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
Agenda

- Overview
- Attack Surface
- Exploit Mitigations
- Conclusion
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)
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

The web is your canvas.

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

- **MSHTML.DLL**
  - `CFontSystemQuery::s_genericFontFamilyMap`
  - `CHtmPre::Tokenize(...)`
  - `CImgTaskWmf::Decode(...)`

- **EdgeHTML.DLL**
  - `CFontSystemQuery::MatchByGenericFamily(...)`
  - `CHtmPre::Exec(...)`
  - `CXPathEvaluator::Evaluate(...)`

**Diff**

- `- CImgTaskWmf`
- `+ CXPathEvaluator`
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:

```plaintext
+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:
  - Function and Method names
  - Strings
  - Imports and Exports
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!CXmIPre::*
- 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
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_rgMimeInfoIImg`
- PNG, JPG, GIF, DDS, TIFF, BMP, HDP, ICO decoding via Windows Imaging Component (WIC)
- WMF and EMF support via GDI was removed in EdgeHTML
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>`
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 (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
DirectWrite is discussed in the “One font vulnerability to rule them all” presentation [1]
Reachable via: JavaScript

Large attack surface that:

- Interacts directly with EdgeHTML DOM objects
- Interacts indirectly with internal EdgeHTML objects and libraries (depends)
DOM API calls can change the state of the DOM tree, DOM objects and other internal EdgeHTML objects.
Unexpected input, unexpected state changes or incorrect state when a DOM API is called can result in memory corruption such as: use-after-frees (above), heap overflows, invalid pointer access, etc.
80 new DOM object types were found in EdgeHTML—New code or new code paths that are reachable.
Enumerating DOM object properties/methods via JavaScript and IDA...
Attack Surface > DOM API > DOM Object Properties/Methods Differing

- ... and then differ 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]
Attack Surface > Summary

- Well-known attack vectors were removed
  - VML (VGX)
  - EMF (GDI)
  - WMF (GDI)
  - EOT (T2EMBED, GDI)

- 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 via EdgeHTML
  - XmlLite
  - MSXML6
  - Windows Imaging Component
  - Adobe Flash Player
  - Media Foundation
  - DirectWrite
  - WinRT PDF Renderer
Exploit Mitigations
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]
Exploit Mitigations > Edge Content Process

- MicrosoftEdgeCP.exe: 64-bit, ASLR (HEASLR, ForceASLR), DEP, and AppContainer
### Exploit Mitigations > Content Process > Mitigations Comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ASLR</td>
<td>Yes (HEASLR, ForceASLR)</td>
<td>Yes (ForceASLR)</td>
<td>Yes (HEASLR, ForceASLR)</td>
<td>Yes (ForceASLR)</td>
<td>Yes (ForceASLR)</td>
</tr>
<tr>
<td>DEP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Process Isolation</td>
<td>AppContainer</td>
<td>Low Integrity</td>
<td>AppContainer</td>
<td>Low Integrity</td>
<td>Low Integrity</td>
</tr>
</tbody>
</table>

- Comprehensive exploit mitigations are applied to the Edge content process (MicrosoftEdgeCP.exe) that hosts EdgeHTML (edgehtml.dll)
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]
Exploit Mitigations > EdgeHTML & Dependencies > Buffer Security Check (/GS)

- Purpose: Detect stack buffer overflows
- Known Bypass/Weakness: Controllable stack buffer pointer/index [1,12]
Exploit Mitigations > EdgeHTML & Dependencies > Control Flow Guard (CFG)

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

**Buckets**
- Small Objects
  - Small Bucket (16)
  - Small Bucket (32)
  - Small Bucket (....)
  - Small Bucket (768)

**Medium Objects**
- Medium Bucket (1024)
- Medium Bucket (1280)
- Medium Bucket (........)
- Medium Bucket (8192)

**Large Objects**
- Large Bucket

**Segment**
- An allocation divided into Pages (allocated via VirtualAlloc)
  - Page #1
  - Page #2
  - Page #3
  - Page #4
  - ...
  - ...

**Block**
- Group of continuous Pages in a Segment
  - Chunk
  - Chunk
  - Chunk

**Garbage Collector**
- (Concurrent, Mark and Sweep)
  - Chunk reference scan
  - MemGC Chunks
  - Stack
  - Registers
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%

<table>
<thead>
<tr>
<th>Value (DWORD)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>MemGC is enabled (default)</td>
</tr>
<tr>
<td>2</td>
<td>Memory Protector is enabled (Force mark-and-reclaim)</td>
</tr>
<tr>
<td>1</td>
<td>Memory Protector is enabled</td>
</tr>
<tr>
<td>0</td>
<td>MemGC and Memory Protector are disabled</td>
</tr>
</tbody>
</table>
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]
Exploit Mitigations > Summary

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

- Additional exploit mitigations applied to EdgeHTML and its dependencies: A number of vulnerabilities will be unexploitable or very difficult to exploit
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
  - Media Foundation
  - DirectWrite
  - WinRT PDF Renderer
  - MemGC
References (More are in the whitepaper)


References (More are in the whitepaper)


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

© 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.