Introduction to Software Reverse Engineering

Arvind S Raj
(arvindsraj@am.amrita.edu)

CSE467 IntroSSOC

B.Tech CSE Jan-May 2017

Looking back and ahead

Till now

- Assembly programming - with C library and using system calls.

Up next

- Software reverse engineering: from assembly to higher level code.

Overview

- Objective: Given an executable file, determine what is does.
- Be able to "read" assembly code i.e. map to higher level code.
- Used in malware and vulnerability analysis.
- Also, a standalone extremely active research area.

Reversing steps for normal binaries

- Simply run the binary: might give some insight into how it works.
- View printable strings in binary. *strings* command helps but use with caution.
- Use function names to guide you on possible behaviour. Names may not be available always.
- Find entry point: *main*, *_start* or entry point using readelf.
- Use a debugger to observe behaviour: makes it easier to understand what is happening.

Reversing steps for normal binaries(cont.)

- Remember objective: why are you reverse engineering this binary?
- Maintain notes, comments etc. Not possible to remember too many details simultaneously.
- Remember System V calling convention.
- Refer to assembly instruction documentation for unknown instructions.

Practice time!

Let's start reversing binaries!