КРИС КАСПЕРСКИ

bhv

ТЕХНИКА ОТЛАДКИ ПРОГРАММ

БЕЗ ИСХОДНЫХ ТЕКСТОВ

ОСНОВНОЙ ИНСТРУМЕНТАРИЙ ХАКЕРА

ВЗЛОМ ПРОГРАММ С ЗАКРЫТЫМИ ГЛАЗАМИ

"ПРОТИВОУГОННЫЕ" СИСТЕМЫ СВОИМИ РУКАМИ

БОРЬБА С КРИТИЧЕСКИМИ ОШИБКАМИ ПРИЛОЖЕНИЙ

ВНЕДРЕНИЕ И УДАЛЕНИЕ ВИРУСНОГО КОДА ИЗ PE-ФАЙЛОВ

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

+CD

PRO

ПРОФЕССИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Крис Касперски

ТЕХНИКА ОТЛАДКИ ПРОГРАММ

БЕЗ ИСХОДНЫХ ТЕКСТОВ

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06
ББК 32.973.26-018.1
К28

Касперски К.

К28 Техника отладки программ без исходных текстов. — СПб.: БХВ-Петербург, 2005. — 832 с.: ил.

ISBN 5-94157-229-8

Даны практические рекомендации по использованию популярных отладчиков, таких как NuMega SoftIce, Microsoft Visual Studio Debugger и Microsoft Kernel Debugger. Показано, как работают отладчики и как противостоять дизасемблированию программы. Описаны основные защитные механизмы коммерческих программ, а также способы восстановления и изменения алгоритма программы без исходных текстов. Большое внимание уделено внедрению и удалению кода из PE-файлов. Материал сопровождается практическими примерами.

Компакт-диск содержит исходные тексты приведенных листингов и полезные утилиты.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. гл. редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Елена Кашлакова*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Наталия Першакова*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 15.08.05.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 67,08.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-229-8

Оглавление

Предисловие 1

Об авторе 1
О чем и для кого эта книга 3

Введение 11

История хакерства 11
История происхождения термина "хакер" 14
Психология хакера 16
Лаборатория искусственного интеллекта и PDP-1 20
Сеть 23
Си и UNIX 26
Конец хакеров шестидесятых 33
RSX-11M 36
Intel 37
Хаос 38
Бытовой компьютер восьмидесятых 40
Рождение современных хакеров, или снова Intel 41

Глава 1. Знакомство с отладочными инструментами 45

1.1. Как работает отладчик 48
Обработка исключений 50
1.2. Что нам понадобится 51
1.3. Особенности отладки в UNIX 53
PTrace — фундамент для GDB 56
PTrace и ее команды 58
Поддержка многопоточности в GDB 60
Краткое руководство по GDB 61
Трассировка системных функций 66
Интересные ссылки 67

1.4. Эмулирующие отладчики и эмуляторы ... 68
Минимальные системные требования ... 70
Выбирай эмулятор себе по руке! ... 71
1.5. Обзор эмуляторов ... 74
DOSBox ... 74
Bochs ... 76
Microsoft Virtual PC ... 77
VMware ... 79
Сводная таблица характеристик эмуляторов ... 80
Разные мелочи ... 81
1.6. Области применения эмуляторов ... 82
Пользователям ... 82
Администраторам ... 83
Разработчикам ... 84
Хакерам ... 87
Как настроить SoftIce под VMware ... 89
Экзотические эмуляторы ... 89
1.7. Кратко об эмуляции процессора ... 90
1.8. BoundsChecker ... 96
Быстрый старт ... 98
Подключение нестандартных DLL ... 101
Пункты меню ... 103
1.9. Хакерские инструменты под UNIX ... 107
Отладчики ... 107
Дизассемблеры ... 111
Шпионы ... 112
Шестнадцатеричные редакторы ... 114
Дамперы ... 115
Автоматизированные средства защиты ... 115

Глава 2. Защитные механизмы и их отладка ... 119

2.1. Классификация защит по роду секретного ключа ... 120
2.2. Создаем защиту и пытаемся ее сломать ... 123
2.3. От exe до crk ... 125
2.4. Знакомство с отладчиком ... 142
Бряк на оригинальный пароль ... 143
Прямой поиск введенного пароля в памяти ... 160
Бряк на функции ввода пароля ... 170
Бряк на сообщения ... 173
Механизм сообщений в Windows 9*x* ... 176
2.5. На сцене появляется IDA ... 177
2.6. Дизассемблер & отладчик в связке ... 209
Из языка IDA-Си ... 212

2.7. Дао регистрационных защит ... 216
Как узнать имя функции по ординалу ... 221
Как сделать исполняемые файлы меньше ... 248
Перехват *WM_GETTEXT* ... 249
2.8. Хеширование и его преодоление ... 251
2.9. Ограничение возможностей ... 267
2.10. Ограничение времени использования ... 286
2.11. Ограничение числа запусков ... 291
2.12. NagScreen ... 293
2.13. Ключевой файл ... 302

Глава 3. Противостояние отладке ... 315

3.1. Обзор способов затруднения анализа программ ... 317
3.2. Приемы против отладчиков реального режима ... 319
3.3. Приемы против отладчиков защищенного режима ... 335
3.4. Как противостоять трассировке ... 348
3.5. Как противостоять контрольным точкам останова ... 354
Несколько грязных хаков, или как не стоит защищать свои программы ... 362
Серединный вызов API-функций ... 363
Вызов API-функций через мертвую зону ... 382
Копирование API-функций целиком ... 385
Как обнаружить отладку средствами Windows ... 388
3.6. Антиотладочные приемы под UNIX ... 389
Паразитные файловые дескрипторы ... 390
Аргументы командной строки и окружение ... 391
Дерево процессов ... 392
Сигналы, дампы и исключения ... 392
Распознавание программных точек останова ... 393
Мы трассируем, нас трассируют ... 394
Прямой поиск отладчика в памяти ... 395
Измерение времени выполнения ... 396
3.7. Основы самомодификации ... 396
Проблемы обновления кода через Интернет ... 409
3.8. Неявный самоконтроль как средство создания неломаемых защит ... 411
Техника неявного контроля ... 413
Практическая реализация ... 415
Как это ломают? ... 425
3.9. Ментальная отладка и дизассемблирование ... 435
Маленькие хитрости ... 459
3.10. Ментальное ассемблирование ... 460
3.11. Краткое руководство по защите ПО ... 465
Джинн из бутылки, или недостатки решений из коробки ... 466

Защита от копирования, распространения серийного номера ... 466
Защита испытательным сроком ... 467
Защита от реконструкции алгоритма ... 467
Защита от модификации на диске и в памяти ... 469
Антидизассемблер ... 470
Антиотладка ... 470
Антимонитор ... 471
Антидамп ... 471
Как защищаться ... 473
Мысли о защитах ... 474
Противодействие изучению исходных текстов ... 474
Противодействие анализу бинарного кода ... 478
3.12. Как сделать свои программы надежнее? ... 481
Причины и последствия ошибок переполнения ... 482
Переход на другой язык ... 483
Использование кучи для создания массивов ... 484
Отказ от индикатора завершения ... 484
Обработка структурных исключений ... 485
Традиции и надежность ... 487
Как с ними борются? ... 488
Поиск уязвимых программ ... 489
Неудачный выбор приоритетов в Си ... 493
3.13. Тестирование программного обеспечения ... 495
Тестирование на микроуровне ... 497
Регистрация ошибок ... 498
Бета-тестирование ... 499
Вывод диагностической информации ... 502
Верификаторы кода языков Си/Си++ ... 504
Демонстрация ошибок накопления ... 505

Глава 4. Примеры реальных взломов ... 509

4.1. Intel C++ 5.0.1 compiler ... 510
4.2. Intel Fortran 4.5 ... 518
4.3. Intel C++ 7.0 compiler ... 523
4.4. Record Now ... 532
4.5. Alcohol 120% ... 535
4.6. UniLink v1.03 от Юрия Харона ... 549
UniLink v1.03 от Юрия Харона II, или переходим от штурма к осаде ... 571
Entry Point и ее окружение ... 572
Передача управления по структурному исключению ... 574
Внутри обработчика ... 581
Таинства stealth-импорта API-функций,
или как устроена *HaronLoadLibrary* ... 586

Таинства stealth-импорта: как устроена *HaronGetProcAddress* ... 589
Таинства *IsDebuggerPresent* ... 599
Таинства загрузки USER32.DLL и ADVAAPI32.DLL ... 601
Конец таинств, или где тот trial, который expired ... 604
4.7. ARJ ... 613
4.8. AVPVE: разбор полетов ... 614
Формат файлов (общее) ... 615
Формат файла языковой поддержки avp.lng ... 616
Формат файлов HLP ... 617
4.9. Bounds-Checker 5 ... 620
4.10. CD-MAN EGA Version ... 621
4.11. F-PROT 2.19 ... 622
4.12. FDR 2.1 ... 627
4.13. HEXEDIT.EXE Version 1.5 ... 629
4.14. SGWW password protection WhiteEagle ... 630
4.15. SOURCER 5.10 ... 631
4.16. Emulated Solar CPU ... 632
4.17. POCSAG 32 ... 635

Глава 5. Критические ошибки приложений и операционной системы ... 639

5.1. Приложения, недопустимые операции и все-все-все ... 640
Доктор Ватсон ... 642
Отладчик Microsoft Visual Studio Debug ... 649
5.2. Обитатели сумеречной зоны, или из морга в реанимацию ... 650
Принудительный выход из функции ... 651
Раскрутка стека ... 654
Передача управления на функцию обработки сообщений ... 658
5.3. Как подключить дамп памяти ... 666
Восстановление системы после критического сбоя ... 676
Подключение дампа памяти ... 677

Глава 6. Формат PE-файлов ... 683

6.1. Особенности структуры PE-файлов в конкретных реализациях ... 684
6.2. Общие концепции и требования, предъявляемые к PE-файлам ... 686
6.3. Структура PE-файла ... 689
6.4. Что можно и что нельзя делать с PE-файлом ... 692
6.5. Описание основных полей PE-файла ... 695
[old-exe] e_magic ... 695
[old-exe] e_cparhdr ... 695
[old-exe] e_lfanew ... 696
[image_file_header] Machine ... 696
[image_file_header] NumberOfSections ... 696
[image_file_header] PointerToSymbolTable/NumberOfSymbols ... 697

[image_file_header] SizeOfOptionalHeader ... 697
[image_file_header] Characteristics ... 698
[image_optional_header] Magic ... 700
[image_optional_header] SizeOfCode/SizeOfInitializedData/SizeOfUninitializedData ... 700
[image_optional_header] BaseOfCode/BaseOfData ... 701
[image_optional_header] AddressOfEntryPoint ... 701
[image_optional_header] ImageBase ... 702
[image_optional_header] FileAlignment/SectionAlignment ... 702
[image_optional_header] SizeOfImage ... 703
[image_optional_header] SizeOfHeaders ... 703
[image_optional_header] CheckSum ... 704
[image_optional_header] Subsystem ... 704
[image_optional_header] DllCharacteristics ... 705
[image_optional_header] SizeOfStackReserve/SizeOfStackCommit, SizeOfHeapReserve/SizeOfHeapCommit ... 706
[image_optional_header] NumberOfRvaAndSizes ... 706
DATA_ DIRECTORY ... 707
Таблица секций ... 709
Экспорт ... 714
Импорт ... 718
Перемещаемые элементы ... 727

Глава 7. Техника внедрения и удаления кода из PE-файлов ... 731

7.1. Понятие X-кода и другие условные обозначения ... 732
7.2. Цели и задачи X-кода ... 733
7.3. Требования, предъявляемые к X-коду ... 736
7.4. Внедрение ... 737
Предотвращение повторного внедрения ... 738
Классификация механизмов внедрения ... 740
Категория A: внедрение в пустое место файла ... 741
Внедрение в регулярную последовательность байтов ... 750
Категория A: внедрение путем сжатия части файла ... 756
Категория A: создание нового NTFS-потока внутри файла ... 758
Категория B: растяжение заголовка ... 761
Категория B: сброс части секции в оверлей ... 763
Категория B: создание своего собственного оверлея ... 767
Категория C: расширение последней секции файла ... 768
Категория C: создание своей собственной секции ... 771
Категория C: расширение серединных секций файла ... 773
Категория Z: внедрение через автозагружаемые DLL ... 776

ПРИЛОЖЕНИЯ 777

Приложение 1. Разгон и торможение Windows NT 779

Структура ядра 780
Типы ядер 782
Почему непригодны тестовые пакеты 784
Обсуждение методик тестирования 786
Разность таймеров 787
Синхронизация 791
ACPI и IRQ 792
Переключение контекста 796
Длительность квантов 802
Обсуждение полученных результатов 805

Приложение 2. Практические советы по восстановлению системы в боевых условиях 807

Аппаратная часть 808
Оперативная память 809
Блок питания 811
И все-все-все 812

Приложение 3. Описание компакт-диска 815

Предметный указатель 817

Предисловие

Взлом — это естественная потребность всякого разумного существа. Тернистый путь познания истинной сути вещей проходит через их разрушение. Оглянитесь вокруг: физики-атомщики расщепляют ядра так, что брызги материи летят, химики-аналитики разбивают длинные молекулы на множество мелких, математики активно используют метод декомпозиции. И никто из них не заслуживает порицания!

Хакерство — это не вандализм. Это проявление природного любопытства к познанию окружающего нас мира. Дизассемблерные листинги, машинные команды, черные экраны SoftIce, напоминающие о первой молодости MS-DOS — все это безумно интересно и увлекательно. Посреди них раскинулся целый мир скрывающих механизмов и защитных кодов. Не ищите его на картах — он существует лишь в обрывках беспорядочно разбросанных по полу распечаток, технических руководствах, автоматически открывающихся на самых интересных местах, и конечно же, многочисленных бессонных ночах, проведенных у монитора.

Это не учебник по взлому и не руководство по защите от хакеров. Таких книг уже написано предостаточно. Баста! Надоело! Перед вами лежат путевые заметки кодокопателя, своеобразный сборник любопытных историй, произведших с мыщъ'ем в киберпространстве. Вы побываете и внутри компиляторов фирмы Intel, и внутри защитных механизмов коммерческих программ, узнаете, как работает отладчик и как правильно держать его в руках. В общем, если не струсите и не отбросите эту книгу прочь, вас ожидает много интересного.

Об авторе

Небрежно одетый мыщъх 28 лет, не обращающий внимания ни на мир, ни на тело, в котором живет, и обитающий исключительно в дебрях машинных кодов и зарослях технических спецификаций. Не общителен, ведет замкнутый

образ жизни хищного грызуна, практически никогда не покидающего свою норку — разве что на звезды посмотреть или на луну (повыть). Высшего образования нет, а теперь уже и не будет; личная жизнь не сложилась, да и вряд ли сложится, так что единственный способ убить время from dusk till dusker — это полностью отдаться работе.

Одержим компьютерами еще со старших классов средней школы (или еще раньше — уже, увы, не помню). Основная специализация — реинжениринг (дизассемблирование), поиск уязвимостей (попросту говоря, "дыр") в существующих защитных механизмах и разработка собственных систем защит. Впрочем, компьютеры — не единственное и, вероятно, не самое главное увлечение в моей жизни. Помимо возни с железом и блужданий в непроходимых джунглях защитного кода, я не расстаюсь с миром звезд и моих телескопов, много читаю, да и пишу тоже (в последнее время как-то больше пишу, чем читаю). Хакерские мотивы моего творчества не случайны и объясняются по-детски естественным желанием заглянуть "под капот" компьютера и малость потыкать его ломом и молоточком, разумеется, фигурально — а как же иначе понять, как эта штука работает?

Если считать хакерами людей, одержимых познанием окружающего мира, то я — хакер.

О чем и для кого эта книга

> If you do accept the society where we are compelled to live, its awfully egoistic way of life and its dirty "profit" values, you may eventually learn how to disable some simple protections, but you'll never be able to crack in the "right" way. You must learn to despise money, governments, televisions, trends, opinion-makers, public opinion, newspapers and all this preposterous, asinine shit if you want to grasp the noble art, coz in order to be emphatic with the code you must be free from all trivial and petty conventions, strange as it may sound. So you better take a good look around you... you'll find plenty of reasons to hate society and act against it, plenty of sparks to crackle programs in the right way... Hope all this did not sound too cretin.
>
> *+ORC an526164@anon.penet.fi*

Изначально эта книга позиционировалась как хрестоматия для профессионалов, однако пробная публикация отдельных глав в сети профессионалам пришлась не по вкусу. Им не понравилось большое количество "воды" и слишком "разжеванные", с их точки зрения, объяснения. Начинающие кодокопатели резонно возражали, что для одного — "вода", для другого — хлеб, пиво и каша в придачу. Понятное дело, каждый читатель хотел видеть книгу такой, какая была бы наиболее удобна ему одному, но удовлетворить интересы всех категорий читателей в одной-единственной книге (к тому же, не претендующей на полноту и новизну излагаемой информации) — невозможно.

Основную ставку автор делает на начинающих — как на наиболее многочисленную и благодарную аудиторию. Профессионалы же вообще не нуждаются в подобных книгах. "Есть, — говорили они мне, — у тебя с десяток интересных страниц, но они размазаны по всему тексту, и потому читать такую книгу можно только по диагонали в порядке общего ознакомления". Нет, не подумайте, что такие заявления меня обидели! Напротив, помогли лучше понять свое место и предназначение.

Чего греха таить — до звания профессионала автору еще очень далеко, и потому позиционировать свои книги для той аудитории, к которой он не принадлежит, мягко говоря, нетактично. Правда, понятия профессионала и начинающего очень условны, и многие начинающие легко уделывают иных профессионалов. Племя подлинных профессионалов очень малочисленно и невелико — в прямом смысле слова, считанные единицы. Так что невоз-

можно сказать заранее: найдете ли вы что-то новое в данной книге или нет. Единственный способ выяснить это — купить ее и прочитать.

Эта книга не для кракеров! Несмотря на то, что в ней рассматриваются и даются в виде законченных технологий механизмы атак на широко распространенные системы, это просто информация, а не руководство к действию. В любом случае правовую ответственность за компьютерный вандализм еще никто не отменял, и прежде чем использовать полученные знания на практике, неплохо бы ознакомиться с уголовным кодексом и запастись адвокатом. В идеальном обществе принимаемые законы лишь закрепляют уже сложившуюся систему отношений, защищая при этом интересы большинства. А чего хочет большинство? Правильно! Хлеба и зрелищ! Ну, с хлебом все более или менее понятно. Даже в странах третьего мира от голода практически никто не умирает. Со зрелищами же ситуация значительно сложнее. На фоне катастрофической деградации аудио/видеоиндустрии борьба с пиратством приобретает воистину угрожающий размах, ущемляя интересы как отдельно взятых пользователей, так и всего мирового сообщества в целом. Власть захватили медиамагнаты, и демократия умерла. Сильные мира сего лоббируют законы, служащие паре десятков миллиардеров и противоречащие интересам остальных. Ряд научных и инженерных исследований в области информационной безопасности частично или полностью запрещен. Потребитель даже не имеет права заглянуть дизассемблером в тот продукт, который ему впаривают. Look, but don't touch! Touch, but don't taste! Taste, but don't swallow! Ситуация дошла до своего логического абсурда, и в воздухе запахло бунтом. Бунтом против тоталитаризма "демократического режима", бунтом против авторского и патентного права, когда один пронырливый коммерсант отнимает у человечества то, что ему принадлежит по праву. Информация — общедоступный ресурс, такой же, как вода и воздух. Мы дети своей культуры. Наши мысли и суждения, которые мы искренне считаем своими, на самом деле представляют комбинацию уже давно придуманного и высказанного. Удачные находки, яркие идеи — все это результат осмысления и переосмысления когда-то услышанного или прочитанного. Вспомните свои разговоры в курилках — даже располагая диктофонной записью, невозможно установить, кто первым высказал идею, а кто ее подхватил. Знание — продукт коллективного разума. Никто не должен единолично им владеть.

Взлом защиты — это выражение протеста против насилия и несправедливости. Сажать за него можно, но бесполезно. Говорят, что в СССР одного человека посадили за то, что он сказал: "У нас нет демократии". А ведь взломщики и защитники информации не только враги, но еще и коллеги. Хакерство и программирование действительно очень тесно переплетены. Создание качественных и надежных защитных механизмов требует навыков низкоуровневой работы с операционной системой, драйверами и оборудованием; знаний архитектуры современных процессоров и учета особенно-

стей кодогенерации конкретных компиляторов, помноженных на "биологию" используемых библиотек. На этом уровне программирования грань между собственно самим программированием и хакерством становится настолько зыбкой и неустойчивой, что я не рисковал бы ее провести.

Начнем с того, что всякая защита, равно как и любой другой компонент программного обеспечения, требует тщательного и всестороннего тестирования на предмет выяснения ее работоспособности. Под работоспособностью в данном контексте понимается способность защиты противостоять квалифицированным пользователям, вооруженным хакерским арсеналом (копировщиками защищенных дисков, эмуляторами виртуальных приводов, оконными шпионами и шпионами сообщений, файловыми мониторами и мониторами реестра). Качество защиты определяется отнюдь не ее стойкостью, но соотношением трудоемкости реализации защиты к трудоемкости ее взлома. В конечном счете, взломать можно любую защиту — это только вопрос времени, денег, квалификации взломщика и усилий, но грамотно реализованная защита не должна оставлять легких путей для своего взлома. Конкретный пример. Защита, привязывающаяся к сбойным секторам (которые действительно уникальны для каждого носителя), бесполезна, если не способна распознать их грубую эмуляцию некорректно заполненными полями EDC/ECC. Еще более конкретный пример. Привязка к геометрии спиральной дорожки лазерного диска, даже будучи реализованной без ошибок, обходится путем создания виртуального CD-ROM привода, имитирующего все особенности структуры оригинального диска. Для этого даже не нужно быть хакером — достаточно запустить Alcohol 120%, ломающий такие защиты автоматически.

Ошибки проектирования защитных механизмов очень дорого обходятся их разработчикам, но гарантированно застраховаться от подобных просчетов — невозможно. Попытка применения "научных" подходов к защите программного обеспечения — чистейшей воды фарс и бессмыслица. Хакеры смеются над академическими разработками в стиле "расчет траектории сферического коня в вакууме", и практически любая такая защита снимается за 15 минут без напряжения извилин. Вот грубый, но наглядный пример. Проектирование оборонной системы военной крепости без учета существования летательных средств позволяет захватить эту самую крепость чуть ли не на простом "кукурузнике" (MS WDB — кукурузник), не говоря уже об истребителях (SoftIce — истребитель, а IDA Pro — еще и бомбардировщик).

Для разработки защитных механизмов следует иметь хотя бы общее представление о методах работы и техническом арсенале противника, а еще лучше — владеть этим арсеналом не хуже противника (т. е. владеть им в совершенстве). Наличие боевого опыта (реально взломанных программ) очень и очень желательно — пребывание в шкуре взломщика позволяет досконально изучить тактику и стратегию наступательной стороны, давая тем самым возможность оптимальным образом сбалансировать оборону. Попросту

говоря, определить и усилить направления наиболее вероятного вторжения хакеров, сосредоточив здесь максимум своих интеллектуальных сил. А это значит, что разработчик защиты должен глубоко проникнуться психологией хакеров, настолько глубоко, чтобы начать мыслить, как хакер.

Таким образом, владение технологией защиты информации предполагает владение технологией взлома. Не зная того, как ломаются защиты, не зная их слабых сторон, не зная арсенала хакеров, невозможно создать стойкую, дешевую, и главное — простую в реализации защиту. Поэтому, описывая технику защиты, было бы, по меньшей мере, нечестно замалчивать технику ее взлома.

Существует мнение, что открытые публикации о дырах в системах безопасности приносят больше вреда, чем пользы, и их следует в обязательном порядке запретить. Другими словами, достойную защиту от копирования мы создать не можем, признавать свои ошибки — не хотим, и чтобы цивилизацию не разрушил первый же залетный дятел, мы должны перестрелять всех дятлов до единого. Вовсе не собираясь апеллировать к заезженной пословице "кто предупрежден — тот вооружен", обратимся за советом к фармацевтической индустрии. Как бы она отнеслась к появлению рекламы, пропагандирующей некий никем не проверенный, но зато невероятно эффективный препарат, исцеляющий немыслимое количество заболеваний и при этом обязательный к применению? Проводить химический анализ препарата вы не имеете права, равно как не имеете права открыто публиковать результаты своих исследований, выявивших, что за личиной "панацеи" скрывается обыкновенный и довольно низкокачественный аспирин с кучей посторонних примесей и целым "букетом" побочных эффектов. Еще бы! Ведь публикации подобного рода заметно охлаждают потребительский пыл, и предпочтение отдается другим препаратам.

Кто виноват: фирма, обманывающая своих покупателей, или исследователи, открывшие покупателям глаза? Если аналогия между фармацевтикой и программным обеспечением кому-то покажется некорректной, пусть он ответит на вопрос: какие задачи решают защитные механизмы и каким требованиям они должны отвечать? Всякая технология имеет свои ограничения и свои побочные эффекты. Рекламные лозунги, позиционирующие защиту как стойкую и абсолютно непрошибаемую, всегда неверны. Коль скоро носитель можно воспроизвести, можно его и скопировать. Весь вопрос в том — как? Запрет на хакерскую деятельность ничего не меняет. Тех, кто занимается несанкционированным клонированием дисков в промышленных масштабах, подобные запреты вряд ли остановят, а вот легальные исследователи будут страдать: профессиональный зуд, видите ли, дает о себе знать. Ну что поделаешь, есть на земле такая категория людей, что не может удержаться от соблазна заглянуть под крышку черного ящика и, дотрагиваясь до вращающихся шестеренок, пытается разобраться: как же все это, блин, работает? Специфика защитных механизмов состоит в том, что дерьмо визуально

ничем не отличается от шедевра. Вам остается уповать лишь на авторитет поставщика или же методично приобретать все продукты один за другим, при этом не будучи уверенным в том, что на рынке вообще присутствуют достойные защитные механизмы. И это действительно так! Качество коммерческих защит настолько низко, что они оказываются не в стоянии справиться с автоматическими копировщиками программ, запущенными обыкновенными пользователями, коих миллионы. Стоит ли говорить, что некопируемость автоматически копировщиками — это минимально разумное требование, предъявляемое ко всякой защите? В идеале же защита должна противостоять квалифицированным хакерам, вооруженных всем необходимым арсеналом программно-аппартаных средств взлома? Качественные защитные алгоритмы есть, но они не представлены на рынке. Почему? Да все потому, что отсутствие достоверной информации о стойкости тех или иных защитных пакетов не позволяет потребителю осуществлять сознательный выбор. А раз так — зачем производителям напрягаться?

Адептам авторского права важно понять: чем интенсивнее ломаются защитные механизмы, тем стремительнее они совершенствуются! Поскольку у разработчиков появляется реальный стимул к созданию действительно качественных и конкурентоспособных защитных пакетов! Начнем с малого: правило Кирхгофа — базовое правило для всех криптографических систем — гласит: стойкость шифра определяется только секретностью ключа. Криптоаналитику известны все детали процесса шифрования и дешифрования, кроме секретного ключа. Отличительный признак качественной защиты — подробное описание ее алгоритма. В самом деле, глупо скрывать то, что каждый хакер может добыть с помощью отладчика, ящика пива и дизассемблера. Собственно говоря, скрупулезное изучение подробностей функционирования защитного механизма можно только приветствовать. В конце концов существует такое понятие, как полнота представления сведений о товаре, и попытка сокрытия явных конструктивных дефектов, строго говоря, вообще не законна. Всякой Хорошей Вещи гласность только на руку, а вот Плохие Вещи боятся ее, как огня.

Апелляция к "цифровой эпохе" и якобы вытекающая отсюда необходимость пересмотра законов — это грязное словоблудие и ничего более. Большое количество судебных исков, вчиненных вполне легальным исследователям защищенных программ, не может не удивлять. Несмотря на то, что подавляющее большинство подобных исков решается мирным путем, сама тенденция выглядит довольно угрожающей. Чего доброго завтра и <Shift>, как "хакерскую" клавишу запретят, поскольку она позволяет отключить автозапуск лазерного диска в OS Windows, а некоторые защитные механизмы как раз на этом и основаны. То, что еще позавчера казалось фантасмагорическим бредом, вчера вылилось в реальный судебный иск.

Закон DMCA (Digital Millennium Copyright Art) действительно запрещает распространять технологии, устройства, продукты и услуги, созданные с целью

обходить существующие системы защиты, что выглядит вполне логично. Правозащитники пытаются уберечь мир от очевидных преступников и вандалов, однако, следует разделять взлом как таковой и исследовательскую деятельность в области информационной безопасности. Взлом, преследующий личную выгоду или совершаемый из хулиганских побуждений, достоин, по меньшей мере, порицания, а по большей — штрафа или тюремного заключения. Причем тяжесть наказания должна быть сопоставима с величиной причинного ущерба. Приравнивать хакеров к террористам, право же, не стоит! Терроризм, равно как и хакерство, демонстрирует катастрофическую неспособность правительства обеспечить должный уровень безопасности, подчеркивая чрезмерную сосредоточенность власть имущих на своих собственных интересах и проблемах. Все! Никаких других точек соприкосновения у террористов и хакеров нет!

Книги, рассматривающие вопросы безопасности исключительно со стороны защиты, грешат тем же, что и конструкторы запоминающих устройств, работающих только на запись: ни то, ни другое не имеет никакого практического применения. Но эта книга не предназначена для разработчиков защит! Я не хотел делать упор ни на одну из категорий, потому что проблемы защиты информации скорее организационные, чем технические. Разработчики ПО не нуждаются в таких книгах и вообще редко прислушиваются к советам по реализации защитных алгоритмов, а хакеры от всего этого просто балдеют и морально разлагаются, теряя стимул к развитию.

Термин "хакер" имеет множественные трактовки, которые было бы бессмысленно перечислять здесь и, тем более, останавливаться на какой-то из них, игнорируя остальные. Но в лучшем значении хакер — это индивидуал, смотрящий в корень и стремящийся разобраться во всем до конца. Для таких людей и предназначена эта книга. Везде, где это только возможно, я буду стремиться к обобщению и постараюсь не акцентировать внимание на конкретных реализациях. Это не означает, что в книге не встретится законченных схем. Напротив, в них не будет недостатка. Но я не ставлю перед собой цель снабдить хакера готовым инструментарием.

Я не буду пытаться научить читателей "ловить рыбу", а рискну пойти немного дальше и привить некоторые навыки самообучения. Из любой самой нестандартной ситуации и при самом скудном инструментарии всегда можно найти выход. Типовые схемы часто оказываются бессильными и бесполезными. Любые навыки слишком привязаны к конкретному окружению. В наше время больших перемен уже поздно хвататься за традиционные схемы обучения. На обучение просто нет времени. Большинству программистов приходится осваивать новые технологии на ходу, и задолго до конца обучения они уже полностью устаревают.

Системы защиты, заметим, развиваются заметно более медленно, и ничего принципиально нового за последние год-два не было придумано или широко внедрено. Это обнадеживает, особенно на фоне деградации качества реа-

лизации широко распространенных систем безопасности. С другой стороны, толковых учебников и изданий по данной тематике тоже не выходило. Это привело к тому, что действительно грамотно спроектированных и качественно реализованных защит сегодня на массовом рынке практически не наблюдается. Большинство авторов, с трудом припоминая школьный курс алгебры, разрабатывают собственные алгоритмы, которые при близком рассмотрении криптостойкими не являются. Все математическое богатство валяется бесхозным и нетронутым, хотя на рынке достаточно много качественных и вполне современных изданий. Я не стремлюсь повторить уже проделанную работу, а просто буду отсылать по ходу дела к конкретным источникам.

Цель этой книги — научить читателя самостоятельно добывать необходимые ему знания и навыки, порой не имея соответствующей литературы и информации. Современное информационное изобилие приводит к атрофированию навыков самостоятельного получения необходимых знаний. Парадоксально на первый взгляд, но недостаток литературы развивает и тренирует мозги куда лучше, чем ее избыток. Сам я осваивал ассемблер 8086 с помощью утилиты debug.com: кроме нее (и свободного времени) у меня в ту пору не было ничего. Логика работы команд изучалась анализом воздействия последних на регистры и память. Эта было утомительное занятие, и свободное владение ассемблером (без учета ряда некритичных команд) ко мне пришло приблизительно через три месяца. Наличие инструкции сократило бы этот срок до двух-трех дней (с учетом знания ассемблера других платформ), но зато не дало бы никаких полезных навыков.

Инструкция исключительно редко бывает исчерпывающей и доступной. Полученные давным-давно навыки актуальны для меня до сих пор, ибо тенденция обратной зависимости качества инструкции от ее объема в последнее время стала угрожающе превращаться из метафоры в реальность. Главный психологический барьер для многих — это ощущение беспомощности перед компьютером. Человек ощущает себя не хозяином ситуации, а незадачливым экзаменуемым, наобум пытающимся угадать, чего же от него хочет машина. Знакомая ситуация: вызываемая функция упрямо ведет себя не так, как описано в документации, и не работает.

В данном случае не помешает прибегнуть к дизассемблированию и анализу ситуации, но, возможно, оптимальным окажется другое решение. В любом случае возможны свои нетривиальные варианты. Наивно полагать, что схема "нажал на кнопку — получил банан" работает во всех ситуациях. Или что банан непременно удастся достать с помощью палки либо ящика. А если нет? Если задача не имеет известного и опробованного решения? Вот тогда и приходится действовать, как говорят, "по обстоятельствам". Научить этому читателя — вот моя задача.

Введение

История хакерства

> I remember the good old days, when computers were mainframes, analysts were magicians, and programmers punced cards.
>
> *Philip Fites, Peter Johnston, Martin Kratz. "Computer viruses"*

Термин "хакер" прочно вошел в разговорный лексикон даже не имеющих никакого отношения к компьютеру людей. А его изначальное значение со временем забылось. Сегодня "хакер" синоним словам "бандит" и "компьютерный вандал". С другой стороны, предпринимаются неоднократные попытки "очистки" термина вводом новых понятий, например, "кракер" — коммерческий взломщик, в противовес якобы бескорыстным в своих побуждениях хакерам. На самом же деле, первые хакеры появились задолго до возникновения компьютеров, более того, задолго до зарождения цивилизации и даже появления человечества. Первым открытием хакеров было то удивительное свойство палки, которое давало возможность охоты, обороны и еще много чего одновременно. Нетривиальное решение задач, за гранью обычного восприятия мира — это главная черта хакеров. Им мало видеть предмет в трех измерениях. Для хакеров каждый предмет — это, прежде всего, объект со своими свойствами, методами и причинно-следственными связями.

На протяжении всей истории человечества всегда находились люди, которые выходили за рамки господствующих установок, традиций и создавали свою философию и субкультуру. По иронии судьбы хакерство оказалось тесно сплетенным с криминалом. Так было во все века, и так и будет до самого последнего вздоха человечества. Почему? Хакерская натура стремится разобраться во всем до конца, разрешить все до мельчайших подробностей, выйти за область определения объекта и проанализировать и испытать его поведение во всех нештатных ситуациях. От простого смертного хакера, прежде всего, отличает исчерпывающее знание предмета. Абсолютное знание по умолчанию подразумевает абсолютную власть над системой. Очень трудно

устоять перед искушением и открывающимися перспективами. Между взломом компьютерной системы и механического сейфа нет принципиальной разницы, чтобы об этом ни говорили. Так что хакеры были всегда, и, по меньшей мере, спекулятивно — связывать их существование с ЭВМ и компьютерными технологиями.

Хакерское сообщество не монолитно в своих побуждениях, целях и мотивах. От невинной шалости до умышленного уничтожения информации очень большой путь. Невозможно осуждать человека за сам факт принадлежности к хакерам. Быть может, в его поступках и нет ничего дурного? К хакерам относят и компьютерных специалистов, занимающихся вопросами безопасности, поскольку им по роду своей деятельности приходится атаковать системы с целью обнаружить (и по возможности устранить) бреши в механизме защиты.

Существует громадное количество людей, интересующихся и влияющих на проблемы безопасности сетевых (да и не только) сообщений. Некоторые из них называют себя хакерами, но имеют скудный багаж знаний, нередко почерпнутый из популярных кинофильмов; другие же, досконально изучив все тонкости защит, могут и не считать себя хакерами, даже если совершают (или не совершают) регулярные атаки. И кто же в этой ситуации хакер, а кто нет?

Попытки ввода ценза на принадлежность к хакерам обречены на провал. Зачем усложнять суть? Хакер — тот, кто атакует систему. Как и зачем он это делает — предмет другого разговора. Человеческий язык направлен на упрощение и обобщение терминов. Любые искусственные построения сметаются временем. Плохо ли это, хорошо ли это — неважно. Никто не в силах изменить ход вещей окружающего мира, поэтому приходится жить по его законам.

Бытует мнение о существовании некоторых признаков принадлежности к хакерам. Это длинные (нечесаные) волосы, пиво, сигареты, пицца в неограниченных количествах и блуждающий в пространстве взгляд. Беглое исследование как будто бы подтверждает справедливость таких утверждений, однако подобные признаки являются не причиной, а следствием. Привязанность к компьютеру заставляет экономнее относиться к свободному времени, порой питаясь всухомятку урывками и на ходу; крепкие напитки вызывают опьянение, затрудняющее мыслительную деятельность, поэтому компьютерные фанатики полностью или частично отказываются от них, переходя на пиво (а то и лимонад). Длинные волосы? Да, они свойственны всем компьютерщикам (и не только им), а вовсе не исключительно хакерам, как, кстати, и все другие "хакерские" признаки.

Еще до недавнего времени считалось: хакер по определению высококлассный специалист. Сегодня же можно атаковать систему, даже не имея никаких представлений о том, как она устроена. Достаточно воспользоваться

доступной информацией или готовой программной реализацией атаки. Поэтому к хакерам приходится причислять всех "кто хочет", потому что "не мочь" уже невозможно. Многие атакующие программы неплохо документированы и имеют простой, интуитивно понятный интерфейс, в большинстве случаев не требующий от "взломщика" ничего, кроме владения мышью.

Так ли велико различие между желанием и умением вторгаться в чужие системы? Кажется, желание в отсутствие умения бесполезно, но это не так. Знания возникают не сами по себе, а приобретаются в процессе обучения. И те, кто хочет, но не может, и те, кто может, но не хочет, — потенциальные злоумышленники, способные на противоправные действия.

Очередной исторический каприз: именно длительная компьютерная анархия позволила легально развиваться целому пласту субкультуры личностей, которые в других отраслях прочно ассоциировались с криминалом. До этого были радиолюбители, которые обходили электронные системы сигнализации, перехватывали секретные передачи и конструировали удивительные по своей природе устройства.

Между перечисленными категориями нет качественной разницы. Более того, обычно работа с компьютером связана с интересом к электронике, и обычное хакерское любопытство затрагивает не только компьютерные, но и любые другие системы защиты.

Я понимаю, что данная трактовка может встретить возражение и является ничем иным, как моим субъективным мнением. И тут мы приходим к любопытной лингвистической проблеме терминов. В самом же деле, любой термин можно описать, перечислив все свойства и лексические вхождения. Независимо от возможных трактовок термина объективная нагрузка на него может быть задана простым перечислением.

Развитие любого языка приводит к тому, что каждый термин достоверен только в момент его определения, после чего начинается размытие и расширение смысловой нагрузки, подминающей под себя изначальную идею создания. Обратимся к лучшему на сегодняшний день исследованию хакерской культуры, "Словарю Жаргона" Э. С. Рэймонда. Словарь гласит (перевод за читателем):

:hacker: [originally, someone who makes furniture with an axe] n.

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.
2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.
3. A person capable of appreciating {hack value}.
4. A person who is good at programming quickly.

5. An expert at a particular program, or one who frequently does work using it or on it; as in `a UNIX hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.)
6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.
7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.
8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence `password hacker', `network hacker'. The correct term is {cracker}.

В рамках данной книги мы будем говорить исключительно о компьютерных хакерах. По непроверенным историческим данным, компьютеры, изначально созданные для узкоспециализированных военных задач, именно хакерами были восприняты, как платформы с безграничными возможностями. В то далекое время математики занимались исключительно гипотетическими машинами, которые имели очень отдаленное отношение к действительности. Воплощенные в груду металла инженерные идеи обогнали даже знаменитых математиков на десятилетия вперед. Предложенная дискретная архитектура была совершенна и привела к развитию соответствующей дискретной математики, большей частью описывая то, что инженеры давно воплотили в жизнь. Так изначально практические компьютерные технологии обогнали математические модели.

Сегодня нам трудно представить, что было время, когда компьютеры обслуживали сливки технической элиты, решая рутинные, поставленные задачи. Впрочем, для начальства не секрет, что свободное и не такое уж свободное машинное время использовалось для личных нужд и исследований персонала. Так и зарождалась субкультура людей, которые открыли в грохочущем монстре вторую Вселенную, свое второе Я. Чудовищное отставание нашей страны в области вычислительной техники, жесткая дисциплина, постоянные репрессии привели к тому, что субкультура хакеров, да и программистов, зародилась целиком в стенах лабораторий США и уже оттуда распространилась на весь мир. В результате даже наши отечественные субкультуры большей частью американизированы, особенно у нас, на фоне массового использования американской программно-аппаратной базы.

История происхождения термина "хакер"

Термин "кракер" (cracker, ломатель), был введен в 1985 г. самими хакерами в знак протеста журналистам, неправильно употребляющим, по их мнению, термин "хакер".

История же возникновения термина "хакер" доподлинно неизвестна. Ее истоки ведут к древнеанглийскому, в котором "хак" обозначал звук топора,

возникающий при ударе о дерево. Но звание хакера присуждалось далеко не всем, а лишь высококлассным столярам и плотникам, создающим едва ли не произведения искусства (другими словами, хакеры представляли собой одержимых плотников).

В шестидесятых годах прошлого столетия, а может быть и раньше, этот термин перекочевал в компьютерную среду. По одной из гипотез, звук "хак" приписывался перфоратору, прокалывающему бумагу; другие уверяют, что так клацали реле. Так или иначе, хакерами стали называть системных программистов, фанатично преданных своему делу, по аналогии с плотниками.

Традиция сохранялась на протяжении более десятка лет, затем новое поколение студентов, впервые увидевших компьютер, окрестило хакерами всех фанатиков без исключения, даже если те были заурядными пользователями.

Примечание

В Массачуссетском технологическом институте и вовсе имеется свое, оригинальное понимание хакеров: "At MIT a hacker is someone who does some sort of interesting and creative work at a high intensity level. This applies to anything from writing computer programs to pulling a clever prank that amuses and delights everyone on campus".

Литературные произведения "The Shockware Rider" Джона Бруннера (John Brunner) 1975 г., "The Adolescence of P-1" Томаса Риана (Thomas Ryan) 1977 г. и наконец, знаменитый "Necromancer" Вильяма Гибсона (Wiliam Gibson), опубликованный в 1984 г., своими героями выбрали компьютерных взломщиков, бросающих вызов обществу. Это совпало с движением панков на западе и было молниеносно подхвачено молодежью. Сейчас ведутся жаркие споры: то ли представителей нового движения хакерами назвали журналисты, то ли те самостоятельно присвоили себе это звание, но с этого момента под хакерами стали подразумевать злоумышленников, мошенников и вандалов всех мастей, отчего легальные хакеры старого поколения по возможности пытались избегать величать себя этим титулом.

В настоящее время термин "хакер" стал поистине всеобъемлющ, отчего потерял всякую ценность и смысл. К хакерам относят откровенных бандитов, совершающих преступления с применением технических средств, просто мелких мошенников, вирусописателей, системных программистов, а порой и просто программистов, в особенности знающих ассемблер, экспертов по безопасности... Считают себя хакерами и те, у кого хватило таланта и способностей запустить готовую атакующую программу даже без осмысления принципа ее работы.

Психология хакера

Где бы ни организовывались вычислительные центры — в бесчисленных местах в Соединенных Штатах, как фактически во всех промышленных районах мира, можно наблюдать блестящих молодых людей, всклокоченных, часто с запавшими, но сияющими глазами, которые сидят за пультами управления вычислительных машин, сжав в напряжении руки в ожидании возможности пустить в ход свои пальцы, уже занесенные над кнопками и клавишами, приковывающими их внимание так же, как брошенная игральная кость приковывает взгляд игрока. Если они не находятся в таком трансе, то часто сидят за столами, заваленными машинными распечатками, которые они сосредоточенно изучают подобно людям, одержимым постижением кабалистического текста. Они работают, чуть ли не до полного изнеможения, по 20–30 часов подряд. Еду, если только они о ней заботятся, им приносят (кофе, кока-кола, бутерброды). Если возможно, они спят около вычислительной машины на раскладушках, но всего несколько часов, а затем — снова за пульт управления или к распечаткам. Их измятая одежда, немытые и небритые физиономии, нечесаные волосы — все свидетельствует о том, что они не обращают внимания ни на свое тело, ни на мир, в котором живут. Они существуют, по крайней мере, когда они так увлечены, лишь в связи с вычислительными машинами и ради них. Они — "машинные наркоманы", одержимые программисты. Это явление наблюдается во всем мире.

Дж. Вейценбаум. "Возможности вычислительных машин и человеческий разум (от суждений к вычислениям)"

Типичный образ хакера конца девяностых прошлого века: молодой человек, так и не получивший систематического образования, открывает в компьютере свою собственную вселенную и уходит в нее целиком. "Как и в других областях человеческой деятельности, спектр отношения людей к програм-

мированию и вычислительным машинам очень широк: от ненависти, через полное безразличие до патологической привязанности или зависимости, которую можно квалифицировать как манию" — писал Николай Безруков в своей монографии "Компьютерная вирусология".

Но в компьютере, в отличие от других видов деятельности, все-таки есть нечто особенное. Легко ли вообразить себе человека, помешанного утром, днем и ночью пылесосить? Зато любителя проводить все свободное и несвободное время за компьютером представить несложно. Какое же это все-таки увлекательное занятие — писать программы! Какое наслаждение смотреть, как они работают, и как приятно видеть результаты прогонов! Это все и работой назвать язык не поворачивается — сплошные удовольствия.

Для объяснения этого феномена выдвинуто и выдвигается множество различных гипотез и предположений. В ход идут аргументы типа: компьютер это собеседник, общаться с которым невероятно интересно; виртуальные игровые миры дают те чувства свободы, силы, самоутверждения, власти, которых нам так не хватает в реальной жизни.

Среди отечественных психологов бытует мнение, что к компьютеру сильнее всего привязываются личности, по тем или иным причинам отвергаемые обществом. Это может быть физическое несовершенство, уродство или инвалидность. Лишенные нормального человеческого общения, эти люди тянутся к компьютеру, как уникальному средству самовыражения и самоутверждения. Впрочем, обратное утверждение не всегда справедливо — компьютерный фанатик не обязательно должен оказываться инвалидом.

Возможно, ближе всех к истине подобрались зарубежные психологи. Среди их пациентов нередко встречались лица, абсолютно не приспособленные к реальной жизни, испытывающие огромные трудности в общении с окружающими людьми, отличающиеся неадекватной реакцией на все происходящее, одним словом, внешне создающие впечатление умственных дегенератов, но вместе с тем превосходно (даже виртуозно) программирующих на компьютере. В большинстве случаев феномен объяснялся тем, что практически все такие пациенты страдали аутизмом.

Аутизмом (от латинского слова autos — "сам", аутизм — погружение в себя) называют тяжелое психическое расстройство, при котором больной самоизолируется и существует как бы вне контакта с окружающим миром, теряя способность формировать эмоциональные привязанности и строить общение с людьми. О причинах аутизма на сегодняшнем уровне развития медицины остается только гадать, но предполагают, что это связано с недоразвитием определенных долей мозга в сочетании с гиперразвитием остальных областей. Другой возможной причиной называют аномальный химический состав мозга, но, так или иначе, аутизм относят к врожденным заболеваниям, хотя временами по этому поводу высказываются серьезные возражения.

Процент заболеваемости колеблется от 4 до 15 случаев на 10 000 детей, значительная часть их которых — мальчики. По статистике только в одних Соединенных Штатах зарегистрировано более 400 000 аутистов, но у 80% из них показатели IQ выше среднего и нередко на уровне гениев.

"С одной стороны, я могу находить ошибки в программах так быстро, что людям становится неловко. Но я совершенно асоциален. Я не могу угадывать намерения и настроение человека по выражению его лица или по его жестам, различать социальные оттенки и, наверное, не умею пользоваться этим языком. У меня не сложились взаимоотношения ни с кем из моих коллег, я определенно существо не коллективное" — рассказывает о себе Петер Леви, один из основателей компании "Accent Technologies", страдающий этим заболеванием.

Аутисты испытывают затруднения в общении даже с близкими людьми, у них отсутствует интерес к окружающему миру, явно выражены страхи, особенности поведения. С самого детства, ощущавшие себя несколько отстраненными от мира, от людей, затрудняющиеся в налаживании контактов со сверстниками, порой битые за свою непохожесть, они находят в компьютере отдушину, средство сравняться с другими людьми или превзойти их.

"В этом мире можно дать волю всем своим идеям и фантазиям — от детских соплей до изощренного садизма. Мир, который полностью зависит от своего могущественного властелина, безраздельно распоряжающегося жизнью и смертью любой твари, которой позволено жить в его мире. Уйти от забот и переживаний грубого материального мира, стать творцом и властителем это мечта большинства, но мечта потаенная, скрытая. Осуществление ее доступно немногим, лишь тем, кто хочет и может" — писал психолог Ю. П. Прокопенко в одной из своих статей, завершая ее таким напутствием: "Хоть с мясом отрывайте свой зад от стула перед компьютером, идите на воздух, общайтесь с людьми. Как бы ни была интересна задача, она не уйдет, а вот жизнь проходит... Если вы чувствуете себя гораздо увереннее в виртуальном мире, чем среди людей, попробуйте посоветоваться с психологом — вдруг чего дельного скажет. Его советы не обязательны для исполнения, но профессиональный опыт может подать вам вашу же проблему с такой неожиданной стороны, что изменится вся система жизненных оценок. Рискните, пусть будет у вас побольше того самого жизненного опыта, которого так не хватает аутисту при любой выраженности этой черты характера".

Попытка связать психическую патологию и компьютерную одержимость не является чем-то новым. Эту мысль отстаивал еще Дж. Вейценбаум в своей монографии "Возможности вычислительных машин и человеческий разум (от суждений к вычислениям)". Вот что он пишет по этому поводу: "Разложение, порождаемое всемогуществом программиста вычислительной машины, проявляется в форме, поучительной для сферы, значительно более обширной, чем мир вычислительной техники. Чтобы оценить его, придется обратиться к примеру психического расстройства, хотя и очень давно известного, но, по-видимому, преобразовавшегося благодаря вычислительным

машинам в новую разновидность — манию программирования". Предположение, что хакерство больше чем образ жизни, склад мышления и простая привязанность к компьютеру, многое проясняет и позволяет пересмотреть свои взгляды на, казалось бы, очевидные факты. Хакерами движет вовсе не желание навредить. Даже когда наглым образом вредят, они лишь стараются обратить на себя внимание, компенсируя недостаток общения. Этот бессознательный порыв может ими самими истолковываться стремлением отомстить обидевшему их человечеству, но это не всегда оказывается так. Человеческий мозг — невероятно сложный механизм, состоящий из множества обособленных скоплений нейронов, с трудом понимающих "язык" друг друга. Сознание — лишь надводная часть айсберга; большая же часть работы протекает на бессознательном уровне. Поэтому мотивы многих поступков так и остаются загадкой. Человек лишь пытается объяснить их так или эдак, зачастую ошибаясь в своих выводах.

Хотя аутистам и присущи внезапные вспышки агрессии, среди одержимых программистов вандализм встречается редко. "Нас обвиняют в том, что мы, мол, чувствуем себя самыми великими, ни во что не ставя людей, которые не работают с компьютерами. Чушь собачья. Как ни стараюсь вот сейчас представить, не получается у меня почувствовать превосходство над, допустим, слесарем потому, что я более или менее умею заставлять компьютер делать то, чего хочу я. Зато он гайки крутит так, как мне в жизни не суметь" — заметил некто по прозвищу Jen.

"Средства массовой информации обычно обращают внимание исключительно на негативные стороны хакерства (взлом и кражу информации, разработку и распространение вирусов), однако такой взгляд является односторонним: для хакеров характерно гипертрофированное увлечение познавательной деятельностью, направленной на выяснение закономерностей работы информационных технологий, что необязательно ведет к каким-либо асоциальным действиям. Наоборот, существует мнение, что многие удачные и полезные идеи в области программного обеспечения были в свое время выдвинуты и реализованы именно хакерами"— писала О. В. Смыслова в своей работе "Методы полевого психологического исследования в сообществе хакеров".

Но все же не всегда хакерство оказывается следствием тяжелой патологии. Часто дело заключается в обычном юношеском максимализме, когда кажется: весь мир твой, и ты его хозяин на правах сильного. Отсюда же идет глубокое убеждение, что информация должна быть свободной, а программное обеспечение — бесплатным. В отличие от неизлечимого аутизма, юношеский максимализм с возрастом проходит: появляется работа, отнимающая все свободное (а у фанатиков и несвободное) время, вырабатывается профессионализм, и надобность доказывать окружающим, что ты не осел, исчезает. А вместе с ней исчезает и сам хакер.

Но не пытайтесь отождествить себя ни с какими портретами. Каждый человек уникален и не подлежит усредняющей классификации.