Tartalomjegyzék

A szerzőről ..... **xxvii**

Bevezető helyett ..... **xxix**

Előszó ..... **xxxv**

Kinek szól a könyv? ..... **xxxv**
A könyv témaköre ..... **xxxvi**
Ami a könyv témakörén túlmutat ..... **xxxvii**

Köszönetnyilvánítás ..... **xxxix**

Információk ..... **xl**

I. rész: A támadó stratégiái

1. Bevezetés az élet játékaiba **3**

1.1. Az önlemásoló struktúrák korai modelljei **4**
1.1.1. Neumann János elmélete az önreprodukáló automatákról 4
1.1.2. Fredkin: szaporodó struktúrák 6
1.1.3. Conway: „Élet” játék 7
1.1.4. Core War: harci programok 11
1.2. A számítógép-vírusok születése **15**
1.3. Automatizált többszöröző kód: a számítógép-vírusok elmélete és definíciója **16**
Hivatkozások 19

2. A rosszindulatú programok elemzésének „varázsa” **21**

2.1. A víruskutatás jellemző sémái **23**
2.2. A vírusok elleni védelem fejlesztése **24**
2.3. A rosszindulatú programok szóhasználata **25**
2.3.1. Vírusok 25
2.3.2. Férgek 25
2.3.2.1. Levélférgek 26

2.3.2.2. Polipok 26
2.3.2.3. Nyulak 26
2.3.3. Logikai bombák 27
2.3.4. Trójai falovak 27
2.3.4.1. Hátsóajtó-programok (backdoor) 28
2.3.4.2. Jelszólopó vírusok 29
2.3.5. Baktériumok 29
2.3.6. Az alkalmazáshibát kihasználó vírusok 29
2.3.7. Letöltők 29
2.3.8. Tárcsázók 30
2.3.9. Dropperek 30
2.3.10. Injektorok 30
2.3.11. Automatizált jogosultságszerzők 31
2.3.12. Kitek (vírusgenerátorok) 31
2.3.13. Spammerprogramok 31
2.3.14. Flooderek 32
2.3.15. Billentyűnaplózók (keyloggerek) 32
2.3.16. Rootkitek 32
2.4. Egyéb típusok 33
2.4.1. Viccprogramok 33
2.4.2. Hoaxok: láncLevelek 33
2.4.3. Egyéb kártevők: hirdető- és kémprogramok 34
2.5. A rosszindulatú számítógépes programok nevezéktana 35
2.5.1. <családnév> 36
2.5.2. <kártevőtípus>:// 36
2.5.3. <platform>/ 36
2.5.4. <csoportnév> 37
2.5.5. <fertőzési_hossz> 37
2.5.6. <változat> 37
2.5.7. [<mutáció>] 37
2.5.8. <módosítók> 37
2.5.9. :<lokalizációazonosító> 38
2.5.10. #<tömörítő> 38
2.5.11. @m vagy @mm 38
2.5.12. !<terjesztőspecifikus_megjegyzés> 38
2.6. A hivatalosan elismert platformnevek annotált listája 38
Hivatkozások 42

3. A kártékony kódok környezete 45

3.1. Függés a számítógép architektúrájától 47
3.2. Processzorfüggőség 48
3.3. Függés az operációs rendszertől 49
3.4. Függés az operációs rendszer verziójától 50

3.5. Fájlrendszerfüggőség 51
3.5.1. Klasztervírusok 51
3.5.2. NTFS-folyamokat fertőző vírusok 53
3.5.3. NTFS tömörítési vírusok 53
3.5.4. Lemezképfájl-fertőzés 54
3.6. Fájlformátum-függőség 54
3.6.1. COM-vírusok a DOS rendszeren 54
3.6.2. EXE-vírusok a DOS rendszeren 54
3.6.3. NE- (New Executable) vírusok 16 bites Windows és OS/2 rendszereken 55
3.6.4. LX-vírusok az OS/2 rendszeren 55
3.6.5. PE-vírusok a 32 bites Windows rendszereken 56
3.6.5.1. A DLL-vírusok 57
3.6.5.2. Natív vírusok 58
3.6.6. ELF-vírusok a UNIX rendszereken 59
3.6.7. Meghajtóprogram-vírusok 60
3.6.8. Tárgykód- és LIB-vírusok 60
3.7. Függés az értelmezett környezettől 61
3.7.1. A Microsoft-alkalmazások makróvírusai 61
3.7.1.1. Makrórongálódás 64
3.7.1.2. Makrókonverzió 66
3.7.1.3. Nyelvfüggőség 66
3.7.1.4. Makróvírusok platformfüggősége 68
3.7.1.5. Makróevolúció és visszafejlődés 69
3.7.1.6. Forráskód, p-kód, végrehajtható kód 70
3.7.1.7. Osztódásos fertőzési stratégiát használó makróvírusok 71
3.7.1.8. Új képlet 72
3.7.1.9. Felhasználói makrók megfertőzése 72
3.7.1.10. Új fájlformátumok: XML 72
3.7.2. REXX-vírusok az IBM rendszereken 73
3.7.3. DCL- (DEC Command Language –DEC parancsnyelv) vírusok a DEC/VMS rendszereken 74
3.7.4. UNIX-shellszkriptek (csh, ksh és bash) 75
3.7.5. VBScript- (Visual Basic Script) vírusok a Windows rendszereken 76
3.7.6. BATCH-vírusok 76
3.7.7. Instant üzenetküldő vírusok az mIRC és a pIRCH szkriptekben 78
3.7.8. SuperLogo vírusok 78
3.7.9. JScript-vírusok 80
3.7.10. Perl-vírusok 80

3.7.11. HTML-levélbe ágyazott JellyScript nyelvű WebTV-férgek 81
3.7.12. Python-vírusok 81
3.7.13. VIM-vírusok 82
3.7.14. EMACS-vírusok 82
3.7.15. TCL-vírusok 82
3.7.16. PHP-vírusok 82
3.7.17. MapInfo-vírusok 83
3.7.18. ABAP-vírusok a SAP rendszereken 83
3.7.19. Windows-súgófájlvírusok – amikor megnyomjuk az F1-et 84
3.7.20. JScript-vírusok az Adobe PDF-dokumentumokban 84
3.7.21. AppleScript-függőség 85
3.7.22. ANSI-függőség 85
3.7.23. Macromedia Flash ActionScript-vírusok 85
2.7.24. HyperTalk-szkriptvírusok 86
3.7.25. AutoLisp-szkriptvírusok 87
3.7.26. Registry-függőség 88
3.7.27. PIF- és LNK-függőségek 88
3.7.28. Lotus Word Pro makróvírusok 89
3.7.29. AmiPro dokumentum vírusok 89
3.7.30. Corel Script-vírusok 90
3.7.31. Lotus 1-2-3 makrófüggőség 90
3.7.32. Windowsos telepítőszkript-függőség 91
3.7.33. AUTORUN.INF és Windows INI-fájlfüggőség 91
3.7.34. HTML- (hipertext jelölőnyelv) függőség 92
3.8. Függés a sebezhetőség fajtájától vagy jelenlététől 92
3.9. Dátum- és időfüggőség 93
3.10. JIT-függőség: a Microsoft .NET vírusai 93
3.11. Függés a tömörített fájlformátumtól 94
3.12. Függés a fájlformátumtól a kiterjesztés alapján 95
3.13. Függés a hálózati protokolltól 97
3.14. Forráskód-függőség 97
3.14.1. Forráskód-trójaivírusok 98
3.15. Erőforrás-függőség a Mac és a Palm rendszerekben 99
3.16. Függés a befogadó fájl méretétől 100
3.17. Hibakereső-függőség 101
3.17.1. Hibakeresőre támaszkodó szándékolt vírusok 102
3.18. Függés a fordító és a kapcsolatszerkesztő (linker) programtól 103
3.19. Függés az eszközfordító rétegtől (Device Translator Layer) 103

3.20. Függés a beágyazott objektum beillesztésétől 107
3.21. Függés az önálló környezettől..... 108
3.22. Többrészes (multipartite) vírusok 109
3.23. Összegzés..... 110
Hivatkozások 111

4. A fertőzési stratégiák osztályozása 115

4.1. Bootvírusok..... 115
4.1.1. Módszerek a fő rendszerindító rekord (MBR) megfertőzésére 117
4.1.1.1. Az MBR megfertőzése a rendszerbetöltő kód cseréjével 117
4.1.1.2. Az MBR-kód cseréje mentés nélkül.....118
4.1.1.3. Az MBR megfertőzése a partíciótábla-bejegyzések módosításával 119
4.1.1.4. Az MBR elmentése a merevlemez végére 119
4.1.2. Módszerek DOS-rendszerindító rekord megfertőzésére 119
4.1.2.1. Szokásos rendszerindítás-fertőző módszerek.....119
4.1.2.2. Külön szektorokat formázó bootvírusok.....120
4.1.2.3. Szektorokat hibásnak megjelölő bootvírusok 121
4.1.2.4. Az eredeti rendszerindító szektort el nem mentő bootvírusok 121
4.1.2.5. A rendszerindító szektort a lemez végére másoló bootvírusok 121
4.1.3. A Windows 95 működése közben is futó bootvírusok 122
4.1.4. Rendszerindító fájlképekre épülő lehetséges támadások hálózati környezetben 122

4.2. Módszerek fájlok megfertőzésére..... 123
4.2.1. Felülíró vírusok.....123
4.2.2. Véletlenszerűen felülíró vírusok 125
4.2.3. Hozzáfűző vírusok.....125
4.2.4. Eléfűző vírusok 127
4.2.5. Klasszikus parazitavírusok.....128
4.2.6. Üregkitöltő vírusok.....129
4.2.7. Széttördelt üregkitöltő vírusok 130
4.2.8. Tömörítővírusok.....132
4.2.9. Amőbavírusok 132
4.2.10. Beágyazott visszafejtő módszer 133
4.2.11. Beágyazottvisszafejtő- és vírustörzsmódszer 135
4.2.12. Álcázott trükkös ugrásmódszer 136
4.2.13. Belépési pontot elrejtő (EPO) vírusok 137
4.2.13.1. Alapvető EPO-módszerek a DOS rendszerben 137
4.2.13.2. Speciális EPO-módszerek a DOS rendszerben 139
4.2.13.3. EPO-vírusok 16 bites Windows rendszereken.....140

4.2.13.4. API-hurok-módszer a Win32 rendszereken 141
4.2.13.5. Függvényhívás-hozzáférés a Win32 rendszereken 143
4.2.13.6. Beemelttábla-csere a Win32 rendszereken 144
4.2.13.7. Utasítás-nyomonkövető módszer a Win32 rendszereken 145
4.2.13.8. „Ismeretlen" belépési pontok használata 145
4.2.13.9. Kódintegráció-alapú EPO-vírusok 146
4.2.14. A jövő fertőzési módszerei: kódépítők 147
4.3. Részletesen a Win32-vírusokról 148
4.3.1. A Win32 API és a támogató platformok 148
4.3.2. 32 bites Windows rendszerek fertőzési módszerei 150
4.3.2.1. Bevezetés a hordozható, futtatható (PE) fájlformátumba 151
4.3.2.1.1. A PE-fejrész 152
4.3.2.1.2. A szekciótábla és a leggyakoribb szekciók 155
4.3.2.1.3. PE-fájl-betöltés: hogyan kapcsolódnak a DLL-ek a végrehajtható fájlokhoz? 158
4.3.2.1.4. PE-fájl-exportok 160
4.3.2.2. Első generációs Windows 95-ös vírusok 161
4.3.2.2.1. Fejrészfertőzés 162
4.3.2.2.2. Eléfűző vírusok 162
4.3.2.3. Hozzáfűző vírusok, amelyek nem vesznek fel új szakaszfejrészt 163
4.3.2.4. Hozzáfűző vírusok, amelyek nem módosítják a belépési pontot 164
4.3.2.5. KERNEL32.DLL-fertőzés 164
4.3.2.6. Kísérő fertőzések 165
4.3.2.7. Széttördelt üregkitöltő fertőzés 165
4.3.2.8. Régi EXE-fejrész Ifanew mezőjének módosítása 166
4.3.2.9. VxD-alapú Windows 95-ös vírusok 167
4.3.2.10. Vxd-ként működő PE-vírusok 167
4.3.2.11. VxD-fertőzés 168
4.3.2.12. A DLL-betöltés fertőzési módszere 169
4.3.3. Win32- és Win64-vírusok a Microsoft Windows rendszerben? 169
4.3.3.1. A Windows 95 és az NT rendszerbetöltőinek lényeges eltérései 170
4.4. Összegzés 172
Hivatkozások 172

5. A memórián belüli stratégiák osztályozása 173

5.1. Közvetlenül fertőző vírusok 173
5.2. Memóriában maradó vírusok 174
5.2.1. Megszakításkezelés és behurkolás 175
5.2.2. Behurkoló rutinok INT 13h-n (bootvírusok) 178
5.2.3. Behurkoló rutinok INT 21h-n (fájlvírusok) 180
5.2.4. Általános memóriatelepítési technikák a DOS-ban 183
5.2.4.1. Önfelderítő módszerek a memóriában 185
5.2.5. Lopakodó vírusok 186
5.2.5.1. Féllopakodó (könyvtárlopakodó) 186
5.2.5.1.1. VxDCall–INT21_elosztókezelő 187
5.2.5.1.2. Importcímtábla (IAT) hurkolása 188
5.2.5.2. Olvasáslopakodó 189
5.2.5.3. Olvasáslopakodó a Windowsban 190
5.2.5.4. Teljesen lopakodó vírusok 191
5.2.5.5. Klaszter- és szektorszintű fájllopakodó 193
5.2.5.6. Hardverszintű lopakodó 194
5.2.6. Lemezgyorsítótár- és rendszerpuffer-fertőzés 195
5.3. Átmenetileg a memóriában maradó vírusok 196
5.4. Részcserélő vírusok (helycserés támadás) 197
5.5. Vírusok a folyamatokban (felhasználói módban) 197
5.6. Kernelmódú vírusok (Windows 9x/Me) 199
5.7. Kernelmódú vírusok (Windows NT/2000/XP) 199
5.8. A hálózaton keresztül memóriába fecskendező vírusok 202
Hivatkozások 202

6. Alapvető önvédelmi stratégiák 203

6.1. Csatornatechnikát (tunneling) alkalmazó vírusok 203
6.1.1. Az eredeti kezelő memória-ellenőrzése 203
6.1.2. Felderítés a hibakereső interfész segítségével 204
6.1.3. Kódemuláció-alapú csatornatechnika 204
6.1.4. A lemez elérése a port I/O segítségével 205
6.1.5. Nem dokumentált függvények használata 205
6.2. Védekező vírusok 205
6.2.1. Visszafejtés-mentesítés 206
6.2.2. Titkosított adatok 206
6.2.3. Kódösszezavarás az elemzés elkerüléséért 208
6.2.4. Utasításkód-keverésre épülő kódolási zűrzavar 209
6.2.5. Az ellenőrző szám használata 209
6.2.6. Tömörített, obfuscate kódolás 210

6.2.7. Hibakeresés-ellenes technikák 211
6.2.7.1. Az INT1 és az INT3 hurkolása x86 esetén....211
6.2.7.2. Az INT1 és az INT3 megszakításvektorainak kiszámítása 212
6.2.7.3. A forráskód ellenőrző számának kiszámítása a töréspontok észleléséhez 212
6.2.7.4. A verem állapotának ellenőrzése a forráskód végrehajtásakor 212
6.2.7.5. INT 1 vagy INT 3 használata másik megszakítás végrehajtására 213
6.2.7.6. Az INT 3 használata kernelmódba lépéshez a Windows 9x rendszeren 213
6.2.7.7. Az INT 0 használata nullával való osztás kivételének a generálásához....214
6.2.7.8. Az INT 3 felhasználása kivétel generálására 214
6.2.7.9. A Win32 használata IsDebuggerPresent() API-val....215
6.2.7.10. Hibakereső program érzékelése rendszerleíró adatbáziskulcsok keresésével 215
6.2.7.11. Hibakereső program érzékelése illesztőprogram-listázással vagy memóriaátvizsgálással 215
6.2.7.12. Visszafejtés SP és ESP (veremmutató) segítségével 215
6.2.7.13. A vírustörzs fordított visszafejtése 215
6.2.7.14. Az előzetes letöltősor támadásai (és amikor visszafelé sülnek el)....216
6.2.7.15. A billentyűzet letiltása....217
6.2.7.16. Kivételkezelők használata 217
6.2.7.17. A hibakeresési regiszterek (DRn) tartalmának törlése....217
6.2.7.18. A videomemória tartalmának ellenőrzése 218
6.2.7.19. A TIB (szálinformációs blokk) tartalmának ellenőrzése 218
6.2.7.20. A CreateFile() API használata (a „Billy Belcebu" módszer)....218
6.2.7.21. Hamming-kód alkalmazása a megszakítási pontok támadására....218
6.2.7.22. Obfuscate fájlformátumok és belépési pontok 219
6.2.8. Antiheurisztika 219
6.2.8.1. Támadások az első generációs Win32-heurisztikák ellen....220
6.2.8.1.1. Új PE-fájl-fertőzési technikák 220
6.2.8.1.2. Egynél több vírusszekció 221
6.2.8.1.3. A gazdafájl fejrészét titkosító eléfűző vírusok....221
6.2.8.1.4. A kihasználatlan terület első szakaszának fertőzése....222

6.2.8.1.5. Az első szakasz fertőzése a fájlszakaszok elmozdításával 222
6.2.8.1.6. Az első szakasz fertőzése tömörítéssel 222
6.2.8.1.7. A belépési pontot elhomályosító technikák 223
6.2.8.1.8. Véletlenszerű belépési pont kiválasztása a kódszakaszban 223
6.2.8.1.9. A fordítóigazítási terület újrahasznosítása 224
6.2.8.1.10. Vírusok, amelyek egyetlen szakaszt sem változtatnak írhatóvá 224
6.2.8.1.11. Pontos fájlfertőzés 224
6.2.8.1.12. Az ellenőrző szám újraszámítása 224
6.2.8.1.13. Meglévő szakaszok újraszámítása 225
6.2.8.1.14. A fejrészfertőzés elkerülése 225
6.2.8.1.15. Importálás megakadályozása sorszámokkal 225
6.2.8.1.16. A nincs CALL-to-POP trükk 226
6.2.8.1.17. A kódméret helyreállítása a fejrészben 226
6.2.8.1.18. Nincs API-sztring-használat 227
6.2.9. Emulációellenes technikák 227
6.2.9.1. A tömörítő (FPU) utasításainak használata 228
6.2.9.2. MMX-utasítások alkalmazása 228
6.2.9.3. Strukturált kivételkezelés használata 229
6.2.9.4. Véletlenszerű víruskód végrehajtása: ez ma vírusnak számít? 229
6.2.9.5. Dokumentálatlan CPU-utasítások használata 230
6.2.9.6. A víruskód brute-force visszafejtésének alkalmazása 230
6.2.9.7. Többszálas vírusfunkció alkalmazása 231
6.2.9.8. Megszakítások alkalmazása polimorfikus visszafejtőkben 231
6.2.9.9. A vezérlés átadása a víruskódnak API használatával 231
6.2.9.10. Hosszú hurkok 232
6.2.10. Csalielkerülő (antigoat) vírusok 232
6.3. Agresszív retrovírusok 232
Hivatkozások 235

7. Fejlett kódevolúciós eljárások és vírusgeneráló csomagok 237

7.1. Bevezetés 237
7.2. A kód evolúciója 237
7.3. Titkosított vírusok 238
7.4. Oligomorf vírusok 244

7.5. Polimorf vírusok ... 245
7.5.1. Az 1260 vírus ... 246
7.5.2. A Dark Avenger Mutation Engine (MtE) ... 247
7.5.3. 32 bites polimorf vírusok ... 249
7.6. Metamorf vírusok ... 253
7.6.1. Mi is az a metamorf vírus? ... 254
7.6.2. Egyszerű metamorf vírusok ... 255
7.6.3. Bonyolultabb metamorf vírusok és permutációs eljárások ... 257
7.6.4. Más alkalmazások megváltoztatása: a végső vírusgenerátor? ... 259
7.6.5. Fejlett metamorf vírusok: a Zmist ... 261
7.6.5.1. Inicializálás ... 262
7.6.5.2. Közvetlen műveleti fertőzés ... 262
7.6.5.3. Permutáció ... 262
7.6.5.4. Hordozható futtatható állományok fertőzése ... 262
7.6.5.5. Kódintegráció ... 264
7.6.6. {W32, Linux}/Simile: Egy keresztplatform-metamorfmotor ... 265
7.6.6.1. Reprodukciós rutin ... 265
7.6.6.2. EPO-mechanizmus ... 265
7.6.6.3. A polimorf dekódoló ... 265
7.6.6.4. A metamorf folyamat ... 267
7.6.6.5. Reprodukció ... 268
7.6.6.6. Aktív tartalom ... 268
7.6.7. A sötét jövő – MSIL metamorf vírusok ... 269
7.7. Vírus-összeállító készletek ... 271
7.7.1. VCS (Virus Construction Set) ... 272
7.7.2. GenVir ... 272
7.7.3. VCL (Virus Creation Laboratory) ... 272
7.7.4. PS-MPC (Phalcon-Skism Mass-Produced Code Generator) ... 273
7.7.5. NGVCK (Next Generation Virus Creation Kit) ... 274
7.7.6. Egyéb csomagok és kódmódosítók ... 274
7.7.7. Megengedett a vírusgeneráló eszközök tesztelése? ... 275
Hivatkozások ... 276

8. Osztályozás a szöveges kimenet (payload) szerint 279

8.1. Aktív tartalom nélkül ... 279
8.2. Nem szándékoltan romboló tartalom ... 280
8.3. Nem káros tartalom ... 281
8.4. Némileg káros tartalom ... 283

8.5. Rendkívül káros tartalom 284
8.5.1. Adatokat felülíró vírusok 284
8.5.2. Data diddler (adattönkretevő) vírusok 286
8.5.3. Adattitkosító vírusok: a „jó, a rossz és a csúf" 286
8.5.4. Hardverrombolók 288
8.6. Túlterheléses (DoS) támadások 289
8.7. Adatlopás: pénzkeresés vírusokkal 291
8.7.1. A jelszóvadászat-támadás 292
8.7.2. Hátsó ajtót (backdoor) tartalmazó vírusok 293
8.8. Összegzés 295
Hivatkozások 295

9. A számítógépes férgek stratégiái 297

9.1. Bevezetés 297
9.2. A számítógépes férgek általános szerkezete 298
9.2.1. Célmeghatározó modul 299
9.2.2. Fertőzés-továbbterjesztő modul 299
9.2.3. Távirányítás és frissítőinterfész 299
9.2.4. Életciklus-vezérlő modul 300
9.2.5. Aktív tartalom 301
9.2.6. Önkövetés 302
9.3. A célmeghatározó modul 302
9.3.1. E-mail címek gyűjtése 302
9.3.1.1. Címjegyzékférgek 303
9.3.1.2. Fájlelemző támadások a merevlemezen 303
9.3.1.3. NNTP-alapú e-mail kollektorok 304
9.3.1.4. E-mail címek gyűjtése a weben 305
9.3.1.5. E-mail címek gyűjtése ICQ-n keresztül 305
9.3.1.6. Felhasználói SMTP és hírcsoport hozzáférésének ellenőrzése menet közben 306
9.3.1.7. Összetett módszerek 307
9.3.2. Hálózati megosztásfelsoroló támadások 307
9.3.3. Hálózatletapogatás és célazonosítás 309
9.3.3.1. Letapogatás előre meghatározott osztálytáblával: a Linux/Slapper féreg 310
9.3.3.2. Véletlenszerű letapogatás: a W32/Slammer féreg 312
9.3.3.3. Összetett letapogatómetódusok: a W32/Welchia féreg 313
9.4. A fertőzések továbbterjesztése 314
9.4.1. Hátsó ajtós módszerrel megrongált rendszerek támadása 314
9.4.2. Egyenrangú hálózatok támadása 315

9.4.3. Támadás azonnali üzenetküldéssel 316
9.4.4. E-mailes féregtámadások és megtévesztési módszerek 317
9.4.5. E-mail csatolmányok beszúrása 317
9.4.6. SMTP-proxy-alapú támadások 318
9.4.7. SMTP-támadások 319
9.4.8. SMTP-továbbterjedés MX-lekérdezést használó szteroidokon 321
9.4.9. NNTP-támadások (Network News Transfer Protocol – hálózati hírtovábbítási protokoll) 322
9.5. Általános módszerek a féregkód átvitelére és végrehajtására 322
9.5.1. Végrehajthatóparancskód-alapú támadások 322
9.5.2. Hivatkozás webhelyekre vagy webproxykra 323
9.5.3. HTML-alapú levelek 324
9.5.4. Távoli bejelentkezésen alapuló támadások 324
9.5.5. Támadások forráskódok beszúrásával 325
9.5.6. Burokkódalapú támadások 326
9.6. A számítógépes férgek frissítési stratégiái 328
9.6.1. Hitelesített frissítések a weben vagy a hírcsoportokon 330
9.6.2. Hátsóajtó-alapú frissítések 334
9.7. Távvezérlés jelek küldésén keresztül 335
9.7.1. Egyenrangú hálózatok vezérlése 336
9.8. Szándékolt és véletlenszerű kölcsönhatások 338
9.8.1. Együttműködés 338
9.8.2. Verseny 340
9.8.3. A jövő: egyszerű féregkommunikációs protokoll? 342
9.9. A vezeték nélküli mobileszközök férgei 342
Hivatkozások 344

II. rész: A védelmező stratégiái

10. Biztonsági rések, sebezhető pontok és puffer-túlcsordulásos támadások 349

10.1. Bevezetés 349
10.1.1. Az összetett támadás meghatározása 349
10.1.2. A fenyegetés 350
10.2. Háttér 351
10.3. A sebezhetőségek típusai 352
10.3.1. Puffertúlcsordulás 352
10.3.2. Első generációs támadások 353

10.3.2.1. Túlcsordulás egy verempufferben 353
10.3.2.2. A verempufferben történt túlcsordulás kihasználása 354
10.3.2.3. A veremalapú túlcsordulási sebezhetőségek okai 354
10.3.3. Második generációs támadások 355
10.3.3.1. Eggyel elszámolt túlcsordulások 355
10.3.3.2. Túlcsordulások a dinamikus memóriaterületen 356
10.3.3.3. A dinamikus memóriaterület 356
10.3.3.4. A sebezhető kód 357
10.3.3.5. A túlcsordulás kihasználása 358
10.3.3.6. Függvénymutatók 360
10.3.4. Harmadik generációs támadások 361
10.3.4.1. Format string támadások 361
10.3.4.2. A dinamikus memóriaterület kezelése 366
10.3.4.3. A bevitel ellenőrzése 367
10.3.4.4. URL-kódolás és -gyűjtés 367
10.3.4.5. MIME-fejrész elemzése 369
10.3.4.6. Az alkalmazás jogainak ellenőrzése 370
10.3.4.7. Biztonságos ActiveX-vezérlők a szkriptíráshoz 370
10.3.4.8. A rendszer módosítása 371
10.3.4.8.1. A NetWare ExecuteOnly attribútum: veszélyesnek tekintik 371
10.3.4.8.2. Csak végrehajtható? 372
10.3.4.8.3. Következtetések 374
10.3.4.9. Hálózatfelsorolás 375
10.4. A jelenlegi és a korábbi veszélyek 376
10.4.1. A Morris internetes féreg, 1988 (veremtúlcsordulás a futtatási burokkódban) 376
10.4.2. Linux/ADM, 1998 (a Morris féreg utánzója) 378
10.4.3. A CodeRed-kitörés, 2001 (a kódinjektáló támadás) 379
10.4.3.1 A puffertúlcsordulás részletei 380
10.4.3.2. A kivételkeret sebezhetősége 382
10.4.4. Linux/Slapper féreg, 2002 (példa a dinamikus memóriaterület túlcsordulására) 382
10.4.4.1. A támadás 383
10.4.4.2. A puffertúlcsordulás 383
10.4.4.3. Duplázás 384
10.4.4.4. A dinamikusmemória-terület címének megszerzése 385
10.4.4.5. Visszaélés a glibc-vel 385
10.4.4.6. Burokkód és fertőzés 387
10.4.4.7. Aki bújt, aki nem 387
10.4.4.8. P2P-támadóhálózat 388

10.4.4.9. A Linux/Slapper-támadásból levont következtetések 388
10.4.5. W32/Slammer féreg, 2003. január (az apró féreg) 389
10.4.5.1. A kihasználás felépítése 389
10.4.5.2. Problémák az ssnetlib.dll-ben (SQL Server 2000) 389
10.4.5.3. Az irányítás átvétele 390
10.4.5.4. Inicializálás 390
10.4.5.5. Reprodukció 391
10.4.5.6. A Slammer féreg támadásából levont következtetések 391
10.4.6. Blaster féreg, 2003. augusztus (burokkódalapú támadás Win32 rendszerek ellen) 392
10.4.6.1. Minden rendszer mehet 392
10.4.6.2. SP4, SP3, SP2, SP1 – gyújtás! 392
10.4.6.3. Második lépés: a shell 393
10.4.6.4. „Egy kis baj van” 393
10.4.6.5. A „pángalaktikus gégepukkasztó” 393
10.4.6.6. MS-DoS 394
10.4.7. Általános puffertúlcsordulás számítógépes vírusokban 394
10.4.8. A W32/Badtrans.B@mm leírása 395
10.4.8.1. A MIME kihasználása 395
10.4.9. A W32/Nimda.A@mm 395
10.4.9.1. Az IIS biztonsági rés 396
10.4.10. A W32/Bolzano leírása 396
10.4.10.1. Az operációs rendszer kernelének a módosítása 397
10.4.11. A VBS/Bubbleboy leírása 398
10.4.11.1. A szkriptíráshoz biztonságos ActiveX-vezérlő kihasználása 398
10.4.12. A W32/Blebla leírása 399
10.4.12.1. Az ActiveX és a gyorsítótár megkerülésének kihasználása 399
10.5. Összegzés 400
Hivatkozások 401

11. Vírusvédelmi módszerek 403

11.1. A víruskeresők első generációja 405
11.1.1. Sztringátvizsgálás 405
11.1.2. Helyettesítőkarakterek 407
11.1.3. Rossz párosítások 409
11.1.4. Általános észlelés 409
11.1.5. Hashalgoritmusok alkalmazása 409
11.1.6. Könyvjelzők 410

11.1.7. Top-and-tail átvizsgálás 411
11.1.8. Belépésipont- és fixpont-átvizsgálás 411
11.1.9. Hipergyors lemezhozzáférés 413
11.2. A víruskeresők második generációja 413
11.2.1. Intelligens átvizsgálás 413
11.2.2. Csontvázészlelés 414
11.2.3. Majdnem pontos azonosítás 414
11.2.4. Pontos azonosítás 415
11.3. Algoritmikus felismerési módszerek 417
11.3.1. Szűrés 419
11.3.2. Statikus visszafejtő észlelése 420
11.3.3. A röntgenmódszer 422
11.4. Kódemulálás 426
11.4.1. Titkosított és polimorf vírusok észlelése emulációval 430
11.4.2. Dinamikusvisszafejtő-észlelés 433
11.5. A metamorf vírusok elleni védekezés 435
11.5.1. Geometrikus észlelés 435
11.5.2. Visszafejtési módszerek 436
11.5.3. Emulátorok alkalmazása a nyomon követéshez 437
11.5.3.1. Az ACG minta észlelése 438
11.5.3.2. Az Evol-minta észlelése 438
11.5.3.3. Negatív és pozitív funkciók 439
11.5.3.4. Emulátoralapú heurisztikák 439
11.6. A 32 bites Windows-vírusok heurisztikus elemzése 441
11.6.1. A kódvégrehajtás az utolsó részben kezdődik 442
11.6.2. A gyanús rész jellemzői 442
11.6.3. A virtuális méret helytelen a PE-fejrészben 442
11.6.4. Lehetséges „rés” a részek között 442
11.6.5. Gyanús programkód átirányítása 443
11.6.6. Gyanús kódrésznév 443
11.6.7. Lehetséges fejrészfertőzés 443
11.6.8. Gyanús import a KERNEL32.DLL könyvtárból sorszám szerint 443
11.6.9. Javított importcímtáblázat 444
11.6.10. Több PE-fejrész 444
11.6.11. Több Windows-fejrész és gyanús KERNEL32.DLL-importok 444
11.6.12. Gyanús áthelyezések 444
11.6.13. A kernel keresése 445
11.6.14. Kernelinkonzisztencia 445
11.6.15. Egy rész betöltése a VMM-címtérbe 445
11.6.16. Hibás kódméret a fejrészben 446
11.6.17. Gyanús jelölőkombináció-példák 446

11.7. Heurisztikus elemzés neurális hálózatokkal 447
11.8. Szokásos és általános helyreállítási módszerek 449
11.8.1. Szabványos vírusirtás 450
11.8.2. Generikus visszafejtők 451
11.8.3. Hogyan működik a generikus vírusirtó? 451
11.8.4. Hogyan győződhet meg a vírusirtó arról, hogy a fájl fertőzött? 452
11.8.5. Hol található a gazdafájl eredeti vége? 452
11.8.6. Hányféle vírustípust kezelhetünk így? 452
11.8.6.1. Betöltőszektor-vírusok 453
11.8.6.2. Fájlvírusok 453
11.8.7. A generikus javítás heurisztikái 454
11.8.8. Általános vírusirtási példák 455
11.9. Immunizálás 456
11.10. Hozzáférés-szabályozó rendszerek 457
11.11. Épségellenőrzés 458
11.11.1. Hibás észlelések 459
11.11.2. Tiszta kiindulóállapot 460
11.11.3. Sebesség 460
11.11.4. Speciális objektumok 460
11.11.5. A módosított objektumok szükségessége 461
11.11.6. Lehetséges megoldások 461
11.12. Meghatározott viselkedés megakadályozása 462
11.13. Gumiszoba (Sandboxing) 464
11.14. Összegzés 465
Hivatkozások 465

12. Memóriaátvizsgálás és vírusirtás 469

12.1. Bevezetés 470
12.2. A Windows NT virtuálismemória-rendszere 472
12.3. Virtuális címterületek 474
12.4. Memóriavizsgálat felhasználói üzemmódban 478
12.4.1. Az NtQuerySystemInformation() titkai 479
12.4.2. Általános folyamatok és különleges rendszerjogosultságok 480
12.4.3. Vírusok a Win32-alrendszerben 481
12.4.4. Saját lapokat kiosztó Win32-vírusok 482
12.4.5. Natív Windows NT-szolgáltatás-vírusok 484
12.4.6. Rejtett ablakfolyamatokat használó Win32-vírusok 484
12.4.7. Végrehajtott kép részét képező Win32-vírusok 485

12.5. A memóriavizsgálat és a lapozás 487
12.5.1. Folyamatok és átvizsgáló fájlképek felsorolása 489
12.6. Vírusirtás a memóriából 489
12.6.1. Adott, víruskódot tartalmazó folyamat befejezése 489
12.6.2. Vírusszálak felismerése és leállítása 490
12.6.3. Víruskód kijavítása aktív lapokon 493
12.6.4. Hogyan végezzük el betöltött DLL-k és futó alkalmazások vírusirtását? 494
12.7. Memóriavizsgálat kernelmódban 494
12.7.1. A folyamatok felhasználói címterének átvizsgálása 495
12.7.2. Az NT-szolgáltatás API belépési pontjainak meghatározása 495
12.7.3. Fontos NT-funkciók kernelmódú memóriaátvizsgálás esetén 496
12.7.4. Folyamatkontextus 497
12.7.5. A címtér felső 2 GB-jának átvizsgálása 498
12.7.6. Hogyan inaktiválhatjuk a szűrő-illesztő programokat megtámadó vírusokat? 498
12.7.7. Csak olvasható kernelmemória kezelése 500
12.7.8. 64 bites platformok kernelmódú memóriaátvizsgálása 501
12.8. Lehetséges támadások a memóriavizsgálat ellen 503
12.9. Összegzés és továbblépés 504
Hivatkozások 506

13. Féregblokkolási módszerek és állomásalapú betörésmegelőzés 507

13.1. Bevezetés 507
13.1.1. Parancsfájl-blokkolás és az SMTP-férgek blokkolása 508
13.1.2. Újszerű támadások blokkolása: CodeRed, Slammer 512
13.2. Módszerek a puffertúlcsordulást okozó támadások elhárítására 512
13.2.1. A forráskódok felülvizsgálata 514
13.2.1.1. Biztonsági frissítések 514
13.2.2. A fordítóprogramok szintjén alkalmazható megoldások 516
13.2.2.1. A StackGuard 517
13.2.2.2 A ProPolice program 518
13.2.2.3 A Microsoft Visual Studio .NET 2003 7.0 és 7.1 519
13.2.3. Operációsrendszer-szintű megoldások és futásidejű kiterjesztések 523
13.2.3.1. Solaris SPARC-processzoron 523

13.2.4. Az alrendszerek kiterjesztései – Libsafe 524
13.2.5. Kernelmódú kiterjesztések 525
13.2.6. Programterelés 527
13.3. Módszerek a férgek blokkolására 527
13.3.1. Beszúrt kód észlelése 528
13.3.1.1. Héjkódok blokkolása a kódbeszúrás észlelésével 529
13.3.2. A küldés blokkolása. Példa az önmagát továbbküldő kód blokkolására 534
13.3.2.1. A W32/Slammer féreg blokkolása 535
13.3.2.2. A W32/CodeRed féreg blokkolása 536
13.3.3. A kivételkezelők érvényesítése 537
13.3.3.1. A kivételkezelők nem megfelelő sorrendje 537
13.3.3.2. Kivételkezelő a dinamikus memóriaterületen vagy a veremben 539
13.3.3.3. A kivételkeret-mutató érvénytelen 540
13.3.4. Egyéb módszerek a vissza-a-LIBC-hez típusú támadások elhárítására 541
13.3.4.1. A folyamatok címterének véletlenszerűvé tétele 542
13.3.4.2. Könyvtári függvények közvetlen meghívásának az észlelése 543
13.3.5. A lapok „GOT” és „IAT” attribútuma 546
13.3.6. Nagyszámú kapcsolat és kapcsolati hiba 546
13.4. Lehetséges jövőbeli féregtámadások 548
13.4.1. A retrofférgek elterjedtségének lehetséges növekedése 548
13.4.2. „Lassú” férgek a radar alatt 548
13.4.3. Polimorf és metamorf férgek 549
13.4.4. A súlyosabb károk veszélye 550
13.4.5. A biztonsági rések felfedezésének automatizálása környezeti tapasztalatok alapján 550
13.5. Összegzés 551
Hivatkozások 553

14. Hálózati szintű védelmi stratégiák 555

14.1. Bevezetés 555
14.2. Az útválasztók hozzáférési listáinak a felhasználása 556
14.3. Védekezés tűzfallal 559
14.4. A hálózati betörésérzékelő rendszerek 562
14.5. Légyfogó rendszerek 564
14.6. Ellentámadások 567
14.7. Korai figyelmeztető rendszerek 568

14.8. A férgek viselkedési mintái a hálózatban 569
14.8.1. A Blaster féreg elfogása 569
14.8.2. A Linux/Slapper féreg elfogása 571
14.8.3. A W32/Sasser.D féreg elfogása 573
14.8.4. A W32/Welchia féreg pingelési kérelmeinek elfogása 575
14.8.5. A W32/Slammer és a kapcsolódó biztonsági rések észlelése 576
14.9. Összegzés 578
Hivatkozások 578

15. A rosszindulatú kódok elemzésének módszerei 581

15.1. Saját víruselemző laborunk 581
15.1.1 Hogyan szerezhető be a szoftver? 583
15.2. Információ, információ, információ 584
15.2.1 Architektúrakalauzok 584
15.2.2. Microsoft-tudásbázis 585
15.3. Dedikált víruselemzés a VMWARE rendszerben 586
15.4. A számítógépes vírusok elemzésének a folyamata 588
15.4.1. Előkészületek 588
15.4.1.1. Gyorsvizsgálat 588
15.4.1.2. Szűrés 588
15.4.1.3. Tisztítás 590
15.4.1.4. A vírus kódjának gyors vizsgálata 591
15.4.1.5. Sztring kiíratása 592
15.4.1.6. Visszafejtés 593
15.4.1.7. „Feketedobozolás” 593
15.4.2. Kicsomagolás 594
15.4.3. Visszafejtés és dekódolás 595
15.4.4. A dinamikus elemzés módszerei 602
15.4.4.1. A fájlváltozások figyelése 603
15.4.4.2. Természetes fertőzésteszt goat- (áldozat) fájlok használatával 605
15.4.4.3. A beállításjegyzék-módosítások figyelése 607
15.4.4.4. Folyamatok és szálak megfigyelése 608
15.4.4.5. A hálózati portok figyelése 608
15.4.4.6. A hálózati forgalom lehallgatása és elfogása 610
15.4.4.7. A rendszerhívások nyomon követése 613
15.4.4.8. Hibakeresés 614
15.4.4.9. Víruselemzés szteroidokon 621
15.5. A rosszindulatú kódok gyűjteményének karbantartása 624
15.6. Automatizált elemzés: a digitális immunrendszer 624
Hivatkozások 627

16. Befejezés 629

Ajánlott olvasmányok 630

A biztonsággal és a korai figyelmeztetéssel kapcsolatos információk 630
Biztonsági frissítések 630
Számítógépesféregjárvány-statisztika 631
A víruskutatással kapcsolatos tanulmányok 631
Vírusirtók készítőinek partneradatai 631
Vírusirtó-tesztelők és hasonló oldalak 633

Tárgymutató 635