Security

Security Problem

- Some systems are inviting targets to thieves.
  - Systems containing payroll and other financial data
- OS and hardware have defences.
  - But the users must conform to the intended use.
- A system is **secure** if resources are used and accessed as intended under all circumstances.
- Security violations can be **intentional** or **accidental**
  - Most default protection mechanisms protect us from accidental attacks.

Keywords

- Intruder and cracker
  - For those attempting to breach security
  - The word *hacker* is used in a wrong way.
    - If the hacker is malicious, we call it a *cracker*
- A **threat** is the potential for security violation
  - Such as the discovery of a vulnerability
- An **attack** is the attempt to break the security.

Types

- Masquerading
  - One participant pretens to be someone else.
  - By masquerading, they breach **authentication**.
    - The correctness of identification
  - Thus, they can gain access to what they would not normally access to.

Types

- Replay attack
  - Consists of malicious or fraudalent repeat of a valid data transmission.
  - Sometimes the replay is the entire attack
    - Try to repeat a money transfer
  - Usually done with *message modification*, to escalate privileges.

Types

- MITM (Man in the middle attack)
  - Attacker sits in the data flow of a communication, masquerades as the sender to the receiver and vice versa.
  - In a network comm, it is possible that a **session hijacking** occurs to make a mitm attack.
    - An active comm. session is intercepted.

Protection

- We must take security measures at 4 levels.
- Physical
  - Site or sites containing the computer systems must be physically secured.
  - Machine rooms and terminals, they must be secured.
  - e.g. Never leave your computer unattended. If you must, Win-L.

Protection

- Human
  - Authorization must be done carefully so only appropriate users can have access.
  - However, authorized users can also be tricked.
    - via *social engineering*
  - Phishing
    - A legitimate-looking email or webpage misleads a user to enter its confidential information.
      - We still see it a lot.
  - Dumpster diving
    - General term for attempting to gather information about someone
      - By looking at trash, finding old papers containing password, etc.
      - Mitnick did these.
        - Book recommendation: Ghost in the wires

Protection

- Operating System
  - List of possible breaches is almost endless.
  - A runaway process can make an accidental DoS attack.
  - A stack overflow could allow an unauthorized process to run etc.

Protection

- Network
  - Data circulates over internet, wireless comm, dial up lines, etc.
  - Interception of these data is dangerous
  - DDoS
    - Botnets

Program Threats

- Trojan Horse
  - A code segment that misuses its environment is called a *trojan horse*.
  - It presents itself as a harmless program. It does not replicate itself, but stays on the system.
  - It can monitor actions, let remote control, download malware, disrupt data.

Program Threats

- Spyware
  - A variation of trojan horse.
  - Sometimes accompanies a program that the user chose to install.
  - Usually comes with freeware or shareware programs.
  - Downloads ads to display, create pop-up browser windows, capture information and return it to central site.

- Spyware is a *micro* example of a **macro** problem.
- Usually, a user of an OS does not need to install network daemons.
- Such are downloaded by two mistakes.
    - First, user runs with more privileges than needed (being an admin)
    - Second, OS may allow by default more privileges to user than it needs.

Program Threats

- Trap Door
  - Designer of a program or system might leave a hole in the software that only he/she is capable of using.
  - Office Space
    - A software rounds up numbers and send the half-cent to their accounts.
  - A clever trap door can be included in a compiler.
    - It generates standard object code as well as a trap door, regardless of the source.
    - So, you cannot see this in the source code, because it is in the compiler code.
  - E.g. You can make your own CPython interpreter and add a trap door in it.

Program Threats

- Logic Bomb
  - A program which only runs when certain conditions are met.
  - e.g. A programmer might create a program which checks whether he was employed or not; if not the program will work.

Program Threats

- Stack & Buffer Overflow
  - Here, the attacker is exploits a bug in the program.
  - Usually a case of *poor programming*
  - Attacker sends more data then the program expects.

- Overflow an input field, command line argument, or input buffer until it writes to stack.
- Overwrite the current return address on stack with the addr. of exploit code.
- Write a simple set of code for the next space in the stack that includes the commands that the atacker wishes to execute
    - Spawn a shell
- Result is usually a *root shell* or *privileged command execution*

- e.g. if a web page form expects are user name to be entered into a field
    - attacker can send the username
        - extra chars to overflow the buffer
        - a new return address to load onto the stack
        - the code they want to run
- when the buffer-reading subroutine returns from execution, the return address will be the exploit code and it will be run.

```c
#include <stdio.h>
#define BUFFER_SIZE 256

int main(int argc, char *argv[])
{
    char buffer[BUFFER_SIZE];

    if(argc < 2) return *1;
    else {
        strcpy(buffer,argv[1]);
        return 0;
    }
}
```

- Creates a char array of size BUFFER_SIZE and copies the contents of the parameters given in as command line args.
- As long the size is smaller than BUFFER SIZE, program works properly
- What happens when argv is larger than it?

- At that point, `strcpy()` function will begin copying from `argv[1]` until it encounters a null terminator or until the program crushes.
- Thus, this program suffers from a potential buffer-overflow problem.
    - The copied data overflows the buffer array.
- A solution to this would be to check for the bounds by using `strncpy(buffer, argv[1], sizeof(buffer)-1)` instead of the written code.

Layout of a typical stack frame

- When a function is invoked, variables defined locally to the function, the parameters passed to the function and the *address* to which control returns once the function exists are stored in a **stack frame**.
- A cracker could execute a buffer-overflow attack.
- Goal is to replace the return address in the stack so it now points to where the attacker wants it.
- We can control the overflow and change the return address to show the address of the payload.

Viruses

- A fragment of code embedded in a legitimate program.
- Self-replicating and designed to infect other programs.
- They can modify files and destroy them.
- Generally a problem for Windows because other OS protects executables from being written.
- Virus dropper
    - A program which inserts the virus into the system.
    - Usually a trojan.
- There are many types of viruses; file, boot, source code, encrypted etc.
    - Adds itself to files, boot sector, source codes, can also be encrypted.

System and Networks Threats

- Attack surface
  - Set of ways in which an attacker can try to break into system.
- Worm
  - A process using the **spawn** mechanism to duplicate itself.
  - Spawns copies of itself, uses up system resources and possibly locking up all other processes.
- Since worms can reproduce themselves, they can shut down the entire network.

Worms

- Robert Morris
  - Unleashed a worm program which was connected to the Internet.
  - Quickly spread and brought down the infected machines.
  - It was designed for rapid reproduction and distribution, but some features of UNIX networking environment provided the means to propagate in the system.

- Worm was made up of two programs:
  - A main program
  - A grappling hook
    - Also called a *bootstrap* or *vector*
- `l1.c` consisted of 99 lines of C code.
  - Connects to the machine where it originated and uploaded a copy of the main worm onto the *hooked* system.
  - Main program proceeded to search for other machines to which the infected system could connect easily.

Port scanning

- Not an attack but means for cracker to detect a system's vulnerabilities.
- You check for which ports are listening.
  - For example *sendmail* uses 25.
  - If you see 25 is open, that means sendmail is using it.
  - Maybe they are using an old version with a bug and they did not patch it.
- nmap (wireshark)
  - Open-source program for network exploration
  - When pointed at a target, it will determine what services are running
    - alongside with names and version.

Denial of Service

- Not aimed at gaining information or stealing resources
- Aim is to disrupt legitimate use of system or facility.
- Generally network based.
  - Either fills the CPU
  - Or fills the network so that you cannot work anymore
- DDoS
  - Distributed denial of service
- Launched from multiple sites at once, toward a common target.
  - Usually by using zombies.
  - Computers being controlled from outside.

Cryptography

- Encryption
  - Encryption algorithms enables the sender of messages to encrypt their messages (M) with a key (K).
  - When M is encrypted with K, it becomes C (Ciphertext)
- Decryption
  - C becomes M when decrypted with K or other key.
  - In *symmetric* cryptography, key *K* is the same for both encryption and decryption.
  - In *asymmetric* cryptography, key is a *pair*.

Symmetric Encryption

- Same key is used to encrypt and decrypt.
- Therefore the key *k* must be a secret.
- We share our keys via *key exchange methods*.
    - Such as Diffie-Hellman.
- There are standards put in place by NIST.
    - DES: Data Encryption Standard
    - 64 bit key length, not enough. Later, they created *Triple DES*, where DES is repeated three times.
    - AES: Advanced Encryption Standard
        - 128, 192, 256 bit key lengths.
        - Works on 128-bit blocks.

Symmetric Encryption

- Stream Cipher and Block Cipher
- **Stream cipher**
  - Designed to encrypt and decrypt a *stream* of bites.
  - RC4 is the most common.
    - Used in MS Office, WEP, SSL etc.
  - The input is a *keystream*.
- **Block cipher**
  - Encrypts and decrypts a block.
  - AES is the standard.

Asymmetric Encryption

- Keys for encryption and decryption are different.
- Also known as *public-key encryption*
- When a key is generated, it is a **pair**.
    - Public key and private key
- Public key is known and private key is private.
- RSA is the most known.
- When someone wants to send you a message, they encrypt that message with your public key.
    - It is known to public.
- However, the only way to open that message is by using the private key.
    - Remember, they were a pair.
- So, since you are the only one with the private key, you can decrypt and read the message.

User authentication

- User must prove that he/she is the one that he/she claims to be.
- Based on one or more of the following three things:
  - The user's possession of something
    - Key card, etc
  - User's knowledge of something
    - Password
  - Attribute of a user
    - Fingerprint, retina, signature, etc

Passwords

- Most common form of authentication.
- Vulnerabilities
  - Can be guessed
  - Accidentally exposed
    - Shoulder surfing
  - Sniffed
    - Crackers fake websites or programs

Securing passwords

- If we store passwords as plaintext, they are vulnerable.
  - People can read them.
  - Even if not stolen, they are open.
- We use *hash functions*.
  - Cryptographic hash functions
  - They have no inverse.
  - Instead of storing plaintext, we store them.
- However, there is another problem

- The hash of two same password *p* will have the same hash.
- When the database is stolen, this can be read and will help the attacker to break them.
- So, we need to make sure even the thing we store in db is different for the same password.
- **Salting**
  - We concetanate a random string to the password, take the hash and store it.
  - For every user we will have a different salt
    - Therefore even if their passwords are the same they will have a different hash.
- There are more advanced ways to store them, but this is the general idea.

One time passwords (OTP)

- Generally used alongside text passwords.
- After the text password is entered the user is asked a different OTP.
  - This is generally sent via SMS or email.
  - Good, because now even if your password is stolen, it is not enough.
  - 2FA
    - Multi-factor authentication
- There are several types of OTP.
  - Most common is SMS.
  - You have 60 seconds to enter the password.
  - How does it work?

- Let's imagine you entered the correct password at 1700.
- In the server, there is an algorithm which creates an OTP for a given time stamp.
- They calculate 60 of them for every second.
    - Maybe for every subsecond.
- Then, you receive an OTP.
    - If that OTP is in that range of 60 passcodes, access is granted.
    - If not, it means it is expired.