Министерство образования Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

В.В. Платонов

МЕТОДЫ ФУНКЦИОНИРОВАНИЯ КОМПЬЮТЕРНЫХ ВИРУСОВ И ЗАЩИТА ОТ НИХ

РУКОВОДСТВО К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

Санкт-Петербург
2003

Платонов В.В. Методы функционирования компьютерных вирусов и защита от них. Руководство к практическим занятиям. СПб.: Изд-во СПбГПУ, 2003. –42 с..

Данное пособие представляет собой руководство к практическим занятиям по дисциплине «Программно-аппаратные средства обеспечения информационной безопасности», изучаемой студентами кафедры «Информационная безопасность компьютерных систем» Санкт -Петербургского государственного политехнического университета.

Представлены методические рекомендации по изучению методов функционирования распространенных компьютерных вирусов и разработке методов защиты от них. Рассматриваются основные теоретические положения, необходимые для выполнения практических занятий.

Пособие может быть полезно при подготовке специалистов по направлению «Информационная безопасность».

Библиогр. 6 наим.

Печатается по решению редакционно-издательского совета Санкт-Петербургского государственного политехнического университета.

© Санкт-Петербургский государственный
политехнический университет, 2003
© В.В.Платонов, 2003

Введение

Данное пособие представляет собой руководство к практическим занятиям по дисциплине «Программно-аппаратные средства обеспечения информационной безопасности», изучаемой студентами кафедры «Информационная безопасность компьютерных систем» (ИБКС) Санкт-Петербургского государственного политехнического университета, и призвано помочь студентам в приобретении практических навыков организации защиты от компьютерных вирусов.

Основным и наиболее распространенным видом вредоносного программного обеспечения (ВПО) являются компьютерные вирусы. В настоящее время проблема компьютерных вирусов является, пожалуй, одной из самых острых в сфере защиты информации по причине необычайного многообразия самих вирусов и широкой масштабности распространения данного явления. По мере развития и усложнения компьютерных систем и программного обеспечения возрастает объем и повышается уязвимость хранящихся в них данных. Вместе с тем происходит и эволюция компьютерных вирусов, они начинают использовать вновь появившиеся технологии для своего распространения, вредоносных воздействий и маскировки. Появляются как модификации или аналоги уже существующих вирусов, так и совершенно новые версии, которые обладают совершенно новыми способами функционирования, что в значительной степени осложняет борьбу с ними.

Способы противодействия компьютерным вирусам можно разделить на профилактику вирусного заражения и уменьшение предполагаемого ущерба от такого заражения; обнаружение, обезвреживание и удаление вирусов. Наиболее эффективными средствами в борьбе и профилактике с компьютерными вирусами безусловно являются антивирусные программы. Но необходимо отметить, что, несмотря на огромные усилия конкурирующих между собой антивирусных фирм, не существует антивирусов, гарантирующих стопроцентную защиту от вирусов.

Цель данного цикла практических занятий – ознакомление с основными методами функционирования компьютерных вирусов и изучение механизмов их воздействий с целью освоения принципов диагностики состояния системы и своевременного противодействия при поражении вирусами. Цикл содержит сведения об основных типах вирусов, алгоритмах их работы, указания к проведению работ, а также сведения об организации и структуре компьютерной системы, необходимые для проведения работ.

Руководство включает в себя 6 работ, которые охватывают основные типы компьютерных вирусов.

При выполнении практических занятий целесообразно использовать следующую литературу:

1. Безруков Н.Н. Компьютерная вирусология. Справочное руководство – Москва, «Диалог», 1991.
2. Касперский Е.В. Компьютерные вирусы: что это такое и как с ними бороться. – Москва, «СК Пресс», 1998.
3. Карпов Б. VBA: специальный справочник. –Спб.: Питер, 2002.

4. Козлов Д.А., Парандовский А.А., Парандовский А.К. Энциклопедия компьютерных вирусов – Москва, «СОЛОН-Р», 2001.
5. Соломон Д., Русинович М. Внутреннее устройство Microsoft Windows 2000. Мастер-класс./Пер. с англ. –Спб.: Питер; М.: Издательско-торговый дом «Русская Редакция», 2001.
6. Финогенов К.Г. самоучитель по системным функциям MS-DOS. –М.: МП «МАЛИП», 1993.

Практическое занятие №1

Цель работы

Изучение принципов функционирования файловых вирусов. Ознакомление с организацией файловой системой MS DOS. На примере ознакомиться с принципами работы BAT-вируса.

1. Теоретические сведения

К данной группе относятся вирусы, которые при своем размножении тем или иным способом используют файловую систему какой-либо (или каких-либо) ОС.

Внедрение файлового вируса возможно практически во все исполняемые файлы всех популярных ОС. На сегодняшний день известны вирусы, поражающие все типы выполняемых объектов стандартной DOS: командные файлы (BAT), загружаемые драйверы (SYS) и выполняемые двоичные файлы (EXE, COM). Существуют вирусы, поражающие исполняемые файлы других операционных систем - Windows 3.x, Windows95/NT, OS/2, Macintosh, UNIX, включая VxD-драйвера Windows 3.x и Windows95.

Существуют вирусы, заражающие файлы, которые содержат исходные тексты программ, библиотечные или объектные модули. Возможна запись вируса также и в файлы данных (но это случается либо в результате ошибки вируса, либо при проявлении его агрессивных свойств).

1.1. Возможные стратегии заражения

По способу заражения файлов вирусы делятся на "overwriting", паразитические ("parasitic"), компаньон-вирусы ("companion"), "link"-вирусы, вирусы-черви и вирусы, заражающие объектные модули (OBJ), библиотеки компиляторов (LIB) и исходные тексты программ.

Overwriting

Данный метод заражения является наиболее простым: вирус записывает свой код вместо кода заражаемого файла, уничтожая его содержимое. Естественно, что при этом файл перестает работать и не восстанавливается. Такие вирусы очень быстро обнаруживают себя, так как операционная система и приложения довольно быстро перестают работать.

Parasitic

К паразитическим относятся все файловые вирусы, которые при распространении своих копий обязательно изменяют содержимое файлов, оставляя сими файлы при этом полностью или частично работоспособными. Основными типами таких вирусов являются вирусы, записывающиеся в начало файлов ("prepending"), в конец файлов ("appending") и в середину файлов ("inserting").

Известны два способа внедрения паразитического файлового вируса в начало файла: первый способ заключается в том, что вирус переписывает начало заражаемого файла в его конец, а сам копируется в освободившееся место; вторым способ - вирус создает в оперативной памяти свою копию, дописывает к ней заражаемый файл и сохраняет полученную конкатенацию на диск.

Наиболее распространенным способом внедрения вируса в файл является дописывание вируса в его конец. При этом вирус изменяет начало файла таким образом, что первыми выполняемыми командами программы, содержащейся в файле, являются команды вируса.

В свою очередь, внедрение вирусов в середину файлов происходит различными методами - путем переноса части файла в его конец или копирования своего кода в заведомо неиспользуемые данные файла ("cavity"-вирусы).

Компаньон-вирусы

К категории "компаньон" относятся вирусы, не изменяющие заражаемых файлов. Алгоритм работы этих вирусов состоит в том, что для заражаемого файла создается файл-двойник, причем при запуске зараженного файла управление получает именно этот двойник, т.е. вирус.

Некоторые компаньон-вирусы используют особенность DOS первым выполнять. COM-файл, если в одном каталоге присутствуют два файла с одним и тем же именем, но различными расширениями имени - .COM и .EXE. Другие при заражении переименовывают файл в какое-либо другое имя, запоминают его (для последующего запуска оригинала из зараженного файла-хозяина) и записывают свой код на диск под именем заражаемого файла. Еще один способ заражения - использовать особенности DOS PATH: либо записать свой код под именем заражаемого файла, но "выше" на один уровень PATH (DOS, таким образом, первым обнаружит и запустит файл-вирус), либо перенести файл-жертву на один подкаталог выше и т.д.

Файловые черви

Файловые черви (worms) при размножении всего лишь копируют свой код в какие-либо каталоги дисков в надежде, что эти новые копии будут когда-либо запущены пользователем. Иногда эти вирусы дают своим копиям "специальные" имена, чтобы подтолкнуть пользователя на запуск своей копии - например, INSTALL.EXE или WINSTART.BAT. Некоторые вирусы могут записывать команду запуска зараженного файла в BAT-файлы.

Link-вирусы

Link-вирусы, как и компаньон-вирусы не изменяют физического содержимого файлов, однако при запуске зараженного файла "заставляют" ОС выполнить свой код. Этой цели они достигают модификацией необходимых полей файловой системы.

OBJ-, LIB-вирусы и вирусы в исходных текстах

Вирусы, заражающие OBJ- и LIB-файлы, записывают в них свой код в формате объектного модуля или библиотеки. Зараженный файл, таким образом, не является

выполняемым и неспособен на дальнейшее распространение вируса в своем текущем состоянии. Носителем же "живого" вируса становится COM- или EXE-файл, получаемый в процессе линковки зараженного OBJ/LIB-файла с другими объектными модулями и библиотеками.

1.2. Алгоритм работы файлового вируса

В наиболее общем виде, получив управление, вирус совершает следующие действия:

- резидентный вирус проверяет оперативную память на наличие своей копии и инфицирует память компьютера, если копия вируса не найдена. Нерезидентный вирус 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, если они есть, дополнительные функции: деструктивные действия, графические или звуковые эффекты и т.д. (Дополнительные 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 (если она есть). Паразитические вирусы при этом либо а) лечат файл, выполняют его, а затем снова заражают, либо б) восстанавливает программу (но не файл) в исходном виде (например, у COM-программы восстанавливается несколько первых байт, у EXE-программы вычисляется истинный стартовый адрес, у драйвера восстанавливаются значения адресов программ стратегии и прерывания), компаньон-вирусы запускают на выполнение своего "хозяина", вирусы-черви и overwriting-вирусы возвращают управление DOS.

Метод восстановления программы в первоначальном виде зависит от способа заражения файла (для конкретного вируса список действий может быть дополнен, пункты могут поменяться местами и значительно расшириться).

1.3. Организация таблицы распределения файлов (FAT)

FAT представляет собой карту дискового пространства, распределяемого под файлы, и состоит из 12-битовых (дискеты) или 16-битовых элементов цифры для каждого имеющегося на диске кластера. Каждый элемент FAT является указателем на следующий занятый файлом кластер. С помощью FAT MS DOS отслеживает последовательность кластеров, принадлежавших файлу, независимо от фактического расположения их на диске. Для незанятых кластеров указатель имеет вид 000. Если кластер поврежден и MS DOS не может считать или записать в него информацию, то такой кластер содержит FF7 (признак сбойного кластера), и MS DOS не использует отмеченные таким образом кластеры при распределении.

Таблица распределения файлов, по сути, является односвязным списком, который DOS использует для отслеживания физического расположения данных на диске и для поиска свободной памяти для новых файлов. Слово по смещению 1Ah в элементе оглавления содержит номер первого кластера в цепочке распределения файла. Соответствующий элемент FAT либо указывает конец цепочки, либо ссылается на следующий элемент.

Повреждение FAT является тяжелой аварией, и при отсутствии резервных копий обычно ведет к значительной потере информации. В некоторых случаях поврежденной оказывается только первая из двух копий FAT. В этом случае можно переписать уцелевшую копию с помощью DiskEdit. Помимо стирания части или всей FAT, возможны мелкие ошибки типа сращивания цепочек кластеров (когда один и тот же кластер принадлежит двум файлам сразу) или зацикливания дерева каталогов. Восстановление облегчается при наличии резервной копии FAT, полученной утилитой IMAGE, которую рекомендуется включать в AUTOEXEC.BAT.

1.4. Корневой каталог

Информация о файлах и каталогах храниться в корневом каталоге. Корневой каталог хранит сведения обо всех содержащихся в нем файлах и подкаталогах. Он занимает семь секторов, начиная с сектора 5. Каждый элемент корневого каталога занимает 32 байта. Структура каталога показана на рис.1. Как видно из рисунка, имя файла состоит не более чем из восьми символов, дополненных справа пробелами. За именем следует факультативное трехсимвольное расширение. При длине менее трех символов оно также дополняется пробелами.

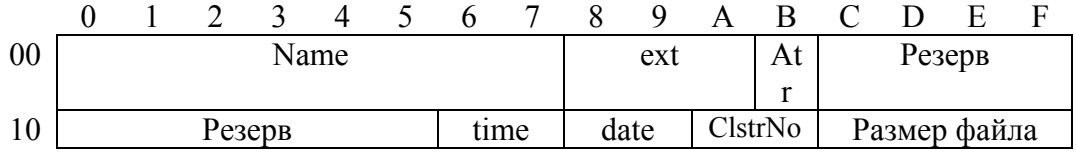

Рис.1. Структура элемента каталога

Границы полей отмечены вертикальными линиями. Используются следующие обозначения (в квадратных скобках указаны: смещение в 16-ричной системе счисления и длина):

name - имя файла [+0;8];
ext - расширение имени [+8;3];
atr - атрибут файла [+0Bh;1];
резерв [+0Ch;0Ah];
time - время создания [+16h;2];
date - дата создания [+18h;2];
ClstrNo- номер начального кластера данных (связь с FAT) [+1Ah;2];
размер файла - размер файла в байтах [+1Ch;4].

Поле атрибутов файла позволяет присваивать файлу определенный "статус" относительно тех или иных операций. Атрибут READ ONLY ("только чтение") запрещает выполнение операций удаления и модификации данного файла. Если

установлен первый бит (атрибут HIDDEN), то данный файл является так называемым скрытым файлом (не выдается командой DIR). Второй бит означает, что это системный файл, в отличие от пользовательских файлов. Используемые биты байта атрибутов показаны на рис.2.

		a	D	V	s	h	R

бит	*** маска ***
0:1 = только чтение	(READ ONLY) (01h)
1:1 = спрятанный	(HIDDEN) (02h)
2:1 = системный	(SYSTEM) (04h)
3:1 = метка тома	(08h)
4:1 = элемент подоглавления	(10h)
5:1 = копия файла НЕ архивировалась	(20h)

Рис.2. Формат поля атрибутов файла

Поля даты и времени приведены на рис.3.

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Час					Минута						Сек/2				

час (0-31) минута (0-63) 2-секундные единицы(0-31)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Год							Месяц				День				

год (0-127) месяц (0-15) день (0-31)

Рис.3. Формат полей даты и времени

Следует обратить внимание на то, что несовпадение систем представления часов, секунд и минут с двоичной системой приводит к наличию некоторого "запаса" в формате поля даты, что позволяет устанавливать "абсурдное" время типа 31 ч 63 мин 62 с. Поскольку ни одна из стандартных команд MS DOS не выдает количество секунд во времени образования файла, то значение "62 секунды" поля секунд может использоваться (и используется) вирусами для отметки зараженных файлов. Аналогично может использоваться поле даты. Год создания файла хранится в IBM PC в виде остатка после вычитания даты создания данного компьютера.

1.5. BAT-файлы

Пакетными или командными файлами (batch) называются текстовые файлы, содержимым которых являются команды MS-DOS (внешние и внутренние). Кроме команд MS-DOS и обращений к исполняемым программам, пакетные файлы могут содержать вызовы других пакетных файлов, специальные команды для управления выводом на экран, специальные команды для организации ветвлений, циклов и метки.

Важным свойством командных файлов является возможность использовать внутри них формальные параметры. При этом обращение к командному файлу имеет вид:

A> имя командного файла параметр1[параметр2 . . .]

Параметры, значения которых будут заданы при обращении к командному файлу, внутри файла будут иметь обозначения **%1**, **%2**, . . . , **%N,** (где N<10).

2. Порядок выполнения работы

1. Подготовка к проведению работы.
   1.1. Создать рабочую директорию, в которой должны находиться файлы:
- command.com
- autoexec.bat
- vir.bat
- Hiew.exe
- Hexedit.exe
- Dosmap.exe

   1.2. Установить значение переменной PATH на данную рабочую директорию.
2. Провести анализ вируса (vir.bat)
   2.1. С помощью текстового редактора изучить вирус.
   2.2.Построить алгоритм работы вируса, установив принципы его функционирования.
3. Проанализировать полученный алгоритм.
   3.1.Определить возможные угрозы, которые могут возникнуть при исполнении данного вируса.
   3.2.Определить возможные варианты обнаружения данного вируса.
   3.3.Предложить варианты защиты от данного вируса.
   3.4.Разработать предложения по обнаружению и защите от вирусов данного класса.

3. Содержание отчета

В отчете необходимо привести:

- алгоритм работы вируса;
- возможные угрозы;
- варианты обнаружения вируса;
- предложения по обнаружению и защите от данного класса вирусов.

4. Контрольные вопросы

1. Алгоритм работы файлового вируса.
2. Возможные стратегии заражения файлов.
3. Структура FAT.

Практическое занятие №2

Цель работы

Ознакомление со структурой исполняемых файлов типа COM и EXE и понятием резидентной программы. Анализ работы файлового вируса.

1. Теоретические сведения

Для MS-DOS существует два основных формата исполняемых программ - COM и EXE.

1.1. COM -программы

Файл COM-формата - это двоичный образ кода и данных программы. Такой файл должен занимать менее 64K. COM-программы содержат единственный сегмент (или, по крайней мере, не содержат явных ссылок на другие сегменты), и при их загрузке не требуется настройка сегментов.

При загрузке и выполнении программного COM-файла [3] сначала инициализируется блок среды (Program Segment Prefix - PSP), выбирая информацию либо из текущей среды системы (случай, принимаемый по умолчанию), либо из среды, указанной порождающим процессом. После установки среды DOS распределяет блок памяти для программы (для COM-программы этот блок занимает всю свободную память). Минимальный размер памяти при этом равен размеру COM-файла плюс память для PSP. Далее COM-программа помещается в память, начиная с PSP:0100 (непосредственно за PSP по смещению 100h).

COM-программы обычно используются для небольших утилит. Они быстрее загружаются, т.к. не требуется настройка сегментов, и занимают меньше места на диске, поскольку заголовок и сегмент стека отсутствуют в загрузочном модуле.

После загрузки COM-файла:

- CS, DS, ES и SS указывают на PSP;
- SP указывает на конец сегмента PSP (обычно 0FFFEh, но может быть и меньше, если полный 64K сегмент недоступен, например, при работе отладчика). Слово по смещению 06h в PSP указывает, какая часть программного сегмента доступна;
- вся память системы за программным сегментом распределена программе;
- слово 00h помещено в стек (командой PUSH);
- IP содержит 100h (первый байт модуля) в результате команды JMP PSP:100.

1.2. EXE - программы

Файл EXE-формата, в отличие от COM-файла, содержит также и информацию для настройки сегментов программы. Она располагается в специальном заголовке, при помощи которого загрузчик выполняет настройку ссылок на сегменты в загруженном модуле.

EXE-программы содержат несколько программных сегментов, включая сегмент кода CS, сегмент данных DS и сегмент стека SS.

Заголовок состоит из двух частей: форматированной части (первые 1Ch байт) и таблицы перемещений (Relocation Table) переменного размера. Форматированная часть заголовка включает 14 двухбайтовых полей – рис. 4.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00h	Sign		PartPag		PageCnt		ReloCn		HdrSize		MinMem		MaxMem		ReloSS	
10h	ExeSP		hkSum		ExeIP		ReloCS		TablOff		Overlay		Rel.Table El			
20h	Смещ. Сегмент				смещ. Сегмент				... ...				... ...			
30h																
...	... ...															
1F0h	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
200h	Начало кода EXE-программы															

Рис.4. Структура заголовка файла EXE-формата

Sign - "подпись" EXE-файла (4D5Ah -- 'MZ') [0;2];

PartPag - длина неполной последней страницы (обычно игнорируется) [2;2];

PageCnt - длина образа загружаемой программы в 512-байтовых страницах, включая заголовок [4;2];

ReloCnt - число элементов в таблице перемещений [6;2];

HdrSize - длина заголовка в 16-байтовых параграфах [8;2];

MinMem - минимум требуемой памяти за концом программы (параграфы) [0Ah;2];

MaxMem - максимум требуемой памяти за концом программы (параграфы) [0Ch;2];

ReloSS - сегментное смещение сегмента стека (для установки SS) [0Eh;2];

ExeSP - значение регистра SP (указателя стека) при запуске [10h;2];

ChkSum - контрольная сумма (отрицательная сумма всех слов файла; в существующих версия MS DOS, к сожалению, при загрузке не проверяется) [12h;2];

ExeIP - значение регистра IP (указателя команд) при запуске [14h;2];

ReloCS - сегментное смещение кодового сегмента (для установки CS) [16h;2];

TablOff - смещение для 1-го элемента перемещения (часто 1Ch) [18h;2];

Overlay - номер оверлея (0 для главного модуля) [1Ah;2].

Как видно из приведенной структуры, заголовок EXE-файла всегда начинается с подписи MZ или, реже, ZM. После загрузки программы загрузчик анализирует эти первые два байта и, если они содержат "MZ", то рассматривает данный файл как EXE-файл, независимо от расширения имени файла (оно может быть и COM и BIN, а не только EXE). Эта особенность MS DOS учитывается не

всеми файловыми вирусами. Поэтому, если при заражении вирус исходит из предположения, что тип файла соответствует расширению его имени, то он неправильно заражает "замаскированные" EXE-файлы, которые после заражения, таким образом, становятся не работоспобными.

EXE-файл загружается, начиная с адреса PSP:0100. В процессе загрузки считывается информация заголовка EXE в начале файла и выполняется перемещение адресов сегментов. После перемещения управление передается загрузочному модулю посредством инструкции далекого перехода (FAR JMP) к адресу CS:IP, извлеченному из заголовка EXE.

В момент получения управления программой EXE-формата:

- DS и ES указывают на PSP;
- CS, IP, SS и SP инициализированы значениями, указанными в заголовке EXE;
- поле PSP MemTop содержит значение, указанное в заголовке EXE. Обычно вся доступная память распределена программе.

1.3. Резидентные программы

Запускаемые на выполнение программы делятся на резидентные и нерезидентные. Резидентная программа по окончании оставляет свой код или часть кода в оперативной памяти, при этом DOS резервирует необходимый для ее работы участок памяти. Затем резидентная программа работает параллельно другим программам, некоторые из резидентных программ могут быть выгружены из памяти. Доступ к резидентной программе осуществляется либо через подмену прерываний, либо непосредственной адресацией. Нерезидентная программа при завершении не оставляет в памяти своего кода, а занимаемая ею память освобождается.

Выход из программы в MS DOS можно выполнить двумя способами:

1. С помощью функции 4Ch (EXIT) в любой момент, независимо от значений регистров.

2. С помощью функции 00h или прерывания 20h, когда CS указывает на PSP.

В этих случаях занимаемая программой память освобождается. Кроме того, MS DOS позволяет завершить программу и оставить ее постоянно резидентной (TSR), используя либо прерывание 27h, либо функцию DOS 31h (KEEP). Последний способ применяется, когда резидентный код длиннее 64K или необходимо сформировать код выхода для родительского процесса.

Концепция TSR-программы используется большинством файловых и всеми бутовыми вирусами для того, чтобы контролировать выполняемые операционной системой операции и заражать выполняемые или открываемые на чтение программы.

2. Порядок выполнения работы

1. Подготовка к проведению работы.

1.1. Создать рабочую директорию, в которой должны находиться файлы:

- command.com

- autoexec.bat
- vir.com
- Hiew.exe
- Файлы-мишени (исполняемые файлы различных расширений и различной длины)

1.2. Установить значение переменной PATH на данную рабочую директорию.

2. Заражение файлов-мишеней.

2.1. Запустить vir.com.

2.2. Запускать на исполнение файлы-мишени и контролировать их заражение.

3. Провести анализ вируса.

3.1. С помощью системных утилит определить:

- Типы заражаемых файлов заражаемых данным вирусом.
- Минимальные длины заражаемых исполняемых файлов.
- Наличие повторных заражений файла вирусом.

3.2. Классифицировать данный вирус.

4. Определить возможные угрозы, которые могут возникнуть при исполнении данного вируса и способы борьбы с ними.

4.1. Определить «сигнатуры» для обнаружения данного вируса.

4.2. Предложить варианты защиты от данного вируса.

4.3. Разработать предложения по обнаружению и защите от вирусов данного класса.

3. Содержание отчета

В отчете необходимо привести:

- результаты наблюдений при проведении заражения;
- типы заражаемых файлов;
- наличие повторного заражения;
- классификация данного вируса;
- возможные угрозы;
- варианты обнаружения вируса, «сигнатура» для его обнаружения;
- предложения по обнаружению и защите от данного класса вирусов.

4. Контрольные вопросы

1. Структура COM файла.
2. Структура EXE файла.
3. Основные отличия форматов COM и EXE.

Практическое занятие №3

1. Теоретические сведения

Загрузочные вирусы заражают загрузочный (boot) сектор флоппи-диска и boot-сектор или Master Boot Record (MBR) винчестера. Принцип действия загрузочных вирусов основан на алгоритмах запуска операционной системы при включении или перезагрузке компьютера - после необходимых тестов установленного оборудования (памяти, дисков и т.д.) программа системной загрузки считывает первый физический сектор загрузочного диска A:, C: или CD-ROM в зависимости от параметров, установленных в BIOS Setup) и передает на него управление.

В случае дискеты или компакт-диска управление получает boot-сектор, который анализирует таблицу параметров диска (BPB - BIOS Parameter Block) высчитывает адреса системных файлов операционной системы, считывает их в память и запускает на выполнение. Системными файлами обычно являются MSDOS.SYS и IO.SYS, либо IBMDOS.COM и IBMBIO.COM, либо других в зависимости от установленной версии DOS, Windows или других операционных систем. Если же на загрузочном диске отсутствуют файлы операционной системы, программа, расположенная в boot-секторе диска выдает сообщение об ошибке и предлагает заменить загрузочный диск.

В случае винчестера управление получает программа, расположенная в MBR винчестера. Эта программа анализирует таблицу разбиения диска (Disk Partition Table), вычисляет адрес активного boot-сектора (обычно этим сектором является boot-сектор диска C:), загружает его в память и передает на него управление. Получив управление, активный boot-сектор винчестера проделывает те же действия, что и boot-сектор дискеты.

При заражении дисков загрузочные вирусы "подставляют" свой код вместо какой-либо программы, получающей управление при загрузке системы. Принцип заражения, таким образом, одинаков во всех описанных выше способах: вирус "заставляет" систему при ее перезапуске считать в память и отдать управление не оригинальному коду загрузчика, но коду вируса.

Заражение дискет производится единственным известным способом - вирус записывает свой код вместо оригинального кода boot-сектора дискеты. Винчестер заражается тремя возможными способами - вирус записывается либо вместо кода MBR, либо вместо кода boot-сектора загрузочного диска (обычно диска C:), либо модифицирует адрес активного boot-сектора в Disk Partition Table, расположенной в 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).

Существует несколько вариантов 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). Возможен и другой способ - размещение первоначального загрузочного сектора в неиспользуемом или редко используемом секторе, например, в одном из секторов винчестера (если такие есть), расположенных между MBR и первым boot-сектором, а на дискете такой сектор выбирается из последних секторов корневого каталога. Некоторые вирусы записывают свой код в последние сектора винчестера, поскольку эти сектора используются только тогда, когда винчестер полностью заполнен информацией (что является довольно редким явлением, если учесть размеры современных дисков).

При заражении большинство вирусов копирует в код своего загрузчика системную информацию, хранящуюся в первоначальном загрузчике (для MBR этой информацией является Disk Partition Table, для Boot-сектора дискет - BIOS Parameter Block). В противном случае система окажется неспособной загрузить себя, поскольку дисковые адреса компонент системы высчитываются на основе этой информации. Такие вирусы довольно легко удаляются переписыванием заново кода системного загрузчика в boot-секторе и MBR - для этого необходимо загрузиться с незараженной системной дискеты и использовать команды SYS для обезвреживания дискет и логических дисков винчестера или FDISK/MBR для лечения зараженного MBR-сектора.

1.1. Алгоритм заражения резидентных вирусов

Практически все загрузочные вирусы резидентные. Они внедряются в память компьютера при загрузке с инфицированного диска. При этом системный загрузчик считывает содержимое первого сектора диска, с которого производится загрузка, помещает считанную информацию в память и передает на нее (т.е. на вирус) управление. После этого начинают выполняться инструкции вируса, который:

- как правило, уменьшает объем свободной памяти (слово по адресу 0040:0013), копируют в освободившееся место свой код и считывает с диска свое продолжение (если оно есть). В дальнейшем некоторые вирусы "ждут" загрузки DOS и восстанавливают это слово в его первоначальное значение. В результате они оказываются расположенными не за пределами DOS, а как отдельные блоки DOS-памяти;
- перехватывает необходимые вектора прерываний (обычно - INT 13H), считывает в память оригинальный boot-сектор и передает на него управление.

В дальнейшем загрузочный вирус ведет себя подобно резидентному файловому: перехватывает обращения операционной системы к дискам и

инфицирует их, в зависимости от некоторых условий совершает деструктивные действия или вызывает звуковые или видеоэффекты.

Существуют нерезидентные загрузочные вирусы - при загрузке они заражают MBR винчестера и дискеты, если те присутствуют в дисководах. Затем такие вирусы передают управление оригинальному загрузчику и на работу компьютера более не влияют.

1.2. Структура управляющей информации на дискете

Первые 12 секторов (0- 11) содержат три управляющих таблицы: загрузчик (BOOT), таблицу распределения файлов (FAT) и корневой каталог (root directory). Схема расположения этих секторов приведена на рис.5.

Рис.5. Схема расположения секторов на дискете

Первый сектор дискеты или раздела винчестера называется загрузочным (бут) сектором и содержит сведения о формате дискеты, а также короткую программу - загрузчик. Структура бут сектора показана на рис.6.

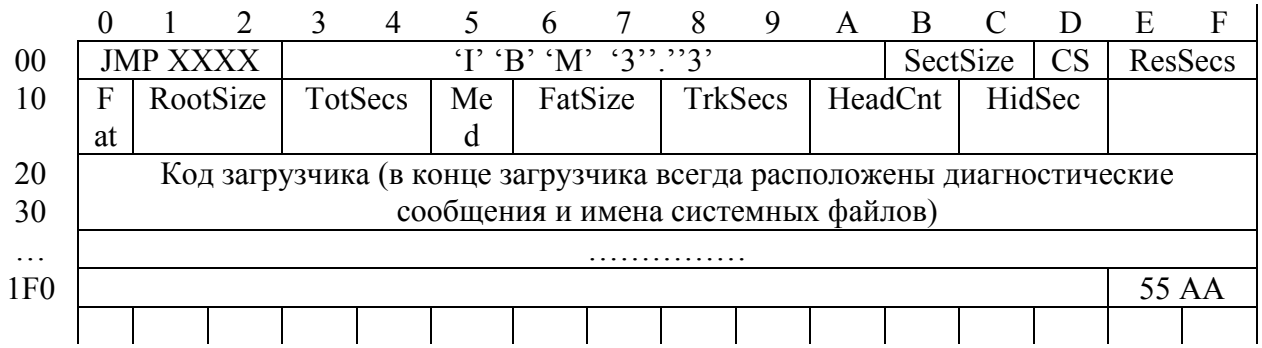

Рис.6. Структура загрузочного сектора

Используемые обозначения:

JMP - NEAR-переход на начало загрузчика; 8 байтов, содержащих произвольную информацию, обычно заносимую программой форматирования диска;
SectSize - количество байтов в одном секторе;
CS(ClustSize) - количество секторов в одном кластере;
ResSecs- количество резервных секторов (перед FAT);
Fat (FatCnt) - количество FAT;
RootSize – максимальное число 32-байтовых элементов корневого каталога;
TotSecs - общее число секторов на носителе (разделе диска);
Med (Media)- дескриптор носителя (то же, что 1-й байт FAT);
FatSize - количество секторов в одной FAT;
TrkSecs - количество секторов в одном треке;
HeadCnt - число головок чтения/записи (поверхностей);
HidnSec- количество спрятанных секторов;
далее - размер форматированной порции корневого сектора.

Программа-загрузчик находит на диске и загружает в оперативную память системный файл IO.SYS (или IBMBIO.SYS). Если файл не найден, то выдается сообщение "Non-System disk or disk error. Replace and strike any key when ready" ("Несистемный диск или ошибка при считывании. Замените и нажмите любую клавишу") и процесс загрузки прекращается. Эта ситуация обычно возникает, когда на компьютере с винчестером пытаются перезагрузиться при установленной и защелкнутой дискете в дисководе А. Если дискета заражена бутовым вирусом, то при выполнении этой рекомендации винчестер также будет заражен.

Второй компонентой бут сектора является таблица параметров. Она содержит важную информацию о структуре дискеты или логического диска (положение и размер FAT, количество секторов в кластере и др.). Эта структура устанавливается при разметке и является статической, т.е. в дальнейшем не меняется. Если таблица параметров испорчена, то информацию с диска прочитать нельзя и при загрузке система зависает.

Бут-сектор винчестера отличается только таблицей параметров. Программа-загрузчик остается неизменной для данной версии ОС. Другие версии операционной системы имеют несколько отличающийся от приведенного загрузчик, однако, его структура неизменна: в конце сектора всегда имеется текст диагностического сообщения и имена системных файлов (например, IBMBIO.COM и IBMDOS.COM).

В конце бут-сектор всегда содержит два идентификационных байта: 55h AAh.

Следует обратить внимание, что первые три байта бут сектора MS DOS стандартны для конкретное версии. Несовпадение этих байтов при просмотре бут сектора должно сразу настораживать, т.к. может свидетельствовать о заражении вирусом.

1.3. Таблица разделов диска - Partition Table (винчестер)

MS DOS различает два типа дисков: физические и логические. Физический диск - это установленное устройство (винчестер). Этот физический диск может быть разделен на несколько логических дисков (разделов). Сведения о положении логических дисков на физическом содержатся в специальной таблице, обычно называемой Partition Table (таблица разделов), которая является составной часть так называемого главного бут сектора винчестера - Master Boot Record (MBR). Название "главный бут-сектор" (буквальный перевод термина Master Boot Record - главная загрузочная запись) связано с тем, что MBR первым загружается в оперативную память, если загрузка операционной системы выполняется с винчестера.

MBR всегда занимает первый сектор винчестера (цилиндр 0, сторона 0, сектор 1). Как видно из рис.7. и рис.8., он состоит из выполняемого кода и таблицы разделов. Последняя расположена в конце сектора и состоит из четырех 16-байтовых элементов. Информация в таблицу разделов заносится утилитой Fdisk или аналогичной ей несистемной утилитой.

	0 1 2 3 4 5 6 7 8 9 A B C D E F	
00	начало кода начальной загрузки MBR	
10 …	…	
1B0	конец кода начальной загрузки MBR	Первый
1C0	Раздел винчестера	Второй
1D0	Раздел винчестера	Третий
1E0	Раздел винчестера	Четвертый
1F0	Раздел винчестера	55 AA

Рис.7. Структура таблицы разделов винчестера

10								Bot	HdS
20	Sec Cyl	Sys HdE	Sec Cyl	младш.	старш.	младш.	старш		

Рис.8. Структура элемента раздела

(поля показаны в тех положениях, в которых они расположены в дампе).
Bot - флаг загрузки (0 - не загружаемый, 80h - загружаемый) [+0,1];
HdS - начало раздела: номер головки [+1,1];
<Sec> <Cyl> - начало раздела: сектор/цилиндр бут сектора[+2,2];

Sys - код системы: 0 - неизв., 1 - DOS 12-бит FAT, 4 - 16-бит[+4,1];
HdE - конец раздела: номер головки [+5,1];
<Sec> <Cyl> - конец раздела: сектор/цилиндр последнего сектора[+6,2];
<младш> <старш> - относительный номер начального сектора[+8,4];
<младш> <старш> - размер (число секторов)[+0Ch,4];
далее (со смещением 10h) следует начало следующего элемента раздела (или 0AA55h для последнего элемента). В таблице разделов используется абсолютная адресация.

Значения цилиндра и сектора занимают 10 и 6 бит соответственно.

Структура элемента раздела показана на рис. 9.

F	E	D	C	B	A	9	8	7	6		5	4	3	2	1	0
c	c	c	c	c	c	c	c	C	C		s	s	s	s	s	s

Рис.9. Структура элемента раздела

Биты, обозначенные большими буквами C, являются старшими, т.е. при дешифровке номера цилиндра они приписываются слева, а не справа.

Наличие таблицы разделов позволяет не только иметь несколько логических дисков, но и несколько операционных систем на одном диске.

Во время загрузки ROM-BIOS загружает MBR и передает управление на ее код. Этот код считывает таблицу разделов, чтобы определить раздел, помеченный как загружаемый (Bootable). Затем в память считывается бут-сектор из этого раздела и ему передается управление. Отсюда следует, что, заражая винчестер, вирус может заменять не только бут-сектор, но и MBR.

Повреждение MBR ведет к тому, что логические диски становятся недоступными. В результате система зависает при загрузке. Для восстановления MBR рекомендуется иметь резервную копию первых секторов каждого логического диска на специальной дискете.

Восстановление таблицы в случае тяжелых повреждений и отсутствия эталона можно провести вручную, пользуясь шаблоном, приведенным на рис.7,8.

Значение "относительного сектора" по смещению 8 в каждом разделе эквивалентно цилиндру, дорожке и сектору начального адреса раздела. Относительный сектор 0 совпадает с цилиндром 0, дорожкой 0, сектором 1. Относительный номер сектора увеличивается сначала по каждому сектору на головке, затем по каждой дорожке и, наконец, по каждому цилиндру.

Применима формула:

отн_сек = (#Цил * сек_на_дор * дор_на_цил) + (#Гол * сек_на_дор) + (#Сек-1)

Разделы винчестера обычно начинаются с четного цилиндра, за исключением первого раздела, который иногда размещается с цилиндра 0, дорожки 0, сектора 2 (поскольку сектор 1 - это главный бут-сектор), а чаще всего начинается со следующей дорожки, оставляя остальные сектора нулевой дорожки пустыми.

Последнее обстоятельство используется рядом бутовых вирусов для хранения там своего хвоста, поэтому содержимое указанных секторов рекомендуется периодически просматривать.

Следует отметить, что помимо бутовых вирусов в указанных секторах могут хранить информацию другие программы, например Advanced Disk Manager. В этом случае заражение винчестера бутовым вирусом, использующим те же сектора, ведет к потере соответствующей информации. Поэтому всегда необходимо иметь резервную копию их содержимого на специальной дискете. На этой же дискете следует хранить копии MBR, бут секторов всех логических дисков, копию CMOS и другую полезную для восстановления системных блоков информацию.

2. Порядок выполнения работы

1. Подготовка к проведению работы.
    1.1. Создать рабочую директорию, в которой должны находиться файлы:
        - command.com
        - autoexec.bat
        - утилиты работы с дисками
        - Hiew.exe
        - Файлы-мишени (исполняемые файлы различных расширений и различной длины)

    1.2. Установить значение переменной PATH на данную рабочую директорию.
    1.3. Проанализировать boot-сектор обычной дискеты.
2. Заражение файлов-мишеней.
    2.1. Вставить «зараженную» дискету. Сделать попытку загрузки с нее.
    2.2. Проанализировать содержимое оперативной памяти.
    2.2. Запускать на исполнение файлы-мишени и контролировать их заражение.
4. Провести запись файла на чистую дискету.
    4.1. После перезагрузки проанализировать boot-сектор дискеты.
    4.2. определить расположение вируса и методы маскировки.
5. Провести анализ вируса.
    5.1. С помощью системных утилит определить:
        - Типы заражаемых файлов заражаемых данным вирусом.
        - Минимальные длины заражаемых исполняемых файлов.
        - Наличие повторных заражений файла вирусом.

    5.2.Классифицировать данный вирус.
6. Определить возможные угрозы, которые могут возникнуть при исполнении данного вируса и способы борьбы с ними.
    6.1. Определить «сигнатуры» для обнаружения данного вируса.
    6.2. Предложить варианты защиты от данного вируса.
    6.3. Разработать предложения по обнаружению и защите от вирусов данного класса.

3. Содержимое отчета

В отчете необходимо привести:

- результаты наблюдений при проведении заражения;
- типы заражаемых файлов;
- особенности заражения дискет и методы маскировки;
- классификацию данного вируса;
- возможные угрозы;
- варианты обнаружения вируса, «сигнатуры» для его обнаружения;
- предложения по обнаружению и защите от данного класса вирусов.

4. Контрольные вопросы

1. Алгоритм заражения загрузочного вируса.
2. Возможные стратегии заражения.
3. Структура и назначение BOOT сектора.
4. Отличие структуры дискет от винчестера.

Практическое занятие №4

1. Теоретические сведения

Поскольку резкой границы между файловыми и бутовыми вирусами нет, некоторые вирусы являются смешанными и заражают как файлы, так и бут-сектор или MBR. Такие "гибридные" вирусы имеют более высокую инфицирующую способность, чем "чистые" типы, однако выявляются легче, чем файловые вирусы, поскольку имеют фиксированное место заражения на винчестере, целостность которого обычно контролируется специальной программой.

Помимо зараженных исполняемых файлов, они могут переноситься на дискетах, содержащих только файлы данных, как обычные бутовые вирусы. Кроме того, загружаясь из MBR, легче обойти резидентные средства контроля, поскольку инсталляция вируса проходит на "чистой" машине.

2. Порядок выполнения работы

1. Подготовка к проведению работы.
   1.1. Создать рабочую директорию, в которой должны находиться файлы:
   - command.com
   - autoexec.bat
   - утилиты работы с дисками
   - Hiew.exe
   - Файлы-мишени (исполняемые файлы различных расширений и различной длины)

   1.2. Установить значение переменной PATH на данную рабочую директорию.
   1.3. Проанализировать boot-сектор обычной дискеты.
2. Заражение файлов-мишеней.
   2.1. Вставить «зараженную» дискету. Сделать попытку загрузки с нее.
   2.2. Проанализировать содержимое оперативной памяти.
   2.2. Запускать на исполнение файлы-мишени и контролировать их заражение.
4. Провести запись файла на чистую дискету.
   4.1. После перезагрузки проанализировать boot-сектор дискеты.
   4.2. определить расположение вируса и методы маскировки.
5. Провести анализ вируса.
   5.1. С помощью системных утилит определить:
   - Типы заражаемых файлов заражаемых данным вирусом.
   - Минимальные длины заражаемых исполняемых файлов.
   - Наличие повторных заражений файла вирусом.

   5.3.Классифицировать данный вирус.
6. Определить возможные угрозы, которые могут возникнуть при исполнении данного вируса и способы борьбы с ними.

6.1. Определить «сигнатуры» для обнаружения данного вируса.
6.4. Предложить варианты защиты от данного вируса.
6.5. Разработать предложения по обнаружению и защите от вирусов данного класса.

3. Содержимое отчета

В отчете необходимо привести:

- результаты наблюдений при проведении заражения;
- типы заражаемых файлов;
- особенности заражения дискет и методы маскировки;
- классификацию данного вируса;
- возможные угрозы;
- варианты обнаружения вируса, «сигнатуры» для его обнаружения;
- предложения по обнаружению и защите от данного класса вирусов.

4. Контрольные вопросы

1. Свойства файловых вирусов, которые могут включать в себя файлово-загрузочные вирусы.
2. Свойства загрузочных вирусов, которые могут включать в себя файлово-загрузочные вирусы.

Практическое задание № 5

Цель работы

Цель работы состоит в ознакомлении с основными механизмами маскировки полиформик и stealth-вирусов.

1. Теоретические сведения

1.1. Stealth-вирусы

Stealth-вирусы (вирусы-невидимки) представляют собой программы, которые перехватывают обращения DOS к пораженным файлам или секторам дисков и либо временно лечат их, либо "подставляют" вместо себя незараженные участки информации. Кроме этого такие вирусы при обращении к файлам используют достаточно оригинальные алгоритмы, позволяющие "обманывать" резидентные антивирусные мониторы. Использование Stealth-алгоритмов позволяет вирусам полностью или частично скрыть себя в системе.

Загрузочные вирусы

Загрузочные стелс-вирусы для скрытия своего кода используют два основных способа. Первый из них заключается в том, что вирус перехватывает команды чтения зараженного сектора (INT 13h) и подставляет вместо него незараженный оригинал. Этот способ делает вирус невидимым для любой DOS-программы. Возможен перехват команд чтения секторов на уровне более низком, чем INT 13h.

Второй способ направлен против антивирусов, поддерживающих команды прямого чтения секторов через порты контроллера диска. Такие вирусы при запуске любой программы (включая антивирус) восстанавливают зараженные сектора, а после окончания ее работы снова заражают диск. Поскольку для этого вирусу приходится перехватывать запуск и окончание работы программ, то он должен перехватывать также DOS-прерывание INT 21h.

С некоторыми оговорками стелс-вирусами можно назвать вирусы, которые вносят минимальные изменения в заражаемый сектор (например, при заражении MBR правят только активный адрес загрузочного сектора - изменению подлежат только 3 байта), либо маскируются под код стандартного загрузчика.

Файловые вирусы

Большинство файловых стелс вирусов использует те же приемы, что приведены выше: они либо перехватывают DOS-вызовы обращения к файлам (INT 21h) либо временно лечат файл при его открытии и заражают при закрытии. Также как и для загрузочных вирусов, существуют файловые вирусы, использующие для

своих стелс-функций перехват прерываний более низкого уровня - вызовы драйверов DOS, INT 25h и даже INT 13h.

Полноценные файловые стелс-вирусы, использующие первый способ скрытия своего кода, в большинстве своем достаточно громоздки, поскольку им приходиться перехватывать большое количество DOS-функций работы с файлами: открытие/закрытие, чтение/запись, поиск, запуск, переименование и т.д., причем необходимо поддерживать оба варианта некоторых вызовов (FCB/ASCII).

Некоторые вирусы используют часть функций полноценного стелс-вируса. Чаще всего они перехватывают функции DOS FindFirst и FindNext (INT 21h, AH=11h, 12h, 4Eh, 4Fh) и "уменьшают" размер зараженных файлов. Такой вирус невозможно определить по изменению размеров файлов, если он резидентно находится в памяти. Программы, которые не используют указанные функции DOS, а напрямую используют содержимое секторов, хранящих каталог, показывают правильную длину зараженных файлов.

Макро-вирусы

Реализация стелс-алгоритмов в макро-вирусах является, наверное, наиболее простой задачей - достаточно всего лишь запретить вызов меню File/Templates или Tools/Macro. Достигается это либо удалением этих пунктов меню из списка, либо их подменой на макросы FileTemplates и ToolsMacro.

1.2. Полиморфик-вирусы

К полиморфик-вирусам относятся те из них, детектирование которых невозможно (или крайне затруднительно) осуществить при помощи так называемых вирусных масок - участков постоянного кода, специфичных для конкретного вируса. Достигается это двумя основными способами - шифрованием основного кода вируса с непостоянным ключом и случайным набором команд расшифровщика или изменением самого выполняемого кода вируса.

Наиболее часто второй способ полиморфизма (изменением выполняемого кода) используется макро-вирусами, которые при создании своих новых копий случайным образом меняют имена своих переменных, вставляют пустые строки или меняют свой код каким-либо иным способом. Таким образом, алгоритм работы вируса остается без изменений, но код вируса практически полностью меняется от заражения к заражению.

Реже этот способ применяется сложными загрузочными вирусами. Такие вирусы внедряют в загрузочные сектора лишь достаточно короткую процедуру, которая считывает и диска основной код вируса и передает на него управление. Код этой процедуры выбирается из нескольких различных вариантов (которые также могут быть разбавлены "пустыми" командами), команды переставляются между собой и т.д. Еще реже этот прием встречается у файловых вирусов - ведь им приходится полностью менять свой код, а для этого требуются достаточно сложные алгоритмы.

Полиморфизм различной степени сложности [2] встречается в вирусах всех типов - от загрузочных и файловых DOS-вирусов до Windows-вирусов и даже макро-вирусов.

2. Порядок выполнения работы

1. Подготовка к проведению работы.
   1.1. Создать рабочую директорию, в которой должны находиться файлы:
   - command.com
   - autoexec.bat
   - svir.com
   - Hiew.exe
   - Файлы-мишени (исполняемые файлы различных расширений и различной длины)

   1.2. Установить значение переменной PATH на данную рабочую директорию.
2. Заражение файлов мишеней.
   2.1. Запустить svir.com
   2.2. Запускать на исполнение файлы-мишени и контролировать их заражение.
   2.3. При отсутствии внешних признаков заражения выполнить перезагрузку.
   2.4. Установить факт заражения мишеней.
3. Провести анализ вируса.
   3.1. С помощью системных утилит определить:
   - Типы заражаемых файлов заражаемых данным вирусом.
   - Минимальные длины заражаемых исполняемых файлов.
   - Наличие повторных заражений файла вирусом.

   3.2.Классифицировать данный вирус.
4. Поиск вариантов обнаружения и защиты.
   4.1.Определить возможные угрозы, которые могут возникнуть при исполнении данного вируса и способы борьбы с ними.
   4.2. Определить «сигнатуры» для обнаружения данного вируса.
   4.3. Предложить варианты защиты от данного вируса.
   4.4.Разработать предложения по обнаружению и защите от вирусов данного класса.

3. Содержание отчета

В отчете необходимо привести:

- результаты наблюдений при проведении заражения;
- типы заражаемых файлов;
- наличие повторного заражения;
- классификация данного вируса;
- возможные угрозы;

- варианты обнаружения вируса, «сигнатура» для его обнаружения;
- предложения по обнаружению и защите от данного класса вирусов.

4. Контрольные вопросы

1. Возможные стратегии сокрытия вирусами своего присутствия.
2. Уровни полиморфизма и возможности обнаружения вируса.

Практическое занятие № 6

Цель работы

Цель работы состоит в ознакомлении с принципами построения и функционирования макровирусов.

1. Теоретические сведения

Макро-вирусы (macro viruses) являются программами на языках (макроязыках), встроенных в некоторые системы обработки данных (текстовые редакторы, электронные таблицы и т.д.). Для своего размножения такие вирусы используют возможности макроязыков и при их помощи переносят себя из одного зараженного файла (документа или таблицы) в другие. Наибольшее распространение получили макро-вирусы для Microsoft Word, Excel и Office97. Существуют также макро-вирусы, заражающие документы Ami Pro и базы данных Microsoft Access.

Для существования вирусов в конкретной системе (редакторе) необходимо наличие встроенного в систему макроязыка с возможностями:

1) привязки программы на макроязыке к конкретному файлу;
2) копирования макропрограмм из одного файла в другой;
3) возможность получения управления макропрограммой без вмешательства пользователя (автоматические или стандартные макросы).

Данным условиям удовлетворяют редакторы Microsoft Word, Office97 и AmiPro, а также электронная таблица Excel и база данных Microsoft Access. Эти системы содержат в себе макроязыки: Word - Word Basic, Excel, Office97 (включая Word97, Excel97 и Access) - Visual Basic for Applications. При этом:

1) макропрограммы привязаны к конкретному файлу (AmiPro) или находятся внутри файла (Word, Excel, Office97);
2) макроязык позволяет копировать файлы (AmiPro) или перемещать макропрограммы в служебные файлы системы и редактируемые файлы (Word, Excel, Office97);
3) при работе с файлом при определенных условиях (открытие, закрытие и т.д.) вызываются макропрограммы (если таковые есть), которые определены специальным образом (AmiPro) или имеют стандартные имена (Word, Excel, Office97).

Данная особенность макроязыков предназначена для автоматической обработки данных в больших организациях или в глобальных сетях и позволяет организовать так называемый "автоматизированный документооборот". С другой стороны, возможности макроязыков таких систем позволяют вирусу переносить свой код в другие файлы, и таким образом заражать их.

На сегодняшний день известны четыре системы, для которых существуют вирусы - Microsoft Word, Excel, Office97 и AmiPro. В этих системах вирусы получают управление при открытии или закрытии зараженного файла, перехватывают

стандартные файловые функции и затем заражают файлы, к которым каким-либо образом идет обращение. По аналогии с MS-DOS можно сказать, что большинство макро-вирусов являются резидентными: они активны не только в момент открытия/закрытия файла, но до тех пор, пока активен сам редактор.

1.1. Макро вирусы в Word/Excel/Office

Физическое расположение вируса внутри файла зависит от его формата, который в случае продуктов Microsoft чрезвычайно сложен - каждый файл-документ Word, Office97 или таблица Excel представляют собой последовательность блоков данных (каждый из которых также имеет свой формат), объединенных между собой при помощи большого количества служебных данных. Этот формат носит название OLE2 - Object Linking and Embedding. Структура файлов Word, Excel и Office97 (OLE2) напоминает усложненную файловую систему дисков DOS: "корневой каталог" файла-документа или таблицы указывает на основные подкаталоги различных блоков данных, несколько таблиц FAT содержат информацию о расположении блоков данных в документе и т.д.

Более того, система Office Binder, поддерживающая стандарты Word и Excel позволяет создавать файлы, одновременно содержащие один или несколько документов в формате Word и одну или несколько таблиц в формате Excel. При этом Word-вирусы способны поражать Word-документы, а Excel-вирусы - Excel-таблицы, и все это возможно в пределах одного дискового файла. То же справедливо и для Office97.

По причине такой сложности форматов файлов Word, Excel и Office97 представить расположение макро-вируса в файле можно схематично, как показано на рис. 10.

Рис. 10. Схема расположение макровируса в файле

Следует отметить, что Word версий 6 и 7 позволяет шифровать присутствующие в документе макросы. Таким образом, некоторые Word-вирусы присутствуют в зараженных документах в зашифрованном (Execute only) виде.

Большинство известных вирусов для Word несовместимы с национальными (в том числе с русской) версиями Word, или наоборот - рассчитаны только на локализованные версии Word и не работают под английской версией. Однако вирус в документе все равно остается активным и может заражать другие компьютеры с установленной на них соответствующей версией Word.

Вирусы для Word могут заражать компьютеры любого класса, а не только IBM-PC. Заражение возможно в том случае, если на данном компьютере установлен текстовый редактор, полностью совместимый с Microsoft Word версии 6 или 7 (например, MS Word for Macintosh). То же справедливо для Excel и Office97.

Следует также отметить, что сложность форматов документов Word, таблиц Excel и особенно Office97 имеет следующую особенность: в файлах-документах и таблицах присутствуют "лишние" блоки данных, т.е. данные, которые никак не связаны с редактируемым текстом или таблицами, либо являются случайно оказавшимися там копиями прочих данных файла. Причиной возникновения таких блоков данных является кластерная организация данных в OLE2-документах и таблицах - даже если введен всего один символ текста, то под него выделяется один или даже несколько кластеров данных. При сохранении документов и таблиц в кластерах, не заполненных "полезными" данными, остается "мусор", который попадает в файл вместе с прочими данными. Количество "мусора" в файлах может быть уменьшено отменой пункта настройки Word/Excel "Allow Fast Save", однако это лишь уменьшает общее количество "мусора", но не убирает его полностью.

Следствием этого является тот факт, что при редактировании документа его размер изменяется вне зависимости от производимых с ним действий - при добавлении нового текста размер файла может уменьшится, а при удалении части текста - увеличиться. То же и с макро-вирусами: при заражении файла его размер может уменьшиться, увеличиться или остаться неизменным.

Следует также отметить тот факт, что некоторые версии OLE2.DLL содержат небольшой недочет, в результате которого при работе с документами Word, Excel и особенно Office97 в блоки "мусора" могут попасть случайные данные с диска, включая конфиденциальные (удаленные файлы, каталоги и т.д.). В эти блоки могут попасть также команды вируса. В результате после лечения зараженных документов активный код вируса удаляется из файла, но в блоках "мусора" могут остаться часть его команд. Такие следы присутствия вируса иногда видимы при помощи текстовых редакторов и даже могут вызвать реакцию некоторых антивирусных программ. Однако эти остатки вируса совершенно безвредны: Word и Excel не обращают на них никакого внимания.

1.2. Принципы работы

Макро-вирусы, поражающие файлы Word, Excel или Office97, как правило, пользуются одним из трех приемов: в вирусе либо присутствует авто-макрос (авто-функция), либо переопределен один из стандартных системных макросов (ассоциированный с каким-либо пунктом меню), либо макрос вируса вызывается

автоматически при нажатии на какую-либо клавишу или комбинацию клавиш. Существуют также полу-вирусы, которые не используют всех этих приемов и размножаются, только когда пользователь самостоятельно запускает их на выполнение.

Таким образом, если документ заражен, при открытии документа Word вызывает зараженный автоматический макрос AutoOpen (или AutoClose при закрытии документа) и, таким образом, запускает код вируса, если это не запрещено системной переменной DisableAutoMacros. Если вирус содержит макросы со стандартными именами, они получают управление при вызове соответствующего пункта меню (File/Open, File/Close, File/SaveAs). Если же переопределен какой-либо символ клавиатуры, то вирус активизируется только после нажатия на соответствующую клавишу.

Например, зараженный макрос со стандартным именем может содержать процедуру, удаляющую другие файлы:

```
Sub Effect() ' Процедура удаления первого попавшегося doc-
файла в каталоге с текущим документом
  On Error Resume Next
  Kill (Dir(ActiveDocument.Path & "\*.doc"))
End Sub
```

Простейшим примером полиморфизма макровируса может быть вставка дополнительных строк в тело вируса:

```
Sub GetRandom() ' Процедура вставки в тело вируса
комментариев произвольной длины и содержания
  On Error Resume Next
  Set Current = MacroContainer
   For Grow = 1 To 20
    Number =
Current.VBProject.VBComponents("Vir").CodeModule.CountOfLines
      RandomLine = Int(Rnd() * Number + 1)
      RemarkLength = Int(Rnd() * 40 + 1)
        For Length = 1 To RemarkLength
          Remark = Remark + Chr$(Int((90 - 65 + 1) * Rnd +
65))
        Next Length

Current.VBProject.VBComponents("Vir").CodeModule.InsertLines
RandomLine, vbTab & "Rem " & Remark
      Remark = ""
  Next Grow
End Sub
```

Также для маскировки вирусы могут отключать некоторые функции Word, которые могут выдавать их присутствие. Например, возможно использование следующих приёмов:

```
Application.ScreenUpdating = False ' запрет обновления окна
Word

Application.DisplayAlerts = wdAlertsNone ' запрет показа
предупреждающих сообщений и сообщений об ошибках

WordBasic.DisableAutoMacros 0 ' пришедшая из Word6 команда
отключения автоматически запускаемых макросов

Options.VirusProtection = False ' отключение предупреждения
об открытии документов с макросами
```

Большинство макро-вирусов содержат все свои функции в виде стандартных макросов Word/Excel/Office97. Существуют, однако, вирусы, использующие приемы скрытия своего кода и хранящие свой код в виде не-макросов. Известно три подобных приема, все они используют возможность макросов создавать, редактировать и исполнять другие макросы. Как правило, подобные вирусы имеют небольшой (иногда - полиморфный) макрос-загрузчик вируса, который вызывает встроенный редактор макросов, создает новый макрос, заполняет его основным кодом вируса, выполняет и затем, как правило, уничтожает (чтобы скрыть следы присутствия вируса). Основной код таких вирусов присутствует либо в самом макросе вируса в виде текстовых строк (иногда - зашифрованных), либо хранится в области переменных документа или в области Auto-text.

2. Порядок выполнения работы

1. Подготовка к проведению работы.
   - 1.1. Создать рабочую директорию, в которой должны находиться файлы:
     - Normal.dot
     - Test1.doc
     - Test2.doc
   - 1.2. В Microsoft Word создать любой документ.
   - 1.3. Проанализировать и зафиксировать текущие установки Word.
2. Заражение.
   - 2.1. Открыть документ test1.doc (Microsoft Word).
   - 2.2. Провести сравнение текущих установок Word с ранее зафиксированными.
   - 2.3. Поизвести действия указанные в п. 2.1 и 2.2. для test2.doc.
3. Произвести анализ вируса
   - 3.1. Изучить принципы построения, основные функции и возможности данных макровирусов.
   - 3.2. Найти основные отличия представленных макровирусов.
4. Поиск вариантов обнаружения и защиты.
   - 4.1.Опираясь на заданные макровирусы, определить возможные признаки наличия макровирусов.

4.2.Найти и реализовать методы удаления «вредных» макросов.
4.3.Предложить методы защиты от макровирусов.

3. Содержание отчета

В отчете необходимо привести:

- результаты наблюдений при проведении заражения;
- отличия двух вирусов.
- возможные признаки наличия макровирусов;
- методы удаления вредоносных макросов;
- предложения по защите от макровирусов.

4. Контрольные вопросы

1. Предпосылки для существования макро-вирусов в системе.
2. Возможные места заражения.
3.Методы сокрытия вирусами своего присутствия.

Приложение 1. Редактор файлов HIEW

Hiew (Hacker's view) - это редактор файлов. Hiew позволяет просматривать файлы формате текста или шестнадцатиричного кода, а также в режиме дизассемблера процессора Pentium(R) Pro.

Основные функции редактора HIEW:

- редактирование файлов неограниченной длины в текстовом и шестнадцатеричном режимах, а также в режиме дизассемблера;
- встроенный ассемблер Pentium(R) Pro;
- поддерживаются NE,LE,LX,PE исполняемые форматы файлов;
- наличие небольшой встроенной системы шифрования/дешифрования;
- мультифайловые поиск и замена;
- поиск ассемблерных команд по шаблону.

Командная строка

Hiew [/SAV=<savefile>] [/INI=<inifile>] [/s]filemask ...[/s][filemask]

где /SAV=<savefile> - откуда брать savefile,
/INI=<inifile> - откуда брать inifile,
[/s] filemask ... [/s][filemask] - можно задавать несколько файлов, включаю маску с шаблоном.

Если в том же каталоге, что и HIEW.EXE находится HIEW.INI то начальные установки берутся из него. Установки из Ini-файла перекрывают значения по умолчанию.

При запуске без параметров Hiew ищет savefile (HIEW.SAV или значение savefile= в INI-файле) и восстанавливает записанное положение (Ctrl/F10 - SaveState).

Опция /s переключает поиск с подкаталогами. Например

hiew /s *.dll *.exe /s *.txt

будет искать .dll и .exe с подкаталогами и .txt только в текущем каталоге.

Информационная строка

Содержание информационной строки редактора приведено на рис.1.

Xxx%	Filename.ext	d	F	R	NE	xxxxxxxx	xxx	----	YYYYYYY	HIEW X.XX (c) SEN
1	2	3	4	5	6	7	8	9	10	

Рис. 1. Содержание информационной строки

На рисунке цифрами обозначено:
1) процентов смещение от всего файла (если bar = P в HIEW.INI);
2) имя файла;
3) направление поиска;
4) область поиска: F - весь текущий файл; B – блок; A - список из командной строки;
5) состояние файла: R - открыт по чтению; W - открыт по записи; U – изменен;
6) тип NewExe
7) текущее
8) в режиме Text - номер первой колонки;
в режиме DeCode - размерность операндов/адресов префикс 'a' говорит об автоматическом определинии размерности кода для LX;
9) закладки: '-' свободно; '1...8' соотв.поз.занята; '*' текущий;
"<Editor>", если в режиме редактирования;
10) длина файла в байтах;

За более детальным описанием функций редактора можно обратиться к файлу Hiewr.xxx (Hiewe.xxx).

В файле HIEW.HLP приводиться подробное описание сочетания клавиш применяемых при редактировании в HIEW. Также HIEW.HLP одновременно является файлом для встроенной помощи - вызывается при помощи клавиш Alt-H.

Приложение 2. Классификация компьютерных вирусов

Основным требованием к классификации должна быть ее объективность, т.е. классификация должна основываться на фиксированном наборе непротиворечиво измеряемых или наблюдаемых признаков. Очевидно, что объективная классификация существенно облегчает систематизацию, распространение и накопление знаний, а также выбор программных средств для борьбы с тем или иным вирусом.

Стандартизованная классификация позволяет четко определить отношение обнаруженного вируса к какой-либо группе, что позволяет в большой степени предсказать его свойства, а также помогает преодолеть проблему путаницы в названиях при обнаружении малораспространенных вирусов.

Безруковым Н.Н. была предложена следующая схема классификации, включающая три основных элемента: код вируса; дескриптор вируса (формализованный список основных свойств); сигнатуру вируса (строка для контекстного поиска данного вируса в зараженной программе). Рассмотрим эти элементы.

Классификационный код вируса

В предлагаемой схеме каждому вирусу присваивается код, состоящий из буквенного префикса, количественной характеристики и факультативного буквенного суффикса. Например, в коде

RCE-1813c RCE - префикс, 1813 - корень (характеристика), а c - суффикс. Кроме того, факультативное расширение, записываемое в конце кода через точку, характеризует группу, к которой относится данный вирус. Например, RCE-1813.IER означает, что данный вирус относится к иерусалимской группе.

Префикс характеризует место расположения головы вируса и состоит из букв и цифр, начинаясь с прописной буквы. В соответствии с этим можно различать следующие типы вирусов:

1) файловые (голова вируса располагается в COM-, EXE-файлах и оверлеях - символы C, E в префиксе. При этом дополнительную букву, отражающую заражение оверлеев в префикс вводить не будем, чтобы избежать его переусложнения, а вынесем в дескриптор);

2) бутовые (голова вируса располагается в бут-секторе или блоке MBR - символы B, D или M в префиксе);

3) пакетные (голова вируса расположена в пакетном файле, т.е. представляет собой строку или программу на языке управления заданиями операционной системы - префикс J).

Так как наряду с "чистыми" вирусами, использующими одну среду, существуют и комбинированные - сочетающие свойства файловых и бутовых вирусов. В таких вирусах вместо первой буквы R используется соответствующую букву префикса бутового вируса, например BCE или MCE (как и бутовые, смешанные вирусы не могут быть нерезидентными).

Характеристика вируса представляет собой количественно измеряемое свойство вируса, допускающее простое определение и отличающееся для большинства типов вирусов. Например, для файловых вирусов в качестве характеристики может использоваться величина приращения длины файлов при заражении ("инфективная длина").

Суффикс используется, когда два разных вируса или два штамма одного и того же вируса имеют одинаковый префикс и характеристику.

Дескриптор вируса

Дескриптор представляет собой систематизацию основных характеристик вируса в закодированном виде. Кодировка состоит из групп символов, начинающихся с заглавной латинской буквы, за которой следуют строчные латинские буквы или цифры. При этом заглавная латинская буква определяет вид характеристики, а следующие за ней маленькие буквы или цифры – значение характеристики для конкретного вируса. Например, в дескрипторе "Xab YcZdmt" имеются три свойства: X - со значением "ab", Y - со значением "c", и Z - со значением "dmt".

Сигнатура вируса

Одной из важных задач классификации является составление списка строк для контекстного поиска (сигнатур). Стандартизация сигнатур особенно важна, когда вирус имеет много штаммов, поскольку формальные схемы, подобные описанным выше классификационному коду и дескриптору, обладают тем недостатком, что некоторые штаммы будут неразличимы в заданном пространстве признаков. В то же время сравнительно легко обеспечить уникальность сигнатуры.

Также, в общем виде, вирусы можно охарактеризовать, разделяя их на классы по следующим основным признакам:

- среда обитания;
- операционная система (ОС);
- особенности алгоритма работы;
- деструктивные возможности.

По среде обитания вирусы можно разделить на:

- файловые (либо различными способами внедряются в выполняемые файлы, либо создают файлы-двойники, либо используют особенности организации файловой системы);
- загрузочные(записывают себя либо в загрузочный сектор диска (boot-сектор), либо в сектор, содержащий системный загрузчик винчестера , либо меняют указатель на активный boot-сектор);

- макро (заражают файлы-документы и электронные таблицы нескольких популярных редакторов);
- сетевые (используют для своего распространения протоколы или команды компьютерных сетей и электронной почты);
- комбинированные.

Заражаемая операционная система (точнее, ОС, объекты которой подвержены заражению) является вторым уровнем деления вирусов на классы. Каждый файловый или сетевой вирус заражает файлы какой-либо одной или нескольких ОС (DOS, Windows, Win95/NT, OS/2 и т.д).

Особенностями алгоритма работы вирусов можно назвать:

- резидентность;
- использование stealth-алгоритмов;
- самошифрование и полиморфичность;
- использование нестандартных приемов.

Под термином "*резидентность*" (DOS'овский термин TSR - Terminate and Stay Resident) понимается способность вирусов оставлять свои копии в системной памяти, перехватывать некоторые события

Stealth-вирусы теми или иными способами скрывают факт своего присутствия в системе. Использование stealth-алгоритмов позволяет вирусам полностью или частично скрыть себя в системе.

К *полиморфик-вирусам* относятся те из них, детектирование которых невозможно (или крайне затруднительно) осуществить при помощи сигнатур вирусов (существует два основных вида полиморфизма: шифрованием основного кода вируса с непостоянным ключем и модификацией прграммы-расшифровщика или изменением самого выполняемого кода вируса).

Различные нестандартные приемы часто используются в вирусах для того, чтобы как можно глубже спрятать себя в ядре ОС, защитить от обнаружения свою резидентную копию, затруднить лечение от вируса (например, поместив свою копию в Flash-BIOS) и т.д.

По деструктивным возможностям вирусы можно разделить на:

- безвредные, т.е. никак не влияющие на работу компьютера (кроме уменьшения свободной памяти на диске в результате своего распространения);
- неопасные, влияние которых ограничивается уменьшением свободной памяти на диске и графическими, звуковыми и пр. эффектами;
- опасные вирусы, которые могут привести к серьезным сбоям в работе компьютера;
- очень опасные, в алгоритм работы которых заведомо заложены процедуры, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, записанную в системных областях памяти и т.п.

СОДЕРЖАНИЕ