Public Research
Influencing Change In DFIR Tools

Dan Pullega
@4n6k

6/9/14
Thanks


(I know I've forgotten some; I apologize to those I've missed)
Outline

• The Shellbags Research Story
  • Why I did it
  • How I did it
  • How you can do it
  • Why you should do it

• The Importance of Public Research
# About Me

<table>
<thead>
<tr>
<th>IT/Networking</th>
<th>InfoSec/Networking</th>
<th>InfoSec/DFIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>FENWICK</td>
<td>DePaul University</td>
<td>Trustwave</td>
</tr>
<tr>
<td></td>
<td>SecDaemons</td>
<td>Deloitte</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4Discovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KPMG</td>
</tr>
</tbody>
</table>

Books/Blogs
Preface

4n6k.com

Shellbags Forensics
Addressing a Misconception

What are they?
How can they help?
How do they work?
INTERPRETATION
Shellbags – What are they? (1)

- **Purpose** is to keep track of Windows Shell (Explorer) window sizes, positions, settings, etc.

- Found in **registry** (per user)

- Entry is created for every newly explored folder.

- Can glean **explored folder’s path, MAC times, time of access, & more.**
Shellbags – What are they? (2)
Shellbags – How can they help?

- Can be used to determine:
  - Folder access (per-user basis)
  - Use of removable storage devices
  - Attacker navigation patterns (if access gained via RDP or other VNC-based vectors)
  - Previously existing folders post-deletion/overwrite.
  - Historical MAC times of folders
  - The means by which a folder was navigated
  - When certain folders were accessed.
Research – Why I Did It

- **I didn’t understand how it worked**
- **A bunch of the “reg times” were the same...**

<table>
<thead>
<tr>
<th>regdate</th>
<th>reg-UTC</th>
<th>full path</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/3/2013</td>
<td>23:40:55.582</td>
<td>Desktop{CLSID_MyComputer}\E:\CONFIDENTIAL\X\</td>
</tr>
<tr>
<td>11/3/2013</td>
<td>23:40:55.582</td>
<td>Desktop{CLSID_MyComputer}\E:\CONFIDENTIAL\Y\</td>
</tr>
<tr>
<td>11/3/2013</td>
<td>23:40:55.582</td>
<td>Desktop{CLSID_MyComputer}\E:\CONFIDENTIAL\Z\</td>
</tr>
</tbody>
</table>

TZWorks sbag v0.33

- **This is how the research project began.**
Research – Why I Did It (Findings)

- Tool Output Misconceptions
  - TZWorks sbag did not perform the following check:

```php
if ($mru != 4294967295 && ($v == $mru)) {
    $item{mrutime} = $key->get_timestamp();
    $item{mrutime_str} = $key->
        get_timestamp_as_string();
    $item{mrutime_str} =~ s/T/ /;
    $item{mrutime_str} =~ s/Z/ /;
}
```

Segment of RegRipper’s shellbags.pl source that determines whether or not to pull the MRU time for given subkey
Research – Why I Did It (Findings)

- **Tool Output Misconceptions**
  - TZWorks sbag & RegRipper’s shellbags.pl
  - sbag did not...
    - check MRU status of entries; multiple entries with same “MRU Time” – huge avenue for misinterpretation
    - identify/parsing MTP device entries; all MTP device entries missing from output
    - parse 0x52 shell items correctly
  - shellbags.pl did not...
    - parse 0x52 shell items correctly
Research – How I Did It (you can too!)

- Start from scratch (assume nothing)
- Compile “known” facts
- Test extensively and granularly
  - Procmon
  - Pull hive, navigate, snapshot, pull hive, analyze
  - Document nature and time of action
  - Do it again; note difference

- DOCUMENT EVERY STEP!
Research – How I Did It (you can too!)

- Reference/cite **ALL** sources
- Reach out to **other researchers**
- Update research with **new developments**
  - Follow up on feedback
- Go **public** with it
  - #DFIR

- **DOCUMENT EVERY STEP!**
Research – How I Did It (you can too!)

VMware Snapshot Manager as a result of shellbags testing
Research – How You Can Do It (1)

- Be practical (use real scenarios)
- Quality over speed/quantity
- Share your [sanitized] data sets
  - ASSUME NOTHING
- Leverage source code and FOSS
  - Tells you exactly what the tool is doing

- Quadruple check your claims.
Research – How You Can Do It (2)

- Using source code effectively

- Easier than low level reversing!

- Full post [here]
Research – Why You Should Do It

• **Tools will get fixed**
  • As a result of public research, both shellbags tools have fixed all reported issues

• **Help other investigators [do the right thing]**
  • Being accurate in this field is crucial

• **Help yourself understand**
  • Come away from it knowing you advanced the field (even if it was in a small way)

• **Show significance of artifact**
  • Others will recognize its importance (e.g. IEF)
Closing Remarks (1)

- **Key Points to Remember:**
  - If you read something or are told something, don’t just blindly believe it. Challenge it and do the work on your own to confirm or deny someone else's claim.
  - Manual testing is valuable; you are only hurting yourself by not knowing how something truly works.
  - Anyone can run a tool. It's knowing what the tool is doing, how to interpret the output, and how to use it effectively that makes you a reliable investigator.
Remember: in the end, you cannot blame a tool. It is the investigator’s job to validate.
Validation & Testing

- **Repeat & Validate Testing**
  - Detailed testing has been performed on shellbags artifacts.
  - The full data sets for this testing have been made available for download in the [original blog post](#).
  - Following along with each step will help in understanding exactly how this artifact works.
Closing Remarks (2)

- **Final Items:**
  - This scratches the surface of shellbags forensics. To get the full story, see the original blog post here:
    - [http://www.4n6k.com/2013/12/shellbags-forensics-addressing.html](http://www.4n6k.com/2013/12/shellbags-forensics-addressing.html)

- Thank you!
Contact

@4n6k
danpullega.com
4n6k.com
References

01. Computer Forensic Artifacts: Windows 7 Shellbags (by Chad Tilbury)
02. Windows Shellbag Forensics (by Willi Ballenthin)
03. Windows Shell Item Format v0.0.21 (by Joachim Metz)
04. Using Shellbag Information to Reconstruct User Activities (by Yuandong Zhu)
05. Windows Forensic Analysis Toolkit 3E (by Harlan Carvey)
06. ShellBag Analysis (by Harlan Carvey)
07. Shellbag Analysis, Revisited...Some Testing (by Harlan Carvey)
08. Shell Item Artifacts, Reloaded (by Harlan Carvey)
09. Windows Shellbag Parser: sbag (by TZWorks)
10. MoVP 3.2 Shellbags in Memory, SetRegTime, & TrueCrypt Volumes (by Jamie Levy)
11. Shellbags Description (by Microsoft KB)
12. NTFS INDX Parsing (by Willi Ballenthin)
13. RecentDocs Artifact (by Joe Garcia)
14. MTP Device Shellbags (by Nicole Ibrahim)