ASPack 2.29

Manual Unpacking Notes

ASPack

- Portable Executable File Compression
  - **NOT meant for File Protection!**
  - In-place de-compression
  - OEP remains unchanged
  - Small and easy to reverse decompression routine

Packed vs. Unpacked

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00009000	00001000	00004C00	00000400	00000000	00000000	0000	0000	E0000060
.rdata	00002000	0000A000	00000C00	00005000	00000000	00000000	0000	0000	C0000040
.data	00003000	0000C000	00000400	00005C00	00000000	00000000	0000	0000	C0000040
.aspack	00002000	0000F000	00001200	00006000	00000000	00000000	0000	0000	E0000060
.adata	00001000	00011000	00000000	00007200	00000000	00000000	0000	0000	E0000040

Packed Executable

	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00008894	00001000	00008A00	00000400	00000000	00000000	0000	0000	60000020
.rdata	00001DC4	0000A000	00001E00	00008E00	00000000	00000000	0000	0000	40000040
.data	00002AC8	0000C000	00001000	0000AC00	00000000	00000000	0000	0000	C0000040

Original Executable

Packed vs. Unpacked

Executable Characteristic	Original	Compressed
Image Base	00400000	00400000
Entry Point	0x1262	0xF001
Number of Sections	3	5
Entry Point Section	.text	.aspack
Size of Executable	47 KB	28.5 KB

Analyzing Decoder Routine

```
.aspack:0040F001
.aspack:0040F001                pusha
.aspack:0040F002                call    loc_40F00A
.aspack:0040F002 ; ---------------------------------------------------------------------------
.aspack:0040F007                db 0E9h
.aspack:0040F008 ; ---------------------------------------------------------------------------
.aspack:0040F008                jmp     short loc_40F00E
.aspack:0040F00A ; ---------------------------------------------------------------------------
.aspack:0040F00A
.aspack:0040F00A loc_40F00A:                            ; CODE XREF: start+1↑j
.aspack:0040F00A                pop     ebp
.aspack:0040F00B                inc     ebp
.aspack:0040F00C                push    ebp
.aspack:0040F00D                retn
.aspack:0040F00D start          endp ; sp-analysis failed
.aspack:0040F00D
```

Analyzing Decoder Routine

```
.aspack:0040F00E                call    loc_40F014
.aspack:0040F00E ; END OF FUNCTION CHUNK FOR start
.aspack:0040F00E ; ---------------------------------------------------------------------------
.aspack:0040F013                db 0EBh
.aspack:0040F014 ; ---------------------------------------------------------------------------
.aspack:0040F014 ; START OF FUNCTION CHUNK FOR start
.aspack:0040F014
.aspack:0040F014 loc_40F014:                            ; CODE XREF: start:loc_40F00E↑j
.aspack:0040F014                pop     ebp
.aspack:0040F015                mov     ebx, 0FFFFFFEDh
.aspack:0040F01A                add     ebx, ebp
.aspack:0040F01C                sub     ebx, 0F000h
.aspack:0040F022                cmp     dword ptr [ebp+488h], 0
.aspack:0040F029                mov     [ebp+488h], ebx
.aspack:0040F02F                jnz     loc_40F400
.aspack:0040F035                lea     eax, [ebp+494h]
.aspack:0040F03B                push    eax
.aspack:0040F03C                call    dword ptr [ebp+0FA9h]
.aspack:0040F042                mov     [ebp+48Ch], eax
.aspack:0040F048                mov     esi, eax
.aspack:0040F04A                lea     edi, [ebp+51h]
.aspack:0040F04D
.aspack:0040F04D loc_40F04D:                            ; CODE XREF: start+5C↓j
.aspack:0040F04D                push    edi
```

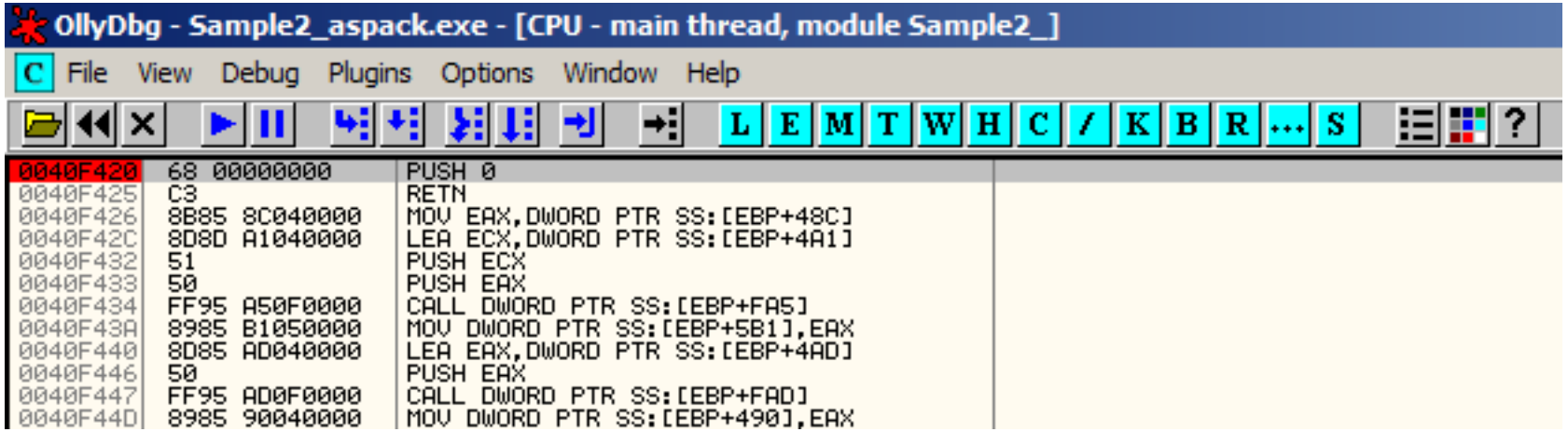
Analyzing Decoder Routine

```
.aspack:0040F062 ; END OF FUNCTION CHUNK FOR start
.aspack:0040F062 ; ---------------------------------------------------------------------------
.aspack:0040F064 aVirtualalloc   db 'VirtualAlloc',0
.aspack:0040F071 aVirtualfree    db 'VirtualFree',0
.aspack:0040F07D aVirtualprotect db 'VirtualProtect',0
.aspack:0040F08C                 dd 959D8B00h, 0B000005h, 8B0A74DBh, 99858703h, 89000005h
.aspack:0040F08C                 dd 0C5B58D03h, 83000005h, 840F003Eh, 10Ah, 68046Ah, 68000010h
.aspack:0040F08C                 dd 1800h, 55FF006Ah, 48858951h, 8B000001h, 0E050446h, 0F000001h
.aspack:0040F08C                 dd 0B784h, 68046A00h, 1000h, 0FF006A50h, 85895155h, 144h
.aspack:0040F08C                 dd 31E8B56h, 4889Dh, 48B5FF00h, 0FF000001h, 53500476h
.aspack:0040F08C                 dd 5C7E8h, 8000B300h, 4D7500FBh, 0EF85FEh, 51500000h, 0C88B5356h
.aspack:0040F08C                 dd 8B05E983h, 144B5h, 0BDB3300h, 782E74C9h, 0E83CAC2Ch
.aspack:0040F08C                 dd 0EB0A74h, 474E93Ch, 0EBEB4943h, 0EB068Bh, 75053E80h
.aspack:0040F08C                 dd 0C10024F3h, 0C32B18C0h, 0C3830689h, 4C68305h, 0EB05E983h
.aspack:0040F08C                 dd 595E5BCEh, 8EB58h, 0
.aspack:0040F15C                 dd 8B000000h, 33E8BC8h, 488BDh, 44B58B00h, 0C1000001h
.aspack:0040F15C                 dd 0A5F302F9h, 0E183C88Bh, 5EA4F303h, 800068h, 0FF006A00h
.aspack:0040F15C                 dd 144B5h, 5E55FF00h, 830CC683h, 850F003Eh, 0FFFFFF2Fh
.aspack:0040F15C                 dd 800068h, 0FF006A00h, 148B5h, 5E55FF00h, 5959D8Bh, 0DB0B0000h
.aspack:0040F15C                 dd 38B0874h, 5998587h, 958B0000h, 488h, 591858Bh, 0D02B0000h
.aspack:0040F15C                 dd 0C28B7974h, 3310E8C1h, 9DB58BDBh, 3000005h, 488B5h
.aspack:0040F15C                 dd 3E8300h, 4E8B6174h, 8E98304h, 3E8BE9D1h, 488BD03h, 0C6830000h
.aspack:0040F15C                 dd 1E8B6608h, 830CEBC1h, 0C7401FBh, 7402FB83h, 3FB8316h
.aspack:0040F15C                 dd 2CEB2074h, 811E8B66h, 0FFFE3h, 4016600h, 661DEB1Fh
```

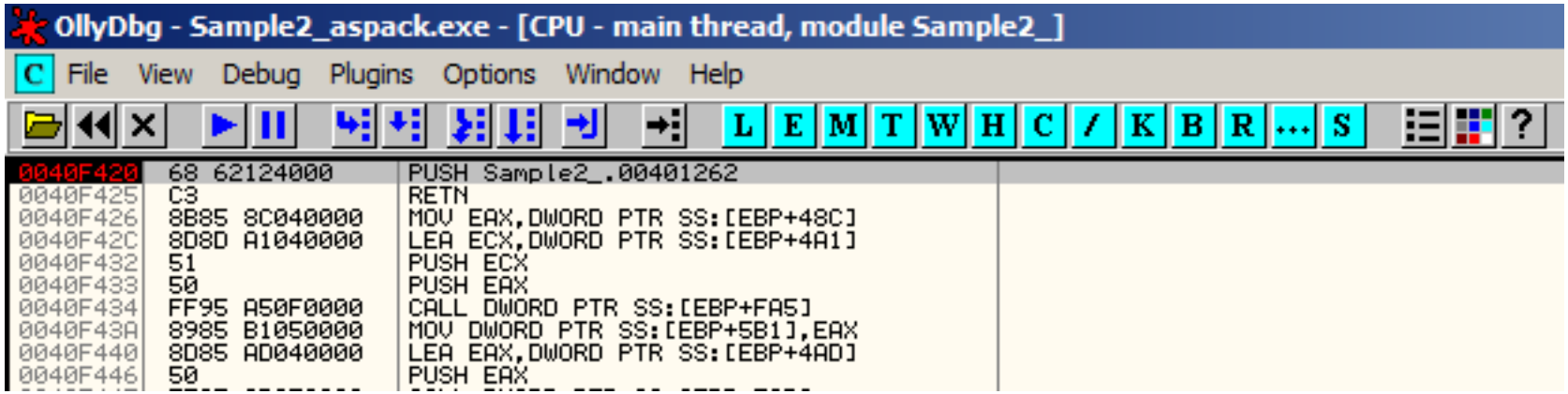
Analyzing Decoder Routine

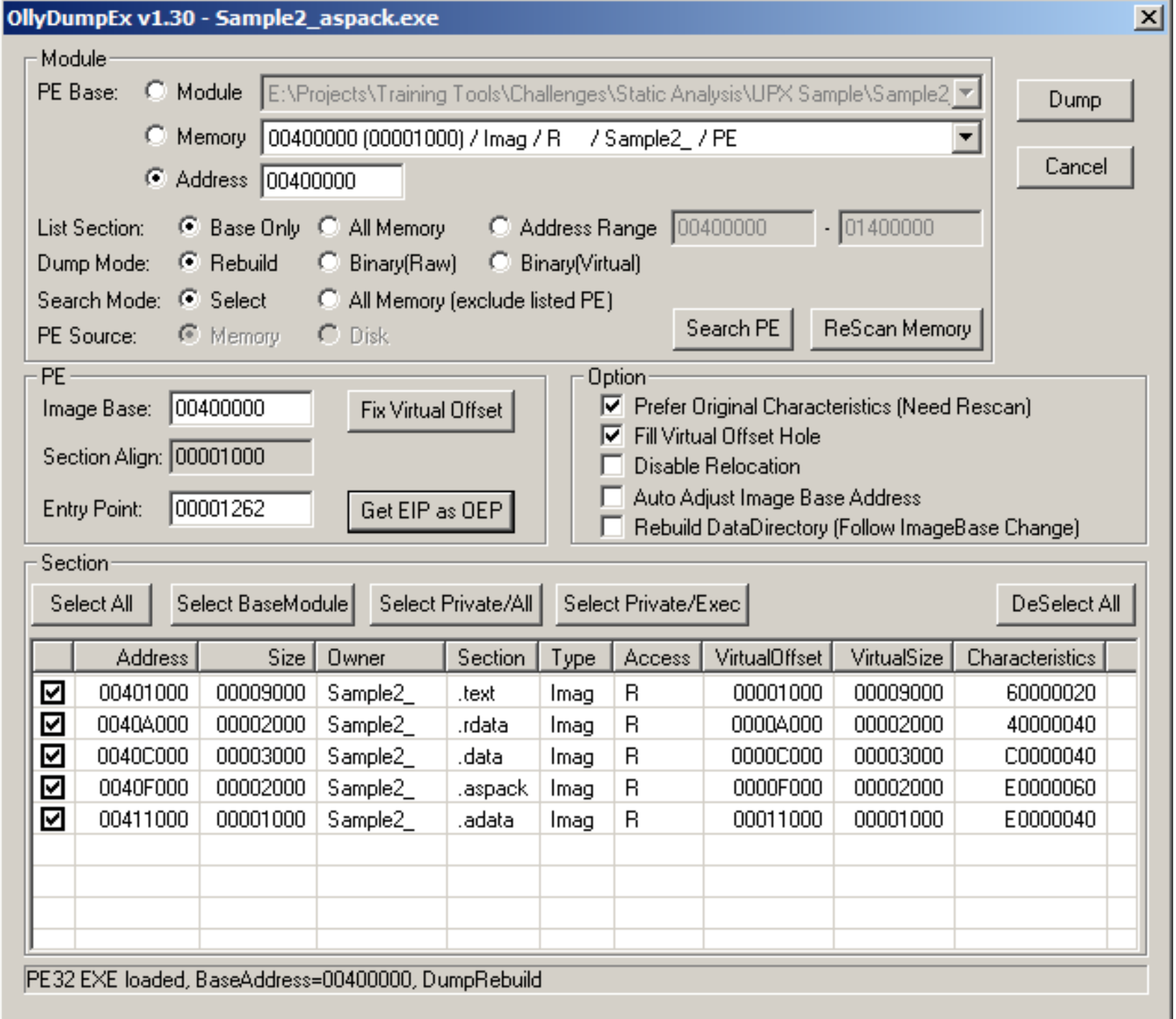
```
.aspack:0040F400 loc_40F400:                             ; CODE XREF: start+2E↑j
.aspack:0040F400                 mov     eax, 1262h
.aspack:0040F405                 push    eax
.aspack:0040F406                 add     eax, [ebp+488h]
.aspack:0040F40C                 pop     ecx
.aspack:0040F40D                 or      ecx, ecx
.aspack:0040F40F                 mov     [ebp+40Eh], eax
.aspack:0040F415                 popa
.aspack:0040F416                 jnz     short loc_40F420 ; Interesting.. results in EIP=0
.aspack:0040F418                 mov     eax, 1
.aspack:0040F41D                 retn    0Ch
.aspack:0040F420 ; ---------------------------------------------------------------------------
.aspack:0040F420
.aspack:0040F420 loc_40F420:                             ; CODE XREF: start+415↑j
.aspack:0040F420                 push    0                ; Interesting.. results in EIP=0
.aspack:0040F425                 retn
.aspack:0040F425 ; END OF FUNCTION CHUNK FOR start
.aspack:0040F425 ; ---------------------------------------------------------------------------
```

Dynamic Analysis



Hardware Breakpoint on Execution




OllyDumpEx v1.30 - Sample2_aspack.exe
Module
PE Base:
Module
E:\Projects\Training Tools\Challenges\Static Analysis\UPX Sample\Sample2
Memory
00400000 (00001000) / Imag / R / Sample2_ / PE
Address
00400000
Dump
Cancel
List Section:
Base Only
All Memory
Address Range
00400000
-
01400000
Dump Mode:
Rebuild
Binary(Raw)
Binary(Virtual)
Search Mode:
Select
All Memory (exclude listed PE)
PE Source:
Memory
Disk
Search PE
ReScan Memory
PE
Image Base:
00400000
Fix Virtual Offset
Section Align:
00001000
Entry Point:
00001262
Get EIP as OEP
Option
Prefer Original Characteristics (Need Rescan)
Fill Virtual Offset Hole
Disable Relocation
Auto Adjust Image Base Address
Rebuild DataDirectory (Follow ImageBase Change)
Section
Select All
Select BaseModule
Select Private/All
Select Private/Exec
DeSelect All
Address Size Owner Section Type Access VirtualOffset VirtualSize Characteristics
00401000 00009000 Sample2_ .text Imag R 00001000 00009000 60000020
0040A000 00002000 Sample2_ .rdata Imag R 0000A000 00002000 40000040
0040C000 00003000 Sample2_ .data Imag R 0000C000 00003000 C0000040
0040F000 00002000 Sample2_ .aspack Imag R 0000F000 00002000 E0000060
00411000 00001000 Sample2_ .adata Imag R 00011000 00001000 E0000040
PE32 EXE loaded, BaseAddress=00400000, DumpRebuild

Decompressed PE

```
; int __cdecl main(int argc, const char **argv, const char **envp)
_main proc near

var_4= dword ptr -4
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h

push    ebp
mov     ebp, esp
push    ecx
mov     [ebp+var_4], offset aMul
mov     eax, [ebp+var_4]
push    eax
push    offset aKeyIsAt0x08x ; "Key is at: 0x%08x\n"
call    sub_401022
add     esp, 8
xor     eax, eax
mov     esp, ebp
pop     ebp
retn
_main endp
```