Name - DIPENDU GHOSH
Course ID - CS6898W
Roll No - CS23M509.

Assignment 2.

1. A vulnerability is a flaw or weakness in a system, software, hardware or process that can be potentially exploited by attackers to compromise the system's security. Vulnerabilities are passive and represent potential risks. They exist due to coding errors, design flaws or configuration mistakes but, by themselves, do not cause harm unless exploited.

An exploit is an actual attack or technique that takes advantage of a vulnerability to cause harm. It is the action taken by an attacker to leverage a vulnerability to gain unauthorized access, steal data, or perform malicious actions. Exploits are active, they involve specific steps or tools to take advantage of a vulnerability.

So to summarize, a weakness or flaw in a system is a vulnerability and a method or attack used to take advantage of that weakness is an exploit.

2

a)

i) -fstack-protector :

This option enables stack protection against buffer overflow attacks by inserting a canary which is a known value on the stack before local variables. The canary is checked before returning from a function. If the canary value has been changed which indicates a buffer overflow, the program is terminated to prevent exploitation. It helps prevent attacks where malicious code tries to overwrite the return address or control flow using buffer overflows.

-O0 :

The -O0 flag disables compiler optimizations. With this option, the compiler generates code without any optimization for performance resulting in a more straight forward translation from the source code to machine code. This is useful during debugging or educational purposes since it ensures that the generated assembly closely matches the original C source code. It makes it easier to trace the program behaviour, as the code remains unoptimized and easier to follow in debuggers.

(ii) ELF stands for Executable and Linkable Format.
ELF is a common file format used for executables, object files, shared libraries and core dumps in Unix-like operating systems. such as Linux. It provides the structure that helps the operating system load, link, and execute the programs. ELF files contain sections such as headers, data, code, and debugging information that the system uses to execute the binary or link it with other binaries.

2. b)

i) 0x1000 and 0x2000 are virtual addresses. In modern operating systems, programs do not directly access physical memory. Instead, they use virtual memory addresses, which are mapped to physical memory by the operating system's memory management unit or MMU. The OS translates these to physical addresses when the program runs.

ii) The initial value of %esp is 0x2000 as mentioned, the stack segment starts at this address during the execution of the program.

iii) Since each segment is 4KB, the stack segment is also 4KB.
So, the amount of stack segment that is free is

$$4096 - 256 = 3840.$$

Now, fact() frame size is as below:

4 bytes for local variable int u
4 bytes for recursive function parameter which is (n-1)
4 bytes for return address.
4 bytes for %ebp (stored frame pointer)

Since in the fact() function there is no 'vulnerable' local variables, such as arrays (buffers) or alloca() to allocate memory to stack, the canary is not added here although the compilation flag of -fstack-protector is used.

$\therefore$ Total = 4+4+4+4 = 16 bytes.

$\therefore$ Max depth/value of $N = \frac{3840}{16}$

$= 240.$

iv) For N>240, the program will try to write outside the stack segment causing a stack overflow and this will ultimately lead to a segmentation fault. The program will eventually crash. or terminate unexpectedly. This happens as the stack memory is exhausted and program cannot continue to allocate the necessary space for further function calls.

3

a) W⊕X → (iii) Prevents execution from certain memory pages.
W⊕X refers to the Write XOR Execute protection also known as NX bit or DEP (Data Execution Prevention). It ensures that memory pages can either be writeable or executable, but not both at the same time. This helps prevent the execution of code from areas of memory marked as writable.

b) Canaries → (i) Enabled mainly by the compiler.
Canaries are a stack protection mechanism inserted by the compiler to detect buffer overflows. A special value called a canary is placed between the buffer and the control data which is the return address. and the value is checked before returning from the function. If the canary is altered, the program terminates, indicating a buffer overflow attempt.

c) ASLR → (iii) Enabled mainly by the Operating System.
Address Space Layout Randomization is a security feature that randomizes the memory address space of key program areas like stack, heap, bibraries each time a program is executed. This makes it harder for attackers to predict where their code or return addresses will reside. ASLR is managed by the operating system.

4.

The users are :-

Teachers - There are 4 teachers who create and manage quiz materials, view answer scripts and edit grades sheets.

TAs - There are 20 TAs who assist in reviewing answer scripts but do not have edit access.

Students - There are 400 students who can only view their own answer scripts and view final grade sheets.

The groups are :-

teachers - The group of 4 Teachers.

ta's - The group of 20 TAs

students - The students will have a general group of 400 students and also individual groups to view the individual answer scripts.

The objects are :-

Question papers :- Files created by teachers for quizzes which become visible to all after the exam.

Answers Scripts - Files that store individual student answers. Teachers and TAs can view them but only the specific student can view their own script.

Grade sheets - Files that store the grades. Editable by teachers but viewable by all students, tas and teachers.

The access policies can be as below:

Question Papers - ~~before exam~~ mapped to the teachers group.

before exam - the permisson is 640 which is

Owner will have read and write access.

Group will have only read access.

Others will have no permissions.

after exam - the permission is changed to 444 which is

Owner will have read permission.

Group will have read permission.

Others will have read permission.

Answer scripts - this ~~is~~ mapped to Teaches and ~~students~~ ~~and~~ TA

group :. the permission is 440.

Owner will have read permission.

Group will have read permission.

Others have no permission.

Grade sheets - this is again mapped to the teaches, students and TAs. The permission is 644.

Owner will have read and write permission.

Group will have read permissions.

Other will have read permissions.

So this is how the system is setup where different users are present and have different levels of access to the files which ensures that only authorized individuals can create, view or modify the objects.

5.

[A] Hardware Interrupt → [ii] Availability of CPU to processes.

Hardware interrupts are used to ensure the availability of the CPU to processes by allowing the CPU to respond to hardware events like I/O operations or timers. that require immediate attention.

[B] CPU rings → [i] Isolate OS from user processes.

CPU rings define different privilege levels in the system, with the OS running in a more privileged ~~ring~~ ring (Ring 0) and user processes running in less privileged rings (Ring 3). This isolates the OS from user processes, preventing user processes from directly interfering with OS operations.

[C] Paging → [iv] Isolate user processes.

Paging is a memory management technique that isolates user processes by giving each process its own virtual memory space. This prevents one process from accessing the memory of another process, thus ensuring memory isolation.

[D] Fat pointers → [iii] Memory Buffer Checks.

Fat pointers are pointers that carry additional information, such as bounds, which help in performing memory buffer checks to prevent buffer overflows and ensure ~~code~~ safe memory access.

6.

There seems to be 2 vulnerabilities in the given program.

Vulnerability 1 - Buffer overflow.

The scanf ("%s", name) function reads user input into the name array but it does not limit the length of the input in case the user enters more than the allocated size of 128 characters. By entering more than 128 characters into name an attacker can overwrite the return address on the stack and potentially exploit the control of the program to execute arbitary code.

Vulnerability 2- Format string.

The printf(name) passes the user input "name" directly to the printf() function without any format string. An attacker could input special formats like %x %x %x to read values from the stack or use %n to write arbitary values to memory which can lead to potentially leaking sensitive data or allow the attacker to write arbitary values to memory.

7.

a) A process in a computer system may be considered as a subject as well as an object. As a subject, the process can initiate actions like reading and writing files, executing programs, or sending network requests. As an object, it can be acted upon by other subjects, such as when another process queries its state, terminates it or sends it signals.

b) The disadvantages of incorporating access control policies in hardware are as below!

i) Complexity: Implementing sophisticated access control in hardware can significantly increase the complexity of the hardware design.

ii) Lack of Flexibility: Hardware is less adaptable compared to software, so updating or modifying access control policies requires redesign or firmware updates, which is costly.

iii) Cost: Incorporating access control into hardware increases development and manufacturing costs.

iv) Scalability: It may not scale well with evolving software applications that require different or more complex access control policies.

c) Memory protection ~~and~~ mechanisms and process isolation enforced by the operating system prevent P1 from accessing or invoking functions in P2. These mechanisms include;

i) Virtual memory: Each process has its own virtual ~~memory~~ address space, isolating it from other processes.

Access Control: The OS restricts a process's access to memory or resources owned by another process.

CPU privilege levels: User-mode processes cannot directly manipulate the memory or execution of other processes, as that would require kernel-mode access.

d) The user identifier (UID) in Unix systems is assigned by the system administrator or during the creation of the user account via system tools like useradd. The UID is stored in the /etc/passwd file along with the user's account details. By default, UID's start from a predefined number. It is 1000 for normal users and with a lower number such as 0 for root. which is reserved for system users.

e) U1 can do or execute the below commands so that U2 can only list the files ~~of~~ in the directory /home/U1/team:

i) chmod 755 /home/U1/team

ii) chmod 600 /home/U1/team/*.

In command (i) U1 is setting the permission of /home/U1/team for to all access for User, read and execute for group G1 and read and execute permissions for others.

Then in command (ii) U1 is setting the permission of all the files or contents of the folder /home/U1/team/ to read and write for user and ~~nother~~ ~~per~~ no other permission for group and others.

When User U2 does a "ls" or "ls -l" on /home/U1/team/ it should see all the contents of the /home/U1/team/ folder.

This type of permissions can be achieved in some other combinations as well. The other combination can be

i) chmod 755 /home/UI/team

ii) chmod 744 /home/UI/team/*

In command (i) the directory permission is set to allowing owner full access and for the group and others access to list all files

In command (ii) the file permissions under the directory is set to allowing the owner full access and for the group and others to read only (which is restricted by the directory permissions).

~~The other combination can be~~

~~i) chmod 755 /home/UI/team.~~

~~ii) chmod 744~~

In ~~both~~ the above cases its seen that the directory /home/UI/team has the execution permission for ~~others~~ So this is the key. The directory needs to have execution permission for others so that the files are viewable but not ~~readable~~.

There can be other options to change ~~ownership~~ group and all on folder levels as well.