Plumbing the Depths: ShellBags

SA Eric R. Zimmerman
saericzimmerman@gmail.com
801-514-4064
@EricRZimmerman
https://binaryforay.blogspot.com/
Why ShellBags?

• The information contained in ShellBags is relevant to forensic examiners, incident response teams, and law enforcement as it quickly and easily provides context around a user's action in addition to their interaction with a computer and its associated resources.

• These include, but are not limited to:
  • Historical record of interactions with a multitude of computer resources
    • TrueCrypt containers, SharePoint, FTP, UNC access, Hyper-V resources, cloud storage
  • Available whether or not you have access to the resources accessed
Topics of discussion

• ShellBag basics
  • Commonalities
  • Types
  • Data structures

• A comparison of ShellBag tools
  • RegRipper
  • sbags
  • ShellBags Explorer/SBECmd

• Using ShellBags Explorer for fun and profit
Before we begin...

<table>
<thead>
<tr>
<th>Time</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
<th>Value 9</th>
<th>Value 10</th>
</tr>
</thead>
</table>
| 0000h| 14 00 | 1F 42 | 25 49 | 1E 03 | 96 7B | C3 4D | B1 31 | E9 06 | ... | B8H..."\n\nBefore we begin...

What is this? (It is from the Registry)

(Hint: It is not a ShellBag!)
ShellBag basics: Types of ShellBags

- Drive letter
- Control panel items
- GUIDs
- Network resources
- Files
- Directories
- MTP devices
- Variable
ShellBag basics: Commonalities between ShellBag layouts

- **Size**
  - First 2 bytes (includes bytes used for size)

- **Type**
  - Many different kinds. One byte at offset 0x03

Luckily, most ShellBags use several common data structures
ShellBag basics: Data structures

• Serialized Property Store
• Extension blocks

• For both structure types
  • What are they for?
  • What do they look like
  • What do they look like decoded?
Serialized property storage

• What are they for?
  • Microsoft Property Store Binary File Format. This file format is a persistence format for a set of properties. Implementers can use this file format to store a set of properties in a file or within another structure.
  • The catch is developers can use it to store anything (which in some cases means it can’t be decoded)
  • Basically, it’s used to store key/value pairs

• (Some of the) Gory details available [here](#)
YOU ARE IN A MAZE OF TWISTY LITTLE PASSAGES, ALL ALIKE.
Serialized property storage (SPS)

Block

- Size (offset 0x00, 4 bytes)
- Version (offset 0x04): Always 1
- Format (offset 0x08, 16 bytes)

Property value (offset 0x18)

- Key/value
- Key/value
## Serialized Property Storage (SPS)

### Key

- **Value size** (offset 0x00, 4 bytes)
- **Name Size** (offset 0x04, 2 bytes)
- **Name** (offset 0x06, 4 bytes)

### Value (offset 0x06 + Name Size)

#### Type

- VT_FILETIME (0x0040) = 64-bit timestamp
- VT_BOOL (0x000B) = Boolean
- VT_CLSID (0x0048) = GUID

#### Data

(how to decode depends on Type)
Serialized property storage: What do they look like on disk?

- Size: 135 bytes
- Signature: 1SPS (0x31535053)
- Format: efb725f1-1a47-a510-f102-608c9eebac1d
- Key/value pairs: 3 of them
  - Key: 10, Value: OCCT
  - Key: 4, Value: Dossier de fichiers
  - Key: 14, Value: 06/07/2013 07:37:46
Extension blocks

• What are they for?
  • Allow reuse of common structures in different places
  • Used to hold such things as MFT info, GUIDs, timestamps, additional shell items (?!)
  • Can also hold SPS objects as well
• Some ShellBags can have dozens of extension blocks in them
Extension blocks

Extension block

Data (offset 0x08)

Size (offset 0x00, 2 bytes)

Version (offset 0x02, 2 bytes)

Signature (offset 0x04, 4 bytes)

Layout varies depending on signature
Extension blocks

First block is 0x56 bytes in size
Signature is beef0004

Second block is 0x2A in size
Signature is beef0013
Extension blocks: Decoded!

Signature: 0xbeef0004
Size: 80
Version: 9
Version Offset: 0x18
Identifier: 2E (Windows 8.1)

Created On:
12/12/2014 5:51:28 PM +00:00
Last Access:
12/12/2014 5:51:28 PM +00:00

Long Name: Hives for lunch
MFT Entry Number: 423308
MFT Sequence Number: 10
File system hint: NTFS
Extension blocks: beef0004

**Timestamps**
- Created
- Modified
- Last accessed

**Version #**
- Indicates which OS generated the extension block

**File reference info**
- Entry #
- Sequence #

**Names**
- Long name (Unicode)
- Localized name (Unicode or ASCII)
Extension blocks: beef0004 (NTFS)

Signature is found at offset 0x04.

The signature is 0x0400EFBE. Since its stored in little endian format, we have to flip it around to get to BEEF0004

Using this knowledge you can spot extension blocks in raw hex easily. Just look for 00 EF BE

and from there you can determine what kind of extension block it is and decode things appropriately.
When dealing with ShellBags created via browsing FAT file systems, things change slightly.

**Creation date and time**
0xC3424D43

DCode shows a date of:
**Mon, 03 June 2013 08:26:26 Local**

In this example, the **Last access date and time** (found at offset 0x12) 0xC24200B0 which converts to:

**Sun, 02 June 2013 22:00:00 Local**

Notice the last access timestamp resolution. This is an indicator this is not NTFS

---

**Note:** FAT timestamps are usually stored using the local time of the computer where they were generated. However, in ShellBags, they are first converted to UTC time before storing them in the Registry. Because of this, we can convert MAC timestamps to any time zone we want to (and ShellBags Explorer makes this easy)
Extension blocks: beef0004 (exFAT)

• Things work similarly with exFAT compared to FAT in that we do not have a sequence number.
• However, one thing that is different is the timestamp resolution. With exFAT, we once again gain greater resolution to include the minutes and seconds.

11/7/2014 10:22:38 PM +00:00
Extension blocks: Determining NTFS vs FAT vs exFAT

- With NTFS, we always get an MFT entry and sequence number
- With FAT and exFAT, we only get an entry number.

- Using our knowledge of how ShellBags record MFT info in beef0004 blocks, we can reliably differentiate between NTFS, FAT, and exFAT formatted disks
Extension blocks: Other fun facts

• On NTFS file systems, the MFT entry and sequence numbers come from the FILE record for the directory.

• With exFAT and FAT, the MFT entry number is the *absolute* offset to the structure defining the directory on disk.

• For FAT and exFAT, this info can be used to find data on quick formatted disks (in addition to traditional carving).

• For NTFS we can make educated guesses if accessed directories were on same physical device based on entry and sequence numbers.
Extension blocks: Other useful stuff

- beef0008: Paths to things in the Recycle bin, the original path, and when the properties for the item in the Recycle bin was viewed
- beef0003, beef0000, beef0019: GUIDs (that often map to folders)
- beef0025: Timestamps
- ??? What will you find?
ShellBag Basics: Drive letter

- This is the simplest case
- Type (offset 0x02): 0x2F
- Nothing more than a drive letter starting at offset 0x03
ShellBag Basics: Control Panel Categories

- Type (offset 0x02): 0x01
- Offset 0x08 determines the category
  - 0x00 = All Control Panel Items
  - 0x01 = Appearance and Personalization
  - 0x02 = Hardware and Sound
  - 0x03 = Network and Internet
  - 0x04 = Sound, Speech and Audio Devices
  - 0x05 = System and Security
  - 0x06 = Clock, Language, and Region
  - 0x07 = Ease of Access
  - 0x08 = Programs
  - 0x09 = User Accounts
  - 0x10 = Security Center
  - 0x11 = Mobile PC
ShellBag Basics: Control Panel GUIDs

- Type (offset 0x02): 0x71
- Offset 0x14: Start of 16 byte GUID
- Once you have bytes you have to convert to GUID format by cutting up the bytes and rearranging them
- GUID == bb64f8a7-bee7-4e1a-ab8d-7d8273f7fdb6 which maps to “Action Center”
- SBE contains over 370 GUID to common name mappings
ShellBag Basics: Root folder GUID

- Type (offset 0x02): 0x1F
- Offset 0x04: Start of 16 byte GUID
- In this case, the GUID is 20d04fe0-3aea-1069-a2d8-08002b30309d which maps to “My Computer”
ShellBag Basics: Directories (fixed)

- Type (offset 0x02): 0x31 (typically)
- Offset 0x08: Last modified date (DOS/FAT format)
- Offset 0x14: Primary name
- Offset varies based on length of primary name: beef0004 extension block. In this case it starts at 0x26 and ends at 0x61
ShellBag Basics: Directories (fixed)

Short name: xwfportable
Modified: 12/23/2014 2:01:08 PM +00:00

Value: xwfportable Shell Type: Directory

Bag Path: BagMRU\0\0, Slot #: 8, MRU Position: 8
Absolute Path: Desktop\My Computer\J:\xwfportable

Extension blocks found: 1
---------------------- Block 0 ----------------------
Signature: 0xbeef0004 Size: 72 Version: 9 Version Offset: 0x1A

Identifier: 2E (Windows 8.1)

Created On: 12/23/2014 6:49:46 PM +00:00
Last Access: 12/23/2014 6:49:46 PM +00:00

Long Name: xwfportable

MFT Entry Number: 3067 MFT Sequence Number: 4
ShellBag Basics: Directories (optical storage)

- **Type (offset 0x02):** 0x00
- **Offset 0x04:** Unique signature, AugM
- **Made up of one or more subshell items**
  - **Offset 0x1C:** Created on date (FAT)
  - **Offset 0x22:** Short name
  - **Offset varies:** beef0004 extension block.
- **This example has 2 subshell items. A second beef004 block starts at offset 0x80**
ShellBag Basics: Files

- Type (offset 0x02): 0x32
- Offset 0x04: Size of file as 32-bit unsigned integer
- Rest looks and feels like a directory with type 0x31 (in fact, it’s the same structure)
- Contains a beef0004 block
ShellBag Basics: Folders inside zip files

- Type (offset 0x02): varies
- Offset 0x24: Usually has a last access timestamp, but can be “N/A” as well
- Up to 2 directory names can be found
  - In this example, the ShellBag contains both the current directory and the parent
- File system view can be seen to far left
ShellBag Basics: Network resources

- Type (offset 0x02): 0xC3
- Offset 0x05: Network resource name
- Captures UNC access as well as Hyper-V related remote file system access
ShellBag Basics: Network resources (FTP)

- Type (offset 0x02): 0x61
- Offset 0x0E: Connect time
- Offset 0x2E: Host name
- Can also contain username (but sadly, no password) if FTP server required it
ShellBag Basics: MTP devices

Root folder

- Volume shell item
  Signature 0x10312005

- File entry shell item
  Signature 0x07192006

Best to see these in SBE as they have a lot of data in them.
ShellBag Basics: Variable

Network resources
- FTP server directory structures
- HTTP and FTP URIs

Directories
- Local
- Remote

MTP type 2 items

GUIDs
- Often map to folders

Zip file contents
- Can contain timestamps

A horrible beast, includes a little of many other types, all of which can contain extension blocks, etc.

Best viewed in SBE due to the variation
A comparison of ShellBag tools

- RegRipper v2.8 (shellbags.pl v20130102)
- sbags v0.39 and v0.40
- ShellBags Explorer v0.0.5.4
ShellBag tools: RegRipper

- CSV output
- Abbreviated ShellBag information
- Fails gracefully on unknown bags/content
ShellBag tools: sbags

- CSV (pipe character) output
- More detailed ShellBag information (metadata, timestamp aggregation)
- Drops data it doesn’t understand
ShellBag tools: SBECmd

- **TSV output**
- **Even more detailed ShellBag information (First/last explored, extension block info, file system info)**
ShellBag tools: ShellBags Explorer

- GUI
- Multiple export formats (json, Excel, TSV)
- Full ShellBag information (First/last explored, extension block info, file system info, hex, etc)
- Time zone support
- Auto discovery of unknown GUIDs, extension blocks, etc
# ShellBag tools: Initial test results

<table>
<thead>
<tr>
<th>Test #</th>
<th>Hive size (MB)</th>
<th># bags RR</th>
<th># bags sbags</th>
<th># bags SBECmd</th>
<th>Parse Time RR*</th>
<th>Parse Time sbags</th>
<th>Parse Time SBECmd**</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.25</td>
<td>4,443’^</td>
<td>4,327” (-116)</td>
<td>4,443</td>
<td>26 sec</td>
<td>3.95 sec</td>
<td>1.54 sec</td>
</tr>
<tr>
<td>2</td>
<td>5.50</td>
<td>175^</td>
<td>173” (-2)</td>
<td>175</td>
<td>1 sec</td>
<td>0.14 sec</td>
<td>0.33 sec</td>
</tr>
<tr>
<td>3</td>
<td>5.50</td>
<td>242’^</td>
<td>237” (-5)</td>
<td>242</td>
<td>&lt; 1 sec</td>
<td>0.23 sec</td>
<td>0.27 sec</td>
</tr>
<tr>
<td>4</td>
<td>6.25</td>
<td>631’^~</td>
<td>612” (-19)</td>
<td>631</td>
<td>2 sec</td>
<td>0.49 sec</td>
<td>0.37 sec</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>303’^~</td>
<td>289” (-14)</td>
<td>303</td>
<td>1 sec</td>
<td>0.22 sec</td>
<td>0.45 sec</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>1,212^</td>
<td>1,177” (-35)</td>
<td>1,212</td>
<td>2 sec</td>
<td>0.71 sec</td>
<td>0.56 sec</td>
</tr>
<tr>
<td>7</td>
<td>1.25</td>
<td>121</td>
<td>121</td>
<td>121</td>
<td>&lt; 1 sec</td>
<td>0.09 sec</td>
<td>0.04 sec</td>
</tr>
<tr>
<td>8</td>
<td>12.3</td>
<td>5,195^</td>
<td>5,191” (-4)</td>
<td>5,195</td>
<td>10 sec</td>
<td>1.63 sec</td>
<td>1.29 sec</td>
</tr>
</tbody>
</table>

* RegRipper times only listed whole seconds
** SBECmd parses the entire hive (including deleted recovery), not just BagMRU branch
^ Parsing errors for Win8.1 hives due to change in extension block
‘ Unknown types
” Missing bags (Bad as data is silently dropped!)
~Wrong/Unknown GUIDs
ShellBag tools: Subsequent test results

• In early February 2015, I notified TZWorks about missing ShellBags using v0.39 of sbags

• TZWorks was sent test cases 1 and 6.

• As of version 0.40, ShellBag *counts* match what SBE is outputting, however...
ShellBag tools: retesting sbags 0.40

<table>
<thead>
<tr>
<th>Test #</th>
<th>Hive size (MB)</th>
<th># bags sbags 0.39</th>
<th># bags sbags 0.40</th>
<th># bags SBECmd</th>
<th>Parse Time sbags 0.40</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.25</td>
<td>4,327” (-116)</td>
<td>4,443</td>
<td>4,443</td>
<td>3.89 sec</td>
</tr>
<tr>
<td>2</td>
<td>5.50</td>
<td>173” (-2)</td>
<td>175</td>
<td>175</td>
<td>0.17 sec</td>
</tr>
<tr>
<td>3</td>
<td>5.50</td>
<td>237” (-5)</td>
<td>242</td>
<td>242</td>
<td>0.24 sec</td>
</tr>
<tr>
<td>4</td>
<td>6.25</td>
<td>612” (-19)</td>
<td>631*</td>
<td>631</td>
<td>0.50 sec</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>289” (-14)</td>
<td>303</td>
<td>303</td>
<td>0.23 sec</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>1,177” (-35)</td>
<td>1,212</td>
<td>1,212</td>
<td>0.81 sec</td>
</tr>
<tr>
<td>7</td>
<td>1.25</td>
<td>121</td>
<td>121</td>
<td>121</td>
<td>0.10 sec</td>
</tr>
<tr>
<td>8</td>
<td>12.3</td>
<td>5,191” (-4)</td>
<td>5,195</td>
<td>5,195</td>
<td>1.37 sec</td>
</tr>
</tbody>
</table>

* Missing path info (Hyper-V related)
" Missing bags (Bad as data is silently dropped!)
SBE processing test case 4
And in sbags 0.40

<table>
<thead>
<tr>
<th>seq#</th>
<th>full path</th>
<th>source subkey/value name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\0\1</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
<tr>
<td></td>
<td>Desktop\</td>
<td>Shell\Bag\RU\13\2\0\1\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0</td>
</tr>
</tbody>
</table>
ShellBag tools: sbags Accuracy

• Device names:
  • sbags
    • Desktop\Control Panel\Hardware and Sound\{CLSID_DeviceCenter}\Provider%5CMicrosoft.Base.DeviceDisplayObjects\DDO:%7B3A59AF9D-5144-5016-AC66-FF31CB3A287D%7D\n  • SBE
    • Desktop\Control Panel\Hardware and Sound\Device Center(Devices and Printers)\Lexmark X5400 Series\n
• Missing folder names
  • sbags
    • Desktop\Device Pairing Folder\n  • Sbags
    • Desktop\Previous Versions Results Delegate Folder\n  • SBE
    • Desktop\ProxyHelper\sbags\Desktop\Previous Versions Results Delegate Folder\
ShellBag tools: sbags Accuracy

- **Unknown GUIDs**
  - **sbags:**
    - Desktop\My Computer\uuid:e55ee164-0817-4500-bb1b-d0491d043d3e\n    - Desktop\My Computer\uuid:2963ddf7-3f4d-4b8b-9187-44571e0cfded\n  - **SBECmd:**
    - Desktop\My Computer\DLNA Media Servers Data Source
    - Desktop\My Computer\DLNA Media Servers Data Source
ShellBag tools: sbags Accuracy

• sbags provides no visibility into folders inside zip files
  • Desktop\My Computer\U:\Downloads\CFTL\CFTL-master.zip\Desktop\My Computer\U:\Downloads\CFTL\CFTL-master.zip
  • Desktop\My Computer\U:\Downloads\CFTL\CFTL-master.zip

• SBECmd output
  • Desktop\My Computer\U:\Downloads\CFTL\CFTL-master.zip\CFTL-master
  • Desktop\My Computer\U:\Downloads\CFTL\CFTL-master.zip\CFTL-master\Scanner
  • Desktop\My Computer\U:\Downloads\CFTL\CFTL-master.zip\CFTL-master\Viewer
ShellBag tools: RegRipper Accuracy

RegRipper has had issues with Windows 8.1 hives due to changes in beef0004 extension block.

On March 4, 2015, Harlan Carvey emailed me an updated version of ShellBags.pl which fixes parsing of hives through Windows 10.
It is not up to a developer to decide what is relevant to include or exclude.

It is better to spectacularly fail and tell the end user than silently drop data.

Without access to all the data, how would you know what was missing?
  - (It could be argued if you have the Registry hives, you do have all the info, but this is a heavy lift when it comes to verification in a hex editor)
Using ShellBags Explorer: Overview

Disclaimer:

We won’t spend a ton of time here due to time constraints. For a lot more details on SBE, watch the two episodes of the Forensics Lunch from November 28, 2014 and December 12, 2014.
Using ShellBags Explorer: Piercing TrueCrypt

• Overview
  • Using Configuration.xml to tell last drive letter TC container mounted on
  • Think about how this can be used in relation to TrueCrypt’s ability to use hidden volumes...are they hidden anymore?
Using ShellBags Explorer: Piercing TrueCrypt
Using ShellBags Explorer: Deduplicating hives

- **How?**
  - SHA-1 of key ShellBag fields
  - Only unique SHA-1 values are included

- **Why?**
  - Seeing user activity on a file system over time
  - Timelines
    - Multiple access timestamps
Using ShellBags Explorer: Deduplicating hives

<table>
<thead>
<tr>
<th>BagPath</th>
<th>Slot</th>
<th>MRUPosition</th>
<th>AbsolutePath</th>
<th>LastWriteTime</th>
<th>SourceFile</th>
<th>MFTEntry</th>
<th>MFTSequenceNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>BagMRU</td>
<td>3</td>
<td>5</td>
<td><code>Desktop\aa</code></td>
<td>11/18/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>1</td>
<td>2</td>
<td><code>Desktop\Common Places</code></td>
<td>11/13/2014 3:42:44 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>1</td>
<td>7</td>
<td><code>Desktop\Common Places</code></td>
<td>11/16/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>2</td>
<td>0</td>
<td><code>Desktop\Control Panel</code></td>
<td>11/13/2014 3:42:44 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>2</td>
<td>6</td>
<td><code>Desktop\Control Panel</code></td>
<td>11/16/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>4</td>
<td>4</td>
<td><code>Desktop\Item103</code></td>
<td>11/18/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>6</td>
<td>3</td>
<td><code>Desktop\KnTToolsEnterprise-2.3.0.2898d</code></td>
<td>11/18/2014 6:45:11 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>0</td>
<td>1</td>
<td><code>Desktop\My Computer\My Computer</code></td>
<td>11/13/2014 3:42:44 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>0</td>
<td>0</td>
<td><code>Desktop\My Computer</code></td>
<td>11/16/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>7</td>
<td>1</td>
<td><code>Desktop\Shared Documents Folder (Users Files)</code></td>
<td>11/16/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU</td>
<td>5</td>
<td>2</td>
<td><code>Desktop\\User\Alice</code></td>
<td>11/16/2014 6:45:11 PM +00:00</td>
<td>UsrClass 2.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A0\0</td>
<td>1</td>
<td>3</td>
<td><code>Desktop\My Computer\Users Files</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A0\0</td>
<td>3</td>
<td>1</td>
<td><code>Desktop\My Computer\\voo</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A0\0</td>
<td>2</td>
<td>2</td>
<td><code>Desktop\My Computer\\voo</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A0\0</td>
<td>4</td>
<td>0</td>
<td><code>Desktop\My Computer\\voo\voo</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A0\0</td>
<td>0</td>
<td>0</td>
<td><code>Desktop\My Computer\\voo\voo</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>5</td>
<td>2</td>
<td><code>Desktop\My Computer\\voo\Backup</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>4</td>
<td>5</td>
<td><code>Desktop\My Computer\\Program Files</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>2</td>
<td>2</td>
<td><code>Desktop\My Computer\\Program Files (x86)</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>2</td>
<td>4</td>
<td><code>Desktop\My Computer\\Program Files (x86)</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>3</td>
<td>3</td>
<td><code>Desktop\My Computer\\ProgramData</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>0</td>
<td>0</td>
<td><code>Desktop\My Computer\\Project Working Folder</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
<tr>
<td>BagMRU\A1\1</td>
<td>0</td>
<td>0</td>
<td><code>Desktop\My Computer\\Project Working Folder</code></td>
<td>11/7/2014 10:20:08 PM +00:00</td>
<td>UsrClass 1.dat</td>
<td>327860</td>
<td>36</td>
</tr>
</tbody>
</table>
Looking to the future

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>Microsoft Windows CurrentVersion</td>
</tr>
<tr>
<td>Explorer</td>
<td>ComDlg32 OpenSavePidlMRU</td>
</tr>
<tr>
<td>GUID</td>
<td></td>
</tr>
<tr>
<td>Extension</td>
<td>blocks</td>
</tr>
<tr>
<td>Serialized</td>
<td>property storage</td>
</tr>
</tbody>
</table>

Remember this?

This is a value from one of the keys under

Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU

Does anything stand out now?

- GUID
- Extension blocks
- Serialized property storage

We can exploit these values just like we do ShellBags!
OpenSavePidlMRU values decoded

Value: User Libraries

Signature: 0xbeef0019

GUID 1: fbb3477e-c9e4-4b3b-a2ba-d3f5d3cd46f9 GUID 1 Folder: Documents Library
GUID 2: 82ba0782-5b7a-4569-b5d7-ec83085f08cc GUID 2 Folder: TopViews

Dumping SPS for metaChunk 0
Sheet #0 => Key: 8, Value: C:\Users\e11732\Documents; Key: 3, Value: folder
Dumping SPS for metaChunk 1
Sheet #0 => Key: 10, Value: Suneel; Key: 15, Value: 07/29/2014 14:51:14; Key: 13, Value: 16; Key: 14, Value: 01/09/2015 18:32:36; Key: 16, Value: 01/09/2015 18:32:36
Sheet #0 => Key: 8, Value: C:\Users\e11732\Documents\Suneel; Key: 3, Value: folder
Dumping SPS for metaChunk 2
Sheet #0 => Key: 10, Value: New folder; Key: 15, Value: 12/04/2014 00:16:12; Key: 13, Value: 16; Key: 14, Value: 01/13/2015 18:30:34; Key: 16, Value: 01/13/2015 18:30:34
Sheet #0 => Key: 8, Value: C:\Users\e11732\Documents\Suneel\New folder; Key: 3, Value: folder
Dumping SPS for metaChunk 3
Sheet #0 => Key: 10, Value: 12-12-2014 Master Data - XXXX_AGGREGATE_REPORT_12_17_2014.xlsx; Key: 12, Value: 0; Key: 13, Value: 128
Sheet #0 => Key: 25, Value: 1212678519; Key: 32, Value: VT_VECTOR data not implemented (yet) See extension block section for contents for now; Key: 24, Value: 12-12-2014 Master Data - XXXX_AGGREGATE_REPORT_12_17_2014.xlsx; Key: 11, Value: .xlsx
Sheet #0 => Key: 8, Value: C:\Users\e11732\Documents\Suneel\New folder\12-12-2014 Master Data - XXXX_AGGREGATE_REPORT_12_17_2014.xlsx; Key: 3, Value: document

End SPS for metaChunk 3
Where to go from here?

RECmd.exe --Hive D:\NTUSER.DAT --sd "04-00-EF-BE" --SuppressData

• For ONE extension block, RECmd found hits in the following keys:
  • ..\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU
  • ..\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRULegacy
  • ..\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU
  • ..\CurrentVersion\Explorer\Modules\CommonPlaces
  • ..\CurrentVersion\Explorer\RecentDocs
  • ..\Shell\Bags\1\Desktop

• In one hive, 930 hits were found. Not ONE of these was in ShellBags!
• Similar results were found when looking for SPS blocks
Questions?

ShellBags

The rest of the Registry