**

  ### If there will be any issue and any messages is tempered then server or client will raise the error #####

SSL has 2 version, : SSL 2 and SSL 3 and TLS has 1,1.1,1.2,1.3 versions.

What is the difference between SSL and TLS?

So what's the difference between SSL and TLS? In conversation, not much and many people continue to use the term SSL. In terms of your server configuration though, it's the difference between vulnerabilities, outdated cipher suites and browser security warnings. When it comes to your servers, you should only have TLS protocols enabled. SSL also supported by the server , but in order to check that we have to run the SSL server test.

What is the difference between the HTTP and HTTPS?

In HTTP, URL begins with “http://” whereas URL starts with “https://”
HTTP uses port number 80 for communication and HTTPS uses 443
HTTP is considered to be insecure and HTTPS is secure
HTTP Works at Application Layer and HTTPS works at Transport Layer, because for HTTPS we required to complete the SSL handshake first to transfer the traffic data and SSL is layer 4 protocol.
In HTTP, Encryption is absent and Encryption is present in HTTPS as discussed above
HTTP does not require any certificates and HTTPS needs SSL Certificates

IPSec:

IPsec is a framework that secure communication at the network or packet processing layer. It can be used to secure one or more data flows between the peers.
IPsec provides the feature like

- Confidentiality
- Integrity
- Origin authentication and
- Anti-reply

IPsec consists of two main protocols.

1. Authentication Header (AH)
    - In this protocol, the IP header and the data is hashed. Using the hash, AH header is created and appended to the packet and create the new packet which contains the [AH Header + IP header + data payload] .
    - Now this new packet will reach the router and it will hash the header and payload. Now both of the hash from sender and receiver (in this case router) has to be matched, even a single bit is changed the AH Header will not match.
2. Encapsulation Security Payload (ESP)
    - ESP also performed at the IP packet layer.
    - This security protocol does provide encryption and integrity of the data packet.
    - The ESP is added after the standard IP header. As it contains the IP header it can be routed as normal IP traffic through the IP routers. Hence, those device who does not support IPsec can also able to carry this packet along with the ESP.
    - It contains six parts,
      Only this two parts are authenticated,
        1. Security Parameter Index
        2. Sequence Number

      Whereas the rest of the part are encrypted during the transmission
        3. Payload data
        4. Padding
        5. Pad length
        6. Next Header
    - It supports multiple encryption protocol and its upto user which one to opt.

Encryption technology:

1. **Tunnel mode:**
    - This encrypt both the IP header and the payload.
    - IPsec with the tunnel mode used when the destination of the packet is different from the security termination point.
    - This tunnel mode is used between two gateways or between endpoint and gateways. [But not between the two actual end points.]
2. **Transport mode:**
    - In this mode, only data portion of each packet is encrypted.
    - This IPsec transport mode can be used between two end points or between endpoint and gateway.

How Does IPsec Work?

- IPsec make uses of tunneling
- But its working can be broken into five major steps.
    1. Interesting Traffic Initiation:
        - The traffic which looks sensitive or deemed interesting considered to be sent using the IPsec.
          How does the data is sensitive or not decide?
          For example, in Cisco routers, access lists can be used to make decisions about encryption of packets by way of crypto map sets.
          We create one more access list stating that if any egress traffic allowed to transmit outside then , router must encrypt the data packet or else send unencrypted data packet.
          When traffic needed to use IPsec, IKE phase one is triggered.
    2. IKE Phase One:
        - In this step, first IPsec will authenticate both end point peers thus protecting the identity of the peers.
        - Then Internet Key Exchange and the Security Association policy is negotiated between the peers.
        - As a result, both the peers have shared secret key which will help in the IKE phase two.
        - In phase the secure tunnel will be created through which the information of the next phase will exchanges securely.

- This phase has to mode,
  1. Main Mode : There are three exchanges between the initiator and receiver.
     - 1st is to exchange the algorithm and hashes
     - 2nd is to generation of the shared secret key (Using Diffie-Hellman exchange)
     - 3rd exchange is for verifying the identity.

  , 2. Aggressive mode:
     - This is faster but not secure, all the information before the tunnel creation is open and risk vulnerable.

  [That's why we used Main mode as a best practice.]

3. IKE Phase two:
   - This phase negotiates information for IPsec SA [In first phase which policy has to be used is negotiated] parameters through the IKE SA.
   - Here as well IPsec policies are shared and then establish IPsec SAs.
   - There is only a single mode (quick mode) in this phase. It exchanges nonce providing replay protection. These nonces generate new shared secret key material. If the lifetime for IPsec expires, it can renegotiate a new SA.
4. Data Transfer:
   - Here the data is safely and securely transmitted through the IPsec tunnel.
   - The sent packets are encrypted and decrypted using the specified encryption in the IPsec SAs.
5. Tunnel Termination:
   - The tunnel may terminate by either deletion or by time out. Time out occurs when the specified time (sec) has passed or when specified number of bytes will have passed through the tunnel.

Firewall/proxy:

Firewall is a network security device which continuously monitor the incoming and outgoing traffic for our network. And based on the security rules configured in our firewall either it will allow or block the data packets. In short, a firewall is used to protect our private internal network from the external Internet network.

We are configured set of security rules for different port through which internet or outside traffic is communicating to our private network.
Firewall will check certain parameters of the packet like Source IP , destination IP or source port or destination port or protocol of the packet then scan all the security rules. Firewall take the decision about the packet.

What are the different type of Firewalls?

Generally, Firewall can be categorized based on their general structure and method of operations.

1. Packet Filtering Firewalls:
    - This firewall creates the check-points at traffic router or switch or any other networking device.
    - Firewall just perform the simple inspection for the packet coming through this checkpoints.
    - The Firewall only check the surface level information like, Source and Destination IP, Source and Destination Port and packet type.
    - This type of firewall doesn't look for the datagram or content of the packet.
    - If the packet failed the inspection then packet will be dropped.
2. Circuit-level gateways firewalls:
    - This type of firewall just verify the TCP handshake. It is just to make sure that the source of the TCP handshake is legitimate.
    - This firewall also do not check the content of the packet hence there is malware content in the packet but TCP handshake is from a trustworthy source then Circuit-level gateway firewalls can't detect it.
3. Stateful Inspection Firewall:
    - This kind of firewall includes both packet level inspection and TCP handshake verification which provides more security. Not the content

4. Next-Generation Firewalls:
    - This kind of firewall perform deep inspection of the packet including the content of the packet, surface level inspection and TCP handshake verification.
    - They also do have extra features like Intrusion Detection System (IDS), as well as intrusion prevention system (IPS). Before we implement this firewall we need to understand what are the features available and supported by that specific firewall.
5. Software Firewall:
    - This type of firewalls are separately installed in any local device to make an individual secure as compared to the rest of the network.
    - We are using this kind of software firewall only for the critical devices.
    - Example, Macfee, Norton, Malwarebytes , AVG.
6. Hardware Firewall:
    - This type of firewalls are Installed physically inside our network and then we create the check points for our hardware firewall and it will monitor the incoming and outgoing traffic on those check-points.

What is the difference between the Stateful and Stateless Firewall?

- The key difference between Stateless and Stateful firewall is that Stateful firewall only inspect the packet based on the surface level information like it's source and destination IP, source and destination port, and match the security rules and either or allow the packet or block the packet.
- Whereas, Stateful firewall perform deep inspection of the packet, TCP handshake verification and surface level information checks. Here deep inspection means it will look inside the packet and content of the datagram. Along with this stateful firewall also able to track the pattern of the packets. In case firewall gets any malicious packet from specific source and once when it gets another packet from the same source statull firewall will block that packet without even further inspection.

What is a proxy server?

- Proxy server does work as an intermediate gateway between the host inside our private network and internet. Proxy server is hide the identity of our internal network from the rest of the internet.
- Proxy servers provide varying levels of functionality, security, and privacy depending on your use case, needs, or company policy.

- What happens is that when we are sending any request to web server , proxy server wrap it with the different Source IP address and then forward the packet to the actual web server. And the same way it forward the response to the true origin or the request. So the rest of the world can not able to track the origin of the packet. Using this mechanism proxy server can hide the identity of the host of our LAN.
- Proxy server also cache the web server response. Which will increase the network performance by reducing the bandwidth usage.
- Proxy server does also support the extra layer of encryption for our network traffic.

Reverse proxy used by the server side.

What is the difference between the proxy server and firewall?

The firewall and proxy server both are one type of network security devices.
But there are three major differences between firewall and proxy server.

1. Functionality
   A firewall essentially blocks communication, while a proxy server simply redirecting the traffic to the different endpoints.
2. Blocking Website:
   Using firewall we may not be able to block the access to the specific website. Where as using the proxy server we may able to redirect the web request of some websites for a certain amount of time period to some other end point s in our network. Form user endpoint, it looks like website is blocked. However, proxy server does have an internal mechanism to redirect web requests to specific web page inside our network.
3. Blocking Program:
   Firewall does have the capability to block certain programs from running in our local users system. Where as proxy server does not have such kind of feature. Proxy server is working for our network as a whole.

SSH:

- Ssh is a secure replacement of the telnet, which send the traffic data without any encryption. When we are using the telnet any one can read the content what we are sending between the client and remote server.
- Where as, SSH is encrypt the traffic data using some cryptographic mechanism before transmitting it.
- For windows we have to use the SSH client like putty, Linux and MacOS can directly do ssh from the terminal.
- You can use the SSH command with the combination of the [username+remote_ip] or [username+remote_hostname]
- Symmetric key used to encrypt the entire communication during the SSH session.
- Both the client and server create the symmetric key individually using an agreed method. The process of creating the symmetric key is carried out by the Key Exchange Algorithm.
- [**During the SSH the symmetric key is never transmitted between the client and server which makes SSH more secure.**]
- **Both the client and server pre decided the Key Exchange Algorithm, then they both use some random public data and separately create the symmetric key by applying KEA on the public data.**
- **So, even if someone gets this public data they can not able to create the symmetric key because the third-party device does not which KEA is used to create the symmetric key for that SSH session.**

- During the SSH when we are typing the password , even that password is transferred in encrypted form.

- Some of the symmetric encryption ciphers = AES, CAST128, BlowFish.

In symmetric encryption , the symmetric key is called as a **shared key#####**
In asymmetric encryption, the keys called , **public key and private key**.####

- In SSH, server is listening on port 22 for SSH connection.
- To complete the SSH process, first TCP handshake will be done between the client and server.
- Which algorithm is used to create the symmetric key for SSH session?
  [Diffie-Hellman Key Exchange Algorithm]

There are two stages to establishing a connection:

1. both the systems must agree upon encryption standards to protect future communications.
2. the user must authenticate themselves. If the credentials match, then the user is granted access.

1. *Session encryption negotiation*

How does the algorithm works at a very basic level?

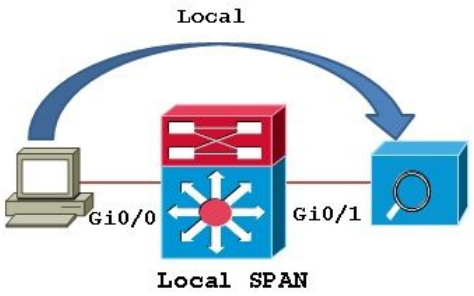
- Both the client and the server agree on a very large prime number, which of course does not have any factor in common. This prime number value is also known as the seed value.
- Next, the two parties agree on a common encryption mechanism to generate another set of values by manipulating the seed values in a specific algorithmic manner. These mechanisms, also known as encryption generators, perform large operations on the seed. An example of such a generator is AES (Advanced Encryption Standard).
- Both the parties independently generate another prime number. This is used as a secret private key for the interaction.
- This newly generated private key, with the shared number and encryption algorithm (e.g. AES), is used to compute a public key which is distributed to the other computer.
- The parties then use their personal private key, the other machine's shared public key and the original prime number to create a final shared key. This key is independently computed by both computers but will create the same encryption key on both sides. Now that both sides have a shared key, they can symmetrically encrypt the entire SSH session. The same key can be used to encrypt and decrypt messages (read: section on symmetrical encryption).

2. *Authenticating the User*
    - *These credentials securely pass through the symmetrically encrypted tunnel, so there is no chance of them being captured by a third party.*

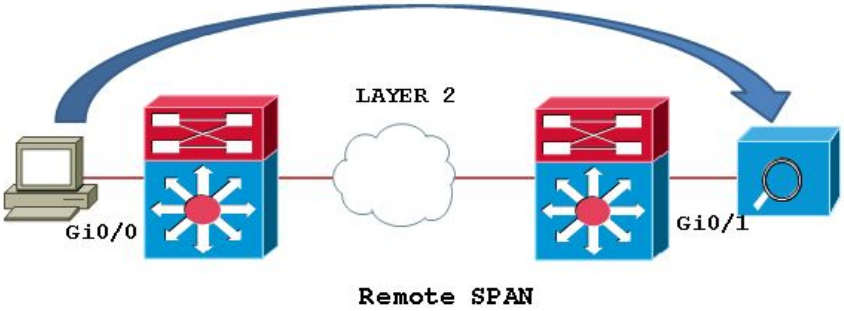
Telnet:

SPAN: [Switch port analyzer]

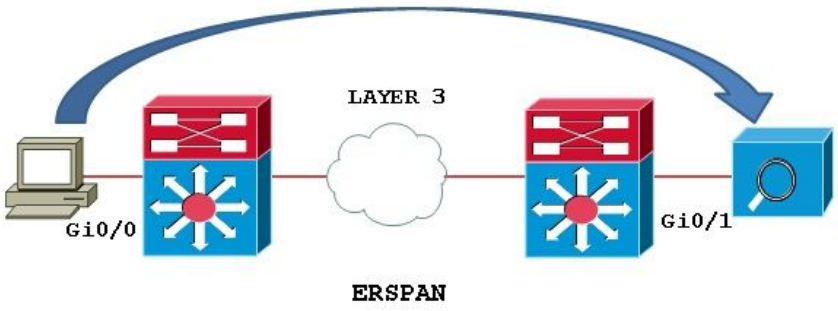
- It is an efficient and high performance traffic monitoring system.
- In case we have configured switch with SPAN on fastethernet port 12 as a source of traffic and fastethernet port 10 as a destination of the SPAN, Then what will happen, whenever any traffic come to port 12 it will be duplicated and also sent out to the destination SPAN device as well as to the actual original destination of the traffic.
- SPAN is used for troubleshooting the network connectivity issue, calculating the network utilization and performance.
- Generally on cisco devices there three types of SPAN supported.
    1. Local SPAN:
        - Duplicate the traffic from one or more interface of the switch and forward that to one or more port of switch which are configured as SPAN destination ports.



    2. Remote SPAN:
        - It is nothing but just an extension of the SPAN.
        - Using the RSPAN we can duplicate the traffci from the multiple ports of the switch and then froward that traffic using a aspecifc VLAN configured for that RSPAN session and transfer the traffic over the switches using trunking to the final destination device of the SPAN.



3. Encapsulated Remote SPAN: (ERSPAN):
    - In ERSPAN, first we create the GRE encapsulated tunnel between our switch and monitoring device. And then just duplicate the traffic from one or more port of the switch and forward that traffic directly to the final destination of the SPAN process using that GRE tunnel.



SMTP: [Simple Mail Transfer Protocol]

-

SNMP:

FTP:

VRRP & HSRP:

VLAN & VxLAN:

LADP:

CDP:

LACP:

VTP:

ACL:

GRE Tunnel:

AAA:

802.1q:

MPLS: