Assignment - PSO 1

Purdue CS426 - Computer Security - Prof. Spafford

Harris Christiansen - January 24, 2018
christih@purdue.edu

Problem 1

Read the Wikipedia description of the Parkerian Hexad

a. What properties does the Parkerian Hexad add to the CIA model of information security? (5 pts)

- The Parkerian Hexad adds the following 3 elements to the CIA security properties:
    - **Possession/Control:** To have the object/data in your possession, even if you cannot access/read it.
    - **Authenticity:** To know that an object/data is what it claims to be (is from the correct person, and has not been altered)
    - **Utility:** To have the ability to access/read/use an object/data, even if you do not have it in your possession.

b. For the items below, what property/properties (could be more than one per situation) of information security are compromised using the Parkerian Hexad? (25 pts.)

1. **A thief has stolen an encrypted hard drive of medical records, but does not have the password or ability to decrypt the drive.**
    - Possession is compromised, as a thief has obtained possession of the data. Utility is **not** compromised, as they do not have the ability to decrypt it.

2. **An ex-employee planted a logic bomb in a company's main server. It will change all user passwords to a value only known to him 1 month after his last day with the company.**
    - Integrity is compromised, as unauthorized modifications have been made to the company’s main server.

- Possession may become compromised, if his logic bomb successfully executes.

3. **An attacker creates a phishing website that looks identical to a major banking website. He was able to get a domain that looks similar to the bank's legitimate domain. He has been sending legitimate looking emails to customers telling them that their checking account is overdrawn, but substituted the log in link with a page on his site. User's have been clicking on the link and unknowingly going to the attackers site. They enter their log in credentials into a form and find that there is a log in error as the attacker saves the credentials while discreetly redirecting them to the original bank site. The unsuspecting user thinks it is just a glitch and tries his same log in password again. This time it works and upon logging into their account they find that their checking account has not been overdrawn. However, two weeks later some users start to notice mysterious payments from their accounts to entities that they do not recognize.**
   - Possession is compromised, as a thief has obtained possession of a users account.

4. **A botnet ring swamps a blog's web servers with requests resulting in a slow down of access for normal users.**
   - Availability is compromised, as an attacker has reduced/eliminated the ability for other users to access the web server's blog.

5. **The private key for a company's web server is obtained by an attacker who has access to sniff packets destined to the server.**
   - Confidentially is compromised, as the company's private key has been leaked.
   - Possession is also compromised, as an attacker has obtained possession of packets destined for the server, and has the ability to read their confidential contents.

Problem 2

Read about <u>backup options.</u>

a. What is a 3-2-1 Backup Strategy? (5 pts.)

- A 3-2-1 backup strategy is a set of recommended rules for backing up files:
- 3: Keep 3 copies of every important file. This means at least one primary copy, and two additional backup copies kept on separate storage devices.
- 2: Store the files on at least 2 different types of storage devices.
- 1: Make sure at least one backup copy is kept off-site. This is incase of catastrophic damage to on-site copies of the data.

b. What are some Pros and Cons of the strategy? Describe how backups are a good, but imperfect solution to disaster recovery? (20 pts.)

- Pros
  - Backups minimize the chance of data-loss by providing a backup copy of the most recent (or sometimes versioned) data, incase of data-loss on the primary storage device (such as hard-drive failure).
  - Some backup services can help increase availability and remote access to your data.
- Cons
  - Restoring from a backup is not an immediate process, as it causes downtime and compromises availability.
  - In the case of corruption or malware, backups may continue to backup the damaged/dangerous files to the backup devices.

c. What are 2 options for offsite locations other than cloud services? (10 pts.)

- An alternative to cloud services is recording backups to physical disks/drives/tapes and shipping those backups to an offsite location. Because this option offers secure, offline, offsite data backup, it is commonly used by banks and governments.
- Options for offsite locations include: other properties you own, friends properties, banks, warehouses, vaults, or buried underground like treasure.

Problem 3

Read Ken Thompson's Paper, Reflections on Trusting Trust

a. What program(s) did he modify and how did he do it? (10 pts)

- The author modified the C compiler for UNIX so that he could introduce a trojan horse into the UNIX login command. This trojan horse added a single known password for which the login command would always return true.
- He made this modification by first adding a pattern match / replace to the compiler that would alter the UNIX login command. He then made a second modification to make the compiler insert these trojan horses into every compiler it compiled. Then, since every version of the compiler is compiled using the previous, the trojan horses would be added to every compiler compiled by that compiler, and so on.

b. What is the point he is making about trusting computer software? (10 pts.)

- Since our machines process so many instructions, it is impractical for us to know for sure what instructions our machines are actually processing. It's possible that the computer software we trust is actually running instructions we would not expect it to run.

Problem 4

Read *at least* Section A of The Protection of Information in Computer Systems

a. List the 5 Functional Levels of Information Protection and write a few sentences in your own words describing each level. (10 pts.)

- **Unprotected Systems:** Systems which have no protections in place for preventing a determined user from accessing all data on a system. This would apply to systems which do not perform permission checks upon access to RAM, file storage, or other processes. An example of such a system is XINU.
- **All-or-nothing Systems:** Systems which provide isolation between users, and may also have a global shared storage space. Such systems would give each user their own space they can store their own files, but nobody else can have access. There may also be a global shared storage space which all users can read/write files on.
- **Controlled Sharing:** Systems which specify on a per file and per user basis who can read/write/execute that file. This requires each file to have a permissions table with an owner and each additional user (or group), as well as their read/write/execute permissions for that file.
- **User-programmed Sharing Controls:** Systems which permit the owner to have custom, specific control over who has access to file on a per case basis. This permits the owner to specify rules such as users can only access this file during specific hours of the day.
- **Putting Strings on Information:** Systems which pass strings or flags with the file from system to system, user to user. These strings are intended to prevent unauthorized sharing or access to files, even after they have been originally accessed. One such implementation is the "locked" flag on files, which prevents a file from being modified, and following the file from device to device.

b. List the 8 Design Principles outlined in the article. Write a few sentences in your own words describing each principle and explain how each one mitigates security flaws in software? (40 pts.)

- **Economy of Mechanism:** Systems and software (particularly protection mechanisms) should have a simple and small design. This is a well known principal (KISS = keep it simple stupid), and has many benefits including that systems are easier to understand, debug, and maintain. This helps mitigate security flaws, as code-inspection, peer review, and testing are easier and more thorough.

- **Fail-safe Defaults:** Protection mechanisms should be designed such that the default case is always to deny the user access. Designing the system this way ensures that only users with permission to access a file/system can do so, whereas mistakes could happen if allow is the default case. This mitigates security flaws by having mistakes in the code result in denying permission to something, rather than incorrectly granting permission.
- **Complete Mediation:** Protection mechanisms should apply to every access to a file/resource, not just some files/resources or some of the time. By applying protection mechanisms globally, it reduces the complexity of keeping track of what will be protected and what isn't, and unifies everything under a single mechanism. This mitigates security flaws by making sure that every single file/resources on the system has correct protection mechanisms in place. Without it, a file/resource that needs protection might accidentally/inadvertently get created/exposed where there is no protection.
- **Open Design:** Design of protection mechanisms should be open/public. By having a public design, security can be verified, and critical flaws can be quickly caught and fixed. Keeping the design secret is not a method of protection. An open design mitigates security flaws by making sure they are discovered and fixed as quickly as possible.
- **Separation of Privilege:** Protection mechanisms that use more than one key to unlock are more secure than those which only use one. By requiring two or more key's to gain access, you decrease the likelihood of an attacker gaining access should he obtain one of the keys. It is even better if you can physically separate the keys, making it harder for an attacker to obtain all the keys. Having more than one key mitigates security flaws by making it harder for an attacker to gain unauthorized access.
- **Least Privilege:** Every user and program of a system should be given the least amount of privileges necessary to do their task. For instance, programs should only be given access to the files necessary to run their program or store user information. Giving minimal privileges mitigates security flaws by ensuring nobody can access anything they shouldn't be able to, or have no reason to.
- **Least Common Mechanism:** Every user/program/mechanism should communicate with the fewest number of other users/programs/mechanisms possible. Each communication path between mechanisms is a potential vulnerability for a malicious mechanism to exploit. Minimizing the number of connections mitigates security flaws by containing private data and making sure it cannot be shared.

- **Psychological Acceptability:** The user interface for users interacting with these protection mechanisms should be designed to enforce the rules of the mechanisms. Where possible, interfaces should be designed such that users do not attempt to perform requests that will result in denied permissions. This mitigates security flaws by keeping users accessing only what they should be accessing.

Problem 5

Assume you can brute force 2^{48} passwords per second. Answer **how long it would take to brute force all the possible passwords** in each case (Show and explain how you arrived at your answer. Format your answer in: Years Months Days Hours Minutes Seconds): (20 pts.)

Note: All calculations were performed with 1 month = 30.45 days. Calculation code available at: http://files.harrischristiansen.com/p3Co

a. A password can be between 8 to 14 characters long. It only contains lowercase letters (a to z: 26 possibilities)

- 0 Years, 0 Months, 2 Days, 18 Hours, 12 Minutes, 32 Seconds

b. A password can be between 8 to 14 characters long. It can contain - lowercase letters (a to z: 26 possibilities) and uppercase letters (A to Z: 26 possibilities).

- 121 Years, 4 Months, 7 Days, 12 Hours, 28 Minutes, 8 Seconds

c. A password can be between 8 to 14 characters long. It can contain - lowercase letters (a to z: 26 possibilities), uppercase letters (A to Z: 26 possibilities), and numbers (0-9: 10 possibilities).

- 1420 Years, 0 Months, 23 Days, 20 Hours, 19 Minutes, 3 Seconds

d. A password can be between 8 to 14 characters long. It can contain - lowercase letters (a to z: 26 possibilities), uppercase letters (A to Z: 26 possibilities), numbers (0-9: 10 possibilities). Through social engineering you know that the first 4 characters in the password are the user's birth year (e.g. 1990)

- Given that the user was born >= 1900
- 0 Years, 0 Months, 7 Days, 0 Hours, 22 Minutes, 14 Seconds