Chapter 6 – Buffer Overflow

* When does Buffer Overflow happen🡪
* When the input fields are unable to handle the input byte size of the user.
* Source String length> Destination Strength length
* What is one of the most famous types of attacks?

Overflowing a buffer assigned to a subroutine. When a program writes data to a buffer and it exceeds the acceptable amount, this can cause some crucial data to be overridden. ( especially in adjacent memory locations)

* Buffer overflow bugs lead to arbitrary code execution. This enables the attacker to exploit this to run his/her own code and take control over the machine.
* Add malicious code by: OVERWRITING THE **Instruction Pointer**
* When a subroutine is called, there is a fixed size of stack
* When it ends, the stack is cleared as well
* Stack overflow is more common than heap overflow attacks
* This is because stacks contain a bunch of nested functions with returning addresses
* These return addresses can be replaced with instruction to conduct the attack

REGISTERS

|  |  |
| --- | --- |
| **EAX** | Accumulator Register, stores values of arithmetic and logical operations, as well as the return values. |
| **EBX** | Base pointer to the data section of the program. |
| **EDX** | I/O pointer, Used for complex operations (division, multiplication) |
| **EBP** | Stack frame base pointer register, points to the start of function stack frame, used to access function arguments |
| **ESP** | Stack pointer, points to the top of the stack |
| **EIP** | Points to the next instruction |
| **ECX** | Used as a counter when using loops |

**Idea Of Attack**

* **The overflow corrupts data on the stack**
* **The idea is to cerate an input to gain control of the critical CPU register to manipulate the memory to create a reverse shell.**

**Protection Against Attacks**

* Safer Library Support
* Manual Auditing Code: Replace unsafe functions with safe ones
* Address Space Layout Randomization: the memory spaces should be random to avoid exploitation

**Steps of Buffer Overflow Exploitation**

1. Attach the program to **immunity debugger**
2. **Fuzz** the program to check the buffer size causing buffer overflow
3. Create a **pattern** to check the valueof **EIP and its offset**
4. **Overwrite the EIP** value
5. Check **bad characters**
6. Find the **JMP point to ESP**
7. Prepare **payload**
8. **Exploit the server to gain access.**

* **What is Mona.py?**

**It is a script that is used to automate and speed specific searches**