

*Посвящается нашим семьям  
и всем помогавшим в создании этой книги*

# СОДЕРЖАНИЕ

От издательства .....	13
Об авторах .....	14
О техническом рецензенте .....	14
Вступительное слово .....	15
Благодарности .....	17
Список аббревиатур .....	18
Введение .....	22
Для кого предназначена эта книга .....	23
Структура книги .....	23
Как читать эту книгу .....	26
<b>Часть I. Руткиты</b> .....	27
<b>Глава 1. Что такое руткит: TDL3</b> .....	28
История распространения TDL3 по миру .....	29
Процедура заражения .....	30
Управление потоком данных .....	32
Скрытая файловая система .....	36
Итог: TDL3 встретил свою Немезиду .....	37
<b>Глава 2. Руткит Festi: самый продвинутый бот для спама и DDoS-атак</b> .....	39
Дело о сети ботов Festi .....	40
Устройство драйвера руткита .....	41
Конфигурационная информация Festi для взаимодействия с командно-управляющим сервером .....	42
Объектно-ориентированная структура Festi .....	43
Управление плагинами .....	44
Встроенные плагины .....	45
Методы противодействия виртуальной машине .....	47
Методы противодействия отладке .....	48
Метод сокрытия вредоносного драйвера на диске .....	49
Метод защиты раздела реестра Festi .....	51
Сетевой протокол Festi .....	52
Фаза инициализации .....	52
Рабочая фаза .....	53
Обход средств обеспечения безопасности и КТЭ .....	54
Алгоритм генерирования доменных имен в случае отказа C&C-сервера .....	57
Вредоносная деятельность .....	57
Модуль рассылки спама .....	58
Проведение DDoS-атак .....	58
Плагин прокси-сервиса .....	60
Заключение .....	61

<b>Глава 3. Обнаружение заражения руткитом</b> .....	62
Методы перехвата .....	63
Перехват системных событий .....	63
Перехват системных вызовов .....	65
Перехват операций с файлами.....	67
Перехват диспетчера объектов .....	68
Восстановление ядра системы .....	71
Великая гонка вооружений с руткитами: ностальгическая нотка .....	72
Заключение .....	74
<b>Часть II. Буткиты</b> .....	75
<b>Глава 4. Эволюция буткита</b> .....	76
Первые буткиты.....	76
Инфекторы загрузочного сектора.....	77
Эволюция буткитов.....	78
Закат эры BSI .....	78
Политика подписания кода режима ядра .....	79
Взлет безопасной загрузки .....	80
Современные буткиты .....	80
Заключение .....	83
<b>Глава 5. Основы процесса загрузки операционной системы</b> .....	84
Общий обзор процесса загрузки Windows .....	85
Старый процесс загрузки .....	86
Процесс загрузки Windows.....	87
BIOS и предзагрузочное окружение .....	87
Главная загрузочная запись .....	88
Загрузочная запись тома и начальный загрузчик программы.....	90
Модуль bootmgr и конфигурационные данные загрузки.....	91
Заключение .....	96
<b>Глава 6. Безопасность процесса загрузки</b> .....	97
Модуль раннего запуска антивредоносной программы.....	97
API обратных вызовов .....	98
Как буткиты обходят ELAM .....	100
Политика подписания кода режима ядра .....	101
Драйверы, подлежащие проверке целостности.....	101
Где находятся подписи драйвера .....	102
Слабость проверки целостности унаследованного кода .....	103
Модуль ci.dll.....	104
Дополнительные защитные меры в Windows 8 .....	106
Технология безопасной загрузки .....	107
Безопасность на основе виртуализации в Windows 10 .....	108
Трансляция адресов второго уровня .....	109
Виртуальный безопасный режим и Device Guard .....	109
Ограничения, налагаемые Device Guard на разработку драйверов.....	110
Заключение .....	111
<b>Глава 7. Методы заражения буткитом</b> .....	112
Методы заражения MBR .....	112

Модификация кода в MBR: метод заражения TDL4.....	113
Модификация таблицы разделов в MBR .....	120
Методы заражения VBR/IPL.....	120
Модификации IPL: Rovnix .....	121
Заражение VBR: Gapz .....	122
Заключение.....	122
<b>Глава 8. Статический анализ буткита с помощью IDA Pro .....</b>	<b>124</b>
Анализ MBR буткита.....	125
Загрузка и дешифрирование MBR .....	125
Анализ службы дисков BIOS.....	129
Анализ зараженной таблицы разделов MBR.....	134
Техника анализа VBR.....	135
Анализ IPL .....	136
Оценка других компонентов буткита.....	136
Продвинутая работа с IDA Pro: написание собственного загрузчика MBR.....	138
Файл loader.hpp .....	138
Реализация accept_file.....	139
Реализация load_file .....	140
Создание структуры, описывающей таблицу разделов .....	141
Заключение.....	142
Упражнения .....	143
<b>Глава 9. Динамический анализ буткита: эмуляция и виртуализация .....</b>	<b>145</b>
Эмуляция с помощью Bochs .....	146
Установка Bochs.....	147
Создание окружения Bochs .....	147
Заражение образа диска .....	150
Использование внутреннего отладчика Bochs.....	152
Комбинация Bochs с IDA.....	153
Виртуализация с помощью VMware Workstation.....	155
Конфигурирование VMware Workstation .....	156
Комбинация VMware GDB с IDA .....	157
Microsoft Hyper-V и Oracle VirtualBox .....	160
Заключение.....	161
Упражнения .....	161
<b>Глава 10. Эволюция методов заражения MBR и VBR: Olmasco .....</b>	<b>163</b>
Сбрасыватель.....	164
Ресурсы сбрасывателя.....	164
Средства трассировки для будущих разработок .....	166
Средства противодействия отладке и эмуляции .....	167
Функциональность буткита .....	169
Метод заражения.....	169
Процесс загрузки зараженной системы .....	170
Функциональность руткита .....	171
Подключение к объекту устройства диска и внедрение полезной нагрузки.....	172
Обслуживание скрытой файловой системы.....	172
Реализация интерфейса транспортного драйвера для перенаправления сетевого трафика .....	175
Заключение.....	176

<b>Глава 11. Буткиты начального загрузчика программы:</b>	
<b>Rovnix and Carberg</b> .....	177
Эволюция Rovnix .....	178
Архитектура буткита .....	179
Заражение системы .....	180
Процесс загрузки после заражения и IPL .....	182
Реализация полиморфного дешифровщика .....	182
Дешифрирование начального загрузчика Rovnix с помощью VMware и IDA Pro ....	184
Перехват управления путем изменения начального загрузчика Windows .....	190
Загрузка вредоносного драйвера .....	193
Функциональность вредоносного драйвера .....	194
Внедрение модуля полезной нагрузки .....	194
Механизмы скрытности и самозащиты .....	196
Скрытая файловая система .....	198
Форматирование раздела под файловую систему Virtual FAT .....	198
Шифрование скрытой файловой системы .....	198
Доступ к скрытой файловой системе .....	199
Скрытый канал связи .....	200
Реальный пример: троян Carberg .....	202
Разработка Carberg .....	202
Усовершенствования сбрасывателя .....	204
Утечка исходного кода .....	205
Заключение .....	205
<b>Глава 12. Garz: продвинутое заражение VBR</b> .....	207
Сбрасыватель Garz .....	208
Алгоритм сбрасывателя .....	210
Анализ сбрасывателя .....	211
Обход NIPS .....	212
Заражение системы буткитом Garz .....	216
О блоке параметров BIOS .....	217
Заражение VBR .....	218
Загрузка вредоносного драйвера .....	220
Функциональность руткита Garz .....	221
Скрытое хранилище .....	224
Самозащита от антивредоносных программ .....	225
Внедрение полезной нагрузки .....	227
Интерфейс взаимодействия с полезной нагрузкой .....	232
Собственный стек сетевых протоколов .....	235
Заключение .....	238
<b>Глава 13. Взлет программ-вымогателей, заражающих MBR</b> .....	239
Краткая история современных программ-вымогателей .....	240
Вымогатель с функциональностью буткита .....	241
Образ действий программ-вымогателей .....	242
Анализ вымогателя Petya .....	244
Получение привилегий администратора .....	244
Заражение жесткого диска (этап 1) .....	245
Шифрование с помощью конфигурационных данных вредоносного начального загрузчика .....	248

Обрушение системы .....	252
Шифрование MFT (этап 2) .....	253
Подводя итоги: заключительные мысли о Petya .....	258
<b>Анализ вымогателя Satana .....</b>	<b>258</b>
Сбрасыватель Satana .....	259
Заражение MBR .....	259
Отладочная информация сбрасывателя .....	260
Вредоносная MBR вымогателя Satana .....	261
Подводя итоги: заключительные мысли о Satana .....	264
<b>Заключение .....</b>	<b>264</b>
<b>Глава 14. Сравнение процессов загрузки с помощью UEFI и MBR/VBR .....</b>	<b>266</b>
Единый расширяемый интерфейс прошивки .....	267
Различия между процессами загрузки через BIOS и UEFI .....	268
Последовательность загрузки .....	268
Разбиение диска на разделы: MBR и GPT .....	269
Прочие отличия .....	270
Особенности таблицы разделов GUID .....	271
Как работает прошивка UEFI .....	275
Спецификация UEFI .....	276
Внутри загрузчика операционной системы .....	278
Начальный загрузчик Windows .....	284
Преимущества прошивки UEFI с точки зрения безопасности .....	287
<b>Заключение .....</b>	<b>288</b>
<b>Глава 15. Современные UEFI-буткиты .....</b>	<b>289</b>
Исторический обзор угроз BIOS .....	290
WinSIN, или первый вредонос, нацеленный на BIOS .....	290
Mebromi .....	291
Краткий обзор других угроз и контрмер .....	292
У любого оборудования есть прошивка .....	296
Уязвимости прошивки UEFI .....	297
Неэффективность битов защиты памяти .....	298
Проверки битов защиты .....	299
Способы заражения BIOS .....	300
Модификация дополнительного ПЗУ неподписанной UEFI .....	302
Добавление или модификация DXE-драйвера .....	304
Как происходит внедрение руткита .....	305
UEFI-руткиты на воле .....	311
Руткит Vector-EDK от группы Hacking Team .....	312
<b>Заключение .....</b>	<b>320</b>
<b>Глава 16. Уязвимости прошивок UEFI .....</b>	<b>321</b>
Почему прошивка может быть уязвимой? .....	322
Классификация уязвимостей UEFI .....	325
Постэксплуатационные уязвимости .....	327
Скомпрометированная цепочка поставок .....	327
Борьба с уязвимостью цепочки поставок .....	329
Исторический обзор защиты прошивок UEFI .....	329
Как работает защита BIOS .....	330

Защита флеш-памяти SPI и ее уязвимости .....	331
Риски неаутентифицированного обновления BIOS .....	334
Защита BIOS с помощью технологии безопасной загрузки.....	335
Intel Boot Guard .....	336
Технология Intel Boot Guard .....	336
Уязвимости Boot Guard .....	337
Уязвимости в модулях SMM.....	339
Что такое SMM.....	339
Эксплуатация обработчиков SMI .....	340
Уязвимости в загрузочном скрипте S3 .....	344
Что делает скрипт S3.....	344
Атаки на слабости загрузочного скрипта S3 .....	345
Эксплуатация уязвимости в загрузочном скрипте S3.....	346
Исправление уязвимости в загрузочном скрипте S3 .....	349
Уязвимости в Intel Management Engine .....	349
История уязвимостей ME .....	349
Атаки на код ME .....	350
Пример: атаки на Intel AMT и BMC.....	351
Заключение.....	354
<b>Часть III. Методы защиты и компьютерно-технической экспертизы .....</b>	<b>355</b>
<b>Глава 17. Как работает безопасная загрузка UEFI .....</b>	<b>356</b>
Что такое безопасная загрузка?.....	357
Детали реализации безопасной загрузки UEFI.....	358
Последовательность загрузки .....	358
Аутентификация исполняемого файла с помощью цифровых подписей .....	359
База данных db .....	361
База данных dbx .....	364
Аутентификация с учетом времени.....	366
Ключи безопасной загрузки .....	366
Безопасная загрузка UEFI: полная картина.....	369
Политика безопасной загрузки.....	370
Защита от буткитов с помощью безопасной загрузки .....	372
Атаки на безопасную загрузку .....	374
Изменение прошивки PI с целью отключения безопасной загрузки .....	374
Модификация переменных UEFI для обхода проверок безопасности.....	375
Защита безопасной загрузки с помощью технологии	
верифицированной и измеренной загрузки.....	377
Верифицированная загрузка.....	378
Измеренная загрузка .....	378
Intel BootGuard .....	378
Где искать ACM .....	379
Изучение FIT.....	382
Конфигурирование Intel BootGuard.....	382
Trusted Boot Board в ARM.....	385
ARM Trust Zone .....	385
Начальные загрузчики в ARM .....	386
Поток выполнения в Trusted Boot.....	388
Верифицированная загрузка и руткиты прошивки .....	389
Заклучение.....	390

<b>Глава 18. Подходы к анализу скрытых файловых систем</b> .....	391
Обзор скрытых файловых систем .....	392
Извлечение данных буткита из скрытой файловой системы .....	393
Извлечение данных из незапущенной системы .....	393
Чтение данных из активной системы .....	394
Подключение к драйверу мини-порта устройства хранения .....	394
Выбор образа скрытой файловой системы .....	400
Программа HiddenFsReader .....	401
Заключение .....	402
<b>Глава 19. Компьютерно-техническая экспертиза BIOS/UEFI: подходы к получению и анализу прошивок</b> .....	403
Ограничения наших методов КТЭ .....	404
Почему компьютерно-техническая экспертиза прошивки так важна .....	404
Атака на цепочку поставок .....	405
Компрометация BIOS через уязвимость прошивки .....	405
Как получить прошивку .....	405
Программный подход к получению прошивки .....	407
Местоположение регистров из конфигурационного пространства PCI .....	408
Вычисление адресов регистров конфигурации SPI .....	409
Использование регистров SPI .....	409
Чтение данных из флеш-памяти SPI .....	412
О недостатках программного подхода .....	413
Аппаратный подход к получению прошивки .....	414
Описание процедуры на примере Lenovo ThinkPad T540p .....	415
Местоположение микросхемы флеш-памяти SPI .....	416
Чтение флеш-памяти SPI с помощью мини-модуля FT2232 .....	418
Анализ образа прошивки с помощью UEFITool .....	420
Какие существуют регионы флеш-памяти SPI .....	421
Просмотр регионов флеш-памяти SPI с помощью UEFITool .....	421
Анализ региона BIOS .....	423
Анализ образа прошивки с помощью Chipsec .....	427
Знакомство с архитектурой Chipsec .....	427
Анализ прошивки с помощью Chipsec Util .....	429
Заключение .....	431
<b>Предметный указатель</b> .....	432



# ОТ ИЗДАТЕЛЬСТВА

## **Отзывы и пожелания**

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com); при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## **Скачивание исходного кода примеров**

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте [www.dmkpress.com](http://www.dmkpress.com) на странице с описанием соответствующей книги.

## **Список опечаток**

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com). Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

## **Нарушение авторских прав**

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и No Starch Press очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

## ОБ АВТОРАХ

**Алекс Матросов** – ведущий специалист по наступательной безопасности в компании NVIDIA. Больше двадцати лет занимается обратной разработкой, продвинутым анализом вредоносных программ, безопасностью на уровне прошивок и методами эксплуатации уязвимостей. До перехода в NVIDIA работал главным исследователем по безопасности в Центре передовых технологий безопасности Intel (SeCoE), больше шести лет работал в группе исследования новых угроз в Intel и занимал должность старшего исследователя по безопасности в компании ESET. Алекс является автором и соавтором многочисленных статей и часто выступает на конференциях по безопасности, в т. ч. REcon, ZeroNights, Black Hat, DEFCON и других. Удостоился награды от компании Hex-Rays за плагин с открытым исходным текстом HexRaysCodeXplorer, который начиная с 2013 года поддерживается командой REhint.

**Евгений Родионов**, PhD, специалист по безопасности в Intel, занимается безопасностью BIOS для клиентских платформ. До этого участвовал во внутренних исследовательских проектах и отвечал за углубленный анализ комплексных угроз в ESET. В сферу его интересов входят безопасность на уровне прошивки, программирование в режиме ядра, технологии противодействия руткитам и обратная разработка. Много раз выступал на конференциях по безопасности, включая Black Hat, REcon, ZeroNights и CARO, является соавтором многочисленных научных статей.

**Сергей Братусь** – научный сотрудник и доцент факультета информатики Дартмутского колледжа. Ранее работал в компании BBN Technologies, где занимался обработкой естественных языков. Братуся интересуют все аспекты безопасности Unix, в особенности безопасность ядра Linux, а также обнаружение и обратная разработка вредоносного ПО в Linux.

## О ТЕХНИЧЕСКОМ РЕЦЕНЗЕНТЕ

**Родриго Рубира Бранко** (BSDaemon) работает главным исследователем по безопасности в корпорации Intel Corporation, где возглавляет группу STORM (Strategic Offensive Research and Mitigations). Родриго обнаружил десятки уязвимостей во многих важных технологиях и опубликовал новаторские работы по эксплуатации, обратной разработке и анализу вредоносных программ. Входит в группу RISE Security Group и является одним из организаторов Hackers to Hackers Conference (H2HC), старейшей конференции по безопасности в Латинской Америке.

# ВСТУПИТЕЛЬНОЕ СЛОВО

Невозможно отрицать тот факт, что вредоносные программы представляют растущую угрозу компьютерной безопасности. Всюду мы видим тревожную статистику, свидетельствующую о росте финансовых потерь, сложности и разнообразии вредоносного ПО. Все больше исследователей, в промышленности и в академических кругах, изучают вредоносное ПО и публикуют свои результаты, пользуясь различными каналами – от блогов и конференций до университетских курсов и книг, посвященных этому предмету. В этих публикациях тема рассматривается под всевозможными углами зрения: обратная разработка, передовые практики, методология и лучшие комплекты инструментов.

Таким образом, дискуссии по поводу инструментов для анализа вредоносного ПО и его автоматизации уже идут и с каждым днем все ширятся. А раз так, возникает вопрос: зачем нужна еще одна книга на эту тему? Что в ней может быть такого, чего нет в других?

Прежде всего, хотя эта книга посвящена обратной разработке передового – я имею в виду *инновационного* – вредоносного ПО, она включает и фундаментальные знания о том, для чего тот или иной фрагмент вредоносной программы вообще написан. В книге объясняются механизмы работы различных обсуждаемых компонентов – от начальной загрузки платформы и загрузки операционной системы до различных частей ядра и программ прикладного уровня, которые все равно рано или поздно обращаются к ядру.

Я сам не раз объяснял, что *фундаментальное* освещение материала – не то же самое, что *базовое*, хотя в обоих случаях речь идет об основополагающих строительных блоках, на которых покоятся компьютеры и вычисления. И под таким углом зрения эта книга – больше, чем обсуждение вредоносного ПО. В ней описывается, как работают компьютеры, как в современных программных стеках используются базовые возможности машины и пользовательские интерфейсы. Зная все это, вы вдруг начинаете *автоматически* понимать, как и почему вещи ломаются и как их можно употребить во вред.

Кто лучше проведет по этому пути, чем авторы, послужной список которых включает распутывание – и неоднократное – по-настоящему изобретательного вредоносного кода, раздвигавшего границы возможного? Добавьте к этому хорошо продуманные и старательные усилия связать свой опыт с основаниями информатики и представить его в более широком контексте, например рассказать о том, как анализировать и классифицировать различные проблемы с концептуально сходными характеристиками, – и вы поймете, почему эта книга должна занять одно из верхних мест в списке ожидающих прочтения.

Но раз содержание и выбранная методология полностью оправдывают потребность в такой книге, то почему никто не написал ее раньше? Я наблюдал за эволюцией этой книги (более того, имел честь быть ее активным участником и, надеюсь, привнес что-то полезное), она заняла несколько лет напряженного труда, несмотря даже на изобилие исходного материала, имевшегося в распоряжении авторов. И я понял, почему никто не попытался написать ее прежде: это не только трудно, но и требует правильного сочетания знаний и навыков (как раз такого, которое есть у авторов), поддержки со стороны редакторов (которую обеспечило издательство No Starch, терпеливо взявшее на себя процесс редактирования и мирившееся с задержками, неизбежными ввиду постоянно меняющейся обстановки в сфере наступательной безопасности) и, наконец, энтузиазма читателей предварительных вариантов книги (которые неустанно гнали работу к финишной черте).

Большое внимание в книге уделено тому, как достигается (или не достигается) доверие в современном компьютере и как различными уровнями и переходами между ними можно злоупотребить, нарушив предположения, принятые следующим уровнем. Задача в том, чтобы высветить две основные проблемы при реализации безопасности: композицию (надлежащее функционирование возможно только при правильном поведении нескольких взаимозависимых уровней) и допущения (каждый уровень должен предполагать, что предыдущий работает правильно). Авторы также делятся своим опытом применения инструментов и подходов к чрезвычайно сложному анализу поведения компонентов, работающих на ранних стадиях загрузки и на самых нижних уровнях операционной системы. Описание такого сквозного межуровневого подхода само по себе достойно отдельной книги и составляет книгу внутри книги. Как читатель я обожаю такие акции «два по цене одного», но лишь немногие авторы их предлагают.

Размышляя о природе знания, я пришел к выводу, что если ты что-то знаешь от и до, то можешь это хакнуть. Применение обратной разработки для понимания кода, который хакает обычное поведение системы, – это потрясающее техническое свершение, которое зачастую приносит много новых знаний. Возможность учиться у профессионалов, в послужном списке которых немало таких свершений и которые готовы поделиться своим пониманием, методами, рекомендациями и опытом, да еще следовать за ними в удобном для себя темпе, – это уникальный шанс. Не упустите его! Идите вглубь: пользуйтесь вспомогательными материалами, практикуйтесь, привлекайте сообщество, друзей и даже профессоров (которые, надеюсь, оценят, сколько полезного эта книга может дать аудитории). Эта книга не просто для чтения – она для изучения.

*Родриго Рубира Бранко*  
(BSDaemon)

# БЛАГОДАРНОСТИ

Мы благодарны всем читателям, купившим предварительные варианты данной книги. Их непрерывная поддержка подгоняла нас вперед, без нее эта книга никогда не была бы закончена. Спасибо всем, кто терпеливо дождался окончательной редакции!

Мы также благодарим всех, кто поддерживал нас в начале этого пути: Дэвида Харли, Джурая Малчо и Якуба Дебски.

Сотрудников издательства No Starch Press, помогавших нам на протяжении пяти лет работы над книгой, так много, что всех не перечислить, поэтому мы выражаем особую признательность Биллу Поллоку (за терпение и внимание к качеству), Лиз Чэдвик и Лорел Чан (без них книга выглядела бы совсем иначе).

Мы высоко ценим отзывы, полученные от Александра Гейзет, Брюса Дэнга, Николая Шлея, Зено Ковача, Алекса Терешкина и всех читателей разных вариантов, приславших свои замечания. Спасибо за указания на опечатки и ошибки, а также за предложения и ободрение.

Огромное спасибо Родриго Рубира Бранко (BSDaemon) за выдающуюся поддержку, техническое рецензирование и вступительное слово.

Мы также благодарны Ильфаку Гульфанову и команде Hex-Rays за поддержку и великолепные инструменты, которыми мы пользовались для анализа обсуждаемых в книге угроз.

Я благодарю свою жену Светлану за поддержку и особенно за терпение, с которым она переносила мои бесконечные отрешенные исследования.

*Алекс Матросов*

Большое спасибо моей семье: жене Евгении и сыновьям Олегу и Леону – за поддержку, воодушевление и понимание.

*Евгений Родионов*

Я обязан многим людям за то, что смог внести свой скромный вклад в эту книгу: авторам и редакторам из журналов Phrack и Uninformed, исследователям из Phenoelit и THC, организаторам и командам Recon, PH-Neutral, Toorcon, Troopers, Day-Con, Shmooscon, Rubi-Con, Berlinsides, H2HC, Sec-T, DEFCON и многим другим. Отдельная благодарность Уильяму Полку, который показал мне, что хакерский подход применим не только к компьютерам, и без чьей помощи я физически не смог бы работать или путешествовать в течение многих лет. И конечно, ничто не могло бы случиться без любви, терпения и поддержки моей жены Анны.

*Сергей Братусь*

# СПИСОК АББРЕВИАТУР

AES	Advanced Encryption Standard (улучшенный стандарт шифрования)
ACM	Authenticated Code Module (модуль аутентифицированного кода)
ACPI	Advanced Configuration and Power Interface (усовершенствованный интерфейс конфигурирования и управления питанием)
AMT	Active Management Technology
APC	asynchronous procedure call (асинхронный вызов процедуры)
APIC	Advanced Programmable Interrupt Controller (расширенный программный контроллер прерываний)
ARM	Advanced RISC Machine
ATA	Advanced Technology Attachment
BCD	Boot Configuration Data (конфигурационные данные загрузки)
BDS	Boot Device Selection (выбор загрузочного устройства)
BIOS	Basic Input/Output System (базовая система ввода-вывода)
BMC	Baseboard Management Controller (контроллер управления материнской платой)
BPB	BIOS Parameter Block (блок параметров BIOS)
BPM	boot policy manifest (манифест политики загрузки)
BSI	boot sector infector (вирус загрузочного сектора)
BSoD	Blue Screen of Death (синий экран смерти)
C&C	command and control (команды и управление)
CBC	cipher block chaining (режим сцепления блоков шифртекста)
CDO	control device object (объект устройства управления)
CHS	Cylinder Head Sector (цилиндр-головка-сектор)
CLR	Common Language Runtime (общезыковая среда выполнения)
COFF	Common Object File Format (формат COFF)
COM	Component Object Model (компонентная объектная модель)
CSM	Compatibility Support Module (модуль поддержки запуска в режиме совместимости)
DBR	DOS Boot Record (загрузочная запись DOS)
DDoS	distributed denial of service (распределенная атака с отказом от обслуживания)
DGA	domain name generation algorithm (алгоритм генерации доменного имени)
DKOM	Direct Kernel Object Manipulation (прямое манипулирование объектом ядра)
DLL	dynamic-link library (динамически компокуемая библиотека)
DMA	direct memory access (прямой доступ к памяти)
DRAM	dynamic random access memory (динамическое запоминающее устройство с произвольной выборкой, ДЗУПВ)
DRM	digital rights management (управление цифровыми правами)
DXE	Driver Execution Environment (среда выполнения драйвера)

EC	Embedded Controller (встраиваемый контроллер)
ECB	Electronic Code Book (режим электронной кодовой книги)
ECC	Elliptic Curve Cryptography (эллиптическая криптография)
EDK	EFI Development Kit (комплект разработки EFI)
EDR	Endpoint Detection and Response (обнаружение и реагирование на угрозы в конечных точках)
EFI	Extensible Firmware Interface (интерфейс расширяемой прошивки)
ELAM	Early Launch Anti-Malware (ранний запуск антивирусного ПО)
ELF	Executable and Linkable Format/Extensible Linking Format (формат исполняемых и объектных файлов)
EPT	Extended Page Tables (расширенные таблицы страниц)
FEK	file encryption key (ключ шифрования файла)
FFS	firmware filesystem (файловая система прошивки)
FIT	Firmware Interface Table (таблицы интерфейсов прошивки)
FPF	field-programmable fuse (программируемый пользователем фьюз)
GDB	GNU Debugger (отладчик GNU)
GDT	Global Descriptor Table (глобальная таблица дескрипторов)
GPT	GUID Partition Table (таблица разделов GUID)
GUID	global unique identifier (глобальный уникальный идентификатор)
HAL	hardware abstraction layer (уровень аппаратных абстракций)
HBA	host-based architecture (серверная архитектура)
HECI	Host-Embedded Controller Interface (интерфейс между ОС и встроенным контроллером)
HIPS	Host Intrusion Prevention System (хостовая система предотвращения вторжений)
HSFC	Hardware sequencing flash control (управление аппаратным заданием последовательности доступа к флеш-памяти)
HSFS	Hardware sequencing flash status (состояния аппаратного задания последовательности доступа к флеш-памяти)
HVCI	Hypervisor-Enforced Code Integrity (целостность кода, гарантируемая гипервизором)
IBV	initial boot block (блок начальной загрузки)
IDT	Interrupt Descriptor Table (таблица дескрипторов прерываний)
IOCTL	Input/Output Control (управление вводом-выводом)
IPL	Initial Program Loader (начальный загрузчик программы)
IRP	input/output request packet (пакет запроса ввода-вывода)
ISH	Integrated Sensor Hub (интегрированный концентратор датчиков)
IV	initialization value (начальное значение)
IVT	Interrupt Vector Table (таблица векторов прерываний)
KEK	key exchange key (ключ для обмена ключами)
KM	key manifest (манифест ключа)
KPP	Kernel Patch Protection (защита ядра от изменения)
LBA	logical block address (логический адрес блока)
LPE	local privilege escalation (расширение локальных привилегий)
MBR	Master Boot Record (главная загрузочная запись)
ME	Management Engine (подсистема процессоров Intel)
MFT	master file table (главная таблица файлов)
MIPS	millions of instructions per second (миллионов команд в секунду)

MSR	model-specific register (модельно-зависимый регистр)
NDIS	Network Driver Interface Specification (спецификация интерфейса сетевого драйвера)
NVRAM	nonvolatile random access memory (энергонезависимое запоминающее устройство с произвольной выборкой)
NX	no-execute (запрет выполнения)
OEM	original equipment manufacturer (изготовитель комплектного оборудования)
OSI	Open Systems Interconnection (стандарт взаимодействия открытых систем)
PCH	Platform Controller Hub (концентратор платформенных контроллеров)
PCR	Platform Configuration Register (регистр конфигурации платформы)
PDO	physical device object (объект физического устройства)
PE	Portable Executable (формат исполняемого файла в Windows)
PEI	Pre-EFI Initialization (фаза, предшествующая инициализации EFI)
PI	platform initialization (инициализация платформы)
PIC	position-independent code (позиционно-независимый код)
PK	platform key (платформенный ключ)
PKI	public key infrastructure (инфраструктура криптографии с открытым ключом)
PMU	Power Management Unit (блок управления питанием)
PnP	plug and play
PoC	proof of concept (доказательство правильности концепции)
POST	Power-On Self-Test (самотестирование при включении питания)
PPI	Pay-Per-Install (оплата по количеству установок)
RCBA	Root Complex Base Address (базовый адрес корневого комплекса)
RCRB	Root Complex Register Block (блок регистров корневого комплекса)
ROP	return-oriented programming (возвратно-ориентированное программирование)
RVI	Rapid Virtualization Indexing (быстрое индексирование виртуализации)
SGX	Software Guard Extensions (расширение архитектуры Intel)
SLAT	Second Level Address Translation (трансляция адресов второго уровня)
SMC	System Management Controller (контроллер управления системой)
SMI	System Management Interrupt (прерывание управления системой)
SMM	System Management Mode (режим управления системой)
SMRAM	system management random access memory (ЗУПВ управления системой)
SPC	Software Publisher Certificate (сертификат издателя ПО)
SPI	Serial Peripheral Interface (последовательный периферийный интерфейс)
SPIBAR	SPI Base Address Register (регистр базового адреса SPI)
SSDT	System Service Descriptor Table (таблица дескрипторов системных служб)
TBB	Trusted Boot Board (плата надежной загрузки)
TDI	Transport Driver Interface (интерфейс транспортного драйвера)
TE	Terse Executable (формат исполняемого файла)



TPM	Trusted Platform Module (доверенный платформенный модуль)
TSA	Time Stamping Authority (уполномоченный по выпуску временных меток)
UAC	User Account Control (управление пользовательскими учетными записями)
UEFI	Unified Extensible Firmware Interface (единый расширяемый интерфейс прошивки)
UID	unique identifier (уникальный идентификатор)
VBR	Volume Boot Record (загрузочная запись тома)
VBS	virtualization-based security (безопасность на основе виртуализации)
VDO	volume device object (объект устройства тома)
VFAT	Virtual File Allocation Table (виртуальная таблица размещения файлов)
VFS	Virtual File System (виртуальная файловая система)
VM	virtual machine (виртуальная машина, VM)
VMM	virtual machine manager (диспетчер виртуальных машин)
VSM	Virtual Secure Mode (виртуальный безопасный режим)
WDK	Windows Driver Kit (комплект разработки драйверов для Windows)
WHQL	Windows Hardware Quality Labs (лаборатория контроля качества оборудования Windows)
WMI	Windows Management Instrumentation (инструментарий управления Windows)

# ВВЕДЕНИЕ



Идея этой книги пришла нам в голову, когда после публикации серии статей и постов в блогах о руткитах и буткитах мы поняли, что эта тема раскрыта куда хуже, чем заслуживает. Мы нутром чуяли, что картина шире, и захотели иметь книгу, которая сводила бы все воедино – обобщала разношерстное собрание трюков, наблюдений за архитектурой операционных систем, паттернов проектирования, применяемых атакующими, и новаторских придумок защитников. Мы искали подобную книгу и не нашли. И тогда решили написать такую, которую сами захотели бы прочитать.

На это у нас ушло четыре с половиной года – дольше, чем мы планировали, и гораздо дольше того срока, который, по нашим прикидкам, могли бы выдержать потенциальные читатели, поддержавшие ознакомительные редакции книги. Если вы один из них и все же читаете эту книгу, мы склоняем голову перед вашей преданностью!

За это время мы наблюдали совместную эволюцию средств нападения и защиты. В частности, мы видели, как Microsoft Windows положила конец нескольким важным направлениям проектирования руткитов и буткитов. Эту историю вы найдете на страницах книги.

Мы также были свидетелями появления новых классов вредоносного ПО, нацеливающихся на BIOS и прошивки чипсетов, до которых текущие средства защиты Windows не могли дотянуться. Мы расскажем об этой совместной эволюции и о том, каковы, скорее всего, будут ее следующие этапы.

Еще одна тема данной книги – развитие методов обратной разработки, нацеленных на ранние стадии процесса загрузки ОС. Так сложилось, что чем раньше выполняется код в длинном процессе загрузки ПК, тем сложнее наблюдать за ним. И эти затруднения долго

путали с безопасностью. Однако же, анализируя буткиты и импланты для BIOS, подрывающие такие низкоуровневые технологии ОС, как Secure Boot, мы видим, что и здесь «безопасность через неведение» ничем не лучше, чем в других разделах информатики. Проходит немного времени (совсем чуть-чуть на временной шкале интернета) – и подход, опирающийся на безопасность через неведение, начинает приносить атакующим больше выгод, чем защитникам. Эта идея еще не получила достаточного освещения в книгах на данную тему, так что мы попытаемся восполнить пробел.

## Для кого предназначена эта книга

Мы адресуем свой труд очень широкому кругу специалистов по информационной безопасности, интересующихся тем, как передовые отряды вредоносного ПО обходят механизмы безопасности на уровне ОС. В фокусе нашего внимания вопросы обнаружения, обратной разработки и эффективного анализа этих продвинутых угроз. Каждая часть книги отражает новый этап эволюционного развития угроз – от появления в виде доказательства правильности концепции до последующего распространения в среде носителей угроз и, наконец, включения в тайный арсенал точно нацеленных атак.

Однако мы ориентируемся на более широкую аудиторию, состоящую не только из аналитиков вредоносного ПО. В частности, мы надеемся, что разработчики встраиваемых систем и специалисты по безопасности облачных систем сочтут книгу полезной, учитывая, сколь опасные масштабы может принять угроза руткитов и других имплантов в их экосистемах.

## Структура книги

Мы начнем с изучения руткитов в части I, где познакомимся с внутренними механизмами ядра Windows, которое исторически послужило площадкой для отработки руткитов. Затем в части II мы сместим акцент на процесс загрузки ОС и буткиты, которые были разработаны после того, как Windows начала укреплять свой режим ядра. Мы разбираем процесс загрузки на этапы с точки зрения атакующего, обращая особое внимание на новые схемы прошивки UEFI и их уязвимости. Наконец, в части III мы поговорим о компьютерно-технической экспертизе классических атак на ОС с помощью руткитов и более современных атак на BIOS и прошивки с помощью буткитов.

### *Часть I. Руткиты*

В этой части описываются классические руткиты уровня ОС во времена их расцвета. Эти исторические примеры руткитов проливают свет на то, как атакующий видит операционную систему изнутри и находит способы надежно внедрить свои импланты, используя собственные структуры ОС.

**Глава 1. Что такое руткит: пример TDL3.** Мы начнем изучение работы руткитов с рассказа об одном из самых интересных руткитов своего времени, основанного на нашем собственном опыте столкновений с различными его вариантами и анализа угроз.

**Глава 2. Руткит Festi: самый продвинутый бот для спама и DDoS-атак.** В этой главе анализируется знаменитый руткит Festi для рассылки спама и организации DDoS-атак, в котором использовались самые передовые на тот момент методы обеспечения скрытности. В частности, руткит включал собственный стек TCP/IP на уровне ядра.

**Глава 3. Обнаружение заражения руткитом.** В этой главе мы продолжим путешествие вглубь ядра операционной системы и расскажем о трюках, с помощью которых атакующие стремятся получить контроль над более глубокими уровнями ядра, например перехватывать системные события и вызовы.

## **Часть II. Буткиты**

Вторая часть книги посвящена эволюции буткитов, условиям, подхлестнувшим эту эволюцию, и методам обратной разработки таких угроз. Мы увидим, как буткиты научились имплантировать себя в BIOS и эксплуатировать уязвимости прошивок UEFI.

**Глава 4. Эволюция буткита.** В этой главе мы подробно рассмотрим движущие силы совместной эволюции, которые породили буткиты и направляли их развитие. Мы опишем некоторые из первых обнаруженных буткитов, в частности знаменитый Elk Cloner.

**Глава 5. Основы процесса загрузки операционной системы.** Рассматриваются внутренние детали процесса загрузки Windows и его изменение со временем. Мы поговорим о главной загрузочной записи, таблицах разделов, конфигурационных данных и модуле *bootmgr*.

**Глава 6. Безопасность процесса загрузки.** Эта глава представляет собой обзор технологий защиты процесса загрузки Windows, в частности модули раннего запуска антивирусного ПО (Early Launch Anti-Malware – ELAM), политику подписания кода режима ядра и ее уязвимости, а также новые средства безопасности на основе виртуализации.

**Глава 7. Методы заражения буткитом.** В этой главе мы тщательно проанализируем методы заражения загрузочных секторов и посмотрим, как они видоизменялись со временем. В качестве примеров будем использовать хорошо известные буткиты: TDL4, Gapz и Rovnix.

**Глава 8. Статический анализ буткита с помощью IDA Pro.** Здесь рассматриваются методы и инструменты статического анализа заражения буткитом. Для примера мы подвергнем анализу буткит TDL4 и предоставим материалы, которые вы сможете ис-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)