Session 0x9
Defense Mechanisms

Security Summer School

ACS/Ixia/Hexcellents

Classes of Defense Mechanisms

- restricting information

Classes of Defense Mechanisms

- restricting information
- limiting control flow

Classes of Defense Mechanisms

- restricting information
- limiting control flow
- runtime checks

Classes of Defense Mechanisms

- restricting information
- limiting control flow
- runtime checks
- least privilege

NX

- protection against code injection (shellcode)
- pages that are executable are not writeable

NX

- protection against code injection (shellcode)
- pages that are executable are not writeable
- bit in page table entries
- requires hardware, kernel support
- enabled/disabled through compiler flags

Bypass NX

- ret-to-plt, ret-to-libc
- Return Oriented Programming (ROP)
- mprotect()

Mechanism

- ELF segments specify required permissions
- loader maps segments in memory pages
- permissions can later be changed using mprotect()

Address Space Layout Randomization (ASLR)

- maps regions at random addresses
- stack, data (heap), shared libraries, VDSO page
- PIE: binary image is also relocated

Address Space Layout Randomization (ASLR)

- achieved RIP control
- no known address to jump to
- ASLR + PIE + ROP = ?

Address Space Layout Randomization (ASLR)

- achieved RIP control
- no known address to jump to
- ASLR + PIE + ROP = ?
- invariants

Address Space Layout Randomization (ASLR)

- achieved RIP control
- no known address to jump to
- ASLR + PIE + ROP = ?
- invariants
  - page alignment
  - contiguous mappings
  - order of mappings
  - value range (entropy)
  - preserved in child processes

Bypass ASLR

- information leak
- partial overwrites
- bruteforce (32 bit)
- NOP sled
- jmp esp

Stack Protection Mechanisms

- address the issue of stack buffer overflows
- stack protector ("canary"): secret value placed in stack frame
- SafeStack (clang): separation in safe stack and unsafe stack
- require compiler support

FORTIFY

- protection against buffer overflows
- library functions are replaced with fortified versions
- perform checks on buffer lengths at runtime
- requires compiler support

RELRO

- protects relocation sections from overwrites (e.g. GOT)
- types: partial, full
- resolve symbols at load time

RELRO

- protects relocation sections from overwrites (e.g. GOT)
- types: partial, full
- resolve symbols at load time
- bypass: other code pointers (GOT in libraries, application-specific pointers, return address)

seccomp

- sandbox based on syscall filtering
- set up by the application at runtime via prctl
- enforced by the kernel
- libseccomp, seccomp-BPF

Classes of Defense Mechanisms (review)

- restricting information:

Classes of Defense Mechanisms (review)

- restricting information: ASLR, PIE, stack protectors, pointer guarding
- limiting control flow:

Classes of Defense Mechanisms (review)

- restricting information: ASLR, PIE, stack protectors, pointer guarding
- limiting control flow: CFI (clang), CFG (Windows)
- runtime checks:

Classes of Defense Mechanisms (review)

- restricting information: ASLR, PIE, stack protectors, pointer guarding
- limiting control flow: CFI (clang), CFG (Windows)
- runtime checks: FORTIFY, seccomp
- least privilege:

Classes of Defense Mechanisms (review)

- restricting information: ASLR, PIE, stack protectors, pointer guarding
- limiting control flow: CFI (clang), CFG (Windows)
- runtime checks: FORTIFY, seccomp
- least privilege: NX, SafeStack, RELRO

Resources

- https://libc.blukat.me/
- https://refspecs.linuxbase.org/LSB_4.1.0/LSB-Core-generic/LSB-Core-generic/libcman.html
- https://medium.com/@HockeyInJune/relro-relocation-read-only-c8d0933faef3
- https://lwn.net/Articles/656307/
- https://lwn.net/Articles/593476/