CIS 751 In-Class Assignment 2

Chuck Zumbaugh

September 12, 2025

1 Changes to prevent buffer overflow attacks from running

- At runtime, randomize the address the program is loaded at. This will prevent an attacker from knowing what address to return to.
- Prevent data on the stack from being executed. Thus, if an attacker tries to return to shell code the system will prevent it from running.
- Prevent the program from returning to an address on the stack. Thus, an attacker could not return to malicious code inserted during the buffer overflow.
- Instead of saving the instruction pointer on the same stack that is used to store data, the system could use another region of memory to manage this. Since this region is separated from where local data is written, it would not be overwritten in the event of a buffer overflow.
- When a function is called, the arguments, EIP, and old EBP are pushed onto the stack (at lower addresses). Perhaps the OS could place a lock on all addresses that are higher than the old EBP when a new stack frame is created. Thus, an attacker attempting to overwrite this data will be prevented from hijacking the EIP.