Cross-drive analysis (CDA) is a forensic technique that correlates information found on multiple digital devices (hard drives, camera cards, cell phones, etc.). Unlike existing approaches for analyzing multiple devices, CDA takes into account the number of devices on which identifiers are found, so that an email address that appears on three or four drives is more important than an email address that appears on just one or two. CDA can also be applied to a large corpus of drives, so that email addresses can automatically be pruned out if they appear in the “background.”
NPS is the Navy’s Research University.

Location: Monterey, CA

Students: 1500
- US Military (All 5 services)
- US Civilian (Scholarship for Service & SMART)
- Foreign Military (30 countries)

Schools:
- Business & Public Policy
- Engineering & Applied Sciences
- Operational & Information Sciences
- International Graduate Studies

NCR Initiative:
- 8 offices on 5th floor, 900N Glebe Road, Arlington
- Current staffing: 4 professors, 2 lab managers, 2 programmers, 4 contractors
- WE ARE HIRING!
- OPEN SLOTS FOR .GOV PHDs!
Outline of the next 35 minutes...

Introducing bulk_extractor
- Overview and history
- Announcing bulk_extractor 1.3!

Cross drive analysis
- What is it?
- Why do it?

Context-sensitive stop lists
- Simple CDA

Using cda_tool
- making stop-lists
Introducing bulk_extractor
Between 2005 and 2008, I interviewed law enforcement regarding their use of forensic tools.

Law enforcement officers wanted a *highly automated* tool for finding:

- Email addresses & Credit card numbers (including track 2 information)
- Phone numbers, GPS coordinates & EXIF information from JPEGs
- Search terms (extracted from URLs)
- All words that were present on the disk (for password cracking)

The tool had to:

- Run on Windows, Linux, and Mac-based systems
- Run with *no* user interaction
- Operate on raw disk images, split-raw volumes, E01 files, and AFF files
- Run at maximum I/O speed of physical drive
- Never crash

Moving technology from the lab to the field has been challenging:

- Must work with evidence files of *any size* and on *limited hardware*.
- Users can't provide their data when the program crashes.
- Users are *analysts* and *examiners*, not engineers.
The tool implements Stream-Based Disk Forensics: Scan the disk from beginning to end; do your best.

1. Read all of the blocks in order.
2. Look for information that might be useful.
3. Identify & extract what's possible in a single pass.

3 hours, 20 min to *read* the data
Primary Advantage: Speed

No disk seeking! (Good for HDs, SSDs, & E01 files)
Easy to parallelize ("embarrassingly parallel")
Reads all the data — allocated files, deleted files, file fragments

Caveats:
- Compressed data must be decompressed
  — *Fragmented, compressed files may not be recovered*
- Can only read at maximum I/O transfer rates if data can be *processed*
  — *Even 24+ cores may not be enough*
- Does not provide file names
  — *File names can be determined with a separate metadata extraction step.*
Fragmented files may not be recovered

ZIP, GZIP & LZMA use *adaptive* compression algorithms.
- Part 1 required to decompress part 2.
- Also an issue for JPEG.

Fortunately, most files are *not* fragmented.
- Individual components of a ZIP can be recovered (e.g. `word/document.xml`)

Most files that *are* fragmented have carvable internal structure:
- Log files, Outlook PST files, etc.
bulk_extractor has three phases of operation: Feature Extraction; Histogram Creation; Post Processing

Output is a directory containing:
- feature files; histograms; carved objects
- Mostly in UTF-8; some XML
- Can be bundled into a ZIP file and process with bulk_extractor_reader.py
Feature files are UTF-8 files that contain extracted data.

# UTF-8 Byte Order Marker; see http://unicode.org/faq/utf_bom.html
# bulk_extractor- Version: 1.3b1-dev2
# Filename: /corp/nps/drives/nps-2009-m57-patents/charlie-2009-12-11.E01
# Feature- Recorder: telephone
# Feature- File- Version: 1.1
...
6489225486   (316) 788-7300  Corrine Porter (316) 788-7300,,,,,,,,Phase I En
6489230027   620-723-2638  ,,,,,,,,,Dan Hayse - 620-723-2638,,,,,,,,Phase I En
6489230346   620-376-4499  Bertha Mangold -620-376-4499,,,,,,,,Phase I En
...
3772517888-GZIP-28322  (831) 373-5555  onterey-(831) 373-5555</nobr>
3772517888-GZIP-29518  (831) 899-8300  Seaside - <nobr>(831) 899-8300</nobr>
5054604751    716-871-2929  a%,888-571-2048,716-871-2929\x0D\x0ACPVC,%,%Cape

Designed for easy processing by python, perl or C++ program

- “Loosely ordered.”
- -GZIP- indicates that data was decompressed
- Non-UTF-8 characters are escaped
Histogram system automatically summarizes features.

```
# UTF-8 Byte Order Marker; see http://unicode.org/faq/utf_bom.html
# bulk_extractor-Version: 1.3b1-dev2
# Filename: /corp/nps/drives/nps-2009-m57-patents/charlie-2009-12-11.E01
# Feature-Recorder: email
# Histogram-File-Version: 1.1

... n=875 mozilla@kewis.ch (utf16=3)
n=651 charlie@m57.biz (utf16=120)
n=605 ajbanck@planet.nl
...
n=288 mattwillis@gmail.com
n=281 garths@oeone.com
n=226 michael.buettner@sun.com (utf16=2)
n=225 bugzilla@babylonsounds.com
n=218 berend.cornelius@sun.com
n=210 ips@mail.ips.es
n=201 mschroeder@mozilla.x-home.org
n=186 pat@m57.biz (utf16=1)
```
New in bulk_extractor 1.3

New supported data types:
- Windows PE Scanner
- Linux ELF Scanner
- VCARD Scanner
- BASE16 scanner
- Windows directory carver

New Histogram options:
- Numeric only option for phone numbers
- Supports new Facebook ID

Better Unicode Support:
- Histograms now UTF-8 / UTF-16 aware
- Feature files are UTF-8 clean

More Reliability:
- Most scanners now use sbuf_t for brokered access to subject data.
- bulk_extractor now tolerant of libewf read errors.
- plug-in system now catches C++ exceptions.

Limited support for file carving:
- packets carved into pcap files
  — (IPv4 and IPv6)
- VCARD carver
Introducing Cross-Drive Analysis
Frequently we get data from evidence and don’t know how to interpret it.

- 5,000 total credit card numbers found on media
- 1,000 distinct credit card numbers found on media
First order cross drive analysis:
Use a collection of data to find outliers.
First order cross drive analysis:
Use a collection of data to find outliers.

Focus on these drives
Second Order Cross Drive Analysis:
Look for correlations between subject drives.

- Unique CCNs
- Total CCNs
Second Order Cross Drive Analysis:
Look for correlations between subject drives.

25 credit card numbers

- Unique CCNs
- Total CCNs
Second Order Cross Drive Analysis:
Look for correlations between subject drives.

- Unique CCNs
- Total CCNs

25 credit card numbers
Same 25 credit card numbers
Second Order Cross Drive Analysis: Look for correlations between subject drives.

Cross-Drive Analysis uses data from multiple drives to provide analysts with a better understanding of their targets.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Manual analysis misses opportunities for correlation.

Different analysts see different hard drives.

Keyword searches don’t connect the dots.
Suppressing bulk_extractor False Positives
For this section we will work with ubnist1

nps-2009-ubnist1 — Bootable Linux USB used to browse USG sites

- `rw-rw-r--  1 simsong  staff  685 Jan  9  2009 narrative.txt`
- `rw-rw-r--  1 simsong  staff 2106589184 Jan  6  2009 ubnist1.gen0.raw`
- `rw-rw-r--  1 simsong  staff 2106589184 Jan  6  2009 ubnist1.gen1.raw`
- `rw-rw-r--  1 simsong  staff 2106589184 Jan  9  2009 ubnist1.gen2.raw`
- `rw-rw-r--  1 simsong  staff 2106589184 Jan  7  2009 ubnist1.gen3.raw`

Four snapshots: gen0, gen1, gen2 & gen3
Hard drives are *filled* with email addresses.

Bulk_extractor finds email addresses in many places:
- Windows binaries; SSL certificates
- Documents
- Cached web pages; Memory; Hibernation Files

UBNIST1 have a LOT of email addresses; each snapshot sees more...

<table>
<thead>
<tr>
<th>gen0</th>
<th>gen1</th>
<th>gen2</th>
<th>gen3</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=3447 <a href="mailto:olly@survex.com">olly@survex.com</a></td>
<td>n=3455 <a href="mailto:olly@survex.com">olly@survex.com</a></td>
<td>n=27364 <a href="mailto:ubuntu-users@lists.ubuntu.com">ubuntu-users@lists.ubuntu.com</a></td>
<td></td>
</tr>
<tr>
<td>n=2237 <a href="mailto:hadess@hadess.net">hadess@hadess.net</a></td>
<td>n=3254 <a href="mailto:ubuntu-devel-discuss@lists.ubuntu.com">ubuntu-devel-discuss@lists.ubuntu.com</a></td>
<td>n=17213 <a href="mailto:ubuntu-motu@lists.ubuntu.com">ubuntu-motu@lists.ubuntu.com</a></td>
<td></td>
</tr>
<tr>
<td>n=2040 <a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>n=2241 <a href="mailto:hadess@hadess.net">hadess@hadess.net</a></td>
<td>n=14291 <a href="mailto:ubuntu-devel-discuss@lists.ubuntu.com">ubuntu-devel-discuss@lists.ubuntu.com</a></td>
<td></td>
</tr>
<tr>
<td>n=990 <a href="mailto:airlied@linux.ie">airlied@linux.ie</a></td>
<td>n=2040 <a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>n=4086 <a href="mailto:language-packs@ubuntu.com">language-packs@ubuntu.com</a></td>
<td></td>
</tr>
<tr>
<td>n=910 <a href="mailto:cjwatson@debian.org">cjwatson@debian.org</a></td>
<td>n=990 <a href="mailto:airlied@linux.ie">airlied@linux.ie</a></td>
<td>n=3481 <a href="mailto:olly@survex.com">olly@survex.com</a></td>
<td></td>
</tr>
<tr>
<td>n=909 <a href="mailto:dsandras@seconix.com">dsandras@seconix.com</a></td>
<td>n=930 <a href="mailto:cjwatson@debian.org">cjwatson@debian.org</a></td>
<td>n=2280 <a href="mailto:ubuntu-desktop@lists.ubuntu.com">ubuntu-desktop@lists.ubuntu.com</a></td>
<td></td>
</tr>
<tr>
<td>n=893 <a href="mailto:rene@debian.org">rene@debian.org</a></td>
<td>n=910 <a href="mailto:rene@debian.org">rene@debian.org</a></td>
<td>n=2239 <a href="mailto:hadess@hadess.net">hadess@hadess.net</a></td>
<td></td>
</tr>
<tr>
<td>n=884 <a href="mailto:anholt@freebsd.org">anholt@freebsd.org</a></td>
<td>n=909 <a href="mailto:dsandras@seconix.com">dsandras@seconix.com</a></td>
<td>n=2040 <a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td></td>
</tr>
<tr>
<td>n=767 <a href="mailto:jirka@5z.com">jirka@5z.com</a></td>
<td>n=884 <a href="mailto:anholt@freebsd.org">anholt@freebsd.org</a></td>
<td>n=1696 <a href="mailto:debian-x@lists.debian.org">debian-x@lists.debian.org</a></td>
<td></td>
</tr>
<tr>
<td>n=760 <a href="mailto:guillem@debian.org">guillem@debian.org</a></td>
<td>n=767 <a href="mailto:jirka@5z.com">jirka@5z.com</a></td>
<td>n=1202 <a href="mailto:strk@keybit.net">strk@keybit.net</a></td>
<td></td>
</tr>
<tr>
<td>n=744 <a href="mailto:wakkerma@debian.org">wakkerma@debian.org</a></td>
<td>n=760 <a href="mailto:guillem@debian.org">guillem@debian.org</a></td>
<td>n=1122 <a href="mailto:bw@benjaminwolsey.de">bw@benjaminwolsey.de</a></td>
<td></td>
</tr>
<tr>
<td>n=708 <a href="mailto:dshaw@jabberwocky.com">dshaw@jabberwocky.com</a></td>
<td>n=744 <a href="mailto:wakkerma@debian.org">wakkerma@debian.org</a></td>
<td>n=1044 <a href="mailto:pkg-perl-maintainers@lists.alioth.debian.org">pkg-perl-maintainers@lists.alioth.debian.org</a></td>
<td></td>
</tr>
<tr>
<td>n=660 <a href="mailto:doogie@debian.org">doogie@debian.org</a></td>
<td>n=708 <a href="mailto:dshaw@jabberwocky.com">dshaw@jabberwocky.com</a></td>
<td>n=1036 <a href="mailto:cjwatson@debian.org">cjwatson@debian.org</a></td>
<td></td>
</tr>
<tr>
<td>n=659 <a href="mailto:brian.cameron@sun.com">brian.cameron@sun.com</a></td>
<td>n=660 <a href="mailto:doogie@debian.org">doogie@debian.org</a></td>
<td>n=1017 <a href="mailto:rene@debian.org">rene@debian.org</a></td>
<td></td>
</tr>
<tr>
<td>n=656 <a href="mailto:dsandras@gnome.org">dsandras@gnome.org</a></td>
<td>n=659 <a href="mailto:brian.cameron@sun.com">brian.cameron@sun.com</a></td>
<td>n=990 <a href="mailto:airlied@linux.ie">airlied@linux.ie</a></td>
<td></td>
</tr>
<tr>
<td>n=654 <a href="mailto:kyoshida@novell.com">kyoshida@novell.com</a></td>
<td>n=656 <a href="mailto:dsandras@gnome.org">dsandras@gnome.org</a></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>n=632 <a href="mailto:priikone@silcnet.org">priikone@silcnet.org</a></td>
<td>n=654 <a href="mailto:kyoshida@novell.com">kyoshida@novell.com</a></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>n=626 <a href="mailto:daniel@fooishbar.org">daniel@fooishbar.org</a></td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

6596 total 6929 total 8734 total 8734 total
It’s important to distinguish email addresses that are relevant to a case from those that are not.

The top address is olly@survex.com

- We should probably ignore Mr. Betts:

Other sources that we might wish to ignore

- Windows binaries; SSL certificates; Sample documents; News Stories
Stop lists specify features to be “stopped”

Stopped features are not ignored!
- Stopped features are moved from email.txt to email_stopped.txt.
- This is important for validation and error-diagnosis.

Stop lists are implemented with the word_and_context_list class.
- “words” — Anything that might be in the “feature” column
  — Email address
  — MD5 hash of the first 4KiB of a file (JPEG)
  — AES key
- “context” — Anything that might be in the “context” column
  — Includes the feature
  — Will suppress a specific instance of a feature
- regular expressions
  — Dramatically slows down the process

bulk_extractor also supports alert lists
- “words” or “context” that should be flagged
stop lists and alert lists are specified with the “-w” and “-r” options.

Usage: bulk_extractor [options] imagefile

... 

-r alert_list.txt  - a file containing the alert list of features to alert  
(can be a feature file or a list of globs)  
(can be repeated.)

-w stop_list.txt   - a file containing the stop list of features (white list  
(can be a feature file or a list of globs)s  
(can be repeated.)

The list can be a list of words or a feature file

- For example

—*stop_list.txt:*
  olly@survex.com
  hadess@hadess.net
  daniel@veillard.com

—*alert_list.txt:*
  daniel@fooishbar.org

- command to use is:

  bulk_extractor -r alert_list.txt -w stop_list.txt -o ubnist1.gen0-v1 ubnist1.gen0.raw
Stop lists processing is reflected in the feature files.

No stop list:

7324005 Aug 4 10:25 ubnist1.gen0/email.txt
    169609 Aug 4 10:25 ubnist1.gen0/email_histogram.txt

$ wc -l ubnist1.gen0/email*
    72965 ubnist1.gen0/email.txt
    6596 ubnist1.gen0/email_histogram.txt

With stop list:

6566160 Aug 4 11:20 email.txt
    169534 Aug 4 11:20 email_histogram.txt
    827940 Aug 4 11:20 email_stopped.txt

$ wc -l ubnist1.gen0-v1/email*
    65241 ubnist1.gen0-v1/email.txt
    6593 ubnist1.gen0-v1/email_histogram.txt
    7729 ubnist1.gen0-v1/email_stopped.txt

$ head ubnist1.gen0-v1/email_stopped.txt
# UTF-8 Byte Order Marker; see http://unicode.org/faq/utf_bom.html
# Filename: /corp/nps/drives/nps-2009-ubnist1/ubnist1.gen0.raw
# Feature-Recorder: email_stopped
# Feature-File-Version: 1.1
317986809-GZIP-47 daniel@veillard.com aniel Veillard <daniel@veillard.com>\x5Cx0A\x5Cx0A\x5Cx0A\x5Cx09* configure.
317986809-GZIP-209 daniel@veillard.com aniel Veillard <daniel@veillard.com>\x5Cx0A\x5Cx0A\x5Cx0A\x5Cx09* libxslt/xs
317986809-GZIP-635 daniel@veillard.com aniel Veillard <daniel@veillard.com>\x5Cx0A

— stop_list.txt:  
olly@survex.com
hadess@hadess.net
daniel@veillard.com

— alert_list.txt:  
daniel@fooishbar.org
Here is more of the email stopped list:

<table>
<thead>
<tr>
<th>Email Address</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-47</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-209</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-635</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-1211</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-1581</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-1692</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-1824</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-1943</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-2085</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-2315</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-2724</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-2940</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-3331</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-3494</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td><a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>317986809-GZIP-3647</td>
<td>Daniel Veillard</td>
</tr>
<tr>
<td>111x627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>317986809-GZIP-275885</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-276902</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-277778</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-277951</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-278061</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-278198</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-278379</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-278529</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
<tr>
<td>317986809-GZIP-278655</td>
<td>330031689</td>
<td>Olly Betts</td>
</tr>
</tbody>
</table>
A feature file can be used as a context stop list. We can use the gen0 email as a stop for gen1

The gen0 feature file becomes a filter:

This isn’t hard to do in practice:

```
$ src/bulk_extractor -w ubnist1.gen0/email.txt -o ubnist1.gen1-filtered_by_gen0 ubnist1.gen1.raw
Reading context stop list ubnist1.gen0/email.txt
Stop list read.
  Total features read: 72961
  List Size: 32084
  Context Strings: 32083
  Regular Expressions: 0
```
The result of filtering gen1 with gen0: We only see the new *uses* of email addresses

8424165 Aug 4 10:24 ubnist1.gen1/email.txt
179549 Aug 4 10:24 ubnist1.gen1/email_histogram.txt

7324005 Aug 4 10:25 ubnist1.gen0/email.txt
169609 Aug 4 10:25 ubnist1.gen0/email_histogram.txt

1083873 Aug 4 11:49 ubnist1.gen1-filtered_by_gen0/email.txt
13867 Aug 4 11:50 ubnist1.gen1-filtered_by_gen0/email_histogram.txt
7805523 Aug 4 11:49 ubnist1.gen1-filtered_by_gen0/email_stopped.txt

The resulting histogram is a histogram of the filtered email.txt:

- n=3178 ubuntu-devel-discuss@lists.ubuntu.com
- n=638 ubuntu-desktop@lists.ubuntu.com
- n=444 debian-x@lists.debian.org
- n=219 debian-boot@lists.debian.org
- n=204 ubuntu-motu@lists.ubuntu.com
- n=144 seb128@debian.org
- n=132 pkg-gnome-maintainers@lists.alioth.debian.org
Stop lists and alert lists have minor impact on performance.

Longer stop lists are slower to process. $t \propto (\text{features} \times \text{stops})$

<table>
<thead>
<tr>
<th>lines in stop list</th>
<th>ubnist1.gen0 Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>54.51 s</td>
</tr>
<tr>
<td>3</td>
<td>55.47 s</td>
</tr>
<tr>
<td>235,886</td>
<td>55.41 s</td>
</tr>
</tbody>
</table>

Regular expressions slow down stop lists dramatically:

<table>
<thead>
<tr>
<th>re lines in stop list</th>
<th>ubnist1.gen0 Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>54.51 s</td>
</tr>
<tr>
<td>3</td>
<td>66.41 s</td>
</tr>
</tbody>
</table>

Stop and alert lists must be applied when bulk_extractor is run.

- A future version may allow filtering after-the-fact.
Context-sensitive stop lists are important when looking for unknown individuals.

Recall all of those email addresses on ubnist1

Although these emails are widely seen, they should not be whitelisted:
- Email addresses can be shared
- Email addresses can be sold
- A Linux developer might be engaged in a criminal enterprise

By using context-sensitive stop lists, we:
- Ignore the email address where it is widely seen
- Will still notice the email address in a new context

Context-sensitive lists need to be maintained!
- Build from default installs of operating systems & applications.
- NIST is running bulk_extractor over the entire NSRL and will make the results available.
- Organizations are free to trade the feature files amongst themselves.
$ python cda_tool.py --help
usage: cda_tool.py [-h] [--netmap] [--idcor] [--makestop MAKESTOP] 
    [--threshold THRESHOLD] [--idfeatures IDFEATURES] 
    reports [reports ...]

Cross Drive Analysis with bulk_extractor output

positional arguments:
  reports               bulk_extractor report directories or ZIP files

optional arguments:
  -h, --help            show this help message and exit
  --netmap              General GraphViz data for network correlation map
  --idcor               Perform identity-based correlation
  --makestop MAKESTOP   Make a stop list of identity features on more than
    THRESHOLD drives
  --threshold THRESHOLD Specify the faction of drives for the threshold
  --idfeatures IDFEATURES
    Specifies feature files used for identity operations

$
cda_tool.py: a platform for cross-drive analysis

cda_tool.py is a python program that:

- Reads 2 or more bulk_extractor reports
- Performs multi-drive correlation
- Outputs the results in a format that can be used for additional processing

Cross-Drive correlation depends on two Python classes:

- bulk_extractor_reader.BulkReport
  - *Reads the directory or a ZIP file of the directory.*
- cda_tool.Correlator
  - *Performs the actual correlation*
Correlation is performed with feature histograms. (This means that local context is ignored.)

Start with drives, features and counts:

<table>
<thead>
<tr>
<th>gen0</th>
<th>gen1</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=3447  <a href="mailto:oly@survex.com">oly@survex.com</a></td>
<td>n=3455  <a href="mailto:oly@survex.com">oly@survex.com</a></td>
</tr>
<tr>
<td>n=2237  <a href="mailto:hadess@hadess.net">hadess@hadess.net</a></td>
<td>n=3254  <a href="mailto:ubuntu-devel-discuss@lists.ubuntu.com">ubuntu-devel-discuss@lists.ubuntu.com</a></td>
</tr>
<tr>
<td>n=2040  <a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
<td>n=2241  <a href="mailto:hadess@hadess.net">hadess@hadess.net</a></td>
</tr>
<tr>
<td>n=990   <a href="mailto:airlied@linux.ie">airlied@linux.ie</a></td>
<td>n=2040  <a href="mailto:daniel@veillard.com">daniel@veillard.com</a></td>
</tr>
<tr>
<td>n=910   <a href="mailto:cjw@adess.net">cjw@adess.net</a></td>
<td>n=990   <a href="mailto:airlied@linux.ie">airlied@linux.ie</a></td>
</tr>
<tr>
<td>n=909   <a href="mailto:dsandras@seconix.com">dsandras@seconix.com</a></td>
<td>n=930   <a href="mailto:cjw@adess.net">cjw@adess.net</a></td>
</tr>
<tr>
<td>n=893   <a href="mailto:rene@adess.net">rene@adess.net</a></td>
<td>n=910   <a href="mailto:rene@debian.org">rene@debian.org</a></td>
</tr>
<tr>
<td>n=884   <a href="mailto:anholt@freebsd.org">anholt@freebsd.org</a></td>
<td>n=909   <a href="mailto:dsandras@seconix.com">dsandras@seconix.com</a></td>
</tr>
<tr>
<td>n=767   <a href="mailto:jirka@5z.com">jirka@5z.com</a></td>
<td>n=884   <a href="mailto:anholt@freebsd.org">anholt@freebsd.org</a></td>
</tr>
<tr>
<td>n=760   <a href="mailto:guillem@freebsd.org">guillem@freebsd.org</a></td>
<td>n=767   <a href="mailto:jirka@5z.com">jirka@5z.com</a></td>
</tr>
<tr>
<td>n=744   <a href="mailto:wakkerma@debian.org">wakkerma@debian.org</a></td>
<td>n=760   <a href="mailto:guillem@debian.org">guillem@debian.org</a></td>
</tr>
<tr>
<td>n=708   <a href="mailto:dshaw@jabberwocky.com">dshaw@jabberwocky.com</a></td>
<td>n=744   <a href="mailto:wakkerma@debian.org">wakkerma@debian.org</a></td>
</tr>
<tr>
<td>n=660   <a href="mailto:doogie@debian.org">doogie@debian.org</a></td>
<td>n=708   <a href="mailto:dshaw@jabberwocky.com">dshaw@jabberwocky.com</a></td>
</tr>
<tr>
<td>n=659   <a href="mailto:brian.cameron@sun.com">brian.cameron@sun.com</a></td>
<td>n=660   <a href="mailto:doogie@debian.org">doogie@debian.org</a></td>
</tr>
<tr>
<td>n=656   <a href="mailto:dsandras@gnome.org">dsandras@gnome.org</a></td>
<td>n=659   <a href="mailto:brian.cameron@sun.com">brian.cameron@sun.com</a></td>
</tr>
<tr>
<td>n=654   <a href="mailto:kyoshida@novell.com">kyoshida@novell.com</a></td>
<td>n=656   <a href="mailto:dsandras@gnome.org">dsandras@gnome.org</a></td>
</tr>
<tr>
<td>n=632   <a href="mailto:priikone@silcnet.org">priikone@silcnet.org</a></td>
<td>n=654   <a href="mailto:kyoshida@novell.com">kyoshida@novell.com</a></td>
</tr>
<tr>
<td>n=626   <a href="mailto:daniel@fooishbar.org">daniel@fooishbar.org</a></td>
<td>...</td>
</tr>
</tbody>
</table>

Correlator produces per-drive counts:

```
{  
  “olly@survex.com”:  
  {“gen0”:3447,  
   “gen1”:3455},  
  “hadess@hadess.net”:  
  {“gen0”:2237,  
   “gen1”:2241},  
  “airlied@linux.ie”:  
  {“gen0”:990,  
   “gen1”:990},  
  “ubuntu-devel-discuss@lists.ubuntu.com”,  
  {“gen1”:3254}  
...  
```
Use cda_tool is to create a stoplist of features present on more than f drives:

```
$ python cda_tool.py --help
usage: cda_tool.py [-h] [--netmap] [--idcor] [--makestop MAKESTOP]
                [--threshold THRESHOLD] [--idfeatures IDFEATURES]
                reports [reports ...]

... positional arguments:
    reports               bulk_extractor report directories or ZIP files

... optional arguments:
    -h, --help            show this help message and exit
    --netmap              General GraphViz data for network correlation map
    --idcor               Perform identity-based correlation
    --makestop MAKESTOP   Make a stop list of identity features on more than
                           THRESHOLD drives
    --threshold THRESHOLD Specify the faction of drives for the threshold
    --idfeatures IDFEATURES Specifies feature files used for identity operations

$ Example:
$ ./cda_tool.py --makestop stoplist.txt --threshold 3 --idfeatures email *.zip
```
On my laptop I had a sample of reports from 2009-12-11:

```
$ ls -l current/tutorial/
  total 289188
  -rw-r--r--  1 22484045 Jul 10 11:16 charlie-2009-12-11.zip
  -rw-r--r--  1    52194 Jul 10 11:16 charlie-work-usb-2009-12-11.zip
  -rw-r--r--  1 18719633 Jul 10 11:19 jo-2009-12-11-002.zip
  -rw-r--r--  1  2702376 Jul 10 11:19 jo-work-usb-2009-12-11.zip
  -rw-r--r--  1  81798305 Jul 10 11:40 terry-2009-12-10.zip
  -rw-r--r--  1  81734461 Jul 10 11:41 terry-2009-12-11-001.zip
  -rw-r--r--  1  82532973 Jul 10 11:42 terry-2009-12-11-002.zip
  -rw-r--r--  1  6089981 Jul 10 11:42 terry-work-usb-2009-12-11.zip
```
We can use cda_tool.py to create a stoplist of identity features on more than 50% of the drives:

```
$ python cda_tool.py --makestop stoplist.txt --threshold 0.5 *.zip
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/terry-work-usb-2009-12-11.E01 for email
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/terry-2009-12-11-002.E01 for email
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/terry-2009-12-11-002.E01 for telephone
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/jo-2009-12-11-002.E01 for ccn
...

Stoplist stoplist.txt created with 763 features

<table>
<thead>
<tr>
<th>DPF</th>
<th>Feature Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>733</td>
</tr>
<tr>
<td>2</td>
<td>345</td>
</tr>
<tr>
<td>3</td>
<td>752</td>
</tr>
<tr>
<td>4</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>702</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

DPF = Drives per Feature
Only features on 4 or more drives were written.
$
The result is a list of features written to stoplist.txt:

```
$ wc stoplist.txt
    763   944  18994 stoplist.txt
$ head -20 stoplist.txt
(000)-000-0000
(111) 555-4444
(222) 555-1111
(222) 555-2222
(506) 298-2000
(777) 777-7777
+254-20-3742774
+254-20-3744660
+254-20-3750230
+31 20 346 9363
+34 91 377 8170
+351 707 5000
+39 02 4586
+420 555 123321
+420 555 654321
+49 69 5007
+81-422-42-7633
0.0014F8A2@mail.groupcare.dk
001-800-2468
098 345.6044
```

This stoplist can be read with bulk_extractor’s -w option.
Use cda_tool.py to find drive affinity...
Use cda_tool.py to find drive affinity...

Cross Drive Correlation

- Drives #74 x #77
  - 25 CCNS in common

- Drives #171 & #172
  - 13 CCNS in common

- Drives #179 & #206
  - 13 CCNS in common

- Same Community College
  - Drives #171 & #172
  - 13 CCNS in common

- Same Medical Center
  - Drives #74 x #77
  - 25 CCNS in common

- Same Car Dealership
  - Drives #179 & #206
  - 13 CCNS in common
We use the formula from Garfinkel 2006

Let:

\[ D = \text{# of drives} \]
\[ F = \text{# of extracted features} \]
\[ d_0 \ldots d_D = \text{Drives in corpus} \]
\[ f_0 \ldots f_F = \text{Extracted features} \]
\[ FP(f_n, d_n) = \begin{cases} 
0 & f_n \text{ not present on } d_n \\
1 & f_n \text{ present on } d_n 
\end{cases} \]

\[ DC(f) = \sum_{n=0}^{D} FP(f, d_n) = \text{# of drives with feature } f \]

\[ S_2(d_1, d_2) = \sum_{n=0}^{F} \frac{FP(f_n, d_1) \times FP(f_n, d_2)}{DC(f_n)} \]
Use the `cda_tool.py --idcor`

```python
cda_tool.py --idc ~/current/tutorial/*.zip
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/terry-work-usb-2009-12-11.E01 for email
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/terry-work-usb-2009-12-11.E01 for telephone
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/charlie-2009-12-11.E01 for ccn
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/charlie-2009-12-11.E01 for email
Scanning /corp/nps/drives/nps-2009-m57-patents-redacted/charlie-2009-12-11.E01 for telephone
...
cda_tool.py --idcor shows the drive affinity between each drive, and the features that contributed

Identity-based correlation: computes drive affinity using TF-IDF
Drive A:  /corp/nps/drives/nps-2009-m57-patents-redacted/terry-2009-12-11-001.E01
Drive B:  /corp/nps/drives/nps-2009-m57-patents-redacted/terry-2009-12-11-002.E01
Score  : 405.73333333332454
Top factors:
    0.3333  4890008118593969
    0.3333  5467663276676632
    0.3333  5540655860834388
    0.2   6543210123456788
    0.3333  5396180804577535

Drive A:  /corp/nps/drives/nps-2009-m57-patents-redacted/terry-2009-12-11-001.E01
Drive B:  /corp/nps/scenarios/2009-m57-patents/drives-redacted/terry-2009-12-10.E01
Score  : 400.73333333332465
Top factors:
    0.3333  4890008118593969
    0.3333  5467663276676632
    0.3333  5540655860834388
    0.2   6543210123456788
    0.3333  5396180804577535

... Drive names come from BE report

No 3D graph yet (sorry!)
In conclusion:
bulk_extractor 1.3: new features & working CDA!

New scanners designed for Windows and malware analysis:
- Windows PE Scanner
- Linux ELF Scanner
- VCARD Scanner
- BASE16 scanner
- Windows directory carver

Cross-Drive analysis in cda_tool.py

Download from https://github.com/simsong/bulk_extractor
- bulk_extractor1.3 in Downloads
- python and plugins moving to git repository!

Contact info:
- simsong@acm.org

Questions?