Linux Memory Forensics: Expanding Rekall Userland Investigation

Johannes Stadlinger*, Frank Block*†, Andreas Dewald*‡

*Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Erlangen, Germany
†ERNW GmbH, Heidelberg, Germany
‡ERNW Research GmbH, Heidelberg, Germany

May 8, 2018
Agenda

Motivation

Background

Goals

Analysis and Plugins

Evaluation

Conclusion
Motivation
Motivation

- Importance and relevance of Memory Forensics is growing [2], [5].

- Most of the previous publications were focusing on kernel specific data (e.g., network connections, running processes, etc.).

- Such information are extractable by known tools like Rekall or Volatility.

- Only a few approaches handling the userspace.

- **However:**
  The Userspace has not yet received that much attention. It also may include data that might be of forensic interest – especially the Heap:
  - Command History
  - Hostnames
  - Username, Passwords
  - …
Background
Background – Former approaches

• RAM as big bulk of data → Pattern-Search Techniques
e.g., bash- oder cmdscan-Plugin von Rekall [7], [8]

• More advanced: Isolate special heap-chunks of certain processes.
e.g., Volatility-Plugin focusing on Notepad by Ligh et. al. [4]

• Cohen [1]:
  • Target: Windows
  • New approach: Knowledge about inner heap structures → New perspectives
  • Applied in Plugins (Volatility): z.B. DNS Client Resolver.

• Block and Dewald [3]:
  • Target: Linux and glibc
  • Analysis of internal structure
  • Development of several Plugins for Rekall (HeapAnalysis).

⇒ Basis of our work
Background – HeapAnalysis-Plugins [3]

- **heapinfo**: Returns statistics about all available chunks.
- **heapdump**: Dumps all chunks into separate files on the local system.

- **heapsearch**: Searches all chunks for strings, pointers, or regex-expression. It is also possible to provide specific addresses of chunks:

```
  Chunk #1  →  Pointer to  →  Target Chunk  →  Pointer to  →  Chunk #2
         |                      |                      |
         |                      |                      |
         |                      |                      |
  Chunk #1  →  Pointer zu  →  Target Chunk  →  Pointer zu  →  Chunk #2
```

- **heaprefs**: Returns all chunks the current chunk contains a reference to:

```
  Chunk #1  ←  Pointer zu  ←  Target Chunk  ←  Pointer zu  ←  Chunk #2
         |                      |                      |
         |                      |                      |
         |                      |                      |
```
Background – HeapAnalysis-Plugins [3]

- **heapinfo**: Returns statistics about all available chunks.
- **heapdump**: Dumps all chunks into separate files on the local system.
- **heapsearch**: Searches all chunks for strings, pointers, or regex-expression. It is also possible to provide specific addresses of chunks:

  ![Diagram](diagram.png)

- **heapprofs**: Returns all chunks the current chunk contains a reference to:

  ![Diagram](diagram2.png)
Goals
Goals

• Focus on Linux Userspace applications.

• Show that the heap indeed contains information of forensic interest (e.g., credentials, history, etc.).

• The examiners should be able to extract information from certain applications without any deeper knowledge about their inner structures.

• Apply and continue the work of Frank Block.
Goals – Concrete

Analyse:

- What data is available?
- How is it structured??
- Where is it stored inside the heap?

Afterwards: Implementation and Deployment of several plugins for the Rekall Framework on the basis of the HeapAnalysis-class.

The following application were analyzed:

- cUrl
- gnome-keyring-d
- seahorse
- ssh
- sshfs
- sqlite
- pwsafe
- owncloud
Goals – Concrete

Analyse:

• *What* data is available?
• *How* is it structured??
• *Where* is it stored inside the heap?

Afterwards: *Implementation* and *Deployment* of several plugins for the Rekall Framework on the basis of the HeapAnalysis-class.

The following application were analyzed:

• cUrl
• gnome-keyring-d
• seahorse
• ssh
• sshfs
• sqlite
• pwsafe
• owncloud
Analysis and Plugins
Analysis and Plugins

Approach for each application

1. Detection: "chunks of interest"
   - heapsearch: string, regex
   - heapdump
   - strings

2. Adjacence of the chunk/structure
   - heapsearch: chunk_addresses
   - heaprefs: chunk_addresses

3. Detection of patterns/starting points

4. Implementation
Plugin 1: curl

Desired data

• Username
• Password

Existing data

• Username
• Password
• filename of output
• URL

<table>
<thead>
<tr>
<th>pid</th>
<th>url</th>
<th>output</th>
<th>user</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1068</td>
<td><a href="https://pool.c0nf.de/curl/2Gb.dat">https://pool.c0nf.de/curl/2Gb.dat</a></td>
<td>outputdummy.file</td>
<td>mem_user</td>
<td>mem_password</td>
</tr>
</tbody>
</table>
Plugin 1: curl

**Desired data**
- Username
- Password

**Existing data**
- Username
- Password
- filename of output
- URL

<table>
<thead>
<tr>
<th>pid</th>
<th>url</th>
<th>output</th>
<th>user</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1068</td>
<td><a href="https://pool.conf.de/curl/2Gb.dat">https://pool.conf.de/curl/2Gb.dat</a></td>
<td>outputdummy.file</td>
<td>mem_user</td>
<td>mem_password</td>
</tr>
</tbody>
</table>
## Plugin 2: gnome_keyring

### Desired data
- Master-Password
- Single password entries

### Existing data
- Meta-information about keyrings
- Name of each password entry
- SSH private keys

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>name</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>989</td>
<td>1</td>
<td>nebenring</td>
<td>Keyring</td>
<td>Entries in total: 3</td>
</tr>
<tr>
<td>989</td>
<td>2</td>
<td>newring</td>
<td>Keyring</td>
<td>Entries in total: 20</td>
</tr>
<tr>
<td>989</td>
<td>3</td>
<td>hauptring</td>
<td>Keyring</td>
<td>Entries in total: 6</td>
</tr>
</tbody>
</table>

Recovered name of keyrings with the numbers of entries it contains

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>name</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>989</td>
<td>1</td>
<td>entryentryentryentry-1</td>
<td>Stored Note</td>
<td>Number in keyring: 1</td>
</tr>
<tr>
<td>989</td>
<td>2</td>
<td>entryentryentryentry-2</td>
<td>Stored Note</td>
<td>Number in keyring: 2</td>
</tr>
<tr>
<td>989</td>
<td>3</td>
<td>entryentryentryentry-3</td>
<td>Stored Note</td>
<td>Number in keyring: 3</td>
</tr>
</tbody>
</table>

Recovered name of keyring entries

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>name</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1002</td>
<td>1</td>
<td>t.b.d</td>
<td>Private SSH key</td>
<td>-----BEGIN RSA PRIVATE KEY-----</td>
</tr>
</tbody>
</table>

Recovered Private SSH keys (ASCII armored)
Plugin 2: gnome_keyring

Desired data

• Master-Password
• Single password entries

Existing data

• Meta-information about keyrings
• Name of each password entry
• SSH private keys

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>name</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>989</td>
<td>1</td>
<td>nebenring</td>
<td>Keyring</td>
<td>Entries in total: 3</td>
</tr>
<tr>
<td>989</td>
<td>2</td>
<td>newring</td>
<td>Keyring</td>
<td>Entries in total: 20</td>
</tr>
<tr>
<td>989</td>
<td>3</td>
<td>hauptring</td>
<td>Keyring</td>
<td>Entries in total: 6</td>
</tr>
</tbody>
</table>

Recovered name of keyrings with the numbers of entries it contains

Recovered name of keyring entries

Recovered Private SSH keys (ASCII armored)

1002 1 t.b.d Private SSH key -----BEGIN RSA PRIVATE KEY------
...
Plugin 3: seahorse

Desired data

- Master-Password
- Single password entries

Existing data

- Name of each password entry (Stored Notes)
- PGP Key details
  - Mail
  - Name
  - Note
  - SHA-1 Fingerprints
- SSH Key details
  - Fingerprint
  - Name
  - File paths
  - Public Key
Plugin 3: seahorse

Desired data

- Master-Password
- Single password entries

Existing data

- Name of each password entry *(Stored Notes)*
- PGP Key details
  - Mail
  - Name
  - Note
  - SHA-1 Fingerprints
- SSH Key details
  - Fingerprint
  - Name
  - File paths
  - Public Key
## Plugin 3: seahorse

<table>
<thead>
<tr>
<th>entry</th>
<th>name</th>
<th>type</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name of password entries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>github</td>
<td>Stored Note</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>pwentry-5</td>
<td>Stored Note</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PGP keys</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><a href="mailto:hans.w@exam.com">hans.w@exam.com</a></td>
<td>Mail</td>
<td><a href="mailto:hans.w@exam.com">hans.w@exam.com</a></td>
</tr>
<tr>
<td>1.1</td>
<td></td>
<td>Name</td>
<td>Hans Wurst</td>
</tr>
<tr>
<td>1.2</td>
<td></td>
<td>Note</td>
<td>test</td>
</tr>
<tr>
<td>1.3</td>
<td></td>
<td>Priv-SHA</td>
<td>3089E99B1599C2E894485B01231C331E48E854F6</td>
</tr>
<tr>
<td>1.4</td>
<td></td>
<td>Pub-SHA</td>
<td>66DD35661FE1695B92F5BBFD2DB18518A1A1F61F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSH keys</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td></td>
<td>Name</td>
<td><a href="mailto:test@test.com">test@test.com</a></td>
</tr>
<tr>
<td>1.2</td>
<td></td>
<td>Path Private</td>
<td>/home/user/.ssh/id_rsa</td>
</tr>
<tr>
<td>1.3</td>
<td></td>
<td>Path Public</td>
<td>/home/user/.ssh/id_rsa.pub</td>
</tr>
<tr>
<td>1.4</td>
<td></td>
<td>Public Key</td>
<td>ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAAQDcmCvR7Rrq</td>
</tr>
</tbody>
</table>
Plugin 4: ssh

Desired data

- Username, Password
- Key(-fragments)
- Command History

Existing data

- Username, Hostname
- IP-Addresses

<table>
<thead>
<tr>
<th>pid</th>
<th>username</th>
<th>source</th>
<th>hostname</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1074</td>
<td>mem_test</td>
<td>10.0.2.15</td>
<td>c0nf.de</td>
<td>188.68.50.8</td>
</tr>
</tbody>
</table>
Plugin 4: ssh

Desired data
- Username, Password
- Key(-fragments)
- Command History

Existing data
- Username, Hostname
- IP-Addresses

<table>
<thead>
<tr>
<th>pid</th>
<th>username</th>
<th>source</th>
<th>hostname</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1074</td>
<td>mem_test</td>
<td>10.0.2.15</td>
<td>c0nf.de</td>
<td>188.68.50.8</td>
</tr>
</tbody>
</table>
Plugin 5: sshfs

Desired data

- Username, Password
- Filelist

Existing data

- Filelist (partial)
- Username, Hostname
- folderpath of the server and clients (partial)

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>name</th>
<th>username</th>
<th>hostname</th>
<th>folder_server</th>
<th>folder_local</th>
</tr>
</thead>
<tbody>
<tr>
<td>1112</td>
<td>1</td>
<td>/</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>2</td>
<td>/.</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>3</td>
<td>/..</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>4</td>
<td>/.aptitude</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>5</td>
<td>/bash_history</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1112</td>
<td>28</td>
<td>/git.pub</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>29</td>
<td>/hereuare.txt</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>30</td>
<td>/letsencrypt</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>31</td>
<td>/owntmp</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>32</td>
<td>/owntmp/..</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>33</td>
<td>/owntmp/..</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
</tbody>
</table>
Plugin 5: sshfs

Desired data
- Username, Password
- Filelist

Existing data
- Filelist (partial)
- Username, Hostname
- folderpath of the server and clients (partial)

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>name</th>
<th>username</th>
<th>hostname</th>
<th>folder_server</th>
<th>folder_local</th>
</tr>
</thead>
<tbody>
<tr>
<td>1112</td>
<td>1</td>
<td>/</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>2</td>
<td>./</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>3</td>
<td>../</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>4</td>
<td>./aptitude</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td>1112</td>
<td>5</td>
<td>./bash_history</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>/git.pub</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>/hereuare.txt</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>/letsencrypt</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>/owntmp</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>/owntmp/</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>/owntmp/..</td>
<td>mem_test</td>
<td>c0nf.de</td>
<td>/home/mem_test</td>
<td>/home/user/tmp</td>
</tr>
</tbody>
</table>
Plugin 6: pwsafe

Desired data

• Master-Password
• Username
• Password
• Title

Existing data

• Username
• Password (!)
• Title
• Group

<table>
<thead>
<tr>
<th>entry</th>
<th>group</th>
<th>title</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Personal</td>
<td>Facebook Copy # 9</td>
<td>hans.wurst</td>
<td>ananas</td>
</tr>
<tr>
<td>15</td>
<td>School</td>
<td>MyCampus</td>
<td>hansw</td>
<td>password123</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>School</td>
<td>MyUni Copy # 9</td>
<td>unishort</td>
<td>secret123</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Plugin 6: pwsafe

#### Desired data
- Master-Password
- Username
- Password
- Title

#### Existing data
- Username
- Password (!)
- Title
- Group

<table>
<thead>
<tr>
<th>entry</th>
<th>group</th>
<th>title</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task: pwsafe (1198)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[...]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Personal</td>
<td>Facebook Copy # 9</td>
<td>hans.wurst</td>
<td>ananas</td>
</tr>
<tr>
<td>15</td>
<td>School</td>
<td>MyCampus</td>
<td>hansw</td>
<td>password123</td>
</tr>
<tr>
<td>[...]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>School</td>
<td>MyUni Copy # 9</td>
<td>unishort</td>
<td>secret123</td>
</tr>
</tbody>
</table>
Plugin 7: sqlite

 Desired data

• Command History

Existing data

• Command History
• For each table
  Complete scheme

---

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>time</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1262</td>
<td>1</td>
<td>2017-08-29 10:16:09Z</td>
<td>quit</td>
</tr>
<tr>
<td>1262</td>
<td>2</td>
<td>2017-08-29 10:16:09Z</td>
<td>;</td>
</tr>
<tr>
<td>1262</td>
<td>3</td>
<td>2017-08-29 10:16:09Z</td>
<td>q</td>
</tr>
<tr>
<td>1262</td>
<td>4</td>
<td>2017-08-29 10:16:09Z</td>
<td>;</td>
</tr>
<tr>
<td>1262</td>
<td>5</td>
<td>2017-08-29 10:16:13Z</td>
<td>.help</td>
</tr>
<tr>
<td>1262</td>
<td>6</td>
<td>2017-08-29 10:19:42Z</td>
<td>.tables</td>
</tr>
</tbody>
</table>

-------------------------------

Extracted Tables:

-------------------------------

Table 1: djcelery_workerstate

-------------------------------

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
</tr>
<tr>
<td>2</td>
<td>hostname</td>
</tr>
<tr>
<td>3</td>
<td>last_heartbeat</td>
</tr>
</tbody>
</table>

...
Plugin 7: sqlite

Desired data

• Command History

Existing data

• Command History

• For each table
  Complete scheme

<table>
<thead>
<tr>
<th>pid</th>
<th>entry</th>
<th>time</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1262</td>
<td>1</td>
<td>2017-08-29 10:16:09Z</td>
<td>quit</td>
</tr>
<tr>
<td>1262</td>
<td>2</td>
<td>2017-08-29 10:16:09Z</td>
<td>;</td>
</tr>
<tr>
<td>1262</td>
<td>3</td>
<td>2017-08-29 10:16:09Z</td>
<td>q</td>
</tr>
<tr>
<td>1262</td>
<td>4</td>
<td>2017-08-29 10:16:09Z</td>
<td>;</td>
</tr>
<tr>
<td>1262</td>
<td>5</td>
<td>2017-08-29 10:16:13Z</td>
<td>.help</td>
</tr>
<tr>
<td>1262</td>
<td>6</td>
<td>2017-08-29 10:19:42Z</td>
<td>.tables</td>
</tr>
</tbody>
</table>

Extracted Tables:

Table 1: djcelery_workerstate

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>hostname</td>
<td>varchar (255)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>last_heartbeat</td>
<td>datetime</td>
<td></td>
</tr>
</tbody>
</table>

[... ]
Plugin 8: owncloud

Desired data

- Username, Password
- Hostname

Existing data

- Username and Password
- Hostname
- Sync-Protocols
  - Timestamp, Filename
  - Folder, Action

<table>
<thead>
<tr>
<th>entry</th>
<th>time</th>
<th>file</th>
<th>folder</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2017-07-16 19:44:25</td>
<td>Documents/Example.odt</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>3</td>
<td>2017-07-16 19:44:25</td>
<td>Photos/Squirrel.jpg</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>4</td>
<td>2017-07-16 19:44:25</td>
<td>Photos/San Francisco.jpg</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>5</td>
<td>2017-07-16 19:44:25</td>
<td>Photos/Paris.jpg</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>6</td>
<td>2017-07-16 19:44:24</td>
<td>Documents</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>7</td>
<td>2017-07-16 19:44:24</td>
<td>Photos</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
</tbody>
</table>

Hostname: https://cloud.c0nf.de
Username: mem_test
Password: mem_password
Plugin 8: owncloud

Desired data
- Username, Password
- Hostname

Existing data
- Username and Password
- Hostname
- Sync-Protocols
  - Timestamp, Filename
  - Folder, Action

<table>
<thead>
<tr>
<th>entry</th>
<th>time</th>
<th>file</th>
<th>folder</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2017-07-16 19:44:25</td>
<td>Documents/Example.odt</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>3</td>
<td>2017-07-16 19:44:25</td>
<td>Photos/Squirrel.jpg</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>4</td>
<td>2017-07-16 19:44:25</td>
<td>Photos/San Francisco.jpg</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>5</td>
<td>2017-07-16 19:44:25</td>
<td>Photos/Paris.jpg</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>6</td>
<td>2017-07-16 19:44:24</td>
<td>Documents</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
<tr>
<td>7</td>
<td>2017-07-16 19:44:24</td>
<td>Photos</td>
<td>ownCloud</td>
<td>Downloaded</td>
</tr>
</tbody>
</table>

Hostname: https://cloud.conf.de
Username: mem_test
Password: mem_password
Plugin 8: owncloud – Structure

Figure: OwnCloud: Structure to receive one entry of the sync-protocol.
Evaluation
Evaluation

- Test environment:
  - Debian "stretch" 32 bit, Kernel Version 4.9.30-2+deb2u5
  - ArchLinux 64 bit, Kernel Version 4.4-66
  - glibc-version: 2.24 and 2.25 (2.27: started)
- Simulate certain user actions for all applications (including special cases)
- Check the results for correctness and completeness.
Evaluation – Performance

![Bar Chart]

- Time in s

- 32-bit vs 64-bit

- Tools:
  - curl
  - ssh
  - sshfs
  - gnome_keyring
  - seahorse
  - pwsafe
  - sqlite
  - owncloud

- May 8, 2018
Conclusion
Conclusion

• A lot of information could be found in the heap that is of forensic interest.
• The work of Block and Dewald could be utilized for further application.
• The developed Tools support the forensic examiners to extract data from the heap.
• Plugins support 32- and 64-bit.
• Expandable for further versions.
Conclusion

Limitations

- Volume of the heap might differ from application to application (e.g., ssh vs. owncloud)
- Results of the password managers are very limited. Concrete passwords are hardly extractable.
- Different versions of the applications.
- Performance for graphical user interfaces.
- Missing connections between data (e.g., gnome_keyring: ssh-keys)

Future Work

- Pull-Request for the official Rekall Master-Branch (in progress)
- Improve existing plugins (Performance, versions, etc.).
- Focus on other applications: Analyse and Implementation of further plugins.
Thank you for your attention!
Referenzen
Referenzen I


Referenzen II


Referenzen III


