



Extracting the Windows Clipboard from Memory

By

James Okolica and Gilbert Peterson

Presented At

The Digital Forensic Research Conference

DFRWS 2011 USA New Orleans, LA (Aug 1st - 3rd)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

<http://dfrws.org>



Extracting the Windows Clipboard from Physical Memory

James S. Okolica
Gilbert L. Peterson





Overview

- **Cyber Forensics and Live Response**
 - Why do we need it? What can we get from it?
 - The Compiled Memory Analysis Tool (CMAT)
- **Windows Clipboard**
 - What it does and how it works (from an API perspective)
 - Under the covers
- **Integrating Clipboard into CMAT**
 - User Side (user32.dll)
 - Kernel Side (win32k.sys)
- **Experimental Set up and Results**
- **Next Steps**



Live Cyber Forensics

- Business Productivity
 - Lost Revenue
 - Supervisory Control and Data Acquisition (SCADA) Systems
 - Concern of the system coming back up
- Acquisition of volatile-only information
 - Network Traffic
 - Active process and user information
- Encrypted Hard Drives
- Memory Resident Malware
- Too much data



Types of Volatile Data

- General Operating System Information
- Services/ Driver Information
- Logged on users and their authentication credentials
- Registry Information
- Process Information
 - Open Files
 - Open Registry Keys
 - Network Connections and Status
 - Dynamic Link Libraries
- Command History
- Clipboard Contents

Sutherland et al (2008). Acquiring Volatile Operating System Data Tools and Techniques. ACM SIGOPS O/S Review 42 (3)
Carvey (2007). Windows Forensic Analysis.



A Compiled Memory Analysis Tool (CMAT)

- **Determines O/S version** (using `_DBGKD_DEBUG_DATA_HEADER64` or finding the kernel PE)
 - Physical Address Extensions enabled/disabled, 32 bit/64 bit
- **Loads O/S specific data structures** (by retrieving PDBs from Microsoft's Symbol Server)
- **Locates O/S-agnostic signatures for processes and registries**
- **Connects users found in the registry with processes**
- **Locates data structures within PEs** (by retrieving PDBs from Microsoft's Symbol Server)
 - Network activity
 - Clipboard data



The Windows Clipboard

- Sharing data between applications
 1. Select an object and send it to a common area
 2. Retrieve the object from the common area
- Observations:
 - Only one object can be in the common area at a time
 - The object can be stored in multiple formats
- History
 - Dynamic Data Exchange (DDE)
 - Object Linking and Embedding (OLE)
 - Compound Object Model (COM)
 - Object Linking and Embedding v2.0 (UDT, Drag & Drop)
 - Active X
 - .NET



Windows Clipboard Format & Functions

- Predefined Formats

- Formats identified by Microsoft when the Clipboard was initially implemented

CT_TEXT	0x0001
CF_BITMAP	0x0002
CF_TIFF	0x0005
CF_WAVE	0x000C

- Private Formats

- Formats developed by vendors (including Microsoft) to enable transfer of proprietary formats (e.g., Microsoft Office objects)

OLE	0xC013
IDataObject	0xC009

- Multi-Formats

- Although only one piece of data can be in the Clipboard at a time, programs can save that data in multiple formats (e.g., MS Office, OLE Object, Unicode, ASCII)

Transferring Text to the Clipboard

```
hGlobal = GlobalAlloc (GHND |
                    GMEM_SHARE, iLength + 1) ;
pGlobal = GlobalLock (hGlobal);

for (i = 0; i < wlength; i++)
    *pGlobal++ = *pString++;

GlobalUnlock (hGlobal);

OpenClipboard (hwn);
EmptyClipboard();

SetClipboardData (CF_TEXT, hGlobal);

CloseClipboard();
```

Retrieving Text from the Clipboard

```
OpenClipboard (hwnd);

hGlobal = GetClipboardData
        (CF_TEXT);
```




Reversing Methodology

- Create a Virtual Machine (VM)
- Execute a copy/paste operation
- Perform dynamic analysis to locate the structures
- Generate a dump file of the VM's memory and duplicate the dynamic analysis

Transferring Text to the Clipboard

```
hGlobal = GlobalAlloc (GHND |
                   GMEM_SHARE, iLength + 1) ;
pGlobal = GlobalLock (hGlobal);

for (i = 0; i < wlength; i++)
    *pGlobal++ = *pString++;

GlobalUnlock (hGlobal);

OpenClipboard (hwn);
EmptyClipboard();

SetClipboardData (CF_TEXT, hGlobal);

CloseClipboard();
```

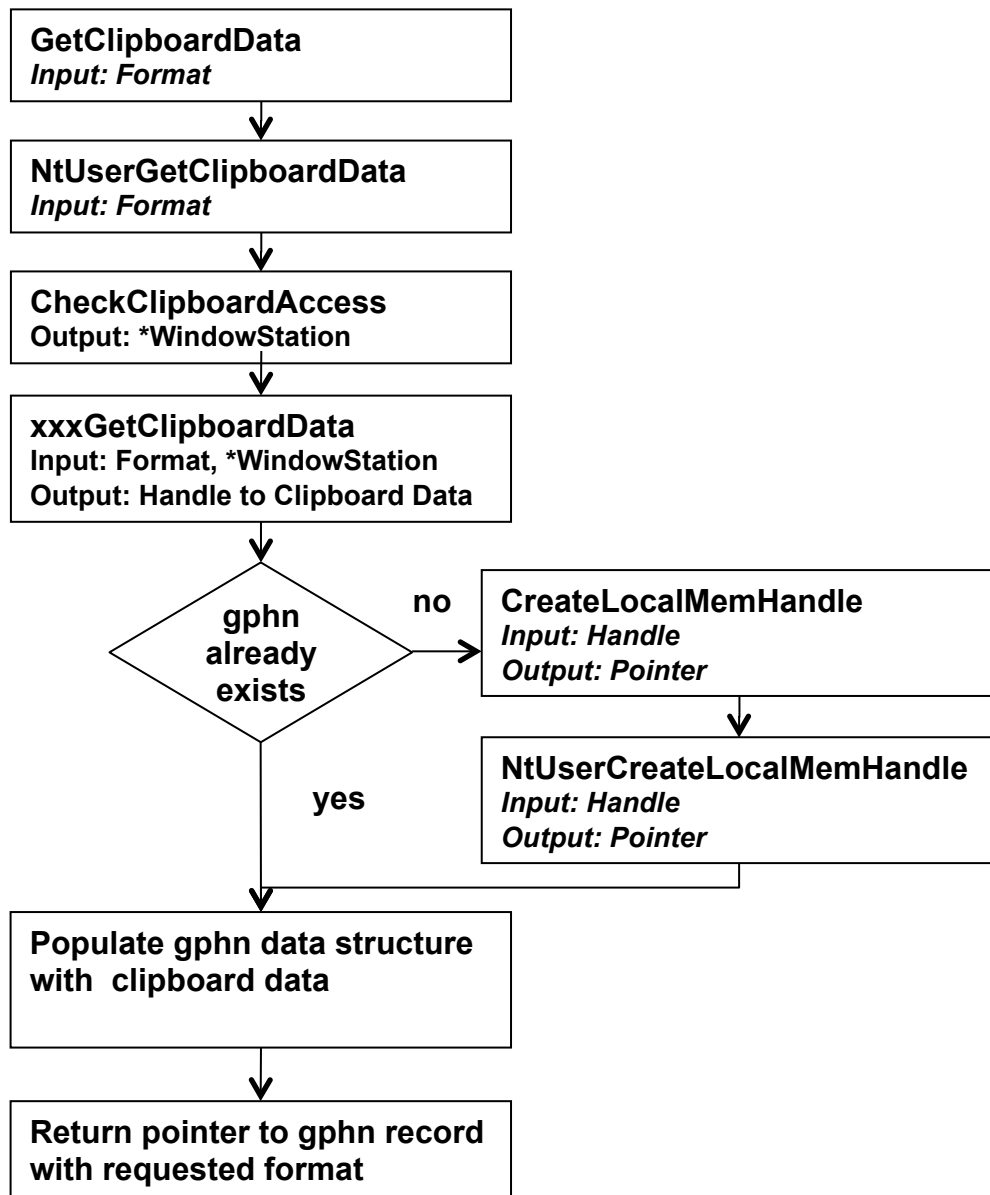
Retrieving Text from the Clipboard

```
OpenClipboard (hwnd);

hGlobal = GetClipboardData
        (CF_TEXT);
```



GetClipboardData

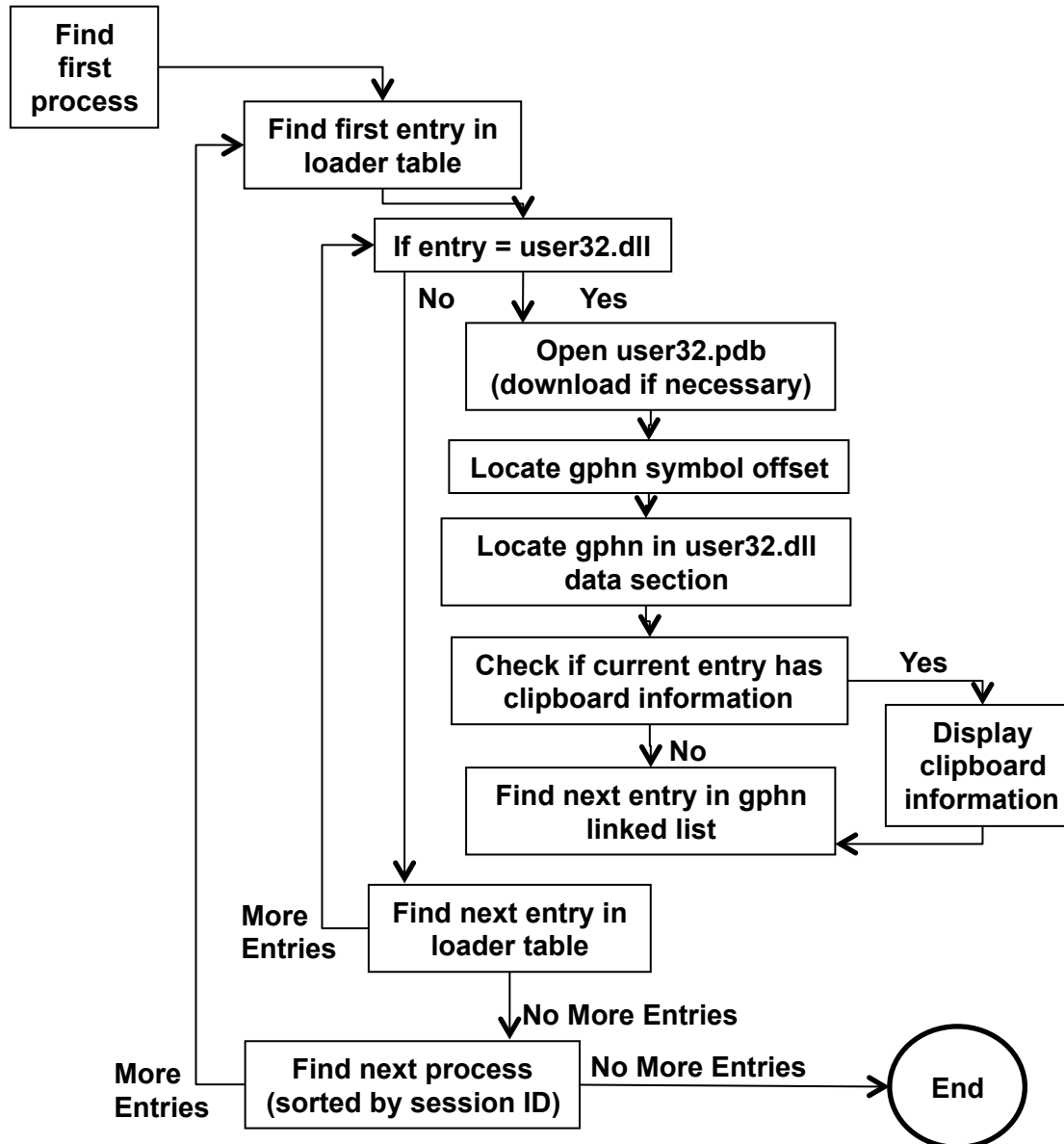


Clipboard Structure

32 bit Offset	64 bit Offset	Data Type	Field Name
0x00	0x00	gphn*	Next
0x04	0x08	uint16_t	Format
0x08	0x10		Unknown
0x0c	0x18	void*	Handle

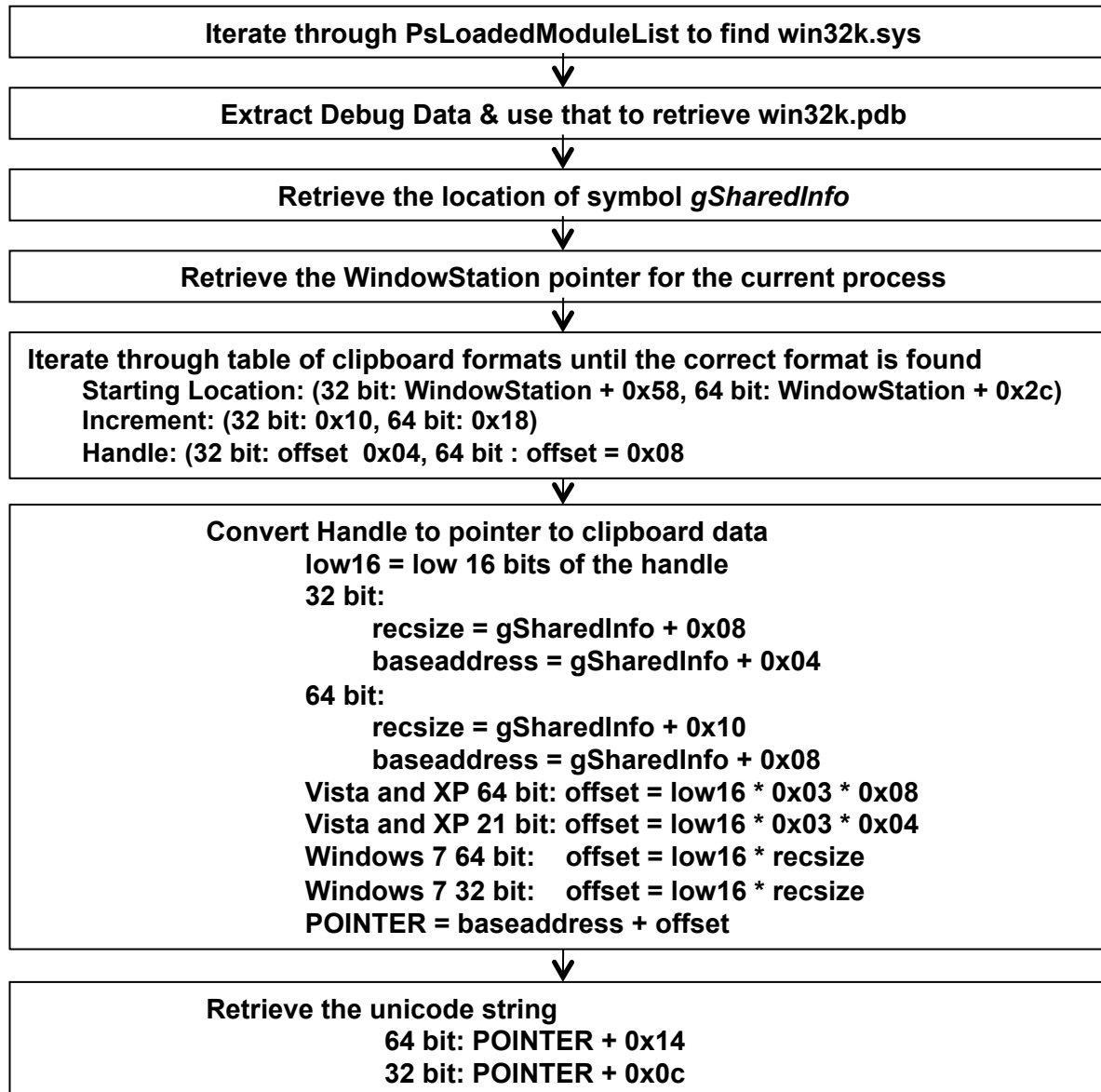


User Clipboard Integration (user32.dll)





Kernel Clipboard Integration (win32k.sys)





Experimental Setup

- DFRWS 2008 Forensic Rodeo
 - 2 Windows XP 32 bit memory dumps
- NIST CFReDS dataset –
 - 1 Windows Vista 32 bit memory dump
 - 2 Windows XP memory 32 bit dumps
- Additional memory dumps
 - 6 operating system configurations
 - Windows XP SP3 32 bit, Windows Vista (pre-SP1) 32 bit, Windows 7 SP3 32 bit
 - Windows XP SP2 64 bit, Windows Vista (pre-SP1) 64 bit, Windows 7 SP3 64 bit
 - For each operating system configuration
 - Memory dump with clipboard data from MS Excel 2007
 - Memory dump with clipboard data from MS Word 2007
 - Memory dump with clipboard data from Notepad



Results

Dataset	Memory Image	Results
DFRWS2008	Dfrws	“Pp -B -p -o out.pl file” command found
CFReDS	Vista-beta2.img	No Clipboard Data Found
CFReDS	Xp-laptop-2005-06-25.img	No Clipboard Data Found
CFReDS	Xp-laptop-2005-07-04-1430.img	Non-textual Clipboard Data Found
Generated	32 bit XP w/ Notepad	Notepad Clipboard Data Found
Generated	32 bit XP w/ MS Word	MS Word Clipboard Data Found
Generated	32 bit XP w/ MS Excel	MS Excel Clipboard Data Found
Generated	64 bit XP w/ Notepad	Notepad Clipboard Data Found
Generated	64 bit XP w/ MS Word	MS Word Clipboard Data Found
Generated	64 bit XP w/ MS Excel	MS Excel Clipboard Data Found



Results

Dataset	Memory Image	Results
Generated	32 bit Vista w/ Notepad	Notepad Clipboard Data Found
Generated	32 bit Vista w/ MS Word	MS Word Clipboard Data Found
Generated	32 bit Vista w/ MS Excel	MS Excel Clipboard Data Found
Generated	64 bit Vista w/ Notepad	Notepad Clipboard Data Found
Generated	64 bit Vista w/ MS Word	MS Word Clipboard Data Found
Generated	64 bit Vista w/ MS Excel	MS Excel Clipboard Data Found
Generated	32 bit Vista w/ Notepad	Notepad Clipboard Data Found
Generated	32 bit Win7 w/ MS Word	MS Word Clipboard Data Found
Generated	32 bit Win7 w/ MS Excel	MS Excel Clipboard Data Found
Generated	64 bit Win7 w/ Notepad	Notepad Clipboard Data Found
Generated	64 bit Win7 w/ MS Word	MS Word Clipboard Data Found
Generated	64 bit Win7 w/ MS Excel	MS Excel Clipboard Data Found



Next Steps

- Proprietary/ Application Specific Formats
 - IDataObjects
 - OLE Objects
- Different ways to copy data
 - Between applications
 - Within an application
 - Drag and Drop
- Formalizing a process for reversing DLLs and Drivers



Questions

?