IBM Security
Intelligence. Integration. Expertise.

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

[v2]

© 2015 IBM Corporation

Agenda

- Overview
- Attack Surface
- Exploit Mitigations
- Conclusion

© 2015 IBM Corporation

Notes

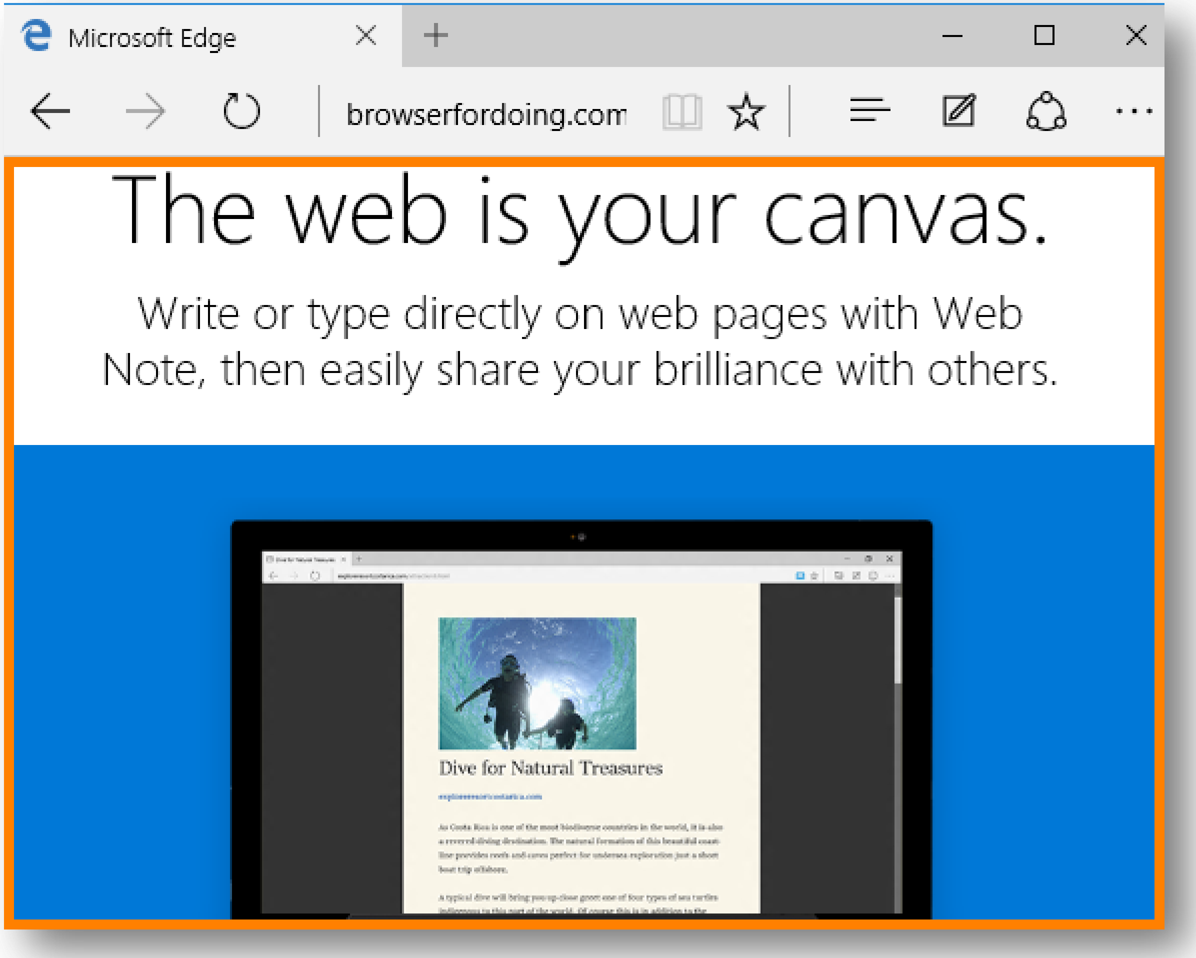
- Detailed whitepaper is available
- All information is based on Microsoft Edge running on 64-bit Windows 10 build 10240 (edgehtml.dll version 11.0.10240.16384)

© 2015 IBM Corporation

Overview

© 2015 IBM Corporation

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.

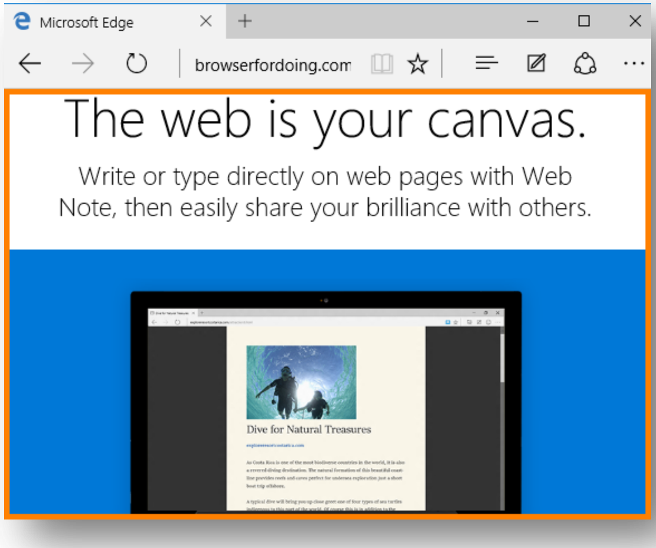
Dive for Natural Treasures

exploreresortcostarica.com

As Costa Rica is one of the most biodiverse countries in the world, it is also a revered diving destination. The natural formation of this beautiful coastline provides reefs and caves perfect for undersea exploration just a short boat trip offshore.

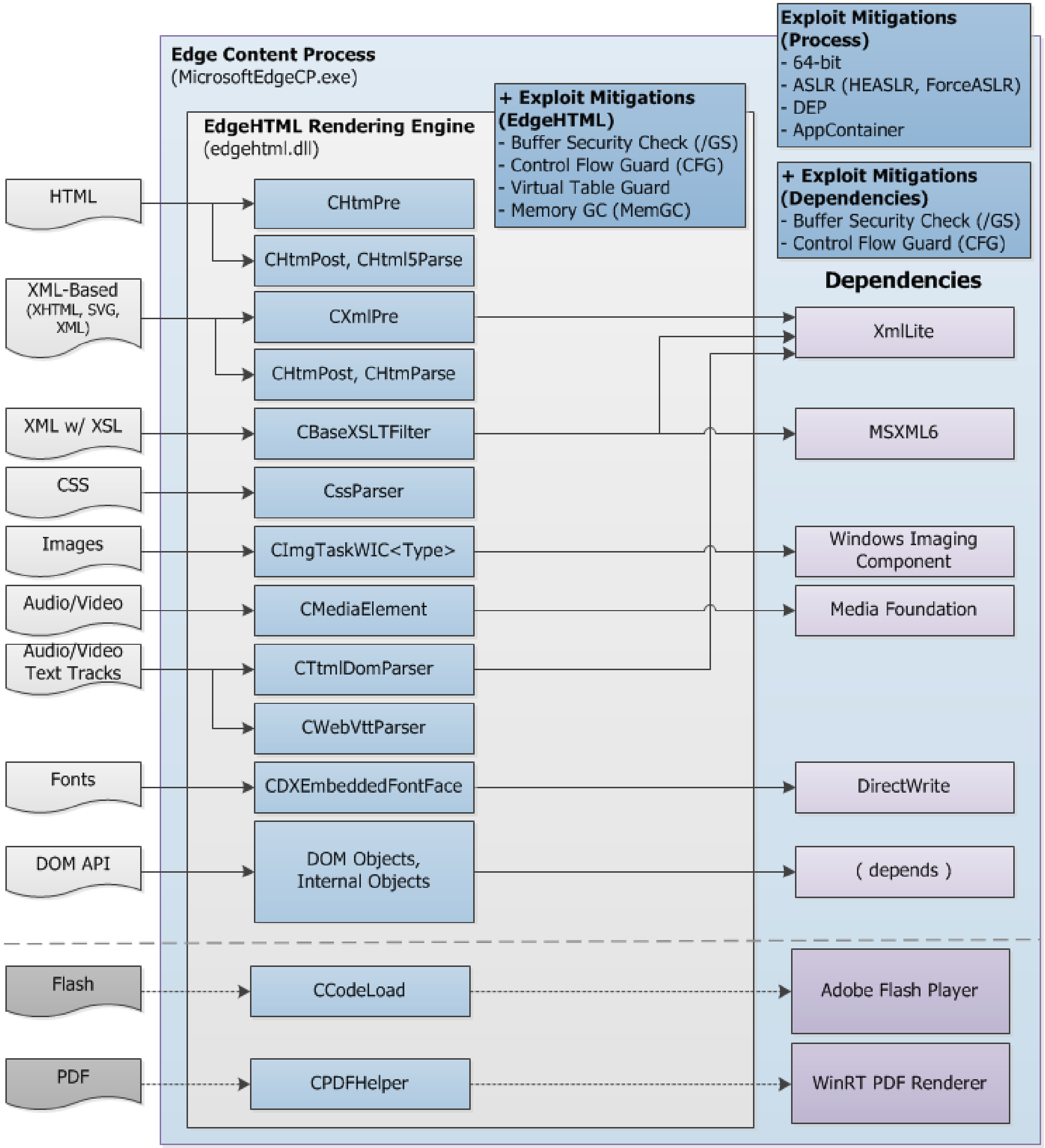
A typical dive will bring you up-close given one of four types of sea turtles indigenous to this part of the world. Of course this is in addition to the

Overview > EdgeHTML Attack Surface Map & Exploit Mitigations



The web is your canvas.

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



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 (HEASLR, ForceASLR)
- DEP
- AppContainer

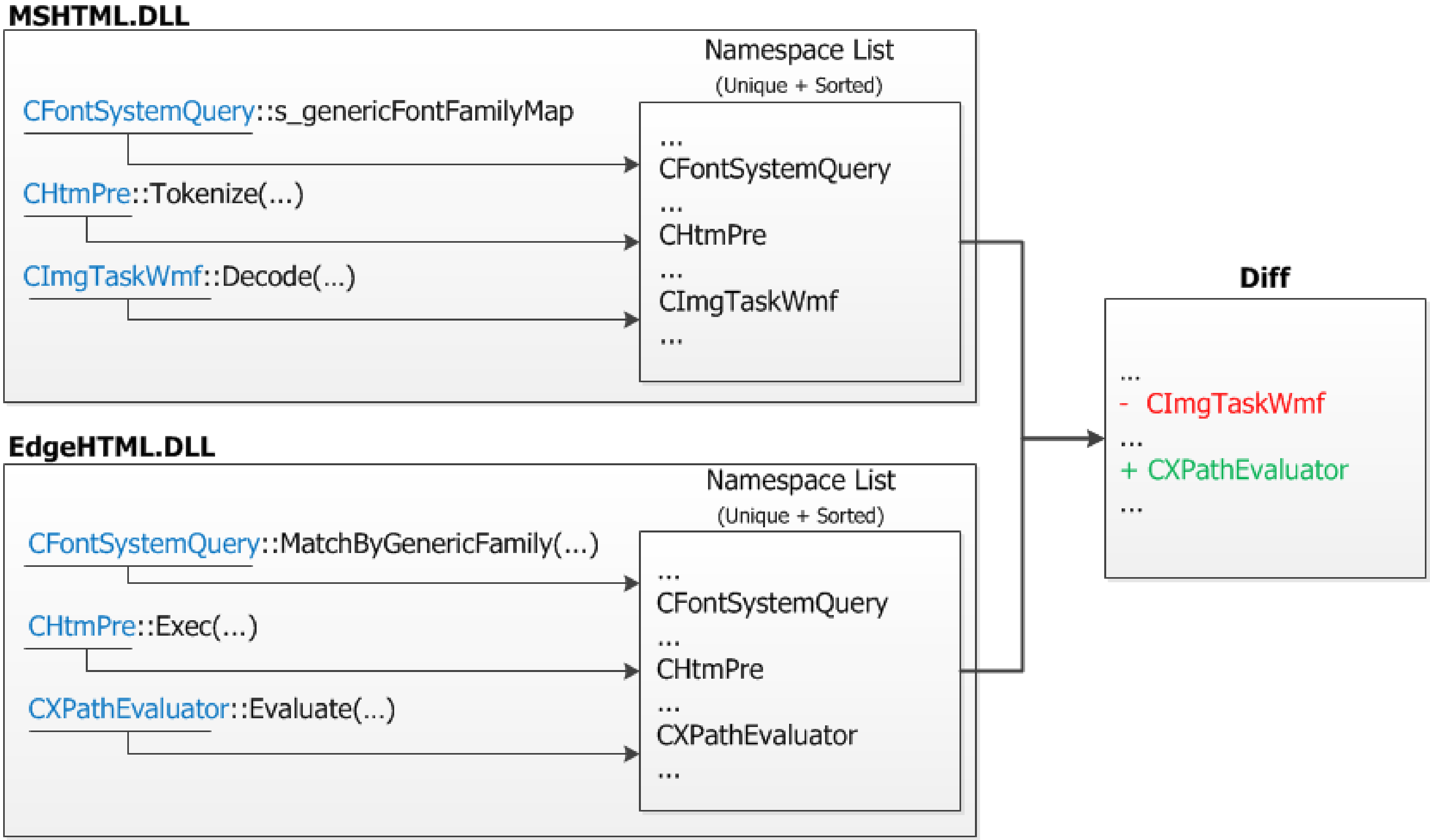
+ Exploit Mitigations (Dependencies)
- Buffer Security Check (/GS)
- Control Flow Guard (CFG)

Dependencies

Overview > Initial Recon: MSHTML and EdgeHTML

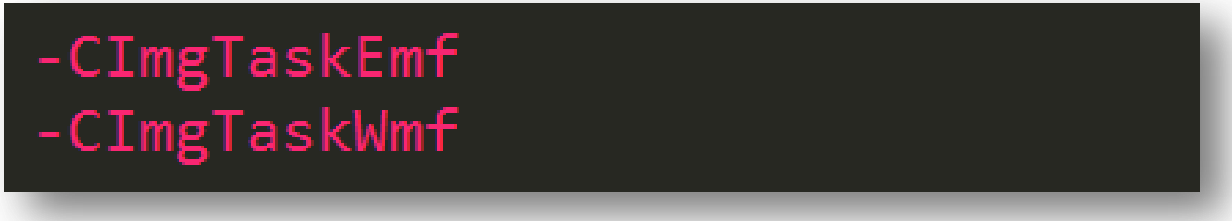
- EdgeHTML is forked from Trident (MSHTML)
- Problem: Quickly identify major code changes (features/functionalities) from MSHTML to EdgeHTML
- One option: Diff class names and namespaces

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



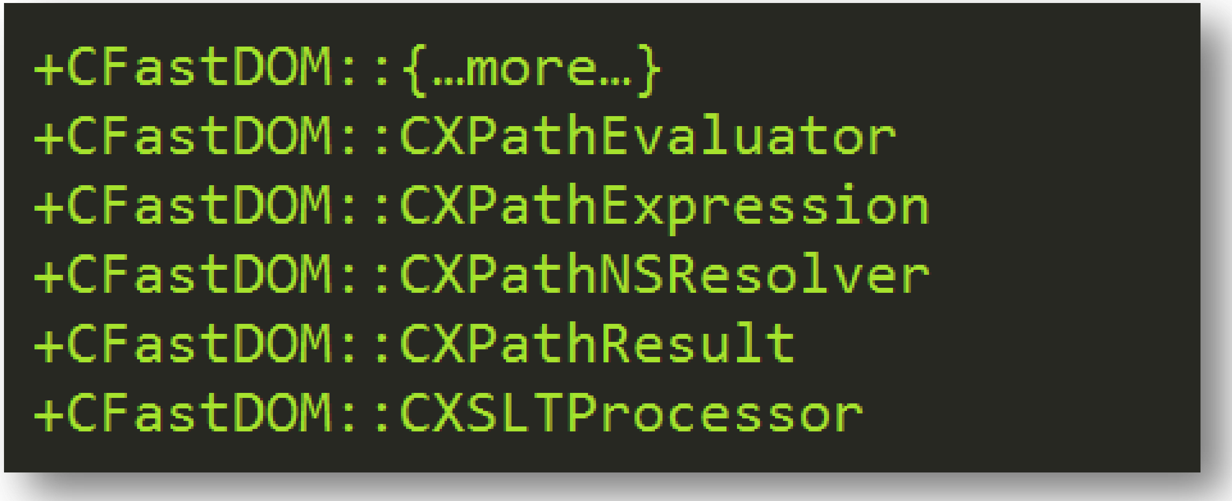
Overview > Initial Recon: Diffing MSHTML and EdgeHTML (Examples)

- Suggests change in image support:



```
-CImgTaskEmf
-CImgTaskWmf
```

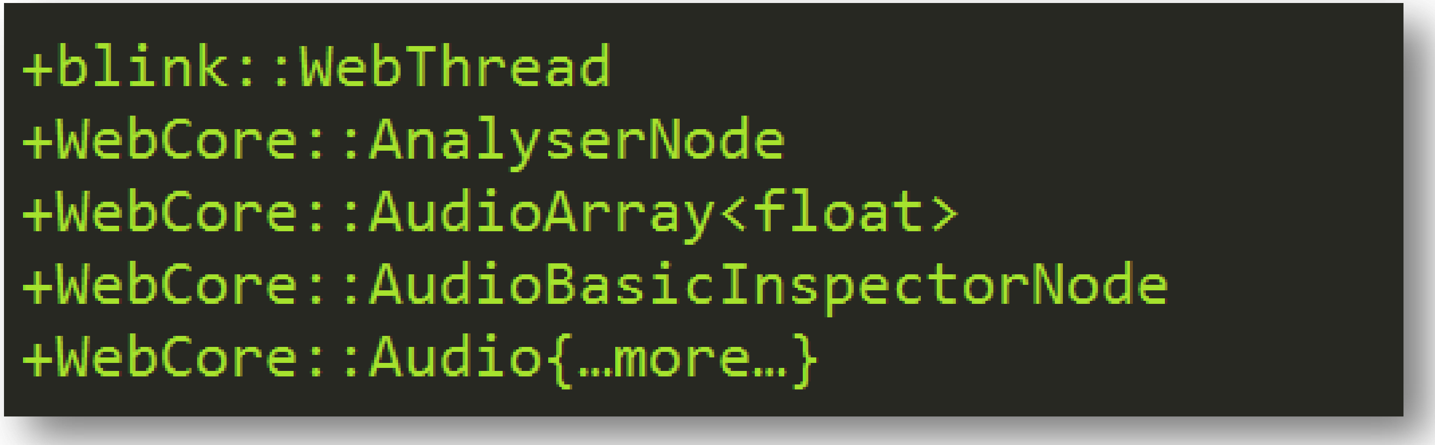
- Suggests new DOM object types:



```
+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…}
```

© 2015 IBM Corporation

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:
  - Function and Method names
  - Strings
  - Imports and Exports

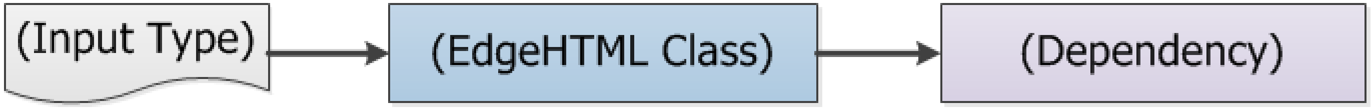
© 2015 IBM Corporation

Attack Surface

© 2015 IBM Corporation

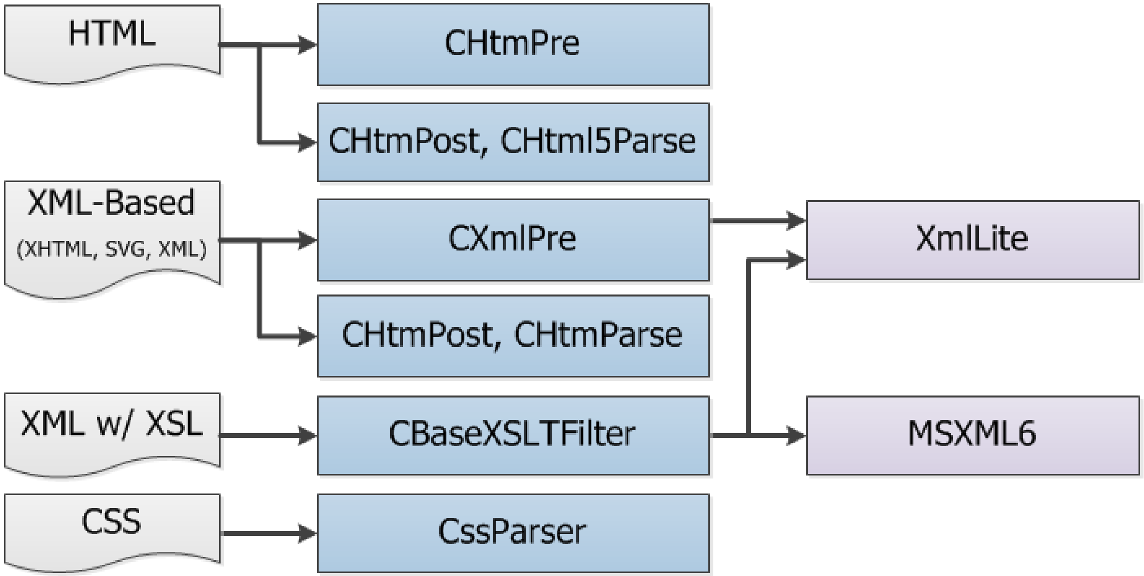
Attack Surface

- 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

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

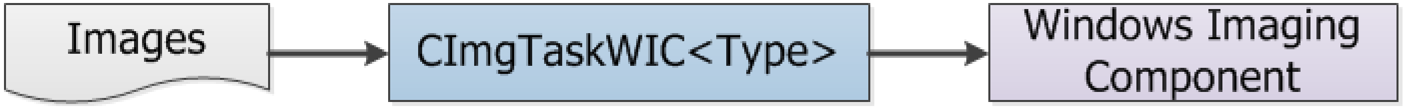
Attack Surface > Markup/Style Parsing > MSXML6

MSXML6

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

© 2015 IBM Corporation

Attack Surface > Image Decoding



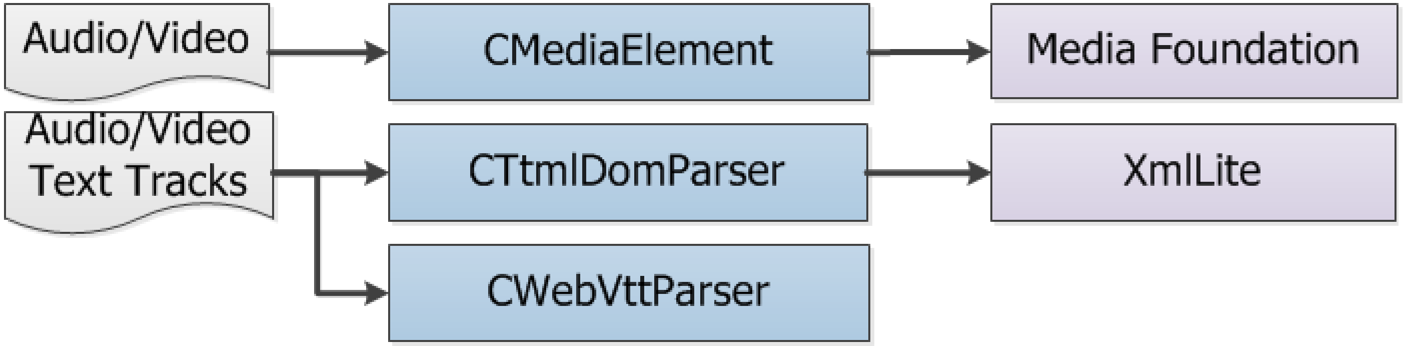
- Reachable via: direct link, <img>, <embed>
- Supported image formats: g_rgMimeInfoImg
- PNG, JPG, GIF, DDS, TIFF, BMP, HDP, ICO decoding via Windows Imaging Component (WIC)
- WMF and EMF support via GDI was removed in EdgeHTML

© 2015 IBM Corporation

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

- 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>
- Supported audio/video containers: g_rgMimeInfoAudio and g_rgMimeInfoVideo
- MP4, MP3, WAV support via Media Foundation (MF)
- TTML & WebVTT support for timed text tracks (captioning) via <track>

© 2015 IBM Corporation

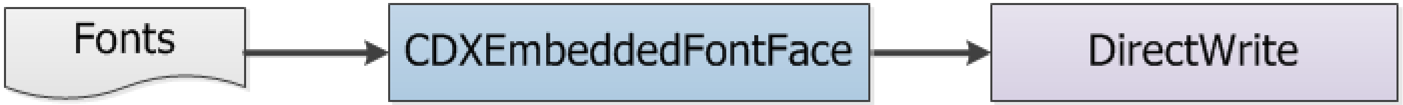
Attack Surface > Audio/Video Decoding > Media Foundation (MF)

Media Foundation

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

© 2015 IBM Corporation

Attack Surface > Font Rendering

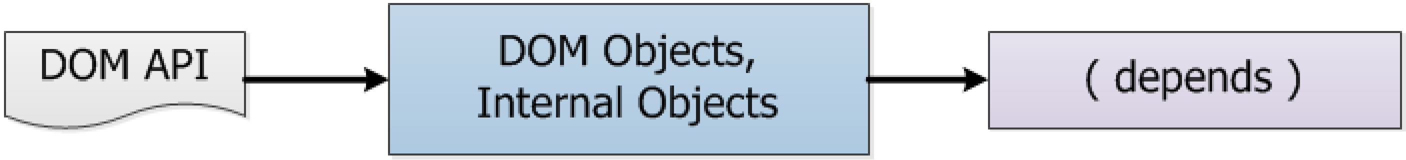


- Reachable via: @font-face CSS rule
- TTF, OTF and WOFF (after TTF/OTF extraction) font support via DirectWrite
- EOT font support was removed in EdgeHTML
  - Removed dependence to T2EMBED and GDI for EOT font parsing

Attack Surface > Font Rendering > DirectWrite

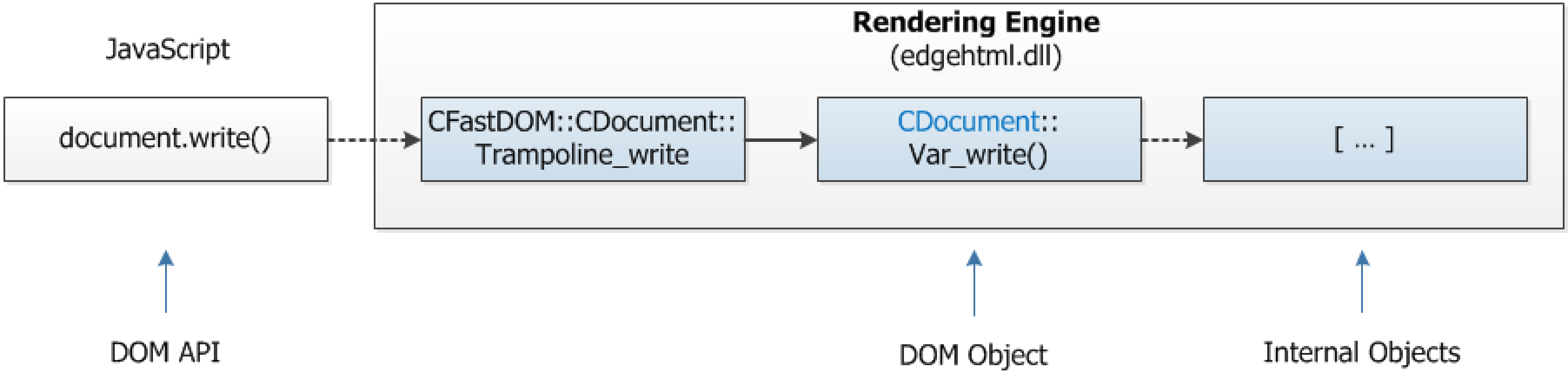
- DirectX Text Rendering API
- Built-in Windows component
- Parses the font in the user-mode process where it (DWrite.dll) is hosted
- IDWriteFactory::CreateCustomFontFileReference() is used by EdgeHTML to register a custom private font
- DirectWrite is discussed in the “One font vulnerability to rule them all” presentation [1]

Attack Surface > DOM API



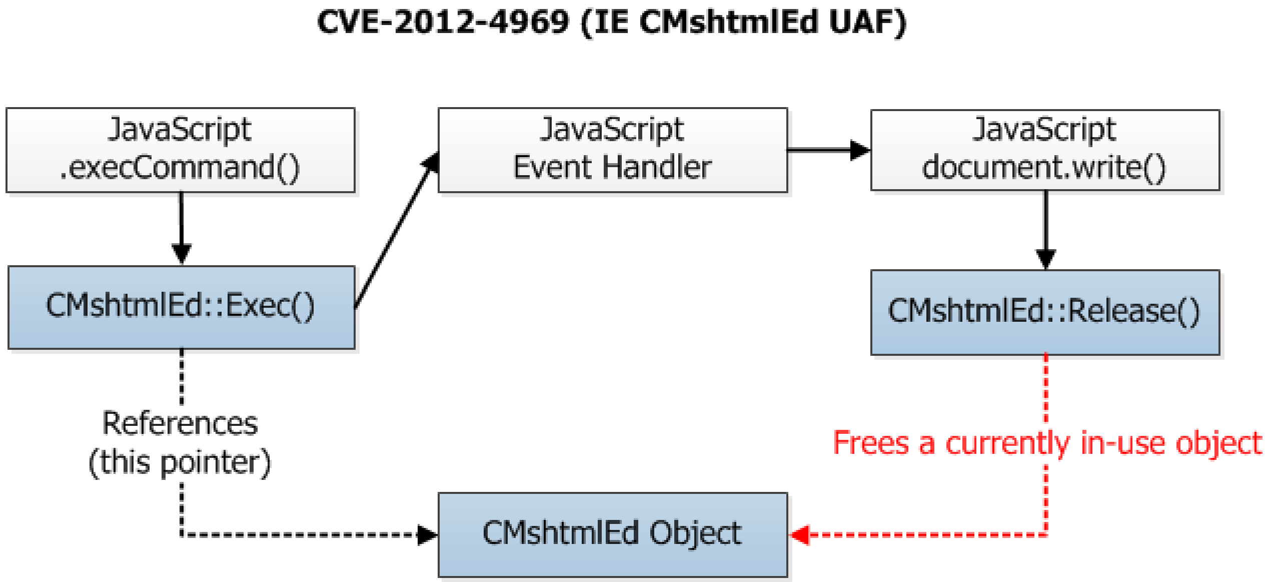
- Reachable via: JavaScript
- Large attack surface that:
  - Interacts directly with EdgeHTML DOM objects
  - Interacts indirectly with internal EdgeHTML objects and libraries (depends)

Attack Surface > DOM API



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

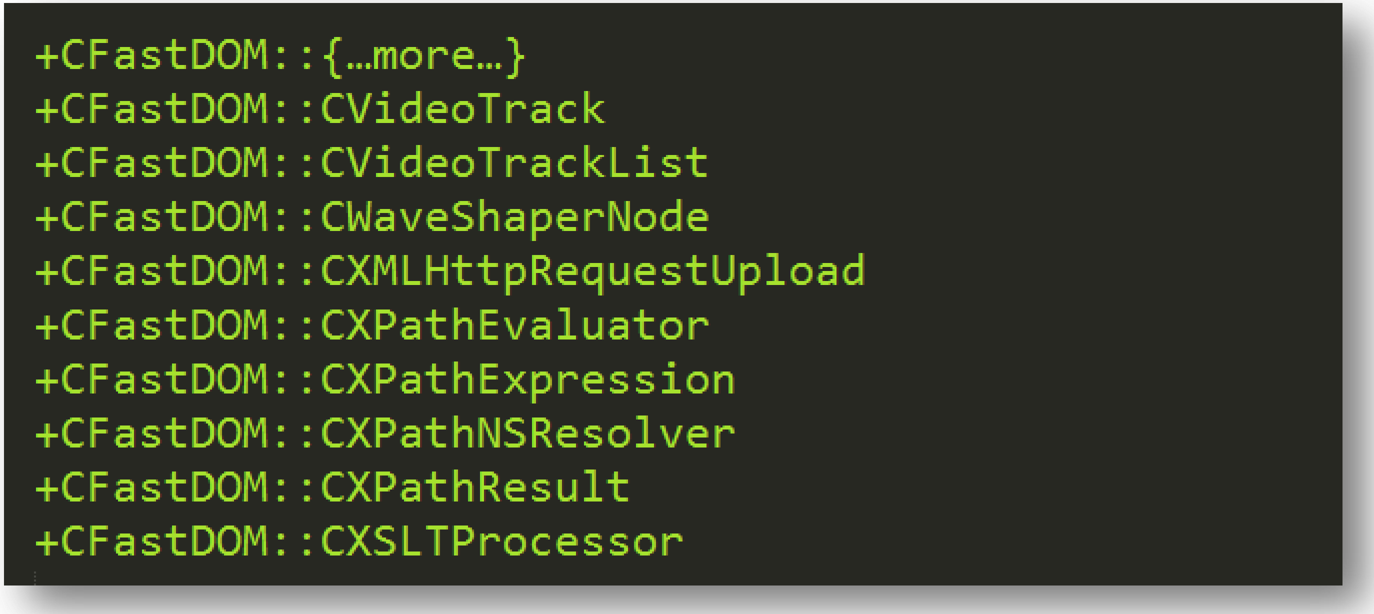
Attack Surface > DOM API


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-frees (above), heap overflows, invalid pointer access, etc.

© 2015 IBM Corporation

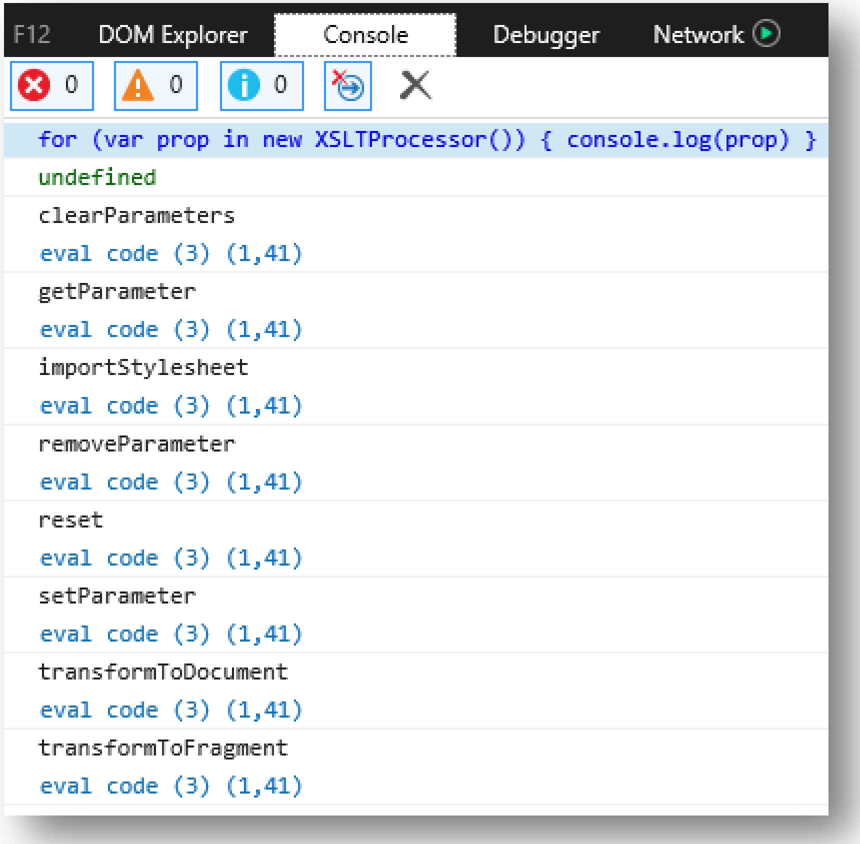
Attack Surface > DOM API > New DOM Object Types



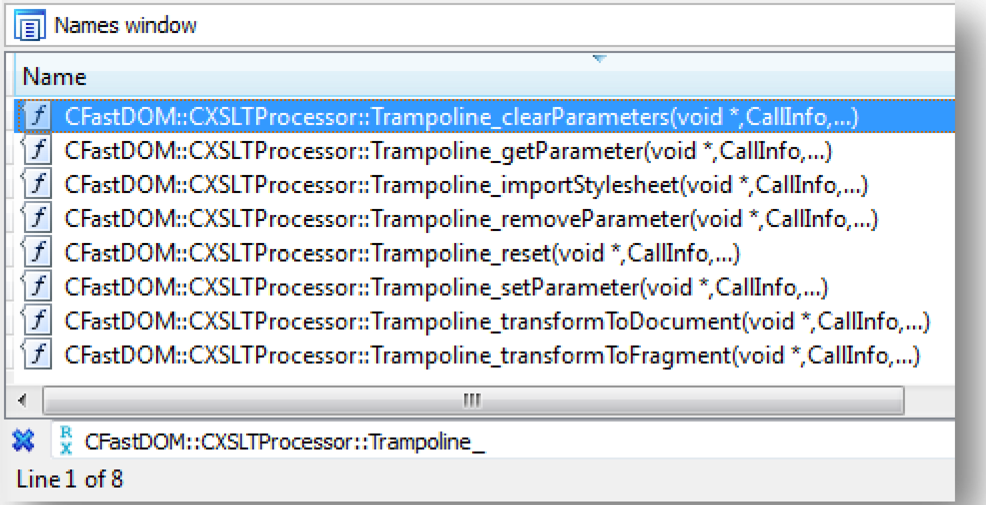
```
+CFastDOM::{…more…}
+CFastDOM::CVideoTrack
+CFastDOM::CVideoTrackList
+CFastDOM::CWaveShaperNode
+CFastDOM::CXMLHttpRequestUpload
+CFastDOM::CXPathEvaluator
+CFastDOM::CXPathExpression
+CFastDOM::CXPathNSResolver
+CFastDOM::CXPathResult
+CFastDOM::CXSLTProcessor
```

- 80 new DOM object types were found in EdgeHTML
  - New code or new code paths that are reachable

Attack Surface > DOM API > DOM Object Properties/Methods Enumeration



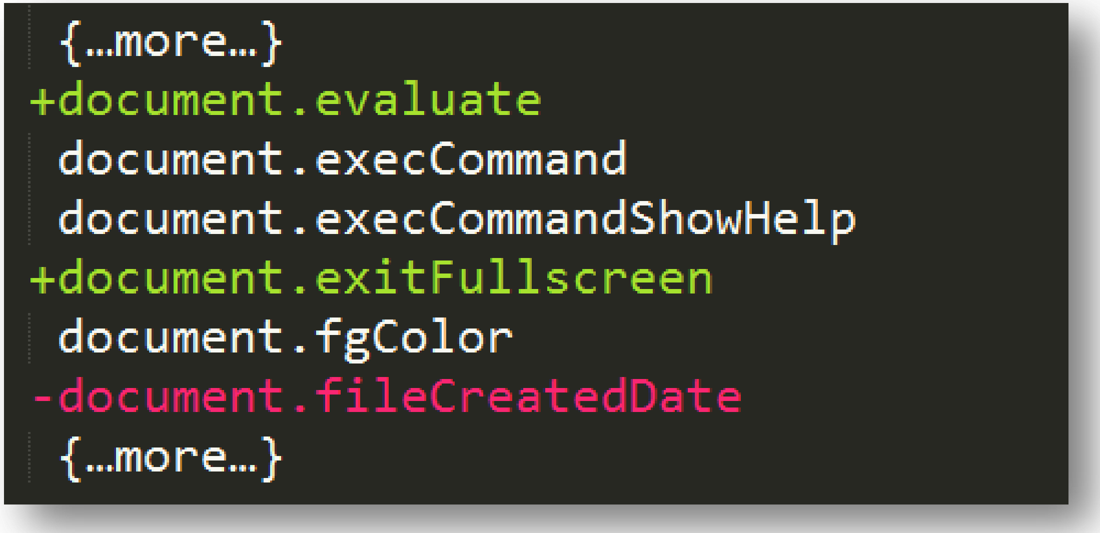
```
for (var prop in new XSLTProcessor()) { console.log(prop) }
undefined
clearParameters
eval code (3) (1,41)
getParameter
eval code (3) (1,41)
importStylesheet
eval code (3) (1,41)
removeParameter
eval code (3) (1,41)
reset
eval code (3) (1,41)
setParameter
eval code (3) (1,41)
transformToDocument
eval code (3) (1,41)
transformToFragment
eval code (3) (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…

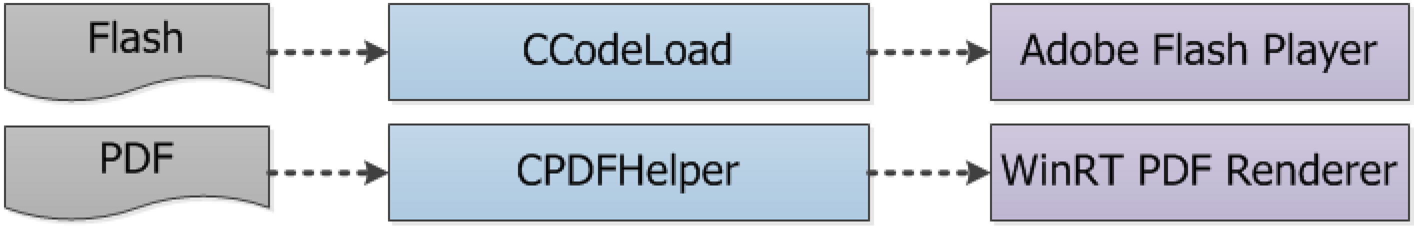
Attack Surface > DOM API > DOM Object Properties/Methods Diffing



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

- … and then diffing them to find out new properties / methods in already-existing DOM object types
  – New code or new code paths that are reachable

Attack Surface > PDF and Flash Renderers

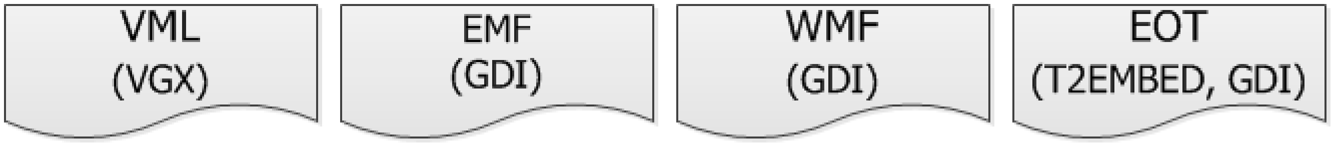


- Built-in/pre-installed complex renderers that can be instantiated by default
  - Additional set of attack surface
  - Functionalities can be repurposed for exploitation
    - CFG Bypass (via Flash JIT -now mitigated) [2]
    - ASLR Bypass (via Flash Vector -now mitigated) [3]

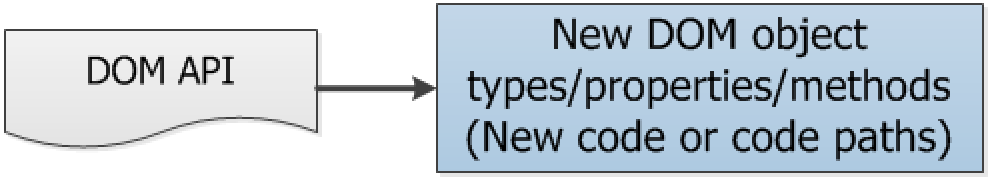
© 2015 IBM Corporation

Attack Surface > Summary

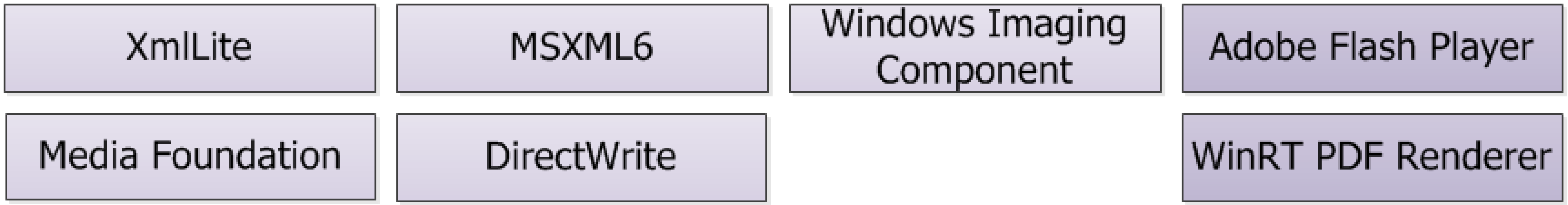
- Well-known attack vectors were removed



- New attack vectors were found in the DOM API



- Remotely-reachable libraries via EdgeHTML



IBM Security

Intelligence. Integration. Expertise.

Exploit Mitigations

© 2015 IBM Corporation

Exploit Mitigations

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

© 2015 IBM Corporation

Exploit Mitigations > Edge Content Process

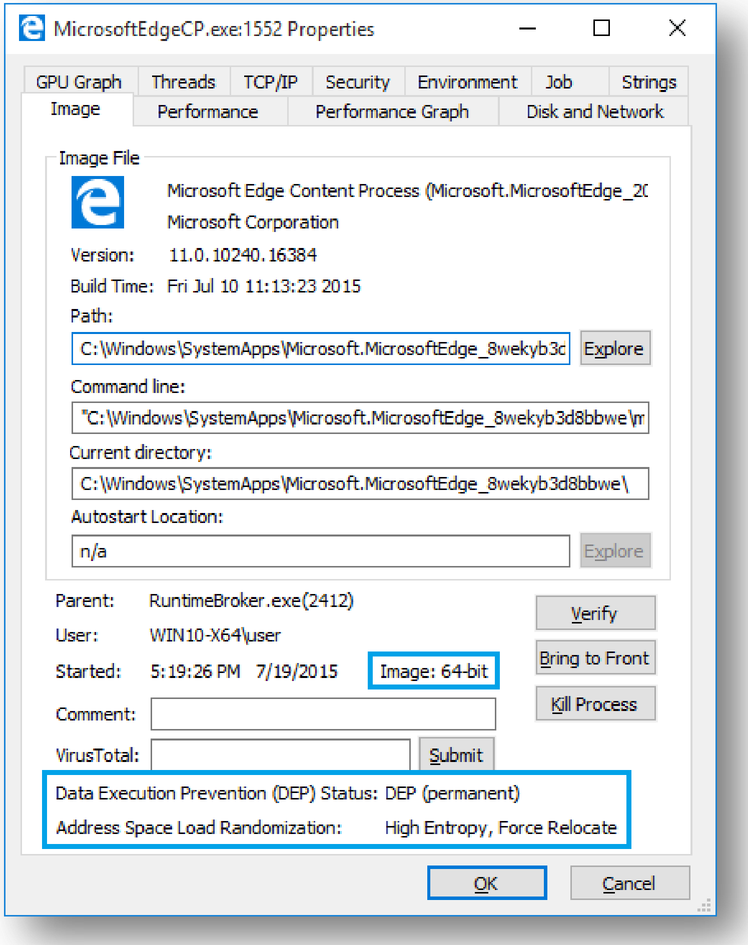


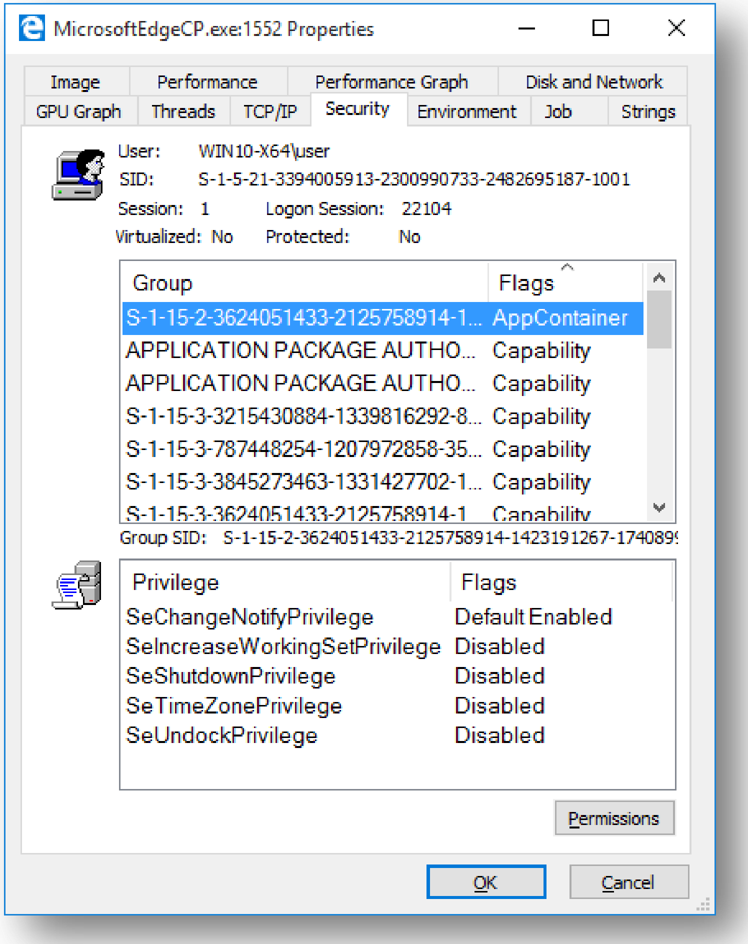
Image File

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



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 (HEASLR, ForceASLR), DEP, and AppContainer

Exploit Mitigations > Content Process > Mitigations Comparison

	Win10/ Edge	Win10/ IE11/	Win8/ ImmersiveIE	Win8/ IE11	Win7/ IE11
64-bit	Yes	No	Yes	No	No
ASLR	Yes (HEASLR, ForceASLR)	Yes (ForceASLR)	Yes (HEASLR, ForceASLR)	Yes (ForceASLR)	Yes (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 (edgehtml.dll)

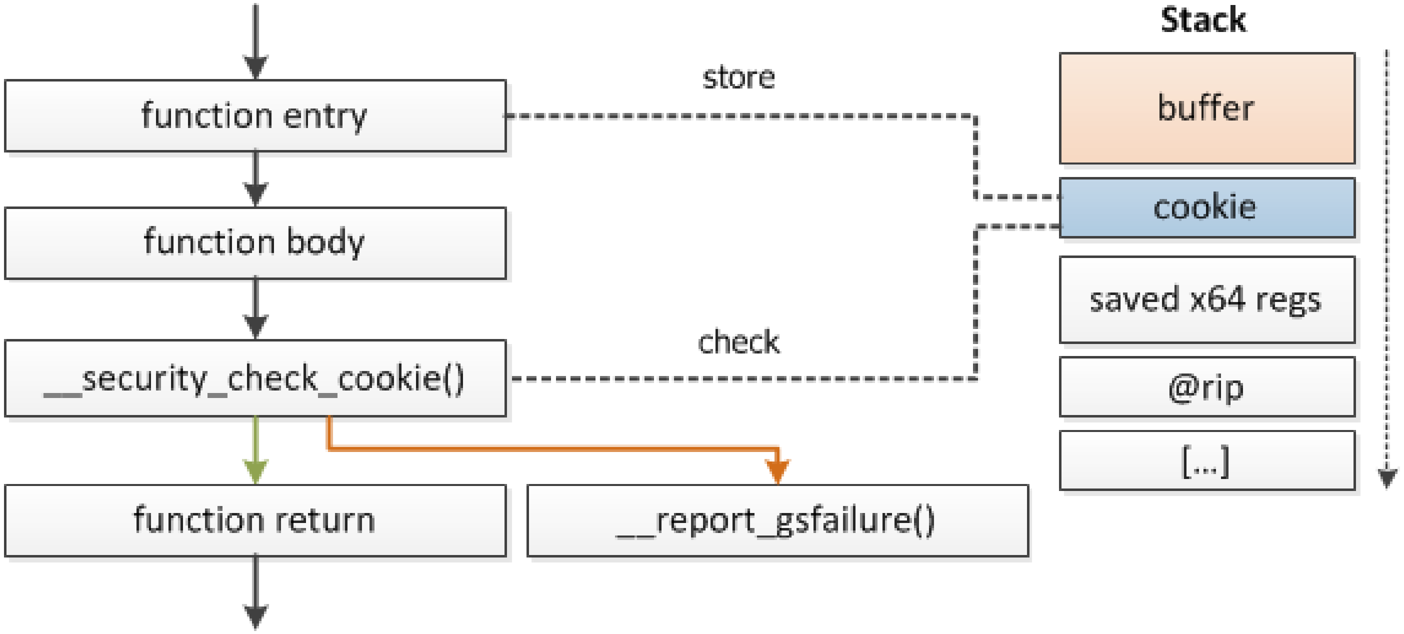
© 2015 IBM Corporation

Exploit Mitigations > Content Process > Known Mitigation Bypass/Weakness

- 64-bit
  - Relative heap spraying (depends) [4,5]
- ASLR+DEP
  - Memory content disclosure (via vulnerabilities) [3,6]
- AppContainer
  - Kernel vulnerabilities [7,8]
  - Vulnerabilities in the broker or higher-privileged processes [9,10,11]
  - Leveraging writable resources [9]

© 2015 IBM Corporation

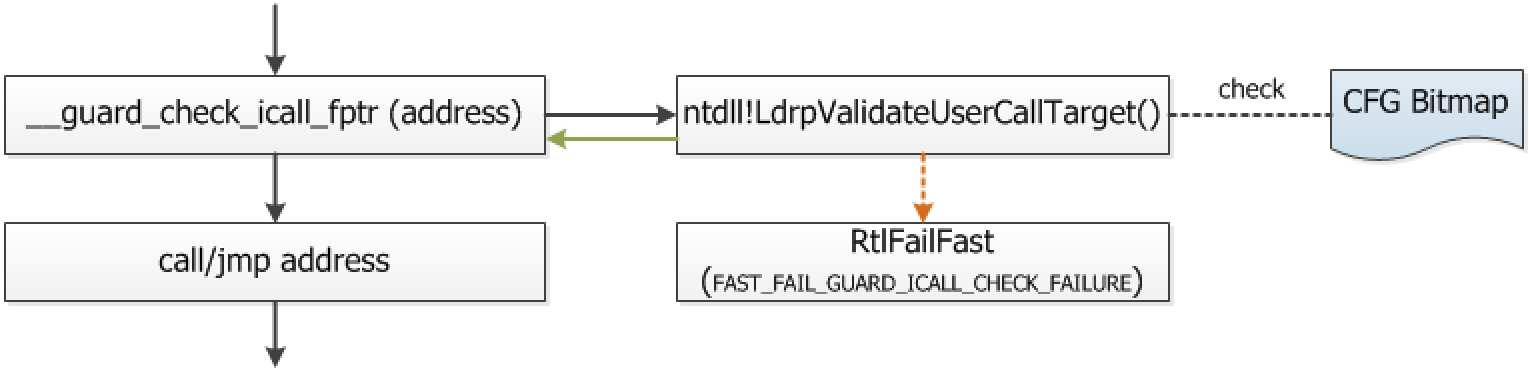
Exploit Mitigations > EdgeHTML & Dependencies > Buffer Security Check (/GS)



- Purpose: Detect stack buffer overflows
- Known Bypass/Weakness: Controllable stack buffer pointer/index [1,12]

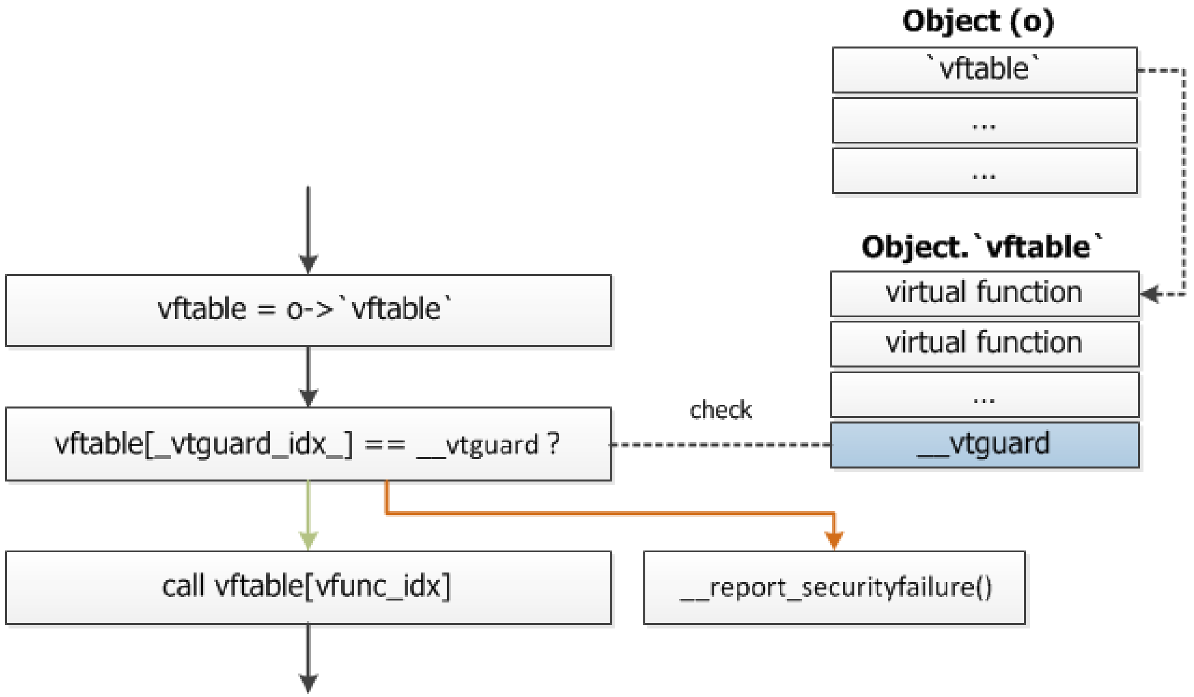
© 2015 IBM Corporation

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


__guard_check_icall_fptr (address)
ntdll!LdrpValidateUserCallTarget()
check
CFG Bitmap
call/jmp address
RtlFailFast
(FAST_FAIL_GUARD_ICALL_CHECK_FAILURE)

- Purpose: Detect and prevent abnormal control flow
- Recently introduced and well-researched [13,14]
- Known Bypass/Weakness:
  - Flash JIT-generated code [2] (now mitigated by JIT-generating a CFG check when generating CALLs)
  - Jumping to a valid API address [5], stack data overwrite [13,5], more [5]…

Exploit Mitigations > EdgeHTML > Virtual Table Guard (VTGuard)



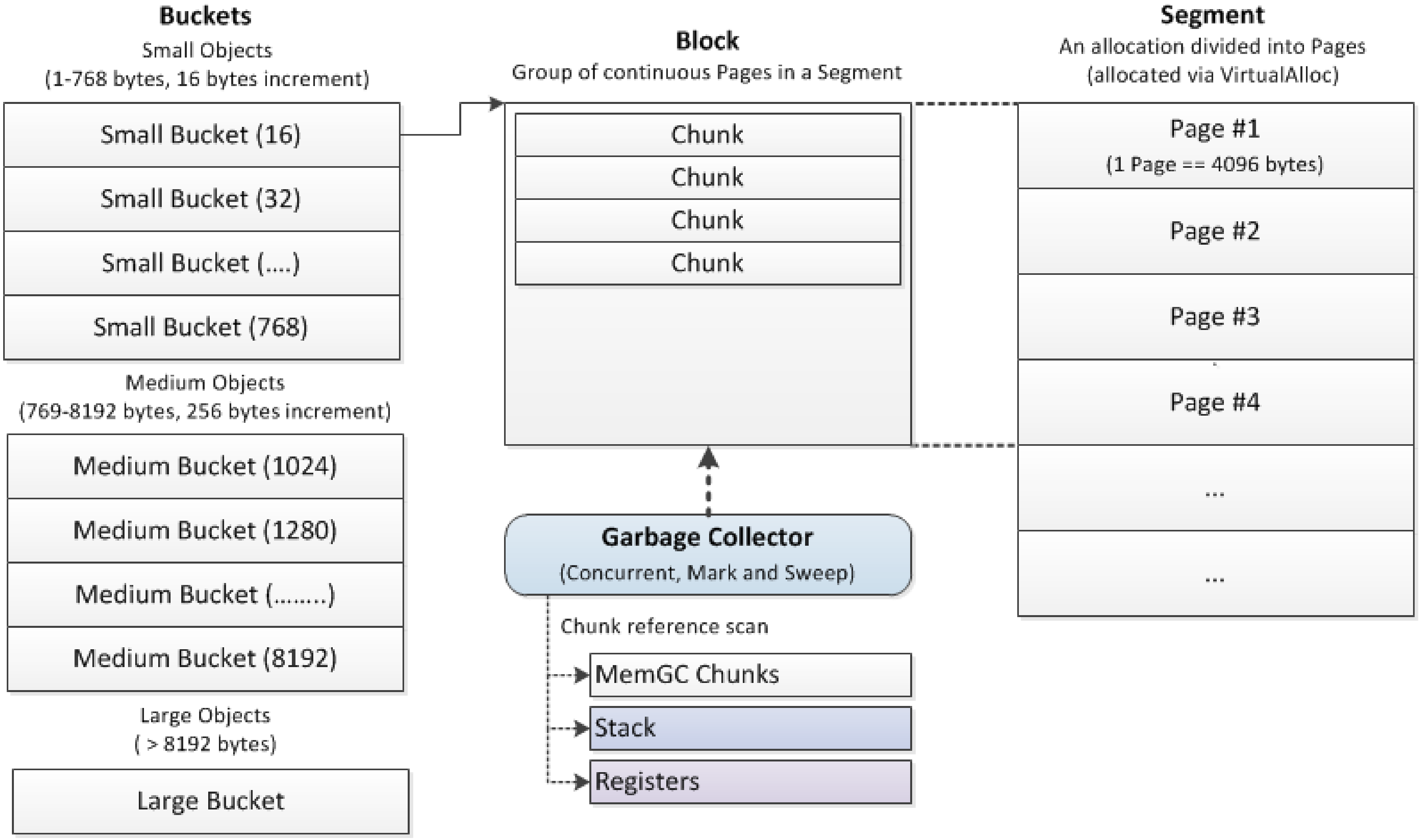
- Purpose: Detect an invalid virtual function table
- Known Bypass/Weakness:
  - Applied only to select EdgeHTML classes
  - Bypassed if address of __vtguard is leaked

Exploit Mitigations > EdgeHTML > Memory GC (MemGC)

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

© 2015 IBM Corporation

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



Exploit Mitigations > EdgeHTML > MemGC > MemGC and Memory Protector Configuration

- Can be configured in Edge and IE via:
  - HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Main OverrideMemoryProtectionSetting=%Value%

Value (DWORD)	Meaning
3	MemGC is enabled (default)
2	Memory Protector is enabled (Force mark-and-reclaim)
1	Memory Protector is enabled
0	MemGC and Memory Protector are disabled

© 2015 IBM Corporation

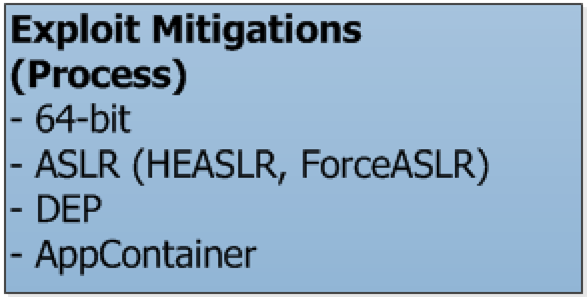
Exploit Mitigations > EdgeHTML > MemGC > Bypass and Related Research

- No known bypass for covered cases as of writing (both MemGC and Memory Protector)
  - Exploits were demonstrated for UAF cases not covered by Memory Protector [15]
  - Memory Protector was leveraged to bypass ASLR on 32-bit IE [15] and approximating the bottom-up allocation address range on 64-bit IE [16]

© 2015 IBM Corporation

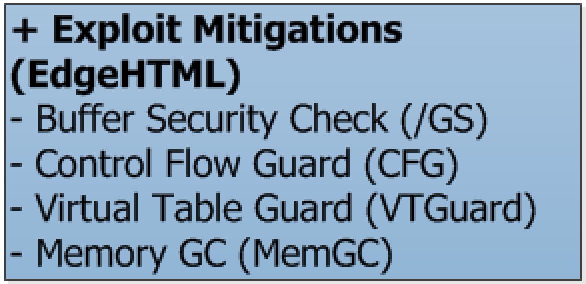
Exploit Mitigations > Summary

- Comprehensive exploit mitigations are applied to the content process: Time-consuming/costly exploit development

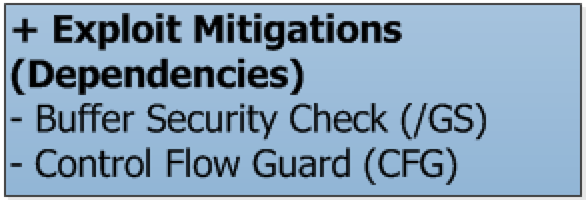


Exploit Mitigations (Process)
- 64-bit
- ASLR (HEASLR, ForceASLR)
- DEP
- AppContainer

- 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)



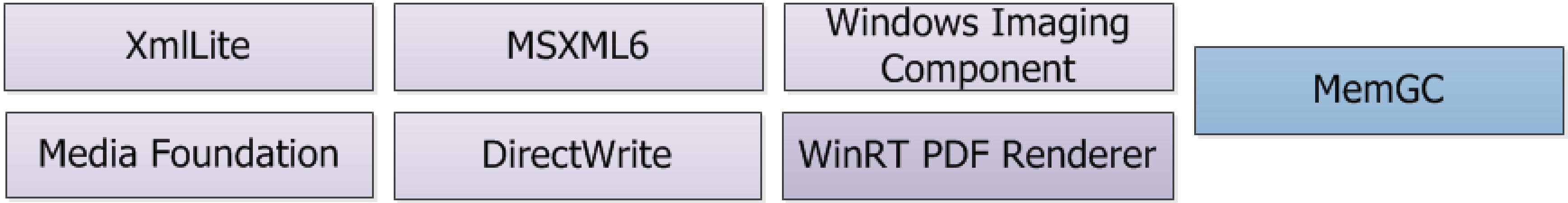
+ Exploit Mitigations (Dependencies)
- Buffer Security Check (/GS)
- Control Flow Guard (CFG)

Conclusion

© 2015 IBM Corporation

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 are in the whitepaper)

- [1] M. Jurczyk, "**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 are 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] M. V. Yason, "**Diving Into IE10's Enhanced Protected Mode Sandbox**," [Online]. Available: https://www.blackhat.com/docs/asia-14/materials/Yason/WP-Asia-14-Yason-Diving-Into-IE10s-Enhanced-Protected-Mode-Sandbox.pdf

- [11] 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

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

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

- [14] 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

- [15] 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

- [16] 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.