

Protection Systems

Last modified on: 3/13/2009, 2:09 PM

Friday, March 13, 2009

1:40 PM

Stack Guard

- used canary before return address where if the canary was "killed" they knew the stack was compromised.

Pointer Guard

- Marks fields as pointers or not pointers
- Pointers are encrypted so if change it won't match the encryption

Address Space Randomization

- Assumptions
 - o Attacker knows approximate layout of memory
- Idea
 - o Take away this knowledge from the attacker
 - o Shift stack start - easy
 - o Also must shift txt and mmap areas
 - Shifting mmap area should be easy because libraries are already designed to be loaded anywhere
 - Effectively 16 bits of random because of possible collisions
 - Need to just recompile with an option to make the txt movable.
 - Also needs to 4kb boundary
 - Finding the location of sleep and adding a constant they can still find this zone.
 - o Heap? 20-bits
- Fixes
 - o 64-bit address space

16-bits of randomness gives about 4 min
- Attacker may not need to guess all randomness simultaneously

Internal Randomization

- Stack
 - o Local variable ordering
 - o Interframe padding
 - o Frame ordering?
 - o Parameter ordering?
- Code Segments
 - o Function ordering
 - o Padding
 - o Instruction choices
 - o Basic blocks
 - o Polymorphic code?
 - o Instruction set randomization
- Heap
 - o Interallocation padding
 - o Ordering