Forensic Timeline Analysis of the Zettabyte File System

Dylan Leigh (s4081906)

Supervisor: AProf. Hao Shi

Centre for Applied Informatics, College of Engineering and Science
Victoria University, Melbourne, Australia

2014

Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Timeline Analysis

Detective: “I need to know the sequence of events on the suspect’s system, when files were created, modified, deleted...”.

Timeline Analysis

- Timestamps from Filesystem
- Internal File Metadata
- Registry and Log files
- Web Browser History / Cache / Cookies

Super-timeline

- Collect all forms of event and collate into a single super-timeline
- Automated by tools such as log2timeline and Plaso

Timeline Analysis

Detective: “I need to know the sequence of events on the suspect’s system, when files were created, modified, deleted...”.

Timeline Analysis

- Timestamps from Filesystem
- Internal File Metadata
- Registry and Log files
- Web Browser History / Cache / Cookies

Super-timeline

- Collect all forms of event and collate into a single super-timeline
- Automated by tools such as log2timeline and Plaso

Timeline Analysis

Detective: “I need to know the sequence of events on the suspect’s system, when files were created, modified, deleted...”.

Timeline Analysis

- Timestamps from Filesystem
- Internal File Metadata
- Registry and Log files
- Web Browser History / Cache / Cookies

Super-timeline

- Collect all forms of event and collate into a single super-timeline
- Automated by tools such as log2timeline and Plaso

Outline

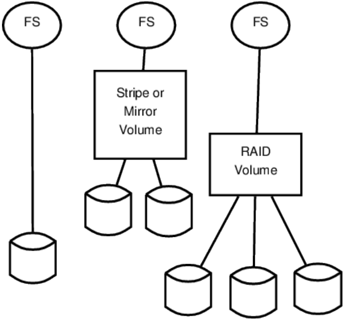
1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

What is ZFS?

"Traditional" Filesystems and Volume Management

"Traditional" Filesystems

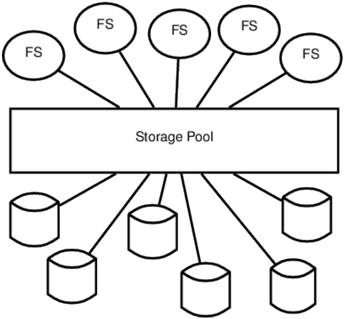
- One Filesystem on a disk or partition
- Volumes (Mirror/RAID) for redundancy
- Volumes abstract a disk to the file system
- Problems:
  - Fixed FS size
  - Fixed Redundancy
  - Inefficient use of storage



Zettabyte File System

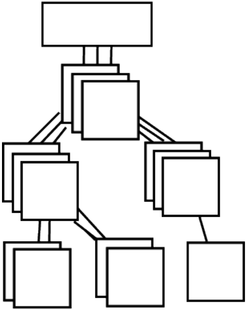
ZFS

- All disks added to a "pool" of storage
- Filesystems created as required from the pool
- Many advantages over traditional volume-based disk management
- Need for new forensic tools



ZFS Internals

- Everything is an object in a tree
- Redundant copies of objects across different devices
- Checksums in pointers allow self-healing
- COW atomic transactions
- Redundancy + Checksums + Copy-On-Write = Guaranteed Integrity



ZFS Features and Usage

ZFS Features

- Scalable and flexible pooled storage - up to 256 zebibytes across 18 billion devices, files up to 16 exbibytes...
- Guaranteed data integrity with automatic recovery
- COW tree allows efficient snapshots, remote replication
- Fine-tuned compression, deduplication and redundancy

ZFS Usage

- Cross-platform OSS Unix; common in servers, NAS
- Devices from Oracle, Netgear, Racktop, iXsystems....
- Cloud backend for Joyent, Delphix, Nexenta...

ZFS Features and Usage

ZFS Features

- Scalable and flexible pooled storage - up to 256 zebibytes across 18 billion devices, files up to 16 exbibytes...
- Guaranteed data integrity with automatic recovery
- COW tree allows efficient snapshots, remote replication
- Fine-tuned compression, deduplication and redundancy

ZFS Usage

- Cross-platform OSS Unix; common in servers, NAS
- Devices from Oracle, Netgear, Racktop, iXsystems....
- Cloud backend for Joyent, Delphix, Nexenta...

Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Digital Forensic Implications of ZFS

Nicole Lang Beebe, Sonia D. Stacy, and Dane Stuckey

Digital Investigation, Elsevier, 2009

- Overview of ZFS forensics
- Advantages (COW copies; temporal state awareness...)
- Disadvantages (compression; dynamic sizes...)
- Based on examination of the documentation and source code
- No empirical analysis of actual file systems
- No specific techniques or guidelines for technicians

ZFS Forensics Research

Data Recovery - Max Bruning

- "ZFS On-Disk Data Walk (Or: Where's My Data)." *OpenSolaris Developer Conference*, 2008
- “Recovering removed file on zfs disk." *(website)*, 2008/2013.

”Zettabyte File System Autopsy”

Andrew Li, *Macquarie University*, 2009

- ZDB Enhancement for locating known data within a disk

”Analysis and Implementation of Anti-Forensics Techniques on ZFS”

Cifuentes, J. and Cano, J. *Revista IEEE America Latina*, 2012

ZFS Forensics Research

Data Recovery - Max Bruning

- "ZFS On-Disk Data Walk (Or: Where's My Data)." *OpenSolaris Developer Conference*, 2008
- “Recovering removed file on zfs disk." *(website)*, 2008/2013.

”Zettabyte File System Autopsy”

Andrew Li, *Macquarie University*, 2009

- ZDB Enhancement for locating known data within a disk

”Analysis and Implementation of Anti-Forensics Techniques on ZFS”

Cifuentes, J. and Cano, J. *Revista IEEE America Latina*, 2012

ZFS Forensics Research

Data Recovery - Max Bruning

- "ZFS On-Disk Data Walk (Or: Where's My Data)." *OpenSolaris Developer Conference*, 2008
- “Recovering removed file on zfs disk." *(website)*, 2008/2013.

"Zettabyte File System Autopsy”

Andrew Li, *Macquarie University*, 2009

- ZDB Enhancement for locating known data within a disk

"Analysis and Implementation of Anti-Forensics Techniques on ZFS”

Cifuentes, J. and Cano, J. *Revista IEEE America Latina*, 2012

Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Research Objectives

- Determine what timeline information can be obtained from ZFS internal structures
- Determine how long useful metadata is retained
- Conduct empirical studies based on real filesystems written to disk
- Determine at least one technique for verifying file timestamps and detecting forged timestamps
- Provide a research basis for developing new forensic timeline tools for ZFS

Methodology Overview

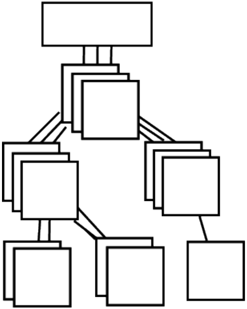
- Create ZFS pools with 1-9 disks
  - flat, mirror, mirror pair and RAIDZ configurations where applicable
- 24 hours of simulated file activity based on typical corporate network storage statistics (multiple studies)
- For each pool configuration:
  - Control (No tampering)
  - System clock reverted while a file saved
  - File timestamp modified with touch
- Metadata saved using ZDB (ZFS Debugger) every 30 minutes

Experiment Overview

- 85 simulations, each 24 hours
- Six elements in four structures examined after the simulation
- 22 extra “large file” simulations (10 minutes each)
- Over 110 GB of ZFS metadata collected

Structures Examined

Structure	Component
Uberblocks	TXG/Timestamp
Block Pointers	Birth TXG
	Vdev Number
File Objects	Object Number
	Generation TXG
Spacemaps	Spacemap TXG

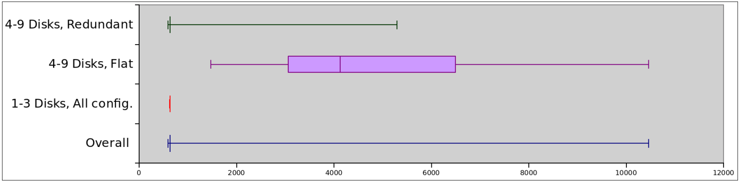


Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

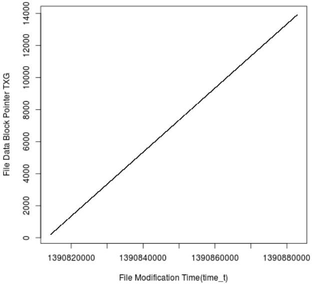
Uberblocks

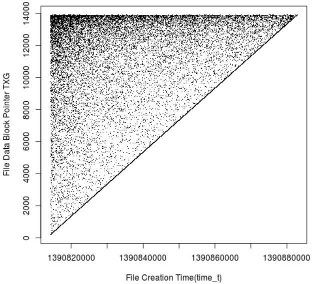
- Contain Transaction Group ID (TXG) and Timestamp
- Effective, but in most cases overwritten after 636 seconds



Block Pointer Birth TXG

```
DVA[0]=<0:24001bc0600:c00> DVA[1]=<0:4a345f03600:c00>
[L1 ZFS plain file] fletcher4 lzjb ... size=4000L/400P
birth=625760L/625760P fill=3 cksum=63e2c4fa2d:40ad...
```

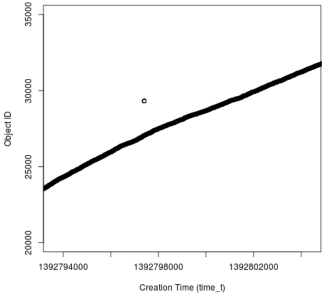




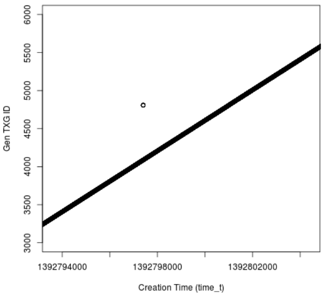
Modification Time of Individual Blocks

```
Object  lvl   ...   dsize  ...  type
     7    2   ...   258K   ...  ZFS plain file ...
...
   gen   25
   size  145408
...
Indirect blocks:
    0 L1  DVA[0]=<0:16f200:400> DVA[1]=<0:3c36c00:400>
       ... size=4000L/400P birth=29L/29P fill=2 ...
    0  L0 DVA[0]=<0:dec00:20000> [L0 ZFS plain file]
        ... size=20000L/20000P birth=25L/25P fill=1 ...
    20000  L0 DVA[0]=<0:14f200:20000> [L0 ZFS plain file]
        ... size=20000L/20000P birth=29L/29P fill=1 ...
```

Object Number and Generation TXG



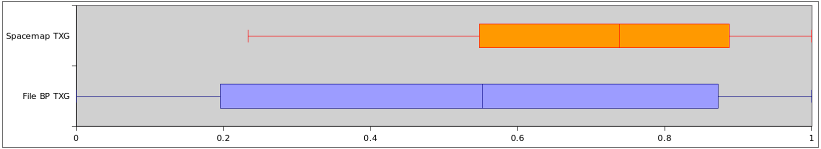
Object Number
(may be reused)



Generation TXG
(never reused)

Spacemaps and Vdev Number

- Both ineffective - too many transient objects
- Spacemap TXG when spacemap (re)written
  - not when the space was allocated
- Spacemaps “condensed” and rewritten frequently



Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Key Findings

1: Comparing birth TXG from the highest level file data Block Pointers is very effective for timeline analysis

2: Modification time of individual data blocks can be determined by comparing their level 0 BP TXG to highest level BP TXG from other files.

3: Generation TXG and Object Number can be used to determine the creation order of objects, and detect false creation times.

4: Uberblocks useful if collected before they are overwritten, typically 10.6 minutes.

5: Spacemaps and the Vdev number not usable for timeline analysis

Key Findings

1: Comparing birth TXG from the highest level file data Block Pointers is very effective for timeline analysis

2: Modification time of individual data blocks can be determined by comparing their level 0 BP TXG to highest level BP TXG from other files.

3: Generation TXG and Object Number can be used to determine the creation order of objects, and detect false creation times.

4: Uberblocks useful if collected before they are overwritten, typically 10.6 minutes.

5: Spacemaps and the Vdev number not usable for timeline analysis

Key Findings

1: Comparing birth TXG from the highest level file data Block Pointers is very effective for timeline analysis

2: Modification time of individual data blocks can be determined by comparing their level 0 BP TXG to highest level BP TXG from other files.

3: Generation TXG and Object Number can be used to determine the creation order of objects, and detect false creation times.

4: Uberblocks useful if collected before they are overwritten, typically 10.6 minutes.

5: Spacemaps and the Vdev number not usable for timeline analysis

Key Findings

1: Comparing birth TXG from the highest level file data Block Pointers is very effective for timeline analysis

2: Modification time of individual data blocks can be determined by comparing their level 0 BP TXG to highest level BP TXG from other files.

3: Generation TXG and Object Number can be used to determine the creation order of objects, and detect false creation times.

4: Uberblocks useful if collected before they are overwritten, typically 10.6 minutes.

5: Spacemaps and the Vdev number not usable for timeline analysis

Key Findings

1: Comparing birth TXG from the highest level file data Block Pointers is very effective for timeline analysis

2: Modification time of individual data blocks can be determined by comparing their level 0 BP TXG to highest level BP TXG from other files.

3: Generation TXG and Object Number can be used to determine the creation order of objects, and detect false creation times.

4: Uberblocks useful if collected before they are overwritten, typically 10.6 minutes.

5: Spacemaps and the Vdev number not usable for timeline analysis

Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Published Publications

BSDCan 2014

- Presented at BSDCan 2014, University of Ottowa, Canada on 16 May 2014
- Proceedings Paper and slides: http://www.bsdcan.org/2014/schedule/events/464.en.html

Plaso Plugins & Publication

- Used in a separate software project to add ZDB analysis to the Plaso timeline software
- ZDB Plaso software article published in *Digital Forensics Magazine, Issue 20, August 2014*

Published Publications

BSDCan 2014

- Presented at BSDCan 2014, University of Ottowa, Canada on 16 May 2014
- Proceedings Paper and slides: http://www.bsdcan.org/2014/schedule/events/464.en.html

Plaso Plugins & Publication

- Used in a separate software project to add ZDB analysis to the Plaso timeline software
- ZDB Plaso software article published in *Digital Forensics Magazine, Issue 20, August 2014*

Future Publications

Linux Users of Victoria

- One hour presentation, early 2015

Incident Response Scenario

- Forensic Investigation of a Simulated Web Server Intrusion
- Submitted for AsiaBSDCon 2015

Future Publications

- Thesis Background and Recommendations content

Future Publications

Linux Users of Victoria

- One hour presentation, early 2015

Incident Response Scenario

- Forensic Investigation of a Simulated Web Server Intrusion
- Submitted for AsiaBSDCon 2015

Future Publications

- Thesis Background and Recommendations content

Future Publications

Linux Users of Victoria

- One hour presentation, early 2015

Incident Response Scenario

- Forensic Investigation of a Simulated Web Server Intrusion
- Submitted for AsiaBSDCon 2015

Future Publications

- Thesis Background and Recommendations content

Outline

1. Introduction to Timeline Forensics and ZFS
   - Timeline Analysis
   - The Zettabyte File System
   - Existing ZFS Forensic Research
2. Experiments and Analysis
   - Objectives & Experiments
   - Structures in Detail
3. Conclusion and Future Work
   - Key Findings
   - Research Output
   - Future Work

Future Work: Next Steps

Survey

- Real data would be better than simulated file activity
- The next phase of this project is to conduct a survey of real ZFS pools from production systems
  - Volunteers submit anonymized ZDB data
    (paths and names removed)
  - Ethics approved

Forging Metadata

- Discussed theoretically in thesis
- Requires regenerating tree checksums

Future Work: Next Steps

Survey

- Real data would be better than simulated file activity
- The next phase of this project is to conduct a survey of real ZFS pools from production systems
  - Volunteers submit anonymized ZDB data (paths and names removed)
  - Ethics approved

Forging Metadata

- Discussed theoretically in thesis
- Requires regenerating tree checksums

Future Work: More Structures, Different Workloads

Other ZFS Structures

- Per-dataset objects (Master Node, Delete Queue, ...)
- ZFS Intent Log
- Snapshots
- Meta Object Set
- Past object trees from previous Uberblocks
- Old blocks on disk no longer referenced...

Other Workloads

- Longer running times, larger disks, more disks...
- ZFS Features: Compression, Deduplication ...
    - Dedicated Log and Cache devices
- Other workloads
    - Desktop, Home NAS, Webserver, VM host, Databases...
    - Pools with many datasets
- Pools providing ZVOLs for other FS

Contact Details and Questions

- Dylan Leigh
    - Email: research@dylanleigh.net
    - Web: http://research.dylanleigh.net
- Questions?