CS590 DSP: Data Security and Privacy **Duc Viet Le**
Spring 2018

Homework #1

Problem 1 (10 pts) Confidentiality, Integrity, Availability.

- (3 pts) State what is Confidentiality, Integrity, and Availability.
  Answer:

  Confidentiality is about avoiding unauthorized disclosure of information and allowing those who are authorized to have access to the information.

  Integrity is about avoiding unauthorized modification of information and allowing authorized parties to modify in a permitted way.

  Availability is about making sure authorized users have access to data or services.

- (3 pts) For each, give two examples where they are violated.
  Answer:

  Confidentiality Using different analysis techniques to detect hidden communications or deanonymize anonymous identities (i.e. Netflix Deanonymization). Another example is cryptography attacks on encryption scheme like padding oracle attack to decrypt ciphertext of CBC mode encryption.

  Integrity One example is that there is no integrity checking for DNS queries, and it's a reason for DNS poisoning attack (i.e. Kaminsky attack). Similarly, there is no integrity checking for ARP queries in LAN network, and it's the reason for ARP poisoning attack.

  Availability One example is about DDos attacks on website services. Another example is that one can prevent authorized user from accessing the system by trying to login to the system with a wrong password multiple time to trigger its defend mechanism.

- (4 pts) Identify two computer security control measures on your computer(s). Which of the three properties Confidentiality, Integrity, and Availability do they aim at providing? What kinds of adversaries they **cannot** defend against?

  Answer:

Security Control Measures	Confidentiality	Integrity	Availability
SSL/TLS communications	X	X	
Bot detection/CAPTCHA			X

  - SSL/TLS may not be able to defend against adversaries who try to perform an DDOS attack by creating lots of connections.
  - Weak bot detection or CAPTCHA may not provide integrity checking or may not have lots of different challenge;therefore, those mechanisms may be subjects for adversaries who perform replay attacks (i.e. record and reuse same responses for certain challenges)

Problem 2 (10 pts) Unix Access Control.

- (2 pts) Explain why the setuid bit is needed in UNIX DAC?
  Answer. Because some operations are not modele as files. System integrity requires more than controlling who can write but how it's written

- (4 pts) What security problems can be caused by using the setuid bit? What can one do to mitigate the problem?
  Answer. Those program can setuid as root; therefore, if those programs are compromised by attackers.Then, attackers will have root access to the system. The program with setuid should be able to drop its privilege either temporary or permanently. The reason is that the program should only use the minimum privilege to perform its task to mitigate the risk of being compromised by attackers.
- (4 pts) Explain how the sticky bit on directories affect UNIX file access control, and why it is needed.
  Answer. If the sticky bit is set then only the directory's owner or root can rename or delete directories. It's needed and normally set for those directories shared by others (e.g \tmp) to prevent users with write and execution permissions from deleting or moving other users' files.

Problem 3 (15 pts) More Unix Access Control. On a UNIX/Linux system, create a directory, that includes sub-directories the following. Submit a printout of running "ls -ailR" on this directory. You can copy/paste the printout into your homework.

- A sub-directory named "dir1" such that any other user on the system can create/delete any files under the directory, but cannot do a listing of the file names in the directory.
  Answer.

```
3148535 drwxrwxr-x 4 dduck dduck  4096 Jan 30 23:46 .
3148533 drwxrwxr-x 3 dduck dduck  4096 Jan 30 12:36 ..
3148540 drwx-wx-wx 2 dduck dduck  4096 Jan 30 23:44 dir1
```

- A sub-directory named "dir2" such that any other user on the system can create files in the directory, can do a listing of the file names in the directory, but can delete only the files owned by the user.
  Answer.

```
3148535 drwxrwxr-x 4 dduck dduck  4096 Jan 30 23:53 .
3148533 drwxrwxr-x 3 dduck dduck  4096 Jan 30 12:36 ..
3148540 drwx-wx-wx 2 dduck dduck  4096 Jan 30 23:44 dir1
3278614 drwxrwxrwt 2 dduck dduck  4096 Jan 30 23:45 dir2
```

- A sub-directory named "dir3" such that any other user can see the file names under the directory, but not access any of the file.
  Answer.

```
3148535 drwxrwxr-x 5 dduck dduck  4096 Jan 31 00:30 .
3148533 drwxrwxr-x 3 dduck dduck  4096 Jan 30 12:36 ..
3148540 drwx-wx-wx 2 dduck dduck  4096 Jan 31 00:13 dir1
3278614 drwxrwxrwt 2 dduck dduck  4096 Jan 31 00:08 dir2
3278618 drwxr-xr-- 2 dduck dduck  4096 Jan 31 00:30 dir3
```

- Create an executable file with name "test" under "dir1" such that the setuid bit on the file is set.
  Answer.

```
./dir1:
total 96
3148540 drwx-wx-wx 2 dduck dduck  4096 Jan 31 00:13 .
3148535 drwxrwxr-x 5 dduck dduck  4096 Jan 31 00:34 ..
3148583 -rwsrwxr-x 1 dduck dduck  8608 Jan 31 00:25 test
```

- Create a hard link with name "test" under "dir2" to the file dir1/test.
  Answer.

```
./dir2:
total 24
3278614 drwxrwxrwt 2 dduck dduck 4096 Jan 31 00:38 .
3148535 drwxrwxr-x 5 dduck dduck 4096 Jan 31 00:36 ..
3148583 -rwsrwxr-x 2 dduck dduck 8608 Jan 31 00:25 test
```

- Create a symbolic link with name "test" under "dir3", and make it point to the file dir1/test.
  Answer.

```
./dir3:
total 8
3278618 drwxr-xr-- 2 dduck dduck 4096 Jan 31 00:39 .
3148535 drwxrwxr-x 5 dduck dduck 4096 Jan 31 00:36 ..
3278621 lrwxrwxrwx 1 dduck dduck    9 Jan 31 00:39 test -> dir1/test
```

- After submitting the printout, delete either dir1/test, and see how this affect dir2/test and dir3/test. Describe your findings.
  Answer.
  - The hard link test on dir2 is executable and able to print out some output.

    ```
    ./dir2/test
    Hello, World!%
    ```
  - The soft link test on dir3 does not work anymore

    ```
    ./dir3/test
    no such file or directory: ./dir3/test
    ```

Problem 4 (15 pts) Read Granham & Denning: "Protection: Principle and Practice"
https://dl.acm.org/citation.cfm?id=1478928

Answer the following problems. Don't write too long, stay within one page for the answer.

1. Describe three places in computing/information systems where access control is used.
   Answer.

   Fileshare services (i.e. Dropbox, google drive): only user that has re login credential can sign in and see his file while his files are stored in the server. He can decide to create a link with different access capability to certain file and share the link with different user.

   Database Management System (i.e. MySQL, SQL server) : Users are given accesses to different tables in the database to do the work.

   Computer Memory: one example is with intel software guard extensions (SGX), it has its own secure memory where none of processes that run outside SGX can have access to it.

2. Describe one experience that you were frustrated by the existence of (or lack of) access control mechanisms and what you wish the access control would be changed. If you have not frustrated by access control before, think about one place where access control could be improved.
   Answer. Under one project working with the Center for Career Opportunities at Purdue, they don't assign their student workers with accesses to different tables in their database. Instead, they have one account that have read/write access to all tables of CCO Database, and the account's login credential is put in plain in one of the configuration file which is accessible to all students. Thus, without the lack of access control, any student with malicious intent may use that credential to obtain valuable information like SSN, email, and address of its users.
3. Identify one vulnerability/attack so that can be attributed to problems in the access control mechanisms?
   Answer. One example given in class is that when owner with high security clearance run a trojan-horse program The trojan-horse program may use access right of the user to write content of top secret file to lower secret file.

Problem 5 (15 pts) Read Norman Hardy's "The Confused Deputy".

http://zoo.cs.yale.edu/classes/cs422/2010/bib/hardy88confused.pdf

- Explain what is the confused deputy problem.
  Answer. When a program runs with authority from two different sources, the program serves two masters and carries authority for each to perform it duties, and it has no way to keep those two instructions apart. In the paper presented, the confused deputy problem happened when the compiler was given home file license (one authority) to write files in its home directory, and by providing (SYSX)BILL as file name, user (other authority) who runs (SYSX)FORT can overwrite billing file with debug information.
- Explain how capability-based system solve the confused deputy problem.
  Answer. In capability-based system, to perform an access to a file, the program will be given a direct capability to the file. The capability will identifies the file and authorizes the program an access on that file. In this case, the os will know directly what file the compiler should have access to.
- Explain how is the confused deputy problem manifested in setuid root programs in UNIX DAC, and how this problem is addressed in UNIX DAC.
  Answer. If the programs with setuid root are compromised by attackers, the attacker will temporarily gain root access on the system. Therefore, he may get access to sensitive information like /etc/passwrd

  The problem is addressed by allowing program changing its effective userid. Programs with setuid root can drop privilege either temporarily or permanently. The idea is that those programs should use minimum privilege needed to perform task, and this follows the least privilege principle in security.
- Recall the weaknesses exploited by the malwares we have examined (Morris). Are they related to the confused deputy problem?

  Answer. Yes. Morris worm exploited trust on those programs including sendmail, remotelogin, fingerd. Basically, the worm exploits the bug those program and gets those privileges of the user whose behalf the program execute (i.e. root privilege in this case.).

Problem 6 (15 pts) Read the following article

- Ken Thompson's "Reflections on Trusting Trust"
  https://dl.acm.org/citation.cfm?id=358210

Write a brief summary which should include:

- How the attack described in Thompson's article work?
  Answer. Ken Thompson described 3 stages of the attack:
  - Stage 1: the attack uses self-reproducing code which is a program that output an exact copy of its source.
  - Stage 2: Self learning code. The idea is that we can train the compiler a certain code we want it to compile, then remove that code, and the compiler will know what to do with it. In the paper, Ken Thompson gave an example of how to train compiler to know the new escape character `\v`. First, we let the compiler know that 11 is ACII value for `\v`. After, installing this compiler, we can safely remove the training code from the source code with no trace left.
  - Stage 3: With self-reproducing code and self-learning code described above, the next step is to install trojan horse into compiler code. An trivial way to insert trojan horse is to include it in the compiler source. However, this code will be detected by analyzing the source code. Therefore, the idea is to include a second trojan horse that use self-reproducing code to insert both trojan horse into compiler code. This, however, requires a learning phase that described in stage 2. First we compile the modfified source to obtain bugged binary, and this bugged binary will reinsert the bugs whenever it compile. This trojan horse will go undetected unless there is some analysis on the binary.
- How can it be used to compromise the security of real world systems?
  Answer. This attack can be used to compromise the security of real world system. Here, the example given in the paper is about inserting reproducing Trojan horse into the C compiler; however this attack can be done at any level to program (i.e. assembler, a loader...). This includes hardware level firmware, and author mentioned that the lower the level of program, the harder to detect this bug. The author stressed that a well-installed microcode bug will be almost impossible to detect.
- What are the most effective ways to defend against the attack?
  Answer. One way is to never trust the code the you did not write. Another way is to spread the awareness about breaking in computer system. It should have some kind of social stigma just like breaking into other person's house, and it's punishable by law just like other crimes.
- What have you learned?
  Answer. the moral lesson is not to trust code that we did not write despite having access to source code. No amount of source-level verification or scrutiny will protect you from using untrusted code.

Problem 7 (20 pts) Read Sections I and II of the report Bell La Padula: Secure Computer Systems: Unified Exposition and Multics Interpretation. Describe the Bell La Padula model as given in the report.

http://seclab.cs.ucdavis.edu/projects/history/CD/bell76.pdf
Answer. The goal of Bella La Padula Model is about the confidentiality.

The paper first talks about how to model different elements in computer system and access between these elements. The enssential problem in access control is to control access of active entities to a set of passive entities based on security policy. It defines active entities on the computer system as subjects and passive entites are object. An entities can be both subject and object. Next, the paper abstractly define actual access modes in computer system as access attributes. Intuitively, those mode are categorized into execute, read, append, and write access. After the description of how to model computer element and actual access mode, we are able to describe a system state. There are four components of a state: the current access set, the structure imposed on the objects (hierarchy), the access permission, and the level function. The first component is the current access set. The current access set consists of triples (i.e. (subject, object, access-attribute)) representing all current access in system. Next, the second component is about a structure imposed on the objects; this component describe the rule of how parent-child relation used in system. The next component involves access permission. Access permission is represented as access matrix in which each row shows access attributes for that subject. Finally, last component is level function; this component talks about defining security level using clearances and formal categories. Therefore, the model notation for a state is `(current access, access permission matrix,level function, hierarchy)`

Then, with all notational conventions described, the paper describes 3 security properties of the access control system. First, the simple security property (ss-property) requires during a state when reading access is performed by a subject, the security level (clearances and set of categories) of the subject must dominate the must dominates the security level of object. Intuitively, subject with lower security level should not have access to object of higher security level. The second property is *-property. This property states that if a subject has both read access to object 1 and write access to object 2, the level of object 1 is dominated by level of object-2. This is no write down property. However, this policy does not apply on trusted subject. Finally, the last security property is the discretionary security property (i.e. need to know). A state satisfies the ds-property provided every current access is permitted by access permission matrix. In other word, at given point of time, the subject is only given access to information if it's necessary.