Finding privacy leaks and stolen data with bulk data analysis and optimistic decoding

Garfinkel, Simson L.
Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/44330

Downloaded from NPS Archive: Calhoun
Finding privacy leaks and stolen data with bulk data analysis and optimistic decoding

Drexel University
Wednesday, Oct. 15, 2014

Simson L. Garfinkel
Naval Postgraduate School
http://simson.net/

The opinions expressed herein are those of the author(s), and are not necessarily representative of those of the Naval Postgraduate School, the Department of Defense (DOD); or, the United States Army, Navy, or Air Force.
NPS is the Navy’s Research University.

Monterey, CA — 1500 students
- US Military
- Civilian (Scholarship for Service & SMART)
- Foreign Military (30 countries)

National Capital Region (NCR) Office
- 900 N Glebe (Ballston)/Virginia Tech building
  ARLINGTON, VA
Digital information is pervasive in today’s society.

Many potential sources of digital information:

- Desktops; Laptops
- Tablets; Cell Phones
- Internet-Based Services
- GPS devices
- Cars

“Digital forensics” — the recovery, analysis & use of this information

- Law enforcement — Document a conspiracy (stock fraud; murder-for-hire; Silk Road)
- DOD — Identify members of a terrorist organization.
- Cyber investigations — Find and understand malware
- Ordinary people — Recover deleted files.
We need digital forensics because digital devices are exceedingly complex.

Typical computer might have:
- Millions of files; dozens of applications
- Data from many different people
- Information in many different formats

Digital forensic tools allow for:
- Viewing hidden information / recovery of deleted files
- Rapidly analyzing large amounts of subject data
- Determine when something happened in the computer’s past
- Establish intent:
  — Recovered search terms
  — User-generate content
- Recover specific information critical to a case:
  — Identity information (contacts, etc.)
  — Contraband information (stolen documents, criminal content)

These tools can also be used for:
- Privacy auditing; software testing
We have created multiple “corpora” to support digital forensics education and research.

The Real Data Corpus (80TB)
- Data from ≈4000 used hard drives, phones, and other devices purchased abroad.
- Organized by country.
- IRB approval required to use.

“GOVDOCS” (500GB)
- ≈1 million files downloaded from US government web servers.
- No restrictions on use or redistribution.

Scenarios (multiple computers, packet captures, memory, etc.) (2TB)
- Designed to teach digital forensics.
  —+50 universities, +1000 students have used in past 3 years.
- No restrictions on use or redistribution.
- Can also be used for original research.

The “digital corpora” are part of a broad research effort in digital forensics.

Identifying high-value data (Triage)
- What is important?
  - Contacts, calendar, documents?
  - Geolocation information?

Making extracted data available for automated analysis
- Data representations (DFXML)
- Programming environments (DFXML Toolkit)

Creating new approaches for correlation (Cross-drive analysis)
- Identify previously unknown organizations or networks
- Identify data that is unusual or emerging

Sector hashing for triage and fragment recovery

This talk is about using forensics tools to find privacy leaks!
Remember Snapchat?
Snapchat promised users that expired images could not be viewed unless “saved.”
Snapchat promised users that expired images could not be viewed unless “saved.”

Is there any way to view an image after the time has expired?
No, snaps disappear after the timer runs out. You can save snaps that you capture by pressing the save button on the preview screen.

What if I take a screenshot?
Screenshots can be captured if you’re quick. The sender will be notified if we detect you have taken a screenshot.

Usernames and Passwords
What do I do if I forgot my password? How do I reset/change my password?
You can reset your password from the Snapchat application. First attempt to login with your username/email and an incorrect password, the app will prompt you to “Try Again” or “Reset Password.” Tap “Reset Password” and follow the instructions sent to your email address to reset the password and access your account.

If you don’t receive a password reset email within an hour or so, be sure to check for it in your spam folder!

What do I do if I forgot my username?
OMG! — Expired images were not actually deleted. They were just hidden from view.

OMG, "Deleted" Snapchat Sexts Can Actually Be Recovered

By Will Oremus | Posted Thursday, May 9, 2013, at 3:56 PM

The premise of Snapchat is simple: Send a photo or short video to a friend, and it will self-destruct after 10 seconds. That way, it won’t wind up on the Internet and ruin anyone’s reputation, friendships, or career.

Needless to say, that has made it a wildly popular choice for sexting. But Snapchat’s appeal goes far beyond that. In an age in which “privacy” and “technology” have become almost antonymous, it has been billed as the anti-Facebook—a communications tool that deletes your data rather than preserving, analyzing, and trading on it. In short, it’s supposed to make messaging fun again.

But the app’s security has never been ironclad. As the media have repeatedly warned parents, and parents in turn warned their kids, message recipients can still save a compromising image by taking a quick screenshot. But Snapchat tries to mitigate the risk somewhat by automatically notifying the sender when that happens. If someone screenshots you, it’s a virtual slap in the face. If they don’t, you can assume you’re in the clear.

Except that apparently you can’t. KSL-TV in Utah reports that an Orem-based firm called Decipher Forensics has figured out a way to recover supposedly deleted images from the recipient’s phone. The process isn’t simple: 24-year-old Decipher forensics examiner Richard Hickman told the network that it takes him about six hours, on average, to image the phone’s data. So far he can only do it with Android devices, though he’s working on doing the same for iOS. But his firm is now offering to perform the recovery procedure for anyone who wants it, from parents
FTC charged Snapchat with deceptive practices. Snapchat settled.

Snapchat Settles FTC Charges That Promises of Disappearing Messages Were False

Snapchat Also Transmitted Users’ Location and Collected Their Address Books Without Notice Or Consent

FOR RELEASE

May 8, 2014


Snapchat, the developer of a popular mobile messaging app, has agreed to settle Federal Trade Commission charges that it deceived consumers with promises about the disappearing nature of messages sent through the service. The FTC case also alleged that the company deceived consumers over the amount of personal data it collected and the security measures taken to protect that data from misuse and unauthorized disclosure. In fact, the case alleges, Snapchat’s failure to secure its Find Friends feature resulted in a security breach that enabled attackers to compile a database of 4.6 million Snapchat usernames and phone numbers.

According to the FTC’s complaint, Snapchat made multiple misrepresentations to consumers about its product that stood in stark contrast to how the app actually worked.
This talk is about finding privacy-sensitive information in stored data.

Like Snapchat, GPS devices also retain private data not visible in the user interface.

Many kinds of data are “encoded.” Finding these requires “optimistic decoding.”

Our study of 1400 drives found thousands of email addresses that were only in compressed data.

Evaluating impact of simple obfuscation
Exploring a consumer GPS device for hidden private data.
The Garmin Nüvi works with GPS information.
The Garmin Nüvi works with GPS information.
The Garmin Nüvi shows where you are…
… but it retains much more information.

(taken with Garmin’s screen capture)
The “Trip log” shows where you have been.
The trip log data are kept in “.gpx” files.

These files are visible when the Garmin GPS is plugged into a computer.

Questions:

• What information is in these files?

• Is there information in these files not visible from the user interface?

• Is there privacy-sensitive information elsewhere on the device?
The file 24.gpx contains track information in XML format.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?><gpx xmlns="http://www.topografix.com/GPX/1/1"
xmlns:gpxx="http://www.garmin.com/xmlschemas/GpxExtensions/v3"
xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v2"
creator="nüvi 1390" version="1.1"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd"
    <metadata>
        <link href="http://www.garmin.com"><text>Garmin International</text></link>
        <time>2013-09-03T20:44:09Z</time>
    </metadata>
    <trk>
        <name>ACTIVE LOG: 23 AUG 2013 10:47</name>
        <trkseg>
            <trkpt lat="38.885255" lon="-77.114185"><ele>99.63</ele><time>2013-08-23T14:47:26Z</time><extensions>
                <gpxtpx:TrackPointExtension><gpxtpx:course>0.00</gpxtpx:course></gpxtpx:TrackPointExtension>
            </extensions></trkpt>
            <trkpt lat="38.885206" lon="-77.114113"><ele>89.54</ele><time>2013-08-23T14:47:49Z</time><extensions>
                <gpxtpx:TrackPointExtension><gpxtpx:speed>1.37</gpxtpx:speed><gpxtpx:course>0.00</gpxtpx:course></gpxtpx:TrackPointExtension>
            </extensions></trkpt>
            <trkpt lat="38.885058" lon="-77.114068"><ele>88.58</ele><time>2013-08-23T14:47:54Z</time><extensions>
                <gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</gpxtpx:speed><gpxtpx:course>180.71</gpxtpx:course></gpxtpx:TrackPointExtension>
            </extensions></trkpt>
            <trkpt lat="38.885008" lon="-77.114062"><ele>88.10</ele><time>2013-08-23T14:47:55Z</time><extensions>
                <gpxtpx:TrackPointExtension><gpxtpx:speed>5.49</gpxtpx:speed><gpxtpx:course>177.88</gpxtpx:course></gpxtpx:TrackPointExtension>
            </extensions></trkpt>
            <trkpt lat="38.884954" lon="-77.114056"><ele>88.10</ele><time>2013-08-23T14:47:56Z</time><extensions>
            </extensions></trkpt>
            <trkpt lat="38.884495" lon="-77.113991"><ele>85.69</ele><time>2013-08-23T14:48:05Z</time><extensions>
            </extensions></trkpt>
            <trkpt lat="38.883690" lon="-77.114060"><ele>80.41</ele><time>2013-08-23T14:48:09Z</time><extensions>
            </extensions></trkpt>
        </trkseg>
    </trk>
</gpx>
```
Each GPX XML record is roughly 266 bytes


The file 24.gpx has 766,553 ÷ 266 ≈ 2,900 tracking points in it.
Information not visible through the user interface:

- Elevation
- Speed
This program extracts and tabulates `gpxtpx:speed` records from the GPX files:

```python
import xml.sax, os, glob

class SpeedReader(xml.sax.ContentHandler):
    def __init__(self):
        xml.sax.ContentHandler.__init__(self)
        self.cdata = ""
        self.speeds = []

    def startElement(self,name,attrs):
        self.inElement = name
        self.cdata = ""

    def characters(self,content):
        self.cdata += content

    def endElement(self,name):
        if name=="gpxtpx:speed":
            self.speeds.append(float(self.cdata))

def get_gpx_speeds(fn):
    sr = SpeedReader()
    try:
        xml.sax.parse(open(fn,"rb"),sr)
    except xml.sax._exceptions.SAXParseException:
        pass
    print(fn,min(sr.speeds),max(sr.speeds),len(sr.speeds))

if __name__=="__main__":
    for fn in glob.glob("*.gpx"):
        get_gpx_speeds(fn)
```
The program documents that the GPS was moving faster than 34 m/s (≈75 MPH)

Lesson:

Retained data frequently contains privacy-sensitive information not visible from the user interface.

<table>
<thead>
<tr>
<th>FILE</th>
<th>MIN</th>
<th>MAX</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.gpx</td>
<td>1.37</td>
<td>28.82</td>
<td>4188</td>
</tr>
<tr>
<td>13.gpx</td>
<td>1.37</td>
<td>28.82</td>
<td>4015</td>
</tr>
<tr>
<td>14.gpx</td>
<td>1.37</td>
<td>30.2</td>
<td>10095</td>
</tr>
<tr>
<td>15.gpx</td>
<td>1.37</td>
<td>28.82</td>
<td>3827</td>
</tr>
<tr>
<td>16.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td>9830</td>
</tr>
<tr>
<td>17.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td>8664</td>
</tr>
<tr>
<td>18.gpx</td>
<td>1.37</td>
<td>26.08</td>
<td>3405</td>
</tr>
<tr>
<td>19.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td></td>
</tr>
<tr>
<td>20.gpx</td>
<td>1.37</td>
<td>32.94</td>
<td></td>
</tr>
<tr>
<td>21.gpx</td>
<td>1.37</td>
<td>32.94</td>
<td></td>
</tr>
<tr>
<td>22.gpx</td>
<td>1.37</td>
<td>32.94</td>
<td></td>
</tr>
<tr>
<td>23.gpx</td>
<td>1.37</td>
<td>32.94</td>
<td></td>
</tr>
<tr>
<td>24.gpx</td>
<td>1.37</td>
<td>32.94</td>
<td></td>
</tr>
<tr>
<td>25.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td>5045</td>
</tr>
<tr>
<td>26.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td>8826</td>
</tr>
<tr>
<td>27.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td></td>
</tr>
<tr>
<td>28.gpx</td>
<td>1.37</td>
<td>34.31</td>
<td></td>
</tr>
<tr>
<td>29.gpx</td>
<td>1.37</td>
<td>28.82</td>
<td>7700</td>
</tr>
<tr>
<td>30.gpx</td>
<td>1.37</td>
<td>28.82</td>
<td>3774</td>
</tr>
<tr>
<td>31.gpx</td>
<td>1.37</td>
<td>17.84</td>
<td>996</td>
</tr>
</tbody>
</table>

129,773 records
20 files
Are there more hidden data?
My Nüvi *always* has 20 files in the /GPS/Archive directory

Can the deleted files be recovered?
Can the deleted *data* be recovered?
Are there more hidden data? My Nüvi *always* has 20 files in the /GPS/Archive directory.

Can the deleted files be recovered? — no

Can the deleted *data* be recovered? — yes.
The Sleuth Kit is an open source digital forensics toolset.

The **fls** program shows ‘deleted’ files and directories on storage devices.

- “*” indicates that the file is deleted:

```
+++++ d/d 23676-144-1:    CrashReports
+++++ r/d * 23442-48-2(realloc):  GoogleUpdateHelper.msi
+++++ r/d * 23442-144-1(realloc):  GoogleUpdateHelper.msi
+++++ r/d * 23439-48-2(realloc):  goopdate.dll
+++++ r/d * 23439-144-1(realloc):  goopdate.dll
+++++ r/d * 23441-48-2(realloc):  GoopdateBho.dll
+++++ r/d * 23441-144-1(realloc):  GoopdateBho.dll
+++++ r/d * 23443-48-2(realloc):  goopdateres_ar.dll
+++++ r/d * 23443-144-5(realloc):  goopdateres_ar.dll
+++++ r/d * 23444-48-2(realloc):  goopdateres_bg.dll
+++++ r/d * 23444-144-5(realloc):  goopdateres_bg.dll
```

The “fls” program shows deleted files with a “*”: 

```
``
We made a forensic image of the GPS “usb drive” and analyzed it with SleuthKit

SleuthKit shows no deleted files in the /GPS/Archive/ directory:

```
$ fls -r SLG-GARMAIN.E01  28
  d/d  28:   GPX
+   r/r 54419718:   Current.gpx
+   d/d 54419720:   Archive
++   r/r 57598982:   21.gpx
++   r/r 57598984:   22.gpx
++   r/r 57598986:   23.gpx
++   r/r 57598988:   24.gpx
++   r/r 57598990:   25.gpx
++   r/r 57598992:   26.gpx
++   r/r 57598994:   27.gpx
++   r/r 57598996:   28.gpx
++   r/r 57598998:   29.gpx
++   r/r 57599000:   30.gpx
++   r/r 57599002:   31.gpx
++   r/r 57599004:   32.gpx
++   r/r 57599006:   12.gpx
++   r/r 57599008:   13.gpx
++   r/r 57599010:   14.gpx
++   r/r 57599012:   15.gpx
++   r/r 57599014:   16.gpx
++   r/r 57599016:   17.gpx
++   r/r 57599018:   18.gpx
++   r/r 57599020:   19.gpx
++   r/r 57599022:   20.gpx
```

Is there deleted Garmin XML that is not in a deleted file?
“Bulk data analysis” looks at the Garmin’s storage device sector-by-sector.

At the sector level, there are many XML records that are not in files.
Our “bulk_extractor” program scans bulk data for forensically significant data.

Output is a directory containing:
- “Feature files;” histograms; carved objects
- Mostly in UTF-8; some XML
- Can be processed with Python-based tools

“Digital media triage with bulk data analysis and bulk_extractor,”
bulk_extractor has a scanner for GPX XML records.

GPX XML found on the drive is stored in the file “gps.txt”

We processed the Garmin image with bulk_extractor and analyzed the GPS records:

```
$ more gps.txt
# BULK_EXTRACTOR-Version: 1.4.1 ($Rev: 10844 $)
# Feature-Recorder: gps
# Filename: SLG-GARMAIN.E01
# Feature-File-Version: 1.1
83886185        ,,,,,,111.53
83886435        2013-06-29T17:31:34Z,38.886777,-77.144822,99.15,10.98,114.3
83886685        2013-06-29T17:31:50Z,38.886160,-77.143053,100.60,8.24,115.7
83886934        2013-06-29T17:32:02Z,38.885877,-77.142233,99.63,5.49,115.76
83887183        2013-06-29T17:32:04Z,38.885827,-77.142125,99.15,4.12,131.29
83887432        2013-06-29T17:32:05Z,38.885727,-77.142082,99.15,4.12,151.06
83887681        2013-06-29T17:32:06Z,38.885727,-77.142003,98.67,5.49,168.00
```

### Table 1: GPS Data

<table>
<thead>
<tr>
<th>Time</th>
<th>Lat</th>
<th>Lon</th>
<th>Elv</th>
<th>Sp</th>
<th>Hd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-06-29T17:31:34Z</td>
<td>38.886777</td>
<td>-77.144822</td>
<td>99.15</td>
<td>10.98</td>
<td>114.3</td>
</tr>
<tr>
<td>2013-06-29T17:31:50Z</td>
<td>38.886160</td>
<td>-77.143053</td>
<td>100.60</td>
<td>8.24</td>
<td>115.7</td>
</tr>
<tr>
<td>2013-06-29T17:32:02Z</td>
<td>38.885877</td>
<td>-77.142233</td>
<td>99.63</td>
<td>5.49</td>
<td>115.76</td>
</tr>
<tr>
<td>2013-06-29T17:32:04Z</td>
<td>38.885827</td>
<td>-77.142125</td>
<td>99.15</td>
<td>4.12</td>
<td>131.29</td>
</tr>
<tr>
<td>2013-06-29T17:32:05Z</td>
<td>38.885727</td>
<td>-77.142082</td>
<td>99.15</td>
<td>4.12</td>
<td>151.06</td>
</tr>
<tr>
<td>2013-06-29T17:32:06Z</td>
<td>38.885727</td>
<td>-77.142003</td>
<td>98.67</td>
<td>5.49</td>
<td>168.00</td>
</tr>
</tbody>
</table>
Thousands of entries are *earlier* than the first GPX file:

```
$ cut -f 2 gps.txt | grep 2013 | sort | head -2
2013-02-10T22:21:43Z,38.884835,-77.159409,92.42,4.12,354.35
2013-02-10T22:21:57Z,38.885625,-77.159516,94.83,6.86,357.18
```

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Allocated</th>
<th>Bulk Data Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Range</td>
<td>June 24 - Oct 12</td>
<td>Feb 10 - Oct 12</td>
</tr>
<tr>
<td># Entries</td>
<td>129,773</td>
<td>180,589</td>
</tr>
</tbody>
</table>

Conclusion:

- A substantial amount of data can be recovered that is invisible.
Garmin could improve privacy and transparency by applying basic privacy engineering.

Example mitigation strategies:

• Do not record information that’s not needed.
  — *Speed & elevation*

• Overwrite GPX files when they are deleted

• Not recording track information when tracks are not displayed.

Improving privacy would negatively impact the value of digital forensics:

• Often the most valuable data in an investigation are hidden data.
  — *Visible data tips off the suspect.*

• Investigators want as much data as possible.
  — *If the system throws out data, it might throw out something valuable.*
In George Ford’s Murder trial, critical information came from a hidden GPS.

- Murder of Shyanne Somers, 12, on July 8, 2007
- Convicted on 2009
- Key evidence: Tracking data from GPS hidden by Ford’s wife

If information is invisible, the criminal won’t try to erase it.
Optimistic decoding finds more private data.
Most digital forensic tools extract data, producing reports and searchable databases.

**Phone**
- Databases
- Email addresses
- SMS records
- Address book records
- “Chat”

**Documents**
- Names
- Email addresses
- Places
- Date/Time information
Forensic tools extract “features” from recognized file types.

For example, email addresses are extracted from Word files by:

• Converting the file to text
• Processing the text with regular expressions.
Regular expressions can also extract features from data not in files — “bulk data.”

[bulk data] ➞ RegEx ➞ Email Addresses

(This is how the GPX scanner works.)

<table>
<thead>
<tr>
<th>Folders.pst</th>
<th>Presentation.pptx</th>
<th>Sequestration.docx</th>
</tr>
</thead>
<tbody>
<tr>
<td>a097 83a1 ed96 26a6 3c69 3d0f 750a 2399</td>
<td>a2b5 bea7 692f 5847 a38a dd53 082c add5</td>
<td>5061 b64c 721d 864b 90b6 b55f bb04 735c</td>
</tr>
<tr>
<td>a2b5 bea7 692f 5847 a38a dd53 082c add5</td>
<td>5061 b64c 721d 864b 90b6 b55f bb04 735c</td>
<td>9448 6730 5453 df64 813e b603 5795 2242</td>
</tr>
<tr>
<td>9448 6730 5453 df64 813e b603 5795 2242</td>
<td>9448 6730 5453 df64 813e b603 5795 2242</td>
<td>e9c8 7454 7322 7cdc b60e 97af 2f64 2728</td>
</tr>
<tr>
<td>e9c8 7454 7322 7cdc b60e 97af 2f64 2728</td>
<td>e9c8 7454 7322 7cdc b60e 97af 2f64 2728</td>
<td>3cfb 84bd 2a84 2dfe 50ea 5935 c349 1513</td>
</tr>
<tr>
<td>3cfb 84bd 2a84 2dfe 50ea 5935 c349 1513</td>
<td>3cfb 84bd 2a84 2dfe 50ea 5935 c349 1513</td>
<td>a9e9 e92c a3f8 6e46 0530 8a88 c7a2 5d2b</td>
</tr>
<tr>
<td>a9e9 e92c a3f8 6e46 0530 8a88 c7a2 5d2b</td>
<td>a9e9 e92c a3f8 6e46 0530 8a88 c7a2 5d2b</td>
<td>d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b</td>
</tr>
<tr>
<td>d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b</td>
<td>d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b</td>
<td></td>
</tr>
</tbody>
</table>
It’s easy to see email addresses in bulk data.

- Folders.pst
- Presentation.pptx
- Sequestration.docx
- Mother.JPG

```
a097 83a1 ed96 26a6 3c69 3d0f 750a 2399 ......&.<i=.u.#.
a2b5 bea7 692f 5847 a38a dd53 082c add5 .....i/XG...S...,  
5061 b64c 721d 864b 90b6 b55f bb04 135c Pa.Lr..K..._...s\  
9448 6730 5453 df64 813e b603 5795 7142 .Hg0TS.d>..W."B 
e9c8 7454 7322 7cd8 b85d 6a26 85b2 65c1 ..tTs"|....../d'(
3cfb 84bd 2a84 2dfe 545d 89e3 2b2d 0c26 <XYZ@COMPANY.COM 
a9e9 e92c a3f8 6e46 0a51 de3a b060 7d7d ...,,..nF.0....]++
d89d 77cc fe1e f637 f515 dva1 1b47 098b ..w....7......G.. 
```
Email addresses are a sequence of characters. This sequence can be encoded in many ways.

XYZ@company.com

- **Unicode UTF-8**: (Email)
  
  5859 5a40 636f 6d70 616e 792e 636f 6d 6d  XYZ@company.com

- **Unicode UTF-16**: (MS Word)
  
  5800 5900 5a00 4000 6300 6f00 6d00 7000 7000  X.y.z.@.c.o.m.p.
  6100 6e00 7900 2e00 6300 6f00 6d00 6d  a.n.y...c.o.m

- **Base 64**: (Email attachments)
  
  5746 6c61 5147 4e76 6258 4268 4268 626e 6b75  Y29tCg===
  5932 3974 4367 3d3d 3d0a 3d3d 3d3d 3d0a

- **Compressed**: (Browser cache)
  
  1f9d 9058 b268 0132 e64d 1b38 61dc e471 51b0 8d02
  ...X.h.2.M.8a..q 51b0 8d02
Compression works by eliminating repeated sequences:

Computers use compression to save memory:

```
5859 5a40 636f 6d70 616e 792e 636f 6d20 XYZ@company.com
4142 4340 636f 6d70 616e 792e 636f 6d20 ABC@company.com
4445 4640 636f 6d70 616e 792e 636f 6d20 DEF@company.com
```

Compressed with “gzip:”

```
1f8b 0800 0000 0000 0203 8b88 8c72 48ce .............rH.
cf2d 48cc abd4 03d2 0a8e 4ece 287c 1757 .-H........N.( | .W
3714 3e00 b455 c1c5 3000 0000 7.>..U..0...
```

Compressed email addresses do not “look” like email addresses!

—Forensic tools must decompress first to identify compressed email addresses.
It’s hard to see compressed email address in bulk data.
It’s hard to see compressed email address in bulk data.

<table>
<thead>
<tr>
<th>Folders.pst</th>
<th>Presentation.pptx</th>
<th>Sequestration.docx</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:XYZ@company.com">XYZ@company.com</a></td>
<td><a href="mailto:ABC@company.com">ABC@company.com</a></td>
<td><a href="mailto:DEF@company.com">DEF@company.com</a></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>e327 962d 6450 3d91 c945 3bed 97a6 76a9 0bbd</td>
<td>.'-dP=..E;......</td>
<td>............rH.</td>
</tr>
<tr>
<td>1a00 0800 0000 0000 0000 0000 0000 0000</td>
<td>....-H......N.(</td>
<td>.W</td>
</tr>
<tr>
<td>8cc abd4 03d2 0000 0000 0000 0000 0000</td>
<td>7.&gt;..U..0........</td>
<td>7.&gt;..U..0........</td>
</tr>
<tr>
<td>714 3e00 b455 c1c5 3f6d a5b7 9c5d 7ed4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0a8e 4ece 287c 1757 3714 3e00 a175</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a097 83a1 ed96 26a6 3c69 3d0f 750a 2399</td>
<td>......&amp;.&lt;i=.u.#.</td>
<td>......&amp;.&lt;i=.u.#.</td>
</tr>
<tr>
<td>a2b5 bea7 692f 5847 a38a dd53 082c add5</td>
<td>....i/XG...S,...</td>
<td>....i/XG...S,...</td>
</tr>
<tr>
<td>5061 b64c 721d 864b 90b6 b55f bb04 735c</td>
<td>Pa.Lr..K..._...s\</td>
<td>Pa.Lr..K..._...s\</td>
</tr>
<tr>
<td>9448 6730 5453 df64 813e b603 5795 2242</td>
<td>.Hg0TS.d&gt;...W.&quot;B</td>
<td>.Hg0TS.d&gt;...W.&quot;B</td>
</tr>
<tr>
<td>e948 7454 7322 7cdc b60e 97af 2f64 2728</td>
<td>..tTs&quot;</td>
<td>....../d'(</td>
</tr>
<tr>
<td>84bd 2a84 2dfe 50ea 5935 c349 1513</td>
<td>&lt;<a href="mailto:XYZ@COMPANY.COM">XYZ@COMPANY.COM</a></td>
<td>&lt;<a href="mailto:XYZ@COMPANY.COM">XYZ@COMPANY.COM</a></td>
</tr>
<tr>
<td>e9 e92c a3f8 6e46 0530 8a88 c7a2 5d2b</td>
<td>....,...nF.0..... ]+</td>
<td>....,...nF.0..... ]+</td>
</tr>
<tr>
<td>d89d 77cc fe1e f637 f3f3 d0af 1b47 c09b</td>
<td>....w.....7......G.</td>
<td>....w.....7......G.</td>
</tr>
</tbody>
</table>
Existing digital forensic tools ignore compressed/encoded features in bulk data.

Compressed file formats:
- PDF file fragments
- ZIP archives
- Microsoft Office “x” files (docx, pptx, xlsx).

Other sources include:
- Web browser caches
- Windows Hibernation files.

How much is being missed?
How many encoded features are conventional tools missing?
Using the Real Data Corpus, we show that optimistic decoding is important for forensic tools.

Start with many hard drives

Extract email addresses with & without optimistic decoding

Are there significant numbers of email addresses that non-optimistic tools would miss?
Email addresses can be in files

Files
- Documents
- Address book
- Email messages

Browser Cache:
- Web mail
- Facebook Data

ABC@company.com
DEF@company.com
Email addresses can be in non-file disk sectors

ABC@company.com
DEF@company.com

XYZ@company.com

Swap Files
Hibernation Files
File fragments
Some may be in both files and in non-files. (A file that’s read into RAM before the system hibernates.)
This Venn Diagram represents email addresses on media.

- ABC@company.com
- DEF@company.com
- XYZ@company.com
- Alice@net.net

Set A: Swap Files, Hibernation Files, File fragments
The number of email addresses in each region depends on the media.
Email addresses can be plain text. “XYZ@company.com”
Email addresses can be compressed or encoded.

“x....rH..-H.......N.(|.W7.>..u..”
Each email address can be present plain, compressed, or both.

- Plain email addresses: XYZ@company.com
- Compressed email addresses: x....rH..-H..... ..N.(|.W7.>...u...
- Both
There are four different conditions for an email address on the media.

1) Plain in Files
2) Comp. in Files
3) Plain in non-files
4) Comp in non-files

Condition #4 is invisible to today’s forensic tools.

How significant is this?
We devised an experiment to determine the size of condition #4 for a specific drive.

First, find and remove the plain email addresses in files.

![Venn diagram](image)
...We remove the addresses compressed and in files....
...Remove email addresses that are not compressed.
...those that remain are the “invisible” email addresses.

- Email addresses in files
  - Plain in files
  - Compressed in files

- Email addresses in non-files
  - Plain in non-files
  - Compressed in non-files

- Email addresses in Slack space; Swap Files

Invisible to today’s tools
bulk_extractor implements recursive re-analysis.

Data that are decompressed are re-analyzed.

bulk_extractor finds both plain and compressed text in bulk data
• Both in “files” and in unallocated space.
bulk_extractor’s internal design:
The “feature files” show where the extracted features were found... and how they were encoded.

Plain text features have numeric offsets:

- **392175418**
  - `WindowsXP@gn.microsoft.com`  
  - Name=`WindowsXP@gn.microsoft.com\015\012`

Compressed features indicated by offset & decoding algorithm:

- **3772517888-GZIP-28322**
  - `user@company.com`  
  - onterey-<nobr>`user@company.com`</nobr>
Post-processing can determine if a feature was in a file. (Both allocated and deleted files.)

Offset: 392175418  
Feature: WindowsXP@gn.microsoft.com  
Context: \012[User]\015\012Name=WindowsXP@gn.microsoft.com \015\012Password=B@ji0  
Filename: WINDOWS/system32/oobe/migx25a.dun  
MD5: 2b00042f7481c7b056c4b410d28f33cf

For each feature, we can category #1, #2, #3 or #4!
We ran bulk_extractor and identify_filenames.py on RDC drive IN10-0138 and examined the email encodings:

<table>
<thead>
<tr>
<th>Emails seen</th>
<th>count</th>
<th>1) Plain in Files</th>
<th>2) Comp. in Files</th>
<th>3) Plain in non-files</th>
<th>4) Comp in non-files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cleartext</td>
<td></td>
<td>358</td>
<td>--</td>
<td>5341</td>
<td>--</td>
</tr>
<tr>
<td>All Comp</td>
<td></td>
<td>--</td>
<td>9</td>
<td>--</td>
<td>135</td>
</tr>
<tr>
<td>GZIP</td>
<td>50</td>
<td>14</td>
<td></td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>HIBER</td>
<td>39</td>
<td>7</td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>HIBER-GZIP</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>PDF</td>
<td>88</td>
<td>1</td>
<td></td>
<td></td>
<td>87</td>
</tr>
<tr>
<td>ZIP</td>
<td>28</td>
<td>7</td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>ZIP-PDF</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>

135 out of 5700 email addresses are invisible to existing tools.
There are thus four kinds of email addresses on media.

Email addresses

in files

Email addresses

in Slack space;
Swap Files

Plain email addresses

XYZ@company.com

Compressed email addresses

x....rH..-H....
..N.(|.W7.>..u..

We analyzed 1,646 RDC disk images with intact file systems. Many email addresses existed only encoded, in non-files.

<table>
<thead>
<tr>
<th>Coding</th>
<th>Drives</th>
<th>Emails</th>
<th>avg</th>
<th>max</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Plain in files</td>
<td>739</td>
<td>81,920</td>
<td>110</td>
<td>4,206</td>
<td>253</td>
</tr>
<tr>
<td>2) Comp in files</td>
<td>355</td>
<td>19,711</td>
<td>55</td>
<td>5,454</td>
<td>388</td>
</tr>
<tr>
<td>3) Plain in non-files</td>
<td>860</td>
<td>1,956,059</td>
<td>2,274</td>
<td>178,073</td>
<td>9,248</td>
</tr>
<tr>
<td>4) Comp in non-files</td>
<td>474</td>
<td>165,481</td>
<td>349</td>
<td>59,376</td>
<td>2,889</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coding</th>
<th>Drives</th>
<th>Emails</th>
<th>avg</th>
<th>max</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE64 Comp</td>
<td>54</td>
<td>219</td>
<td>4</td>
<td>50</td>
<td>7</td>
</tr>
<tr>
<td>BASE64-GZIP Comp</td>
<td>2</td>
<td>64</td>
<td>32</td>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td>GZIP Comp</td>
<td>234</td>
<td>66,195</td>
<td>282</td>
<td>9,103</td>
<td>981</td>
</tr>
<tr>
<td>GZIP-BASE64 Comp</td>
<td>7</td>
<td>44</td>
<td>6</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>GZIP-GZIP Comp</td>
<td>15</td>
<td>12,663</td>
<td>844</td>
<td>11,845</td>
<td>2,944</td>
</tr>
<tr>
<td>GZIP-GZIP-BASE64 Comp</td>
<td>2</td>
<td>38</td>
<td>19</td>
<td>30</td>
<td>11</td>
</tr>
<tr>
<td>GZIP-GZIP-GZIP Comp</td>
<td>4</td>
<td>58</td>
<td>14</td>
<td>38</td>
<td>14</td>
</tr>
<tr>
<td>GZIP-GZIP-ZIP Comp</td>
<td>1</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>GZIP-PDF Comp</td>
<td>5</td>
<td>38</td>
<td>7</td>
<td>30</td>
<td>11</td>
</tr>
<tr>
<td>GZIP-ZIP Comp</td>
<td>6</td>
<td>49</td>
<td>8</td>
<td>30</td>
<td>9</td>
</tr>
<tr>
<td>HIBER Comp</td>
<td>79</td>
<td>1,433</td>
<td>18</td>
<td>217</td>
<td>44</td>
</tr>
<tr>
<td>PDF Comp</td>
<td>162</td>
<td>2,352</td>
<td>14</td>
<td>238</td>
<td>31</td>
</tr>
<tr>
<td>ZIP Comp</td>
<td>388</td>
<td>85,252</td>
<td>219</td>
<td>59,369</td>
<td>3,025</td>
</tr>
<tr>
<td>ZIP-BASE64 Comp</td>
<td>5</td>
<td>30</td>
<td>6</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>ZIP-BASE64-GZIP Comp</td>
<td>2</td>
<td>65</td>
<td>32</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>ZIP-GZIP Comp</td>
<td>14</td>
<td>261</td>
<td>18</td>
<td>132</td>
<td>34</td>
</tr>
<tr>
<td>ZIP-PDF Comp</td>
<td>26</td>
<td>115</td>
<td>4</td>
<td>18</td>
<td>4</td>
</tr>
</tbody>
</table>

Some drives had more than 10,000 compressed email addr.
Different encoding formats have different significance.

Likely PII sources:
- BASE64 — Email attachments
- GZIP — Web browser caches (esp. JSON)
- PDF — Personal Documents (PII)
- ZIP-PDF — PDF archives

Public Data:
- BASE64-GZIP — Open Source Software
- ZIP — Typically from JAR files
- ZIP-ZIP-ZIP — WinZip distributions
HIBER-GZIP typically contains web browser memory.

• JSON object downloaded from Facebook by compressed HTTP
• In RAM, written to HIBER on disk when the system went into sleep.
• Frequently captures SSL-encrypted data.

Conclusion:
• Optimistic encoding is important for finding identity information.
Finding more information with optimistic de-obfuscation
In 2013, a DoD partner told us that pre-processing forensic data with (XOR 255) yielded additional data. This didn’t make a lot of sense.

• What additional data?
• (XOR 255) is not a strong encryption algorithm!

I worked with four interns to figure out why.

• CDT Aubin Heffernan, USMA
• CDT Scott Horras, USMA
• CDT Kyle Gorak, USMA
• Ms. Carolina Zarate, Poolesville High School

Our plan:

• XOR all data in the Real Data Corpus
• Look for “interesting” information not visible without XOR.
We created a new “scanner” that inverts each buffer (XOR 255) and then reprocesses.
We searched for valid data that had been XOR’ed.

These kinds of data can be automatically recognized with high reliability.

DATA XOR(255) Windows PE header
DATA XOR(255) URL
DATA XOR(255) ZIP “Local Header”
DATA XOR(255) JPEG or Exif
We found substantial XOR’ed data.

These kinds of data can be recognized with high reliability.
We examined anti-virus systems and found:

- Malware used XOR(255) to hide download URLs
- AV XOR’ed Malware that was put into QUARANTINE
- VirusTotal did not recognize uploaded malware that had been XORed.

**XOR(255) in commercial software:**

- Real Audio — to obscure Dr. Yuriy Reznik’s email address.
- Nero 7 — to hide a watermark (http://www.nero.com)

**XOR(255) in Exfil’ed data:**

- Fragments of a ZIP file that had been XOR’ed.
- Contents were Excel spreadsheets with names & salary data.
XOR(255) is throughout the real data corpus:

<table>
<thead>
<tr>
<th>Year</th>
<th># URL</th>
<th># WinPE</th>
<th># ZIP</th>
<th>#Exif</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1981</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1985</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1990</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1996</td>
<td>2</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1997</td>
<td>185</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1998</td>
<td>443</td>
<td>126</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1999</td>
<td>261</td>
<td>526</td>
<td>44</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>252</td>
<td>526</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>2001</td>
<td>593</td>
<td>238</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2002</td>
<td>734</td>
<td>234</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2003</td>
<td>224</td>
<td>87</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004</td>
<td>1,359</td>
<td>427</td>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>2005</td>
<td>2,640</td>
<td>184</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2006</td>
<td>1,934</td>
<td>3,840</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2007</td>
<td>315</td>
<td>16,782</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2008</td>
<td>1,376</td>
<td>1,973</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2009</td>
<td>1,722</td>
<td>489</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2010</td>
<td>802</td>
<td>468</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2011</td>
<td>14,594</td>
<td>8</td>
<td>74</td>
<td>0</td>
</tr>
<tr>
<td>2013</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2014</td>
<td>49</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2016</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2018</td>
<td>818</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2019</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2023</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2027</td>
<td>346</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2029</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2030</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2033</td>
<td>218</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2037</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2080</td>
<td>20</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2081</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>no file</td>
<td>252,742</td>
<td>11,550</td>
<td>4,594</td>
<td>130</td>
</tr>
<tr>
<td>Total</td>
<td>281,712</td>
<td>37,533</td>
<td>4,769</td>
<td>130</td>
</tr>
</tbody>
</table>

Table 1: Validated XOR features by year for the analyzed drives, where the “year” is corresponds to the modification time of the file within which each XOR-encoded feature was found. “no file” indicates that the XOR-encoded features could not be located to a specific file. Timestamps prior to 1996 and after 2011 are likely the result of an improperly set system clock or on-disk corruption and are reported here for completeness.
XOR(255) was found in drives from (all) 21 countries.

<table>
<thead>
<tr>
<th>country</th>
<th>total drives</th>
<th>drives with XOR WinPE</th>
<th>drives with XOR URL</th>
<th>drives with XOR ZIP</th>
<th>drives with XOR exif</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANGLADESH</td>
<td>57</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BOSNIA AND HERZEGOVINA</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CANADA</td>
<td>48</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CHINA</td>
<td>807</td>
<td>25</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EGYPT</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GERMANY</td>
<td>37</td>
<td>22</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>GHANA</td>
<td>19</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GREECE</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INDIA</td>
<td>603</td>
<td>185</td>
<td>77</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>ISRAEL</td>
<td>260</td>
<td>84</td>
<td>39</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>MEXICO</td>
<td>173</td>
<td>73</td>
<td>16</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>MONACO</td>
<td>11</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>NEW ZEALAND</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PAKISTAN</td>
<td>81</td>
<td>31</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PALESTINE, STATE OF</td>
<td>140</td>
<td>39</td>
<td>8</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>SINGAPORE</td>
<td>34</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SWITZERLAND</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>THAILAND</td>
<td>17</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>TURKEY</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>UNITED ARAB EMIRATES</td>
<td>87</td>
<td>62</td>
<td>7</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>2,411</strong></td>
<td><strong>581</strong></td>
<td><strong>171</strong></td>
<td><strong>50</strong></td>
<td><strong>7</strong></td>
</tr>
</tbody>
</table>

Table 2: Incidence of drives with Validated XOR features, by country
The current XOR implementation significantly increases processing time.

<table>
<thead>
<tr>
<th>test image</th>
<th>Size</th>
<th>without XOR</th>
<th>with XOR</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>nps-2009-domexusers</td>
<td>40GB</td>
<td>522 sec</td>
<td>799 sec</td>
<td>+53%</td>
</tr>
<tr>
<td>nps-2011-2tb</td>
<td>2TB</td>
<td>34,140 sec</td>
<td>58,147 sec</td>
<td>+70%</td>
</tr>
</tbody>
</table>

Table 4: Observed processing times for `bulk_extractor` with and without XOR scanner.

There are opportunities to make de-obfuscation more efficient.

Solution 1 — Only use for specific cases.

Solution 2 — Examine data *before* XORing.

- Use cryptanalysis and knowledge of file headers, magic numbers, etc.
Digital forensics — what’s next?

This work — Bulk Data Analysis and Optimistic Decoding
• Basic engineering: more decompression algorithms, improved efficiency
• Science:
  —Combine bulk data with file analysis for more complete understanding of media
  —Develop approaches to increase precision and cluster returned results

Technical Privacy Auditing
• Developing a methodology for performing bit-level privacy audits.
  —Consumer operating systems, cell phones, in-vehicle systems.
  —Transport: analysis of app-to-server communications

Other forensic targets
• Small devices / Internet Of Things.
• Cloud
In conclusion:

There are persistent reports of sensitive information being left behind on consumer devices.

We showed how we can find hidden privacy-sensitive information in consumer GPS devices.

Optimistic decoding finds data present but hidden.

- De-compression
- De-obfuscation

The tools and some of the data discussed here are free for download.

- https://github.com/simsong/bulk_extractor
- http://www.digitalcorpora.org/