Index of the training Reversing & Exploiting with Free Tools
by RICARDO NARVAJA (Translated by Fare9 & Arrizen)

EXERCISES:
https://docs.google.com/document/d/1OqD-AgsQzDWX9Y3v2cdQuf1xlYb_0GjJ8il9FuS1VUE/edit#

Part 1 - 18/10/2019

Installation of necessary tools.

Installing IDA Free
Installing Radare
Installing Ghidra
Installing X64DBG
Installing WINDBG
Installing Python 3
Installing Pycharm Community edition

https://docs.google.com/document/d/1w1xd7HpFoYXZjOWAjYX9-VLZG4KjX0-wLqhdQyMmcqs

Part 2 - 26/10/2019

Debugging With X64DBG

Bug definition
Vulnerability definition
Exploit definition
Buffer definition
Buffer Overflow definition
Definition of ampersand '&' (address of)
Debugging Stack1 exercise with X64DBG
Prologue and Epilogue in 32 bits architecture
Our definition of Horizon

https://docs.google.com/document/d/1kx6XrT5qbI8T_F1mkWw57RoY8YbN9XD1m6YDCuUycOI/

Part 3 - 6/11/2019

First steps about Static Reverse Engineering.

Binary Diffing Definition
Static Analysis of STACK1 with IDA FREE
Static Analysis of STACK1 with Radare2
Static Analysis of STACK1 with Radare2 using CUTTER GUI
Static Analysis of STACK1 with Ghidra

Installing DIA SDK
Static Analysis of STACK2 with IDA FREE
Static Analysis of STACK2 with RADARE2
Static Analysis of STACK2 with RADARE2 using CUTTER GUI
Static Analysis of STACK2 with Ghidra

https://docs.google.com/document/d/1Yb62bc85wTs4FS3jc1P_603UBStJcS673trkkTg142s

Part 4 - 20/11/2019

Defining the necessary concepts for stack exercises 3 and 4.

Defining what is an invalid character or bad char
Basic script to test bad chars.
Adapting basic script to test bad chars in Stack1
Solution for Stack3
Proposals for solving Stack4

https://docs.google.com/document/d/1eefm92s2iCSs9pBp2bpQ6zRvBvAt9-CB4Lf5OiKx4w8

Part 5 - 15/12/2019

Solving the Stack4 in IDA Free

Analyzing statically stack4 in IDA Free
Restriction for BAD CHARS
Static Representation of the stack in the main function
Acceptable solution of stack4 with crash
Professional Solution of stack4 without crash nor hardcoding

https://docs.google.com/document/d/1d-fVZfLwaPdgcuXbXL2u6xgEB4Duy-Us6_a3TPPbiUE

Part 6 - 24/03/2020

Solving the ABO1 with Radare

Solving the ABO1 Using RABIN2
Useful binaries included in Radare
RAHASH2
RAX2
RASM2
RADIFF2
Differences between virtual address and file offset
Static Reversing
Getting latest builds of Radare
Static map of the stack
Debugging
Shellcode with Resolver
Resolver of 32 bits.
Executing standalone with Shellcode

Using windbg
Thread Environment Block (TEB)
Process Environment Block (PEB)
IMAGE_DOS_HEADER
IMAGE_NT_HEADERS
IMAGE_OPTIONAL_HEADERS
IMAGE_DATA_DIRECTORY_ARRAY
IMAGE_EXPORT_DIRECTORY

https://docs.google.com/document/d/1ZzGAIvK7cKEuycDaPpUOT4vI-rYzjqMqac6vRJ2Zufs

Part 7 - 29/03/2020

Solving the ABO2 with GHIDRA.

Installing GHIDRA 9.1.1
Exploiting SEH in 32 bits.
TIB & TEB
EXCEPTION REGISTRATION RECORDS
POP POP RET
SCRIPT Get-PESecurity

https://docs.google.com/document/d/1drx6Jm4RGpJDDSxq1pSdM-_1dsHyTeHxElqxFHcAPcc

Part 8 - 29/03/2020

Solving ABO3 with IDA Free.

MS DIA SDK ERROR
Installing Visual C++ 2008 with redistributable
Source code
Static Analysis of ABO3
Solution Script

https://docs.google.com/document/d/1uiynmHHPHriZUTsv8PVHByj1Qdp7HN5r

Part 9 - 29/03/2020

Solving ABO4 with RADARE.

Updating RADARE & CUTTER
Reversing ABO4 using RADARE
Debugging with x64dbg
Reversing of ABO4 using CUTTER

https://docs.google.com/document/d/1J3nWanwLT_5siHgTNdQD685QTDnQrnPU/

Part 10 - 09/04/2020

What are the mitigations?
What is DEP?
What is Wow64?
What is SetDEPProcessPolicy?
Using PROCESS EXPLORER
What is ROP?
What are GADGETS?
Let's reverse the 32-bit executable with DEP in GHIDRA.
Script To Use If The Program Has No Dep Enabled
Minirop. For Practice
Using Rp ++

https://docs.google.com/document/d/1-_GO6z7mK3GryGHss6xqY8rxYvr_mABA6b9n-kAiDio/edit

Part 11 - 11/04/2020

Roping Step By Step
What Is Aslr?
Questions:
1) Do you have modules that do not have ASLR?
2)Does the process have the VirtualAlloc or VirtualProtect function imported into any module that does not have ASLR enabled?
3) Is the data already located in the stack to start roping?
4) There are no invalid characters? Can I use any character I want?
Let's look at the VirtualAlloc function on MSDN.
Roping Without Happy Ending
Roping With A Happy End
Bypassing Dep

https://docs.google.com/document/d/1YrOneOB8hy2sOMsWDp57boanskhcUbL8SIyrq1yahU/edit?usp=drivesdk

Part 12 - 12/04/2020

Starting with 64 bits
Calling Conventions in 32 bits
Reverse Order
Calling Convention __CDECL
Calling Convention __STDCALL

Calling Convention in 64 bits
Microsoft x64 Calling Convention
Registers in 64 bits
Reversing in 64 bits
Resolving the exercise of 64 bits: the first part of the ROP

https://drive.google.com/open?id=1n9boXgqxOSQ8IyE1FhSHXju53D412dIAvzGPbpMCSPQ

PARTE 13 - 23/03/2020

Resolution Of The 64 Bit Exercise
Script With Complete Solution
Resolver 64 Bits
64 Bit Structures
Finding The Image Base Of Kernel32
Finding The Address Of Winexec
Calling Winexec

https://drive.google.com/open?id=13mE2nXuuruJqDXgFQRdN28obKL-OL-3nZ7tJltFu6BY

Part 14 - 18/04/2020

Analyzing The Difficulty Of The Second Exercise Of 32 Bits
Answering Questions
1) Do You Have Modules That Do Not Have Aslr?
2) Do You Have Imported Virtualalloc Or Virtualprotect In Any Module That Does Not Have Aslr?
3) Is The Data Already Located In The Stack To Start Roping?
4) Can I Pass Any Character, That Is, There Are No Invalid Characters?
Final Score
A Tip To Help you To Solve.

https://drive.google.com/open?id=13oo98CBdNoPGm0cwcu2VU2ZYv4tau8iXfQveRMQEmUs

Part 15 - 26/04/2020

Reversing Exercise 2 - 32 Bits With Radare
Open Stored Project (PO)
Renaming Functions (AFN)
Questions About A Command (?)
Navigating Menus (M)

Window Mode(w)
Searching
Finding References.
Searching Gadgets
Reversing Statically
HOUSTON I HAVE A PROBLEM.
Basic Script To Start Roping

https://docs.google.com/document/d/1JyRNHI-g7BU3Z_S6WL4REK8MnA9wX0RP91yR5W1suOo/edit#

Part 16 - 05/11/2020

SEARCHING FOR GADGETS WITH RADARE
TESTING RADARE DEBUGGER
HOUSTON, WE HAVE A PROBLEM.
SUFFERING ENOUGH.
RUNNING A DIFFERENT SHELLCODE
SCRIPT FINAL WITH SOLUTION FOR PYTHON 3

https://docs.google.com/document/d/1wNSJn4EV9aUFBJTxVzW58sU-Yh0u6N-zDSkFOjY8o8I/edit
.

Part 17 - 23/05/2020

REVERSING THE 64 BITS EXERCISE WITH GHIDRA
DECOMPILING WITH GHIDRA
ROPING IN 64 BITS
MICROSOFT TRICKS
NOTE TO MICROSOFT
MORE TRICKS FROM MICROSOFT: DOES NOT WORK WINEXEC.
WORKING DESPITE EVERYTHING

https://t.co/INae9WP81M?amp=1

Part 18 - 02/06/2020

IDC SCRIPTING
SCRIPT TO FIND GADGETS
FINAL SCRIPT

https://docs.google.com/document/d/1Z3G4Xazm99u89RRqEv_N9YNFUpPQEzaSTooEPX0huqM/edit?usp=sharing