RAPID RECOGNITION OF BLACKLISTED FILES AND FRAGMENTS

MICHAEL MCCARRIN
BRUCE ALLEN

MANY THANKS TO:

- OSDFCon and Basis
- Bruce Allen
- Scott Young
- Joel Young
- Simson Garfinkel

All of whom have helped with this project immensely.

PROBLEM STATEMENT:
GIVEN A LIST OF INTERESTING FILES, FIND ALL DEVICES THAT CONTAIN AT LEAST ONE.

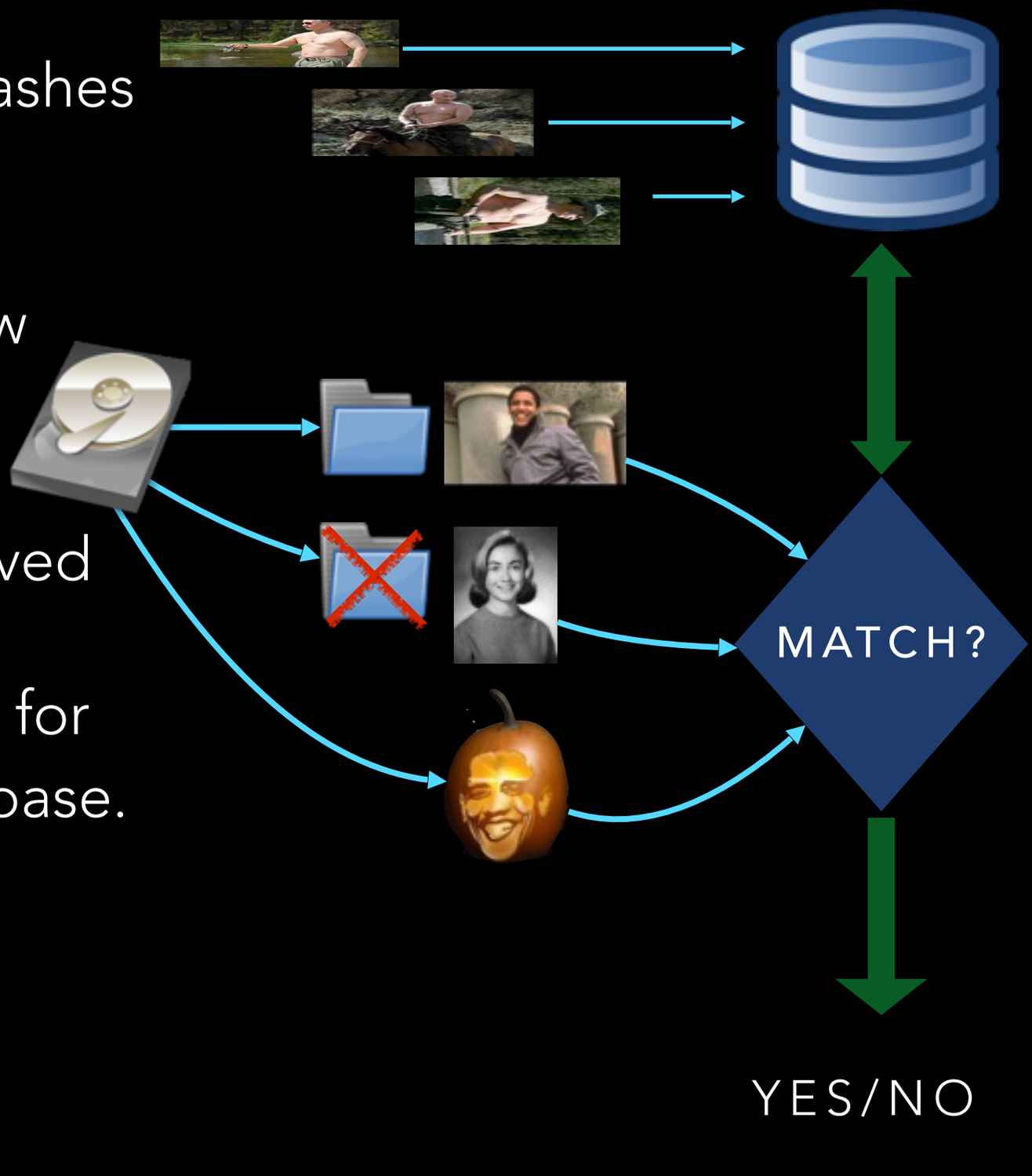
A COMMON SOLUTION:
HASH EVERYTHING. COMPARE HASHES.

1. Keep a database of the hashes of your interesting files.

2. Extract files from each new device you see:
   - Overt + undeleted + carved

3. Hash extracted files. Look for matches against the database.

4. Solved!



THERE ARE MANY COMMON SCENARIOS WHERE FILE HASHING DOESN'T WORK.

- Comparing cryptographic hashes only finds **exact** matches.

- Flip a bit, the hash changes completely.

- Everybody knows this already.

- Usually it comes up when people are worrying about mildly clever adversaries.

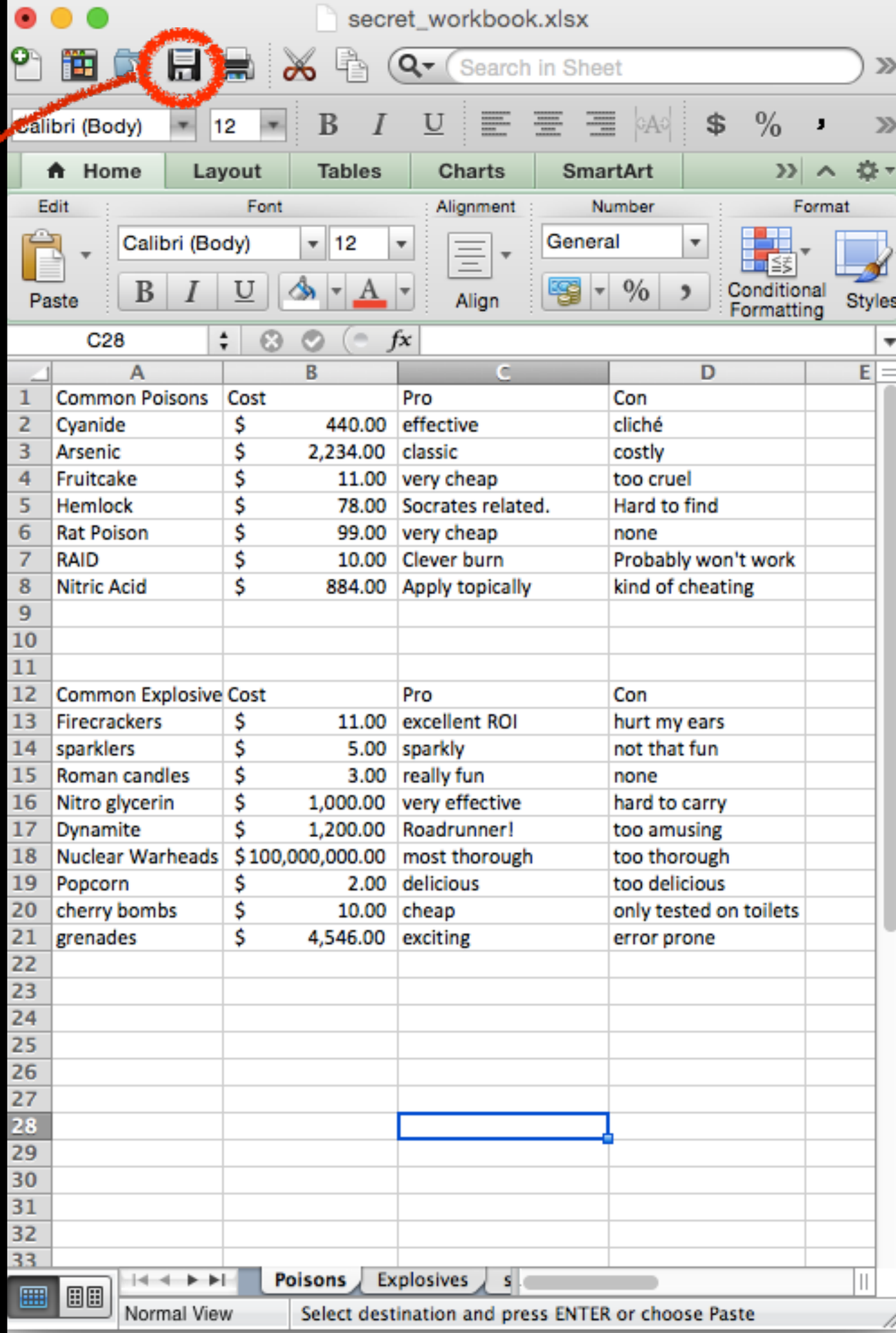
- **But**… there are many more mundane reasons to worry.

EXAMPLE 1: EXCEL CHANGES FILE CONTENT WHEN YOU SAVE

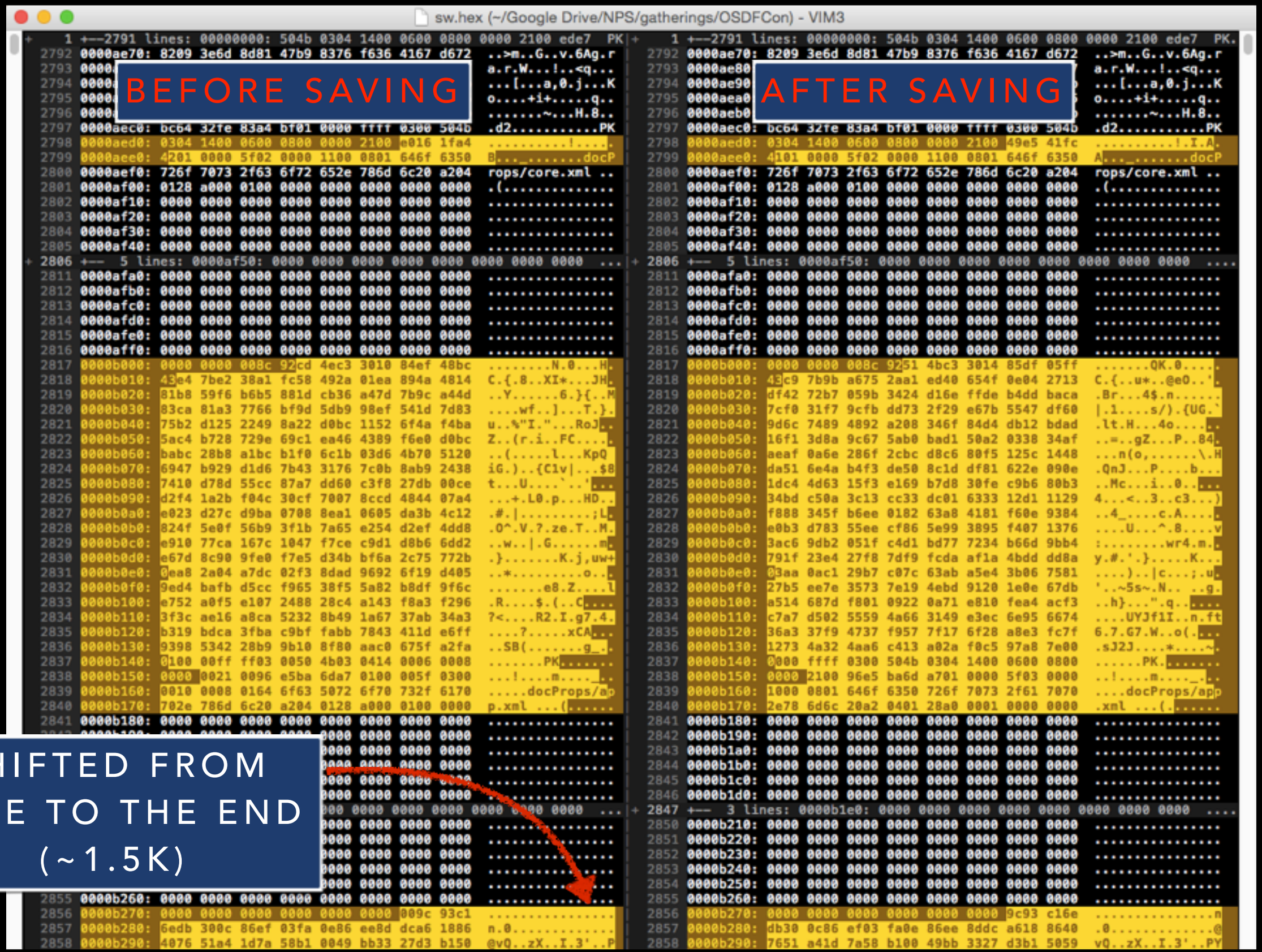
"GET NEW MD5"

How to get a new MD5 for your Excel file:

- Open your file.
- Click save.
- Complete!



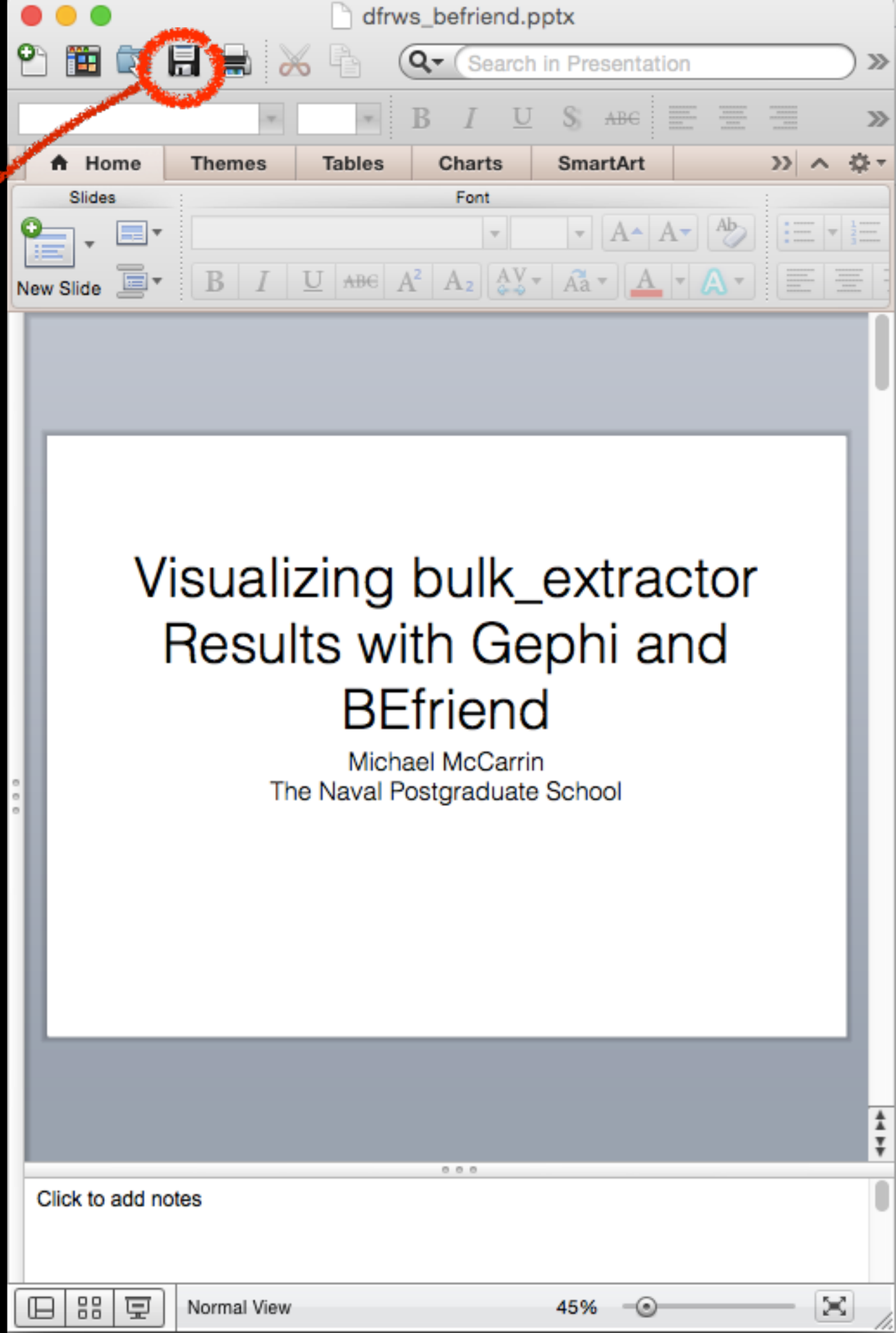
HERE'S THE HEX VIEW OF THE CHANGES



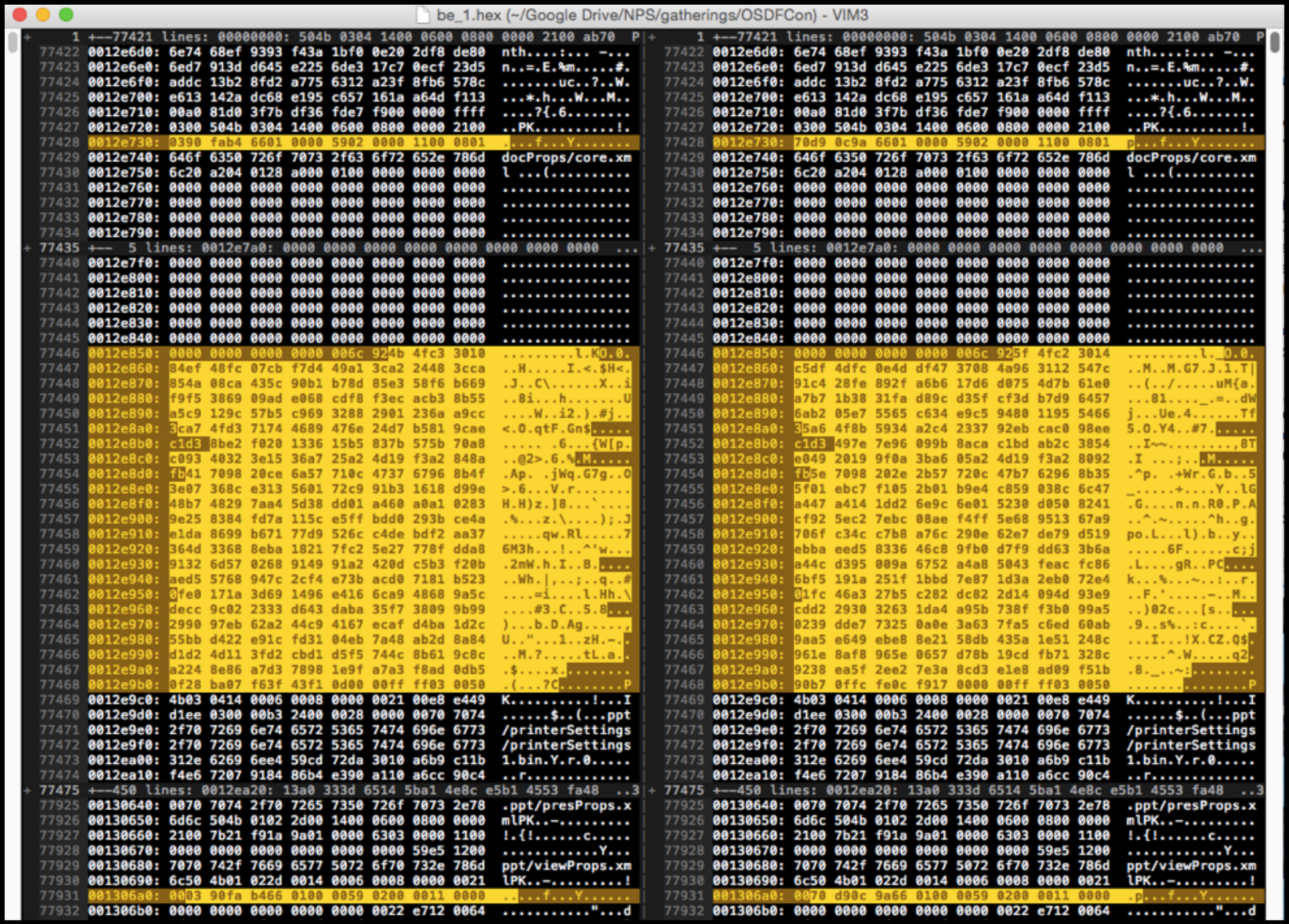
EXAMPLE 2: POWERPOINT EXHIBITS SIMILAR BEHAVIOR

"GET NEW MD5"

- Not as many changes, but one byte is enough.
- Note that the user content is 100% unchanged!



DIFF OF POWERPOINT HEX DUMPS BEFORE AND AFTER SAVING:



MD5 (PLAN.PDF) =
0E2CDDFF785730F97C93E087A270250A

M57 PATENTS EXTENSION

My Clever Plan

Jo Smith

1 USE MATH

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

$$\begin{aligned}(x+y)^3 &= (x+y)^2(x+y)\\ &= (x^2+2xy+y^2)(x+y)\\ &= (x^3+2x^2y+xy^2)+(x^2y+2xy^2+y^3)\\ &= x^3+3x^2y+3xy^2+y^3\end{aligned} \tag{1.1}$$

Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

1.1 NOW FOR THE CLEVER BIT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

$$A = \begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} \tag{1.2}$$

Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

MD5 (PLAN.PDF) =
04206496CA24F93A75C79186572E8670

M57 PATENTS EXTENSION

My Clever Plan

Jo Smith

1 USE MATH

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

$$\begin{aligned}(x+y)^3 &= (x+y)^2(x+y)\\ &= (x^2+2xy+y^2)(x+y)\\ &= (x^3+2x^2y+xy^2)+(x^2y+2xy^2+y^3)\\ &= x^3+3x^2y+3xy^2+y^3\end{aligned} \tag{1.1}$$

Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

1.1 NOW FOR THE CLEVER BIT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

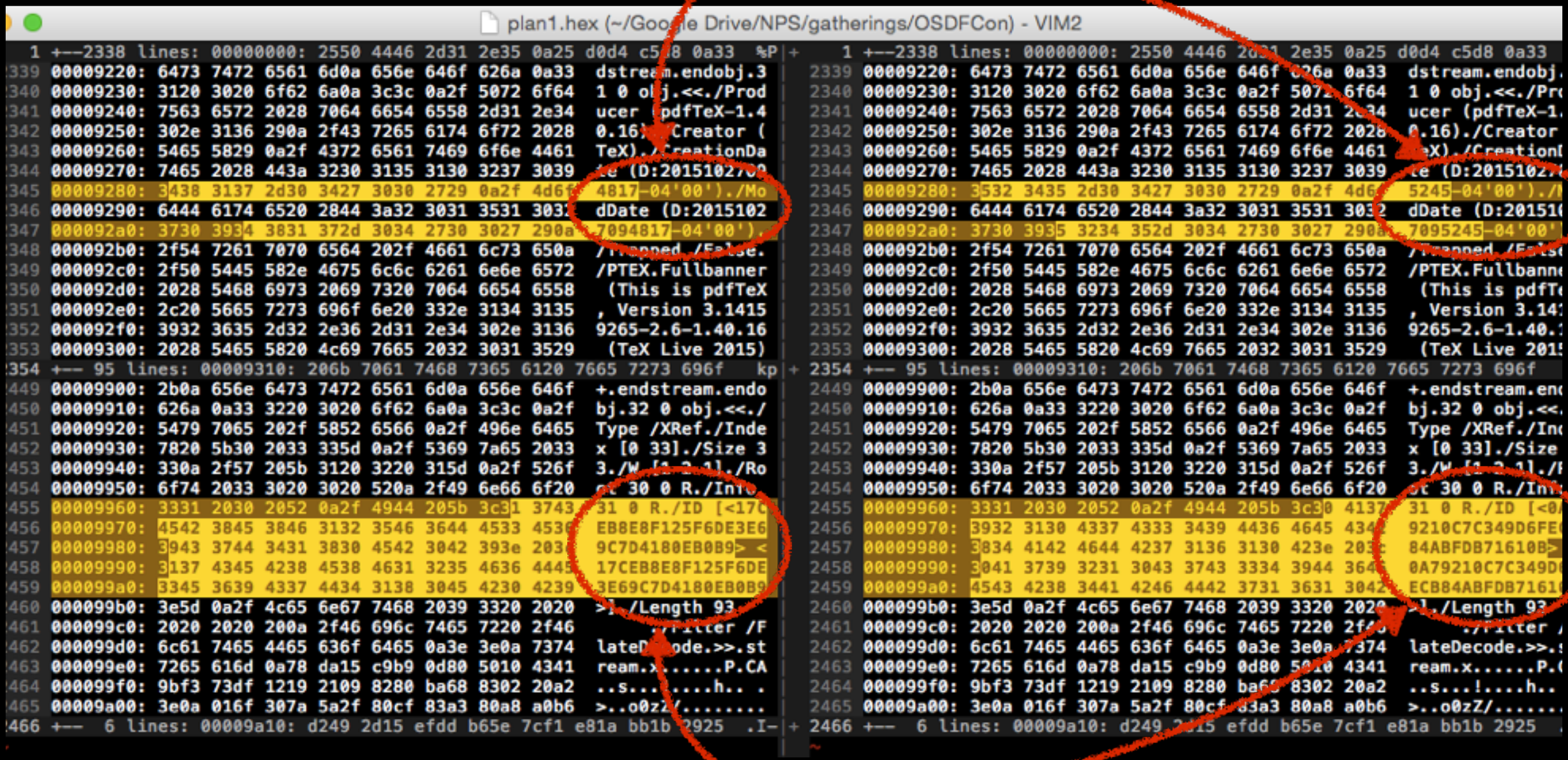
$$A = \begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} \tag{1.2}$$

Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

EXAMPLE 3: TWO "IDENTICAL" PDFS CREATED BY PDFLATEX

AGAIN, HERE IS THE HEX VIEW.

IF YOU LOOK CLOSELY YOU CAN SEE THE TIME STAMP HAS CHANGED



CONCLUSION: MANY APPLICATIONS CHANGE FILE-LAYER CONTENT.

- The point is not that your investigation relies on Excel files or finding PowerPoint presentations.
- The point is: **applications don’t care** about forensic integrity.
- The changes above are **transparent** to the user.

COROLLARY: WE MAY BE MISSING EVIDENCE.

Other things you won't find with file hashes:

- Files that are **deleted** and partially overwritten.
- Slightly **edited** files.
- **Corrupted** files in overt space.
- Badly **carved** files.
- **Fragments** of files in unallocated space.

WORSE: THESE ARE CASES WHERE THE SUSPECT IS NOT EVEN TRYING.

- At most, they require someone to click save or empty the recycling bin.
- The state produced by **default behavior** is much more concerning than any single clever person.
- Many reasons to improve this process, but the most compelling is **economic:**
  - **Solving the problem should be cost less than creating it.**

AN ARGUMENT FOR "APPROXIMATE MATCHING" STRATEGIES:

- Not a fancy thing: we often just want to find files with the **exact same** user-level content.

- Fundamentally, the approach is not that different:

1. Cut the object into pieces.

2. Hash the pieces.

3. Count matching hashes.

THIS PROBLEM HAS MANY KNOWN SOLUTIONS.

Numerous sophisticated matching schemes:

- Rabin fingerprinting, ssdeep, sdhash, frag_find, sector hashing…
  - Lots of good stuff—we try to build on it.
- A skilled examiner could find everything we mentioned.
- Many groups have in-house solutions.
- **Why write a new tool?**

REASONS FOR A NEW TOOL

1. Nothing on the previous slide will change the economic problem.

2. Make already developed techniques more available.

3. Getting it right is harder than it first seems.

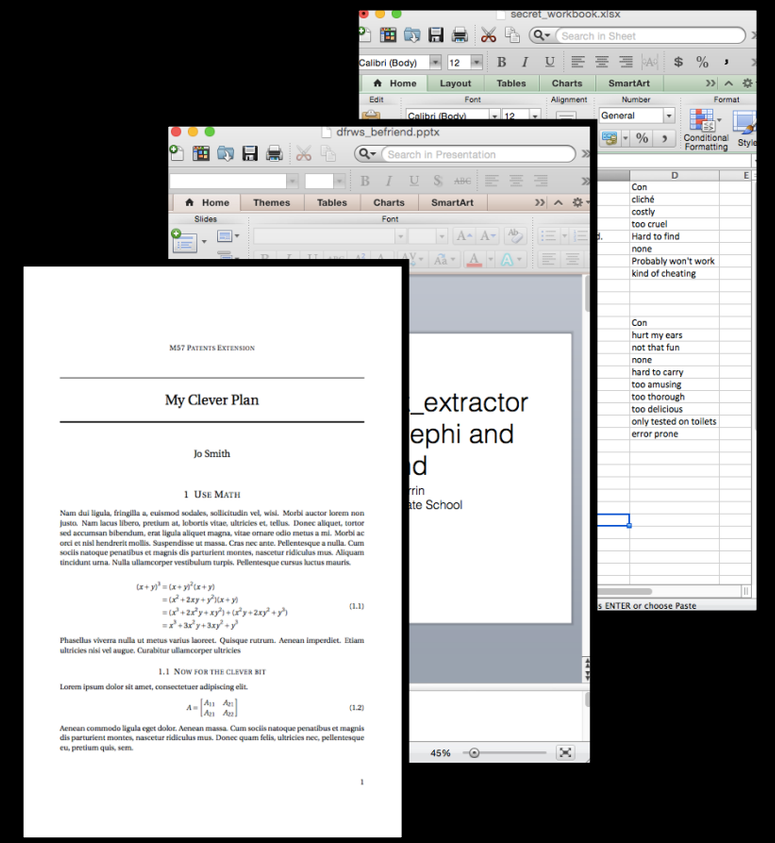
   - In particular, the feature selection problem has not been solved.

OUR GOALS: TRIVIAL, SCALABLE, COMPREHENSIBLE.

- **Trivial**: It has to be easier to discover targets than it is to obscure them.
- **Scalable**: Should be able to search for millions of files at once.
- **Comprehensible**: operation of tool and interpretation of results must be intuitive.

SECTORSCOPE DEMO

- Target: the 3 files shown above.
- Media: a modified disk image from the m57 patents scenario (exFAT).



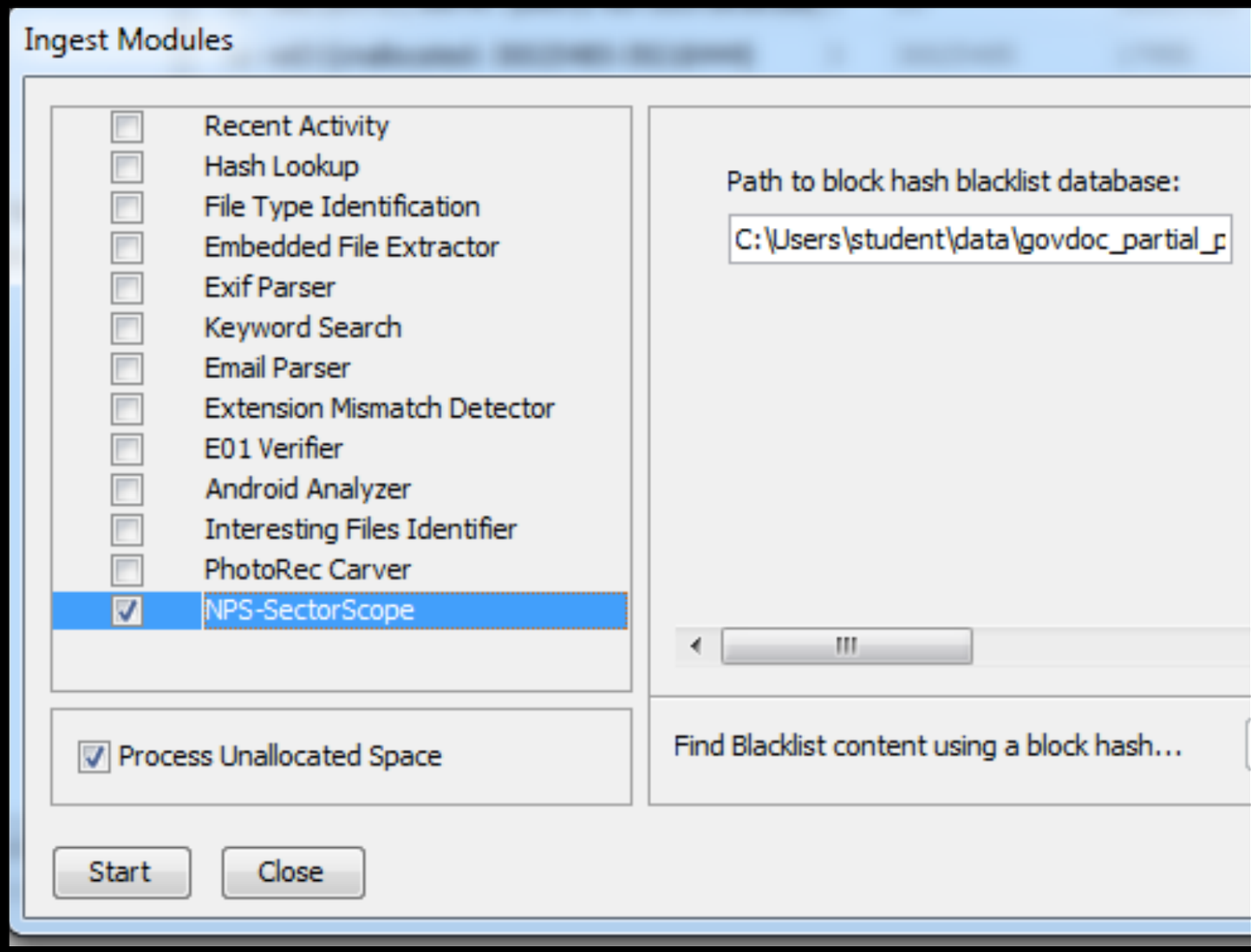
- Files were added to a subfolder on the image then deleted.
- Subfolder was deleted then replaced so it is not trivially easy to recover through Autopsy / TSK.

STEP 1: CREATE A BLACKLIST

- We store our blacklist in hashdb: a fast, lightweight key-value store we designed for this project.
- Blacklist files are hashed in 512-byte chunks and imported into the database using the hashdb bulk_extractor scanner.
- At the moment, this requires a command line.

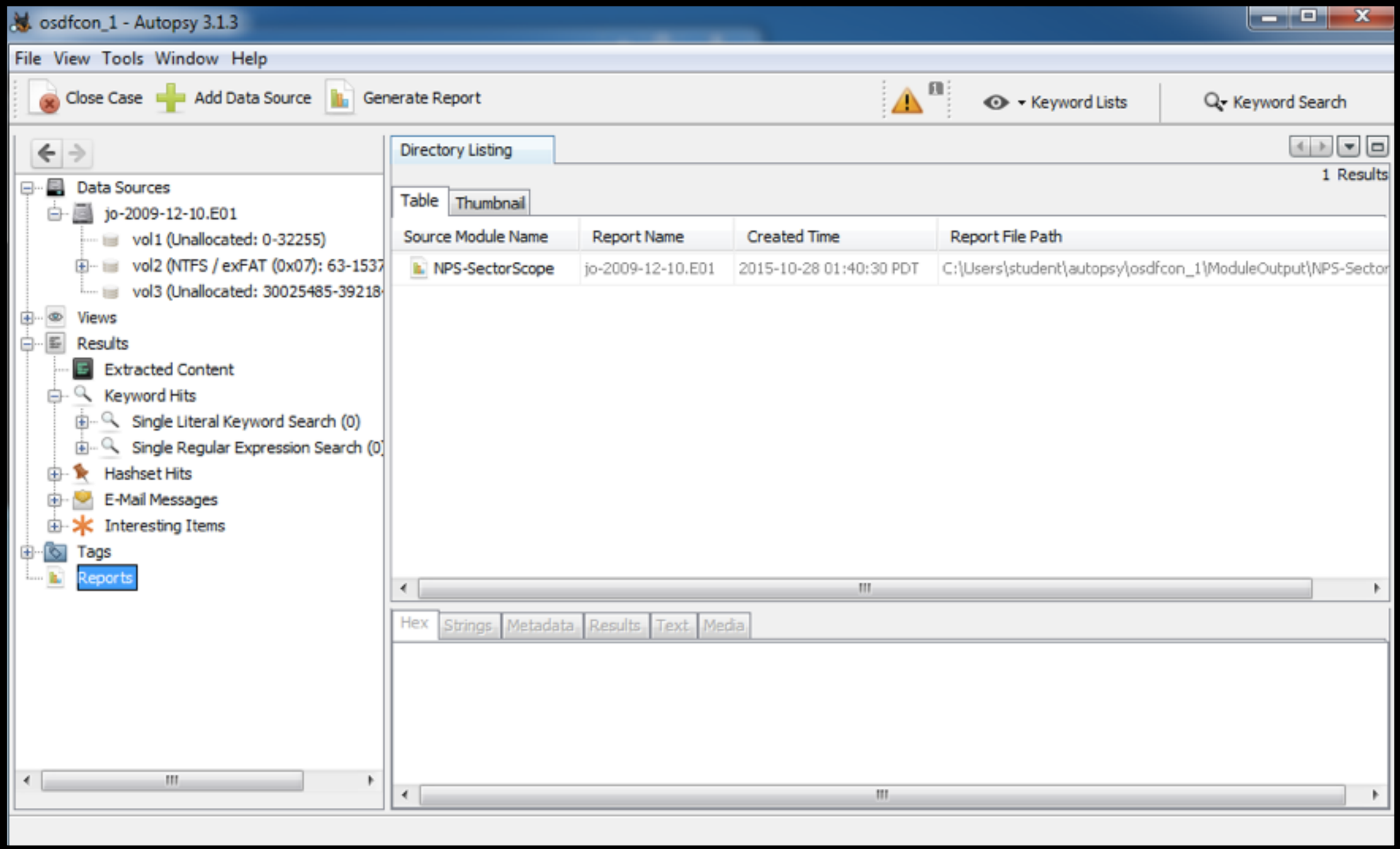
STEP 2: SCAN MEDIA

- Could do this at the command line…
- Let's use Autopsy.
- Install SectorScope and the SectorScope plugin.
- Add your disk image to the case.
- Run SectorScope ingest module.

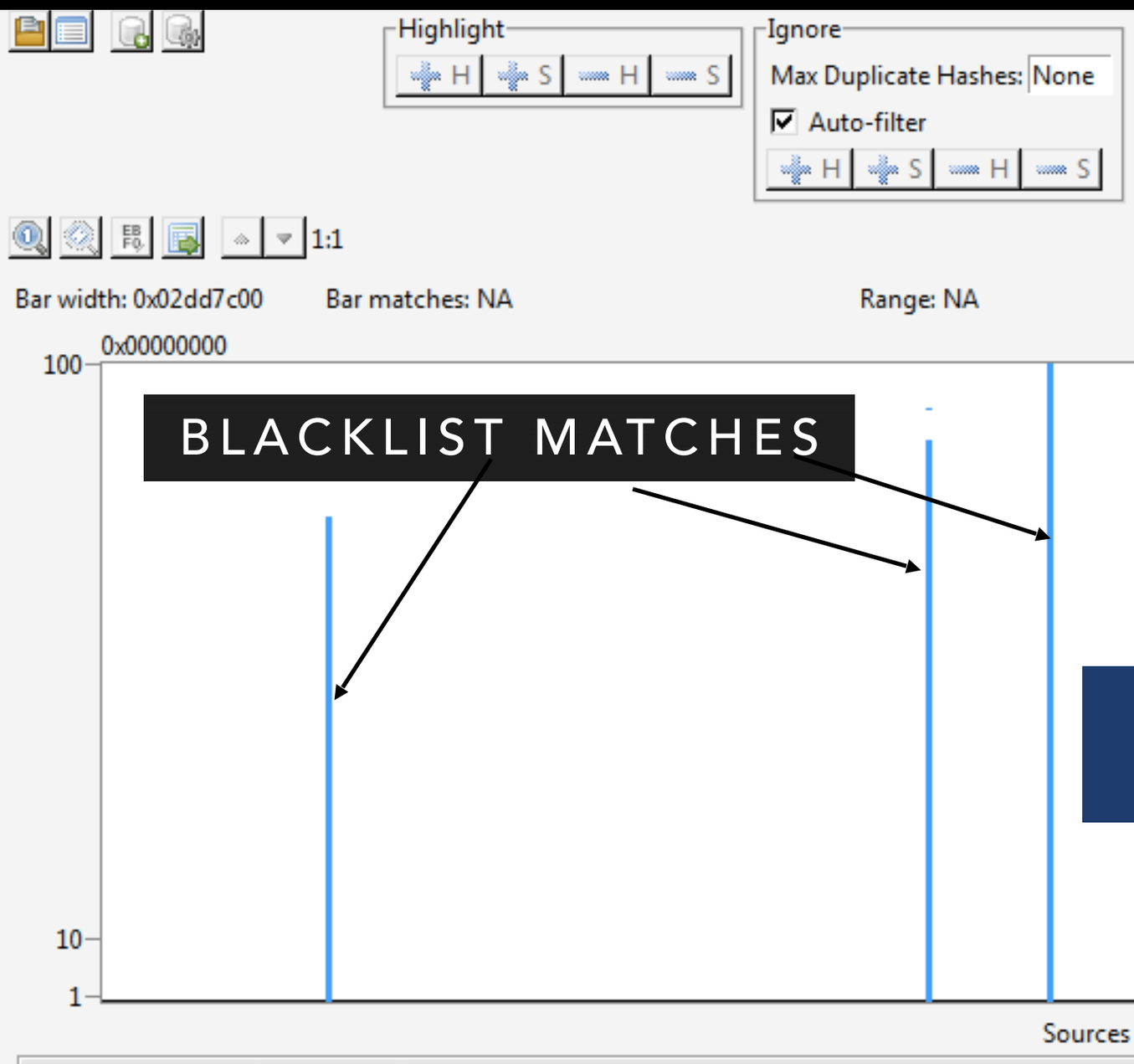


STEP 3: ANALYZE THE RESULTS.

Open SectorScope from the Reports listing:



RESULTS FROM OUR 3 FILE BLACKLIST.



Sources

ID	%Match	#Match	#M(h)	Size	Repository Name	Filename
1	97.9%	2384	0	1.2MiB	blacklist1	blacklist/dfrws_befriend.pptx
2	97.4%	76	0	38.6KiB	blacklist1	blacklist/plan.pdf
3	94.6%	88	0	46.2KiB	blacklist1	blacklist/secret workbook.xlsx

SOURCES FOUND

PAN AND ZOOM TO NAVIGATE THE MEDIA.

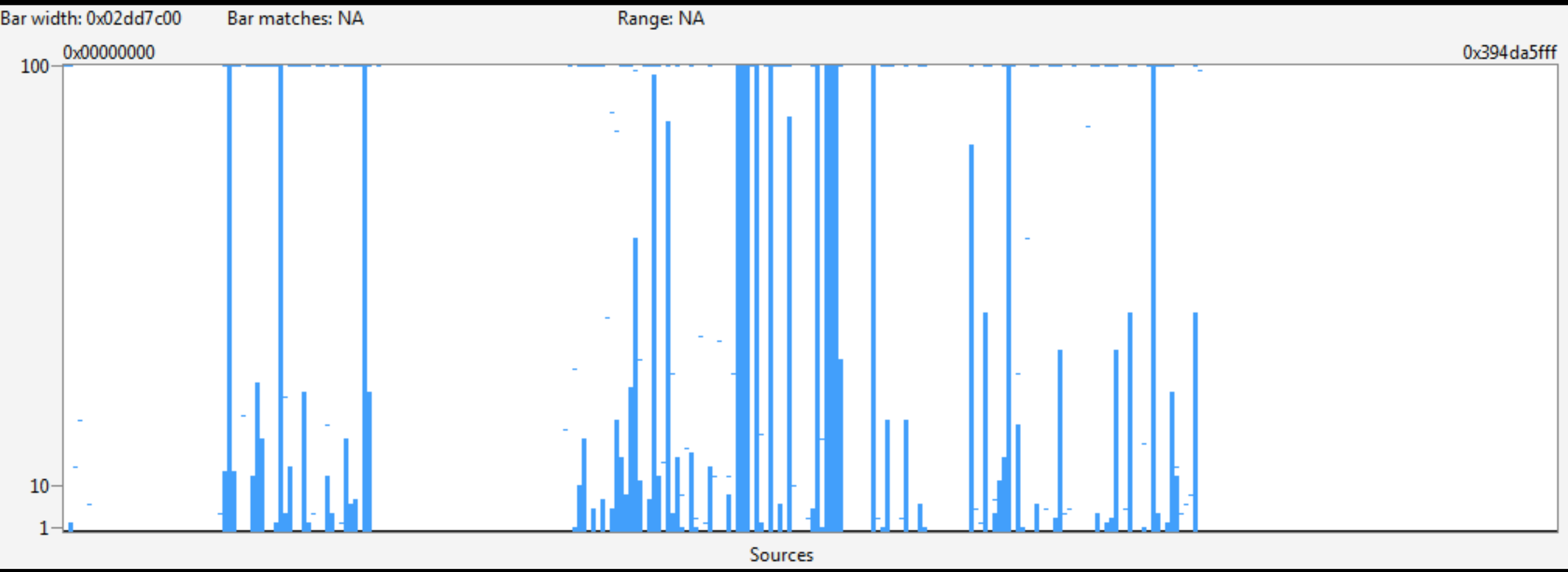
- The list is sorted by percent of file found.
- Highlight a file to see where it is.
- Ignore it to get it off the screen.
- Select a range to see which sources are there.
- Adjust the resolution.
- Open a hex-dump of the data.

LET'S LOOK AT A NON-TRIVIAL EXAMPLE.

THE SCAN RUNS BULK_EXTRACTOR'S HASHDB MODULE TO QUERY THE BLACKLIST.

- Scanning took 109 seconds to scan against 20,772,713 hashes
- Lenovo Thinkpad with 8G of RAM, and a Dual Core i7-5600U CPU (2.6 GHz)
- You can beat this speed with faster hardware
- Compare to 99 seconds to scan against 2,607 hashes: a factor of a thousand takes 10 seconds longer.

THINGS GET MORE COMPLICATED WITH 20 MILLION HASHES.

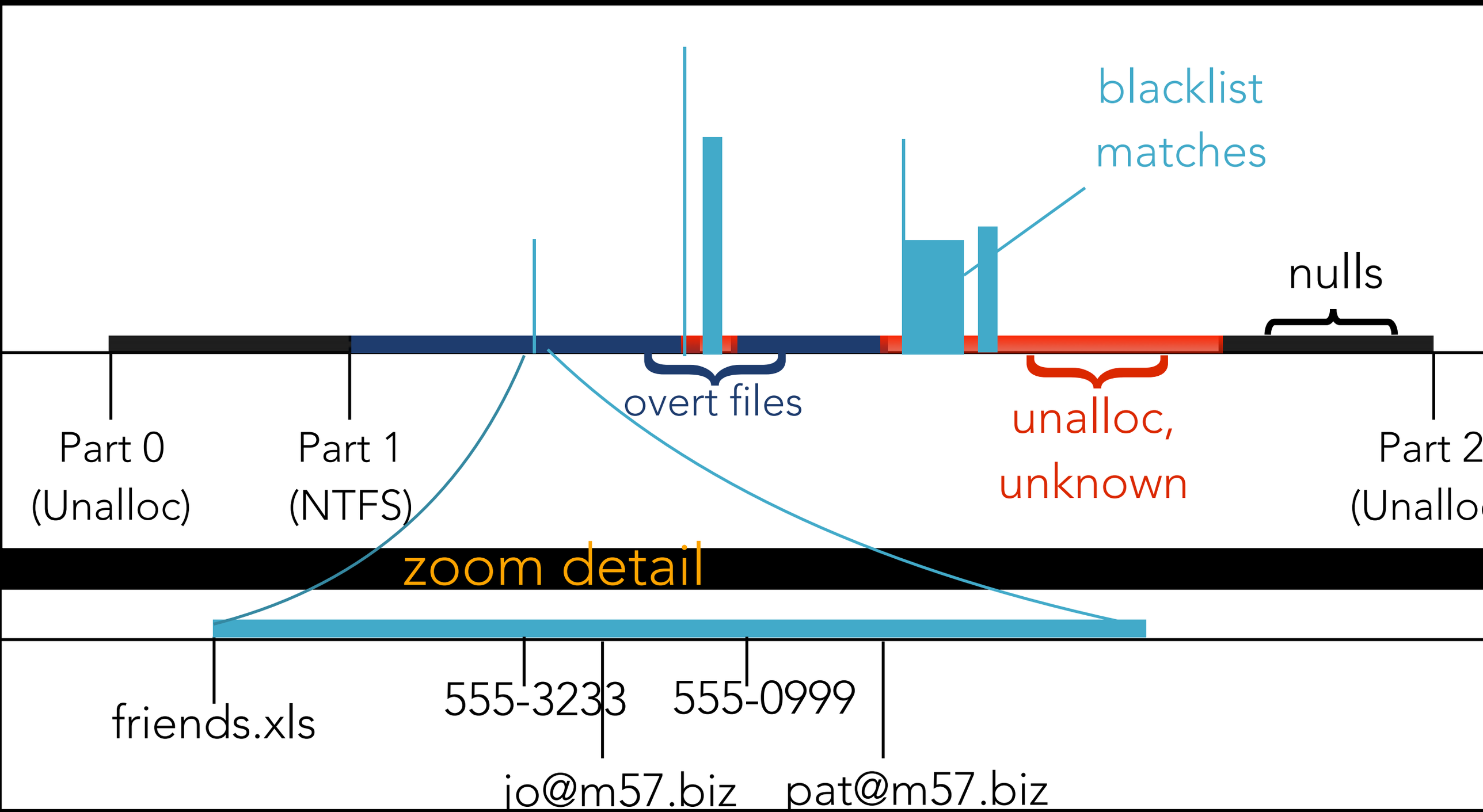


- Added from govdocs—not all high entropy.
- Need to use noise reduction to filter out false positives.

NOISE REDUCTION IS TRICKY

- Down at the level of a 512-byte block, it's not really clear what a "match" tells us.
- Some blocks are extremely rare. (2^{4096} is a lot of possibilities).
- Some blocks show up all over the place.
- We use an algorithm published in a recent DFRWS paper:
  Garfinkel, S. L., & McCarrin, M. (2015). Hash-based carving: Searching media for complete files and file fragments with sector hashing and hashdb. Digital Investigation, 14, S95-S105.

FUTURE WORK: ANNOTATE WITH TSK AND BULK_EXTRACTOR DATA



YOU NEED TO KNOW WHAT YOU DON'T KNOW.

- Few forensics tools will tell you this.
- How much of the image contains data that you know nothing about at all?
- Our end goal is to bring available tools together to map out every sector.

Questions?

Contact:
Michael McCarrin
mrmccarr@nps.edu

Bruce Allen
bdallen@nps.edu

SectorScope on GitHub:
https://github.com/NPS-DEEP/NPS-SectorScope