

Introduccion a la Ingeniería Inversa

Análisis de malware, cracking de software...

Carlos Ledesma Peña Fernando Díaz Urbano

Grupo de Desarrollo de Google, Málaga

Índice

Introducción

Entorno

Herramientas

Lenguajes

Seguridad

Estado actual

Caso práctico

GNU Radio

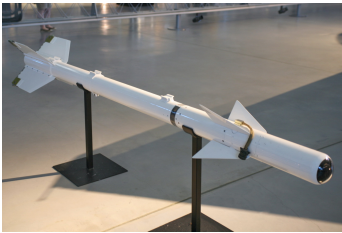
Pager POCSAG

Para terminar

Conclusiones

Utilidades

Antecedentes



Misil K-13 soviético aire-aire



CDP MPC 1600-1

Aplicaciones típicas en informática

- **Mantenimiento de programas sin código fuente**
Interacción de componentes, creación de documentación...
- **Análisis de malware**
Conocer objetivos y procedimientos de un malware
- **Búsqueda de vulnerabilidades**
En programas de código cerrado
- **Modding y cracking**
Añadir o remover funcionalidades

... Entre otras.

Aplicaciones típicas en informática

- **Mantenimiento de programas sin código fuente**
Interacción de componentes, creación de documentación...
- **Análisis de malware**
Conocer objetivos y procedimientos de un malware
- **Búsqueda de vulnerabilidades**
En programas de código cerrado
- **Modding y cracking**
Añadir o remover funcionalidades

... Entre otras.

¿Qué debemos conocer?

- **El entorno**

Sistema operativo, proceso, ejecutable, librería...

- **Las herramientas**

Desensamblador, depurador, monitor de recursos...

- **Los lenguajes**

Ensamblador (x86), bytecode (CIL), scripting (Perl)...

- **Los patrones típicos**

Reconocer un bucle while, una inyección en un proceso...

Tipos de análisis

Estático

- Permite examinar todos los caminos y valores de variables
- Examen directo de ramas de ejecución poco frecuentes
- No efectivo contra código automodificable o muy ofuscado

Dinámico

- Sólo se ejecuta un camino cada vez, dependiente del entorno
- Más fácil entender lógica complicada viendo la ejecución
- Efectivo contra código automodificable u ofuscado

Índice

Introducción

Entorno

Herramientas

Lenguajes

Seguridad

Estado actual

Caso práctico

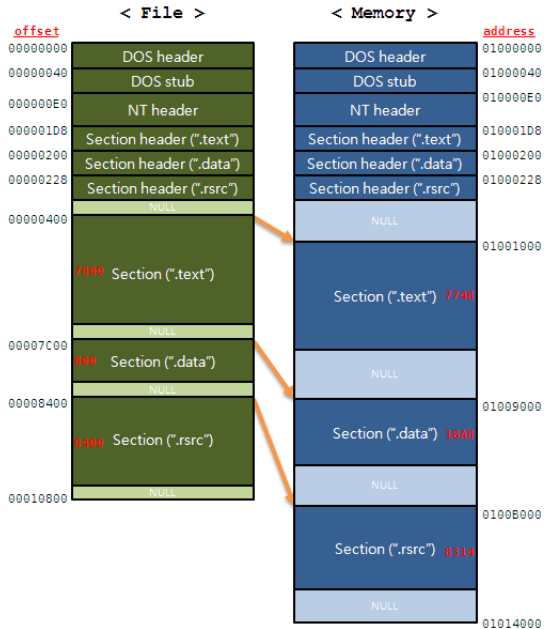
GNU Radio

Pager POCSAG

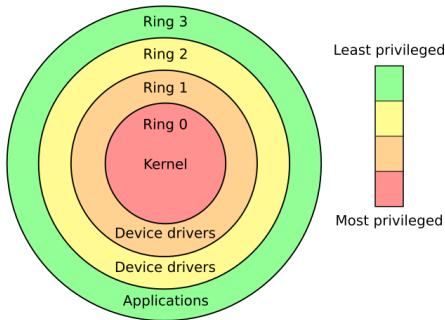
Para terminar

Conclusiones

Utilidades



Anillos de protección



HyperVisor

- **AMD-V:** Pacifica
- **Intel VT-x:** Vanderpool

Índice

Introducción

Entorno

Herramientas

Lenguajes

Seguridad

Estado actual

Caso práctico

GNU Radio

Pager POCSAG

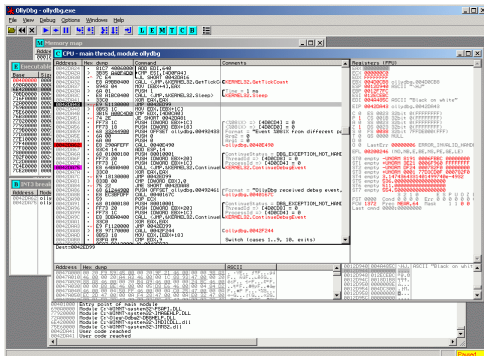
Para terminar

Conclusiones

Utilidades

Ollydbg

- Posee GUI
- User-level debugging(Ring 3)
- Más sencillo para principiantes



Índice

Introducción

Entorno

Herramientas

Lenguajes

Seguridad

Estado actual

Caso práctico

GNU Radio

Pager POCSAG

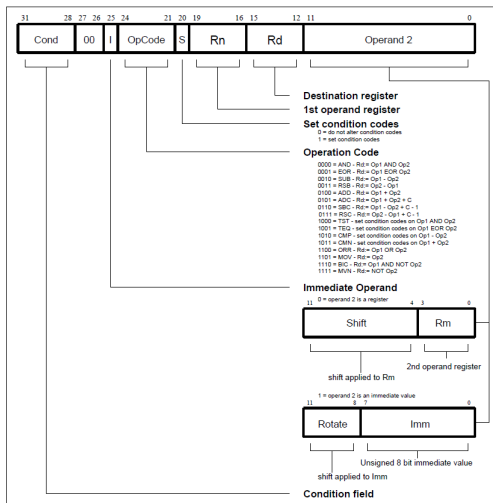
Para terminar

Conclusiones

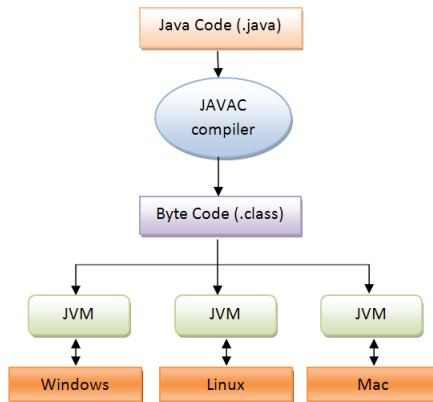
Utilidades



Opcodes



Bytecode





Seguridad

- **2005:** Versión mejorada de *BlueSniper*.
- **2007:** Ataque sobre teclados inalámbricos.
- Ataque sobre el pasaporte europeo.
- **2008:** Michael Ossmann repasa en Black Hat sobre el estado de la seguridad de las radiocomunicaciones, y advierte que SDR accesible es peligroso.
- Ataque sobre el sistema de tarjetas del metro de Boston y de pago remoto en peajes.
- **2009:** Ataque práctico sobre *GSM*.
- **2010:** Lectura de *RFID* a larga distancia.
- **2011:** Ataque sobre *GRPS/EDGE* y *UMTS/HSPA*.



Estado actual

- En 2010, Eric Fry se da cuenta de algo extraño al realizar ingeniería inversa a un *driver* de un dispositivo *USB* para recepción *FM* y *DAB+*. Lo que viaja del dispositivo al PC no es audio, sino muestras de la señal en una etapa intermedia entre la señal de radiofrecuencia y el audio.
- En 2012 nace el proyecto *rtl-sdr*, que proporciona una interfaz para usar estos dispositivos como SDR's (sólo recepción, pero muy asequibles).
- Interés en integrar SDR en la comunidad de pentesting, con nuevas herramientas que permiten inyectar paquetes de diversos protocolos al vuelo.

Índice

Introducción

Entorno

Herramientas

Lenguajes

Seguridad

Estado actual

Caso práctico

GNU Radio

Pager POCSAG

Para terminar

Conclusiones

Utilidades

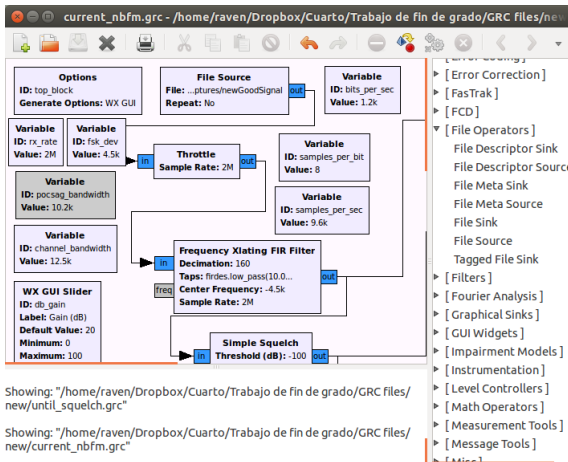


¿Qué es GNU Radio?

GNU Radio es un entorno de desarrollo *open source* multiplataforma de procesamiento de señales en general, si bien está especializado en SDR, pero no limitado a ello.

- Ofrece una interfaz gráfica, *GRC*, además de las interfaces para *Python* y *C++*. La de *Python* es una envoltura de la de *C++*, y la gráfica una envoltura de la de *Python*.
- La interfaz gráfica sirve para crear diagramas de flujo, con conexiones entre bloques que representan funciones de procesamiento de señales.
- Los bloques pueden ser de entrada o salida, para interactuar con el exterior (parte hardware de SDR, tarjeta de audio, disco duro...), o de entrada y salida, implementando funciones en sí.

GRC



Showing: "/home/raven/Dropbox/Cuarto/Trabajo de fin de grado/GRC files/new/until_squelch.grc"

Showing: "/home/raven/Dropbox/Cuarto/Trabajo de fin de grado/GRC files/new/current_nbfm.grc"

¿Qué es un pager POCSAG?





Al ataque

- Tras escanear en el rango de frecuencias que menciona el *pager* en su parte de atrás, y los avisos que alcanzo a capturar se emiten en la misma frecuencia.
- Construyo el diagrama de flujo (no sin mucho esfuerzo) para decodificar con arreglo al estándar *POCSAG* y efectivamente, se ajusta al estándar. Cada dispositivo tiene un ID, y suena cuando se emite el suyo.
- Modifico el diagrama de flujo y creo un bloque personalizado para *GRC* para imprimir en consola los ID's según se capturan los avisos.



Dificultades encontradas

- Dominio completamente nuevo para mí, y falta de base sólida a la hora de resolver los problemas (días de diagnóstico por problema).
- *GRC* no está hecho para aprender a base de prueba y error desde el principio, no es fácil saber qué está fallando ni por qué (curva de aprendizaje elevada).
- Limitaciones del hardware, mi portátil usa *USB 2.0*, lo que limita el ancho de banda capturable de una vez, además de no tener potencia de procesamiento suficiente y descartar muestras si se usaban varias operaciones simultáneamente.

Índice

Introducción

Entorno

Herramientas

Lenguajes

Seguridad

Estado actual

Caso práctico

GNU Radio

Pager POCSAG

Para terminar

Conclusiones

Utilidades

Conclusiones

- Desde el punto de vista económico y humano, es necesario invertir en la seguridad de los sistemas informáticos. No sólo se protege de las malas intenciones, sino de las buenas intenciones equivocadas.
- La "seguridad" por oscuridad no es seguridad, si un sistema necesita que su forma de funcionar no sea pública para ser seguro, no es seguro igualmente. *Sólo la clave debe ser desconocida para el resto.*
- No se le ha prestado suficiente atención a la seguridad de las radiocomunicaciones en el pasado, y ahora se dispone de herramientas basadas en SDR que facilitan aprovecharse de sistemas vulnerables. Hay que prestarle atención desde ya.

Utilidades

- Aprender conocimientos básicos de radio (inquietud personal).
- Incorporar una nueva herramienta de trabajo (es posible que se incorpore en labores de pentesting).
- Servir de guía de inicio rápido a SDR a los investigadores del departamento.
- Obtener el título de Graduado en Ingeniería Informática.

¡Gracias por vuestra atención!