

T-110.6220

Course Intro

Antti Tikkanen



Reverse Engineering Malware

```

.text:00401250 E8 BB DA 0E 00 89 44 24 04 A1 2C A3 57 00 8B 40 F++..ëD$í,úw.ï@
.text:00401260 10 89 04 24 E8 27 D5 0E 00 8B 15 2C A3 57 00 E9 ë$F'+.ï$,úw.T
.text:00401270 4B FF FF FF 8D B6 00 00 00 00 8D BF 00 00 00 00 K ì|....ì+....
.text:00401280 55 89 E5 83 EC 08 C7 04 24 01 00 00 00 FF 15 18 Uësâ8|$... $
.text:00401290 A3 57 00 E8 B8 FE FF FF 90 8D B4 26 00 00 00 00 úw.F+| Éì|&....
.text:004012A0 55 89 E5 83 EC 08 C7 04 24 02 00 00 00 FF 15 18 Uësâ8|$... $
.text:004012B0 A3 57 00 E8 98 FE FF FF 90 8D B4 26 00 00 00 00 úw.Fÿ| Éì|&....
.text:004012C0 55 8B 0D 54 A3 57 00 89 E5 5D FF E1 8D 74 26 00 UïTúw.ës] ßit&.
.text:004012D0 55 8B 0D 34 A3 57 00 89 E5 5D FF E1 90 90 90 90 Uï4úw.ës] ßÉÉÉÉ
.text:004012E0 83 EC 7C B8 70 B5 4E 00 89 44 24 34 B8 74 30 4F â8|+p|N.ëD$4+t00
.text:004012F0 00 89 44 24 38 8D 44 24 60 89 44 24 3C B8 90 13 .ëD$8ìD$`ëD$<+É
.text:00401300 40 00 89 44 24 40 8D 44 24 1C 89 7C 24 74 89 5C @.ëD$@ìD$ë|të\
.text:00401310 24 6C 89 74 24 70 89 6C 24 78 89 64 24 44 89 04 $lèt$peì$xèd$Dë
.text:00401320 24 E8 3A BE 0E 00 8B BC 24 80 00 00 00 85 FF 0F $F:+.ï+$Ç...à ¤
.text:00401330 84 8B 00 00 00 C7 04 24 10 20 57 00 8B 94 24 80 äï...|$ w.ïö$Ç
.text:00401340 00 00 00 8D 44 24 50 89 44 24 04 BE 88 E1 56 00 ...ìD$PëD$+êßV.
.text:00401350 31 DB 89 74 24 50 B9 01 00 00 00 89 54 24 54 89 1|ët$P|...ëT$Të
.text:00401360 5C 24 58 89 4C 24 20 E8 D4 59 00 00 89 44 24 04 \XëL$ F+Y..ëD$
.text:00401370 C7 04 24 10 20 57 00 E8 B4 5A 00 00 85 C0 74 2E |$ w.F|Z..à+t.
.text:00401380 8B 40 08 BA E8 EC 56 00 89 54 24 50 EB 34 66 90 ï@|F8V.ët$Pd4fé
.text:00401390 B8 E8 EC 56 00 89 44 24 50 8B 44 24 24 89 04 24 +F8V.ëD$PìD$$ë$
.text:004013A0 B8 FF FF FF FF 89 44 24 20 E8 72 C4 0E 00 B8 E8 + ëD$ Fr-.+F
.text:004013B0 EC 56 00 89 44 24 50 89 F6 8D BC 27 00 00 00 00 8V.ëD$Pë÷ì+'....
.text:004013C0 31 C0 89 44 24 18 8D 44 24 1C 89 04 24 E8 6E BE 1+ëD$ìD$ë$Fn+
.text:004013D0 0E 00 8B 44 24 18 8B 5C 24 6C 8B 74 24 70 8B 7C .ïD$ï\$lìt$pi|
.text:004013E0 24 74 8B 6C 24 78 83 C4 7C C3 8D B6 00 00 00 00 $tìl$xâ-|+ì|....

```

Course Staff

- Staff
 - Main lecturer: Antti Tikkanen
 - Course assistant: Päivi Tynninen
- Visiting lecturers from F-Secure
 - Mikko Hyppönen
 - Paolo Palumbo (static analysis, interpreted languages)
 - Jarkko Turkulainen (obfuscation, dynamic analysis)
 - Karmina Aquino (interpreted languages)
 - Christine Bejerasco (web threats)
 - Broderick Aquilino (Mac)
 - Jarno Niemelä (corporate espionage)
- No visiting hours, only by appointment

Lectures

Date	Topic
21.1.2015	Course Intro + Threats Today
28.1.2015	Static Analysis I
4.2.2015	Static Analysis II
11.2.2015	Dynamic Analysis
25.2.2015	Windows for Reverse Engineers
4.3.2015	Binary Obfuscation and Protection
11.3.2015	Malware Analysis Automation
18.3.2015	Analyzing Interpreted Languages The Malicious Web
25.3.2015	Mac Malware Analysis Industrial Espionage
1.4.2015	Course Project Instructions

Homework

The screenshot displays the OllyDbg interface for the process fspeed.exe, CPU - main thread, module fspeedp. The left pane shows the disassembled code, the middle pane shows the CPU registers, and the right pane shows the memory dump.

Disassembled Code (Left Pane):

```

.text:00403E5F align 10h
.text:00403E60 :----- SUBROUTINE -----
.text:00403E60 ; int __stdcall sub_403E60(void *Dst, char)
.text:00403E60 sub_403E60 proc near ; CODE XREF: 5
.text:00403E60 Dst = dword ptr 4
.text:00403E60 arg_4 = byte ptr 0
.text:00403E60
.text:00403E60 push ebx
.text:00403E61 mov ebx, [esp+4+Dst]
.text:00403E65 cmp ebx, 0FFFFFFFh
.text:00403E68 push esi
.text:00403E69 mov esi, ecx
.text:00403E6B jbe short loc_403E72
.text:00403E6D call sub_4139E2
.text:00403E72 loc_403E72: ; CODE XREF: 5
.text:00403E72 mov eax, [esi+10h]
.text:00403E75 mov eax, ebx
.text:00403E77 jnb short loc_403E92
.text:00403E79 mov eax, [esi+10h]
.text:00403E7B push eax ; MaxCount
.text:00403E7D push ebx ; Dst
.text:00403E7F mov ecx, esi
.text:00403E80 call sub_403FF0

```

CPU Registers (Middle Pane):

Register	Value
EAX	00000000
ECX	0012FFB0
EDX	7C9BEB94 ntdll.KiFastSystemCallRet
EBX	7FFDF000
ESP	0012FFC4
EBP	0012FF00
ESI	FFFFFFFF
EDI	7C910738 ntdll.7C910738
EIP	0047B478 fspeedp.<ModuleEntryPoint>

Memory Dump (Right Pane):

Address	Hex dump	ASCII
0012FFC4	7C910738	RETURN to kernel(32.7C910738)
0012FFC8	7C910738	ntdll.7C910738
0012FFCC	FFFFFFFF	
0012FFD0	7FFDF000	
0012FFD4	80543FFD	
0012FFD8	0012FFC8	
0012FFDC	80D1B020	
0012FFE0	FFFFFFFF	End of SEH chain
0012FFE4	7C839A00	SE handler
0012FFE8	7C816FE0	kernel32.7C816FE0
0012FFFC	00000000	
0012FF00	00000000	
0012FF04	00000000	
0012FF08	00000000	
0012FF0C	0047B478	fspeedp.<ModuleEntryPoint>
0012FF10	00000000	

Analysing fspeedp: 2141 heuristical procedures, 10 calls to known functions

Final Project

- At the end of the course we'll have a mandatory final project
- Course Project 2015: multistage reverse engineering challenge
 - Again, individual work: share ideas, write your own answers
- Deliverables
 - Written report about your solutions
 - Possible source code, IDA databases, ...

Grading

- Lectures are not mandatory but very much recommended
- Tool trainings are not mandatory but very much recommended
- Homework
 - 3 rounds, each 20 % of your course grade
- Final project
 - 40 % of your course grade

Practical Malware Analysis

*The Hands-On Guide to
Dissecting Malicious
Software*

Michael Sikorski
and Andrew Honig

Foreword by Richard Bejtlich



No official course book, this is
a good option though!

Sign up now!

- We can take only up to 35 students
- Sign up in Noppa (don't ask me how..)
- Course **signup will close before next week's lecture**