BSc (Hons) in IT Specialized in CS
Year 2, 2022

Tutorial 05
SNP **Buffer Overflow Attacks**

---

1. What is a Buffer Overflow Attack

   Attackers exploit buffer overflow issues by overwriting the memory of an application. This changes the execution path of the program, triggering a response that damages files or exposes private information. For example, an attacker may introduce extra code, sending new instructions to the application to gain access to IT systems.

   If attackers know the memory layout of a program, they can intentionally feed input that the buffer cannot store, and overwrite areas that hold executable code, replacing it with their own code. For example, an attacker can overwrite a pointer (an object that points to another area in memory) and point it to an exploit payload, to gain control over the program.

2. Types of Buffer Overflow Attacks

   Stack-based buffer overflows are more common, and leverage stack memory that only exists during the execution time of a function.

   Heap-based attacks are harder to carry out and involve flooding the memory space allocated for a program beyond memory used for current runtime operations.

3. How to Prevent Buffer Overflows

   Address space randomization (ASLR)
   Data execution prevention
   Structured exception handler overwrite protection (SEHOP)

4. Answer the following questions concerning how a program is stored in memory during its execution.

   1) Which segment of memory has contents that remain unchanged during program execution?

2) Does the programmer have complete control over how the stack is organized?

3) What is the relationship between the order in which variables appear in a function and the order in which these same variables are stored in a function's stack frame?