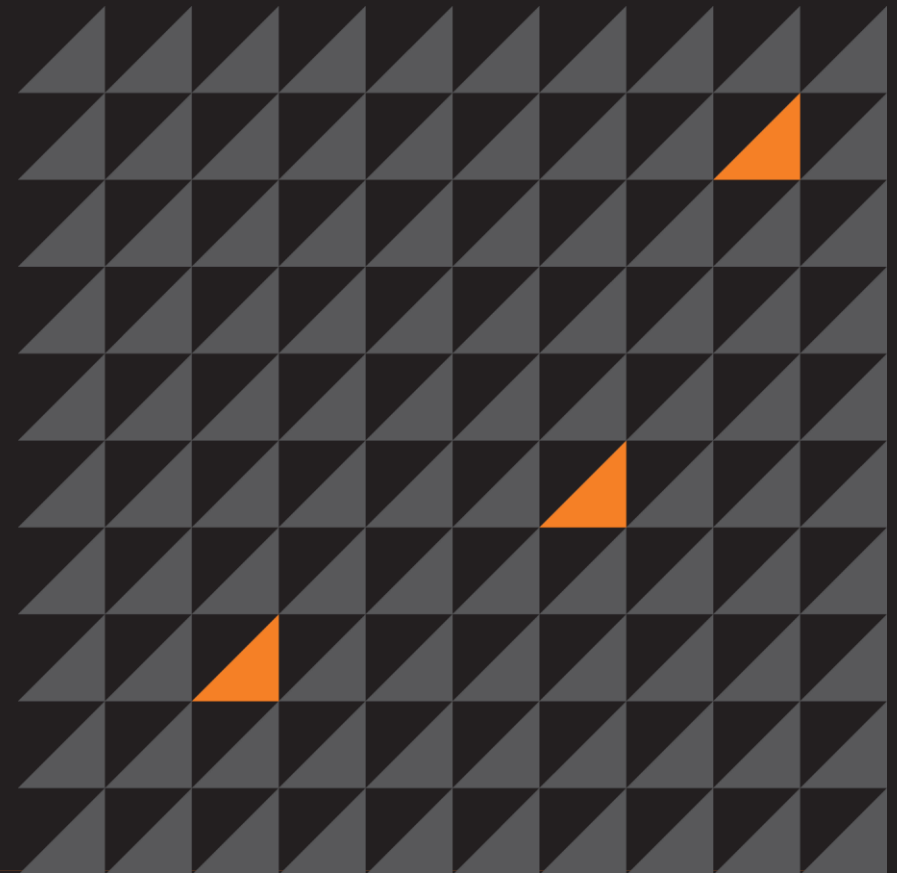




# UNDERSTANDING THE MICROSOFT OFFICE 2013 PROTECTED-VIEW SANDBOX

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

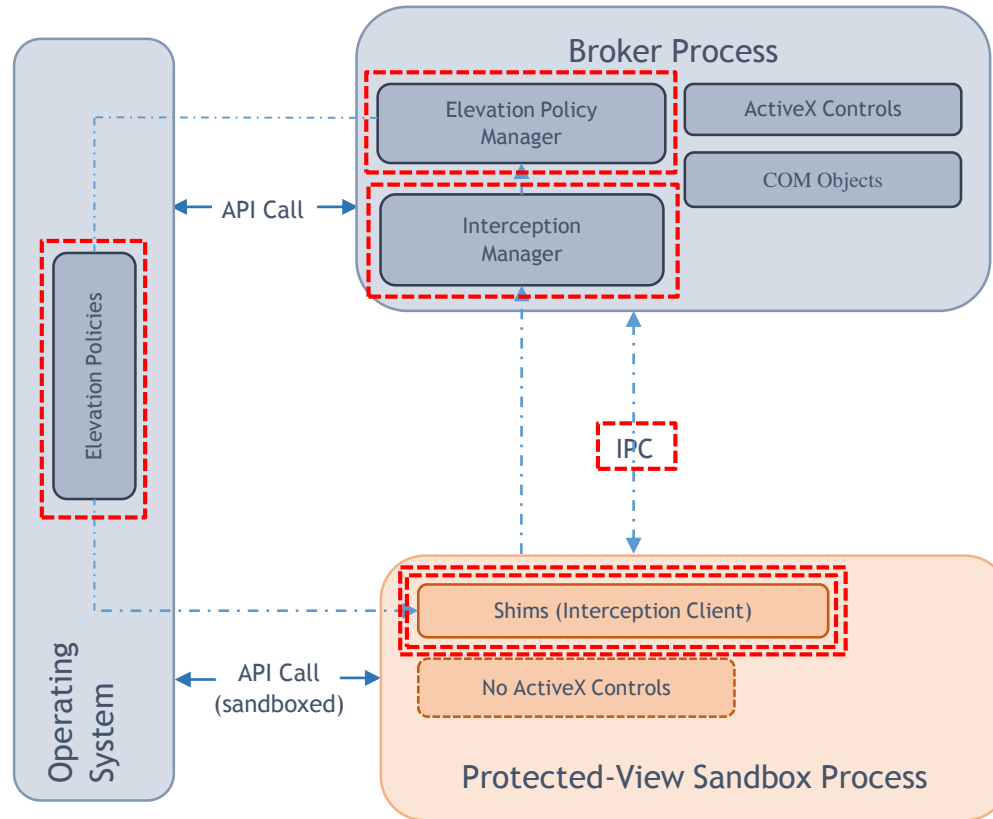
---

- 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





## 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\*	

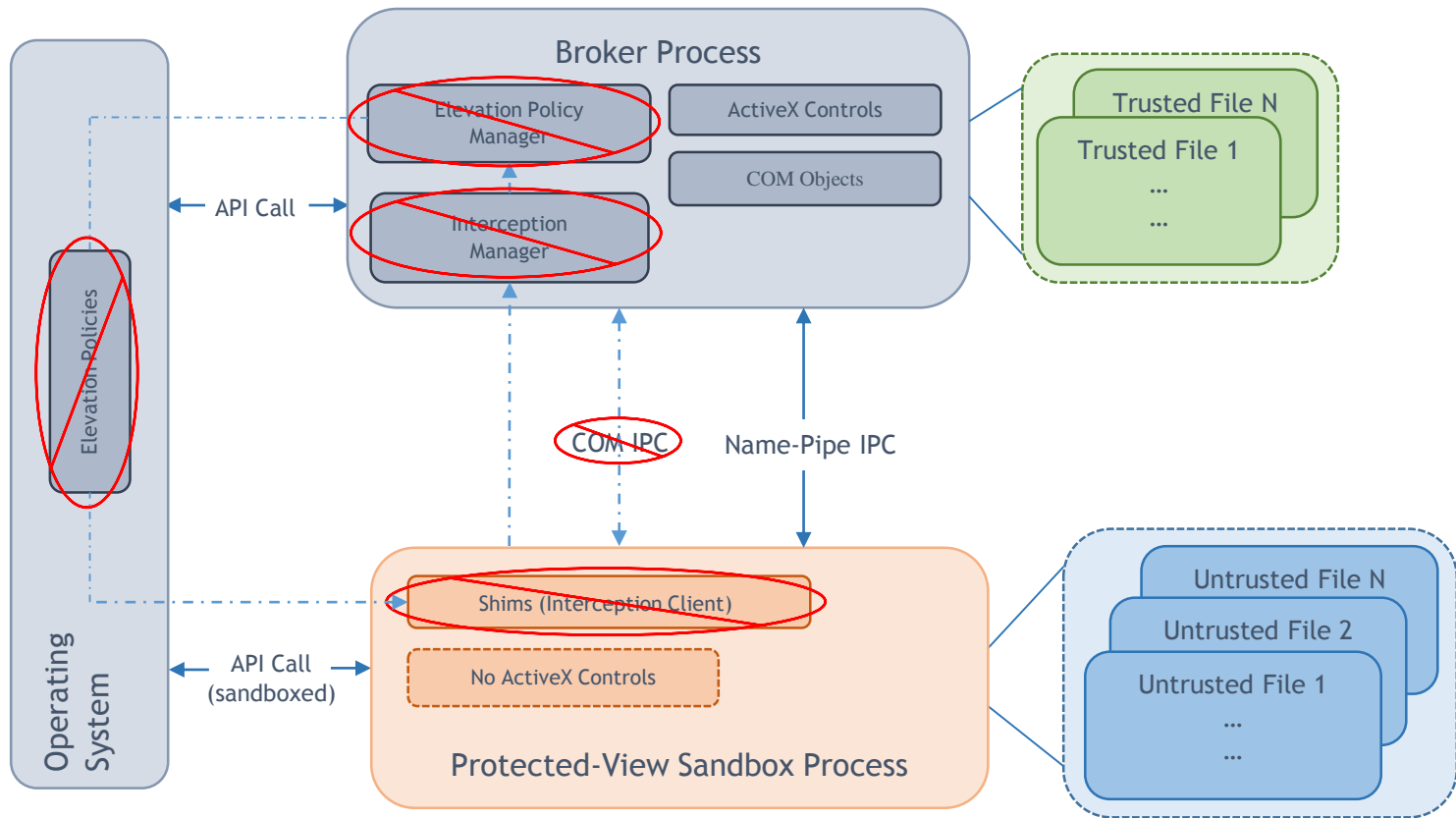


## INTERNALS: ARCHITECTURE

---

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

## INTERNALS: ARCHITECTURE



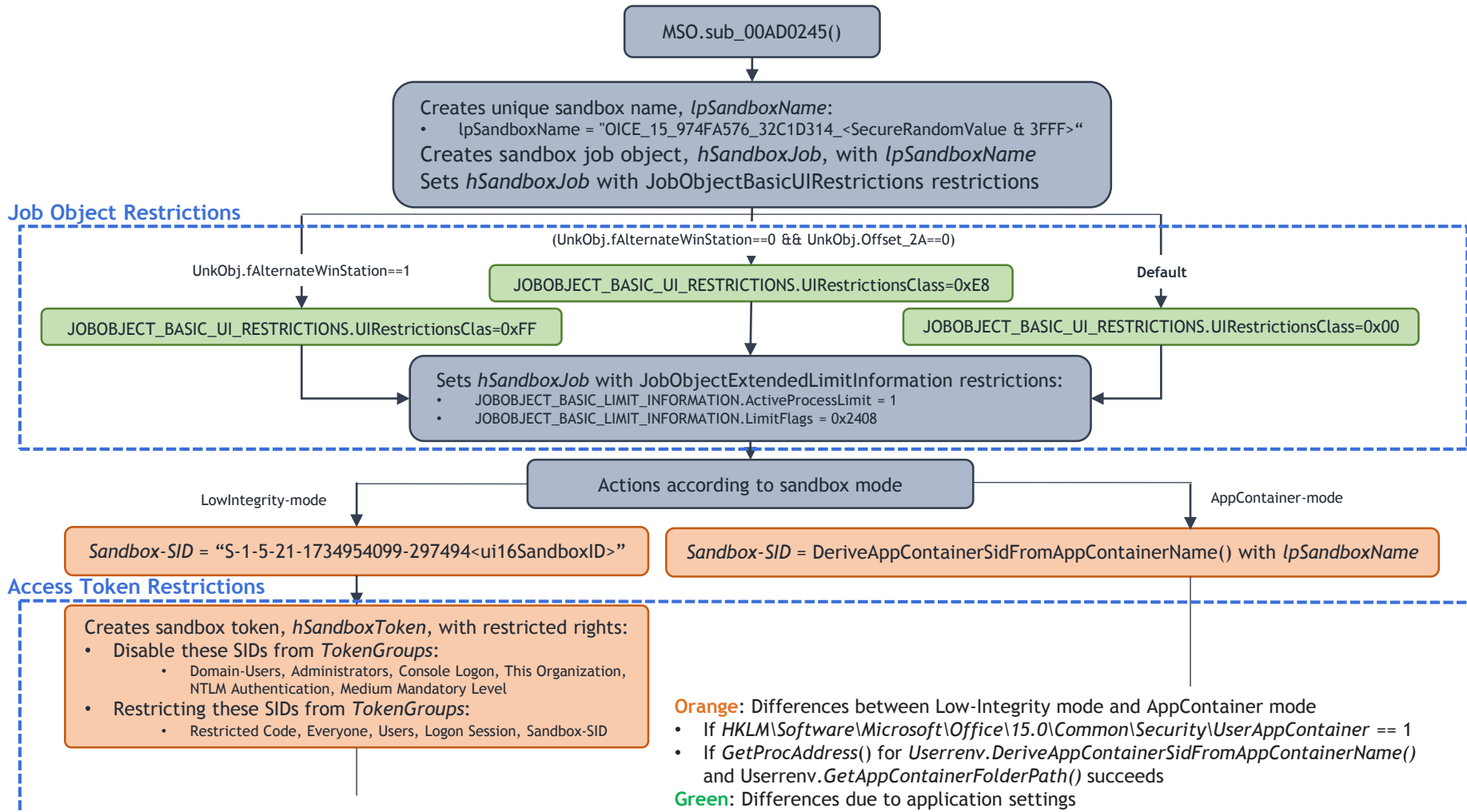


## INTERNALS: INITIALIZATION

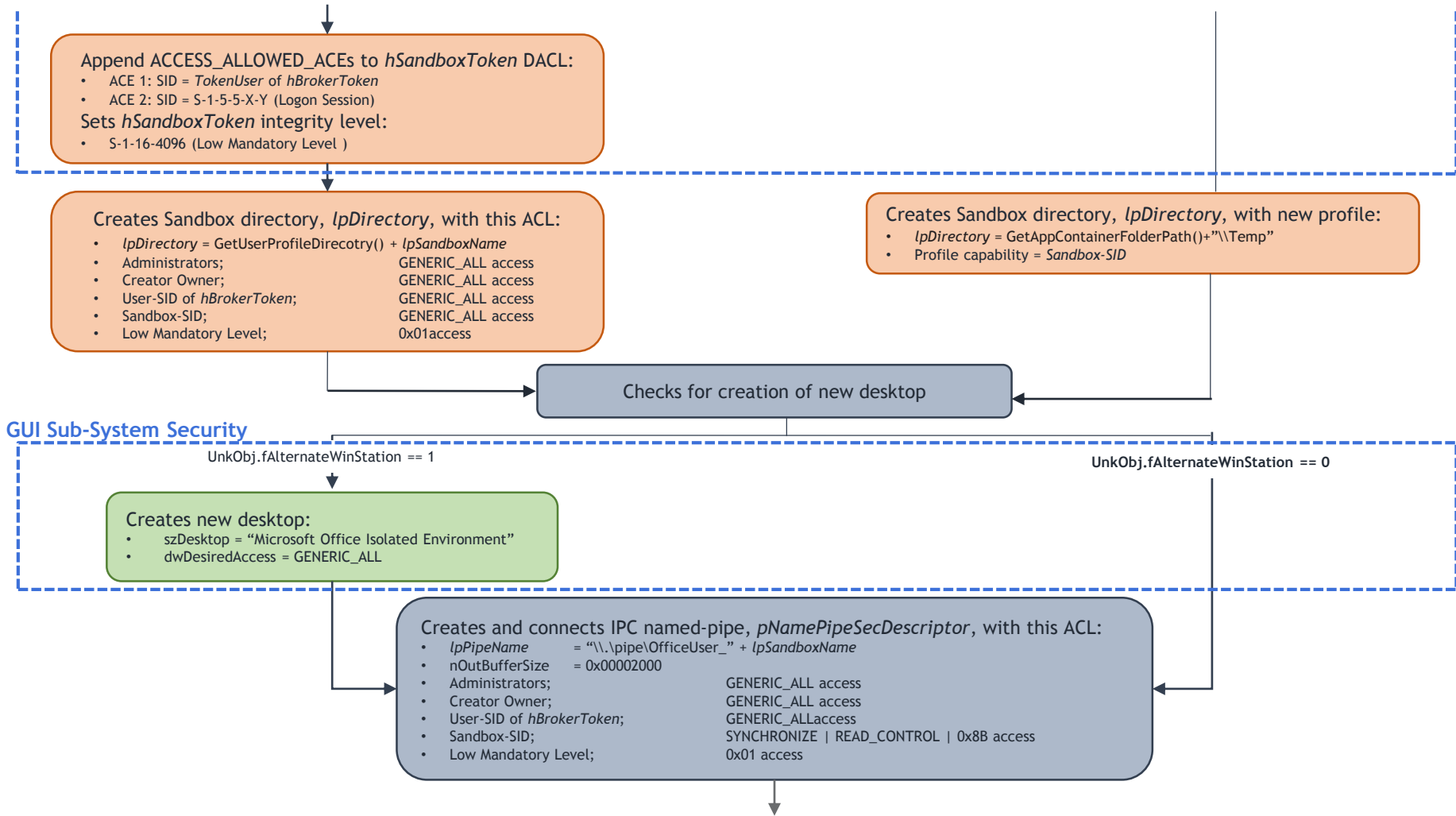
---

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

# INTERNALS: INITIALIZATION

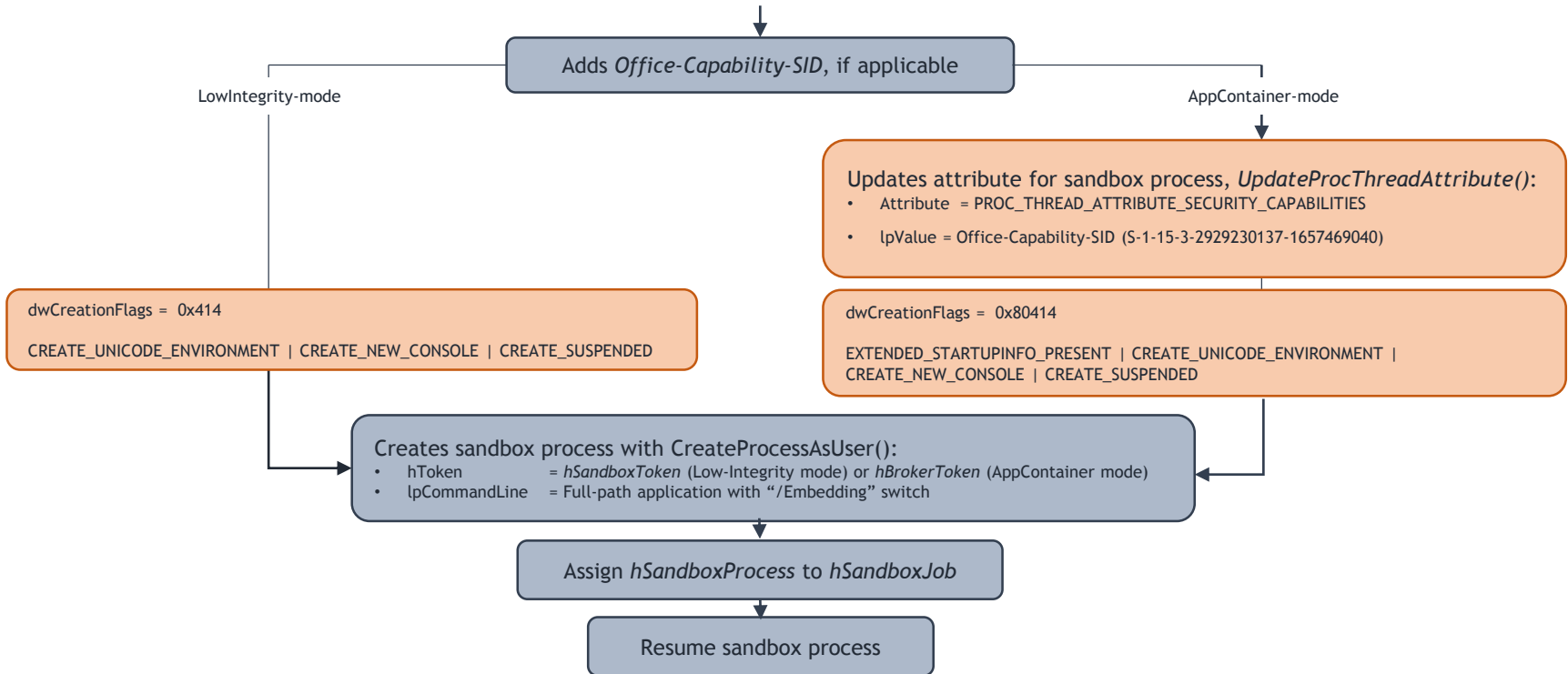


# INTERNALS: INITIALIZATION





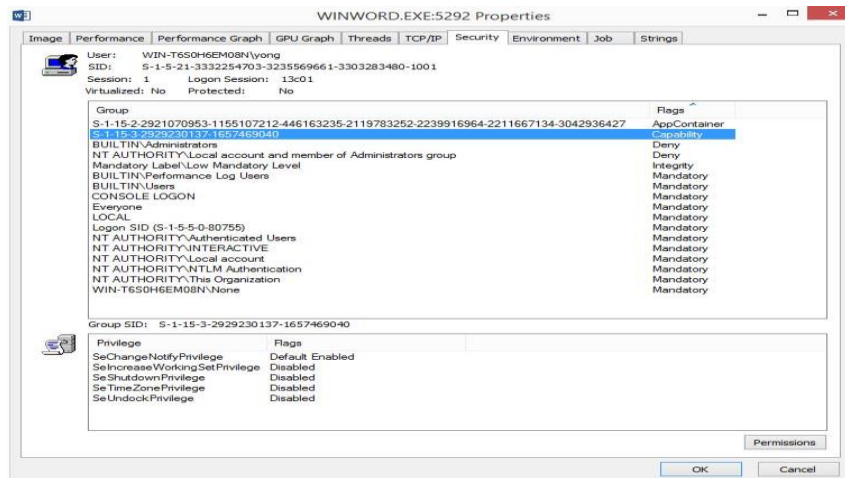
## INTERNALS: INITIALIZATION



- 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

---

File Locations	Access Mask
<b>Sandbox-SID (S-1-15-2-**-**-**-**)</b>	
%UserProfile%\AppData\Local\Packages\<lpSandboxName>*	STANDARD_RIGHTS_ALL   0x1FF
<b>Office-Capability-SID (S-1-15-3-2929230137-1657469040)</b>	
None	None

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

## INTERNALS: RESTRICTIONS

Registry Keys	Access Mask
<b>Sandbox-SID (S-1-15-2-**-**-**-**)</b>	
HKCR\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\ <i>&lt;Sandbox-SID&gt;</i>	KEY_READ
HKCR \Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\ <i>&lt;Sandbox-SID&gt;</i> \Children	KEY_ALL_ACCESS
HKCR \Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\ <i>&lt;lpSandboxName&gt;</i> *	KEY_ALL_ACCESS
HKCR \Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\ <i>&lt;Sandbox-SID&gt;</i>	KEY_READ
HKCR \Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\ <i>&lt;Sandbox-SID&gt;</i> \Children	KEY_ALL_ACCESS
HKCR \Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\ <i>&lt;lpSandboxName&gt;</i> *	KEY_ALL_ACCESS
HKEY_USERS\ <i>&lt;WinUser-SID&gt;</i> \Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\ <i>&lt;Sandbox -SID&gt;</i>	KEY_READ
HKEY_USERS\ <i>&lt;WinUser-SID&gt;</i> \Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\ <i>&lt;Sandbox -SID&gt;</i> \Children	KEY_ALL_ACCESS
HKEY_USERS\ <i>&lt;WinUser-SID&gt;</i> \Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\ <i>&lt;lpSandboxName&gt;</i> *	KEY_ALL_ACCESS
<b>Office-Capability-SID (S-1-15-3-2929230137-1657469040)</b>	
HKCU\Software\Microsoft\Office\*	KEY_READ
HKEY_USERS\ <i>&lt;WinUser-SID&gt;</i> \Software\Microsoft\Office\*	KEY_READ

- 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

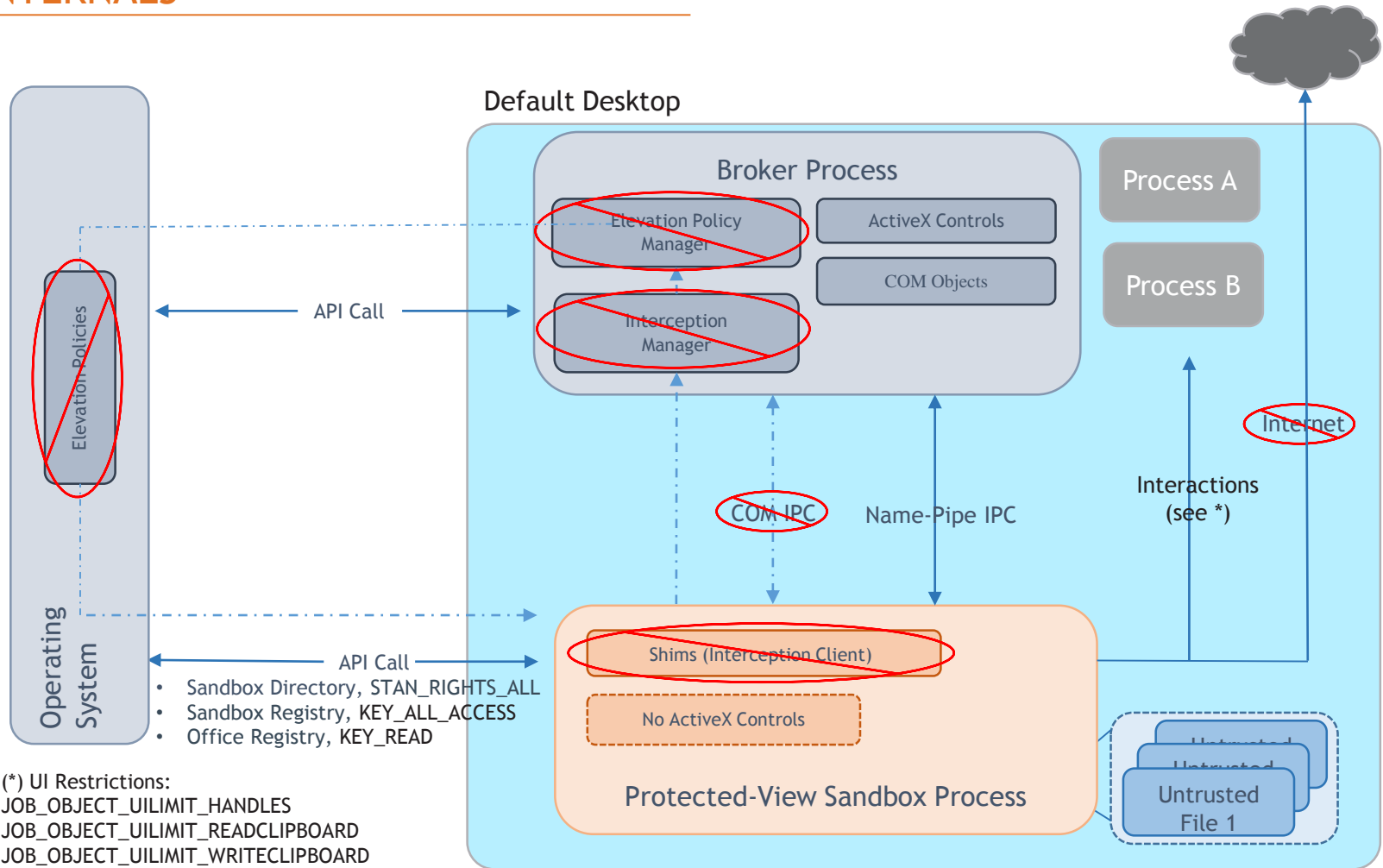


## INTERNALS: RESTRICTIONS

---

- 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  
 ...

([https://msdn.microsoft.com/en-us/library/windows/desktop/ms684152\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms684152(v=vs.85).aspx))

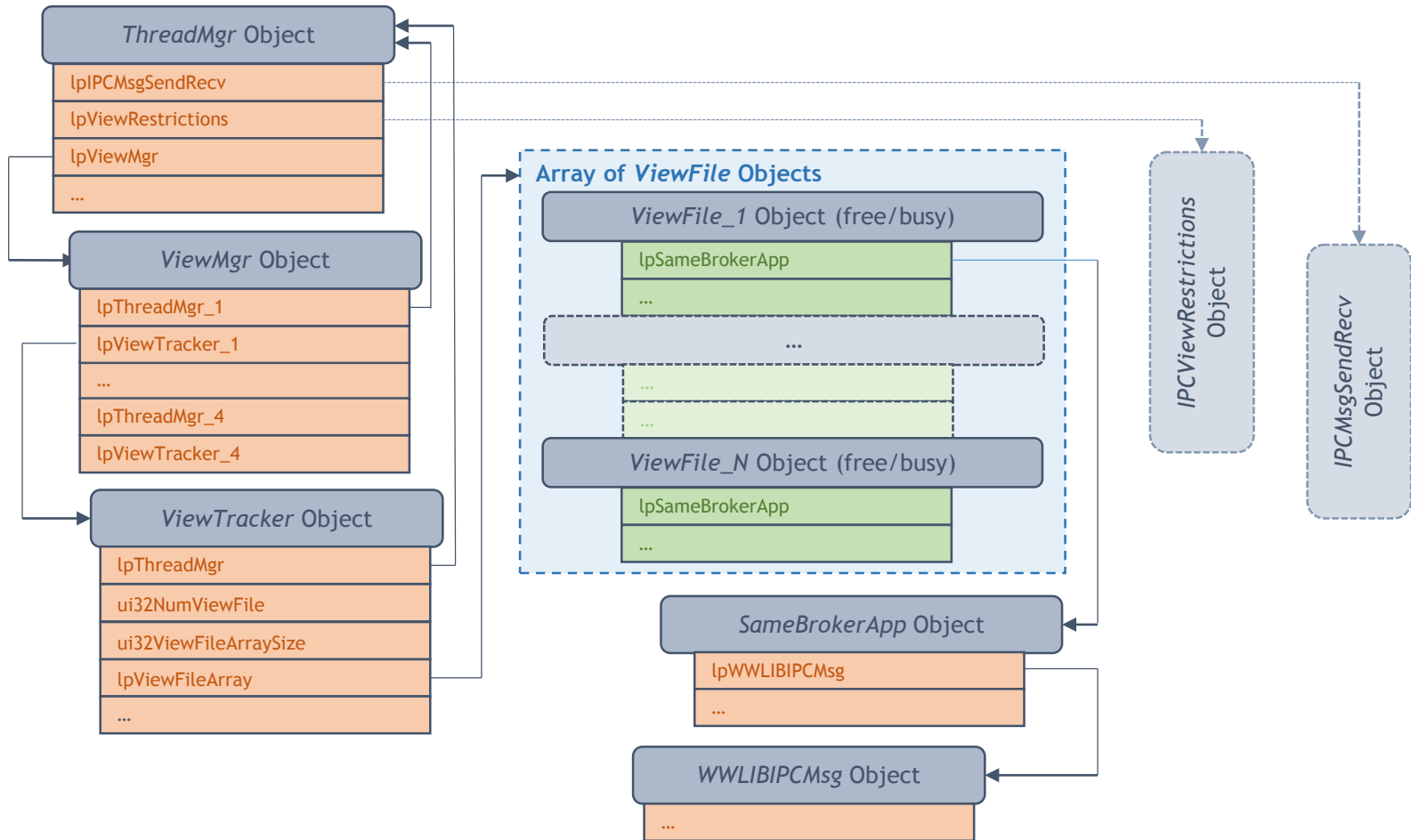


# MS OFFICE 2013 PROTECTED-VIEW SANDBOX

## INTER-PROCESS COMMUNICATION (IPC) MECHANISM

- Internal Objects
- Format of IPC Messages
- Purpose of IPC Messages

# IPC: INTERNAL OBJECTS





## IPC: INTERNAL OBJECTS

**ThreadMgr Object**

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

**ViewMgr Object**

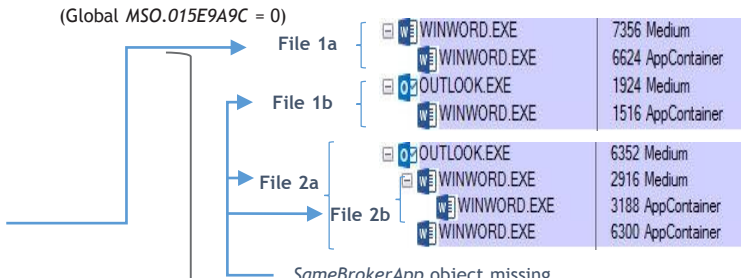
Offset	Size	Field	Comment
04	UINT32	ui32Num	Number of untrusted files + 2
08	LPVOID	lpThreadMgr_1	-
0C	LPVOID	lpViewTacker_1	Pointer to <i>ViewTracker</i> object

**ViewTracker Object**

Offset	Size	Field	Comment
0C	LPVOID	lpThreadMgr	Pointer to <i>ThreadMgr</i> object
14	UINT32	ui32NumViewFile	Number of Protected-View files, or number of busy slots in <i>ViewFilesArray</i>
18	UINT32	ui32ViewFileArraySize	Size of <i>ViewFilesArray</i>
1C	UINT32	ui32Unknown	Higher 2 bytes are used to calculate new <i>ui32ViewFilesArraySize</i> when <i>ViewFilesArray</i> is full
20	LPVOID	lpViewFileArray	Pointer to an array of <i>ViewFile</i> objects

# IPC: INTERNAL OBJECTS

ViewFile Object (size 0x1C)			
Offset	Size	Field	Comment
04	UINT32	ui32ViewID	Unique ID to identify respective untrusted file.
08	HWND	hOPHWnd	hWnd for "OPH Previewer Window" class
0C	LPWSTR	lpwFileName	Pointer to full-path to original file
10	LPWSTR	lpwTemporaryFileName	Pointer to full-path to temp file in sandbox dir
14	LPVOID	lpSameBrokerApp	Pointer to <i>SameBrokerApp</i> object
18	UINT32	ui32SessionEnableHyperlinks	Used in Tag 0x091000 (TRUE or FALSE)



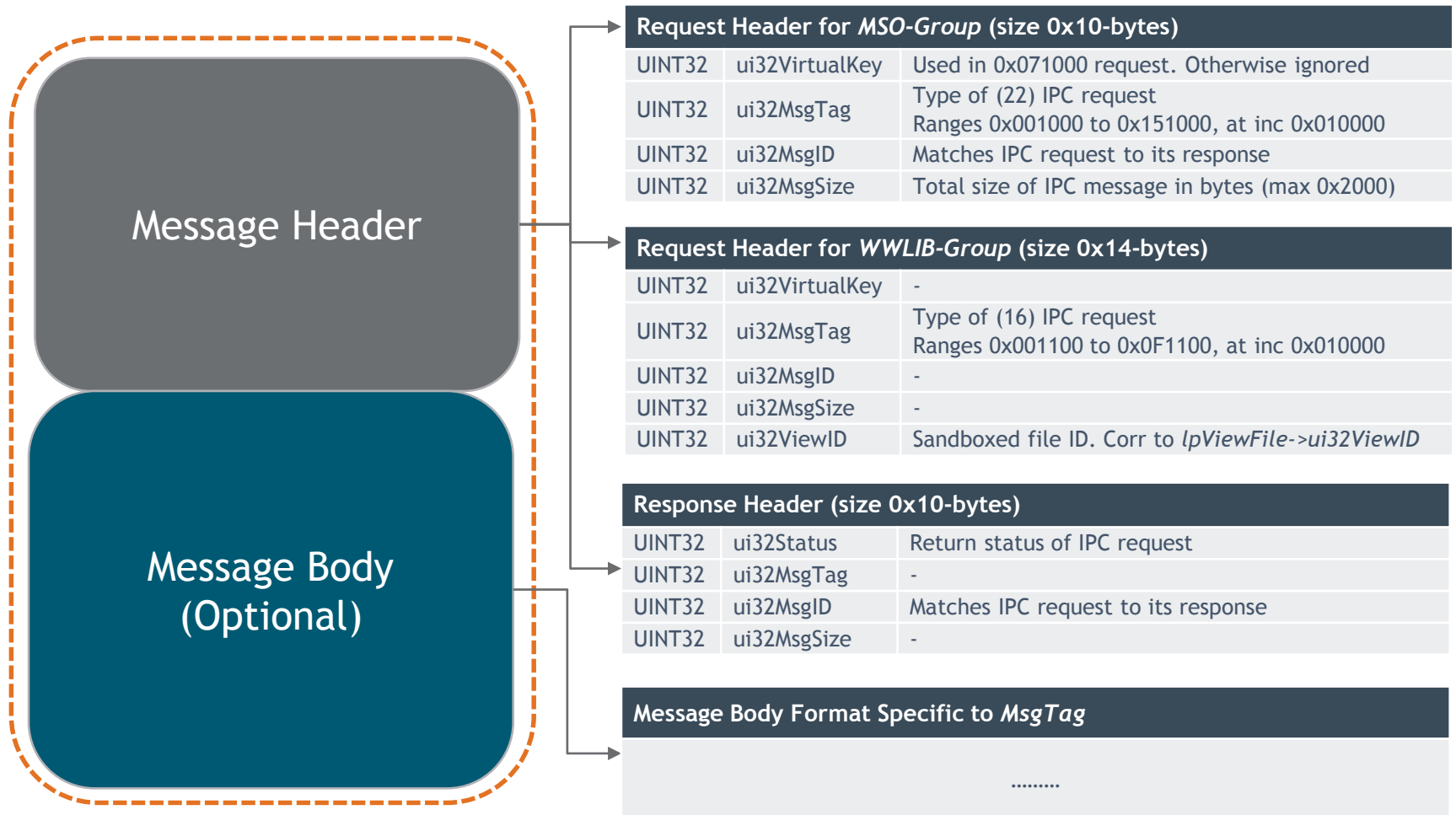
SameBrokerApp Object			
Offset	Size	Field	Comment
0C	LPVOID	lpWWLIBIPCMsg	Pointer to <i>WWLIBIPCMsg</i> object
84	HWND	hOPHParentWnd	Used in 0x061000 and 0x101000 IPC message
90	LPVOID	lpDRMStream	Used in 0x081000 IPC message
B0	LPVOID	lpTaskList	Used in 0x0B1000 IPC message

MSO-Group of IPC Messages

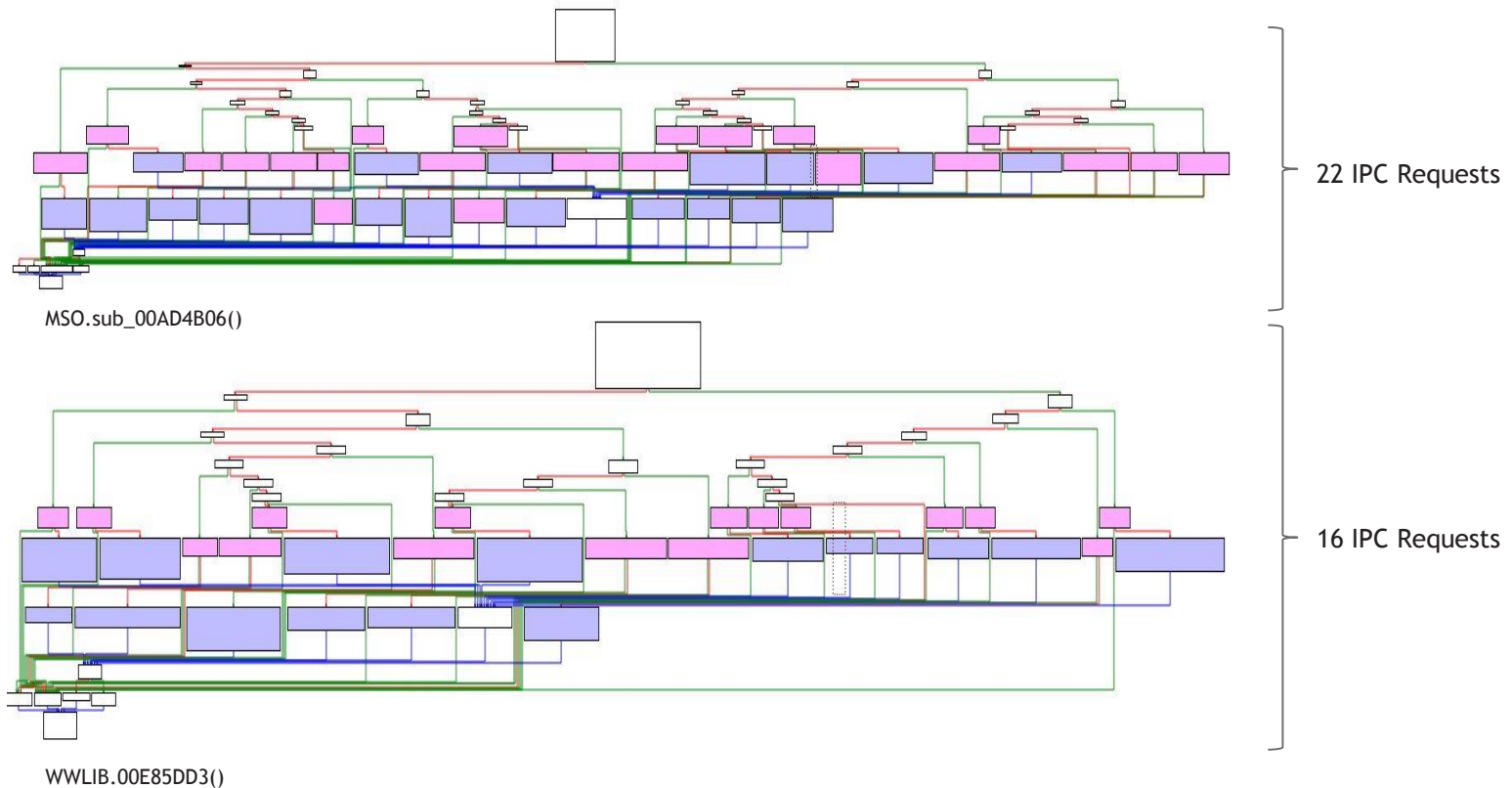
WWLIBIPCMsg Object			
Offset	Size	Field	Comment
1C	UINT32	ui32IPCOA1100	Used in 0x0A1100 IPC message
20	UCHAR [0x2C]	uchIPC091100Contents	Buffer storing IPC 0x091100 message contents
4C	UINT32	ui32IPC071100MsgID	MsgID of IPC 0x071100 message
50	UINT32	ui32IPC081100MsgID	MsgID of IPC 0x081100 message
54	UINT32	ui32IPC091100MsgID	MsgID of IPC 0x091100 message
58	UINT32	ui32IPC031100MsgID	MsgID of IPC 0x031100 message
5C	UINT32	ui32IPC041100MsgID	MsgID of IPC 0x041100 message
60	UINT32	ui32IPC0E1100MsgID	MsgID of IPC 0x0E1100 message
64	UCHAR [0x24]	uchIPC041100Contents	Buffer storing IPC 0x041100 message contents
8C	UCHAR [0x1D4]	uchIPC031100Contents	Buffer storing IPC 0x031100 message contents

WWLIB-Group of IPC Messages

## IPC: MESSAGE FORMAT



## IPC: MESSAGE FORMAT



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

**Purple:** Service functions

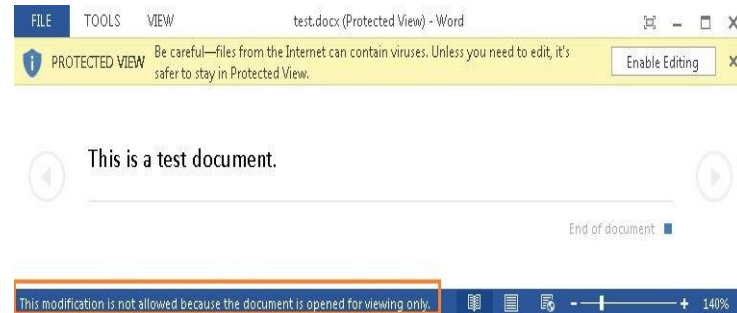


## IPC: MESSAGE 0X061000

### Message Body for 0x061000 Request

Size	Field	Comment
UINT32	ui32ViewID	-
DWORD	dwMsg	Any of WM_KEYDOWN, WM_KEYUP, WM_CHAR, WM_SYSKEYDOWN, WM_SYSKEYUP or WM_SYSCHAR
WPARAM	wParam	-
LPARAM	lParam	-

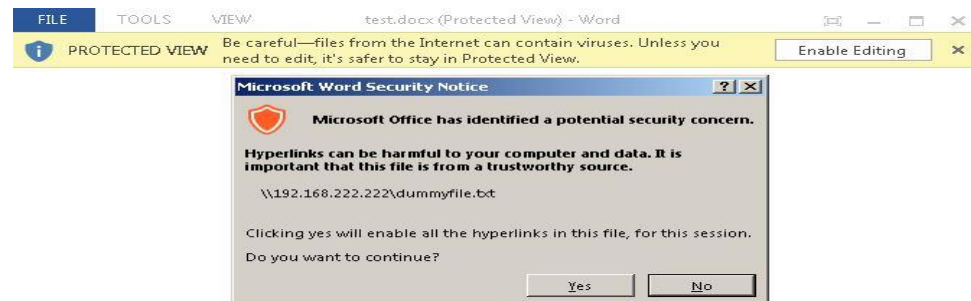
- PostMessageW() request to “Microsoft Word Document” window
- Cannot disable Protected-View:
  - Sandbox window in focus
  - Limited set of *dwMsg*



## IPC: MESSAGE 0X091000

Message Body for 0x091000 Request		
Size	Field	Comment
UINT32	ui32ViewID	-
LPWSTR	pwzTarget	Parameter in <i>HlinkNavigateToStringReference()</i>
LPWSTR	pwzLocation	Parameter in <i>HlinkNavigateToStringReference()</i>
DWORD	grfHLNF	Parameter in <i>HlinkNavigateToStringReference()</i>
UINT32	ui32Unknown	-
HWND	hWndParent	<i>CreateWindowExW()</i> , for user-permission prompt window
WCHAR[]	wzTarget	Parameter in <i>HlinkNavigateToStringReference()</i>
WCHAR[]	wzLocation	Parameter in <i>HlinkNavigateToStringReference()</i>

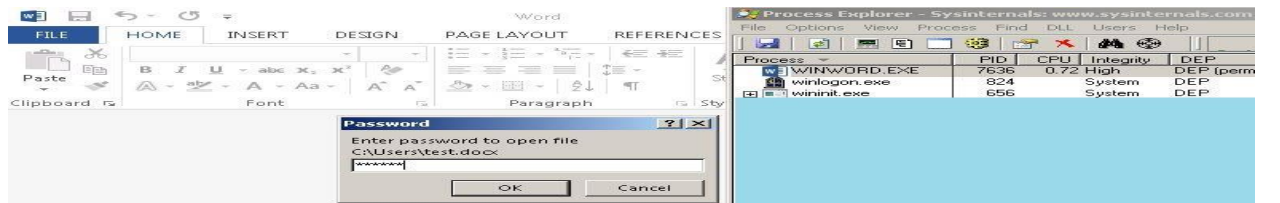
- *HlinkNavigateToStringReference()* request to visit URL
- User permission
  - *lpViewFile->ui32SessionEnableHyperlinks*
  - Per file per session



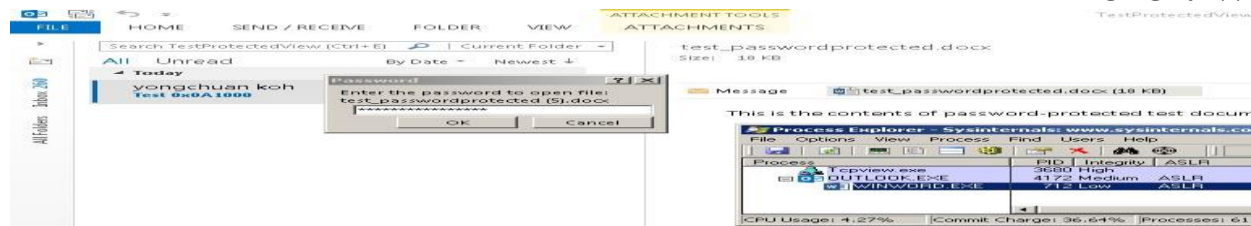
## IPC: MESSAGE 0X0A1000

- “Enter Password” prompt for password-protected file

Message Body for 0x0A1000 Request		
Size	Field	Comment
UINT32	ui32ViewID	-
Message Body for 0x0A1000 Response		
LPWSTR	lpwPassword	Pointer to wzPassword
UINT16	ui16PasswordSize	Wide-char length of wzPassword
WCHAR[]	wzPassword	-



Broker: WINWORD



Broker: OUTLOOK



## 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

Message Body for 0x0C1000 Request		
Size	Field	Comment
HANDLE	hSandboxSharedMem	-
HANDLE	hEventBrokerIsDone	Broker sets this event after DW20.EXE exits and before it terminates sandbox
LPWSTR	lpwAdditionalWerFileName	Pointer to <i>wzAdditionalWerFileName</i> , or Null
UINT16	ui16AdditionalWerFileNameSize	-
WCHAR[]	wzAdditionalWerFileName	An additional file to be submitted to WER server, to be created by Protected-View sandbox

## IPC: MESSAGE 0X0C1000

### Possible Format of *hBrokerShareMem* Shared Memory

Size	Field	Comment
UINT32	0x00009C9C	Size of <i>hBrokerSharedMem</i> memory
UINT32	ui32ProtectedViewPID	Sandbox Process-ID
UINT32	ui32ProtectedViewTID	TID of faulting thread in sandbox. Copied from <i>hSandboxSharedMem</i> offset 0x0C.
UINT32	uiProtectedViewEIP	EIP of faulting instruction in sandbox. Copied from <i>hSandboxSharedMem</i> offset 0x10.
LPVOID	lpProtectedViewPEP	Exception pointers in sandbox. Copied from <i>hSandboxSharedMem</i> offset 0x14.
		.....
HANDLE	hSandboxProcess	Sandbox process handle
		.....
WCHAR[0x104]	wzModulesList	List of modules loaded in sandbox, separated by Null
WCHAR[0x400]	wzWerSubmitFilesList	List of files to submit to WER with <i>CWatsonReport::AddFilesToReport()</i> , separated by ' ': <ul style="list-style-type: none"> <li>• <b>Sandbox-directory</b> + <i>wzAdditionalWerFileName</i></li> <li>• %Temp% + "winword.exe.OsrHost.dmp.dat"</li> <li>• %Temp% + "winword.exe.OsrHost.cvr.dat"</li> </ul>
UINT32	ui32CrashParamFlag	If Null, next 11 fields are ignored. Copied from <i>hSandboxSharedMem</i> offset 0x8470
WCHAR[0xFF]	wzEventType	<i>pwzEventType</i> in <i>WerReportCreate()</i> Copied from <i>hSandboxSharedMem</i> offset 0x8474
WCHAR[0xFF]	wzParam0	<i>pwzValue</i> for WER_P0 in <i>WerReportSetParameter()</i> Copied from <i>hSandboxSharedMem</i> offset 0x8672
		.....
WCHAR[0xFF]	wzParam9	<i>pwzValue</i> for WER_P9 in <i>WerReportSetParameter()</i> Copied from <i>hSandboxSharedMem</i> offset 0x9860
		.....

# IPC: MESSAGE 0X0C1000

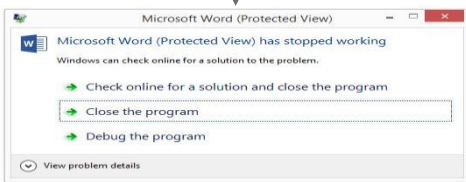
## Windows Error Reporting

WINWORD.EXE	4892	0.08	Medium
WINWORD.EXE	3592	< 0.01	AppContainer
DW20.EXE	5740		Medium
DWWIN.EXE	5612	0.02	Medium

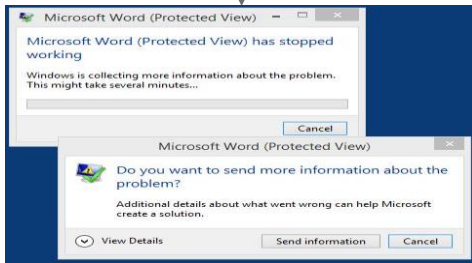
```

<?xml version="1.0" encoding="UTF-16"?>
<WERREPORT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <MACHINEINFO machineName="WIN-NUNVQ4E223" os="6.3.9600.2.0.0.256-48" locid="04090404" />
  <USERINFO userName="NUNVQ4E223" />
  <APPLICATIONINFO appName="Microsoft Word (Protected View)" appPath="C:\Program Files\Microsoft Office\Office15\WINWORD.EXE" />
  <EVENTINFO reportType="2" eventTime="130772894289500" eventType="APPLICATION_FAULT_EXCEPTION">
    <SIGNATURE>
      <PARAMETER id="0" name="Application Name" value="WINWORD.EXE"/>
      <PARAMETER id="1" name="Application Version" value="15.0.4569.1504"/>
      <PARAMETER id="2" name="Application Timestamp" value="526960d"/>
      <PARAMETER id="3" name="Fault Module Name" value="unknown"/>
      <PARAMETER id="4" name="Fault Module Version" value="0.0.0.0"/>
      <PARAMETER id="5" name="Fault Module Timestamp" value="00000000"/>
      <PARAMETER id="6" name="Exception Code" value="00000000"/>
      <PARAMETER id="7" name="Exception Offset" value="00000000"/>
      <SECONDARYPARAMETER name="LCID" value="1033"/>
      <SECONDARYPARAMETER name="skulcid" value="1033"/>
    </SIGNATURE>
    <FILES>
      <FILE filetype="4" filename="crash.doc"/>
      <FILE filetype="5" filename="WERInternalMetadata.xml"/>
    </FILES>
  </EVENTINFO>
</WERREPORT>
  
```

Level 1 Error Data



"I want more information"

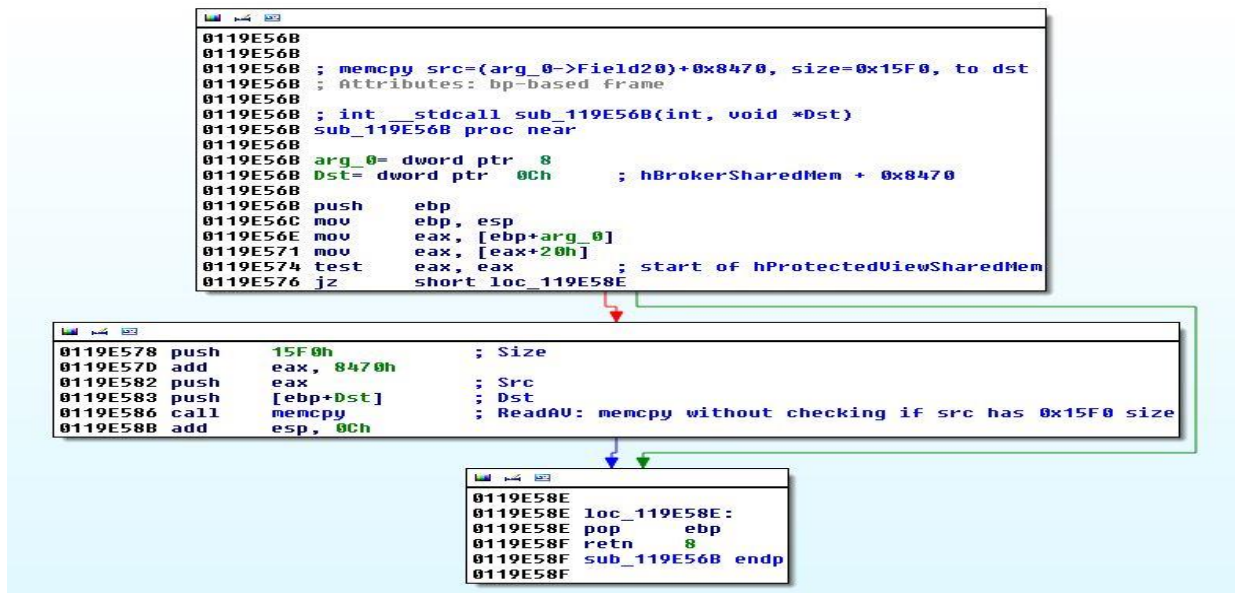


Level 2 Error Data



## 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

```

0119DCBB push  '\ '
0119DCBD pop  ecx
0119DCBE mov  [edi+eax*2], cx ; append '\ ' to local_dst
0119DCD2 inc  eax ; copied_size++
0119DCD3 sub  ebx, eax ; ebx = remaining dst buffer size
0119DCD5 push ebx ; SizeInWords
0119DCD6 lea  esi, [edi+eax*2]
0119DCD9 push esi ; Dst: start of (empty) local_dst buffer
0119DCDA push [ebp+wzAdditionalWerFileName] ; Src
0119DCDB call _wcsncpy_s ; Wrapper for wcsncpy_s(Dst,SizeInWords,Src,_TRUNCATE)
0119DCDD ; Returns pointer to end of Dst (including NULL)
0119DCDD mov  ebx, eax
0119DCD2 push '\ ' ; Ch
0119DCD6 sub  ebx, edi ; ebx = total copied size now, including NULL
0119DCD8 push esi ; Str: wzAdditionalWerFileName copied to local_dst[0x104]
0119DCD9 sar  ebx, 1
0119DCDB call ds:_imp_wcschr ; check if there is '\ ' in filename (ie: src, without fullpath)
0119DCE1 pop  ecx
0119DCE2 pop  ecx
0119DCE3 pop  esi
0119DCE4 test eax, eax
0119DCE6 jz   short loc_119DCF9

0119DCE8 backslash_found:
0119DCE8 push 63673945h
0119DCE9 call 0119DCE0 ; Original6125
0119DCF2 xor  eax, eax
0119DCF4 mov  [edi], ax ; local_dst[0x104]
0119DCF7 xor  ebx, ebx

0119DCF9 loc_119DCF9:
0119DCF9 push edi
0119DCF9 call PUIpc_CheckIfFullPathIsDirectory ; Returns 1 if fullpath filename (arg_0) is NOT directory
0119DCF9 test  eax, eax
0119DD01 jnz  short loc_119DD14
  
```



## 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

Message Body for 0x0F1000 Request		
Size	Field	Comment
UINT32	ui32ViewID	-
UINT32	ui32MessageID	0x465 or 0x469
LPWSTR	lpwData	Pointer to <i>wzData</i> , or Null
LPWSTR	lpwHelpID	Pointer to <i>wzHelpID</i> , or Null
UINT16	ui16DataSize	-
WCHAR[0x104]	wzData	“DEV”, “SHAPESHEET”, or others
UINT16	ui16HelpIDSize	-
WCHAR[0x104]	wzHelpID	“NoHelp”, or others

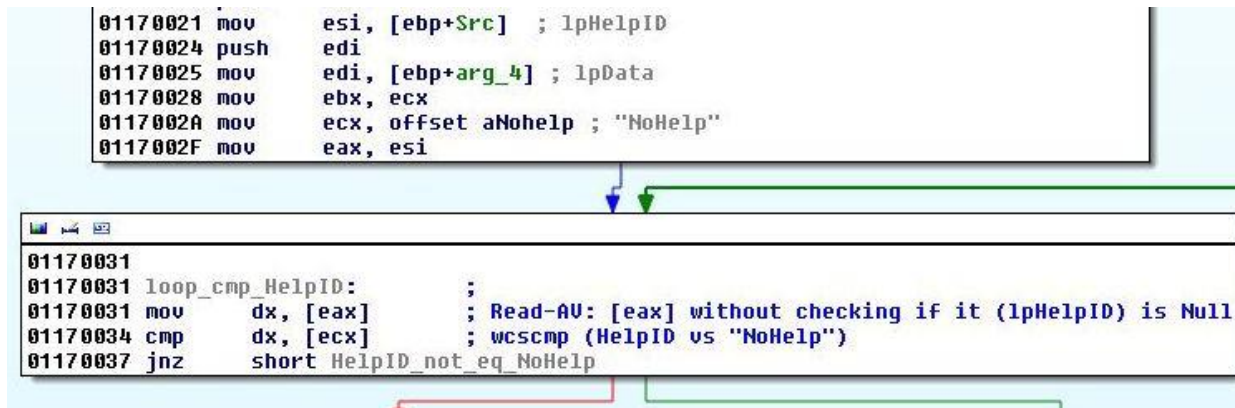
## IPC: MESSAGE 0X0F1000

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

```
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
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

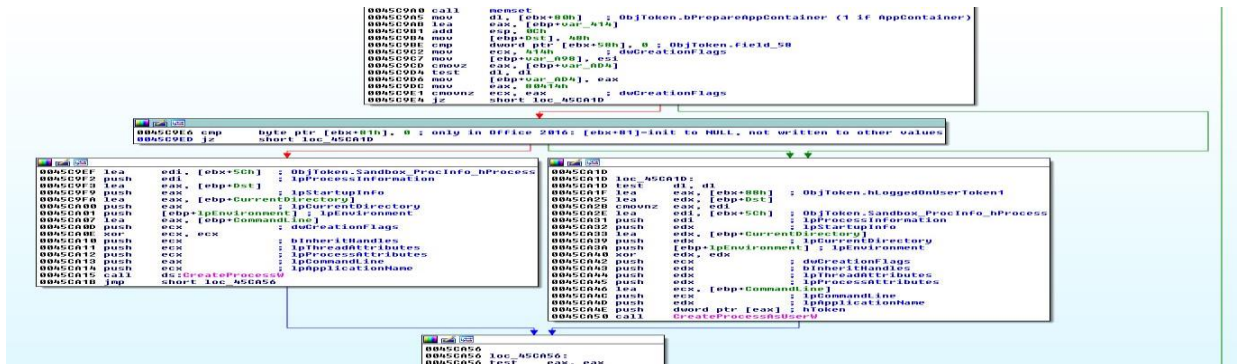


## MS OFFICE 2016: SANDBOX INTERNALS

- 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])



```

00401000 call 00401000 ; memset
00401001 mov  di, [ebx+80h] ; ObjToken.bPrepareAppContainer (1 if AppContainer)
00401002 lea  eax, [ebp+var_414]
00401003 add  esp, 0C0h
00401004 mov  [ebp+var_414], 50h
00401005 mov  ecx, 810h ; ObjToken.Field_50
00401006 mov  [ebp+var_408], 8 ; dCreationFlags
00401007 mov  ecx, [ebp+var_404], esi
00401008 test  di, di
00401009 mov  [ebp+var_404], eax
0040100A mov  ecx, esi
0040100B cmovz short inc_40C0E
0040100C jmp  0040100E

0040100E mov  byte ptr [ebx+81h], 0 ; only in Office 2016: [ebx+81]-init to NULL, not written to other values
0040100F jmp  loc_40C0E

00401010 lea  edi, [ebx+5Ch] ; ObjToken.Sandbox_ProcInfo_hProcess
00401011 push edi
00401012 lea  edi, [ebp+var_404] ; IpProcessInformation
00401013 push edi
00401014 lea  edi, [ebp+CurrentDirectory] ; IpStartUpInfo
00401015 push edi
00401016 lea  edi, [ebp+Environment] ; IpCurrentDirectory
00401017 push edi
00401018 lea  edi, [ebp+Command_Line] ; dCreationFlags
00401019 push edi
0040101A mov  ecx, esi
0040101B push ecx ; OtherAttributes
0040101C push ecx ; IpThreadAttributes
0040101D push ecx ; IpProcessAttributes
0040101E push ecx ; IpCommand_Line
0040101F push ecx ; IpApplicationName
00401020 call 00401020 ; CreateProcessW
00401021 jmp  short loc_40C0E

00401022 loc_40C0D:
00401023 inc  dword ptr [ebx+81h] ; ObjToken.loggedOnUserToken
00401024 test  di, di
00401025 mov  ecx, [ebp+var_404] ; ObjToken.Sandbox_ProcInfo_hProcess
00401026 cmovz  edi, esi ; IpProcessInformation
00401027 push  edi ; IpStartUpInfo
00401028 lea  edi, [ebp+CurrentDirectory] ; IpCurrentDirectory
00401029 push  edi ; IpThreadAttributes
0040102A push  edi ; IpProcessAttributes
0040102B push  edi ; IpCommand_Line
0040102C lea  edi, [ebp+Command_Line]
0040102D push  edi ; IpApplicationName
0040102E push  dword ptr [eax] ; IpToken
0040102F call 0040102F ; CreateProcessAsUserW

00401030 loc_40C0E:
00401031 loc_40C0E:
00401032 test  eax, eax
    
```

## 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

```

0119DCBB push '\
0119DCBD pop ecx
0119DCBE mov [edi+eax*2], cx ; append '\' to local_dst
0119DC22 inc eax ; copied_size++
0119DC23 sub ebx, eax ; ebx = remaining dst buffer size
0119DC25 push ebx ; SizeInWords
0119DC26 lea esi, [edi+eax*2]
0119DC29 push esi ; Dst: start of (empty) local_dst buffer
0119DC2A push [ebp+wzAdditionalVerFileName] : Src
0119DC2C call _wcsncpy_s ; Wrapper for wcsncpy_s(Dst,SizeInWords,Src,_TRUNCATE)
0119DC2D ; Returns pointer to end of Dst (including NULL)
0119DC2E mov ebx, eax
0119DC2F push '\
0119DC30 sub ebx, edi ; ebx = total copied size now, including NULL
0119DC32 push esi ; Str: wzAdditionalVerFileName copied to local_dst[0x104]
0119DC34 sar ebx, 1
0119DC36 call ds:_imp_wcschr ; check if there is '\' in filename (ie: src, without fullpath)
0119DC38 pop ecx
0119DC39 pop ecx
0119DC3A pop esi
0119DC3B test eax, eax
0119DC3D jz short loc_119DCF9

```

```

0119DCE8 backslash_found:
0119DCE8 push 63673965h
0119DCE9 call Ordinal16125
0119DCEB xor eax, eax
0119DCEC mov [edi], ax ; local_dst[0x104]
0119DCEE xor ebx, ebx

```

```

0119DCF9 loc_119DCF9:
0119DCF9 push edi
0119DCF9 call PUIpc_CheckIfFullPathIsDirectory ; Returns 1 if Fullpath Filename (arg_0) is NOT directory
0119DCF9 test eax, eax
0119DD01 jnz short loc_119DD14

```

MS Office 2013

```

0086DBEC push '\
0086DBEE pop ecx
0086DBEF mov [ebx+eax*2], cx
0086DBF3 inc eax
0086DBF4 sub edi, eax
0086DBF6 push edi ; SizeInWords
0086DBF7 lea esi, [ebx+eax*2]
0086DBFA push esi ; Dst
0086DBFB push [ebp+Src] ; Src
0086DBFE call _wcsncpy_s
0086DC03 mov edi, eax
0086DC05 push '\
0086DC07 sub edi, ebx
0086DC09 push esi ; Str
0086DC0A sar edi, 1
0086DC0C call ds:wcschr
0086DC12 pop ecx
0086DC13 pop ecx
0086DC14 pop esi
0086DC15 test eax, eax
0086DC17 jz short loc_86DC2A

```

```

0086DC19 push 63673965h
0086DC1E call Mso20Win32Client_803
0086DC23 xor eax, eax
0086DC25 xor edi, edi
0086DC27 mov [ebx], ax

```

```

0086DC2A loc_86DC2A: ; lpFileName
0086DC2A push ebx
0086DC2B call PUIpc_CheckIfFullPathIsDirectory
0086DC30 test eax, eax
0086DC32 jnz short loc_86DC45

```

MS Office 2016

## MS OFFICE 2016: IPC 0X161000

---

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

Message Body for 0x161000 Request		
Size	Field	Comment
UINT32	ui32ViewID	-
UINT32	hUnprotectWnd	Window handle to unprotect for.
UINT32	ui32Boolean	If 0, unprotects window with <i>IpcUnprotectWindow()</i> or <i>_IpcUnprotectWindowNoDRM()</i> . If 1, protects window with <i>IpcProtectWindow()</i> or <i>_IpcProtectWindowNoDRM()</i>



## 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)...

## REFERENCES

---

- Yason, Mark Vincent. "**DIVING INTO IE 10'S ENHANCED PROTECTED MODE SANDBOX**". [Online] <https://www.blackhat.com/docs/asia-14/materials/Yason/WP-Asia-14-Yason-Diving-Into-IE10s-Enhanced-Protected-Mode-Sandbox.pdf>.
- Forshaw, James. "**IE11 Sandbox Escapes**". [Online] <https://github.com/tyranid/IE11SandboxEscapes>.
- Keetch, Tom. "**Practical Sandboxing on the Windows Platform**". [Online] <https://www.hackinparis.com/slides/hip2k11/12-EscapingWindowsSandboxes.pdf>.
- MSDN. "**How to: Configure Microsoft Error Reporting**". [Online] Microsoft Corporation. [https://msdn.microsoft.com/en-us/library/office/bb219076\(v=office.12\).aspx](https://msdn.microsoft.com/en-us/library/office/bb219076(v=office.12).aspx).
- MSDN. "**Naming Files, Paths, and Namespaces**". [Online] [https://msdn.microsoft.com/en-us/library/windows/desktop/aa365247\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa365247(v=vs.85).aspx).
- Vreugdenhil, Peter. "**ADOBE SANDBOX: WHEN THE BROKER IS BROKEN**". [Online] <https://cansecwest.com/slides/2013/Adobe%20Sandbox.pdf>.
- Ionescu, Alex. "**New Security Assertions in Windows 8**". [Online] <http://www.alex-ionescu.com/?p=69>.
- .....



Thank You!

Questions?