Studying copy-on-read and copy-on-write techniques on block device level to aid in large environment forensics

E. van den Haak

System and Network Engineering
University of Amsterdam

Master Thesis, July 2014
Background

Forensics on cloud solutions and large environments

- Sheer volume of data
- (Remote) Acquisition is very hard
  - Making a copy of all data is impossible
  - Making data available remotely is a long procedure
Focus on server block device level
- Copy only relevant data to local storage
  - Copy-on-Read
- Enable live forensics without interfering with original block device
  - Copy-on-Write

Important aspects
- Data integrity
- Reproducible
- Storable
What is a good way to mount block devices read only and store read and changed data in separate sparse files?

- What methods exist that allow copy-on-write and copy-on-read on block device level?
- Can these methods be effectively used to do remote data acquisition while storing read- and changed data locally?
- If necessary, how can an existing method be modified in order to meet the requirements of this research?
Related Research

- Forensic mount tool Xmount[1]
- NIST Cloud Computing Forensic Science Challenges[2]
Existing methods

Methods that either support copy-on-read or copy-on-write

- Xmount
- Fusecow
- Bcache
Ideal situation

Block layer

CoRaW layer

User layer

Block device

CoRaW device

User

CoR File (evidence)

CoW File

Read(b)

Write(p, v)

Read(b)

NULL

NULL

Return(v)

Return(v)
Both Xmount and Fusecow

- Open source
- C
- GPL

Scope

- Copy-on-read file
- Read only feature

\[^1\]Sources on github[3]
Copy-on-write implementation (existing)

```
write(3,X); write(D,Z); write(8,Y)
```

- Fusecow has two separate files
- Xmount puts bitmap into header of CoW file
Copy-on-read implementation

```
read(1); read(E); read(3)
```
Copy-on-read implementation remount

/dev/zero

<table>
<thead>
<tr>
<th>0-4</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A-E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bitmap

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>1</td>
</tr>
</tbody>
</table>

Copy-on-read file

B O D

Virtual block device

<table>
<thead>
<tr>
<th>0-4</th>
<th>B</th>
<th>0</th>
<th>D</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A-E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>O</td>
</tr>
</tbody>
</table>
Test setup

- CoW
- CoR
- Evidence
- Mount
- Live Acquisition
- Block device (OS)
- CoRaW device
- QEMU
Test setup

/dev/zero

Mount

CoW

CoW

CoR

Run

Evidence

Block device (OS)

Mount

CoRaW device

Live Acquisition

QEMU
Second test setup

1. Block device
   
2. Block device
   
3. /dev/zero
   
   - Mount
   - hash(read(x))
   - Evidence
   - Compare results

CoR

CoRaW device

DD
Results

QEMU

- Fusecoraw works flawless

DD
Results

QEMU

- Fusecoraw works flawless
- Xmount has trouble remounting as it performs lots of tests

DD
Results

QEMU

- Fusecoraw works flawless
- Xmount has trouble remounting as it performs lots of tests
  - For now Read only or Copy-on-Read file as Copy-on-Write file

DD
Results

QEMU

- Fusecoraw works flawless
- Xmount has trouble remounting as it performs lots of tests
  - For now Read only or Copy-on-Read file as Copy-on-Write file
  - Requires future work

DD
Results

QEMU

- Fusecoraw works flawless
- Xmount has trouble remounting as it performs lots of tests
  - For now Read only or Copy-on-Read file as Copy-on-Write file
  - Requires future work

DD

- Both techniques work as expected, hashes match.
Both proof-of-concepts perform a good job
- Remounting writable works only with Fusecoraw
- No issue for current concept

Read data is persistent

Fusecoraw recommended if writable remounting is desired

Xmount recommended if not
Future Research

- Fusecoraw
- Xmount
- Integrate in concept
References

Gillen Daniel. 

xmount, 2008. 


Nist cloud computing forensic science challenge (draft), 2014. 


Eric van den Haak. 

Evdh's git repository, 2014. 

https://github.com/evdh-nl.