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., Volalitlity-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`.

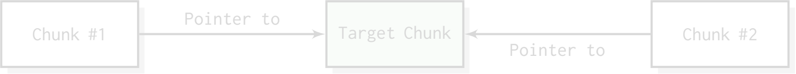
- Block and Dewald [3]:
  - Target: Linux and glibc
  - Analysis of internal structure
  - Development of several Plugins for Rekall (*HeapAnalysis*).

  ⇒ **Basis of our work**

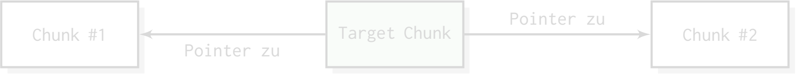
Background – HeapAnalysis-Plugins [3]

- **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:

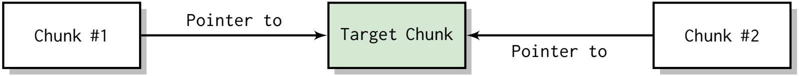


- **heaprefs**: Returns all chunks the current chunk contains a reference to:

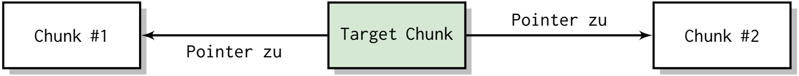


Background – HeapAnalysis-Plugins [3]

- **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:



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



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 knowlege 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

```
 pid                    url                      output          user       password
------ ----------------------------------- -----------------  ---------- --------------
 1068  https://pool.c0nf.de/curl/2Gb.dat   outputdummy.file   mem_user   mem_password
```

Plugin 1: `curl`

Desired data

- Username
- Password

Existing data

- Username
- Password
- filename of output
- URL

```
 pid                    url                    output          user       password
------ ----------------------------------- -----------------  ---------- --------------
 1068  https://pool.c0nf.de/curl/2Gb.dat   outputdummy.file   mem_user   mem_password
```

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

```
   pid   entry                name                   type                  value
  ------ ----- ---------------------------- ------------------ -------------------------
------------------------------------------------------------------
Recovered name of keyrings with the numbers of entries it contains
------------------------------------------------------------------
   989      1   nebenring                    Keyring            Entries in total: 3
   989      2   newring                      Keyring            Entries in total: 20
   989      3   hauptring                    Keyring            Entries in total: 6
---------------------------------
Recovered name of keyring entries
---------------------------------
   989      1   entryentryentryentry-1       Stored Note        Number in keyring: 1
   989      2   entryentryentryentry-2       Stored Note        Number in keyring: 2
   989      3   entryentryentryentry-3       Stored Note        Number in keyring: 3
------------------------------------------
Recovered Private SSH keys (ASCII armored)
------------------------------------------
   1002     1   t.b.d       Private SSH key    -----BEGIN RSA PRIVATE KEY-----
                                               ...
```

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

```
   pid   entry               name                    type                value
  ------ ----- ---------------------------- ------------------ -------------------------
------------------------------------------------------------------
Recovered name of keyrings with the numbers of entries it contains
------------------------------------------------------------------
   989    1    nebenring                    Keyring            Entries in total: 3
   989    2    newring                      Keyring            Entries in total: 20
   989    3    hauptring                    Keyring            Entries in total: 6
---------------------------------
Recovered name of keyring entries
---------------------------------
   989    1    entryentryentryentry-1       Stored Note        Number in keyring: 1
   989    2    entryentryentryentry-2       Stored Note        Number in keyring: 2
   989    3    entryentryentryentry-3       Stored Note        Number in keyring: 3
------------------------------------------
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

```
 entry          name           type                         content
 ----- ------------------ -------------- ------------------------------------------------
------------------------
Name of password entries
------------------------
  1     github                 Stored Note
[...]
  6     pwentry-5              Stored Note
--------
PGP keys
--------
  1     hans.w@exam.com
  1.1                          Mail           hans.w@exam.com
  1.2                          Name           Hans Wurst
  1.3                          Note           test
  1.4                          Priv-SHA       3089E99B1599C2E894485B01231C331E48E854F6
  1.5                          Pub-SHA        66DD35661FE1695B92F5BBFD2DB18518A1A1F61F
--------
SSH keys
--------
  1     test@test.com
  1.1                          Fingerprint    b1:fd:2b:9b:62:ba:f7:ec:44:a6:c2:20:b2:85:fa:58
  1.2                          Name           test@test.com
  1.3                          Path Private   /home/user/.ssh/id_rsa
  1.4                          Path Public    /home/user/.ssh/id_rsa.pub
  1.5                          Public Key     ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDcmCvR7Rrq
                                              ...
```

Plugin 4: ssh

Desired data

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

Existing data

- Username, Hostname
- IP-Addresses

```
 pid      username          source           hostname        destination
------ ----------------- ---------------- ----------------- ----------------
 1074  mem_test           10.0.2.15        c0nf.de           188.68.50.8
```

Plugin 4: ssh

Desired data

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

Existing data

- Username, Hostname
- IP-Addresses

```
  pid       username          source          hostname        destination
------ ---------------- --------------- ---------------- ---------------
  1074   mem_test         10.0.2.15       c0nf.de          188.68.50.8
```

Plugin 5: sshfs

Desired data

- Username, Password
- Filelist

Existing data

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

```
 pid   entry        name          username    hostname      folder_server     folder_local
------ ----- ---------------- ----------- ------------- ---------------- ---------------
 1112    1   /                mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    2   /.               mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    3   /..              mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    4   /.aptitude       mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    5   /.bash_history   mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 ...
 1112   28   /git.pub         mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   29   /hereuare.txt    mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   30   /letsencrypt     mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   31   /owntmp          mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   32   /owntmp/.        mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   33   /owntmp/..       mem_test    c0nf.de       /home/mem_test    /home/user/tmp
```

Plugin 5: `sshfs`

Desired data

- Username, Password
- Filelist

Existing data

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

```
 pid   entry         name          username    hostname     folder_server    folder_local
------ ----- ---------------- ----------- ------------- ---------------- ---------------
 1112    1    /                mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    2    /.               mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    3    /..              mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    4    /.aptitude       mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112    5    /.bash_history   mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 ...
 1112   28    /git.pub         mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   29    /hereuare.txt    mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   30    /letsencrypt     mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   31    /owntmp          mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   32    /owntmp/.        mem_test    c0nf.de       /home/mem_test    /home/user/tmp
 1112   33    /owntmp/..       mem_test    c0nf.de       /home/mem_test    /home/user/tmp
```

Plugin 6: `pwsafe`

Desired data

- Master-Password
- Username
- Password
- Title

Existing data

- Username
- Password (!)
- Title
- Group

```
  entry   group          title                username              password
  ----- ---------- -------------------- -------------------- --------------------
-------------------
Task: pwsafe (1198)
-------------------
[...]
   14   Personal   Facebook Copy # 9    hans.wurst           ananas
   15   School     MyCampus             hansw                password123
[...]
   42   School     MyUni Copy # 9       unishort             secret123
```

Plugin 6: `pwsafe`

Desired data

- Master-Password
- Username
- Password
- Title

Existing data

- Username
- Password (!)
- Title
- Group

```
  entry   group          title               username             password
  ----- ---------- -------------------- -------------------- --------------------
-------------------
Task: pwsafe (1198)
-------------------
[...]
   14    Personal   Facebook Copy # 9    hans.wurst           ananas
   15    School     MyCampus             hansw                password123
[...]
   42    School     MyUni Copy # 9       unishort             secret123
```

Plugin 7: `sqlite`

Desired data

- Command History

Existing data

- Command History
- For each table
  Complete scheme

```
   pid   entry           time                      command
  ------ ----- --------------------- ------------------------
   1262    1   2017-08-29 10:16:09Z  quit
   1262    2   2017-08-29 10:16:09Z  ;
   1262    3   2017-08-29 10:16:09Z  q
   1262    4   2017-08-29 10:16:09Z  ;
   1262    5   2017-08-29 10:16:13Z  .help
   1262    6   2017-08-29 10:19:42Z  .tables
-----------------
Extracted Tables:
-----------------
-----------------------------
Table 1: djcelery_workerstate
-----------------------------
           1   id                    integer
           2   hostname              varchar(255)
           3   last_heartbeat        datetime
[...]
```

Plugin 7: `sqlite`

Desired data

- Command History

Existing data

- Command History
- For each table
  Complete scheme

```
   pid   entry          time                     command
  ------ ----- --------------------- ------------------------
   1262    1   2017-08-29 10:16:09Z  quit
   1262    2   2017-08-29 10:16:09Z  ;
   1262    3   2017-08-29 10:16:09Z  q
   1262    4   2017-08-29 10:16:09Z  ;
   1262    5   2017-08-29 10:16:13Z  .help
   1262    6   2017-08-29 10:19:42Z  .tables
-----------------
Extracted Tables:
-----------------
-----------------------------
Table 1: djcelery_workerstate
-----------------------------
           1   id                    integer
           2   hostname              varchar(255)
           3   last_heartbeat        datetime
[...]
```

Plugin 8: `owncloud`

Desired data

- Username, Password
- Hostname

Existing data

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

```
  entry         time                          file                   folder      action
  ----- -------------------- ------------------------------ ------------ ----------
------------------------------
Hostname: https://cloud.c0nf.de
Username: mem_test
Password: mem_password
------------------------------
  1     2017-07-16 19:44:28  ownCloud Manual.pdf            ownCloud     Downloaded
  2     2017-07-16 19:44:25  Documents/Example.odt          ownCloud     Downloaded
  3     2017-07-16 19:44:25  Photos/Squirrel.jpg            ownCloud     Downloaded
  4     2017-07-16 19:44:25  Photos/San Francisco.jpg       ownCloud     Downloaded
  5     2017-07-16 19:44:25  Photos/Paris.jpg               ownCloud     Downloaded
  6     2017-07-16 19:44:24  Documents                      ownCloud     Downloaded
  7     2017-07-16 19:44:24  Photos                         ownCloud     Downloaded
```

Plugin 8: `owncloud`

Desired data

- Username, Password
- Hostname

Existing data

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

```
  entry        time                          file                 folder      action
  ----- -------------------- -------------------------------- ------------- ----------
-------------------------------
Hostname: https://cloud.c0nf.de
Username: mem_test
Password: mem_password
-------------------------------
  1     2017-07-16 19:44:28  ownCloud Manual.pdf              ownCloud      Downloaded
  2     2017-07-16 19:44:25  Documents/Example.odt            ownCloud      Downloaded
  3     2017-07-16 19:44:25  Photos/Squirrel.jpg              ownCloud      Downloaded
  4     2017-07-16 19:44:25  Photos/San Francisco.jpg         ownCloud      Downloaded
  5     2017-07-16 19:44:25  Photos/Paris.jpg                 ownCloud      Downloaded
  6     2017-07-16 19:44:24  Documents                        ownCloud      Downloaded
  7     2017-07-16 19:44:24  Photos                           ownCloud      Downloaded
```

Plugin 8: owncloud – Structure

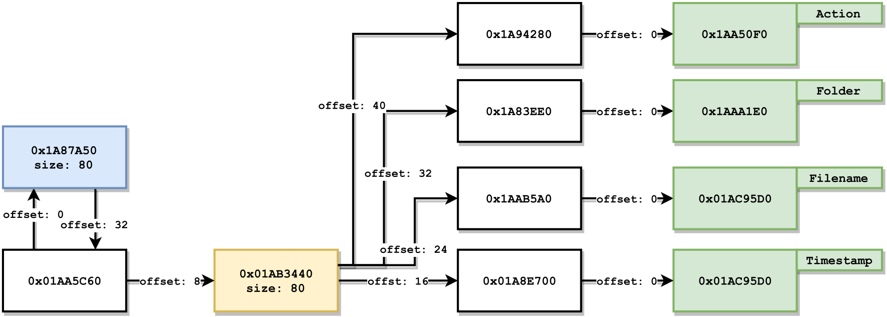


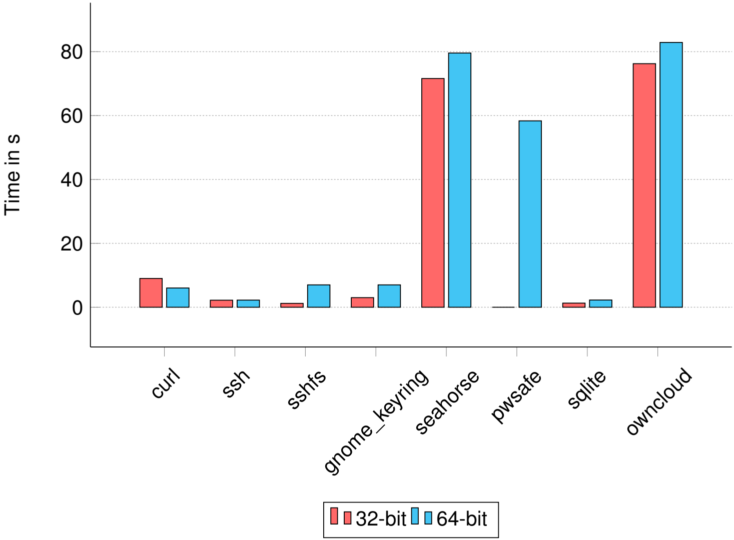
Figure: OwnCloud: Structure to receive one entry of the sync-protocol.

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



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!

Questions? Feedback? Suggestions? Criticism?

Referenzen

Referenzen I

[1] M. Cohen, “Forensic analysis of windows user space applications through heap allocations”, in *Computers and Communication (ISCC), 2015 IEEE Symposium on*, IEEE, 2015, pp. 237–244.

[2] A. Aljaedi, D. Lindskog, P. Zavarsky, R. Ruhl, and F. Almari, “Comparative analysis of volatile memory forensics: Live response vs. memory imaging”, in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, IEEE, 2011, pp. 1253–1258.

[3] F. Block and A. Dewald, “Linux memory forensics: Dissecting the user space process heap”, Friedrich-Alexander-Universität Erlangen-Nürnberg, Dept. of Computer Science, Tech. Rep. CS-2017-02, Apr. 2017.

[4] M. H. Ligh, A. Case, J. Levy, and A. Walters, *The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory*. John Wiley & Sons, 2014.

Referenzen II

[5] E. Casey, *Digital evidence and computer crime: Forensic science, computers, and the internet*. Academic press, 2011.

[6] G. Inc., *Rekall memory forensic framework*, http://www.rekall-forensic.com/, [Online; accessed 16-May-2017].

[7] ——,*Rekall: Scan the bash process for history.* [Online; accessed 16-May-2017]. [Online]. Available: http://www.rekall-forensic.com/docs/Manual/Plugins/Linux/#bash.

[8] ——,*Rekall: Extract command history*, [Online; accessed 16-May-2017]. [Online]. Available: http://www.rekall-forensic.com/docs/Manual/Plugins/Windows/#cmdscan.

[9] F. S. Foundation, *The gnu c library*, [Online; accessed 16-May-2017]. [Online]. Available: https://www.gnu.org/software/libc/.

[10] T. V. Foundation, *Volatility*, http://www.volatilityfoundation.org/, [Online; accessed 16-May-2017].

Referenzen III

[11] J. N. Ferguson, “Understanding the heap by breaking it”, *black Hat USA*, pp. 1–39, 2007.

[12] F. Adelstein, “Live forensics: Diagnosing your system without killing it first”, *Communications of the ACM*, vol. 49, no. 2, pp. 63–66, 2006.

[13] S. L. Garfinkel, “Digital forensics research: The next 10 years”, *digital investigation*, vol. 7, S64–S73, 2010.

[14] A. Case, L. Marziale, C. Neckar, and G. G. Richard, “Treasure and tragedy in kmem_cache mining for live forensics investigation”, *Digital Investigation*, vol. 7, S41–S47, 2010.