Research on Empirical Digital Forensics at the Naval Postgraduate School

Neil C. Rowe

U.S. Naval Postgraduate School
Monterey, California, USA
ncrowe@nps.edu
http://faculty.nps.edu/ncrowe

Empirical digital forensics

- This looks at real data and makes hypotheses about correlations and causations.
- This is similar to what field biologists do in observing and studying an ecosystem.
- It requires a reasonably large collection of representative data.
- This can use methods from machine learning and big-data processing.
- This talk will report some of our research projects:
    1. Machine translation of file paths
    2. Excluding uninteresting files from an investigation
    3. Finding interesting personal names on a drive
    4. Finding the most likely places for malware
    5. Assessing the success of factory resets on a mobile device

Main test corpus

- 3850 drives in our main corpus (977 without an operating system):
  - Images of 3203 non-mobile drives of the Real Data Corpus (RDC), a collection obtained from used equipment purchased in 32 non-US countries around world over 20 years, of which 95% ran the Windows operating system.
  - 411 mobile devices from the RDC, research sponsors, and our school.
  - 236 randomly selected classroom and laboratory computers at our school (metadata and hash values only). These were big.
- The corpori are publicly available with access restrictions.
- Artifact data extracted from the RDC with the Bulk Extractor tool: email addresses, phone numbers, bank-card numbers, GPS data, IP addresses, URLs, keyword searches, zip files, and rar files.

Methods we use from machine learning

- Much of our work involves extracting probabilistic clues from forensic data.
- Explainability of results is important to us. So we use:
  - Naïve Bayes: $o\left(U \,\middle|\, E_1 \& E_2 \& \ldots \& E_N\right) = o(U|E_1)o(U|E_2) \ldots o(U|E_N)o(U)^{1-N}$ where “o” is “odds”
  - Linear models: $t = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_{11} x_{11}$
  - Case-based reasoning: Majority vote of the K “nearest neighbors” to the current case among cases seen before
  - Set covering: Results of a search in the space of possible boolean rules to find the set of rules with the best F-score
  - Neural networks: Multiple layers of linear models alternating with nonlinearities

Task 1: Translation of file paths

Why translate file paths?

- 3.6% of the language of file paths in the (non-US) part of our corpus is not English or computer terms, ignoring punctuation and digits.
- Much of this non-English language is important for investigators as it often represents user-created files.
- The language of the file name is often the language of the file.
- Translation of file names need not be perfect since preliminary investigations only need to decide file relevance.
- Translation of everything first to a single language like English is the easiest.

Obstacles to path translation

- Sending a whole path to a translator errs on some interpolated English words when English should be echoed.
  - Systran translated “Temporary Internet Files” on a Mexican drive as “Temporary Internet You Case Out”.
- 23.1% of our paths changed language at least twice. We must translate each directory segment separately.
  - Example : Documents and Settings/defaultuser/Mes documents/Ma musique/Desktop.ini.
- There often aren’t enough character bigrams in a path to adequately guess the language.
  - Tool LA-Strings thought "obj viewsspt viewssrc vs lk“ was most likely Latvian.
- Country of origin is not always a good predictor of the language – e.g. Chinese is all over the world.

Our approach

1. Use SleuthKit and Fiwalk tool to get paths, convert them to UTF-8.
2. Exclude paths without a word of at least three characters that is not known English or not a computer code like “jpeg”.
3. For each directory:
   a) Extract the words in each path in a directory
   b) Get probabilities of languages using five clues.
   c) For each path segment in the directory:
      i. Infer the most likely language of each segment using three clues.
      ii. Translate the segments using one of three methods.
   d) Insert the translated words into original paths with analogous punctuation and case.
   e) Put translated path into DFXML metadata with new tag <englishfilename>.

Example translations we produced

Applications/Microsoft Office X/Office/Assistenten-Vorlagen/Kataloge/Kapsel

was translated to:

Applications/Microsoft Office X/Office/Assistants-Were-present/Catalogs/Cap

top.com/تصميماتي/السلسلة المعلوماتية.jpg

was translated to:

top.com/My designs/The computer-based series.jpg

Note analogous punctuation and case to the originals.

Sources of dictionary and translation data

34 languages currently handled, 1.2 million words

- English wordlists (currently 403,000 words) from several online sources
- Wikipedia: Good for everyday words (but most one-letter and two-letter words excluded to handle code-like names like “ab8e6rs”)
- Google Translate output of the 32,015 English words occurring at least 10 times in the corpus (except when identical): Good for technical words
- Transliterations of 18 European languages
- Manual entry of common computer abbreviations
- Inferred compound words by automated splitting and testing: Both to recognize the language and get the translation

We must address transliteration

- Many users attempt to do their languages on an English keyboard.
- This means they transliterate characters as necessary.
- The mapping is straightforward for European characters, but more complex otherwise.
- We create transliterated dictionaries to match with words in file paths, for the 18 most unproblematic languages.
- This is particularly helpful for Spanish and French.
- It didn’t work well for Arabic, which has many transliteration ambiguities.

Automatically finding compound words

- Automated analysis by splitting found 185,248 potential compound words to check, in the unrecognized words of the corpus.
- To reduce false alarms, splits had to involve words of at least four characters (except for Chinese), where both were known words of the same language.
- English examples: arabportal, mainparts, cityhospital, seatdisplay.
- Recognizing foreign-language compounds permits automated inference of a translation.
- Examples: horadormir -> hour sleep, ventadirecta -> sale direct, producktregistierung -> manufacture registration, weichzeichnen -> flexible chart.

Aggregation of directory words

- Directory: WINNT/Profiles/adrian/Menú Inicio/ Programas/Accesorios/Multimedia on Mexican drives contained:
  - Control de volumen.lnk
  - Grabadora de sonidos.lnk
  - Reproductor de CD.lnk
  - Reproductor de medios.lnk
- Words extracted for this directory:

  control de volumen grabadora de sonidos reproductor de cd reproductor de medios
- All are Ascii. But 11/12 words are in a Spanish dictionary, 2/12 are in an English dictionary, 3/12 are in an computer-term dictionary.
- So guess this directory is Spanish.
- Weight a language by inverse of log of size of its word list (following Zipf's Law).

Character distributions (unigrams)

- We compute conditional probabilities of a language given its character based on the dictionaries. E.g.: “a” with umlaut has probability 0.54 for Finnish, 0.30 for Swedish, 0.11 for German, 0.05 for other languages.
- Weight of a language: $\exp[(1/M)\sum_{i=1}^{M}\ln(\max(p_{i,L}, c_{i,L}))]$

  ranging over given words where p is the conditional probability and c is a lower bound for previously-unseen characters.
- We also assign characters to one of 20 categories by Unicode codepoint numeric range.
  - This enables us to assign never-seen characters to categories.
  - It also permits statistics on the categories for each language. This gives another way to identify the language.

Other methods to identify the language were tested

- LA-Strings: A character-bigram tool.
- Character type: 20 broad classes of characters.
- Country of origin: We used a standard table of language percentages by country.
- Keywords in the path: Certain words indicate language encodings, like standard abbreviations for languages.
- Inheritance from the languages of the directory above a given directory.

Combining the language clues

- Combining clues for a directory language L:

$$c_d w_{L,dictionary} + c_c w_{L,characters} + c_o w_{L,country} + c_k w_{L,keywords} + c_l w_{L,LA-Strings}$$

Justification: Clues may be missing, so situation is disjunctive.

- Combining clues for a path segment for L:

$$w_{L,dictionary} w_{L,characters} \sqrt{w_{L,directory}}$$

Justification: All three clues must be strong for a good candidate, so situation is conjunctive.

Testing clues in directory language identification

Factors	Raw accuracy	Adjusted accuracy
All	0.721	0.904
All without character types and inheritance	0.798	0.934
All without LA-Strings	0.694	0.904
All without dictionary lookup	0.662	0.836
All without character distributions	0.703	0.898
All without country	0.722	0.886
All without path keywords	0.793	0.897
Just dictionary lookup	0.649	0.857
Just character distributions	0.359	0.775

"Adjusted accuracy" combines transliterated with untransliterated, ignores confusion of English with untranslatable, and weights misidentification of English by 1/3. Conclusion: Character types and inheritance do not provide useful clues, LA-Strings maybe, others yes.

Confusion matrix for directory-segment language identification on random sample of 3518 directories

Overall adjusted accuracy was 93.7%. We got 93.5% on a different random sample of 29 million new drives, so there was little training bias. “t-” means transliterated. Rows denote true language.

	ar	de	en	es	fr	he	it	ja	ko	nl	ru	tr	zh	t-de	t-es	t-fr	t-he	t-hi	t-it	oth	un
ar	799		2																	8	3
de		14	3											36							10
en			301											2	4	1			1	3	273
es			4	107										2	370					11	85
fr			2		25											9					2
he						179															1
it			1				9								1						
ja								6													1
ko			1						17												1
nl			1							3											1
ru											2										
tr												17								8	6
zh			1										67								1
t-ar			2																	1	8
t-de														2							
t-es				1											80						1
t-fr																8					1
t-he																					1
t-hi			1											1						8	3
t-it			1												1				3		
oth	7		1									2								9	5
un			7												3	2	2			8	952

Testing of path-segment translation

We used 200 examples each and judged results ourselves.
Conclusion: Google Translate is significantly better than the others.

Language /Measure	Spanish	French	Japanese
Word-for-word OK	.72	.74	.57
Systran OK	.65	.61	.75
Google Translate OK	.81	.80	.92
None OK	.07	.03	.04
Word-for-word best	.55	.65	.48
Systran best	.52	.55	.48
Google Translate best	.78	.75	.85

Example translations

Language: words for translation	Word-for-word translation	Systran translation	Google Translate translation
Spanish: entren ser lider	come be head	they enter to be leader	come to be leader
Polish: magazyn kratownica	repository truss	warehouse grate	storage grid
Japanese: デスクトップの 表示	desktop display	Indication of desktop	Show Desktop
Arabic: مشكلة سقوط السارية	problem downfall applicable	Shaper of falling contagious	Problem of the fall of the applicable
Chinese: 陆行鸟饲 å x 手 x e x c	陆行鸟饲 å x hand x e x c	Goes by land the bird to raise å x x e x c	The land line Torikai å x hand x e x c
French: premierbaiser pps	first kiss pps	premierbaiser pps	premierbaiser pps
French: tetes de vainqueurs pps	heads of winners pps	suck winners ps	heads of winners pps

File names with a given number of words

Number of words	Percentage of translatable file segments having that number of words
1	20.5%
2	26.1%
3	17.9%
4	11.1%
5	7.4%
6	4.0%
7	2.4%
8	1.9%
>8	8.7%

So 46.6% of all translatable file segments are one-word or two-word, and their translations could be provided by dictionary lookup.

Conclusions about file-name translation

- Translation of file paths in harder that it seems.
- Success in translation requires handling each segment of a path separately.
- Success in language identification requires aggregating words from the same directory over a corpus.
- Surprisingly, character bigrams didn’t help much. But dictionary lookup and unigrams did help.
- On translation quality, Google Translate was clearly the best. Systran performance was equaled by a simple word-for-word translation substitution.
- To translate to other languages, first translate to English and then to the target language.

Task 2: Inferring interesting files

Another project: Eliminating files uninteresting to a forensic investigation.

- In most investigations, only a small fraction of the files on a drive are worth study.
- Most investigators define “uninteresting” intuitively – can we make intuition more precise?
- Our definition: Files containing no user-created nor user-discriminating information. (This is for criminal investigations not involving malware.)
- Standard method: Eliminate files whose hash values on their contents match those in published hash sets like the NSRL (U.S. National Software Reference Library). Rationale: Well-known software or downloads do not help most investigations.
- Weaknesses: It doesn’t match many files. Standard sets have difficulty keeping pace with new software and Web pages, and they don’t include files created only once programs are installed such as configuration files and manifests.

Identify the uninteresting

Our experiments suggest that files be identified as uninteresting if they meet any two of the following criteria:

- Frequent hash values (on minimum number of drives)
- Frequent paths (on minimum number of drives)
- Frequent filename-directory pairs
- Unusually busy times for a drive
- Unusually busy weeks in the corpus (and file has an unusually common extension in that week)
- Unusually frequent file sizes (when associated frequently with the same extension)
- Membership in directories containing already-identified mostly-uninteresting files
- Paths containing known uninteresting directories
- Files with uninteresting extensions

Thresholds for these were obtained experimentally with goal of 1% error on each individually.

Example same-hash files

Hash: 4E5D7F227E64BF650249C90EFAF400D05939194B

Not in NSRL, size 35766, predominant extension pop, libmagic data “MS Windows Help Data”

3 occurrences as path Drivers/Pointing/APOINTNL.POP

1 occurrence as path $OrphanFiles/DRIVERS/Pointing/APOINTNL.POP (deleted)

3 occurrences as path 笔记本索å°¼Z1XZG驱å¨/Pointing/APOINTNL.POP

1 occurrence as path SONY SZ13C/Drivers/Pointing/APOINTNL.POP

2 occurrences as path sony z1xzg驱å¨/Pointing/APOINTNL.POP

1 occurrence as path _ony-z1/Pointing/_POINTNL.POP (deleted)

1 occurrence as path sony z1xzg驱å¨/Pointing/_POINTNL.POP (deleted)

1 occurrence as path Programme/Apoint/ApointNL.pop

1 occurrence as path VAIO Applications/Drivers/Pointing/APOINTNL.POP

1 occurrence as path Drivers Backup/FS4_Drivers/Drivers/Pointing/APOINTNL.POP

1 occurrence as path sonysz32Audio/å¤份的驱å¨/Alps Pointing-device for VAIO/

1 occurrence as $OrphanFiles/ALPSPO~1/ApointNL.pop (deleted)

2 occurrences as path $OrphanFiles/VAIO/DRIVERS/POINTING/APOINTNL.POP (deleted)

Example same-path files

Program Files/Common Files/Adobe/Help/en_US/Adobe Reader/8.0/WS1A103696-4D61-4dca-BA3D-BBA4D1823D82.html:

- 107 occurrences of size 7026 with hash 16F64E648E044B1AF68D233394E9BBA7AE61E96E (not in NSRL hash values)
- 31 occurrences of size 6264 with hash A632411D9A48C6233CE53BF45D7B305F6ADBD70D (in NSRL hash values)
- 1 occurrence of size 7146 with hash 4C84A4E0E7C9ABAF829FE6E73DA67B5EC0558439 (not in NSRL hash values) (possible malware?)
- 9 occurrences with no hash, either size 7026 or 6264

Example common bottom directory/filename pairs

Subpath: intro/congratulations.jpg, no occurrences of any hash value in NSRL

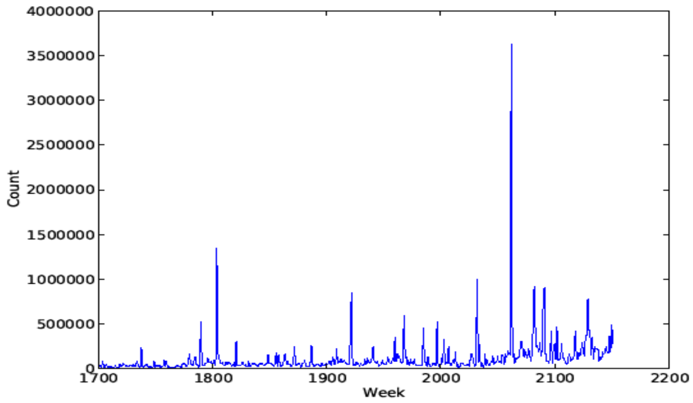
- 29 occurrences size 171079 with hash 1A8D4B336DB19B9D44068F3066D574232119659A
- 1 occurrence size 101697 with hash 8005ED87B2F92C10A52FDB768C91E11A4B84C486
- 1 occurrence size 171079 with hash 114E22142FDC857668F6EFF7D57B2412C4D221EA (deleted)

Example busy time on a drive

- 168 files on an Indian drive were created around 2010-04-28 12:27:30.
- All were in directory Program Files/ pdfconverter.com.
- 82 did not have hash values in NSRL.
- They had a variety of extensions.
- This is a frequent pattern for a software download.

Example busy week on a corpus

- Some weeks had especially large numbers of file/directory creations over the entire corpus. The count per week had occasional spikes (see graph). We looked for spikes 100 times the average and files in directories then that had at least 10 times more than the average usage.



- Two such records not in NSRL:
  - ProgramData/Microsoft/User Account Pictures/Default Pictures/..
  - ProgramData/Microsoft/Windows/Start Menu/Programs/Windows Media Player.lnk

Example frequent size for files

- 2106 files in our corpus had size 1691, classified as an unusually common size.
- Only 168 of those had hash values in the NSRL.
- One hash value for size 1691 occurred 954 times, and another 250 times. But many occurred only once.
- File extensions that were unusually common were .resx (1306 occurrences), .gif (397), .dat (257), and .manifest (146).
- The .resx and .manifest files were all in the Microsoft Windows operating system.

Example uninteresting directory by “contagion”

- Program Files/Chessmaster Challenge/Images has 126 files on one drive in our corpus.
- Many (but not all) files were created at the same time, apparently by download.
- The downloads are marked as uninteresting by the time-uninterestingness method.
- Then by “contagion”, the remaining files in this directory were considered uninteresting as well.

Example uninteresting top directories

- Paint Shop Pro (software)
- Program Files/NFS 6 (game)
- ProgramData/Microsoft/Crypto (operating system)
- FOUND.008 (operating system)
- System Volume Information (operating system)
- Applications/AudoCad 2009 (software)

Example uninteresting extensions

- dat (operating-system data)
- mui (Microsoft installer)
- tga (graphics)
- exe (executable)
- e01 (disk image)
- lex (lexicon)
- lproj (configuration)
- pgp (cryptography)
- ibf (game data)

Eliminating uninteresting files with our system

Set	Number of files (millions)	Number of distinct hashes (millions)
Full corpus	262.7	35.80
Full corpus minus NSRL	200.1 (76.2%)	33.42 (93.4%)
Same minus those files identified by 2 of 9 methods as uninteresting	56.2 (21.4%)	19.27 (53.8%)
Same minus uninteresting hashless or default-hash files	46.0 (17.5%)	19.26 (53.8%)
Same plus potentially interesting files	59.5 (22.6%)	20.32 (56.8%)

Comparison of our 9 uninteresting-file clues on a 20,000-file training set

Method	Parameter values	True positives	False positives	False negatives	Precision	Recall
Frequent hashes	mindrives=5	10816	25	7555	.9977	.5888
Frequent paths	mindrives=20	11430	26	7941	.9975	.5677
Frequent immediate directories	mindrives=50, segcount=2	7824	22	10547	.9972	.4259
Created with many other files	min count within minus = 50	7408	55	10963	.9926	.4032
Busy weeks	weekmult=5, pathmult=100	6500	13	11871	.9980	.3538
Frequent sizes	mincount=10, mindev=20	4093	85	14278	.9797	.2228
Directory is mostly uninteresting	mindircount=40, fracmin=0.8	8137	534	9999	.9813	.3791
Uninteresting top directory	none	14274	60	4133	.9958	.7755
Uninteresting extension	none	3762	9	14609	.9976	.2048

Clues to explicitly-interesting files weren’t so good

Method	Mincount and minfrac	True positives	False positives	False negatives	Precision	Recall
Hashes occurring only once for a given path	30, 0.8	3	451	1204	.0066	.0025
Path names occurring only once for a given hash	30, 0.8	2	201	1205	.0099	.0017
Files created in atypical weeks for their directories	30, 0.8	0	20	1207	.0000	.0000
Inconsistency between extension and magic-number analysis	None	5	2602	1202	.0019	.0041
Inconsistency in file size for the same hash value	None	0	67	1207	.0000	.0000
Explicitly identified interesting extension or directory	None	552	1478	655	.2719	.4573

Task 3: Finding useful personal names on drives

Forensic use of personal names

- Email addresses, phone numbers, bank-account numbers, street addresses, and IP addresses are useful forensic artifacts in many investigations.
- Personal names could be even better since most criminal and intelligence investigations focus on people.
- However, regular expressions are not much help in finding personal names: Checking long wordlists is necessary.
- And many possible personal names found on drives are useless – often they are sales contacts or English words not usually used as personal names.
- Thus we developed methods for filtering possible personal names found on drives.

Example personal-name candidates

Good:

- “John Smith” 555-1234
- smith_john_r@hotmail.com
- ingledorfer@yahoo.com
- sendto: mark smith

Bad:

- Smith Electric, 862 Main
- Ask Siri
- Herman Melville
- Huckleberry Finn
- John
- Mark field

Our approach

- Bootstrap on existing Bulk Extractor (digitalcorpora.org/downloads/bulkextractor) output for finding email addresses, phone numbers, and Web URLs.
- Find names in the “context” argument supplied in Bulk Extractor output, the 16 characters before and after the artifact.
- This will be much faster than search for names directly since regular expressions can find these types of artifacts quickly.
- Rate the names using 11 clues.
- Combine evidence with either Naive Bayes, a linear model, or case-based reasoning.
- Find the best threshold for classifying true personal names in a manually tagged training set.

Some statistics

- The program we wrote found 302,242,805 personal-name candidates from 2222 drives in our corpus with a name.
  - Names were one to six words.
  - 5,921,992 were distinct (though names like “John” are ambiguous).
  - There were 61,365,153 files on these drives.
  - Several hundred drives had no files but many names, apparently due to imperfect disk wiping.
- Personal-name candidates found:
  - 95.6% in email data
  - 1.0% in phone-number data
  - 3.4% in URL data (only 6.5% of which were useful)
  - 0.0% in CCN (bank-card) data

Splitting and combining strings to find names

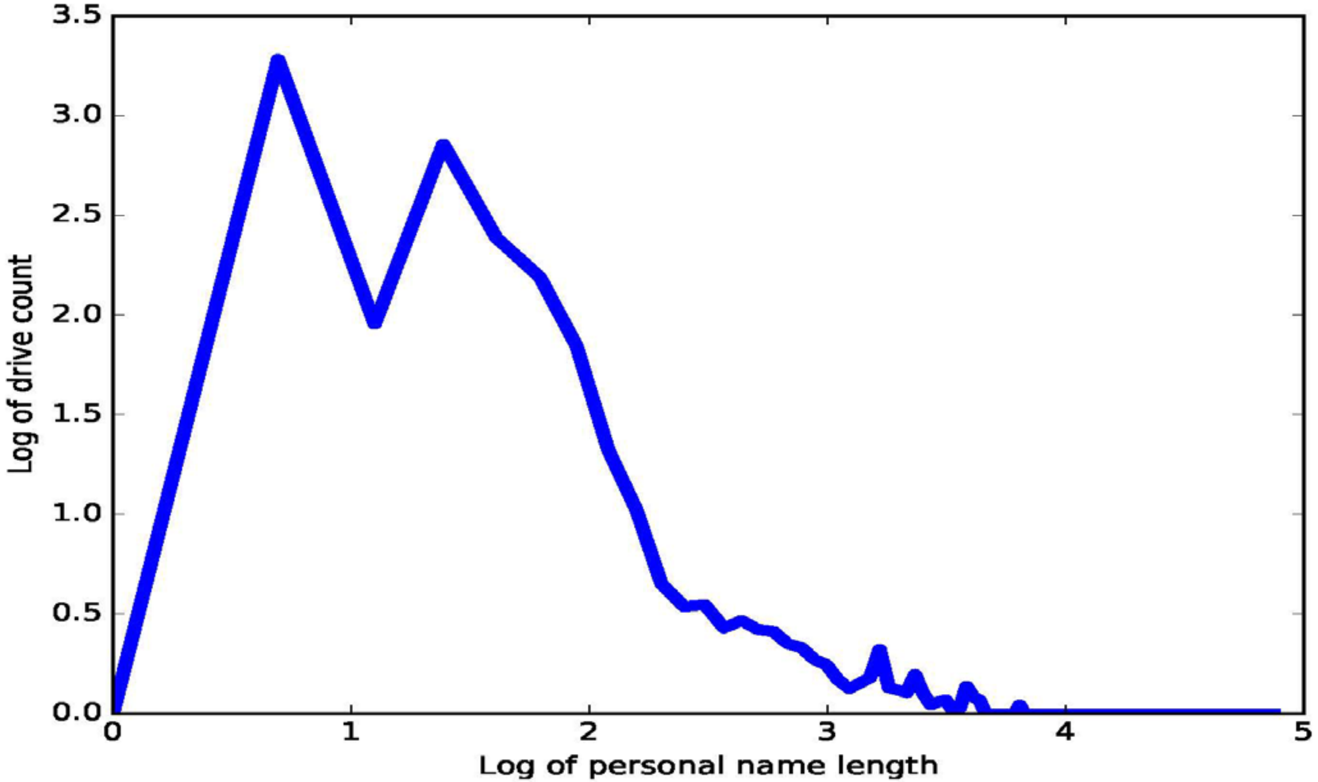
- Names were frequently compounds, e.g. “johnjsmith”.
- Words not found in our dictionaries or personal-names list were split every possible way to find recognizable words.
- Names separated by a space, “_”, or “.’ can often make multiword names, e.g. “John A Smith” and “john_a_smith”.
- Names can be combined with punctuation, e.g. “John Anthony Smith”.
- Name candidates obey some constraints. Consider:
  - “John Smith john.a.smith@Hotmail.com”. We can extract “John Smith” and “John A. Smith”, but we should not combine them because they have words in common.
  - “Mark field” can’t be a name with inconsistent capitalization.

Factors for rating personal-name candidates

- Length in characters
- Lower case, upper case, initial capital letter, or mixed case
- Whether the name is delimited by punctuation marks
- Whether the name is followed by a number (as often in personal email addresses)
- Whether the name is a single word or multiple words
- Whether the name is commonly used as a non-name
- The total count of the name over all drives
- The number of drives on which the name occurs
- The average number of occurrences of the name per drive
- Whether the name occurs near a .com, .org, .gov, .mil or .biz, excluding mail servers
- Strength of a nearby email address or phone number

Number of drives on which a name occurs as a function of name length in characters.

This says we should weight the number of drives on which a name occurs by this curve. This applies to multiword names too.



A training set

- 5639 name candidates randomly selected from all the 302 million name candidates found.
- We inspected each manually.
- 1127 we identified as forensically interesting personal names and 4522 as not forensically interesting.
  - Most decisions were easy from context.
  - A few required Internet research.
  - A few were ambiguous and were removed from the training set.

Results on evaluating clue strengths (1)

Clue	Odds on training set	Standard deviation on training set
Length ≤ 5 characters	0.168	0.006
Length > 5 characters	0.272	0.006
All lower case	0.319	0.006
All upper case	0.150	0.015
Capitalized only	0.172	0.006
Mixed case	0.134	0.012
Delimited both sides	0.361	0.009
Delimited on one side	0.301	0.013
No delimiters	0.158	0.004
Followed by a digit	1.243	0.077
No following digit	0.214	0.004
Single word	0.236	0.005
Multiple words	0.249	0.007
Ambiguous word	0.055	0.004
Not ambiguous word	0.294	0.006

Results on evaluating clue strengths (2)

Clue	Odds on training set	Standard deviation on training set
≤ 9 occurrences in corpus file names	0.451	0.011
> 9 occurrences in corpus file names	0.162	0.004
Normalized number of drives ≤ 153	0.421	0.009
Normalized number of drives > 153	0.112	0.004
≤ 399 occurrences per drive	0.189	0.004
> 399 occurrences per drive	0.664	0.025
Organizational domain name nearby	0.009	0.001
No organizational domain name nearby	0.760	0.015
Prior to any clues	0.241	0.004

Overall results

Training model	Average best F-score with cross-validation	Threshold at average best F-score
Naïve Bayes, odds form	0.6681 (100-fold, no cross-modal clues)	0.2586 (0.907 recall with 0.570 precision)
Linear model	0.6435 (100-fold, no cross-modal)	0.3807
Case-based reasoning	0.6383 (10-fold, no cross-modal)	Average best multiplier 1.97
Set covering	0.6000 (no cross-modal)	None
Naïve Bayes including rating on nearby email addresses with linear model	0.7990 (100-fold, with cross-modal)	0.2889 (0.900 recall with 0.695 precision)

Side issue: Identifying the owner of a drive

- Easy ways to identify the owner of a drive don't always work:
  - Often the common names are vendor or help contacts.
  - Often possible names are used as non-names.
  - User directories can be aliases, they only show people who log in, and do not give frequencies of use.
- Thus it's better to use our ratings of personal names to determine ownership.
- We found that the highest-count personal name with rating over 0.2 was the drive owner in 8 of 11 cases where we could confirm ownership. In the remaining 3 cases, one was second, one was fourth, one was 20th (apparently a drive used by many people).

Conclusions on finding personal names forensically

- Our methods eliminated 71.3% of the personal-name candidates found on a drive, at best F-score of 67.4%.
- This should reduce the workload for criminal and intelligence investigators by a factor of 3.5.
- Reduction can be further improved by using cross-modal email ratings if available.
- This work bootstraps on routine Bulk Extractor data, so it requires little additional processing time. Many investigations routinely run Bulk Extractor to find artifacts.
- Once the interesting personal names are found, we can build better social-network graphs connecting them.

Task 4: Identifying the most likely places for malware on a drive

The malware identification problem

- To find malware, we can use:
  - Signature checking (can’t find truly new malware)
  - Static anomaly analysis (can make mistakes)
  - Behavioral analysis (can take time)
  - Reputation of source (you don’t always know the source, and reputations can be bought)
- Checking systems for malware can be arduous. A full signature scan of a typical computer’s secondary storage takes hours.
- Hence anti-malware software usually offers “quick scans” that checks the most likely locations.
- But there’s little empirical justification for where they look.
- This work tries to remedy that.

Malware identification methods we tested

- Files in our corpus whose SHA-1 hash values were tagged as threats in the database of the Bit9 Forensic Service.
- Files whose hash values matched those in the Open Malware corpus of 3 million files.
- Files whose hash values matched those in the VirusShare database of about 18 million files.
- Files identified as threats by Symantec antivirus software. This was done only on a sample of files extracted from the corpus.
- Files identified as threats by ClamAV open-source antivirus software in the same sample of files tested by Symantec.

Whitelists we tested

- Hashcodes of files explicitly whitelisted by Bit9 on our corpus, minus those identified as malware by our five methods.
- The June 2014 version of the National Software Reference Library Reference Data Set (NSRL) of commonly seen software hash values. It does not guarantee to exclude malware.
- A random sample of our corpus minus those identified as malware by any of our five methods. It likely included unrecognized malware but the number was likely too low to influence analysis results, judging by the rate of identified malware.

Intersections of the hash sets

	BW	NW	RW	BT	BV	OM	VS	SM	CA
BW	707917/705004	192121/192090	57301/57301	22/0	0/0	5160/0	929/0	94/0	298/0
NW	192121/192090	2167233/2167048	140841/140834	591/0	4554/0	2582/0	4093/0	6/0	43/0
RW	57301/57301	140841/140834	809168/809158	0/0	512/0	0/0	2363/0	5/0	0/0
BT	22/0	591/0	0/0	239284/238704	0/0	418/409	28/28	289/280	400/393
BV	0/0	4554/0	512/0	0/0	10062/5462	113/25	6/0	0/0	1/0
OM	5160/0	2582/0	0/0	418/409	113/25	7338/4786	187/121	745/719	1002/981
VS	929/0	4093/0	2363/0	28/28	6/0	187/121	151706/145449	19/19	33/32
SM	94/0	6/0	5/0	289/280	0/0	745/719	19/19	1434/1401	880/877
CA	298/0	43/0	0/0	400/393	1/0	1002/981	32/32	880/877	2598/2555

BW=Bit9 whitelist, NW=NSRL whitelist, RW=corpus whitelist, BT=Bit9 threat, BV=Bit9 vulnerable, OM=OpenMalware, VS= VirusShare, SM=Symantec, CA=ClamAV. Counts: files/hashes.

Rates of malware in different sources (files/hashes)

	Bit9 threats in our corpus	Bit9 identified vulnerable in our corpus	Open Malware corpus in our corpus	VirusShare corpus in our corpus	Symantec Endpoint Protection on corpus sample	ClamAV Antivirus on corpus sample
School drives	.000000, .000000	.000099, .000004	.000092, .000070	.000047, .000020	.000000, .000000	.000009, .000010
Real Data Corpus	.000049, .000186	.000249, .000481	.000200, .000741	.005296, .002696	.000057, .000188	.000114, .000205
Mobile drives	.000052, .000105	.000000, .000000	.000061, .000152	.326605, .210274	.000038, .000103	.000052, .000124
Microsoft Windows drives	.000039, .000136	.000191, .000167	.000174, .000338	.000147, .000083	.000045, .000078	.000085, .000132
Embedded files	.000083, .000039	.000033, .000012	.000482, .003000	.000138, .000084	.000334, .000678	.000892, .000570
All drives	.000139, .000141	.000166, .000160	.000156, .000340	.004741, .001597	.000043, .000083	.000083, .000141

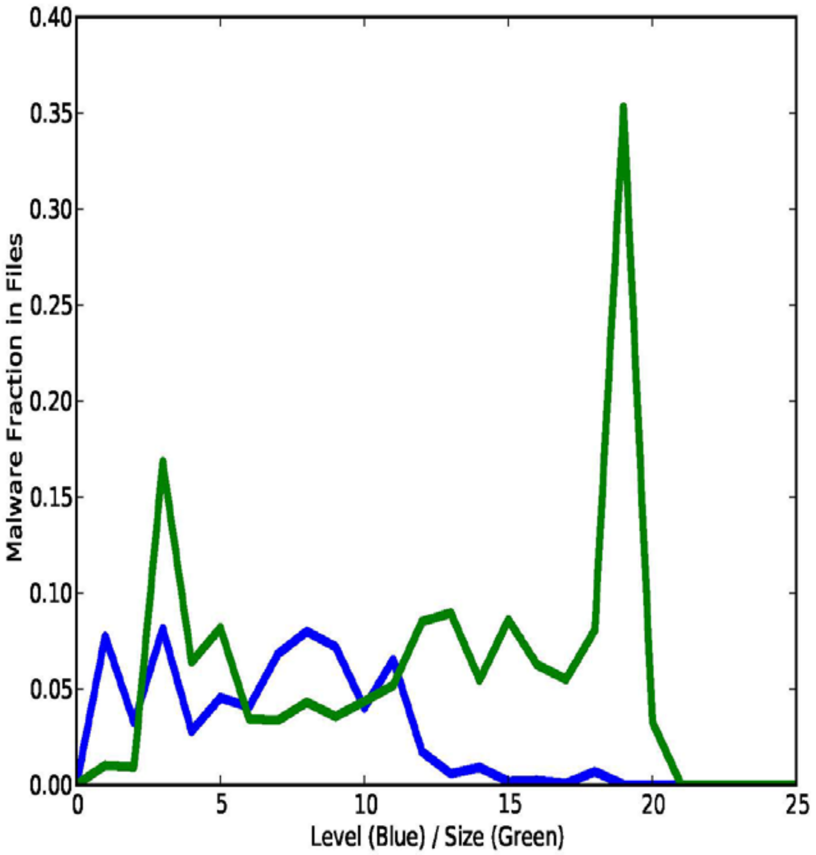
Measured clue strengths in standard deviations above expected value (based on files/based on hashes) (1)

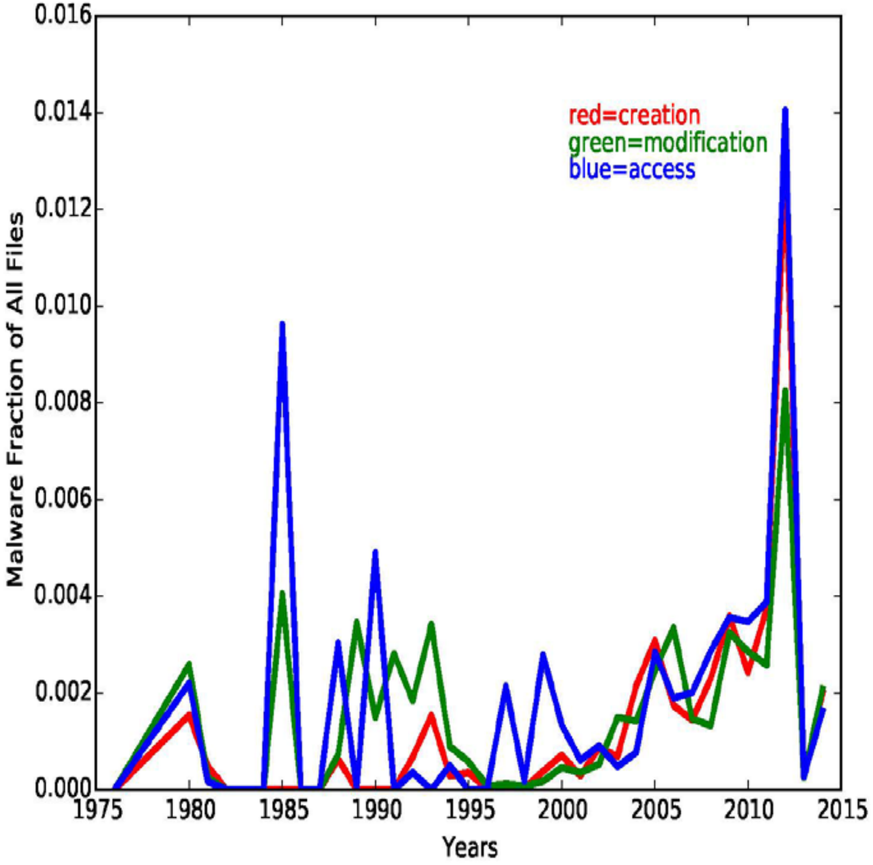
Malware set	Bit9	Open Malware	VirusShare	Symantec	ClamAV
Total count in corpus identified as malicious	35,202/ 1,201	763,199/ 7,331	1,006,412/ 151,621	11,085/ 626	25,972/ 1,662
Total count in corpus identified as nonmalicious	12,094,989 /303,332	12,094,989 /303,332	12,094,989 /303,332	12,094,989 /303,332	12,094,989 /303,332
File size 0 or 1	-1.3/-0.3	-6.1/-0.7	1750/-0.9	-0.7/-0.2	-1.1/-0.3
Rounded log file size = 5	-8.5/-5.5	-223/-19.5	129/112	-27.7/-5.2	-9.6/-7.1
Rounded log file size = 10	-44.8/9.3	287/22.5	-187/-9.0	-33.6/-6.5	-36.2/-3.1
Rounded log file size = 15	-11.2/2.5	-33.9/23.4	-28.1/16.8	10.0/4.6	35.0/16.9
Level = 1	121.0/26.2	-46.6/9.7	-77.6/-47.1	-3.6/-2.4	124/14.7
Level = 5	-37.9/2.5	-65.7/7.4	5.2/-85.2	-20.7/-5.3	-29.8/5.5
Level = 10	-39.4/-6.6	-182/-15.8	-22.8/-8.2	11.0/-1.2	-17.7/-8.0
Level = 15	-23.8/-2.3	-110/-5.3	-99.2/-24.5	-13.4/-1.6	-17.5/-2.7
Deleted file	130.0/4.2	-213/3.1	595/1159	17.0/-1.4	115/10.6
Extension/ libmagic incompatible	-116.5/6.4	-108.4/-4.1	-161/-60.0	-5.3/9.1	-2.6/6.6
Odd creation time	88.8/17.7	-79.0/9.1	-73.1/-46.1	-6.9/-2.0	4.3/13.2
Rare hash value	-0.3/na	2.1/na	-1.6/na	-0.2/na	-0.3/na
Rare extension	996/2151	75.6/583	175/-24.4	3068/2874	1782/3287
Double extension	-25.1/-1.6	-119/-6.2	-77.1/-17.0	-5.9/12.8	-19.5/5.3
Long extension	-1.9/-0.9	-10.4/-1.5	17.1/8.0	-1.3/-0.7	-1.5/-0.2
Encryption extension	-8.5/-1.9	28.8/-4.2	-41.2/-17.6	-4.4/-0.6	-7.3/-2.2
Odd characters in path	-10.4/6.4	-90.1/7.2	-32.8/29.0	-26.5/-3.6	-38.2/-0.6
Repeated pattern in path	1.7/-0.4	-12.3/0.8	-8.5/16.4	-1.2/-0.3	13.3/75.5
Misspelling in path	-6.5/-1.2	-30.4/-1.2	-354/-11.9	-0.4/-0.8	-2.6/0.1

Measured clue strengths in standard deviations above expected value (based on files/based on hashes) (2)

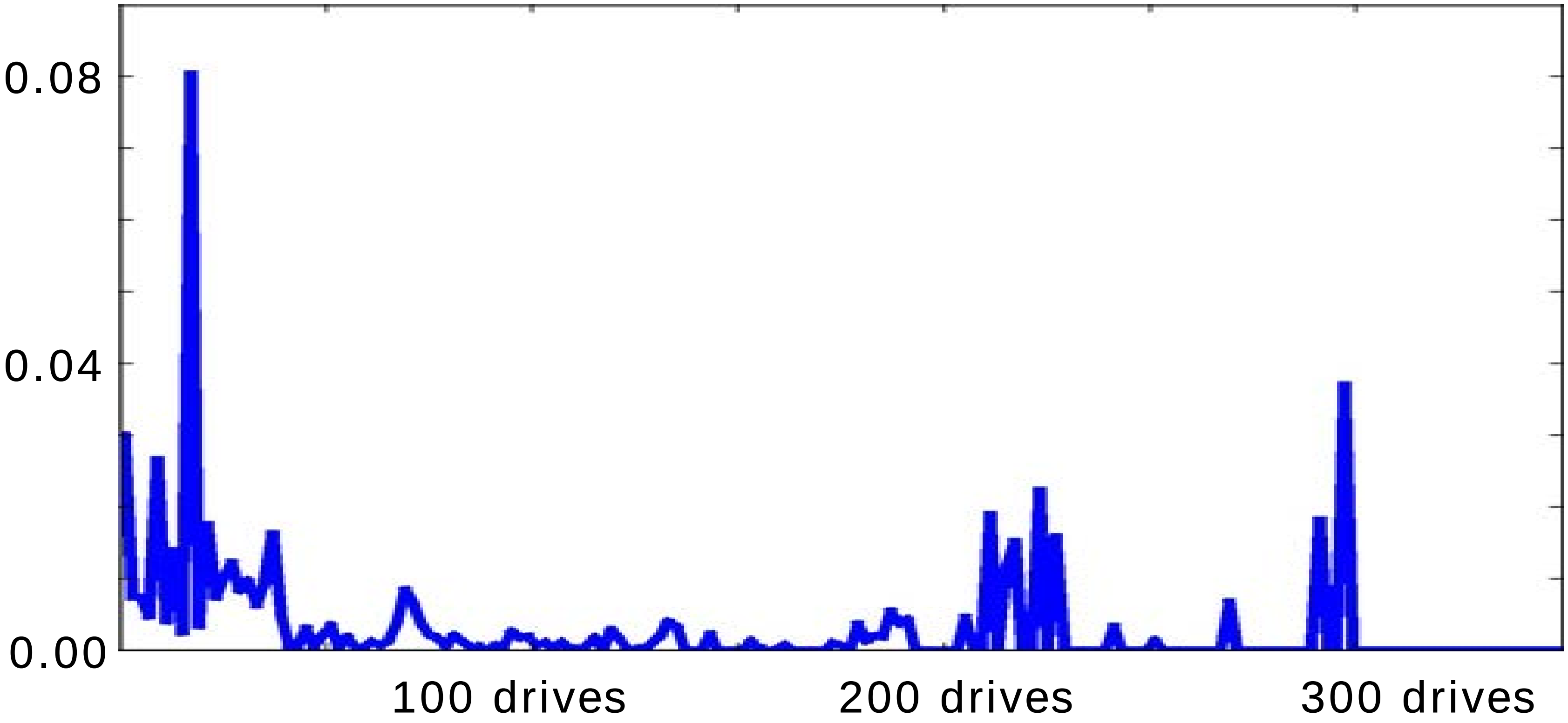
Malware set	Bit9	Open Malware	VirusShare	Symantec	ClamAV
E: None	-78.5/-10.5	-355/-27.5	226/-17.9	-42.9/-8.2	-67.5/-12.8
E: Photograph	-15.8/-5.3	-100/-14.8	9.8/110	-4.3/-3.1	-18.9/-4.6
E: Link	-5.9/4.8	-36.5/-1.8	-24.9/-15.9	-4.1/-1.0	-6.5/-1.1
E: Video	-8.5/-2.2	-16.8/-5.0	-28.5/-12.6	-4.7/-1.6	-7.2/-2.5
E: Executable	134/54.5	1979/162	-199/-166	158/18.2	26.6/25.3
E: Drive image	-3.6/-1.1	-16.2/-2.3	25.2/-8.1	-2.0/-0.8	-3.1/-1.3
E: Query	-12.5/-0.9	-58.1/-2.3	-24.4/20.8	-7.0/-0.7	-10.7/-1.1
E: Installation	758.1/6.4	-70.9/-5.0	-125/-43.8	-12.4/-2.2	702/-0.4
E: Networking	-5.0/-0.8	-22.5/-0.9	-3.7/-8.5	-2.9/-0.6	-4.4/-0.9
E: Hardware	-0.9/-0.4	595/1.8	-26.1/-14.0	-2.8/-0.9	-4.3/-1.5
E: Engineering	-13.8/-2.5	-64.2/-6.2	-64.2/-27.0	-7.7/-1.8	-11.8/-3.0
E: Miscellaneous	-3.0/-1.5	-10.2/-1.1	48.9/19.0	-2.0/-1.1	4.8/2.1
T: Root	51.8/18.3	-75.9/6.8	45.5/25.2	-60.4/-0.1	50.2/6.8
T: Hardware	115.1/-2.3	22.2/3.0	125/47.9	47.8/11.3	21.9/42.4
T: Temporaries	20.7/2.6	-219/8.1	-279/-121	33.3/14.4	-25.6/-2.3
T: Games	57.5/-2.5	-60.4/3.5	-27.0/-46.1	12.5/-2.7	-12.8/-3.2
T: Miscellaneous	40.4/24.7	-40.3/20.2	-50.5/-42.1	-5.8/-1.8	14.9/33.8
I: Op. system	-43.0/6.0	704.2/9.5	276/-56.0	-3.6/-8.8	-27.4/-9.1
I: Backup	206/-5.7	-207/-16.6	-246/-80.0	-25.2/-5.5	256/-5.7
I: Audio	244/-2.2	-45.7/-3.8	461/90.7	-2.0/-2.6	-4.7/0.3
I: Data	-24.3/2.5	-137/-6.4	120/60.0	-12.8/-2.0	-23.5/-4.4
I: Security	7.0/14.5	85.2/20.8	49.8/18.3	65.1/20.1	12.1/2.0
I: Games	38.5/1.0	11.6/10.8	711/907	9.1/-0.2	-7.4/-0.5
I: Miscellaneous	59.3/15.8	50.7/28.4	48.2/-22.5	4.7/-1.5	103/50.1

Malware fraction in corpus by file depth, size logarithm, and years





Probability of malware versus number of drives on which a hash-value occurs



This is worrisome news for reputation-based malware detection, because a significant fraction of malware hash values occurred over many drives in our corpus.

Building a better “quick scan”

- We can rank files for malware inspection based on a Naive Bayes combination of our best clues.
- We used:

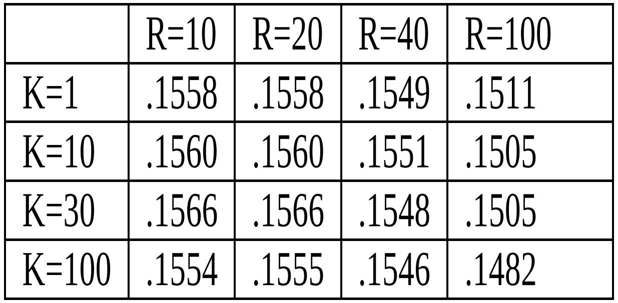
$$o(\mathrm{M} \mid (\mathrm{C}_1 \wedge C_2 \wedge \ldots \wedge C_M)) =$$

$$[o(\mathrm{M} \mid \mathrm{C}_1)o(\mathrm{M} \mid \mathrm{C}_2)\ldots o(\mathrm{M} \mid \mathrm{C}_M) / (\mathrm{o(M)})^{N-1}]^{(1/\mathrm{N})}$$

$$o(\mathrm{M} \mid \mathrm{C}) = [(n(\mathrm{M\&\,C}) + (K * \mathrm{n(M)} / \mathrm{n(O)}) / \mathrm{n(M)}] /$$

$$[(\mathrm{n(O\&\,C)} + K) / \mathrm{n(O)}]$$

The effect of R and M on performance was not dramatic:



	R=10	R=20	R=40	R=100
K=1	.1558	.1558	.1549	.1511
K=10	.1560	.1560	.1551	.1505
K=30	.1566	.1566	.1548	.1505
K=100	.1554	.1555	.1546	.1482

Testing and results on contextual identification

- We tested three random partitions of our corpus:
    - 612,818 instances of malware and 128,776,919 instances of non-malware for training.
    - We used one half for training and one half for testing.
    - Recall values were 0.343, 0.305 and 0.333.
    - Precision values were 0.213, 0.211, and 0.211.
    - F-scores were 0.263, 0.249, and 0.259.
- Note little variation in the results with training sample.
- If one is willing to accept a much lower precision of 0.010 with our methods, one can obtain a better recall of 0.650.
- By comparison, selecting only the executable files gave 0.005 precision, 0.190 recall, and F-score of 0.0097. Hence our methods give 5 times better precision with 1.7 times better recall over inspecting executables alone.
- Selecting only the files in the operating system gave 0.003 precision and 0.189 recall, even worse.

Conclusions on malware localization

- Different malware identification methods find significantly different things.
- Some classic clues to malware (e.g. rare file extensions and deletion status) were confirmed and others were not (e.g. misspellings, double extensions, and occurrence in the operating system).
- We got 5 times better precision (fraction of malware in files identified as malware) with 70% better recall (fraction of malware detected) than the approach of inspecting executables alone.
- Our methods also ran significantly faster than signature checking, and can be used before other kinds of malware analysis.

Task 5: Analysis of effects of factory resets on mobile devices

Introduction

- Increasing criminal activity uses mobile devices.
- Devices provide a way to erase user information ("factory resets") upon resale or discard of a device.
- Operating-system vendors claim that factory resets erase user data.
- But anecdotal evidence reports that resets do not delete all such data.
- Our work was the first systematic test of what resets do.

Detailed Experiments on Two Phones

- First we tested two phones:
  - Android Samsung Galaxy SIII
  - Apple iPhone 4S
- After taking images and resetting the phones, we emplaced specific user files.
- Emplacement included downloading files, visiting Web sites, taking pictures with the camera, and installing software.
- We checked whether the emplaced files remained after a factory reset.

Analysis: Android Test Phone

- It did a poor job of deleting user files on reset.
- No downloaded user files were deleted (including txt, doc, pdf, and ppt), and neither their caches.
- No camera images were deleted.
- Third-party software was deleted, but not Kindle and DropBox files.
- 968 files were deleted and 65 were added by the reset.
- Executables not reduced much, but “copies and backup” were mostly eliminated.

Some files left on the Android

File	Description
CACHE/Root/recovery/last_log	Ascii recovery log
SYSTEM/Root/addon.d \/blacklist	Four hexadecimal MD5 hash values
SYSTEM/Root/etc/apns-conf.xml	Ascii phone carrier IP address
SYSTEM/Root/etc/audio_policy.conf	Ascii audio devices listing
USERDATA/Root/media/0/amazonmp3/temp/log.txt	Ascii log file of Amazon Cloud Player
USERDATA/Root/media/0/Android/data/com.andrew.apollo/cache/ImageCache/3910b1e0ccab19bc46fd9db27cca49c9.0	Binary image cache data
USERDATA/Root/system/users/userlist.xml	Ascii User ID information
USERDATA/Root/drm/fwdlock/kek.dat	Lock data
USERDATA/Root/media/0/Android/data/com.dropbox.android /files/scratch/09thesis_regan.pdf	PDF document of previous phone user

Analysis: Apple iOS Test Phone

- Most user files were deleted except for those in the operating-system directories. (But arbitrary user files could not be downloaded.)
- Resets deleted 17,914 files and added 115.
- Files in operating-system directories went from 29,812 to 27,621, so they were not much affected.
- Major things left were caches, configuration data, and Facebook data.
  - Almost all files deleted were within the last month.
  - We found Ascii files after the reset despite Apple's claims of uniform encryption.

Analysis: iOS Test Phone

File	Description
System/InnsbruckTaos11B554a.N90OS/System/Library/PrivateFrameworks/Preferences.framework/SupplementalLocaleData.plist	Binary location and language settings
System/InnsbruckTaos11B554a.N90OS/usr/share/mecabra/ja /rerank.dat	Binary resource rankings
Data/Data/Keychains/keychain-2.db	Ascii keys
Data/Data/logs/lockdownd.log	Ascii security event log
Data/Data/mobile/Applications/B8AD4B05-2518-4570-8447-7BE2BFDA8F9F/Library/Preferences/com.apple.mobilesafari.plist	Ascii browser preferences
Data/Data/mobile/Library /BulletinBoard/SectionInfo.plist	Ascii bulletin board index
Data/Data/mobile/Library/Caches/com.apple.springboard /Cache.db-wal	Ascii screen cache for user "wal"
Data/Data/mobile/Library/Cookies/com.apple.itunesstored.2.sqlitedb	Ascii cookies for iTunes
System/InnsbruckTaos11B554a.N90OS/System/Library/PrivateFrameworks/Preferences.framework/ SupplementalLocaleData.plist	Binary location and language settings

Experiments: 21 Devices

- Some we could reset ourselves and they had data from previous users.
  - Some came from research projects at our school.
  - Some were personal devices.
- In addition, some pre-reset and post-reset image pairs came from the international Real Drive Corpus.
- Devices:
  - Apple iOS: iPhone 2, 2G, 4, 4S, some were modified OSs
  - Google Android: Samsung Galaxy SIII, HTC Droid Eris, Motorola Atrix 4G, Huawei U8500, HTC Flyer tablet
  - BlackBerry: 8100 Pearl, 8300 Curve
- 2 additional devices did not work at all, 2 could not be imaged by Cellebrite, and 3 failed on reset.

Summary counts on the 21 devices

File Count Type	Pre-reset	Post-reset
Total Files	349,915	200,987
iPhone Files	299,058	176,907
Android Files	50,846	24,058
Exact matches pre-reset and post-reset	140,320	140,320
Subsequent matches on filename and hash value but not all directories	34,228	36,540
Subsequent matches on hash value alone	9,269	12,911
Subsequent matches on full path alone	2,849	2,836
Subsequent matches on full path ignoring digits alone	6,448	256
Remaining unmatched	156,801	8,124

Extension Counts: Pre/Post-reset

Type of file	Pre/Post	Type of file	Pre/Post
E: No extension	36561/21078	E: Video	303/90
E: Operating system	106168/104406	E: Source code	1791/736
E: Graphics	98618/27522	E: Executables	3432/2856
E: Camera pictures	15443/3967	E: Disk image	13828/1932
E: Temporaries	733/159	E: Log	599/73
E: Web pages	1418/680	E: Copies and backup	7347/905
E: Documents	3089/1233	E: XML	5193/1045
E: Spreadsheets	425/356	E: Configuration	20788/18379
E: Compressed	601/278	E: Games	3741/1048
E: Audio	16427/8313		

Immediate Directory Counts: Pre/Post-reset

Type of file	Pre/Post	Type of file	Pre/Post
D: Root	1012/966	D: Data	18300/9771
D: Operating system	122625/117701	D: Programs	3616/2876
D: Hardware	1128/319	D: Documents	6211/1036
D: Temporaries	12141/2928	D: Sharing	7500/2368
D: Pictures	17950/4328	D: Security	2953/2749
D: Audio	10812/7814	D: Games	53722/0
D: Video	2570/0	D: Applications	84593/46696
D: Web	2714/277	Deleted files	20087/4181

Analysis within Files

- The Cellebrite Physical Analyzer can search for some categories of file contents:
  - Installed applications and their usage
  - Contacts, call logs, and cookies
  - Location, maps, wireless networks, and IP connections
  - User accounts, user dictionary, SMS, and passwords
  - Content file types and directory information
- Bulk Extractor
  - Open-Source tool for searching for particular content
  - Can extract from compressed files

Data Counts 1a (Post/Pre-reset)

Device	p1	p2	p3	P4	p5	p6	p7	p8
Type	I	I	I	I	I	I	I	I
App. Usage	0/0	0/0	1/199	0/0	1/125	1/56	9/23	0/23
Call Log	0/0	0/2	0/103	0/0	0/107	0/104	0/13	0/105
Contacts	0/0	0/0	0/209	0/0	0/1461	0/2366	0/0	0/284
Cookies	0/0	0/0	0/5	0/0	0/0	0/43	0/0	0/6
Installed Apps.	34/34	34/34	23/127	28/34	23/142	23/56	0/24	0/79
IP Connections	0/2	0/2	0/2	0/1	0/0	0/1	0/0	0/7
Locations	0/0	0/0	0/1	0/0	0/0	5/10	0/0	0/72
Maps	0/0	0/0	0/12	0/0	0/0	0/2	0/0	0/19
Passwords	0/6	0/5	0/0	0/1	0/0	0/0	0/0	0/0

Data Counts 1b (Post/Pre-reset)

Device	p1	p2	p3	P4	p5	p6	p7	p8
SMS Messages	0/0	0/2	0/30	0/0	0/1152	0/50	0/0	0/672
User Acts.	0/0	1/1	1/1	1/1	1/1	1/6	1/1	1/3
User Dict.	0/0	0/1	0/161	0/0	0/30	0/312	0/0	0/819
Wireless	0/0	1/1	0/1	0/0	0/0	0/0	0/0	0/0
Images	3714/3716	3715/3716	2488/16541	2631/2716	2488/25106	5888/14477	2491/2611	2488/13705
Audio	1/1	1/1	2/2512	1/1	2/1202	2/1125	2/2	2/1120
Text	159/161	159/164	11/392	20/34	11/1689	12/67	12/21	12/135
Databases	31/38	32/43	13/60	21/50	23/54	12/63	13/24	23/55
Config.	2797/2969	2831/2959	1349/4798	1976/2969	1352/6978	1345/2237	1348/1382	1349/10930
Apps.	6/6	6/10	164/489	227/304	164/458	164/200	164/310	164/670

Data Counts 2a (Post/Pre-reset)

Device	p9	p10	p11	p12	p13	p14	p15	p16
Type	A	A	A	A	A	A	A	A
App. Usage	0/0	1/139	0/0	0/102	0/0	0/0	0/0	0/0
Call Log	0/0	0/0	0/0	0/52	0/5	0/6	0/0	7/192
Contacts	0/0	0/5	0/0	0/48	0/0	0/2	0/0	0/65
Cookies	0/0	0/3	0/0	0/0	0/0	0/5	0/0	0/0
Installed Apps.	30/30	48/102	25/43	23/32	26/70	20/24	12/44	0/0
IP Conns.	0/0	0/2	0/0	0/3	0/0	0/1	0/0	0/0
Locations	0/0	0/0	0/5	0/0	0/10	0/0	0/0	0/0
Maps	0/15	0/5	0/8	0/0	0/0	0/0	0/0	0/0
Passwords	0/0	0/0	0/0	0/1	0/0	0/1	0/1	0/0

Data Counts 2b (Post/Pre-reset)

Device	p9	p10	p11	p12	p13	p14	p15	p16
User Accounts	1/1	1/1	0/0	0/0	0/1	0/1	0/0	0/1
User Dict.	0/132	0/50	0/0	0/84	0/40	0/1	0/0	0/0
Wireless	0/1	0/1	0/0	0/1	0/0	0/0	0/1	0/0
Images	150/150	764/764	11/11	1815/1815	42/42	15/15	9/9	616/616
Audio	1/1	1/1	1/1	1/1	0/0	4/4	1/1	243/263
Text	130/130	48/48	0/0	132/132	1/1	0/0	4/4	1/1
Databases	5/65	12/45	0/0	25/41	0/0	10/24	0/0	16/36
Config.	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
Apps.	0/0	313/313	0/0	7/7	3/3	1/1	24/24	0/3
SMS Msg.	0/80	0/5	0/0	0/121	0/0	0/2	0/0	0/66

Data Counts 3a (Post/Pre-reset)

Device	p18	p20	p21	p25
Type	A	I	B	B
App. Usage	0/0	0/0	0/0	0/0
Call Log	0/1	0/100	0/11	0/61
Contacts	0/0	0/477	0/0	5/252
Cookies	0/17	0/0	0/0	0/16
Installed Apps.	34/34	0/0	0/0	0/1
IP Connections	1/6	0/0	0/0	0/0
Locations	0/0	0/0	0/0	0/9
Maps	0/0	0/0	0/0	0/0
Passwords	5/9	0/0	0/0	0/1

Data Counts 3b (Post/Pre-reset)

Device	p18	P20	P21	p25
User Accounts	1/1	0/0	0/0	0/2
User Dictionary	0/6	0/0	0/0	0/2
Wireless Networks	3/7	0/0	0/0	0/1
Images	3716/3743	0/ 7	0/3	71/159
Audio	1/1	0/1	0/0	1/1
Text	243/263	0/0	0/0	440/3031
Databases	37/58	0/0	0/0	24/55
Configuration	2850/2953	0/0	0/0	0/0
Applications	6/6	0/0	0/0	388/390
SMS Messages	0/0	0/4	0/0	0/76

Bulk Extractor applied to post-reset images

- Many email addresses and phone numbers found within files, but mostly for vendors.
- IP addresses found on several devices.
- Root certificates found on many devices.
- A few SHA1 and MD5 hashes found on some devices.
- Password found on one device.
- Geolocation data found on one device.
- Bulk Extractor claimed to find 157 additional files that Cellebrite did not find – but some of these may be spurious due to recent tests.

Conclusions on device resets

- The “factory reset” did not erase all user data on any device we tested, contradicting manufacturer claims.
- Android devices did not delete most emplaced user files.
- Both Android and iPhones did not delete a variety of indirect data about users, and it varied with device.
- On average 42% of the files on a device were deleted by the reset, mostly third-party software.
- Modified files were also found after the reset, and even some new files, indicating routine system operations.
- Year of the operating-system release did not affect results much, so things are not getting better.
- This suggests that reset devices may still have forensically valuable information.

Recommended procedure for erasing devices

1. Perform a software or “factory” reset.
2. Manually delete any remaining user files that have been moved to unconventional locations (like a compressed file in the root directory).
3. Manually delete remaining user data from software directories.
4. Delete remaining cache files, browser history files, cookies, and settings files.
5. Delete zero-size files.
6. Overwrite deleted data with zeros.
7. Remove the SIM card and any other removable storage.

Commercial software such as data erasing tools will be generally necessary for steps 2, 3, and 6.