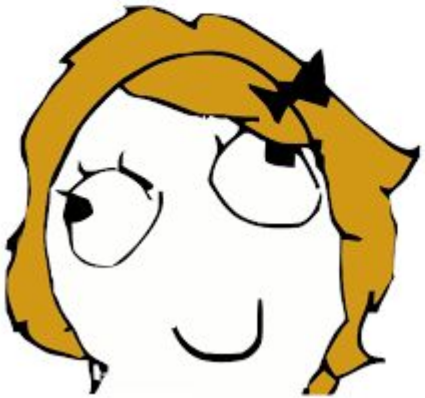
Automated Unpacking of Malware with Memory Forensics

Raphaela Mettig

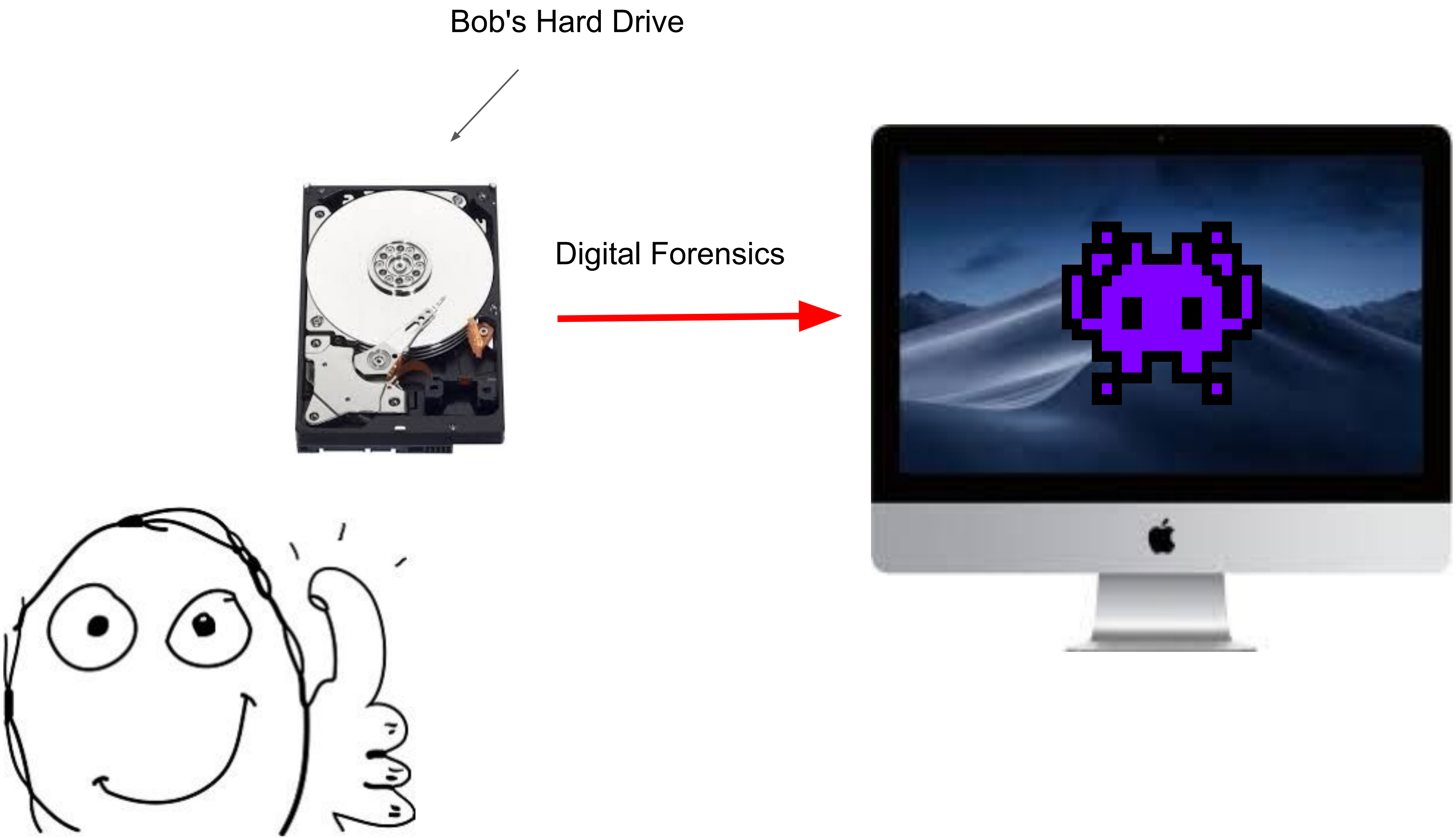
Story Time

Alice



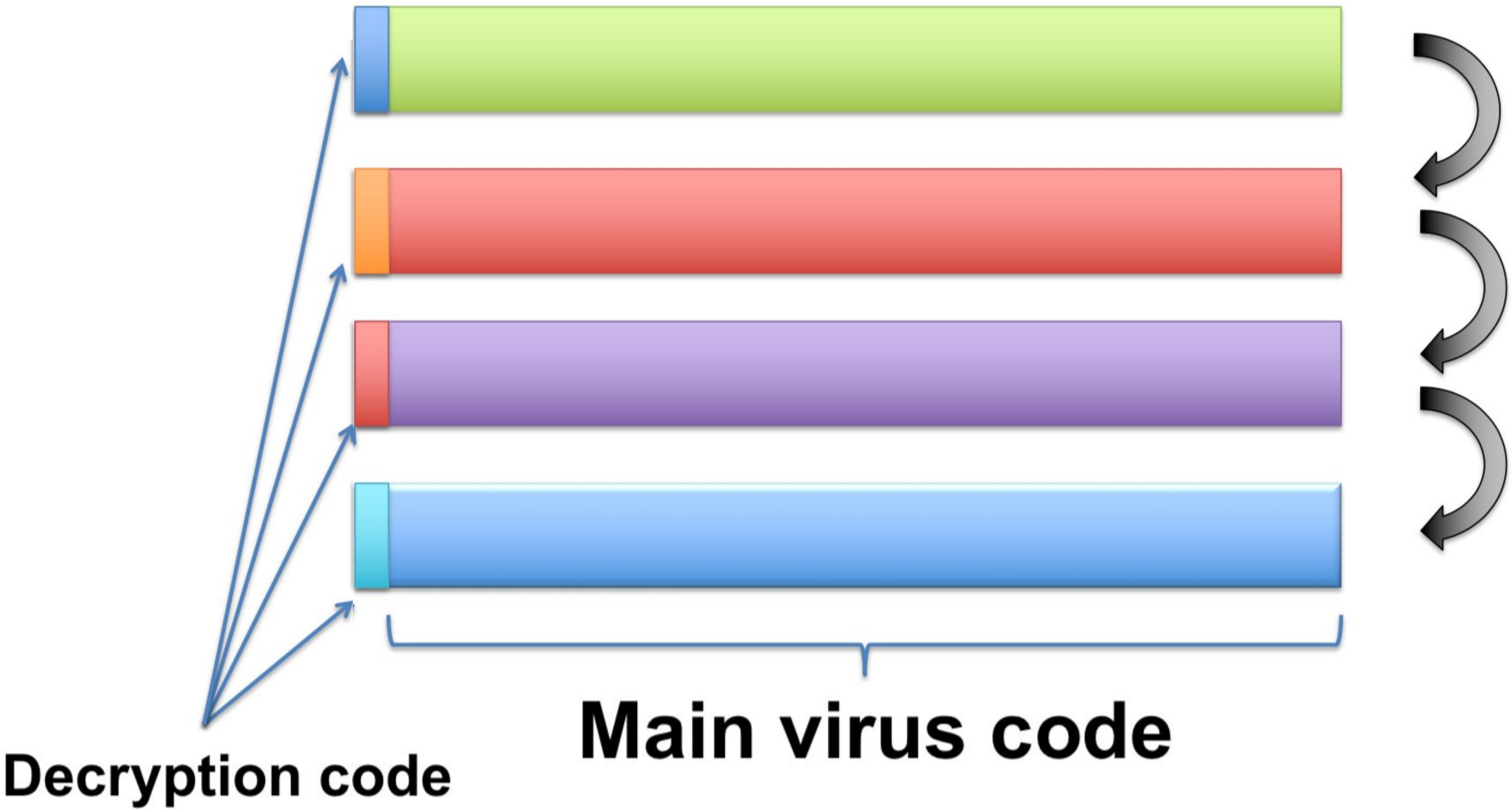
Bob's Computer

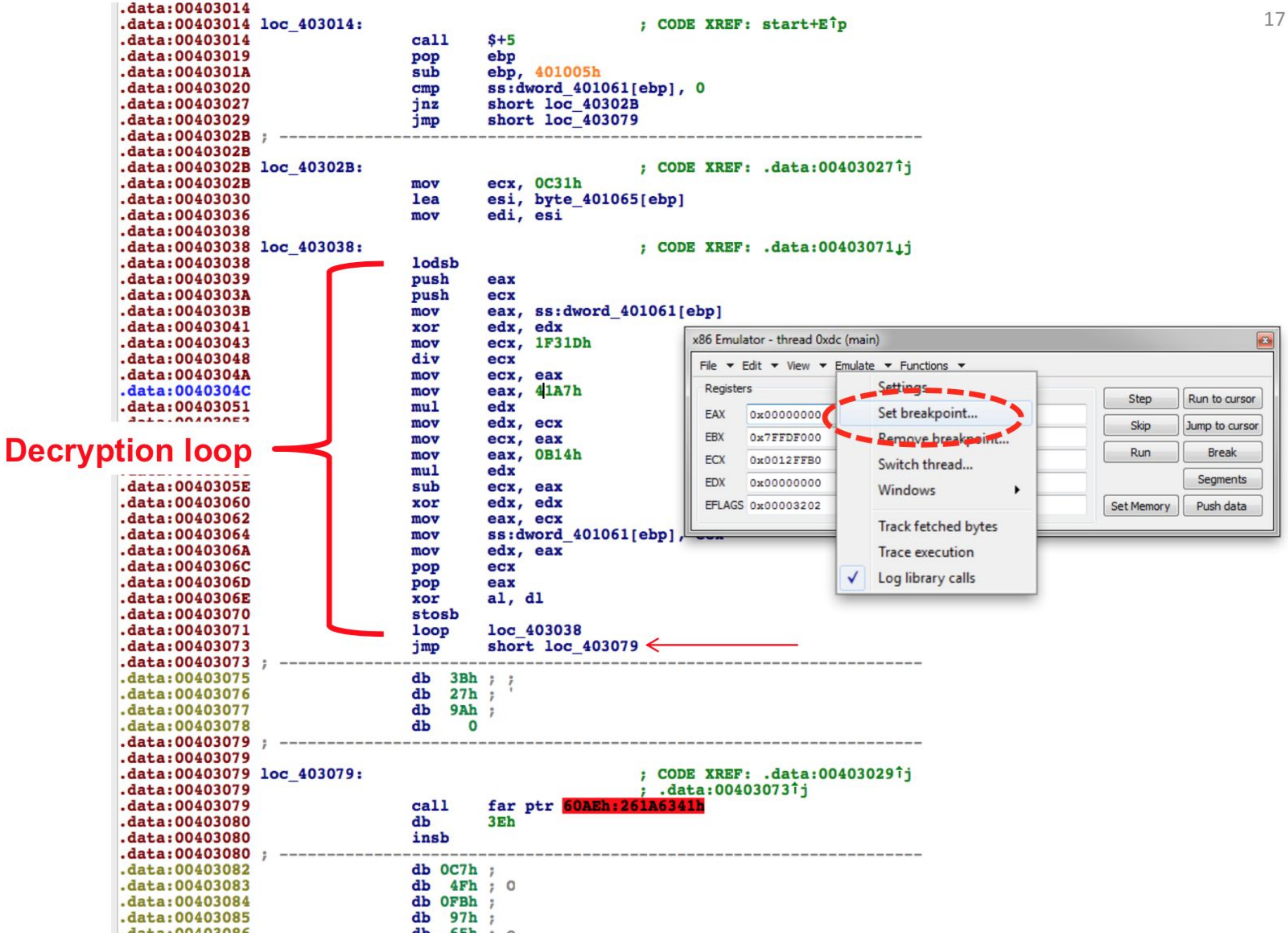
RAM
1 011 01 1 0 10
110 1 0
0
1



Bob's Hard Drive

Digital Forensics


Main virus code
Decryption code



```
.data:00403014
.data:00403014 loc_403014:                         ; CODE XREF: start+E↑p
.data:00403014                 call    $+5
.data:00403019                 pop     ebp
.data:0040301A                 sub     ebp, 401005h
.data:00403020                 cmp     ss:dword_401061[ebp], 0
.data:00403027                 jnz     short loc_40302B
.data:00403029                 jmp     short loc_403079
.data:0040302B ; ---------------------------------------------------------------
.data:0040302B
.data:0040302B loc_40302B:                         ; CODE XREF: .data:00403027↑j
.data:0040302B                 mov     ecx, 0C31h
.data:00403030                 lea     esi, byte_401065[ebp]
.data:00403036                 mov     edi, esi
.data:00403038
.data:00403038 loc_403038:                         ; CODE XREF: .data:00403071↓j
.data:00403038                 lodsb
.data:00403039                 push    eax
.data:0040303A                 push    ecx
.data:0040303B                 mov     eax, ss:dword_401061[ebp]
.data:00403041                 xor     edx, edx
.data:00403043                 mov     ecx, 1F31Dh
.data:00403048                 div     ecx
.data:0040304A                 mov     ecx, eax
.data:0040304C                 mov     eax, 41A7h
.data:00403051                 mul     edx
.data:00403053                 mov     edx, ecx
                               mov     ecx, eax
                               mov     eax, 0B14h
                               mul     edx
.data:0040305E                 sub     ecx, eax
.data:00403060                 xor     edx, edx
.data:00403062                 mov     eax, ecx
.data:00403064                 mov     ss:dword_401061[ebp], ecx
.data:0040306A                 mov     edx, eax
.data:0040306C                 pop     ecx
.data:0040306D                 pop     eax
.data:0040306E                 xor     al, dl
.data:00403070                 stosb
.data:00403071                 loop    loc_403038
.data:00403073                 jmp     short loc_403079
.data:00403073 ; ---------------------------------------------------------------
.data:00403075                 db  3Bh ; ;
.data:00403076                 db  27h ; '
.data:00403077                 db  9Ah ;
.data:00403078                 db    0
.data:00403079 ; ---------------------------------------------------------------
.data:00403079
.data:00403079 loc_403079:                         ; CODE XREF: .data:00403029↑j
.data:00403079                                     ; .data:00403073↑j
.data:00403079                 call    far ptr 60AEh:261A6341h
.data:00403080                 db      3Eh
.data:00403080                 insb
.data:00403080 ; ---------------------------------------------------------------
.data:00403082                 db 0C7h ;
.data:00403083                 db  4Fh ; O
.data:00403084                 db 0FBh ;
.data:00403085                 db  97h ;
.data:00403086                 db  65h ; e
```

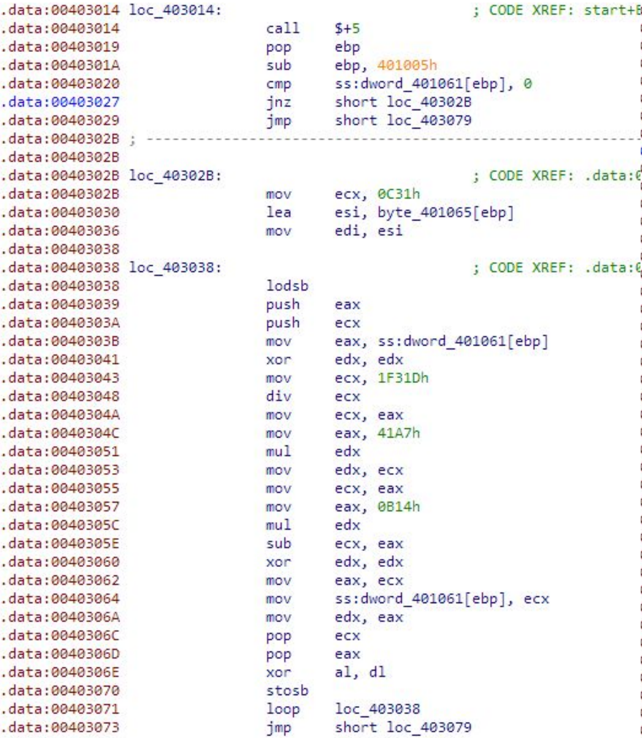
Decryption loop

x86 Emulator - thread 0xdc (main)

Registers

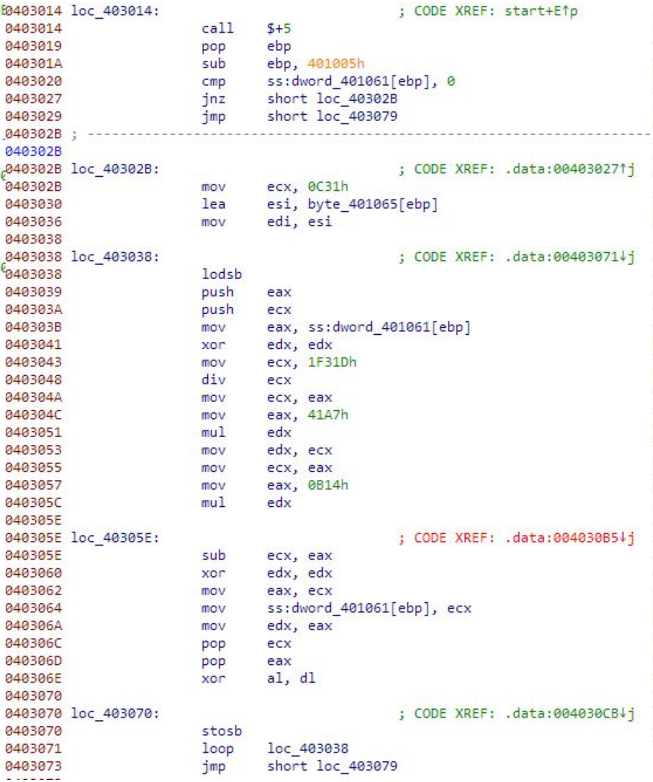
EAX 0x00000000
EBX 0x7FFDF000
ECX 0x0012FFB0
EDX 0x00000000
EFLAGS 0x00003202

1



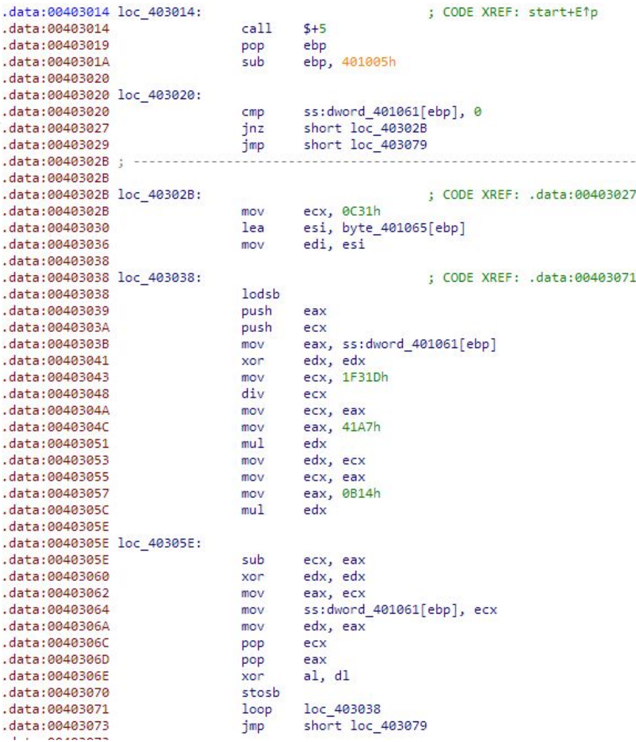
```
.data:00403014 loc_403014:                             ; CODE XREF: start+E
.data:00403014                 call    $+5
.data:00403019                 pop     ebp
.data:0040301A                 sub     ebp, 401005h
.data:00403020                 cmp     ss:dword_401061[ebp], 0
.data:00403027                 jnz     short loc_40302B
.data:00403029                 jmp     short loc_403079
.data:0040302B ; ---------------------------------------------------------------------------
.data:0040302B
.data:0040302B loc_40302B:                             ; CODE XREF: .data:0
.data:0040302B                 mov     ecx, 0C31h
.data:00403030                 lea     esi, byte_401065[ebp]
.data:00403036                 mov     edi, esi
.data:00403038
.data:00403038 loc_403038:                             ; CODE XREF: .data:0
.data:00403038                 lodsb
.data:00403039                 push    eax
.data:0040303A                 push    ecx
.data:0040303B                 mov     eax, ss:dword_401061[ebp]
.data:00403041                 xor     edx, edx
.data:00403043                 mov     ecx, 1F31Dh
.data:00403048                 div     ecx
.data:0040304A                 mov     ecx, eax
.data:0040304C                 mov     eax, 41A7h
.data:00403051                 mul     edx
.data:00403053                 mov     edx, ecx
.data:00403055                 mov     ecx, eax
.data:00403057                 mov     eax, 0B14h
.data:0040305C                 mul     edx
.data:0040305E                 sub     ecx, eax
.data:00403060                 xor     edx, edx
.data:00403062                 mov     eax, ecx
.data:00403064                 mov     ss:dword_401061[ebp], ecx
.data:0040306A                 mov     edx, eax
.data:0040306C                 pop     ecx
.data:0040306D                 pop     eax
.data:0040306E                 xor     al, dl
.data:00403070                 stosb
.data:00403071                 loop    loc_403038
.data:00403073                 jmp     short loc_403079
```

2

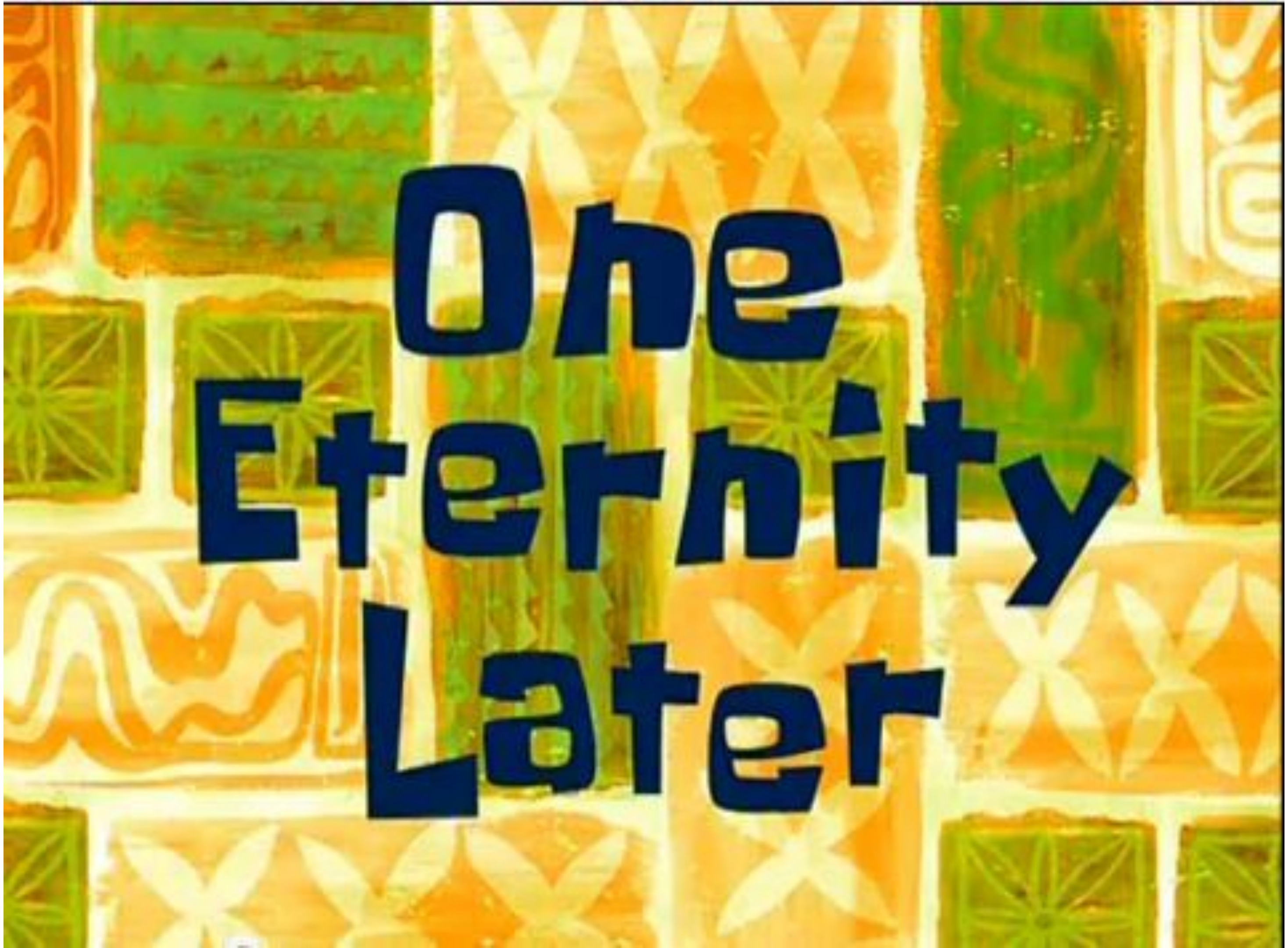


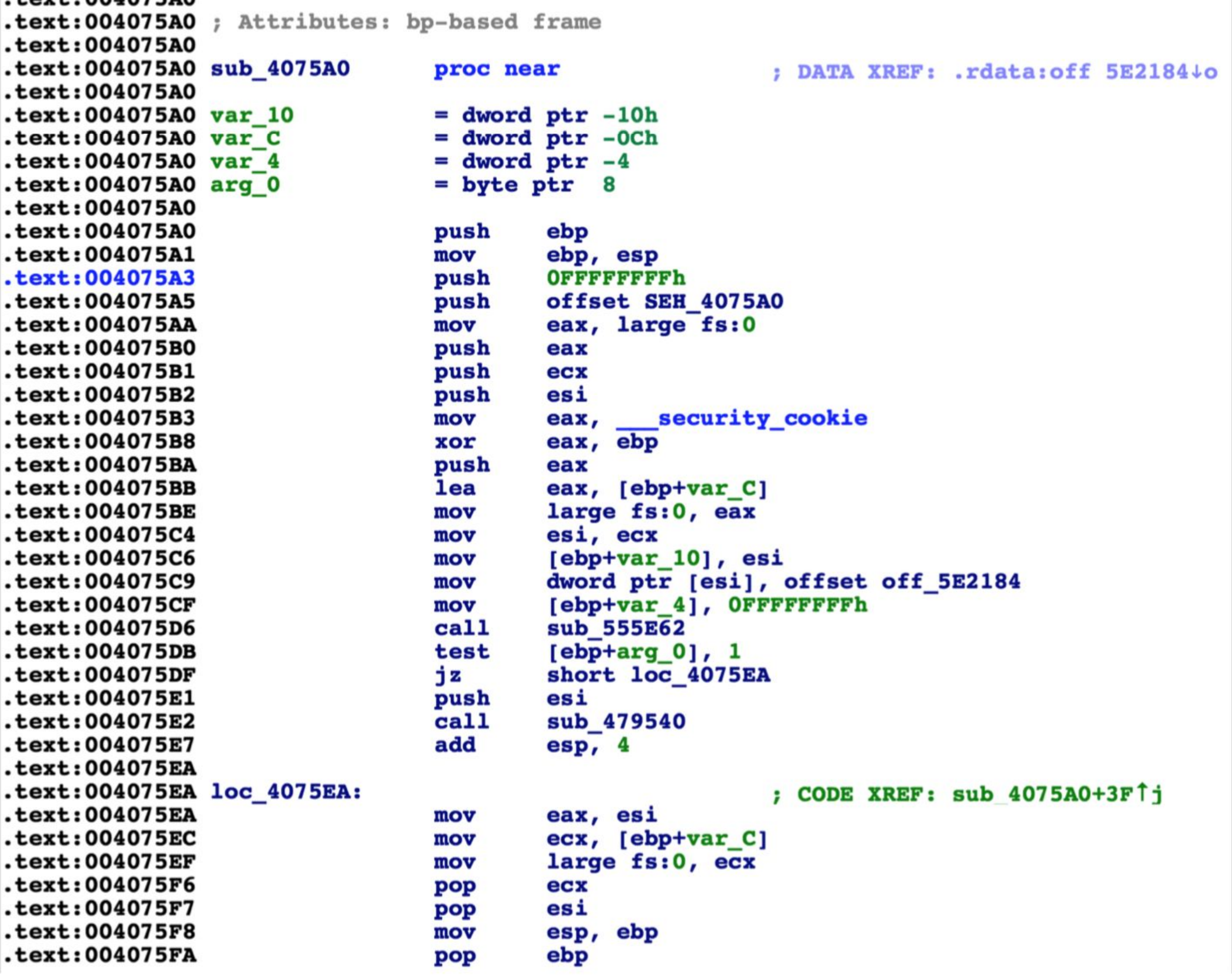
```
0403014 loc_403014:                             ; CODE XREF: start+E↑p
0403014                 call    $+5
0403019                 pop     ebp
040301A                 sub     ebp, 401005h
0403020                 cmp     ss:dword_401061[ebp], 0
0403027                 jnz     short loc_40302B
0403029                 jmp     short loc_403079
040302B ; ---------------------------------------------------------------------------
040302B
040302B loc_40302B:                             ; CODE XREF: .data:00403027↑j
040302B                 mov     ecx, 0C31h
0403030                 lea     esi, byte_401065[ebp]
0403036                 mov     edi, esi
0403038
0403038 loc_403038:                             ; CODE XREF: .data:00403071↓j
0403038                 lodsb
0403039                 push    eax
040303A                 push    ecx
040303B                 mov     eax, ss:dword_401061[ebp]
0403041                 xor     edx, edx
0403043                 mov     ecx, 1F31Dh
0403048                 div     ecx
040304A                 mov     ecx, eax
040304C                 mov     eax, 41A7h
0403051                 mul     edx
0403053                 mov     edx, ecx
0403055                 mov     ecx, eax
0403057                 mov     eax, 0B14h
040305C                 mul     edx
040305E
040305E loc_40305E:                             ; CODE XREF: .data:004030B5↓j
040305E                 sub     ecx, eax
0403060                 xor     edx, edx
0403062                 mov     eax, ecx
0403064                 mov     ss:dword_401061[ebp], ecx
040306A                 mov     edx, eax
040306C                 pop     ecx
040306D                 pop     eax
040306E                 xor     al, dl
0403070
0403070 loc_403070:                             ; CODE XREF: .data:004030CB↓j
0403070                 stosb
0403071                 loop    loc_403038
0403073                 jmp     short loc_403079
```

3



```
.data:00403014 loc_403014:                             ; CODE XREF: start+E↑p
.data:00403014                 call    $+5
.data:00403019                 pop     ebp
.data:0040301A                 sub     ebp, 401005h
.data:00403020
.data:00403020 loc_403020:
.data:00403020                 cmp     ss:dword_401061[ebp], 0
.data:00403027                 jnz     short loc_40302B
.data:00403029                 jmp     short loc_403079
.data:0040302B ; ---------------------------------------------------------------------------
.data:0040302B
.data:0040302B loc_40302B:                             ; CODE XREF: .data:00403027
.data:0040302B                 mov     ecx, 0C31h
.data:00403030                 lea     esi, byte_401065[ebp]
.data:00403036                 mov     edi, esi
.data:00403038
.data:00403038 loc_403038:                             ; CODE XREF: .data:00403071
.data:00403038                 lodsb
.data:00403039                 push    eax
.data:0040303A                 push    ecx
.data:0040303B                 mov     eax, ss:dword_401061[ebp]
.data:00403041                 xor     edx, edx
.data:00403043                 mov     ecx, 1F31Dh
.data:00403048                 div     ecx
.data:0040304A                 mov     ecx, eax
.data:0040304C                 mov     eax, 41A7h
.data:00403051                 mul     edx
.data:00403053                 mov     edx, ecx
.data:00403055                 mov     ecx, eax
.data:00403057                 mov     eax, 0B14h
.data:0040305C                 mul     edx
.data:0040305E
.data:0040305E loc_40305E:
.data:0040305E                 sub     ecx, eax
.data:00403060                 xor     edx, edx
.data:00403062                 mov     eax, ecx
.data:00403064                 mov     ss:dword_401061[ebp], ecx
.data:0040306A                 mov     edx, eax
.data:0040306C                 pop     ecx
.data:0040306D                 pop     eax
.data:0040306E                 xor     al, dl
.data:00403070                 stosb
.data:00403071                 loop    loc_403038
.data:00403073                 jmp     short loc_403079
```


One
Eternity
Later



```
.text:004075A0 ; Attributes: bp-based frame
.text:004075A0
.text:004075A0 sub_4075A0      proc near                ; DATA XREF: .rdata:off 5E2184↓o
.text:004075A0
.text:004075A0 var_10          = dword ptr -10h
.text:004075A0 var_C           = dword ptr -0Ch
.text:004075A0 var_4           = dword ptr -4
.text:004075A0 arg_0           = byte ptr  8
.text:004075A0
.text:004075A0                 push    ebp
.text:004075A1                 mov     ebp, esp
.text:004075A3                 push    0FFFFFFFFh
.text:004075A5                 push    offset SEH_4075A0
.text:004075AA                 mov     eax, large fs:0
.text:004075B0                 push    eax
.text:004075B1                 push    ecx
.text:004075B2                 push    esi
.text:004075B3                 mov     eax, ___security_cookie
.text:004075B8                 xor     eax, ebp
.text:004075BA                 push    eax
.text:004075BB                 lea     eax, [ebp+var_C]
.text:004075BE                 mov     large fs:0, eax
.text:004075C4                 mov     esi, ecx
.text:004075C6                 mov     [ebp+var_10], esi
.text:004075C9                 mov     dword ptr [esi], offset off_5E2184
.text:004075CF                 mov     [ebp+var_4], 0FFFFFFFFh
.text:004075D6                 call    sub_555E62
.text:004075DB                 test    [ebp+arg_0], 1
.text:004075DF                 jz      short loc_4075EA
.text:004075E1                 push    esi
.text:004075E2                 call    sub_479540
.text:004075E7                 add     esp, 4
.text:004075EA
.text:004075EA loc_4075EA:                              ; CODE XREF: sub_4075A0+3F↑j
.text:004075EA                 mov     eax, esi
.text:004075EC                 mov     ecx, [ebp+var_C]
.text:004075EF                 mov     large fs:0, ecx
.text:004075F6                 pop     ecx
.text:004075F7                 pop     esi
.text:004075F8                 mov     esp, ebp
.text:004075FA                 pop     ebp
```

Memory Analysis

Use plugins to analyze:

Running processes

Hidden processes

Hooks that hide malware

Network connections

Encryption keys

Private browsing data

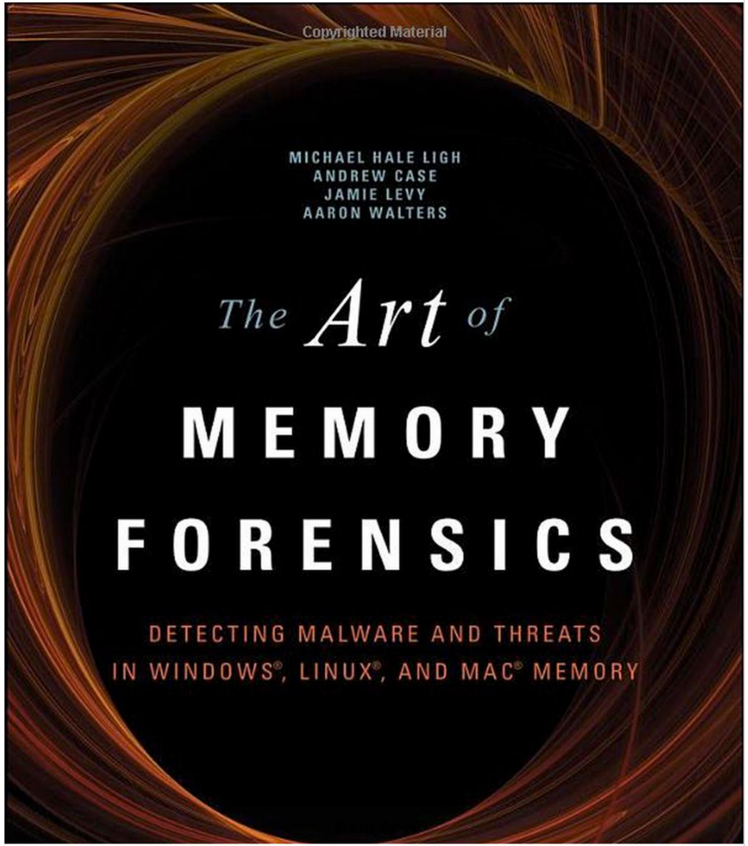
Clipboard data

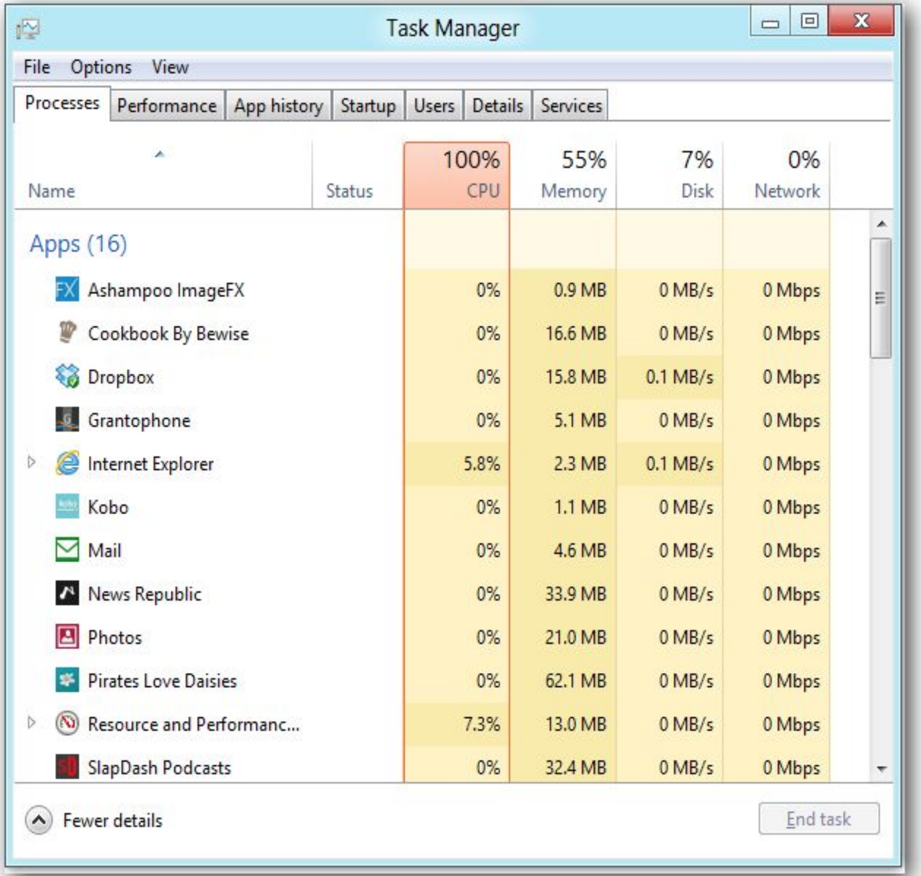
Volatile registry branches

Command history

Window hierarchy

+ "easily" develop plugins


Copyrighted Material
MICHAEL HALE LIGH
ANDREW CASE
JAMIE LEVY
AARON WALTERS
The Art of MEMORY FORENSICS
DETECTING MALWARE AND THREATS IN WINDOWS®, LINUX®, AND MAC® MEMORY



Name	Status	100% CPU	55% Memory	7% Disk	0% Network
Apps (16)					
Ashampoo ImageFX		0%	0.9 MB	0 MB/s	0 Mbps
Cookbook By Bewise		0%	16.6 MB	0 MB/s	0 Mbps
Dropbox		0%	15.8 MB	0.1 MB/s	0 Mbps
Grantophone		0%	5.1 MB	0 MB/s	0 Mbps
Internet Explorer		5.8%	2.3 MB	0.1 MB/s	0 Mbps
Kobo		0%	1.1 MB	0 MB/s	0 Mbps
Mail		0%	4.6 MB	0 MB/s	0 Mbps
News Republic		0%	33.9 MB	0 MB/s	0 Mbps
Photos		0%	21.0 MB	0 MB/s	0 Mbps
Pirates Love Daisies		0%	62.1 MB	0 MB/s	0 Mbps
Resource and Performanc...		7.3%	13.0 MB	0 MB/s	0 Mbps
SlapDash Podcasts		0%	32.4 MB	0 MB/s	0 Mbps

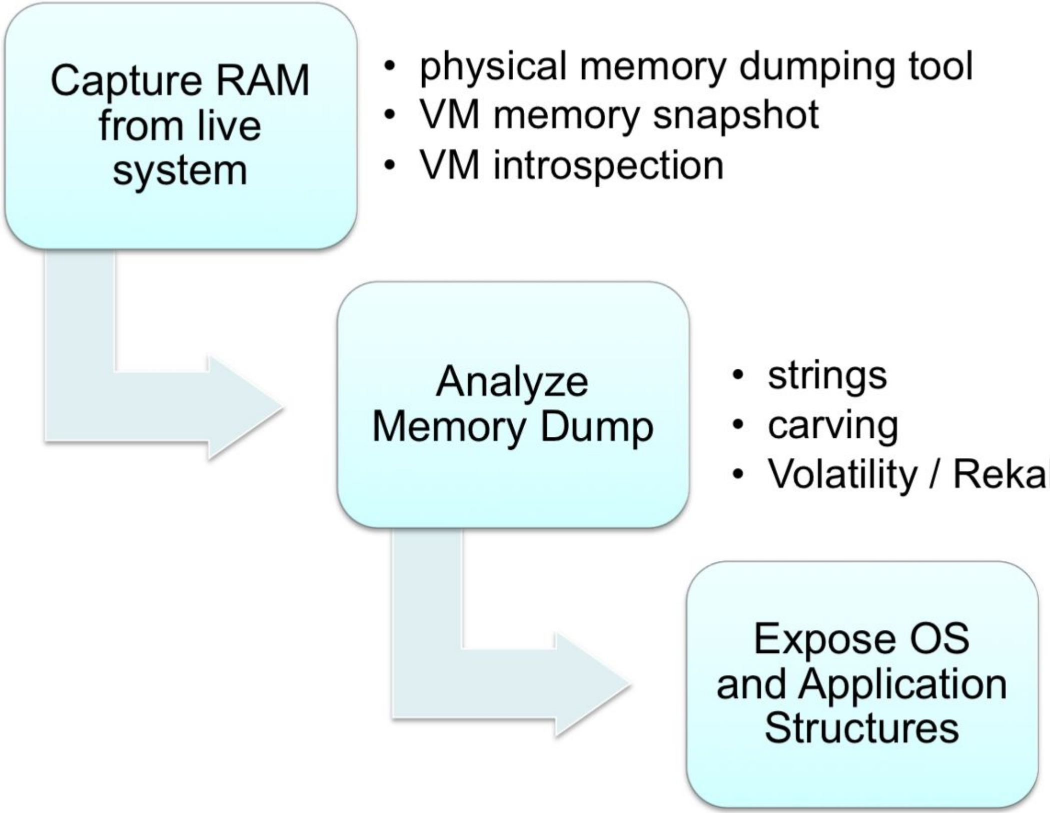
Windows Process Lists

```
$ python vol.py -f lab.mem --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.4
Offset(V)  Name                PID   PPID  Thds    Hnds  Sess  Start
---------- -------------- ------ ----- ----- ------- -----  -------------------
0x823c8830 System              4      0    56     537 -----
0x81e7e180 smss.exe          580      4     3      19 -----  2013-03-14 03:02:22
0x82315da0 csrss.exe         644    580    10     449     0  2013-03-14 03:02:25
0x81f37948 winlogon.exe      668    580    18     515     0  2013-03-14 03:02:26
0x81fec128 services.exe      712    668    15     281     0  2013-03-14 03:02:27
[snip]
0x81eb4300 vmtoolsd.exe     1684   1300     6     213     0  2013-03-14 03:02:45
0x8210b9c8 IEXPLORE.EXE     1764   1300    16     642     0  2013-03-14 03:03:04
0x81e79020 firefox.exe       180   1300    27     447     0  2013-03-14 03:03:05
0x81cb63d0 wuauclt.exe      1576   1072     3     104     0  2013-03-14 03:03:40
0x81e86bf8 alg.exe          1836    712     5     102     0  2013-03-14 03:04:00
0x8209eda0 wscntfy.exe      2672   1072     1      28     0  2013-03-14 03:04:01
0x82013340 jucheck.exe      2388   1656     2     104     0  2013-03-14 03:07:45
0x81e79418 thunderbird.exe  3832   1300    30     339     0  2013-03-14 03:12:54
0x8202b398 AcroRd32.exe     3684    180     0 -------     0  2013-03-14 14:19:16
0x81ecd3c0 cmd.exe          3812   3684     1      33     0  2013-03-14 14:19:29
0x81f55bd0 a[1].php         2280   3812     1     139     0  2013-03-14 14:19:30
0x8223b738 IEXPLORE.EXE     2276   2280     7     280     0  2013-03-14 14:19:32
0x822c8a58 AcroRd32.exe     2644    180     0 -------     0  2013-03-14 14:40:16
```

Parent / Child Relationships

```
$ python vol.py -f lab.mem --profile=WinXPSP3x86 pstree
Volatility Foundation Volatility Framework 2.4

[snip]

0x82263378:explorer.exe              1300   1188     11    363 2013-03-14 03:02:42
. 0x81e85da0:TSVNCache.exe           1556   1300      7     53 2013-03-14 03:02:43
. 0x81e79020:firefox.exe              180   1300     27    447 2013-03-14 03:03:05
.. 0x8202b398:AcroRd32.exe           3684    180      0 ------ 2013-03-14 14:19:16
... 0x81ecd3c0:cmd.exe               3812   3684      1     33 2013-03-14 14:19:29
.... 0x81f55bd0:a[1].php             2280   3812      1    139 2013-03-14 14:19:30
..... 0x8223b738:IEXPLORE.EXE        2276   2280      7    280 2013-03-14 14:19:32
```

Memory Forensics Workflow



Capture RAM from live system

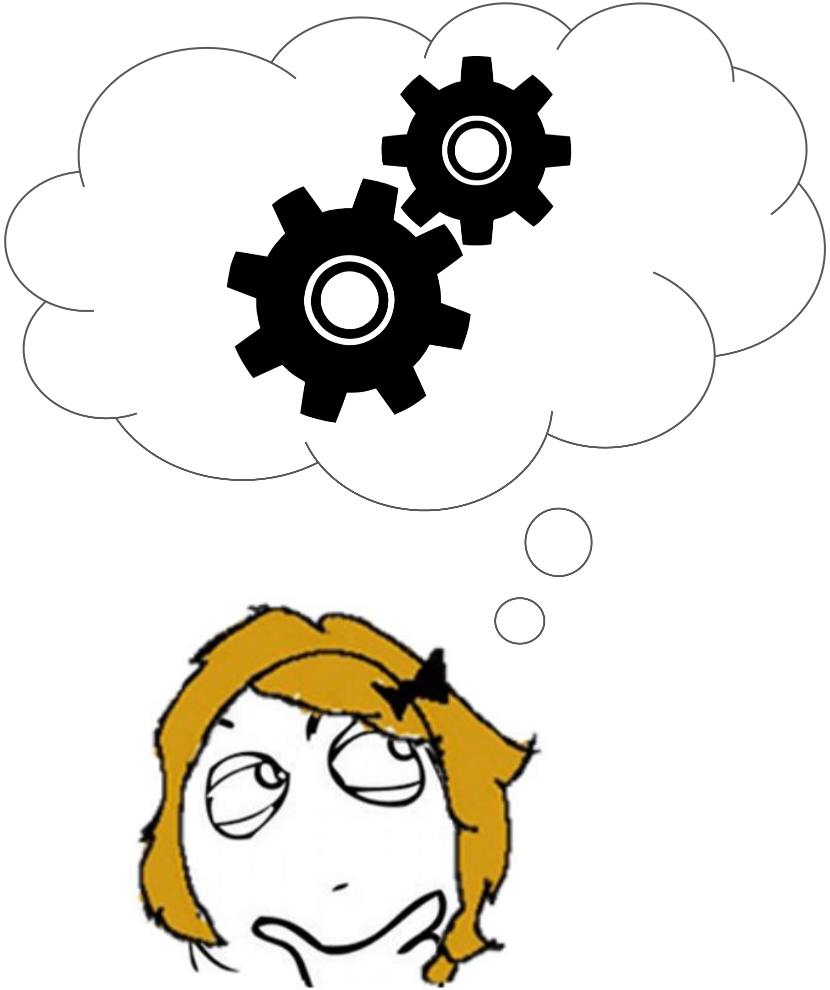
- physical memory dumping tool
- VM memory snapshot
- VM introspection

Analyze Memory Dump

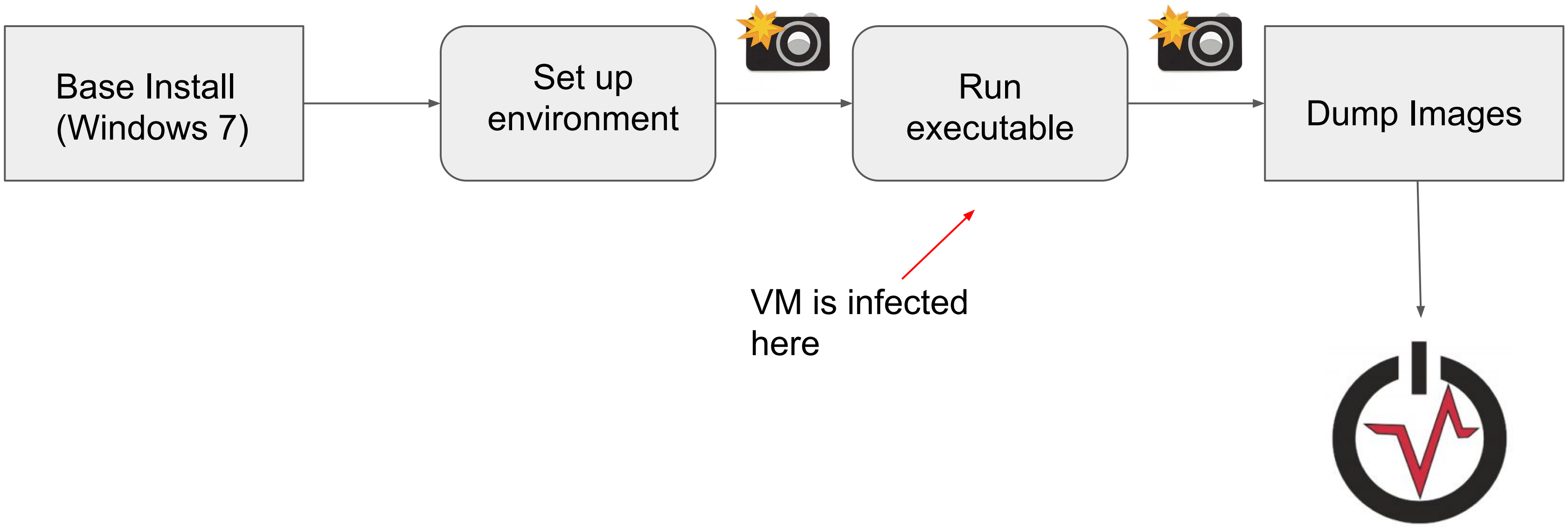
- strings
- carving
- Volatility / Rekall

Expose OS and Application Structures

- to yield useful evidence



A Pipeline



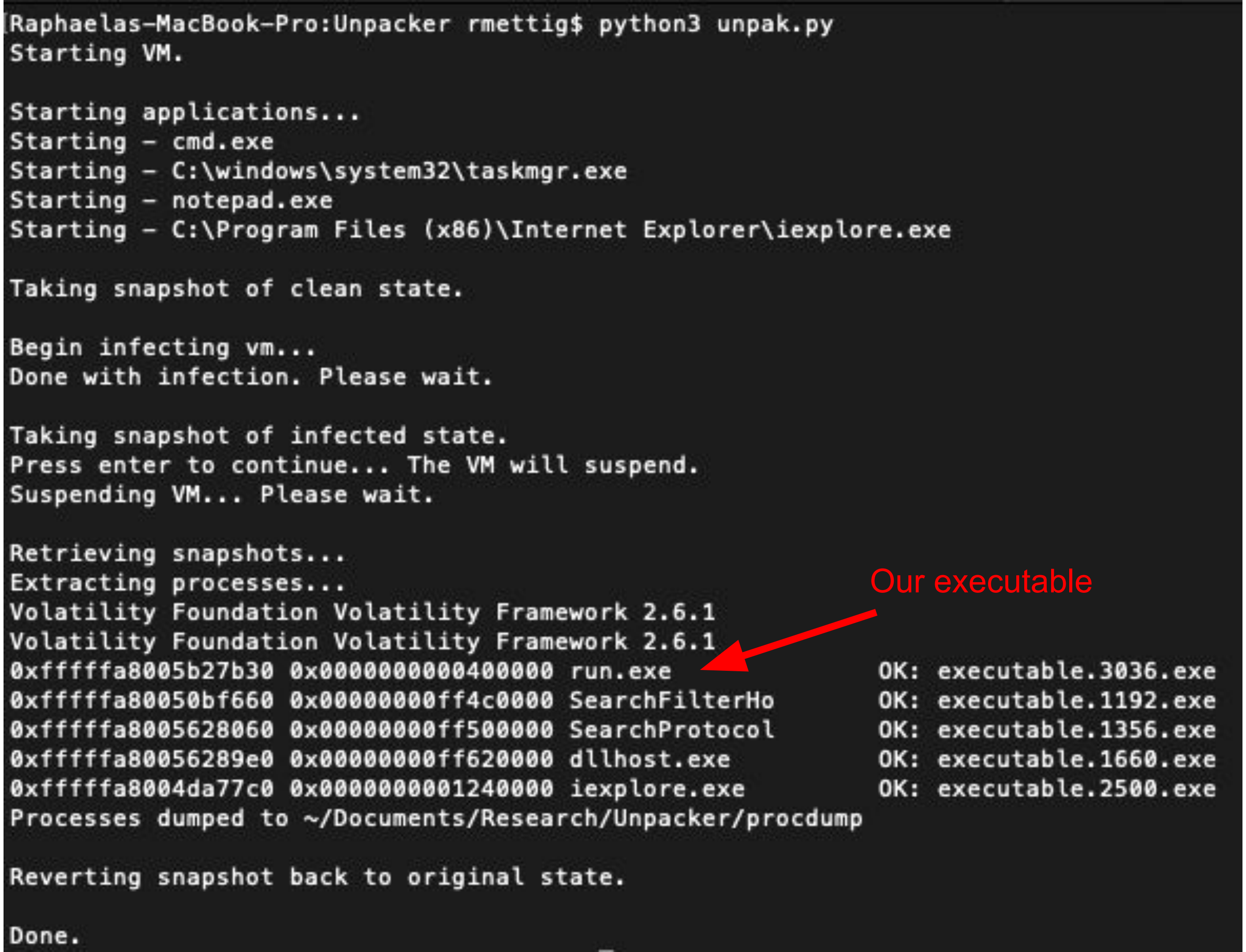
Base Install (Windows 7)

Set up environment

Run executable

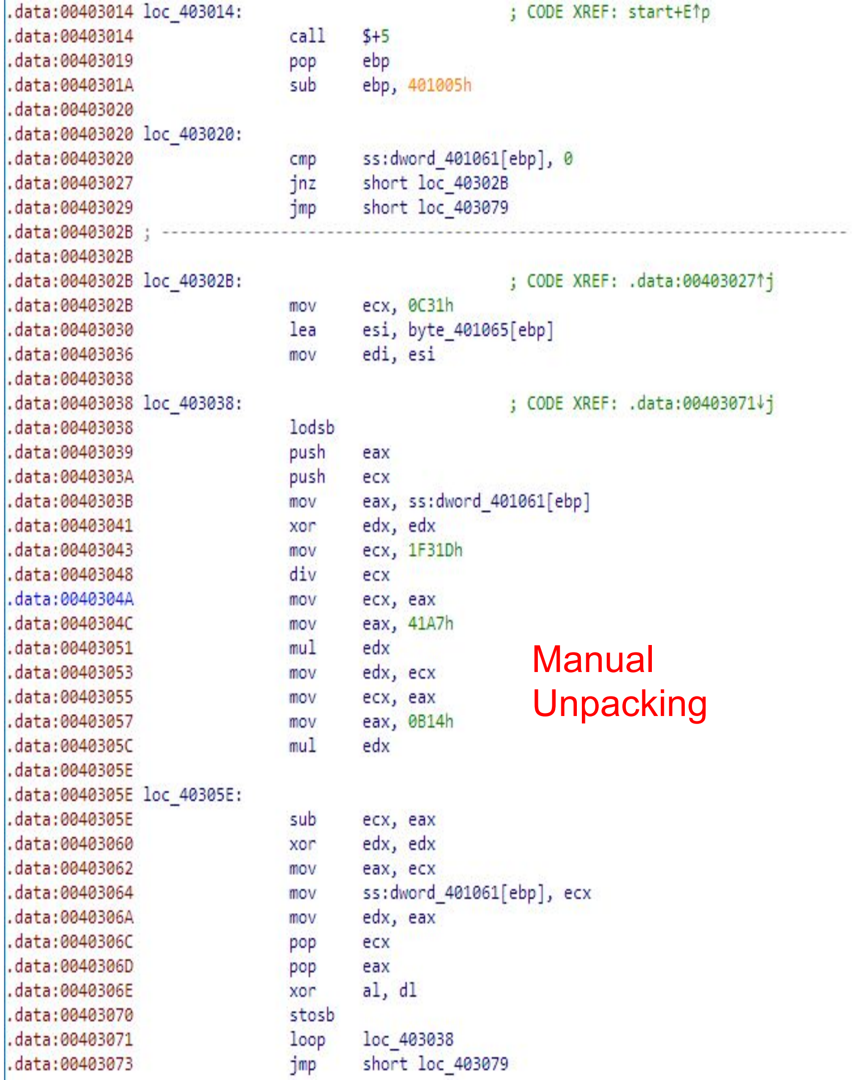
Dump Images

VM is infected here



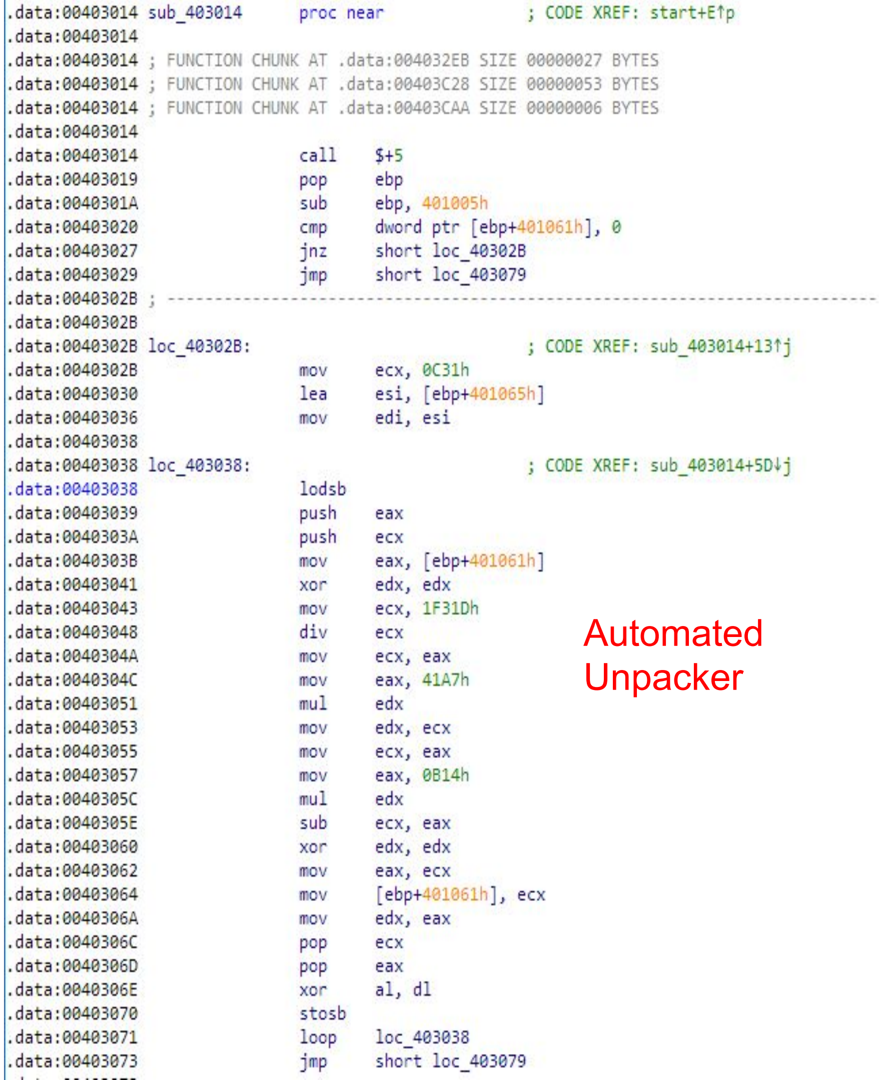
```
[Raphaelas-MacBook-Pro:Unpacker rmettig$ python3 unpak.py
Starting VM.

Starting applications...
Starting - cmd.exe
Starting - C:\windows\system32\taskmgr.exe
Starting - notepad.exe
Starting - C:\Program Files (x86)\Internet Explorer\iexplore.exe

Taking snapshot of clean state.

Begin infecting vm...
Done with infection. Please wait.

Taking snapshot of infected state.
Press enter to continue... The VM will suspend.
Suspending VM... Please wait.

Retrieving snapshots...
Extracting processes...
Volatility Foundation Volatility Framework 2.6.1
Volatility Foundation Volatility Framework 2.6.1
0xffffa8005b27b30 0x0000000000400000 run.exe               OK: executable.3036.exe
0xffffa80050bf660 0x00000000ff4c0000 SearchFilterHo        OK: executable.1192.exe
0xffffa8005628060 0x00000000ff500000 SearchProtocol        OK: executable.1356.exe
0xffffa80056289e0 0x00000000ff620000 dllhost.exe           OK: executable.1660.exe
0xffffa8004da77c0 0x0000000001240000 iexplore.exe          OK: executable.2500.exe
Processes dumped to ~/Documents/Research/Unpacker/procdump

Reverting snapshot back to original state.

Done.
```

Our executable



```
.data:00403014 loc_403014:                          ; CODE XREF: start+E↑p
.data:00403014                 call    $+5
.data:00403019                 pop     ebp
.data:0040301A                 sub     ebp, 401005h
.data:00403020
.data:00403020 loc_403020:
.data:00403020                 cmp     ss:dword_401061[ebp], 0
.data:00403027                 jnz     short loc_40302B
.data:00403029                 jmp     short loc_403079
.data:0040302B ; ---------------------------------------------------------------------------
.data:0040302B
.data:0040302B loc_40302B:                          ; CODE XREF: .data:00403027↑j
.data:0040302B                 mov     ecx, 0C31h
.data:00403030                 lea     esi, byte_401065[ebp]
.data:00403036                 mov     edi, esi
.data:00403038
.data:00403038 loc_403038:                          ; CODE XREF: .data:00403071↓j
.data:00403038                 lodsb
.data:00403039                 push    eax
.data:0040303A                 push    ecx
.data:0040303B                 mov     eax, ss:dword_401061[ebp]
.data:00403041                 xor     edx, edx
.data:00403043                 mov     ecx, 1F31Dh
.data:00403048                 div     ecx
.data:0040304A                 mov     ecx, eax
.data:0040304C                 mov     eax, 41A7h
.data:00403051                 mul     edx
.data:00403053                 mov     edx, ecx
.data:00403055                 mov     ecx, eax
.data:00403057                 mov     eax, 0B14h
.data:0040305C                 mul     edx
.data:0040305E
.data:0040305E loc_40305E:
.data:0040305E                 sub     ecx, eax
.data:00403060                 xor     edx, edx
.data:00403062                 mov     eax, ecx
.data:00403064                 mov     ss:dword_401061[ebp], ecx
.data:0040306A                 mov     edx, eax
.data:0040306C                 pop     ecx
.data:0040306D                 pop     eax
.data:0040306E                 xor     al, dl
.data:00403070                 stosb
.data:00403071                 loop    loc_403038
.data:00403073                 jmp     short loc_403079
```

Manual Unpacking



```
.data:00403014 sub_403014      proc near               ; CODE XREF: start+E↑p
.data:00403014
.data:00403014 ; FUNCTION CHUNK AT .data:004032EB SIZE 00000027 BYTES
.data:00403014 ; FUNCTION CHUNK AT .data:00403C28 SIZE 00000053 BYTES
.data:00403014 ; FUNCTION CHUNK AT .data:00403CAA SIZE 00000006 BYTES
.data:00403014
.data:00403014                 call    $+5
.data:00403019                 pop     ebp
.data:0040301A                 sub     ebp, 401005h
.data:00403020                 cmp     dword ptr [ebp+401061h], 0
.data:00403027                 jnz     short loc_40302B
.data:00403029                 jmp     short loc_403079
.data:0040302B ; ---------------------------------------------------------------------------
.data:0040302B
.data:0040302B loc_40302B:                             ; CODE XREF: sub_403014+13↑j
.data:0040302B                 mov     ecx, 0C31h
.data:00403030                 lea     esi, [ebp+401065h]
.data:00403036                 mov     edi, esi
.data:00403038
.data:00403038 loc_403038:                             ; CODE XREF: sub_403014+5D↓j
.data:00403038                 lodsb
.data:00403039                 push    eax
.data:0040303A                 push    ecx
.data:0040303B                 mov     eax, [ebp+401061h]
.data:00403041                 xor     edx, edx
.data:00403043                 mov     ecx, 1F31Dh
.data:00403048                 div     ecx
.data:0040304A                 mov     ecx, eax
.data:0040304C                 mov     eax, 41A7h
.data:00403051                 mul     edx
.data:00403053                 mov     edx, ecx
.data:00403055                 mov     ecx, eax
.data:00403057                 mov     eax, 0B14h
.data:0040305C                 mul     edx
.data:0040305E                 sub     ecx, eax
.data:00403060                 xor     edx, edx
.data:00403062                 mov     eax, ecx
.data:00403064                 mov     [ebp+401061h], ecx
.data:0040306A                 mov     edx, eax
.data:0040306C                 pop     ecx
.data:0040306D                 pop     eax
.data:0040306E                 xor     al, dl
.data:00403070                 stosb
.data:00403071                 loop    loc_403038
.data:00403073                 jmp     short loc_403079
```

Automated Unpacker

Future Work

- Support different OSs (Windows only, at the moment)
- Tailored Volatility Plugins, not just for unpacking
- Volume

That's all Folks!