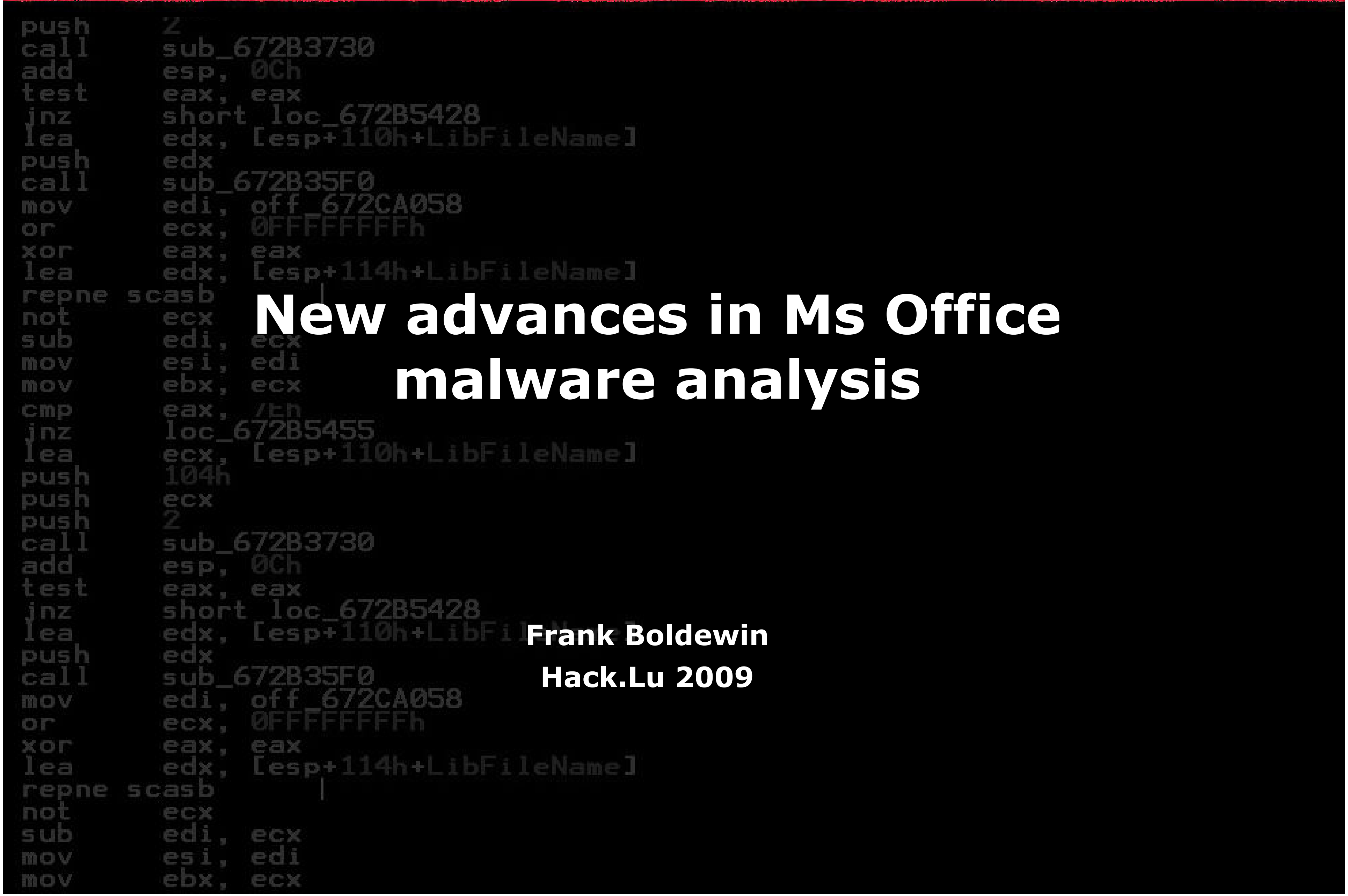


```
push        2
call        sub_672B3730
add         esp, 0Ch
test        eax, eax
jnz         short loc_672B5428
lea         edx, [esp+110h+LibFileName]
push        edx
call        sub_672B35F0
mov         edi, off_672CA058
or          ecx, 0FFFFFFFFh
xor         eax, eax
lea         edx, [esp+114h+LibFileName]
repne scasb
not         ecx
sub         edi, ecx
mov         esi, edi
mov         ebx, ecx
cmp         eax, 7Eh
jnz         loc_672B5455
lea         ecx, [esp+110h+LibFileName]
push        104h
push        ecx
push        2
call        sub_672B3730
add         esp, 0Ch
test        eax, eax
jnz         short loc_672B5428
lea         edx, [esp+110h+LibFi
push        edx
call        sub_672B35F0
mov         edi, off_672CA058
or          ecx, 0FFFFFFFFh
xor         eax, eax
lea         edx, [esp+114h+LibFileName]
repne scasb
not         ecx
sub         edi, ecx
mov         esi, edi
mov         ebx, ecx
```

New advances in Ms Office malware analysis

Frank Boldewin

Hack.Lu 2009

Agenda

- Introduction to MS Office exploitation
- Some MS Office exploits since 2006
- Short introduction to the OLESS format
- Example of a malicious MS Office document structure
- Typical MS Office Shellcode behavior
- Status Quo to MS Office document analysis
- Introduction to OfficeMalScanner

Introduction to MS Office exploitation

- MS Office commonly exploited since 2006
- Existing exploits in the wild exploit unexceptional the older OLESS file format.
- Currently no known bugs in the newer XML based MS Office format.

Some MS Office exploits since 2006

CVE-2006-0009	Powerpoint	MS06-012 (March 2006)
CVE-2006-0022	Powerpoint	MS06-028 (June 2006)
CVE-2006-2492	Word	MS06-027 (June 2006)
CVE-2006-3434	Powerpoint	MS06-062 (October 2006)
CVE-2006-3590	Powerpoint	MS06-048 (August 2006)
CVE-2006-4534	Word	MS06-060 (October 2006)
CVE-2006-4694	Powerpoint	MS06-058 (October 2006)
CVE-2006-5994	Word	MS07-014 (February 2007)
CVE-2006-6456	Word	MS07-014 (February 2007)
CVE-2007-0515	Word	MS07-014 (February 2007)
CVE-2007-0671	Excel	MS07-015 (February 2007)
CVE-2007-0870	Word	MS07-024 (May 2007)
CVE-2008-0081	Excel	MS08-014 (March 2008)
CVE-2008-4841	Word	MS09-010 (April 2009)
CVE-2009-0238	Excel	MS09-009 (April 2009)
CVE-2009-0556	Powerpoint	MS09-017 (May 2009)

Short introduction to the OLESS format

- OLESS Header
- FAT FS
  - SectorNumbers
  - OLESS directory entries
- Data is divided into directories (storages) and files (streams)

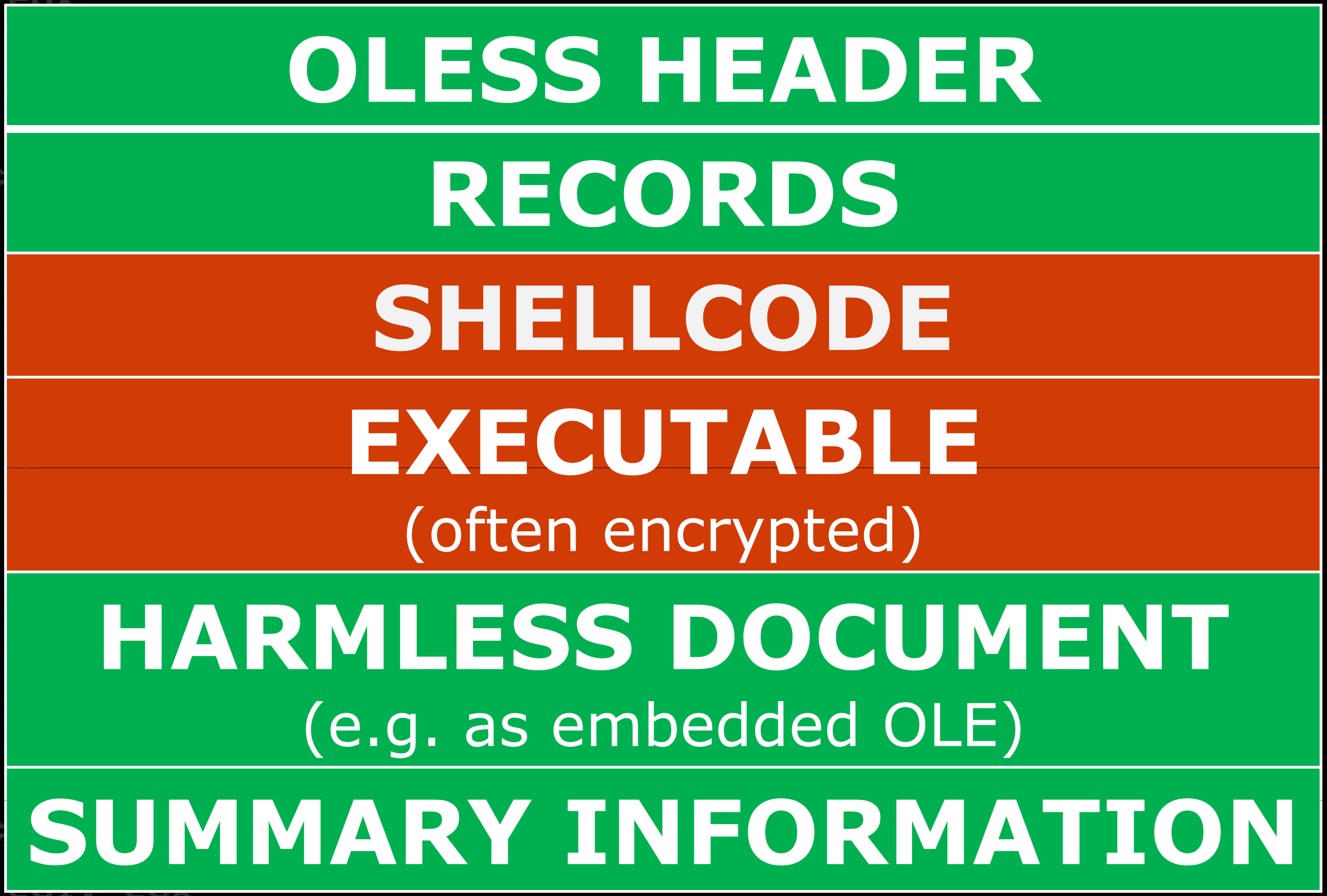
Short introduction to the OLESS format

- Depending on the application streams may contain
  - Macros
  - Graphics
  - Tables
  - Sounds
  - Animations
  - ….

Short introduction to the OLESS format

- Parsing can be done using the Win32 COM API
  - StgOpenStorage()
  - IStorage methods
  - IStream methods

Example of a malicious MS Office document structure



OLESS HEADER

RECORDS

SHELLCODE

EXECUTABLE
(often encrypted)

HARMLESS DOCUMENT
(e.g. as embedded OLE)

SUMMARY INFORMATION

Typical MS Office Shellcode behavior

- When a bug in a MS Office application gets triggered...
  - Shellcode executes
  - Finds itself by open file handles enumeration and file size checking
  - SetFilePointer to encrypted PE-File(s), decrypt, drop and execute
  - Drop harmless embedded MS Office document and start to look innocent

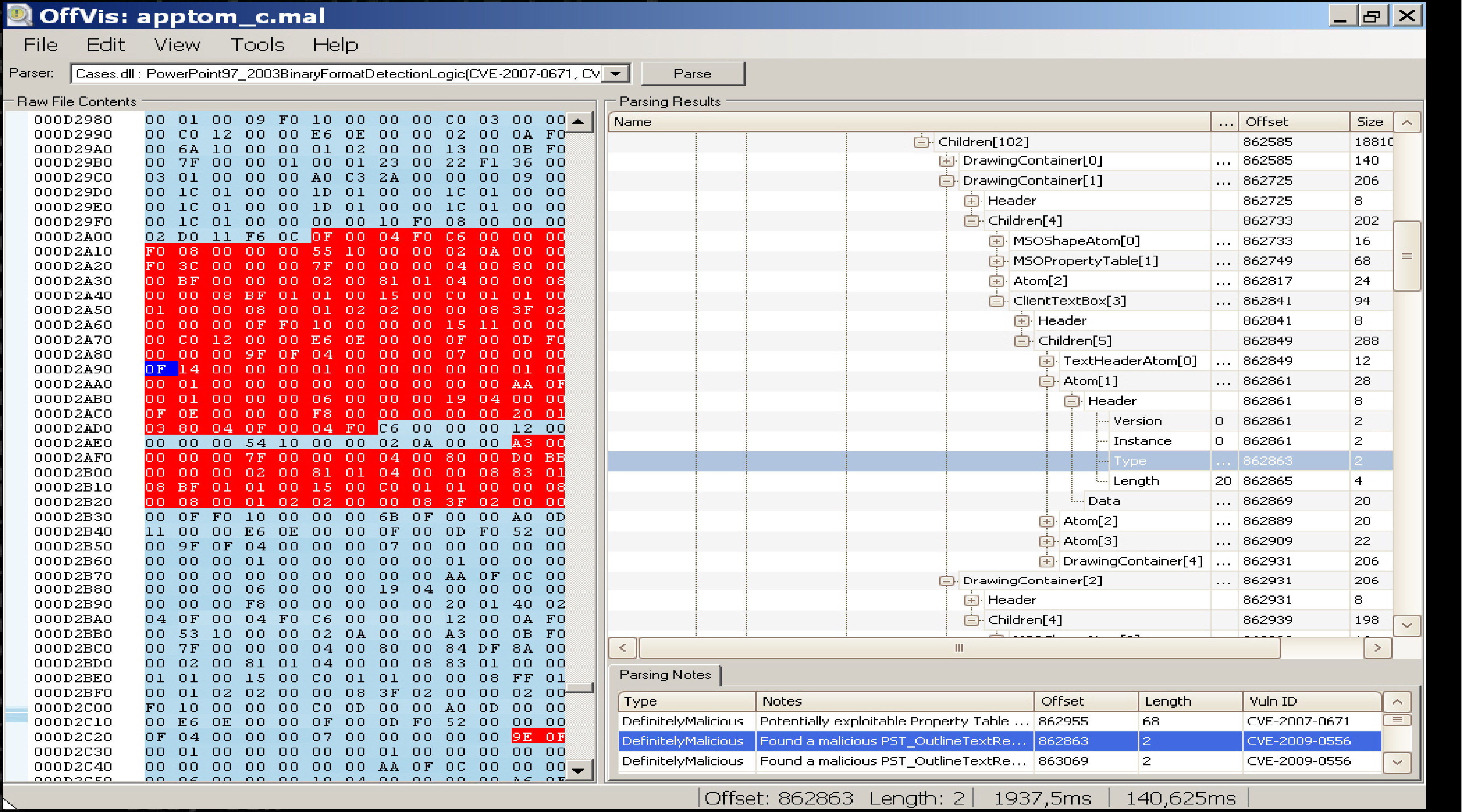
Status Quo to MS Office document analysis

- Not much public information about MS-Office malware analysis available
- Microsoft Office Binary File Format Specification (since Feb. 2008)
- Bruce Dang`s talk „Methods for Understanding Targeted Attacks with Office Documents”

Available tools for Ms Office analysis

- DFView (oldschool Microsoft OLE structure viewer)
- Officecat (signature based CLI utility)
- FlexHex Editor (OLE compound viewer)
- OffVis - (Office binary file format visualization tool)

OffVis in action



Parser: Cases.dll : PowerPoint97_2003BinaryFormatDetectionLogic(CVE-2007-0671, Cv

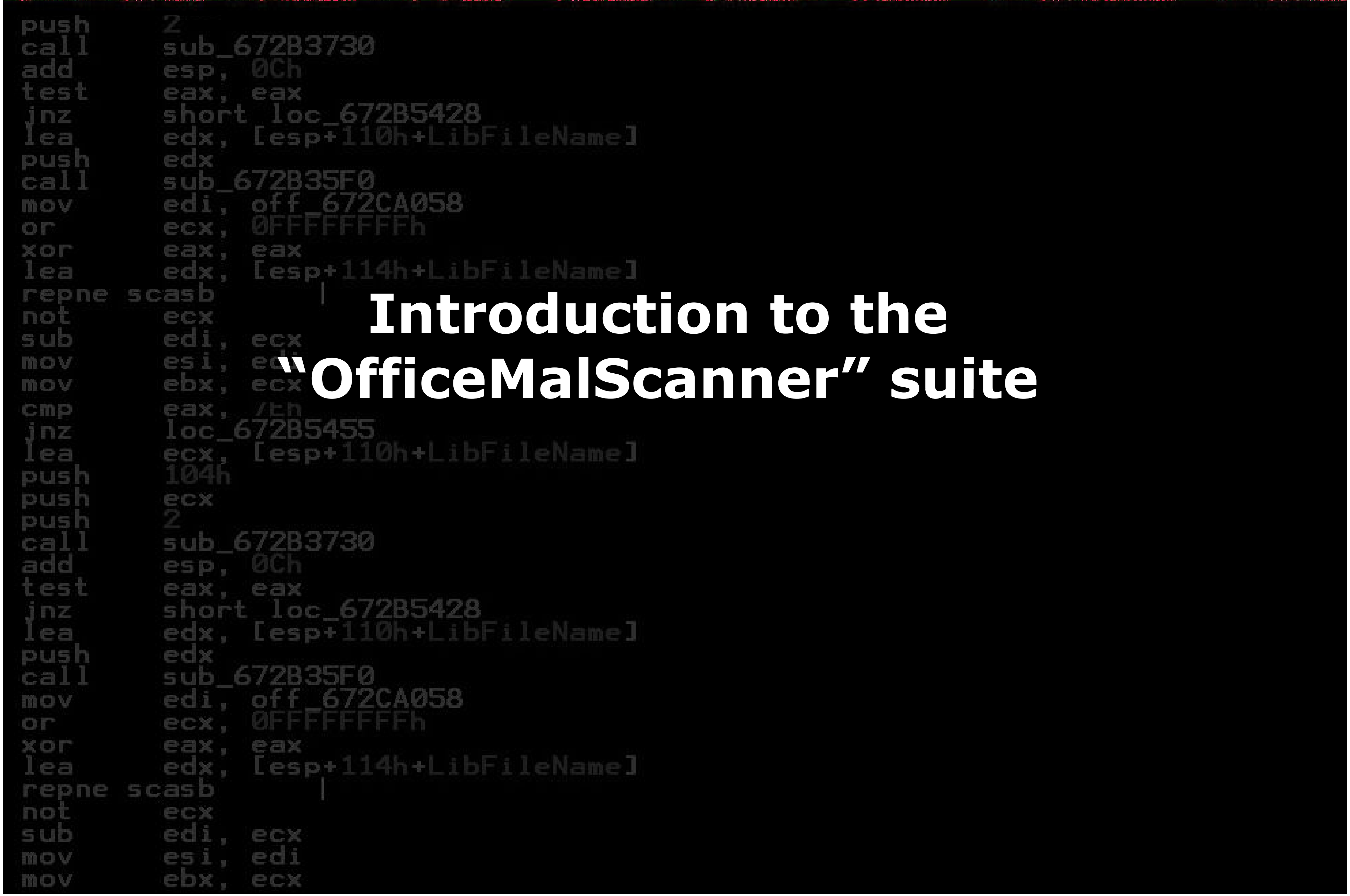
```
000D2980   00 01 00 09 F0 10 00 00 00 C0 03 00 00
000D2990   00 C0 12 00 00 E6 0E 00 00 02 00 0A F0
000D29A0   00 6A 10 00 00 01 02 00 00 13 00 0B F0
000D29B0   00 7F 00 00 01 00 01 23 00 22 F1 36 00
000D29C0   03 01 00 00 00 A0 C3 2A 00 00 00 09 00
000D29D0   00 1C 01 00 00 1D 01 00 00 1C 01 00 00
000D29E0   00 1C 01 00 00 1D 01 00 00 1C 01 00 00
000D29F0   00 1C 01 00 00 00 00 10 F0 08 00 00 00
000D2A00   02 D0 11 F6 0C 0F 00 04 F0 C6 00 00 00
000D2A10   F0 08 00 00 00 55 10 00 00 02 0A 00 00
000D2A20   F0 3C 00 00 00 7F 00 00 00 04 00 80 00
000D2A30   00 BF 00 00 00 02 00 81 01 04 00 00 08
000D2A40   00 00 08 BF 01 01 00 15 00 C0 01 01 00
000D2A50   01 00 00 08 00 01 02 02 00 00 08 3F 02
000D2A60   00 00 00 0F F0 10 00 00 00 15 11 00 00
000D2A70   00 C0 12 00 00 E6 0E 00 00 0F 00 0D F0
000D2A80   00 00 00 9F 0F 04 00 00 00 07 00 00 00
000D2A90   0F 14 00 00 00 01 00 00 00 00 00 01 00
000D2AA0   00 01 00 00 00 00 00 00 00 00 00 AA 0F
000D2AB0   00 01 00 00 00 06 00 00 00 19 04 00 00
000D2AC0   0F 0E 00 00 00 F8 00 00 00 00 00 20 01
000D2AD0   03 80 04 0F 00 04 F0 C6 00 00 00 12 00
000D2AE0   00 00 00 54 10 00 00 02 0A 00 00 A3 00
000D2AF0   00 00 00 7F 00 00 00 04 00 80 00 D0 BE
000D2B00   00 00 00 02 00 81 01 04 00 00 08 83 01
000D2B10   08 BF 01 01 00 15 00 C0 01 01 00 00 08
000D2B20   00 08 00 01 02 02 00 00 08 3F 02 00 00
000D2B30   00 0F F0 10 00 00 00 6B 0F 00 00 A0 0D
000D2B40   11 00 00 E6 0E 00 00 0F 00 0D F0 52 00
000D2B50   00 9F 0F 04 00 00 00 07 00 00 00 00 00
000D2B60   00 00 00 01 00 00 00 00 00 01 00 00 00
000D2B70   00 00 00 00 00 00 00 00 00 AA 0F 0C 00
000D2B80   00 00 00 06 00 00 00 19 04 00 00 00 00
000D2B90   00 00 00 F8 00 00 00 00 00 20 01 40 02
000D2BA0   04 0F 00 04 F0 C6 00 00 00 12 00 0A F0
000D2BB0   00 53 10 00 00 02 0A 00 00 A3 00 0B F0
000D2BC0   00 7F 00 00 00 04 00 80 00 84 DF 8A 00
000D2BD0   00 02 00 81 01 04 00 00 08 83 01 00 00
000D2BE0   01 01 00 15 00 C0 01 01 00 00 08 FF 01
000D2BF0   00 01 02 02 00 00 08 3F 02 00 00 02 00
000D2C00   F0 10 00 00 00 C0 0D 00 00 A0 0D 00 00
000D2C10   00 E6 0E 00 00 0F 00 0D F0 52 00 00 00
000D2C20   0F 04 00 00 00 07 00 00 00 00 00 9E 0F
000D2C30   00 01 00 00 00 00 00 01 00 00 00 00 00
000D2C40   00 00 00 00 00 00 00 AA 0F 0C 00 00 00
```

```
Name                                      ...   Offset   Size
Children[102]                                   862585   18810
  DrawingContainer[0]                     ...   862585   140
  DrawingContainer[1]                     ...   862725   206
    Header                                      862725   8
    Children[4]                                 862733   202
      MSOShapeAtom[0]                     ...   862733   16
      MSOPropertyTable[1]                 ...   862749   68
      Atom[2]                             ...   862817   24
      ClientTextBox[3]                    ...   862841   94
        Header                                  862841   8
        Children[5]                             862849   288
          TextHeaderAtom[0]               ...   862849   12
          Atom[1]                         ...   862861   28
            Header                              862861   8
              Version                     0     862861   2
              Instance                    0     862861   2
              Type                        ...   862863   2
              Length                      20    862865   4
            Data                          ...   862869   20
          Atom[2]                         ...   862889   20
          Atom[3]                         ...   862909   22
          DrawingContainer[4]             ...   862931   206
  DrawingContainer[2]                     ...   862931   206
    Header                                      862931   8
    Children[4]                                 862939   198
```

Type	Notes	Offset	Length	Vuln ID
DefinitelyMalicious	Potentially exploitable Property Table ...	862955	68	CVE-2007-0671
DefinitelyMalicious	Found a malicious PST_OutlineTextRe...	862863	2	CVE-2009-0556
DefinitelyMalicious	Found a malicious PST_OutlineTextRe...	863069	2	CVE-2009-0556

Offset: 862863 Length: 2 | 1937,5ms | 140,625ms

Introduction to the “OfficeMalScanner” suite



```
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
cmp     eax, 7Eh
jnz     loc_672B5455
lea     ecx, [esp+110h+LibFileName]
push    104h
push    ecx
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
```

OfficeMalScanner features

- OfficeMalScanner is a forensic tool for analysts to find malicious traces in MS Office documents.
- Features:
  - SCAN
  - BRUTE
  - DEBUG
  - INFO
  - INFLATE

SCAN mode (Shellcode scanner)

- GetEIP (4 Methods)

```
           CALL  NEXT
NEXT:      POP reg
------------------------------------------
           JMP [0xEB] 1ST
2ND:       POP reg
1ST:       CALL 2ND
------------------------------------------
           JMP [0xE9] 1ST
2ND:       POP reg
1ST:       CALL 2ND
------------------------------------------
           FLDZ
           FSTENV [esp-0ch]
           POP reg
```

SCAN mode (Shellcode scanner)

- Find Kernel32 base (3 methods)

```
MOV reg, DWORD PTR FS:[30h]
---------------------------------------------
XOR reg_a,reg_a
MOV reg_a(low-byte), 30h
MOV reg_b, fs:[reg_a]
---------------------------------------------
PUSH 30h
POP reg_a
MOV reg_b, FS:[reg_a]
```

- Find structured exception handling

```
MOV reg, DWORD PTR FS:[00h]
```

SCAN mode (Shellcode scanner)

- **API Hashing**

```
LOOP:   LODSB
        TEST    al, al
        JZ      short OK
        ROR     EDI, 0Dh   (or 07h)
        ADD     EDI, EAX
        JMP     short LOOP
 OK:    CMP     EDI, …
```

- **Indirect function call**

```
PUSH DWORD PTR [EBP+val]
CALL[EBP+val]
```

SCAN mode (Shellcode scanner)

- **Suspicious strings**
  - UrlDownloadToFile
  - GetTempPath
  - GetWindowsDirectory
  - GetSystemDirectory
  - WinExec
  - ShellExecute
  - IsBadReadPtr
  - IsBadWritePtr
  - CreateFile
  - CloseHandle
  - ReadFile
  - WriteFile
  - SetFilePointer
  - VirtualAlloc
  - GetProcAddr
  - LoadLibrary

SCAN mode (Shellcode scanner)

- Easy decryption trick

  LODS(x)

  XOR or ADD or SUB or ROL or ROR

  STOS(x)

- Embedded OLE Data (unencrypted)
  - Signature: \xD0\xCF\x11\xE0\xA1\xB1\x1a\xE1
  - Gets dumped to disk

SCAN mode (Shellcode scanner)

- **Function Prolog**

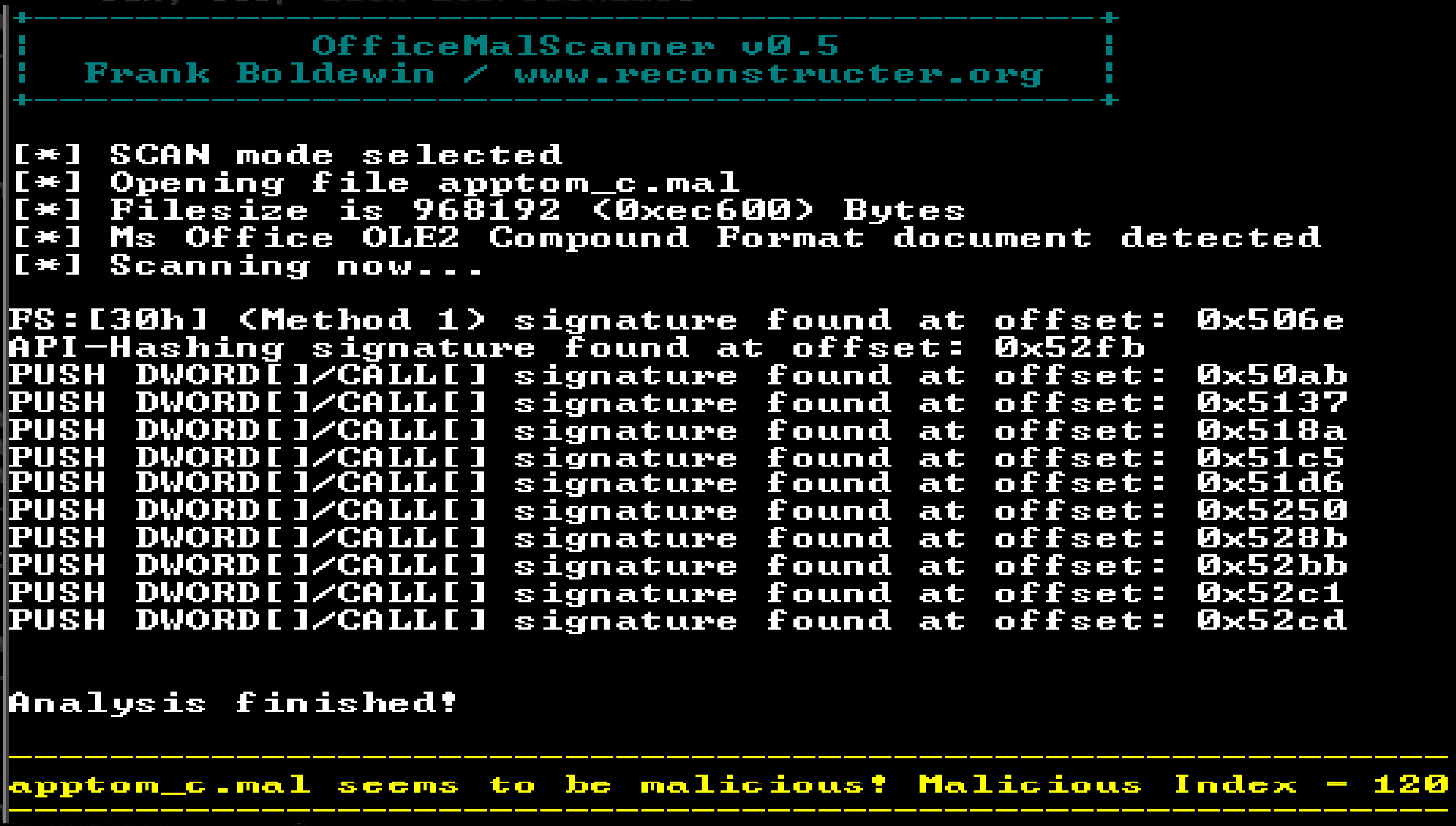
```
PUSH EBP
MOV EBP, ESP
SUB ESP, <value> or ADD ESP, <value>
```

- **PE-File Signature (unencrypted)**

```
Offset 0x0          == MZ
Offset 0x3c         == e_lfanew
Offset e_lfanew     == PE
```

Found PE-files are dumped to disk

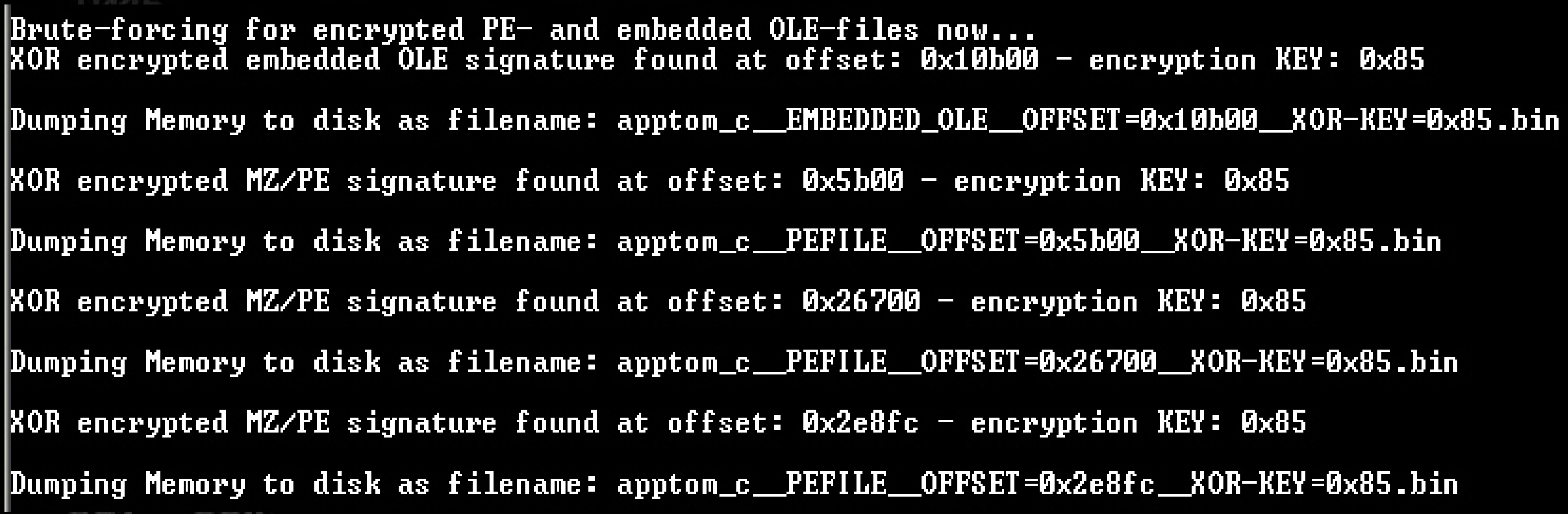
SCAN mode in action



```
+--------------------------------------------------------+
|               OfficeMalScanner v0.5                    |
|   Frank Boldewin / www.reconstructer.org               |
+--------------------------------------------------------+

[*] SCAN mode selected
[*] Opening file apptom_c.mal
[*] Filesize is 968192 (0xec600) Bytes
[*] Ms Office OLE2 Compound Format document detected
[*] Scanning now...

FS:[30h] (Method 1) signature found at offset: 0x506e
API-Hashing signature found at offset: 0x52fb
PUSH DWORD[]/CALL[] signature found at offset: 0x50ab
PUSH DWORD[]/CALL[] signature found at offset: 0x5137
PUSH DWORD[]/CALL[] signature found at offset: 0x518a
PUSH DWORD[]/CALL[] signature found at offset: 0x51c5
PUSH DWORD[]/CALL[] signature found at offset: 0x51d6
PUSH DWORD[]/CALL[] signature found at offset: 0x5250
PUSH DWORD[]/CALL[] signature found at offset: 0x528b
PUSH DWORD[]/CALL[] signature found at offset: 0x52bb
PUSH DWORD[]/CALL[] signature found at offset: 0x52c1
PUSH DWORD[]/CALL[] signature found at offset: 0x52cd


Analysis finished!

---------------------------------------------------------------------------
apptom_c.mal seems to be malicious! Malicious Index = 120
---------------------------------------------------------------------------
```

BRUTE mode

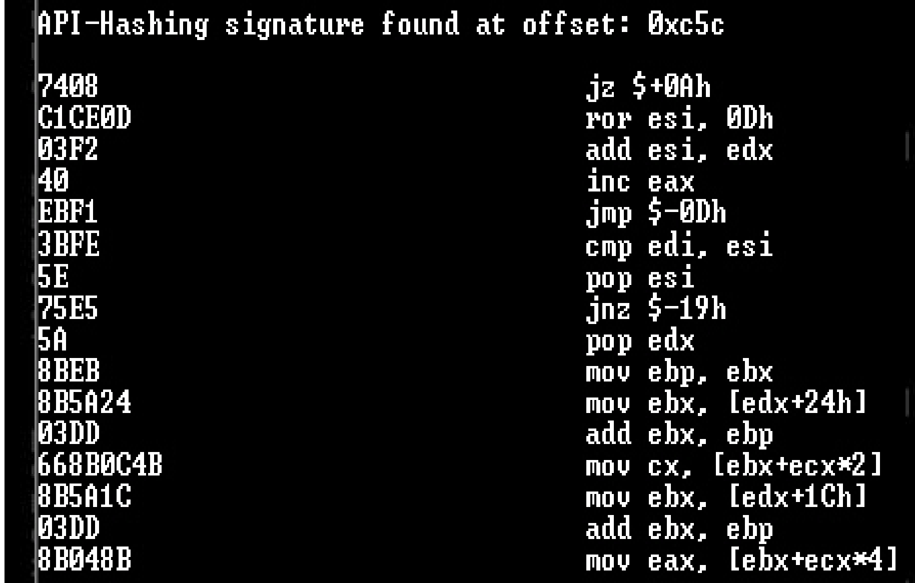
- Easy XOR + ADD 0x0 – 0xff buffer decryption
  - After decryption
    - Embedded OLE check
    - PE-file signature check
- Found files get dumped to disk



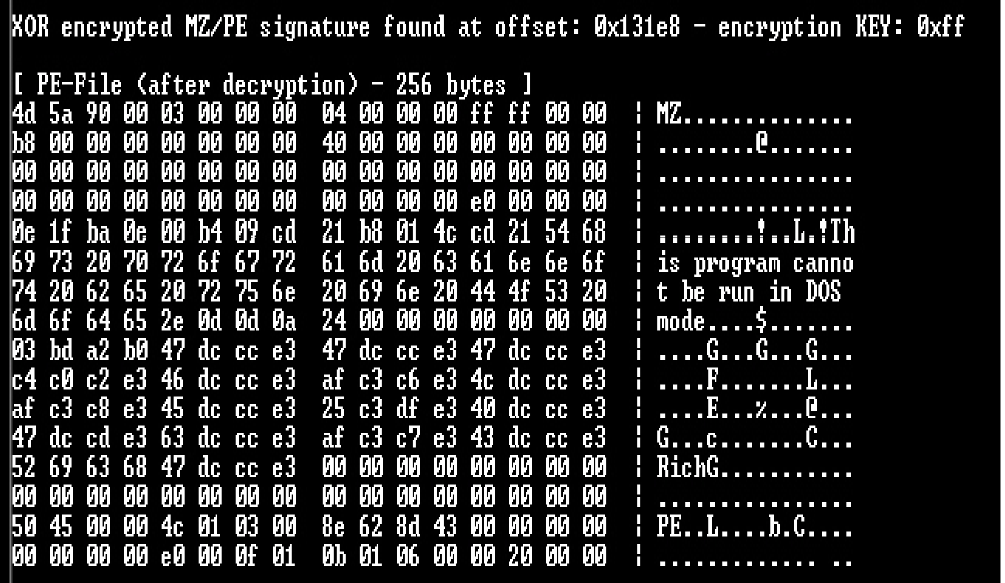
```
Brute-forcing for encrypted PE- and embedded OLE-files now...
XOR encrypted embedded OLE signature found at offset: 0x10b00 - encryption KEY: 0x85

Dumping Memory to disk as filename: apptom_c__EMBEDDED_OLE__OFFSET=0x10b00__XOR-KEY=0x85.bin

XOR encrypted MZ/PE signature found at offset: 0x5b00 - encryption KEY: 0x85

Dumping Memory to disk as filename: apptom_c__PEFILE__OFFSET=0x5b00__XOR-KEY=0x85.bin

XOR encrypted MZ/PE signature found at offset: 0x26700 - encryption KEY: 0x85

Dumping Memory to disk as filename: apptom_c__PEFILE__OFFSET=0x26700__XOR-KEY=0x85.bin

XOR encrypted MZ/PE signature found at offset: 0x2e8fc - encryption KEY: 0x85

Dumping Memory to disk as filename: apptom_c__PEFILE__OFFSET=0x2e8fc__XOR-KEY=0x85.bin
```

DEBUG mode

- The Debug mode displays:
  - Disassembly for detected code
  - Hexdata for detected strings and PE-files



```
API-Hashing signature found at offset: 0xc5c

7408                                 jz $+0Ah
C1CE0D                               ror esi, 0Dh
03F2                                 add esi, edx
40                                   inc eax
EBF1                                 jmp $-0Dh
3BFE                                 cmp edi, esi
5E                                   pop esi
75E5                                 jnz $-19h
5A                                   pop edx
8BEB                                 mov ebp, ebx
8B5A24                               mov ebx, [edx+24h]
03DD                                 add ebx, ebp
668B0C4B                             mov cx, [ebx+ecx*2]
8B5A1C                               mov ebx, [edx+1Ch]
03DD                                 add ebx, ebp
8B048B                               mov eax, [ebx+ecx*4]
```



```
XOR encrypted MZ/PE signature found at offset: 0x131e8 - encryption KEY: 0xff

[ PE-File (after decryption) - 256 bytes ]
4d 5a 90 00 03 00 00 00  04 00 00 00 ff ff 00 00  ¦ MZ..............
b8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ¦ ........@.......
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ¦ ................
00 00 00 00 00 00 00 00  00 00 00 00 e0 00 00 00  ¦ ................
0e 1f ba 0e 00 b4 09 cd  21 b8 01 4c cd 21 54 68  ¦ ........!..L.!Th
69 73 20 70 72 6f 67 72  61 6d 20 63 61 6e 6e 6f  ¦ is program canno
74 20 62 65 20 72 75 6e  20 69 6e 20 44 4f 53 20  ¦ t be run in DOS
6d 6f 64 65 2e 0d 0d 0a  24 00 00 00 00 00 00 00  ¦ mode....$.......
03 bd a2 b0 47 dc cc e3  47 dc cc e3 47 dc cc e3  ¦ ....G...G...G...
c4 c0 c2 e3 46 dc cc e3  af c3 c6 e3 4c dc cc e3  ¦ ....F.......L...
af c3 c8 e3 45 dc cc e3  25 c3 df e3 40 dc cc e3  ¦ ....E...%...@...
47 dc cd e3 63 dc cc e3  af c3 c7 e3 43 dc cc e3  ¦ G...c.......C...
52 69 63 68 47 dc cc e3  00 00 00 00 00 00 00 00  ¦ RichG...........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ¦ ................
50 45 00 00 4c 01 03 00  8e 62 8d 43 00 00 00 00  ¦ PE..L....b.C....
00 00 00 00 e0 00 0f 01  0b 01 06 00 00 20 00 00  ¦ ............. ..
```

Malicious index rating

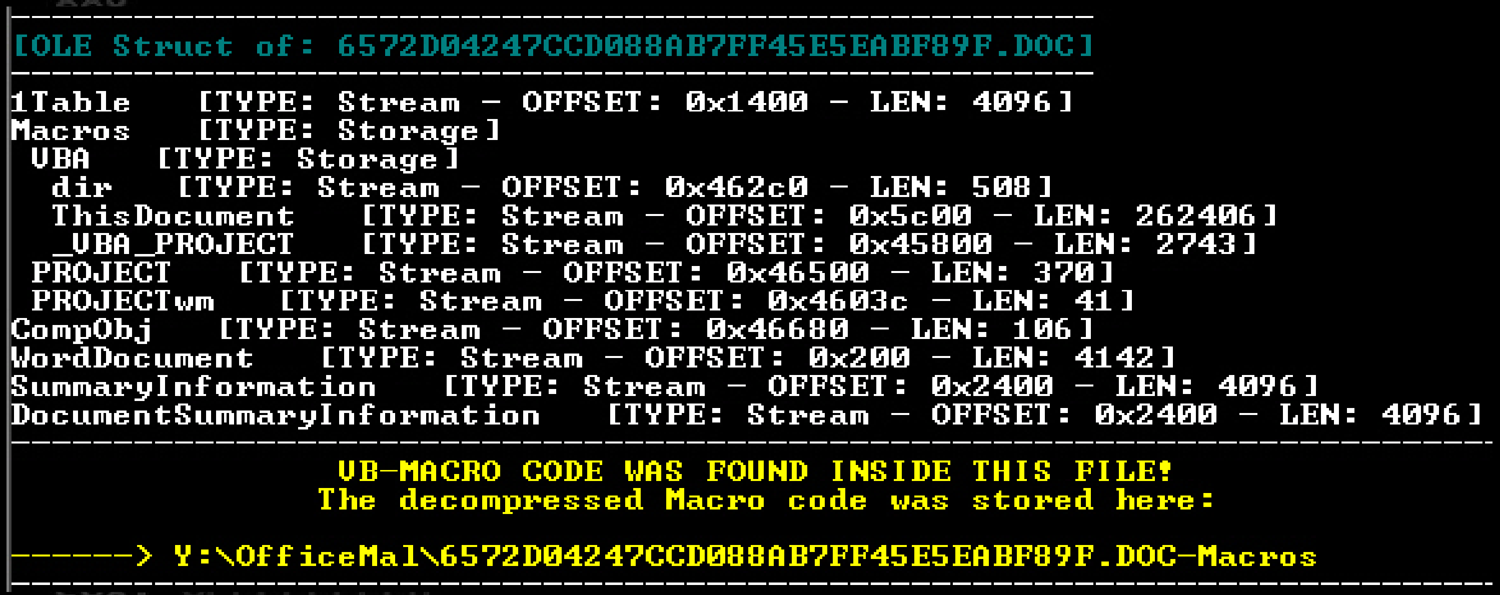
- The malicious index rating can be used for automated analysis as threshold.
- Every suspicious trace increases the malicious index counter depending on its hazard potential.

- Index scoring

Executables	: 20
Code	: 10
STRINGS	: 2
OLE	: 1

INFO mode

- The INFO mode dumps OLE structures, offsets, length and saves found VB-Macro code to disk

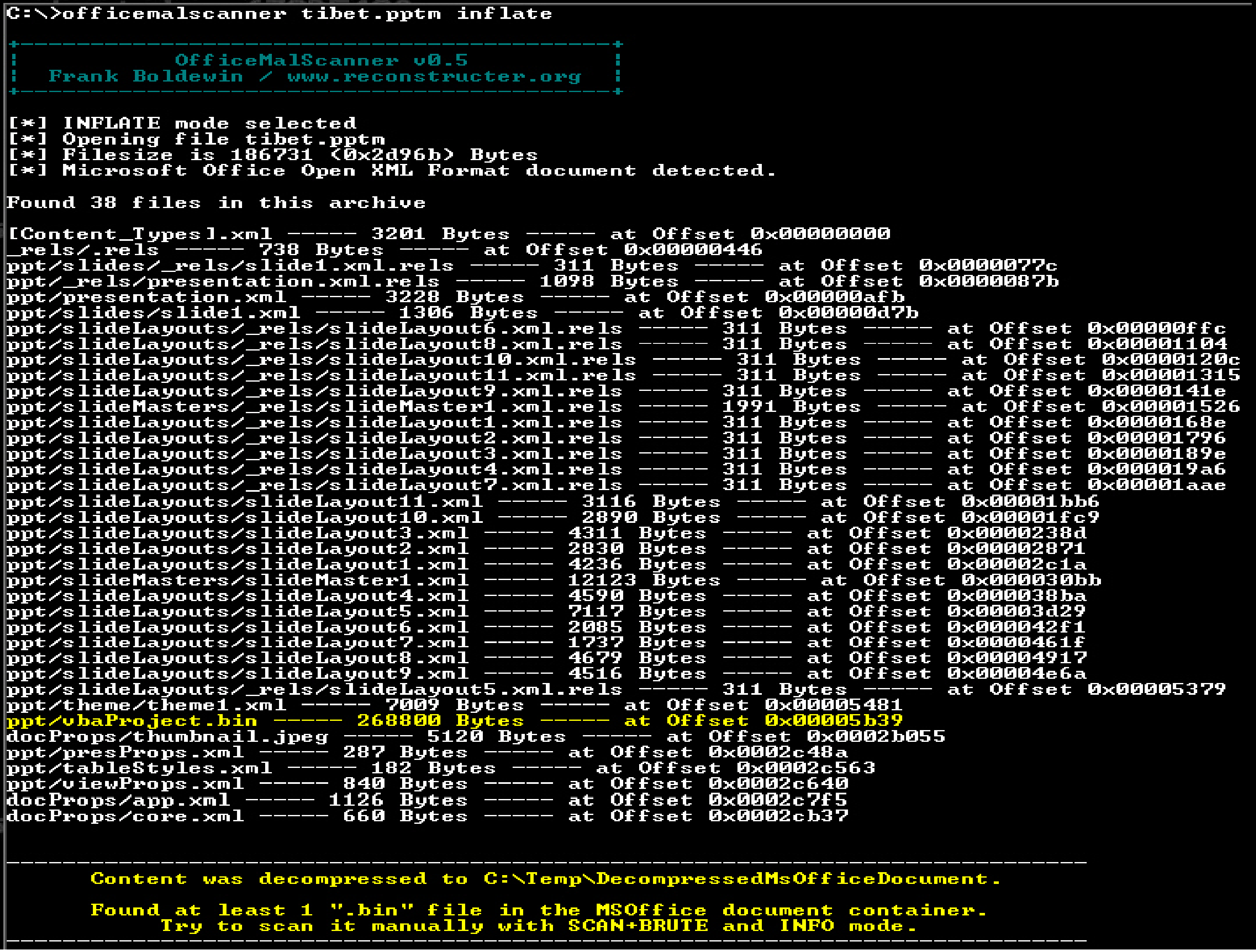


```
------------------------------------------------------------
[OLE Struct of: 6572D04247CCD088AB7FF45E5EABF89F.DOC]
------------------------------------------------------------
1Table    [TYPE: Stream - OFFSET: 0x1400 - LEN: 4096]
Macros    [TYPE: Storage]
 VBA    [TYPE: Storage]
  dir    [TYPE: Stream - OFFSET: 0x462c0 - LEN: 508]
  ThisDocument    [TYPE: Stream - OFFSET: 0x5c00 - LEN: 262406]
  _VBA_PROJECT    [TYPE: Stream - OFFSET: 0x45800 - LEN: 2743]
 PROJECT    [TYPE: Stream - OFFSET: 0x46500 - LEN: 370]
 PROJECTwm    [TYPE: Stream - OFFSET: 0x4603c - LEN: 41]
CompObj    [TYPE: Stream - OFFSET: 0x46680 - LEN: 106]
WordDocument    [TYPE: Stream - OFFSET: 0x200 - LEN: 4142]
SummaryInformation    [TYPE: Stream - OFFSET: 0x2400 - LEN: 4096]
DocumentSummaryInformation    [TYPE: Stream - OFFSET: 0x2400 - LEN: 4096]
----------------------------------------------------------------------------------
                VB-MACRO CODE WAS FOUND INSIDE THIS FILE!
               The decompressed Macro code was stored here:

------> Y:\OfficeMal\6572D04247CCD088AB7FF45E5EABF89F.DOC-Macros
----------------------------------------------------------------------------------
```

INFLATE mode

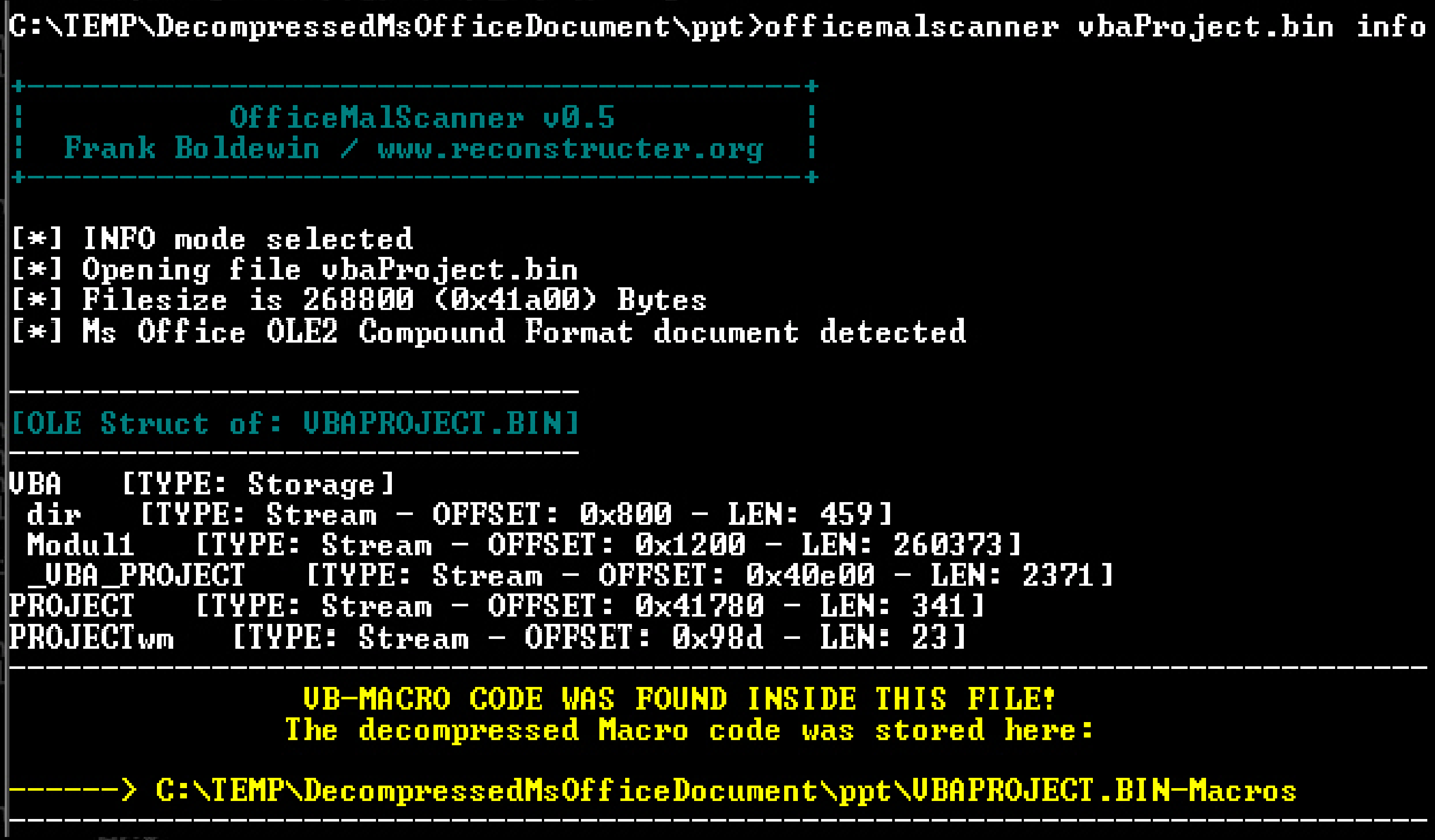
- Decompresses Ms Office 2007 documents, into a temp dir and marks potentially malicious files.
- Documents with macros included (docm, pptm and xlsm) contain .bin files, usually vbaproject.bin (Old MSOffice format)
- Such files could host malicious macro code and can extracted using the OfficeMalScanner INFO mode.

INFLATE mode – Usage STEP 1



```
C:\>officemalscanner tibet.pptm inflate

+------------------------------------------+
|           OfficeMalScanner v0.5           |
|  Frank Boldewin / www.reconstructer.org   |
+------------------------------------------+

[*] INFLATE mode selected
[*] Opening file tibet.pptm
[*] Filesize is 186731 (0x2d96b) Bytes
[*] Microsoft Office Open XML Format document detected.

Found 38 files in this archive

[Content_Types].xml ------ 3201 Bytes ------ at Offset 0x00000000
_rels/.rels ------ 738 Bytes ------ at Offset 0x00000446
ppt/slides/_rels/slide1.xml.rels ------ 311 Bytes ------ at Offset 0x0000077c
ppt/_rels/presentation.xml.rels ------ 1098 Bytes ------ at Offset 0x0000087b
ppt/presentation.xml ------ 3228 Bytes ------ at Offset 0x00000afb
ppt/slides/slide1.xml ------ 1306 Bytes ------ at Offset 0x00000d7b
ppt/slideLayouts/_rels/slideLayout6.xml.rels ------ 311 Bytes ------ at Offset 0x00000ffc
ppt/slideLayouts/_rels/slideLayout8.xml.rels ------ 311 Bytes ------ at Offset 0x00001104
ppt/slideLayouts/_rels/slideLayout10.xml.rels ------ 311 Bytes ------ at Offset 0x0000120c
ppt/slideLayouts/_rels/slideLayout11.xml.rels ------ 311 Bytes ------ at Offset 0x00001315
ppt/slideLayouts/_rels/slideLayout9.xml.rels ------ 311 Bytes ------ at Offset 0x0000141e
ppt/slideMasters/_rels/slideMaster1.xml.rels ------ 1991 Bytes ------ at Offset 0x00001526
ppt/slideLayouts/_rels/slideLayout1.xml.rels ------ 311 Bytes ------ at Offset 0x0000168e
ppt/slideLayouts/_rels/slideLayout2.xml.rels ------ 311 Bytes ------ at Offset 0x00001796
ppt/slideLayouts/_rels/slideLayout3.xml.rels ------ 311 Bytes ------ at Offset 0x0000189e
ppt/slideLayouts/_rels/slideLayout4.xml.rels ------ 311 Bytes ------ at Offset 0x000019a6
ppt/slideLayouts/_rels/slideLayout7.xml.rels ------ 311 Bytes ------ at Offset 0x00001aae
ppt/slideLayouts/slideLayout11.xml ------ 3116 Bytes ------ at Offset 0x00001bb6
ppt/slideLayouts/slideLayout10.xml ------ 2890 Bytes ------ at Offset 0x00001fc9
ppt/slideLayouts/slideLayout3.xml ------ 4311 Bytes ------ at Offset 0x0000238d
ppt/slideLayouts/slideLayout2.xml ------ 2830 Bytes ------ at Offset 0x00002871
ppt/slideLayouts/slideLayout1.xml ------ 4236 Bytes ------ at Offset 0x00002c1a
ppt/slideMasters/slideMaster1.xml ------ 12123 Bytes ------ at Offset 0x000030bb
ppt/slideLayouts/slideLayout4.xml ------ 4590 Bytes ------ at Offset 0x000038ba
ppt/slideLayouts/slideLayout5.xml ------ 7117 Bytes ------ at Offset 0x00003d29
ppt/slideLayouts/slideLayout6.xml ------ 2085 Bytes ------ at Offset 0x000042f1
ppt/slideLayouts/slideLayout7.xml ------ 1737 Bytes ------ at Offset 0x0000461f
ppt/slideLayouts/slideLayout8.xml ------ 4679 Bytes ------ at Offset 0x00004917
ppt/slideLayouts/slideLayout9.xml ------ 4516 Bytes ------ at Offset 0x00004e6a
ppt/slideLayouts/_rels/slideLayout5.xml.rels ------ 311 Bytes ------ at Offset 0x00005379
ppt/theme/theme1.xml ------ 7009 Bytes ------ at Offset 0x00005481
ppt/vbaProject.bin ------ 268800 Bytes ------ at Offset 0x00005b39
docProps/thumbnail.jpeg ------ 5120 Bytes ------ at Offset 0x0002b055
ppt/presProps.xml ------ 287 Bytes ------ at Offset 0x0002c48a
ppt/tableStyles.xml ------ 182 Bytes ------ at Offset 0x0002c563
ppt/viewProps.xml ------ 840 Bytes ------ at Offset 0x0002c640
docProps/app.xml ------ 1126 Bytes ------ at Offset 0x0002c7f5
docProps/core.xml ------ 660 Bytes ------ at Offset 0x0002cb37


-----------------------------------------------------------------------------
      Content was decompressed to C:\Temp\DecompressedMsOfficeDocument.

      Found at least 1 ".bin" file in the MSOffice document container.
           Try to scan it manually with SCAN+BRUTE and INFO mode.
-----------------------------------------------------------------------------
```

INFLATE mode – Usage STEP 2

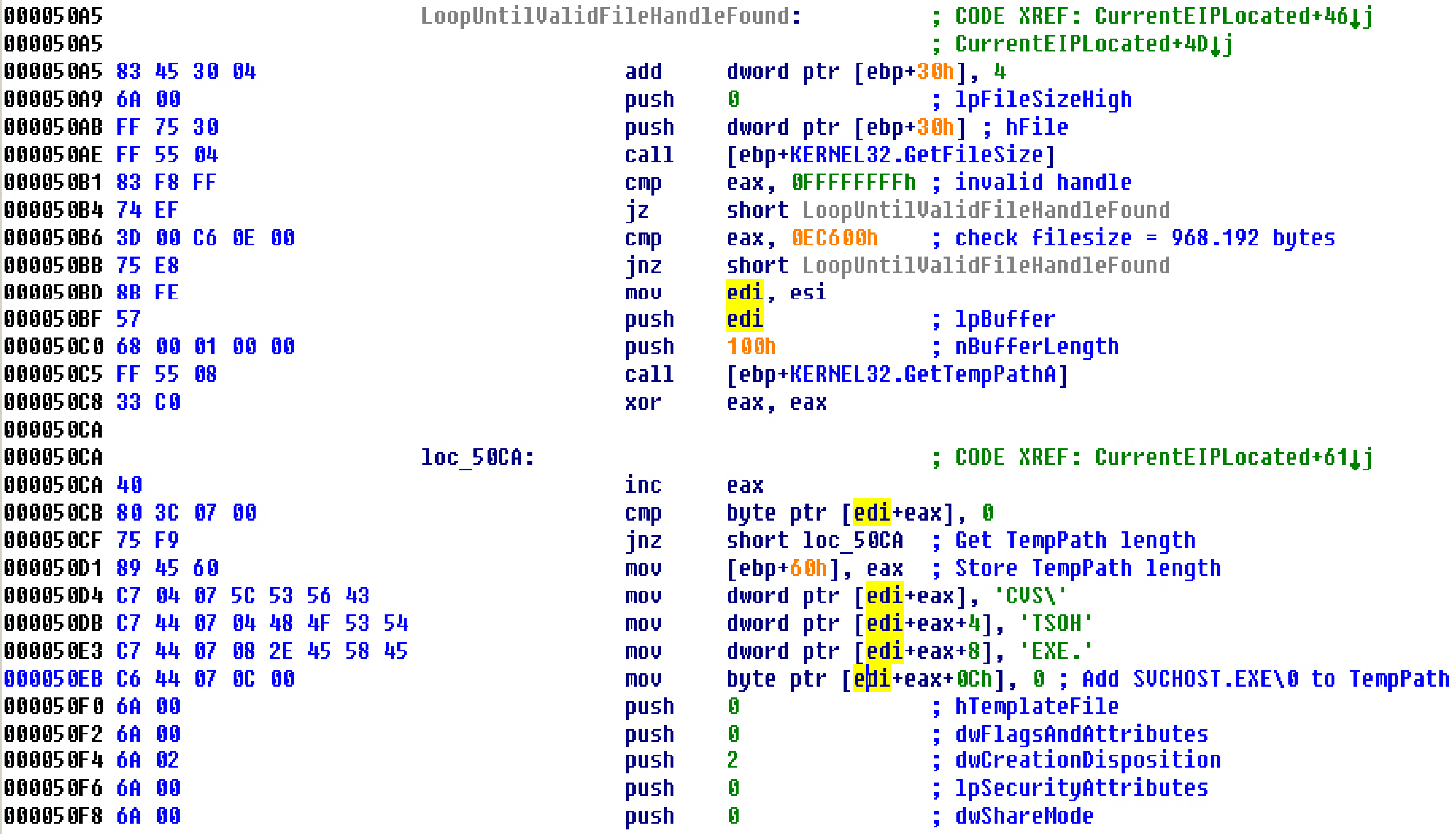


```
C:\TEMP\DecompressedMsOfficeDocument\ppt>officemalscanner vbaProject.bin info

+------------------------------------------+
|           OfficeMalScanner v0.5           |
|  Frank Boldewin / www.reconstructer.org   |
+------------------------------------------+

[*] INFO mode selected
[*] Opening file vbaProject.bin
[*] Filesize is 268800 (0x41a00) Bytes
[*] Ms Office OLE2 Compound Format document detected

-------------------------------
[OLE Struct of: VBAPROJECT.BIN]
-------------------------------
VBA   [TYPE: Storage]
 dir   [TYPE: Stream - OFFSET: 0x800 - LEN: 459]
 Modul1   [TYPE: Stream - OFFSET: 0x1200 - LEN: 260373]
 _VBA_PROJECT   [TYPE: Stream - OFFSET: 0x40e00 - LEN: 2371]
PROJECT   [TYPE: Stream - OFFSET: 0x41780 - LEN: 341]
PROJECTwm   [TYPE: Stream - OFFSET: 0x98d - LEN: 23]
-----------------------------------------------------------------------------
                VB-MACRO CODE WAS FOUND INSIDE THIS FILE!
               The decompressed Macro code was stored here:

------> C:\TEMP\DecompressedMsOfficeDocument\ppt\VBAPROJECT.BIN-Macros
-----------------------------------------------------------------------------
```

MalHost-Setup
A shellcode runtime environment

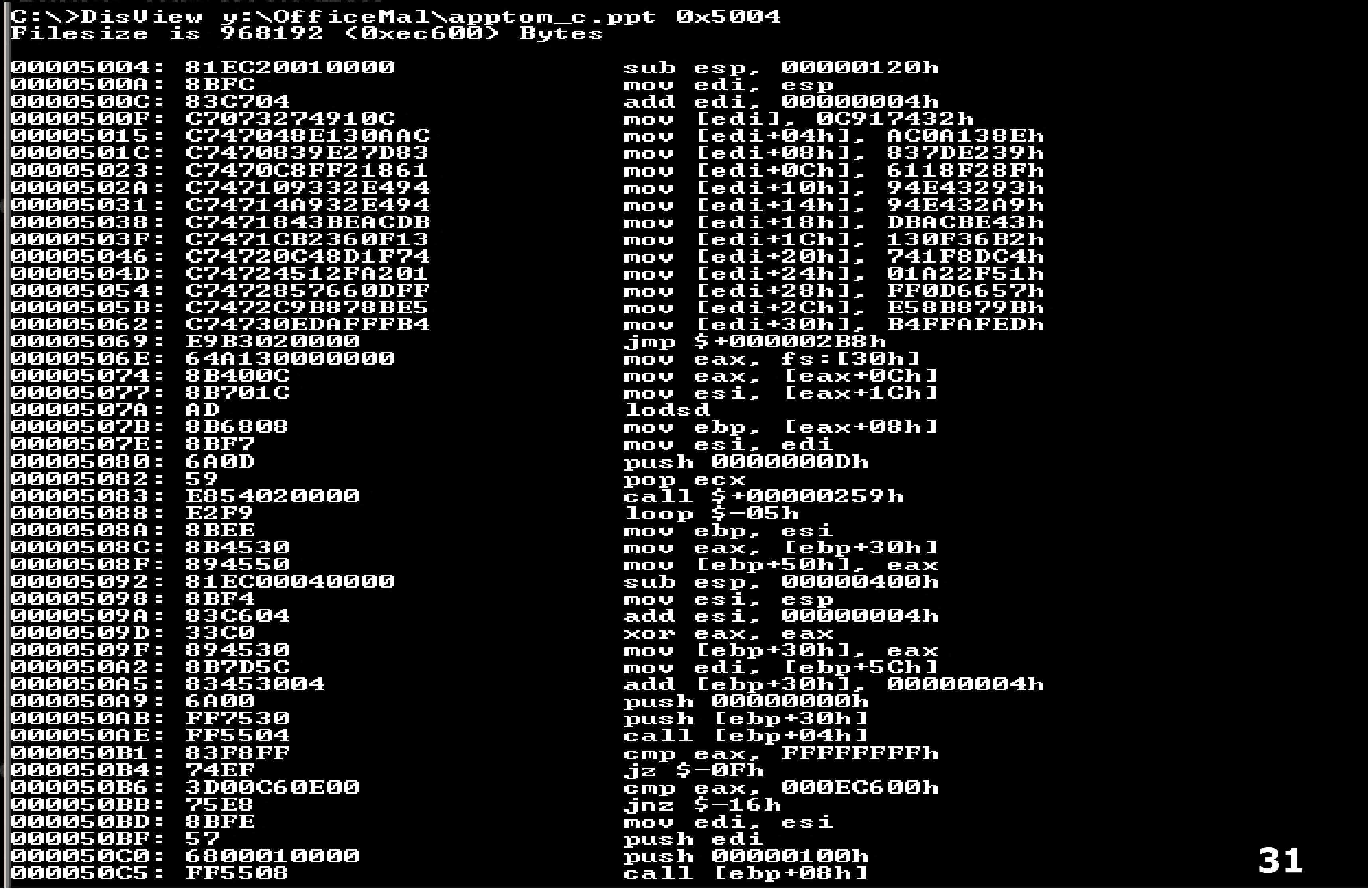
```
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
cmp     eax, 7Eh
jnz     loc_672B5455
lea     ecx, [esp+110h+LibFileName]
push    104h
push    ecx
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
```

MalHost-Setup – Typical shellcode requirements illustrated



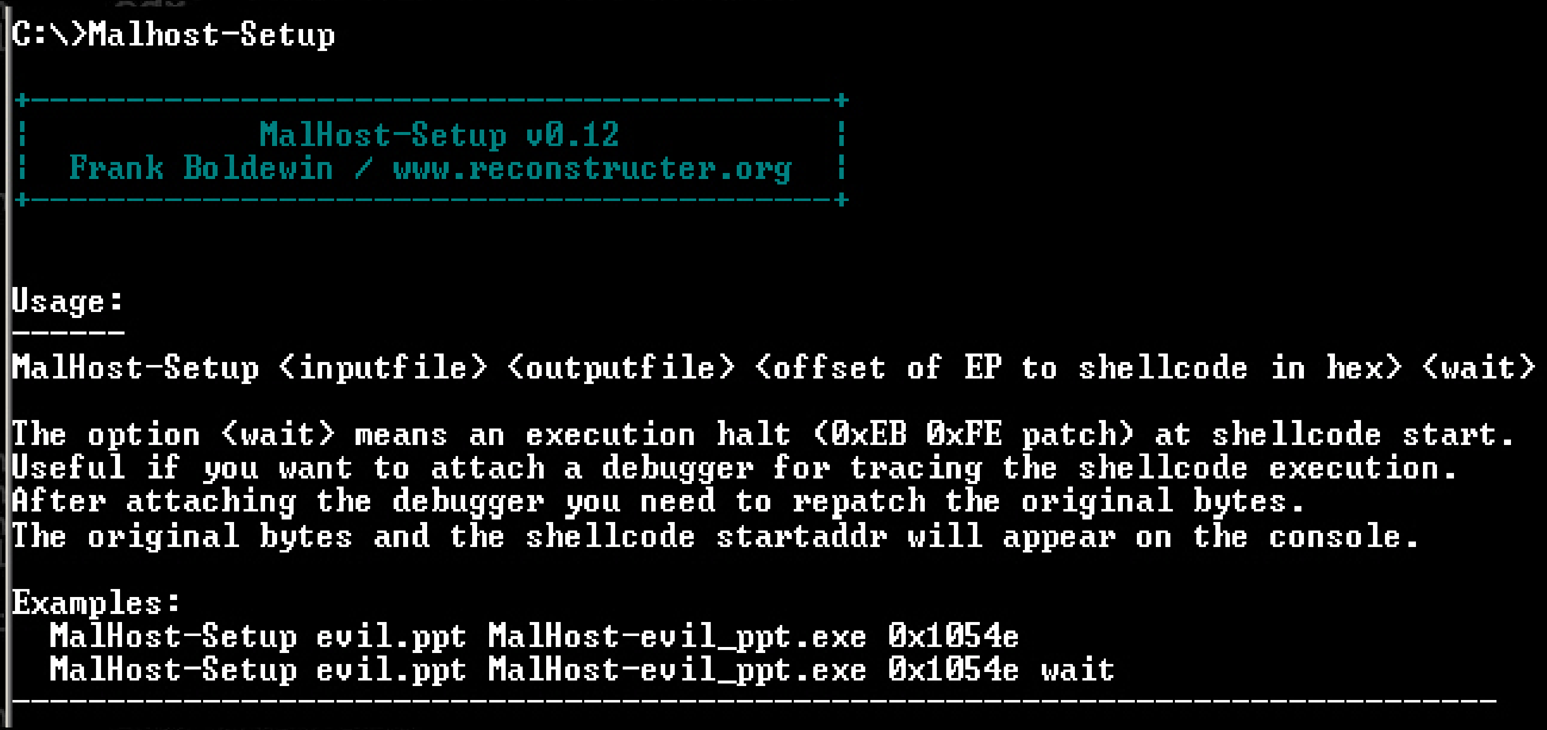
```
000050A5                       LoopUntilValidFileHandleFound:            ; CODE XREF: CurrentEIPLocated+46↓j
000050A5                                                                 ; CurrentEIPLocated+4D↓j
000050A5 83 45 30 04                   add     dword ptr [ebp+30h], 4
000050A9 6A 00                         push    0                ; lpFileSizeHigh
000050AB FF 75 30                      push    dword ptr [ebp+30h] ; hFile
000050AE FF 55 04                      call    [ebp+KERNEL32.GetFileSize]
000050B1 83 F8 FF                      cmp     eax, 0FFFFFFFFh ; invalid handle
000050B4 74 EF                         jz      short LoopUntilValidFileHandleFound
000050B6 3D 00 C6 0E 00                cmp     eax, 0EC600h     ; check filesize = 968.192 bytes
000050BB 75 E8                         jnz     short LoopUntilValidFileHandleFound
000050BD 8B FE                         mov     edi, esi
000050BF 57                            push    edi              ; lpBuffer
000050C0 68 00 01 00 00                push    100h             ; nBufferLength
000050C5 FF 55 08                      call    [ebp+KERNEL32.GetTempPathA]
000050C8 33 C0                         xor     eax, eax
000050CA
000050CA                       loc_50CA:                                 ; CODE XREF: CurrentEIPLocated+61↓j
000050CA 40                            inc     eax
000050CB 80 3C 07 00                   cmp     byte ptr [edi+eax], 0
000050CF 75 F9                         jnz     short loc_50CA   ; Get TempPath length
000050D1 89 45 60                      mov     [ebp+60h], eax   ; Store TempPath length
000050D4 C7 04 07 5C 53 56 43          mov     dword ptr [edi+eax], 'CVS\'
000050DB C7 44 07 04 48 4F 53 54       mov     dword ptr [edi+eax+4], 'TSOH'
000050E3 C7 44 07 08 2E 45 58 45       mov     dword ptr [edi+eax+8], 'EXE.'
000050EB C6 44 07 0C 00                mov     byte ptr [edi+eax+0Ch], 0 ; Add SVCHOST.EXE\0 to TempPath
000050F0 6A 00                         push    0                ; hTemplateFile
000050F2 6A 00                         push    0                ; dwFlagsAndAttributes
000050F4 6A 02                         push    2                ; dwCreationDisposition
000050F6 6A 00                         push    0                ; lpSecurityAttributes
000050F8 6A 00                         push    0                ; dwShareMode
```

MalHost-Setup – Finding the shellcode-start with DisView



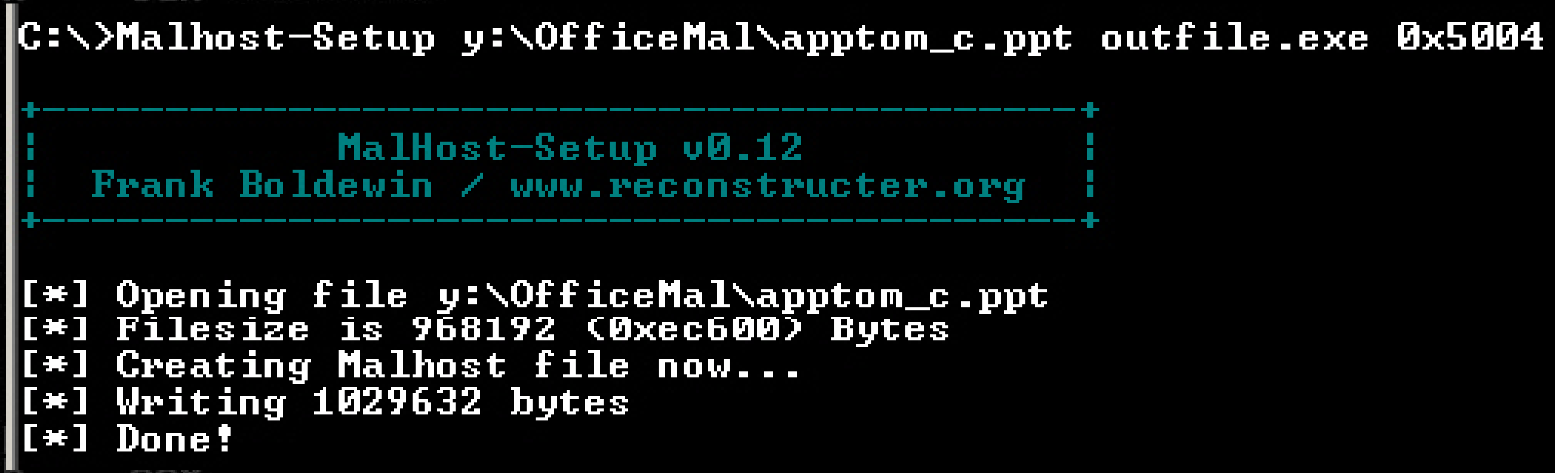
```
C:\>DisView y:\OfficeMal\apptom_c.ppt 0x5004
Filesize is 968192 (0xec600) Bytes

00005004: 81EC20010000             sub esp, 00000120h
0000500A: 8BFC                     mov edi, esp
0000500C: 83C704                   add edi, 00000004h
0000500F: C7073274910C             mov [edi], 0C917432h
00005015: C747048E130AAC           mov [edi+04h], AC0A138Eh
0000501C: C7470839E27D83           mov [edi+08h], 837DE239h
00005023: C7470C8FF21861           mov [edi+0Ch], 6118F28Fh
0000502A: C747109332E494           mov [edi+10h], 94E43293h
00005031: C74714A932E494           mov [edi+14h], 94E432A9h
00005038: C7471843BEACDB           mov [edi+18h], DBACBE43h
0000503F: C7471CB2360F13           mov [edi+1Ch], 130F36B2h
00005046: C74720C48D1F74           mov [edi+20h], 741F8DC4h
0000504D: C74724512FA201           mov [edi+24h], 01A22F51h
00005054: C7472857660DFF           mov [edi+28h], FF0D6657h
0000505B: C7472C9B878BE5           mov [edi+2Ch], E58B879Bh
00005062: C74730EDAFFFB4           mov [edi+30h], B4FFAFEDh
00005069: E9B3020000               jmp $+000002B8h
0000506E: 64A130000000             mov eax, fs:[30h]
00005074: 8B400C                   mov eax, [eax+0Ch]
00005077: 8B701C                   mov esi, [eax+1Ch]
0000507A: AD                       lodsd
0000507B: 8B6808                   mov ebp, [eax+08h]
0000507E: 8BF7                     mov esi, edi
00005080: 6A0D                     push 0000000Dh
00005082: 59                       pop ecx
00005083: E854020000               call $+00000259h
00005088: E2F9                     loop $-05h
0000508A: 8BEE                     mov ebp, esi
0000508C: 8B4530                   mov eax, [ebp+30h]
0000508F: 894550                   mov [ebp+50h], eax
00005092: 81EC00040000             sub esp, 00000400h
00005098: 8BF4                     mov esi, esp
0000509A: 83C604                   add esi, 00000004h
0000509D: 33C0                     xor eax, eax
0000509F: 894530                   mov [ebp+30h], eax
000050A2: 8B7D5C                   mov edi, [ebp+5Ch]
000050A5: 83453004                 add [ebp+30h], 00000004h
000050A9: 6A00                     push 00000000h
000050AB: FF7530                   push [ebp+30h]
000050AE: FF5504                   call [ebp+04h]
000050B1: 83F8FF                   cmp eax, FFFFFFFFh
000050B4: 74EF                     jz $-0Fh
000050B6: 3D00C60E00               cmp eax, 000EC600h
000050BB: 75E8                     jnz $-16h
000050BD: 8BFE                     mov edi, esi
000050BF: 57                       push edi
000050C0: 6800010000               push 00000100h
000050C5: FF5508                   call [ebp+08h]
```

MalHost-Setup – Help screen



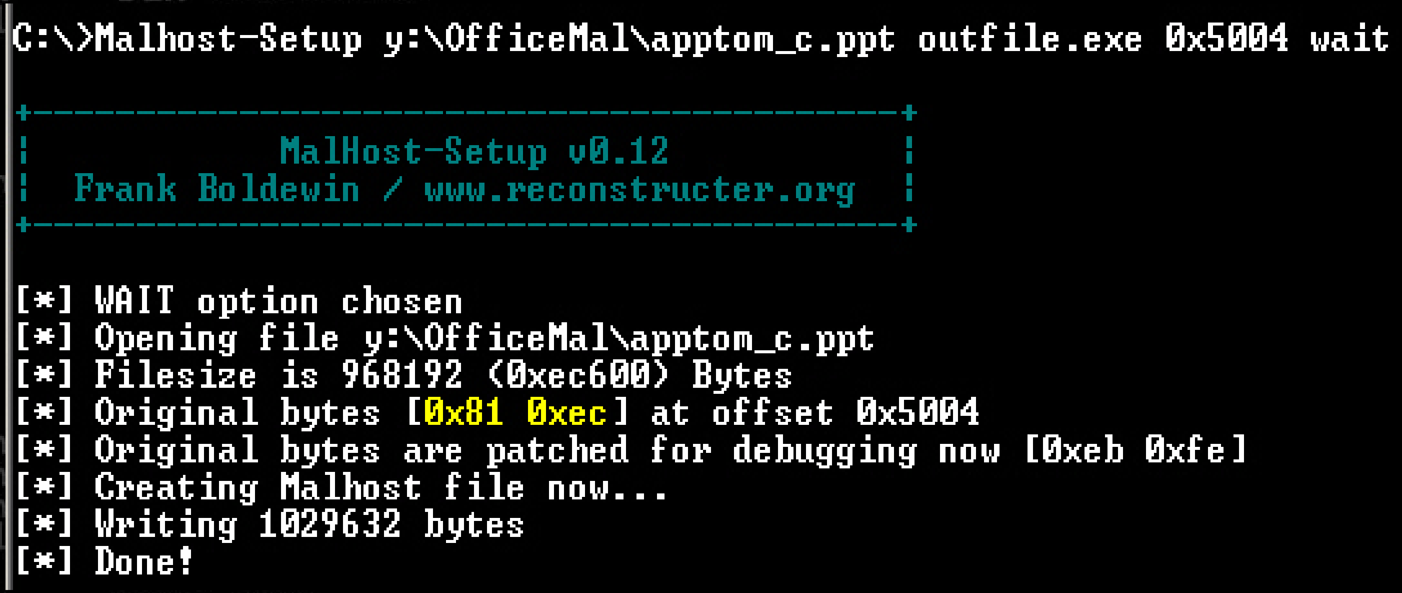
```
C:\>Malhost-Setup

+-------------------------------------------+
¦             MalHost-Setup v0.12           ¦
¦  Frank Boldewin / www.reconstructer.org   ¦
+-------------------------------------------+


Usage:
------
MalHost-Setup <inputfile> <outputfile> <offset of EP to shellcode in hex> <wait>

The option <wait> means an execution halt (0xEB 0xFE patch) at shellcode start.
Useful if you want to attach a debugger for tracing the shellcode execution.
After attaching the debugger you need to repatch the original bytes.
The original bytes and the shellcode startaddr will appear on the console.

Examples:
  MalHost-Setup evil.ppt MalHost-evil_ppt.exe 0x1054e
  MalHost-Setup evil.ppt MalHost-evil_ppt.exe 0x1054e wait
-------------------------------------------------------------------------------
```

MalHost-Setup – Configuration (unattended mode)



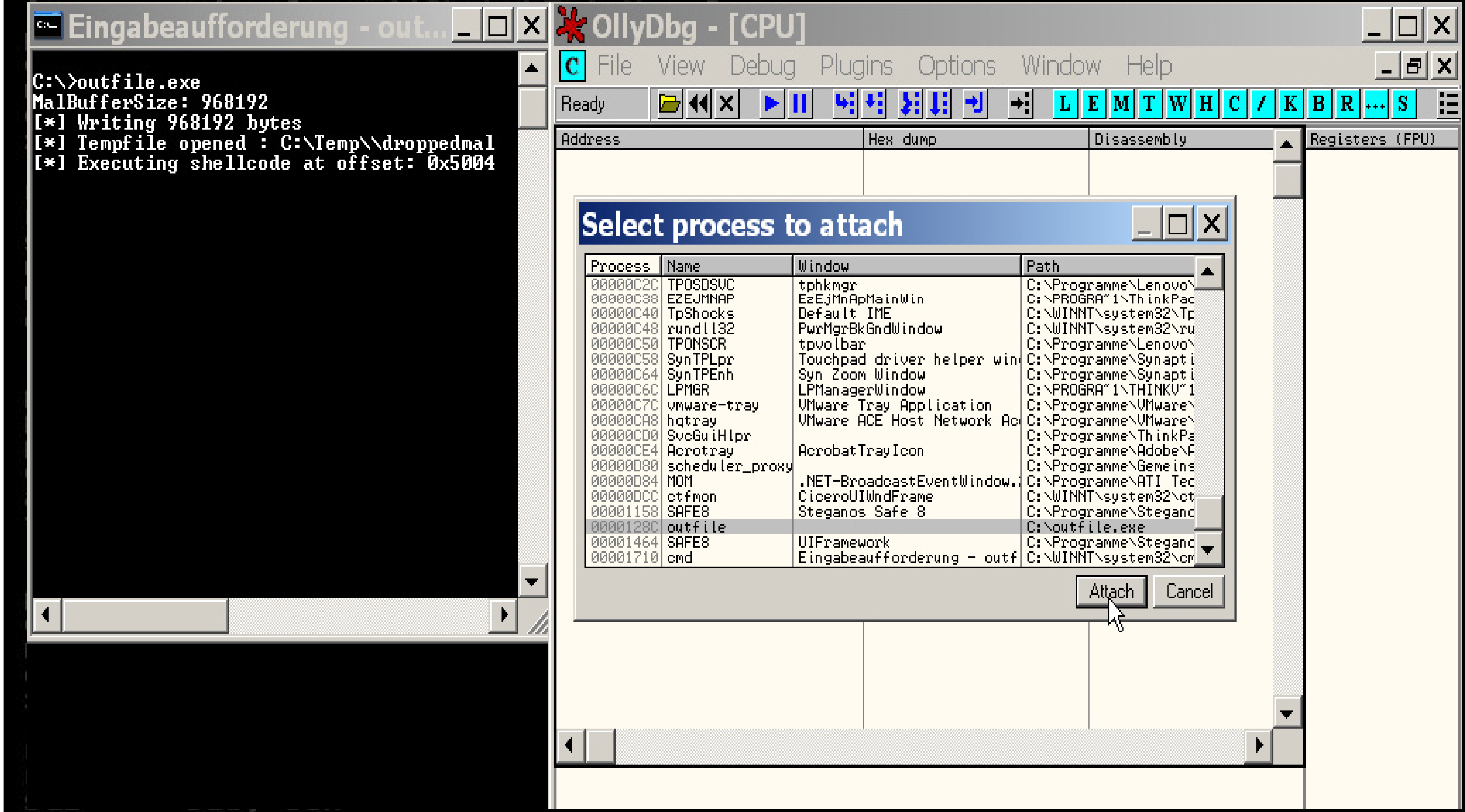
```
C:\>Malhost-Setup y:\OfficeMal\apptom_c.ppt outfile.exe 0x5004

+-------------------------------------------+
|              MalHost-Setup v0.12          |
|  Frank Boldewin / www.reconstructer.org   |
+-------------------------------------------+

[*] Opening file y:\OfficeMal\apptom_c.ppt
[*] Filesize is 968192 (0xec600) Bytes
[*] Creating Malhost file now...
[*] Writing 1029632 bytes
[*] Done!
```

MalHost-Setup – Configuration – (debug mode)



```
C:\>Malhost-Setup y:\OfficeMal\apptom_c.ppt outfile.exe 0x5004 wait

+------------------------------------------+
|              MalHost-Setup v0.12          |
|  Frank Boldewin / www.reconstructer.org   |
+------------------------------------------+

[*] WAIT option chosen
[*] Opening file y:\OfficeMal\apptom_c.ppt
[*] Filesize is 968192 (0xec600) Bytes
[*] Original bytes [0x81 0xec] at offset 0x5004
[*] Original bytes are patched for debugging now [0xeb 0xfe]
[*] Creating Malhost file now...
[*] Writing 1029632 bytes
[*] Done!
```

MalHost-Setup – Debugging



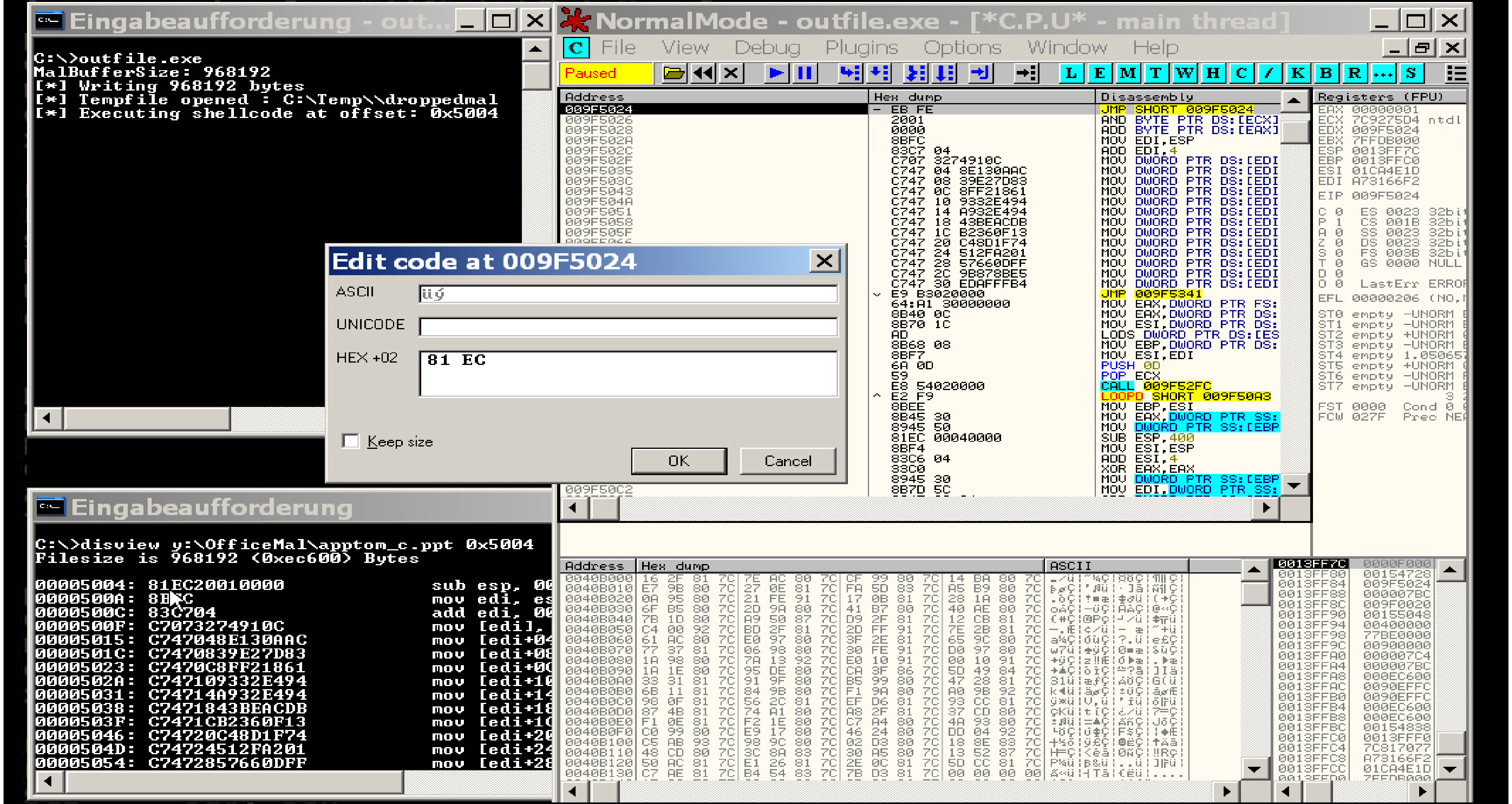
Eingabeaufforderung - out...
OllyDbg - [CPU]
C File View Debug Plugins Options Window Help
Ready
L E M T W H C / K B R ... S
Address
Hex dump
Disassembly
Registers (FPU)
Attach
Cancel

```
C:\>outfile.exe
MalBufferSize: 968192
[*] Writing 968192 bytes
[*] Tempfile opened : C:\Temp\\droppedmal
[*] Executing shellcode at offset: 0x5004
```

Select process to attach

Process	Name	Window	Path
00000C2C	TPOSDSVC	tphkmgr	C:\Programme\Lenovo\
00000C38	EZEJMNAP	EzEjMnApMainWin	C:\PROGRA~1\ThinkPac
00000C40	TpShocks	Default IME	C:\WINNT\system32\Tp
00000C48	rundll32	PwrMgrBkGndWindow	C:\WINNT\system32\ru
00000C50	TPONSCR	tpvolbar	C:\Programme\Lenovo\
00000C58	SynTPLpr	Touchpad driver helper win	C:\Programme\Synapti
00000C64	SynTPEnh	Syn Zoom Window	C:\Programme\Synapti
00000C6C	LPMGR	LPManagerWindow	C:\PROGRA~1\THINKV~1
00000C7C	vmware-tray	VMware Tray Application	C:\Programme\VMware\
00000CA8	hqtray	VMware ACE Host Network Ac	C:\Programme\VMware\
00000CD0	SvcGuiHlpr		C:\Programme\ThinkPa
00000CE4	Acrotray	AcrobatTrayIcon	C:\Programme\Adobe\A
00000D80	scheduler_proxy		C:\Programme\Gemeins
00000D84	MOM	.NET-BroadcastEventWindow.:	C:\Programme\ATI Tec
00000DCC	ctfmon	CiceroUIWndFrame	C:\WINNT\system32\ct
00001158	SAFE8	Steganos Safe 8	C:\Programme\Stegano
0000128C	outfile		C:\outfile.exe
00001464	SAFE8	UIFramework	C:\Programme\Stegano
00001710	cmd	Eingabeaufforderung - outf	C:\WINNT\system32\cm

MalHost-Setup – Debugging



```
C:\>outfile.exe
MalBufferSize: 968192
[*] Writing 968192 bytes
[*] Tempfile opened : C:\Temp\\droppedmal
[*] Executing shellcode at offset: 0x5004
```

```
C:\>disview y:\OfficeMal\apptom_c.ppt 0x5004
Filesize is 968192 (0xec600) Bytes

00005004: 81EC20010000              sub esp, 00
0000500A: 8BEC                      mov edi, es
0000500C: 83C704                    add edi, 00
0000500F: C7073274910C              mov [edi],
00005015: C747048E130AAC            mov [edi+04
0000501C: C7470839E27D83            mov [edi+08
00005023: C7470C8FF21861            mov [edi+0C
0000502A: C747109332E494            mov [edi+10
00005031: C74714A932E494            mov [edi+14
00005038: C7471843BEACDB            mov [edi+18
0000503F: C7471CB2360F13            mov [edi+1C
00005046: C74720C48D1F74            mov [edi+20
0000504D: C74724512FA201            mov [edi+24
00005054: C7472857660DFF            mov [edi+28
```

```
Address    Hex dump              Disassembly
009F5024   EB FE                 JMP SHORT 009F5024
009F5026   2001                  AND BYTE PTR DS:[ECX]
009F5028   0000                  ADD BYTE PTR DS:[EAX]
009F502A   8BFC                  MOV EDI,ESP
009F502C   83C7 04               ADD EDI,4
009F502F   C707 3274910C         MOV DWORD PTR DS:[EDI
009F5035   C747 04 8E130AAC      MOV DWORD PTR DS:[EDI
009F503C   C747 08 39E27D83      MOV DWORD PTR DS:[EDI
009F5043   C747 0C 8FF21861      MOV DWORD PTR DS:[EDI
009F504A   C747 10 9332E494      MOV DWORD PTR DS:[EDI
009F5051   C747 14 A932E494      MOV DWORD PTR DS:[EDI
009F5058   C747 18 43BEACDB      MOV DWORD PTR DS:[EDI
009F505F   C747 1C B2360F13      MOV DWORD PTR DS:[EDI
           C747 20 C48D1F74      MOV DWORD PTR DS:[EDI
           C747 24 512FA201      MOV DWORD PTR DS:[EDI
           C747 28 57660DFF      MOV DWORD PTR DS:[EDI
           C747 2C 9B878BE5      MOV DWORD PTR DS:[EDI
           C747 30 EDAFFFB4      MOV DWORD PTR DS:[EDI
           E9 B3020000           JMP 009F5341
           64:A1 30000000        MOV EAX,DWORD PTR FS:
           8B40 0C               MOV EAX,DWORD PTR DS:
           8B70 1C               MOV ESI,DWORD PTR DS:
           AD                    LODS DWORD PTR DS:[ES
           8B68 08               MOV EBP,DWORD PTR DS:
           8BF7                  MOV ESI,EDI
           6A 0D                 PUSH 0D
           59                    POP ECX
           E8 54020000           CALL 009F52FC
           E2 F9                 LOOPD SHORT 009F50A3
           8BEE                  MOV EBP,ESI
           8B45 30               MOV EAX,DWORD PTR SS:
           8945 50               MOV DWORD PTR SS:[EBP
           81EC 00040000         SUB ESP,400
           8BF4                  MOV ESI,ESP
           83C6 04               ADD ESI,4
           33C0                  XOR EAX,EAX
           8945 30               MOV DWORD PTR SS:[EBP
009F50C2   8B7D 5C               MOV EDI,DWORD PTR SS:
```

```
Registers (FPU)
EAX 00000001
ECX 7C9275D4 ntdl
EDX 009F5024
EBX 7FFDB000
ESP 0013FF7C
EBP 0013FFC0
ESI 01CA4E1D
EDI A73166F2

EIP 009F5024

C 0  ES 0023 32bi
P 1  CS 001B 32bi
A 0  SS 0023 32bi
Z 0  DS 0023 32bi
S 0  FS 003B 32bi
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR

EFL 00000206 (NO,N

ST0 empty -UNORM
ST1 empty -UNORM
ST2 empty +UNORM
ST3 empty -UNORM
ST4 empty 1.05065
ST5 empty +UNORM
ST6 empty -UNORM
ST7 empty -UNORM
               3
FST 0000  Cond 0
FCW 027F  Prec NEA
```

Edit code at 009F5024

ASCII: üý

UNICODE:

HEX +02: 81 EC

□ Keep size

```
Address   Hex dump
0040B000  16 2F 81 7C 7E AC 80 7C CF 99 80 7C 14 BA 80 7C
0040B010  E7 9B 80 7C 27 0E 81 7C FA 5D 83 7C A5 B9 80 7C
0040B020  0A 95 80 7C 21 FE 91 7C 17 0B 81 7C 28 1A 80 7C
0040B030  6F B5 80 7C 2D 9A 80 7C 41 B7 80 7C 40 AE 80 7C
0040B040  7B 1D 80 7C A9 50 87 7C D9 2F 81 7C 12 CB 81 7C
0040B050  C4 00 92 7C BD 2F 81 7C 2D FF 91 7C 7E 2B 81 7C
0040B060  61 AC 80 7C E0 97 80 7C 3F 2E 81 7C 65 9C 80 7C
0040B070  77 37 81 7C 06 98 80 7C 30 FE 91 7C D0 97 80 7C
0040B080  1A 98 80 7C 7A 13 92 7C E0 10 91 7C 00 10 91 7C
0040B090  1A 1E 80 7C 95 DE 80 7C CA 3F 86 7C 5D 49 84 7C
0040B0A0  33 31 81 7C 91 9F 80 7C B5 99 80 7C 47 28 81 7C
0040B0B0  6B 11 81 7C 84 9B 80 7C F1 9A 80 7C A0 9B 92 7C
0040B0C0  98 0F 81 7C 56 2C 81 7C EF D6 81 7C 93 CC 81 7C
0040B0D0  87 4B 81 7C 74 A1 80 7C A8 2F 81 7C 37 CD 80 7C
0040B0E0  F1 0E 81 7C F2 1E 80 7C C7 A4 80 7C 4A 93 80 7C
0040B0F0  C0 99 80 7C E9 17 80 7C 46 24 80 7C DD 04 92 7C
0040B100  C5 AB 93 7C 98 9C 80 7C 02 D3 80 7C 18 8E 83 7C
0040B110  48 CD 80 7C 3C 8A 83 7C 30 A5 80 7C 13 52 87 7C
0040B120  50 AC 81 7C E1 26 81 7C 2E 0C 81 7C 5D CC 81 7C
0040B130  C7 AE 81 7C B4 54 83 7C 7B D3 81 7C 00 00 00 00
```

```
0013FF7C   0000F000
0013FF80   00154728
0013FF84   009F5024
0013FF88   000007BC
0013FF8C   009F0020
0013FF90   00155048
0013FF94   00400000
0013FF98   77BE0000
0013FF9C   00900000
0013FFA0   000007C4
0013FFA4   000007BC
0013FFA8   000EC600
0013FFAC   0090EFFC
0013FFB0   0090EFFC
0013FFB4   000EC600
0013FFB8   000EC600
0013FFBC   00154838
0013FFC0   0013FFF0
0013FFC4   7C817077
0013FFC8   A73166F2
0013FFCC   01CA4E1D
```

OfficeMalScanner Suite Download

http://www.reconstructer.org/code/OfficeMalScanner.zip

Questions?

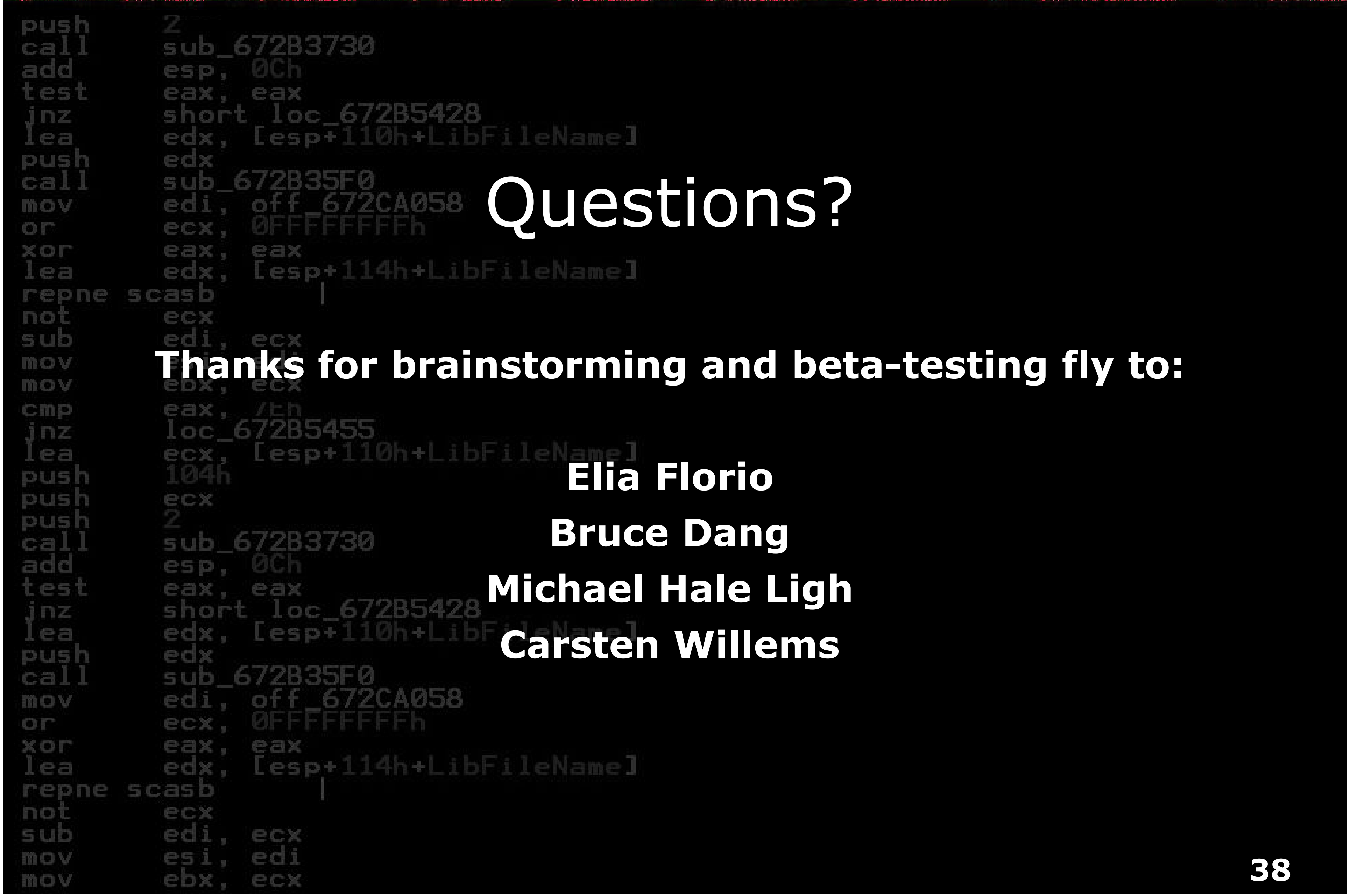
Thanks for brainstorming and beta-testing fly to:

Elia Florio

Bruce Dang

Michael Hale Ligh

Carsten Willems



```
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb      |
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
cmp     eax, 7Eh
jnz     loc_672B5455
lea     ecx, [esp+110h+LibFileName]
push    104h
push    ecx
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb      |
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
```