Network Security

Network Layers

Open Systems Interconnection (OSI) Model

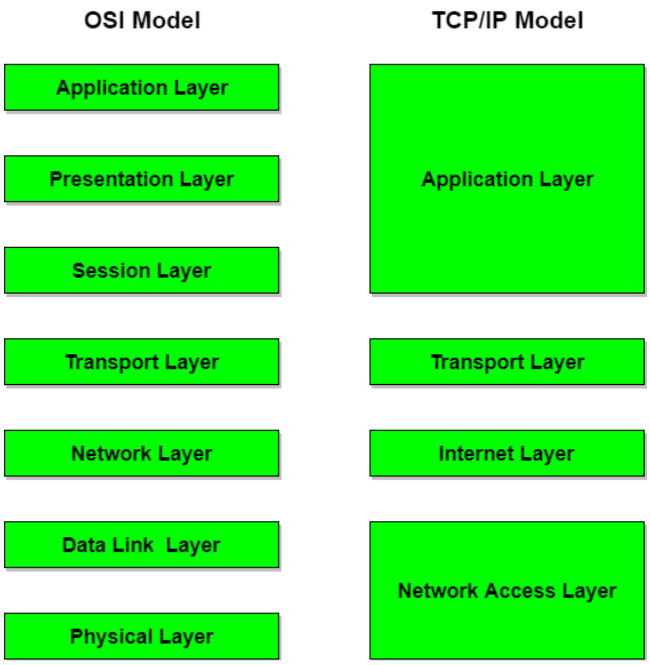
The Open Systems Interconnection (OSI) model is a conceptual/reference model that standardizes the communication functions of a telecommunication or computing system.

7 – Application
6 – Presentation
5 – Session
4 – Transport
3 – Network
2 – Data Link
1 – Physical

Transmission Control Protocol / Internet Protocol (TCP/IP) Reference Model

Application (e.g. HTTP)
Transport (e.g. TCP, UDP)
Layer 3 – IP / Internet Protocol
Layer 2 – Data Link
Layer 1 - Physical

where Layers 2 and 1 are the Network Access/Interface. In some models, these two layers are summarized into a single Network Interface layer.



OSI vs TCP/IP

So why are there two models when it comes to Network Layers? There are actually differences between them.

OSI (Open System Interconnection)	**TCP/IP (Transmission Control Protocol / Internet Protocol)**
OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user.	TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network.
In OSI model the transport layer guarantees the delivery of packets.	In TCP/IP model the transport layer does not guarantees delivery of packets. Still the TCP/IP model is more reliable.
Follows vertical approach.	Follows horizontal approach.
OSI model has a separate Presentation layer and Session layer.	TCP/IP does not have a separate Presentation layer or Session layer.
Transport Layer is Connection Oriented.	Transport Layer is both Connection Oriented and Connection less.
Network Layer is both Connection Oriented and Connection less.	Network Layer is Connection less.
OSI is a reference model around which the networks are built. Generally it is used as a guidance tool.	TCP/IP model is, in a way implementation of the OSI model.
Network layer of OSI model provides both connection oriented and connectionless service.	The Network layer in TCP/IP model provides connectionless service.
OSI model has a problem of fitting the protocols into the model.	TCP/IP model does not fit any protocol
Protocols are hidden in OSI model and are easily replaced as the technology changes.	In TCP/IP replacing protocol is not easy.
OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them. It is protocol independent.	In TCP/IP, services, interfaces and protocols are not clearly separated. It is also protocol dependent.
It has 7 layers	It has 4 layers

But in general, the differences do not really matter for this module.

Why do we use network layering?

It partitions a complex communication system into several abstraction layers. For example, we can view the layer-N protocol to be built upon a virtual connection at layer N-1. In other words, it just passes down the message, and let the level below do its job.

This is the concept called encapsulation in networking.

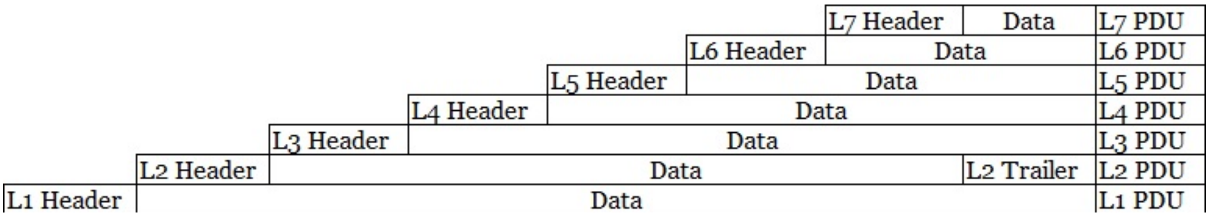
Encapsulation and Protocol Data Unit (PDU)
In networking, encapsulation is the method of designing **modular** communication protocols in which logically separate functions in the network are abstracted from their underlying structures via inclusion or information hiding within higher level objects, i.e. the lower layers are unable to discern which part of the data came from which higher level layer.

The roles and responsibilities are clear:

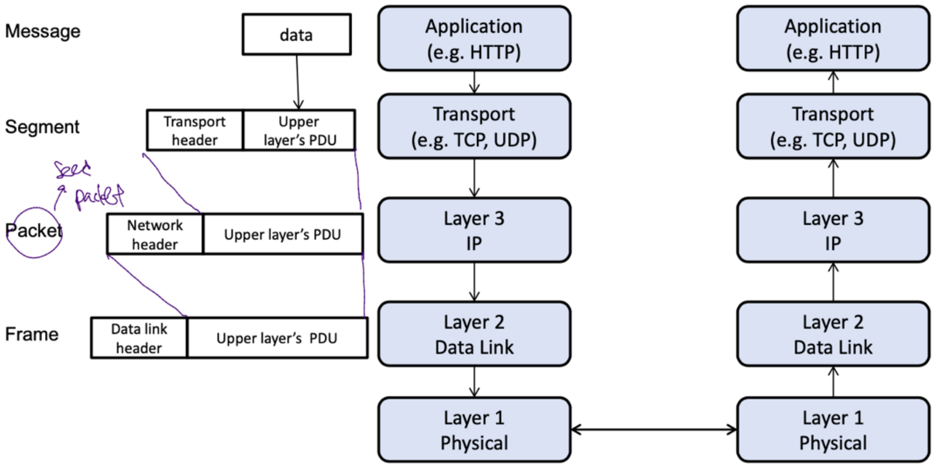
- Physical Layer: Physical transmission of the data
- Link Encapsulation Layer: Local area networking
- Internet Protocol Layer: Global addressing of local computers
- Transmission Control Protocol (Transport) Layer: Select the process or application i.e. the port which specifies the service such as a Web or TFTP server

During encapsulation, each layer builds a protocol data unit (PDU) by adding a header (and sometimes trailer) containing control information to the PDU from the layer above. For example, in the Internet Protocol suite, the contents of a web page are encapsulated with an HTTP header, then by a TCP header, an IP header, and, finally, by a frame header and trailer. The frame is forwarded to the destination node as a stream of bits, where it is decapsulated (or de-encapsulated) into the respective PDUs and interpreted at each layer by the receiving node.

The result of encapsulation is that each lower layer provides a service to the layer or layers above it, while at the same time each layer communicates with its **corresponding layer on the receiving node.** These are known as adjacent-layer interaction and same-layer interaction, respectively.



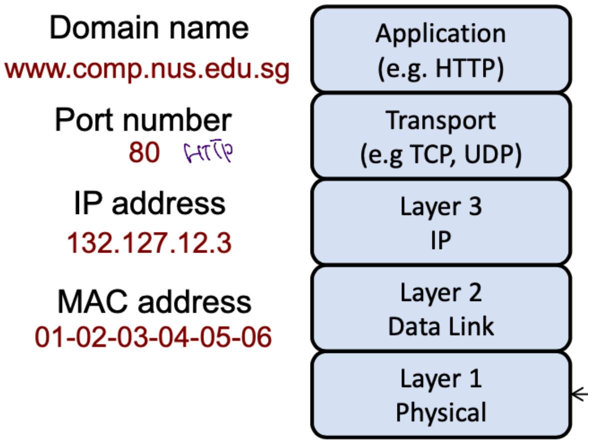
*Graphical representation of the PDUs in the **OSI model***



*Graphical representation of the PDUs in the **IP/TCP model***

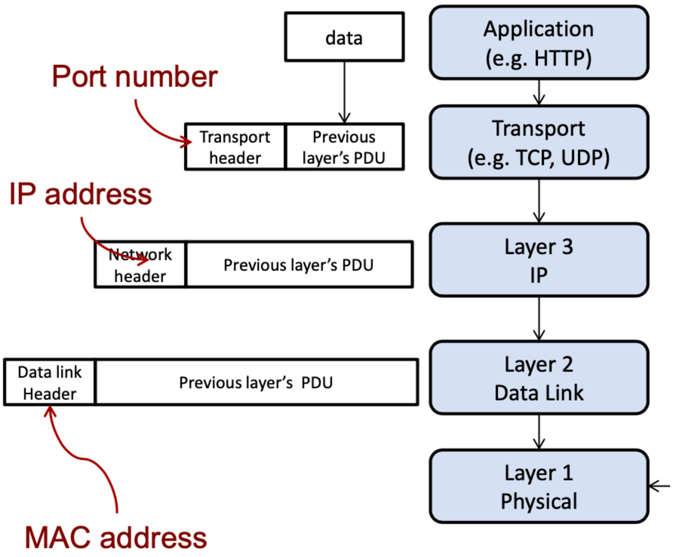
Addressing Schemes

As mentioned above, each layer has its own function in addressing. Refer to the diagrams below:



Different Addressing Schemes at Different Layers

Note: MAC here stands for Medium Access Control, not the same MAC in cryptography



Different Addressing Schemes in Headers

Hops

A hop happens when a packet is passed from one network segment to the next, usually through routers, from the source to the destination. The hop count refers to the number of intermediate devices through which data must pass between source and destination.

On a layer 3 network such as Internet Protocol (IP), each router along the data path constitutes a hop. By itself, this metric is, however, not useful for determining the optimum network path, as it does not take into consideration the speed, load, reliability, or latency of any particular hop, but merely the total count. Nevertheless, some routing protocols, such as Routing Information Protocol (RIP), use hop count as their sole metric.

Each time a router receives a packet, it modifies the packet, decrementing the time to live (TTL). The router discards any packets received with a zero TTL value. This prevents packets from endlessly bouncing around the network in the event of routing errors. Routers are capable of managing hop counts, but other types of network devices (e.g. Ethernet hubs and bridges) are not.

What are the differences between a hub, a switch and a router?

Hub

Hub is commonly used to connect segments of a LAN (Local Area Network). A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets. Hub acts as a common connection point for devices in a network.

A hub is the least expensive, least intelligent, and least complicated of the three. Its job is very simple: anything that comes in one port is sent out to the others. That's it.

Switch

A switch operates at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI (Open Systems Interconnection) Reference Model and therefore support any packet protocol. LANs that use switches to join segments are called switched LANs or, in the case of Ethernet networks, switched Ethernet LANs. In networks, the switch is the device that filters and forwards packets between LAN segments.

A switch is slightly smarter than a hub in that it learns where the sender of a message is, such that any subsequent messages destinated for that sender need only be sent to that single port.

Router

A router is connected to at least two networks, commonly two LANs or WANs (Wide Area Networks) or a LAN and its ISP's (Internet Service Provider's) network. The router is generally located at gateways, the places where two or more networks connect. Using headers and forwarding tables, router determines the best path to forward the packets. In addition, router uses protocols such as ICMP (Internet Control Message Protocol) to communicate with each other and configures the best route between any two hosts. In a word, router forwards data packets along with networks.

Consumer-grade routers perform (at minimum) two additional and important tasks: DHCP and NAT.

DHCP (Dynamic Host Configuration Protocol) is how dynamic IP addresses are assigned. When it first connects to the network, a device asks for an IP address to be assigned to it, and a DHCP server responds with an IP address assignment. A router connected to your ISP-provided internet connection will ask your ISP's server for an IP address; this will be your IP address on the internet. Your local computers, on the other hand, will ask the router for an IP address, and these addresses are local to your network.

NAT – Network Address Translation – is the way the router *translates* the IP addresses of packets that cross the internet/local network boundary. When computer "A" sends a packet, the IP address that it's "from" is that of computer "A" – 192.168.0.1, for example. When the router passes that on to the internet, it replaces the local IP address with the internet IP address assigned by the ISP – 1.2.3.4, for example. It also keeps track, so if there's a response the router knows to do the translation in reverse, replacing the internet IP address with the local IP address for machine "A", and then sending that response packet on to machine "A".

A side effect of NAT is that machines on the internet cannot initiate communications to local machines; they can only respond to communications initiated by them. This means that the router also acts as an effective firewall.

What can a router's web admin do?

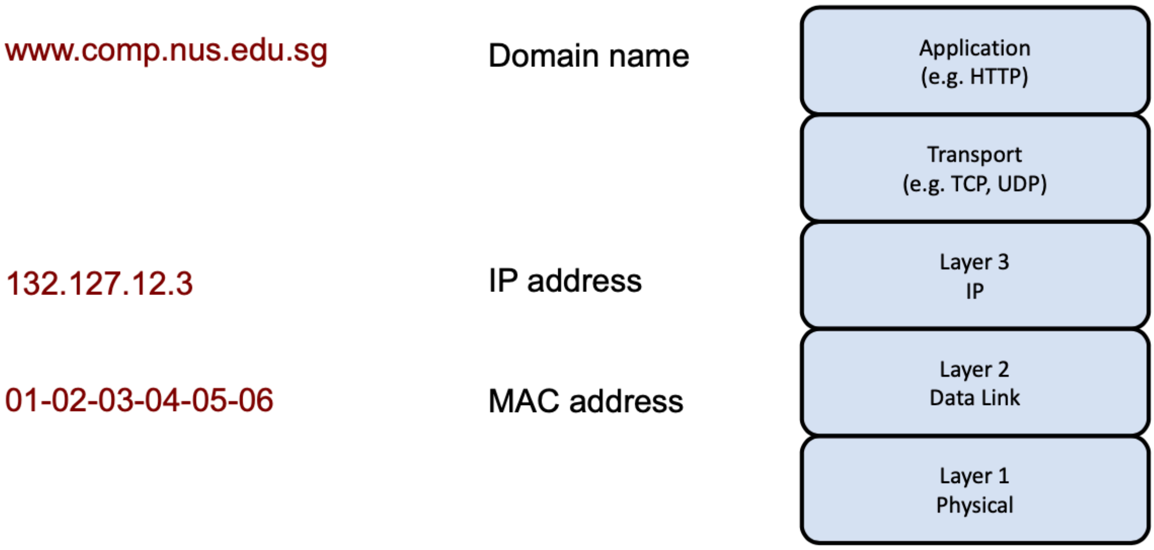
A router's web admin can do a lot of things, given that the admin is well-versed in this area. The admin can do the following:

1. Set another computer to intercept all traffic
2. If insecure email is used, email account details can be intercepted in clear whenever the email software checks the mail automatically
    a. Further incoming email can be blocked
    b. Individual messages can be deleted
3. With details in email, any password resets can be intercepted and used to gain access to secure sites
4. Cookies can be stolen if they are not uniformly secure, allowing access into an account without the password
5. See the end points of encrypted web traffic, though not the content itself
6. Change DNS settings to point to a domain under their control, as your computer would use the router as a DNS server when looking up what IP address a certain domain corresponds to.
7. If the router is part of an entire network of routers, compromising one router is enough to take over all of them, as they trust each other and there are routing protocols that can be subverted.

Network Attacks

Name Resolution and Attacks

Each peer entity (computer systems connected to each other via the internet) has a name. A single node may have different name at a different layer. For example:



When a peer entity uses the virtual connection in the layer below, it needs to find the corresponding name mapping. For example, finding the IP address of a domain name. Protocols that perform name mappings are known as resolution protocols.

Many initial design of resolution protocols did not take security into account, and thus it was easy for attackers to manipulate the outcome.

One such resolution protocol is the Domain Name System (DNS), which maps domain names to their respective IP addresses. It uses a hierarchical decentralized naming system. An attacker can thus target the association of the domain name with the IP address.

Another resolution protocol is the Address Resolution Protocol (ARP), which associates or maps IP addresses (logical addresses) with MAC addresses (physical addresses). It uses a broadcast mechanism on a local network. An attacker on the local network can target the association.

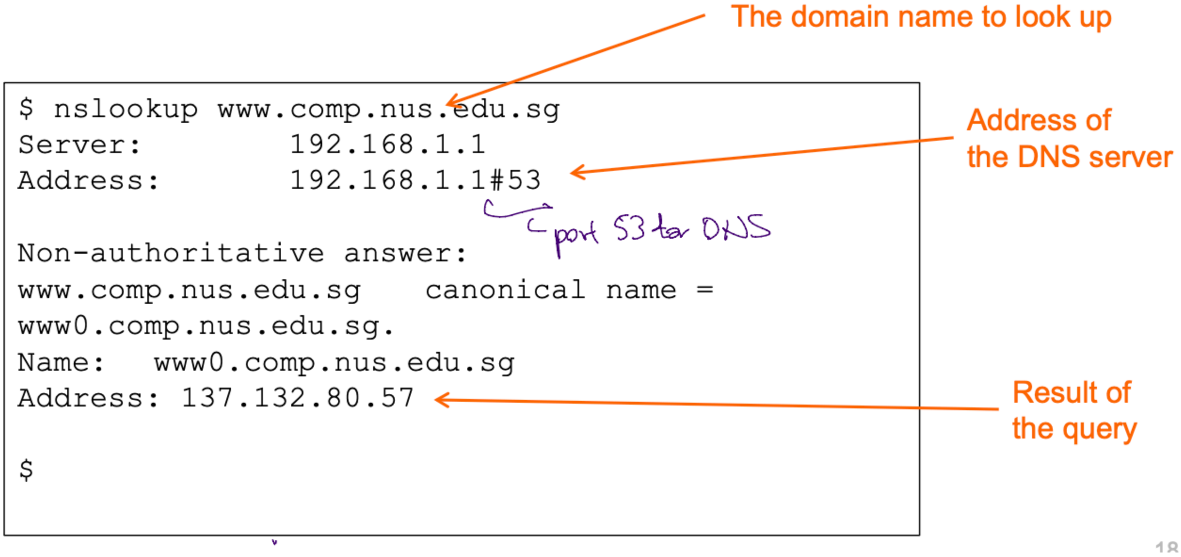
Domain Name System (DNS)

Given a domain name (e.g. www.comp.nus.edu.sg), its IP address can be found by either looking up a locally stored host table, or by querying a DNS server. The process is known as **name resolution**.

The entity (aka client) that initiates the query is called the resolver. If the address is found, we say that the domain name is resolved.

Each query contains a 16-bit number, known as Query ID (QID). The response from the name server must also contains a QID. If the QID in the response doesn't match the QID in the query, the client rejects the answer.

Note that no encryption or MAC is involved, as in the original design consideration, the QID is probably not meant for authentication, but as an efficient way to match multiple queries.



```
$ nslookup www.comp.nus.edu.sg
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.comp.nus.edu.sg    canonical name =
www0.comp.nus.edu.sg.
Name:   www0.comp.nus.edu.sg
Address: 137.132.80.57

$
```

The domain name to look up

Address of the DNS server

port 53 for DNS

Result of the query

18

Local DNS Attack

Let us consider the case of Alice, who is at a café using an unprotected WIFI connection to surf the web. She visits www.comp.nus.edu.sg, and types the domain name into the browser's address bar.

The browser makes a query to a DNS server to determine the IP address, then connects to the IP address obtained.

However, if there is an attacker at the physical layer, i.e. in the café accessing the same WIFI, the attacker can sniff data from the unprotected communication channel and spoof data into it as well. The attacker cannot modify or remove data already sent by Alice.

Should the attacker own a web server with a spoofed SoC website, the attacker can spoof a reply with the same QID as Alice's query with the attacker's own IP address as the message. Since the attacker is closer to Alice, the attacker's reply will reach Alice first, before the reply from the DNS server does. Alice takes the first reply as the answer and connects to the fake IP address.

More about DNS

- DNS operates at the **application layer.**
- Although the attacker is at the physical layer, for ease of analysis, we can assume that the attacker is just below the application layer. That is, there exists some virtual connection that can send the message across.
- Hence, the previous portion doesn't mention about the MAC and IP addresses, etc., of the DNS server.
- The DNS is an important component as it resolves the domain name. Hence, an DNS server can be the "single-point-of-failure" for the network.
- A DoS attacks, instead of attacking a Web server, could attack the DNS server instead.

Denial of Service Attacks

DoS attacks target availability, preventing some service from being accessible and usable upon demand by an authorised entity, delaying time-critical operations.

Types of DoS Attacks

	Stopping Service	Exhausting Resources
Local Attack (Easily detected and punished/mitigated)	Process killing Process crashing System reconfiguring	Spawning processes Filling up the file system
Remote Attack (over the Internet)	Sending malformed packet attacks Requires vulnerabilities, which are easily patched up	Packet flooding, as the system cannot tell if the request is valid or invalid

More on the above:

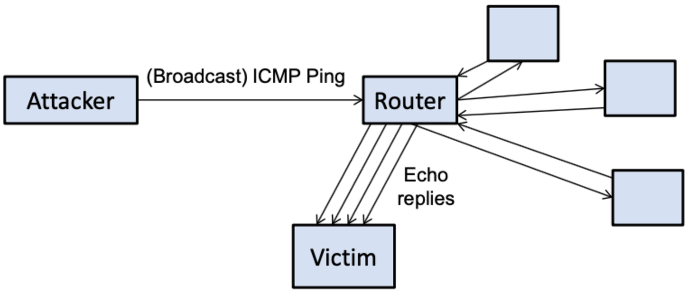
- Local Attacks
  - Can be more easily tracked than remote attacks
- Malformed Packet Attacks
  - Sending malformed packets remotely does not usually work on updated operating systems, since it requires vulnerabilities
- Packet Flooding Attacks
  - Many effective DOS attacks simply remotely flood the victims with an overwhelming amount of requests or data
  - The attacker can amplify small traffic to obtain a large amount of traffic, typically done by using available public servers (Internet infrastructure), such as DNS, NTP and CharGen.

DoS Example 1: ICMP/Smurf Flood Attack

Why "smurf"? This is because this attack can bring down big targets.

This is an attack that makes use of public servers to target a specific victim IP address. This is done via:

1. An attacker sends the **"ICMP PING"** request to a router, instructing the router to broadcast this request.
2. The request's source IP address is spoofed and replaced with the victim IP address.
3. The router thus broadcasts this request.
4. Every entity that receives this request will reply to it by sending an **"Echo reply"** to the source, which is the victim
5. The victim is overwhelmed with "Echo reply"s from the entire network.
6. The attacker thus takes advantage of the **amplification effect** to attack the victim.

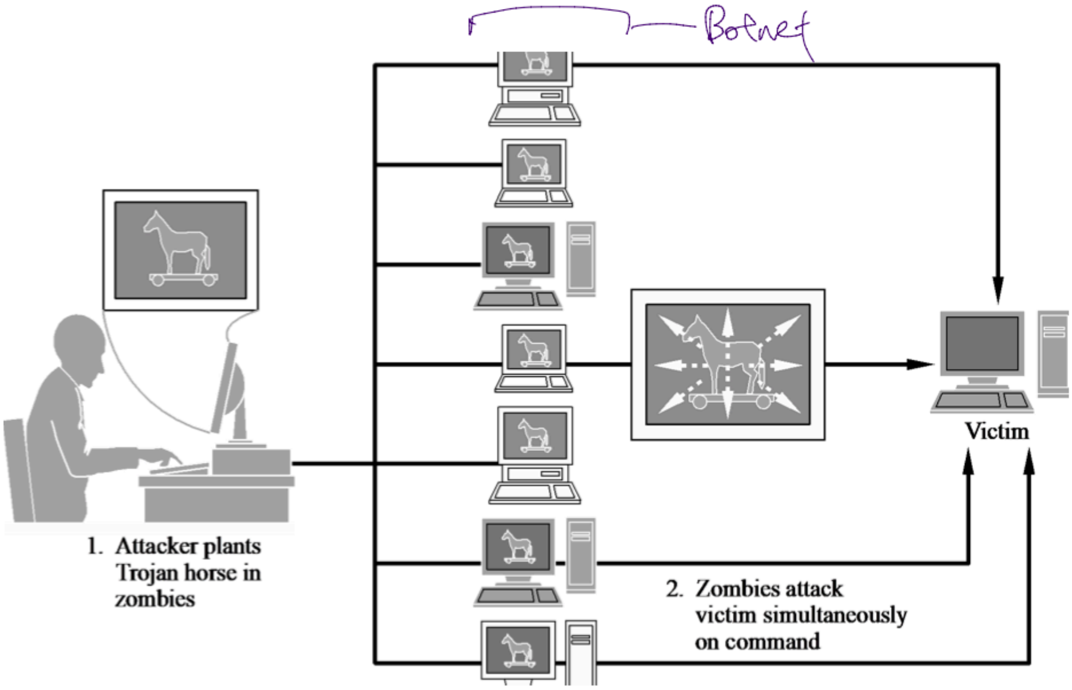


Is this attack still effective? Fortunately not, as most routers are now configured to not broadcast the requests. To prevent this attack, the measure is to simply disable a feature that was previously thought to be useful.

DoS Example 2: Application-Layer DoS Attack (HTTP Get)
The trick is to simply flood a web server with HTTP requests. For this attack to be effective, a large number of attackers are required, since each attacker can only send requests at a low rate.

When DoS is carried out by a large number of attackers, this is called **Distributed Denial of Service (DDoS).**

Distributed Denial of Service (DDoS)


Botnet
Victim
1. Attacker plants Trojan horse in zombies
2. Zombies attack victim simultaneously on command

As shown above, DDoS is normally achieved via a botnet.

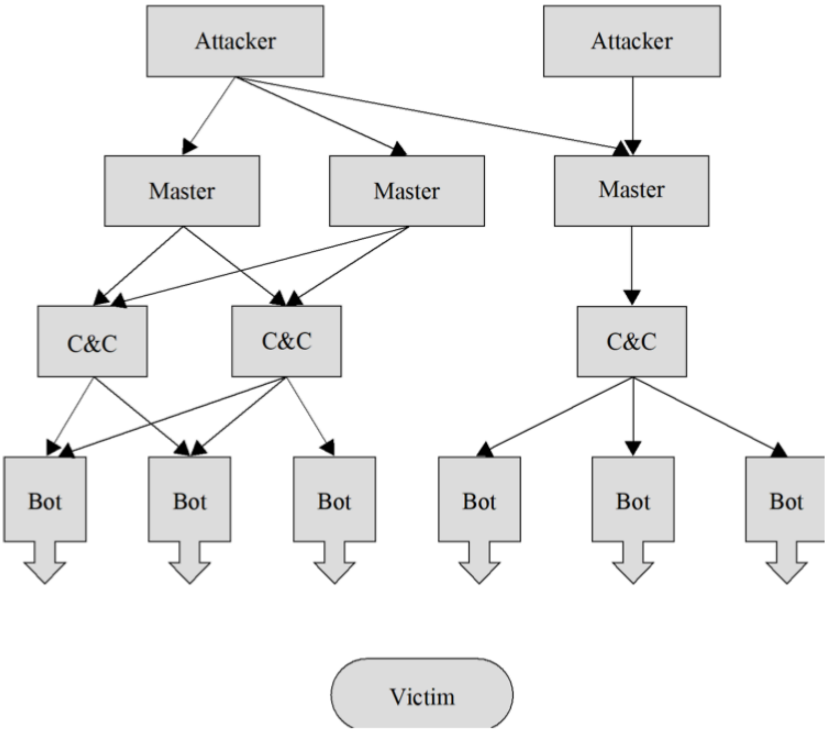
A bot, or zombie, is a compromised machine, and a botnet, or zombie army, is a large collection of connected bots, communicating via covert channels.

Why via covert channels? This is to prevent the owner of the zombie computer from noticing. Most owners of zombie computers are unaware that their system is being used in this way. Because the owner tends to be unaware, these computers are metaphorically compared to zombies.

A botnet has a **command-and-control** mechanism and can thus be controlled by a single individual to carry out a Distributed Denial of Service attack.

Some other possible usages of a botnet includes:

- Vulnerability Scanning
- Anonymising HTTP Proxy
- Email Address Harvesting
- Cipher Breaking

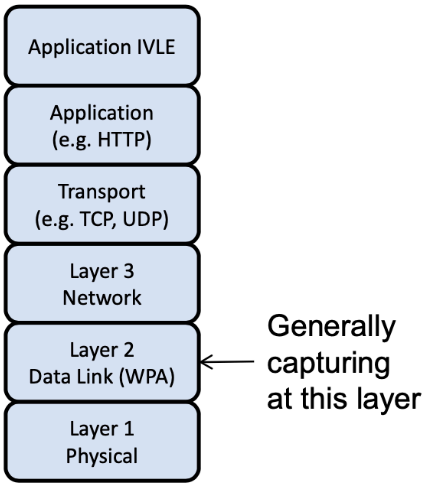


How a botnet may look like

Useful Tools

Wireshark (Packet Analyser)

Wireshark is a popular, free and open-source network packet analyser. It generally performs capturing at the **link layer**. This depends on the operating system and hardware, however. It essentially captures "interactions" between the operating system and the network card driver.



Some things that Wireshark can do:

- View list of packets
- View packet details
- View packet bytes
- Filter for packets
- Follow TCP stream

Nmap (Port Scanner)

What is a port? Previously, we saw that the port number was assigned at the **Transport Layer** of the TCP/IP model.

A port helps a server decide which application process to handle an incoming packet. By saying that a process or service is "listening" to a particular port, we mean that the process is running and ready to process packets with that particular port number. We also say a port is "open" when there exists such a process running in the server.

Well-known port numbers:

- 1: TCP Port Service Multiplexer
- 7: Echo Protocol
- 17: Quote of the Day
- 19: Character Generator Protocol (CHARGEN)
- 20: File Transfer Protocol (FTP) data transfer
- 21: File Transfer Protocol (FTP) control
- 22: Secure Shell (SSH), secure logins, file transfers (scp, sftp) and port forwarding
- **25: Simple Mail Transfer Protocol (SMTP), used for email routing between mail servers**
- 43: WHOIS Protocol
- 53: Domain Name System (DNS)
- **80: Hypertext Transfer Protocol (HTTP)**
- 220: Internet Message Access Protocol (IMAP), version 3
- **443: Hypertext Transfer Protocol over TLS/SSL (HTTPS)**
- 465: Authenticated SMTP over TLS/SSL (SMTPS)
- **515: Line Printer Daemon (LDP), print service**
- 666: Doom, first online first-person shooter

Port scanning is thus the process of determining which ports are open on hosts in a network. Ports are somewhat like the "doors" into each machine, hence port scanning is somewhat like knocking on the doors.

A port scanner is thus a tool to perform port scanning. It is useful for both attackers and network administrators to scan for vulnerabilities. Nmap is a very popular port scanner.

Nmap is a full featured port-scanning tool:

- Command-line tool with a GUI frontend
- Installation: sudo apt-get install nmap, zenmap

- Usage: nmap [ScanType(s)] [Options] {target specification}
- Examples:
  - TCP ACK scan (a stealthier scan): nmap -sA
  - OS fingerprinting: nmap -O
  - Service/version detection: nmap -sV

Network Protection

Cryptography

There are several cryptographic techniques that can help us achieve **confidentiality** (via encryption) and **authenticity** (via MAC, PKI, and Strong Authentication) over a public communication channel, even if the adversary can sniff and spoof the data.

There are various security protocols that essentially achieve that, but operates at different layers. Some prominent protocols are:

- TLS/SSL
- WIFI Protected Access II (WPA2)
- Internet Protocol Security (IPsec)

Issues faced when we discuss security protocols and attacks:

- Often, when we discuss a security protocol, we indicate the layer that the protocol aims to protect.
  - Complication: Some protections span across multiple layers, or do not provide full protection of the targeted layer.
- When analysing an attack, it is also insightful to figure out at which layer the attacker resides.
  - Complication: Likewise, some attacks span across multiple layers. In such situations, trying hard to pinpoint the layer could sometimes be very confusing.

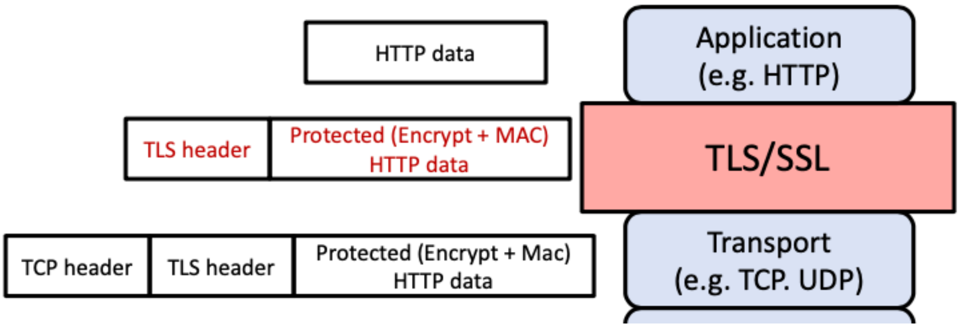
Thus, we have the following general guideline:

A security protocol that protects layer k would protect information from that layer and above against an attacker sitting at layer k-1 and below

For example, if an attacker resides at layer 1 and there is a security protocol that protects layer 3, what is protected by the security protocol is the information generated in layer 3 and above, but what is not protected is the information generated in layer 2.

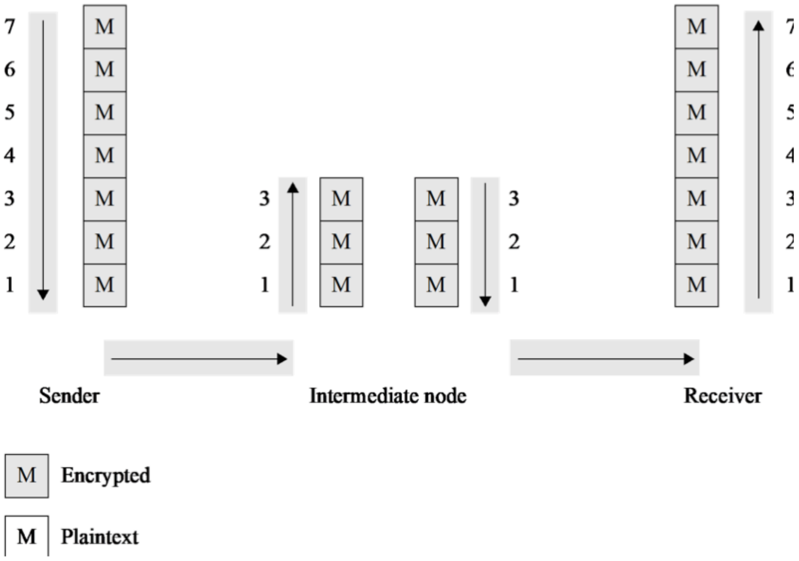
Secure Sockets Layer / Transport Layer Security (SSL/TLS)
The SSL/TLS sits **on top of the transport layer.** In other words, when an application, such as a browser or an email agent, wants to send data to the other end point, it first passes the data and the destination IP address to the SSL/TLS.

Next, SSL/TLS protects the data using encryption (for confidentiality) and MAC (for authenticity), then it instructs the transport layer to send the protected data. An end-to-end encryption is performed.

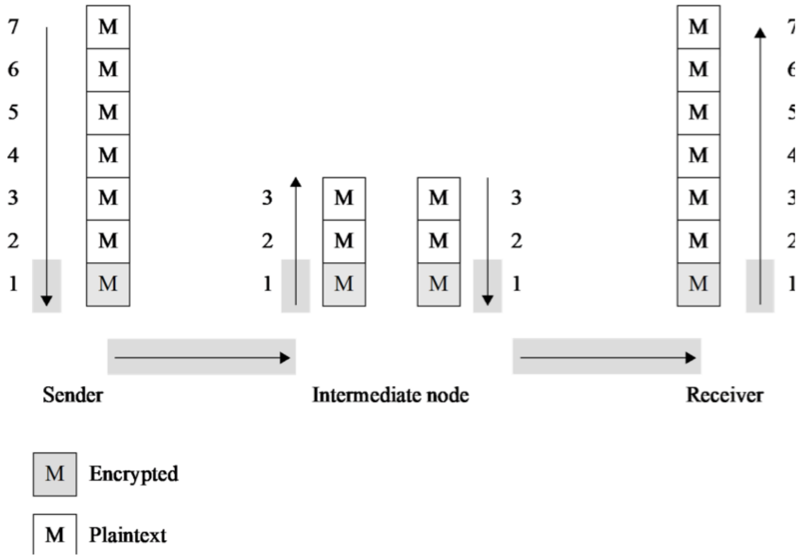

HTTP data
Application (e.g. HTTP)
TLS header
Protected (Encrypt + MAC) HTTP data
TLS/SSL
TCP header
TLS header
Protected (Encrypt + Mac) HTTP data
Transport (e.g. TCP. UDP)

The receiver end-point decrypts the received data **at the corresponding layer**.

What is end-to-end encryption?
End-to-end encryption is called as such because data stays encrypted from one end of its journey to the other. In this case, it is actually encrypted once it leaves the application layer, and remains encrypted until it reaches above the transport layer of the receiver side. Below is a diagram to illustrate the point.



This is in contrast with Link Encryption, of Hop-by-Hop Encryption. In Link Encryption, the encryption occurs at the Data Link and Physical layers, and the packet is decrypted at every device between the two ends. Below is a diagram to illustrate the point.



Examples
Let us see some examples of how SSL/TLS works.

1. Alice accesses the LumiNUS web application to upload her report, a.pdf, to the LumiNUS server.
    a. Note that LumiNUS uses HTTPS, which in turn employs SSL/TLS.
2. Alice's machine does the following:
    a. The "LumiNUS client" passes the file a.pdf to HTTPS, which in turns passes it to TLS
    b. TLS protects the data by encryption and MAC
    c. TLS passes the protected data to the transport layer
3. The LumiNUS server carries out the following:
    a. The transport layer passes the protected layer to TLS
    b. TLS decrypts the data and verify the MAC for integrity
    c. TLS passes the decrypted data to the LumiNUS application.

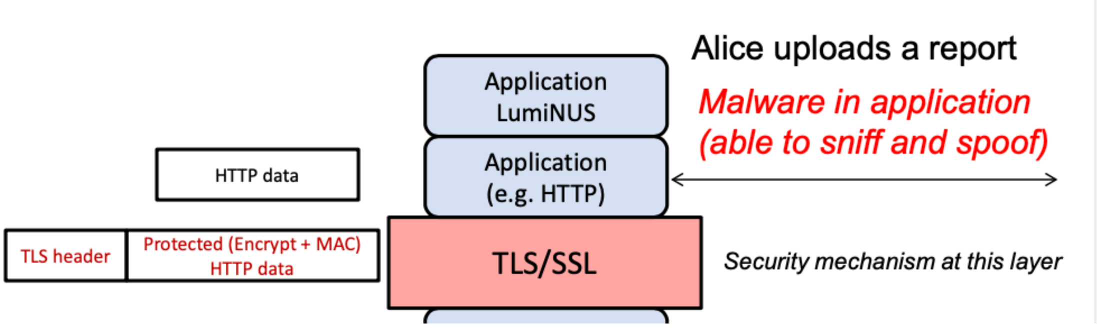
Note that in the process, "handshaking" occurs, where the two parties establish their session keys.

Attack Scenario 1: Attacker at the Physical Layer
Suppose there is an attacker at the physical layer who can sniff and spoof the message at that layer. Alice then uploads her report in that café using their free and open WIFI, that has no WPA protection. Hence, anyone in the café has access to the physical layer, and can sniff and spoof messages in that layer.

Can the attacker learn:
- Alice's uploaded report?
    - No, as it is protected by SSL/TLS, which protects the application layer, and information from that layer would be protected from an attacker below it, i.e. physical layer.
- The fact that Alice is visiting the LumiNUS website i.e. can the attacker lean the website's IP address?
    - Yes, as the information is contained in the IP headers from the network layer, which is not protected by TLS/SSL that sits on top of the transport layer from the attacker below it at the data link or physical layer.

Attack Scenario 2: Attacker at the Application Layer
Suppose that there is an adversary at the application layer. For example, a malicious JavaScript code is injected into LumiNUS and is being executed by Alice's computer. Can the malicious script learn:

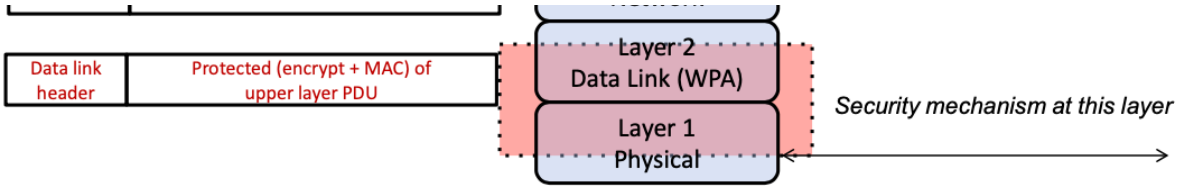
- Alice's report?
  - Yes, as the SSL/TLS does not protect information from its layer and above from attackers who are also on its layer or above. The malware in the application is still able to sniff and spoof.
- Alice's MAC address?
  - No, as the MAC address is defined at layer 2. An attacker at the application layer is unable to attack "downwards", and can only attack upwards.



WIFI Protected Access II (WPA2)

WPA2 is a popular protocol employed in home WIFI access points, and is more secure than Wired Equivalent Privacy (WEP), which is broken, and WPA.

WPA2 provides protection at layer 2 (Link) and layer 1 (Physical). However, not all information in layer 2 are protected.


Data link header
Protected (encrypt + MAC) of upper layer PDU
Layer 2
Data Link (WPA)
Layer 1
Physical
Security mechanism at this layer

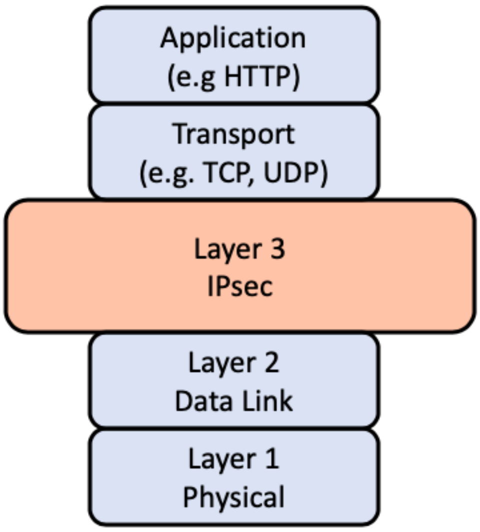
Attack Scenario: Attacker at the Physical Layer
Suppose there is an attacker at the physical layer who is able to sniff and spoof information, and Alice uploads a report. Can the attacker learn:

- Alice’s report?
  - No, as data from the upper layers have been encrypted with AES.
- The fact that Alice is visiting LumiNUS website?
  - No
- The MAC address (which is assigned in the link layer)?
  - The MAC address is never encrypted, as the MAC itself is required to enable the packet to reach the router and enable the router to send packets back.

Anyone within the range of the network might be able to see the traffic, but it will be scrambled with the most up-to-date encryption standards. WPA2 uses AES and keys that are 64 hexadecimal digits long.

WPA3 was announced in January 2018.

Internet Protocol Security (IPsec)



IPsec provides integrity and authenticity protection of IP addresses, but not confidentiality. Hence, attackers are unable to "spoof" the source IP addresses, but they can still learn the source and destination IP addresses of sniffed packets.

It is a mechanism whose goal is to protect the IP layer. The following is the detailed description:

IPsec is a is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing **mutual authentication between agents at the beginning of the session** and **negotiation of cryptographic keys to be used during the session**. IPsec can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway.

Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.

IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite, while some other Internet security systems in widespread use, such as Transport Layer Security (TLS) and Secure Shell (SSH), operate in the upper layers at Application layer. Hence, **only IPsec protects any application traffic over an IP network**. Applications can be automatically secured by IPsec at the IP layer.

Firewall

Having SSL/TLS and WPA2 is still insufficient when it comes to protecting the network. There are concerns with Denial of Service attacks, which they cannot prevent. SSL/TLS and WPA2 does not protect us when we interact with many services and applications, such as the DNS server. It is no practical, due to efficiency, to establish a SSL/TLS to the DNS server for a DNS query, hence we are susceptible to DNS spoofing.

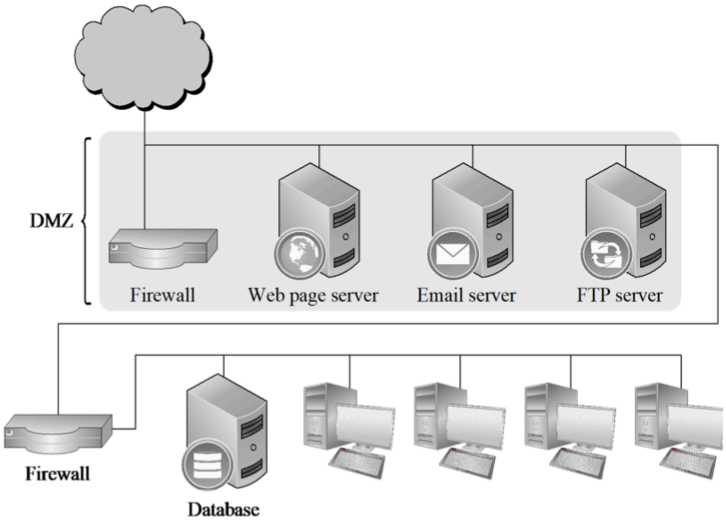
There is a need to control the flow of traffic between networks, especially between the untrusted public network (Internet) and the trusted internal network.

What is a Firewall?
A firewall is a device or program that controls the flow of network traffic between networks or hosts that employ **differing security postures.** It sits at the border between networks and looks at addresses, services and other characteristics of traffic. It then controls what traffic is allowed to enter the network (ingress filtering), or leave the network (egress filtering).

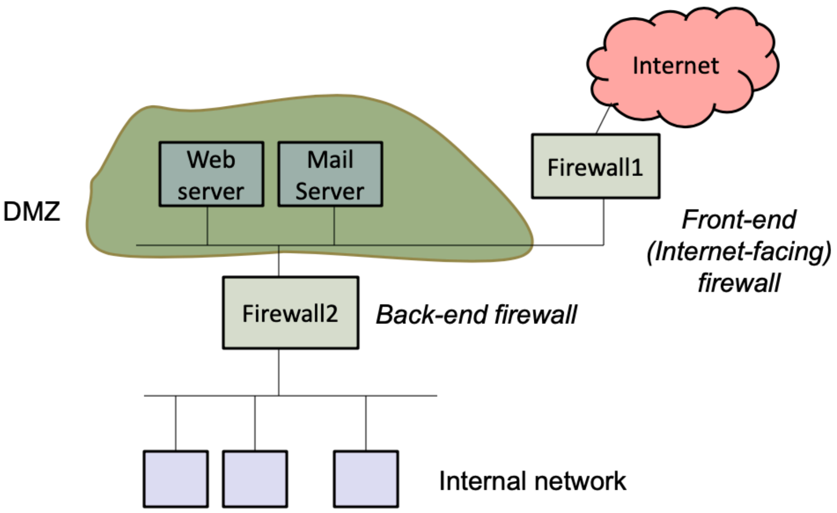
Demilitarized Zone (DMZ)
There is thus a concept of a DMZ – a small sub-network that exposes the organisation’s external service to the (untrusted) Internet.

The purpose of a DMZ is to add an additional layer of security to an organization's local area network (LAN): an external network node can access only what is exposed in the DMZ, while the rest of the organization's network is firewalled. The DMZ functions as a small, isolated network positioned between the Internet and the private network and, if its design is effective, allows the organization extra time to detect and address breaches before they would further penetrate into the internal networks.

In this case, the hosts most vulnerable to attack are those that provide services to users outside of the local area network, such as e-mail, Web and Domain Name System (DNS) servers. Because of the increased potential of these hosts suffering an attack, they are placed into this specific subnetwork in order to protect the rest of the network should any of them become compromised.



As seen above, this is a dual firewall architecture or 2-firewall setting. It is the most secure approach, according to Corlton Fralick – to use two firewalls to create a DMZ. The first firewall (also called the "front-end" or "perimeter" firewall) must be configured to allow traffic destined to the DMZ only. The second firewall (also called "back-end" or "internal" firewall) only allows traffic to the DMZ from the internal network.



This setup is considered more secure since two devices would need to be compromised. There is even more protection if the two firewalls are provided by two different vendors, because it makes it less likely that both devices suffer from the same security vulnerabilities.

Firewall Design
A firewall enforces a set of rules provided by the network administrator.

An example of rules for Firewall-2 (back-end firewall) would be:

- Block HTTP
- Allow from Internal Network to Mail Server: SMTP, POP3

An example of rules for Firewall-1 (front-end firewall) would be:

- Allow from anywhere to Mail Server: SMTP only

How the rules are to be specified thus differs based on the devices and software. Here is a sample firewall configuration.

Rule No	Protocol Type	Source Address	Destination Address	Designation Port	Action
1	TCP	*	192.168.1.*	25	Permit
2	TCP	*	192.168.1.*	69	Permit
3	TCP	192.168.1.*	*	80	Permit
4	TCP	*	192.168.1.18	80	Permit
5	TCP	*	192.168.1.*	*	Deny
6	UDP	*	192.168.1.*	*	Deny

*(any) matches any value

The table is processed in a top-down manner, and the first matching rule determines the action taken. Hence, the most specific rule is on top, and the most general rule is last.

Types of Firewall

There are 6 types of firewalls in the textbook, but they are usually grouped into 3 types:

1. **(Traditional) Packet Filters**

Filters packets based on information in packet headers.

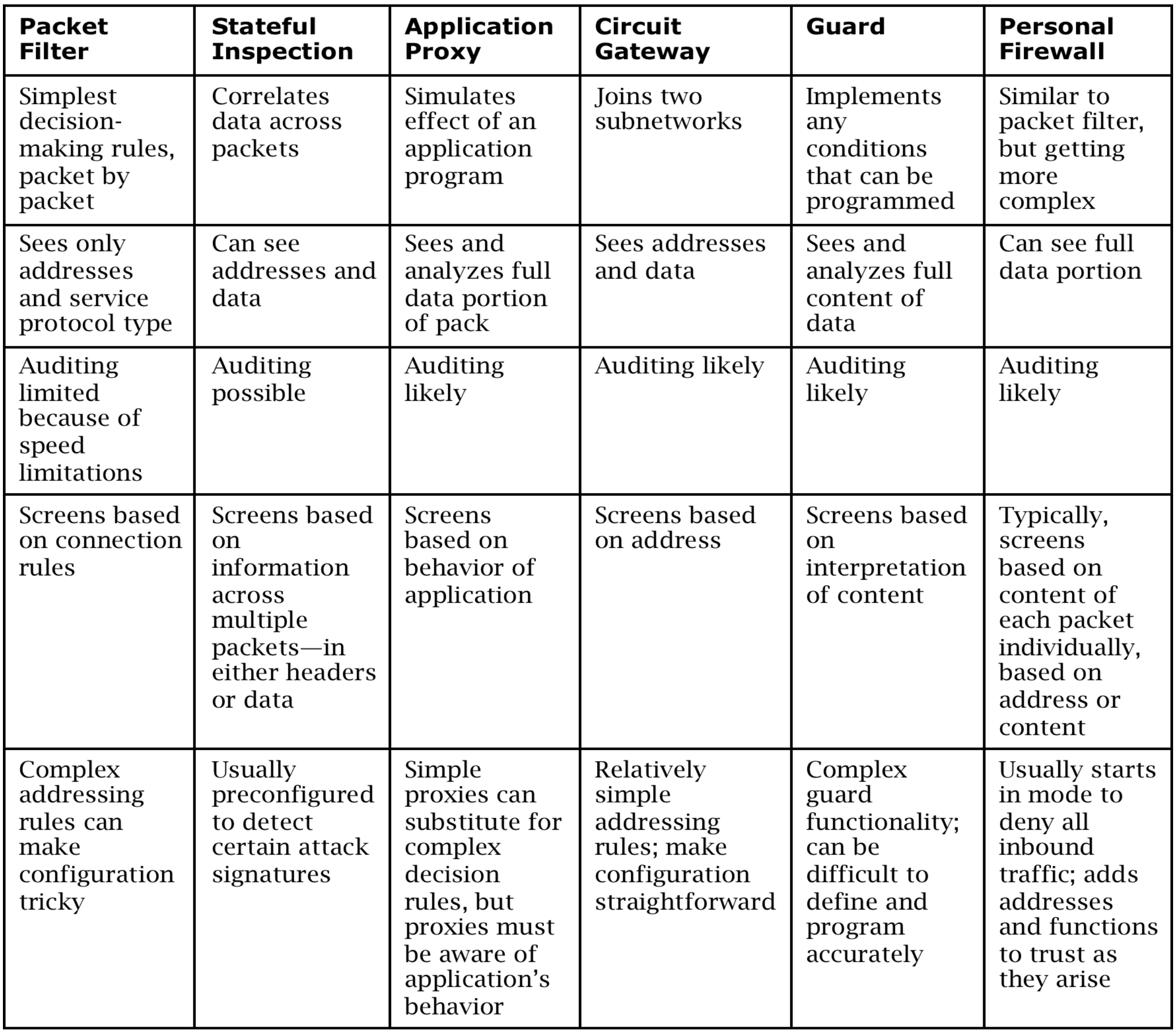
2. **Stateful-Inspection (Packet Filters):**

Maintains a state table of all active connections, and filters packets based on active connection states.

3. **Application Proxy**

Understands application logic and acts as a relay of application-level traffic.

Below are optional information on the various types of firewalls:



Packet Filter	Stateful Inspection	Application Proxy	Circuit Gateway	Guard	Personal Firewall
Simplest decision-making rules, packet by packet	Correlates data across packets	Simulates effect of an application program	Joins two subnetworks	Implements any conditions that can be programmed	Similar to packet filter, but getting more complex
Sees only addresses and service protocol type	Can see addresses and data	Sees and analyzes full data portion of pack	Sees addresses and data	Sees and analyzes full content of data	Can see full data portion
Auditing limited because of speed limitations	Auditing possible	Auditing likely	Auditing likely	Auditing likely	Auditing likely
Screens based on connection rules	Screens based on information across multiple packets—in either headers or data	Screens based on behavior of application	Screens based on address	Screens based on interpretation of content	Typically, screens based on content of each packet individually, based on address or content
Complex addressing rules can make configuration tricky	Usually preconfigured to detect certain attack signatures	Simple proxies can substitute for complex decision rules, but proxies must be aware of application's behavior	Relatively simple addressing rules; make configuration straightforward	Complex guard functionality; can be difficult to define and program accurately	Usually starts in mode to deny all inbound traffic; adds addresses and functions to trust as they arise

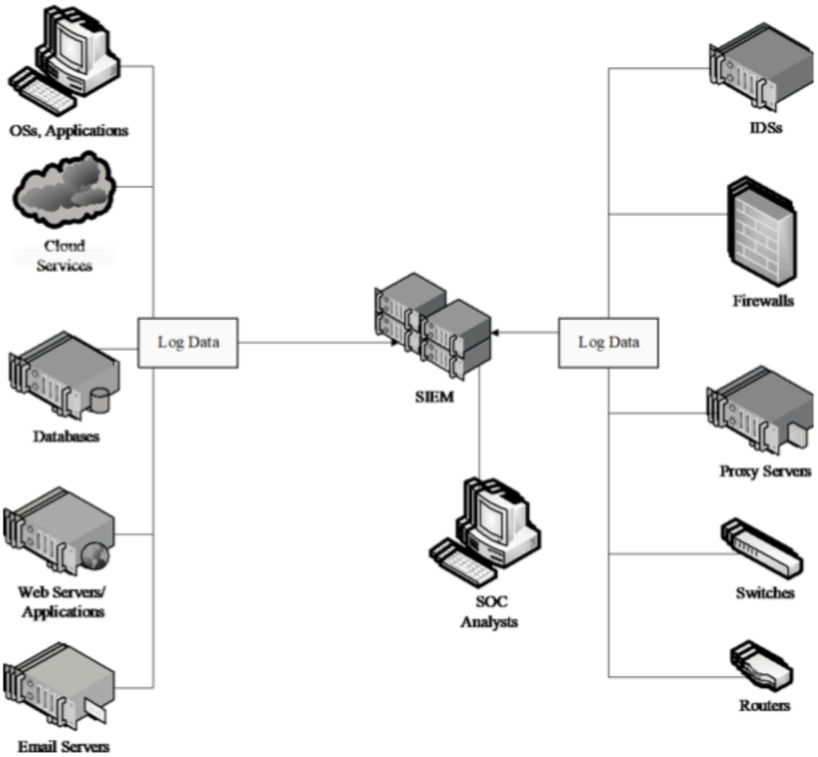
Network Security Management

There is a need to continuously monitor and adjust network characteristics. Hence, some best practices have been adopted to do so:

Security Operations Center (SOC)
A centralised unit in an organisation that monitors the IT systems and deals with security issues.

Security Information and Event Management (SIEM)
Pronounced as "SIM", SIEM is an approach that aims to provide real-time analysis of security alerts generated by network hardware and network applications. This may include the following capabilities:

- Data aggregation and correlation
- Event alerting
- Compliance report generation
- Forensic analysis



Firewall Design Case Study (from Tutorial 6)
Assume that we have two firewalls and the following machines:

1. Lab
    a. Total of 100 machines in labs for students to:
        i. prepare for reports
        ii. search for materials on the web
    b. There are also network printers in the labs
2. Teachers
    a. Every teacher has a PC in the teacher room
    b. They use the PCs to:
        i. enter students' grades
        ii. send/receive emails
        iii. prepare teaching materials
        iv. print exam questions
        v. perform web searches
    c. There are also network printers in the teacher rooms
3. Web-server
    a. School's web server
4. Email-server
    a. School's SMTP email server
5. SQL-server
    a. Stores the student database
    b. Some information can be accessed via a web-based application hosted in the Web-server
    c. For example, the app can allow students to update their mobile phone numbers
    d. Some other information can be accessed only by the teachers

These are the more precise requirements:

1. Prevent cases where exam questions get mistakenly printed in the Lab
2. Protect the SQL server
3. Block outbound packets that do not have legitimate source IP addresses, as some students may be running hacking tools that generate spoofed source IP addresses
4. We ignore the detailed issue of routing, i.e. we do not consider the internet gateway and Network Address Translation (NAT). For simplicity, we just assume that all machines use "public" IP addresses.

What we can do is the following:

Internal ← (IN) F_2 (OUT) → DMZ ← (IN) F_1 (OUT) → Internet

With the following setup:

- DMZ
    - Web-server
    - Email-server
    - Lab
        - We place it here since Lab PCs do not contain any important data and we want to segregate Lab and Teachers as required
    - Lab-printers
- Internal
    - Teachers
    - Teacher-printers
    - SQL-server

We then configure firewalls F_1 and F_2 as such:

Source IP	Dest IP	Source Port	Dest Port	Direction	Action
Web-server	*	HTTP	*	OUT	Allow
*	Web-server	*	HTTP	IN	Allow
Email-server	*	SMTP	*	OUT	Allow
*	Email-server	*	SMTP	IN	Allow
Lab	*	*	HTTP	OUT	Allow
*	Lab	HTTP	*	IN	Allow
Teachers	*	*	HTTP	OUT	Allow
*	Teachers	HTTP	*	IN	Allow
*	*	*	*	*	Block

Table 1: Firewall rules for the front-end / outer firewall F_1

Source IP	Dest IP	Source Port	Dest Port	Direction	Action
SQL-server	Web-server	SQL	*	OUT	Allow
Web-server	SQL-server	*	SQL	IN	Allow
Teachers	*	*	HTTP	OUT	Allow
*	Teachers	HTTP	*	IN	Allow
Teachers	Email-server	*	SMTP	OUT	Allow
Email-server	Teachers	SMTP	*	IN	Allow
*	*	*	*	*	Block

Table 2: Firewall rules for the back-end / inner firewall F_2

Note that requirement 3, which aims to block outgoing packets with illegitimate source IP addresses (egress filtering), is automatically met by the given rule sets. Any other firewall rules can be added as necessary, such as those needed to allow DNS and HTTPS traffic.

Type of Firewall: The firewalls above inspect only a few important fields of network packets. Thus they are packet filtering firewalls, which operate at the IP layer.