Data Driven Software Security

Sweety Chauhan, Microsoft Security Response Center (MSRC)

August 25th 2016

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary
## Security Fix (F³) process: Find, Fix, Finish

<table>
<thead>
<tr>
<th>FIND</th>
<th>FIX</th>
<th>FINISH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FIND</strong></td>
<td><strong>FIX</strong></td>
<td><strong>FINISH</strong></td>
</tr>
<tr>
<td>Find security vulnerabilities in Microsoft products, services and devices, through multiple sources such as network of ‘Security Researchers’ across the world, internal fuzzing or variant investigations</td>
<td>Coordinate and drive teams across the company to fix the vulnerability across all of products, services, and devices</td>
<td>Document and release the fix to customers so that they can get protected fast</td>
</tr>
<tr>
<td><strong><a href="mailto:Secure@microsoft.com">Secure@microsoft.com</a></strong> – 24/7 response for receiving security vulnerability reports</td>
<td>Comprehensively test and fix all variants – and on all supported platforms – not just the vulnerability that was reported</td>
<td>Ship fixes to online services as soon as it’s ready in order to protect customers as soon as possible</td>
</tr>
<tr>
<td><strong>Coordinated Vulnerability Disclosure (CVD)</strong> – responsible approach to the reporting of vulnerabilities in order to help protect customers</td>
<td></td>
<td>Simultaneously release the fix across all software products, to limit the opportunity for attackers to reverse-engineer the fix</td>
</tr>
</tbody>
</table>
Mitigations

- Analysis of Vulnerability reports and trends
- Develop mitigations to make it harder to exploit
  - Eliminating Classes of Vulnerabilities
  - Breaking Exploitation Techniques
High level Vulnerability and Exploit Trends

# of RCE/EOP CVEs by patch year

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total (of CVEs)</td>
<td>121</td>
<td>111</td>
<td>133</td>
<td>155</td>
<td>218</td>
<td>199</td>
<td>141</td>
<td>287</td>
<td>300</td>
<td>414</td>
</tr>
<tr>
<td>Linear (Total)</td>
<td>121</td>
<td>111</td>
<td>133</td>
<td>155</td>
<td>218</td>
<td>199</td>
<td>141</td>
<td>287</td>
<td>300</td>
<td>414</td>
</tr>
</tbody>
</table>

% of RCE & EOP CVEs exploited within 30 days of patch

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploited within 30 days of patch</td>
<td>24</td>
<td>18</td>
<td>19</td>
<td>25</td>
<td>61</td>
<td>43</td>
<td>25</td>
<td>21</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Not known to be exploited</td>
<td>97</td>
<td>93</td>
<td>114</td>
<td>130</td>
<td>157</td>
<td>156</td>
<td>116</td>
<td>266</td>
<td>262</td>
<td>396</td>
</tr>
</tbody>
</table>
## Windows 10 Mitigations

### Platform
- **Control Flow Guard (CFG)**
  Enabled by default for user mode
- **Hypervisor Enforced Code Integrity (HVCI)**
  Enforcing strong code integrity for kernel mode
- **User Mode Font Driver (UMFD)**
  Blocking of untrusted fonts to help eliminate attack vectors via font parsing

### Edge
- **AppContainer**
- **64-bit by default**
- **MemGC**
Control Flow Guard

- Built-in protection against memory corruption
- Lightweight security checks at compile-time
- Enabling CFG help make Windows secure platform to build and run applications on
Secure Boot
Ensures that everything that boots on a platform is signed by a trusted authority
Includes Secure Firmware Updates and “Platform” Secure Boot

Kernel Mode Code Integrity (KMCI)
Feature in Windows that ensures that any code running in kernel is signed by a trusted authority
Virtual Secure Mode (VSM) Architecture

Hyper Guard
- A component of secure kernel introduced in Windows 10 1511
- Always active when secure kernel is enabled
- Uses hypervisor capabilities to block modifications where possible:
  - Machine State Registers (MSR)
  - Descriptor Table Registers (DTR)
  - Control registers (CR)

Hypervisor-Enforced Code Integrity (HVCI)

Prior to Windows 10, CI enforcement was done from within the Kernel.

HVCI protects the management of the code integrity checks using virtualization based security.

For HVCI based systems, enforcement will be in VSM:
- Pages can only be marked executable from VSM after verification in VSM.
- Eliminates most memory based attacks.
UMFD (User Mode Font Driver)

- GDI font rendering executes in a kernel mode context before Windows 10
  - GDI contains four font drivers: TrueType, OpenType, Bitmap, and Vector
- Provides an attractive attack surface for both remote and local-authenticated kernel elevation of privilege attacks.
  - Even unsuccessful exploitation attempts would still cause a bug check
UMFD (User Mode Font Driver) Mitigations

• With UMFD in place, any of the font parsing bugs would now yield code execution inside App Container that runs in the user mode
• Addressed significant influx of font parsing bugs
• As a result, there are no more direct remote elevation to kernel privileges in GDI font parsing.
Windows 10 Anniversary Update Mitigations

- Windows Kernel 64-bit ASLR Improvements
- Child Process Restrictions
- CFG Improvements
- Edge Isolation and Containment Improvements
- Secure Kernel: Guest VSM support
64-bit kernel address space layout is now fully dynamic
- PXE randomization is applied to various internal kernel regions

Page table self-reference map & PFN database are randomized
- Custom fixups are used to update constant references to page table location in NT/HAL
- Secure kernel ensures these fixups are safely applied when enabled

GDI shared handle table no longer discloses kernel addresses
Breaks techniques described in research and used in the wild
Child Process Restrictions

- Processes can now restrict child processes from being created
- Prevents code execution via launching a child process
- No longer possible to bypass CFG by calling WinExec & related APIs
- Property of the process token and thus inherited during impersonation
- Enabled via UpdateProcThreadAttribute at process creation time
Control Flow Guard Improvements

- **Longjmp Hardening**
  - Enabled for all Windows binaries

- **Addressed CFG check coverage gaps**
  - `RtlRemoteCall`, Visual C++ CRT delayload, etc

- **Switched to CFG “dispatch mode” on 64-bit by default**

<table>
<thead>
<tr>
<th>CFG ‘Check’ Mode (Windows 10)</th>
<th>CFG ‘Dispatch’ Mode (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mov rcx,&lt;&lt;icall target&gt;&gt;</code></td>
<td><code>mov rax,&lt;&lt;icall target&gt;&gt;</code></td>
</tr>
<tr>
<td><code>call [__guard_check_icall_icall_fptr]</code></td>
<td><code>call [__guard_dispatch_icall_icall_fptr]</code></td>
</tr>
<tr>
<td><code>call rcx</code></td>
<td></td>
</tr>
</tbody>
</table>
Edge Isolation and Containment Improvements

- Flash moved out-of-process to a separate AppContainer
  - Improved isolation and containment of Flash
- Win32k system call filtering enabled for Edge content processes
  - Reduced kernel attack surface
Secure Kernel: Guest VSM support

- Includes HVCI and Hyper Guard support
- Host hypervisor exposes VSM capabilities (VTL protections) to guests
- Each guest brings its own secure kernel instance (runs in guest)
  - Not tied to host OS secure kernel servicing, etc.
- Host OS secure kernel instance does not mediate guest OS VSM
  - Host hypervisor only exposes the VSM primitives to the guest
Microsoft Bounty Programs

1. **Online Services Bug Bounty** (including Microsoft Azure services)
2. **Microsoft Edge RCE on Windows Insider Preview Bug Bounty**
3. **.NET Core and ASP.NET Core RC2 Bug Bounty**
4. **Mitigation Bypass Bounty**
5. **Bounty for Defense**
Resources

1. MSDN Documentation
2. Platform Mitigations
3. SRD Blog
4. Microsoft Bounty Programs
© 2016 Microsoft. All rights reserved. Microsoft, Windows and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.