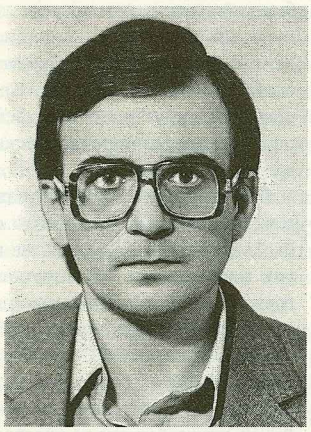
АНТИВИРУСЫ



Н.Н. БЕЗРУКОВ

Классификация вирусов Попытка стандартизации

Актуальность проблемы защиты компьютера от вирусов обусловила необходимость организации оперативного обмена информацией и взаимодействия специалистов, работающих в данной области, что невозможно без разработки стандартной классификации компьютерных вирусов.

"Интеркомпьютер" начинает публикацию статьи, основанной на версии 3.0 первой части монографии Н.Н. Безрукова "Компьютерная вирусология" (ее предполагает выпустить в 1990 г. издательство "Наука"). Первая редакция этой работы, в которой сделана попытка систематизировать сведения о компьютерных вирусах, появилась в сентябре 1989 г. В дальнейшем текст монографии обновлялся ежемесячно. Очередные версии работы (на дискете) распространяются бесплатно на ежемесячном семинаре "Системное программирование", который проводится в Киеве под руководством автора (каждый второй четверг месяца в ауд. 4-205 Киевского института инженеров гражданской авиации, начало семинара в 15.00) и в Москве на аналогичном семинаре под руководством Л.Г. Бунича (каждый третий четверг месяца в помещении конференц-зала ЦНСБ ВАСХНИЛ, начало семинара в 11.00).

В этом номере рассматриваются принципы классификации; классификационные таблицы и пояснения к ним будут опубликованы в следующем номере "Интеркомпьютера".

Компьютерный вирус - это программа, обладающая способностью к скрытому саморазмножению в среде стандартной операционной системы компьютера с помощью включения в исполняемый код (загружаемые программы, компоненты операционной системы, пакетные файлы или компилируемый текст) своей, возможно, модифицированной копии, которая сохраняет способность к дальнейшему размножению. Такое определение является вариантом определения, данного Ф. Коуэном (F. Cohen) - автором первого серьезного математического исследования компьютерных вирусов, в статье *Computational Aspects of Computer Viruses*, опубликованной в августовском номере журнала "Computers & Security" за 1989 г. Пионером среди отечественных исследователей в этой области был А.А. Чижов, который в 1987 - 1988 гг. провел ряд экспериментов с компьютерными вирусами.

Необходимо подчеркнуть, что применение стандартной классификации существенно облегчает накопление и распространение знаний в любой области, позволяя, в частности, однозначно определять как известные вирусы и их разновидности, так и новые, только что появившиеся вирусы и их еще неисследованные разновидности (штаммы). При этом следует использовать ограниченный набор сравнительно простых и непротиворечивых признаков, для определения которых не требуется серьезный анализ зараженных программ и элементов операционной системы. Кроме того, переход к стандартной классификации облегчит пользователю выбор антивирусных средств (очень часто эти средства, разработанные программистами какой-то организации для решения конкретной проблемы, затем распространяются по всей стране, хотя и не имеют документации). В ряде случаев новые антивирусные средства разрабатываются только потому, что отсутствует информация о возможностях существующих антивирусных программ.

Если разработчики не сообщают типы вирусов, на которые ориентированы те или иные антивирусные программы, или сообщают лишь придуманные ими неформальные названия ("клички") этих программ, то пользователям приходится выяснять их применимость экспериментальным путем. Например, так называемый "Венский" вирус (С-648 по предлагаемой классификации) разработчиками программы-фага FAG_OM назван Omega, что, конечно, свидетельствует об их изобретательности, но ничего не говорит потенциальным пользователям этой антивирусной программы.

К сожалению, в настоящее время при классификации используются в основном именно такие "клички". Анализ этих названий позволил выявить три типа "кличек", учитывающих:

место обнаружения или разработки вируса

(например, вирусы Lehigh, Jerusalem, Vienna, Alameda);

содержащиеся в теле вируса текстовые строки или выдаваемые вирусом сообщения (например, вирусы Vacsina, Eddie, Dark Avenger, Disk Killer, sUMsDos);

эффект, вызываемый вирусом, (например, вирусы Time Bomb, DOS-62, Cascade, Black Friday).

При этом один и тот же вирус может иметь множество названий, и новое название вируса, использованное разработчиком той или иной антивирусной программы, далеко не всегда соответствует новому вирусу. Перечень названий многих широко известных вирусов напоминает список имен арабского шейха. Например, мне приходилось встречать не менее девяти названий вируса, обнаруженного в декабре 1987 г. в Иерусалимском университете (RCE-1813 по предлагаемой классификации), три из которых - Israeli virus ("Израильский вирус"), Jerusalem ("Иерусалим") и PLO ("ООП") - относятся к первому типу, два названия (sUMsDos и sU) - ко второму типу, и, наконец, еще четыре: Black Hole ("Черная дыра"), Black Friday ("Черная пятница"), Friday 13 ("Пятница, 13-е") и "Вирус замедления" - к третьему типу. Данный вирус "вырезает" в левом углу экрана "черную дыру", удаляет с диска все файлы, которые запускаются по пятницам, пришедшимся на 13-е число, и, кроме того, примерно через 20 мин после запуска зараженной программы искусственно замедляет работу компьютера в несколько сотен раз.

Исследование вируса на зараженном компьютере - наиболее распространенная ошибка, которую допускают неопытные пользователи.

Конечно, такое многообразие названий создает определенные трудности, особенно если учесть, что рассматриваемый вирус имеет несколько штаммов, отличающихся особенностями функционирования.

Принципы построения классификации

Основным требованием к классификации вирусов является ее объективность - классификация должна основываться на фиксированном наборе непротиворечиво измеряемых или наблюдаемых признаков. В идеальном случае эти признаки следует выбирать так, чтобы, скажем, два разработчика антивирусных средств, независимо работающих в Киеве и Москве, использовали для одного и того же вируса одинаковое название и разные названия для разных вирусов. Это позволит быстро выявлять новые вирусы или их штаммы.

Очевидно, что объективная классификация вирусов существенно облегчит систематизацию, распространение и накопление знаний в области компьютерной вирусологии, а также упростит выбор программных средств для борьбы с тем или иным вирусом.

Следует подчеркнуть важность не просто данной классификации как таковой, но и принятие ее в качестве *стандартной*. Уже сейчас отсутствие стандартной классификации вирусов приводит к ряду нежелательных явлений:

каждый разработчик антивирусных программ использует свою уникальную классификацию вирусов, из которой часто неясно, о каком конкретно вирусе идет речь, тем более что помимо основного типа вируса обычно существует ряд его штаммов со сходными, но не идентичными свойствами;

у пользователей наблюдается тенденция аппроксимировать общее число вирусов количеством известных им антивирусных программ, и прежде всего, программ-фагов, которые, образно говоря, "выкусывают" тело вируса из зараженной программы, восстанавливая тем самым ее работоспособность в близком к первоначальному состоянии.

Такая аппроксимация приводит к существенной переоценке общего числа имеющихся компьютерных вирусов, однако человек быстро "рационализирует" этот факт, разбивая один реальный вирус на несколько "виртуальных" и приписывая каждому из них собственный набор признаков. Так, автору приходилось сталкиваться с "самодельной" классификацией, в которую вирусы C-648 и RCE-1813 входили в двух вариантах каждый, причем второму варианту вируса C-648 приписывались свойства вируса RCE-1813 (замедление работы компьютера).

В процессе чтения лекций и проведения семинаров по данной проблеме, а также в процессе работы над рукописью книги "Компьютерная вирусология" была выработана схема классификации вирусов, включающая в себя три таких основных элемента, как:

классификационный код вируса (напоминает код в классификации транзисторов);

дескриптор вируса (формализованный список его основных свойств);

сигнатура вируса (строка для контекстного поиска данного вируса в зараженной программе).

Классификационный код вируса

В предлагаемой классификации каждому вирусу присваивается классификационный код, состоящий из буквенного префикса, цифровой части (корня), которая содержит характеристику вируса, и необязательного буквенного суффикса. Например, в классификационном коде RC-1704f буквы RC - это префикс, 1704 - корень (характеристика), а f - суффикс.

Главным требованием к классификационному коду является возможность определения большинства свойств вируса на незараженном компьютере. Выполнение каких-либо действий по исследованию вируса на зараженном компьютере является наиболее распространенной и одновременно наиболее грубой ошибкой, которую допускают неопытные пользователи. Следует подчеркнуть, что любые действия на компьютере, зараженном неизвестным вирусом, сопряжены с определенным риском вызвать срабатывание "троянской" компоненты вируса. Кроме того, резидентный вирус с целью маскировки может перехватывать запросы и искажать выдаваемую информацию.

В настоящий момент мне известен ряд вирусов, обладающих указанным свойством. Так, файловые вирусы, входящие в состав группы так называемых TP-вирусов (вирусы этой группы имеют номера, хранящиеся в предпоследнем байте кода вируса в 16-ричном виде), начиная с вируса TP-34, обладают интересным свойством: при попытке трассировать зараженную программу резидентный вирус "выкусывает" вирус из программы, "подсовывая" отладчику уже излеченную программу. Аналогично бутовые вирусы, известные под названием "Пакистанских" (Brain, Ashar и др.), при попытке просмотреть бутсектор на зараженном компьютере "подсовывают" пользователю оригинальный бутсектор, сохраненный вирусом в одном из секторов, отмеченных, как дефектные (и тем самым исключенных из распределения под файлы).

Буквенный префикс. Этот элемент классификации характеризует среду размножения вируса. Все вирусы, с этой точки зрения, можно разделить на пять основных типов:

файловые вирусы, заражающие файлы типа COM и/или EXE (префикс C, E или CE);

бутовые вирусы, или вирусы загрузчиков, заражающие бутсектор (boot sector) жесткого и гибкого дисков, т.е. загружаемый сектор, либо блок MBR - Master Boot Record (префикс B, D или M);

драйверные вирусы, заражающие драйверы устройств или запускающие себя включением в файл CONFIG.SYS дополнительной строки (префикс S);

пакетные вирусы, использующие для своего запуска язык управления заданиями операционной системы или написанные на этом языке (префикс J);

компилируемые вирусы, включающие свой код в исходный текст некоторых компилируемых программ (префикс T).

Следует упомянуть так называемые *сетевые вирусы* (точнее, *репликаторы*), обеспечивающие "рассылку" своего кода всем абонентам сети или их части. Среди них встречаются как пакетные вирусы (например, вирус Christmas Tree написан на достаточно хорошо известном в нашей стране языке управления заданиями REXX операционной системы VM - наиболее распространенной для компьютеров семейства IBM/370), так и компилируемые вирусы (например, вирус P. Морриса, мл., имеющий компилируемую компоненту). В MS-DOS встречаются только два из упомя-

нутых выше типов вирусов: файловые и бутовые (вирусы общего назначения и вирусы загрузчиков по терминологии И.Ш. Карасика).

Буква R в префиксе показывает, что вирус является резидентным (после запуска зараженной программы такой вирус закрепляется в оперативной памяти и, перехватывая некоторые прерывания, активизируется при определенных действиях операционной системы).

Количественная характеристика. Эта часть классификационного кода вируса представляет собой достаточно просто определяемую количественную характеристику какого-либо свойства вируса; эти характеристики для большинства типов вирусов отличаются друг от друга. Например, для файловых вирусов в качестве характеристики можно использовать нормированное приращение длины файла при заражении.

Буквенный суффикс. В классификационном коде суффикс (необязательный элемент) отражает дополнительную информацию о вирусе, например наличие штаммов, неотличимых по префиксу и характеристике).

Конечно, рассмотренный выше классификационный код вируса не позволяет описать все основные свойства вируса. Однако предлагаемая систематизация свойств вирусов, надеюсь, представляет значительный интерес как для разработчиков антивирусных программ, так и для пользователей, поскольку позволяет объединять разнородные факты, относящиеся к поведению того или иного вируса, облегчая их запоминание и сопоставление.

Дескриптор вируса

В качестве второго элемента классификации выбран дескриптор вируса, отражающий систематизацию основных характеристик вируса в закодированном виде. Дескриптор вируса состоит из групп символов, начинающихся с прописной латинской буквы, за которой следуют строчные латинские буквы или цифры. При этом прописная латинская буква определяет вид дескриптора, а следующие за ней строчные буквы или цифры соответствуют значению дескриптора для конкретного вируса. Например, в дескрипторе XabYcZdmt отражены три свойства: X со значением "ab", Y со значением "c", и Z со значением "dmt".

Сигнатура

Поскольку все известные в настоящее время вирусы можно обнаружить с помощью контекстного поиска, применяя антивирусные программы-детекторы, одной из важных задач классификации является составление перечня сигнатур, т.е. списков строк для контекстного поиска. С помощью сигнатур осуществляется "детектирование" впервые используемого программного обеспечения на наличие вирусов, что позволяет повысить степень защищенности компьютера.

Хотя в дальнейшем в качестве сигнатур применяются только текстовые строки, для них можно использовать и регулярные выражения. Последние существенно устойчивее к некоторым мутациям вирусов и, кроме того, при меньшей длине обеспечивают лучшее качество распознавания (меньшее количество ложных срабатываний). Все это делает их предпочтительнее простых текстовых строк. Версию классификационных таблиц с сигнатурами из регулярных выражений автор предполагает опубликовать несколько позднее.

Стандартизация сигнатур особенно важна для вирусов, имеющих много штаммов, поскольку формальные обозначения вируса, подобные описанным выше классификационному коду и дескриптору, обладают тем недостатком, что в заданном пространстве признаков некоторые штаммы нельзя отличить друг от друга. В то же время сравнительно легко обеспечить уникальность сигнатуры, по крайней мере, для известных типов вирусов (теоретически можно создать вирус, не имеющий ни одной сигнатуры, т.е вирус, который принципиально нельзя обнаружить с помощью контекстного поиска).

Очевидно, что вероятность обнаружения вируса с помощью сигнатуры, соответствующей участку, содержащему команды, выше, чем у сигнатуры участка, содержащего данные, например текстовые строки (так как последние могут быть модифицированы). Поэтому выбор сигнатуры целесообразно выполнять на основе анализа реконструированного исходного текста вируса. Длина сигнатуры не должна быть слишком большой (длинную сигнатуру невозможно запомнить и труднее ввести вручную); в то же время при недостаточной длине сигнатуры или при выборе для нее нехарактерных участков кода вируса будет происходить много ложных срабатываний, что весьма нежелательно.

Строго говоря, список строк для контекстного поиска, выбранный в качестве сигнатуры, не должен содержаться ни в одной из наиболее распространенных в MS-DOS системных программ, включая, конечно, и компоненты MS-DOS. Таким образом, для выбора отвечающей указанным требованиям сигнатуры необходим ряд экспериментов, в ходе которых сами сигнатуры становятся предметом сравнения и анализа.

В настоящее время имеется ряд удачных программ-детекторов, позволяющих обнаружить вирус с помощью поиска соответствующих строк (сигнатур) в файлах; эти сигнатуры естественно "принять за основу" (например, сигнатуры, используемые в двух известных американских программах-детекторах: SCAN фирмы McAfee Associates и VIRSCAN фирмы IBM). Для определенности обозначим буквой M сигнатуру, используемую программой-детектором SCAN, а буквой I - сигнатуру, используемую программой-детектором VIRSCAN.

Необходимо отметить, что сигнатуры для ряда вирусов, распространяющихся в нашей стране (С-534, С-623, С-529 и др.) в существующих версиях этих программ отсутствуют, а сигнатуры для группы TP-вирусов неудачны. В таких случаях в статье используются выбранные автором сигнатуры, которые обозначены буквой B.

Кроме того, в теле некоторых вирусов встречаются характерные текстовые строки. Такие сигнатуры, обозначенные буквой T, будем использовать как вспомогательные.

При наличии сигнатуры проверку зараженности файлов вирусом данного типа можно выполнять, применяя не только специализированные антивирусные программы (из них наиболее удачной, по-моему, является программа Virus Locator (VL), разработанная А.В. Шеховцовым, которая позволяет проводить поиск сигнатур в каталоге или заданных его ветвях), но и пакеты Norton Utilities (NU) или PCTools, благо у пользователя они всегда под рукой (в последнем случае для просмотра всех файлов можно использовать режим глобального поиска по диску). Следует отметить, что антивирусная программа Virus Locator получила первую премию по классу программ-детекторов и фагов на конкурсе бесплатных антивирусных программ, который проводился в феврале 1990 г. Киевским семинаром "Системное программирование".

Окончание статьи и классификационные таблицы будут опубликованы в следующем номере "Интеркомпьютера".

Читатели, желающие регулярно получать текущие версии монографии "Компьютерная вирусология" на дискетах, могут подписаться на электронный бюллетень СОФТПАНОРАМА Киевского семинара "Системное программирование". По вопросам подписки обращайтесь по адресу: 252680, Киев-58, ГСП, Проспект космонавта Комарова 1, Киевский институт инженеров гражданской авиации (КИИГА), корп. 3, Авиацентр НТТМ (тел. 484-94-63).

ФАКТЫ

Фирма Foresight Resources разработала версию САПР Draflx, работающую в среде Windows, - Draflx Windows CAD. Хотя этот пакет может работать только с двумерной графикой, но, выполняя большинство функций таких мощных пакетов, как AutoCAD и Cadkey, он стоит гораздо дешевле - 695 дол.

АНТИВИРУСЫ

Классификация антивирусных программ

И.Ш. КАРАСИК

В предыдущем номере "Интеркомпьютера" рассматривались вопросы, связанные с "анатомией" и "физиологией" вирусов различных типов. В этом номере я хочу познакомить вас с классификацией антивирусных программ, позволяющих защитить компьютер от проникновения вирусов или, по крайней мере, обезопасить его от их вредного воздействия.

Классификация антивирусных программ, приведенная ниже, основана на различных особенностях вирусов, проявляющихся в разные моменты их жизни. Следует подчеркнуть, что пользователю совершенно безразлично, кто испортил ему информацию: вирус или троянская программа, поэтому, говоря об антивирусных программах, я все время буду иметь в виду еще и оказание какого-то противодействия троянским программам (давно обещанный разговор о троянских программах состоится позднее).

При всей идейной простоте антивирусных средств разработка качественных антивирусных программ - дело достаточно тонкое и весьма трудоемкое. Во всяком случае, для создания хорошей антивирусной программы необходимо иметь богатую, постоянно пополняемую коллекцию вирусов, выделять действительно новые вирусы и изучать их, т.е. проводить собственные исследования в области тайного проникновения вирусов в систему.

Я уже не говорю о том, что надо знакомиться с новыми версиями операционной системы (ОС), следить за публикациями о подробностях ее функционирования и т.д.

Детекторы вирусов и фаги

Итак, что же выдает присутствие вируса в компьютере? В первую очередь, наличие кодов вируса в теле некоторых исполняемых файлов либо в известных секторах жесткого или гибкого диска. На этом основано функционирование специализированных антивирусных программ-детекторов, предназначенных для обнаружения конкретных вирусов. Теоретически обнаружение вируса, заразившего файл, - дело относительно несложное. Нужно только знать, что ищешь, т.е. знать специфику конкретного вируса (ключевые поля, устойчивые коды и т.п.). Кроме этого, следует представлять механизм внедрения вируса в программу или, если это вирус загрузчика, знать, в каком месте на диске размещает он свои коды. Располагая такой информацией, можно написать антивирусную программу-детектор для обнаружения конкретного вируса. Детектор вирусов загрузчиков должен просматривать конкретные сектора жесткого или гибкого диска, детектор вирусов общего назначения - файлы типа COM или EXE по всему дереву каталогов или по его части.

Хочу отметить, что согласно последней информации об IBM-совместимых компьютерах вирусы общего назначения могут поражать не только файлы типа COM или EXE, но и оверлейные части исполняемых файлов (оверлеи), а также драйверы, вставляемые в DOS во время загрузки ОС. Во всяком случае, в документации на программу SCAN49 ее разработчик Дж. Макафи (J. McAfee), известный специалист по борьбе с вирусами, упоминает о таких возможностях. Приходится верить ему на слово, так как ни моим коллегам, ни мне такие вирусы еще не встречались. Сказанное выше означает, что даже для поиска исполняемых файлов по дереву каталогов необходимо писать специальные программы, поскольку для оверлейных частей исполняемых файлов нет стандарта на расширение (правда, в программе SCAN49 есть специальный ключ /E, с помощью которого можно задавать расширения оверлеев и драйверов).

Вообще, приведенное выше утверждение о том, что писать программы-детекторы достаточно просто, не следует понимать буквально. Прозрачна только идея, при написании же программ-детекторов для конкретных вирусов необходимо учитывать ряд особенностей, которые я и попытался описать ниже.

Первая особенность связана с выбором той части текста программы, по которой можно выявить наличие вируса. Несмотря на кажущуюся простоту, это довольно тонкий вопрос. Не стоит идентифицировать вирус по каким-то заметным цепочкам литер или кодов, и уж совсем наивно пытаться обнаружить вирус, используя для поиска видимые глазом текстовые поля сообщений - такие поля не влияют на работу вируса, и их изменить проще всего. При существующем сервисе на IBM-совместимых компьютерах внести изменения в исполняемый файл очень просто - это может сделать и разработчик вируса, и даже любой маломальски грамотный пользователь. Программа-детектор, основанная на идентификации

Название	Разработчик	Характеристика
AIDSTEST	Д. Лозинский	Фаг. Известная мне версия распознает 21 вирус. Первая премия по классу фагов на Киевском семинаре (руководитель Н. Безруков).
ANTI-KOT	О. Котик	Фаг. Последняя версия распознает и удаляет 13 вирусов.
DIAGLOT	Г. Агасандян	Детектор. Версия DIAGLOT-12 позволяет распознавать 12 вирусов.
DOCTOR	А. Чижов	Фаг. Версия 1.52 распознает 22 вируса. Вторая премия по классу фагов на Киевском семинаре (руководитель Н. Безруков).
SCAN49	Дж. Макафи	Детектор. Версия 1.7V49 распознает 51 вирус.
VR	А. Осипенко	Фаг. Версия 3.50 распознает 20 вирусов и некоторые необычные файлы.

вируса по таким признакам, уже не узнает его, хотя вирус практически не изменился. Так и рождаются слухи о новых штаммах вирусов, вызванные, как правило, написанными "в лоб" программами-детекторами.

Ориентироваться, конечно, следует на какие-то характерные для данного вируса шаблоны, т.е. последовательности (цепочки) машинных команд. Достаточно разумным представляется искать эти последовательности в окрестности точки входа в вирус или в той части вируса (для вирусов с резидентной частью), где проверяется активность его резидентной части. В любом случае выбор цепочки машинных команд в настоящий момент - вопрос, скорее, искусства, чем науки (особенно если вспомнить, что интеллектуальные вирусы обладают свойством мутации, а также способностью к интерференции, т.е. могут в ограниченной степени менять свой код или коды каких-то других вирусов). Не забудьте, что, в конце концов, вирус - изделие рук человеческих, и при необходимости разработчик может изменить любую его часть. Правда, история знает случаи чрезвычайной устойчивости формулировок: так, после Зингера никому еще не удалось получить патент на швейную машину, иголка которой имела бы ушко, расположенное в нижней части. Возможно, и для вирусов удастся найти такие инвариантные последовательности кодов. Во всяком случае, Дж. Макафи в документации к своей программе SCAN49 говорит именно о последовательностях команд (исследования И. Митюрина позволили выявить эти последовательности). Правильный выбор цепочек команд для поиска позволяет значительно уменьшить число шаблонов, по которым ведется поиск: несколько штаммов одного вируса могут быть идентифицированы по одной, характерной для этих штаммов цепочке.

Следующая особенность, которую мне хочется отметить, - это алгоритм поиска инвариантных последовательностей кодов вирусов в исполняемых файлах. При некорректной организации поиска кодов вирусов или при недостаточно жестких дополнительных проверках подозрительных программ применение детектора может дать странные результаты. Так, одна из первых диагностических программ - DIAG, разработанная польским специалистом Е. Собчиком (J. Sobczyk), утверждала, что инфицирована вирусом, хотя, конечно, это было не так. Просто из-за некорректной реализации поиска кодов вирусов программа DIAG, обнаружив в своем теле шаблон для поиска вируса, принимала его за часть кода вируса. Например, одна из версий известной антивирусной программы ANTI-KOT сообщала, что эта программа длиной менее 600 байт поражена вирусом длиной 648 байт. Организация поиска характерных кодов вирусов важна еще и потому, что программа может быть одновременно заражена несколькими вирусами или одним вирусом несколько раз.

Качественная программа-детектор, безусловно, должна правильно реагировать на такие ситуации, сообщая, какими вирусами или сколько раз инфицирован исполняемый файл.

Последняя из особенностей, связанная с реализацией качественных программ-детекторов, относится к эффективности поиска инвариантных последовательностей кодов вирусов. Детекторам довольно часто приходится просматривать значительные объемы данных; для того чтобы эта работа могла быть выполнена за приемлемое время, необходимы эффективные алгоритмы поиска инвариантных последовательностей команд.

Некоторые разработчики антивирусных программ-детекторов детально исследуют вирусы и реализуют в своих программах возможности настоящих антивирусных программ-фагов, т.е. программ, которые умеют не только обнаруживать, но и удалять из инфицированной программы код вируса, восстанавливая тем самым исходное состояние программы (правда, не всегда точно).

На мой взгляд, важнее провести "детектирование" программ, так как при обнаружении вируса почти всегда можно отыскать неинфицированную копию исходной программы в архиве. Конечно, никто не откажется заодно и "вылечить" пораженную программу, если только лечение будет выполнено корректно. Совершенно очевидно, что это значительно более сложная задача, чем просто обнаружение вируса в программе. Я бы сказал, что по этой причине доверия к программам-фагам соответственно меньше.

В настоящее время разработано множество программ-детекторов и фагов. В таблице перечислены некоторые наиболее популярные программы-детекторы и фаги, за качество которых я могу поручиться.

В документации на антивирусную программу SCAN49 упоминается созданная фирмой IBM программа-детектор, которая по ассортименту диагностируемых вирусов и по виду цепочек команд для поиска не совпадает с программой SCAN49. Существует мнение, что цепочки команд для поиска эта программа берет из специального текстового файла, содержащего описания вирусов, и поэтому для нее добавление нового вируса к списку обнаруживаемых - тривиальная операция.

Достоинства качественных программ-де-

Возможна ли идеальная классификация?

Прежде чем начать разговор об антивирусных программах, хотелось бы отметить, что антивирусные программы делятся на универсальные (реагирующие на заражение любым вирусом) и специализированные (реагирующие на заражение лишь определенными вирусами и абсолютно нечувствительные к присутствию любых других вирусов). Идеалом, конечно, является универсальная программа, которая четко обнаруживает и надежно защищает от любого вируса.

Приведенная в статье классификация антивирусных программ не является исчерпывающей и точной.

Во-первых, постоянно появляются все более изощренные вирусы, разрабатываются все более совершенные программы противодействия им и, соответственно, размываются границы между классами антивирусных программ.

Во-вторых, после прочтения статьи вам станет ясно, что качественная антивирусная программа должна реализовывать функции антивирусных программ сразу нескольких классов (так, собственно, и происходит на деле). Наконец, разработано множество программ для проверки различных эвристических соображений. Эти программы практически не отражены в приведенной классификации.

текторов и фагов совершенно очевидны: они позволяют точно и недвусмысленно определить наличие в компьютере вируса, а в некоторых случаях и оперативно излечить инфицированные программы.

Принципиальными недостатками даже самых совершенных программ-детекторов являются: их неуниверсальность (программа-детектор рассчитана на обнаружение вируса конкретного типа) и запаздывание по отношению к появлению новых вирусов (детекторы всегда появляются с опозданием, поскольку, чтобы научить детектор распознавать вирус, нужно предварительно выделить и исследовать его).

При написании детекторов и фагов следует помнить, что эти программы, имеющие дело с самыми разнообразными опасными вирусами, должны быть защищены (по крайней мере, должны уметь обнаруживать свое заражение!), иначе они очень быстро превращаются в источник заражения вирусами. Последнее замечание относится также к архивированным программам, т.е. к программам, которые хранятся в архиве (в сжатом виде). Дело в том, что исполняемые файлы часто помещают в архивы, а после архивации, конечно, ни один детектор не может обнаружить вирус в зараженной программе. Правда, в документации на программу-детектор SCAN49 упоминаются какие-то интерфейсные программы, которые позволяют сканировать исполняемые файлы в архивированном виде.

Программы-вакцины

Этот класс универсальных антивирусных программ применяется только для борьбы с вирусами общего назначения.

Учитывая опасность вирусного заражения, можно позаботиться о защите своей программы еще во время ее написания, встроив в нее некоторую диагностическую часть, реагирующую на внедрение вируса. Можно, например, запомнить в каком-нибудь месте программы длину некоторых исполняемых модулей или последовательность машинных кодов в окрестности точки входа в программу; вычислить и запомнить какой-нибудь вариант контрольной суммы программы и т.д. В процессе работы ваша программа должна проверять все эти характеристики, определяемые при создании исполняемого файла. Несовпадение текущих и запомненных (эталонных) значений этих характеристик свидетельствует о несанкционированном изменении программы, а это прямое указание на ее инфицирование. Не стоит, однако, думать о возможности вирусного заражения при написании любой программы; существует, наконец, множество уже написанных программ, которые никто и не предполагал защищать.

Можно ли защитить ранее написанные программы? Конечно, можно. Мне известны, по крайней мере, три отечественные программы-вакцины: STAMPER А. Чижова, PROTECT Д. Стефанкова и СТРАЖ Ф. Шерстюка.

Принцип действия программы-вакцины состоит в том, что она внедряется в защищаемую программу подобно вирусу общего назначения и запоминает перечисленные выше характеристики последней. При запуске защищенной, но не зараженной программы управление получает программа-вакцина, проверяющая состояние защищенной программы примерно так же, как это сделал бы программист.

Допустим, что после установки защиты исполняемую программу заражает какой-нибудь вирус. Как работает такая дважды модифицированная программа? Понятно, что сначала управление получает вирус, который делает свое "черное" дело, и помешать этому программа-вакцина, конечно, не может. После завершения инициализации вирус должен передать управление исполняемой программе; на самом же деле в процессе установки защиты исполняемая программа была так изменена, что управление получает программа-вакцина. Она проверяет текущее состояние исполняемой программы, которое, естественно, отличается от эталонного (изменения внес вирус при инфицировании этой программы), и немедленно поднимает тревогу, обнаружив отличие. В некоторых случаях, например, если повреждения программы при ее инфицировании не слишком велики или объем запомненной программой-вакциной информации достаточно велик, программа-вакцина может "вылечить" зараженную программу (примерами могут быть такие вакцины, как СТРАЖ, PROTECT и т.п.).

Вирусы загрузчиков заражают не исполняемые файлы, а файловую систему в целом, поэтому для борьбы с ними изложенный выше способ защиты использовать непосредственно нельзя, однако в модифицированном виде он годится и для этой цели.

Достоинство программ-вакцин состоит в том, что с их помощью можно защитить любые программы, а также их оверлейные части от всех вирусов общего назначения, в том числе и от еще не разработанных вирусов.

Недостатки рассмотренного выше способа борьбы тоже понятны:

- программы-вакцины не могут обнаружить заражение программы, если оно произошло до момента включения защиты;
- длина защищенной программы возрастает;
- время загрузки каждой защищенной программы увеличивается;
- вирус, зная "устройство" программы-вакцины, может обойти защиту;
- программы-вакцины не помогают в случае троянских программ.

Программы слежения за состоянием файловой системы

Программы этого класса, очень похожие на программы-вакцины, описанные выше, реагируют на попытки заражения исполняемых файлов, следя за их состоянием: запоминают длину и контрольные суммы избранных исполняемых файлов, дату их создания и другие характеристики. Отличие рассматриваемых программ от программ-вакцин состоит в том, что характеристики исполняемых файлов запоминаются в отдельных файлах. При этом длины исполняемых файлов не увеличиваются, и процесс слежения за их состоянием для вируса остается незамеченным, т.е. вирус в принципе не может узнать о том, что состояние каких-то файлов контролируется, и поэтому не может имитировать запомненное состояние.

Программы слежения за состоянием файловой системы универсальны, т.е. реагируют на попытку заражения любым вирусом; отслеживания различных версий вирусов не требу-

Не только обнаружение, но и борьба

По моему твердому убеждению, самое важное требование пользователя к антивирусной программе - быстро и надежно установить факт заражения компьютера вирусом. Этой информации достаточно для того, чтобы начать борьбу с незваным пришельцем. Конечно, если есть возможность использовать специально сконструированный для такой борьбы инструмент, результата можно достичь намного быстрее и с меньшими затратами, но это, по выражению математиков, не является необходимым условием.

У разработчиков антивирусных программ рано или поздно появляется соблазн использовать свои знания о вирусах для того, чтобы предоставить в распоряжение пользователя инструмент не только для обнаружения вирусов, но и для борьбы с ними. Для некоторых классов антивирусных программ это - естественное развитие их возможностей. Для программ, занимающихся поиском вирусов в исполняемых файлах, такое усовершенствование состоит в том, чтобы научить эти программы не только обнаруживать, но и удалять вирусы из зараженных файлов. Так программы-детекторы вирусов превращаются в программы-фаги. Для программ-мониторов проверка ситуаций, характерных для конкретных вирусов, пожалуй, тоже оправданна. Возможно, что при этом программа-монитор в какой-то степени теряет свою универсальность, но приобретает при этом дополнительный "запас прочности".

Для программ других классов такое расширение поля их деятельности едва ли разумно: программа разбухает, становится аморфной и неудобной в употреблении.

Название	Разработчик	Характеристика
CRCDOS	Р. Фейт (R. Faith)	Работает в двух режимах: сбор информации о файлах, перечисленных в списке, и проверка актуальной информации на совпадение с запомненной в момент ее сбора.
Vacine 1.3	Фирма Art Hill	Запоминает информацию о системных файлах, создает каталог с копиями этих файлов под странными именами.
DLI	В. Герасимов	При первом запуске запоминает состояния файлов на логическом диске; при повторном запуске проверяет совпадение текущего и запомненного состояний файлов, сообщая об изменениях, уничтожении и создании новых файлов.

ется. Эти программы можно модифицировать так, чтобы они обнаруживали и заражения вирусами загрузчиков.

Однако при всей своей привлекательности полностью решить проблему борьбы с вирусами эти программы не могут, так как обладают рядом недостатков:

1. Эффективность программ слежения зависит от частоты их запуска - если программой слежения пользоваться раз в год, лучше этого не делать вообще.

Регулярно приходится тратить значительное время на просмотр файловой системы (затраты времени зависят от объема информации, записанной на ваших дисках, числа просматриваемых файлов, характера проводимых программой слежения проверок и их качества). Правда, это проблемы решаемые - запускает же операционная система UNIX при каждой перезагрузке программу FSCK, предназначенную для просмотра и "ремонта" файловой системы, тем более, что реально нужно следить за состоянием только наиболее часто используемых программ.

2. Программы слежения за состоянием файловой системы большого практического значения не имеют, так как они бессильны против троянских программ, разрушающих файловую систему (когда файловая система сильно разрушена, следить, пожалуй, уже не за чем).

3. Программы слежения, подобно программам-вакцинам, не в состоянии обнаружить заражение программы, если оно произошло до момента включения защиты, т.е. они не реагируют на уже зараженные программы.

4. Применение программ слежения требует от пользователя определенной компьютерной грамотности.

К настоящему времени разработано множество программ слежения за состоянием системных файлов (возможности этих программ по отслеживанию и даже восстановлению системных файлов патологически гипертрофированы, но, к сожалению, ничего, кроме слежения за состоянием этих файлов, подобные программы делать не умеют). Наличие такого большого числа программ слежения за состоянием системных файлов тем более странно, что пока известен лишь один специализированный вирус - Lechigh, поражающий только системные файлы.

В таблице приведены программы слежения, с которыми мне приходилось часто работать.

Программы-мониторы

В настоящее время можно, наверное, говорить как о классических программах-мониторах, которые являются универсальными антивирусными программами и позволяют в какой-то степени активно противодействовать вирусам и троянским программам, так и о более совершенных программах-мониторах, которые, частично утратив свойства универсальности, позволяют более эффективно (избирательно) противостоять вирусам.

Классическая программа-монитор старается заблокировать не только процесс размножения (репликации) вирусов, который является неотъемлемой частью жизни вирусов, но и вообще любые опасные, по мнению программы-монитора, попытки доступа к жесткому или гибкому диску. Такая программа-монитор пытается перлюстрировать запросы на доступ к дискам: если она считает, что запросы на модификацию данных опасны, то поднимает тревогу и при отсутствии разрешения на модификацию блокирует исполнение запросов. Для проверки правильности запросов на изменение данных эти программы используют различные алгоритмы, отслеживая попутно те или иные события, имеющие, возможно, отношение к деятельности вирусов.

Следует отметить, что программам этого класса присущи два недостатка, сильно снижающих их ценность:

1. Наличие на компьютерах BIOS и возможность непосредственно выдавать команды ввода-вывода в DOS делают надежды на качественную защиту дисков довольно призрачными. Я готов поверить в то, что вирус не будет выдавать команды ввода-вывода из соображений экономии длины кода, но почему бы ему не передавать управление непосредственно в точку входа прерывания int 13? Найти эту точку - не такая уж неразрешимая задача; обнаружить же и, тем более, заблокировать прямую передачу управления в BIOS в незащищенном режиме (real mode) невозможно.

2. Действия, которые программы-мониторы должны отслеживать и блокировать, по-существу не являются криминальными. Многие прикладные программы при обращении к данным на дисках используют те же приемы, что и вирусы. Например, в редакторе Brief при модификации файлов часто применяется доступ по абсолютному адресу сектора. Поэтому на программы-мониторы обрушивается целый "шквал" запросов ввода-вывода, среди которых трудно выделить запросы, действительно обусловленные деятельностью вирусов. Дело осложняется тем, что вирусы, как правило, не пользуются штатными средствами DOS, а модифицируют управляющие блоки непосредственно. Аппаратные же средства защиты памяти в DOS не работают, поэтому программы-мониторы, чтобы не пропустить момент запуска вируса, вынуждены обеспечивать защиту постоянно, а не только в моменты реальной опасности.

Хорошая программа-монитор обеспечивает защиту дисков на разных уровнях:

- на системном уровне работы с файлами (при их создании, открытии, позиционировании, чтении и записи) с помощью блока FCB (File Control Block) или с помощью доступа к файлам (File Handle), подобного осуществляемому при использовании ОС UNIX;
- на уровне доступа к секторам по абсолютным адресам и на уровне физической адресации (при выборе номера цилиндра, дорожки и сектора).

Известно, что один системный запрос вызывает серию запросов на уровне физической адресации, и программе-монитору приходится разбираться, чем вызваны эти запросы на доступ: действительно системными запросами или "творчеством" вируса либо троянской программы. Выше было показано, что сделать это корректно не представляется возможным, поэтому программы-мониторы часто поднимают тревогу в совершенно безобидных ситуациях. Это не только создает неприятности для пользователя, но быстро приводит к тому, что у него вырабатывается "соглашательский" стереотип, в результате чего применение программы-монитора становится практически бессмысленным.

Следует подчеркнуть, что несмотря на такие серьезные недостатки, программы-мониторы позволяют активно бороться и с вирусами (почти с момента их появления в компьютере, блокируя процесс репродукции), и с троянскими программами (в том случае, если они начинают уж очень существенно разрушать файловую систему).

Ниже в качестве примера перечислены не-

которые наиболее популярные программы-мониторы:

- VIRBLK - простая программа-монитор, которая поднимает тревогу при попытке изменить или создать файл типа COM или EXE;
- ANTI4US - более сложная программа-монитор, отслеживающая разные события и обладающая возможностью перенастройки в процессе работы;
- FSP - наиболее совершенная программа-монитор, проверяющая законность доступа к файлам на всех уровнях; эта программа минимально "шумит", отслеживая различные "полезные" события, например изменения в КМОП-памяти. Имеется возможность настроить программу FSP на защиту определенных логических частей диска и осуществить таким образом отслеживание состояния файловой системы (при своем старте или при запуске избранных исполняемых файлов программа FSP проверяет контрольные суммы этих файлов; можно указать и имена исполняемых файлов, и способ проверки контрольной суммы). К сожалению, версия 1.41 программы FSP и версия 2.0 пакета Norton Commander сильно взаимодействуют друг с другом.

К программам-мониторам тесно примыкает ряд других программ, разрабатывавшихся не для борьбы с вирусами, а для других целей, но неожиданно оказавшихся подспорьем в борьбе с вирусами. К числу таких программ относятся программы обслуживания диска, разделения доступа (например, ADM, WATCHDOG), а также некоторые программы кэширования диска, позволяющие защищать от записи какие-то из обслуживаемых ими логических устройств (например, программа SC, разработанная фирмой MAXTOR). В ряде случаев программы обслуживания или кэширования диска позволяют получить лучшие по сравнению с обеспечиваемыми программами-мониторами результаты в борьбе с вирусами хотя бы потому, что, в отличие от последних, реализуют какие-то варианты защиты с помощью пароля, а не простое подтверждение запроса. Программы обслуживания диска, часто встраиваемые в операционную систему с помощью драйверов, имеют больше возможностей для корректного разделения запросов на *правильные* (обусловленные нормальным функционированием системы) и *вирусные* (связанные с наличием вирусов). Однако без использования режима защиты процессора (protection mode) ни программы кэширования, ни программы обслуживания диска, ни пакеты разделения доступа, ни даже программы-мониторы не обеспечивают надежной защиты компьютера от вредного воздействия вирусов.

Современные программы-мониторы отслеживают более специфические события, обусловленные действиями вирусов, чем классические программы-мониторы. Поэтому для них не характерен высокий уровень ложных тревог, являющийся бичем классических мониторов.

Например, SCANRES - программа-монитор, разработанная тем же Макафи, сканирует исполняемый файл в процессе загрузки, пытаясь обнаружить в этом файле вирус. Конечно, универсальность у такого монитора полностью потеряна, и в смысле вирусной специализации эта программа-монитор эквивалентна программе-детектору (точнее, это резидентный детектор); от монитора, однако, осталась оперативность реагирования.

Программа-монитор ANTI-KOR, разработанная О. Котиком, построена исходя из эвристических соображений: после завершения исполнения каждой программы монитор просматривает таблицу векторов прерываний и состояние блоков управления памятью. По мнению разработчика, вирус общего назначения с резидентной частью или вирус загрузчика внесет какие-то изменения в распределение векторов прерываний и памяти. Рассматриваемая программа-монитор сохраняет свойство универсальности по отношению к вирусам с резидентной частью и вирусам загрузчиков, но не реагирует на вирусы без резидентной части и троянские программы. Следует отметить, что это относительно новая программа, поэтому информация об ее свойствах носит предварительный характер.

Антивирусные программы SBM (разработчики Б. Мостовой и В. Еременко) и CHECK21 (разработчик В. Двоеглазов) - призеры по классу мониторов Киевского семинара (руководитель Н. Безруков). Эти программы, используя изящные приемы работы с прерываниями по времени, не позволяют вирусам перехватывать прерывание int 21 и в большинстве случаев успешно подавляют репликационную активность вирусов (к сожалению, подобные их действия не распространяются на троянские программы).

Мне хотелось бы предложить свой вариант программы-монитора, которая годится для борьбы с вирусами общего назначения, имеющими резидентную часть. Особенность этих вирусов состоит в том, что во время старта они вынуждены проверять, активна ли уже их резидентная часть. Программа-монитор, эмулирующая правильный ответ на запрос вируса об активности его резидентной части, защищает компьютер не хуже программы-монитора, которая отслеживает запросы на доступ к диску, сообщая пользователю о "сакраментальном" запросе и ставя его в известность о попытке заражения вирусом. Образно говоря, такая программа очень точно моделирует иммунитет живых организмов по отношению к перенесенной инфекции; конечно, прежде чем научить программу эмулировать активность резидентной части нового вируса, нужно обнаружить вирус и как-то исследовать его. В этом и состоит один из основных недостатков предлагаемой программы-монитора. Этот недостаток присущ и детекторам, однако следует помнить, что для написания программы-монитора и программы-детектора необходимо исследовать различные аспекты функционирования вируса.

Хочу подчеркнуть, что пессимизм, прозвучавший при описании программ-мониторов, не стоит понимать слишком буквально, так как до настоящего времени эти программы являются единственным средством активного противодействия вирусам или троянским программам (особенно это относится к различным не классическим, а более совершенным программам-мониторам, поскольку разработчики вирусов еще не обратили на них внимания).

При просмотре вновь полученных потенциально зараженных или, возможно, троянских программ с помощью мониторов программы с

Вниманию пользователей пакетов Lotus, Framework, Brief, Multi-Edit и других пакетов прикладных программ, имеющих достаточно мощный аппарат макрокоманд и допускающих создание пользовательских макросов!

В августовском номере журнала "Computers & Security" за 1989 г. один из его редакторов Х. Дж. Хайленд (H. J. Highland) в рубрике *Random Bits & Bytes* описал душераздирающую историю конструирования макровируса в среде Lotus. Насколько я понял, автор рассматривает все-таки не вирус, поскольку о размножении вредной макрокоманды речь не идет, а троянскую (в соответствии с принятой нами терминологией) макрокоманду - макрос; он утверждает, что "аналогичные приемы можно использовать для создания троянских макрокоманд в любой другой программе, позволяющей пользователю писать свои макросы".

Я хочу обратить внимание читателей на то, что такие троянские макрокоманды очень специфичны. Вероятно, для обнаружения этих макрокоманд ни одна из ранее разработанных программ-детекторов не годится, так как проявляют они себя только во время работы в той программной среде, для которой написаны. Тот способ обнаружения троянских макросов, который предлагает Х.Дж. Хайленд, специфичен для пакета программ Lotus. Кстати, этот способ похож на те простые способы обнаружения вирусов, которые описаны в статье об устройстве вирусов (см. "Интеркомпьютер", 1989, N 2). Во всяком случае, будьте внимательны и обращайте внимание на любые несуразности, которые могут возникать при работе с данными. Помните, что троянские программы появились уже и в среде прикладных пакетов.

В книге *Computer Viruses: A High-Tech Decease* Р. Бергера (R. Burger) рассмотрен вирус, написанный на языке пакетных файлов DOS.

Результаты серьезного математического исследования вирусов, выполненного знаменитым Ф. Коуэном (F. Cohen), приведены в статье *Computational Aspects of Computer Viruses*, опубликованной в том же августовском номере журнала "Computers & Security".

аномальными свойствами удается выявить еще до того, как разрушительные действия вирусов или троянских программ проявятся в полной мере; это позволяет активно противостоять натиску инфекции.

Завершая обзор антивирусных программ, мне бы хотелось упомянуть еще об одном весьма распространенном классе программ, которые, строго говоря, нельзя назвать антивирусными. Речь идет о программах, предназначенных для демонстрации содержимого оперативной памяти компьютера и определения состояния векторов прерываний. Эти программы очень полезны, а в ряде случаев просто необходимы при сражениях с вирусами, так как позволяют с разной степенью детализации (правда, иногда не совсем корректно) просмотреть, как используется оперативная память: вирусы загрузчиков и вирусы общего назначения с резидентной частью постоянно находятся в оперативной памяти, и эффективные программы, предназначенные для демонстрации содержимого памяти, позволяют быстро установить факт заражения компьютера вирусом. К настоящему времени разработано множество подобных программ: MEM, MAPMEM, VTSR, MI, PCSTAT, PCMAP.

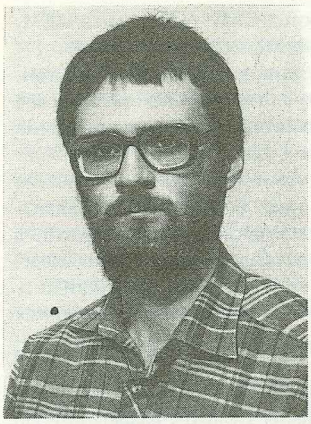
Особенность этих программ состоит в том, что вирусы могут отлично маскировать свое присутствие в памяти, и лишь одна программа из перечисленных выше - PCMAP - позволяет установить сам факт заражения компьютера вирусом загрузчика. Большинство резидентных частей кодов вирусов можно разместить в буферах DOS, где обнаружить их очень сложно. Например, вирус, инфицировавший системный драйвер, формально становится частью DOS, поэтому его весьма трудно обнаружить.

Мне хотелось бы обратить внимание читателей на программу PCMAP, разработанную Д. Стефанковым, которая подробно и с максимальной точностью демонстрирует "анатомию" использования оперативной памяти и векторов прерывания компонентами DOS и резидентными программами. Эта программа рассчитана на профессионалов, которым она предоставляет уникальные средства демонстрации содержимого оперативной памяти как для обнаружения вирусов, так и для исследования функционирования отладчиков и пакетов прикладных программ. Еще одной приятной особенностью программы PCMAP является "вирусоустойчивость": при старте она проверяет свою длину и длины некоторых других кодов, сообщая пользователю о факте своего заражения; в некоторых случаях эта программа способна самовосстанавливаться.

Хочу сообщить читателям, что в апрельском номере журнала "PC Magazine" за 1989 г. (с. 193 - 228) помещен достаточно подробный обзор западных антивирусных программ и пакетов. Еще больше антивирусных программ рассмотрено в книге *Computer Viruses* Р. Робертса (R. Roberts), но там информация более старая и менее подробная.

В одном из следующих номеров "Интеркомпьютера" предполагается привести список троянских программ с минимальными комментариями, обсудить способы "лечения" зараженного компьютера, а также поговорить о тривиальных, но весьма эффективных приемах "техники безопасности" при работе с зараженным компьютером.

АНТИВИРУСЫ



Ф.Н. ШЕРСТЮК

Вирусы и антивирусы на IBM-совместимых ПК

Большинству пользователей персональных компьютеров (ПК) приходилось сталкиваться с компьютерными вирусами, а тот, кто с ними еще не знаком, к сожалению, скоро познакомится: вирусы все чаще поражают ПК. Происхождение их самое разнообразное: вирусы пишут хулиганствующие программисты для того, чтобы потешить свое самолюбие или заработать деньги на продаже антивирусов (антивирусных программ), разрабатывают даже в кругах, близких к таким крупным фирмам-разработчикам ПК, как IBM, в целях борьбы с фирмами, производящими компьютеры-клоны.

Несколько советов

Известно, что в программировании, как и в медицине, лучший способ не заразиться - это профилактика. Для предотвращения заражения компьютера вирусами рекомендуется следовать приведенным ниже советам.

Пользуйтесь только проверенными программами (абсолютной гарантии, конечно, никто дать не может, но программа с фирменной дискеты или программа, которой ваш хороший знакомый регулярно пользовался не менее полугода, вероятно, не инфицирована).

Избегайте "коллекционирования" программ, которыми вы не пользуетесь.

Помните: самыми распространенными источниками вирусов являются игры и антивирусные программы.

Компьютеры пользователей, которые следуют этим советам, крайне редко заражаются (компьютер применяется для реальной работы, выполняемой с помощью сложившегося инструментария, для расширения которого нужны очень веские основания; на диске нет места для "коллекционирования" ненужных программ; не хватает времени для компьютерных игр).

Необходимо иметь на защищенных от записи дискетах копии всех файлов, с которыми вы работаете, в том числе и текстовых (существуют вирусы, которые вносят не сразу заметные изменения в текстовые файлы). Особое внимание следует обратить на резервные копии операционной системы (их должно быть не менее двух).

Если вирус все же проник в ваш компьютер, то самый простой и надежный (хотя и наиболее трудоемкий) способ избавиться от него, не "подцепив" при этом новый вирус, состоит в следующем: перезагрузите операционную систему (нажав кнопку RESET или выключив/включив питание, а не с помощью комбинации клавиш Ctrl-Alt-Del) с дискеты, содержащей незараженную копию операционной системы, отформатируйте все незащищенные диски и дискеты (существуют вирусы, которые заражают не файлы, а диски), затем восстановите на дисках всю информацию с резервных копий. Пусть этот совет также не покажется утопией - я знаю организацию, где подобная процедура проводилась на 10 компьютерах приблизительно раз в неделю (так часто там появлялись новые вирусы).

Способы "лечения"

Скорее всего, советы, изложенные выше, для вас не вполне приемлемы, и вам, вероятно, придется воспользоваться антивирусными программами. Все антивирусные программы можно разделить на специализированные и универсальные.

Большинство известных антивирусных программ (AIDSTEST, VDEATH, SERUM3, ANTI-KOT и множество других) являются *специализированными*: каждая из них обнаруживает наличие одного или нескольких конкретных вирусов, никак не реагируя на присутствие остальных (это так называемые программы-детекторы). Некоторые из этих программ могут даже "вылечить" зараженный файл (это так называемые программы-фаги).

Недостатками описанных антивирусных программ являются:

слишком узкая специализация: уже сейчас известно множество вирусов, и число их стремительно растет, в частности за счет появления новых разновидностей (штаммов), вызванных мутациями - изменениями, которые хулиганствующий программист может легко внести в вирус, после чего соответствующий антивирус перестанет опознавать этот вирус;

обилие ошибок из-за того, что такие программы, как правило, пишутся в спешке;

большая вероятность заражения при использовании (практика показывает, что именно специализированные антивирусные программы чаще всего являются разносчиками вирусов).

Универсальные антивирусные программы предназначены для борьбы с целыми классами вирусов. В отличие от специализированных

антивирусных программ, предназначенных для борьбы с уже исследованными вирусами, универсальные антивирусные программы можно использовать для борьбы как с известными, так и с неизвестными (еще ненаписанными) вирусами. Мне приходилось сталкиваться с универсальными антивирусами двух типов: резидентными антивирусами и программами-ревизорами.

К резидентным антивирусам относятся ANTI4US, FLUSHOT, CDM и несколько других программ, которые препятствуют "размножению" вирусов, для чего при обнаружении подозрительных действий запрашивают у пользователя их подтверждения.

Основными недостатками этих программ являются назойливость (действия, требующие подтверждения, в подавляющем большинстве случаев совершаются вовсе не вирусами, а вполне невинными программами) и низкая надежность (разработчикам вирусов известно несколько простых и весьма эффективных способов обхода таких антивирусных программ). Кроме того, резидентные антивирусные программы "съедают" много оперативной памяти и вступают в клавиатурные конфликты с другими программами.

Программа-ревизор сравнивает некоторые параметры файлов (длину, контрольную сумму и т. п.) с эталонными, что позволяет обнаружить, подвергался ли файл несанкционированным изменениям. Однако в силу того, что такая программа работает довольно долго, ревизии проводятся не чаще одного раза за сеанс работы с компьютером либо не над всеми файлами.

Печальные выводы

Из вышесказанного можно (и нужно) сделать вывод, что в настоящий момент антивирусные программы либо неэффективны, либо опасны, либо и то, и другое вместе. На мой взгляд, причины возникновения этой печальной ситуации таковы:

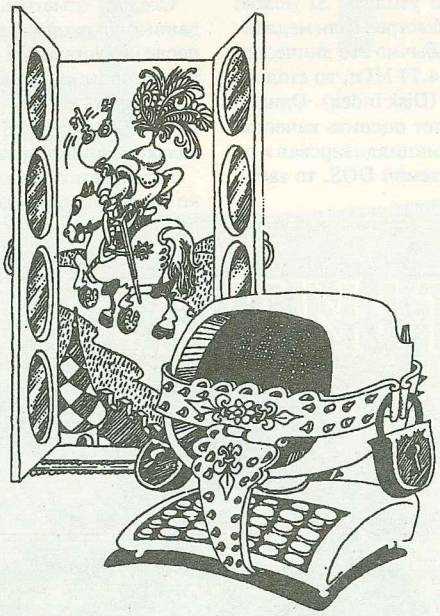
1. Программисты высокой квалификации уделяют разработке антивирусов недостаточно внимания. Это естественно: вирусы у них практически не заводятся; кроме того, у таких программистов есть значительно более серьезные задачи, чем написание антивирусов.

2. Центр тяжести в борьбе с вирусами в настоящее время, к сожалению, перенесен на "индивидуальный террор", т.е. на разработку специализированных антивирусных программ. Применение этого метода, проистекающего от неверия в полную победу над вирусами, с ростом числа вирусов неизбежно приведет к полному поражению. По количеству и качеству универсальные антивирусные программы в настоящее время серьезно уступают специализированным, что является серьезной методологической ошибкой.

3. Другая ошибка, регулярно допускаемая при создании универсальных антивирусных программ, состоит в непонимании того факта, что никакая антивирусная программа не должна сковывать свободу действий пользователя, поскольку она предназначена для облегчения его деятельности.

4. К сожалению, сведения об "анатомии" вирусов в настоящее время получили чрезвычайно широкое распространение (они даже публикуются в печати). Результат, на мой взгляд, может быть только один: резкий рост числа штаммов имеющихся вирусов и появление новых отечественных вирусов.

5. Некоторым людям наличие вирусов выгодно, например тем, кто продает антивирусные программы или предлагает свои услуги по обеззараживанию компьютеров, либо тем, кто хочет какие-то свои огрехи списать на действия вирусов, и т.д.



Что же делать?

На мой взгляд, недопустимо упускать инициативу из своих рук, отдавая ее разработчикам вирусов. Следует подчеркнуть, что несмотря на обилие самих вирусов, идей, положенных в их основу, совсем немного, причем существенного роста числа этих идей пока не предвидится. Все имеющиеся на сегодняшний день методы борьбы с вирусами практически исчерпали себя, необходимо разработать принципиально новые. В связи с этим я хочу поделиться некоторыми соображениями.

Все известные мне вирусы можно разделить на три типа:

1. *Специализированные системные вирусы.* Эти вирусы портят системные файлы (IO.SYS, MSDOS.SYS, IBMBIO.COM, IBMDOS.COM, COMMAND.COM), не удлиняя их (наименее распространенный тип вирусов).

2. *Вирусы загрузчиков.* Они портят загрузчик и перехватывают прерывание int 13. Вирусы этого типа достаточно широко распространены.

3. *Вирусы общего назначения.* Эти вирусы инфицируют как прикладные, так и системные программы, т.е. удлиняют файлы типа EXE и COM. Это наиболее распространенный тип вирусов.

Борьба с вирусами каждого из этих типов требует индивидуального подхода, но общая идеология разработки соответствующих антивирусов состоит в том, что если нельзя предотвратить размножение вирусов (опыт применения резидентных универсальных антивирусов показывает, что это действительно так), то не стоит и пытаться это делать, но выявить факт заражения надо на возможно более ранней стадии. Желательно также попробовать удалить вирус из зараженного файла или с диска. Обе эти задачи можно решить, лишь располагая достаточной информацией о параметрах эталона - незараженного файла или диска, и достаточно часто сравнивая объект возможного заражения с эталоном. Посмотрим, как эту идеологию применить для борьбы с вирусами каждого из перечисленных выше типов.

Поскольку число системных файлов невелико, то имея их эталонные копии можно из файла AUTOEXEC.BAT запускать программу сравнения системных файлов с эталонной копией.

Вирус загрузчика перехватывает прерывание int 13 еще до начала загрузки DOS. Поэтому для обнаружения такого вируса необходимо знать, указывает вектор прерывания int 13 в момент начала загрузки на ПЗУ (в этом случае все нормально) или на ОЗУ (это означает, что вектор прерывания кем-то перехвачен - вероятно, вирусом). Для обнаружения вируса достаточно написать сравнительно простой драйвер и поместить его первым среди драйверов, расположенных в файле CONFIG.SYS. Если хранить копию загрузчика и адрес ПЗУ, на который должен указывать вектор прерывания int 13, то легко "вылечить" компьютер от вируса загрузчика.

В конец каждого файла типа EXE или COM можно дописать программу-"стража", которой в момент запуска файла передается управление; этот "страж" проверяет, не удлинился ли файл, к которому он "прицеплен", т.е. каждый файл, защищенный "стражем", в момент запуска сам проверяет, не заражен ли он. Если в теле этого "стража" хранить также контрольную сумму файла и копии его "уязвимых мест", т.е. тех областей файла, которые скорее всего будут попорчены вирусом, то, основываясь на этой информации, зараженный файл можно восстанавливать (это возможно только в том случае, когда вирус не повредил никаких мест, кроме "уязвимых").

Говоря о реализуемости изложенного выше, следует иметь в виду, что к настоящему времени разработано множество программ для сравнения длины файла с эталоном; драйвер для защиты от вирусов загрузчиков, по-видимому, написать несложно; программа-"страж", которую я уже написал, успешно эксплуатируется в нескольких организациях, "страж" во всех случаях обнаруживает факт заражения и в подавляющем большинстве случаев позволяет "вылечить" зараженный файл.