ZFS Timeline Forensics Quick Reference

Dylan Leigh

16 May 2014 - Version 1.0 - Latest version available from http://research.dylanleigh.net/

Disk Image Warning

While using the following commands directly on operating zpools may be educational; any real forensic investigation should never work with the physical disks.

A block-by-block master copy of each disk should be saved with `dd` or an appropriate forensic tool, a digest of the master copy saved, and then the master copy along with the original disks should be sealed and preserved. Only work with a copy of the master copy, and use the digest to verify it is not modified.

The `zpool import` command may be useful for importing disk images as a ZFS pool. Make sure the `readonly` and `altroot/-R` options are used.

I. Useful ZFS Commands

- Import disk images in a directory as a zpool:
  ```
  zpool import -R <alternate-root-dir> -o readonly=on
          -d <directory>
  ```
- Status of zpools - mentions missing or faulted devices:
  ```
  zpool status -v [pool]
  ```
- Builtin history - includes zpool creation, settings changes, filesystem and snapshot creation:
  ```
  zpool history -il [pool]
  ```
- List all properties from a pool:
  ```
  zpool get all <pool>
  ```
- List all ZFS Filesystems:
  ```
  zfs list
  ```
- List all ZFS Filesystems, Snapshots and Clones:
  ```
  zfs list -t all
  ```

II. Getting Data from ZDB

Note: -P turns on "parseable" output (e.g. "2012314 bytes"instead of "2MB")
For most options, repeating it increases verbosity.

- Pool configuration:
  ```
  zdb -C <pool>
  ```
- Uberblocks from a device:
  ```
  zdb -P -uuu -l <device>
  ```
- Objects and Block Pointers from a Dataset:
  ```
  zdb -P -bbbbbb -dddddd <poolname>/<dataset>
  ```
- Spacemaps for all metaslabs in a pool:
  ```
  zdb -P -mmmmmm <poolname>
  ```

III. Analysing uberblocks

- Compare Uberblock TXG and timestamps:
  ```
  grep t[ix][gm] <uberblock-data-file> | grep -v contiguous
  ```

IV. Analyzing Dataset Objects

- Extract Gen TXG / Crtime pairs:
  (note: this only works neatly because these entries are next to each other in the output. Later versions of ZDB might break this)
  ```
  grep -A 1 '^[[:space:]]*crtime' <dataset-data-file>
  ```

- Locate files with multiple Block Pointers to data:
  ```
  awk '($0 ~ /ZFS plain file/ && int($2) > 1) {print}'
          <dataset-data-file>
  ```