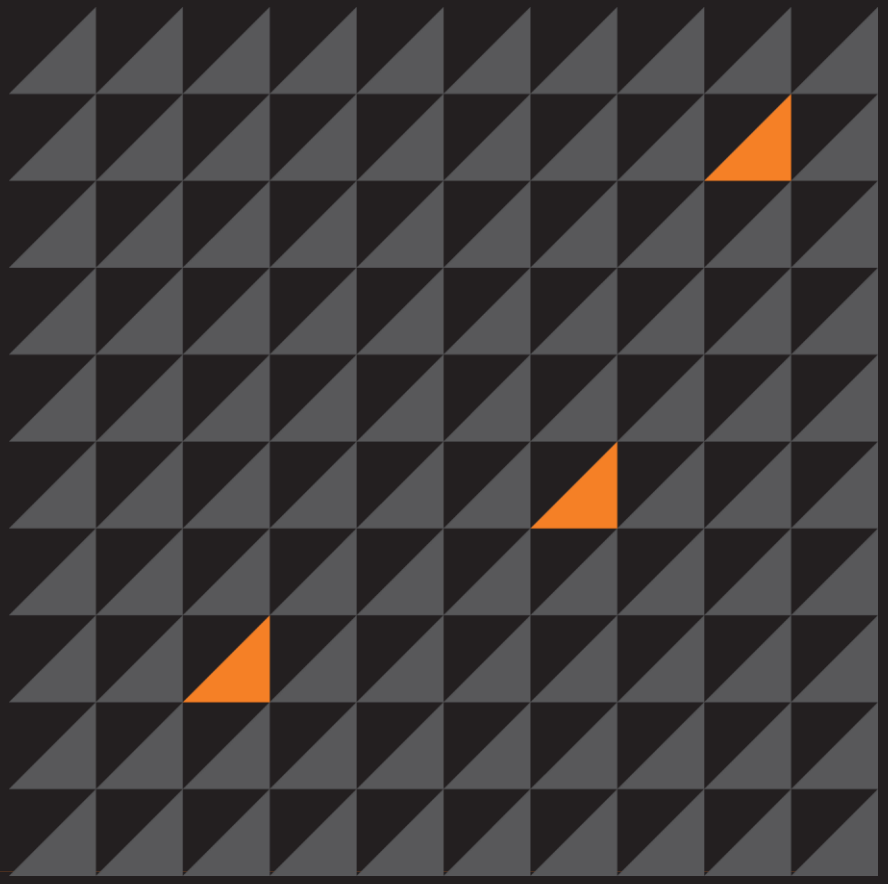
UNDERSTANDING THE MICROSOFT OFFICE 2013 PROTECTED-VIEW SANDBOX

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

MWR LABS

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

MWR LABS

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

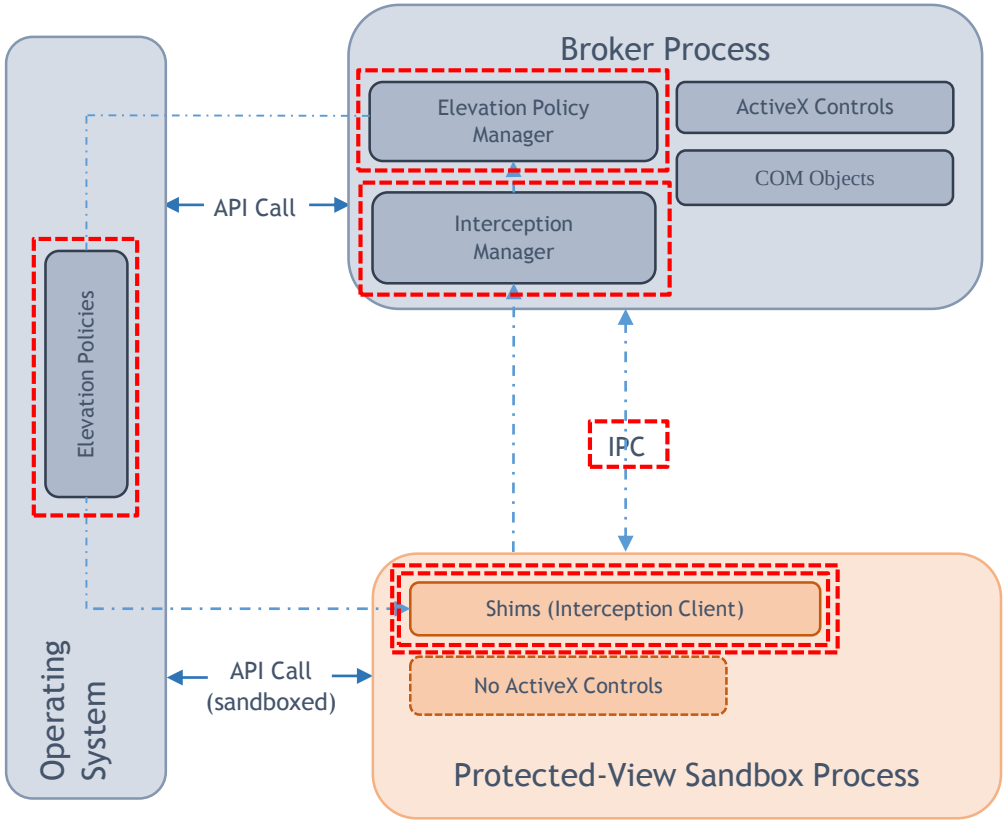
- Architecture
- Initialization
- Restrictions

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

Brower 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

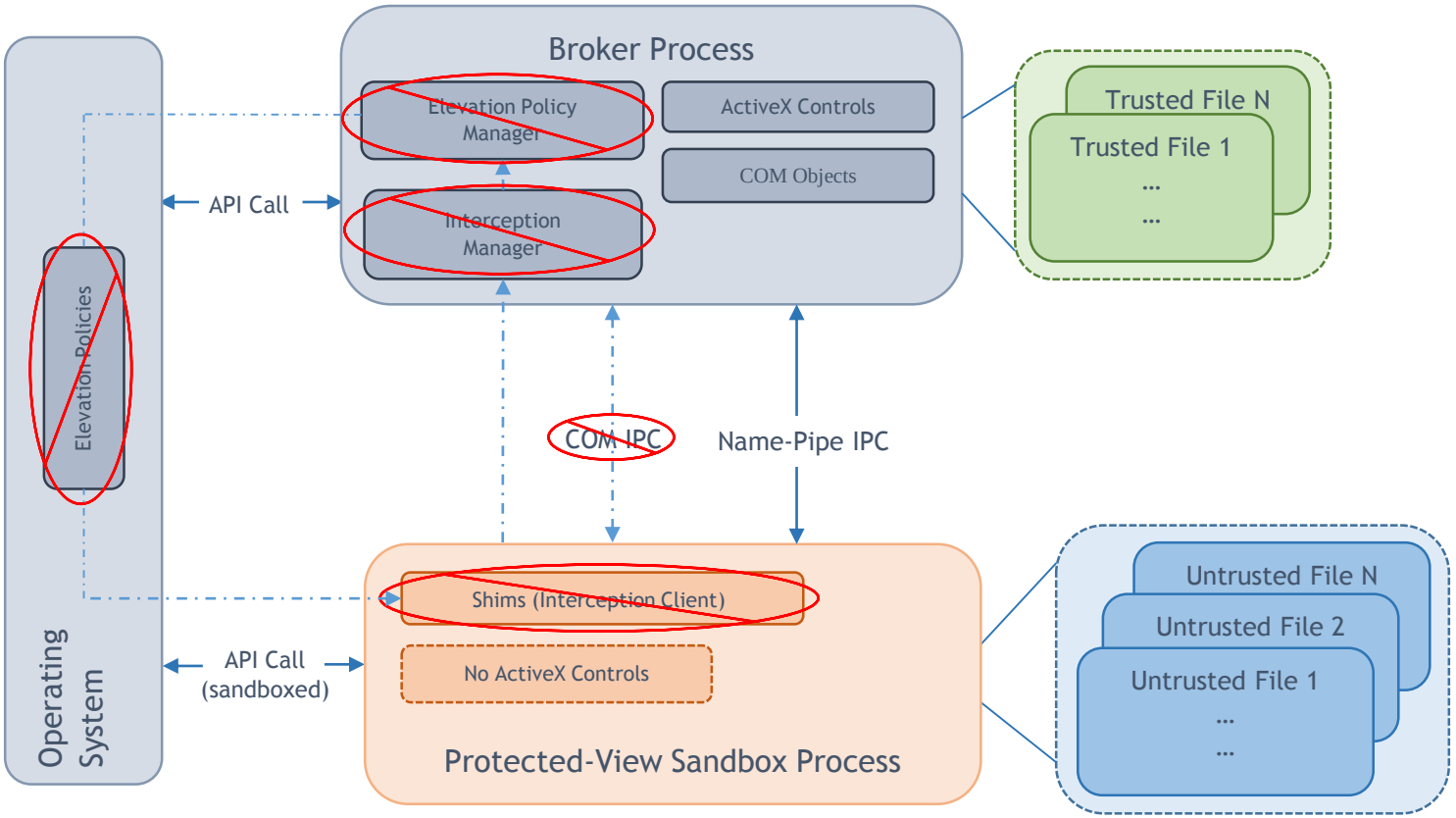
- Elevation Policy Component
  - In IE, elevation Policies are stored as registry keys
    - <AppName>|<AppPath>|<CLSID>|<LaunchPolicyValue> format
  - Checks for new registry keys with this format
    - MS Office 2007 vs MS Office 2013
  - Elevation Policy component not present in Protected-View

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…

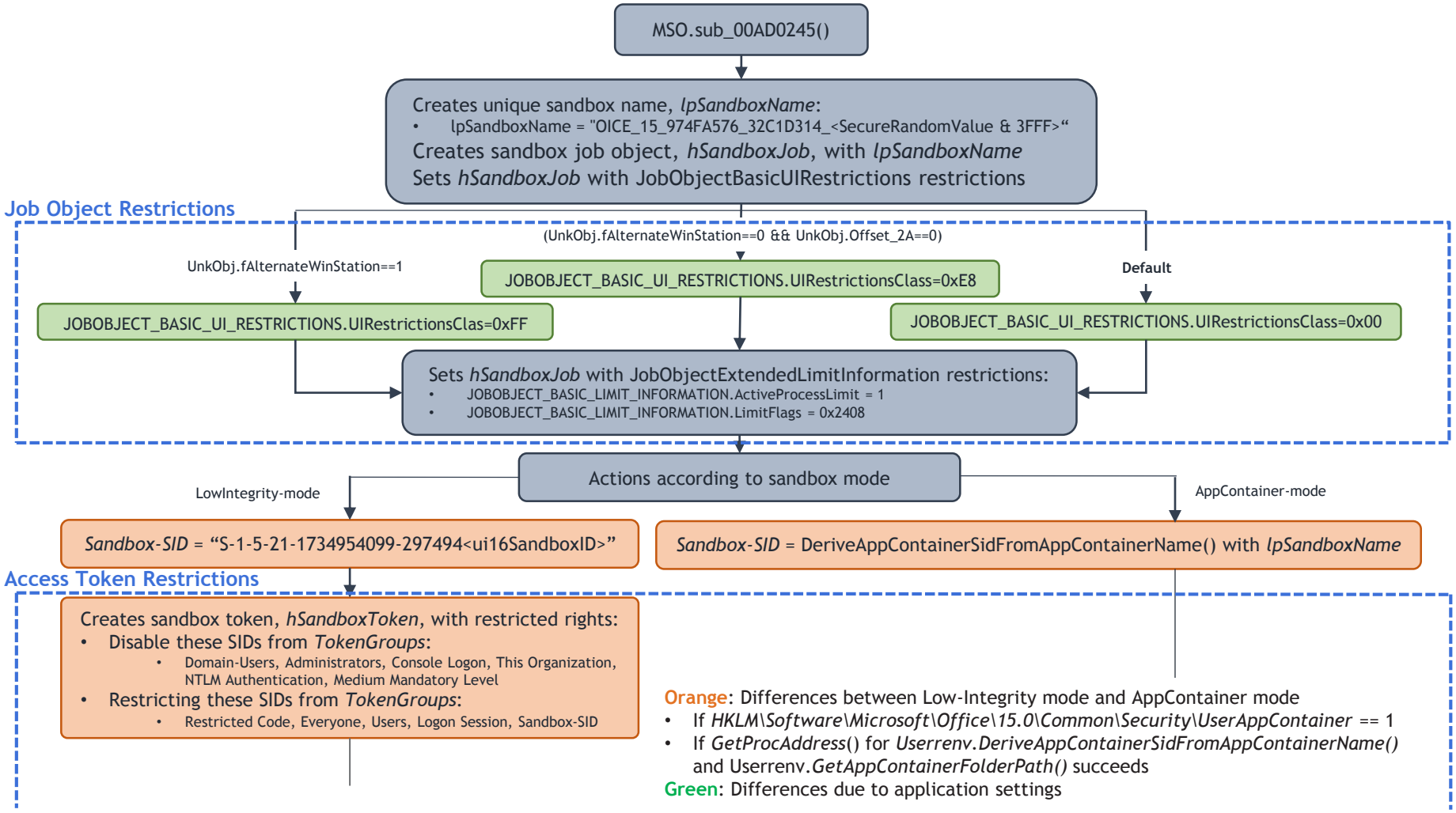
INTERNALS: ARCHITECTURE



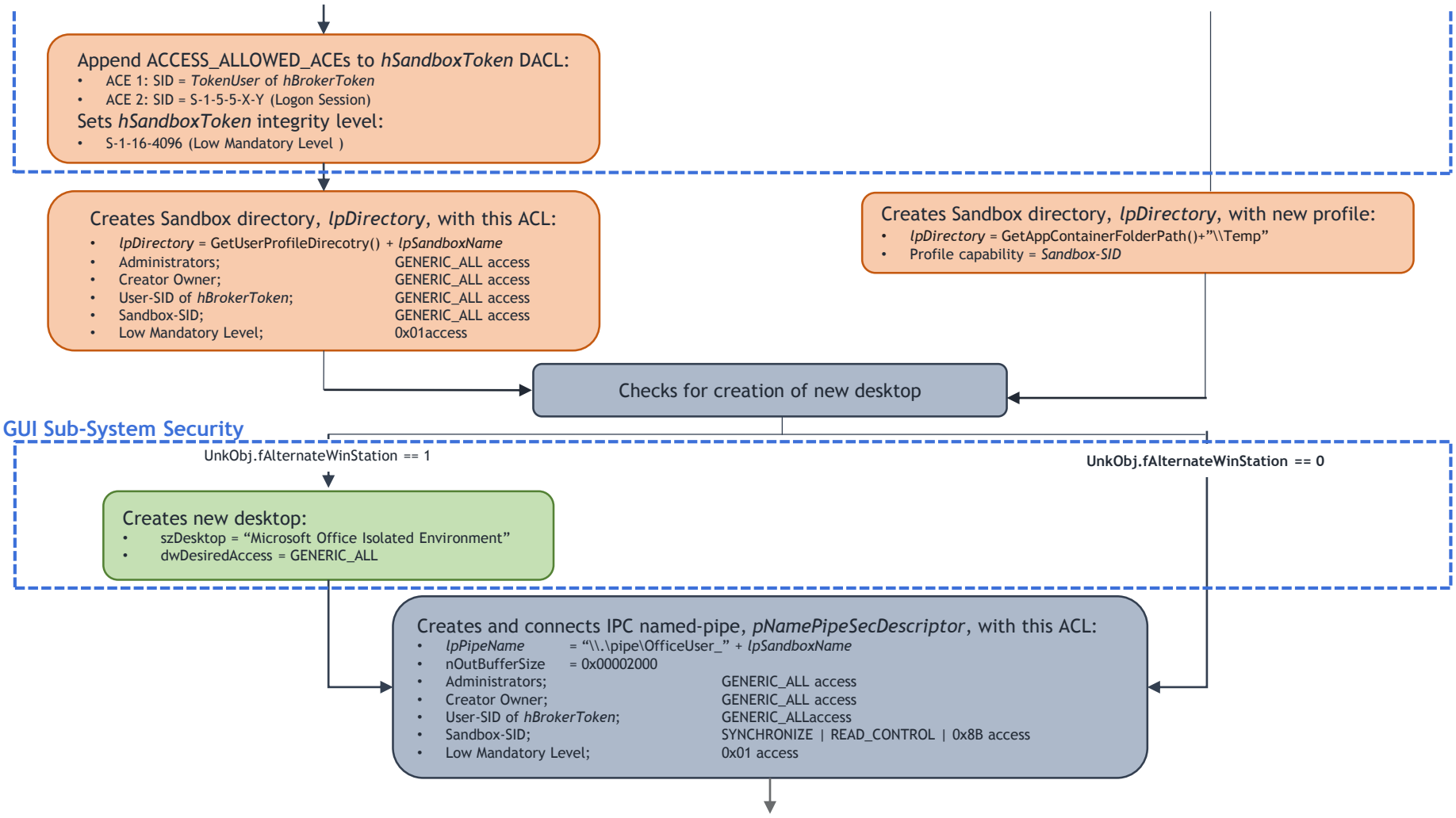
INTERNALS: INITIALIZATION

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

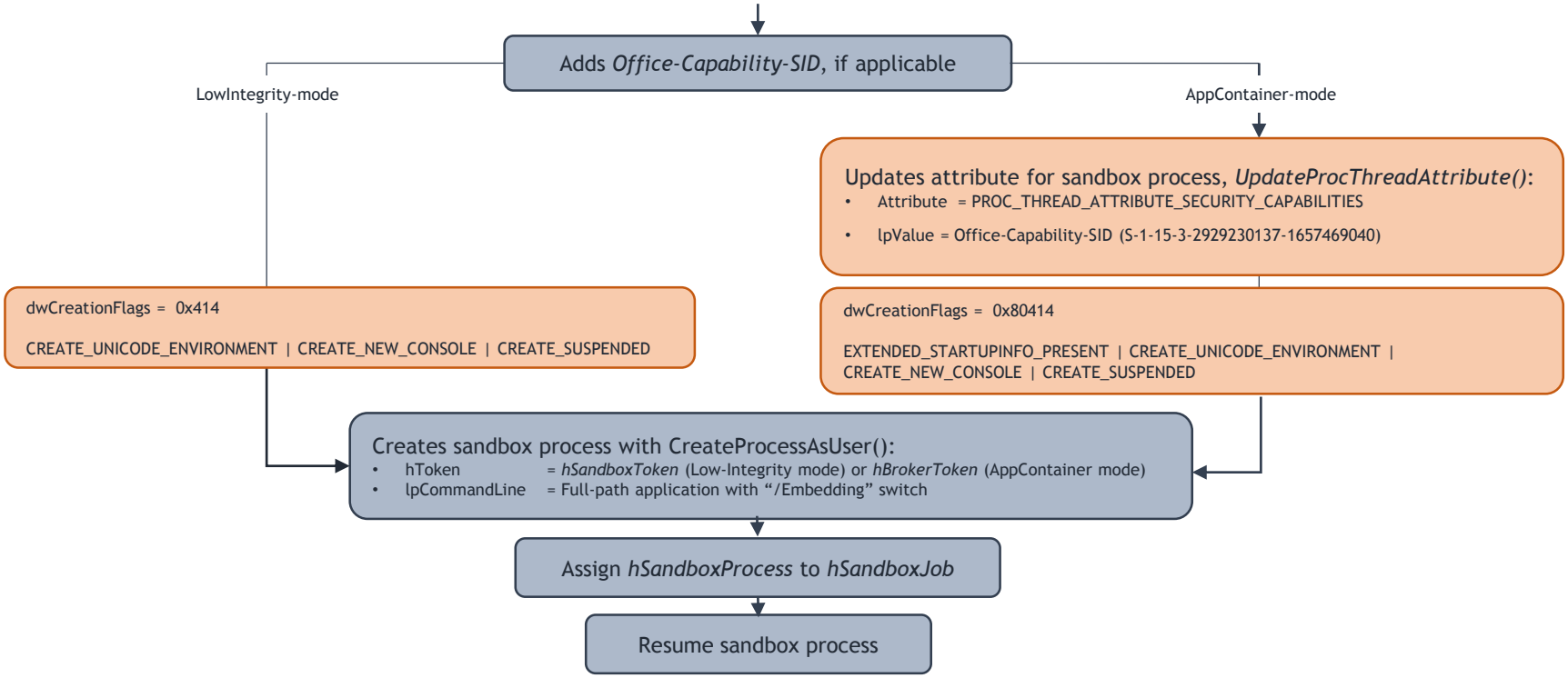
INTERNALS: INITIALIZATION



INTERNALS: INITIALIZATION



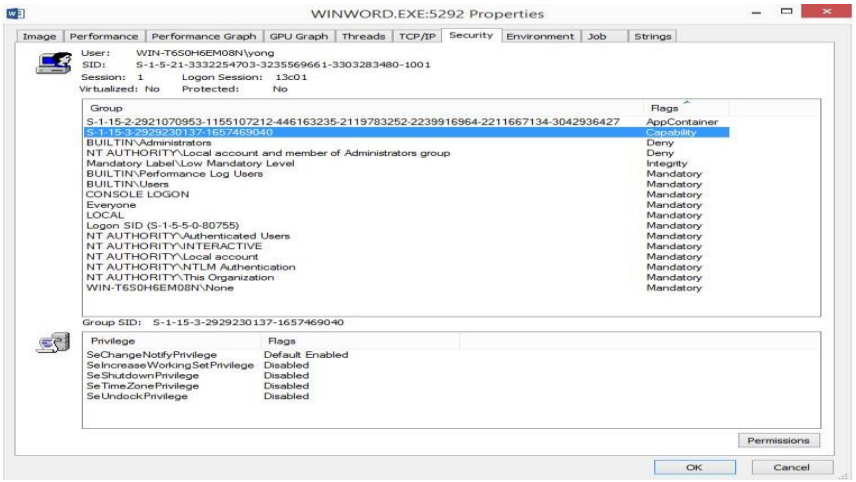
INTERNALS: INITIALIZATION



- 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
Sandbox-SID (*S-1-15-2-*-*-*-*-*-*-)**	
%UserProfile%\AppData\Local\Packages\<lpSandboxName>*	STANDARD_RIGHTS_ALL | 0x1FF
Office-Capability-SID (*S-1-15-3-2929230137-1657469040*)	
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

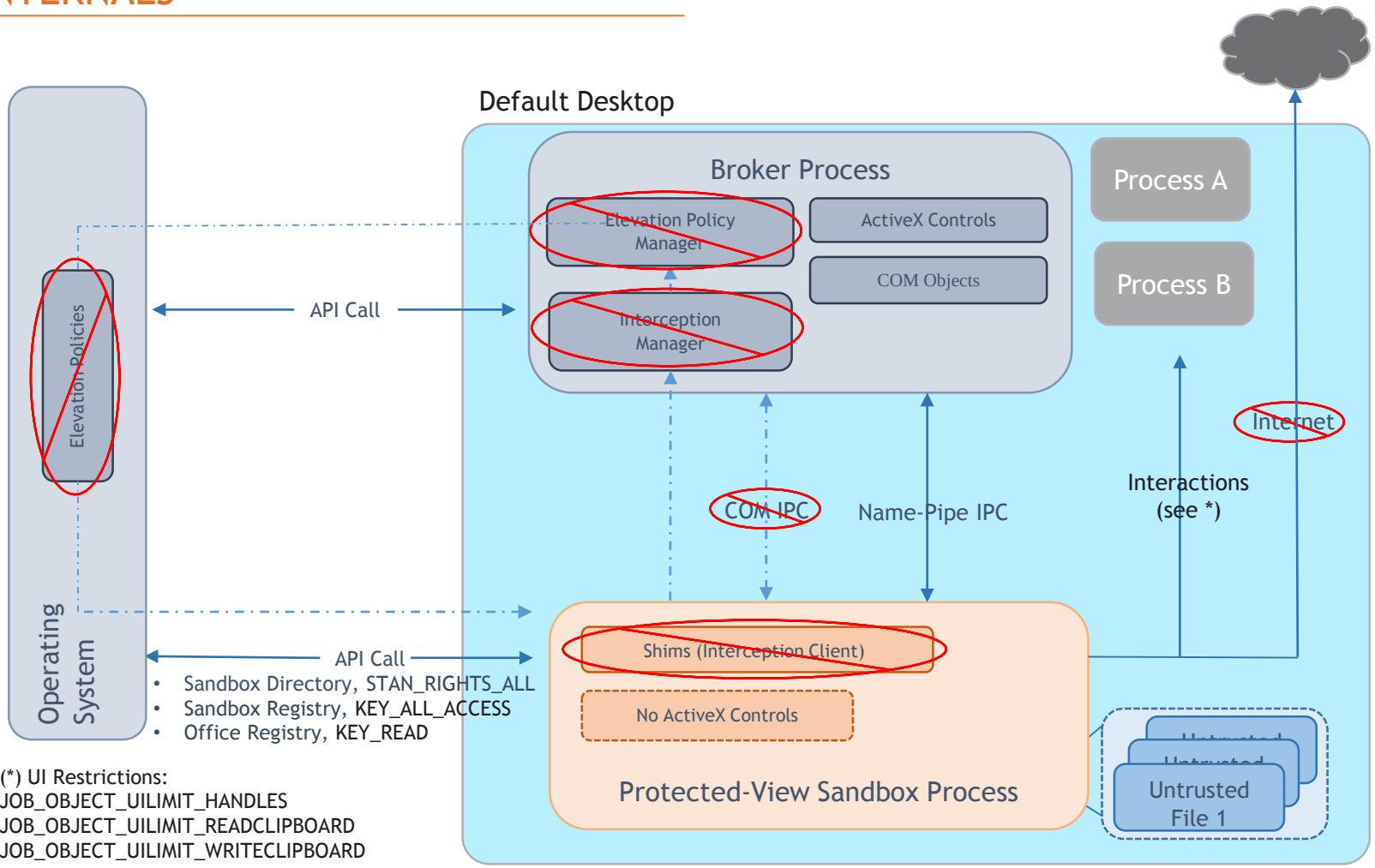
Registry Keys	Access Mask
Sandbox-SID (*S-1-15-2-*-*-*-*-*-*-)**	
HKCR\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\<*Sandbox-SID*>	KEY_READ
HKCR \Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Mappings\<*Sandbox-SID*>\Children	KEY_ALL_ACCESS
HKCR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



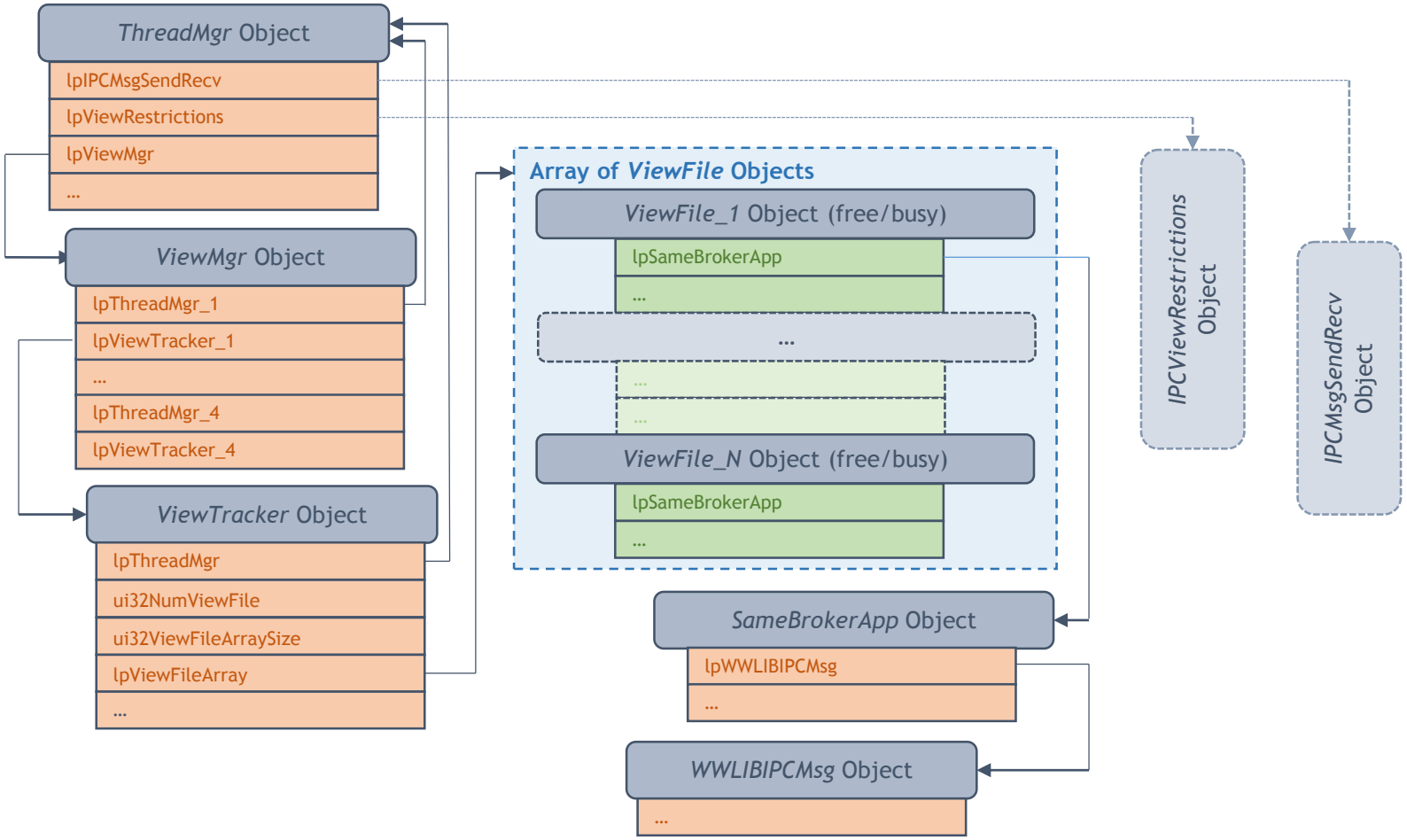
(*) UI Restrictions:
JOB_OBJECT_UILIMIT_HANDLES
JOB_OBJECT_UILIMIT_READCLIPBOARD
JOB_OBJECT_UILIMIT_WRITECLIPBOARD
…
(https://msdn.microsoft.com/en-us/library/windows/desktop/ms684152(v=vs.85).aspx)

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

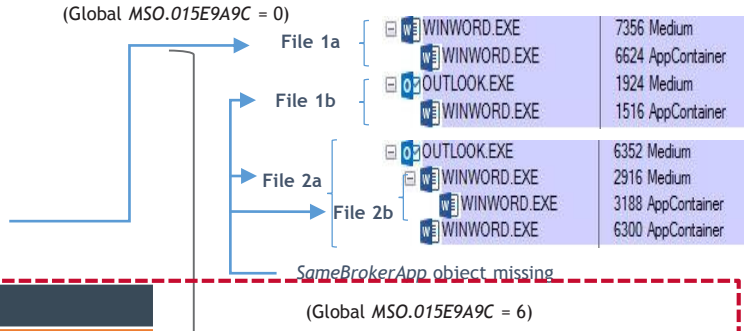
ThreadMgr Object			
Offset	**Size**	**Field**	**Comment**
08	LPVOID	lpIPCMsgSendRecv	Pointer to an object that sends/receives IPC messages
0C	LPVOID	lpViewRestrictions	Pointer to an object describing the sandbox restrictions
10	LPVOID	lpViewMgr	Pointer to *ViewMgr* object

ViewMgr Object			
Offset	**Size**	**Field**	**Comment**
04	UINT32	ui32Num	Number of untrusted files + 2
08	LPVOID	lpThreadMgr_1	-
0C	LPVOID	lpViewTacker_1	Pointer to *ViewTracker* object

ViewTracker Object			
Offset	**Size**	**Field**	**Comment**
0C	LPVOID	lpThreadMgr	Pointer to *ThreadMgr* object
14	UINT32	ui32NumViewFile	Number of Protected-View files, or number of busy slots in *ViewFilesArray*
18	UINT32	ui32ViewFileArraySize	Size of *ViewFilesArray*
1C	UINT32	ui32Unknown	Higher 2 bytes are used to calculate new *ui32ViewFilesArraySize* when *ViewFilesArray* is full
20	LPVOID	lpViewFileArray	Pointer to an array of *ViewFile* 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 *SameBrokerApp* object
18	UINT32	ui32SessionEnableHyperlinks	Used in Tag 0x091000 (TRUE or FALSE)



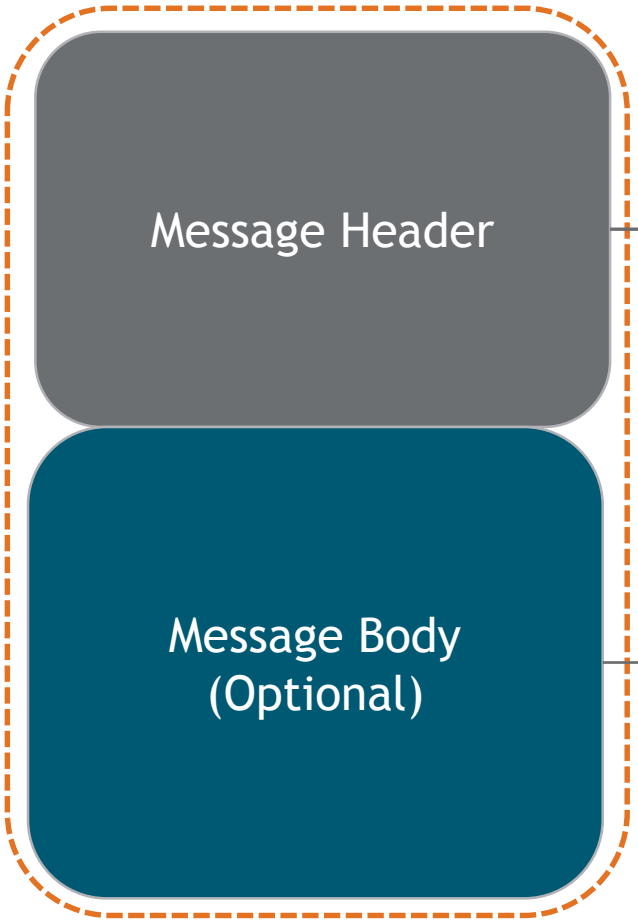
SameBrokerApp Object			
Offset	**Size**	**Field**	**Comment**
0C	LPVOID	lpWWLIBIPCMsg	Pointer to *WWLIBIPCMsg* object
84	HWND	hOPHParentWnd	Used in 0x061000 and 0x101000 IPC message
90	LPVOID	lpDRMStream	Used in 0x081000 IPC message
B0	LPVOID	lpTaskList	Used in 0x0B1000 IPC message

MSO-Group of IPC Messages

WWLIBIPCMsg Object			
Offset	**Size**	**Field**	**Comment**
1C	UINT32	ui32IPC0A1100	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



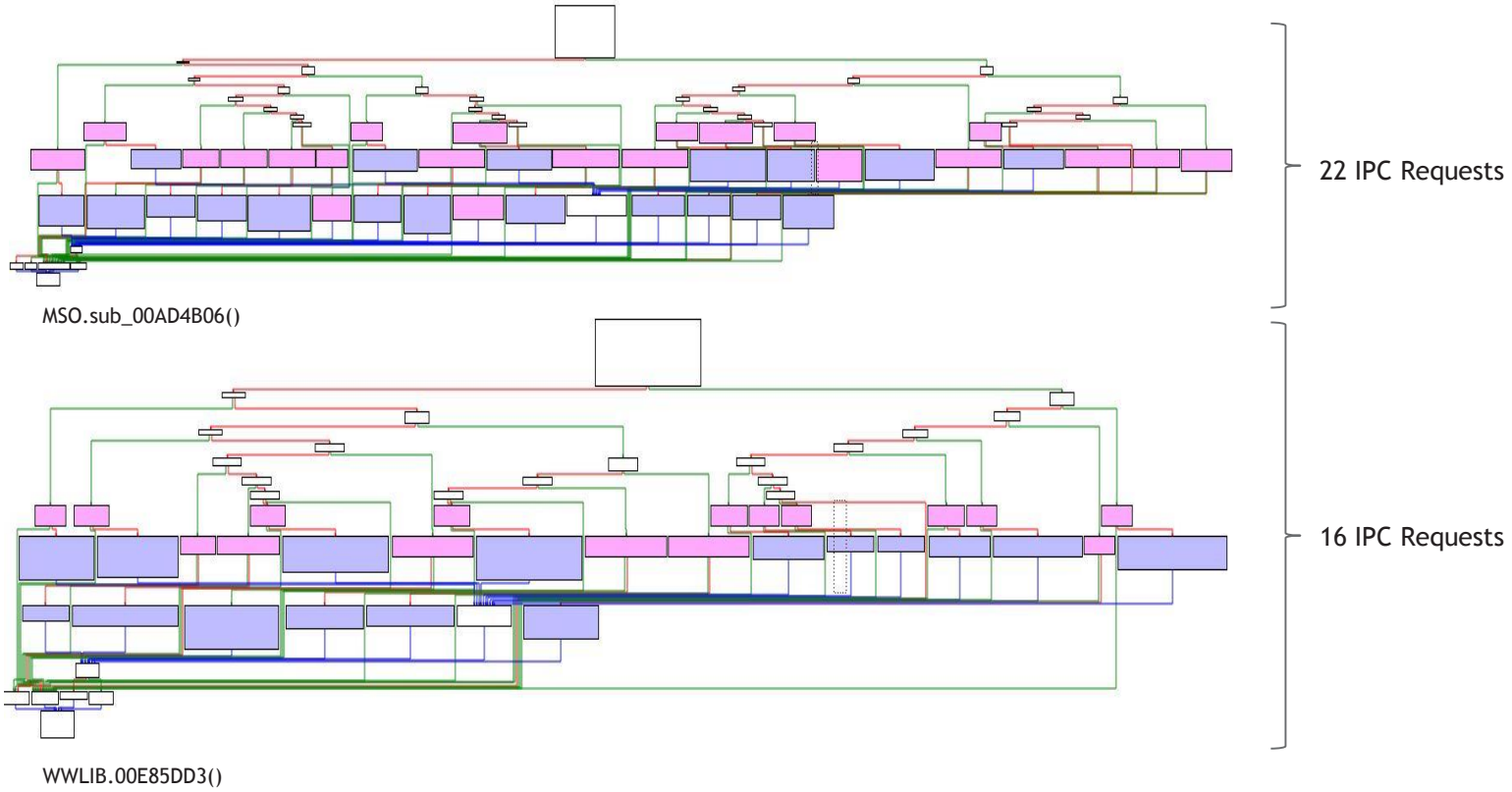
Request Header for *MSO-Group* (size 0x10-bytes)		
UINT32	ui32VirtualKey	Used in 0x071000 request. Otherwise ignored
UINT32	ui32MsgTag	Type of (22) IPC request Ranges 0x001000 to 0x151000, at inc 0x010000
UINT32	ui32MsgID	Matches IPC request to its response
UINT32	ui32MsgSize	Total size of IPC message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

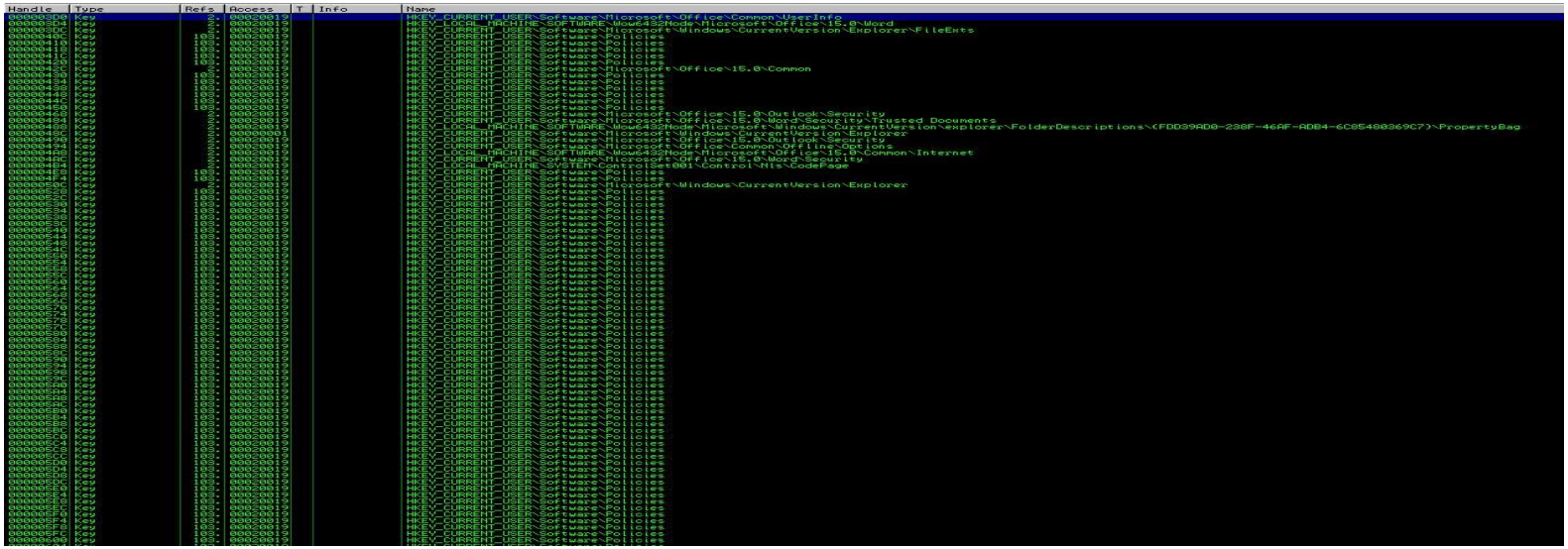


Pink: Sanity checks on IPC message, according to MsgTag

Purple: Service functions

IPC: MESSAGE 0X001000

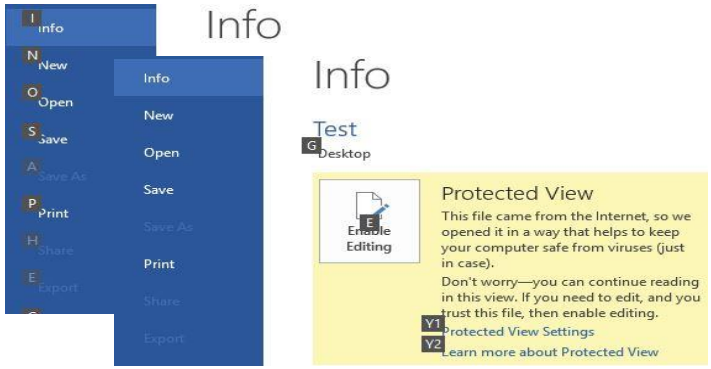
- No message body
- Registry key “HKEY_CURRENT_USER\Software\Policies”
  - Duplicate and return, with KEY_READ access
  - Work-around sandbox restrictions
  - Does not keep duplication count

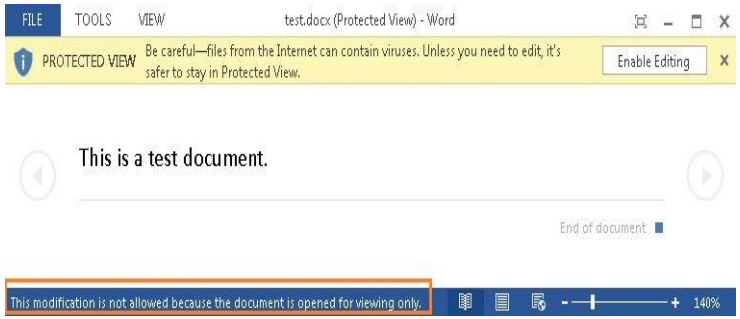


IPC: MESSAGE 0X061000

Message Body for 0x061000 Request		
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

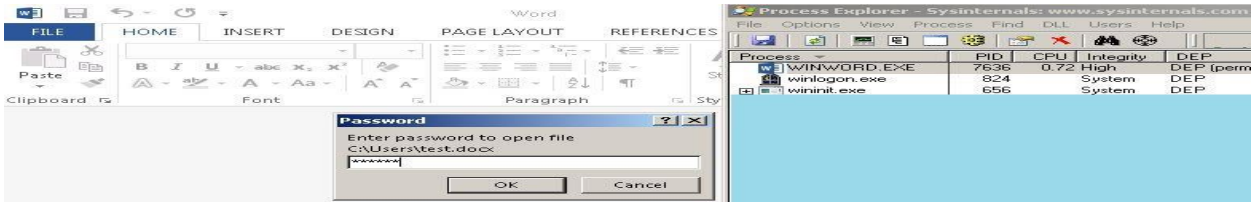
Message Body for 0x091000 Request		
Size	**Field**	**Comment**
UINT32	ui32ViewID	-
LPWSTR	pwzTarget	Parameter in *HlinkNavigateToStringReference()*
LPWSTR	pwzLocation	Parameter in *HlinkNavigateToStringReference()*
DWORD	grfHLNF	Parameter in *HlinkNavigateToStringReference()*
UINT32	ui32Unknown	-
HWND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

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

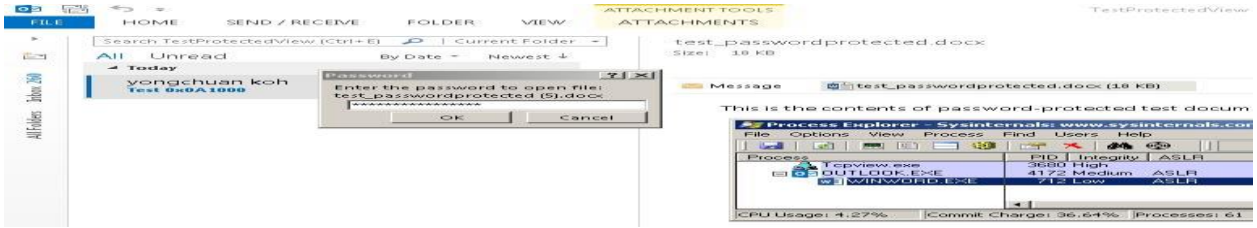
IPC: MESSAGE 0X0A1000

- "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

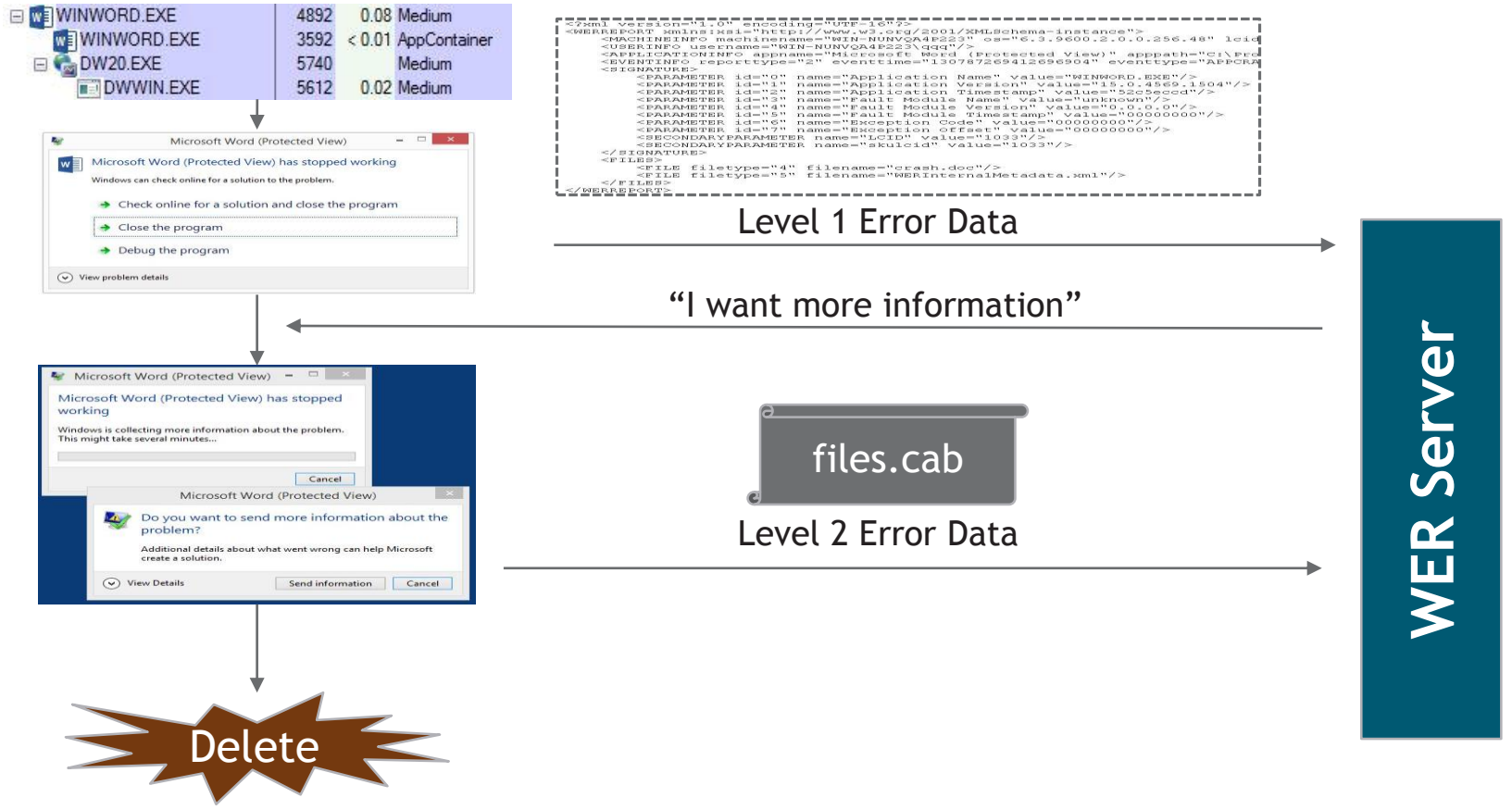
Message Body for 0x0C1000 Request		
Size	**Field**	**Comment**
HANDLE	hSandboxSharedMem	-
HANDLE	hEventBrokerIsDone	Broker sets this event after DW20.EXE exits and before it terminates sandbox
LPWSTR	lpwAdditionalWerFileName	Pointer to *wzAdditionalWerFileName*, or Null
UINT16	ui16AdditionalWerFileNameSize	-
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 *hBrokerSharedMem* memory
UINT32	ui32ProtectedViewPID	Sandbox Process-ID
UINT32	ui32ProtectedViewTID	TID of faulting thread in sandbox. Copied from *hSandboxSharedMem* offset 0x0C.
UINT32	uiProtectedViewEIP	EIP of faulting instruction in sandbox. Copied from *hSandboxSharedMem* offset 0x10.
LPVOID	lpProtectedViewPEP	Exception pointers in sandbox. Copied from *hSandboxSharedMem* 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 *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
		.....

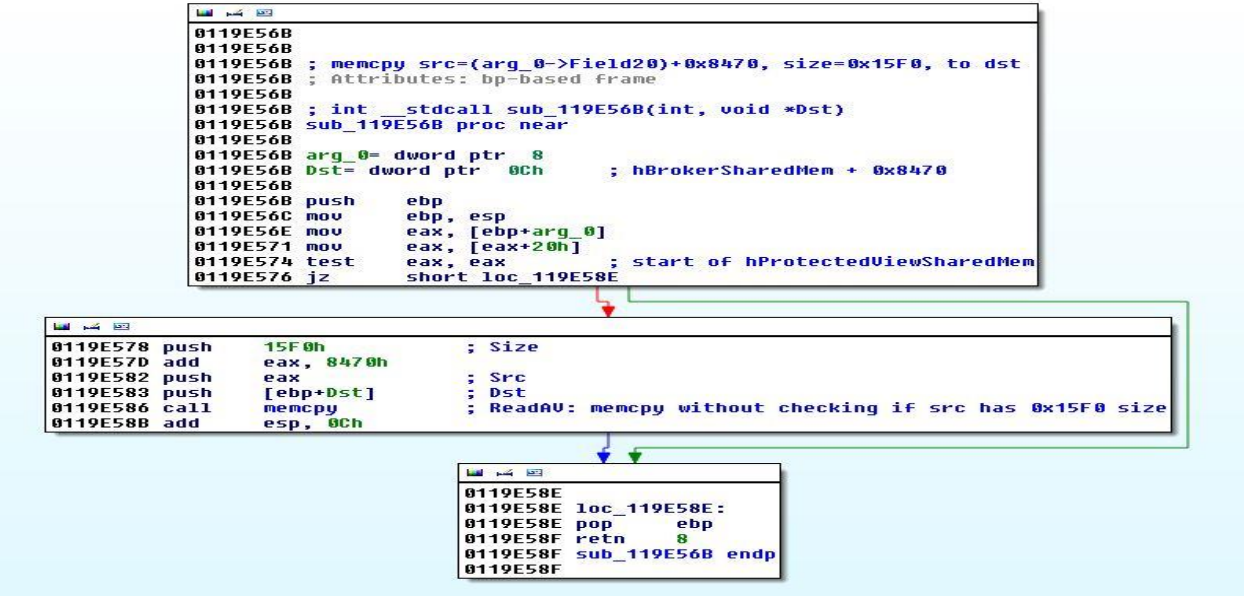
IPC: MESSAGE 0X0C1000

Windows Error Reporting



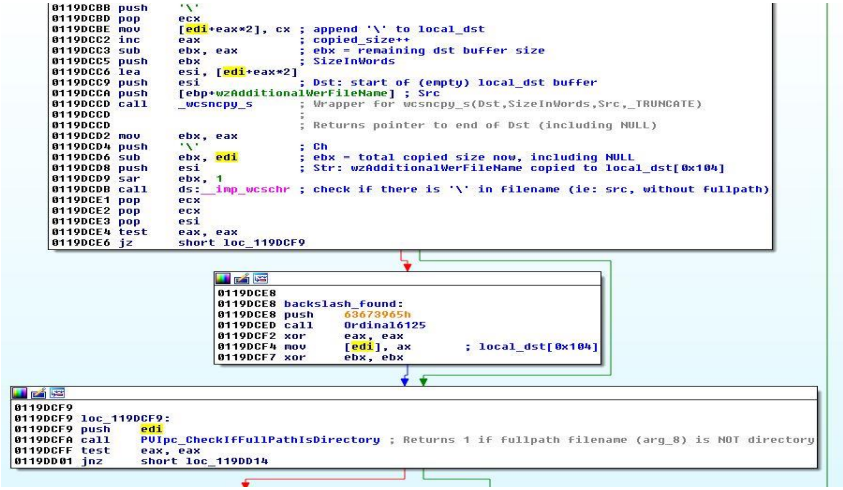
IPC: MESSAGE 0X0C1000

- 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…

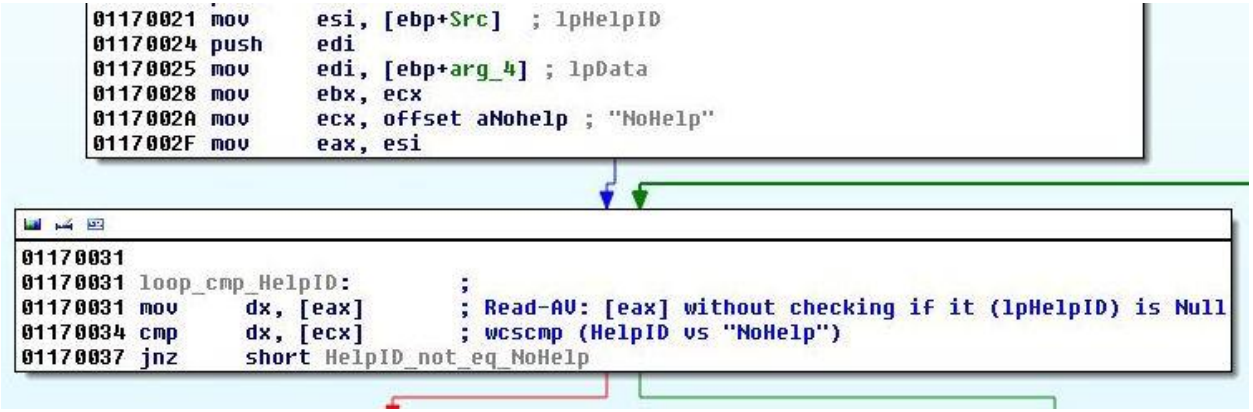
IPC: MESSAGE 0X0F1000

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

Message Body for 0x0F1000 Request		
Size	**Field**	**Comment**
UINT32	ui32ViewID	-
UINT32	ui32MessageID	0x465 or 0x469
LPWSTR	lpwData	Pointer to *wzData*, or Null
LPWSTR	lpwHelpID	Pointer to *wzHelpID*, 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



MS OFFICE 2013 PROTECTED-VIEW SANDBOX

MICROSOFT OFFICE 2016

- Sandbox Internals
- IPC Messages

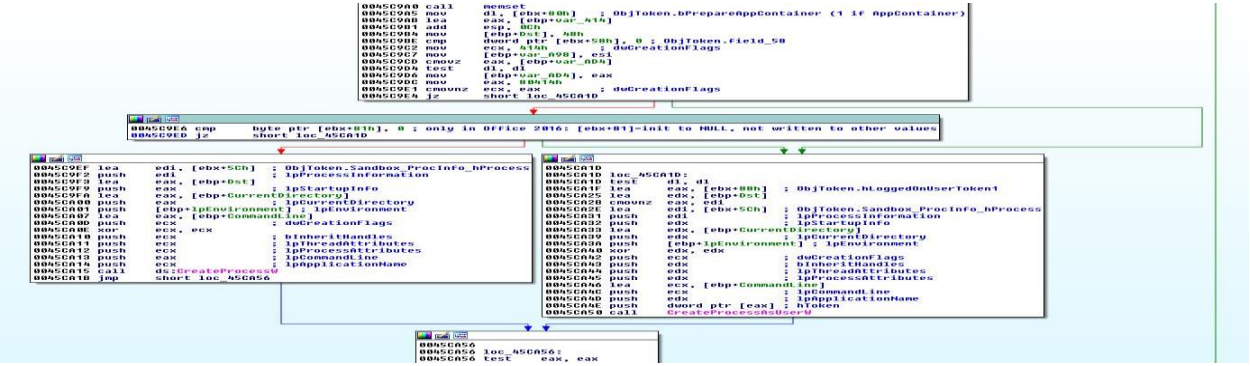
MWR LABS

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

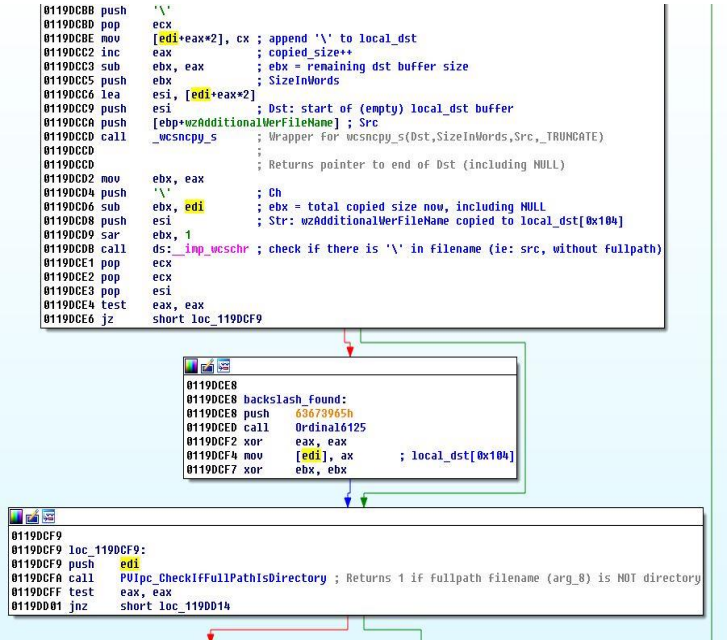
MS OFFICE 2016: SANDBOX INTERNALS

- Sandbox Initialization
  - New *CreateProcessW()* option in addition to *CreateProcessAsUserW()*
  - Does not seem reachable (BYTE [ebx+81h])

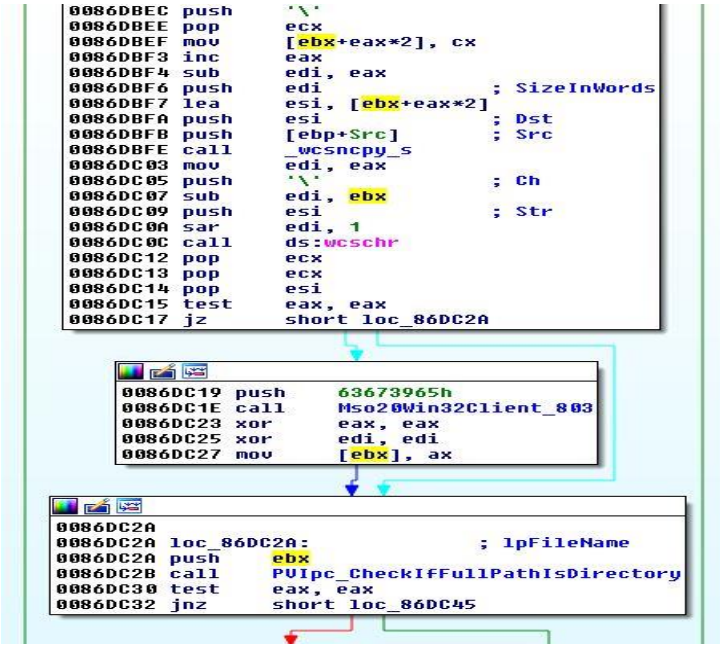


MS OFFICE 2016: IPC 0X0C1000

- 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 2013



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 *IpcUnprotectWindow()* or *_IpcUnprotectWindowNoDRM()*. If 1, protects window with *IpcProtectWindow()* or *_IpcProtectWindowNoDRM()*

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.
- MSDN. "**Naming Files, Paths, and Namespaces**". [Online] 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?