Computer System Security

UNIT -1

Introduction to Computer Security

- Computer security is the protection of computer systems and information from harm, theft, and unauthorized use. It is the process of preventing and detecting unauthorized use of your computer system.
- Often people confuse computer security with other related terms like information security and cybersecurity. One way to ascertain the similarities and differences among these terms is by asking what is being secured. For example,
- Information security is securing information from unauthorized access, modification & deletion.
- Computer Security means securing a standalone machine by keeping it updated and patched.
- Cybersecurity is defined as protecting computer systems, which communicate over the computer networks.

Components of Computer Security

The components of a computer system that needs to be protected are:

- **Hardware**, the physical part of the computer, like the system memory and disk drive
- **Firmware**, permanent software that is etched into a hardware device's non-volatile memory and is mostly invisible to the user, like code inside a printer.
- **Software,** the programming that offers services, like operating system, word processor, internet browser to the user .

The CIA Triad

- ***Confidentiality*** *is* ensuring that information is available only to the intended audience
- ***Integrity*** *is* protecting information from being modified by unauthorized parties
- ***Availability*** *is* data is available to the users only when needed.
- In simple language, computer security is making sure information and computer components are usable but still protected from people or software that shouldn't access it or modify it.

Computer security threats

Viruses

A computer virus is a malicious program which is loaded into the user's computer without user's knowledge. It replicates itself and infects the files and programs on the user's PC. The ultimate goal of a virus is to ensure that the victim's computer will never be able to operate properly or even at all.

Computer Worm

A computer worm is a software program that can copy itself from one computer to another, without human interaction. The potential risk here is that it will use up your computer hard disk space because a worm can replicate in great volume and with great speed.

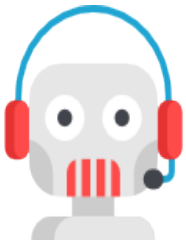
Phishing

Disguising as a trustworthy person or business, phishers attempt to steal sensitive financial or personal information through fraudulent email or instant messages. Phishing in unfortunately very easy to execute. You are deluded into thinking it's the legitimate mail and you may enter your personal information.

Botnet

A botnet is a group of computers connected to the internet, that have been compromised by a hacker using a computer virus. An individual computer is called 'zombie computer'. The result of this

threat is the victim's computer, which is the bot will be used for malicious activities and for a larger scale attack like DDoS.



Rootkit

A rootkit is a computer program designed to provide continued privileged access to a computer while actively hiding its presence. Once a rootkit has been installed, the controller of the rootkit will be able to remotely execute files and change system configurations on the host machine.

Keylogger

Also known as a keystroke logger, keyloggers can track the real-time activity of a user on his computer. It keeps a record of all the keystrokes made by user keyboard. Keylogger is also a very powerful threat to steal people's login credential such as username and password.

Real life goals of computer security

1) Protects the system against viruses, worms, spyware, and other unwanted programs.

2) Protection against data from theft.

3) Protects the computer from being hacked.

4) Minimizes computer freezing and crashes.

5) Gives privacy to users

6) Improved security of cyberspace.

7) Increase in cyber defense.

8) Increase in cyber speed.

9) Protecting company data and information.

10) Protects individual private information.

11) Protects networks and resources.

12) Fight against computer hackers and identity theft.

13) The main advantage that could be achieved with the utilization of these effective cyber-security mechanisms is the protection of networks from various false nodes which try to gain unauthorized access to the network.

14) It identifies the vulnerabilities and weak entities that can aid an attacker to attack the system or the server. These vulnerabilities upon identification can help the teams to secure the systems in order to prevent such attacks from happening.

Cyber Sample Attacks

1.Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks:

- A denial-of-service attack overwhelms a system's resources so that it cannot respond to service requests. A DDoS attack is also an attack on system's resources, but it is launched from a large number of other host machines that are infected by malicious software controlled by the attacker.
- Unlike attacks that are designed to enable the attacker to gain or increase access, denial-of-service doesn't provide direct benefits for attackers. For some of them, it's enough to have the satisfaction of service denial. However, if the attacked resource belongs to a business competitor, then the benefit to the attacker may be real enough. Another purpose of a DoS attack can be to take a system offline so that a different kind of attack can be launched.

2. Man-in-the-middle (MitM) attack A MitM attack occurs when a hacker inserts itself between the communications of a client and a server. Here are some common types of man-in-the-middle attacks:

Session hijacking

- In this type of MitM attack, an attacker hijacks a session between a trusted client and network server. The attacking computer substitutes its IP address for the trusted client while the server continues the session, believing it is communicating with the client. For instance, the attack might unfold like this:
- A client connects to a server.
- The attacker's computer gains control of the client.
- The attacker's computer disconnects the client from the server.
- The attacker's computer replaces the client's IP address with its own IP address and spoofs the client's sequence numbers.

- The attacker's computer continues dialog with the server and the server believes it is still communicating with the client.

IP Spoofing

- IP spoofing is used by an attacker to convince a system that it is communicating with a known, trusted entity and provide the attacker with access to the system. The attacker sends a packet with the IP source address of a known, trusted host instead of its own IP source address to a target host. The target host might accept the packet and act upon it.

Replay

- A replay attack occurs when an attacker intercepts and saves old messages and then tries to send them later, impersonating one of the participants. This type can be easily countered with session timestamps or nonce (a random number or a string that changes with time).

3. Phishing and spear phishing attacks

- Phishing attack is the practice of sending emails that appear to be from trusted sources with the goal of gaining personal information or influencing users to do something. It combines social engineering and technical trickery. It could involve an attachment to an email that loads malware onto your computer. It could also be a link to an illegitimate website that can trick you into downloading malware or handing over your personal information.
- To reduce the risk of being phished, you can use these techniques:
- **Critical thinking** — Do not accept that an email is the real deal just because you're busy or stressed or you have 150 other unread messages in your inbox. Stop for a minute and analyze the email.
- **Hovering over the links** — Move your mouse over the link, but **do not click it**! Just let your mouse cursor h over over the link and see where would actually take you. Apply critical thinking to decipher the URL.
- **Analyzing email headers** — Email headers define how an email got to your address. The "Reply-to" and "Return-Path" parameters should lead to the same domain as is stated in the email.

4.SQL injection attack

SQL injection has become a common issue with database-driven websites. It occurs when a malefactor executes a SQL query to the database via the input data from the client to server. SQL commands are inserted into data-plane input (for example, instead of the login or password) in order to run predefined SQL commands. A successful SQL injection exploit can read sensitive data from the database, modify (insert, update or delete) database data, execute administration operations (such as shutdown) on the database, recover the content of a given file, and, in some cases, issue commands to the operating system

5. Eavesdropping attack

- Eavesdropping attacks occur through the interception of network traffic. By eavesdropping, an attacker can obtain passwords, credit card numbers and other confidential information that a user might be sending over the network. Eavesdropping can be passive or active:
- **Passive eavesdropping** — A hacker detects the information by listening to the message transmission in the network.
- **Active eavesdropping** — A hacker actively grabs the information by disguising himself as friendly unit and by sending queries to transmitters. This is called probing, scanning or tampering.

Control Hijacking

Control hijacking is a way of preventing hijacking attacks.

These are of three types:

1.Buffer overflow attacks

2.Interger overflow attacks

3.Format string attack

A **buffer** is a temporary area for data storage. When more data (than was originally allocated to be stored) gets placed by a program or system process, the extra data overflows. It causes some of that data to leak out into other buffers, which can corrupt or overwrite whatever data they were holding.

In a **buffer-overflow attack,** the extra data sometimes holds specific instructions for actions intended by a hacker or malicious user; for example, the data could trigger a response that damages files, changes data or unveils private information.

- Attacker would use a buffer-overflow exploit to take advantage of a program that is waiting on a user's input. There are two types of buffer overflows: stack-based and heap-based. Heap-based, which are difficult to execute and the least common of the two, attack an application by flooding the memory space reserved for a program. Stack-based buffer overflows, which are more common among attackers, exploit applications and programs by using what is known as a stack: memory space used to store user input.
- Integer overflows occur when the result of an arithmetic operation is a value, that is too large to fit in the available storage space. To clarify the problem, I'll introduce the term *process register*. Process registers represent an amount of storage available in digital processors and its width defines the range of values that can be represented.

Binary register width	Maximum representable value
8 bits	2^8 - 1 = 255

16 bits	2^16 - 1 = 65,535
32 bits	2^32 - 1 = 4,294,967,295
64 bits	2^64 - 1 = 18,446,744,073,709,551,615

Integer overflow Attacks

- The following example helps to clarify what exactly leads to an arithmetic overflow. Let's assume we have three 16-bit unsigned integer values *a*, *b* and *c*. For *a*, the maximum 16-bit representable value 0xffff (hexadecimal value of 65535) is assigned, and for *b* the value of 0x1 (hexadecimal value of 1). If we add *a* and *b* and store the result in *c*, the addition would lead to an arithmetic overflow:

a = 0xffffffff

b = 0x1

r = a + b

The value *0x10000* is too large for a 16-bit binary register, so the addition results in an arithmetic overflow.

Format String attack

- Before we explain the Format String Attack, we need to know what the **format string bug** is.
- Format String bug is the one of the most common vulnerability in programs c.
- Format string is a bug that occurs when format string printf (%d, %s) used in the printf () function is used in the wrong form.
- Vulnerable code: #include<stdio.h>

  int main (int argc,char **argv)

  { printf(argv[1]);

  }
- Safer code: #include<stdio.h>

  int main (int argc,char **argv)

  { printf(“%s”,argv[1]);

  }
- This is because the computer recognizes the input value as a formatting character rather than a character
- Format string attack generates an error when a developer accidentally writes a printf () code without variable, and hacker can use thus error to steal the root.

Two Vulnerabilities used in Format string attack

1.If there is no format string factor after last entered format string, in terms of stack, from the time the printf () function is called, printf () consider in order from the stack top's content as printf ()'s factors.

2.These format string store the number of bytes printed by printf () to int type pointer.

%n store as 4 bytes and %hn store as 2 bytes.

Format string vulnerabilities are a class of bug that take advantage of an easily avoidable programmer error. If the programmer passes an attacker-controlled buffer as an argument to a printf (or any of the related functions, including sprintf, fprintf, etc), the attacker can perform writes to arbitrary memory addresses.

The following program contains such an error:

```
int main (int argc, char** argv)
{
   char buffer [100];
   strncpy(buffer, argv[1], 100);

   // We are passing command line
   // argument to printf
   printf(buffer);

   return 0;
}
```

How can we prevent format string attack?

- There are several prevention methods that we can use:
- Always specify a format string as part of program, not as an input.
- If possible, make the format string a constant. Extract all the variable parts as other arguments to the call.
- Use defences such as Format Guard. Rare at design time.
- Steadily to the patch system. The kernel development and security settings are more about Set UID and complement these vulnerabilities
- Normal use of the printf function like below does not cause any problems.

The Marketplace Vulnerabilities

- Software vulnerabilities and "exploits" are used to get remote access to both stored information and information generated in real time. When most people use the

same software, as is the case in most of countries today given the monopolistic nature of internet content and service providers, one specific vulnerability can be used against thousands if not millions of people. In this context, criminals have become interested in such vulnerabilities. A 2014 report from McAfee's Centre for Strategic and International Studies estimates that the cost of cybercrime and cyberespionage is somewhere around $160 billion per year. Worldwide, countries have appointed public institutions to deal with this issue, but they will likely conflict with the interest of their own government to access people's information in order to prevent crime. As a result, both national security agencies and criminals hide certain software vulnerabilities from both users and the original developer. This type of vulnerability is known as a zero-day exploit.

- Classically, black markets—like illegal weapons or narcotics—require a huge network of trusted parties to perform the transactions of deal-making, document forgery, financial transfers and illicit transport, among others. As it is very difficult to enforce any legal agreement within these networks, many criminal organizations recruit members close to home. This proximity element increases the cost of transaction as more intermediaries are required for transnational transactions, decreasing the overall profit of the original seller.
- Zero-days, on the other hand, are virtual products and can be easily sold without intermediaries over the internet as available technologies are strong enough to provide anonymity at a very low cost. Even if there is a need for intermediaries, "unwitting data mules" can be used to avoid any evidence of wrong doing. This is why the black market is so lucrative compared to gray markets.
- Gray markets, which involve transactions with public institutions in charge of national security, usually require the use of third parties to hide the traces of their transactions. The Hacking Team archive, for example, contains alleged contracts with the Ecuadorian National Secretariat of Intelligence where they used two intermediaries: Robotec and Theola. In the same archive, it is said that third-party companies Cicom and Robotec negotiated the contracts on behalf of the FBI and DEA respectively. It is less likely that white markets face the same problem as it is not in their interest to hide the transaction, it is quite the opposite because companies actively promote the use of their new patches.

Defences against control hijacking

- Preventing hijacking attacks:
- **Fix bugs:** It includes Software **Auditing** and rewrite software in a type safe language (Java, ML). We can use following tools for software auditing such as Coverity, Prefast/Prefix.
- Concede overflow but prevent code execution **(Platform Defense).**
- Add runtime code to detect overflows exploits (**Runtime Defense).**

Defence against control Hijacking-Platform Defence

(1). Marking memory as non-execute (DEP): The 1st approach to platform defense was Data Execution Prevention(DEP). It Prevent attack code execution by marking stack and heap as non-executable •

- Only data is allowed on stack, code cannot be written.

Limitations: –

- Some apps need executable heap (e.g. Just in time compilers).
- Can be easily bypassed using Return Oriented Programming (ROP).

Attack: Return Oriented Programming(ROP): Control hijacking without injecting code

- **Response: Randomization**

ASLR: (Address Space Layout Randomization)

- Map shared libraries to rand location in process memory.
- Attacker cannot jump directly to exec function.

Other randomization methods:

– Sys-call randomization: randomize sys-call id's

Note: Everything in process memory must be randomized

- stack, heap, shared libs, base image Instruction Set Randomization (ISR)

Defence against control Hijacking-Run-Time Defense

- Many run-time checking techniques ...

we only discuss methods relevant to overflow protection.

Solution 1: StackGuard

- Run time tests for stack integrity.
- Embed "canaries" in stack frames and verify their integrity prior to function return.
- Canary Types

• Random canary:

– Random string chosen at program startup.

– Insert canary string into every stack frame.

– Verify canary before returning from function.

• Exit program if canary changed. Turns potential exploit into DoS.

– To corrupt, attacker must learn current random string.

• Terminator canary: Canary = {0, newline, linefeed, EOF}

– String functions will not copy beyond terminator.

– Attacker cannot use string functions to corrupt stack.

Stack Guard implemented as a GCC patch.

– Program must be recompiled

• Minimal performance effects: 8% for Apache

• Note: Canaries do not provide full protection

– Some stack smashing attacks leave canaries unchanged

• Heap protection: Point Guard

– Protects function pointers and set jmp buffers by encrypting them:

e.g. XOR with random cookie

– Less effective, more noticeable performance effects

Advanced Control Hijacking techniques- Heap Spray Attacks

- **Heap-based control hijacking:**

1. use Javascript to spray heap with shellcode (and NOP slides)

2. Then point vtable ptr anywhere in spray area

Javascript heap spraying

```
var nop = unescape("%u9090%u9090")
while (nop.length < 0x100000) nop += nop
var shellcode = unescape("%u4343%u4343%...");
var x = new Array ()
for (i=0; i<1000; i++) {
x[i] = nop + shellcode;
}
```

Pointing func-ptr almost anywhere in heap will cause shellcode to execute.

Defenses

• Protect heap function pointers (e.g. Point Guard)

• Better browser architecture:

– Store JavaScript strings in a separate heap from browser heap

• OpenBSD heap overflow protection:

• Nozzle [RLZ'08] : detect sprays by prevalence of code on heap

Error 404 Hacking digital India part1 chase

- Creating a Trojan file such as android .apk file that will be distributed all over the internet and the person whoso ever download this file, their mobile phone will be hacked easily.
- They are always bounded with other games like Candy Crush , Mini Militia and Clash of Cl
- We may never know but that file may contain a backdoor.
- **"WannaCry" ransomware** is a cyber attack which is like a crypto locker that will not harm your system but it will encrypt everything inside your computer.

- Once data is encrypted then it will ask for a decryption password and to decrypt the data you need a key.
- That key will be generated only when you will pay a certain amount of money into a bit coin address(bit coin is a kind of virtual address) That's why its called **Ransomware**.
- The more things you connect on the internet , greater vulnerabilities over there. Assume everything you do on the internet is already hacked.
- There are 3 potential cyber attacks"
- Web Application Attack
- Mobile Application Attack
- Network Attack
- **Phishing** is a cyber-attack which creates duplicate web pages and ask for login and other personal details. It is a fraudulent practice of sending emails and legitimate websites in order to induce individuals to reveal personal information.
- When we use wifi then there is a possibility of **man-in-the-middle attack** (Attacker intercepts the messages between two parties who believe that they are communicating directly.
- Maximum hacks happen because people were asked to click on certain links and then certain technical controls exploited on their systems using which people either pivoted to other servers and systems or extracted their own systems.
- Indian IT act is thoroughly outdated. India doesn't have any dedicated laws to deal with digital payments, also doesn't have a low on cyber security or a dedicated legislation on privacy and data protection.
- Cyber Security awareness has come to from school level because now at school level everyone has a cell phone.