UNDERSTANDING THE MICROSOFT OFFICE 2013 PROTECTED-VIEW SANDBOX

REcon 2015
Yong Chuan Koh
MWR Infosecurity
#whoami

- Yong Chuan, Koh (@yongchuank)
- Security Consultant, MWR Infosecurity (SG)
- Source code review, binary analysis, malware analysis, etc...
- Research Interests
  - Reverse Engineering
  - Bugs-Finding
  - Exploitation
Outline

• Introduction
• Sandbox Internals
• Inter-Process Communication (IPC) Mechanism
• Microsoft Office 2016
• Conclusion
MS OFFICE 2013 PROTECTED-VIEW SANDBOX

INTRODUCTION
INTRODUCTION

• Sandboxing 101
  – Wikipedia: “...a sandbox is a security mechanism for separating running programs...A sandbox typically provides a tightly controlled set of resources for guest programs to run in, ...A sandbox is implemented by executing the software in a restricted operating system environment, thus controlling the resources (...) that a process may use...”
  – Request broker to work around certain restrictions
• Protected-View Sandbox
  – Introduced since MS Office 2010
  – Only untrusted files are rendered in sandbox
  – Read-only mode
INTRODUCTION

• Motivation
  – Many excellent sandboxing researches
    • IE EPM: “Diving Into IE10’s EPM”, “IE11 Sandbox Escapes”
    • Chrome: “The Chrome Sandbox”
    • Adobe Reader: “Playing in the Reader X Sandbox”, etc
  – No Protected-View publication since 2010
    • Community or MS
• Objective
  – Sandbox restrictions
  – Broker tasks
• Refer to whitepaper for details
• Disclaimer: No RCE 0-day in this presentation
MS OFFICE 2013 PROTECTED-VIEW SANDBOX

SANDBOX INTERNALS

• Architecture
• Initialization
• Restrictions
INTERNALS: ARCHITECTURE

• Methodology
  – Need to “sketch” Protected-View sandbox architecture
  – By comparing against IE sandbox model
    • Likelihood of code-reuse + components
    • Thoroughly researched by many
INTERNALS: ARCHITECTURE

Browser Sandbox Architecture

Diagram showing the architecture of a browser sandbox, including:
- Operating System
- Elevation Policies
- Broker Process
  - Elevation Policy Manager
  - Interception Manager
  - ActiveX Controls
  - COM Objects
- Shims (Interception Client)
- Protected-View Sandbox Process
- API Calls (sandboxed)
- IPC (Inter-Process Communication)
INTERNALS: ARCHITECTURE

• Interception Component
  – Used by sandbox to redirect selected API calls
  – Implemented with API hooking (inline-hooking, IAT hooking or EAT hooking)
  – Checks for patching in sandbox process in function prologues, IAT and EAT
  – Interception component not present in Protected-View
INTERNALS: ARCHITECTURE

- Elevation Policy Component
  - In IE, elevation Policies are stored as registry keys
    - `<AppName> | <AppPath> | <CLSID> | <LaunchPolicyValue>` format
  - Checks for new registry keys with this format
    - MS Office 2007 vs MS Office 2013
  - Elevation Policy component not present in Protected-View

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\15.0\Common\OverridePointerMode
Common\MathFonts\*

Common\LCCache\WordDocBibs\1033\*
Word\Resiliency\DisabledItems\*

Common\LCCache\WordDocParts\1033\*
Word\Security\Trusted Documents\LastPurgeTime

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\15.0\Common\COM Compatibility\{CLSID}\*
Common\Config\{CLSID}\*

User Settings\Excel_Core\Create\Software\Microsoft\Internet_Explorer\ProtocolExecute\"\WarnOnOpen

Excel\Document Inspectors\*
```

INTERNALS: ARCHITECTURE

- Inter-Process Communication (IPC) Component
  - Fundamental in any sandboxing implementation
  - (Name-Pipe) IPC component is present
  - More details later...
INTERNALS: ARCHITECTURE

- Broker Process
  - Elevation Policy Manager
  - Interception Manager
  - ActiveX Controls
  - COM Objects

- Operating System
  - Elevation Policies

- Protected-View Sandbox Process
  - Shims (Interception Client)
  - No ActiveX Controls

- API Calls
  - API Call (sandboxed)

- IPC
  - COM-IPC
  - Name-Pipe IPC

- Trusted and Untrusted File Systems
  - Trusted File N
  - Trusted File 1
  - Untrusted File N
  - Untrusted File 2
  - Untrusted File 1
INTERNALS: INITIALIZATION

- Sandboxing implementation ("Practical Sandboxing on the Windows Platform"):
  - Restricted access token
  - GUI sub-system security
  - Job object restrictions
**INTERNALS: INITIALIZATION**

MSO.sub_00AD0245()

Creates unique sandbox name, `lpSandboxName`:
- `lpSandboxName = "OICE_15_974FA576_32C1D314_<SecureRandomValue & 3FFF>"

Creates sandbox job object, `hSandboxJob`, with `lpSandboxName`
Sets `hSandboxJob` with JobObjectBasicUIRestrictions restrictions

\[JOBOBJECT_BASIC_UI_RESTRICTIONS.UIRestrictionsClass = 0xFF\]
\[JOBOBJECT_BASIC_UI_RESTRICTIONS.UIRestrictionsClass = 0xE8\]
\[JOBOBJECT_BASIC_UI_RESTRICTIONS.UIRestrictionsClass = 0x00\]

UnkObj.fAlternateWinStation == 1 (UnkObj.fAlternateWinStation == 0 & UnkObj.Offset_2A == 0)

Sets `hSandboxJob` with JobObjectExtendedLimitInformation restrictions:
- \[JOBOBJECT_BASIC_LIMIT_INFORMATION.ActiveProcessLimit = 1\]
- \[JOBOBJECT_BASIC_LIMIT_INFORMATION.LimitFlags = 0x2408\]

Actions according to sandbox mode

Sandbox-SID = DeriveAppContainerSidFromAppContainerName() with `lpSandboxName`

Orange: Differences between Low-Integrity mode and AppContainer mode
- If HKLM\Software\Microsoft\Office\15.0\Common\Security\UserAppContainer == 1
- If GetProcAddress() for Userenv.DeriveAppContainerSidFromAppContainerName() and Userenv.GetAppContainerFolderPath() succeeds

Green: Differences due to application settings

Access Token Restrictions

Creates sandbox token, `hSandboxToken`, with restricted rights:
- Disable these SIDs from **TokenGroups**:
  - Domain-Users, Administrators, Console Logon, This Organization, NTLM Authentication, Medium Mandatory Level
- Restricting these SIDs from **TokenGroups**:
  - Restricted Code, Everyone, Users, Logon Session, Sandbox-SID
INTERNALS: INITIALIZATION

Append ACCESS_ALLOWED_ACEs to hSandboxToken DACL:
- ACE 1: SID = TokenUser of hBrokerToken
- ACE 2: SID = S-1-5-5-X-Y (Logon Session)

Sets hSandboxToken integrity level:
- S-1-16-4096 (Low Mandatory Level)

Creates Sandbox directory, lpDirectory, with this ACL:
- lpDirectory = GetUserProfileDirectory() + lpSandboxName
- Administrators; GENERIC_ALL access
- Creator Owner; GENERIC_ALL access
- User-SID of hBrokerToken; GENERIC_ALL access
- Sandbox-SID; GENERIC_ALL access
- Low Mandatory Level; 0x01 access

Creates Sandbox directory, lpDirectory, with new profile:
- lpDirectory = GetAppContainerFolderPath() + "\Temp"
- Profile capability = Sandbox-SID

Checks for creation of new desktop

GUI Sub-System Security

UnkObj.fAlternateWinStation == 1

Creates new desktop:
- szDesktop = "Microsoft Office Isolated Environment"
- dwDesiredAccess = GENERIC_ALL

Creates and connects IPC named-pipe, pNamePipeSecDescriptor, with this ACL:
- lpPipeName = "\pipe\OfficeUser_" + lpSandboxName
- nOutBufferSize = 0x00002000
- Administrators; GENERIC_ALL access
- Creator Owner; GENERIC_ALL access
- User-SID of hBrokerToken; GENERIC_ALL access
- Sandbox-SID; SYNCHRONIZE | READ_CONTROL | 0x8B access
- Low Mandatory Level; 0x01 access
INTERNALS: INITIALIZATION

- **GUI sub-system security:** No desktop isolation
- **Job object restrictions:** No UI restrictions
  - "DIVING INTO IE 10’S ENHANCED PROTECTED MODE SANDBOX"
    - Read/Write to clipboard, screen scraping, screen captures
INTERNALS: RESTRICTIONS

- AppContainer based on capabilities, defines sandbox “boundary”
- In IE, capabilities are defined in Winnt.h or registry key
- In Protected-View, only 1 capability is assigned
  - S-1-15-3-2929230137-1657469040
  - Undocumented and unique to MS Office
## INTERNALS: RESTRICTIONS

<table>
<thead>
<tr>
<th>File Locations</th>
<th>Access Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandbox-SID (S-1-15-2-<em>.</em>/-<em>/-</em>/-<em>/-</em>/-<em>/-</em>)</td>
<td></td>
</tr>
<tr>
<td>%UserProfile\AppData\Local\Packages&lt;lpSandboxName&gt;*</td>
<td>STANDARD_RIGHTS_ALL</td>
</tr>
<tr>
<td>Office-Capability-SID (S-1-15-3-2929230137-1657469040)</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

- Sandbox-SID restricts access to “%UserProfile\AppData\Local\Packages\<sandbox-name>” directory
- Capability-SID does not allow access to file locations
# INTERNALS: RESTRICTIONS

<table>
<thead>
<tr>
<th>Sandbox-SID (S-1-15-2-...-*)</th>
<th>Access Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKCR\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings&lt;Sandbox-SID&gt;</td>
<td>KEY_READ</td>
</tr>
<tr>
<td>HKCR[Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings&lt;Sandbox-SID&gt;\Children</td>
<td>KEY_ALL_ACCESS</td>
</tr>
<tr>
<td>HKCR\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage(&lt;pSandboxName&gt;)*</td>
<td>KEY_ALL_ACCESS</td>
</tr>
<tr>
<td>HKCR\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings&lt;Sandbox-SID&gt;</td>
<td>KEY_READ</td>
</tr>
<tr>
<td>HKCR\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage(&lt;pSandboxName&gt;)*</td>
<td>KEY_ALL_ACCESS</td>
</tr>
<tr>
<td>HKCR\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings&lt;Sandbox-SID&gt;\Children</td>
<td>KEY_ALL_ACCESS</td>
</tr>
<tr>
<td>HKCR\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage(&lt;pSandboxName&gt;)*</td>
<td>KEY_ALL_ACCESS</td>
</tr>
<tr>
<td>HKEY_USERS&lt;WinUser-SID&gt;\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings&lt;Sandbox-SID&gt;</td>
<td>KEY_READ</td>
</tr>
<tr>
<td>HKEY_USERS&lt;WinUser-SID&gt;\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings&lt;Sandbox-SID&gt;\Children</td>
<td>KEY_ALL_ACCESS</td>
</tr>
<tr>
<td>HKEY_USERS&lt;WinUser-SID&gt;\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage(&lt;pSandboxName&gt;)*</td>
<td>KEY_ALL_ACCESS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Office-Capability-SID (S-1-15-3-2929230137-1657469040)</th>
<th>Access Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKCU\Software\Microsoft\Office*</td>
<td>KEY_READ</td>
</tr>
<tr>
<td>HKEY_USERS&lt;WinUser-SID&gt;\Software\Microsoft\Office*</td>
<td>KEY_READ</td>
</tr>
</tbody>
</table>

- **Sandbox-SID restricts access to sandbox-related registry keys**
  - Mostly KEY_ALL_ACCESS access
- **Capability-SID restricts access to Office-related registry keys**
  - Only KEY_READ access
  - HKCU\Software\Microsoft\Office\15.0\Word\Security\Trusted Locations
  - HKCU\Software\Microsoft\Office\15.0\Word\File MRU
INTERNALS: RESTRICTIONS

• Capability-SID does not allow network outbound connections
  – WSAEACCES “Permission Denied” error
INTERNALS

(*) UI Restrictions:
- JOB_OBJECT_UILIMIT_HANDLES
- JOB_OBJECT_UILIMIT_READCLIPBOARD
- JOB_OBJECT_UILIMIT_WRITECLIPBOARD

MS OFFICE 2013 PROTECTED-VIEW SANDBOX

INTER-PROCESS COMMUNICATION (IPC) MECHANISM

• Internal Objects
• Format of IPC Messages
• Purpose of IPC Messages
**IPC: INTERNAL OBJECTS**

- **ThreadMgr Object**
  - lpIPCMsgSendRecv
  - lpViewRestrictions
  - lpViewMgr
  - ...

- **ViewMgr Object**
  - lpThreadMgr_1
  - lpViewTracker_1
  - ...
  - lpThreadMgr_4
  - lpViewTracker_4

- **ViewTracker Object**
  - lpThreadMgr
  - ui32NumViewFile
  - ui32ViewFileArraySize
  - lpViewFileArray
  - ...

- **Array of ViewFile Objects**
  - **ViewFile_1 Object (free/busy)**
    - lpSameBrokerApp
    - ...
    - ...
  - **ViewFile_N Object (free/busy)**
    - lpSameBrokerApp
    - ...

- **SameBrokerApp Object**
  - lpWWLIBIPCMsg
  - ...

- **WWLIBIPCMsg Object**
  - ...

- **IPCViewRestrictions Object**
  - ...

- **IPCMsgSendRecv Object**
# IPC: INTERNAL OBJECTS

## ThreadMgr Object

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>08</td>
<td>LPVOID</td>
<td>lpIPCMsgSendRecv</td>
<td>Pointer to an object that sends/receives IPC messages</td>
</tr>
<tr>
<td>0C</td>
<td>LPVOID</td>
<td>lpViewRestrictions</td>
<td>Pointer to an object describing the sandbox restrictions</td>
</tr>
<tr>
<td>10</td>
<td>LPVOID</td>
<td>lpViewMgr</td>
<td>Pointer to ViewMgr object</td>
</tr>
</tbody>
</table>

## ViewMgr Object

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>UINT32</td>
<td>ui32Num</td>
<td>Number of untrusted files + 2</td>
</tr>
<tr>
<td>08</td>
<td>LPVOID</td>
<td>lpThreadMgr_1</td>
<td>-</td>
</tr>
<tr>
<td>0C</td>
<td>LPVOID</td>
<td>lpViewTacker_1</td>
<td>Pointer to ViewTracker object</td>
</tr>
</tbody>
</table>

## ViewTracker Object

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0C</td>
<td>LPVOID</td>
<td>lpThreadMgr</td>
<td>Pointer to ThreadMgr object</td>
</tr>
<tr>
<td>14</td>
<td>UINT32</td>
<td>ui32NumViewFile</td>
<td>Number of Protected-View files, or number of busy slots in ViewFilesArray</td>
</tr>
<tr>
<td>18</td>
<td>UINT32</td>
<td>ui32ViewFileArraySize</td>
<td>Size of ViewFilesArray</td>
</tr>
<tr>
<td>1C</td>
<td>UINT32</td>
<td>ui32Unknown</td>
<td>Higher 2 bytes are used to calculate new ui32ViewFileArraySize when ViewFilesArray is full</td>
</tr>
<tr>
<td>20</td>
<td>LPVOID</td>
<td>lpViewFileArray</td>
<td>Pointer to an array of ViewFile objects</td>
</tr>
</tbody>
</table>
### IPC: INTERNAL OBJECTS

#### ViewFile Object (size 0x1C)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>UINT32</td>
<td>ui32ViewID</td>
<td>Unique ID to identify respective untrusted file.</td>
</tr>
<tr>
<td>08</td>
<td>HWND</td>
<td>hWnd</td>
<td>hWnd for “OPH Previewer Window” class</td>
</tr>
<tr>
<td>0C</td>
<td>LPWSTR</td>
<td>lpwFileName</td>
<td>Pointer to full-path to original file</td>
</tr>
<tr>
<td>10</td>
<td>LPWSTR</td>
<td>lpwTemporaryFileName</td>
<td>Pointer to full-path to temp file in sandbox dir</td>
</tr>
<tr>
<td>14</td>
<td>LPVOID</td>
<td>lpSameBrokerApp</td>
<td>Pointer to SameBrokerApp object</td>
</tr>
<tr>
<td>18</td>
<td>UINT32</td>
<td>ui32SessionEnableHyperlinks</td>
<td>Used in Tag 0x091000 (TRUE or FALSE)</td>
</tr>
</tbody>
</table>

#### SameBrokerApp Object

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0C</td>
<td>LPVOID</td>
<td>lpWWLIBIPCMsg</td>
<td>Pointer to WWLIBIPCMsg object</td>
</tr>
<tr>
<td>84</td>
<td>HWND</td>
<td>hWnd</td>
<td>hWnd used in 0x061000 and 0x101000 IPC message</td>
</tr>
<tr>
<td>90</td>
<td>LPVOID</td>
<td>lpDRMStream</td>
<td>Used in 0x081000 IPC message</td>
</tr>
<tr>
<td>B0</td>
<td>LPVOID</td>
<td>lpTaskList</td>
<td>Used in 0x0B1000 IPC message</td>
</tr>
</tbody>
</table>

#### WWLIBIPCMsg Object

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1C</td>
<td>UINT32</td>
<td>ui32IPC0A1100</td>
<td>Used in 0xD0A1100 IPC message</td>
</tr>
<tr>
<td>20</td>
<td>UCHAR</td>
<td>uchIPCO91100Contents</td>
<td>Buffer storing IPC 0x091100 message contents</td>
</tr>
<tr>
<td>4C</td>
<td>UINT32</td>
<td>ui32IPC071100MsgID</td>
<td>MsgID of IPC 0x071100 message</td>
</tr>
<tr>
<td>50</td>
<td>UINT32</td>
<td>ui32IPC081100MsgID</td>
<td>MsgID of IPC 0x081100 message</td>
</tr>
<tr>
<td>54</td>
<td>UINT32</td>
<td>ui32IPC091100MsgID</td>
<td>MsgID of IPC 0x091100 message</td>
</tr>
<tr>
<td>58</td>
<td>UINT32</td>
<td>ui32IPC031100MsgID</td>
<td>MsgID of IPC 0x031100 message</td>
</tr>
<tr>
<td>5C</td>
<td>UINT32</td>
<td>ui32IPC041100MsgID</td>
<td>MsgID of IPC 0x041100 message</td>
</tr>
<tr>
<td>60</td>
<td>UINT32</td>
<td>ui32IPC0E1100MsgID</td>
<td>MsgID of IPC 0x0E1100 message</td>
</tr>
<tr>
<td>64</td>
<td>UCHAR</td>
<td>uchIPCO41100Contents</td>
<td>Buffer storing IPC 0x041100 message contents</td>
</tr>
<tr>
<td>8C</td>
<td>UCHAR</td>
<td>uchIPCO31100Contents</td>
<td>Buffer storing IPC 0x031100 message contents</td>
</tr>
</tbody>
</table>
IPC: MESSAGE FORMAT

Message Header

Request Header for MSO-Group (size 0x10-bytes)
- UINT32 ui32VirtualKey: Used in 0x071000 request. Otherwise ignored
- UINT32 ui32MsgTag: Type of (22) IPC request
  Ranges 0x001000 to 0x151000, at inc 0x010000
- UINT32 ui32MsgID: Matches IPC request to its response
- UINT32 ui32MsgSize: Total size of IPC message in bytes (max 0x2000)

Request Header for WWLIB-Group (size 0x14-bytes)
- UINT32 ui32VirtualKey: 
- UINT32 ui32MsgTag: Type of (16) IPC request
  Ranges 0x001100 to 0x0F1100, at inc 0x010000
- UINT32 ui32MsgID: 
- UINT32 ui32MsgSize: 
- UINT32 ui32ViewID: Sandboxed file ID. Corr to lpViewFile->ui32ViewID

Response Header (size 0x10-bytes)
- UINT32 ui32Status: Return status of IPC request
- UINT32 ui32MsgTag: 
- UINT32 ui32MsgID: Matches IPC request to its response
- UINT32 ui32MsgSize: 

Message Body Format Specific to MsgTag

.........
IPC: MESSAGE FORMAT

**Pink**: Sanity checks on IPC message, according to MsgTag

**Purple**: Service functions
IPC: MESSAGE 0X001000

- No message body
- Registry key “HKEY_CURRENT_USER\Software\Policies”
  - Duplicate and return, with KEY_READ access
  - Work-around sandbox restrictions
  - Does not keep duplication count
IPC: MESSAGE 0X061000

Message Body for 0x061000 Request

<table>
<thead>
<tr>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>UINT32</td>
<td>ui32ViewID</td>
<td>-</td>
</tr>
<tr>
<td>DWORD</td>
<td>dwMsg</td>
<td>Any of WM_KEYDOWN, WM_KEYUP, WM_CHAR, WM_SYSKEYDOWN, WM_SYSKEYUP or WM_SYSCHAR</td>
</tr>
<tr>
<td>WPARAM</td>
<td>wParam</td>
<td>-</td>
</tr>
<tr>
<td>LPARAM</td>
<td>lParam</td>
<td>-</td>
</tr>
</tbody>
</table>

- PostMessageW() request to “Microsoft Word Document” window
- Cannot disable Protected-View:
  - Sandbox window in focus
  - Limited set of dwMsg

Protected View

This file came from the Internet, so we opened it in a way that helps to keep your computer safe from viruses (just in case).

Don’t worry—you can continue reading in this view. If you need to edit, and you trust this file, then enable editing.
Protected View Settings
Learn more about Protected View

This is a test document.

End of document
**IPC: MESSAGE 0X091000**

<table>
<thead>
<tr>
<th>Message Body for 0x091000 Request</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
</tr>
<tr>
<td>UINT32</td>
</tr>
<tr>
<td>LPWSTR</td>
</tr>
<tr>
<td>LPWSTR</td>
</tr>
<tr>
<td>DWORD</td>
</tr>
<tr>
<td>DWORD</td>
</tr>
<tr>
<td>HWND</td>
</tr>
<tr>
<td>WCHAR[]</td>
</tr>
<tr>
<td>WCHAR[]</td>
</tr>
</tbody>
</table>

- **HlinkNavigateToStringReference()** request to visit URL
- User permission
  - lpViewFile->ui32SessionEnableHyperlinks
  - Per file per session
**IPC: MESSAGE 0X0A1000**

- “Enter Password” prompt for password-protected file

<table>
<thead>
<tr>
<th>Message Body for 0x0A1000 Request</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
</tr>
<tr>
<td>UINT32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message Body for 0x0A1000 Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
</tr>
<tr>
<td>LPWSTR</td>
</tr>
<tr>
<td>UINT16</td>
</tr>
<tr>
<td>WCHAR[]</td>
</tr>
</tbody>
</table>
IPC: MESSAGE 0X0C1000

- Windows Error-Reporting Request
  - Typical: AppA -> DWWIN.EXE (on AppA)
  - WINWORD: Sandbox ->[0x0C1000]- Broker -> DWWIN.EXE (on Sandbox)
    - JOBOBJECT_BASIC_LIMIT_INFORMATION.ActiveProcessLimit = 1
- Definition
  - hSandboxSharedMem: Memory that sandbox shares with broker
  - hBrokerSharedMem: Memory that broker shares with DWWIN

<table>
<thead>
<tr>
<th>Message Body for 0x0C1000 Request</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
</tr>
<tr>
<td>HANDLE</td>
</tr>
<tr>
<td>HANDLE</td>
</tr>
<tr>
<td>LPWSTR</td>
</tr>
<tr>
<td>UINT16</td>
</tr>
<tr>
<td>WCHAR[]</td>
</tr>
</tbody>
</table>
### IPC: MESSAGE 0x0C1000

#### Possible Format of `hBrokerShareMem` Shared Memory

<table>
<thead>
<tr>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>UINT32</td>
<td>0x00009C9C</td>
<td>Size of <code>hBrokerSharedMem</code> memory</td>
</tr>
<tr>
<td>UINT32</td>
<td>ui32ProtectedViewPID</td>
<td>Sandbox Process-ID</td>
</tr>
<tr>
<td>UINT32</td>
<td>ui32ProtectedViewTID</td>
<td>TID of faulting thread in sandbox. Copied from <code>hSandboxSharedMem</code> offset 0x0C.</td>
</tr>
<tr>
<td>UINT32</td>
<td>uiProtectedViewEIP</td>
<td>EIP of faulting instruction in sandbox. Copied from <code>hSandboxSharedMem</code> offset 0x10.</td>
</tr>
<tr>
<td>LPVOID</td>
<td>lpProtectedViewPEP</td>
<td>Exception pointers in sandbox. Copied from <code>hSandboxSharedMem</code> offset 0x14.</td>
</tr>
<tr>
<td>HANDLE</td>
<td>hSandboxProcess</td>
<td>Sandbox process handle</td>
</tr>
<tr>
<td>WCHAR[0x104]</td>
<td>wzModulesList</td>
<td>List of modules loaded in sandbox, separated by Null</td>
</tr>
</tbody>
</table>
| WCHAR[0x400] | wzWerSubmitFilesList      | List of files to submit to WER with `CWatsonReport::AddFilesToReport()`, separated by ‘|’:
|          |                              | • Sandbox-directory + wzAdditionalWerFileName                                               |
|          |                              | • %Temp% + "winword.exe.OsrHost.dmp.dat"                                                    |
|          |                              | • %Temp% + "winword.exe.OsrHost.cvr.dat"                                                    |
| UINT32   | ui32CrashParamFlag          | If Null, next 11 fields are ignored. Copied from `hSandboxSharedMem` offset 0x8470         |
| WCHAR[0xFF] | wzEventType                | `pwzEventType` in `WerReportCreate()` Copied from `hSandboxSharedMem` offset 0x8474         |
| WCHAR[0xFF] | wzParam0                   | `pwzValue` for WER_P0 in `WerReportSetParameter()` Copied from `hSandboxSharedMem` offset 0x8672 |
| WCHAR[0xFF] | wzParam9                   | `pwzValue` for WER_P9 in `WerReportSetParameter()` Copied from `hSandboxSharedMem` offset 0x9860 |
IPC: MESSAGE 0X0C1000

Windows Error Reporting

<table>
<thead>
<tr>
<th>Process</th>
<th>Priority</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WINWORD.EXE</td>
<td>0.03</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>WINWORD.EXE</td>
<td>&lt; 0.01</td>
<td>AppContainer</td>
<td></td>
</tr>
<tr>
<td>DW20.exe</td>
<td>0.02</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>DW20.exe</td>
<td>0.03</td>
<td>Medium</td>
<td></td>
</tr>
</tbody>
</table>

Level 1 Error Data

“"I want more information""

files.cab

Level 2 Error Data

Delete
IPC: MESSAGE 0x0C1000

• Issue 1: Read-AV
  – Copies 0x15F0 bytes from hSandboxSharedMem to hBrokerSharedMem
  – Parameters for WerReportCreate() and WerReportSetParameter()
  – No src-buffer size check
IPC: MESSAGE 0X0C1000

- Issue 2: Directory-Traversal
  - Level 2 Error Data
  - DWWIN.EXE adds (full-path) wzAdditionalWerFileName file to .cab with WatsonReport::AddFilesToReport()
- No backslash character in wzAdditionalWerFileName field
- Prepends sandbox-directory to wzAdditionalWerFileName field
IPC: MESSAGE 0X0C1000

- MSDN
  - "File I/O functions in the Windows API convert "/" to "/" as part of converting the name to an NT-style name..."
- Use "/" in wzAdditionalWerFileName to traverse out of sandbox container
  - wzAdditionalWerFileName = “../Desktop/thisismyfile.txt”
IPC: MESSAGE 0X0C1000

• Demo
IPC: MESSAGE 0X0C1000

• Effect of issue 2
  – Steal arbitrary file if WER server is compromised ("Watson.microsoft.com" by default, or enterprise WER server)
  – Delete arbitrary file (by DWWIN.EXE)
• Response from MSRC
  – "...We’ve completed our investigation on this issue but this doesn't meet our current bug bar for servicing. We are looking at incorporating this fix as part of our future security plans ..."
  – Well, see “Microsoft Office 2016” section...
IPC: MESSAGE 0x0F1000

- “Microsoft Help” request
  - CreateProcessW (""CLVIEW.EXE" "WINWORD" "Word""), or
  - HlinkNavigateToStringReference (MS Office Developer Help website)
- Depends on request fields

<table>
<thead>
<tr>
<th>Message Body for 0x0F1000 Request</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
</tr>
<tr>
<td>UINT32</td>
</tr>
<tr>
<td>UINT32</td>
</tr>
<tr>
<td>LPWSTR</td>
</tr>
<tr>
<td>LPWSTR</td>
</tr>
<tr>
<td>UINT16</td>
</tr>
<tr>
<td>WCHAR[0x104]</td>
</tr>
<tr>
<td>UINT16</td>
</tr>
<tr>
<td>WCHAR[0x104]</td>
</tr>
</tbody>
</table>
IPC: MESSAGE 0X0F1000

- Null-Deference Read-AV
  - `lpwHelpID` vs “NoHelp”, w/o checking `lpwHelpID` for NULL

```assembly
01170021 mov   esi, [ebp+Src]  ; lpHelpID
01170024 push  edi
01170025 mov   edi, [ebp+arg_4]  ; lpData
01170028 mov   ebx, ecx
0117002a mov   ecx, offset aNohelp  ; “NoHelp”
0117002f mov   eax, esi

01170031 loop_cmp_HelpID:  
01170031 mov   dx, [eax]  ; Read-AV: [eax] without checking if it (lpHelpID) is Null
01170034 cmp   dx, [ecx]  ; wcscmp (HelpID vs "NoHelp")
01170037 jnz  short HelpID_not_eq_NoHelp
```
MS OFFICE 2013 PROTECTED-VIEW SANDBOX

MICROSOFT OFFICE 2016

• Sandbox Internals
• IPC Messages
MS OFFICE 2016: SANDBOX INTERNALS

• Sandbox Code
  – Common routines moved to individual modules (Mso20win32client.dll, Mso30win32client.dll, etc)
  – New Win8 security assertions
  – Diffing tools would not work well

• Sandbox Restrictions
  – No new capability-SID
  – Sandbox boundary remain unchanged
MS OFFICE 2016: SANDBOX INTERNALS

- Sandbox Initialization
  - New `CreateProcessW()` option in addition to `CreateProcessAsUserW()`
  - Does not seem reachable (BYTE [ebx+81h])
MS OFFICE 2016: IPC 0X0C1000

- Directory-Traversal Issue
  - 13 Feb 2015: Response from MSRC ("..fix as part of future security plans")
  - ~05 May 2015: MS Office 2016 Preview is released
MS OFFICE 2016: IPC 0X161000

- New IPC message
- Only for OUTLOOK-loaded
- To protect/unprotect sandbox window for RMS-protected files

<table>
<thead>
<tr>
<th>Size</th>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>UINT32</td>
<td>ui32ViewID</td>
<td></td>
</tr>
<tr>
<td>UINT32</td>
<td>hUnprotectWnd</td>
<td>Window handle to unprotect for.</td>
</tr>
<tr>
<td>UINT32</td>
<td>ui32Boolean</td>
<td>If 0, unprotects window with IpcUnprotectWindow() or _IpcUnprotectWindowNoDRM(). If 1, protects window with IpcProtectWindow() or _IpcProtectWindowNoDRM()</td>
</tr>
</tbody>
</table>
MS Office 2016: IPC 0x031100

- Larger message size (0x1E8 -> 0x1F4 bytes)
- No change in handling of message
MS OFFICE 2013 PROTECTED-VIEW SANDBOX

CONCLUSION
CONCLUSION

• Simplistic sandbox architecture
• Reduced IPC messages
  – Adobe Reader (200+) vs MS Office (38)
• Sandbox Internals
  – No desktop isolation
  – No job (UI) restrictions
• IPC Mechanism
  – 2 read-AVs
  – Directory-traversal
• What to expect...
  – Not much changes
    • 1 new ICP messages
    • New CreateProcessW() option
• Still a good sandbox (kernel-bugs aside)...

Labs.mwrinfosecurity.com  |  © MWR Labs
REFERENCES

Thank You!

Questions?