Cooperative Mode: Comparative Storage Metadata Verification Applied To The Xbox 360

By

Alex Nelson, Erik Steggall and Darrell Long

Presented At

The Digital Forensic Research Conference

DFRWS 2014 USA Denver, CO (Aug 3rd- 6th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

http:/dfrws.org

Cooperative Mode: Comparative storage metadata verification applied to the XBox 360

Alex J. Nelson[1,2], Erik Q. Steggall[1], Darrell D. E. Long[1]

[1] University of California, Santa Cruz
[2] Prometheus Computing, LLC.

DFRWS, August 5, 2014

Acknowledgements

- This presentation results from research supported by the Naval Postgraduate School Assistance Grant/Agreement No. N00244-12-1-0066 awarded by the NAVSUP Fleet Logistics Center San Diego (NAVSUP FLC San Diego). The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Introduction

- Post-mortem storage analysis:

  The disks may be dead, but the understanding is moving.

- Storage forensics studies results of *implementations*; we're lucky to have *specifications*.

  - File system code some of the most active changes in Linux [Lu, FAST'13]

  - Complete specifications for NTFS (technically) unavailable
    - Heavily reverse-engineered
    - But did your forensic tool get it right?
    - Did your subject's OS get it right?
    - How right?

Good news / bad news

- Bad news: We can't ask how *right* storage analysis is.
  - "Correct" needs a *formal* specification of storage system states.
  - Nobody has specified all possible faults of storage states:
    - Faulty OS
    - Faulty file system code
    - Arbitrary adversarial corruption
- Good news: We can ask how *wrong* an analysis is.

Cooperative mode: *N*-Version Programming

- *N*-Version Programming [Chen, FTCS'78]:
  - Implement the same formal specification *N* times.
  - Compare the results; consensus (or average) answer is taken.

- In storage analysis:
  - Have *N* parsers enumerate storage metadata.
    - All storage analytics stem from metadata.
  - List the metadata differences - those are tool bugs!

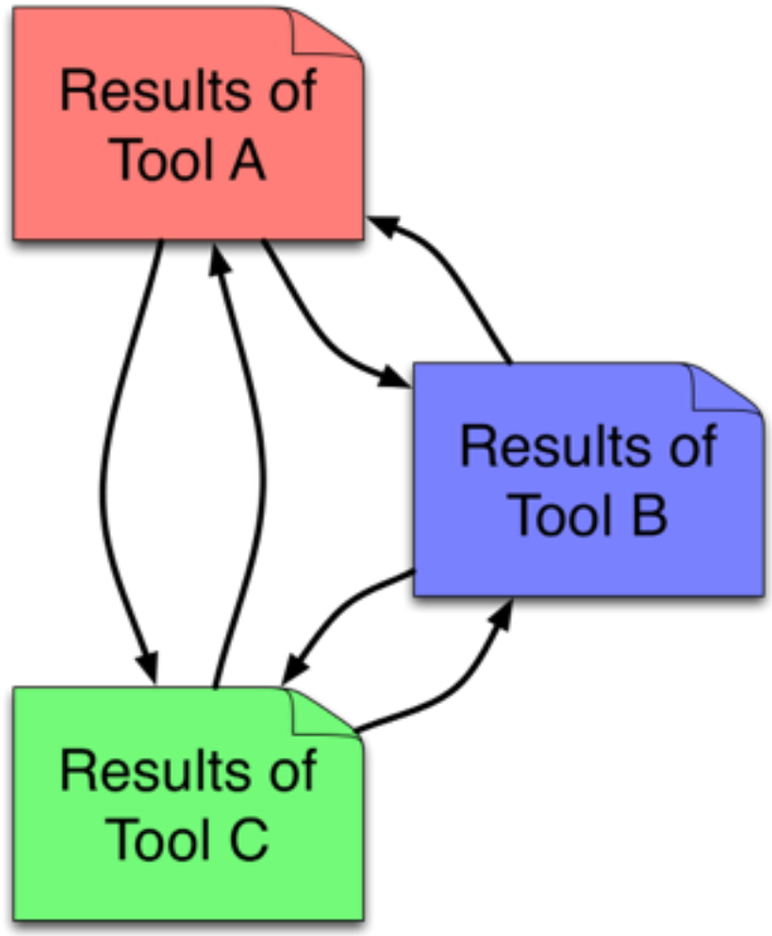
Effective tool comparison

- Storage parser comparison needs:

  - 1 in-common output format, for comparison;
    1 comparison algorithm.
    - # of comparisons = double edges of complete graph
      = N^2-N **work**.
    - **"Work" is designing a tool comparison.**
    - Just use one algorithm.

  - Minimal human work.
    - Need scripting interfaces,
      but GUIs preferred today [Hibshi, IMF'11]

  - Diversity in code bases.
    - You can fool **some** of the forensics tools **all** of the time,
      and **all** of the forensics tools **some** of the time,
      but not **all** of the forensics tools **all** of the time.



Why XBox 360?

- Game consoles are computers too

(Image credit: pcper.com)

- XBox 360 hard drive: standard SATA
  - Specialized file system, XTAF, like FAT
  - Specialized volume management
- File system analysis code available from “The Internet”
  - Not advertised in bigger forensic products
  - Your enigmatic phone may have equivalent “analysis support”
  - How incorrect is the code?
  - We can find out!

Contributions

- N-Version Programming in storage forensics

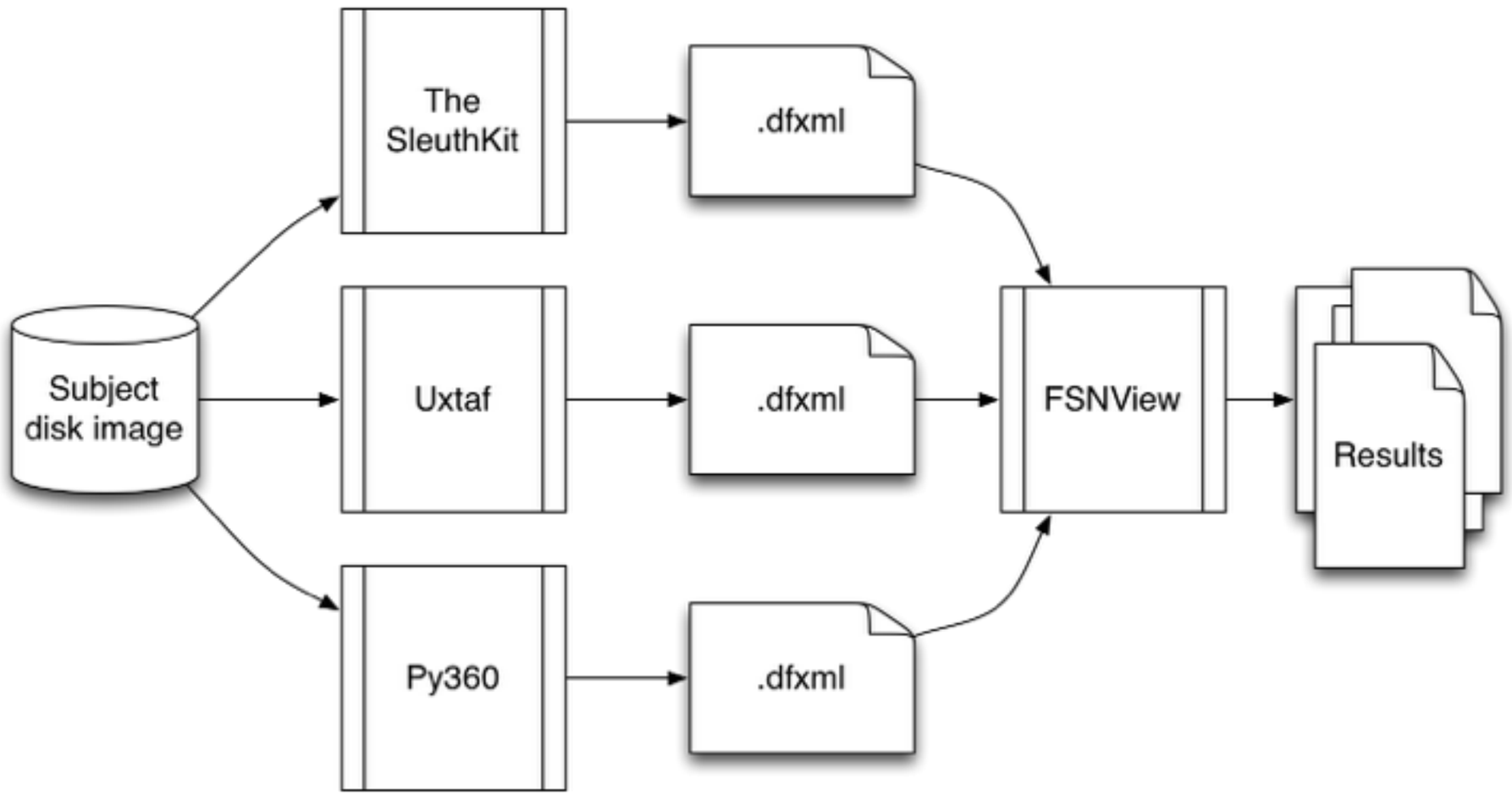
- Code
  - Three XTAF → DFXML parsers.
  - New *idifference* implementation.
  - New DFXML Python library.
  - DFXML validating schema.

  - File system parser comparator, *FSNView.*
  - A helper file system, *UPartsFS*.

- Measurement reporting for file system differences

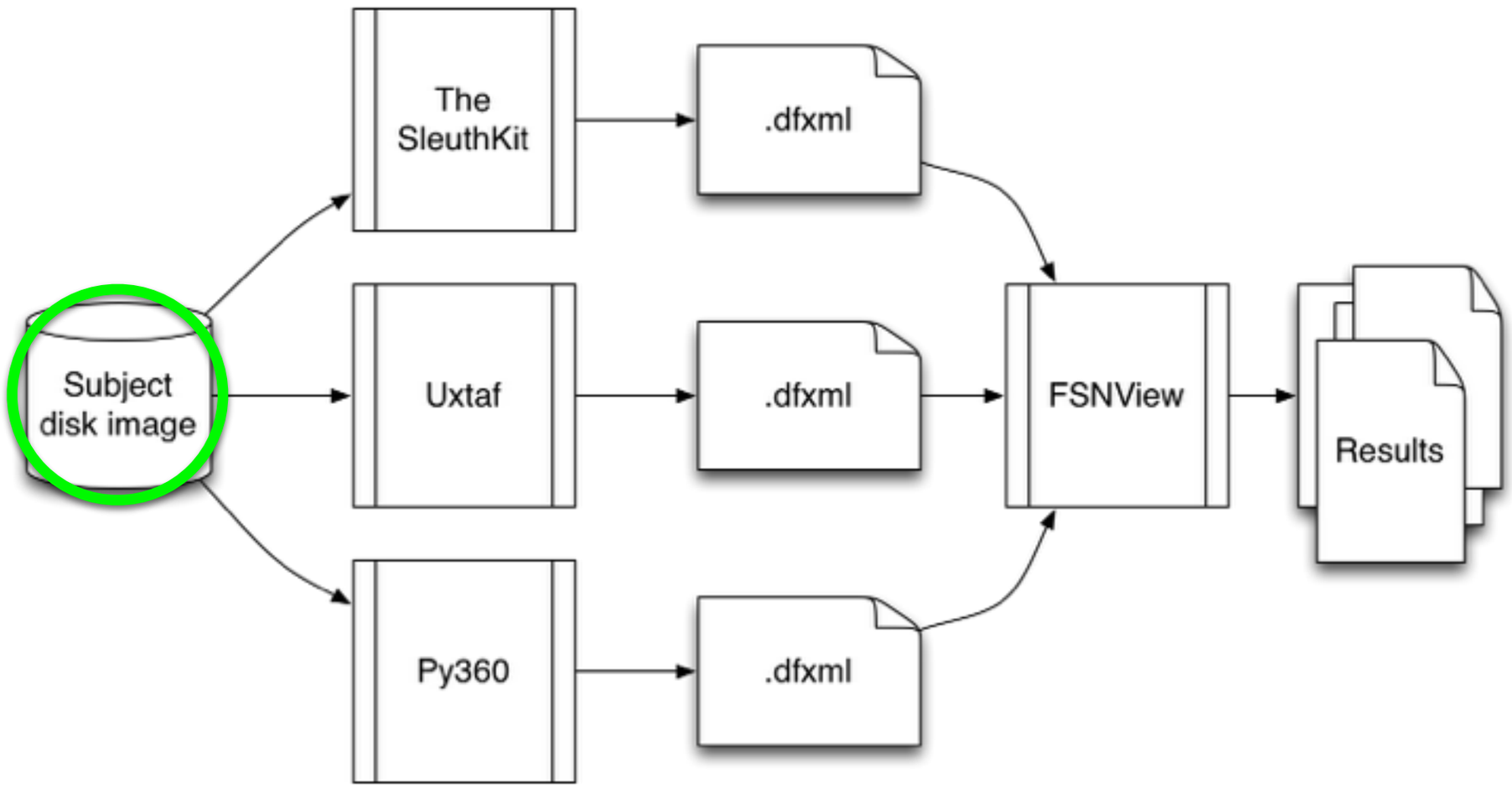
Talk outline

- Analytic subject: XTAF
- DFXML updates
- Making three XTAF → DFXML parsers
- Comparing the results



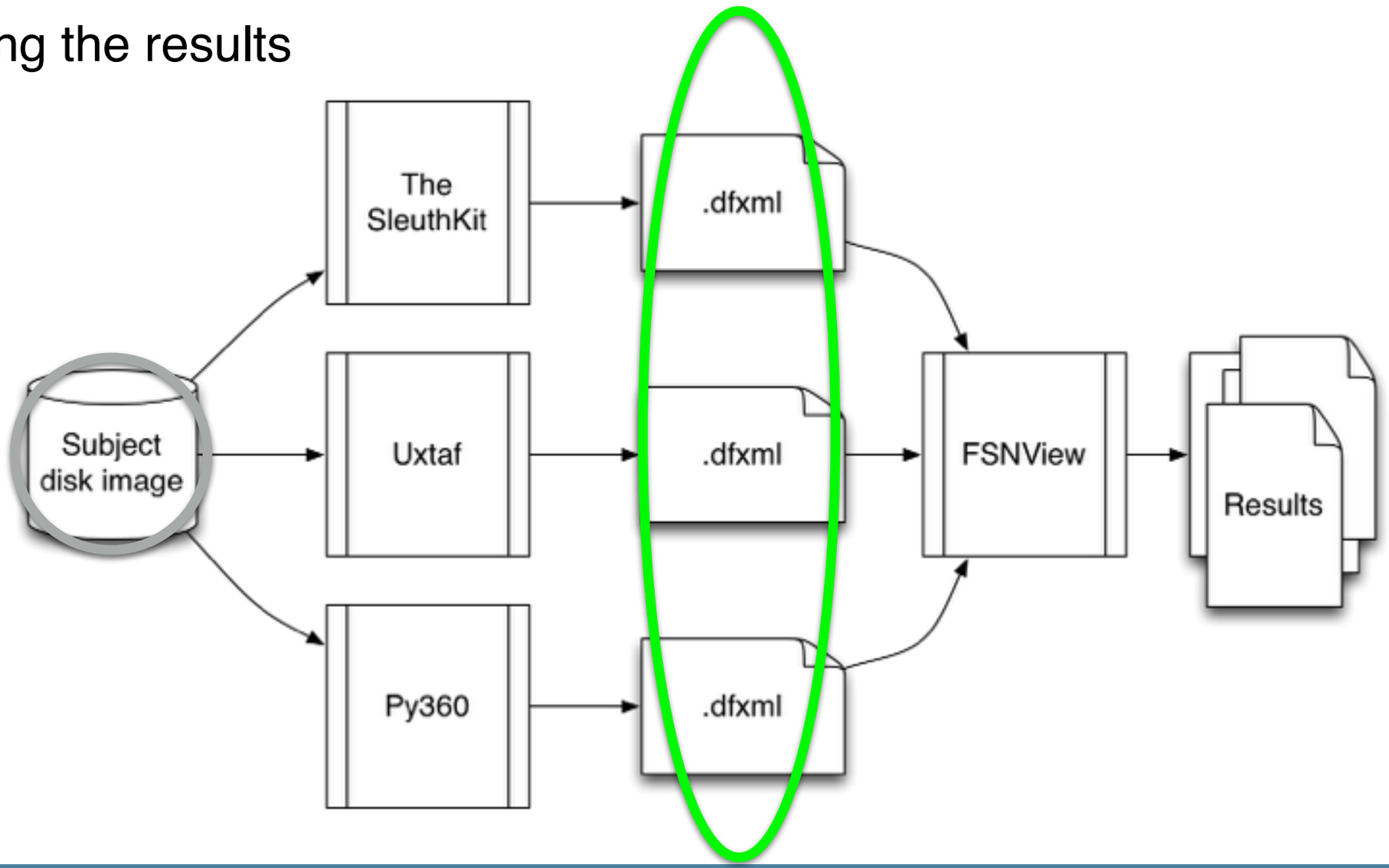
Talk outline

- Analytic subject: XTAF
- DFXML updates
- Making three XTAF → DFXML parsers
- Comparing the results



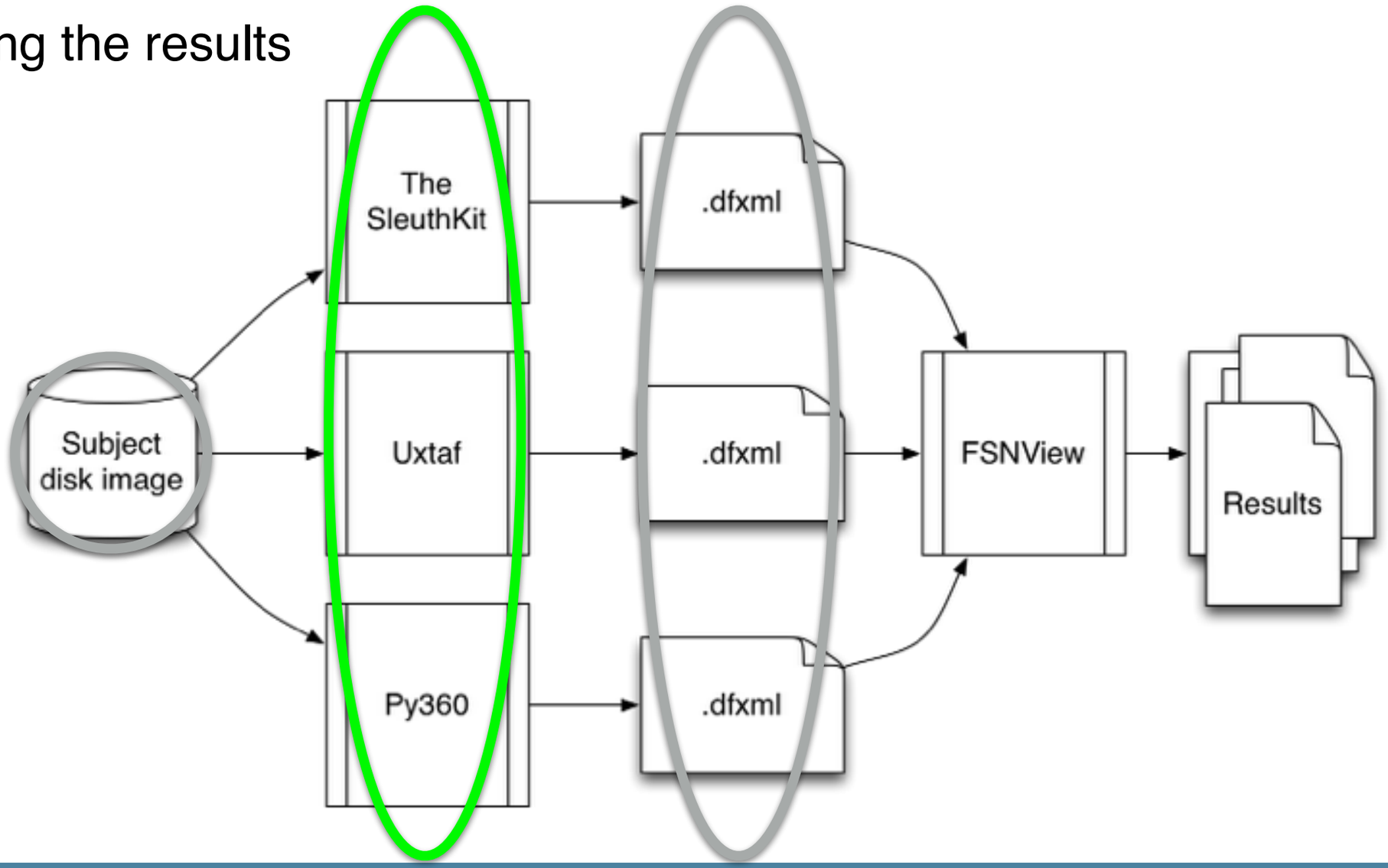
Talk outline

- Analytic subject: XTAF
- DFXML updates
- Making three XTAF → DFXML parsers
- Comparing the results



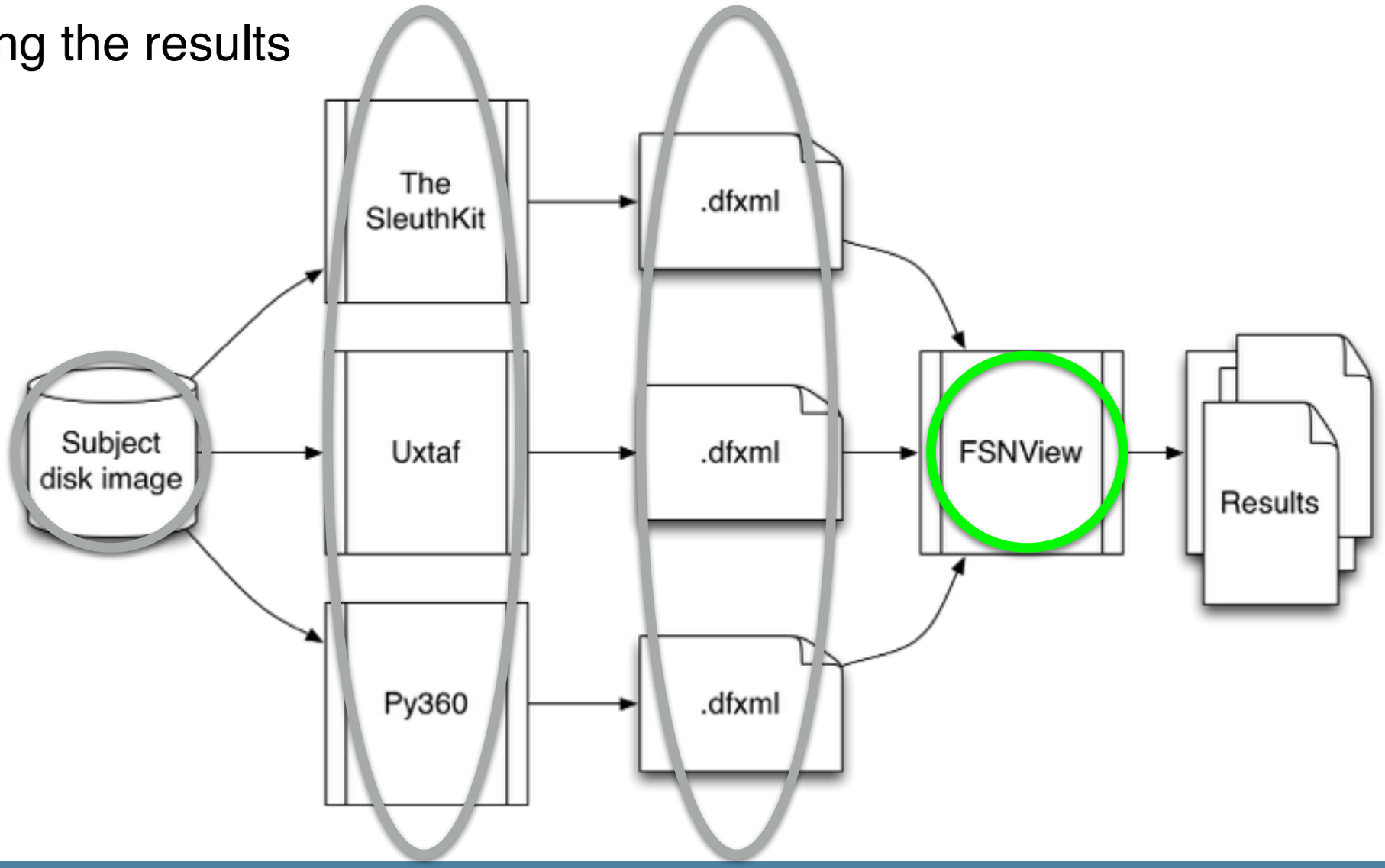
Talk outline

- Analytic subject: XTAF
- DFXML updates
- Making three XTAF → DFXML parsers
- Comparing the results



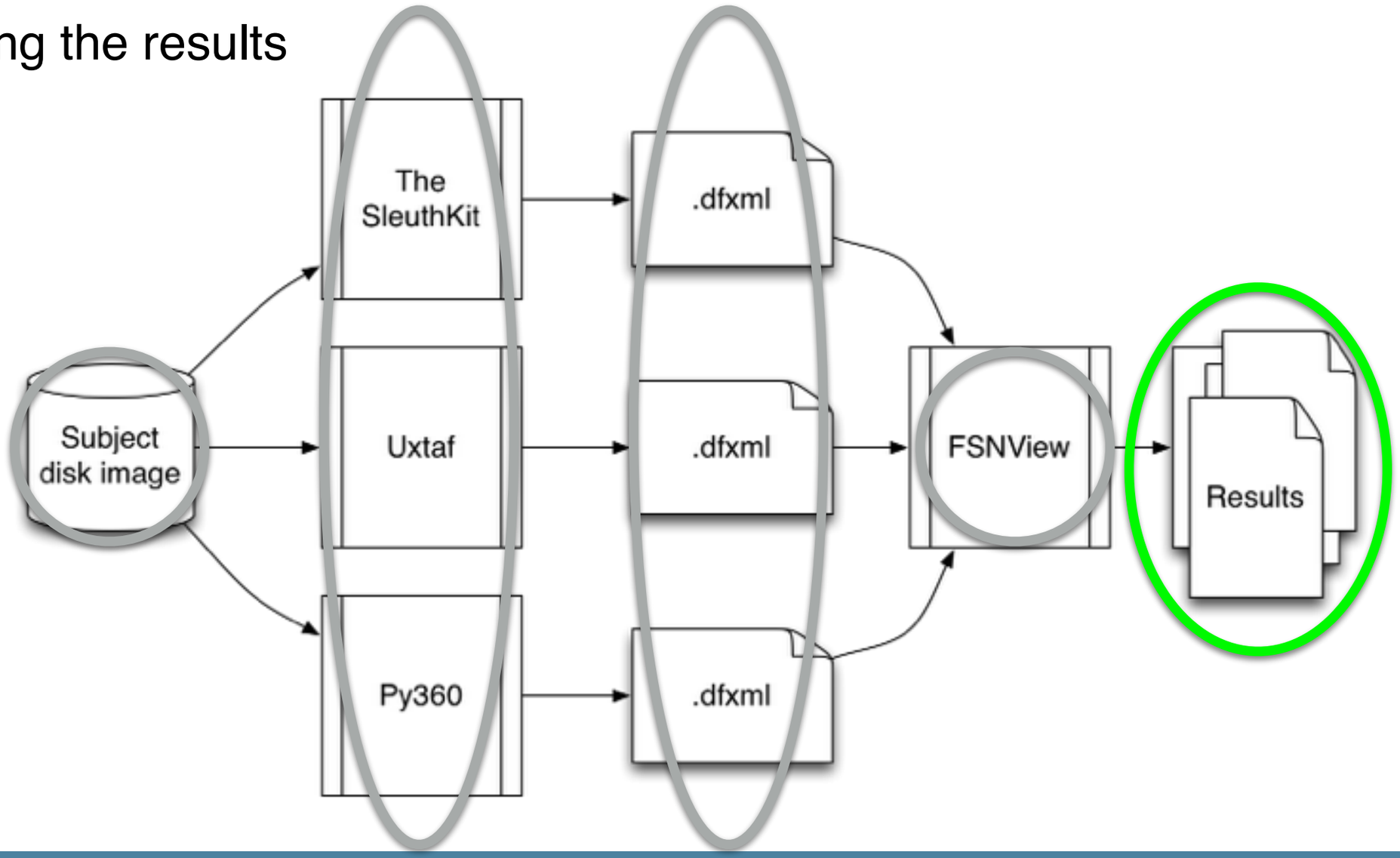
Talk outline

- Analytic subject: XTAF
- DFXML updates
- Making three XTAF → DFXML parsers
- Comparing the results



Talk outline

- Analytic subject: XTAF
- DFXML updates
- Making three XTAF → DFXML parsers
- Comparing the results



CENTER FOR RESEARCH IN STORAGE SYSTEMS

DATA

XTAF

- A variant of FAT
  - Looks, acts like FAT16 and FAT32
  - Uses different data structures
  - No parent directory references
  - ASCII filenames
    - Users don’t name directly
- Partitions managed with hard-coding
  - Signature: ASCII bytes “XTAF” at one of six known offsets
  - Two encrypted partitions
    - (This is not a cryptanalysis talk.)

Data creation steps

- Wireless networking USB device removed.

- Unplugged frequently.
  - Reset clock to 2005 frequently.

- All actions were taken using the normal interface.
  - Single and multi-player games
  - Sending messages
  - XBox Live purchasing with pre-paid, now-0'd debit cards

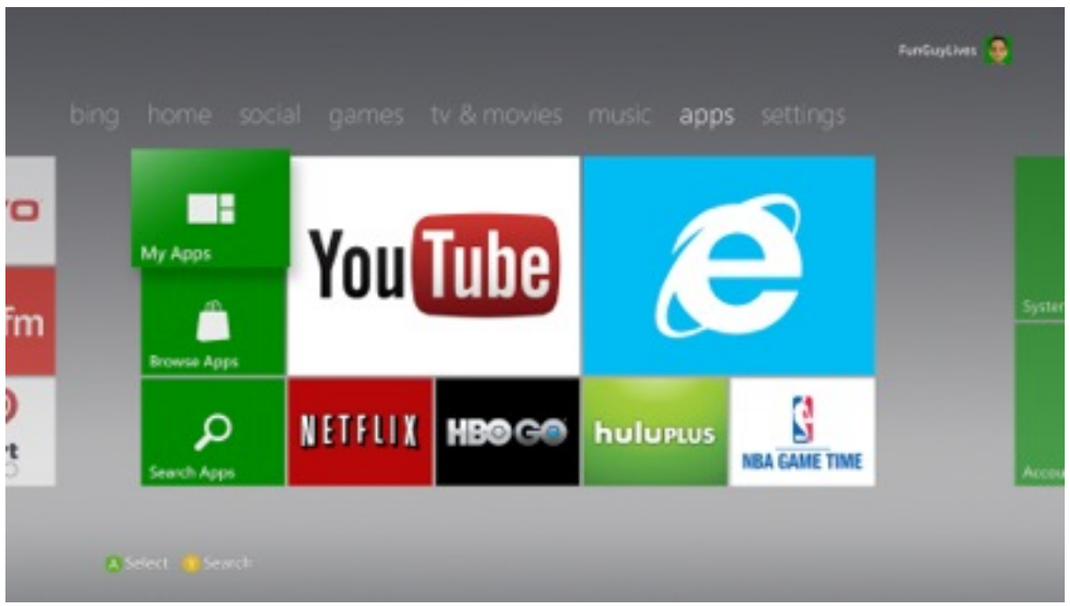


Image credit: Kotaku

CENTER FOR RESEARCH IN STORAGE SYSTEMS

CODE

idifference, Episode III
(*Revenge of the diff)*

- Original *idifference* [Garfinkel, DFRWS'12a]:
  - Identified matching files between file sets.
  - Identified changes in matching files.
  - Reported changes.
  - Good program; not great *library*.
- We reimplemented *idifference* and the DFXML library.
  - Fundamental diff. job, matching files, now a separate program.
    - Deletions make matching challenging.
  - Difference representation now part of DFXML library.
    - *Differential annotations* to DFXML.
  - *idifference2*: same results, new internals.
  - Further differential analytics are now possible.

DFXML now validating, read-write

- DFXML generators need agreement on what to generate.
  - XML schema developed: https://github.com/dfxml-working-group/dfxml_schema
  - Works with *xmllint.*

- New DFXML library, Objects.py.
  - Writable DFXML documents.
  - Writable properties (`f.filename = "$FAT1"`).
    - An aside: Getters, setters, and transparency to clients: https://archive.org/details/SeanKellyRecoveryfromAddiction (10min video).
  - Iterators (`for f in objects: …`) instead of callbacks.
    - Based on, and should feel like, `xml.etree.ElementTree`.
  - Now can report *metadata* locations.

Storage parsers and adaptations

- The SleuthKit and Fiwalk
  - In C/C++, by Carrier, Garfinkel, many others
  - Venerable FAT parser
  - Adapted copy of FAT parser to XTAF data structures

- Uxtaf
  - In C, by René Ladan
  - Merged "`ls`" and "`cd`" shell functions into DFXML generator

- Py360
  - In Python (2.x), by "Arkem"
  - Includes *file* analytics
  - Added DFXML Object population into file system walk

UPartsFS

- Problem with file system parsers:
  Sometimes, they only know file systems.

- UPartsFS: Partition table as a userspace file system.
  - Presents each partition as a raw image file.
  - Uses FUSE & TSK volume library.

- Advantages:
  - *File system* parsers don't have to worry about *partition management*.
  - No offset acrobatics in parsers.
  - Don't need kernel module to read a new partition scheme.

FSNView: All of the above

- "File System N View"
  - ("Fusion View" already taken.)
- Input:
  - XBox 360 hard drive; or pre-made DFXML files.
- Output:
  - Summary of diffs, as file count.
  - Breakout of diffs, in each metadata dimension.
    - Tables in LaTeX and HTML.
  - Lists of differing file paths (normal *idifference* output).

CENTER FOR RESEARCH IN STORAGE SYSTEMS

RESULTS - Lessons learned

Summary

- Scale: ~300–350 files & directories.

	fiwalk	py360	uxtaf
Partitions processed	5	6	4
Allocated directories	65	58	56
Allocated files	293	231	231
Allocated other	0	0	0
Unallocated directories	1	14	8
Unallocated files	2	15	11
Unallocated other	15	0	0

Differences – files

- Differencing is not a symmetric operation.

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
Additional files	13	3	53	0	55	12
Allocated	5	3	37	0	37	2
Unallocated	8	0	16	0	18	10
Missed files	53	55	32	12	22	0
Allocated	37	37	5	2	3	0
Unallocated	16	18	27	10	19	0
Renamed files	0	0	0	0	0	0
Allocated	0	0	0	0	0	0
Unallocated	0	0	0	0	0	0

Differences – files

- Differencing is not a symmetric operation.

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
Additional files	13	3	53	0	55	12
Allocated	5	3	37	0	37	2
Unallocated	8	0	16	0	18	10
Missed files	53	55	32	12	22	0
Allocated	37	37	5	2	3	0
Unallocated	16	18	27	10	19	0
Renamed files	0	0	0	0	0	0
Allocated	0	0	0	0	0	0
Unallocated	0	0	0	0	0	0

Differences – files

- Differencing is not a symmetric operation.

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
Additional files	13	3	53	0	55	12
Allocated	5	3	37	0	37	2
Unallocated	8	0	16	0	18	10
Missed files	53	55	32	12	22	0
Allocated	37	37	5	2	3	0
Unallocated	16	18	27	10	19	0
Renamed files	0	0	0	0	0	0
Allocated	0	0	0	0	0	0
Unallocated	0	0	0	0	0	0

Differences – files

- Differencing is not a symmetric operation.

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
Additional files	13	3	53	0	55	12
Allocated	5	3	37	0	37	2
Unallocated	8	0	16	0	18	10
Missed files	53	55	32	12	22	0
Allocated	37	37	5	2	3	0
Unallocated	16	18	27	10	19	0
Renamed files	0	0	0	0	0	0
Allocated	0	0	0	0	0	0
Unallocated	0	0	0	0	0	0

Differences – files

- Disagreement on which directories to walk.

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
Additional files	13	3	53	0	55	12
Allocated	5	3	37	0	37	2
Unallocated	8	0	16	0	18	10
Missed files	53	55	32	12	22	0
Allocated	37	37	5	2	3	0
Unallocated	16	18	27	10	19	0
Renamed files	0	0	0	0	0	0
Allocated	0	0	0	0	0	0
Unallocated	0	0	0	0	0	0

Differences – files

- Disagreement on which directories to walk.

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
Additional files	13	3	53	0	55	12
Allocated	5	3	37	0	37	2
Unallocated	8	0	16	0	18	10
Missed files	53	55	32	12	22	0
Allocated	37	37	5	2	3	0
Unallocated	16	18	27	10	19	0
Renamed files	0	0	0	0	0	0
Allocated	0	0	0	0	0	0
Unallocated	0	0	0	0	0	0

Differences – file metadata

Differences in...	fi→p3	fi→ux	p3→fi	p3→ux	ux→fi	ux→p3
SHA-1 (dirs)	6	60	1	55	52	55
SHA-1 (files)	8	2	2	8	1	8
SHA-1 (other)	0	0	0	0	0	0
Filesize (dirs)	62	61	53	0	52	0
Filesize (files)	0	0	1	0	1	0
Filesize (other)	0	0	0	0	0	0
Modified time	0	0	0	0	0	0
Access time	0	0	0	0	0	0
Metadata change time	0	0	0	0	0	0
Creation time	0	0	1	0	1	0
Data byte runs (dirs)	1	61	0	64	52	64
Data byte runs (files)	1	0	2	0	1	0

Some lessons learned

- *Disagreeing metadata?* You may have found a re-allocated file.

- *Non-unique allocated-file paths?* Are you reading deleted content right?

- *Trusting non-essential metadata?* For shame.
  - (Directory entry “file name length” is only length up to a period.)

- Some tools missed files with user names.
  - Tool diversity helped Bulk Extractor ascribe artifacts.

Future work

- *More data, please!*
  - Links on forensics wiki corpora page
- *Let's clarify file system definitions.*
  - Refine DFXML vocabulary
  - Converge on "Correct" state of test data
- Testing for file systems with inodes *and* directory entries
  - NTFS: Fiwalk vs. EnCase vs. ????
    https://github.com/Sebastienbr/DFXML-EnCase

Conclusion: Tool comparison helps you, now.

- Human-cheap file system metadata comparison is available.
  - You can know whether a tool will report a file incorrectly.

- Check: *Is your "Smoking gun" file a discrepancy?*
  - Look for its name/hash in the diff lists.

- Check: Are your discrepancies hiding smoking guns?
  - Evidence of anti-forensic masking tactics? (Anti-your-forensics?)

Questions?

- My email:
  - ajnelson@cs.ucsc.edu
- Errata, disk images, and code locations at:
  - https://users.soe.ucsc.edu/~ajnelson/research/nelson_dfrws14/

Image credits

- http://www.kotaku.com.au/2012/10/xbox-360-dashboard-update-oh-hey-internet-explorer-for-your-tv/
- http://www.pcper.com/reviews/General-Tech/New-Xbox-360-S-Slim-Teardown-Opened-and-Tested/Enter-new-guy

CENTER FOR RESEARCH IN STORAGE SYSTEMS

BACKUP SLIDES

CENTER FOR RESEARCH IN STORAGE SYSTEMS

Paper Outline

1. Introduction
    1. Outline
2. Background: Theory and frameworks
    1. Digital Forensics XML
    2. Differential analysis
3. Analytic subject: Xbox 360 and the XTAF file system
    1. The XTAF filesystem
    2. Partition management
4. Designing XTAF data
5. Improving DFXML and differencing for tool evaluation
    1. Formalizing the DFXML language
    2. Implementing new DFXML Python bindings
    3. Modularizing idifference.py
    4. Byte runs to note more than content locations
6. Programs extended for DFXML comparison
    1. Uxtaf
    2. Py360
    3. The SleuthKit
7. Tools developed for DFXML comparison
    1. UPartsFS: Extending single-partition file system parsers
    2. FSNView: A single-data, multi-interpreter DFXMLreporter
8. Evaluating multi-tool analysis of Xbox 360 storage
    1. Artifact recovery
9. Related work
    1. Practices
    2. Other tool comparison in storage forensics
    3. Xbox analysis
10. Future research
11. Conclusion

Differential DFXML sample: NTFS

```
<fileobject delta:changed_file="1" delta:modified_file="1">

  <filename>$MFT</filename>

  <filesize delta:changed_property="1">30490624</filesize>

  <mtime>2009-11-08T15:44:29Z</mtime>

  <byte_runs delta:changed_property="1">
   <byte_run file_offset="0" fs_offset="3221225472" img_offset="3221257728" len="30490624"/>
  </byte_runs>

  <hashdigest delta:changed_property="1" type="sha1">dca2e60259189a2b2b74c8526d7a2e3f084c8e04</hashdigest>

  <delta:original_fileobject>
   <filename>$MFT</filename>
   <filesize>29556736</filesize>
   <mtime>2009-11-08T15:44:29Z</mtime>
   <byte_runs>
    <byte_run file_offset="0" fs_offset="3221225472" img_offset="3221257728" len="29556736"/>
   </byte_runs>
   <hashdigest type="sha1">104fd989c490d05aa4ec8abcf05269dcb591e68d</hashdigest>
  </delta:original_fileobject>

</fileobject>
```