NETWORK SECURITY ESSENTIALS

BCA – IV

Credits – 4

Evaluations – 5

The course is designed to build an understanding of various network security components, protocols and creating the awareness about the issues due to security.

Pre-requisites: An understanding of Basic Computer Networking and security

UNIT 3

Security issues in Internet protocols: TCP, DNS, and routing

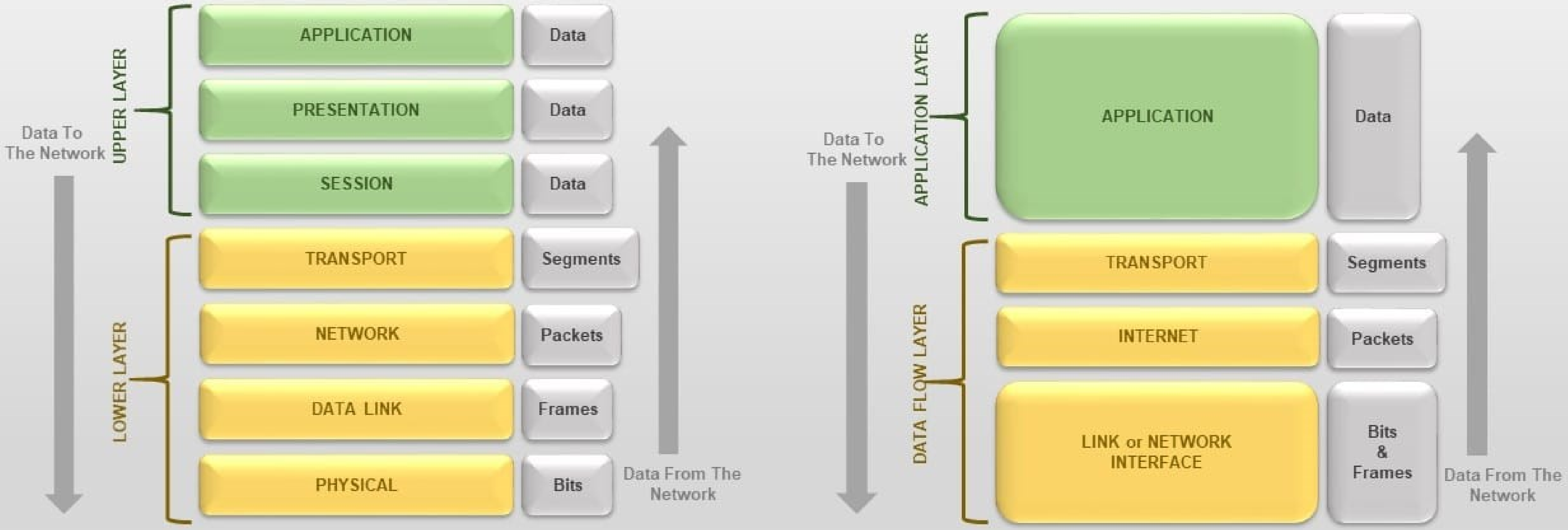
TCP – TRANSMISSION CONTROL PROTOCOL

TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation through which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules defining the Internet.

TCP is a connection-oriented protocol, which means a connection is established and maintained until the application programs at each end have finished exchanging messages. It determines how to break application data into packets that networks can deliver, sends packets to and accepts packets from the network layer, manages flow control and -- because it is meant to provide error-free data transmission -- handles retransmission of dropped or garbled packets and acknowledges all packets that arrive.

In the Open Systems Interconnection (OSI) communication model, TCP covers parts of Layer 4, the transport layer, and parts of Layer 5, the session layer.

OSI MODEL vs TCP/IP MODEL



Data To The Network

UPPER LAYER

APPLICATION — Data

PRESENTATION — Data

SESSION — Data

LOWER LAYER

TRANSPORT — Segments

NETWORK — Packets

DATA LINK — Frames

PHYSICAL — Bits

Data From The Network

Data To The Network

APPLICATION LAYER

APPLICATION — Data

DATA FLOW LAYER

TRANSPORT — Segments

INTERNET — Packets

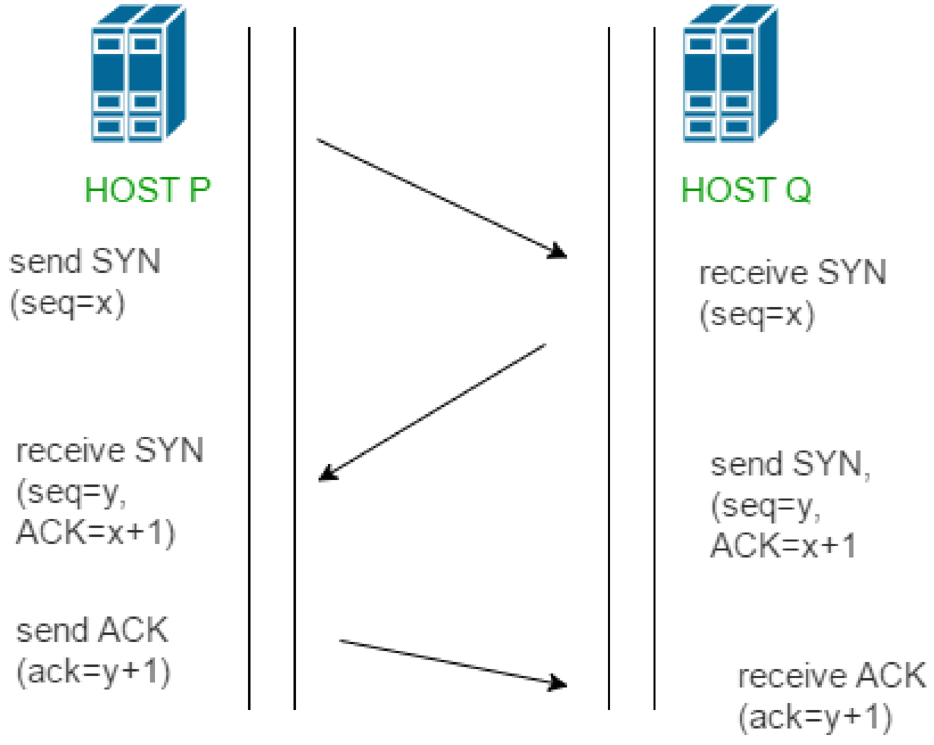
LINK or NETWORK INTERFACE — Bits & Frames

Data From The Network

TCP is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. Three-way handshake (active open), retransmission, and error-detection adds to reliability but lengthens latency.

Applications that do not require reliable data stream service may use the User Datagram Protocol (UDP), which provides a connectionless datagram service that prioritizes time over reliability. UDP is used in multicast, whereas TCP can’t be used in multicast.

CONNECTION ESTABLISHMENT – 3 WAY HANDSHAKE

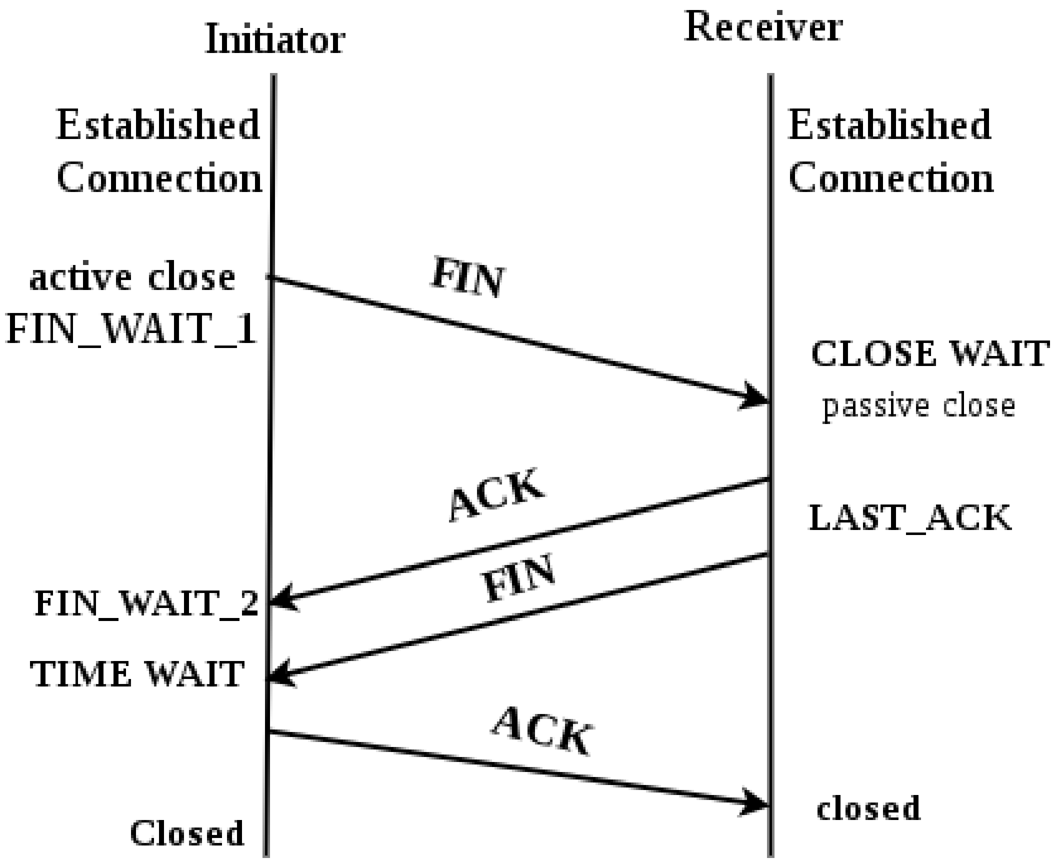


- **Step 1 (SYN) :** In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with

- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with

- **Step 3 (ACK) :** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer. The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged. The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged. With these, a full-duplex communication is established.

CONNECTION TERMINATION



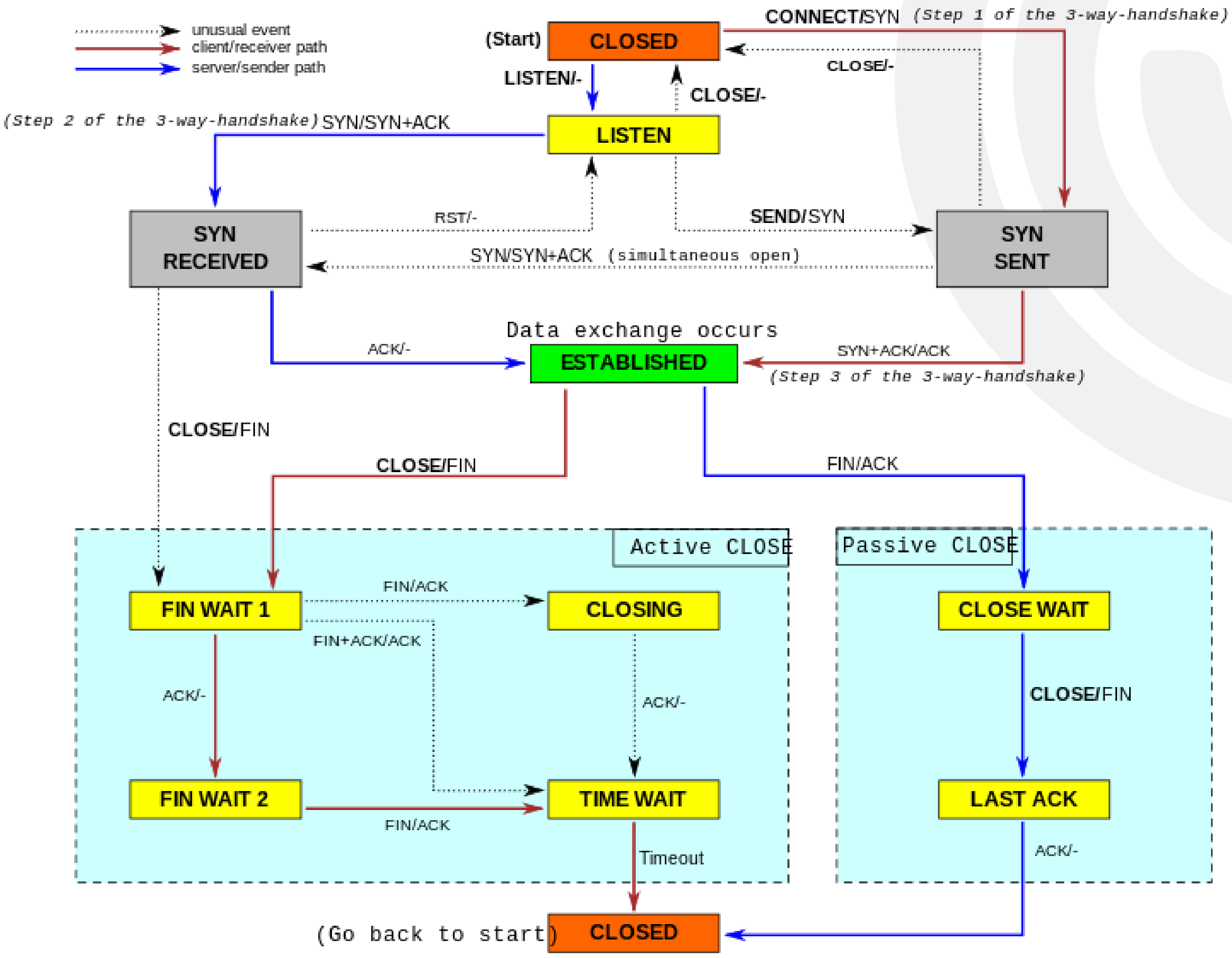
1.Step 1 (FIN From Client) –
Suppose that the client application decides it wants to close the connection. (Note that the server could also choose to close the connection). This causes the client send a TCP segment with the **FIN** bit set to **1** to server and to enter the **FIN_WAIT_1** state. While in the **FIN_WAIT_1** state, the client waits for a TCP segment from the server with an acknowledgment (ACK).

2.Step 2 (ACK From Server) –
When Server received FIN bit segment from Sender (Client), Server Immediately send acknowledgement (ACK) segment to the Sender (Client).

3.Step 3 (Client waiting) –
While in the **FIN_WAIT_1** state, the client waits for a TCP segment from the server with an acknowledgment. When it receives this segment, the client enters the **FIN_WAIT_2** state. While in the **FIN_WAIT_2** state, the client waits for another segment from the server with the FIN bit set to 1.

4.Step 4 (FIN from Server) –
Server sends FIN bit segment to the Sender(Client) after some time when Server send the ACK segment (because of some closing process in the Server).

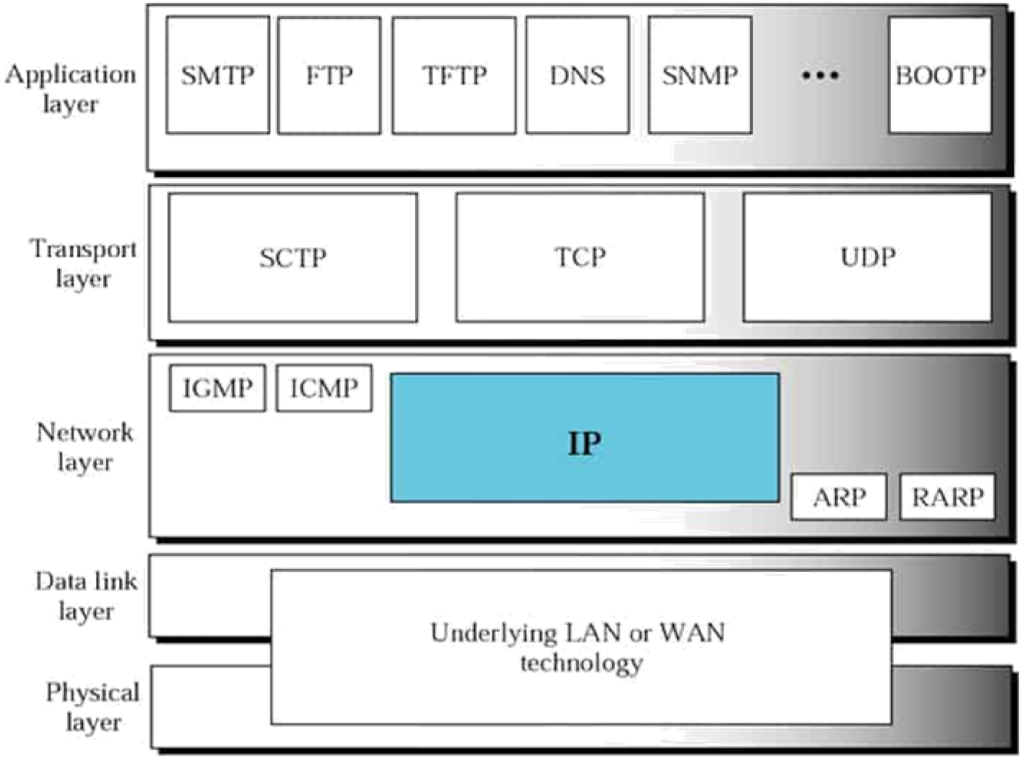
5.Step 5 (ACK from Client) –
When Client receive FIN bit segment from the Server, the client acknowledges the server's segment and enters the **TIME_WAIT** state. The **TIME_WAIT** state lets the client resend the final acknowledgment in case the **ACK** is lost.The time spent by client in the **TIME_WAIT** state is depend on their implementation, but their typical values are 30 seconds, 1 minute, and 2 minutes. After the wait, the connection formally closes and all resources on the client side (including port numbers and buffer data) are released.



TCP State diagram

During the lifetime of a TCP connection, the local end-point undergoes a series of state changes

INTERNET PROTOCOL



The **Internet Protocol** (**IP**) is the principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its **routing function** enables internetworking, and essentially establishes the Internet.

Routing is the process of selecting a path for traffic in a network or between or across multiple networks. Broadly, routing is performed in many types of networks, including circuit-switched networks, such as the public switched telephone network (PSTN), and computer networks, such as the Internet.

The design of the Internet protocol suite adheres to the end-to-end principle. Under the end-to-end principle, the network infrastructure is considered inherently unreliable at any single network element or transmission medium and is dynamic in terms of the availability of links and nodes. No central monitoring or performance measurement facility exists that tracks or maintains the state of the network. For the benefit of reducing network complexity, the intelligence in the network is purposely located in the end nodes.

As a consequence of this design, the **Internet Protocol only provides best-effort delivery and its service is characterized as unreliable. It is a connectionless protocol**, in contrast to connection-oriented communication. Various fault conditions may occur, such as data corruption, packet loss and duplication. Because routing is dynamic, meaning every packet is treated independently, and because the network maintains no state based on the path of prior packets, different packets may be routed to the same destination via different paths, resulting in out-of-order delivery to the receiver.

All fault conditions in the network must be detected and compensated by the participating end nodes. **The upper layer protocols of the Internet protocol suite are responsible for resolving reliability issues.**

THE ADVANTAGES OF TCP/IP PROTOCOL SUITE

•It is an industry–standard model that can be effectively deployed in practical networking problems.

•It is interoperable, i.e., it allows cross-platform communications among heterogeneous networks.

•It is an open protocol suite. It is not owned by any particular institute and so can be used by any individual or organization.

•It is a scalable, client-server architecture. This allows networks to be added without disrupting the current services.

•It assigns an IP address to each computer on the network, thus making each device to be identifiable over the network. It assigns each site a domain name. It provides name and address resolution services.

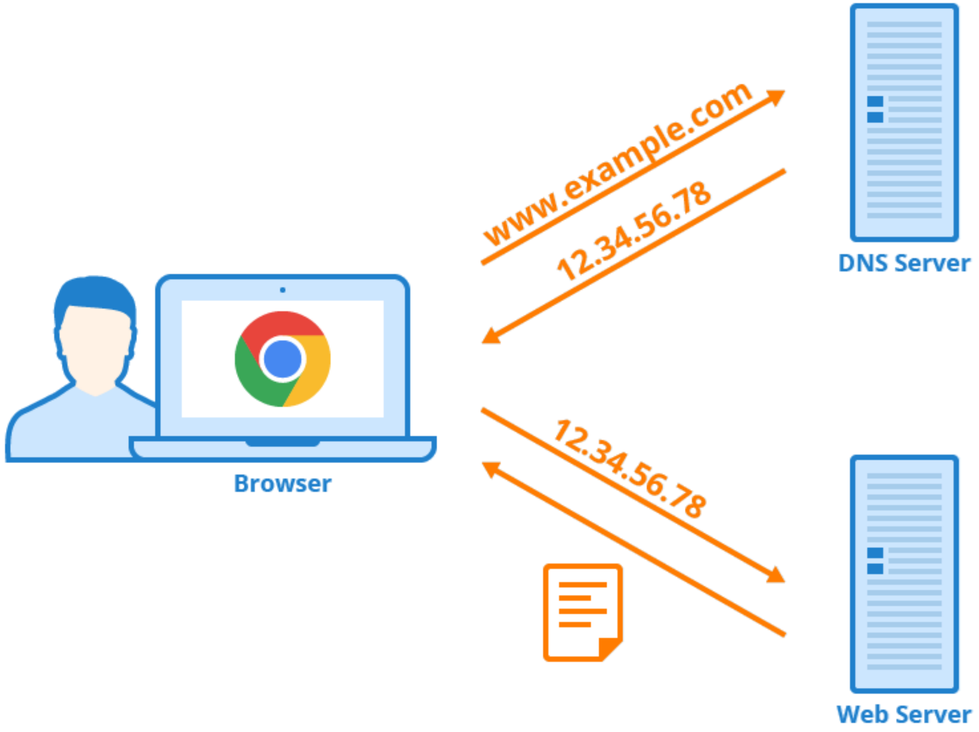
THE DISADVANTAGES OF THE TCP/IP MODEL

•It is not generic in nature. So, it fails to represent any protocol stack other than the TCP/IP suite. For example, it cannot describe the Bluetooth connection.

•It was originally designed and implemented for wide area networks. It is not optimized for small networks like LAN (local area network) and PAN (personal area network).

•Among its suite of protocols, TCP and IP were carefully designed and well implemented. Some of the other protocols were developed ad hoc and so proved to be unsuitable in long run. However, due to the popularity of the model, these protocols are being used even 30–40 years after their introduction.

DOMAIN NAME SYSTEM (DNS)



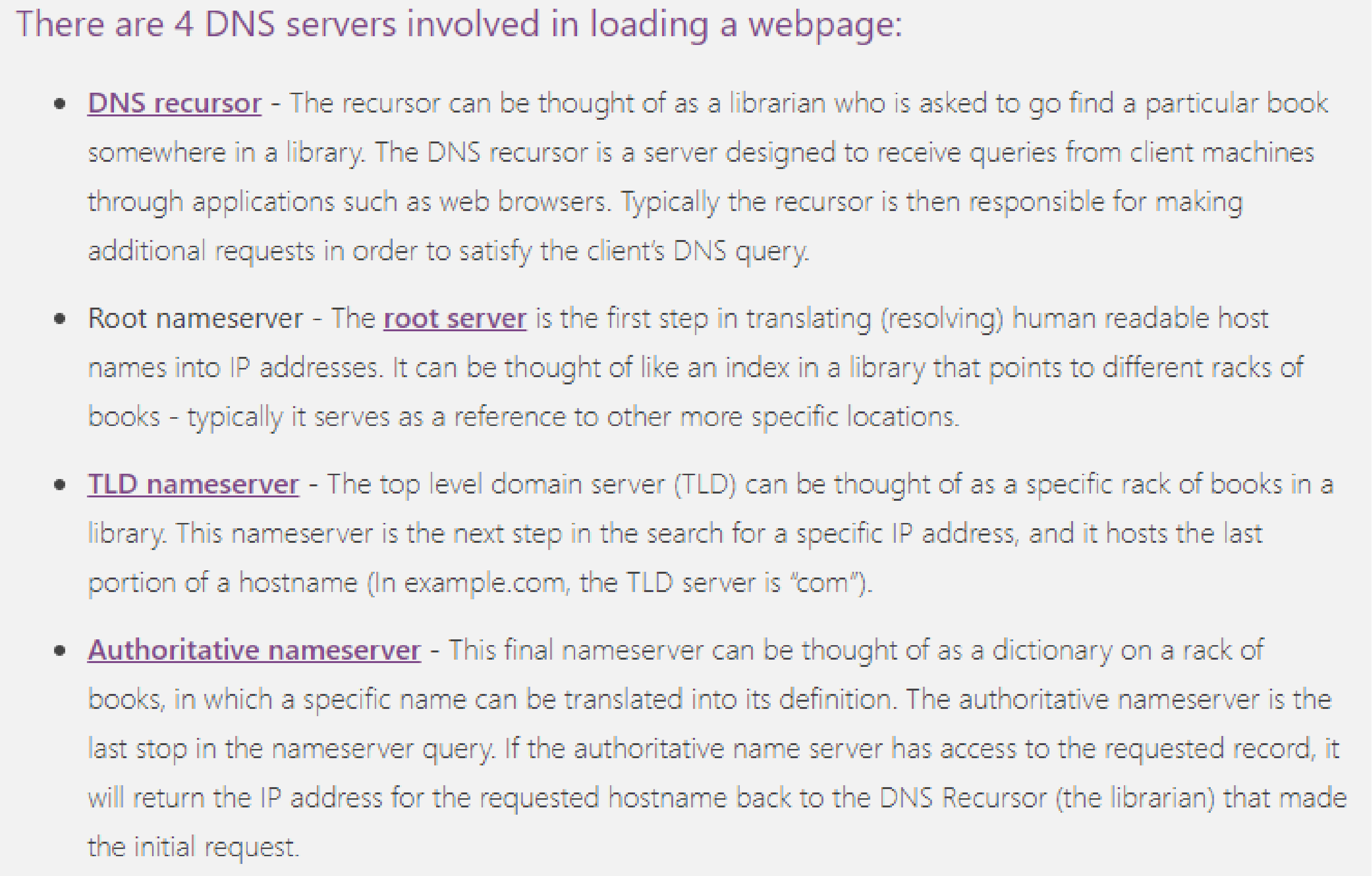
The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like google.com or yahoo.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

How does DNS work?

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

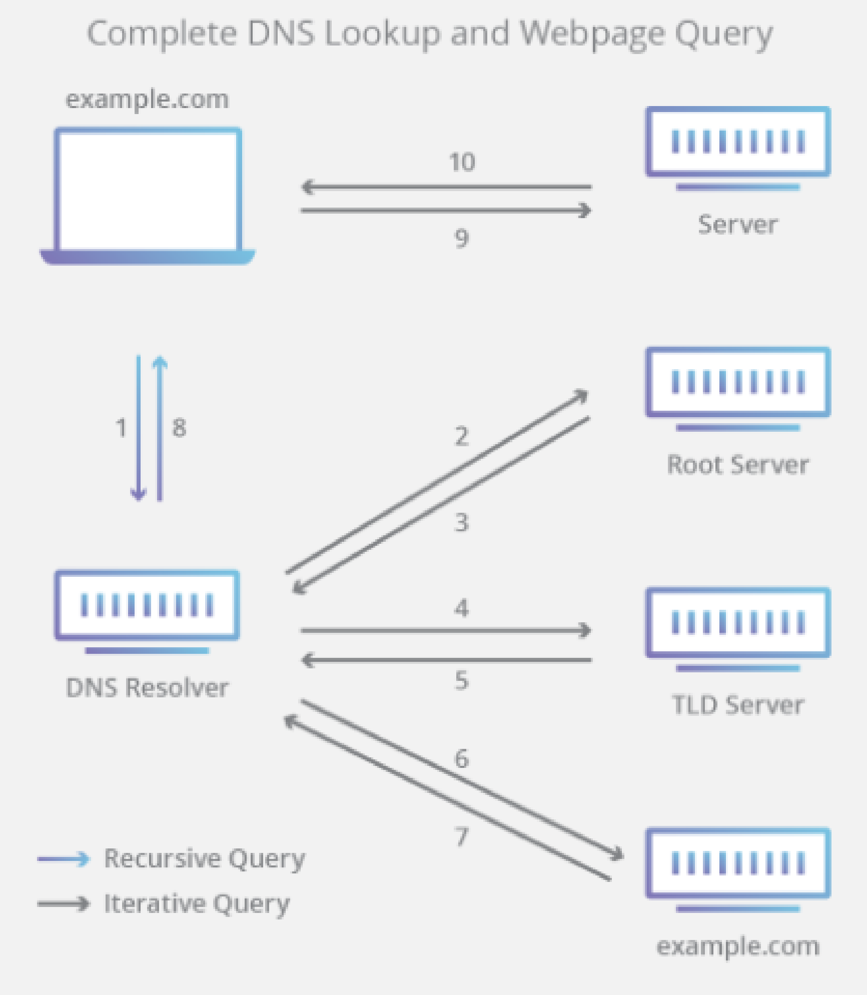
In order to understand the process behind the DNS resolution, it's important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs " behind the scenes" and requires no interaction from the user's computer apart from the initial request.



There are 4 DNS servers involved in loading a webpage:

- DNS recursor - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the recursor is then responsible for making additional requests in order to satisfy the client's DNS query.
- Root nameserver - The root server is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.
- TLD nameserver - The top level domain server (TLD) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is "com").
- Authoritative nameserver - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

Steps in DNS Lookup



1. A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
2. The resolver then queries a DNS root nameserver (.).
3. The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
4. The resolver then makes a request to the .com TLD.
5. The TLD server then responds with the IP address of the domain’s nameserver, example.com.
6. Lastly, the recursive resolver sends a query to the domain’s nameserver.
7. The IP address for example.com is then returned to the resolver from the nameserver.
8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

10. The browser makes a HTTP request to the IP address.
11. The server at that IP returns the webpage to be rendered in the browser (step 10).

Note: Often DNS lookup information will be cached either locally inside the querying computer or remotely in the DNS infrastructure. There are typically 8 steps in a DNS lookup. When DNS information is cached, steps are skipped from the DNS lookup process which makes it quicker.

TYPES OF DNS QUERIES

DNS queries are the computer code that tells the DNS servers what kind of query it is and what information it wants back. There are three basic DNS queries in a standard DNS lookup.

•**Recursive query:** In a recursive query the computer requests an IP address or the confirmation that the DNS server doesn't know that IP address.

•**Iterative query:** An iterative query the requester asks a DNS server for the best answer it has. If the DNS server doesn't have the IP address, it will return the authoritative name server or TLD name server. The requester will continue this iterative process until it finds an answer or times out.

•**Non-recursive query:** A DNS resolver will use this query to find an IP address that it doesn't have in its cache. These are limited to a single request to limit network bandwidth usage.

DNS CACHE + CACHING FUNCTIONS

DNS Cache is a repository of domain names and IP addresses that are stored on a computer, so it doesn't have to ask for the IP address every time. Imagine if every time any user tried to go to www.abcd.com, DNS had to query the authoritative name server at ABCD. The traffic would be overwhelming! The very thought of that much traffic is why we have DNS caching.

DNS caching has two major goals:
•Speed up DNS requests
•Reduce bandwidth of DNS requests across the internet

The DNS cache methodology does have some issues, however:
•DNS changes need time to propagate – meaning it could be a while before every DNS server has their cache updated to latest IP data
•DNS cache is a potential attack vector for hackers

There are a few different types of DNS caching used on the internet:
•**Browser DNS caching:** Current browsers circa 2018 have built in DNS caching functionality. Resolving a DNS with the local cache is fast and efficient.
•**Operating System (OS) DNS caching:** Your computer is a DNS client, and there is a service on your computer that manages DNS resolution and requests. This DNS cache is also local and therefor fast and requires no bandwidth.
•**Recursive resolving DNS caching:** Each DNS recursor has a DNS cache, and it stores any IP address that it knows to use for the next request

ROUTING

Packets, which are the atomic unit of information in packet-switched communication networks, are exchanged between the nodes (a node might be an end device, a router or a data generating device, etc.).

The process of transferring these packets of information from their source node to the destination node with one or more hops in between along the most optimum path is called as ‘Routing’. Routers and switches are the devices that are used for the purpose which work on the routing protocols and algorithms they are configured with.

These packets are taken care of by the L3 layer of the OSI Reference Model’s network layer.

When a packet is introduced in the network and received by one of the routers, it reads the packet’s headers to understand the destination and checks its routing table marked with its metrics to see what would be the next best hope for the packet to reach the destination optimally. Then, it pushes the packet to the next node, and the above process repeats at the new node too until the packet reaches the destination node.

Routing metrics

These tables have information based on which packet switching takes place in the most optimal path. And this information is different metrics or variables which the routing algorithms look for and then decide their path. The standard metrics include –

1.**Path Length:** In this, the administrator will assign costs to each path (between two nodes). The path length will be the sum of all the path costs. The path with the less path length will be chosen as the most optimal one.

2.**Delay:** This is the measure of time it takes for the packet to route from source to destination. This depends on many factors like network bandwidth, the number of intermediate nodes, congestion at nodes, etc. Sooner the transfer, the better the Quality of Service (QoS).

3.**Bandwidth:** This refers to the amount of data a link can transfer through it. Usually, the enterprise lease the network line to achieve a higher link and bandwidth.

4.**Load:** Load refers to the traffic which a router or a link is handling. The unbalanced or unhandled load might cause congestion and a lower rate of transmission packet losses.

5.**Communication Cost:** This is the operational expense which the company incurs by sending the packets on the leased line between the nodes.

6.**Resilience and Reliability:** This refers to the error handling capacity of the router and the routing algorithms. If some nodes in the network fail, then the resilience and reliability measure will show us how well the other nodes can handle the traffic.

1. Static Routing

This type is the optimal path between all possible pairs of sources & destinations in the given network is pre-defined and fed into the routing table of the network's routers.

Advantages

•There is no CPU overhead for the routers to decide the next hop for the packet as the paths are predefined.

•This offers higher security as the administrator has autonomy over packet flow permissions along a defined path.

•Between the routers, no bandwidth would be used (for tasks like updating its table, etc.)

Disadvantages

•It will be difficult for the administrator to identify and pre-define an optimal path from all possible combinations of source & destination nodes for larger network topology.

•The administrator would be expected to be thorough in the concepts of networks and topology. Transition to a new administrator would consume time so as understand the topology and policies that are defined.

2. Dynamic Routing

This type gives the router the ability to discover the network by protocols like OSPF (Open Shortest Path First) and RIP (Routing Information Protocol), updates the routing table by itself and effectively decides upon the path that the incoming packet must follow to reach its destination.

Advantages

•This is easy to configure.

•It would be efficient in order to discover some remote network and execute routing there.

Disadvantages

•When one of the routers in the network implementing dynamic routings discovers change or generates an update, it broadcasts it to all the nodes. Thus, consuming a higher amount of bandwidth.

•It is relatively less secure than static.

Types of Routing Algorithms

There are two types of algorithms:

1. Adaptive

The routes are decided dynamically based on the changes in the network topology.

- **Distance Vector Routing:** In this algorithm, each router maintains it's a table containing an entry for each router in the network. These entries are updated periodically. This is also called the Bellman-Ford Algorithm. Originally, this was the ARPANET algorithm.
- **Link State Routing:** LSR discovers the neighbours, measures the cost to each neighbour, then constructs the packets and sends them along the computed shortest path.

2. Non-Adaptive

The routes are decided in a static fashion by the routers.

- **Flooding:** In this, you send the packets to every other neighbouring router & they in-turn to the same, and by some path, the packet reaches its destination. This duplicates the packets, but the reliability is very high in a type of routing. This is mostly used in defense networks, distributed databases, wireless networks, and populating the routing tables.

SECURITY ISSUES WITH INTERNET PROTOCOLS

TCP/IP protocol suite is not perfect. There exists a number of serious security flaws inherent in the protocol design or most of TCP/IP implementation. The network hackers just utilize these security holes to perform various network attacks. Among the hacking techniques, three of them are commonly used and reflect some typical problems in TCP/IP protocol suite.

Refer to the document TCP_IP networkattacks.pdf shared on Teams

SECURITY ISSUES WITH DNS

DNS is organized into a tree-like infrastructure where the first level contains topmost domains, such as *.com* and *.org*. The second level nodes contain general, traditional domain names. The ‘leaf’ nodes on this tree are known as hosts. DNS works similar to a database which is accessed by millions of computer systems in trying to identify which address is most likely to solve a user’s query.

Types of Attacks:

1.Denial of service (DoS) –

An attack where the attacker renders a computer useless (inaccessible) to the user by making a resource unavailable or by flooding the system with traffic.

2.Distributed denial of service (DDoS) –

The attacker controls an overwhelming amount of computers (hundreds or thousands) in order to spread malware and flood the victim’s computer with unnecessary and overloading traffic. Eventually, unable to harness the power necessary to handle the intensive processing, the systems will overload and crash.

3.DNS spoofing (also known as DNS cache poisoning) –

Attacker will drive the traffic away from real DNS servers and redirect them to a “pirate” server, unbeknownst to the users. This may cause in the corruption/theft of a user’s personal data.

Types of attacks (Contd.)

4.Fast flux –

An attacker will typically spoof his IP address while performing an attack. Fast flux is a technique to constantly change location-based data in order to hide where exactly the attack is coming from. This will mask the attacker's real location, giving him the time needed to exploit the attack. Flux can be single or double or of any other variant. A single flux changes address of the web server while double flux changes both the address of web server and names of DNS serves.

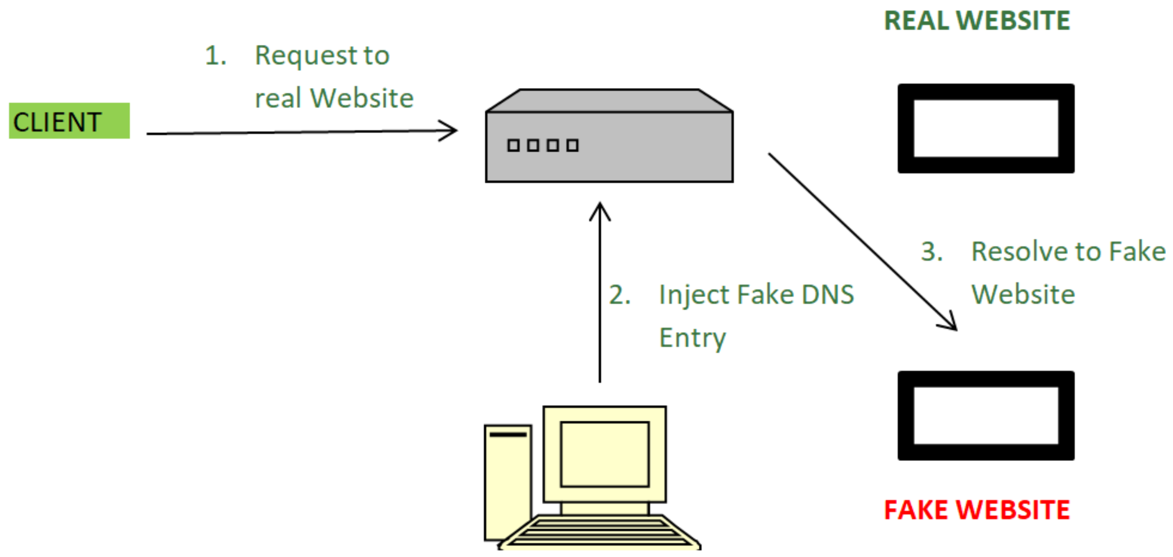
5.Reflected attacks –

Attackers will send thousands of queries while spoofing their own IP address and using the victim's source address. When these queries are answered, they will all be redirected to the victim himself.

4.Reflective amplification DoS –

When the size of the answer is considerably larger than the query itself, a flux is triggered, causing an amplification effect. This generally uses the same method as a reflected attack, but this attack will overwhelm the user's system's infrastructure further.

DNS SPOOFING OR DNS CACHE POISONING



DNS Spoofing

DNS Spoofing means getting a wrong entry or IP-address of the requested site from DNS server. Attackers find out the flaws in DNS system and take control and will redirect to a malicious website.

Image –

1.Request to Real Website: User hit a request for particular website it goes to DNS server to resolve the IP-address of that website.

2.Inject Fake DNS entry: Hackers already take control over the DNS server by detecting the flaws and now they add false entry in DNS server.

3.Resolve to Fake Website: Since fake entry in DNS server redirect user to wrong website.

To Prevent From DNS Spoofing –

DNS Security Extensions (DNSSEC) is used to add an additional layer of security in DNS resolution process to prevent security threats such as DNS Spoofing or DNS cache poisoning.

DNSSEC protects against such attacks by digitally ‘signing’ data so you can be assured it is valid.

DNS TUNNELING

What is DNS Tunneling?

DNS Tunneling is a strategy for a digital exploit that encodes the information of different programs or protocols in DNS inquiries and responses. DNS tunneling frequently incorporates information payloads that can be added to an exploited domain name server and used to control a distant system and applications.

Why DNS Tunneling is a Problem?

DNS was initially made for name resolution and not for data exchange, so it’s regularly not seen as a malignant interchange of information and data exfiltration danger.

Since DNS is entrenched and confided protocol, attackers realize that organizations seldom investigate DNS packets for malevolent movement. DNS has less consideration and most organizations center assets around breaking down web or email traffic where they believe, attacks regularly occur.

As a general rule, constant endpoint checking is needed to discover and forestall DNS tunneling. Besides, tunneling application bundles have become an industry and are uncontrollably accessible on the Internet, so hackers don’t generally require specialized advancement to execute DNS tunneling exploits.

1.DNS tunneling exploits can give aggressors a consistently accessible back channel to exfiltrate taken information. It depends on utilizing DNS as a covert correspondence channel to sidestep a firewall.

2.Hackers tunnel various types of protocols like SSH or HTTP with DNS, at that point secretly pass stolen information or passage IP traffic.

3.A DNS tunnel can be utilized as a full controller channel for an already exploited inside host. This lets secretly the hackers move records out of the organization, download new code to the existing malware, or have total distant admittance to the servers, etc.

4.DNS tunnels can likewise be utilized to bypass captive portals, to abstain from paying for Wi-Fi services.

5.DNS tunneling utilizes the DNS protocol to tunnel the malware and other information through a client-server model.

ATTACKS ON ROUTING PROTOCOLS

Wireless Sensor Network (**WSN**) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.

Attacks on routing protocols: most of the routing protocols for WSNs are vulnerable to various types of attacks. Some of these attacks are listed below.

- Routing table overflow: in this type of attack, an adversary node advertises routes to non-existent nodes, to the authorized node present in the network. The main objective of such an attack is to cause an overflow of the routing tables, which would in turn prevent the creation of entries corresponding to new routes to authorized nodes. Proactive routing protocols are more vulnerable to this attack compared to reactive routing protocols.

- Routing table poisoning: in this case, the compromised nodes in the network send fictitious routing updates or modify genuine route update packets sent to other honest nodes. Routing table poisoning may result in sub-optimal routing, congestion in some portions of the network, or even make some parts of the network inaccessible.

- Packet replication: in this attack, an adversary node replicates stale packets. This consumes additional bandwidth and battery power and other resources available to the nodes and also causes unnecessary confusion in the routing process.

- Route cache poisoning: in reactive (i.e. on-demand) routing protocols such as ad hoc on-demand distance vector (AODV) (Perkins, et al., 1999), each node maintains a route cache which holds information regarding routes that have become known to the node in the recent past. Similar to routing table poisoning, an adversary can also poison the route cache to achieve similar objectives.

- Rushing attack: on-demand routing protocols that use duplicate suppression during the route discovery process are vulnerable to this attack (Hu et al., 2003b). An adversary node which receives a route request packet from the source node floods the packet quickly throughout the network before other nodes which also receive the same route request packet can react. Nodes that receive the legitimate route request packets assume those packets to be duplicates of the packet already received through the adversary node and hence discard those packets. Any route discovered by the source node would contain the adversary node as one of the intermediate nodes. Hence, the source node would not be able to find secure routes, that is, routes that do not include the adversary node. It is extremely difficult to detect such attacks in WSNs

Some of the sniffing-based attacks, an enterprise routing infrastructure can easily be attacked with man-in-the-middle and other attacks designed to corrupt or change the routing tables with the following results:

- Traffic redirection—In this attack, the adversary is able to redirect traffic, enabling the attacker to modify traffic in transit or simply sniff packets.
- Traffic sent to a routing black hole—Here the attacker is able to send specific routes to null0, effectively kicking IP addresses off of the network.
- Router denial-of-service (DoS)—Attacking the routing process can result in a crash of the router or a severe degradation of service.
- Routing protocol DoS—Similar to the attack previously described against a whole router, a routing protocol attack could be launched to stop the routing process from functioning properly.
- Unauthorized route prefix origination—This attack aims to introduce a new prefix into the route table that shouldn't be there. The attacker might do this to get a covert attack network to be routable throughout the victim network.