SNS COLLEGE OF TECHNOLOGY

(Autonomous)

COIMBATORE – 35

DEPARTMENT OF COMPUTER SIENCE AND ENGINEERING (UG & PG)

Third Year Computer Science and Engineering, 5th Semester

UNIT I – CYBER SCURITY FUNDAMENTALS

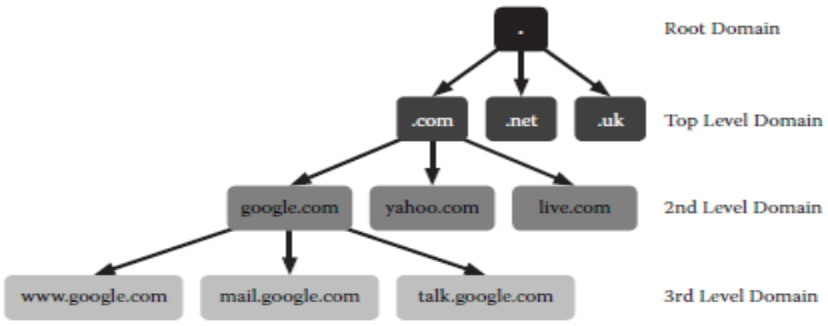
Topic Name : Domain Name System (DNS) & Firewall

Domain Name System (DNS)

This section explains the fundamentals of the domain name sys-tem (DNS), which is an often overlooked component of the Web's infrastructure, yet is crucial for nearly every networked application. Many attacks, such as fast-flux and DNS application, take advantage of weaknesses in the DNS design that emphasize efficiency over security. Later sections will discuss some attacks that abuse the DNS and will build upon the base information provided in this section.

DNS is a fundamental piece of the Internet architecture. Knowledge of how the DNS works is necessary to understand how attacks on the system can affect the Internet as a whole and how criminal infrastructure can take advantage of it. The Internet Protocol is the core protocol the Internet uses. Each computer with Internet access has an assigned IP address so that other systems can send traffic to it. Each IP address consists of four numbers between 0 and 255 separated by periods, such as 74.125.45.100.

These numbers are perfect for computers that always deal with bits and bytes but are not easy for humans to remember. To solve this problem, the DNS was invented in 1983 to create easy-to-remember names that map to IP address. The primary goal that the designers of the DNS had in mind was scalability. This goal grew from the failure of the previous solution that required each user to download a multithousand -line file named hosts.txt from a single server.



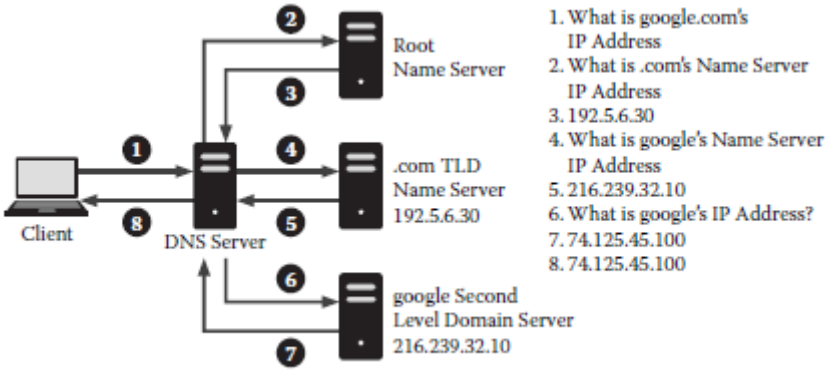
The hierarchical structure of the domain name system (DNS).

The DNS uses computers known as name servers to map domain names to the corresponding IP addresses using a database of records. Rather than store information for every domain name in the system, each DNS server must only store the information for its domain. For instance, the name server gotgoogle.com keeps information for www. google.com and mail.google.com but not for www.yahoo.com. Name servers are granted authority over a domain by the domain above them, in this case .com. When a name server has this authority, it aptly receives the title of authoritative name server for that domain.

The hierarchical nature that defines the DNS is also a key to the resolution process. Resolution is the process of mapping a domain to an IP address, and resolvers are the programs that perform this function. Due to the nature of the resolution process, resolvers fall into two categories: recursive and nonrecursive.

The below diagram show the steps required for a resolver to complete this process. The first step in resolving www.google.com is contacting the root name server to find out which name server is authoritative for .com. Once the resolver has this information, it can query the .com name server for the address of the google.com name server. Finally, the resolver can query the google.com name server for the address of www.google.com and pass it on to a Web browser or other program.

The below diagram depicts the most common way for systems to resolve domain names: by contacting a recursive DNS server and allowing it to do the work. A nonrecursive resolver (like the one used by a home PC) will only make a single request to a server, expecting the complete answer back. Recursive resolvers follow the chain of domains, requesting the address of each name server as necessary until reaching the final answer. Using recursive DNS servers also makes the system much more efficient due to caching. Caching occurs when a DNS server already knows what the answer to a question is, so it does not need to look it up again before responding to the query. The addresses of the root server and the .com server are usually cached due to the frequency with which systems request them. The DNS stores information in Resource Records (RR). These records are separated by type, and each one stores different information about a domain.

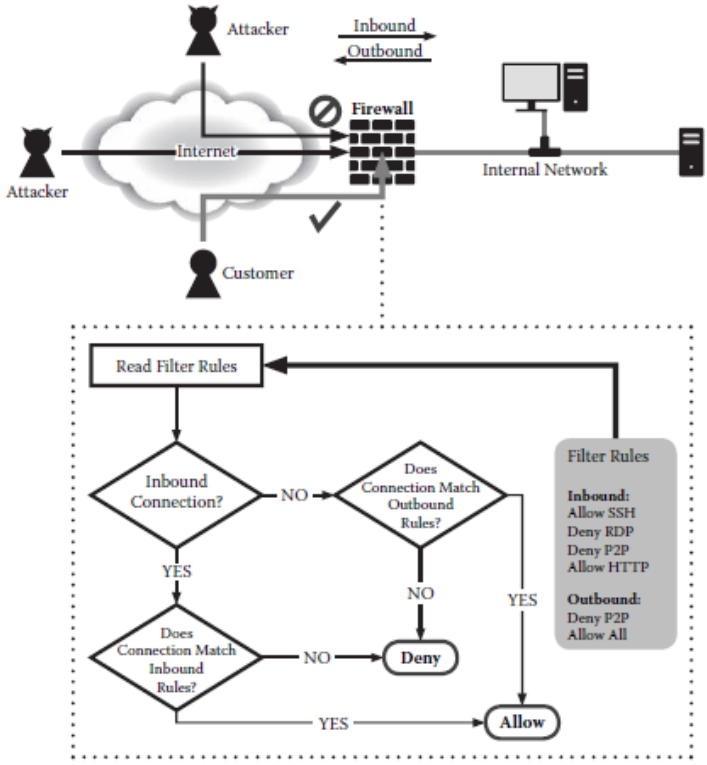


Resolution of google.com using a recursive DNS server.

A record for www.google.com in the Internet class (specified by IN). During the development of DNS, additional classes were created, but the Internet class is the only one commonly used today. The answer session includes the information from the question, the IP address for the domain, and a time-to-live (TTL) value. The TTL specifies the number of seconds for which the data in the record are valid. This value is the key to the caching system described above; as without a TTL, the servers would not know how long any particular data could be cached.

Firewalls

Firewalls are network devices or software that separates one trusted network from an untrusted network (e.g., the Internet) by means of rule-based filtering of network traffic.



A basic firewalled network.

There are three basic types of firewall: packet-filtering firewalls, stateful firewalls, and application gateway firewalls. While each of these different firewall types performs the same basic operation of filtering undesirable traffic, they go about the task in different manners and at different levels of the network stack.

The above diagram show identifies the firewall as a separate physical device at the boundary between an untrusted and trusted network, in reality a firewall is merely software. This does not mean that physical, separate devices are not firewalls, but merely that these devices are simply computers running firewall software. Host-based firewalls have found their way into most operating systems. Windows XP and later versions have a built-in firewall called the Windows Firewall. Linux- and Unix-based computers use ipchains or iptables (depending on the age and type of the operating system [OS]) to perform firewall functionality therefore, it is important to understand that firewalls can exist at different locations within a network, not just at the perimeter of a network.

Packet-Filtering Firewalls

The most rudimentary of firewalls is the packet-filtering firewall. Packet-filtering firewalls work at the IP level of the network. Most routers integrate this type of firewall to perform basic filtering of packets based on an IP address. The principle behind packet-filtering firewalls is that the firewall bases the decision to allow a packet from one network into another network solely on the IP address of the source and destination of the packet.

Packet-filtering firewalls can also expand on the basic principle of IP-address-only filtering by looking at the Transmission Control Protocol (TCP) or User Diagram Protocol (UDP) source and destination ports. In this mode, the firewall operates in nearly the same fashion as the packet-filtering firewalls operating on the IP address. For a packet to pass through the firewall, the source IP and port and the destination IP and port must match at least one rule in the filter list. More advanced routers and even some higher-end switches offer this functionality.

Stateful Firewalls

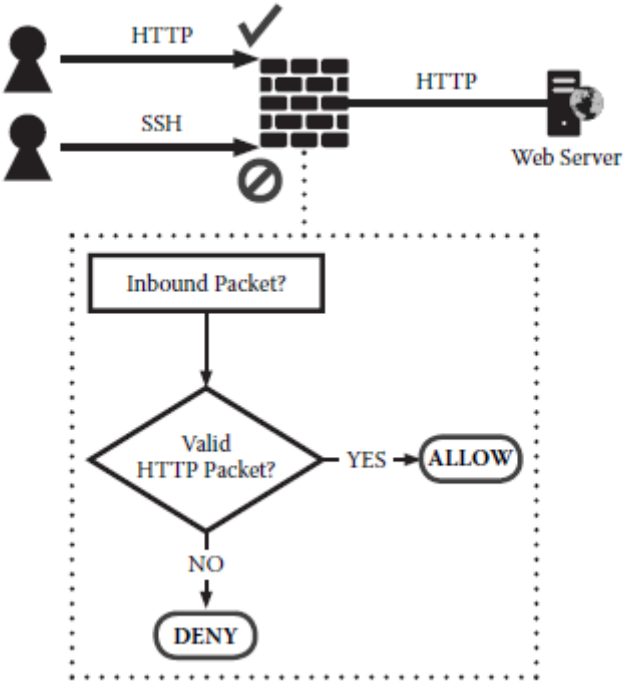
Simple packet-filtering firewalls suffer from one significant downside: they do not take into consideration the state of a connection, only the endpoints of the connection. Stateful firewalls allow only properly established connections to traverse the firewall's borders. While packet filtering is still a key element of these firewalls, the firewall also pays attention to the state of the connection. Once the firewall allows a successful connection between two hosts using the three-way TCP handshake, the firewall records the occurrence of a valid session between the two hosts. If an attacker attempts to generate an invalid session, such as by sending an ACK (acknowledgment) prior to sending a SYN (synchronize), the firewall identifies the packet as an invalid state and subsequently blocks the connection.

It is the ability to determine the order and state of a communication session that allows stateful firewalls to make faster determinations about incoming packets.

Application Gateway Firewalls

Application gateway firewalls, also known as proxies, are the most recent addition to the firewall family. These firewalls work in a similar manner to the stateful firewalls, but instead of only understanding the state of a TCP connection, these firewalls understand the protocol associated with a particular application or set of applications. A classic example of an application

gateway firewall is a Web proxy or e-mail-filtering proxy. A Web proxy, for instance, understands the proper HTTP protocol and will prevent an improperly constructed request from passing. These proxies also prevent unknown protocols from passing through. For example, a properly configured HTTP proxy will not understand an SSH connection and will prevent the establishment of the connection.



An application gateway filtering known and unknown protocols

This level of packet inspection cannot occur with either a packet-filtering or stateful firewall, as neither firewall type looks at the application layer of the network stack. By identifying improperly constructed packets for a given protocol, the application gateway firewalls may prevent some types of protocol-specific attacks; however, if a particular protocol's definition allows for such vulnerability, the gateway will provide no protection.

Firewalls come in a variety of forms, from simple packet filtering to the more complex proxy. The topic of firewalls is complex and extremely well documented. To understand the importance of firewalls, the minute details of their operation can be avoided, but it is critical to understand the high-level concepts of their operation. Understanding the basics of how firewalls process traffic and how that processing prevents unwanted intrusions is the key to understanding the security of firewalls.