Trabajo Práctico
Análisis de un programa binario

Gustavo Grieco (gustavo.grieco@gmail.com)

R-222 Arquitectura del Computador

Introducción

El objetivo de este trabajo es que el alumno aprenda las técnicas para la inspección de un programa binario (compilado sin información de depuración o su fuente) y la deducción de información relevante del mismo. Dichas técnicas se encuadran dentro de un área llamada Ïngeniería Inversa". Utilizar ingeniería inversa para analizar un binario es básicamente realizar documentación sobre el mismo, pero sin utilizar la documentación provista por el programador original del software. Este trabajo práctico pretende abordar una ínfima parte de esta área, pero con el objetivo de incentivar al alumno a que averigüe más del tema.

Herramientas

Se provee al alumno información concreta de herramientas que deben ser usadas para resolver este trabajo práctico:

- gdb: Este programa permite depurar paso a paso un ejecutable. Es una herramienta básica, pero muy poderosa para la detección de errores en tiempo de ejecución. En este trabajo podemos utilizarlo para desensamblar e inspeccionar el estado del programa analizado.
- objdump: Este programa permite obtener información detallada de un programa compilado, así como las instrucciones ensamblador que contiene, las secciones de datos inicializadas o por inicializar y otros datos relevantes. Se puede obtener el código máquina de un binario ejecutando:
  objdump -d binario
- ltrace: Este programa permite identificar rápidamente llamadas a librerías o syscalls realizadas por un binario durante su ejecución, así también como sus argumentos. Se pueden obtener la información de llamadas a funciones externas y syscalls ejecutando:
  ltrace -s ./binario
- strings: Este sencillo programa revisa un archivo arbitrario buscando listas de caracteres imprimibles (i.e. strings) y los muestra por salida estándar.

- radare: Es un conjunto de herramientas para realizar análisis de binarios de múltiples plataformas. La herramienta principal combina un depurador y un desensamblador, además de ser fácilmente extensible usando diversos lenguajes (python, java, etc). Se recomienda al alumno instalar y leer la documentación apropiada para aprender a utilizar los comandos de esta herramienta ya que facilita considerablemente el proceso de análisis de binarios.

Otras herramientas podrían ser aceptables para realizar este trabajo. La ingeniería inversa es un problema difícil y muchas veces no se puede realizar de manera completa sobre un programa por lo tanto cualquier herramienta adicional que aporte información será bienvenida. En cualquier caso, consulte al docente.

Información Inicial

El alumno deberá analizar el archivo binario provisto. Este programa posee un mensaje cifrado, que intenta descifrar cada vez que se lo ejecuta. Por alguna razón, el descifrado falla. El alumno deberá identificar los datos del mensaje cifrado, cómo se descifra, las condiciones bajo las cuales dicho proceso falla y el mensaje descifrado.

Metodología

Para realizar el trabajo el alumno debe resolver los siguientes puntos, respondiendo las preguntas que se plantean:

- Describa toda la información general del binario analizado que pueda. Por ejemplo: ¿Que arquitectura utiliza? ¿Incluye símbolos de depuración?, ¿Que compilador se utilizó?, ¿Que funciones estándar de la librería de C se utilizan? ¿Qué función se utiliza para que el binario se comporte de manera no determinista?
- Determine la dirección de la primera instrucción del main del binario.
- Utilizando el grafo provisto por radare2 (ya generado aquí) describa el funcionamiento de la programa desde el punto de vista de los nodos de dicho grafo. Este grafo le será muy útil ya que es como una especie de "mapa"del binario. ¿Qué indica el color de cada flecha? ¿Por qué la información de las llamadas a funciones no fue incluida como más aristas en el grafo?
- Identifique el mensaje codificado.
- Analice la información de saltos condicionales. Una buena idea es empezar buscando variables locales alocadas en el stack. ¿Qué técnica se utilizó para encriptar el mensaje original? Detalle como se encripta dicho mensaje en las instrucciones del binario.

- Analice las instrucciones de ensamblador relevantes para saber porqué el mensaje se desencripta correcta o incorrectamente. Este paso puede tomar su tiempo: Reúna cada una de las pistas de los items anteriores e intente descartar la mayor cantidad de código ensamblador para el análisis manual.

Características adicionales

El alumno puede extender el trabajo (opcionalmente) con las siguientes mejoras:

- Modificar o inducir al binario para siempre decodifique correctamente (si no lo hizo ya).
- Modificar el binario para que decodifique un mensaje a elección del alumno.

Entrega del Trabajo

El trabajo será evaluado por la cátedra mediante una presentación de un informe. El alumno debe entregar dicho informe incluyendo datos académicos (integrantes del grupo, legajos, fechas) y reportando problemas y soluciones encontradas durante la realización del trabajo y posibles extensiones al mismo.

Material y Referencias

How does linux execute my main()?
Radare, the reverse engineering framework
Using GDB without debugging symbols on x86?
Reversing: Secrets of Reverse Engineering de Eldad Eilam
Executable and Linkable Format (ELF)