



Professional Information Security Training and Services

OFFENSIVE
security
www.offensive-security.com

Advanced Windows Exploitation Techniques

Matteo Memelli

Alexandru Uifalvi

Igor Frankovic

All rights reserved to Offensive Security, 2017 ©

No part of this publication, in whole or in part, may be reproduced, copied, transferred or any other right reserved to its copyright owner, including photocopying and all other copying, any transfer or transmission using any network or other means of communication, any broadcast for distant learning, in any form or by any means such as any information storage, transmission or retrieval system, without prior written permission from the author.

This page intentionally left blank.

Table of Contents

Module 0x00 Introduction	8
Module 0x01 Custom Shellcode Creation.....	9
Lab Objectives	9
<i>Overview</i>	9
System Calls and “The Windows Problem”	10
Talking to the Kernel.....	11
Finding kernel32.dll: PEB Method.....	12
<i>Exercise</i>	18
Resolving Symbols: Export Directory Table Method	19
<i>Working with the Export Names Array</i>	20
<i>Computing Function Names Hashes</i>	23
<i>Fetching Function's VMA</i>	26
<i>Exercise</i>	29
NULL free Position Independent Shellcode (PIC).....	30
<i>Exercise</i>	31
WriteFile Shellcode	32
<i>Exercise</i>	37
Wrapping Up.....	39
Module 0x02 DEP/ASLR/EMET Bypass and Sandbox Escape via Flash HeapSpray.....	40
Lab Objectives	40
<i>Overview</i>	40
Ret2Lib Attacks and Their Evolution	43
Return Oriented Programming Exploitation	43
ASLR	49
Debugger automation: Pykd and findrop.py.....	50
<i>Exercise</i>	57
Flash Player Heap Internals Key Points	58
Heap Spray: The Technique	62
<i>Exercise</i>	69
Heap Overflow Case Study: CVE-2015-3104 POC	70
<i>Exercise</i>	73
Heap Overflow Case Study: A Deeper Look at the Bug.....	74
<i>Exercise</i>	78
Heap Overflow Case Study: Allocation Control.....	79

<i>Exercise</i>	86
Heap Overflow Case Study: Gaining Read/Write access to the memory space.....	87
<i>Exercise</i>	92
Heap Overflow Case Study: Finding our object in memory.....	93
<i>Exercise</i>	93
Heap Overflow Case Study: Defeating ASLR	94
<i>Exercise</i>	96
Heap Overflow Case Study: Gaining code execution	97
<i>Exercise</i>	103
Heap Overflow Case Study: Stack Pivoting.....	104
<i>Exercise</i>	104
Heap Overflow Case Study: Defeating DEP.....	106
<i>Exercise</i>	108
<i>Exercise</i>	109
<i>Exercise</i>	113
Restoring the execution flow	114
<i>Exercise</i>	115
Sandbox Escape.....	116
<i>Exercise</i>	126
Enhanced Mitigation Experience Toolkit (EMET).....	127
Testing EMET 5.52 Protections on CVE-2015-3104.....	128
Disable vs Bypass.....	130
Disarming EMET: Theory	132
Disabling EMET: Practice (CVE-2015-3104)	135
<i>Exercise</i>	136
Defeating EAF	138
<i>Exercise</i>	139
Wrapping Up.....	140
Module 0x03 Kernel Drivers Exploitation (32-bit)	141
Lab Objectives.....	141
<i>Overview</i>	141
Windows I/O System and Device Drivers.....	141
Communicating with Drivers.....	142
I/O Control Codes	143
Privilege Levels and Ring0 Payloads.....	143
Token Stealing Payload	145
SEP Case Study: Kernel Pool Overflow	150

SEP Case Study: Vulnerability Overview	151
SEP Case Study: Way Down in ring0 Land.....	156
SEP Case Study: Bypassing Device Driver Checks.....	162
<i>Exercise</i>	163
SEP Case Study: Triggering the Overflow.....	164
<i>Exercise</i>	178
SEP Case Study: Allocation Control.....	179
<i>Exercise</i>	190
SEP Case Study: Object Header Manipulation.....	191
<i>Exercise</i>	193
SEP Case Study: The Header Issue	194
<i>Exercise</i>	195
SEP Case Study: The Quota Issue	196
<i>Exercise</i>	196
SEP Case Study: EIP Hunting	197
<i>Exercise</i>	208
SEP Case Study: Elevation.....	209
<i>Exercise</i>	215
<i>Extra Mile</i>	215
Wrapping up	216
Module 0x04 64-bit Kernel Driver Exploitation	217
Lab Objectives	217
<i>Overview</i>	217
64-bit Address Space.....	218
64-bit Main Enhancements.....	220
Windows-On-Windows Emulation	222
64-bit Exploitation: General Concepts.....	224
CVE-2015-5736 Case Study: Vulnerability Overview.....	226
CVE-2015-5736 Case Study: Overwriting the callback function.....	227
<i>Exercise:</i>	233
CVE-2015-5736 Case Study: Triggering the Vulnerable Code	234
<i>Exercise:</i>	237
CVE-2015-5736 Case Study: SMEP says hello.....	238
<i>Exercise</i>	239
CVE-2015-5736 Case Study: Introduction to memory paging and structures	240
<i>Exercise:</i>	247
CVE-2015-5736 Case Study: The PML4 self-reference entry.....	248

CVE-2015-5736 Case Study: Stack Pivoting.....	252
<i>Exercise:</i>	258
CVE-2015-5736 Case Study: Bypassing SMEP	259
<i>Exercise:</i>	263
CVE-2015-5736 Case Study: Restoring the execution flow.....	264
<i>Exercise:</i>	266
CVE-2015-5736 Case Study: PML4 self-reference entry randomization	267
CVE-2015-5736 Case Study: GDI/User objects and Read/Write primitives	268
CVE-2015-5736 Case Study: Accelerator Table allocation size.....	270
CVE-2015-5736 Case Study: Leaking the Accelerator Table kernel address	276
CVE-2015-5736 Case Study: Bitmap allocation size	280
CVE-2015-5736 Case Study: Leaking the Bitmap object kernel address	286
CVE-2015-5736 Case Study: Understanding the pvScan0 Read/Write primitive.....	291
<i>Exercise:</i>	294
CVE-2015-5736 Case Study: Overwriting the pvScan0	295
<i>Exercise:</i>	299
CVE-2015-5736 Case Study: Restoring the execution flow using ROP gadgets.....	300
CVE-2015-5736 Case Study: Flipping the U/S bit	313
<i>Exercise:</i>	314
CVE-2015-5736 Case Study: Optimisation – Increasing allocation reliability.....	315
CVE-2015-5736 Case Study: Optimisation – Monitoring Disk Usage	317
Wrapping Up.....	318