

Security Management

Unit 8

Security

Security refers to providing a protection system to computer system resources such as CPU, memory, disk, software programs and most importantly data/information stored in the computer system. If a computer program is run by an unauthorized user, then he/she may cause severe damage to computer or data stored in it. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc.

The goal of Security System

- There are several goals of system security. Some of them are as follows:

1. Integrity

- Unauthorized users must not be allowed to access the system's objects, and users with insufficient rights should not modify the system's critical files and resources.

2. Secrecy

- The system's objects must only be available to a small number of authorized users. The system files should not be accessible to everyone.

3. Availability

- All system resources must be accessible to all authorized users, i.e., no single user/process should be able to consume all system resources. If such a situation arises, service denial may occur. In this case, malware may restrict system resources and preventing legitimate processes from accessing them.

Security

Security Problems

- protecting systems from deliberate attacks, either internal or external, from individuals intentionally attempting to steal information, damage information, or otherwise deliberately wreak havoc in some manner.

Some of the most common types of violations include:

- **Breach of Confidentiality** - Theft of private or confidential information, such as credit-card numbers, trade secrets, patents, secret formulas, manufacturing procedures, medical information, financial information, etc.
- **Breach of Integrity** - Unauthorized modification of data, which may have serious indirect consequences. For example a popular game or other program's source code could be modified to open up security holes on users systems before being released to the public.
- **Breach of Availability** - Unauthorized destruction of data, often just for the "fun" of causing havoc and for bragging rites. Vandalism of web sites is a common form of this violation.
- **Theft of Service** - Unauthorized use of resources, such as theft of CPU cycles, installation of daemons running an unauthorized file server, or tapping into the target's telephone or networking services.
- **Denial of Service, DoS** - Preventing legitimate users from using the system, often by overloading and overwhelming the system with an excess of requests for service.

Security

- There are four levels at which a system must be protected:
- **Physical** - The easiest way to steal data is to pocket the backup tapes. Also, access to the root console will often give the user special privileges, such as rebooting the system as root from removable media. Even general access to terminals in a computer room offers some opportunities for an attacker, although today's modern high-speed networking environment provides more and more opportunities for remote attacks.
- **Human** - There is some concern that the humans who are allowed access to a system be trustworthy, and that they cannot be coerced into breaching security. However more and more attacks today are made via **social engineering**, which basically means fooling trustworthy people into accidentally breaching security.
 - **Phishing** involves sending an innocent-looking e-mail or web site designed to fool people into revealing confidential information. E.g. spam e-mails pretending to be from e-Bay, PayPal, or any of a number of banks or credit-card companies.
 - **Dumpster Diving** involves searching the trash or other locations for passwords that are written down. (Note: Passwords that are too hard to remember, or which must be changed frequently are more likely to be written down somewhere close to the user's station.)
 - **Password Cracking** involves divining users passwords, either by watching them type in their passwords, knowing something about them like their pet's names, or simply trying all words in common dictionaries. (Note: "Good" passwords should involve a minimum number of characters, include non-alphabetical characters, and not appear in any dictionary (in any language), and should be changed frequently. Note also that it is proper etiquette to look away from the keyboard while someone else is entering their password.)

Security

- **Operating System** - The OS must protect itself from security breaches, such as runaway processes (denial of service), memory-access violations, stack overflow violations, the launching of programs with excessive privileges, and many others.
- **Network** - As network communications become ever more important and pervasive in modern computing environments, it becomes ever more important to protect this area of the system. (Both protecting the network itself from attack, and protecting the local system from attacks coming in through the network.) This is a growing area of concern as wireless communications and portable devices become more and more prevalent.

User Authentication

- Protection, dealt with making sure that only certain users were allowed to perform certain tasks, i.e. that a users privileges were dependent on his or her identity. But how does one verify that identity to begin with?

Passwords

- Passwords are the most common form of user authentication. If the user is in possession of the correct password, then they are considered to have identified themselves.
- Separate passwords could be implemented for separate activities, such as reading this file, writing that file, etc. In practice most systems use one password to confirm user identity, and then authorization is based upon that identification.

Password Vulnerabilities

- Passwords can be guessed.
 - Intelligent guessing requires knowing something about the intended target in specific, or about people and commonly used passwords in general.
 - Brute-force guessing involves trying every word in the dictionary, or every valid combination of characters. For this reason good passwords should not be in any dictionary (in any language), should be reasonably lengthy, and should use the full range of allowable characters by including upper and lower case characters, numbers, and special symbols.
- "Shoulder surfing" involves looking over people's shoulders while they are typing in their password.
 - Even if the lurker does not get the entire password, they may get enough clues to narrow it down, especially if they watch on repeated occasions.

- Passwords echoed as stars or dots still give clues, because an observer can determine how many characters are in the password.
- "Packet sniffing" involves putting a monitor on a network connection and reading data contained in those packets.
 - However you should still never e-mail a password, particularly not with the word "password" in the same message or worse yet the subject header.
- Long hard to remember passwords are often written down, particularly if they are used seldomly or must be changed frequently. Hence
- Passwords can be given away to friends or co-workers, destroying the integrity of the entire user-identification system.
- Most systems have configurable parameters controlling password generation and what constitutes acceptable passwords.
 - They may be user chosen or machine generated.
 - They may have minimum and/or maximum length requirements.
 - They may need to be changed with a given frequency. (In extreme cases for every session.)
 - A variable length history can prevent repeating passwords.
 - More or less stringent checks can be made against password dictionaries.

Encrypted Passwords

- Modern systems do not store passwords in clear-text form, and hence there is no mechanism to look up an existing password.
- Rather they are encrypted and stored in that form. When a user enters their password, that too is encrypted, and if the encrypted version match, then user authentication passes.
- The encryption scheme was once considered safe enough that the encrypted versions were stored in the publicly readable file `"etc/passwd"`.
 - They always encrypted to a 13 character string, so an account could be disabled by putting a string of any other length into the password field.
 - Modern computers can try every possible password combination in a reasonably short time, so now the encrypted passwords are stored in files that are only readable by the super user. Any password-related programs run as `setuid root` to get access to these files. (`/etc/shadow`)
 - A random seed is included as part of the password generation process, and stored as part of the encrypted password. This ensures that if two accounts have the same plain-text password that they will not have the same encrypted password. However cutting and pasting encrypted passwords from one account to another will give them the same plain-text passwords.

One-Time Passwords

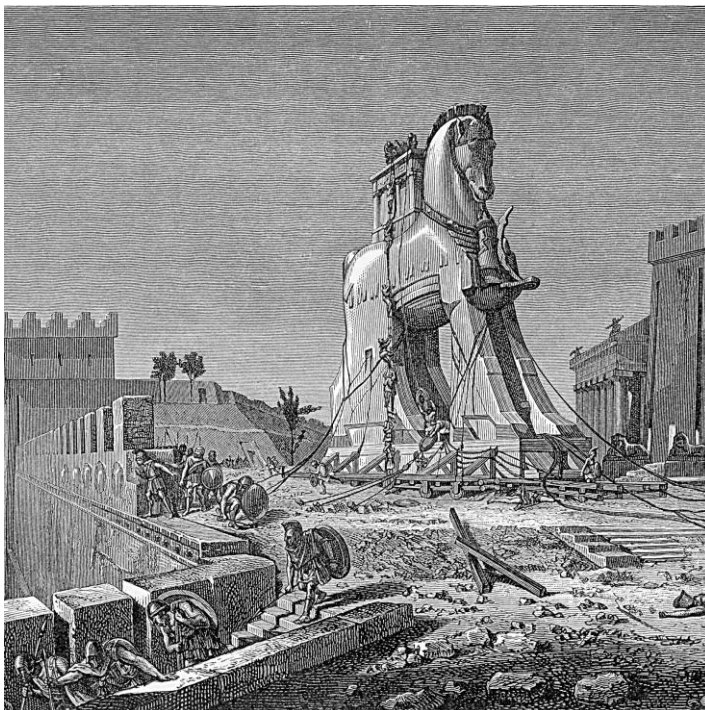
- One-time passwords resist shoulder surfing and other attacks where an observer is able to capture a password typed in by a user.
 - These are often based on a **challenge** and a **response**. Because the challenge is different each time, the old response will not be valid for future challenges.
 - For example, The user may be in possession of a secret function $f(x)$. The system challenges with some given value for x , and the user responds with $f(x)$, which the system can then verify. Since the challenger gives a different (random) x each time, the answer is constantly changing.
 - A variation uses a map (e.g. a road map) as the key. Today's question might be "On what corner is SEO located?", and tomorrow's question might be "How far is it from Navy Pier to Wrigley Field?" Obviously "Taylor and Morgan" would not be accepted as a valid answer for the second question!
 - Another option is to have some sort of electronic card with a series of constantly changing numbers, based on the current time. The user enters the current number on the card, which will only be valid for a few seconds. A **two-factor authorization** also requires a traditional password in addition to the number on the card, so others may not use it if it were ever lost or stolen.
 - A third variation is a **code book**, or **one-time pad**. In this scheme a long list of passwords is generated, and each one is crossed off and cancelled as it is used. Obviously it is important to keep the pad secure.

Biometrics Passwords

- Biometrics involve a physical characteristic of the user that is not easily forged or duplicated and not likely to be identical between multiple users.
 - Fingerprint scanners are getting faster, more accurate, and more economical.
 - Palm readers can check thermal properties, finger length, etc.
 - Retinal scanners examine the back of the users' eyes.
 - Voiceprint analyzers distinguish particular voices.
 - Difficulties may arise in the event of colds, injuries, or other physiological changes.

User Authorization

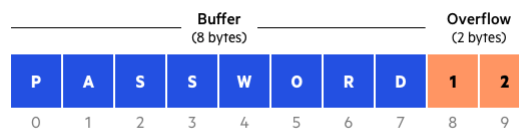
- Authorization is the process of giving someone permission to do or have something. In multi-user computer systems, a system administrator defines for the system which users are allowed access to the system and what privileges of use (such as access to which file directories, hours of access, amount of allocated storage space, and so forth).
- Assuming that someone has logged in to a computer operating system or application, the system or application may want to identify what resources the user can be given during this session. Thus, authorization is sometimes seen as both the preliminary setting up of permissions by a system administrator and the actual checking of the permission values that have been set up when a user is getting access.
- We may assign a user several form of authorization:
 1. Read Authorization
 2. Insert Authorization
 3. Update Authorization
 4. Delete Authorization
- We may assign the user or process, all, none or combination of these types of authorization.
- Authorization deals with the access rights on the file.



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** – They seem to be attractive and harmless cover programs but are really harmful hidden programs that can be used as the virus carrier. In one of the versions of Trojan, the User is fooled to enter confidential login details on an application. Those details are stolen by a login emulator and can be further used as a way of information breaches.
- 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 generally 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
- Buffer Overflow:** Buffers are memory storage regions that temporarily hold data while it is being transferred from one location to another. A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the data to the buffer overwrites adjacent memory locations.

Program Threats



- For example, a buffer for log-in credentials may be designed to expect username and password inputs of 8 bytes, so if a transaction involves an input of 10 bytes (that is, 2 bytes more than expected), the program may write the excess data past the buffer boundary.
- Buffer overflows can affect all types of software. They typically result from malformed inputs or failure to allocate enough space for the buffer. If the transaction overwrites executable code, it can cause the program to behave unpredictably and generate incorrect results, memory access errors, or crashes.

Types of Buffer Overflow Attacks

- Stack-based buffer overflows** are more common, and leverage stack memory that only exists during the execution time of a function.
- Heap-based attacks** are harder to carry out and involve flooding the memory space allocated for a program beyond memory used for current runtime operations.

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.