IBM Security

Intelligence. Integration. Expertise.

IBM

# Understanding the Attack Surface and Attack Resilience of Project Spartan’s (Edge) New EdgeHTML Rendering Engine

**Mark Vincent Yason**

IBM X-Force Advanced Research

yasonm[at]ph[dot]ibm[dot]com

@MarkYason

[v1]

# Agenda

- Overview
- Attack Surface
- Exploit Mitigations
- Conclusion

# Notes

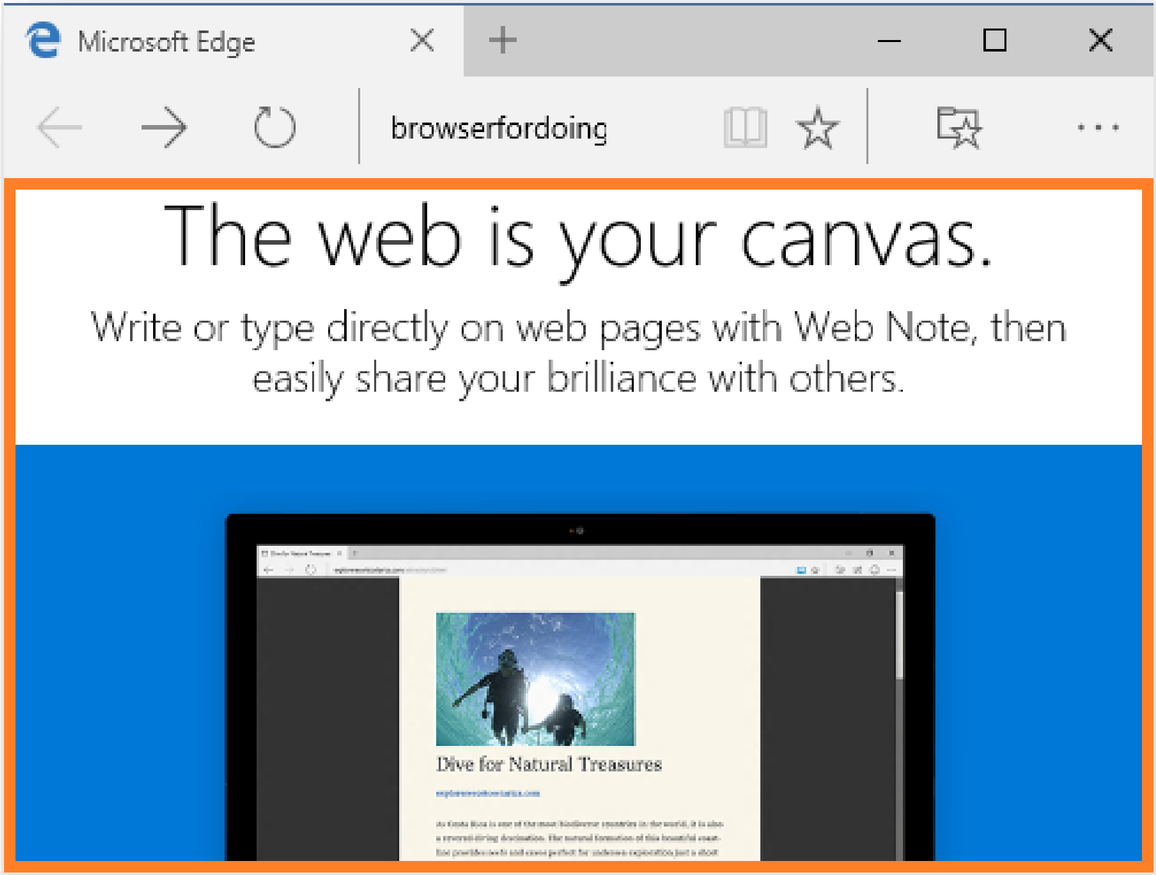
- Detailed whitepaper is available
- Technical information are based on Edge running on 64-bit Windows 10 TP10074 (edgehtml.dll version 11.0.10074.0)
- Edge content process name and process screenshots are based on Edge running on 64-bit Windows 10 TP10240

IBM Security

Intelligence. Integration. Expertise.

# Overview

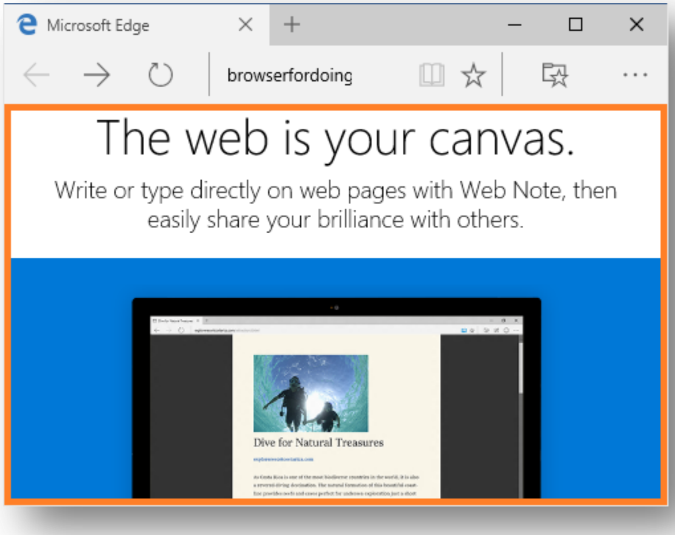
# Overview > EdgeHTML Rendering Engine



The web is your canvas.

Write or type directly on web pages with Web Note, then easily share your brilliance with others.

# Overview > EdgeHTML Attack Surface Map & Exploit Mitigations



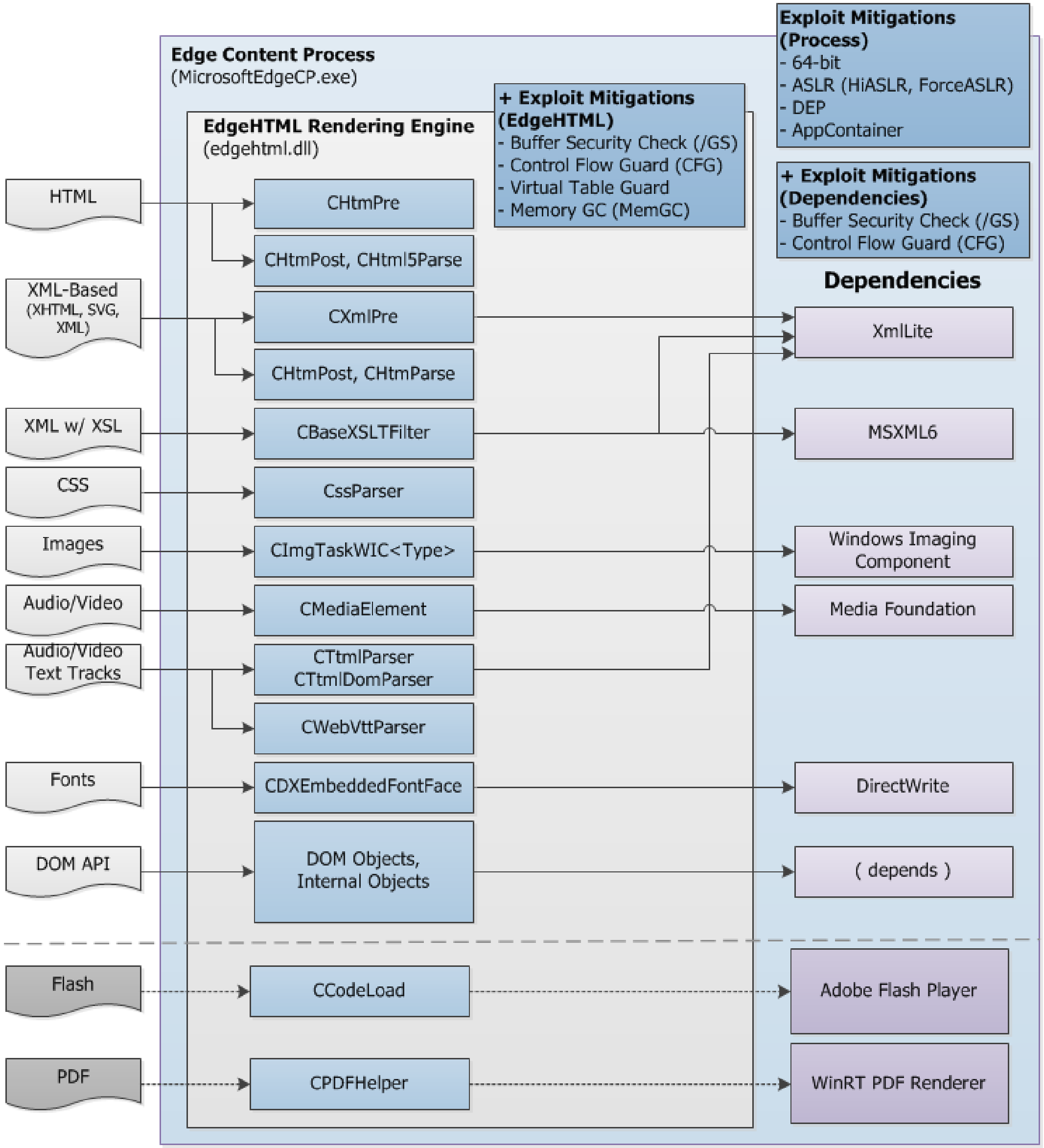
Microsoft Edge

browserfordoing

The web is your canvas.

Write or type directly on web pages with Web Note, then easily share your brilliance with others.

Dive for Natural Treasures



**Edge Content Process**
(MicrosoftEdgeCP.exe)

**EdgeHTML Rendering Engine**
(edgehtml.dll)

**+ Exploit Mitigations (EdgeHTML)**
- Buffer Security Check (/GS)
- Control Flow Guard (CFG)
- Virtual Table Guard
- Memory GC (MemGC)

**Exploit Mitigations (Process)**
- 64-bit
- ASLR (HiASLR, ForceASLR)
- DEP
- AppContainer

**+ Exploit Mitigations (Dependencies)**
- Buffer Security Check (/GS)
- Control Flow Guard (CFG)

**Dependencies**

HTML → CHtmPre, CHtmPost, CHtml5Parse

XML-Based (XHTML, SVG, XML) → CXmlPre, CHtmPost, CHtmParse

CXmlPre → XmlLite

XML w/ XSL → CBaseXSLTFilter → XmlLite, MSXML6

CSS → CssParser

Images → CImgTaskWIC<Type> → Windows Imaging Component

Audio/Video → CMediaElement → Media Foundation

Audio/Video Text Tracks → CTtmlParser CTtmlDomParser, CWebVttParser

CTtmlParser CTtmlDomParser → XmlLite

Fonts → CDXEmbeddedFontFace → DirectWrite

DOM API → DOM Objects, Internal Objects → ( depends )

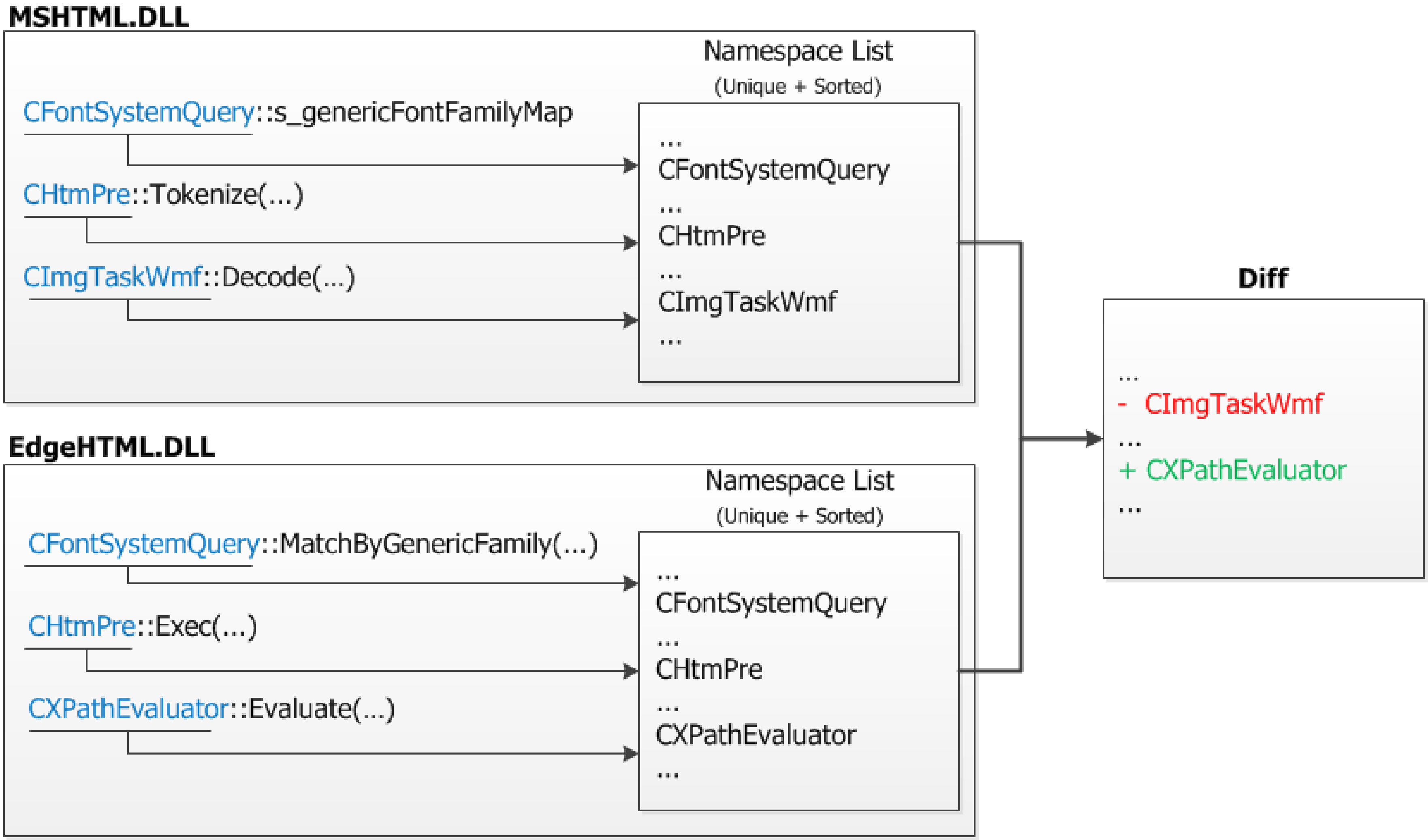
Flash → CCodeLoad → Adobe Flash Player

PDF → CPDFHelper → WinRT PDF Renderer

# Overview > Initial Recon: MSHTML and EdgeHTML

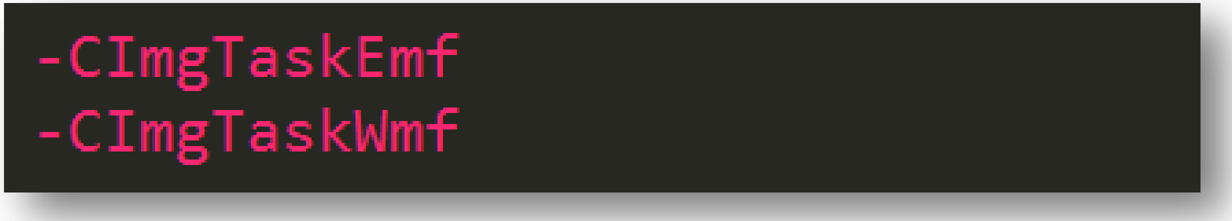
- EdgeHTML is forked from Trident (MSHTML)
- Problem: Quickly identify major code changes (features) from MSHTML to EdgeHTML
- One option: Diff classes and namespaces

# Overview > Initial Recon: Diffing MSHTML and EdgeHTML (Method)



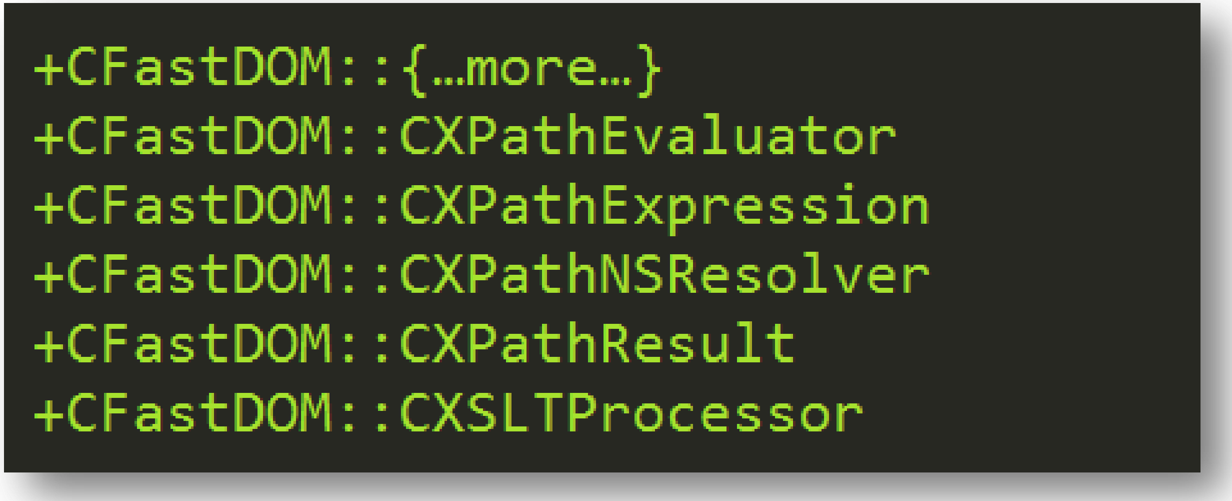
# Overview > Initial Recon: Diffing MSHTML and EdgeHTML (Examples)

- Suggests change in image support:



```
-CImgTaskEmf
-CImgTaskWmf
```

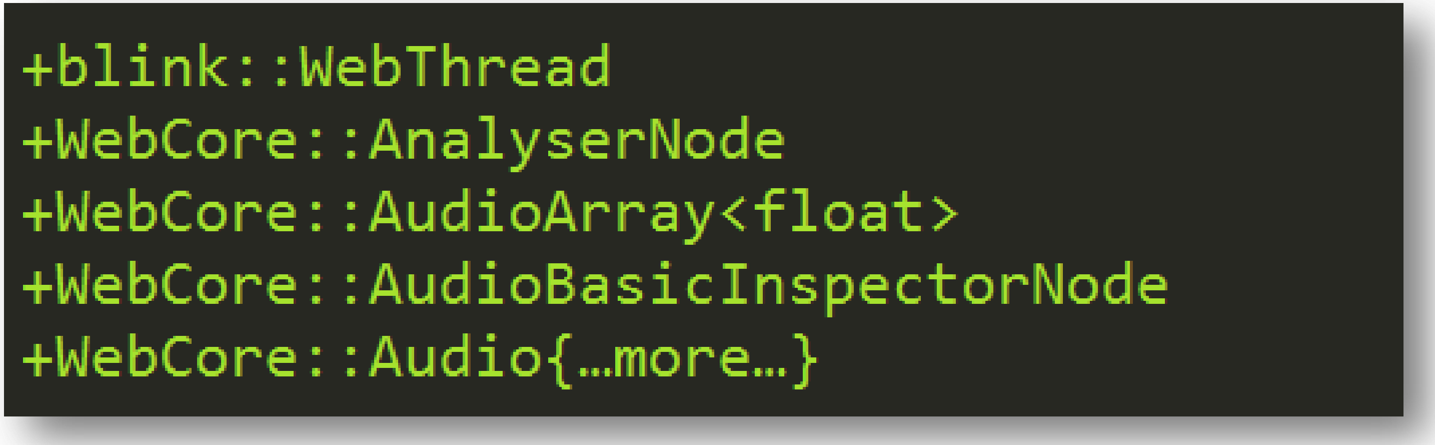
- Suggests new DOM object types:



```
+CFastDOM::{…more…}
+CFastDOM::CXPathEvaluator
+CFastDOM::CXPathExpression
+CFastDOM::CXPathNSResolver
+CFastDOM::CXPathResult
+CFastDOM::CXSLTProcessor
```

# Overview > Initial Recon: Diffing MSHTML and EdgeHTML (Examples)

- Suggests ported code from another rendering engine (Blink) for Web Audio support:



```
+blink::WebThread
+WebCore::AnalyserNode
+WebCore::AudioArray<float>
+WebCore::AudioBasicInspectorNode
+WebCore::Audio{…more…}
```

# Overview > Initial Recon: Diffing MSHTML and EdgeHTML (Notes)

- Further analysis needed
  - Renamed class/namespace results into a new namespace plus a deleted namespace
- Requires availability of symbols
  - Bindiffing is another option
- Same rudimentary diffing method can be applied to:
  - Functions, Methods
  - Strings
  - Imports, Exports

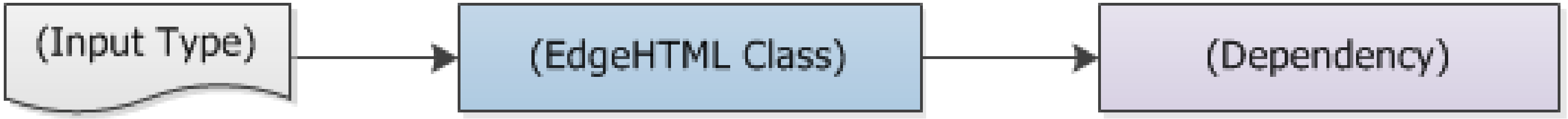
IBM Security

Intelligence. Integration. Expertise.

# Attack Surface

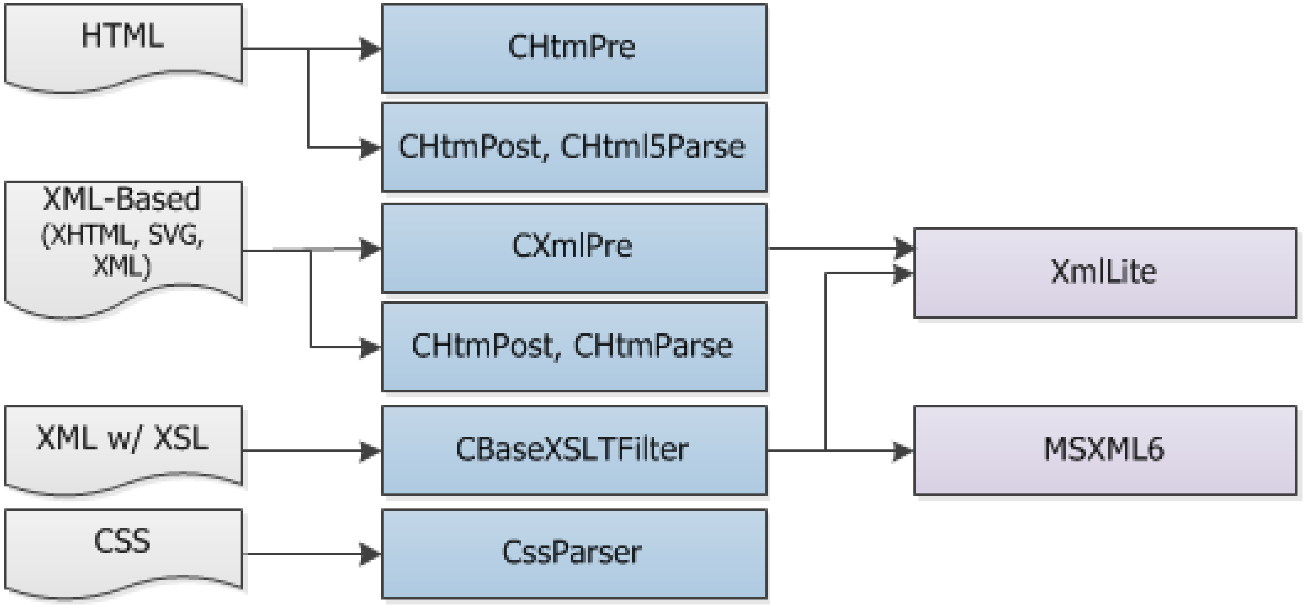
# Attack Surface

- Legend for the next slides



- EdgeHTML class is the entry point for parsing/processing
  - Most use other EdgeHTML classes
  - Analysis can start by setting a breakpoint on the listed EdgeHTML class methods, i.e.:
    - (WinDbg)> bm edgehtml!CXmlPre::*

# Attack Surface > Markup/Style Parsing



- HTML & CSS parsing are done by EdgeHTML classes
- XmlLite is used for parsing XML-based markups
- MSXML6 is used for XML transformation
- VML support (binary behaviors) was removed in EdgeHTML

# Attack Surface > Markup/Style Parsing > XmlLite

XmlLite

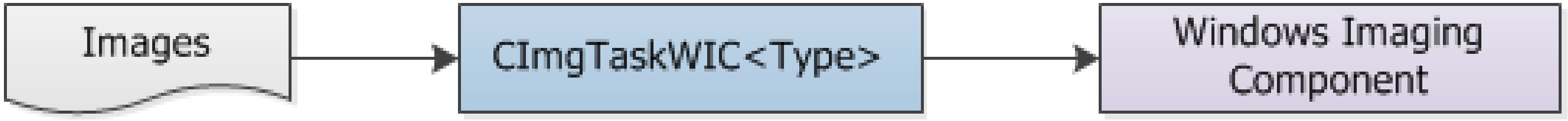
- Lightweight XML parser
- Built-in Windows component
- IXmlReader interface is used by EdgeHTML for reading nodes from XML-based markups

# Attack Surface > Markup/Style Parsing > MSXML6

MSXML6

- Comprehensive XML parser
- Built-in Windows component
- IXMLDOMDocument interface is used by EdgeHTML for transforming XML that references an XSL stylesheet

# Attack Surface > Image Decoding



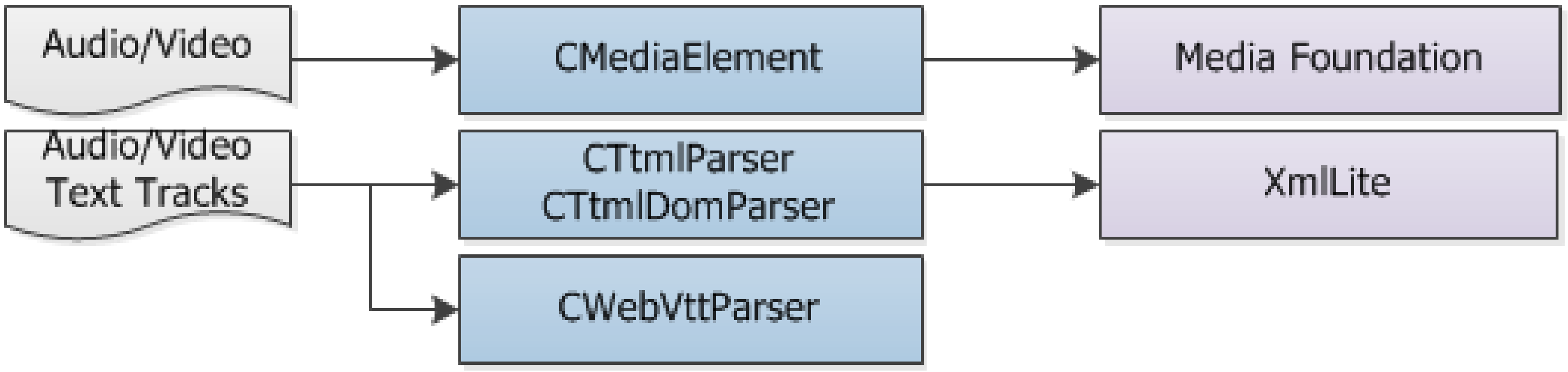
- Reachable via: direct link, <img>, <embed>
- Supported image formats: g_rgMimeInfoImg
- PNG, JPG, GIF, DDS, TIFF, BMP, HDP, ICO decoding via Windows Imaging Component (WIC)
- WMF and EMF support via GDI32 was removed in EdgeHTML

# Attack Surface > Image Decoding > Windows Imaging Component (WIC)

Windows Imaging Component

- Image decoder/encoder for multiple image formats
- Built-in Windows component
- IWICImagingFactory->CreateDecoder() is used by EdgeHTML to instantiate the decoder for a particular image format

# Attack Surface > Audio/Video Decoding



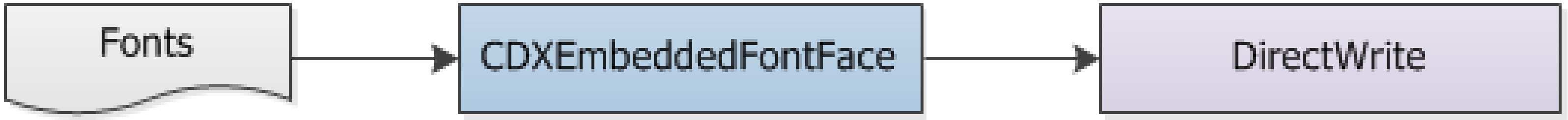
- Reachable via: direct link, <audio>, <video>, <track>
- Supported audio/video containers: g_rgMimeInfoAudio and g_rgMimeInfoVideo
- MP4, MP3, WAV support via Media Foundation (MF)
- TTML & WebVTT support for timed text tracks
  - –XmlLite is used for TTML parsing

# Attack Surface > Audio/Video Decoding > Media Foundation (MF)

Media Foundation

- Framework for audio/video processing
- Built-in Windows component
- IMFMediaEngine is used by EdgeHTML to setup the media source and control playback

# Attack Surface > Font Rendering



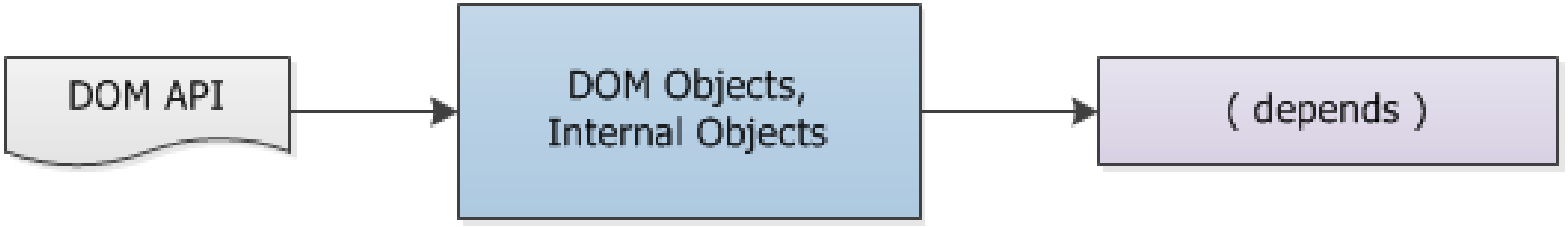
- Reachable via: @font-face CSS rule
- TTF, OTF and WOFF font support via DirectWrite
- EOT support was removed in EdgeHTML
  - Removed dependence to T2EMBED for EOT parsing

# Attack Surface > Font Rendering > DirectWrite

DirectWrite

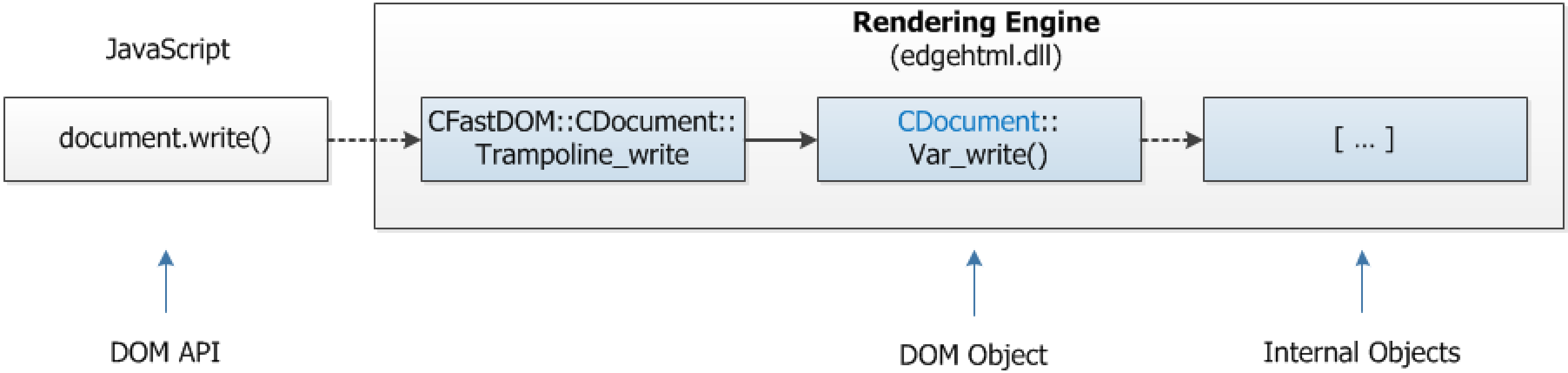
- DirectX text rendering API
- Built-in Windows component
- IDWriteFactory->CreateCustomFontFileReference() is used by EdgeHTML to register a custom private font
- DirectWrite is mentioned in the “One font vulnerability to rule them all” presentation [1]

# Attack Surface > DOM API



- Reachable via: JavaScript
- Large attack surface that:
  - Interacts directly with EdgeHTML DOM objects
  - Interacts indirectly with internal EdgeHTML objects and libraries (depends)

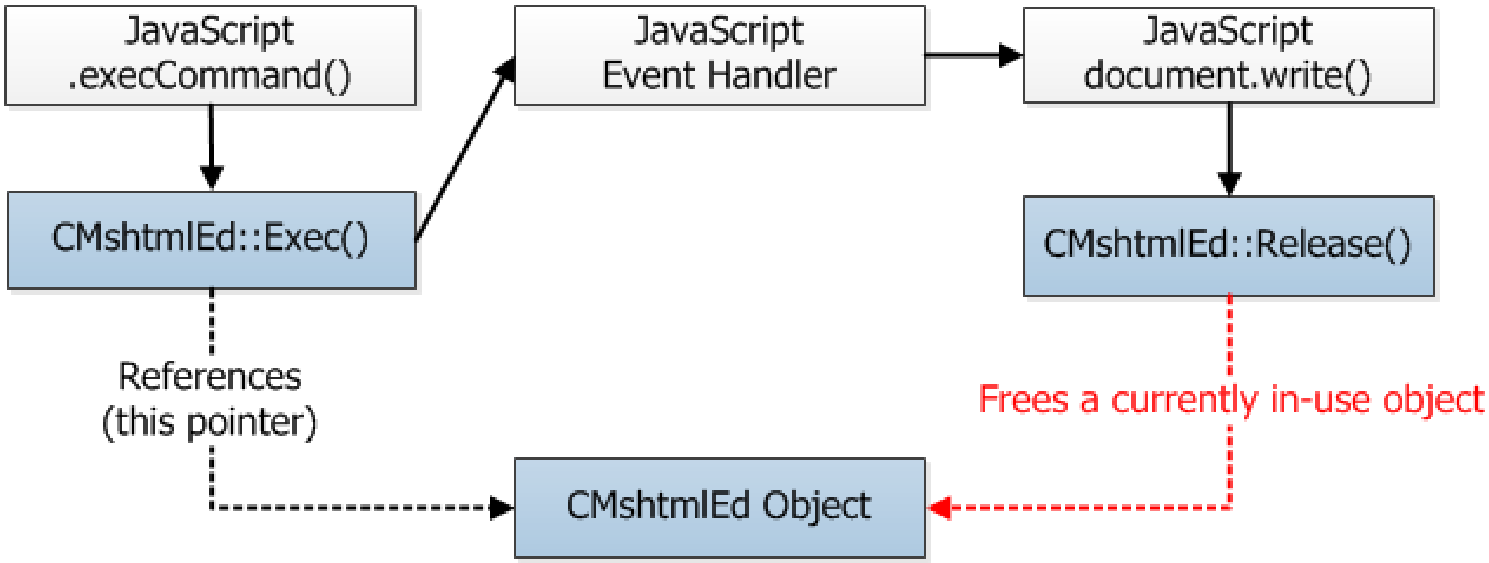
# Attack Surface > DOM API



- DOM API calls can change the state of the DOM tree, DOM objects and other EdgeHTML internal objects

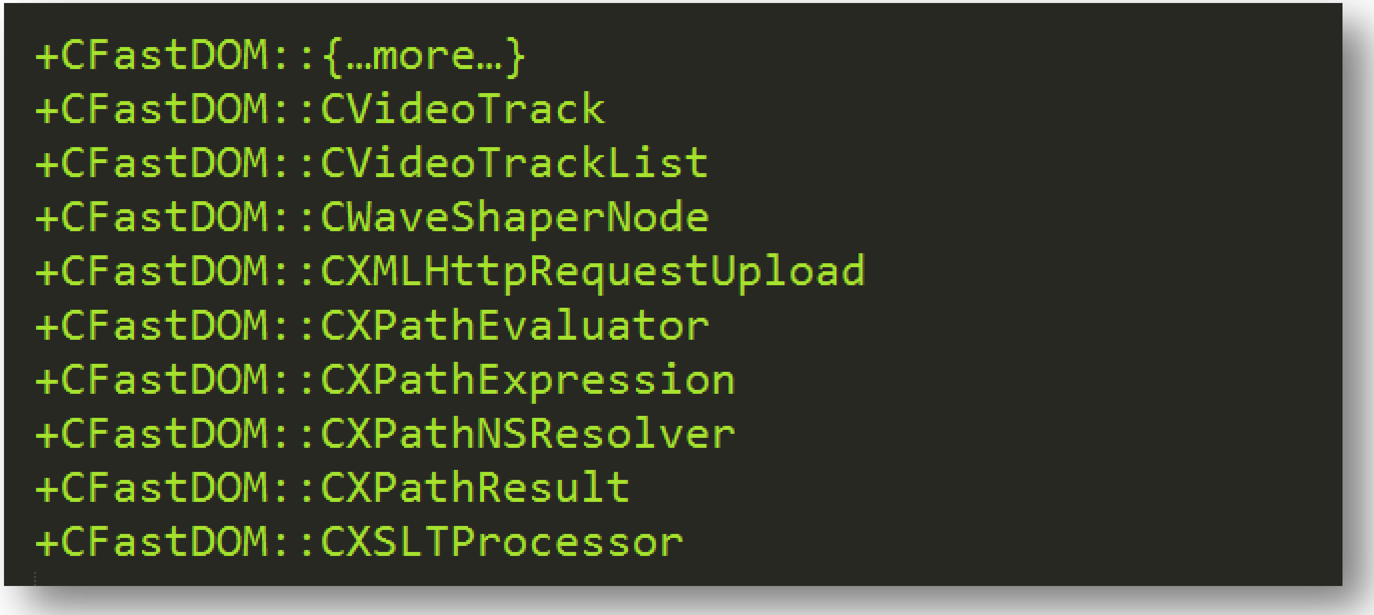
# Attack Surface > DOM API

CVE-2012-4969 (IE CMshtmlEd UAF)


JavaScript .execCommand()
JavaScript Event Handler
JavaScript document.write()
CMshtmlEd::Exec()
CMshtmlEd::Release()
References (this pointer)
Frees a currently in-use object
CMshtmlEd Object

- Unexpected input, unexpected state changes or incorrect state when a DOM API is called can result to memory corruption, such as: use-after-free (above), heap overflows, invalid pointer access, etc.

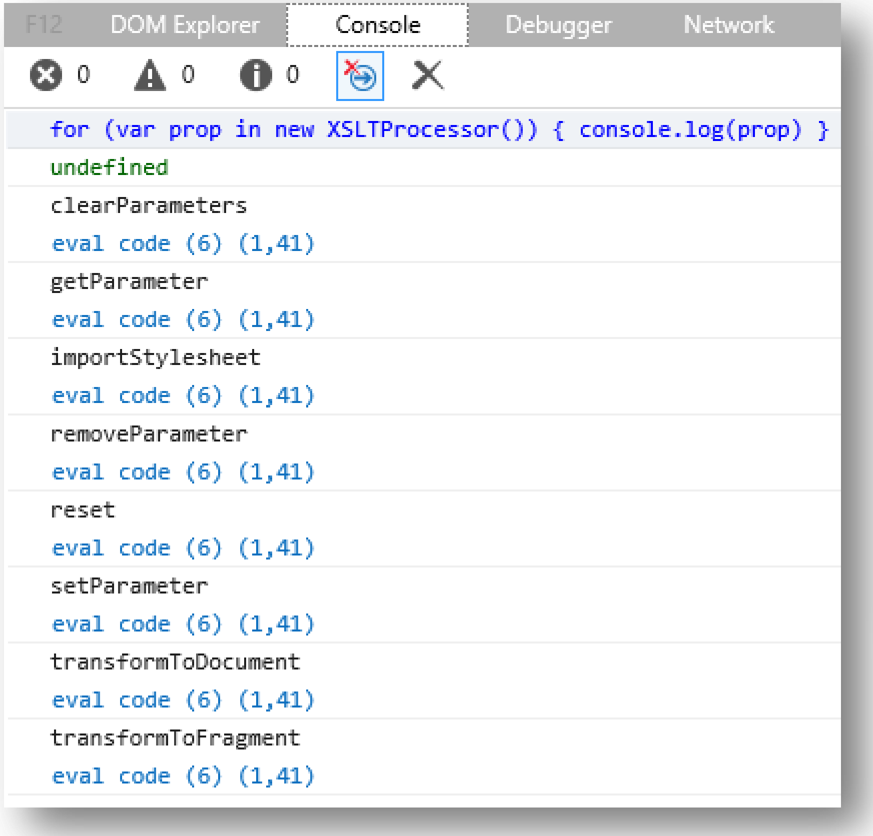
# Attack Surface > DOM API



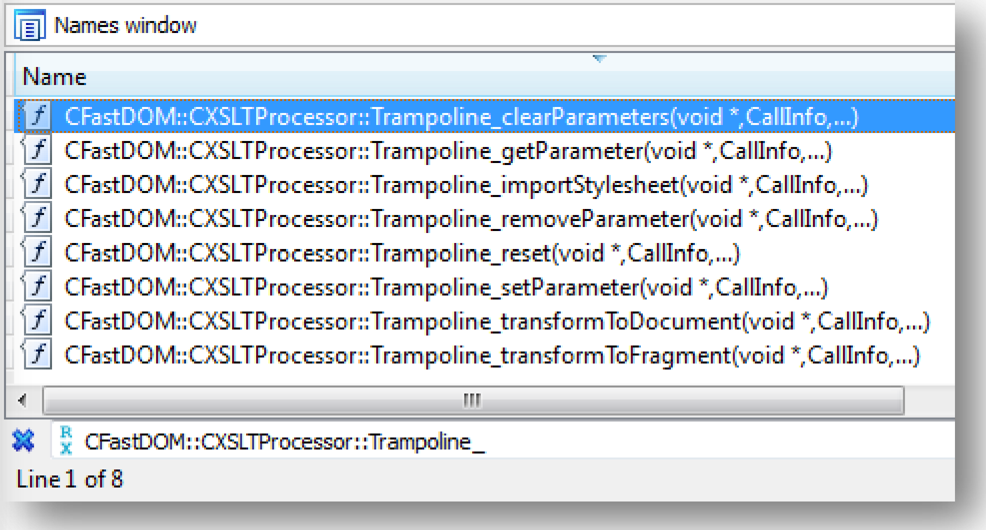
```
+CFastDOM::{…more…}
+CFastDOM::CVideoTrack
+CFastDOM::CVideoTrackList
+CFastDOM::CWaveShaperNode
+CFastDOM::CXMLHttpRequestUpload
+CFastDOM::CXPathEvaluator
+CFastDOM::CXPathExpression
+CFastDOM::CXPathNSResolver
+CFastDOM::CXPathResult
+CFastDOM::CXSLTProcessor
```

- Over 60+ new DOM object types were found in EdgeHTML
  - –New code or new code paths that are reachable

# Attack Surface > DOM API



```
for (var prop in new XSLTProcessor()) { console.log(prop) }
undefined
clearParameters
eval code (6) (1,41)
getParameter
eval code (6) (1,41)
importStylesheet
eval code (6) (1,41)
removeParameter
eval code (6) (1,41)
reset
eval code (6) (1,41)
setParameter
eval code (6) (1,41)
transformToDocument
eval code (6) (1,41)
transformToFragment
eval code (6) (1,41)
```



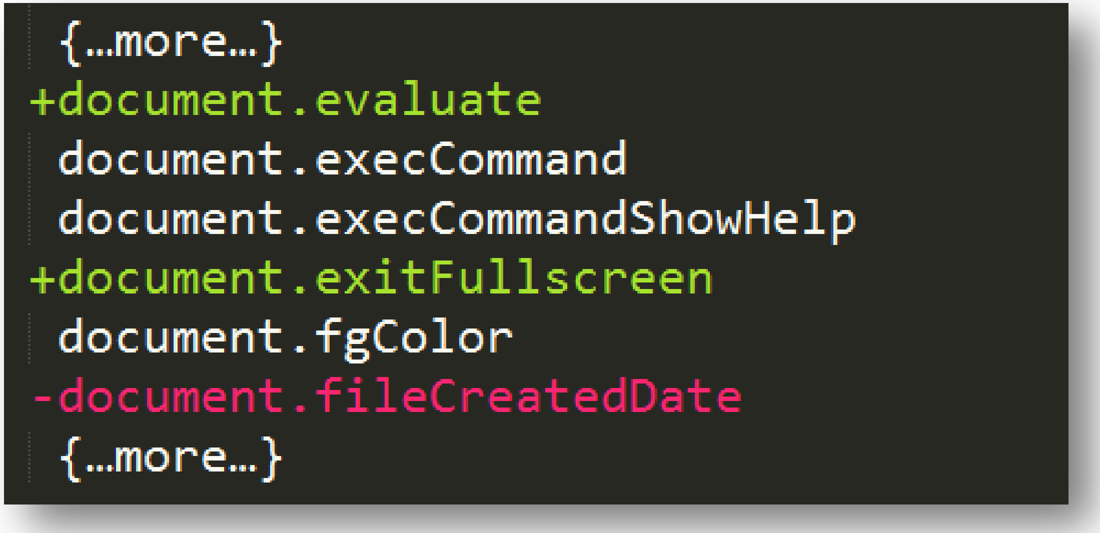
```
CFastDOM::CXSLTProcessor::Trampoline_clearParameters(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_getParameter(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_importStylesheet(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_removeParameter(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_reset(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_setParameter(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_transformToDocument(void *,CallInfo,...)
CFastDOM::CXSLTProcessor::Trampoline_transformToFragment(void *,CallInfo,...)
```

CFastDOM::CXSLTProcessor::Trampoline_

Line 1 of 8

- Enumerating DOM object properties/methods via JavaScript and IDA…

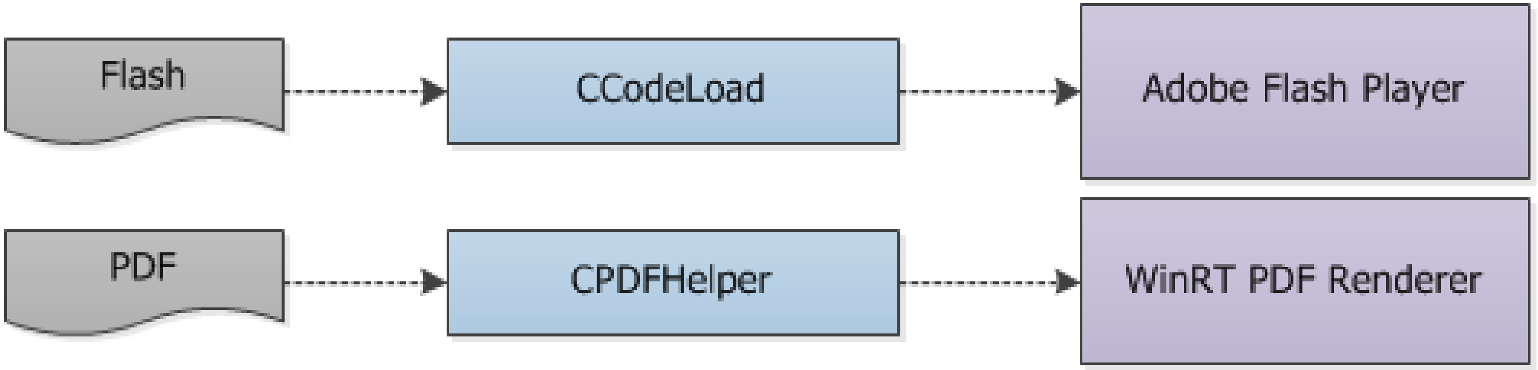
# Attack Surface > DOM API



```
 {…more…}
+document.evaluate
 document.execCommand
 document.execCommandShowHelp
+document.exitFullscreen
 document.fgColor
-document.fileCreatedDate
 {…more…}
```

- … and then diffing them to find out new properties / methods
  - New code or new code paths that are reachable

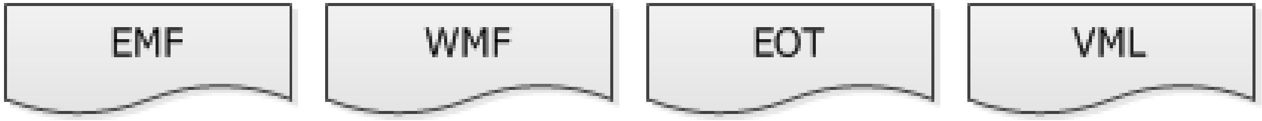
# Attack Surface > PDF and Flash Renderers



- Built-in/pre-installed complex renderers that can be instantiated by default
  - –Additional set of attack surface
  - –Functionalities can be repurposed for exploitation
    - CFG Bypass (Flash JIT) [2]
    - ASLR Bypass (Flash Vector object) [3]

# Attack Surface > Summary

- Well-known attack vectors were removed



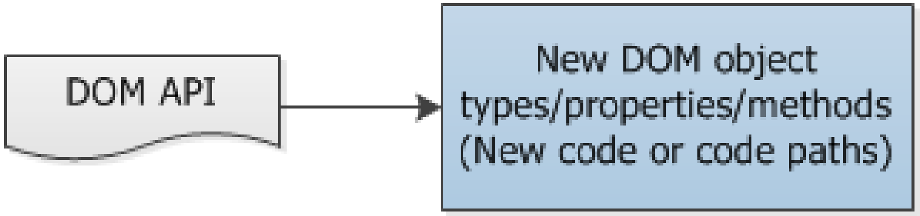
EMF

WMF

EOT

VML

- New attack vectors were found in the DOM API



DOM API

New DOM object types/properties/methods (New code or code paths)

- Remotely-reachable libraries thru EdgeHTML

XmlLite

MSXML6

Windows Imaging Component

WinRT PDF Renderer

Media Foundation

DirectWrite

Adobe Flash Player

IBM Security

Intelligence. Integration. Expertise.

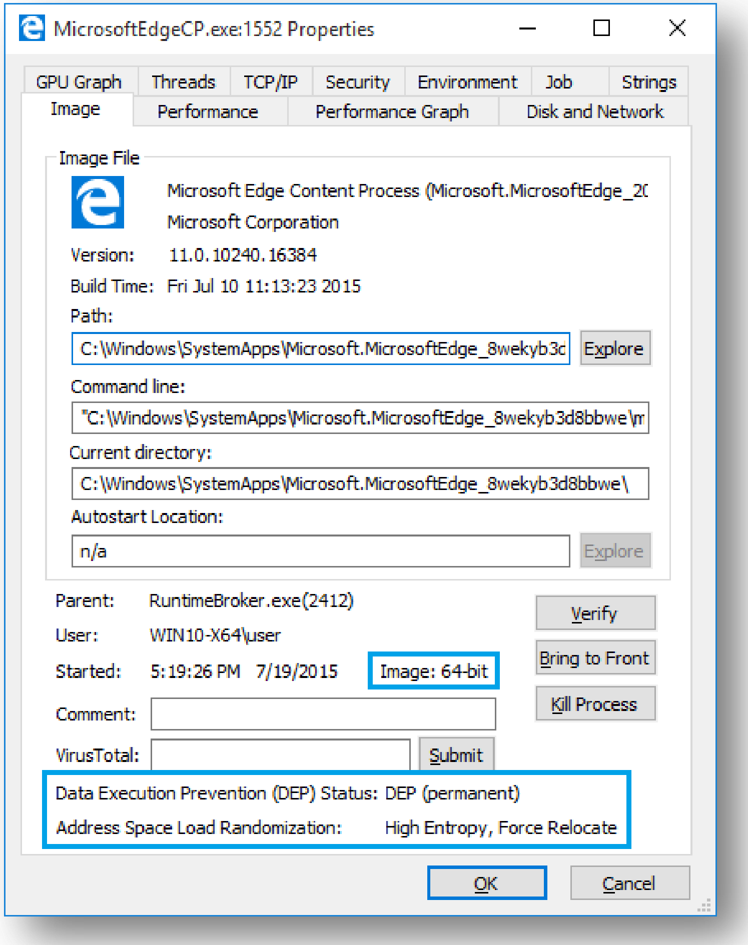
IBM

# Exploit Mitigations

# Exploit Mitigations

- Discussion of exploit mitigations applied to:
  - Content process where EdgeHTML is hosted
  - EdgeHTML and its dependencies
  - Specific to EdgeHTML
- Known/published bypass or weakness researched/discovered by various security researchers are discussed and [referenced]

# Exploit Mitigations > Edge Content Process



MicrosoftEdgeCP.exe:1552 Properties

Image File

Microsoft Edge Content Process (Microsoft.MicrosoftEdge_20

Microsoft Corporation

Version: 11.0.10240.16384

Build Time: Fri Jul 10 11:13:23 2015

Path: C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3c

Command line: "C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\m

Current directory: C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\

Autostart Location: n/a

Parent: RuntimeBroker.exe(2412)

User: WIN10-X64\user

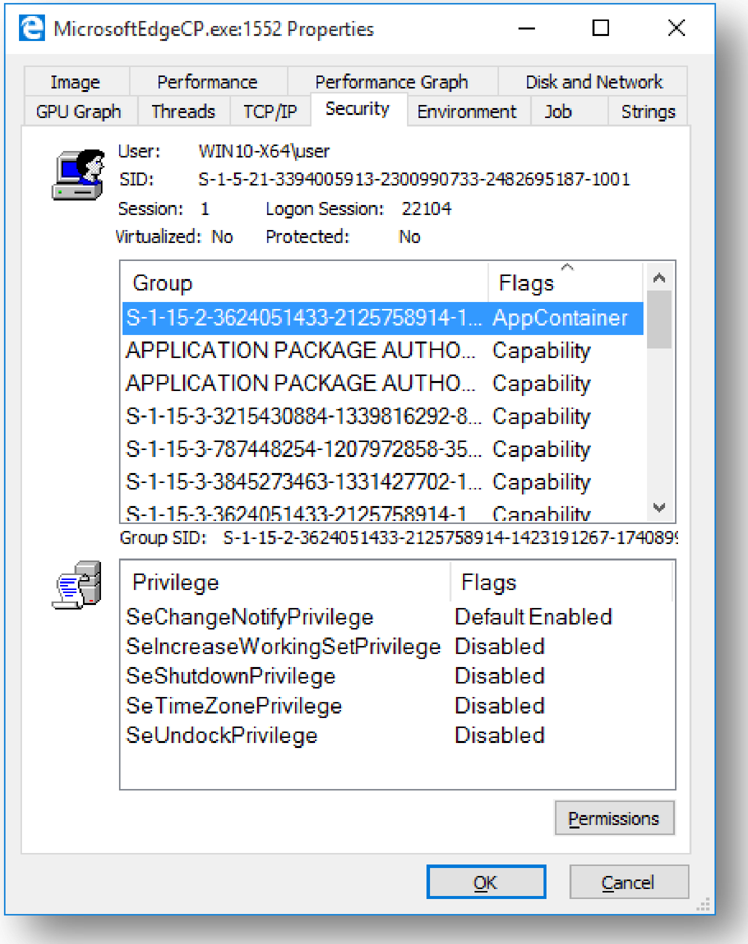
Started: 5:19:26 PM 7/19/2015 Image: 64-bit

Comment:

VirusTotal:

Data Execution Prevention (DEP) Status: DEP (permanent)

Address Space Load Randomization: High Entropy, Force Relocate



MicrosoftEdgeCP.exe:1552 Properties

User: WIN10-X64\user

SID: S-1-5-21-3394005913-2300990733-2482695187-1001

Session: 1 Logon Session: 22104

Virtualized: No Protected: No

| Group | Flags |
|---|---|
| S-1-15-2-3624051433-2125758914-1... | AppContainer |
| APPLICATION PACKAGE AUTHO... | Capability |
| APPLICATION PACKAGE AUTHO... | Capability |
| S-1-15-3-3215430884-1339816292-8... | Capability |
| S-1-15-3-787448254-1207972858-35... | Capability |
| S-1-15-3-3845273463-1331427702-1... | Capability |
| S-1-15-3-3624051433-2125758914-1 | Capability |

Group SID: S-1-15-2-3624051433-2125758914-1423191267-174089

| Privilege | Flags |
|---|---|
| SeChangeNotifyPrivilege | Default Enabled |
| SeIncreaseWorkingSetPrivilege | Disabled |
| SeShutdownPrivilege | Disabled |
| SeTimeZonePrivilege | Disabled |
| SeUndockPrivilege | Disabled |

- MicrosoftEdgeCP.exe: 64-bit, ASLR (HiASLR, ForceASLR), DEP, and AppContainer

# Exploit Mitigations > Content Process (Comparison)

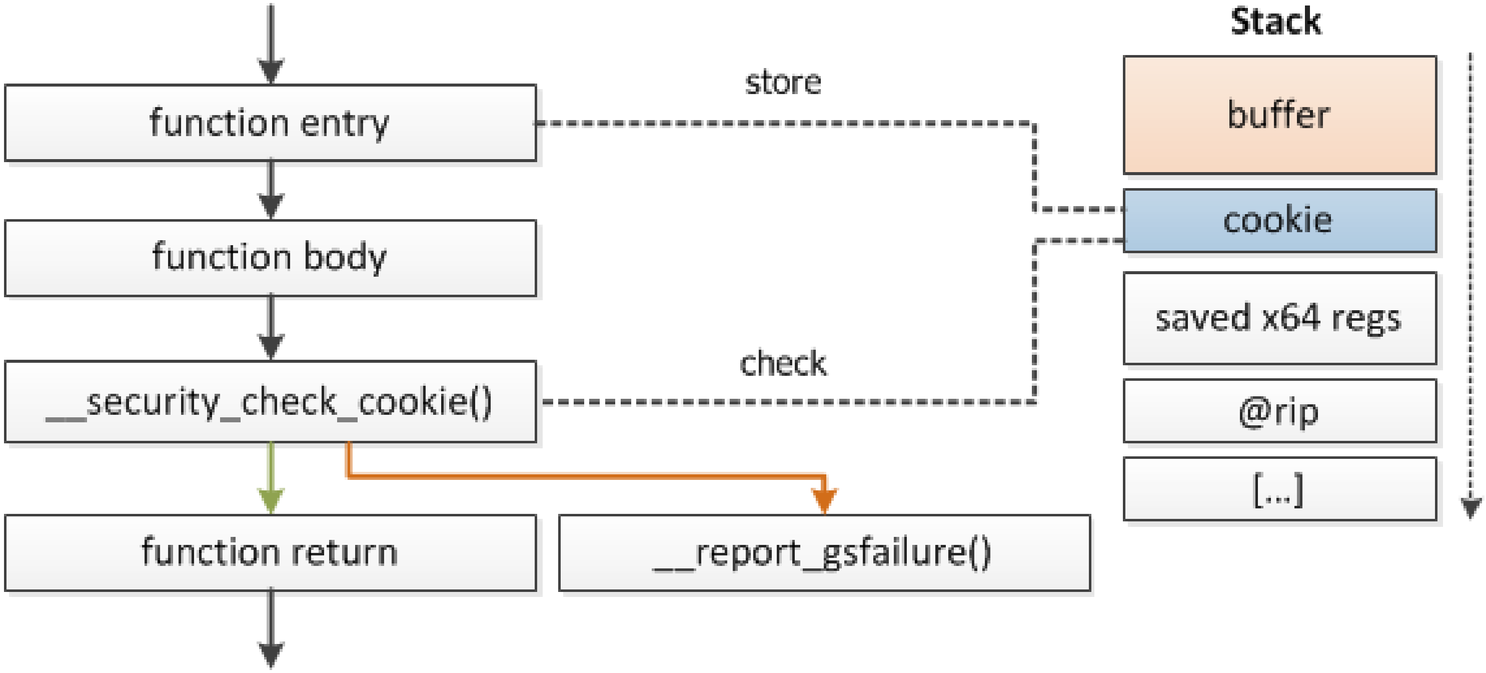
| | Win10/<br>Edge | Win10/<br>IE11/ | Win8/<br>ImmersiveIE | Win8/<br>IE11 | Win7/<br>IE11 |
|---|---|---|---|---|---|
| **64-bit** | **Yes** | **No** | **Yes** | **No** | **No** |
| **ASLR** | **Yes**<br>(HiASLR, ForceASLR) | **Yes**<br>(ForceASLR) | **Yes**<br>(HiASLR, ForceASLR) | **Yes**<br>(ForceASLR) | **Yes**<br>(ForceASLR) |
| **DEP** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |
| **Process Isolation** | **AppContainer** | **Low Integrity** | **AppContainer** | **Low Integrity** | **Low Integrity** |

- Comprehensive exploit mitigations are applied to the Edge content process (MicrosoftEdgeCP.exe) that hosts EdgeHTML

# Exploit Mitigations > Content Process > Known Mitigation Bypass/Weakness

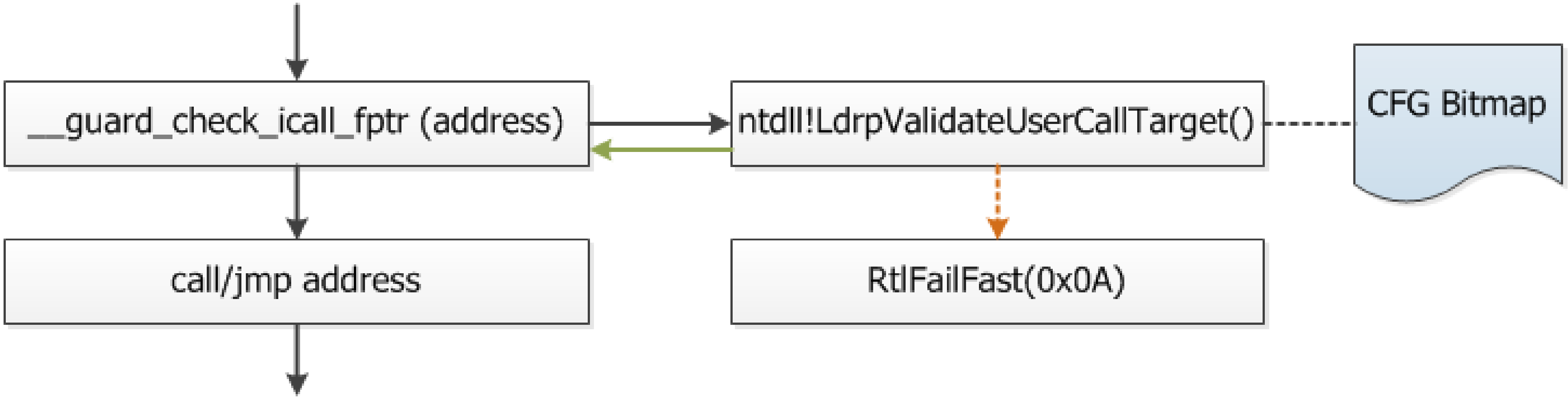
- 64-bit
  - Relative heap spraying [4, 5]
- ASLR+DEP
  - Memory Content Disclosure [3,6]
- AppContainer
  - Kernel vulnerabilities [7,8]
  - Vulnerabilities in the broker or higher-privileged processes [9,10]
  - Leveraging writable resources [9]

# Exploit Mitigations > EdgeHTML & Dependencies > Buffer Security Check (/GS)



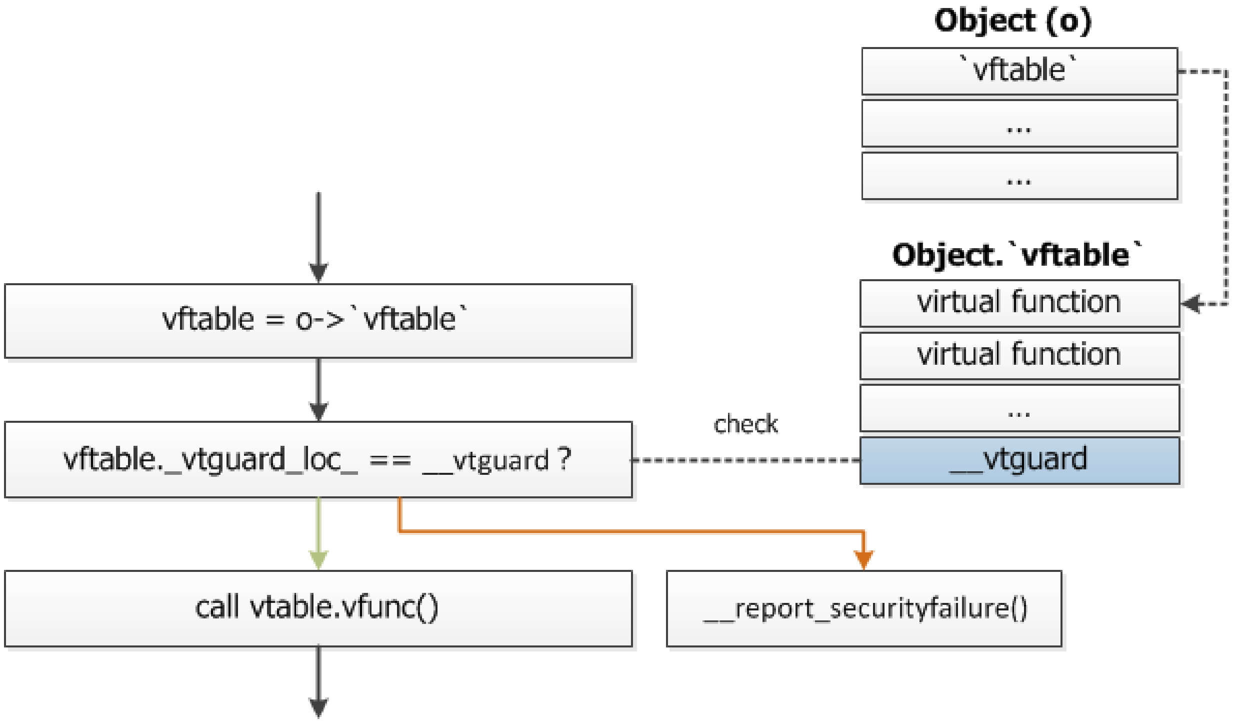
- Purpose: Detect stack buffer overflows
- Known Bypass/Weakness: Controllable stack buffer pointer/index [1, 11]

# Exploit Mitigations > EdgeHTML & Dependencies > Control Flow Guard (CFG)


__guard_check_icall_fptr (address)
ntdll!LdrpValidateUserCallTarget()
CFG Bitmap
call/jmp address
RtlFailFast(0x0A)

- Purpose: Disrupt ROP-based exploits
- Recently introduced and well-researched [12, 13]
- Known Bypass/Weakness:
  – Dynamic Code: Flash JIT-generated code [2]
  – Jumping to valid APIs, stack data overwrite, more…[5]

# Exploit Mitigations > EdgeHTML > Virtual Table Guard (VTGuard)

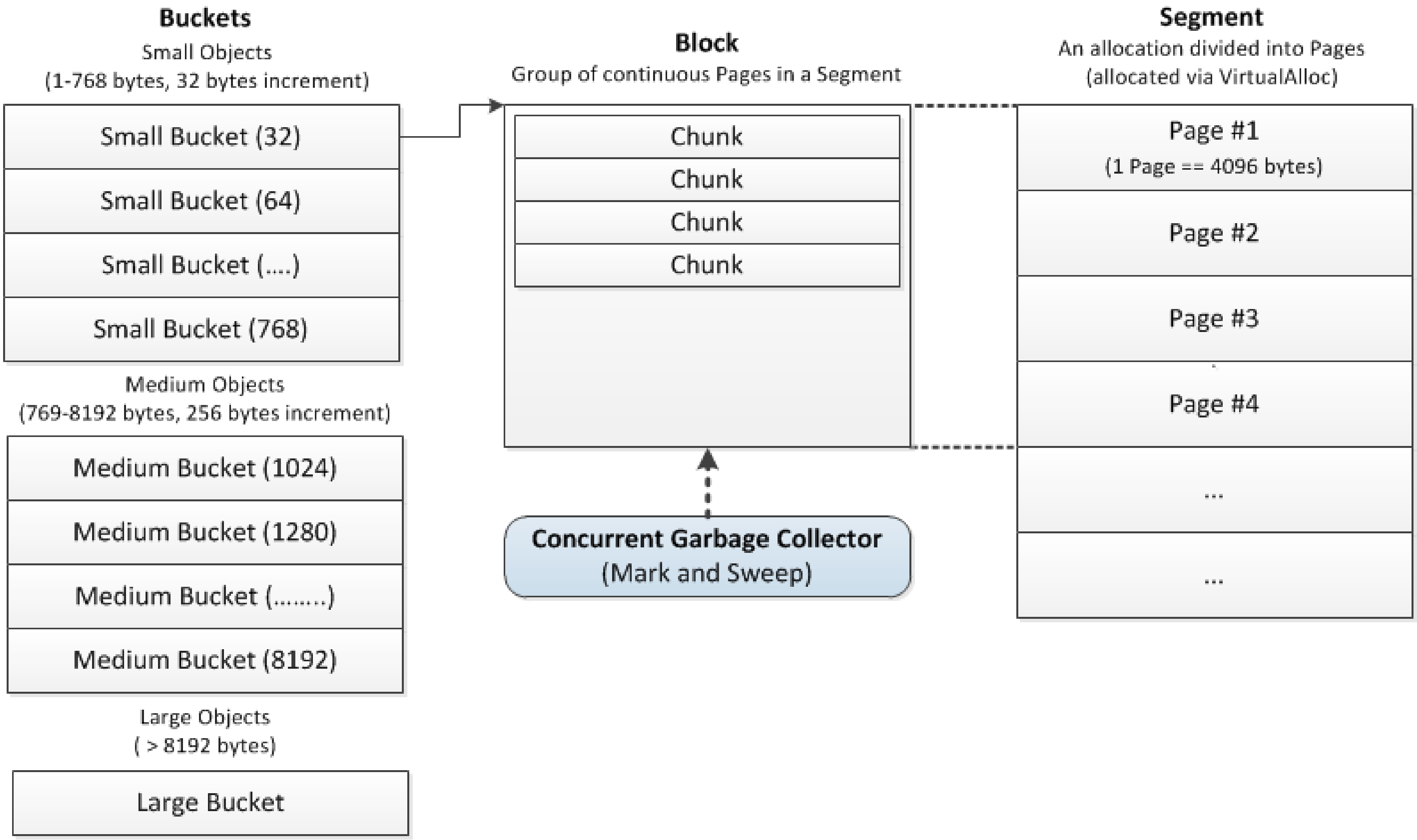


- Purpose: Detect an invalid virtual function table
- Known Bypass/Weakness:
  - –Applied only to select EdgeHTML classes
  - –Bypassed if __vtguard address is leaked

# Exploit Mitigations > EdgeHTML > Memory GC (MemGC)

- Introduced in EdgeHTML and MSHTML on Win10
- Purpose: Mitigate exploitation of use-after-frees
  - Prevent freeing of still-referenced memory chunks
- Improvement and successor to Memory Protector
  - Recursively scans MemGC chunks, registers and the stack for references
- Uses a separate managed heap (MemGC heap) and a concurrent mark and sweep garbage collector

# Exploit Mitigations > EdgeHTML > MemGC Heap (Edge x64)



**Buckets**

Small Objects
(1-768 bytes, 32 bytes increment)

Small Bucket (32)
Small Bucket (64)
Small Bucket (....)
Small Bucket (768)

Medium Objects
(769-8192 bytes, 256 bytes increment)

Medium Bucket (1024)
Medium Bucket (1280)
Medium Bucket (........)
Medium Bucket (8192)

Large Objects
( > 8192 bytes)

Large Bucket

**Block**

Group of continuous Pages in a Segment

Chunk
Chunk
Chunk
Chunk

**Concurrent Garbage Collector**
(Mark and Sweep)

**Segment**

An allocation divided into Pages
(allocated via VirtualAlloc)

Page #1
(1 Page == 4096 bytes)

Page #2

Page #3

Page #4

...

...

# Exploit Mitigations > EdgeHTML > Memory GC (MemGC)

- No known bypass for covered cases as of writing
  - Research on leveraging its predecessor (Memory Protector) to bypass ASLR [14] and approximating bottom-up allocation address range [15] are published
- Interesting potential research topics on MemGC:
  - Internals (algorithms, data structures map, etc.)
  - Grooming the MemGC heap
  - Attacking the MemGC heap metadata
  - Bypassing MemGC

# Exploit Mitigations > Summary

- Comprehensive process-level exploit mitigations are applied: Time-consuming/costly exploit development

**Exploit Mitigations (Process)**
- 64-bit
- ASLR (HiASLR, ForceASLR)
- DEP
- AppContainer

- Additional exploit mitigations applied to EdgeHTML and its dependencies: A number of vulnerabilities will be unexploitable or very difficult to exploit

**+ Exploit Mitigations (EdgeHTML)**
- Buffer Security Check (/GS)
- Control Flow Guard (CFG)
- Virtual Table Guard (VTGuard)
- Memory GC (MemGC)

**+ Exploit Mitigations (Dependencies)**
- Buffer Security Check (/GS)
- Control Flow Guard (CFG)

# Conclusion

# Conclusion

- New attack vectors in rendering engines will be introduced in the parsing of new markup/style specs and in the DOM API to support new web standards
- New attack vectors in EdgeHTML are balanced by comprehensive exploit mitigations in place
- Interesting research topics related to EdgeHTML (internals, audit, fuzzing, bypass):

XmlLite

MSXML6

Windows Imaging Component

MemGC

Media Foundation

DirectWrite

WinRT PDF Renderer

# References (More in the whitepaper)

- [1] J. Mateusz , "**One font vulnerability to rule them all**," [Online]. Available: http://j00ru.vexillium.org/dump/recon2015.pdf

- [2] F. Falcón, "**Exploiting CVE-2015-0311, Part II: Bypassing Control Flow Guard on Windows 8.1 Update 3**," [Online]. Available: https://blog.coresecurity.com/2015/03/25/exploiting-cve-2015-0311-part-ii-bypassing-control-flow-guard-on-windows-8-1-update-3/

- [3] H. Li , "**Smashing the Heap with Vector: Advanced Exploitation Technique in Recent Flash Zero-day Attack**," [Online]. Available: https://sites.google.com/site/zerodayresearch/smashing_the_heap_with_vector_Li.pdf

- [4] I. Fratric, "**Exploiting Internet Explorer 11 64-bit on Windows 8.1 Preview**," [Online]. Available: http://ifsec.blogspot.com/2013/11/exploiting-internet-explorer-11-64-bit.html

- [5] Y. Chen, "**The Birth of a Complete IE11 Exploit Under the New Exploit Mitigations**," [Online]. Available: https://syscan.org/index.php/download/get/aef11ba81927bf9aa02530bab85e303a/SyScan15%20Yuki%20Chen%20-%20The%20Birth%20of%20a%20Complete%20IE11%20Exploit%20Under%20the%20New%20Exploit%20Mitigations.pdf

- [6] F. Serna, "**The info leak era on software exploitation**," [Online]. Available: https://media.blackhat.com/bh-us-12/Briefings/Serna/BH_US_12_Serna_Leak_Era_Slides.pdf

- [7] T. Ormandy and J. Tinnes, "**There's a party at ring0 and you're invited**," [Online]. Available: https://www.cr0.org/paper/to-jt-party-at-ring0.pdf

- [8] Nils and J. Butler, "**MWR Labs Pwn2Own 2013 Write-up - Kernel Exploit**," [Online]. Available: https://labs.mwrinfosecurity.com/blog/2013/09/06/mwr-labs-pwn2own-2013-write-up---kernel-exploit/

# References (More in the whitepaper)

- [9] J. Forshaw, "**Digging for Sandbox Escapes - Finding sandbox breakouts in Internet Explorer**," [Online]. Available: https://www.blackhat.com/docs/us-14/materials/us-14-Forshaw-Digging-For_IE11-Sandbox-Escapes.pdf

- [10] P. Sabanal and M. V. Yason, "**Digging Deep Into The Flash Sandboxes**," [Online]. Available: https://media.blackhat.com/bh-us-12/Briefings/Sabanal/BH_US_12_Sabanal_Digging_Deep_WP.pdf

- [11] C. Evans, "**What is a "good" memory corruption vulnerability?**," [Online]. Available: http://googleprojectzero.blogspot.com/2015/06/what-is-good-memory-corruption.html

- [12] MJ0011, "**Windows 10 Control Flow Guard Internals**," [Online]. Available: http://powerofcommunity.net/poc2014/mj0011.pdf

- [13] J. Tang, "**Exploring Control Flow Guard in Windows 10**," [Online]. Available: http://sjc1-te-ftp.trendmicro.com/assets/wp/exploring-control-flow-guard-in-windows10.pdf

- [14] A.-A. Hariri, S. Zuckerbraun and B. Gorenc, "**Abusing Silent Mitigations: Understanding weaknesses within Internet Explorer's Isolated Heap and MemoryProtection**," [Online]. Available: http://h30499.www3.hp.com/hpeb/attachments/hpeb/off-by-on-software-security-blog/599/1/WP-Hariri-Zuckerbraun-Gorenc-Abusing_Silent_Mitigations.pdf

- [15] I. Fratric, "**Dude, where's my heap?**," [Online]. Available: http://googleprojectzero.blogspot.com/2015/06/dude-wheres-my-heap.html

**Statement of Good Security Practices:** IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

# THANK YOU

www.ibm.com/security

IBM Security

Intelligence. Integration. Expertise.

**© Copyright IBM Corporation 2015. All rights reserved.** The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and / or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.