

**Rootkity i bootkity : zwalczanie współczesnego złośliwego oprogramowania i zagrożeń nowej generacji / Alex Matrosov, Eugene Rodionov i Sergey Bratus z przedmową Rodrigo Robira Branco. –
Wydanie I. – Warszawa, 2020**

Spis treści

PRZEDMOWA <i>Rodrigo Rubia Branco</i>	xxi
PODZIĘKOWANIA	xxv
SKRÓTY	xxvii
WPROWADZENIE	xxxii
Dlaczego warto przeczytać tę książkę?	xxxii
Co znajdziemy w tej książce?	xxxiii
Część 1: Rootkity	xxxiii
Część 2: Bootkity	xxxiii
Część 3: Obrona i techniki śledcze	xxxv
Jak czytać tę książkę	xxxvi
CZĘŚĆ I: ROOTKIT	1
1	
CZYM JEST ROOTKIT: STUDIUM PRZYPADKU TDL3	3
Historia dystrybucji TDL3 w świecie	4
Procedura infekcji	5
Kontrola przepływu danych	8
Bring Your Own Linker	8
Jak działają hooki trybu jądra w TDL3	9
Ukryty system plików	12
Podsumowanie: TDL3 spotyka swoją Nemesis	13
2	
ROOTKIT FESTI: NAJBARDZIEJ ZAAWANSOWANY BOT SPAMOWY I DDOS	15
Przypadek botnetu Festi	16
Analiza sterownika rootkita	17
Informacje konfiguracyjne Festi do komunikacji z C&C	18
Zorientowana obiektowo platforma Festi	19
Zarządzanie wtyczkami	20
Wbudowane wtyczki	21
Techniki obrony przed uruchomieniem w maszynie wirtualnej	23
Techniki antydebuggerowe	25
Metoda ukrywania złośliwego sterownika na dysku	25
Metoda ochrony klucza rejestru Festi	28

Sieciowy protokół komunikacyjny Festi	29
Faza inicjowania	29
Faza robocza	30
Omijanie oprogramowania antywirusowego i oprogramowania do analizy śledczej	31
Algorytm generowania domeny na wypadek awarii C&C	34
Złośliwa funkcjonalność	35
Moduł Spam	35
Mechanizm DDoS	36
Wtyczka proxy Festi	37
Podsumowanie	38

3

OBSERWOWANIE INFЕКCJI ROOTKITAMI	41
Metody przechwytywania	42
Przechwytywanie zdarzeń systemowych	42
Przechwytywanie wywołań systemowych	44
Przechwytywanie operacji plikowych	46
Przechwytywanie dyspozytora obiektów	48
Przywracanie jądra systemu	50
Wielki rootkitowy wyścig zbrojeń: uwaga nostalgiczna	51
Podsumowanie	53

CZĘŚĆ II: BOOTKITY 55

4

EWOLUCJA BOOTKITU	57
Pierwsze bootkity	58
Programy infekujące sektor rozruchowy	58
Elk Cloner i Load Runner	58
Wirus Brain	59
Ewolucja bootkitów	60
Koniec ery BSI	60
Zasada podpisywania kodu trybu jądra	60
Pojawienie się Secure Boot	61
Nowoczesne bootkity	62
Podsumowanie	64

5

PODSTAWY PROCESU ROZRUCHU SYSTEMU OPERACYJNEGO	65
Ogólny przegląd procesu rozruchu systemu Windows	66
Starszy proces rozruchowy	67
Proces rozruchu Windows	68
BIOS i środowisko przeduruchomieniowe	69
Master Boot Record	69
Volume Boot Record i Initial Program Loader	71
Moduł bootmgr oraz Boot Configuration Data	73
Podsumowanie	77

6	
ZABEZPIECZENIA PROCESU ROZRUCHU	79
Moduł Early Launch Anti-Malware	80
Procedury wywołań zwrotnych API	80
Jak bootkity omijają ELAM?	82
Zasada podpisywania kodu trybu jądra	83
Sterowniki trybu jądra podlegające testom integralności	83
Lokalizacja podpisów sterowników	84
Słabość integralności starszego kodu	85
Moduł ci.dli	87
Zmiany w mechanizmach ochronnych Windows 8	89
Technologia Secure Boot	90
Zabezpieczenia oparte na wirtualizacji w Windows 10	91
Second Level Address Translation	91
Virtual Secure Mode i Device Guard	92
Ograniczenia Device Guard dotyczące tworzenia sterowników	93
Podsumowanie	94
7	
TECHNIKI INFЕКCJI BOOTKITÓW	95
Techniki infekcji MBR	96
Modyfikowanie kodu MBR: technika infekcji TDL4	96
Modyfikowanie tablicy partycji MBR	103
Techniki infekcji VBR/IPL	104
Modyfikacje IPL: Rovnix	104
Infekcja VBR: Gapz	105
Podsumowanie	106
8	
STATYCZNA ANALIZA BOOTKITU PRZY UŻYCIU IDA PRO	107
Analizowanie MBR zainfekowanego przez bootkit	108
Ładowanie i deszyfrowanie MBR	108
Analizowanie usługi dyskowej BIOS	113
Analiza tablicy partycji zainfekowanego MBR	118
Techniki analizy VBR	119
Analizowanie IPL	120
Inne komponenty bootkitów	121
Zaawansowane wykorzystanie IDA Pro: tworzymy niestandardowy loader MBR	122
Poznajemy loader.hpp	122
Implementowanie aecept_file	123
Implementowanie load_file	124
Tworzenie struktury tablicy partycji	125
Podsumowanie	127
Ćwiczenia	127

9

DYNAMICZNA ANALIZA BOOTKITU: EMULACJA

I WIRTUALIZACJA

129

Emulacja przy użyciu Bochs	130
Instalowanie Bochs	131
Tworzenie środowiska Bochs	131
Infekowanie obrazu dysku	134
Korzystanie z wewnętrznego debugera Bochs	136
Łączenie Bochs z IDA	137
Wirtualizacja przy użyciu VMware Workstation	139
Konfigurowanie VMware Workstation	140
Łączenie VMware GDB z IDA	141
Microsoft Hyper-V i Oracle VirtualBox	146
Podsumowanie	146
Ćwiczenia	146

10

EWOLUCJA TECHNIK INFEKOWANIA MBR I VBR: OLMASCO

149

Dropper	150
Zasoby droppera	151
Telemetria celem przyszłego rozwoju	152
Triki zapobiegające debugowaniu i emulacji	153
Funkcjonalność bootkitu	155
Technika infekcji bootkitem	155
Proces rozruchu zainfekowanego systemu	156
Funkcjonalność rootkitu	158
Zahaczanie obiektu urządzenia dysku twardego i wstrzyknięcie payloadu	158
Obsługa ukrytego systemu plików	158
Implementowanie Transport Driver Interface w celu przekserowania łączności sieciowej	161
Podsumowanie	163

11

BOOTKITY IPL: ROVIX I CARBERP

165

Ewolucja bootkitu Rovnix	166
Architektura bootkitu	167
Infekowanie systemu	168
Proces rozruchowy po infekcji i IPL	171
Implementowanie polimorficznego deszyfratora	171
Rozszyfrowywanie programu ładującego Rovnix przy użyciu VMware i IDA Pro	172
Przejmowanie kontroli przez łatanie rozruchowego programu Windows	179
Ładowanie złośliwego sterownika trybu jądra	182
Funkcjonalność sterownika trybu jądra	184
Wstrzykiwanie modułu payloadu	184
Mechanizmy ukrywania i samoobrony	186
Ukryty system plików	187

Formatowanie partycji jako Virtual FAT	188
Szyfrowanie ukrytego systemu plików	188
Uzyskiwanie dostępu do ukrytego systemu plików	189
Ukryty kanał łączności	190
Historia przypadku: połączenie z trojanem Carberp	192
Rozwój Carberp	192
Ulepszenia droppera	194
Wyciek kodu źródłowego	196
Podsumowanie	196

12

GAPZ: ZAAWANSOWANA INFIEKCJA VBR	197
Dropper Gapz	198
Algorytm droppera	200
Analiza droppera	201
Omijanie HIPS	202
Infekowanie systemu bootkitem Gapz	207
Przypomnienie informacji o bloku parametrów BIOS-u	207
Infekowanie VBR	209
Ładowanie złośliwego sterownika trybu jądra	210
Funkcjonalność rootkitu Gapz	212
Ukryty magazyn	215
Samoobrona przed oprogramowaniem antywirusowym	216
Wstrzyknięcie payloadu	218
Interfejs komunikacyjny payloadu	224
Niestandardowy stos protokołu sieciowego	226
Podsumowanie	229

13

ROZWÓJ RANSOMWARE MBR	231
Krótką historia ransomware	232
Ransomware z funkcjonalnością bootkitu	233
Modus operandi ransomware	234
Analiza ransomware Petya	236
Uzyskiwanie uprawnień administratora	236
Infekowanie dysku twardego (krok 1)	237
Szyfrowanie przy użyciu danych konfiguracyjnych złośliwego programu ładującego	241
Prowokowanie awarii systemu	245
Szyfrowanie MFT (krok 2)	245
Podsumowanie: końcowe przemyślenia na temat Petya	251
Analiza ransomware Satana	252
Dropper Satana	252
Infekowanie MBR	252
Informacje debugowania droppera	254
Złośliwy MBR Satana	255
Podsumowanie: końcowe przemyślenia na temat Satana	257
Podsumowanie	258

14

ROZRUCH UEFI A PROCES ROZRUCHOWY MBR/VBR 259

Unified Extensible Firmware Interface	260
Różnice między starszym procesem rozruchowym BIOS a UEFI	261
Przepływ procesu rozruchu	262
Partycjonowanie dysku: MBR kontra GPT	262
Inne różnice	263
Specyficzne cechy tablicy partycji GUID	265
Jak działa oprogramowanie wbudowane UEFI	269
Specyfikacja UEFI	271
Wewnątrz loadera systemu operacyjnego	272
Windows Boot Loader	278
Zalety UEFI z punktu widzenia bezpieczeństwa	281
Podsumowanie	282

15

WSPÓŁCZESNE BOOTKIT Y UEFI 283

Przegląd historycznych zagrożeń BIOS-u	284
WinCIH, pierwszy złośliwy program atakujący BIOS	285
Mebromi	285
Przegląd innych zagrożeń i środków zaradczych	286
Ali Hardware Has Firmware	291
Podatności UEFI	292
(Nie)skuteczność bitów ochrony pamięci	293
Sprawdzanie bitów ochrony	294
Sposoby infekowania BIOS-u	295
Modyfikowanie niepodpisanego Option ROM UEFI	297
Dodanie lub zmodyfikowanie sterownika DXE	299
Istota iniekcji rootkitu	300
Rootkity UEFI wykorzystywane w atakach	306
Rootkit Vector-EDK od Hacking Team	307
Podsumowanie	316

16

PODATNOŚCI OPROGRAMOWANIA UKŁADOWEGO UEFI 317

Co sprawia, że oprogramowanie wbudowane jest podatne?	318
Klasyfikowanie podatności UEFI	322
Podatności „po przejęciu”	323
Podatności umożliwiające przejęcie łańcucha zaopatrzenia	324
Neutralizowanie podatności łańcucha dostaw	325
Historia zabezpieczeń oprogramowania układowego UEFI	325
Jak działają zabezpieczenia BIOS-u	326
Zabezpieczenia pamięci flash SPI i ich podatności	327
Ryzyko związane z niewierzytelnioną aktualizacją BIOS-u	330
Ochrona BIOS-u przy użyciu Secure Boot	331
Intel Boot Guard	332
Technologia Intel Boot Guard	332

Podatności w Boot Guard	333
Podatności w modułach SMM	336
Istota SMM	336
Nadużywanie programów obsługi SMI	336
Podatności w skrypcie rozruchu S3	341
Istota S3 Boot Script	341
Wyszukiwanie słabości w skrypcie rozruchu S3	342
Wykorzystanie podatności skryptu rozruchu S3	343
Naprawianie podatności skryptu rozruchu S3	346
Podatności w Intel Management Engine	346
Historia podatności ME	347
Ataki na kod ME	347
Studium przypadku: ataki na Intel AMT i BMC	348
Podsumowanie	351

CZĘŚĆ III: OBRONA I TECHNIKI ANALIZY 353

17

JAK DZIAŁA UEFI SECURE BOOT 355

Czym jest Secure Boot?	356
Szczegóły implementacji UEFI Secure Boot	357
Sekwencja rozruchu	357
Uwierzytelnianie plików wykonywalnych za pomocą podpisów cyfrowych	359
Baza danych db	360
Baza danych dbx	364
Uwierzytelnienie oparte na czasie	365
Klucze Secure Boot	366
UEFI Secure Boot: pełny obraz	368
Zasady Secure Boot	370
Ochrona przed bootkitami przy użyciu Secure Boot	373
Atakowanie Secure Boot	374
Łatanie oprogramowania PI w celu wyłączenia Secure Boot	374
Modyfikowanie zmiennych UEFI w celu obejścia testów zabezpieczeń	376
Ochrona Secure Boot za pomocą weryfikowanego lub mierzonego rozruchu	377
Weryfikowany rozruch	378
Mierzony rozruch	379
Intel BootGuard	379
Odszukiwanie ACM	380
Poznanie FIT	382
Konfigurowanie Intel BootGuard	384
ARM Trusted Boot Board	386
ARM Trust Zone	386
Programy ładujące w architekturze ARM	387
Przebieg Trusted Boot	389
Weryfikowany rozruch kontra rootkity oprogramowania układowego	391
Podsumowanie	392

18

SPOSOBY ANALIZOWANIA UKRYTYCH SYSTEMÓW PLIKÓW 393

Przegląd ukrytych systemów plików	394
Odczytywanie danych bootkitu z ukrytego systemu plików	395
Odczytywanie danych z systemu w stanie offline	395
Odczytywanie danych w działającym systemie	396
Zahaczanie sterownika miniportu urządzenia pamięci masowej	396
Analizowanie obrazu ukrytego systemu plików	403
Narzędzie HiddenFsReader	403
Podsumowanie	405

19

ŚLEDZTWA BIOS/UEFI: PODEJŚCIA DO ZDOBYWANIA OPROGRAMOWANIA UKŁADOWEGO I ANALIZ 407

Ograniczenia naszych technik badania	408
Dlaczego badanie oprogramowania układowego jest ważne	408
Atakowanie łańcucha dostaw	409
Przełamanie zabezpieczeń BIOS-u przez podatność oprogramowania układowego	409
Pozyskiwanie oprogramowania układowego	410
Podejście programowe do uzyskiwania oprogramowania układowego	411
Lokalizowanie rejestrów obszaru konfiguracji PCI	412
Obliczanie adresów rejestru konfiguracji SPI	413
Korzystanie z rejestrów SPI	414
Czytanie danych z pamięci flash SPI	416
Uwzględnianie niedoskonałości podejścia programowego	417
Sprzętowe podejście do uzyskiwania oprogramowania układowego	419
Przegląd analizy przypadku Lenovo ThinkPad T540p	420
Lokalizowanie układu pamięci flash SPI	421
Odczytywanie pamięci flash SPI przy użyciu modułu FT2232	422
Analizowanie obrazu oprogramowania układowego przy użyciu UEFITool	425
Poznawanie regionów pamięci flash SPI	425
Przeglądanie regionów pamięci flash SPI przy użyciu UEFITool	426
Analizowanie regionu BIOS-u	428
Analizowanie obrazu oprogramowania układowego przy użyciu Chipsec	432
Poznawanie architektury Chipsec	433
Analizowanie oprogramowania układowego przy użyciu Chipsec Util	434
Podsumowanie	437

SKOROWIDZ 439