CS 3423 Operating Systems

Fall Semester 2019

Prof. Pai H. Chou

Assignment 15

Due Date: Sunday, December 29, 2019, 11:59pm

Scope: Chapter 16 Security
Chapter 17 Protection

## 1. Definitions and Short Answers

1. What are examples of the following types of breach of security?
	1. breach of **confidentiality**
	2. breach of **integrity**
	3. breach of **availability**
	4. theft of **service**
	5. denial of **service**

 a. breach of confidentiality: unauthorized read of data

 b. breach of integrity: unauthorized modification of data

 c. breach of availability: unauthorized destruction of data

 d. theft of service: unauthorized use of resources

 e. denial fo service: prevention of legitimate use

1. What is the meaning of the following
	1. **masquerading**
	2. **replay attack**

a. masquerading: 偽裝, pretending to be an authorized user

b. replay attack: Snooper captures message, "replays" as if from original sender

 ( 竊聽 + 偽裝成原本的 sender 重複送出相同 data )

1. Of the four levels of security, give examples of ways they can be breached
	1. physical security
	2. human security
	3. OS security
	4. network security

 a. physical security: Data centers, servers, connected terminals

 b. human security: Social engineering ( 釣魚網站 ), dumpster diving 翻垃圾箱

 c. OS security: Protection mechanisms, debugging

 d. network security: Intercepted communications, interruption of communication

1. In the movie WarGames, why can the dialup modem to the school's gradebook system be a security problem? What is a better solution if modems are still to be used?
Modem dialup -> Anyone can call the phone to connect the modem

 solution: authenticate, call-back (認證, 回撥)

1. How does a **trojan horse** gain a user's access rights?

 Trojan Horse: Malware that disguises itself as legitimate software

1. Explain how a **trojan mule** can be set up to steal the free Wi-Fi username and password.
Trojan Mule: Fake Login interface

User input username, passward -> incorrect -> goback to true login interface

1. Is **spyware** a kind of trojan horse? What specifically makes it spyware?
Spyware: one kind of Trojan Horse.

sends confidential information back to intruder(hacker)

1. What is a **trap door**? Why is it hard to detect even if you review the entire source code?

 Trap door: specific user ID / password that circumvents nromal security procedures.

 ex: "admin", "admin"

 Trap door may hide in compiler.

1. How is a **logic bomb** a special case of a trap door? What would be an example of why someone would want to put one in?

 Logic boom: special case of back door, conditionally enabled.

 工程師 A 在公司 C 工作，打 code 順便放進 logic boom, 某天被炒魷魚 -> BOOM!!

1. In **stack buffer-overflow** type of attack,
	1. Which C-library functions contain the **loophole** that can be exploited? What is the loophole? What alternative functions should you call instead?
	2. How does the attacker change the **target address** to jump (or return) to?
	3. How does the attacker **inject code** into the attacked system?
	4. How can hardware help prevent this type of attack?

 a. strcpy, gets, sprintf: may cause stack / memory overflow, alternative function: strncpy, fgets, snprintf

 b. Write past arguments on the stack into the return address on stack

 c. padded with NOPs as needed so it overwrites return address with address of this blue code!!!

 d. Adding bit to page table to indicate "non-executable" state

 Disabling stack execution

1. Where are different places where a virus can reside? Specifically
	1. parasitic
	2. boot virus
	3. macro virus
	4. root kit
	5. source code virus
	6. tunning
	7. multipartite

 a. parasitic: file virus

 b. boot virus: boot sector of system, loaded on boot before OS

 c. macro virus: VisualBasic script, Excel macros ??

 d. root kit: Unix backdoor to root, now any virus that attacks OS

 e. source code virus: modifies source code

 f. tunning: installing itself in interrupt handling chain

 g. multipartite: infects multiple parts of system from boot sector to memory and files, hard to detect

1. How do the different types of virus make it even harder to detect?
	1. polymorphic
	2. encrypted
	3. stealth
	4. tunneling
	5. multipartite

 a. polymorphic: changes its own representation

 b. encrypted: virus code itself is encrypted

 c. stealth: modifies system calls

 d. tunneling: interrupt handling chain

 e. multipartite: infects multiple parts of system from boot sector to memory and files, hard to detect.

1. What is the meaning of **secure by default**?

 secure by default: disable most services by default

1. Name two ways that **masquerading** can be done.

 masquerading: Man-in-the-middle, spoofing

1. Explain how **DNS spoofing** can be used in **man-in-the-middle** attack. Let A and B denote the two parties communicating and C is the attacker.
redirects request to attacker's own server

 A <--> C <--> B

 A is attacker's server

1. How does a **denial of service** (DoS) attack work? How does **distributed DoS** (DDoS) make it harder to stop?
Denial of Service: Overwhelm the targeted computer ( send too many request at the same time )

Distributed denial-of-service (DDOS): from multiple sites at once

1. What does the attacker gain by **port scanning**?
Port Scanning: automated send special request to a range of ports

 -> get the answer of service -> detect the OS, version, Network service on system

1. What is the meaning of **ciphertext**?
ciphertext: encrypted messages
2. In **symmetric** encryption vs **asymmetric** encryption,
	1. which one uses the **same key** for both encryption and decryption? Which one uses **different keys**?
	2. if it uses different keys, can they be just any pair of keys or do they need to be chosen together?
	3. which one has higher complexity?

 a. symmetric encryption: use the same key for decryption/encryption, asymmetric: different keys

 b. 猜 together，講義有寫generate together

 c. asymmetric has higher time complexity

1. What is the difference between **block cipher** and **stream cipher**?

 block cipher vs stream cipher:

 block cipher: encrypt data in sepcific-sized blocks. ( 64-bit, 128-bit block )

 Stream cipher: encrypt data as a stream of bits or bytes 當成 1 串來 encrypt 不分塊

1. If you use public/private key encryption for A to send a message to B,
	1. whose key (and which key) is used by A to encrypt the message to be sent to B?
	2. whose key (and which key) is used by B to decrypt the message sent by A?

 a. A use public key to encrypt

 b. B use private key to decrypt

1. How are both asymmetric encryption and symmetric encryption both used to **set up a secure channel** and then for A and B to **communicate** with each other?

 A, B 之間 message 以 symeetric encryption 傳輸

 encrypt the key of symmetric encryption by asymmetric encryption

 send the encrypted key

 use their own private key to decrypt the key of symmetric encryption

1. What is an example of a **domain** in Unix?

 domain: set of access right

 domain in Unix = user-id

1. The access right is represented as <object-name, rights-set>.
	1. What are examples of objects?
	2. What are examples of rights-set?

 objects: software(file, program), hardware(keyboard, CPU, memoru, disk, printer)

 rights-set: power set of {read, write, execute}

1. In Unix, consider the case where user A launches a program that is owned by B.
	1. if the **setuid bit** is **not** set, what domain is the process?
	2. if the **setuid bit** is set, what domain is the process?
	3. Does the passwd program (for changing password) have setuid bit set or not? Why?

 a. setuid bit = off: domain = A

 b. setuid bit = on: domain = B

 c. the passwd program (for changing password) doesn't have setuid bit set

 If it has setuid bit -> user can modify others password.

1. In the access matrix M, such as

	1. What is the proper interpretation of M[D1, F1], which has the value "read"?
	2. What is the proper interpretation of M[D2, D4], which has the value "switch, control"?

 a. domain D1 can read object F1

 b. D2 can switch to D4

 change rights on resource without being the owner of the resource

1. What is a "**limited copy**" access right? How is it different from a general **copy** access right?

 limited copy: copy an access right R\* from Dij to Dkj as R

 Dkj cannot be copied because it is not R\*

 可以從 有 \* 的 copy right, 但沒 \* 的無法 copy

1. Does a domain have to be an owner to **transfer** access right to another domain? Does it retain the right after the transfer?

 Transfer ( 把 right 給別人 自己的沒了, no need to be owner )

 • transfer the access right from Di to Dk

 • the right is removed from Di after transfer

1. If an **access matrix** is implemented as a straightforward global table,
	1. What are the advantages?
	2. What are the disadvantages?

 a. What are the advantages? easy to implement, faster search for access right

 b. What are the disadvantages? Large table -> memory usage is too large, difficult to group object

1. Between **access list** and capability list implementations of the access matrix, compare the following features and explain which one is more or less efficient for what reason.
	1. find the set of access rights for a given domain
	2. revoke a domain's access right to given resource
	3. passing access list by a resource vs. passing capability list by a domain

 a. find the set of access rights for a given domain: capability list wins

 b. revoke a domain's access right to given resource: access list wins

 c. passing access list by a resource vs. passing capability list by a domain:

 access list is faster. capability list( controlled by OS )

1. How can **access list scheme** improve performance for checking a domain's capabilities? What assumption must be made about any changes in the access rights?

 On first access to object: search access list

 • If allowed, OS creates capability and attach to process

1. if **lock key** is used to implement access matrix,
	1. can there be multiple keys working with a given lock?
	2. does a **lock** encode an object or a domain's list of unique bit patterns?
	3. does a **key** encode an object or a domain's unique list of bit patterns?

 a. can there be multiple keys working with a given lock?

 Nope, one lock, one key

 b. does a lock encode an object or a domain's list of unique bit patterns?

 lock is on object

 c. does a key encode an object or a domain's unique list of bit patterns?

 key is on domain.

## 2. Programming Exercise

There is no programming exercise this week, but be sure you are caught up with your project checkpoint.