Lecture 4

DESIGNING SECURE SYSTEMS

Specific design principles underlie the design and implementation of mechanisms for supporting security policies. These principles build on the ideas of simplicity and restriction.

Principle of Least Privilege

This principle restricts how privileges are granted. The principle of least privilege states that a subject should be given only those privileges that it needs in order to complete its task. If a subject does not need an access right, the subject should not have that right. Furthermore, the function of the subject (as opposed to its identity) should control the assignment of rights. If a specific action requires that a subject's access rights be augmented, those extra rights should be relinquished immediately on completion of the action.

Principle of Fail-Safe Defaults

This principle restricts how privileges are initialized when a subject or object is created.The principle of fail-safe defaults states that, unless a subject is given explicit access to an object, it should be denied access to that object.

This principle requires that the default access to an object is none. Whenever access, privileges, or some security-related attribute is not explicitly granted, it should be denied. Moreover, if the subject is unable to complete its action or task, it should undo those changes it made in the security state of the system before it terminates. This way, even if the program fails, the system is still safe.

Principle of Economy of Mechanism

This principle simplifies the design and implementation of security mechanisms.

The principle of economy of mechanism states that security mechanisms should be as simple as possible.

If a design and implementation are simple, fewer possibilities exist for errors. The checking and testing process is less complex, because fewer components and cases need to be tested. Complex mechanisms often make assumptions about the system and environment in which they run. If these assumptions are incorrect, security problems may result.

Principle of Complete Mediation

This principle restricts the caching of information, which often leads to simpler implementations of mechanisms.

The principle of complete mediation requires that all accesses to objects be checked to ensure that they are allowed.

Whenever a subject attempts to read an object, the operating system should mediate the action. First, it determines if the subject is allowed to read the object. If so, it provides the resources for the read to occur. If the subject tries to read the object again, the system should check that the subject is still allowed to read the object. Most systems would not make the second check. They would cache the results of the first check and base the second access on the cached results.

Principle of Open Design

This principle suggests that complexity does not add security.

The principle of open design states that the security of a mechanism should not depend on the secrecy of its design or implementation.

Designers and implementers of a program must not depend on secrecy of the details of their design and implementation to ensure security. Others can ferret out such details either through technical means, such as disassembly and analysis, or through nontechnical means, such as searching through garbage receptacles for source code listings (called "dumpster-diving"). If the strength of the program's security depends on the ignorance of the user, a knowledgeable user can defeat that security mechanism. The term "security through obscurity" captures this concept exactly.

Principle of Separation of Privilege

This principle is restrictive because it limits access to system entities.

The principle of separation of privilege states that a system should not grant permission based on a single condition.

This principle is equivalent to the separation of duty principle discussed earlier. Company checks for more than $75,000 must be signed by two officers of the company. If either does not sign, the check is not valid. The two conditions are the signatures of both officers.

Similarly, systems and programs granting access to resources should do so only when more than one condition is met. This provides a fine-grained control over the resource as well as additional assurance that the access is authorized.

Principle of Least Common Mechanism

This principle is restrictive because it limits sharing.

The principle of least common mechanism states that mechanisms used to access resources should not be shared.

Sharing resources provides a channel along which information can be transmitted, and so such sharing should be minimized. In practice, if the operating system provides support for

virtual machines, the operating system will enforce this privilege automatically to some degree.

Principle of Psychological Acceptability

This principle recognizes the human element in computer security.

The principle of psychological acceptability states that security mechanisms should not make the resource more difficult to access than if the security mechanisms were not present.

Configuring and executing a program should be as easy and as intuitive as possible, and any output should be clear, direct, and useful. If security-related software is too complicated to configure, system administrators may unintentionally set up the software in a nonsecure manner. Similarly, security-related user programs must be easy to use and must output understandable messages. If a password is rejected, the password changing program should state why it was rejected rather than giving a cryptic error message. If a configuration file has an incorrect parameter, the error message should describe the proper parameter.

FIREWALLS

A firewall is an approach to security; it helps implement a larger security policy that defines the services and access to be permitted, and it is an implementation of that policy in terms of a network configuration, one or more host systems and routers, and other
security measures such as advanced authentication in place of static passwords.
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.

What is being protected by firewalls?

Your data

- Secrecy - what others should not know
- Integrity - what others should not change
- Availability - your ability to use your own systems

Your resources

Your systems and their computational capabilities

Your reputation

- Confidence is shaken in your organization
- Your site can be used as a launching point for crime
- You may be used as a distribution site for unwanted data
- You may be used by impostors to cause serious problems

- You may be viewed as “untrusted” by customers and peers

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.

Firewall Security and Concepts

- The amount of security required for an entity is based on the security threat
- If you do not know what your threat is to the Intranet systems, it is extremely
- difficult to properly secure the environment and all systems interconnected
- Network compartmentalization is the buzzword for this type of effort
- Switching technology is a big help, but it does not tell you who is going where
- and why - that’s what analysis is all about
- Not knowing the threat causes false security to be deployed and money spent in the wrong places

The main reasons for systems and computers not being secure are

- Lack of password encryption
- Lack of personnel with experience
- Lack of management backing
- Authority
- Responsibility
- Legal and political issues
- Lack of recurring effort
- Budget

Firewall Components

The primary components (or aspects) of a firewall are:

- Network policy,
- Advanced authentication mechanisms,
- Packet filtering, and Application gateways.

F I R E W A L L D E S I G N P O L I C Y

The firewall design policy is specific to the firewall. It defines the rules used to implement the service access policy. One cannot design this policy in a vacuum isolated from understanding issues such as firewall capabilities and limitations, and threats and vulnerabilities associated with TCP/IP. Firewalls generally implement one of two basic

design policies:

- permit any service unless it is expressly denied, and
- deny any service unless it is expressly permitted.

A firewall that implements the first policy allows all services to pass into the site by default, with the exception of those services that the service access policy has identified as disallowed. A firewall that implements the second policy denies all services by default, but then passes those services that have been identified as allowed. This second policy follows the classic access model used in all areas of information security.

The first policy is less desirable, since it offers more avenues for getting around the firewall, e.g., users could access new services currently not denied by the policy (or even addressed by the policy) or run denied services at non-standard TCP/UDP ports that aren't denied by the policy. Certain services such as X Windows, FTP, Archie, and RPC cannot be filtered easily, and are better accommodated by a firewall that implements the first policy. The second policy is stronger and safer, but it is more difficult to implement and may impact users more in that certain services such as those just mentioned may have to be blocked or restricted more heavily.

Packet Filtering

IP packet filtering is done usually using a packet filtering router designed for filtering packets as they pass between the router's interfaces. A packet filtering router usually can filter IP packets based on some or all of the following fields:

- source IP address,
- destination IP address,
- TCP/UDP source port, and
- TCP/UDP destination port.

Problems with Packet Filtering Routers

Packet filtering routers suffer from a number of weaknesses.

Packet filtering rules are complex to specify and usually no testing facility exists for verifying the correctness of the rules (other than by exhaustive testing by hand). Some routers do not provide any logging capability, so that if a router's rules still let dangerous packets through, the packets may not be detected until a break-in has occurred.

Often times, exceptions to rules need to be made to allow certain types of access that normally would be blocked. But, exceptions to packet filtering rules sometimes can make the filtering rules so complex as to be unmanageable. For example, it is relatively straightforward to specify a rule to block all inbound connections to port 23 (the TELNET server). If exceptions are made, i.e., if certain site systems need to accept TELNET connections directly, then a rule for each system must be added. Sometimes the addition of certain rules may complicate the entire filtering scheme. As noted previously, testing a complex set of rules for correctness may be so difficult as to be impractical.

Some packet filtering routers do not filter on the TCP/UDP source port, which can make the filtering rule set more complex and can open up ``holes" in the filtering scheme.

APPLICATION GATEWAYS

To counter some of the weaknesses associated with packet filtering routers, firewalls need to use software applications to forward and filter connections for services such as TELNET and FTP. Such an application is referred to as a proxy service, while the host

running the proxy service is referred to as an application gateway. Application gateways and packet filtering routers can be combined to provide higher levels of security and flexibility than if either were used alone.
As an example, consider a site that blocks all incoming TELNET and FTP connections using a packet filtering router. The router allows TELNET and FTP packets to go to one host only, the TELNET/FTP application gateway. A user who wishes to connect inbound to a site system would have to connect first to the application gateway, and then to the destination host, as follows:

- a user first telnets to the application gateway and enters the name of an internal host,
- the gateway checks the user's source IP address and accepts or rejects it according
- to any access criteria in place,
- the user may need to authenticate herself (possibly using a one-time password
- device),
- the proxy service creates a TELNET connection between the gateway and the
- internal host,
- the proxy service then passes bytes between the two connections, and
- the application gateway logs the connection.

Application gateways have a number of general advantages over the default mode of permitting application traffic directly to internal hosts. These include:

information hiding, in which the names of internal systems need not necessarily be made known via DNS to outside systems, since the application gateway may be the only host whose name must be made known to outside systems,

robust authentication and logging, in which the application traffic can be preauthenticated before it reaches internal hosts and can be logged more effectively than if logged with standard host logging,

cost-effectiveness, because third-party software or hardware for authentication or logging need be located only at the application gateway, and

less-complex filtering rules, in which the rules at the packet filtering router will be less complex than they would if the router needed to filter application traffic and direct it to a number of specific systems. The router need only allow application traffic destined for the application gateway and reject the rest.

A disadvantage of application gateways is that, in the case of client-server protocols such as TELNET, two steps are required to connect inbound or outbound. Some application gateways require modified clients, which can be viewed as a disadvantage or an advantage, depending on whether the modified clients make it easier to use the firewall.

CIRCUIT-LEVEL GATEWAYS

A circuit-level gateway relays TCP connections but does no extra processing or filtering of the protocol. For example, the TELNET application gateway example provided here would be an example of a circuit-level gateway, since once the connection between the source and destination is established, the firewall simply passes bytes between the systems. Another example of a circuit-level gateway would be for NNTP, in which the NNTP server would

connect to the firewall, and then internal systems' NNTP clients would connect to the firewall. The firewall would, again, simply pass bytes.

Firewall Architectures

Firewalls can be configured in a number of different architectures, provided various levels their risk profile to the type of firewall architecture selected. The following sections describe typical firewall architectures and sample policy statements.

Multi-homed host

A multi-homed host is a host (a firewall in this case) that has more than one network interface, with each interface connected to logically and physically separate network segments. A dual-homed host (host with two interfaces) is the most common instance of a multi-homed host.

A dual-homed firewall is a firewall with two network interfaces cards (NICs) with each interface connected to a different network. For instance, one network interface is typically connected to the external or untrusted network, while the other interface is connected to the internal or trusted network. In this configuration, an important security tenet is not to allow traffic coming in from the untrusted network to be directly routed to the trusted network - the firewall must always act as an intermediary.

Routing by the firewall shall be disabled for a dual-homed firewall so that IP packets from one network are not directly routed from one network to the other.

Screened host

A screened host firewall architecture uses a host (called a bastion host) to which all outside hosts connect, rather than allow direct connection to other, less secure internal hosts. To achieve this, a filtering router is configured so that all connections to the internal network from the outside network are directed towards the bastion host.

If a packet-filtering gateway is to be deployed, then a bastion host should be set up so that all connections from the outside network go through the bastion host to prevent direct Internet connection between the ORGANIZATION network and the outside world.

Screened subnet

The screened subnet architecture is essentially the same as the screened host architecture, but adds an extra strata of security by creating a network which the bastion host resides (often called a perimeter network) which is separated from the internal network. A screened subnet will be deployed by adding a perimeter network in order to separate the internal network from the external. This assures that if there is a successful attack on the bastion host, the attacker is restricted to the perimeter network by the screening router that is connected between the internal and perimeter network.

Types of Firewalls

There are different implementations of firewalls, which can be arranged in different ways.

The various firewall implementations are discussed below.

Packet Filtering Gateways

Packet filtering firewalls use routers with packet filtering rules to grant or deny access based on source address, destination address and port. They offer minimum security but at a very low cost, and can be an appropriate choice for a low risk environment. They are

fast, flexible, and transparent. Filtering rules are not often easily maintained on a router, but there are tools available to simplify the tasks of creating and maintaining the rules. Filtering gateways do have inherent risks including:
The source and destination addresses and ports contained in the IP packet header are the only information that is available to the router in making decision whether or not to permit traffic access to an internal network.

They don't protect against IP or DNS address spoofing.

An attacker will have a direct access to any host on the internal network once access has been granted by the firewall.

Strong user authentication isn't supported with some packet filtering gateways.

They provide little or no useful logging.

Application Gateways
An application gateway uses server programs (called proxies) that run on the firewall. These proxies take external requests, examine them, and forward legitimate requests to the internal host that provides the appropriate service. Application gateways can support functions such as user authentication and logging.

Because an application gateway is considered as the most secure type of firewall, this configuration provides a number of advantages to the medium-high risk site:

The firewall can be configured as the only host address that is visible to the outside network, requiring all connections to and from the internal network to go through the firewall.

The use of proxies for different services prevents direct access to services on the internal network, protecting the enterprise against insecure or misconfigured internal hosts.

Strong user authentication can be enforced with application gateways.

Proxies can provide detailed logging at the application level.

Hybrid or Complex Gateways
Hybrid gateways combine two or more of the above firewall types and implement them in series rather than in parallel. If they are connected in series, then the overall security is enhanced; on the other hand, if they are connected in parallel, then the network security perimeter will be only as secure as the least secure of all methods used. In medium to high-risk environments, a hybrid gateway may be the ideal firewall implementation.