



Professional Information Security Training and Services

**OFFENSIVE**  
**security**  
[www.offensive-security.com](http://www.offensive-security.com)

# Advanced Windows Exploitation Techniques

Matteo Memelli

Alexandru Uifalvi

All rights reserved to Offensive Security, 2015 ©

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

|   |           |
|---|-----------|
| <b>Module 0x00 Introduction .....</b>                       | <b>7</b>  |
| <b>Module 0x01 Custom Shellcode Creation.....</b>           | <b>8</b>  |
| Lab Objectives .....  | 8         |
| <i>Overview</i> .....                                       | 8         |
| System Calls and “The Windows Problem” .....                | 9         |
| Talking to the Kernel.....                                  | 10        |
| Finding kernel32.dll: PEB Method.....                       | 11        |
| <i>Exercise</i> .....                                       | 17        |
| Resolving Symbols: Export Directory Table Method .....      | 18        |
| <i>Working with the Export Names Array</i> .....            | 19        |
| <i>Computing Function Names Hashes</i> .....                | 22        |
| <i>Fetching Function's VMA</i> .....                        | 25        |
| <i>Exercise</i> .....                                       | 27        |
| MessageBox Shellcode.....                                   | 28        |
| <i>Exercise</i> .....                                       | 30        |
| Position Independent Shellcode (PIC).....                   | 31        |
| <i>Exercise</i> .....                                       | 32        |
| Wrapping Up.....  | 33        |
| <b>Module 0x02 DEP/ASLR Bypass and Heap Spraying .....</b>  | <b>34</b> |
| Lab Objectives .....  | 34        |
| <i>Overview</i> .....                                       | 34        |
| Ret2Lib Attacks and Their Evolution .....                   | 37        |
| Return Oriented Programming Exploitation .....              | 37        |
| ASLR .....  | 43        |
| Debugger automation: Pykd and findrop.py.....               | 44        |
| <i>Exercise</i> .....                                       | 51        |
| JavaScript Heap Internals Key Points .....                  | 52        |
| Heap Spray: The Technique .....                             | 55        |
| Heap Spray Case Study: CVE-2011-2371 POC.....               | 60        |
| <i>Exercise</i> .....                                       | 63        |
| Heap Spray Case Study: A Deeper Look at the Bug.....        | 64        |
| Heap Spray Case Study: Mapping the Object in Memory .....   | 66        |
| <i>Exercise</i> .....                                       | 71        |
| Heap Spray Case Study: Controlling the Execution Flow ..... | 72        |

|   |            |
|---|------------|
| <i>Exercise</i> .....   | 75         |
| Heap Spray Case Study: Stack Pivoting .....                   | 76         |
| <i>Exercise</i> .....   | 79         |
| Heap Spray Case Study: Pointers Stunts.....                   | 80         |
| <i>Exercise</i> .....   | 85         |
| Heap Spray Case Study: When 1bit = Shell.....                 | 86         |
| <i>Exercise</i> .....   | 89         |
| Heap Spray Case Study: The Hard Way.....                      | 90         |
| <i>Exercise</i> .....   | 95         |
| <i>Exercise</i> .....   | 100        |
| Defeating DEP .....   | 101        |
| <i>Exercise</i> .....   | 102        |
| <i>Exercise</i> .....   | 103        |
| <i>Exercise</i> .....   | 108        |
| Enhanced Mitigation Experience Toolkit (EMET).....            | 109        |
| Testing EMET 5.2 Protections on CVE-2011-2371 .....           | 110        |
| Disable vs Bypass.....  | 112        |
| Disarming EMET: Theory .....                                  | 113        |
| Disabling EMET: Practice (CVE-2011-2371) .....                | 116        |
| <i>Exercise</i> .....   | 117        |
| Defeating EAF .....   | 120        |
| <i>Exercise</i> .....   | 121        |
| <i>Extra Mile</i> .....                                       | 122        |
| Wrapping Up.....  | 123        |
| <b>Module 0x03 Kernel Drivers Exploitation (32-bit) .....</b> | <b>124</b> |
| Lab Objectives .....  | 124        |
| <i>Overview</i> .....   | 124        |
| Windows I/O System and Device Drivers .....                   | 124        |
| Communicating with Drivers.....                               | 125        |
| I/O Control Codes .....                                       | 126        |
| Privilege Levels and Ring0 Payloads.....                      | 126        |
| Token Stealing Payload .....                                  | 129        |
| SEP Case Study: Kernel Pool Overflow .....                    | 134        |
| SEP Case Study: Vulnerability Overview .....                  | 135        |
| SEP Case Study: Way Down in ring0 Land.....                   | 140        |
| SEP Case Study: Bypassing Device Driver Checks.....           | 146        |
| <i>Exercise</i> .....   | 147        |

|   |            |
|---|------------|
| SEP Case Study: Triggering the Overflow.....                | 148        |
| <i>Exercise</i> .....                                       | 162        |
| SEP Case Study: Allocation Control.....                     | 163        |
| <i>Exercise</i> .....                                       | 176        |
| SEP Case Study: Object Header Manipulation.....             | 177        |
| <i>Exercise</i> .....                                       | 180        |
| SEP Case Study: The Header Issue .....                      | 181        |
| <i>Exercise</i> .....                                       | 182        |
| SEP Case Study: The Quota Issue .....                       | 183        |
| <i>Exercise</i> .....                                       | 183        |
| SEP Case Study: EIP Hunting .....                           | 184        |
| <i>Exercise</i> .....                                       | 196        |
| SEP Case Study: Elevation.....                              | 197        |
| <i>Exercise</i> .....                                       | 203        |
| <i>Extra Mile</i> .....                                     | 204        |
| Wrapping up .....   | 205        |
| <b>Module 0x04 64-bit Kernel Driver Exploitation .....</b>  | <b>206</b> |
| Lab Objectives .....  | 206        |
| <i>Overview</i> .....                                       | 206        |
| 64-bit Address Space.....                                   | 207        |
| 64-bit Main Enhancements.....                               | 209        |
| Windows-On-Windows Emulation .....                          | 211        |
| 64-bit Exploitation: General Concepts.....                  | 213        |
| MS14-058 Case Study: Vulnerability Overview.....            | 215        |
| MS14-058 Case Study: Triggering the Vulnerable Code.....    | 219        |
| <i>Exercise</i> .....                                       | 231        |
| MS14-058 Case Study: Mapping our Path.....                  | 232        |
| <i>Exercise</i> .....                                       | 236        |
| MS14-058 Case Study: “Arbitrary” Kernel Overwrite.....      | 237        |
| <i>Exercise</i> .....                                       | 240        |
| MS14-058 Case Study: Escalating Privileges (Theory) .....   | 241        |
| <i>Exercise</i> .....                                       | 246        |
| MS14-058 Case Study: Escalating Privileges (Practice) ..... | 247        |
| <i>Exercise</i> .....                                       | 261        |
| <i>Extra Mile</i> .....                                     | 261        |
| Wrapping up .....   | 262        |