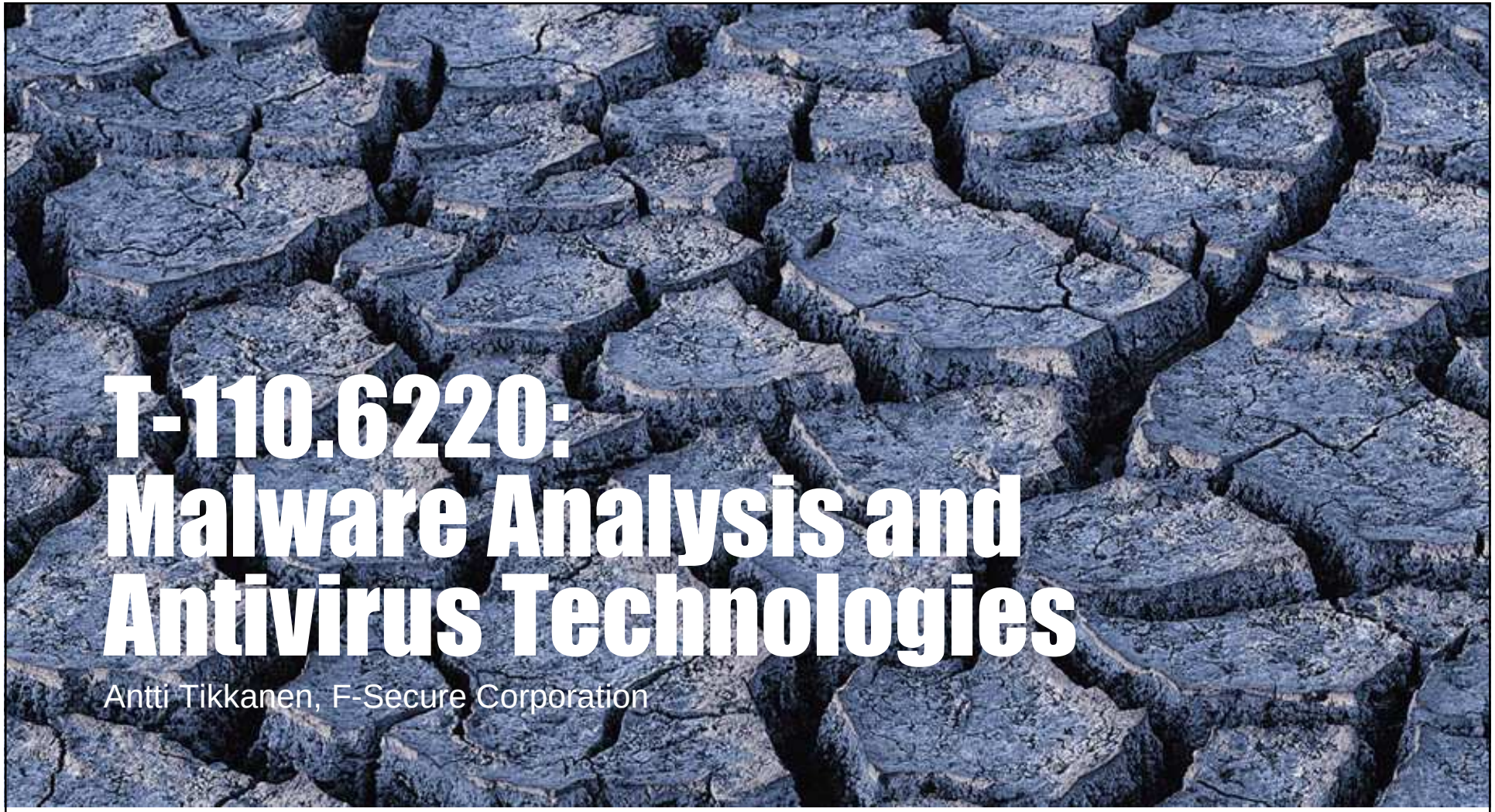


T-110.6220: Malware Analysis and Antivirus Technologies

Antti Tikkanen, F-Secure Corporation

F-SECURE®

BE SURE.

Introduction

The course teaches students what malicious code is and how it can be detected and analyzed.

Topics of the course include

- Reverse engineering and debugging malware
- Unpacking and decrypting malware
- Windows with an antivirus perspective
- Antivirus engine basics
- Spyware and mobile malware

This course includes a homework project that requires programming skills

Course staff

Lecturer

- Antti Tikkanen, F-Secure Corporation

Assistant

- Laura Takkinen

Visiting lecturers from F-Secure

- Mikko Hyppönen
- Gergely Erdelyi (reverse engineering)
- Jarkko Turkulainen (unpacking)
- Jarno Niemelä (mobile)
- Stefan Lundström (spyware)

Course information

Lectures Wednesdays at 16-18 in lecture hall TU1 (TUAS building)

- A few exceptions, see the web page
- All lectures (as well as all other material) are in English

To pass this course, you must complete:

1. three homework rounds
2. the course project

Grade based on points from homeworks and project

Prerequisites

1. T-110.4100 Computer networks
2. T-106.1220 Data Structures and Algorithms
3. Basic understanding of Windows OS or other OS internals
   (e.g. "T-106.5150 Operating Systems Project")
4. Computer architecture
   (e.g. "S-87.3190 Computer Architecture")
5. C or Assembly programming skills

Course material

The Art of Computer Virus Research and Defense

Peter Szor (Author)

ISBN 978-0321304544,

Addison-Wesley Professional, 2005

We won't go through all of the book, but lecturers will refer to it quite often for additional information.

Course material (tools for homeworks)

IDA Pro 4.9 Disassembler and Debugger

- http://www.hex-rays.com/idapro/idadownfreeware.htm

OllyDbg 1.10 Debugger

- http://www.ollydbg.de

Debugging Tools for Windows

- http://www.microsoft.com/whdc/devtools/debugging/default.mspx

You can freely install these to your own machines as well!

Contact information

Course web page:

- http://www.tml.tkk.fi/Opinnot/T-110.6220/

Mailing list

- T-110.6220@tml.hut.fi (both lecturer and assistant)

Signing up

By e-mail, see instructions here:

- http://www.tml.tkk.fi/Opinnot/T-110.6220/2008/sign_up.html

We can only accept 40 students

- If needed, we'll prioritize based on completed pre-requirement courses and your study programme

Deadline for sign up is Sunday 20th January at midnight!

More about course content

We have three important themes:

1. **Reverse engineering (RE).** The process of discovering the technological principles of a device, object or system through analysis of its structure, function and operation (Wikipedia).
2. **Windows**. To understand the sample you are reversing, you need to understand the environment (the OS) at a low level.
3. **Antivirus technologies**. We focus on the classical part of client side technology, the file scanning engine.

Homeworks

The course includes three individual homework assigments

1. Reverse engineering with IDA Pro
2. Debugging with OllyDbg/WinDbg
3. Malware taxonomy and malware in 2008

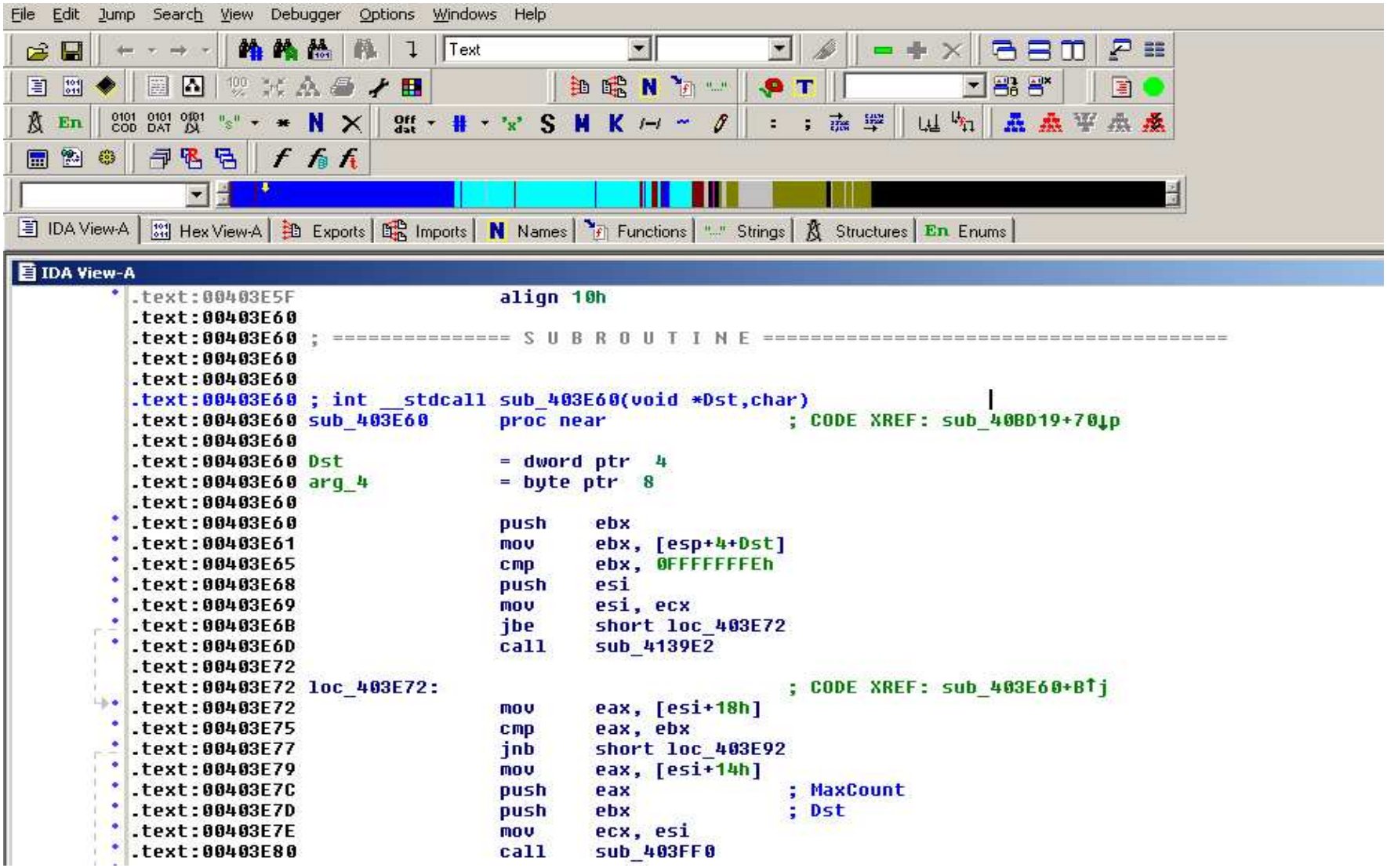
We will have a lab session to introduce the tools (TBA)

- Not mandatory, but very warmly recommended to pass the homework

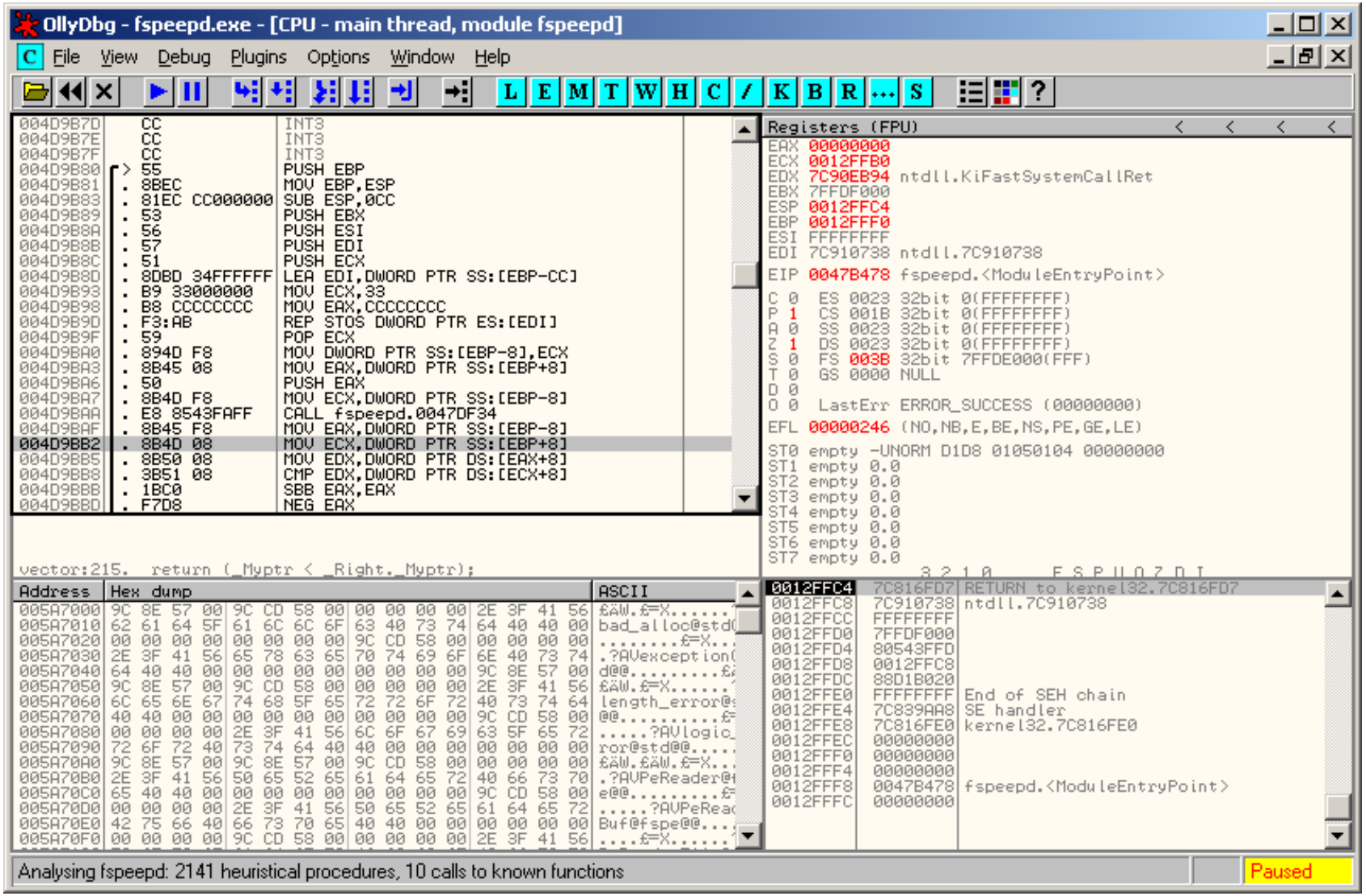
Detailed descriptions published later

You will not be handling real malware!

Homework #1


File Edit Jump Search View Debugger Options Windows Help
Text
IDA View-A
Hex View-A
Exports
Imports
Names
Functions
Strings
Structures
Enums
IDA View-A
.text:00403E5F align 10h
.text:00403E60
.text:00403E60 ; =============== S U B R O U T I N E =======================================
.text:00403E60
.text:00403E60
.text:00403E60 ; int __stdcall sub_403E60(void *Dst,char)
.text:00403E60 sub_403E60 proc near ; CODE XREF: sub_40BD19+70↓p
.text:00403E60
.text:00403E60 Dst = dword ptr 4
.text:00403E60 arg_4 = byte ptr 8
.text:00403E60
.text:00403E60 push ebx
.text:00403E61 mov ebx, [esp+4+Dst]
.text:00403E65 cmp ebx, 0FFFFFFFEh
.text:00403E68 push esi
.text:00403E69 mov esi, ecx
.text:00403E6B jbe short loc_403E72
.text:00403E6D call sub_4139E2
.text:00403E72
.text:00403E72 loc_403E72: ; CODE XREF: sub_403E60+B↑j
.text:00403E72 mov eax, [esi+18h]
.text:00403E75 cmp eax, ebx
.text:00403E77 jnb short loc_403E92
.text:00403E79 mov eax, [esi+14h]
.text:00403E7C push eax ; MaxCount
.text:00403E7D push ebx ; Dst
.text:00403E7E mov ecx, esi
.text:00403E80 call sub_403FF0

Homework #2


OllyDbg - fspeepd.exe - [CPU - main thread, module fspeepd]
File View Debug Plugins Options Window Help
004D9B7D CC INT3
004D9B7E CC INT3
004D9B7F CC INT3
004D9B80 > 55 PUSH EBP
004D9B81 . 8BEC MOV EBP,ESP
004D9B83 . 81EC CC000000 SUB ESP,0CC
004D9B89 . 53 PUSH EBX
004D9B8A . 56 PUSH ESI
004D9B8B . 57 PUSH EDI
004D9B8C . 51 PUSH ECX
004D9B8D . 8DBD 34FFFFFF LEA EDI,DWORD PTR SS:[EBP-CC]
004D9B93 . B9 33000000 MOV ECX,33
004D9B98 . B8 CCCCCCCC MOV EAX,CCCCCCCC
004D9B9D . F3:AB REP STOS DWORD PTR ES:[EDI]
004D9B9F . 59 POP ECX
004D9BA0 . 894D F8 MOV DWORD PTR SS:[EBP-8],ECX
004D9BA3 . 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]
004D9BA6 . 50 PUSH EAX
004D9BA7 . 8B4D F8 MOV ECX,DWORD PTR SS:[EBP-8]
004D9BAA . E8 8543FAFF CALL fspeepd.0047DF34
004D9BAF . 8B45 F8 MOV EAX,DWORD PTR SS:[EBP-8]
004D9BB2 . 8B4D 08 MOV ECX,DWORD PTR SS:[EBP+8]
004D9BB5 . 8B50 08 MOV EDX,DWORD PTR DS:[EAX+8]
004D9BB8 . 3B51 08 CMP EDX,DWORD PTR DS:[ECX+8]
004D9BBB . 1BC0 SBB EAX,EAX
004D9BBD . F7D8 NEG EAX
vector:215. return (_Myptr < _Right._Myptr);
Registers (FPU)
EAX 00000000
ECX 0012FFB0
EDX 7C90EB94 ntdll.KiFastSystemCallRet
EBX 7FFDF000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C910738 ntdll.7C910738
EIP 0047B478 fspeepd.<ModuleEntryPoint>
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDE000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM D1D8 01050104 00000000
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
Address Hex dump ASCII
0012FFC4 7C816FD7 RETURN to kernel32.7C816FD7
0012FFC8 7C910738 ntdll.7C910738
0012FFCC FFFFFFFF
0012FFD0 7FFDF000
0012FFD4 80543FFD
0012FFD8 0012FFC8
0012FFDC 88D1B020
0012FFE0 FFFFFFFF End of SEH chain
0012FFE4 7C839AA8 SE handler
0012FFE8 7C816FE0 kernel32.7C816FE0
0012FFEC 00000000
0012FFF0 00000000
0012FFF4 00000000
0012FFF8 0047B478 fspeepd.<ModuleEntryPoint>
0012FFFC 00000000
Analysing fspeepd: 2141 heuristical procedures, 10 calls to known functions
Paused

Course project

Topic: Designing and implementing an antivirus engine

- Details given on the lecture of Wednesday 9th April

Submission will include

1. Source code of the engine
   (we like C/C++, but you can go for Java or Python as well)
2. A short whitepaper explaining the solution
3. A demo session

This is also an individual assignment

Fighting Online Crime

F-Secure's CRO Mikko Hyppönen gives our first real lecture

Fighting Online Crime

- Who is the enemy? Where is he from?
- How money is being made with malware
- How modern antivirus labs work
- Where do we find the new samples
- How do we analyse them
- How to locate the criminals

Tuesday 22nd January in T1, 16-18

Questions?