[Unit 5: Designing Internet Systems and Servers]
Internet Technology (CSC-402)

Jagdish Bhatta

Central Department of Computer Science & Information Technology
Tribhuvan University

Designing of Internet System Network Architecture:

The term ”network architecture” is commonly used to describe a set of abstract principles for the technical design of protocols and mechanisms for computer communication. Network architecture represents a set of deliberate choices out of many design alternatives, where these choices are informed by an understanding of the requirements. In turn, the architecture provides a guide for the many technical decisions required to standardize network protocols and algorithms. The purpose of the architecture is provide coherence and consistency to these decisions and to ensure that the requirements are met.

Network architecture is a set of high-level design principles that guides the technical design of the network, especially the engineering of its protocols and algorithms. To flesh out this simple definition, we have examples of the constituents of the architecture and how it is applied. A network architecture must typically specify:

- Where and how state is maintained and how it is removed.
- What entities are named
- How naming, addressing, and routing functions inter-relate and how they are performed.
- How communication functions are modularized, e.g., into “layers” to form a “protocol stack”.
- How network resources are divided between flows and how end-systems react to this division, i.e., fairness and congestion control.
- Where security boundaries are drawn and how they are enforced.
- How management boundaries are drawn and selectively pierced.
- How differing QoS is requested and achieved.

As an example, the following list is a brief summary of the requirements underlying the original Internet architecture. This list is ordered with the most important requirements first;

- **Internetworking:** existing networks must be interconnected.

- **Robustness:** Internet communication must continue despite loss of networks or routers.

- **Heterogeneity:** The Internet architecture must accommodate a variety of network

- **Distributed management:** The Internet architecture must permit distributed management of its resources

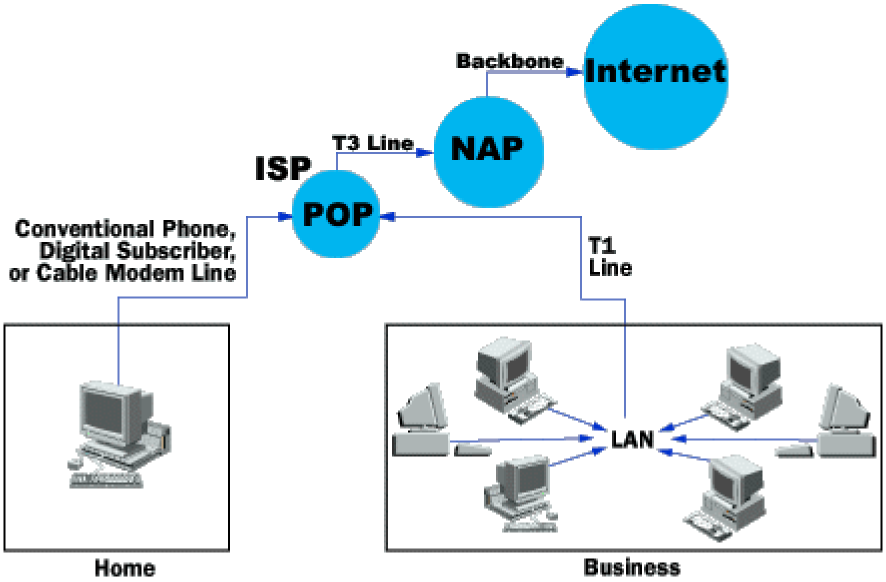
- **Cost:** The Internet architecture must be cost effective.

- **Ease of Attachment:** The Internet architecture must permit host attachment with a low level of effort.

- **Accountability:** The resources used in the internet architecture must be accountable.

Components of Internet Network Architecture:

Internet system architecture is defined as the arrangement of different types of parts of computer or the network hardware to configure or setup the internet technology is known as internet network architecture. Different types of devices or the hardware is required to setup up the internet network architecture. It can operate with the both networks such as wired or either wirelessly.



There are lots of components that are involved in maintaining the architecture of the internet technology. Some important parts that are used to configure the networking of the internet technology are as follows

Satellite: A major part of the internet network architecture is the satellite. Satellite plays a vital role in catching and distributing the signals over the network and the users use the internet network to search different types of information at any time.

Network Adapters: There are different types of network adapters that are used to configure or setup the internet technology on your operating system. First install the network adapters in the system then install its software for the sake of its proper working or compatibility. **Some common network adapters that are used for access of the internet are LAN cards or modems etc.**

Routers: As we know that this technology is also operates wirelessly so some components that are used to configure the internet network technology wireless router plays an important role and it is also the main part of the architecture. **It is defined as the device**

that is used to transmit data from one place to another in the form of packets that are called as data packets is known as router. These data packets are also called data gram.

Access Points: A special type of routing device that is used to transmit the data between wired and wireless networking device is called as AP. **It is often connected with the help of wired devices such as Ethernet.** It only transmits or transfers the data between wireless internet technology and wired internet network technology by using infra structure mode of network. One access point can only support a small group of networks and works more efficiently. It is operated less than hundred feet. It is denoted by AP.

Clients: Any kind of device such as personal computers, Note books, or any kind of mobile devices which are inter linked with wireless network area referred as a client of internet network architecture.

Bridges: A special type of connectors which is used to establish connections between wired network devices such as Ethernet and different wireless networks such as wireless LAN. It is called as bridge. It acts as a point of control in internet network architecture.

Building Blocks of Internet Architecture

Data Formatting: In the Internet, all types of digital information are encapsulated in packets with a standard format defined by the IP protocol. At a minimum, a host needs to be able to send and receive packets in that format in order to be connected to the Internet. It further includes issues like packet encapsulation, IP header formats, packet fragmentation and reassembly.

Addressing: In internet, defining addressing schemes is next important aspect. The process portion of the address definition, called port, has been standardized as part of both TCP and UDP header formats, and the network and host portions of the address definition have been combined into one 32-bit value, called IP address, which should be globally unique. Further includes defining DNS, DHCP, Subnetting , NAT.

Dynamic Routing: Routing in the context of the Internet is about maintaining consistent forwarding tables at the routers, in accordance with the network's store and forward communication paradigm. In the early days of the Internet, routing was not a major issue because of the small number of networks connected to it. Routing within a network was often done with a private protocol and routes between networks were static and manually set up. Later, as the size of the Internet grew, dynamic routing became necessary as the topology of the network constantly changed.

The forwarding is done by matching subnet prefixes. A typical forwarding table at a router, often referred to as the Forwarding Information Base (FIB) for the router, contains entries (i.e., routes) in the format of: <network prefix>, <next hop>, <metric>. <network prefix> is the subnet prefix of the destination network for this route, <next hop> the interface to use as part of this route to reach the destination, and <metric> a measure of goodness of this route. To forward a packet, the router first looks up its FIB with the packet's destination address and looks for subnet prefix matches. When the packet's destination address matches multiple routes in the FIB, the router chooses the route with the longest prefix match, i.e., with the most matching network bits. If there are more than one longest prefix matches, the router uses the <metric> field to break the tie. After determining a route, the router forwards the packet to the output port as defined by that route's <next hop>.

Resource Allocation: Resource allocation did not receive serious consideration in the original design of the Internet architecture because of the datagram service principle. However, as the reach of the Internet extends and the access speed increases, latency or loss sensitive applications such as video phone start to be deployed. These applications require the network to provide some minimum level of performance guarantee with respect to throughput, packet delay, packet loss rate, etc. A new catch phrase, quality of services (QoS), has since been coined by the networking community to refer to the level of performance guarantee a computer network provides. The various approaches to enforce this include traffic engineering, Differentiated services model, Integrated services model, Multiprotocol level switching.

Security: Today, with the Internet becoming an open infrastructure for ecommerce and E-government, security is one of the most pressing issues faced by the Internet community. Several security mechanisms such as firewall, virtual private network, transport layer security, secure email, and public key infrastructure (PKI), have been added to the Internet architecture with some level of successes.

Choice of Platforms

Software Platforms for servers:

Every website needs a reliable web server to be hosted on, so that it can be accessed via internet users. Today, in web hosting market there are many types of web servers available running on different platform to select.

There are at least three categories of server software platforms you need to consider:

- Choose a network computing operating system that fits the size, needs and resources of your business. A **networking operating system** (**NOS**), also referred to as the Dialoguer, is the software that runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions. The network operating system is designed to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks. The most popular network operating systems are Microsoft Windows Server 2003, Microsoft Windows Server 2008,UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

- Pick a file server platform that's reliable and secure to protect your company's data.

- Use web server platform software that can handle the amount of traffic you'll get and that has the functionality you want.

The most popular platforms and web servers are:

- **UNIX and Linux running Apache web server**
- **Window NT/2000 running Internet Information Server (IIS)**

How do you choose your web server platform?

If your website is purely made up of static web pages (i.e. HTML files), then any web hosting platform will work fine for you. However, if your website allows dynamic content, you will most likely need to run specific server-side functionality such as CGI scripts, JSP, ASP, SSI or PHP. In this case, UNIX platform web hosting will be ideal for your requirement.

On the other hand, if you need to use specific applications that require Windows to run such as ASP, .Net, MS Access, Microsoft SQL server or Cold Fusion, then you will need to find a web hosting providers that support Microsoft's Windows NT platform. Otherwise, all other server-side functionalities such as PHP, Perl and MySQL can be supported by UNIX platform.

While common programs such as Perl, PHP, Flash etc run on both UNIX and Window platform. Many other free open source software programs are available only for UNIX than for Windows. As a result, UNIX hosting is less expensive than Window hosting. So, if hosting cost is a big concern to you, then you should consider UNIX or Linux hosting.

Software Development Platforms/ Frameworks

- **IIS: Internet information service is** a web server application and set of feature extension modules created by Microsoft for use with Microsoft Windows

- **AMP Platforms**
    - **Lamp: LAMP** is an acronym for a solution stack of free, open source software, referring to the first letters of Linux (operating system), Apache HTTP Server, MySQL (database software) and originally Perl (but now sometimes PHP or Python), principal components to build a viable general purpose web server. The software combination has become popular because it is free of cost, open-source, and therefore easily adaptable, and because of the ubiquity of its components which are bundled with most current Linux distributions. When used together, they form a solution stack of technologies that support application servers.

- **Wamp: WAMP**s are packages of independently-created programs installed on computers that use a Microsoft Windows operating system. WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.
- **MAMP:** The acronym **MAMP** refers to a set of free software programs commonly used together to run dynamic web sites on servers running the Apple Macintosh operating system, Mac OS X:
    - **M**ac OS X, the operating system;
    - **A**pache, the Web server;
    - **M**ySQL, the database management system (or database server);
    - **P** for **P**HP, **P**erl, or **P**ython, all programming languages used for web development.

Any open source Web platform made up of these software programs and built upon Mac OS X is a MAMP.

- <u>Cross Platform</u>

- **XAMPP:**

- **Zend Server Community**

<u>Hardware Platform for servers:</u>

Hardware requirements for servers vary, depending on the server application. Absolute CPU speed is not usually as critical to a server as it is to a desktop machine . Servers' duties to provide service to many users over a network lead to different requirements such as fast network connections and high I/O throughput. Since servers are usually accessed over a network, they may run in headless mode without a monitor or input device. Processes that are not needed for the server's function are not used. Many servers do not have a graphical user interface (GUI) as it is unnecessary and consumes resources that could be allocated elsewhere. Similarly, audio and USB interfaces may be omitted.

To increase reliability, most of the servers use memory with error detection and correction, redundant disks, redundant power supplies and so on. Such components are also frequently hot swappable, allowing technicians to replace them on the running server without shutting it down.

Beside server computer the important hardware resources to establish a successful client/server model include; gateways, routers, network bridges, switches, hubs, and repeaters

Server Concepts: WEB, Proxy, RADIUS, MAIL:

Web Server:

Web server can refer to either the hardware (the computer) or the software (the computer application) that helps to deliver Web content that can be accessed through the Internet. The most common use of web servers is to host websites, but there are other uses such as gaming, data storage or running enterprise applications. The primary function of a web server is to deliver web pages on the request to clients using the Hypertext Transfer Protocol (HTTP). This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and scripts.

A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary memory, but this is not necessarily the case and depends on how the web server is implemented. While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to create HTML documents dynamically ("on-the-fly") as opposed to returning static documents. The former is primarily used for retrieving and/or modifying information from databases. The latter is typically much faster and more easily cached. Web servers are not always used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring and/or administering the device in question. This usually means that no additional software has to be installed on the client computer, since only a web browser is required (which now is included with most operating systems).

A web server (program) has defined load limits, because it can handle only a limited number of concurrent client connections (usually between 2 and 80,000, by default between 500 and 1,000) per IP address (and TCP port) and it can serve only a certain maximum number of requests per second depending on its own settings, the HTTP request type, whether the content is static or dynamic, whether the content is cached, and the hardware and software limitations of the OS of the computer on which the web server runs.

When a web server is near to or over its limits, it becomes unresponsive.

At any time web servers can be overloaded because of:

- Too much legitimate web traffic. Thousands or even millions of clients connecting to the web site in a short interval, e.g., Slashdot effect; the term "Slashdot effect" refers to phenomenon of a website becoming virtually unreachable because too many people are hitting it after the site was mentioned in an interesting article on the popular Slashdot news service.
- Distributed Denial of Service attacks. A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer or network resource unavailable to its intended users;
- Computer worms that sometimes cause abnormal traffic because of millions of infected computers (not coordinated among them);
- XSS viruses can cause high traffic because of millions of infected browsers and/or web servers;
- Internet bots. Traffic not filtered/limited on large web sites with very few resources (bandwidth, etc.);
- Internet (network) slowdowns, so that client requests are served more slowly and the number of connections increases so much that server limits are reached;
- Web servers (computers) partial unavailability. This can happen because of required or urgent maintenance or upgrade, hardware or software failures, back-end (e.g., database) failures, etc.; in these cases the remaining web servers get too much traffic and become overloaded.

Proxy Server:

In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server. The proxy server evaluates the request as a way to simplify and control their complexity. Today, most proxies are web proxies, facilitating access to content on the World Wide Web.

In an enterprise that uses the Internet, a proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service. A proxy server is associated with or part of a gateway server that separates the enterprise network from the outside network and a firewall server that protects the enterprise network from outside intrusion.

A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering requirements, the proxy server, assuming it is also a cache server, looks in its local cache of previously downloaded Web pages. If it finds the page, it returns it to the user without needing to forward the request to the Internet. If the page is not in the cache, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the page from the server out on the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user.

A proxy server has a variety of potential purposes, including:

- To keep machines behind it anonymous, mainly for security.
- To speed up access to resources (using caching). Web proxies are commonly used to cache web pages from a web server.
- To apply access policy to network services or content, e.g. to block undesired sites.
- To access sites prohibited or filtered by your ISP or institution.
- To log / audit usage, i.e. to provide company employee Internet usage reporting.
- To bypass security / parental controls.
- To circumvent Internet filtering to access content otherwise blocked by governments.
- To scan transmitted content for malware before delivery.
- To scan outbound content, e.g., for data loss prevention.
- To allow a web site to make web requests to externally hosted resources (e.g. images, music files, etc.) when cross-domain restrictions prohibit the web site from linking directly to the outside domains.

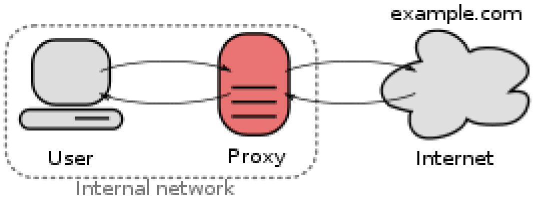
Proxy server can be placed in the user's local computer or at various points between the user and the destination servers on the Internet.

A proxy server that passes requests and responses unmodified is usually called a gateway or sometimes tunneling proxy.

A forward proxy is an Internet-facing proxy used to retrieve from a wide range of sources (in most cases anywhere on the Internet).

A reverse proxy is (usually) an Internet-facing proxy used as a front-end to control and protect access to a server on a private network, commonly also performing tasks such as load-balancing, authentication, decryption or caching.

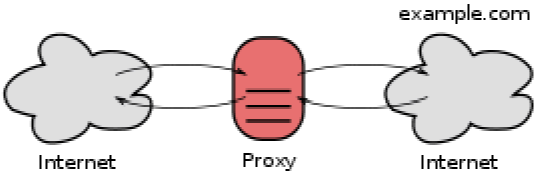
Forward proxies



A forward proxies are those taking requests from an internal network and forwarding them to the Internet. Forward proxies are proxies where the client server names the target server to connect to. Forward proxies are able to retrieve from a wide range of sources (in most cases anywhere on the Internet).

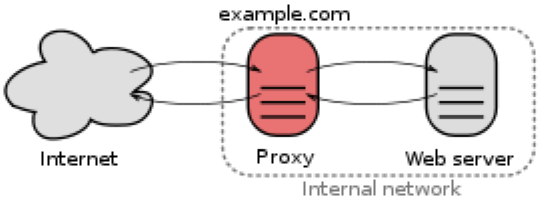
The terms "forward proxy" and "forwarding proxy" are a general description of behavior (forwarding traffic) and thus ambiguous. Except for Reverse proxy, the types of proxies described in this article are more specialized sub-types of the general forward proxy concept.

Open proxies



An open proxy is one forwarding requests from and to anywhere on the Internet. An open proxy is a forwarding proxy server that is accessible by any Internet user. An anonymous open proxy allows users to conceal their IP address while browsing the Web or using other Internet services. There are varying degrees of anonymity however, as well as a number of methods of 'tricking' the client into revealing itself regardless of the proxy being used.

Reverse proxies



A reverse proxy is one taking requests from the Internet and forwarding them to servers in an internal network. Those making requests connect to the proxy and may not be aware of the internal network. A reverse proxy (or surrogate) is a proxy server that appears to clients to be an ordinary server. Requests are forwarded to one or more origin servers which handle the request. The response is returned as if it came directly from the web server.

Reverse proxies are installed in the neighborhood of one or more web servers. All traffic coming from the Internet and with a destination of one of the neighborhood's web servers goes through the proxy server. The use of "reverse" originates in its counterpart "forward proxy" since the reverse proxy sits closer to the web server and serves only a restricted set of websites.

There are several reasons for installing reverse proxy servers:

- **Encryption / SSL acceleration:** when secure web sites are created, the SSL encryption is often not done by the web server itself, but by a reverse proxy that is equipped with SSL acceleration hardware. Furthermore, a host can provide a single "SSL proxy" to provide SSL encryption for an arbitrary number of hosts; removing the need for a separate SSL Server Certificate for each host, with the downside that all hosts behind the SSL proxy have to share a common DNS name or IP address for SSL connections.
- **Load balancing:** the reverse proxy can distribute the load to several web servers, each web server serving its own application area. In such a case, the reverse proxy

may need to rewrite the URLs in each web page (translation from externally known URLs to the internal locations).

- **Serve/cache static content:** A reverse proxy can offload the web servers by caching static content like pictures and other static graphical content.
- **Compression:** the proxy server can optimize and compress the content to speed up the load time.
- **Spoon feeding:** reduces resource usage caused by slow clients on the web servers by caching the content the web server sent and slowly "spoon feeding" it to the client. This especially benefits dynamically generated pages.
- **Security:** the proxy server is an additional layer of defense and can protect against some OS and WebServer specific attacks. However, it does not provide any protection to attacks against the web application or service itself, which is generally considered the larger threat.
- **Extranet Publishing:** a reverse proxy server facing the Internet can be used to communicate to a firewalled server internal to an organization, providing extranet access to some functions while keeping the servers behind the firewalls. If used in this way, security measures should be considered to protect the rest of your infrastructure in case this server is compromised, as its web application is exposed to attack from the Internet.

Use of Proxy servers:

Filtering:

A content-filtering web proxy server provides administrative control over the content that may be relayed in one or both directions through the proxy. It is commonly used in both commercial and non-commercial organizations (especially schools) to ensure that Internet usage conforms to acceptable use policy. In some cases users can circumvent the proxy, since there are services designed to proxy information from a filtered website through a non filtered site to allow it through the user's proxy

A content filtering proxy will often support user authentication, to control web access. It also usually produces logs, either to give detailed information about the URLs accessed by specific users, or to monitor bandwidth usage statistics

Some common methods used for content filtering include: URL or DNS blacklists, URL regex filtering, MIME filtering, or content keyword filtering. Some products have been known to employ content analysis techniques to look for traits commonly used by certain types of content providers.

Requests made to the open internet must first pass through an outbound proxy filter. The web-filtering company provides a database of URL patterns (regular expressions) with associated content attributes. This database is updated weekly by site-wide subscription, much like a virus filter subscription. The administrator instructs the web filter to ban broad classes of content (such as sports, pornography, online shopping, gambling, or social networking). Requests that match a banned URL pattern are rejected immediately.

Caching:

A caching proxy server accelerates service requests by retrieving content saved from a previous request made by the same client or even other clients. Caching proxies keep local copies of frequently requested resources, allowing large organizations to significantly reduce their upstream bandwidth usage and costs, while significantly increasing performance. Most ISPs and large businesses have a caching proxy. Caching proxies were the first kind of proxy server. Another important use of the proxy server is to reduce the hardware cost. An organization may have many systems on the same network or under control of a single server, prohibiting the possibility of an individual connection to the Internet for each system. In such a case, the individual systems can be connected to one proxy server, and the proxy server connected to the main server.

DNS Proxy:

A DNS proxy server takes DNS queries from a (usually local) network and forwards them to an Internet Domain Name Server. It may also cache DNS records.

Gateways to private networks:

Proxy servers can perform a role similar to a network switch in linking two networks.

Accessing services anonymously:

An anonymous proxy server, sometimes called a web proxy, generally attempts to anonymize web surfing. There are different varieties of anonymizers. The destination server (the server that ultimately satisfies the web request) receives requests from the anonymizing proxy server, and thus does not receive information about the end user's address. However, the requests are not anonymous to the anonymizing proxy server, and so a degree of trust is present between the proxy server and the user. Many of them are funded through a continued advertising link to the user.

RADIUS:

Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers to connect and use a network service.

Because of the broad support and the ubiquitous nature of the RADIUS protocol, it is often used by ISPs and enterprises to manage access to the Internet or internal networks, wireless networks, and integrated e-mail services. These networks may incorporate modems, DSL, access points, VPNs, network ports, web servers, etc

RADIUS serves three functions:

- **to authenticate users or devices before granting them access to a network,**

- **to authorize those users or devices for certain network services and**
- **to account for usage of those services.**

RADIUS servers use the AAA concept to manage network access in the following two-step process, also known as an "AAA transaction". AAA stands for “authentication, authorization and accounting”.

Authentication and authorization: The user or machine sends a request to a Remote Access Server (RAS) to gain access to a particular network resource using access credentials. The credentials are passed to the RAS device via the link-layer protocol - for example, Point-to-Point Protocol (PPP) in the case of many dialup or DSL providers or posted in an HTTPS secure web form. In turn, the RAS sends a RADIUS Access Request message to the RADIUS server, requesting authorization to grant access via the RADIUS protocol.

This request includes access credentials, typically in the form of username and password or security certificate provided by the user. Additionally, the request may contain other information which the RAS knows about the user, such as its network address or phone number, and information regarding the user's physical point of attachment to the RAS.
The RADIUS server checks that the information is correct using authentication schemes like PAP, CHAP or EAP. The user's proof of identification is verified, along with, optionally, other information related to the request, such as the user's network address or phone number, account status and specific network service access privileges. Historically, RADIUS servers checked the user's information against a locally stored flat file database. Modern RADIUS servers can do this, or can refer to external sources-commonly SQL, Kerberos, LDAP, or Active Directory servers—to verify the user's credentials.

The RADIUS server then returns one of three responses to the RAS ;

1. Access Reject
2. Access Challenge
3. Access Accept.

Access Reject - The user is unconditionally denied access to all requested network resources. Reasons may include failure to provide proof of identification or an unknown or inactive user account.

Access Challenge - Requests additional information from the user such as a secondary password, PIN, token or card. Access Challenge is also used in more complex authentication dialogs where a secure tunnel is established between the user machine and the Radius Server in a way that the access credentials are hidden from the RAS.

Access Accept - The user is granted access. Once the user is authenticated, the RADIUS server will often check that the user is authorized to use the network service requested. A given user may be allowed to use a company's wireless network, but not its VPN service, for example. Again, this information may be stored locally on the RADIUS server, or may be looked up in an external source like LDAP or Active Directory.

Each of these three RADIUS responses may include a Reply-Message attribute which may give a reason for the rejection, the prompt for the challenge, or a welcome message for the accept. The text in the attribute can be passed on to the user in a return web page.

Authorization attributes are conveyed to the RAS stipulating terms of access to be granted.

For example: the following authorization attributes may be included in an Access-Accept.

- The specific IP address to be assigned to the user
- The address pool from which the user's IP should be chosen
- The maximum length that the user may remain connected

Accounting: When network access is granted to the user by the NAS, an Accounting Start (a RADIUS Accounting Request packet containing an Acct-Status-Type attribute with the value "start") is sent by the NAS to the RADIUS server to signal the start of the user's network access. "Start" records typically contain the user's identification, network address, point of attachment and a unique session identifier.

Periodically, Interim Update records (a RADIUS Accounting Request packet containing an Acct-Status-Type attribute with the value "interim-update") may be sent by the NAS to the RADIUS server, to update it on the status of an active session. "Interim" records typically convey the current session duration and information on current data usage.

Finally, when the user's network access is closed, the NAS issues a final Accounting Stop record (a RADIUS Accounting Request packet containing an Acct-Status-Type attribute with the value "stop") to the RADIUS server, providing information on the final usage in terms of time, packets transferred, data transferred, reason for disconnect and other information related to the user's network access.

Typically, the client sends Accounting-Request packets until it receives an Accounting-Response acknowledgement, using some retry interval. The primary purpose of this data is that the user can be billed accordingly; the data is also commonly used for statistical purposes and for general network monitoring.

RADIUS is commonly used to facilitate roaming between ISPs, for example:

- by companies which provide a single global set of credentials that are usable on many public networks;
- by independent, but collaborating, institutions issuing their own credentials to their own users, that allow a visitor from one to another to be authenticated by their home institution,

Mail Server:

A mail server is a computer that serves as an electronic post office for email. Mail exchanged across networks is passed between mail servers that run specially designed software. This software is built around agreed-upon, standardized protocols for handling mail messages and the graphics they might contain.

Mail servers can be broken down into two main categories: outgoing mail servers and incoming mail servers. Outgoing mail servers are known as SMTP, or Simple Mail Transfer Protocol, servers. Incoming mail servers come in two main varieties. POP3, or Post Office Protocol, version 3, servers are best known for storing sent and received messages on PCs' local hard drives. IMAP, or Internet Message Access Protocol, servers always store copies of messages on servers. Most POP3 servers can store messages on servers, too, which is a lot more convenient.

There are several different components to the email system. They work together to move, deliver or retrieve your email.

Mail Transfer Agent (MTA): The MTA does a great deal of the hard work in moving mail around as it is responsible to move the mail from the local MTA to the destination MTA on the Internet. The Mail Transfer Agent works closely with DNS in making this all happen. The MTA uses a specific language SMTP to transfer mail on port 25, which is a standard. Several examples of MTAs are Sendmail, Postfix, and QMAIL.

Mail Delivery Agent (MDA): The MDA will receive the mail destined for the local network from the MTA and then will make this mail available for the user. The MDA will use POP3 on port 110 or IMAP on port 143 to make this available to users. Examples of MDA are Dovecot or Cyrus-IMAP.

Mail User Agent (MUA): The MUA is the client program that the end user uses to retrieve and view email. Users are able to view web based email with a browser but will user tools like Outlook, Thunderbird, Mutt, or Evolution to download mail to the local machine. When you send or read your email, the only part you see is the MUA, which is a fancy way of saying email client. But, there’s a lot more than that involved. To send an email, you’ll first sit down at your computer and fire up Thunderbird, or whichever other email client that you’re using. When you compose the message and click on the send button, the MUA will send it to the MTA. (The MTA could either be on the corporate network or at your ISP.) This MTA will send the message to successive MTA’s until it gets to the MTA that serves the email recipient. This MTA will then send the message to a Post Office Protocol (POP)/Internet Mail Application Protocol (IMAP) server. This server will store the email until the recipient accesses it with her MUA. Of course, this same process could take place within a corporate network instead of across the Internet.

The Process of Sending an Email:

Now that you know the basics about incoming and outgoing mail servers, it will be easier to understand the role that they play in the emailing process. The basic steps of this process are outlined below for your convenience.

Step #1: After composing a message and hitting send, your email client - whether it's Outlook Express or Gmail - connects to your domain's SMTP server. This server can be named many things; a standard example would be smtp.example.com.

Step #2: Your email client communicates with the SMTP server, giving it your email address, the recipient's email address, the message body and any attachments.

Step #3: The SMTP server processes the recipient's email address - especially its domain. If the domain name is the same as the sender's, the message is routed directly over to the domain's POP3 or IMAP server - no routing between servers is needed. If the domain is different, though, the SMTP server will have to communicate with the other domain's server.

Step #4: In order to find the recipient's server, the sender's SMTP server has to communicate with the DNS, or Domain Name Server. The DNS takes the recipient's email domain name and translates it into an IP address. The sender's SMTP server cannot route an email properly with a domain name alone; an IP address is a unique number that is assigned to every computer that is connected to the Internet. By knowing this information, an outgoing mail server can perform its work more efficiently.

Step #5: Now that the SMTP server has the recipient's IP address, it can connect to its SMTP server. This isn't usually done directly, though; instead, the message is routed along a series of unrelated SMTP servers until it arrives at its destination.

Step #6: The recipient's SMTP server scans the incoming message. If it recognizes the domain and the user name, it forwards the message along to the domain's POP3 or IMAP server. From there, it is placed in a sendmail queue until the recipient's email client allows it to be downloaded. At that point, the message can be read by the recipient.

Cookies:

A cookie, also known as an HTTP cookie, web cookie, or browser cookie, is usually a small piece of data sent from a website and stored in a user's web browser while a user is browsing a website. When the user browses the same website in the future, the data stored in the cookie can be retrieved by the website to notify the website of the user's previous activity. Cookies were designed to be a reliable mechanism for websites to remember the state of the website or activity the user had taken in the past. This can include clicking particular buttons, logging in, or a record of which pages were visited by the user even months or years ago.

Perhaps most importantly, authentication cookies are the most common method used by web servers to know whether the user is logged in or not, and which account they are logged in under. Without such a mechanism, the site would not know whether to send a page containing sensitive information, or require the user to authenticate himself by logging-in. The security of an authentication cookie generally depends on the security of the issuing website and the user's web browser. If not implemented correctly, a cookie's data can be intercepted by a hacker to gain unapproved access to the user's data and possibly to the originating website.

Cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site. Other than being set by a web server, cookies can also be set by a script in a language such as JavaScript, if supported and enabled by the Web browser.

The cookies consist of several values;

1. **The name and value** are encoded into the cookie and represent the state. The interpretation is that the name has an associated value.
2. **The expires field** indicates when the cookie is valid. Expired cookies are discarded; they are not to be given out. If this field is not present, the cookie will be deleted at the end of the session.
3. **The domain** states the domain for which the cookie is intended. It consists of the last n fields of the domain name of a server. For example, domain=.adv.com specifies that the cookie is to be sent to any requesting server in the adv.com domain. A domain field must have at least one embedded "." in it. There is no requirement that a cookie be sent from a host in the domain. This can be used to track certain types of accesses, as discussed below.
4. **The path** further restricts the dissemination of the cookie. When a Web server requests a cookie, it provides a domain. Cookies that match that domain may be sent to the server. If the server specifies a path, the path must be the leading substring of the path specified in the cookie.
5. If **the secure field** is set, the cookie will be sent only over secured connections (that is, to "https" or "http" over SSL).

Cookies can contain authentication information, both user-related and host-related. Using cookies for authentication treats them as tokens supplied by the browser to validate (or state and validate) an identity. Depending on the sensitivity of the interactions with the server, protecting the confidentiality of these cookies may be critical.

Types of cookie:

Session cookie: A user's session cookie for a website exists only while the user is reading and navigating the website. When an expiry date or validity interval is not set at cookie creation time, a session cookie is created. Web browsers normally delete session cookies when the user exits the browser.

Persistent cookie: A persistent cookie will outlast user sessions. If a persistent cookie has its Max-Age set to 1 year, then, within the year, the initial value set in that cookie would be sent back to the server every time the user visited the server. This could be used to record a vital piece of information such as how the user initially came to this website. For this reason persistent cookies are also called tracking cookies.

Secure cookie: A secure cookie has the secure attribute enabled and is only used via HTTPS, ensuring that the cookie is always encrypted when transmitting from client to server. This makes the cookie less likely to be exposed to cookie theft via eavesdropping.

HttpOnly cookie: The HttpOnly cookie is supported by most modern browsers. On a supported browser, an HttpOnly session cookie will be used only when transmitting HTTP (or HTTPS) requests, thus restricting access from other, non-HTTP APIs (such as JavaScript). This restriction mitigates but does not eliminate the threat of session cookie theft via cross-site scripting (XSS). This feature applies only to session-management cookies, and not other browser cookies.

Third-party cookie: First-party cookies are cookies set with the same domain (or its subdomain) in your browser's address bar. Third-party cookies are cookies being set with different domains from the one shown on the address bar (i.e. the web pages on that domain may feature content from a third-party domain - e.g. an advertisement run by www.advexample.com showing advert banners). (Privacy setting options in most modern browsers allow you to block third-party tracking cookies).

For example: Suppose a user visits www.example1.com, which sets a cookie with the domain ad.foxytracking.com. When the user later visits www.example2.com, another cookie is set with the domain ad.foxytracking.com. Eventually, both of these cookies will be sent to the advertiser when loading their ads or visiting their website. The advertiser can then use these cookies to build up a browsing history of the user across all the websites this advertiser has footprints on.

Supercookie: A "supercookie" is a cookie with a public suffix domain, like .com, .co.uk or k12.ca.us. Most browsers, by default, allow first-party cookies—a cookie with domain to be the same or sub-domain of the requesting host. For example, a user visiting www.example.com can have a cookie set with domain www.example.com or .example.com, but not .com. A supercookie with domain .com would be blocked by browsers; otherwise, a malicious website, like attacker.com, could set a supercookie with domain .com and potentially disrupt or impersonate legitimate user requests to example.com. The Public Suffix List is a cross-vendor initiative to provide an accurate list of domain name suffixes changing. Older versions of browsers may not have the most up-to-date list, and will therefore be vulnerable to certain supercookies.

The term "supercookies" is sometimes used for tracking technologies that do not rely on HTTP cookies. Two such "supercookie" mechanisms were found on Microsoft websites: cookie syncing that respawned MUID cookies, and ETag cookies

Zombie cookie: A zombie cookie is any cookie that is automatically recreated after a user has deleted it. This is accomplished by a script storing the content of the cookie in some other locations, such as the local storage available to Flash content, HTML5 storages and other client side mechanisms, and then recreating the cookie from backup stores when the cookie's absence is detected.

Uses of Cookies:

Session Management:

Cookies may be used to maintain data related to the user during navigation, possibly across multiple visits. Cookies were introduced to provide a way to implement a "shopping cart" (or "shopping basket"), a virtual device into which users can store items they want to purchase as they navigate throughout the site.

Shopping basket applications today usually store the list of basket contents in a database on the server side, rather than storing basket items in the cookie itself. A web server typically sends a cookie containing a unique session identifier. The web browser will send back that session identifier with each subsequent request and shopping basket items are stored associated with a unique session identifier.

Allowing users to log in to a website is a frequent use of cookies. Typically the web server will first send a cookie containing a unique session identifier. Users then submit their credentials and the web application authenticates the session and allows the user access to services.

Personalization:

Cookies may be used to remember the information about the user who has visited a website in order to show relevant content in the future. For example a web server may send a cookie containing the username last used to log in to a website so that it may be filled in for future visits.

Many websites use cookies for personalization based on users' preferences. Users select their preferences by entering them in a web form and submitting the form to the server. The server encodes the preferences in a cookie and sends the cookie back to the browser. This way, every time the user accesses a page, the server is also sent the cookie where the preferences are stored, and can personalize the page according to the user preferences. For example, the Wikipedia website allows authenticated users to choose the webpageskin they like best; the Google search engine once allowed users (even non-registered ones) to decide how many search results per page they want to see.

Tracking:
Tracking cookies may be used to track internet users' web browsing. This can also be done in part by using the IP address of the computer requesting the page or the referrer field of the HTTP request header, but cookies allow for greater precision.

This can be demonstrated as follows:
If the user requests a page of the site, but the request contains no cookie, the server presumes that this is the first page visited by the user; the server creates a random string and sends it as a cookie back to the browser together with the requested page;

From this point on, the cookie will be automatically sent by the browser to the server every time a new page from the site is requested; the server sends the page as usual, but also stores the URL of the requested page, the date/time of the request, and the cookie in a log file.

By analyzing the log file collected in the process, it is then possible to find out which pages the user has visited, and in what sequence.

Load Balancing: Proxy Arrays

Most commonly, the term load balancing refers to distributing incoming HTTP requests across Web servers in a server farm, to avoid overloading any one server. Because load balancing distributes the requests based on the actual load at each server, it is excellent for ensuring availability and defending against denial of service attacks.

Load balancing is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The load balancing service is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server.

Why is load balancing of servers needed?

Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server. Busy Web sites typically employ two or more Web servers in a load balancing scheme. If one server starts to get swamped, requests are forwarded to another server with more capacity. Load balancing can also refer to the communications channels themselves.

If there is only one web server responding to all the incoming HTTP requests for your website, the capacity of the web server may not be able to handle high volumes of

incoming traffic once the website becomes popular. The website's pages will load slowly as some of the users will have to wait until the web server is free to process their requests. The increase in traffic and connections to your website can lead to a point where upgrading the server hardware will no longer be cost effective.

In order to achieve web server scalability, more servers need to be added to distribute the load among the group of servers, which is also known as a *server cluster*. The load distribution among these servers is known as load balancing. Load balancing applies to all types of servers (application server, database server), however, we will be devoting this section for load balancing of web servers (HTTP server) only.

Application

One of the most common applications of load balancing is to provide a single Internet service from multiple servers, sometimes known as a server farm. Commonly, load-balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol (NNTP) servers and Domain Name System (DNS) servers. Lately, some load balancers evolved to support databases; these are called database load balancers.

For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting backend servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.

How?

When multiple web servers are present in a server group, the HTTP traffic needs to be evenly distributed among the servers. In the process, these servers must appear as one web server to the web client, for example an internet browser. The load balancing mechanism used for spreading HTTP requests is known **as *IP Spraying*.** The equipment used for IP spraying is also called the 'load dispatcher' or 'network dispatcher' or simply, the 'load balancer'. In this case, the IP sprayer intercepts each HTTP request, and redirects them to a server in the server cluster. Depending on the type of sprayer involved, the architecture can provide scalability, load balancing and fail-over requirements.

Proxy Server Arrays are way to handle loads in internet Proxy Server provides a feature called proxy arrays**.** An array is a peer-to-peer configuration of proxy servers instead of a hierarchy. **A proxy array is a solution whereby one or multiple proxy servers operate as a single cache for client requests**. Each Proxy Server that belongs to the proxy array performs the following functions:

- Maintains membership information for the proxy array. This information provides input on which array members are available and which array members are unavailable. A client therefore has to only query one array member because each member of the proxy array maintains information on all other array members.

- Uses a hash algorithm to perform routing decisions. The hash algorithm uses the following factors:
  List of current available servers.
  URL of client request.
  Load factor.

Arrays have the following benefits:

- Users and application sessions are *load-balanced* across the array. To scale to more users, simply add more SGD servers to the array. See Load Balancing for more details.
- With more than one server, there is no single point of failure. You can decommission a server temporarily with the minimum of disruption to your users.
- Configuration information, including all the objects in your organizational hierarchy, is replicated to all array members. All array members have access to all information.

An array contains the following:

- **One primary server.** This server is the authoritative source for global SGD information, and maintains the definitive copy of the organizational hierarchy, called the local repository.
- **One or more secondary servers.** The primary server replicates information to these servers.

Load Balancing Approaches

Load balancing of servers by an IP sprayer can be implemented in different ways. These methods of load balancing can be set up in the load balancer based on available load balancing types. There are various algorithms used to distribute the load among the available servers.

Random Allocation

In a random allocation, the HTTP requests are assigned to any server picked randomly among the group of servers. In such a case, one of the servers may be assigned many more requests to process, while the other servers are sitting idle. However, on average, each server gets its share of the load due to the random selection.

***Pros*:** Simple to implement.

***Cons*:** Can lead to overloading of one server while under-utilization of others.

Round-Robin Allocation

In a round-robin algorithm, the IP sprayer assigns the requests to a list of the servers on a rotating basis. The first request is allocated to a server picked randomly from the group, so that if more than one IP sprayer is involved, not all the first requests go to the same server. For the subsequent requests, the IP sprayer follows the circular order to redirect the request. Once a server is assigned a request, the server is moved to the end of the list. This keeps the servers equally assigned.

***Pros*:** Better than random allocation because the requests are equally divided among the available servers in an orderly fashion.

***Cons*:** Round robin algorithm is not enough for load balancing based on processing overhead required and if the server specifications are not identical to each other in the server group.

Weighted Round-Robin Allocation

Weighted Round-Robin is an advanced version of the round-robin that eliminates the deficiencies of the plain round robin algorithm. In case of a weighted round-robin, one can assign a weight to each server in the group so that if one server is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the IP sprayer will assign two requests to the powerful server for each request assigned to the weaker one.

***Pros*:** Takes care of the capacity of the servers in the group.

***Cons*:** Does not consider the advanced load balancing requirements such as processing times for each individual request.

Dynamic Round Robin (Called **Dynamic Ratio** on the BIG-IP)

It is similar to Weighted Round Robin, however, weights are based on continuous monitoring of the servers and are therefore continually changing. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. This *Application Delivery Controller* method is *rarely* available in a simple load balancer.

The configuration of a load balancing software or hardware should be decided on the particular requirement. For example, if the website wants to load balance servers for static HTML pages or light database driven dynamic webpages, round robin will be sufficient. However, if some of the requests take longer than the others to process, then advanced load balancing algorithms are used. The load balancer should be able to provide intelligent monitoring to distribute the load, directing them to the servers that are capable of handling them better than the others in the cluster of server.

Server Setup and Configuration Guidelines: *Needs to be finalized*

Security and System Administration Issues:

Security issues:

An organization's servers provide a wide variety of services to internal and external users, and many servers also store or process sensitive information for the organization. Some of the most common types of servers are Web, email, database, infrastructure management, and file servers.

A *server* is a host that provides one or more services for other hosts over a network as a primary function. For example, a file server provides file sharing services so that users can access, modify, store, and delete files. Another example is a database server that provides database services for Web applications on Web servers. The Web servers, in turn, provide Web content services to users' Web browsers. There are many other types of servers, such as application, authentication, directory services, email, infrastructure management, logging, name/address resolution services (e.g., Domain Name Server [DNS]), print, and remote access

Servers are frequently targeted by attackers because of the value of their data and services. For example, a server might contain personally identifiable information that could be used to perform identity theft. The following are examples of common security threats to servers:

- Malicious entities may exploit software bugs in the server or its underlying operating system to gain unauthorized access to the server.

- Denial of service (DoS) attacks may be directed to the server or its supporting network infrastructure, denying or hindering valid users from making use of its services.

- Sensitive information on the server may be read by unauthorized individuals or changed in an unauthorized manner.

- Sensitive information transmitted unencrypted or weakly encrypted between the server and the client may be intercepted.

- Malicious entities may gain unauthorized access to resources elsewhere in the organization's network via a successful attack on the server.

- Malicious entities may attack other entities after compromising a server. These attacks can be launched directly (e.g., from the compromised host against an external server) or indirectly (e.g., placing malicious content on the compromised server that attempts to exploit vulnerabilities in the clients of users accessing the server).

The classic model for information security defines three objectives of security: maintaining confidentiality, integrity, and availability. *Confidentiality* refers to protecting information from being accessed by unauthorized parties. *Integrity* refers to ensuring the authenticity of information—that information is not altered, and that the source of the information is genuine. *Availability* means that information is accessible by authorized users. Each objective addresses a different aspect of providing protection for information.

When addressing server security issues, it is an excellent idea to keep in mind the following general information security principles:

- **Simplicity**—Security mechanisms (and information systems in general) should be as simple as possible. Complexity is at the root of many security issues.

- **Fail-Safe**—If a failure occurs, the system should fail in a secure manner, i.e., security controls and settings remain in effect and are enforced. It is usually better to lose functionality rather than security.

- **Complete Mediation**—Rather than providing direct access to information, mediators that enforce access policy should be employed. Common examples of mediators include file system permissions, proxies, firewalls, and mail gateways.

- **Open Design**—System security should not depend on the secrecy of the implementation or its components.

- **Separation of Privilege**—Functions, to the degree possible, should be separate and provide as much granularity as possible. The concept can apply to both systems and operators and users. In the case of systems, functions such as read, edit, write, and execute should be separate. In the case of system operators and users, roles should be as separate as possible. For example, if resources allow, the role of system administrator should be separate from that of the database administrator.

- **Least Privilege**—This principle dictates that each task, process, or user is granted the minimum rights required to perform its job. By applying this principle consistently, if a task, process, or user is compromised, the scope of damage is constrained to the limited resources available to the compromised entity.

- **Psychological Acceptability**—Users should understand the necessity of security. This can be provided through training and education. In addition, the security mechanisms in place should present users with sensible options that give them the usability they require on a daily basis. If users find the security mechanisms too cumbersome, they may devise ways to work around or compromise them. The objective is not to weaken security so it is understandable and acceptable, but to train and educate users and to design security mechanisms and policies that are usable and effective.

- **Least Common Mechanism**—When providing a feature for the system, it is best to have a single process or service gain some function without granting that same function to other parts of the system. The ability for the Web server process to access a back-end database, for instance, should not also enable other applications on the system to access the back-end database.

- **Defense-in-Depth**—Organizations should understand that a single security mechanism is generally insufficient. Security mechanisms (defenses) need to be layered so that compromise of a single security mechanism is insufficient to compromise a host or network. No "silver bullet" exists for information system security.

- **Work Factor**—Organizations should understand what it would take to break the system or network's security features. The amount of work necessary for an attacker to break the system or network should exceed the value that the attacker would gain from a successful compromise.

- **Compromise Recording**—Records and logs should be maintained so that if a compromise does occur, evidence of the attack is available to the organization. This information can assist in securing the network and host after the compromise and aid in identifying the methods and exploits used by the attacker. This information can be used to better secure the host or network in the future. In addition, these records and logs can assist organizations in identifying and prosecuting attackers.

Administration issues

The Server administration includes designing, installing, administering, and optimizing company servers and related components to achieve high performance of the various business applications supported by tuning the servers as necessary. This includes ensuring the availability of client/server applications, configuring all new implementations, and developing processes and procedures for ongoing management of the server environment. Where applicable, the Server Administrator will assist in overseeing the physical security, integrity, and safety of the data center/server farm.

Server Administration is handled by the administrators. Server administrators are system architects responsible for the overall design, implementation, and maintenance of a server. Network administrators are responsible for the overall design, implementation, and maintenance of a network.

The Vital activities include handling and analyzing log files, performing regular server backups, recovering from server compromises, testing server security regularly, and performing remote administration securely.

Logging

Logging is a cornerstone of a sound security posture. Capturing the correct data in the logs and then monitoring those logs closely is vital.35 Network and system logs are important, especially system logs in the case of encrypted communications, where network monitoring is less effective. Server software can provide additional log data relevant to server-specific events.

Reviewing logs is mundane and reactive, and many server administrators devote their time to performing duties that they consider more important or urgent. However, log files are often the only record of suspicious behavior. Enabling the mechanisms to log information allows the logs to be used to detect failed and successful intrusion attempts and to initiate alert mechanisms when further investigation is needed. Procedures and tools need to be in place to process and analyze the log files and to review alert notifications.
Server logs provide—

- Alerts to suspicious activities that require further investigation
- Tracking of an attacker's activities
- Assistance in the recovery of the server
- Assistance in post-event investigation
- Required information for legal proceedings.

Server Backup Procedures

One of the most important functions of a server administrator is to maintain the integrity of the data on the server. This is important because servers are often some of the most exposed and vital hosts on an organization's network. The server administrator needs to perform backups of the server on a regular basis for several reasons. A server could fail as a result of a malicious or unintentional act or a hardware or software failure. In addition, Federal agencies and many other organizations are governed by regulations on the backup and archiving of server data. Server data should also be backed up regularly for legal and financial reasons.

Security Testing Servers

Periodic security testing of servers is critical. Without periodic testing, there is no assurance that current protective measures are working or that the security patch applied by the server administrator is functioning as advertised. Although a variety of security testing techniques exists, vulnerability scanning is the most common. Vulnerability scanning assists a server administrator in identifying vulnerabilities and verifying whether the existing security measures are effective. Penetration testing is also used, but it is used less frequently and usually only as part of an overall penetration test of the organization's network

Authorization and Authentication

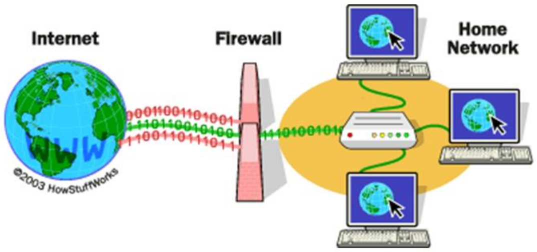
Vulnerability Scanning

Vulnerability scanners are automated tools that are used to identify vulnerabilities and misconfigurations of hosts. Many vulnerability scanners also provide information about mitigating discovered vulnerabilities. Vulnerability scanners attempt to identify vulnerabilities in the hosts scanned. Vulnerability scanners can help identify out-of-date software versions, missing patches, or system upgrades, and they can validate compliance with or deviations from the organization's security policy. To accomplish this, vulnerability scanners identify OSs, server software, and other major software applications running on hosts and match them with known vulnerabilities in their vulnerability databases.

Firewalls:

Firewall is hardware device or software applications that act as filters between a company's private network and the internet. It protects networked computers from intentional hostile intrusion that could compromise confidentiality or result in data corruption or denial of service by enforcing an access control policy between two networks.

The main purpose of a firewall system is to control access to or from a protected network (i.e., a site). It implements a network access policy by forcing connections to pass through the firewall, where they can be examined and evaluated. A firewall system can be a router, a personal computer, a host, or a collection of hosts, set up specifically to shield a site or subnet from protocols and services that can be abused from hosts outside the subnet. A firewall system is usually located at a higher level gateway, such as a site's connection to the Internet, however firewall systems can be located at lower-level gateways to provide protection for some smaller collection of hosts or subnets. The main function of a firewall is to centralize access control. A firewall serves as the gatekeeper between the untrusted Internet and the more trusted internal networks. The earliest firewalls were simply routers.



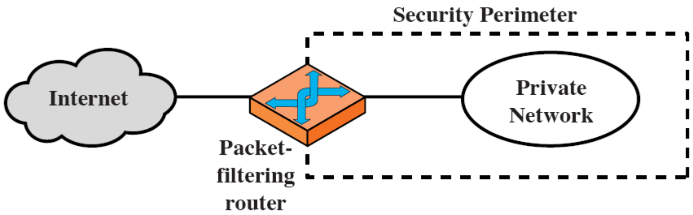
Firewalls provide several types of protection:

- They can block unwanted traffic.
- They can direct incoming traffic to more trustworthy internal systems.
- They hide vulnerable systems, which can't easily be secured from the Internet.
- They can log traffic to and from the private network.
- They can hide information like system names, network topology, network device types, and internal user ID's from the Internet.
- They can provide more robust authentication than standard applications might be able to do.

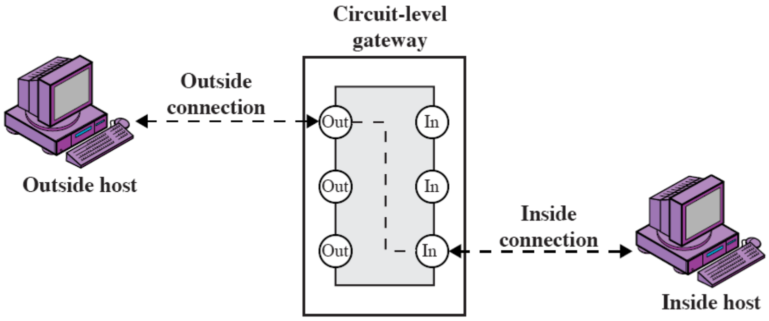
What does a firewall do?

A firewall examines all traffic routed between the two networks to see if it meets certain criteria. If it does, it is routed between the networks, otherwise it is stopped. It can be used to log all attempts to enter the private network and trigger alarms when hostile or unauthorized entry is attempted. Firewalls can filter packets based on their source and destination addresses and port numbers. This is known as address filtering. Firewalls can also filter specific types of network traffic. This is also known as protocol filtering because the decision to forward or reject traffic is dependent upon the protocol used, for example HTTP, ftp or telnet. Firewalls can also filter traffic by packet attribute or state.
There are different types of firewalls available in today's world some of them are:

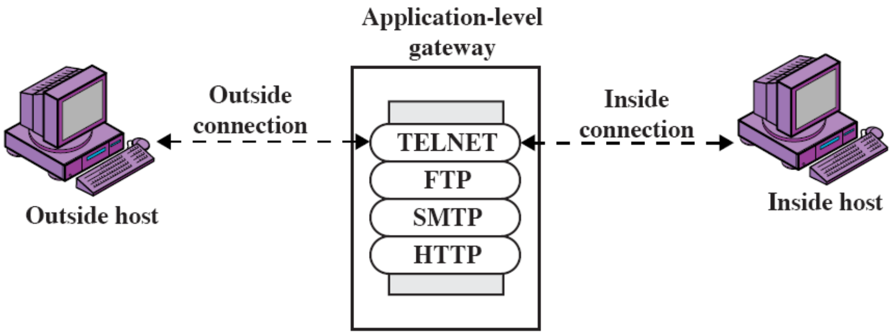
Packet Filters: Packet filtering firewalls work at the network layer (OSI model), or the IP layer (TCP/IP). In this each packet is compared to a set of criteria before it is forwarded. Depending on the packet and the criteria, the firewall can drop, forward the packet or send a message to the originator. Rules can be source and destination IP address, source and destination port number and protocol used. The advantages of packet filtering firewalls is their low cost and low impact on network performance



Circuit Level Gateways: It work at the session layer (OSI model), or the TCP layer (TCP/IP). They monitor TCP handshaking between packets to determine whether a requested session is legitimate. Information passed to a remote computer through a circuit level gateway appears to have originated from the gateway. This is useful for hiding information about protected networks. Circuit level gateways are relatively inexpensive and have the advantage of hiding information about the private network they protect. On the other hand, they do not filter individual packets.



Application Gateways: Application level gateways, also called proxies, are similar to circuit-level gateways except that they are application specific. They can filter packets at the application layer of the OSI model. Incoming or outgoing packets cannot access services for which there is no proxy. In plain terms, an application level gateway that is configured to be a web proxy acts as the server to the internal network and client to the external network. Because they examine packets at application layer, they can filter application specific commands such as http: post and get, etc. Application level gateways can also be used to log user activity and logins. They offer a high level of security, but have a significant impact on network performance.



Stateful Multilayer Inspection Firewall: It combines the aspects of the other three types of firewalls. They filter packets at the network layer, determine whether session packets are legitimate and evaluate contents of packets at the application layer. They rely on algorithms to recognize and process application layer data instead of running application specific proxies. Stateful multilayer inspection firewalls offer a high level of security, good performance and transparency to end users. They are expensive however, and due to their complexity are potentially less secure than simpler types of firewalls if not administered by highly competent personnel.

Content Filtering:

Content filtering is the technique whereby content is blocked or allowed based on analysis of its content, rather than its source or other criteria. It is most widely used on the internet to filter email and web access.

Content filtering on Web commonly is named Web filtering.

Content filtering is commonly used by organizations such as offices and schools to prevent computer users from viewing inappropriate web sites or content, or as a pre-emptive security measure to prevent access of known malware hosts. **Filtering rules are typically set by a central IT department and may be implemented via software on individual computers or at a central point on the network such as the proxy server or internet router.** Depending on the sophistication of the system used, it may be possible for different computer users to have different levels of internet access.

On the Internet, content filtering (also known as *information filtering*) is the use of a program to screen and exclude from access or availability Web pages or e-mail that is deemed objectionable. Content filtering is used by corporations as part of Internet firewall computers and also by home computer owners, especially by parents to screen the content their children have access to from a computer.

Content filtering usually works by specifying character strings that, if matched, indicate undesirable content that is to be screened out. Content is typically screened for pornographic content and sometimes also for violence- or hate-oriented content. Critics of content filtering programs point out that it is not difficult to unintentionally exclude desirable content.

Content filtering and the products that offer this service can be divided into Web filtering, the screening of Web sites or pages, and e-mail filtering, the screening of e-mail for spam or other objectionable content.

Filters can be implemented in many different ways: by a software program on a personal computer via network infrastructure such as proxy servers that provide Internet access.

Browser based filters: Browser based content filtering solution is the most lightweight solution to do the content filtering, and is implemented via a third party browser extension.

Client-side filters: This type of filter is installed as software on each computer where filtering is required. This filter can typically be managed, disabled or uninstalled by anyone who has administrator-level privileges on the system.

Content-limited (or filtered) ISPs: Content-limited (or filtered) ISPs are Internet service providers that offer access to only a set portion of Internet content on an opt-in or a mandatory basis. Anyone who subscribes to this type of service is subject to restrictions. The type of filters can be used to implement government, regulatory or parental control over subscribers.

Network-based filtering: This type of filter is implemented at the transport layer as a transparent proxy, or at the application layer as a web proxy. Filtering software may include data loss prevention functionality to filter outbound as well as inbound information. All users are subject to the access policy defined by the institution. The filtering can be customized, so a school district's high school library can have a different filtering profile than the district's junior high school library.

Search-engine filters: Many search engines, such as Google and Alta Vista offer users the option of turning on a safety filter. When this safety filter is activated, it filters out the inappropriate links from all of the search results. If one knows the actual URL of a website that features sexual explicit or 18 + content, they have the ability to access it without using a search engine. Engines like Lycos, Yahoo, and Bing offer kid-oriented versions of their engines that permit only children friendly websites.

There are four options for content filtering: Client based software, server based software, stand-alone appliances (hardware) and managed service.

- **Client based software** - Client based software is installed on the workstation that is used to surf the Internet. This software is in addition to web browsers which offer a certain level of content filtering. Software requires separate installation and maintenance for each workstation. Client based software may be used in environments with a limited number of users.
- **Server based software** - Server based software is installed on a host server such as a web server, proxy server, or firewall and is maintained using administrator software. Server based software allows for the central control of an entire network or single subnet with a single installation. Because of cost concerns and the required network infrastructure, server-based software is typically used in mid to large-sized environments.
- **Stand-alone appliance** - Stand-alone appliances are hardware devices that can be installed on a network you wish to filter and monitor. These are generally higher performance solutions because they use dedicated hardware and are optimized for filtering. Appliance solutions can be scaled for deployment in small to large environments.
- **Managed Service** - This solution involves outsourcing a portion or all of a company's security infrastructure including content filtering. Third-party managed security solutions have been deployed in all environments.

Some Internet Service Providers (ISP's) provide content filtering as part of a business grade broadband offering.

Content Filtering Types

- **From Address**

1. From specific addresses
2. From specific domains
3. From trusted senders

- **Contains Specific Words or Phrases**

4. Subject
5. Body Text
6. Subject or Body Text
7. From Address
8. To Address
9. Email Headers
10. Anywhere in Message

- **To Address**

11. To Specific Addresses
12. To Specific Domains
13. Only to Me
14. My Address in To Field
15. My Address not in To Field
16. My Address in To or CC Field

- **Attachments**

17. Has any Attachment
18. Specific Filenames
19. Specific Extensions
20. Over Specific Size

- **Other**

21. Flagged as High Priority
22. Flagged as Normal Priority
23. Flagged as Low Priority
24. Message is Automated (no return address)
25. Message under Size
26. Message over Size
27. Received in Date Range
28. Sent through a Specific Server (by IP)
29. Spam Probability

Email Filtering:

Email filtering is the processing of email to organize it according to specified criteria. Most often this refers to the automatic processing of incoming messages, but the term also applies to the intervention of human intelligence in addition to anti-spam techniques, and to outgoing emails as well as those being received.

Email filtering softwares are given emails as inputs. For its output, it might pass the message through unchanged for delivery to the user's mailbox, redirect the message for delivery elsewhere, or even throw the message away. Some mail filters are able to edit messages during processing.

Mail filters can operate on inbound and outbound email traffic. Inbound email filtering involves scanning messages from the Internet addressed to users protected by the filtering system or for lawful interception. Outbound email filtering involves the reverse - scanning email messages from local users before any potentially harmful messages can be delivered to others on the Internet. One method of outbound email filtering that is commonly used by Internet service providers is transparent **SMTP proxying**, in which email traffic is intercepted and filtered via a transparent proxy within the network. Outbound filtering can also take place in an email server. Many corporations employ data leak prevention technology in their outbound mail servers to prevent the leakage of sensitive information via email.

SMTP Proxy:

SMTP Proxies are commonly used to process and filter inbound and outbound email traffic. SMTP proxy can be used to control email messages and email content. The proxy scans SMTP messages for a number of filtered parameters, and compares them against the rules in the proxy configuration.

With an SMTP proxy filter you can:

- Adjust timeout, maximum email size, and line length limit to make sure the SMTP proxy does not use too many network resources and can prevent some types of attacks.
- Customize the deny message that users see when an email they try to receive is blocked.
- Filter content embedded in email with MIME types and name patterns.
- Limit the email addresses that email can be addressed to and automatically block email from specific senders.

SMTP proxies are specialized Mail Transfer Agents (MTAs) that, similar to other types of proxy servers, pass SMTP sessions through to other MTAs without using the store-and-forward approach of a typical MTA. When an SMTP proxy receives a connection, it initiates another SMTP session to a destination MTA. Any errors or status information from the destination MTA will be passed back to the sending MTA through the proxy

SMTP proxies come in a few fundamental flavors:

- **Synchronous** - each SMTP client connection causes the proxy to establish a single connection with a downstream mail server.
- **Multiplexing** - the proxy establishes downstream connections to the mail server only as needed, and by intelligently juggling a pool of SMTP connections; this juggling protects the downstream mail server from excessive connection concurrency
- **Transparent** - the proxy is inserted into the network between clients and servers, masquerading itself in such a way that the client and server believe they are talking directly to each other, even though there is a proxy in the middle.