Е.Воеводин, 913 гр.

Компьютерные вирусы и способы борьбы с ними.

Итак, что же такое вирус в компьютерной терминологии? На мой взгляд, точного определения здесь дать нельзя. Да, это программа, но какая? Очевидно, приносящая тот или иной вред ПО (программному обеспечению), мешающая работать, искажающая результаты деятельности и т. д. Данный список можно продолжать бесконечно долго, но найти грань, которая отделяла бы безобидную программу от опасной, очень и очень сложно.

Но перейдем собственно к систематизации. Существует несколько основных критериев, по которым можно классифицировать вирусы:

- поддерживаемая ОС (операционная система)
- среда обитания
- способ заражения
- особенности алгоритма работы
- деструктивные возможности.

Что касается первого пункта, здесь все достаточно понятно без особых объяснений – речь идет о способности вируса воздействовать на объекты той или иной операционной системы.

Теперь о среде обитания вирусов. Здесь можно выделить четыре категории:

- Файловые вирусы - чтобы размножиться, используют файловую систему, внедряясь в исполняемые файлы практически любой ОС: DOS, Windows, Unix/Linix, MacOS, OS/2 etc., независимо от ее версии. Иногда это могут быть даже исходные тексты программ, библиотечные или объектные модули.

- Макровирусы иногда не выделяют в отдельный класс, причисляя их к предыдущему. На мой взгляд, подобная систематизация не совсем верна, поскольку их пишут на макроязыках, встроенных в некоторые системы обработки данных - чаще всего это текстовые редакторы или электронные таблицы.

- Загрузочные вирусы заражают загрузочный сектор гибкого диска или MBR (Master Boot Record) винчестера. Они подставляют свой код вместо программы, которая должна получать управление при старте системы; кроме того, зачастую переносят boot-сектор в другую область носителя.

- Сетевые вирусы активно используют различные протоколы и возможности LAN (Local Area Network) или WAN (Wide Area Network). Основная отличительная особенность сетевых вирусов состоит в их способности самостоятельно передавать свой код на удаленную рабочую станцию или сервер.

Существует также великое множество различных комбинаций перечисленных вирусов – например, файлово-загрузочные. Бороться с ними еще сложнее, а наносимый ущерб еще ощутимее; конечно, и алгоритм работы подобных вирусов гораздо изысканнее.

На очереди не менее важная характеристика – способ инфицирования системы. Он может быть резидентным и нерезидентным. В первом случае в ОЗУ компьютера находится часть тела вируса, перехватывающая обращения ОС, например, к тем же файлам, и заражающая их. Подобные вирусы активны в течение всего сеанса работы.

Нерезидентные же, наоборот, действуют только в течение определенного промежутка времени, оставляя оперативную память "стерильно чистой".

Теперь обсудим особенности алгоритмов работы вирусов. Здесь тоже можно выделить огромное количество категорий. Остановимся на самых распространенных:

- Компаньоны - никогда не изменяют существующих файлов, а создают файл-спутник с одинаковым именем и похожим расширением. Например, в DOS'е наряду с file.exe появляется file.com - естественно, зараженный. Достаточно в командной строке дать директиву запуска file (без расширения), и ОС выполнит file.com (приоритет расширения *.com выше, чем *.exe), после чего вирус ворвется в систему.

- Сетевые черви - действуют подобно компаньонам, только они распространяются по сети. Иногда они оставляют дополнительные файлы на дисках после себя, а иногда ограничиваются пространством ОЗУ. Как и предыдущие, содержимое дисков машины они не изменяют.

- Паразиты, наоборот, при своем распространении ведут себя деструктивно по отношению к дискам и файлам.

- Невидимки - куда более сложные вирусы. Их название неслучайно, они действительно могут полностью или частично скрывать себя в системе. Иногда невидимки временно лечат пораженные участки или подставляют вместо них "здоровые", перехватывая соответствующие обращения ОС.

- Полиморфные - почти совершенные вирусы. Соответственно, их еще сложнее обнаружить - они не содержат сигнатур, т. е. ни одного постоянного участка кода. В большинстве случаев два образца одного и того же полиморфного вируса не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика. Такой прием очень популярен. В том или ином виде его можно встретить во многих существующих разновидностях вирусов. Иногда его еще называют самошифрованием.

- Студенческие. Данные вирусы не отличаются особой сложностью или оригинальностью, они зачастую нерезидентны и содержат ошибки. В одних случаях это спасает, в других приводит к еще более плачевным результатам.

Наконец, обратимся к деструктивным возможностям. Обычно различают четыре категории вирусов (хотя деление весьма условно):

- Переселенцы - влияние таких вирусов на систему можно свести только к распространению, а значит, лишь некоторому уменьшению количества свободных ресурсов.

- Шутники - лучше всего их воспримут люди с развитым чувством юмора. Помимо некоторого естественного уменьшения общей производительности системы, они могут... спеть песенку или нарисовать на экране солнышко. Все зависит от фантазии самого программиста и от его уровня культуры.

- Диверсанты - приводят к серьезным сбоям в работе системы.

- Разрушители - уничтожают программы, данные, затирают системные области и т. п.

Закончив классификацию вирусов по всем возможным параметрам, самое время сказать: существует и множество других вредных программ, так или иначе нацеленных на нанесение ущерба системе. Это троянские кони, проявляющие себя в самый неподходящий момент, злые шутки с сообщениями о нанесении якобы ущерба компьютеру, хотя на самом деле ничего не происходит, есть даже конструкторы вирусов и полиморфик-генераторы. С их помощью в наш век высоких технологий даже ребенок без особых усилий может создать весьма грозное оружие.

Теперь давайте сделаем краткий обзор некоторых, наиболее распространённых алгоритмов заражения.

Загрузочный вирус.

Загрузочные вирусы заражают загрузочный (Boot) сектор флоппи-диска и Boot-сектор или Master Boot Record (MBR) винчестера. При инфицировании диска вирус в большинстве случаев переносит оригинальный Boot-сектор (или MBR) в какой-либо другой сектор диска (например, в первый свободный). Если длина вируса больше длины сектора, то в заражаемый сектор помещается первая часть вируса, остальные части размещаются в других секторах (например, в первых свободных). Затем вирус копирует системную информацию, хранящуюся в первоначальном загрузчике, в свои коды и записывает их в загрузочный сектор (для MBR этой информацией является Disk Partition Table, для Boot-сектора дискет - BIOS Parameter Block).

Существует несколько способов размещения на диске первоначального загрузочного сектора и продолжения вируса:

- в сектора свободных кластеров логического диска
- в неиспользуемые или редко используемые системные сектора
- в сектора, расположенные за пределами диска.

Если продолжение вируса размещается в секторах, которые принадлежат свободным кластерам диска (при поиске этих секторов вирусу приходится анализировать таблицу размещения файлов - FAT), то, как правило, вирус помечает в FAT эти кластеры как сбойные (так называемые псевдосбойные кластеры). Этот способ используется вирусами "Brain", "Ping-Pong" и в дальнейшем будет именоваться как СПОСОБ "BRAIN". Вирусы семейства "Stoned" используют другой метод - они размещают старый загрузочный сектор в неиспользуемом или редко используемом секторе. На винчестере этот сектор является одним из секторов (если такие есть), расположенных между MBR и первым Boot-сектором, а на дискете этот сектор выбирается из последних секторов корневого каталога. В дальнейшем этот метод будет называться СПОСОБ "STONED".

Реже используется метод сохранения продолжения вируса за пределами диска, этот метод пока встречался только при заражении вирусом флоппи-дисков. Для этого вирусу приходиться форматировать на диске дополнительный трек (метод нестандартного форматирования), например, 40-й трек на 360K дискете. Конечно, существуют и другие методы размещения вируса на диске, например, вирусы семейства "Azusa" содержат в своем теле стандартный загрузчик MBR и при заражении записываются поверх оригинального MBR без его сохранения. В качестве некого итога можно привести обобщённый алгоритм работы загрузочного вируса:

- уменьшение объема свободной памяти
- считывание с диска своего продолжения (если оно есть)

- перенос своего тела в другую область памяти
- установка необходимых векторов прерываний

- дополнительные действия

- копирование в память оригинального Boot-сектора и передача на него управления

Резидентный вирус.

DOS предусматривает единственный способ создания резидентных (TSR) модулей (помимо драйверов, указываемых в CONFIG.SYS) – при помощи функции KEEP (INT 21h, AH=31h или INT 27h). Многие файловые вирусы для маскировки своего распространения используют другой способ – обрабатывая системные области, управляющие распределением памяти (MCB), выделяют для себя свободный участок памяти, помечают его как занятый и переписывают туда свою копию. Некоторые вирусы внедряют свои TSR-копии в свободные участки памяти в таблице векторов прерываний, в рабочие области DOS, в память, отведенную под системные буферы, в блоки памяти UMB, EMS и XMS.

Известны два способа проверки резидентным вирусом наличия своей копии в памяти компьютера.

Первый заключается в том, что вирус вводит новую функцию некоторого прерывания, действие которой заключается в возврате значения "я здесь". При старте вирус обращается к ней, и, если возвращенное значение совпадает со значением "я здесь", то память компьютера уже заражена и повторное заражение не производится.

При проверке вторым способом вирус просто сканирует память компьютера. Оба способа могут в той или иной мере сочетаться друг с другом.

При инфицировании оперативной памяти вирус ищет свободное место в памяти и записывает туда свою копию. Затем вирус переопределяет одно или несколько прерываний, необходимых ему для поиска заражаемых файлов, для выполнения деструктивных действий или звуковых и видеоэффектов.

При инфицировании файлов нерезидентные и некоторые резидентные вирусы ищут на диске (дисках) эти файлы при помощи функций DOS FindFirst и FindNext (INT 21h, AH=11h,12h,4Eh,4Fh). Резидентные вирусы используют более широкий список функций DOS, при обращении к которым происходит заражение файла. Фактически в этом списке присутствуют все функции, по значениям входных или выходных параметров которых можно определить имя файла, к которому идет обращение (к таким параметрам относятся значения соответствующих регистров или областей памяти). В результате, можно выделить несколько опасных функций прерывания 21h:

- функция выполнения (EXEC, AX=4B00)

- функция загрузки в память (AH=4Bh)

- функция поиска (FindFirst и FindNext, AH=11h,12h,4Eh,4Fh)

- функция создания (Create, AH=3Ch)

- функция открытия (Open, AH=3Dh)

- функция закрытия (Close, AH=3Eh)

- функция изменения атрибутов (ChMode, AH=43h)

- функция переименования (Rename, AH=56h)

Теперь время перейти к самому интересному. Как же вирусы заражают файлы.

Файловые вирусы.

Вирус может внедриться в файлы трех типов:

- командные файлы (BAT)
- загружаемые драйверы (SYS) в том числе IO.SYS и MSDOS.SYS
- выполняемые двоичные файлы (EXE, COM) включая новые типы EXE-файлов (NEW EXE) и NLM-файлы, выполняемые в операционных системах типа MS-Windows, OS/2, Novell Netware и т.д.

Возможно внедрение вируса в файлы данных, но эти случаи возникают либо в результате ошибки вируса, либо при проявлении вирусом своих агрессивных свойств. Конечно, возможно существование вирусов, заражающих файлы, которые содержат исходные тексты программ, библиотечные или объектные модули, но подобные способы распространения вируса слишком экзотичны и поэтому в дальнейшем не рассматриваются.

На сегодняшний день известно всего несколько видов BAT-вируса. Они не представляют интереса, так как достаточно примитивны и очень просто обнаруживаются, после чего удаляются.

Внедрение вируса в SYS-файл

Вирусы, внедряющиеся в SYS-файл, приписывают свои коды к телу файла и модифицируют адреса программ стратегии (Strategy) и прерывания (Interrupt) заражаемого драйвера (встречаются вирусы, изменяющие адрес только одной из программ). При инициализации зараженного драйвера вирус перехватывает соответствующий запрос операционной системы, передает его драйверу, ждет ответа на этот запрос, корректирует его и остается в оперативной памяти вместе с драйвером в одном блоке памяти. Такой вирус может быть чрезвычайно опасным и живучим, так как внедряется в оперативную память при загрузке DOS раньше любой антивирусной программы, если она, конечно, тоже не является драйвером.

Внедрение вируса в COM- и EXE-файлы

Выполняемые двоичные файлы имеют форматы COM или EXE, отличающиеся заголовком и способом запуска программ на выполнение. Расширение имени файла (".COM" или ".EXE") не всегда соответствует действительному формату файла, что, правда, никак не влияет на работу программы. Файлы COM и EXE заражаются по-разному, следовательно, вирус должен отличать файлы одного формата от другого. Вирусы решают эту задачу двумя способами: одни анализируют расширение имени файла (".COM", ".EXE"), другие - заголовок файла.

Первый способ далее будет называться заражением .COM- (или .EXE-) файлов, второй - заражением COM- (или EXE-) файлов. В большинстве случаев вирус инфицирует файл корректно, т.е. по информации, содержащейся в теле вируса, можно полностью восстановить зараженный файл. Но вирусы, как и большинство программ, часто содержат незаметные с первого взгляда ошибки. Из-за этого даже вполне корректно написанный вирус может необратимо испортить файл при его заражении. Например, вирусы, различающие типы файлов по расширению имени (.COM, .EXE), очень опасны, так как портят файлы, у которых расширение имени не соответствует внутреннему формату.

Файловые вирусы при распространении внедряются в тело заражаемого файла: в его начало, конец или середину. Существует несколько возможностей внедрения вируса в середину файла: он может быть скопирован в таблицу настройки адресов EXE-файла ("BootExe"), в область стека файла COMMAND.COM ("Lehigh"), может "раздвинуть" файл или переписать часть файла в его конец, а свои коды в освободившееся место ("April1st.Exe", "Phoenix"), и т. д. Кроме того, копирование вируса в середину файла может произойти в результате ошибки вируса - в этом случае файл может быть необратимо испорчен. Встречаются и другие способы внедрения вируса в середину файла, например, вирус "Mutant" применяет метод компрессирования некоторых участков файла.

Внедрение вируса в начало файла

Известны три способа внедрения вируса в начало файла. Первый способ заключается в том, что вирус переписывает начало заражаемого файла в его конец, а сам копируется в освободившееся место. При заражении файла вторым способом вирус создает в оперативной памяти свою копию, дописывает к ней заражаемый файл и сохраняет полученную конкатенацию на диск. При заражении третьим способом вирус записывает свои коды в начало файла, не сохраняя старого содержимого начала файла. Естественно, что при этом файл перестает работать и не восстанавливается. Некоторые вирусы, поражающие файлы первым и вторым способом, дописывают блок информации и в конец файла (например, вирус "Jerusalem" по этому блоку отличает зараженные файлы от незараженных).

Внедрение вируса в конец файла

Наиболее распространенным способом внедрения вируса в файл является дописывание вируса в конец этого файла. При этом вирус изменяет начало файла таким образом, что первыми выполняемыми командами программы, содержащейся в файле, являются команды вируса. В COM-файле в большинстве случаев это достигается изменением его первых трех (или более) байтов на коды инструкции JMP Loc_Virus (или в общем случае - на коды программы, передающей управление на тело вируса). EXE-файл либо переводится в формат COM-файла и затем заражается как COM-файл, либо модифицируется заголовок файла. В заголовке EXE-файла изменяется значение стартового адреса (CS:IP) и значение длины выполняемого модуля (файла), реже – регистры-указатели на стек (SS:SP), контрольная сумма файла и т.д. Дополнительно к этому, длины файлов перед заражением могут увеличиваться до значения, кратного параграфу (16 байт).

Алгоритм работы файлового вируса

Вирус, после передачи ему управления, совершает следующие наиболее общие для всех вирусов действия:

- восстанавливает программу (но не файл) в исходном виде (например, у COM-программы восстанавливаются первые несколько байт, у EXE-программы вычисляется истинный стартовый адрес, у драйвера восстанавливаются значения адресов программ стратегии и прерывания);

- если вирус резидентный, то он проверяет оперативную память на наличие своей копии и инфицирует память компьютера, если копия вируса не найдена. Если вирус нерезидентный, то он ищет незараженные файлы в текущем и (или) корневом оглавлении, в оглавлениях, отмеченных командой PATH, сканирует дерево каталогов логических дисков, а затем заражает обнаруженные файлы;

- выполняет деструктивные действия, производит графические или звуковые эффекты и т.д. Данные функции резидентного вируса могут вызываться спустя некоторое время после активизации в зависимости от текущего времени, конфигурации системы, внутренних счетчиков вируса или других условий. В этом случае вирус при активизации обрабатывает состояние системных часов, устанавливает свои счетчики и т.д.

- возвращает управление основной программе.

Метод восстановления программы в первоначальном виде зависит от способа заражения файла. Если вирус внедряется в начало файла, то он либо сдвигает коды зараженной программы на число байт, равное длине вируса, либо перемещает часть кода программы из ее конца в начало, либо восстанавливает файл на диске, а затем запускает его. Если вирус записался в конец файла, то при восстановлении программы он использует информацию, сохраненную в своем теле при заражении файла. Это может быть длина файла, несколько байт начала файла в случае COM-файла или несколько байтов заголовка в случае EXE-файла. Если же вирус записывается в середину файла специальным образом, то при восстановлении файла он использует также специальные алгоритмы.

О виусах и способах заражения компьютеров можно было бы говорить ещё очень и очень долго, но давайте теперь порассуждаем о том, как же, всё-таки, самому, без помощи различных антивирусных средств, попытаться найти и обезвредить вирус.

Обнаружение вируса.

Обнаружение загрузочного вируса.

Для обнаружения загрузочного вируса необходимо загрузиться с чистой дискеты и, запустив DiskEditor, заглянуть 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 COh (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. Небходимо проверить 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.

Теперь проверяем boot-сектор MS-DOS, он обычно расположен в секторе в 0/1/1. Его внешний вид для сравнения можно найти как в книге Е. Касперского, так и на любой "чистой" машине.

Итак, 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, можно оставить 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 байт 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 можно использовать и стандартный отладчик debug. Загружаем подозреваемую на наличие вируса программу (в чистой операционной системе) и теперь можно попытаться визуально распознать наличие вируса по коду. Вот на что надо обратить 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", "Ь291"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.

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. Поэтому смотрим на лицевую панель компьютера. Если в наличии дисководы обоих типов (3.5" и 5.25"), то придется отформатировать 4 дискеты на 4 стандартных формата: 360Кбайт, 720Кбайт, 1.2Мбайт и 1.44Мбайт. Затем при помощи программы DiskEditor внимательно рассмотрим и постараемся запомнить внешний вид boot-секторов этих дискет (0/0/1), хотя бы первые байты (естественно, все это делается на чистой машине). Вставляем не защищенные от записи дискеты по очереди в дисководы "больной" машины и (обязательно) обращаемся к ним: пытаемся прочитать каталог, записать, прочитать и удалить какие-либо файлы. Наконец, на чистой машине при помощи DiskEditor вновь просматриваем сектор 0/0/1. Если на какой-либо дискете он изменился, при помощи того же DiskEditor снимаем образ всей дискеты в файл. Вирус пойман. 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 copymaster 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 программу, подобную приведённой ниже программе test.com.

test.com

```
cseg segment

assume cs:cseg, ds:cseg, ss:cseg

org -lOOh
Start:

db 1249 dup (OFAh,90h,OFBh,OF8h)

1 60 Методы борьбы с вирусами

mov ah,4Ch

int 21h
```

```
cseg ends

End Start
```

Скопируем приманку на зараженную машину. Выполним над ней как можно больше операций: запустим, скопируем в другое место винчестера и на дискету, переместим, просмотрим её в NC и DOS. 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, четвертые – вообще во всех каталогах винчестера. скопируем приманку во все каталоги диска (запускаем из корневого каталога).

Теперь выбираем заведомо зараженную программу и запускаем ее N раз, постоянно изменяя время и дату. Выбираем тот файл приманки, который изменил длину. Вот вирус и пойман.

Некоторые аспекты программирования современных вирусов.

Прежде всего, хотелось бы отметить, что с появлением и массовым распространением таких операционных систем, как Windows 2000 и Windows XP, создание вирусов, работающих по общей схеме, стало довольно затруднительным занятием. Ведь теперь всё программное обеспечение, если это конечно не драйверы, происходит в так называемом Ring-3, то есть под привилегиями пользователя. Теперь нельзя напрямую записывать свой код в файл, общаться с портами и устройствами.

Теперь, создание вирусов должно происходить на совершенно другом уровне. В основе большинства вирусов, написанных под Win32, лежит перехват API функций (помните прерывания в DOS-е?), либо работа напрямую с сервисами VXD в Window9x.

Приведём пример из жизни.

Для начала приведем стандартный алгоритм простого нефайлового вируса ("form", "stone" , проч.). Перехватив тем или иным способом процесс загрузки с винчестера, вирус, если он собирается быть активным (резидентным) в течение всего сеанса работы компьютера, использует перехват прерывания(ий) с целью следить за обращениями пользовательских программ к дисководу(ам) и, выбрав момент, перенести свое тело на дискету в ее BOOT- сектор.

Самое простое, что можно сделать (и было сделано много раз) в этом направлении – это перехватить сервис BIOS прерывание int 13h, предназначенное для работы с жесткими дисками и дискетами. Такой выбор объясняется следующими причинами:

- Вирус получает управление перед или после выполнения программ MBR или BOOT активного раздела жесткого диска, когда доступными для перехвата являются только сервисы BIOS – операционная система еще не активизировала свои прерывания;

- Большинство пользовательских программ при работе с дискетами косвенно через сервисы OS вызывают int 13h и несложные проверки на номер диска, сектора и дорожки позволяют легко идентифицировать момент заражения дискеты.

Таким образом, все, что необходимо для функционирования такого класса программ – это OS, использующая для работы с накопителями ТОЛЬКО сервисы, предоставляемые BIOS.

Однако что же будет в "плохом" для нас случае, когда операционная система по тем или иным причинам не желает использовать стандартные средства для доступа к драйверам, а норовит использовать свои драйвера?

Игорь Коваль в своей книге исследовал работу классического резидентного (на int 13h) вируса под Windows, и обнаружил, что после загрузки системы вирус перестает быть активным. Точнее говоря, он становится псевдоактивным – с винчестером OS продолжает работать через него, но вот вызовы программ (автор уделил особое внимание дос-окошкам – Norton Commander) для работы с дисководами до него не доходят. Следовательно, вирус не может активизироваться для заражения дискеты, поскольку он отлавливает момент обращения именно к дисководам (скажем, сравнение номера диска в регистре dl <=1). Оказалось, что Windows, по словам автора, использует свой драйвер для работы с дискетами, а вот для жесткого диска "предпочитает" работу стандартными средствами …

Решение И. Коваля

Столкнувшись с вышеприведенной проблемой, И. Коваль пошел по пути использования перехвата сервиса DOS для работы с дисками, а именно – int 21h, функции 0eh (смена текущего диска). Алгоритм триггера активизации вируса в этом случае даже проще, чем в случае int 13h – достаточно следить за попытками выбрать через эту функцию дисковод.

Осталось решить, как перейти от момента загрузки компьютера к перехваченному прерыванию int 21h.

Общий принцип перехода заключался в активном ожидании загрузки DOS и перехвате int 21h после его появления.

Автор показал, что ожидающая программа не может использовать очевидные на первый взгляд прерывания int 1ch, int 08h и int09h. Дело в том, что Windows перестает использовать стандартные биос-обработчики этих прерываний в своей работе и восстанавливает вектор int 21h .

Решением оказалось использование вектора int 16h. Обработчик этого вектора "остается в живых" после загрузки, и, более того, вызывается во всех без исключения программах Windows – даже таких, как Word.

Схематично работу такого вируса можно представить следующим образом:
При первом получении управления в процессе загрузки mbr вирус перехватывает int 16h. Обработчик прерывания int 16h следит за вектором int 21h, вызывая его недокументированной функцией AX=0babch – контроль на перехват. Если вектор int 21h не отвечает стандартным образом, то считается, что необходимо выполнить перехват int 21h. Обработчик int 21h следит за сменой диска, контролируя функцию 0eh.

Автор утверждает, что такой вирус нормально работает в DOS окошках и осталяет "на будущее" реализацию работоспособной программы во всех приложениях Windows.

Оказалось – все так и есть, правда, остается непонятным детальный механизм перехвата int 21h с вектора int 16h. Вектор int 16h вызывается не при нажатии любой

клавиши, а только специальных (типа shift). Видимо, это делается для того, чтобы биос мог отследить состояние своих флажков.

Однако как же написать работоспособный под Windows вирус, который:

- Перехватывает прерывание int 13h при загрузке и использует только его;
- Работает во всех задачах (таких, как Word);
- Не использует особенностей OS (в данном случае – перехват чисто dos-ого прерывания int 21h, а является потенциально опасным для любых OS, работающих по тем или иным причинам с винчестером через BIOS).

Проблема с активностью вполне решается таким перехватом. Как уже было сказано, перехватчик int 13h вызывается при ВСЕХ операциях с винчестером. Остается поймать момент обращения к дисководу. На бессмысленность периодических запросов (а не засунули ли в дисковод чего-нибудь ?) указал сам И. Коваль.

Рассмотри процесс форматирования дискеты (неважно, из Windows приложения или из NC). OS должна записать на него свой новенький бут-сектор. А где она его может взять? Конечно, с винчестера! А кто винчестер контролирует?! Мы! Так чего же мы ждем: (Скажем, только что с винта был прочитан сектор, и он сейчас в es:[bx])

```
push es              ; Check for BOOT Signature...
cmp  word ptr es:[bx+01feh],0aa55h
jnz  NoBootRec
cmp  byte ptr es:[bx],0ebh
jnz  NoBootRec       ; Real BOOT Record ?!
mov  ah,02h          ; Try to Infect !!!
mov  ds:[si+8],ah    ; It works under Windows too )))
```

Такой алгоритм работает не только в случае форматирования, но также и в случае обращения к дисководу.

А теперь давайте рассмотрим довольно оригинальный вирус W2K.Stream под знаменитую Windows2000, с необычным алгоритмом реализации.

Вирус был создан в конце августа двумя хакерами из Чехии под кодовыми названиями Benny и Ratter. На данный момент случаев заражения вирусом зарегистрировано не было, однако его работоспособность и присутствие всех необходимых функций для существования в "диком виде" не вызывают сомнений.

"Данный вирус, несомненно, открывает новую страницу в истории создания компьютерных вирусов", - комментирует Евгений Касперский, руководитель антивирусных исследований компании, - "Технология внедрения в файлы при помощи замещения потоков делает процесс обнаружения и удаления вирусов исключительно сложным".

В отличие от существовавших ранее способов заражения файлов (добавление тела вируса в начало, конец или любое другое место тела программы), "Stream" использует возможность файловой системы NTFS (Windows NT/2000) поддерживать множественные потоки данных. Например, в файлах Windows 95/98 (FAT) внутренняя структура программы представлена лишь одним потоком – собственно ее телом.

В Windows NT/2000 (NTFS) таких потоков может быть много: как независимых программных модулей, так и разнообразной дополнительной служебной информации (права доступа к файлу, отметки о шифрации, времени обработки и т.д.). Это придает файлу гибкость, позволяя пользователям создавать новые потоки данных для хранения дополнительных атрибутов файлов или целых программ. "Stream" первым из известных вирусов использует возможность создавать множественные потоки данных NTFS для заражения файлов. Для этого он выполняет следующую последовательность действий.

- Сначала вирус создает дополнительный поток под именем "STR" и переносит туда оригинальное содержимое программы.
- После этого он записывает свое тело в основной поток вместо самой программы. Таким образом, при запуске зараженной программы сначала будет выполняться основной поток, содержащий вирус, который после окончания работы передаст управление "родительской" программе.

"По умолчанию, антивирусные программы проверяют только основной поток. В случае с данным вирусом проблем с его обнаружением и удалением ни у кого не возникнет", - продолжает Евгений Касперский, - "Однако, нет никаких гарантий, что вирусы не "переползут" в дополнительные потоки. В таком случае, большинству антивирусных компаний просто придется кардинально перестраивать их программы".

Технические детали

Это первый известный Windows-вирус, который при заражении файлов использует метод "stream companion". Этот метод основан на возможности файловой системы NTFS создавать дополнительные блоки данных ("потоки" данных - streams), которые ассоциированы с конкретным файлом.

На файловой системе NTFS каждый файл имеет как минимум один стандартный поток данных, к которому можно обратиться по имени файла. Каждый файл может также иметь дополнительные потоки данных, которые имеют свои персональные имена (filename:streamname). Стандартный поток в файле является собственно телом файла (в до-NTFS-ных терминах). Например, при запуске EXE-файла его код и данные считываются из стандартного потока; при открытии документа, его текст считывается также из стандартного потока. Дополнительные потоки данных в файлах могут иметь данные произвольного типа (например, данные о правах доступа к данному файлу). Они являются принадлежностью файла и жестко к нему привязаны. Потоки данных в файле не могут быть прочитаны и изменены без упоминания имени файла; при удалении файла удаляются все его потоки; при переименовании файла к его потокам затем можно обращаться только при помощи нового имени файла. Стандартные утилиты просмотра или редактирования потоков данных в поставке Windows отсутствуют. Для "ручного" просмотра можно использовать специализированные утилиты, например, FAR с необходимым набором "плагинов".

Работа вируса

Вирус является приложением Windows (PE EXE-файл). Упакован утилитой компрессии PE EXE-программ Petite. Имеет размер около 4K. При запуске заражает все файлы в текущем каталоге и возвращает управление программе-носителю. При заражении файла вирус переносит его тело (стандартный поток) в новый поток (который имеет имя STR) и затем записывает свой код в стандартный поток файла. Таким образом, тело зараженного файла (стандартный поток) оказывается замещенным кодом вируса, а первоначальное содержимое файла оказывается перемещенным в его поток. При запуске зараженного файла Windows считывает и выполняет его стандартный поток (т.е. код вируса). Windows также показывает у всех зараженных файлов одинаковую длину - длину файла-вируса (поскольку Windows сообщает длину только основного потока файла). Для того, чтобы возвратить управление первоначальному коду зараженного файла, вирус всего-лишь запускает на выполнение соответствующий поток файла, используя его имя: "FileName:STR".

В принципе, вирус работоспособен на любой операционной системе, использующей файловую систему NTFS. Однако его автор встроил в вирус проверку установленного ПО. Таким образом, запуск вируса происходит только в случае, если компьютер работает под управлением Windows 2000.

Вот так можно заражать файлы даже в такой защищённой операционной системе, как Windows 2000, не смотря на защищённый режим процессора, защиту памяти и прочие решения Microsoft-а.

При работе над данным эссе использовались следующие ресурсы:

1. http://www.microsoft.com/rus/windows2000/library/security/
2. http://www.viruslist.com/viruslist.asp?id=4351
3. http://neworder.box.sk/codebox.links.php?&key=trojgi
4. http://www.viruslist.com
5. http://wasm.ru
6. Энциклопедия компьютерных вирусов. Д. А. Козлов, А.А Парандовский.
7. http://www.citforum.ru
8. http://www.xakep.ru
9. http://security.nnov.ru