**Лекция 8. Quest.**

**В конце прошлой Лекции было записано.**

**Ответы на вопросы 4 – 4.0 – 4.5 СДАТЬ ДО НАЧАЛА ЛЕКЦИИ 25.10.2023!!!**

**Домашнее задание: На листочках НАБРАТЬ (можно использовать книгу про ассемблер) и распечатать ПОЛНЫЙ ТЕКСТ АНТИВИРУСА, чтобы быть готовым к следующей Лекции 8, которая называется «Quest: Kill The Hog».**

***НЕ СПИСЫВАТЬ! НЕ ИСПОЛЬЗОВАТЬ ЧРТ – У КАЖДОГО ДОЛЖЕН БЫТЬ СВОЙ СОБСТВЕННОРУЧНО НАБРАННЫЙ ТЕКСТ! В СЛУЧАЕ ОБНАРУЖЕНИЯ ЧРТ ВСЕ РАБЫ И РАБОВЛАДЕЛЬЦЫ ПОЛУЧАТ ПО МИНУС 256 баллов в рейтинг. ПРОСЬБА! ОТОРВИТЕСЬ ОТ СВОИХ ГАДЮК, МАРОК КОФЕ, ВСЕМИРНОЙ ЛИПКОСТИ и прочих помоек: просто наберите текст – может хоть что-то «зацепит». Тем более, что этот текст понадобится для лабораторной работы №2. Часть 2.***

***А теперь непосредственно Quest!!!***

***Лекция 8. 25.10.2023. AntiDOg Quest.***

***На прошлой лекции Вы по диагонали или по вертикали «прочитали» о том, что такое «AntiDHog68». Был записан алгоритм функционирования антивируса, определено, что будет являться Signature, приведено описание того, что такое макрокоманда, макроопределение, макрорасширение. В процессе написания антивируса был использован аппарат ПРОЦЕДУР. Исходя из предположения, что в процессе чтения Лекции 8 Вы нетривиально проДУРКовали, и выполнили домашнее задание (приготовив тексты антивируса) сегодняшняя Лекция будет построена так:***

1. ***Вы ещё раз изучаете текст антивируса, но вникая в детали, так как известно, что "Дьявол скрывается в мелочах».***
2. ***В процессе изучения Вы должны будете ПИСЬМЕННО НА СВОИХ ЛИСТОЧКАХ с текстом антихога (на обратной стороне или рядышком по тексту. И если кто-то типа Веренича решит сэкономить – пусть пеняет на себя) ОТВЕЧАТЬ НА ВОПРОСЫ ПО ТЕКСТУ ЛЕКЦИИ (именно на занятии – в течение 17-30 – 18-50). НА ЛИСТОЧКАХ!!!! А в конце занятия ЛИСТОЧКИ сдаются мне !!!***
3. ***А до конца ЭТОЙ недели, необходимо ВЫПОЛНИТЬ ДОМАШНЕЕ ЗАДАНИЕ - набрать исходные тексты, присутствующие в вопросах, на языке ассемблер, транслировать, линковать, выполнить, продемонстрировать ПРАВИЛЬНЫЕ ОТВЕТЫ на все 16 вопросов в отчёте.***

***Итак, погнали….***

***MoveFPos macro F\_Handle, FPos***

***mov ax,4200h***

***mov bx,F\_Handle***

***xor cx,cx***

***mov dx,FPos***

***int 21h***

***endm***

***PutStr macro Text***

***mov ah,09h***

***mov dx,offset Text***

***int 21h***

***endm***

***Вопрос 1:***

* 1. ***Дать (взяв его из Лекций Зубовичаи указав номера лекций, страницы и сколько раз даётся это описание) описание функции 9 21h прерывания на примере, в котором область памяти Text определена следующим образом:***

***Text db ‘Mama’, 10, ‘Mila’, 13, ‘ramu’, ‘10’,13,”$”***

* 1. ***И ответить на вопрос: «Что будет выведено на экран»?***

***Переходим к записи 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***

***Вопрос 2. Какое макрорасширение будет сформировано ПРЕПРОЦЕССОРОМ по макрокоманде PutStr.***

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

***mov ah,1ah***

***mov dx,offset DTA***

***mov cx,27h***

***int 21h***

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

***mov ah,4eh***

***mov dx,offset a\_MaskForVir***

***int 21h***

***. . .***

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

***Вопрос 3. Доказать или опровергнуть, что «первый совпадающий по маске файл» определяется «текущим признаком сортировки» файлов в папке (по имени, по Дате изменения, по типу, по размеру).***

***С помощью команд пересылки символов-байтов, к которым относится и команда movsb, и с помощью директивы rep сохраним полученное в DTA имя найденного файла в области памяти FNAME. Команда Movsb 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***

***Вопрос 4: Адрес области памяти FN\_Ofs определен следующим образом:***

***FN\_Ofs equ offset DTA+1eh. Что такое Equ, что такое адрес, что такое DTA + 1eh и что означает offset.***

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

***Вопрос 5: Нарисовать в шестнадцатеричном виде содержимое регистра СХ после выполнения команды MOV CX, -128. И что произойдёт, если вместо фрагмента:***

***mov si,FN\_Ofs***

***mov di,offset FName***

***mov cx,13***

***rep movsb***

***записать***

***mov si,FN\_Ofs***

***mov di,offset FName***

***mov cx,-128***

***rep movsb***

***Дома с помощью турбо-отладчика доказать, что как Вы написали на листочке, так и будет…***

***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-файла:***

***mov ax,3d10h***

***mov dx,offset FName***

***int 21h***

***Вопрос 6: Записать фрагмент программы на языке ассемблер ДЛЯ ВЫВОДА НА ЭКРАН имени файла и Handle файла, имя которого находится в FName.***

***Установить начальные значения для процедур:***

***mov FHandle,ax***

***mov ah,40h***

***mov cl,44h***

***nop***

***nop***

***Вопрос 7: Зачем здесь ЗАПИСАНЫ две команды Nop. Что будет, если их убрать?***

***Вопрос 8: Далее осуществляется вызов процедуры InfoAboutFile. Что происходит при вызове процедуры, как передаётся управление процедуре и что происходит со стеком. Готовя отчёт к концу недели, используя отладчик показать, что находится в стеке до вызова процедуры и после выхода из процедуры, определить адрес ТОЧКИ ВОЗВРАТА из этой процедуры в головную программу. НАРИСОВАТЬ!!!!***

***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***

***Вопрос 9.В чём разница между действиями: УБИТЬExecutable и УДАЛИТЬExecutable. Ответ типа: «Это соответствующий перевод слов Kill и Remove» - НЕ ПРИНИМАЕТСЯ…***

***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***

***Вопрос 10: Построить макрорасширение, соответствующее макрокоманде***

***MoveFPos FHandle 0eh***

***mov ah,3fh***

***mov bx,FHandle***

***mov cx,14***

***mov dx,offset SignatureArray***

***int 21h***

***cld***

***push ds***

***pop es***

***Вопрос 11: Зачем ЗАПИСАНЫ эти команды Push и Pop.***

***mov si,offset VirSignature***

***mov di,offset SignatureArray***

***mov cx,14***

***repe cmpsb***

***Вопрос 12: Дать ПОЛНОЕ (нарисовать!!!!) ОПИСАНИЕ того, ЧТО ПРОИСХОДИТ в результате repe cmpsb, дать определение Префикса Repe и КОМАНДЫ Cmpsb!!!! Не помоечно!!***

***jnz Finish***

***mov SignatureFound,1***

***Finish:***

***Ret***

***Вопрос 13: Нарисовать, что происходит ПО КОМАНДЕ Ret. А В отчёте до конца недели ПОКАЗАТЬ, ЧТО НАХОДИТСЯ В СТЕКЕ ДО ВЫПОЛНЕНИЯ КОМАНДЫ RET и ПОСЛЕ ВЫПОЛНЕНИЯ КОМАНДЫ RET! Используя ОТЛАДЧИК!***

***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***

***Вопрос 14: Сколько байт, каких (в 16 с.с.) ЗАПИСЫВАЕТСЯ (посредством использования какой функции и В КАКОЙ ФАЙЛ) при вызове процедуры KillExecutable.***

***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***

***Вопрос 15: Дать подробное описание команд***

***Lodsb и 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 16: Какие действия осуществляются транслятором при трансляции ДИРЕКТИВЫ End start. Нарисовать.***

***Ещё раз!!! До конца недели НЕОБХОДИМО, ИСПОЛЬЗУЯ ОТЛАДЧИК, РИВЕИМИТ НА ВСЕ 16 вопросов Questa и прислать на почту ПОЛНЫЙ\_ПОЛНЫЙ отчёт – ответы на все 16 вопросов, продемонстрировав использование отладчика!!!***