**Лекция 14. AntiDоg**

**Вопросы:**

* 1. **На прошлой лекции был записан алгоритм антивирусной программы для DHog68. Записать его.**
  2. **Из скольки байт состоит сигнатура вируса Dhog68.**

**На прошлой лекции мы успели определить МАКРОКОМАНДУ, записав соответствующее макроопределение: Установить маркер текущего байта в файле, заданного 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.

**PutStr macro Text**

**mov ah,09h**

**mov dx,offset Text**

**int 21h**

**endm**

**Как видим, макроОПРЕДЕЛЕНИЯ записываются до основного текста на языке записи алгоритмов ассемблер. Переходим к описанию алгоритма:**

**seg000 segment byte public 'CODE'**

**assume cs:seg000**

**org 100h**

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

**public start**

**start proc near**

**PutStr TitleStr**

**TitleStr db '+----------------------------+',13,10**

**db '|AntiDHog - antivirus example|',13,10**

**db '+----------------------------+',13,10,'$'**

**В результате ПРЕПРОЦЕССОРОМ будет сформировано МАКРОРАСШИРЕНИЕ =>**

**Mov ah,09**

**Mov dx, offset TitleStr**

**Int 21h**

**Далее установим адрес области памяти DTA, в которой, как мы знаем, обычно хранится информация о найденных с помощью соответствующей функции 21H прерывания:**

**mov ah,1ah**

**mov dx,offset DTA**

**mov cx,27h**

**int 21h**

установить адрес DTA

Вход

* AH = 1aH
* DS:DX = адрес для DTA

Выход

* нет =

Описание:

устанавливает адрес DTA.

**DTA db 43 dup(0)**

**Вопросы: 14.3. Почему и зачем в регистр СХ помешается значение, равное 27h, а область памяти DTA описана как 43 байта?**

* 1. **Зачем нужна эта область DTA, если она есть в PSP?**

**Находим в текущей папке первый совпадающий по маске файл:**

**mov ah,4eh**

**mov dx,offset a\_MaskForVir**

**int 21h**

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

**С помощью команд пересылки символов-байтов, к которым относится например команда movsb,и с помощью директивы rep сохраним полученное в DTA имя найденного файла в области памяти FNAME. Команда Mobsb делает только то, что делает: Берёт содержимое одного байта ОП, адрес которого равен (DS):(SI), пересылает (КОПИРУЕТ) содержимое этого байта в байт, адрес которого определяется парой регистров (ES):(DI) И после чего ДОБАВЛЯЕТ к регистрам SI и DI по единичке к каждому!!!**

**BegScan:**

**push ds**

**pop es**

**mov si,FN\_Ofs**

**mov di,offset FName**

**mov cx,13**

**rep movsb**

***FN\_Ofs equ offset DTA+1eh***

**FName db 128 dup(0)**

***Префикс REP предполагает выполнение подпрефиксной команды столько раз, чему равно содержимое регистра СХ, рассмтриваемое как целое положительное число с фиксированной точкой.***

***Вопрос: 14.5.Сколько раз будет выполнена команда movsb в следующих случаях:***

* 1. ***Mov CX, 0***

***Rep movsb***

* 1. ***Mov cx,-1***

***Rep movsb***

***Проверить!!!***

**Получить атрибуты файла**

**mov ah,43h**

**mov al,0**

**mov dx,offset FName**

**int 21h**

**Установить атрибуты файла**

**mov ah,43h**

**mov al,1**

**mov dx,offset FName**

**mov cl,0**

**int 21h**

**Получить Handle-файл:**

**mov ax,3d10h**

**mov dx,offset FName**

**int 21h**

**Установить начальные значения для процедур:**

**mov FHandle,ax**

**mov ah,40h**

**mov cl,44h**

**nop**

**nop**

**Вывести на экран сообщение об анализируемом файле:**

**call InfoAboutFile**

**Осуществить проверку о наличии в анализируемом файле соответствующей сигнатуры:**

**call ReadSignature**

**cmp SignatureFound,0**

**je NextFile**

**Если сигнатура найдена, необходимо «убить» исполнимый модуль:**

**call KillExecutable**

**а затем удалить его:**

**call RemoveExecutable**

**NextFile:**

**Вывести на экран сообщение о результате обработки очередного файла:**

**call CureInfo**

**Найти очередной совпадающий по маске файл:**

**mov ah,4fh**

**int 21h**

**jnb BegScan**

**int 20h**

**start endp**

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

**DTA db 43 dup(0)**

**FN\_Ofs equ offset DTA+1eh**

**FName db 128 dup(0)**

**IName db 128 dup(0)**

**SignatureFound db 0**

**SignatureArray db 14 dup(0)**

**VirSignature db 0cdh,21h,0b4h,43h,0b0h,01h,0bah**

**db 9eh,00h,0b1h,00h,0cdh,21h,0b8h**

**Int20Cmd db 0cdh,20h**

**FHandle dw 0**

**TitleStr db '+----------------------------+',13,10**

**db '|AntiDHog - antivirus example|',13,10**

**db '+----------------------------+',13,10,'$'**

**NormStr db ' - Ok',13,10,'$'**

**CureStr db ' - deleted!',13,10,'$'**

**ReadSignature proc near**

**mov SignatureFound,0**

**MoveFPos FHandle 0eh**

**mov ah,3fh**

**mov bx,FHandle**

**mov cx,14**

**mov dx,offset SignatureArray**

**int 21h**

**cld**

**push ds**

**pop es**

**mov si,offset VirSignature**

**mov di,offset SignatureArray**

**mov cx,14**

**repe cmpsb**

**jnz Finish**

**mov SignatureFound,1**

**Finish:**

**ret**

**ReadSignature endp**

**KillExecutable proc near**

**MoveFPos FHandle 0**

**mov ah,40h**

**mov bx,FHandle**

**mov cx,2**

**mov dx,offset Int20Cmd**

**int 21h**

**mov ah,3eh**

**mov bx,FHandle**

**int 21h**

**ret**

**KillExecutable endp**

**RemoveExecutable proc near**

**mov ah,41h**

**mov dx,offset FName**

**int 21h**

**ret**

**RemoveExecutable endp**

**InfoAboutFile proc near**

**push ds**

**pop es**

**mov si,offset FName**

**mov di,offset IName**

**NextChar:**

**lodsb**

**stosb**

**cmp al,0**

**jne NextChar**

**dec di**

**mov byte ptr [di],'$'**

**PutStr IName**

**ret**

**InfoAboutFile endp**

**CureInfo proc near**

**mov ah,9**

**mov dx,offset NormStr**

**cmp SignatureFound,0**

**je ContMsg**

**mov dx,offset CureStr**

**ContMsg:**

**int 21h**

**ret**

**CureInfo endp**

**seg000 ends**

**end start**

Вопрос 14.6. Какая разница между «убить» и «удалить» Executable?

Вопрос 14.7. Какая ошибка допущена в названии Лекции 14.

Домашнее задание – ДО конца дня (до 23:59) 13.12.2021. Используя DosBox и вирус DHog68 «ЗАРАЗИТЬ» все файлы в выбранной для этого папке. Используя отладчик удостовериться, что все файлы «заражены». Отладить AntiDHog, поместить антивирус в папку с зараженными текстами и удостовериться, что этот антивирус «работает».