# Computer networking and management

Business, Management

Task 1

1. Firewall technologies

The basic building-block technologies of a firewall are a packet filtering router, which is typically a piece of hardware; circuit gateways which are composed of two software modules; and proxy software which are also called application gateways. These basic building blocks of firewalls can be combined, mixed and matched to create a variety of firewall designs. This section explains the working of two of the basic building blocks.

a. Packet filtering Router

Packet filleting routers, placed between two networks, are the simplest type of firewall. A router sits at a network junction or intersection and directs the network traffic i. e. packets towards it correct definition. Packet filtering routers are the most common and oldest firewall device in use. A packet filtering router applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions i. e. from and to the internal network. Filtering rules are based on information contained in a network packet such as source IP address, destination IP address, source and destination transport-level address, IP protocol field and interface (Stallings, 2007: 359-360). The filtering rules are set by the network administrator. In establishing a set of filtering rules, the main idea is to list permitted or denied network address and ports. In this way, routers can restrict access to certain subnets or sets of both. Some information services are difficult to set up securely with a packet filtering router firewall because the protocol

requires that a call be made from outside to inside the firewall to set up the connection. In other words, the outside party must call back. In addition, the rules in a packet filtering router table are can be complex to set up, test, and maintain. If the filtering rules in the table aren't right, unauthorized traffic may be passing through the firewall unnoticed. In general, packet filtering routers alone are not flexible enough and not provide an adequate level of fine-grained control to be adequate firewalls. The figure below shows the operation of a typical packet filtering router firewall (Anonymous, 2002: 199).

Fig – 1, Packet filtering router (Stallings, 2007: 359)

b. Application Gateway

An application level gateway firewall is also known as a proxy firewall. A proxy is a mechanism that copies packets from one network to another; the copy process also changes the source and destination address to protect the identity of the internal or private network. An application-level gateway firewall filters traffic based on the Internet Service i. e. application used to transmit or receive data. Each type of application must have its own unique proxy server. Thus, an application-level gateway firewall comprises numerous individual proxy servers. This type of firewall negatively affects network performance because each packet must be examined and processed as it passes through the firewall. Application-level gateways are known as second-generation firewalls, and they operate at the application i. e. 7th or highest layer of the OSI model (Stewart, Tittel, Chapple, 2005: 98). As the application gateways examine the packets at the application layer, they can filter applications specific commands. This cannot be done with either packet filtering or circuit-level firewalls because they do not know the

application-level information. For this reason application-level gateways can also be used to log user activity and logins. They offer a high level of security though they do affect the network performance due to the use of context switches that slow down network access. They are not transparent to end users and require manual configuration for each client computer. The figure below shows the operation of a typical application-level gateway firewall (Stallings, 2007: 364).

Fig – 2, Application-level gateway firewall (Stallings, 2007: 359)

Read also about Threshold Capabilities

2. Real world Example of firewall working

The following is example is that of the Los Angeles site at Security Corp. the company has its LAN connected via a firewall. The firewall which the company has decided to use is the Cisco's PIX firewall Model PIC 515. This model is intended for the enterprise core of small to medium size businesses. This product has wire speed performance and the pipe carries the ability to handle up to 170 Mbps of clear-text throughput. The chassis is a configurable IU pizza box that is intended for rack mounting and comes with an slot for an additional single-port or four-port Fast Ethernet interface. This allows the inside, outside and up to four additional service networks. The base unit is based on 200MHz Intel Pentium MMX with 32 MB RAM and 8 MB flash. The licensing is flexible, so enterprises are able to purchase only what they need. The restricted license limits the number of interfaces to three and does not support high availability. The unrestricted license allows

for an increase in RA < from 32 MB to 64 MB and up to six interfaces together with failover capacity.

It is well known that merely having a firewall is not at all enough for network security. There needs to be a clear security policy as per organizationalgoalsand objectives, which would define the access requirements. Security Corp. has such a policy in place and has a clear demarcation of the network requirements by different levels of users. PIX is considered to be an extremely good firewall option for small and medium businesses. The drawback of PIX is that configuring it for inbound and outbound traffic required multiple steps. The process can however be made easier by first controlling the outbound traffic. Once outbound access it controlled, allowing the inbound access is relatively easy.

Task 2

1. Operation of a Web cache

Web caches preserve a local version of responses served from origin servers to the client browser. The web cache keeps track of responses served for specific URL requests, and if there are instructions to store the response in the cache, it remembers them for a specific period. Next time when the same URL is requested, the web cache intercepts the request and returns the stored response from its storage to the client browser. Using a web cache can significantly reduce the server load and response time to users, and smooth the network traffic when the cache is physically close to the users. The web cache is also sometimes known as an HTTP proxy server and is located between a web server and web browsers in a network and

provides services for all browsers within a local network. The HTTP proxy server forwards requests in behalf of clients and relays responses from servers (Davies, Schulte, Barnett, 2004: 51). The figure below shows normal web browsing using an HTTP proxy server.

Fig – 3, Normal web browsing using HTP proxy server (Deibert, Palfrey, Rohozinski, Zittrain, 2008: 62)

The six actions shown above are:

i. User requests www. example. org/page/html

ii. DNS looks up for www. example. org

iii. Look up response: www. example. org is 192. 0. 2. 166

iv. Get web page: www. example. org/page. html at 192. 0. 2. 166

v. Here is www. example. org/page. html

vi. Here is www. example. org/page. html

(Deibert, Palfrey, Rohozinski, Zittrain, 2008: 62)

However, as well as improving the performance, an HTTP proxy server can also block websites. The proxy decides whether requests for web pages should be permitted, and if so, send the request to the web server hosting the requested content. Since the full content of the request is available, web pages can be filtered, not just entire web servers or domains (Deibert, Palfrey, Rohozinski, Zittrain, 2008: 62).

2. Operation of a Content Distribution Network CDN

A content distribution network, CDN, is essentially a network that is able to distribute content closer to the end user, to provide faster and consistent response time. CDNS may come in different flavors. CDNS are designed to deliver large volumes of traffic quickly and efficiently. CDN has multiple data centers and generally the data is replicated across each data center. This is done both for data integrity, so that if there is a powerfailuresomewhere the files are still available, and also for speed. Most CDNs also rely on caching to make downloads happen faster (Verma, 2003: 13). CDNs minimize the delay that the end user experiences when requesting to view or download content such as HTML pages, data files, images and streaming media by deploying and managing a network of edge caches or proxy caches which maintain content replicas that emanate from an origin server. These edge caches are closer to end user in terms of lower response latency i. e. small number of router hops, higher bandwidth connections etc. The operation of a CDN is shown in the figure below. Algorithms for cache management are employed to optimize resource utilization based on user content consumption patterns. CDNs may also supports content management features such as content preloading, prioritization, publication and expiry dates, and digital rights management. In addition to the figure given below, many other CDN architectures are possible including the use of multicast distribution and peer-to-peer caching (Furht, 2006: 114).

Fig – 4, Operation of a Content Distribution Network (Furht, 2006: 114)

3. Comparison between web cache and CDN

Both web caching and content distribution networks aim to scale the network and distribute content to the network edge. They significantly improve user

experience by serving web content locally. However, although web caching solves some scalability and performance problems, content providers rarely have control of cached objects and the QoS end user receives. CDNs have been developed to address these issues. By using CDNs, content providers are able to not only have global; reach, but also get guaranteed and measurable QoS and control over content (Dixit, Prasad, 2003: 550). The HTTP client server model does not scale well as the number of clients and the network bandwidth utilization increase. The server is a choke point, and a single point of failure. Adding redundancy and load balancing can help, but this does not deal with overall network load. Content distribution networks are designed to minimize the delay that the end user experiences when requesting to view or download content such as HTML pages, images, data files and streaming media (Furht, 2006: 114).

The proxy server may have a cache which is shared by several clients. Static documents which are designated as cacheable may be stored in the proxy cache when first requested by a client. A subsequent request for the object from another client sharing the cache would obtain the object from proxy cache instead of going out to the network the fetch the objects. CDNs cache content remotely from servers at edges of the network. Websites will pay to use a CDN to offload requests and move contents closer to clients. By contrast, proxy caches function on behalf of clients, and website does not pay to use a proxy site (Getov, Gerndt, Hoisie, Malony, Miller, 2002: 217).

Task 3

1. Differences between TCP and UDP

Both TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are standard protocols used by applications to send information over a network. The choice of which to use for a given application depends on the design goals of the application. This section outlines the main differences between the UDP and TCP transport layer network services

Ø The main difference between TCP and UCP is that TCP is connection oriented while UDP is connectionless. This means that UDP does not make sure that all packets that are sent are also received because it has no provisions for checking whether all packets really arrive at the destination. Because of this USP is sometimes referred to as the unreliable protocol. TCP on the other hand has provisions for session set up and session maintenance as both the hosts that are a part of a TCP connection keep track of the conversation. TCP keeps track of the transmission and ensures that all parts of the transmission are successful and hence is called a reliable protocol (Robbins et al. 2003: 724; Huston, Johnson, Syyid, 2004: 208).

Ø UDP is based on messages while TCP is based on byte streams. For unconnected UDP sockets, if an application sends three 256 byte datagrams, the receiving host will receive anywhere from zero to all three of them. A datagram that is received will however be the complete 256 byte datagram sent – it will not be broken up or combined with some other datagram. This makes UDP more record oriented. In case of TCP sends the three 256 byte buffers of data, the receiving host will receive 768 bytes of data in the same order they were transmitted but the data can be received in any number of separate chunks without any guarantee of where the breaks between chunks will be. Hence, for TCP a way is needed to extract the streamed data

correctly, referred to as unmarshalling (Robbins et al. 2003: 724; Huston et al., 2004: 208).

Ø UDP makes no guarantees about the arrival or order of data whereas TCP guarantees that any data received is precisely what was sent and that it arrives in order. In other words, even when no errors occur anywhere in the network, UDP can deliver messages out of order (Huston et al., 2004: 208). While UDP does deliver the messages to the application in the order they are received, the UDP packets may travel different routes on the Internet and may not arrive in the order in which they are sent. In contrast, the network subsystem of the receiving host buffers TCP packets and uses sequence numbers to deliver bytes to the application in the order they were sent (Robbins et al. 2003: 724).

Ø Whereas TCP is a one-to-one connection between two peers, UDP offers several modes of operation: unicast, broadcast, and multicast. Unicast is a one-to-one operation similar to TCP. In broadcast mode, each datagram sent is broadcast to every listener on the network or sub network the datagram is broadcast on. The multicast mode consists of application joined in multicast modes with unique IP addresses. Any datagram sent to a multicast group is received only by those stations subscribed to the group (Robbins et al. 2003: 724; Huston el al., 2004: 208).

Ø If TCP cannot deliver data to the remote host, it eventually reports the failure by returning an error. UDP is unreliable. The network might drop UDP packets and never deliver them to the remote host. UDP does not notify either the sender or the receiver that an error has occurred (Robbins et al. 2003: 724 Huston el al., 2004: 208).

Ø TCP uses a three-way handshake before it starts a data transfer process. UDP just starts the data transmission process without any formal preliminaries (Robbins et al. 2003: 724).

2. Why the use of UDP for streaming applications is controversial?

The transport mechanism for streaming media is not adequately catered for by either TCP or UDP. While it is obvious that TCP is the choice for file-transfer type applications such as e-mail, UDP is the favored transport protocol for the delivery of audio and video applications. The decision is controversial because UDP provides a connectionless and unreliable datagram service with a minimum of overhead. This means that while there is a significant risk of data packets being lost or not arriving in sequence, the protocol allows the streaming applications to transmit as fast as they can. Another disadvantage of USP is that it does not enforce any congestion control. Thus, the excessive use of USP in the Internet can result in congestion collapse. In contrast TCP provides a reliable connection between two terminals on top of an unreliable communications network which may include wireless links, and also enforces congestion control. However, this congestion control may limit the transmission rate to sometimes even below the video bit rate (Salkintzis, 2004: 11-31).

The choice of UDP over TCP in streaming applications is controversial because eit does not follow any guidelines of logical selection. Even the standards body do not give any clues as to the reason for the choice. For instance, both USP and TCP are supported by 3GPP, however, most 3GPP application use UDP for streaming. This is confusing because research has clearly shown that TCP may provide acceptable quality at a lower application

complexity. However, the fact that UDP supports multicating and TCP does not, may be another point in its favor since this is an efficient way to utilize the scarce wireless bandwidth for video streaming (Smyth, 2004: 349).

Task 4

1. Single Sign On (SSO)Technology

Single Sign-On (SSO) can be defined as " an authentication mechanism that allows a single identity to be shared across multiple applications" (Contesti, Andre, Waxvik, Henry, Goins, 2007: 15). An SSO technology allows the user to authenticate once and gain access to multiple resources without needing to re-authenticate. As is obvious, primary purpose of SSO is for the convenience of the user as this et up eliminates the situation where separate passwords and user IDs are required for each application. SSO hence deals with userID and passwords pains by eliminating the need for users to remember the myriad of userIDs and passwords beyond their initial network log-in. SSO securely stores the userIDs and passwords that each user needs and automatically retrieves them for the user when required (Contesti et al., 2007: 15).

Single time authentication does not mean that an SSO system unidies the account information, usually userID and password, for all services, even if that is a popular interpretation for many people. Instead, it hides the account information for the participation services behind only one account. Thus, the user longs on to the SSO system once and the system manages the logins for the specific services the user chooses to work with. Especially, the system does not automatically perform a login for the user at all services

managed by the SSO system. A login takes place only at those services that the user chooses to work with (Tipton, Henry, 2006: 180).

A single sign-on is often referred to as reduced sign-on and federated ID management. By implementing an SSO solution that incorporates a single instance of user credentials leveraged by multiple systems, an organization's employees as well as partners can be offered access with limited management. In practice, most SSO systems operate as follows: First, a user provides a userID and password to a primary login program which authenticates the user against a master system, usually the server. Once authenticated, the user can request access to additional systems. When he/she does so, the SSO system retrieves the user's password for the new system, and starts a new session with the new system using that password. The figure below shows a common SSO architecture.

Fig – 5, Common SSO architecture (Tipton, Henry, 2006: 180)

Though single sign-on services have been used for many years, the lack of universal adoption and cost of integration has made their use in Web applications highly undesirable. These difficulties have led to the creation of SSO services targeted specifically to authentication on the Web. One of the most popular of these systems is the Microsoft Passport Service. Passport provides a single user authentication service and repository of user information. Websites and users initially negotiate secrets during the Passport registration process. In all cases, these secrets are known only to the Passport servers and the registering identity (Bidgoli, 2006: 42).

The initial authentication in SSO system can take place using any authentication type or combination of types, while the authentication to the subsequent systems can occur using an entirely different authentication type. Several vendors have come out with various forms of SSO technology to aid with authentication within large enterprises. Some of these SSO products are: CA-Unicenter, IntelliSoft SnareWorks, Kerberos, SESAME, Kryptoknoght, NetSP, Memco Proxima, Tivoli Global Sign-on, x. 509 etc (Jacobs, Posluns, Clemmer, Dalton, and Rogers, 2003: 45, 46).

2. Comparison of Password Synchronization and SSO Technology

As is mentioned earlier, an SSO technology allows a user to authenticate one time and then access resources in theenvironmentwithout needing to re-authenticate. This sounds similar to password synchronization, but it is not. With password synchronization, a product takes the user's password and updates each user account on each different system and application with that one password. That means the password must be typed by the user in each and every application and system which he/she wishes to access. When the user requests to access a network application, the application will send over a request for credentials, but the SSO system will responds to the application for the user. So in SSO environments, the SSO system intercepts the login prompts from network systems and applications and fills in the necessary identification and authentication information, i. e. the userID and password for the user. However, though password synchronization and single sign-on are different technologies, they still have the same vulnerability. If an attacker uncovers a user's credential set, he/she can have access to all

the resources that the legitimate user may have access to (Harris, 2007: 173).

3. Advantages of SSO Technology

Following are the advantages of SSSO solutions:

a. Efficient log-on process: Users require fewer passwords to remember and are prompted less to perform their job functions (Tipton, Henry, 2006: 180).

b. User may create strong passwords: With the reduced number of passwords to remember, users can remember a single, very strong password that can also be changed often (Tipton, Henry, 2006: 180).

c. No need for multiple passwords: The introduction of an SSO system translates into a single-use credential for users (Tipton, Henry, 2006: 180).

d. Time-out and attempt thresholds enforced across entire platform: The time-out threshold is used to protect against a user being away from his workstation but still logged on for an extended period, thereby leaving it available to an intruder who could continue with the user’s session. In this case, the workstation would be disconnected after selected period of inactivity. The attempt threshold is used to protect against an intruder attempting to obtain an authentic ID and password combination by trying many combinations. In this case, the workstation would be locked after a pre-specified number of trials till the administrator gives back the privileges (Tipton, Henry, 2006: 180).

e. Centralized administration: Given the singularity of the access control mechanism, administrators are offered central administrative interface to support the enterprise (Tipton, Henry, 2006: 180).

f. Easier administration: The process of administration of the network is easier as can be seen from the following situations:

i. When a new employee is hired, all of the accounts on all the systems that the employee needs to access can be quickly added from a single administration point.

ii. When an existing employee ends his services, all access can be quickly and simultaneously restricted at a single administration point.

iii. If an existing user loses his token or forgets his password, the administrator can quickly update the iser’s authentication credentials from a single administration point.

g. Good return on investment: SSO generally offers a good return on investment for the enterprise. The reduced administrative costs can often pay for the cost of implementing an SSO in a short period of time. However, it should be noted that if scripting is used to facilitate the use of SSO, the typical reduced administration costs associated with SSO could be negated.

4. Disadvantages of SSO Technology

Following are the disadvantages of SSSO solutions:

a. A compromised password allows intruders into all authorized resources: If the credential used for total access is compromised, the attacker would then have the privileges and capacity assigned to the original

user. Although risks are similar in nature to typical user name and password combination, SSO introduces complete access via a single account. An attacker would only be limited by the enterprise wide assigned privileges, as opposed to the assigned privileges for a given system (Tipton, Henry, 2006: 181).

b. Inclusion of unique platform may be challenging: SSO is complex and requires significant integration to be effective. It is not uncommon for a large enterprise to utilize hundreds, if not thousands of applications running on a wide variety of operating systems, each with their own approach to user management. Therefore, significant planning and analysis should be performed prior to embarking on a SSO solution (Tipton, Henry, 2006: 181).

c. Difficult to implement across the enterprise: Many systems use proprietary authentication systems that will not work well with standard SSO systems (Contesti, Andre, Waxvik, Henry, Goins, 2007: 16).

d. Time consuming to implement properly: Many underestimate the amount of time necessary to properly implement SSO across all systems in the enterprise (Contesti, Andre, Waxvik, Henry, Goins, 2007: 16).

e. Expensive to implement: Because of the difficulty and time involved top properly implement SSO, it is expensive. A redundant authentication server is required to avoid a single point of failure (Contesti, Andre, Waxvik, Henry, Goins, 2007: 16).

f. Finally implementing SSO means that all of the ussers' credentials for the company's resources are stored in one location. If an attacker was able

to break into this storehouse, he/she could access and do whatever he/she wanted witht hecompany’s resources.

5. SSO Technology types

a. Kerberos Single Sign-on

The Kerberos single sign-on system was developed to improve both security and user satisfaction. The name Kerberos refers to the mythical three-headed dog guarding the gates to the underworld. Kerberos provides security when the end points of the network are safe but the transmission path cannot be trusted, for instance, when the servers and workstations are trusted but the network is not. The concept of this operation is for the user to log in once to Kerberos. After login, the Kerberos system authenticates the user and grants access to all resources. Kerberos is a ticket authentication protocol based on symmetric cryptography. Kerberos uses the following components:

i. Key Distribution Center (KDC): Holds user and service cryptographic keys, provides authentication services, and creates and distributes session keys

ii. Authentication Service (AS): Functional component of the KDC that actually performs the authentication.

iii. Principals: All entities that use the Kerberos protocol for authentication are referred to as principals, which could be users, application, resources, or services.

iv. Realm: A set of principles, which are grouped together logically by an administrator. A KDC is responsible for one or more realms of principle.

v. Ticket granting service (TGS): The part of the KDC that creates and distributes tickets to the principles containing session keys.

vi. Ticket: An Authentication token

vii. Secret and session keys: Symmetric keys used for authentication purposes and data encryption.

(Harris, 2007: 43)

The operation of Kerberos is as follows:

i. The user authenticates to the Kerberos workstation software. Authentication may be a password or biometric method.

ii. The workstation software authenticates to the Kerberos server.

iii. Shared encryption keys are used. A network access ticket is created by Kerberos.

iv. A Kerberos access ticket is sent to the workstation, signed in the workstation’s shared encryption key. All other network servers receive a similar ticket granting the workstation access to shared servers

v. The user is automatically signed in to all servers.

(Cannon, Bergmann, Pamplin, 2008, 412)

The belief is that a user with a strong password and strong encryption will improve overall security. Unfortunately, Kerberos works only with specially modified versions of software designed for use with Kerberos. Merely

installing Kerberos will not improve security. There are compatibility problems with different versions of implementation. Special skills are required to made a Kerberos installation successful. First, knowledgeable installer will understand how to use separate domains to partition Kerberos access for better security. Second, restoring data from tape backup is quite involved. The Kerberos system must be shut down and the date rolled back to the timestamp of the file being restored. As soon as the file is restored, the time clocks must be rolled forwards again with the system resynchronized for the users. Any compromise of the Key Distribution Center (KDC) means that the entire system is compromised and must be shut down. Using Kerberos required highly experienced system administrators (Cannon, Bergmann, Pamplin, 2008, 412). The following are the characteristics and weakness pertaining to Kerberos:

Ø Provides authentication, confidentiality, and integrity, but not availability or no-repudiation

Ø The KDC can be a single-point-of-failure.

Ø Secret keys are stored on users' workstations.

Ø Session keys are stored on users' workstations in a cache or key tables.

Ø Network traffic is not protected if encryption is not enabled.

Ø KDC must be readily available and support the number of requests it receives from principles.

Ø All principals must have Kerberos software installed.

Ø Requires trusted, synchronized clocks within the network.

Ø The KDC should not allow any non-Kerberos network activity to take place.

(Harris, 2007: 44)

The figure below shows the design of a Kerberos single sign-on.

Fig – 6, Kerberos single sign-on (Cannon, Bergmann, Pamplin, 2008, 412)

b. SESAME

The Secure European System for Applications in a Multi-vendor Environment, SESAME project is a single sign-on technology developed to Kerberos functionality and improves upon its weaknesses. SESAME was funded in part by the Commission of the European Union's RACE program. It is a distributed access control for SSO that uses symmetric and asymmetric cryptographic techniques. It halpsto provide global access identity. SESAME uses symmetric and asymmetric cryptographic techniques to authenticate subjects to network resources. The architecture of a SESAME implementation requires numerous components (Contesti et al. 2007: 21)

Ø Client side components

- User Sponsor
- Client application
- APA client
- Security manager

Ø Domain Security Server

- Authentication server
- Privilege attribute server

- Key distribution server
- Security information management base

Ø Server side components

- PAC validation facility
- Security manager
- Server application

The figure below gives an overview of the various components of SESAME.

Fig – 7, Overview of SESAME Components (Ashley, Vandenwauver, 1999: 99)

The following is a simplified explanation of the SESAME process:

i. The user contacts the authentication server.

ii. The user receives the authentication certificate.

iii. The certificate is delivered to a privilege attribute server.

iv. The user receives a privilege attribute certificate.

v. The certificate is presented to the target that is to be accessed.

vi. An access control decision is made based on the certificate presented, as well as the access control list attached to the resource.

(Contesti et al. 2007: 21)

SESAME is very similar to Kerberos and both can be accessed through the General Security Services Application Programming Interface, GSS-API, which is a generic API for client-to-server authentication. Using standard APIs enables vendors to communicate with and use each other’s functionality.

However, there are notable differences between the two technologies. While Kerberos uses tickets to authenticate subjects to objects, SESAME uses Privileged Attribute Certificate PACs, which contain the subject's identity, access capabilities for the object, access time period, and lifetime of the PAC. The PAC is digitally signed so the object can validate it came from the trusted authentication server, which is referred to as the Privileged Attribute Server, PAS. The PAS holds a similar role to that of KDC within Kerberos. After a user successfully authenticates to the authentication server, AS, he is presented with a toke to give to the PAS. The PAS then creates a PAC for the user to the resource he is trying to access (Harris, 2007: 206). The figure below shows the basic overview of the SESAME process.

Fig – 8, Overview of SESAME process (Harris, 2007: 206)

c. Directory Services

A directory service is a centralized database of resources available to the network. It can be thought of as a telephone directory of network services and assets. A network service is a mechanism that identifies resources such as printers, filer servers, domain controllers, peripheral devices etc. on a network and provides a war to make them available to users and programs. A network directory service contains information about these different resources, providing a naming scheme, and a hierarchical database that outlines characteristics such as a name, logical and physical location, subjects that can access them, and the operations that can be carried out on the resources. It provides user access to network resources transparently without needing to know their exact location and access steps required to

access them. These issues are taken care for the user in the background (Harris, 2007: 42).

Users, clients, and processes consult the directory service to learn where a desired system of resource resides. Then once this address or location is known, access can be directed towards it. A directory service must be authenticated before queries and lookup activities can be performed. Even after authentication, the directory service will only reveal information to a subject based on that subject's assigned privileges. Some well-known commercial directory services include Microsoft's Active Directory and Novell's NetWare Directory Services (NDS), named eDirectory (Stewart, Tittel, Chapple, 2005: 22).

Directory services are tools that help organize and manage complex networks since they allow data files, application, and other information to be quickly and easily relocated within a network. This greatly simplifies administrative tasks, and it allows programmers and developers to better utilize network resources. The more current methods treat data and other network resources as objects. This object-oriented approach allows information to be stored and accessed based on certain characteristics or attributed. In addition to creating and storing data, directory service must publish appropriate data to users. Most directory service have implemented a model of hierarchy similar to the one illustrated in the figure below (Pastore, Dulaney, 2006: 238). This hierarchy allows an object to be uniquely identified to directory users.

Fig – 9, Directory service showing unique identification of a user(Pastore, Dulaney, 2006: 238)

Security for directory services is critical and is typically accomplished by using both authentication and access control. Some directory services used in the networking in present times have been described below:

i. LDAP: Lightweight Directory Access Protocols (LDAP) is a standard directory access protocol that allows queries to be made of different directories, specifically pared-down X-500-based directories. If a directory services supports LDAP, that directory can be queried with a n LDAP client, but it is the protocol LDAP that is growing in popularity and is being used extensively in online white and yellow pages. LDAP is the main access protocol used by Active Directory, which will be discussed next. It operates, by default, at port 389. The LDAP syntax uses commas between names (Pastore, Dulaney, 2006: 239).

ii. Active Directory: Microsoft implemented a directory service called Active Directory (AD) with Windows 2000. For Microsoft products, AD is the backbone for all security, access, and network implementations. AD gives administrators full control of resources. It is a propriety directory service that provides services for other directory services such as LDAP. One or more servers manage AD functions; these servers are connected in a tree structure that allows information to be shared or controlled through the entire AD structure (Pastore, Dulaney, 2006: 239). In conjunction with Active Directory, LDAP uses four different name types:

Ø Distinguished Name: A Distinguished Name (DN) exists for every object in AD. These values can’t be duplicates and must be unique. This is the full path of object, including any containers

Ø Relative Distinguished Name: A Relative Distinguished Name (RDN) doesn't need to be a wholly unique value as long as there are no duplicates within the organizational unit (OU). As such, an RDN is the portion of the name that is unique within its container.

Ø Use Principal Name: A User Principle Name (UPN) is often referred to as a friendly name. It consists of the user account and the user's domain name and is used to identify the user, like an e-mail address.

Ø Canonical Name: The Canonical Name (CN) is the DN given in a top-down notation.

(Pastore, Dulaney, 2006: 239)

iii. X. 500: The International Telecommunications Union (ITU), an international standards group for directory services in the late 1980s, implemented the X-500 standard, which was the basis for later models of directory services of directory services such as LDAP. The major problem in the industry in implementing a full-blown X-500 structure revolved around the complexity of the implementation. Novell was one of the first manufacturers to implement X. 500 in its NetWare NDS product (Pastore, Dulaney, 2006: 239).

iv. eDirectory: eDirectory is the backbone for new Novell networks. It stores information on all system resources, users, and any other relevant information about the systems attached to a NetWare server. eDirectory is an upgrade and replacement for NDS, and it has gained wide acceptable in the community (Pastore, Dulaney, 2006: 239).

d. Security Domains

A domain is merely a set of resources available to a subject. A subject can be a user, process, or application. Within an operating system, a process has a domain, which is a set of system resources available to the process to carry out its tasks. These resources can be memory segments, hard drive space, operating system services, and other processes. In a network environment, a domain is a set of physical and logical resources that is available., which can include routers, file servers, FTP service, web servers et. The term security domain builds upon the definition of domain by adding the fact those resources within this logical structure, domain, are working under the same security policy and managed by the same group. So, a network administrator may put all of the accounting personnel, computers, and network resources in Domain 1 and all of the management personnel, computers,, and network resources in Domain 2. These items fall into these individual containers because they not only carry out similar types of business functions, but also, and more importantly, have same type of trust level. It is this common trust level that allows entities to be managed by one security policy (Harris, 2007: 206-207).

The different domains are separated by logical boundaries, such as firewalls with ACLs, directory services making access decisions, and objects that have their own ACLs indicating which individuals and groups can carry out operations on them. All these security mechanisms are example7s of components that enforce the security policy for each domain (Harris, 2007: 20).

Domains can be architected in a hierarchical manner that dictates the relationship between the different domains and the ways in which subjects

within the different domains can communicate. Subjects can access resources of equal or lowest trust level. The figure below shows an example of hierarchical network domains. Theircommunicationchannels are controlled by security agents such as firewalls, router ACLs, directory services, and the individual domains are isolated by using specific subnet mask address (Harris, 2007: 207).

Fig – 10, Network domains are used to separate different network segments (Harris, 2007: 207)

A domain does not necessarily pertain only to network devices and segmentations, but can also apply to users and processes. The figure below shows how users and processes can have more granular domains assigned to them individually based on their trust level. Group 1 has a high trust level and can access both a domain of its down trust level, Domain 1, and a domain of a lower trust level, Domain 2. User 1, who has a lower trust level can access only the domain at his trust level and nothing higher. The system enforces these domains with access privileges and rights provided by the file system and operating system security kernel (Harris, 2007: 208).

Fig – 11, Subjects can access specific domains based on their trsut levels (Harris, 2007: 208)

Domains are of the type single sign-on because several different types of technologies available today are used to define and enforce these domains and security policies mapped to them: domain controllers in a Windows environment, enterprise resource management (ERM) products, Microsoft Passport, and the various products that use SSO functionality. The goal of

each of them is to allow a user to sign in at one time and be able to access the different domains available without having to reenter any other credentials (Harris, 2007: 208).

6. Types of SSO Technologies

Two types of single sign-on technology are available to most enterprises:

Ø Native Single sign-on: The native single sign-on describes a situation in which all of the involved systems have the capability to have the same password and other access control setting, parameters, and values. These systems also have incorporated the capability to communicate between systems so that if an end user changes his or her end-user identification code or related password, it is changed on all of the system at the same time (Krist, 1998: 9-12).

Ø Pseudo single sign-on: This artificial approach is based on an application that, after installation, takes place in front of the true security processing in terms of what programs get processed before others by the central processing unit. The application can be programmed with the specifications of the security software of each system, and can respond interactively with each of those applications just as the end user would. This front-end software can handle most requests for end-user identification codes and passwords and can attempt to handle any other responses that are required (Krist, 1998: 9-12).

7. Different ways of implementation of SSO

SSO can be implemented in many ways. Following are the two of the most popular ways of SSO implementation.

Ø Script-based single sign-on: This is an older method that is not very much used today. Here the user logs into a primary network operating system and the NOS stored the passwords and authentication mechanism for other systems. When the user logs on, the network operating system passes the authentication credentials to all other systems. The drawback to script-based SSO is typically those passwords are stored in plaintext and they are transmitted over the network in plaintext. Another major drawback to the methods is the lack of encryption of sensitive information, userIDs, and passwords. Any attacker running a network sniffer can uncover the authentication credentials and bypass the access control system in place (Peltier, 2006: 201).

Ø Host-based single sign-on: This SSO method is more commonly used currently. Host-based SSO uses a centralized authentication server that all applications and systems utilize for authentication purposes. One of the major drawbacks of any type of host-based SSO is that there is an SPF or a Single Point of failure. If that authentication server goes down, the network becomes non-functional. Because no authentication can take place it is as if all systems are lost on the network (Peltier, 2006: 201-202).

REFERENCES

Anonymous, (2002), Maximum Security: A Hackers Guide to Protecting Your computer Systems and Network, 4th Edition, Indianapolis: Sams Publishing

Ashley P, Vandenwauver M, (1999), Practical Intranet Security: Overview of the State of the Art and Available Technologies, Boston: Springer & KluwerAcademicPublishers

Bidgoli H, (2006), Handbook of Information Security: Threats, Vulnerabilities, Prevention, Detection, and Management, Vol: 3, Danvers, Massachusetts: John Wiley and Sons

Cannon DL, Bergmann TS, Pamplin B, (2008), CISA Certified Information Systems Auditor Study Guide: Certified Information System Auditor: Study Guide, 2nd Edition, Danvers, Massachusetts: John Wiley and Sons

Contesti DL, Andre D, Waxvik E, Henry PA, Goins BA, (2007), Official (ISC) 2 Guide to the SSCP CBK, Danvers, Massachusetts: CRC Press

Davies J, Schulte W, Barnett M, (2004), Formal Methods and Software Engineering: 6th International Conference on Formal Engineering Methods, ICFEM 2004, Seattle, WA, USA, November 8-12, 2004: Proceedings, New York: Springer

Deibert R, Palfrey JG, Rohozinski R, Zittrain J, (2008), Access Denied: The Practice and Policy of Global Internet Filtering, Cambridge, Massachusetts: MIT Press

Dixit S, Prasad R, (2003), Wireless IP and Building the Mobile Internet, Boston: Artech House

Furht B, (2006), Encyclopedia of Multimedia, Florida: Springer

Getov V, Gerndt M, Hoisie A, Malony A, Miller B, (2002), Performance Analysis and Grid Computing: Selected Articles from the Workshop on Performance Analysis and Distributed Computing, August 19-23, 2002, Dagstuhl, Germany, Norwell, Massachusetts: Springer & Kluwer Academic Publishers

Harris S, (2007), CISSP All-in-one Exam Guide: Exam Guide, 4th Edition, New York: McGraw-Hill Professional

Huston SD, Johnson JCE, Syyid U, (2004), The ACE Programmer's Guide: Practical Design Patterns for Network and Systems Programming, Boston, Massachusetts: Addison-Wesley ; Pearson Publication

Jacobs J, Posluns J, Clemmer L, Dalton M, Rogers R, (2003), SSCP Systems Study Guide and DVD Training System: Study Guide ; Dvd Training System, Rockland, Massachusetts: Syngress

Krist MA, (1998), Standard for Auditing Computer Applications: Using Teams to Improve the Audit Process, 2nd Edition, Danvers, Massachusetts: CRC Press

Pastore MA, Dulaney E, (2006), CompTIA Security+ Study Guide: Exam SY0-101, 3rd Edition, Danvers, Massachusetts: John Wiley and Sons

Robbins KA, Robbins S, (2003), UNIX Systems Programming: Communication, Concurrency, and Threads, 4th Edition, New Jersey: Prentice Hall PTR

Salkintzis AK, (2004), Mobile Internet: Enabling Technologies and Services, Danvers, Massachusetts: CRC Press

Smyth P, (2004), Mobile and Wireless Communications: Key Technologies and Future Applications, IET

Stallings W, (2007), Network Security Essentials: Applications and Standards, 3rd Edition, New Jersey: Prentice Hall

Stewart JM, Tittel E, Chapple M, (2005), CISSP: Certified Information Systems Security Professional Study Guide, 3rd edition, Alameda, California: John Wiley and Sons

Tipton HF, Henry K, (2006), Official (ISC) 2 Guide to the CISSP CBK, Danvers, Massachusetts: CRC Press

Verma DC, (2003), Content Distribution Networks: An Engineering Approach, New York: Wiley-IEEE