TOOL VALIDATION: WAS THAT SUPPOSED TO HAPPEN?

NIST PERSPECTIVE AND PROGRAMS

JIM LYLE

cftt@nist.gov

DISCLAIMER

Certain trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose. No financial interest.

NIST & CFTT

- CFTT – Computer Forensics Tool Testing project
- Established in 2000 to provide a methodology for testing computer forensic software tools
- Develop . . .
  - Tool Requirements
  - Methods to test tools against tool requirements
  - Test data sets (CFReDS) and tools to generate test data sets
- Federated Testing – Test plan in a box: select test data, run tests, record results, generate standard format test report and share results.

CFTT TESTING

There are CFTT test methodologies (requirements and test plans) & Test Reports for:

- Digital Data Acquisition
- Write Blocking
- Drive Wiping
- Deleted File Recovery (Meta-data based)
- Registry Forensics
- String Search
- Mobile Devices
- File Carving

5 CHARACTERIZING RELIABILITY

- Daubert – criteria to help assess reliability admissibility of scientific testimony
    - Tested
    - Peer review
    - Error rate
    - Standards & controls
    - General acceptance
- Daubert, Kuhmo Tire & GE v. Joiner.
- FRE 702

ARE THE RESULTS OF A FORENSIC EXAM RELIABLE?

- The court wants to know if results are reliable.
- Digital & Multimedia Forensic practitioners are confident that tools and methods are reliable, but . . .
- Saying "I know it works" is just not acceptable. There needs to be some objective support to the claim.
- How to characterize and communicate tool reliability?
- Other forensic disciplines use error rates to describe chance of false positive, false negative or otherwise inaccurate results.
- Why not Digital?

ADDRESSING DAUBERT

- Error Rate works great to characterize matching two items –
- Examples:
  - Using hashes to see if two files match or if a file has changed
  - DNA
- Error Rate is not so great on characterizing software tools because of the way software fails
- Examples:
  - Might systematically omit something – Partial acquire
  - Might put unrelated items together –
    recover a file with data from multiple sources
- Testing is a dark art – studied since the 1950's, but software is still buggy

ERROR RATES

- Error rates are usually statistical in nature or there is some kind of random variable from a population
- Usual way to compute an error rate for a method is to study a large population and count the number of times the method gives wrong answers.
- With digital data the technology changes rapidly and new technology often changes the proportion of right and wrong answers.
- For deleted file recovery the file system format: matters: FAT, NTFS, ExFAT, HFS+, APFS, ext4
  - Need different deleted file recovery algorithms for each
  - Fat only points to first block, NTFS has pointers to all blocks, HFS+ clears all block pointers

ALGORITHM VS IMPLEMENTATION

- Error rate for Hash algorithms is based on the math in the algorithm, but actual performance is based on the implementation.
- Are two files the same – The probability two file hashes match is a very very very very very small number that is close to zero.
- An implementation may have an error rate quite different from the algorithm
- It depends on how software can fail . . .

HOW DOES SOFTWARE FAIL?

- Software failure is not statistical in nature . . .
- Software (usually) does the same thing (wanted or not) with the same input. – systematic errors
- But wait, you often see the question "What is the error rate of this or that tool?"
- For software this is the wrong question. It is possible to derive an error rate from observation of a large number of trials, but the rate is an average and not relevant to any specific case
- Need to ask what limitations have been revealed by testing?
- Keep in mind that technology changes steadily

FORENSIC TOOL FAILURE EXAMPLES

- Acquire Digital Data
  - Fail to acquire last few sectors of a device
  - Report incorrect digital device size
  - Treatment of bad sectors
- Write Block
  - Fail to block all write commands on all interfaces
  - Block some read commands too.
- Mobile Devices
  - Partial acquire of some artifacts
  - Get the phone’s number wrong
  - Different failures for different devices

13 MORE TOOL FAILURES

- String Search
  - Strings missed in specific combinations of parameters, e.g., UTF-8 Chinese found, but not if UTF-16
  - Latin based character string (English, French, German, Italian, etc.) reported twice for UTF-16, reported once for UTF-8.
  - ASCII strings not found in unallocated space.
- File Carving for JPG, GIF, TIFF, etc . . .
  - Tool reported lots of false positives
  - Tool confused by file fragmentation
  - Color map corrupted
  - Carved file has elements from more than one source

14 V VS V – TERMINOLOGY

- Verification is the same as Validation, Right?
- Lots of confusion – NIST/CFTT just uses "tool testing"
- Usual English definition of **verification***: the process of establishing the truth, accuracy, or validity of something.
- Usual English definition of **validation***: the action of checking or proving the validity or accuracy of something.
- In the Software Engineering context, a subtle distinction:
  - Validation: Check if building the right tool – testing & design review
  - Verification: Check if the tool is built right – software testing

* Definitions from Mac Dictionary app

SOFTWARE V&V

- Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [IEEE-STD-610]
- Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]
- Software validation ensures that "you built the right thing" and confirms that the product, as provided, fulfills the intended use and goals of the stakeholders.
- Software verification ensures that "you built it right" and confirms that the product, as provided, fulfills the plans of the developers.

TO DO LIST FOR TESTING

You need three things:

1. Decide what the tool or process needs to do – write requirements
2. Decide on a method (algorithm) to meet the requirements
3. Implement the selected algorithm (Write software)

- Validation is checking if the algorithm meets the requirements
- Verification is checking that the software correctly implements the algorithm

HOW TO TEST SOFTWARE

- As many different methods as there are programmers
- Lots of general approaches – details differ – in general it's all about failure
  - Start with requirements
  - Make a list of items that can be tested (i.e., you can create a test case that can fail)
  - Create test data sets
    - You need to have tight control of everything
    - You need to know what results the tool should report
    - It's tempting to use the data from the last several investigations with multiple tools to try and establish what a tool should report, but this is not reliable and you don't know if it contains data elements you need
  - Run the test cases

18 TESTING EXAMPLE – FILE CARVING

- Many file types have recognizable signatures in the file data
  - Graphic – jpeg, gif, png, bmp & tiff
  - Video – mp4, wmv, 3gp, ogv, mov, avi
  - Document – doc, docx, xls, xlsx, pdf, ppt & pptx
  - Archive – zip, rar, 7z, gz & tar
  - Others -- ???
- Can't test all at once

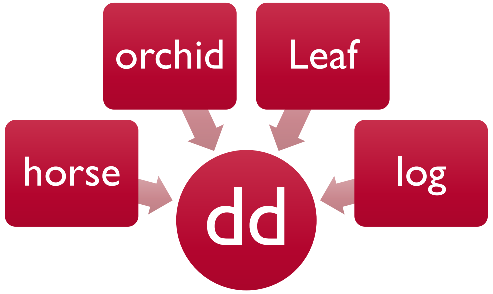
19 TESTING ISSUES -- IDENTIFY REQUIREMENTS

- Dozens of parameters that might affect tool behavior
- Focus on most important parameters
  - Completeness
  - Fragmentation
  - Embedded pictures (thumbnails)
  - Tool option settings (use default values)
- Be aware of other issues like . . .
  - File type specific characteristics
  - Compression level
  - Thumbnails
  - EXIF data
  - Audio track

DATA SETS FOR GRAPHIC FILES

- Collection of separate graphic files:
  - Barn.gif
  - Winter.tiff
  - River.png
  - Oak.jpg
  - Also bmp
- Eight files of each type
- Can construct "dd disk image file"

BASE DD FILE – COMPLETE & CONTIGUOUS PICTURE FILES



orchid

Leaf

horse

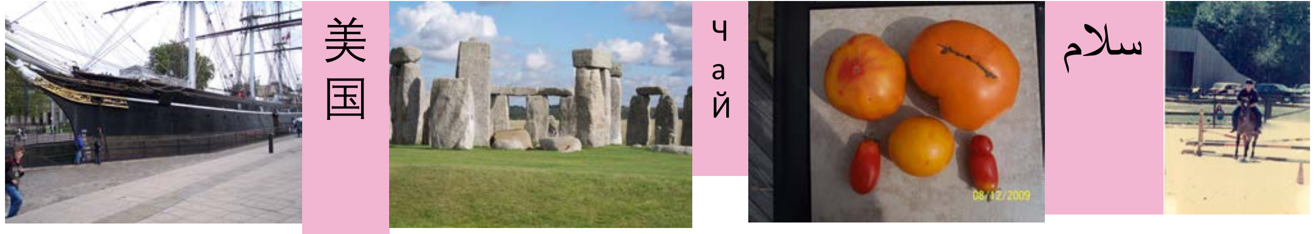
log

dd

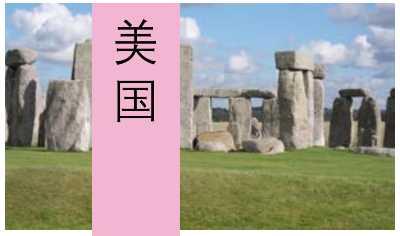
Zero fill to end of last sector

CONSTRUCTING OTHER IMAGES

- Padded with cluster sized blocks of non-English text between pictures



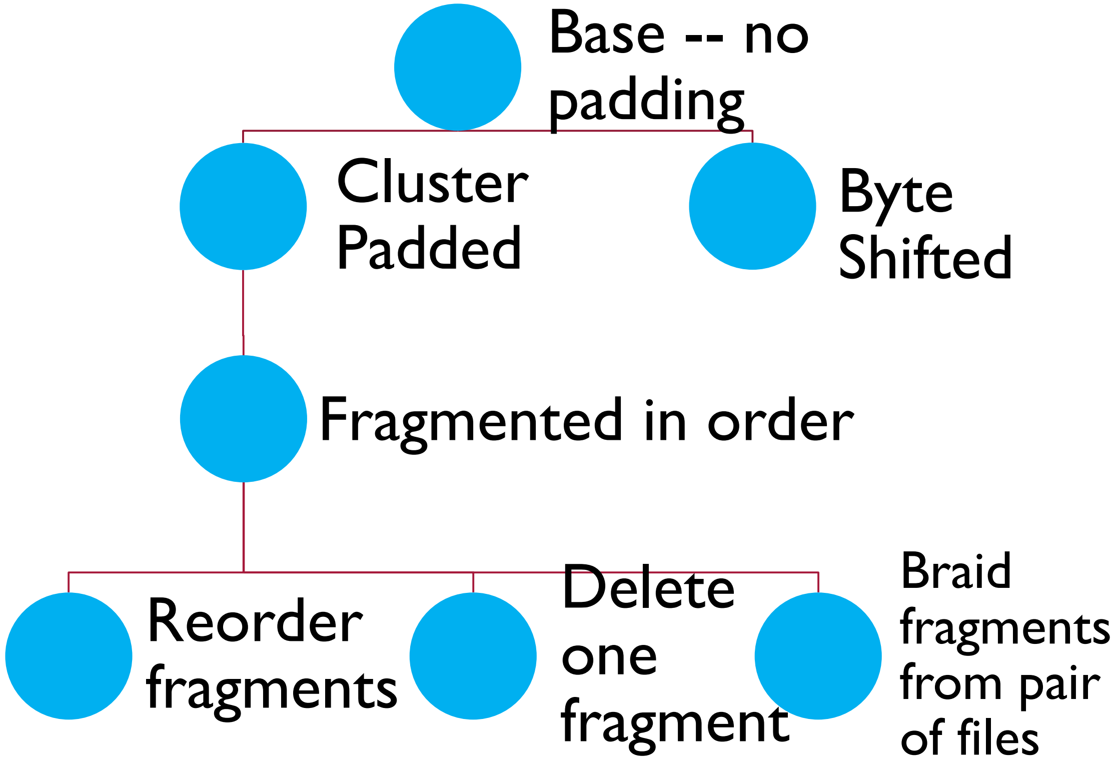
- Fragmented (in order)



Other dd images

- Fragmented (out of order)
- Braided (two files intertwined)
- Incomplete files
- Non-aligned to sectors

23 CARVING TEST IMAGES



Base -- no padding

Cluster Padded

Byte Shifted

Fragmented in order

Reorder fragments

Delete one fragment

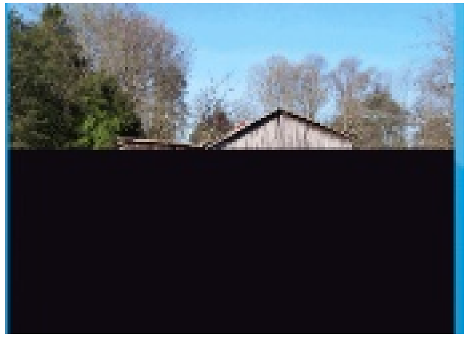
Braid fragments from pair of files

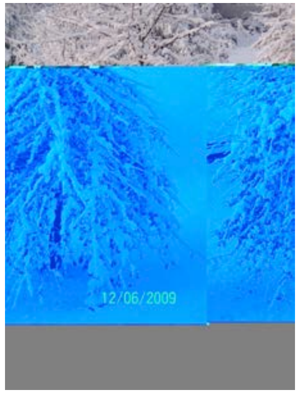
MEASURING RESULTS

- Two approaches –
  - Visibility driven – does the tool produce usable (viewable) results
  - Data driven – See what the tool actually does in relation to ground truth
    - Measure fraction of returned data that belongs
    - Measure fraction of possible data returned
- Methods are complementary

VISIBILITY DRIVEN MEASUREMENT

- Each file checked for visibility by two independent observers
- Resolve differences if disagreement





Category	Visibility
Viewable Complete	Flaws – minor or none
Viewable Incomplete	Flaws – partial, multiple files
Not viewable	Data matches file type, Flaw prevents display
False Positive	Data doesn’t match file type

DATA-DRIVEN MEASUREMENT

- We know the ground truth
- Based on sectors present in carved files and information retrieval based statistics – evaluate returned data
  - Relevant – sector comes from a source file in dd file
  - Retrieved – sector returned in a carved file
- P = (relevant ∧ retrieved)/retrieved -- fraction of retrieved sectors from a source file -- **how much noise returned**
- R = (relevant ∧ retrieved)/relevant – fraction of relevant sectors retrieved – **how much stuff missed**
- F = 2 x (P x R)/(P + R) - average of P & R

A RABBIT-HOLE OF INTERESTING BEHAVIOR

- One tool (A) recovered 8 tiff files from the unpadded dd file
- F score for tiff files was 1.00
- But, only one file was viewable, seven were not viewable
- Examination of the eight files – last sector of tiff file replaced by noise in the carved file
- That last sector is critical to having a displayable file
- Other tools on same data –
  - Tool B Carved 4 with 3 viewable
  - Tool C Carved 10, none viewable
  - Tool D Carved 8, all viewable
- Without both measures we wouldn't know how close the tool was. Maybe an investigator can repair the file and extract a critical piece of evidence

OTHER RESOURCES

- www.SWGDE.org
- CFTT & CFReDS:
  - www.cftt.nist.gov -- Test Requirements, Test Plans & Tool Test Reports
  - www.cfreds.nist.gov – Test Data Sets
- UK Forensic Science Regulator:
  - https://www.gov.uk/government/organisations/forensic-science-regulator
  - https://www.gov.uk/government/publications/method-validation-in-digital-forensics

SUMMARY

- Most tasks not likely to have an error rate (except for some tasks like matching)
- Even if you have an error rate it can be misleading or obsolete
- Use tool testing to uncover tool limitations
- Work from a list of requirements
- Design Test Data to try to get the tool to fail sidetracked
- Look at SWGDE ”Error Mitigation Document” for guidance on reporting on tool limitations

CONTACTS

Jim Lyle
JLYLE@NIST.GOV

Barbara Guttman
BARBARA.GUTTMAN@NIST.GOV

Rick Ayers
RICHARD.AYERS@NIST.GOV

http://www.cfreds.nist.gov	Test Data Sets
http://www.cftt.nist.gov	Test Reports