APOKLIPTIKO

El presente es un índice de los temas tratados por Ricardo Narvaja en su curso "introducción al cracking con OllyDbg desde cero" al convertirse este en una guía obligada de consulta tanto para el cracker principiante como para quienes ya tienen conocimientos de este arte he tratado de hacerlo lo más detallado posible para poder hallar rápidamente el tema que estemos buscando.

He adjuntado también material publicado directamente en la lista de CLS que tiene que ver con algún tema tratado en el curso y que consideré importante a pesar de no formar parte oficialmente de los tutes de Ricardo, por ejemplo:

Capitulo XLIV

Desempacado de ACProtect1.09g.f con todas las funciones habilitadas (3ª parte)

- Arreglando los stolen bytes y el OEP
- Reparando los antidump

¿Cómo injertar si la sección a crear ya está ocupada? (Anexo parte 44)

Esto indica que el tema contiene un comentario dentro del mismo .pdf. Los he insertado como comentario para evitar el hecho de adjuntar archivos.

Esperando sea útil, aquí va esta contribución, si consideras que algo debe ser corregido o agregado puedes contactarme al siguiente mail: apokliptiko@yahoo.com indicándome el tema exacto a ser considerado y en que parte del tute se halla.

Capitulo I

Areas del OllyDbg

- Desemsamblado
- Registros
- Dump
- Stack o pila

Otras ventanas del OllyDbg

- L View Log
- E View Executables
- M View Memory
- T View Threads
- W View Windows
- H View Handles
- C View CPU
- / View Patches

- K View Call Stack
- B View BreakPoints
- R View References
- ... View Run Trace

Configuración del OllyDbg como JIT (Just in Time Debugger)

Como agregar plugins al OllyDbg

Atajos del teclado

- F2
- F7
- F8
- F9
- F12

Capitulo II

Sistemas numéricos

- Binario
- Decimal
- Hexadecimal
  - Numeros positivos y negativos en hexadecimal

Caracteres ASCII

¿Qué es el stack o pila?

Capitulo III

¿Qué son los registros y para qué sirven?

- División de los registros (EAX ➔ AH | AL)

¿Cómo cambiar los valores de los registros?

¿Qué son los flags o banderas?

- El flag O o flag **O**verflow (desbordamiento)
- El flag A o flag **A**uxiliar
- El flag P o flag de **P**aridad
- El flag Z o flag **Z**ero (cero)
- El flag S o **S**ign flag (flag de signo)
- El flag C o **C**arry flag (flag de acarreo)
- El flag T, D e I

Capitulo IV

Instrucciones ASM

- NOP
- PUSH
- POP

- PUSHAD
- POPAD
- PUSHA
- POPA
- MOV
- MOVSX
- MOVZX
- LEA
- XCHG

Capitulo V

Instrucciones matemáticas

- INC
- DEC
- ADD
- ADC
- SUB
- SBB
- MUL
- IMUL
- DIV
- XADD
- NEG

Instrucciones lógicas

- AND
- OR
- XOR
- NOT

Capitulo VI

Comparaciones y saltos condicionales

- CMP
- TEST

Saltos

- JMP
- JE O JZ
- JNE O JNZ
- JS
- JNS
- JP O JPE
- JNP O JNPE
- JO
- JNO
- JB
- JNB
- JBE

- JNBE
- JL
- JA JG JAE JGE

Capitulo VII

Instrucciones assembler

- CALL
- RET

Capitulo VIII

Instrucciones para loops o ciclos (bucles)*

- LOOP
- LOOPZ LOOPE
- LOOPNZ LOOPNE

Instrucciones para el manejo de cadenas de bytes

- MOVS
- REP
- REPE REPZ
- LODS
- STOS
- CMPS

Modos de direccionamiento

- Directo
- Indirecto

*En este capítulo hay explicación de como formar un loop.

Capitulo IX

Definiciones

- Entry Point
- DLL
- Funciones de exportación APIs

¿Como sacar el listado de APIs que usa un programa? [Name (label) in current module CTRL+N]

¿Como usa un programa las APIs?

Capitulo X

BreakPoints

- BreakPoint común o BPX
- BreakPoints en memoria o Memory BreakPoint

Capitulo XI

- **Hardware BreakPoints o HBP**
- **BreakPoints condicionales o conditional breakpoint**
- **BreakPoints condicionales con logueo o conditional log breakpoints**

Capitulo XII

¿Como aprovechar los mensajes en Windows?

- Message BreakPoints o BMSG

Capitulo XIII

Pescando y revirtiendo seriales

- Harcoded
  - API GetDlgItemTextA

Capitulo XIV

Solución del crackme propuesto en el capitulo anterior

- API lstrcmpA
- API GetWindowTextA

Otro hardcoded

- API memset
- API lstrlen

Capitulo XV

Solución del crackme propuesto en el capitulo anterior (Splish)

Solución del serial de un crackme empaquetado (Sambo)

- Uso de los mensajes de Windows

Capitulo XVI

Crackmes con serial variable

- Crackme de CrueHead
- Crackme Splish parte de name/serial
  - Instrucción CDQ
  - Instrucción IDIV ESI

Capitulo XVII

Solución del crackme propuesto en el capitulo anterior (Mexcrk1)

Juego Canasta v5.0*

*En este programa el botón ‘OK’ para ingresar el serial está desactivado.

Capitulo XVIII

Uso de los mensajes de Windows para pescar seriales (WM_KEYUP)

- Crackme de Stzwei (crackme_4stz)

Capitulo XIX

Detección del debugger

- API IsDebuggerPresent
  - Localización del byte correspondiente
  - Parchando el crackme para no ser detectados
  - Cambiando el flag para no ser detectados
  - Ocultar el debugger con un plugin (HideDebugger 1.23f)

Capitulo XX

Detección del OllyDbg por medio del nombre del proceso

- API OpenProcess
- API EnumProcessesModules
- API GetModuleBaseName

Uso de la API GetProcAddress

Capitulo XXI

Más métodos antidebugging

- API CreateToolhelp32Snapshot
- API Process32First
- API Process32Next
- API TerminateProcess
- API FindWindow
- API EnumWindows

Capitulo XXII

Antidebugging

Excepciones no manejadadas por OllyDbg (Unhandled Exceptions)
API SetUnhandledExceptionFilter
API ZwQueryInformationProcess
API UnhandledExceptionFilter

Capitulo XXIII

Fin del estudio de métodos antidebugging

- Ubicación de los bytes:
  - NTGloblalFlag
  - HeapFlags

API OutDebugString

Capitulo XXIV

Solución al crackme del capitulo anterior (antisocial)
- Usando plugins y sin plugins

Capitulo XXV

Manejo de excepciones
- Concepto de excepción

Diferentes tipos de excepciones
- Acceso a memoria no válida
- División entre cero (0)
- Instrucción no valida intento de ejecución de instrucción privilegiada

¿Qué ocurre cuando se genera una excepción?

¿Qué es el SEH?

¿Como se instala un manejador de excepciones?

¿Cómo cambiar el permiso de las secciones desde olly?

Capitulo XXVI

Uso del olly parcheado 5 (olly parcheado para buscar OEPs)

Cracking en Visual Basic (VB)
- **Significado de las partes del nombre de una API de VB**
  - **Ejemplos de APIs de:**
    - Conversión de datos
    - Mover datos
    - Comparaciones
    - Matemáticas
    - Misceláneas

Capitulo XXVII

¿Cómo está compuesto un .exe hecho en VB?
- El método del 4C

Capitulo XXVIII

La guerra total
- **Otra foma de eliminar nags en VB**
  - Injertando en la .dll de VB

Capitulo XXIX

P-Code (Pseudo-Code)

- Reconocimiento, estudio y cracking (la API MethCallEngine)
- Algunos OPCODES

Capitulo XXX

Fin del estudio de programas en P-CODE

- Más OPCODES
- Solución del crackme dejado en el capítulo anterior

Capitulo XXXI

Nociones iniciales de desempaquetado (desempacado)

- ¿Para que se empaca un programa?
- Esquema del funcionamiento de un programa empacado
- Concepto de OEP
- ¿Qué es un loader y como funciona? (una explicación breve)

Capitulo XXXII

Método de trabajo con archivos empacados
Métodos para llegar al OEP

- Mirar o buscar opcodes en el listado muerto del programa empacado antes de ejecutar
- Usar el buscador de OEPs que tiene el olly
- Usar el olly parchado 5 (olly parcheado para buscar OEPs)
- Método del pushad
- OEPs en programas hechos con VB (native o P-Code)
- Método de las excepciones
- Usar alguna API muy usada por el empacador
- Método de la primera API ejecutada por el programa

Capitulo XXXIII

¿Qué es la IAT?
¿Cómo la llena el sistema?

¿Qué es la IT?

Uso de LordPE (para dumpear)

Capitulo XXXIV

Desempacado manual de UPX y reparación de la IAT

- ¿Cómo reparar la IAT?
- Uso de PE Tools (para dumpear)

Capitulo XXXV

Desempacado de AsPack 2.12

- Uso de OIllyDump (para dumpear)
- Uso de Import REConstructor (para reparar la IAT)

Capitulo XXXVI

Desempacado de Crunch 5.0.0 o Bit-Arts

Desempacado de tElock 0.98b1

- Introducción a las entradas de la IAT redireccionadas

Capitulo XXXVII

Métodos para reparar entradas de la IAT redireccionadas

- **Método manual**
  - Uso del traceador condicional de olly
- Uso de los plugins del Import REConstructor
- Uso de los traceadores genéricos del propio Import REContructor
- Uso del JMP – CALL mágico (salto mágico)

Capitulo XXXVIII

Desempacado de yoda's Protector 1.3

Capitulo XXXIX

Uso del olly parcheado 4

Desempacado de PELock1.06.d (1ª parte)

Logueo propio de las excepciones

Estudio de los stolen bytes

- ¿Cómo reconocer si existen stolen bytes?
- Arreglo de código con stolen bytes mediante binary copy y binary paste

Stolen code

Capitulo XL

Scripts para tElock y UPX

Capitulo XLI

Desempacado de PELock1.06.d (2ª parte)

Métodos antidump

- Agregando una sección que crea el packer en tiempo de ejecución

Capitulo XLII

Desempacado de ACProtect1.09g.f con todas las funciones habilitadas (1ª parte)

- Usando un script que restaure los hardware breakpoints borrados por el empacador para llegar al OEP

Capitulo XLIII

Desempacado de ACProtect1.09g.f con todas las funciones habilitadas (2ª parte)

- Usando un script que repare la IAT

Capitulo XLIV

Desempacado de ACProtect1.09g.f con todas las funciones habilitadas (3ª parte)

- Arreglando los stolen bytes y el OEP
- Reparando los antidump

¿Cómo injertar si la sección a crear ya está ocupada? (Anexo parte 44)

Capitulo XLV

Desempacado de ReCrypt0.80*

Otro método usando Estricnina y Pokemon Anti Atach (Anexo escrito por Arapumk)**

*Este crackme no corre tan fácilmente en olly y Ricardo muestra algunos métodos que pueden servir en otros casos previo a mostrar la forma de lograr que corra en olly.
**Otra forma de atacar el mismo empacador. Arapumk muestra un método alternativo con la excelencia a la que nos tiene acostumbrados.

Capitulo XLVI

Un programa protegido con protecciones personalizadas, patrick.exe (1ª parte)

- Comprobación de que programa lo abrió (API Process323Next)
- Comprobación de si los módulos corresponden a Explorer.exe (API Module32First)
- Comprobación de si es el Explorer.exe mediante la ruta (API GetWindowsDirectory)
- Obtener la ruta del crackme (API GetModuleFileName)
- Verificación de cuantos procesos están corriendo en el crackme (API CreateMutex)
- Creación de un segundo proceso (API CreateProcess)

Capitulo XLVII

Un programa protegido con protecciones personalizadas, patrick.exe (2ª parte)

- Más protecciones y formas de pasarlas

- Corriendo el patrick.exe en olly sin tantas problemas

Usando un método propio para loguear las APIS que usa el programa

Capitulo XLVIII

Desempacado de PeSpin1.3.04.f (1ª parte)

- Reparación de los stolen bytes

Capitulo XLIX

Desempacado de PeSpin1.3.04.f (2ª parte)
Reparación de la IAT
Reparación de los antidump

Capitulo L

Desempacado de ReCrypt 0.80

- API OutputDebugString

Capitulo LI

Desempacado de ASProtect.2.3.04.26.a (1ª parte)*

- Uso de los plugins OllyBone y el Weasle
- ¿Qué es OllyBone y qué hace?

Problemas y otras dudas de la parte 51 (véase el comentario)

*Aquí se trabaja con drivers de sistema (.sys)

Capitulo LII

Desempacado de ASProtect.2.3.04.26.a (2ª parte)

- Solución al redireccionamiento mediante un script de Hiei

Capitulo LIII

Desempacado de TPPpack (versión desconocida)*

- Hallar el OEP y reparar el stolen code mediante un script de Ularteck

*Ricardo dentro de su tute publica la explicación a manera de tute que escribió el mismo Ularteck y se ayuda del tute de Marciano concurso 97 nivel 4 para la explicación de reparar la IAT.

Capitulo LIV

Desempacado de ExeCryptor2.2.50.a con compresión máxima de recursos/código /datos, sin protecciones habilitadas (1ª parte)

- Uso del TLS callback (formas de hallarlo con diferentes herramientas)

Capitulo LV

Desempacado de ExeCryptor2.2.50.a con compresión máxima de recursos/código /datos, sin protecciones habilitadas (2ª parte)

- Haciendo el script para reparar la IAT
- Dumpeado