

Enumerating and Analyzing Storage Data with Digital Forensics XML

Alex Nelson

Prometheus Computing, LLC
University of California, Santa Cruz



❖ Contractor at NIST

- Working with NSRL team

❖ UC Santa Cruz Ph.D. Candidate

- Thesis title:

Software signature derivation from sequential digital forensic analysis

❖ Here because of work with Digital Forensics XML

- ❖ Originally by Dr. Simson Garfinkel
 - One of the goals:
Analyze storage without needing the storage image
- ❖ Language to describe files, file systems, and partitioning systems
 - All inode (/MFT entry) metadata
 - File content checksums
 - Location metadata
 - Generating-process provenance
 - Generic enough for any file system
- ❖ Schema available
 - Documents elements
 - Validates DFXML documents with *xmllint*

DFXML language bindings

- ❖ C bindings (by Garfinkel)
 - Library for writing DFXML documents
- ❖ Original Python bindings, *dfxml.py* (by Garfinkel)
 - Read-only objects
 - SAX processing with callback functions
- ❖ Objects.py bindings (by Nelson)
 - Read-write objects
 - xml.etree (SAX-like) processing with loops and *iterparse()*

DFXML research background

❖ File system differencing

- Based on DFXML software
- On second differencing algorithm version

❖ Differencing research:

- Changes between file system states
[Garfinkel, DFRWS 2012]
 - Two metadata manifests in, differences out
- Discrepancies in tool reports
[Nelson, DFRWS 2014]
- Registry effects measurements
(Ongoing)

DFXML-based projects

❖ Implemented:

- *iredact.py* (Garfinkel, Woods) – Redact disk image
 - Improvements on BitCurator development agenda
- *UPartsFS* (Nelson) – Disk partitions as big virtual files
 - FUSE file system
 - (Not actually DFXML-based, but helps with tool analysis)

❖ To implement:

- DFXMLFS.py – Treat DFXML file like a regular file system
 - FUSE file system, in Python
- Parsers for old file systems (?)
- ?

References

- ❖ DFXML Library:
<https://github.com/simsong/dfxml>
- ❖ DFXML Schema:
https://github.com/dfxml-working-group/dfxml_schema
- ❖ Diskprint workflow:
https://github.com/ajnelson/diskprint_workflow
- ❖ Garfinkel, DFRWS 2012:
<http://dfrws.org/2012/proceedings/DFRWS2012-6.pdf>
- ❖ Nelson, DFRWS 2014:
<http://dfrws.org/2014/proceedings/DFRWS2014-6.pdf>