

Offensive Development Practitioner Lab Guide





Table of Contents

TABLE OF CONTENTS	2
LICENSING	11
1.1 FORTRA LICENSE DISCLAIMER	11
2 INTRODUCTION	12
2.1 ABOUT THE OFFENSIVE DEVELOPMENT PRACTITIONER (ODP) CERTIFICATION COURSE	12
2.2 COURSE APPROACH	12
2.3 COURSE SUPPORT	12
3 DEPLOYING THE LAB ENVIRONMENT VIA TERRAFORM	13
3.1 INTRODUCTION	13
3.2 DECLARING STATIC CREDENTIALS IN THE PROVIDER.TF FILE	14
3.3 REQUESTING QUOTA INCREASES FOR VM RESOURCES	19
3.4 SHUTTING DOWN / MANAGING COSTS FOR ODPC LAB INSTANCES WITH TERRAFORM	21
4 GUACAMOLE WALKTHROUGH	23
4.1 CONSOLE SESSIONS	23
4.2 HOW TO UPLOAD FILES FROM THE HOST TO THE GUACAMOLE ENVIRONMENT (CONSOLE SESSION)	24
4.3 HOW TO DOWNLOAD FILES FROM THE GUACAMOLE ENVIRONMENT TO THE HOST (CONSOLE SESSION)	26
4.4 HOW TO USE THE GUACAMOLE CLIPBOARD	27
4.5 LAB ENVIRONMENT	29
4.6 DEVELOPMENT	30
5 PORTABLE EXECUTABLE (PE) PRIMER	31
5.1 PE FORMAT	31
5.2 LAB	37
6 WINDOWS API PRIMER	41
6.1 BACKGROUND	41
6.2 LAB	43
6.3 RESOLVING ISSUES	47
6.4 EXERCISES	49
7 CONVERTING PE FILES TO SHELLCODE	50



8 FINDING THE EDR ACTIVE PROTECTION DLL	52
8.1 NtWriteProcessMemory is Hooked	52
8.2 NtWriteProcessMemory is Not Hooked	61
8.3 EXERCISES	61
9 PROCESS INJECTION: CREATEREMOTEThread	62
9.1 PROCESS INJECTION PRIMER	62
9.2 VirtualAllocEx()	63
9.3 WriteProcessMemory()	65
9.4 CreateRemoteThread()	67
9.5 Shellcode Generation	68
9.6 Encoding Shellcode	69
9.7 PROCESS INJECTION VS EDRs	71
9.8 EXERCISES	72
10 SHELLCODE STORAGE (TEXT SECTION)	73
10.1 CODE	74
10.2 HUNTING OUR SHELLCODE	77
10.3 EXERCISES	81
11 SHELLCODE STORAGE (RESOURCES SECTION)	82
11.1 RESOURCES PRIMER	82
11.2 CREATING RESOURCES	82
11.3 FINDING OUR RESOURCE	83
11.4 COMPILING RESOURCES	84
11.5 HUNTING OUR SHELLCODE	85
11.6 EXERCISES	86
12 PROCESS INJECTION: PROCESS HOLLOWING	87
12.1 PROCESS HOLLOWING INTRODUCTION	87
12.2 CODE ANALYSIS	89
12.3 LAB	97
12.4 EXERCISES	104
13 PROCESS INJECTION: EARLY BIRD	105
13.1 EARLY BIRD INTRODUCTION	105
13.2 Is QueueUserAPC Hooked Nowadays?	107
13.3 ADVANCED LOGGING WITH SYSMON	109
13.4 EXERCISES	114



14 HIDING IMPORTS VIA DYNAMIC RESOLUTION	115
14.1 WINDOWS API DYNAMIC RESOLUTION PRIMER	115
14.2 EXERCISES	118
15 WALKING THE EAT AND THE PEB	119
15.1 EAT PRIMER	119
15.2 WALKING THE EAT	120
15.3 FORWARDERS	122
15.4 AVOIDING LOADLIBRARY AND GETMODULEHANDLE	122
15.5 TESTING THE CODE	128
15.6 EXERCISES	129
16 DYNAMIC RESOLUTION WITH HASHES	130
16.1 WHY HASHES?	130
16.2 THE CODE	130
16.3 EXERCISES	136
17 INTRODUCTION TO COBALT STRIKE	137
17.1 COBALT STRIKE INTRODUCTION	138
17.2 MAIN COMPONENTS	139
17.3 CLIENT VIEW	141
17.4 STAGERS (OPSEC UNSAFE)	142
17.5 REDIRECTORS	143
17.6 COBALT STRIKE TEAM SERVER CONFIGURATION	143
17.7 GETTING YOUR FIRST BEACON	145
17.8 BEACON INTERACTION	150
17.9 EXERCISES	153
18 COBALT STRIKE: PROFILES AND BOFS	154
18.1 COBALT STRIKE PROFILES	154
18.2 HTTP CONFIG	155
18.3 TLS CERTS	156
18.4 GET SECTION	157
18.5 KNOW YOUR TOOLS	158
18.6 POST SECTION	161
18.7 POST-EXPLOITATION	162
18.8 PROFILE VARIANTS	166
18.9 COBALT STRIKE BOFS	168
18.10 WRITING YOUR FIRST BOF	168
18.11 AGGRESSOR SCRIPTS AND BOFS	171



18.12 EXERCISES	178
<u>19 INTRODUCTION TO HAVOC C2 FRAMEWORK</u>	<u>179</u>
19.1 INTRODUCTION TO HAVOC C2 FRAMEWORK	179
19.2 HAVOC CLIENT OVERVIEW: SESSIONS, EVENT VIEWER, AND TEAM CHAT	183
19.3 GENERATING AND USING A CUSTOM HAVOC C2 PROFILE	194
19.4 EXERCISES	199
<u>20 WINDOWS API DIRECT SYSCALLS</u>	<u>200</u>
20.1 INTRODUCTION	200
20.2 FINDING THE SSN	201
20.3 THE CODE	203
20.4 EXERCISES	206
<u>21 WINDOWS API INDIRECT SYSCALLS</u>	<u>207</u>
21.1 INTRODUCTION	207
21.2 FINDING THE SYSCALL INSTRUCTION ADDRESS	208
21.3 THE CODE	210
21.4 EXERCISES	217
<u>22 DYNAMIC SSN RESOLUTION</u>	<u>218</u>
22.1 BACKGROUND	218
22.2 HALOSGATE	218
22.3 THE CODE	221
22.4 WEAPONIZING HALOSGATE	227
22.5 EXERCISES	230
<u>23 API UNHOOKING</u>	<u>231</u>
23.1 OBSERVING THE PATCH	233
23.2 PATCHING THE PATCH – NTAPIs	237
23.3 PATCHING THE PATCH – GENERIC	241
23.4 EXERCISES	243
<u>24 PROCESS INJECTION: CARO-KANN</u>	<u>244</u>
24.1 EVENT TRACING FOR WINDOWS	244
24.2 KERNEL CALLBACK ROUTINES	246
24.3 CARO-KANN	246
24.4 PROJECT STRUCTURE	248
24.5 INJECTION TIME	257



24.6 DEFENDER'S ANALYSIS	258
24.7 EXERCISES	260
<u>25 PROCESS INJECTION: TP TIMER INSERTION</u>	<u>261</u>
25.1 PROCESS THREAD POOLS	261
25.2 ATTACKING THE THREAD POOL VIA ITS TIMER QUEUE	262
25.3 ANALYZING THE INJECTION	266
25.4 EXERCISES	268
<u>26 PROCESS INJECTION: EARLY CASCADE</u>	<u>269</u>
26.1 EARLY CASCADE INTRODUCTION	269
26.2 FINDING AN ALTERNATE CALLBACK POINTER	270
26.3 AVOIDING DEADLOCK	271
26.4 CONTROLLING SHIM ENGINE POINTERS	272
26.5 SELF APC INJECTION AS A STAGER	272
26.6 PROJECT STRUCTURE	273
26.7 INJECTION TIME	281
26.8 EXERCISES	283
<u>27 SUBVERT TRUST CONTROLS: CODE SIGNING</u>	<u>284</u>
27.1 INTRODUCTION TO CODE SIGNING	284
27.2 VERIFYING AND VIEWING AUTHENTICODE SIGNATURES	285
27.3 SIGNATURE TRANSPLANTS	288
27.4 INJECTING CODE IN SIGNATURES	293
27.5 LEGITIMATE SIGNING	297
27.6 INDIRECT SIGNED EXECUTION	299
27.7 EXERCISES	300
<u>28 MODULE STOMPING</u>	<u>301</u>
28.1 INTRODUCTION	301
28.2 THE CODE	301
28.3 FINDING THE DIFFERENCE	303
28.4 EXERCISES	304
<u>29 RETURN ADDRESS PATCHING</u>	<u>305</u>
29.1 BACKGROUND	305
29.2 THE CODE	308
29.3 EXERCISES	314
<u>30 STACK SPOOFING WITH SYNTHETIC FRAMES</u>	<u>315</u>



30.1 FUNCTION FRAMES	315
30.2 MANUALLY SYNTHESIZING OUR FRAMES	318
30.3 PROGRAMMATICALLY SYNTHESIZING OUR OWN FRAMES	322
30.4 SPOOFING THE STACK	325
30.5 EXERCISES	328
<u>31 ANTI-MALWARE SCAN INTERFACE (AMSI) BYPASS</u>	<u>329</u>
31.1 BYPASS AMSI WITH POWERSHELL	329
31.2 BYPASS AMSI WITH .NET	340
31.3 EXERCISES	346
<u>32 PAYLOAD GUARDRAILS</u>	<u>347</u>
32.1 IDENTIFYING THE DOMAIN	347
32.2 IDENTIFYING A TYPICAL USER WORKSTATION	348
32.3 VALIDATE EXECUTABLE NAME	348
32.4 EXPENSIVE CALCULATIONS	350
32.5 EXERCISES	351
<u>33 CUSTOM REFLECTIVE DLL LOADERS</u>	<u>352</u>
33.1 REFLECTIVE DLL LOADER INTRODUCTION	352
33.2 CUSTOM COBALT STRIKE REFLECTIVE DLL LOADERS	353
33.3 ELUSIVEMICE	355
33.4 ACELDR	358
33.5 HUNTING BEACONS	360
33.6 IAT HOOKING	361
33.7 EXERCISES	362
<u>34 DLL PROXYING</u>	<u>363</u>
34.1 INITIAL ACCESS WITH DLL PROXYING	364
34.2 LNK WEAPONIZATION	373
34.3 LOADER LOCK	379
34.4 EXERCISES	379
<u>35 P/INVOKE AND D/INVOKE</u>	<u>380</u>
35.1 MANAGED CODE	380
35.2 PINVOKE	380
35.3 DINVOKE	384
35.4 CSWHISPERS - INSTALLATION	385
35.5 CSWHISPERS – DINVOKE USAGE	387
35.6 CSWHISPERS – INDIRECT SYSCALL USAGE	389



35.7 EXERCISES	392
<u>36 APPDOMAIN MANAGER INJECTION</u>	<u>393</u>
36.1 BACKGROUND	393
36.2 COMPILING THE CODE	394
36.3 THE CONFIGURATION FILE	394
36.4 STRONG NAME SIGNING	395
36.5 UNDERSTANDING THE CODE	395
36.6 INJECTION TIME!	398
36.7 ANALYSIS	402
36.8 EXERCISES	404
<u>37 CLICKONCE PAYLOAD DEVELOPMENT</u>	<u>405</u>
37.1 SETTING UP OUR MACHINE	406
37.2 SEARCHING FOR A TARGET APP TO BACKDOOR	406
37.3 DOWNLOADING THE CLICKONCE APPLICATION	408
37.4 SEARCHING FOR WEAK PERMISSION DEPENDENCIES	411
37.5 ANALYZING THE APPLICATION'S WORKFLOW	414
37.6 INJECTING THE BACKDOOR	418
37.7 BUILDING THE BACKDOORED DEPLOYMENT PACKAGE	420
37.8 TESTING THE BACKDOORED DEPLOYMENT PACKAGE	424
37.9 EXERCISES	426
<u>38 WINDOWS KERNEL INTRODUCTION</u>	<u>427</u>
38.1 BACKGROUND	427
38.2 DEBUGGING THE WINDOWS KERNEL: LOCAL	428
38.3 DEBUGGING THE WINDOWS KERNEL: REMOTE	430
38.4 EXERCISES	430
<u>39 KERNEL TELEMETRY: KERNEL CALLBACK NOTIFICATIONS</u>	<u>431</u>
39.1 PROCESS, THREAD, IMAGE LOAD NOTIFICATION CALLBACKS	431
39.2 FINDING NOTIFY ROUTINE ARRAYS	432
39.3 FINDING AN ARRAY ENTRY'S DRIVER FUNCTION	433
39.4 BLINDING THE EDR TO NEW PROCESSES	435
39.5 TELEMETRY REVIEW	439
39.6 EXERCISES	439
<u>40 KERNEL TELEMETRY: EVENT TRACING FOR WINDOWS</u>	<u>441</u>
40.1 BACKGROUND	441
40.2 EVENT TRACING FOR WINDOWS THREAT INTELLIGENCE (ETWTI) PROVIDER	441



40.3 MONITORING ETW EVENTS	442
40.4 DISABLING ETW IN KERNEL LAND	444
40.5 EXERCISES	448
<u>41 KERNEL PROTECTIONS: PROCESS PROTECTION</u>	<u>449</u>
41.1 BACKGROUND	449
41.2 THE PROTECTION HIERARCHY	449
41.3 FINDING AND MANIPULATING PPL IN EPROCESS	450
41.4 EXERCISES	453
<u>42 ATTACKING THE KERNEL: BRING YOUR OWN VULNERABLE DRIVER (BYOVD)</u>	<u>454</u>
42.1 BACKGROUND	454
42.2 DRIVER SIGNATURE ENFORCEMENT (DSE)	455
42.3 KERNEL VULNERABILITIES	455
42.4 MICROSOFT PROTECTIONS: VULNERABLE DRIVER BLOCK LIST	455
42.5 SUBVERTING OUR LIMITATIONS: WEAPONIZING SIGNED, UNKNOWN VULNERABLE DRIVERS	456
42.6 INSTALLING THE VULNERABLE DRIVER	456
42.7 COMMUNICATING WITH THE DRIVER	457
42.8 ANALYZING THE VULNERABLE DRIVER POC EXPLOIT	457
42.9 A WORD ON THIS DRIVER	461
42.10 EXERCISES	461
<u>43 BYOVD: DISABLING NOTIFICATION CALLBACKS</u>	<u>462</u>
43.1 BACKGROUND & A WORD ON OFFSETS	462
43.2 CALCULATING OUR OFFSETS	462
43.3 EXPLOITING THE DRIVER TO REMOVE CALLBACKS	463
43.4 EXERCISES	464
<u>44 BYOVD: DISABLING KERNEL ETW PROVIDERS</u>	<u>465</u>
44.1 READ PRIMITIVE: WALKING THE STRUCTURES	465
44.2 WRITE PRIMITIVE: DISABLING THE ETWTI PROVIDER	466
44.3 EXERCISES	468
44.4 BYOVD CHALLENGE	469
<u>45 BYOVD: PORTING TO COBALT STRIKE</u>	<u>470</u>
45.1 STAGING OUR BOF	470
45.2 EXERCISE	475
<u>46 THE FINAL BINARY – YOUR LAST CHALLENGE</u>	<u>476</u>



46.1 CROWDSTRIKE	476
46.2 SOPHOS	476
46.3 BITDEFENDER	477
46.4 ELASTIC	477
46.5 CORTEX (PALO ALTO NETWORKS)	478
46.6 MICROSOFT DEFENDER FOR ENDPOINT (MDE)	478
<u>47 APPENDICES</u>	<u>479</u>
47.1 ABBREVIATIONS	479