

Índice

Índice	7
Prólogo	13
Descargo de responsabilidades	15
El Tao del Hacker	17
I. Una aproximación Zen.....	18
II. Hacking, Kung Fu y Ninjas	20
III. Qué nos depara el futuro.....	22
Capítulo 0 Introducción al exploiting	25
0.1. Requisitos previos	26
0.2. Un pequeño laboratorio	26
0.3. El mundo real	28
0.4. Wargames: Plataforma de aprendizaje	30
Capítulo I Stack Overflows: Un Mal Interminable	33
1.1. ¿Qué es un buffer overflow?	33
1.2. Fallos de segmentación (DoS).....	36
1.3. Motivos subyacentes	37
1.4. Aplicaciones Setuid (suid).....	43
1.5. Payloads.....	44
1.6. Su primer exploit	47
1.7. GDB: El debugger de Linux.....	49
1.8. Prácticas de programación segura	58
1.9. Solucionario Wargames.....	63

1.10. Dilucidación.....	67
1.11. Referencias.....	68
Capítulo II Shellcodes en arquitecturas IA32.....	69
2.1. Sintaxis AT&T vs Intel.....	69
2.2. ¿Qué es un shellcode?.....	71
2.3. Llamadas de sistema (syscalls).....	72
2.4. Métodos de referenciación.....	76
2.4.1. Viaje al pasado.....	77
2.4.2. Viaje al presente.....	79
2.4.3. Alternativa FNSTENV.....	80
2.5. Port binding.....	81
2.6. Conexión inversa.....	84
2.7. Egg Hunters.....	86
2.8. Shellcodes polimórficos.....	88
2.9. Dilucidación.....	94
2.10. Referencias.....	95
Capítulo III Atacando el Frame Pointer.....	97
3.1. Abuso del Frame Pointer.....	97
3.1.1. Análisis del problema.....	97
3.1.2. Ejecución de código.....	100
3.2. Off-by-One Exploit.....	105
3.2.1. Precondiciones.....	106
3.3. Dilucidación.....	108
3.4. Referencias.....	109
Capítulo IV Métodos Return to Libc.....	111
4.1. Prueba de concepto (PoC).....	111
4.1.1. Evasión de bytes <i>null</i>	115
4.1.2. Métodos interesantes.....	116
4.2. Exploits avanzados.....	117
4.2.1. Encadenamiento de funciones.....	117
4.2.2. Falseo de frames.....	119

4.3. Solucionario Wargames.....	123
4.4. Dilucidación	125
4.5. Referencias	126
Capítulo V Métodos complementarios.....	127
5.1. Técnica Ret to Ret	127
5.2. Técnica de Murat.....	131
5.3. Jump to ESP: Windows Style.....	133
5.4. ROP (Return Oriented Programming).....	136
5.5. Integer overflows.....	141
5.6. Variables no inicializadas.....	145
5.7. Exploits Remotos.....	146
5.8. Dilucidación	150
5.9. Referencias	150
Capítulo VI Explotando format strings	151
6.1. Análisis del problema	151
6.1.1. Leer de la memoria	153
6.1.2. Parámetro de acceso directo.....	153
6.1.3. Escribir en la memoria	154
6.2. Objetivos primarios	155
6.2.1. DTOR (Destruyores).....	155
6.2.2. GOT (Tabla de Offsets Global).....	156
6.3. Prueba de concepto.....	157
6.3.1. Cambios de orden.....	159
6.4. Format Strings como Buffer Overflows	160
6.5. Objetivos secundarios.....	161
6.5.1. Estructuras __atexit.....	161
6.5.2. setjmp() y longjmp()	166
6.5.3. VTable y VPTR en C++.....	167
6.6. Solucionario Wargames.....	168
6.7. Dilucidación	173
6.8. Referencias	174

Capítulo VII Medidas preventivas y evasiones.....	175
7.1. ASLR no tan aleatorio	175
7.2. StackGuard y StackShield.....	180
7.2.1. StackGuard	180
7.2.2. StackShield	181
7.3. Stack Smash Protector (ProPolice)	182
7.4. Relocation Read-Only (RELRO).....	188
7.5. Fortify Source	190
7.6. Reemplazo Libsafe.....	192
7.7. ASCII Armored Address Space	193
7.8. Jaulas con chroot()	195
7.9. Instrumentación de código	198
7.10. Rompiendo las Reglas: Todo en Uno	200
7.11. Dilucidación.....	208
7.12. Referencias.....	209
Capítulo VIII Heap Overflows: Exploits básicos	211
8.1. Un poco de Historia	211
8.1.1. ¿Qué es un Heap Overflow?.....	212
8.1.2. Convenciones	213
8.2. Algoritmo Malloc de Doug Lea.....	213
8.2.1. Organización del Heap	214
8.2.2. Algoritmo free().....	216
8.3. Técnica Unlink.....	217
8.3.1. Teoría.....	218
8.3.2. Componentes de un Exploit.....	219
8.4. Técnica Frontlink	222
8.4.1. Conocimientos previos	222
8.4.2. Explotación.....	223
8.5. Otros bugs: double free() y use after free().....	228
8.5.1 Double free()	228
8.5.2 Use after free().....	230
8.6. Peligros en los manejadores de señales	231
8.7. Solucionario Wargames	233

8.8. Dilucidación	236
8.9. Referencias	236
Capítulo IX Heap Overflows: Exploits avanzados	237
9.1. La muerte de Unlink	237
9.2. The House of Mind	238
9.2.1. Método Fastbin	246
9.3. The House of Prime	250
9.3.1. unsorted_chunks()	254
9.4. The House of Spirit	255
9.5. The House of Force	257
9.6. The House of Lore	260
9.6.1. Heap Debugging	260
9.6.2. Corrupción SmallBin	262
9.6.3. Corrupción LargeBin	268
9.7. Gestor de memoria seguro	272
9.7.1. Dnmalloc	274
9.7.2. OpenBSD Malloc	274
9.8. Heap Spraying y Heap Feng Shui	275
9.8.1 Heap Spraying	275
9.8.2 Heap Feng Shui	276
9.9. Dilucidación	277
9.10. Referencias	277
Capítulo X Explotación en espacio de kernel	279
10.1. ¿Dónde juegan los mayores?	280
10.2. Derreferencia de punteros nulos	282
10.3. Condiciones de carrera	285
10.4. Desbordamientos de buffer	286
10.5. Dilucidación	287
10.6. Referencias	288
Apéndice I Solucionario Nebula Wargame	289

Glosario de términos	317
Índice alfabético.....	319
Índice de imágenes	321
Libros publicados	325