Analysis of Redundant Traces for Consistency

July 24, 2009

Dr. Fred Cohen
President - California Sciences Institute
CEO – Fred Cohen & Associates

Outline

- Background and Introduction
  - My Background
  - Previous work
  - Redundant traces & type C and D consistency
- Feature and characteristic extraction
- Building sieves and counting things
- Summary, conclusions, and further work

My background

- California Sciences Institute
  - 501(c)(3) non-profit California research and educational institution - WASC accreditation candidacy pending
  - Ph.D. Program in digital forensics (Fall 2009)
- Fred Cohen & Associates
  - Enterprise information protection consulting
  - Digital forensics (high fees – no guarantees)
- Fred Cohen – Digital forensics
  - POST certified instructor, FLETC instructor, books and book chapters, papers, testimony in Federal, State, and Local courts

Previous work

- Carrier, Gladyshev, Willassen
  - Model the forensic analysis process in terms of consistency and inconsistency
- Stallard and Levitt
  - Semantic integrity checking (consistency)
- Cohen
  - Trace consistency: type C (internal) and D (external)
- My basic notion and approach
  - To make a science of digital forensics, we need a physics and a theory for applying it
  - This is about the theory and its limits

Basic notions of DFE

- The evidence is a set of traces
  - A “trace” is a “bag of bits”
  - Normally an ordered sequence
  - It is the result of some digital process
  - The question is: “What process?”
  - How do we find out?
  - How sure are we? Why are we this sure?
- The evidence is latent in nature and technical
  - You need tools to see it and experts to explain it
  - What tools, and how can you trust them?
  - What experts, and how credible are they?

California Sciences Institute

The context of the model

Legal context

- Legal theory
- Application
- Methodology
- Jurisdiction
- Case type
- Proof standard
- Calendar
- Strategies
- Costs

Evidence

- Identify
- Collect
- Preserve
- Transport
- Store
- Analyze
- Interpret
- Attribute
- Reconstruct
- Present
- Destroy

Tools

- Methodology
- History
- Pedigree
- Reliability
- Testing
- Calibration
- Function
- Limitations

Litigants

- Due care
- Retention

People

- *Scientific*
- Knowledge
- Skills
- Experience
- Training
- Education
- *Non-scientific*
- Clarifying
- Observation
- Honesty
- Integrity
- Competence

Challenges

- Make/Miss
- Content
- Context
- Meaning
- Process
- Relationship
- Ordering
- Time
- Location
- Corroboration
- Consistency
- Accident/intent
- False positive
- False negative

Legal process

- Pre-legal
- First filing
- Notice
- Preservation
- Productions
- Disclosures
- Depositions
- Motions
- Sanctions
- Admissibility
- Pre-trial
- Testimony
- Disposition

Admissibility

- Relevance
- Original writing
- Authenticity
- Probative > Prejudicial
- Hearsay

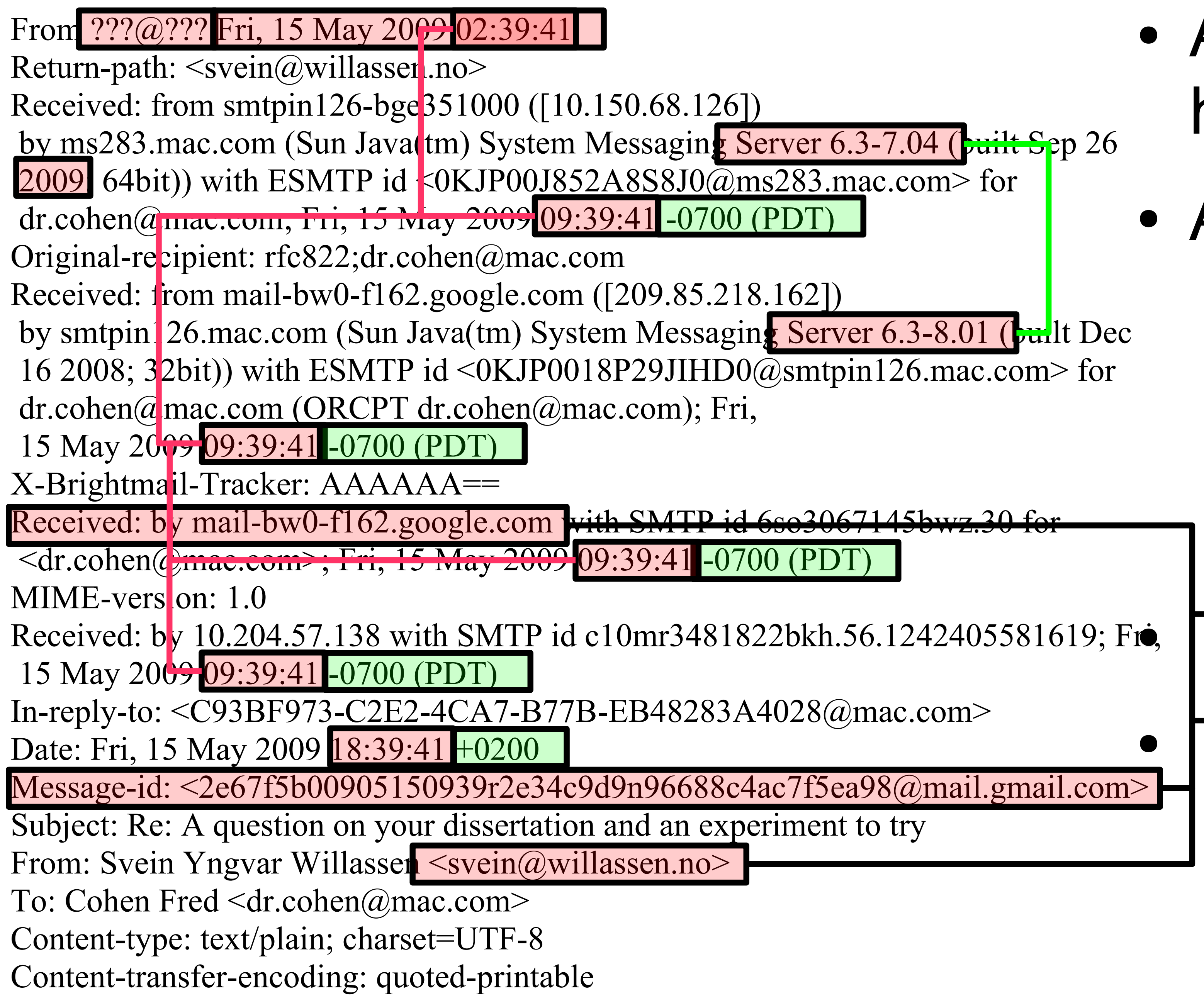
Drill down at http://all.net/

The model

- Laws, Violations, Resources, Schedule
- Hypothesized claims: $H=\{H_1, ..., H_n\}, H \subset V$
- Events: $E: \{e_1, ..., e_o\}$ (statements, etc. non DFE)
- Traces: $T:(t_1, ..., t_q)$ {all subsequences of T}
  - All subsets of the bag of bits
- Trace (internal) consistency: $C:TxT \rightarrow [-1...1]$
- Demonstration consistency: $D:TxE^* \rightarrow [-1..1]$
- $P:\{p_1, ..., p_n\}, \forall p \in P, p \rightarrow \{c \subset C, d \subset D, c \not\subset C, d \not\subset D\}$
  - Procedures that produce c, d, c, d

California Sciences Institute

Example: an email extract



```
From ???@??? Fri, 15 May 2009 02:39:41
Return-path: <svein@willassen.no>
Received: from smtpin126-bge351000 ([10.150.68.126])
 by ms283.mac.com (Sun Java(tm) System Messaging Server 6.3-7.04 (built Sep 26
2009; 64bit)) with ESMTP id <0KJP00J852A8S8J0@ms283.mac.com> for
 dr.cohen@mac.com; Fri, 15 May 2009 09:39:41 -0700 (PDT)
Original-recipient: rfc822;dr.cohen@mac.com
Received: from mail-bw0-f162.google.com ([209.85.218.162])
 by smtpin126.mac.com (Sun Java(tm) System Messaging Server 6.3-8.01 (built Dec
 16 2008; 32bit)) with ESMTP id <0KJP0018P29JIHD0@smtpin126.mac.com> for
 dr.cohen@mac.com (ORCPT dr.cohen@mac.com); Fri,
 15 May 2009 09:39:41 -0700 (PDT)
X-Brightmail-Tracker: AAAAAA==
Received: by mail-bw0-f162.google.com with SMTP id 6so3067145bwz.30 for
 <dr.cohen@mac.com>; Fri, 15 May 2009 09:39:41 -0700 (PDT)
MIME-version: 1.0
Received: by 10.204.57.138 with SMTP id c10mr3481822bkh.56.1242405581619; Fri,
 15 May 2009 09:39:41 -0700 (PDT)
In-reply-to: <C93BF973-C2E2-4CA7-B77B-EB48283A4028@mac.com>
Date: Fri, 15 May 2009 18:39:41 +0200
Message-id: <2e67f5b00905150939r2e34c9d9n96688c4ac7f5ea98@mail.gmail.com>
Subject: Re: A question on your dissertation and an experiment to try
From: Svein Yngvar Willassen <svein@willassen.no>
To: Cohen Fred <dr.cohen@mac.com>
Content-type: text/plain; charset=UTF-8
Content-transfer-encoding: quoted-printable
```

- An email header
- Asserted as:
  - Original writing
  - Received in New Jersey
- Type C 🟥
- Type D 🟩

Fred Cohen & Associates

What's the problem?

- Type C problems identified (so far)
  - “From “ separator ???@??? and date format
  - “From “ offset from last Received (False+)
  - Received: times in the same second (how fast?)
  - Gmail message-ID but emitted from non-gmail account (passes through Google later – Google added AFTER earlier “Received:”?)
  - Message server built after Message Received!
  - Server versions inverted w.r.t. Build time stamps
- Type D problems identified (so far)
  - Received in NJ inconsistent with all time zones
- Lots of traces extracted from the original trace

Size of the space

- T is the size of all sets of all states
  - In a particular matter, T is the available traces
  - For m bits of traces, $|T|=\sum(m!n)2^n$ for n=1 to m
    - 64 bit trace→$3*10^{31}$ possible actual traces
- C is $|T|^2$
    - 64 bit trace →10^{63}
- D is |T|*|power set of E|
- Exhausting C or D is infeasible for any real case
  - Exhausting consistency checks is infeasible
  - What is a “thorough” job?

This is only the beginning

- Which if these are actually spoliation?
  - And how do we tell?
- How many more traces are there?
  - In this specific sequence?
  - Are there other sequences?
  - How about cross-sequence C consistency?
- How do these relate to other events?
  - Version numbers of servers and dates and times
  - Anchor events tying down other facets
  - Character sets available on machines at times
- Where does it end?

Forensic procedures

- P is the size of all instruction sequences executed on all subsets of T and E
- $|\text{Instruction set}|^{|\text{number of instructions executed}|}$
  - 100 instruction instruction set
  - 10^9 instructions per second for 1 second
  - $|P| \approx$ 1 followed by 10^{18} 0's.
- |P| in reality is – perhaps 10^3-10^4?
  - scientific methodology properly applied
  - executed by tools that have been tested, calibrated, demonstrated to be reliable
  - Applied by suitable experts

Returning to the example

- How many more traces are there?
  - We now know the answer – and it hurts!
- How many more procedures may there be?
  - An enormous number in total – but which are probative and how reliable are they?
  - We don't even know how many more there may be for a single email header!
- How do we test the reliability of the apparent inconsistencies?
  - We need an experimental base and samples and lots of procedures to test

Outline

- Background and Introduction
- Feature and characteristic extraction
- Building sieves and counting things
- Summary, conclusions, and further work

We need forensic procedures

- One approach to creating forensic procedures
  - Bottom up from a bag-of-bits to “meaning” in “context”
  - Trace typing: What are the top-level syntax and semantics of the bag of bits?
    - What is the symbol set in use? Assume and test
    - Examine headers and media type = O(1)
    - JDLR, statistical, lexical analysis = O(m+n)
      - m types, n bits, string search or lexical analysis
  - NOTE: Only if no errors are found!!!
    - Inconsistencies with all known syntax
  - NOTE: ignores things that “look right”
    - Virtualization? Other boot media? Steganography

Up the bag-of-bits stack

- Finding exact copies
  - Exacts string (bits) search = O(m+n)
    - m=size of trace, n=size of target string
    - For multi-targets, n=size of largest target
  - Note: many implementations do not do as well
- Searching for regular expressions / build parser
  - LALR parser = O(n+m)
    - Size of the parse tree + size of the trace
    - Linear time (regular expressions, BNF, ...)
    - Assuming that the language is such a language
  - Note: many languages are not (e.g., human, gif)
  - Note: a failed parse leads to non-linear time

Up the bag-of-bits stack 2

- Equivalent content in different formats
  - As soon as it's not an exact match...
    - Inexact match implies equivalence classes
    - If the class sets are differentiable by LALR parser remains linear time. But if not...
  - Example, date and time stamps / pictures
    - From different time zones EST v. GMT v PST
      - Easy to do – class sets make easy equivalence
    - In different formats (e.g., 02/07/08)
      - 2002-07-08 or 2008-02-07 or 2008-07-02?
      - Impossible to be certain how to parse / compare
- In general, impossible to do equivalence matching correctly – or almost correctly fast

Normalization for matching

- Normalize to commensurate language
  - A photograph ⇒ description of colored regions, separations, edge lines, etc.
  - Dates and times ⇒ UTC, yyyy-mm-dd-hh-mm-ss.dddd
  - Words ⇒ lower-case, space-separated, spell corrected,
- Match normalized form
  - Same number of regions? Color values within Δ?
  - Order by date and time < / = / >?
  - Look for sequences? nvan.?
- Retain pointers back to originals
  - Allows for traceability

Problems with normalization

- These are no longer exact matches
  - What is the basis for this similarity?
  - How can you show that the class sets are valid?
  - How can you claim when it is close enough?
  - We only have an exact consistency theory today
- This requires a scientific methodology
  - Theory of similarity and refutation mechanism
  - Experimental basis for confirming/refuting
  - Enough relevant experimental results to provide reliability information
  - Peer reviewed publications showing limits and probative value

Generating characteristics

- For a given set of characteristics, chunking the characteristics into different sequences and differentiating between sequences is O(n·log(n))
  - Essentially, create m^n n-tuples and identify all of the n-tuples and where they fit
  - Do this ∀ traces, ∀ n-tuples, ∀ symbol sets to generate all sets of characteristics
  - Identify features from the set of characteristics and use the features as a basis for comparison
- All the same problems as normalization
  - Consistency is a function of context

Consistency analysis

- Ordering assumptions and out of order analysis
  - Causality implies time ordering of events
  - Extract times O(n) and detect out of order O(nc)
    - n is total trace length
    - c is complexity of comparison
  - BUT: Clock skews must be taken into account
    - Must consider all possible orderings
    - Jitter and skew $\Rightarrow$ ordering is not precise
    - And reliability/retransmission/store and forward
  - A sliding window can be used, but
    - O(nk!) where k=window size in relevant records
    - $\forall\ t_1, t_2 : |t_1 - t_2| < \Delta \Rightarrow t_1 \approx t_2$ – but what Δ to use?

Consistency analysis

- Sourcing and travel patterns
  - Compare message times to each other O(2·n·m)
    - n = number of hops, m= number of messages
  - Lots of other source and travel pattern issues
- Consistency of related records
  - Time to compare records is O(x·n·log(n))
    - x = time to associate records to each other
    - n = number of entries after sorting
  - When association is strict = O(n·log(n))
- Anchor events and external correlation
  - Correlate to anchor events O(n) for n events

Outline

- Background and Introduction
- Feature and characteristic extraction
- Building sieves and counting things
- Summary, conclusions, and further work

Derived traces

- It's often easier to work with content by reformatting it preprocessing it
  - Extract relevant trace(s)
  - Reformat or preprocess into new form (D-trace)
  - Retain linkage back to original trace (O-trace)
  - Perform calculations on D-trace
  - Assert that the results apply to the O-trace
- But be careful!!!
  - $\forall$ O-traces and D-traces $D_1 < D_2 \Rightarrow O_1 < O_2$?
  - Not always!!! You have to know what you are doing or you will come to false conclusions

Other sorts of derivations

- Translations (O(n), n=length of trace)
  - EBCDIC → ASCII?
  - there-fore → therefore?
  - ⇥ → ␣␣␣␣␣␣␣␣?
  - Continued lines into single lines?
    - Be careful!!!
    - It depends on the use of the trace
- Other translations (unknown complexity)
  - French → English?
  - Java → Lisp?
    - Be careful!!!
    - It depends on the use of the trace

More analysis

- Counting things O(1) up to a size limit...
- Combined mechanisms and error handling
  - Each of these has potentially different error mechanisms and modes
  - When you combine them, you may compound errors in a wide range of ways
    - A+9=J?
    - Derived trace search finds things never present?
    - Each invertible on its own but not together?
  - Error output may be used for computation in the next phase!
    - Better check intermediate values – but how?

Outline

- Background and Introduction
- Feature and characteristic extraction
- Building sieves and counting things
- Summary, conclusions, and further work

Is this all obvious stuff?

- In some sense, each of these results better be pretty darned obvious and readily verified
  - So there is nothing new here?
- What is new?
  - Up from bag-of-bits
    - Has not been described elsewhere?
  - We are now able to count and compare things
    - Maybe - in some cases – if we are careful
    - But there are now far more error modes known
  - We are sure – that we are not so sure anymore
  - But we have a methodology and method to test

So what next?

- Extensive testing of this new scientific methodology to make it more definitive
  - What are the real limits of these methods?
  - How reliable are which techniques?
  - How do we measure the reliability of results?
  - How do we test tools against this methodology?
  - How do we now talk about our results?
- The creation of well-understood methods and their properties and limitations
  - These are the classes of errors with that sort of procedure performed on this kind of derived trace and related back to that original trace

How do we say this?

- The traces are internally and externally (in)consistent based on the following checks:
  - T_1 is consistent with T_2 because...
  - T_3 is inconsistent with T_4 because...
  - T_5 is consistent with E_1 because...
  - T_6 is inconsistent with E_2 because...
- And for more complex situations:
  - I did X to get T_1 and T_1 is consistent with E_2
  - I did Y to get T_1 and E_2 combined with T_2 is consistent with E_3

More complicated things

- Based on the procedures I undertook (list here), the trace of Message 1 is consistent with the account named J sending Message 1 to the account named K at or about the time and date specified.
- I applied the following procedures (list here) to the following traces (list here) to try to determine if J sent forged Message 1, and I did not identify any inconsistencies that would tend to indicate that Message 1 was forged by J or anyone else.

What's next?

- This paper only covers the rudimentary forms of analysis in widespread use today
  - Further work is needed to characterize other classes of consistency checking in analysis, including analysis of effects of parallelism
  - Detection of similarity rather than more precise matches
  - Addressing issues of mixed symbol sets and other similar environmental factors,
  - Analysis of possible consistencies and inconsistencies of missing traces and use of this to guide future events
  - Validation requirements for the methods used

Thank You

http://calsci.org/ - calsci at calsci.org

http://all.net/ - fc at all.net

Further Reading

- F. Cohen, "Challenges to Digital Forensic Evidence", ASP Press, 2008 ISBN#1-878109-41-3
- F. Cohen, "Digital Forensic Evidence Examination", ASP Press, 2009, ISBN#1-878109-44-8.
- T. Stallard and K. Levitt, "Automated Analysis for Digital Forensic Science: Semantic Integrity Checking", ACSAC-2003
- S. Willassen, "Hypothesis-based investigation of digital timestamps", in Advances in Digital Forensics IV, Ray and Shenoi ed., 2008.
- B. Carrier, “A Hypothesis-based Approach to Digital Forensic Investigations, Dissertation from Purdue University.
- P. Gladyshev, “Formalising Event Reconstruction in Digital Investigations”, Dissertation, University College of Dublin, 2004
- F. Cohen, "Two models of digital forensic analysis", IEEE/SADFE-2009, Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering, May 21, 2009