Dynamic Analysis

nafiez

- Executing code either on physical machine or emulator
- Tracing the code that gets executed
- Analyzed code execution

Various level of analysis can be approach

- CPU instructions
- CPU exceptions (interrupts, page faults, etc.)
- CPU memory access
- OS System Calls (kernel level)
- OS APIs
- OS high-level activity (registry, filesystem, etc.)
- Network activity

Common Problems

Static	Dynamic
Cost of reversing (time, etc.)	Execution path depends on environment
Code obfuscation and packer	Logic visibility
Code coverage	Performance (emulator, etc.)
	Scalability (hardware, etc.)

Debugging 101

Get to know your debugger tho

- Software debugging consists of
  - User-mode – ring 3
  - Kernel-mode – ring 0
- For some cases, symbols are important
- Kernel debugging can be done either locally or remote
- This class will focus more on user-mode debugging
- Understand debugger features (next slide)

Single Stepping

- Execute the application one instruction at a time

Software Breakpoints

- Break execution of target process at specific address
- Mostly implemented using *INT 3*
  - Debugger writes a byte with value *0xCC* to memory address
  - You won’t see this modification in memory view
- No limitation in software breakpoints
- Cannot break on reads or write addresses, just execution

Hardware Breakpoints

- CPU debug registers provide support for up to four HW breakpoints
- Mostly related to x86 debug registers
  - DR0-3 store the linear addresses to be monitored
  - DR7 configures type of event
    - Break on execution, read and read/write
- Does not modify code bytes
- Limited number of breakpoints

Read and Write Memory

- Debugger should be able to read and performs write into the virtual memory space of the debuggee
- This can be through normal Windows API functions
  - ReadProcessMemory()
  - WriteProcessMemory()

Initial Breakpoint

- First time debugger gets control of the target
- Anything else than system breakpoint, application can run before you can control it

Use the EXE you compile for the analysis

fire up debugger