CS-3103 : Operating Systems : Sec-A (NB) :

Protection & Security…..

Protection

- A computer system is a collection of processes and objects. By ***objects***, we mean both ***hardware objects*** (such as, CPU, memory segments, printers, disks, tape drives) and ***software objects*** (such as files, programs, semaphores).
- Each object has a unique name that differentiates it from all other objects in the system, and each can be accessed only through well-defined and meaningful operations.
- The operations that are possible may depend on the object. Example: on a CPU, we can only execute. Memory segments can be read and written, whereas a CD-ROM or DVD-ROM can only be read. Tape drives can be read, written, and rewound. Data files can be created, opened, read, written, closed, and deleted; program files can be read, written, executed, and deleted.

Goals of Protection

- Obviously to prevent malicious misuse of the system by users or programs. See chapter 15 for a more thorough coverage of this goal.
- To ensure that each shared resource is used only in accordance with system policies, which may be set either by system designers or by system administrators.
- To ensure that errant programs cause the minimal amount of damage possible.
- Note that protection systems only provide the mechanisms for enforcing policies and ensuring reliable systems. It is up to administrators and users to implement those mechanisms effectively.

Principles of Protection

- The principle of least privilege dictates that programs, users, and systems be given just enough privileges to perform their tasks.
- This ensures that failures do the least amount of harm and allow the least of harm to be done.
- For example, if a program needs special privileges to perform a task, it is better to make it a SGID program with group ownership of "network" or "backup" or some other pseudo group, rather than SUID with root ownership. This limits the amount of damage that can occur if something goes wrong.
- Typically each user is given their own account, and has only enough privilege to modify their own files.
- The root account should not be used for normal day to day activities - The System Administrator should also have an ordinary account, and reserve use of the root account for only those tasks which need the root privileges

The Security Problem

- The Security Problem
- Authentication
- Program Threats
- System Threats
- Threat Monitoring
- Encryption

- Security must consider external environment of the system, and protect it from:
  - unauthorized access.
  - malicious modification or destruction
  - accidental introduction of inconsistency.
- Easier to protect against accidental than malicious misuse.

What does Operating System Security (OS Security) mean?

- ***Operating system security (OS security) is the process of ensuring OS integrity, confidentiality and availability.*** OS security refers to specified steps or measures used to protect the OS from threats, viruses, worms, malware or remote hacker intrusions. OS security encompasses all preventive-control techniques, which safeguard any computer assets capable of being stolen, edited or deleted if OS security is compromised.

- **What is operating system security?**
- **How do operating systems contribute to system security?**
- **Alternatively, if we're trying to develop a secure system, what do we demand from the OS?**

What is Security?

- Informal: *Security is keeping unauthorized entities from doing things you don't want them to do.*
- More formal: Confidentiality, integrity, availability
- What is the operating system's role?

Authentication

- User identity most often established through *passwords*, can be considered a special case of either keys or capabilities.
- Passwords must be kept secret.
  - Frequent change of passwords.
  - Use of “non-guessable” passwords.
  - Log all invalid access attempts.

Program Threats

- Trojan Horse
  - Code segment that misuses its environment.
  - Exploits mechanisms for allowing programs written by users to be executed by other users.
- Trap Door
  - Specific user identifier or password that circumvents normal security procedures.
  - Could be included in a compiler.

Program Threats

Operating system's processes and kernel do the designated task as instructed. If a user program made these process do malicious tasks, then it is known as **Program Threats**. One of the common example of program threat is a program installed in a computer which can store and send user credentials via network to some hacker. Following is the list of some well-known program threats.

- **Trojan Horse** – Such program traps user login credentials and stores them to send to malicious user who can later on login to computer and can access system resources.
- **Trap Door** – If a program which is designed to work as required, have a security hole in its code and perform illegal action without knowledge of user then it is called to have a trap door.
- **Logic Bomb** – Logic bomb is a situation when a program misbehaves only when certain conditions met otherwise it works as a genuine program. It is harder to detect.
- **Virus** – Virus as name suggest can replicate themselves on computer system. They are highly dangerous and can modify/delete user files, crash systems. A virus is generatlly a small code embedded in a program. As user accesses the program, the virus starts getting embedded in other files/ programs and can make system unusable for user

System Threats

- Worms – use spawn mechanism; standalone program
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs.
  - Grappling hook program uploaded main worm program.
- Viruses – fragment of code embedded in a legitimate program.
  - Mainly effect microcomputer systems.
  - Downloading viral programs from public bulletin boards or exchanging floppy disks containing an infection.
  - *Safe computing*.

System Threats

System threats refers to misuse of system services and network connections to put user in trouble. System threats can be used to launch program threats on a complete network called as program attack. System threats creates such an environment that operating system resources/ user files are misused. Following is the list of some well-known system threats.

- **Worm** – Worm is a process which can choked down a system performance by using system resources to extreme levels. A Worm process generates its multiple copies where each copy uses system resources, prevents all other processes to get required resources. Worms processes can even shut down an entire network.

- **Port Scanning** – Port scanning is a mechanism or means by which a hacker can detects system vulnerabilities to make an attack on the system.

- **Denial of Service** – Denial of service attacks normally prevents user to make legitimate use of the system. For example, a user may not be able to use internet if denial of service attacks browser's content settings.

Something You Know: Passwords

- Very common
- Very easily guessed
- Originally stored in plaintext, but that's a very bad idea
- Today, passwords are usually stored hashed
- However — some network authentication schemes, such as challenge/response, require plaintext (or equivalent)

One Time passwords

One-time passwords provide additional security along with normal authentication. In One-Time Password system, a unique password is required every time user tries to login into the system. Once a one-time password is used, then it cannot be used again. One-time password are implemented in various ways.

- **Random numbers** – Users are provided cards having numbers printed along with corresponding alphabets. System asks for numbers corresponding to few alphabets randomly chosen.
- **Secret key** – User are provided a hardware device which can create a secret id mapped with user id. System asks for such secret id which is to be generated every time prior to login.
- **Network password** – Some commercial applications send one-time passwords to user on registered mobile/ email which is required to be entered prior to login.

Possible Questions & Answers

A password may become known to other users in a variety of ways. Is there a simple method for detecting that such an event has occurred? Explain your answer.

Ans. : One method is to set the system up so that whenever a user logs in, the system prints out the date and time the user was logged on to the system. The user can then verify whether the last log in time looks right, and change his or her password and take other steps if the last log in time looks suspicious.

Authentication

Why doesnt $D(k_d, N)(E(k_e, N)(m))$ provide authentication of the sender? To what uses can such an encryption be put?

$D(k_d, N)(E(k_e, N)(m))$ means that the message is encrypted using the public key and then decrypted using the private key. This scheme is not sufficient to guarantee authentication since any entity can obtain the public keys and therefore could have fabricated the message. However, the only entity that can decrypt the message is the entity that owns the private key, which guarantees that the message is a secret message from the sender to the entity owning the private key; no other entity can decrypt the contents of the message.

What commonly used computer programs are prone to man-in-the-middle attacks? Discuss solutions for preventing this form of attack.

Any protocol that requires a sender and a receiver to agree on a session key before they start communicating is prone to the man-in- the-middle attack. For instance, if one were to implement on a secure shell protocol by having the two communicating machines to identify a common session key, and if the protocol messages for exchanging the session key is not protected by the appropriate authentication mechanism, then it is possible for an attacker to manufacture a separate session key and get access to the data being communicated between the two parties.
In particular, if the server is supposed to manufacture the session key, the attacker could obtain the session key from the server, communicate its locally manufactured session key to the client, and thereby convince the client to use the fake session key.
When the attacker receives the data from the client, it can decrypt the data, re-encrypt it with the original key from the server, and transmit the encrypted data to the server without alerting either the client or the server about the attacker's presence.
Solution to prevent man-in-the-middle attack: Such attacks could be avoided by using ***digital signatures*** to authenticate messages from the server. If the server could communicate the session key and its identity in a message that is guarded by a digital signature granted by a certifying authority, then the attacker would not be able to forge a session key, and therefore the man-in-the-middle attack could be avoided.

Possible Questions & Answers

What are two advantages of encrypting data stored in the computer system?

Answer: 1) Encrypted data are guarded by the operating system's protection facilities, as well as a password that is needed to decrypt them. 2) Two keys are better than one when it comes to security.

Discuss how the asymmetric encryption algorithm can be used to achieve the following goals.

i) Authentication: the receiver knows that only the sender could have generated the message.

ii) Secrecy: only the receiver can decrypt the message.

iii) Authentication and secrecy: only the receiver can decrypt the message, and the receiver knows that only the sender could have generated the message.

Answer: Let k_e^s be the public key of the sender, k_e^r be the public key of the receiver, k_d^s be the private key of the sender, and k_e^s be the private key of the receiver. Authentication is performed by having the sender send a message that is encoded using k_d^s. Secrecy is ensured by having the sender encode the message using k_e^r. Both authentication and secrecy are guaranteed by performing double encryption using both k_d^s and k_e^r.

Possible Questions & Answers

Differentiate between threat and attack.

"A threat is a category of objects, persons, or other entities that represents a constant danger to an asset".

"An attack is an act or event that exploits vulnerability".

Main difference between threat and attack is a threat can be either intentional or unintentional where as an attack is intentional. Threat is a circumstance that has potential to cause loss or damage whereas attack is attempted to cause damage. Threat to the information system doesn't mean information was altered or damaged but attack on the information system means there might be chance to alter, damage, or obtain information when attack was successful.

What do you mean by theft of service and denial of service?

Theft of service: 1) Application Level Attack; 2) Attacker Gains Increased Access To Restricted or Limited Resources; 3) Opportunistic Attack; 4) Typically does not result in system administration access. **TOS** means a violation that involves unauthorized use of resources. For example, an intruder (or intrusion program) may install a daemon on a system that acts as a file server.

Denial of Service: A denial of service (DoS) attack is an incident in which a user or organization is deprived of the services of a resource they would normally expect to have. In a distributed denial-of-service, large numbers of compromised systems (sometimes called a botnet) attack a single target.

Possible Questions & Answers

- Explain the term “breach of confidentiality” and “breach of availability”.

“Breach of confidentiality”: This type of violation involves unauthorized reading of data (or theft of information). Typically, a breach of confidentiality is the goal of an intruder. Capturing secret data from a system or a data stream, such as credit-card information or identity information for identity theft, can result directly in money for the intruder.

“Breach of availability”: This violation involves unauthorized destruction of data. Some crackers would rather wreak havoc and gain status or bragging rights than gain financially. Website defacement is a common example of this type of security breach.

Explain masquerading and replay attack.

Masquerading and Replay attack: Attackers use several standard methods in their attempts to breach security. The most common is **masquerading**, in which one participant in a communication pretends to be someone else (another host or another person). By masquerading, attackers breach **authentication**, the correctness of identification; they can then gain access that they would not normally be allowed or escalate their privileges—obtain privileges to which they would not normally be entitled. Another common attack is to replay a captured exchange of data. A **replay attack** consists of the malicious or fraudulent repeat of a valid data transmission. Sometimes the replay comprises the entire attack—for example, in a repeat of a request to transfer money. But frequently it is done along with **message modification**, again to escalate privileges. Consider the damage that could be done if a request for authentication had a legitimate user’s information replaced with an unauthorized user’s.

Possible Questions & Answers

Discuss how one-time password scheme is implemented

To avoid the problems of password sniffing and shoulder surfing, a system can use a set of paired passwords. When a session begins, the system randomly selects and presents one part of a password pair; the user must supply the other part. In this system, the user is challenged and must respond with the correct answer to that challenge.

This approach can be generalized to the use of an algorithm as a password. Such algorithmic passwords are not susceptible to reuse. That is, a user can type in a password, and no entity intercepting that password will be able to reuse it. In this scheme, the system and the user share a symmetric password. The password pw is never transmitted over a medium that allows exposure. Rather, the password is used as input to the function, along with a challenge ch presented by the system. The user then computes the function $H(pw, ch)$. The result of this function is transmitted as the authenticator to the computer. Because the computer also knows pw and ch, it can perform the same computation. If the results match, the user is authenticated. The next time the user needs to be authenticated, another ch is generated, and the same steps ensue. This time, the authenticator is different. This one-time password system is one of only a few ways to prevent improper authentication due to password exposure.

One-time password scheme

One-time password systems are implemented in various ways. Commercial implementations use hardware calculators with a display or a display and numeric keypad. These calculators generally take the shape of a credit card, a key-chain dongle, or a USB device. Software running on computers or smartphones provides the user with $H(pw, ch)$; pw can be input by the user or generated by the calculator in synchronization with the computer. Sometimes, pw is just a personal identification number (PIN). The output of any of these systems shows the one-time password. A one-time password generator that requires input by the user involves two-factor authentication. Two different types of components are needed in this case—for example, a one-time password generator that generates the correct response only if the PIN is valid. Two-factor authentication offers far better authentication protection than single-factor authentication because it requires "something you have" as well as "something you know."

Another variation on one-time passwords uses a code book, or one-time pad, which is a list of single-use passwords. Each password on the list is used once and then is crossed out or erased. The commonly used S/Key system uses either a software calculator or a code book based on these calculations as a source of one-time passwords. Of course, the user must protect his code book, and it is helpful if the code book does not identify the system to which the codes are authenticators.

- https://searchsecurity.techtarget.com/definition/asymmetric-cryptography