Digital Forensics File Carving Advances

Team:
Jay Smith KoreLogic
Klayton Monroe KoreLogic
Andy Bair MITRE

Version 1.1 (October 2006)


Agenda

- **Introduction to Digital File Carving**
- 2006 File Carving Challenge
- Methodology
- Conclusion

Computer Forensics Overview

- What Can Effective Forensics Accomplish?
  - Produce corroborating evidence that puts a person at the keyboard at a specific time
  - Recover deleted data (e.g., files, images, email, etc.)
  - Discovery of when files were modified, created, deleted, etc.
  - What applications were installed, even if they were then uninstalled
  - Web sites a user visited…
- What Forensics Cannot Do…
  - Data recovery is impossible if the media is physically destroyed.
  - If the media is <u>securely</u> overwritten, recovery is, at best, very complicated, and often impossible


File Carving Fundamentals

- Definition – Identifying and recovering files based on analysis of file formats
- File carving is a powerful technique because it can
  - Identify and recover files of interest from raw, deleted, or damaged file system, memory, or swap space data
  - Assist in recovering files and data that may not be accounted for by the operating system and file system
  - Assist in simple data recovery


File Carving Details

- Many file types have well-known values or magic(5) numbers in the first bytes of the file header
- Most file carvers
  - Identify specific types of file headers and/or footers
  - Carve out blocks between these two boundaries
  - Stop carving after a user-specified or set limit has been reached
- Unfortunately, not all file types have a standard footer signature, so determining the end can be difficult -- thus the need for limits

File Carving Example

- JPEG files start with *0xffd8* and end with *0xffd9*
- To recover a JPEG file:
  - Find the locations of its header and footer
  - And carve everything between those two endpoints (inclusive)

Hexdump of sample.jpg

```
ff d8 ff e0 00 10 4a 46  49 46 00 01 01 01 00 50  |......JFIF.....P|
... Data ...
28 a2 80 3f ff d9                                 |(..?..|
```

Computer Forensics Challenges

- In general, many more forensic cases today
- Investigations can be lengthy
  - Machines tied up for days during investigations
  - Forensic targets with GB or TB of storage.
  - Still need rapid turnaround, especially in time-sensitive cases involving potential loss of life or property -- think terrorists
- File Carving Challenges
  - One problem faced by forensic practitioners is that existing file carving tools typically produce many false positives and can miss key evidence.
  - Need file carving algorithms that identify more files and reduce the number of false positives

Agenda

- Introduction to Digital File Carving
- **2006 File Carving Challenge**
- Methodology
- Conclusion

DFRWS

- Digital Forensic Research Workshop
  - `http://www.dfrws.org`
- Initiated in 2001
- Objectives (paraphrased)
  - Identify & create processes for applying scientific method in forensics
  - Develop research focused on practitioner
  - Presentation of evidence that meets heightened scrutiny of the courts
- Workshop held annually in various US cities
- Issue forensic challenges leading up to workshop


File Carving Challenge Data

- 50 MB raw file
  - No file system
  - JPEG, ZIP, HTML, Text, & MS Office files & fragments

    dfrws-2006-challenge.zip (ZIP of raw file, 41 MB)
    dfrws-2006-challenge.raw.gz (gzip of raw file, 41 MB)
    MD5 of raw file: bd09d612fc8b3f92662b98f9456f2ada

- Extract as many full files as possible
- Develop tools to solve challenge
  - All source code must be released

Team Goals

- Design and develop file carving algorithms to
  - Identify more files
  - Reduce number of false positives
- Discover more about the current state of file carving
- Contribute lessons learned to workshop & opensource
- Use existing tools from The FTimes Project as our base
  - Extend as needed
- Determine effectiveness of FTimes in dig mode to
  - Identify and enumerate well-known SOFs & EOFs
  - Identify & enumerate file structures or landmarks

Team Environment

- Developed tools and techniques on:
  - FreeBSD 6.[01]
  - Slackware Linux 10.2.0
- Final results produced on FreeBSD 6.0
  - Note: Should be able to reproduce results on any other UNIX system but technical difficulties may arise

Analysis Tools

- Primary OS-native tools:
  - bc (calculations, hex/decimal conversions)
  - dd (data carving and general manipulation)
  - file (data typing)
  - hexdump (data viewing)
  - perl (scripting)
  - sh (scripting)
- Secondary OS-Native tools:
  - gcc (C programming)
  - md5 (or md5sum)


Analysis Tools (Cont.)

- Primary add-on tools, libraries, and modules:
  - FTimes-3.7.0 (mapping, digging, XMagic, and carving)
  - bvi-1.3.1 (data viewing and occasionally editing)
  - foremost-1.1 (benchmark and 2nd opinion)
  - ole-dump (OLE verifier)
  - scalpel-1.54 (benchmark and 2nd opinion)
  - tidy (HTML verifier)
  - unzip552 (ZIP verifier and general extraction tool)
  - xv-3.10a (image viewer)
  - Microsoft Office (document viewer)
- Secondary add-on tools, libraries, and modules:
  - Digest-1.10 (MD5)
  - Digest-SHA1-2.10 (SHA1)
  - Image-TestJPG-0.9 (JPEG verifier)
  - gnuplot-4.0.0 (plotting entropy and averages)
  - mysql-5.0.9-beta (analysis queries based on ftimes output)
  - libOle (contains source for ole-dump)
  - pcre-6.6 (regular expression engine for ftimes)
  - stegdetect-0.5 (potential image verifier)
  - OpenOffice-2.0.3 (document viewer)
  - gqview-2.0.1 (image viewer)
  - WinZip (ZIP verifier and general extraction tool)

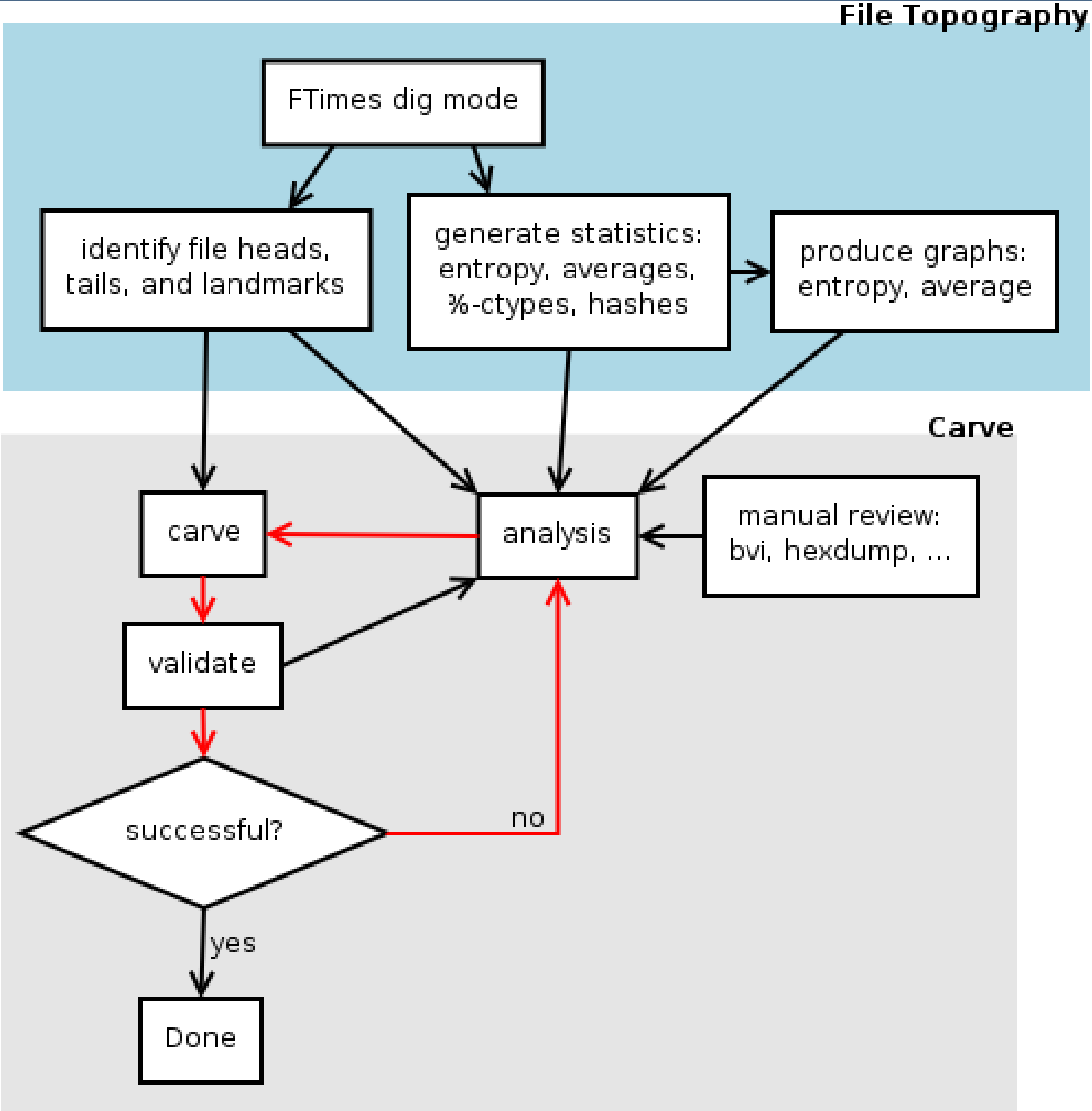
Terminology (1)

- **SOF** - start of file
- **EOF** - end of file
- **FAT** - file allocation table
- **OLE** - object linking and embedding, Microsoft's framework for compound documents
- **XMagic - Extended Magic**; This is a line of Magic that was inspired by the original file(1) Magic. XMagic is part of FTimes.

Terminology (2)

- **Entropy**
  - Measure of randomness
  - Range = 0-8; 8 = most random; 0 = least random
  - Dramatic entropy changes can indicate file boundary
- **Sliding Entropy:**
  - Calculating entropy for each sequential file data block
- **Sliding Average:**
  - Calculating average for each sequential file data block
- **Sliding Hash (MD5 and SHA1)**
  - Calc message digests for each sequential file data blk
  - Bashed against 1+ subject images
  - Can use to locate duplicate blocks

Team Methodology



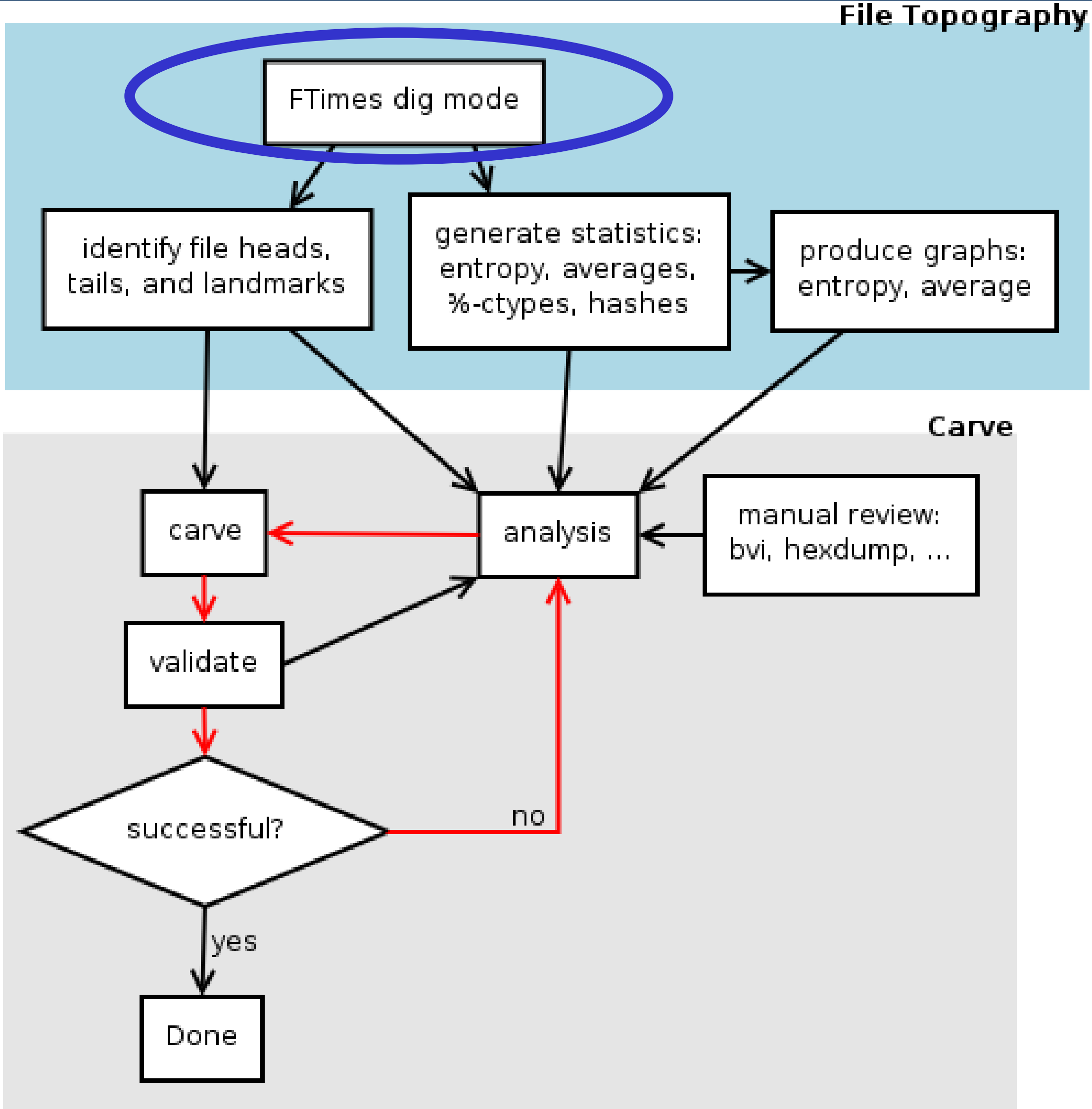
Hypotheses Used to Create Methodology

- Application-specific parsers better than custom
  - Use existing tools and libraries as validators
- Legitimate files will start on sector boundary
  - Non-sector aligned files likely to be embedded
- Blocks of one file encompassed by another file
  - Slack space, entropy tests, and byte distribution may help reveal edges
- Carve most well-defined file types first
  - Use boundary info as SOF/EOF edges for other file types

Agenda

- Introduction to Digital File Carving
- 2006 File Carving Challenge
- **Methodology**
- Conclusion

Methodology



FTimes Overview

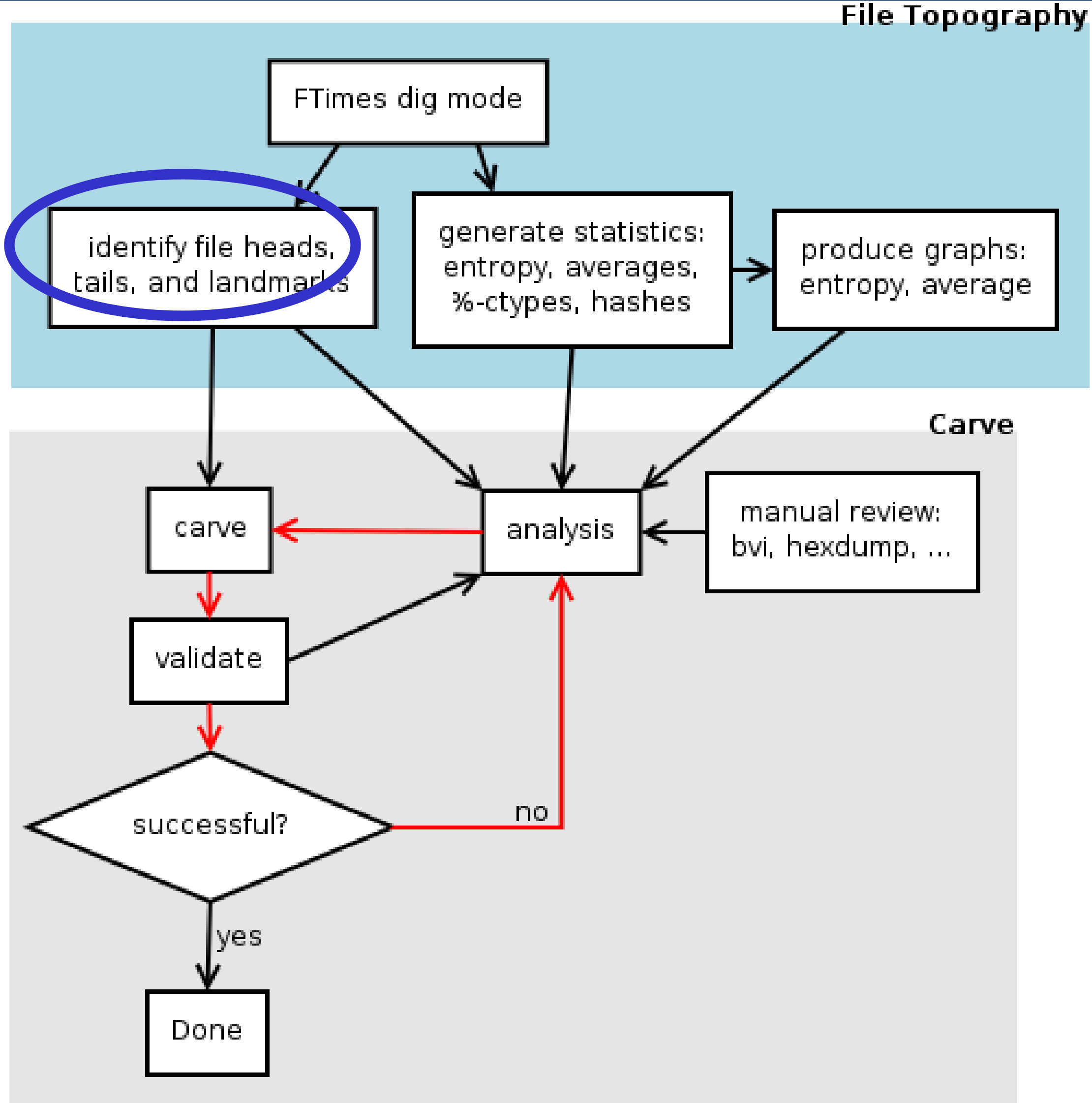
http://ftimes.sourceforge.net/FTimes/index.shtml

- System baselining and evidence collection tool
- Gather/develop topographical information & attributes about directories and files in a manner conducive to intrusion and forensic analysis
- Lightweight: small footprint, command line interface
- Used dig (“search”) mode in conjunction with XMagic to develop topography

FTimes Dig Mode

- Search through directories and files looking for user-specified regular expressions or sequence(s) of bytes
- 3 tiers of searching
  - Basic – DigStringNormal, DigStringNoCase
  - Advanced – DigStringRegExp
  - Expert – DigStringXMagic

Methodology



FTimes – Identify File Heads and Tails

JPEG with 2 thumbnails:

combined.cfg

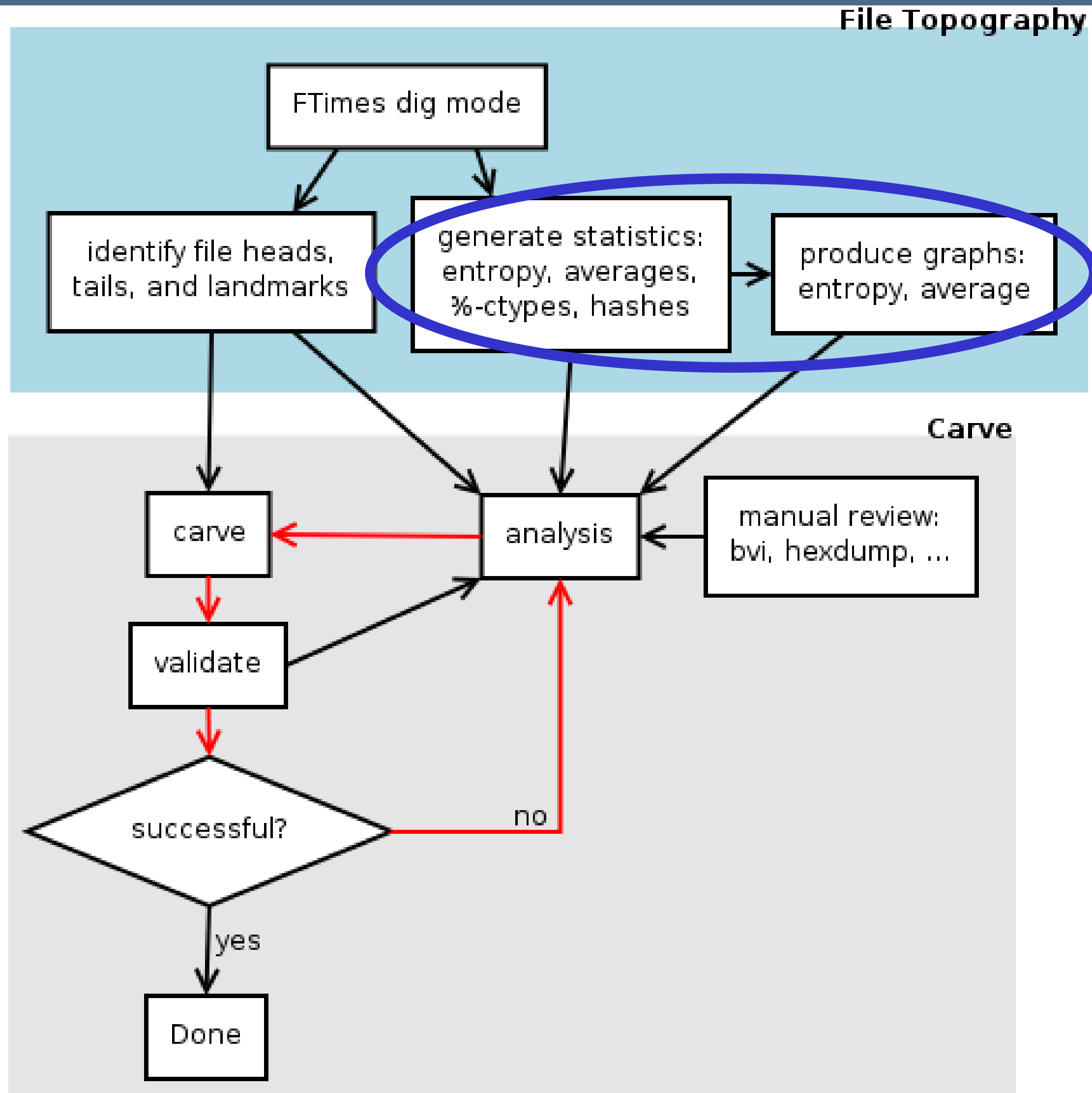
```
DigStringRegExp=(?s)(\xff\xd8....JFIF)    sof.jpeg
DigStringNormal=%ff%d9                    eof.jpeg
```

```
ftimes -diglean combined.cfg challenge.raw
```

combined.dig

```
"challenge.raw"|regexp|sof.jpeg|1980416|%ff%d8%ff%e0%00%10JFIF
"challenge.raw"|regexp|sof.jpeg|1980748|%ff%d8%ff%e0%00%10JFIF
"challenge.raw"|normal|eof.jpeg|1986297|%ff%d9
"challenge.raw"|regexp|sof.jpeg|1995443|%ff%d8%ff%e0%00%10JFIF
"challenge.raw"|normal|eof.jpeg|2000992|%ff%d9
"challenge.raw"|normal|eof.jpeg|2267600|%ff%d9
```

Methodology



FTimes
XMagic Introduction

http://ftimes.sourceforge.net/FTimes/XMagic.shtml

- Used XMagic to develop statistics (entropy, averages, %-ctypes, ...)
- To understand XMagic, requires knowledge of the file(1) command and magic(5)
- Magic number – special constant (traditionally) used to identify a particular type of file (e.g., tcpdump magic is 0xa1b2c3d4)
- file(1) command – determines file types using magic numbers
- Typical file(1) command usage:

```
$ file ftimes.zip
  ftimes.zip: Zip archive data, at least v2.0 to extract
```

File and magic example

```
$ file ftimes.zip
```

```
magic:
0     string    PK\003\004     Zip archive data
>4    byte      0x09           \b, at least v0.9 to extract
>4    byte      0x0a           \b, at least v1.0 to extract
>4    byte      0x0b           \b, at least v1.1 to extract
>4    byte      0x14           \b, at least v2.0 to extract
```

```
ftimes.zip:
0  1  2  3  4  5  6  7
50 4b 03 04 14 00 00 00   |PK......|
```

```
ftimes.zip: Zip archive data, at least v2.0 to extract
```

XMagic vs. Magic (1)

- Split operator/value pair into separate fields
- Supports
  - Regular expression Magic via Perl Compatible Regular Expressions (PCRE)
  - Block-based entropy calculations
  - Block-based average calculations
  - Block-based percent calculations for ctype(3) character classes
  - Block-based hash calculations (MD5 and SHA1)
  - Several different test operators for all of its block-based tests

XMagic vs. Magic (2)

- Test operator/value (if test operator absent in Magic, implied operator is '=‘)

```
Magic:    0    string          \037\235          compress'd data
XMagic:   0    string       =  \037\235          compress'd data
```

- Place holder when the test value is to be ignored:

```
Magic:    >6   byte       x                      type %c
XMagic:   >6   byte       x     -                type %c
```

- Convert a series of string/[Bbc] tests to the equivalent regexp test:

```
Magic: 0 string/B =  \=pod\n                   Perl POD document
Magic: 0 string/B =  \n\=pod\n                 Perl POD document
Magic: 0 string/B =  \=head1\                  Perl POD document
Magic: 0 string/B =  \n\=head1\                Perl POD document
Magic: 0 string/B =  \=head2\                  Perl POD document
Magic: 0 string/B =  \n\=head2\                Perl POD document
XMagic:0 regexp   =~ ^\n?=(?:pod\n|head[12]) Perl POD document
```

XMagic vs. Magic (3)

- Convert a search/<number> test to an equivalent regexp:<number> test

```
Magic:   0   search/20  =  foo  The venerable %s document
XMagic:  0   regexp:20  =~ foo  The venerable %s document
```

- Block-based test types to harvest various topographical information:

```
XMagic:   0    byte                         x  -  512
XMagic:   >&0  row_entropy_1:512            x  -  \b|%f
XMagic:   >&0  row_average_1:512            x  -  \b|%f
XMagic:   >&0  percent_ctype_alnum:512      x  -  \b|%f
XMagic:   >&0  sha1:512                     x  -  \b|%s
```

XMagic and Challenge entropy, %-ctype, ...

stats-512.cfg.xmagic

```
# XMagic
0     byte                   x   -   512
>&0   row_entropy_1:512      x   -   \b|%f
>&0   row_entropy_2:512      x   -   \b|%f
...
```

stats-512.cfg

```
Basename=-
DigStringXMagic=stats-512.cfg.xmagic stats-512
...
```

```
ftimes -diglean stats-512.cfg challenge.raw
```

```
name|type|tag|offset|string
"challenge.raw"|xmagic|stats-512|0|512|4.656387|7.282739|...
"challenge.raw"|xmagic|stats-512|512|512|4.667385|7.244524|...
...
```

Compute and Plot Sliding Entropy/Average Statistics

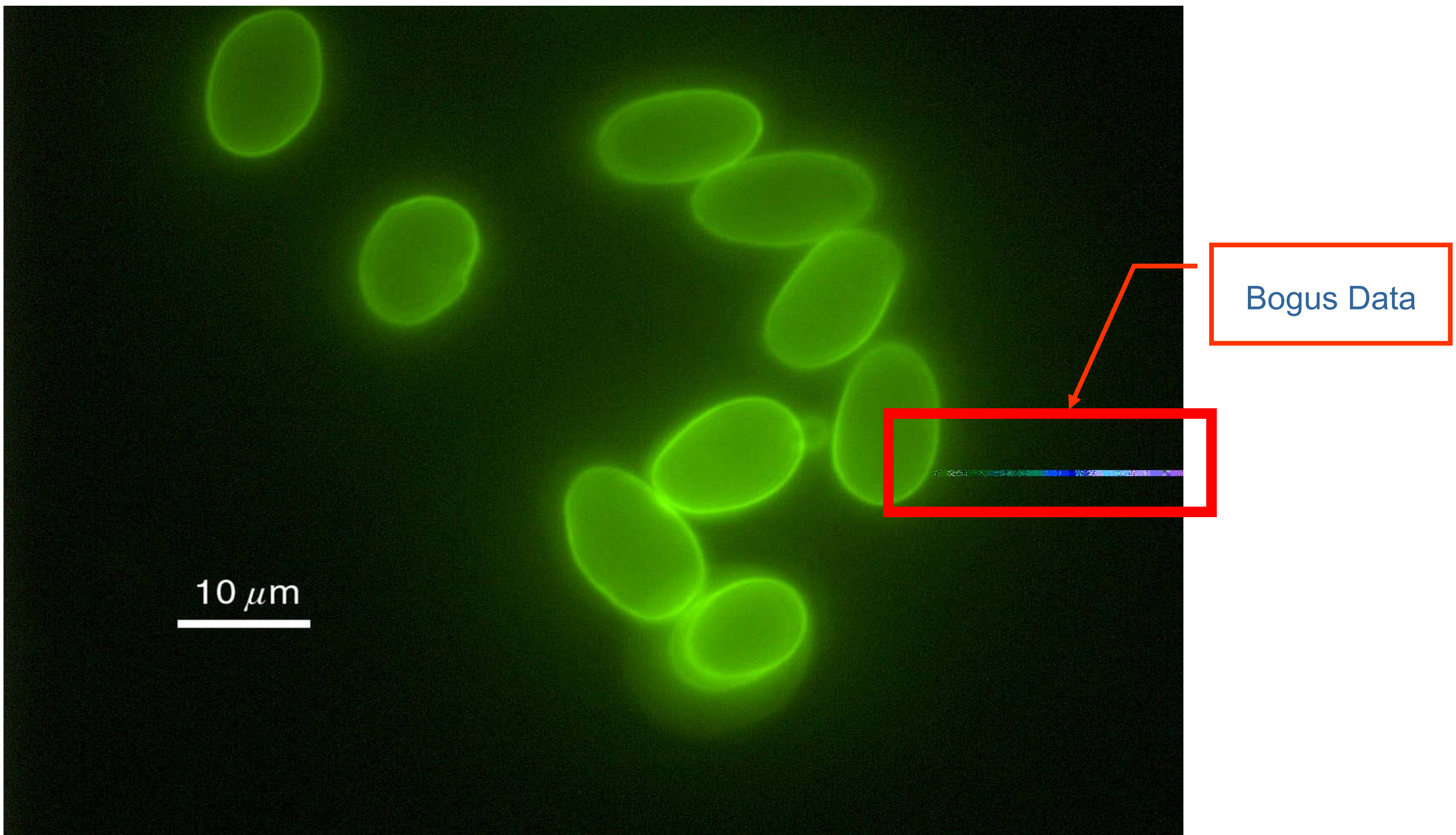
- Sliding entropy & average good for detecting data stream edges
  - Typically occurs on block boundary
- Sliding entropy can be used to classify different data types:
  - Entropy 4-6: TEXT- and HTML-based blocks
  - Entropy 7-8: ZIP- and JPEG-based
- Used FTimes + XMagic to collect stats and topographical info:
  - Compute sliding entropy & average values over subject image
  - Plot entropy and average values

Using Entropy and do_itrim

Example of Extracting a JPEG Image

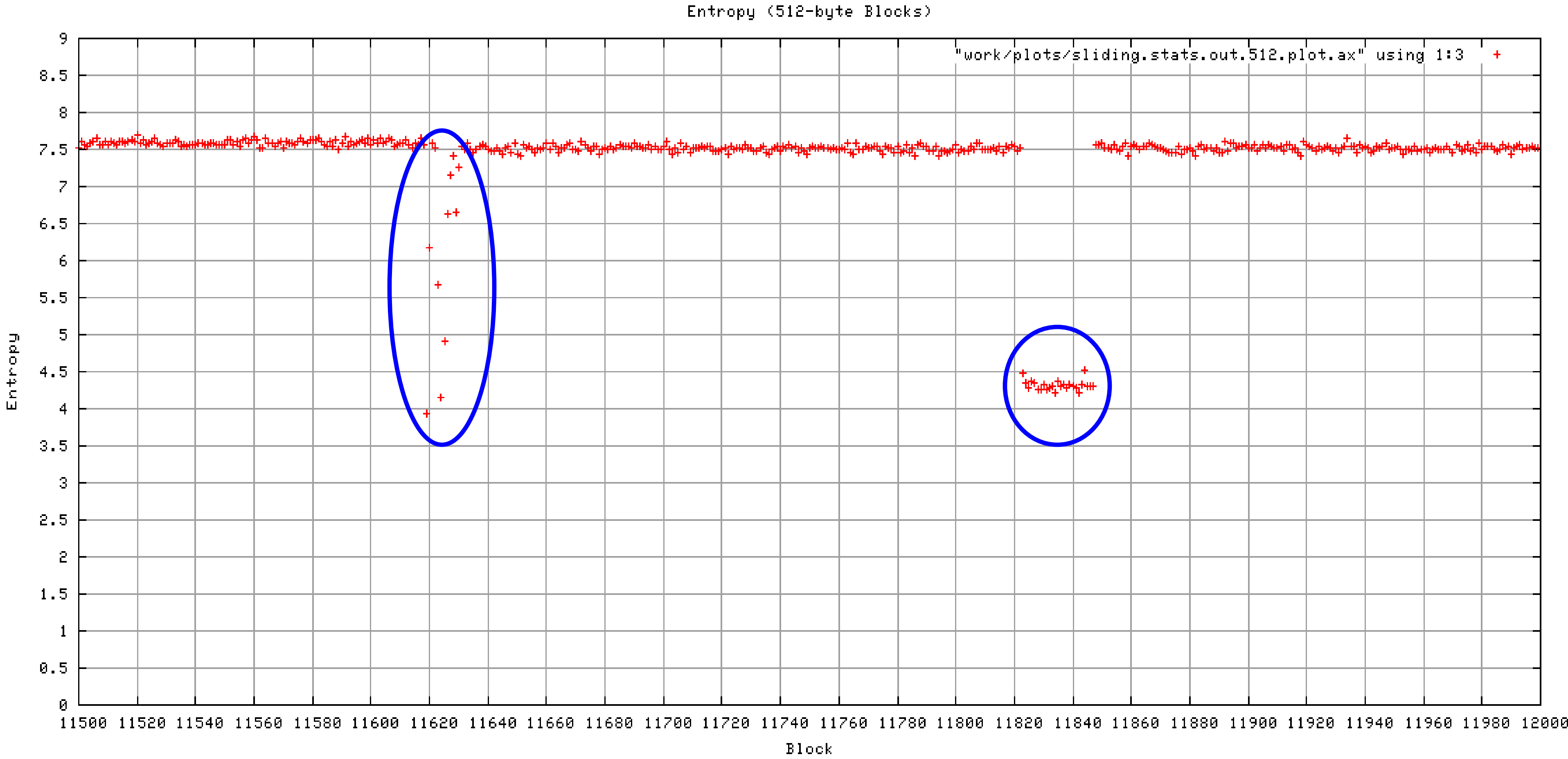
Stage 1 Carve

We used sliding entropy graphs to see if we could determine the portion to trim out using do_itrim. Notice the portion on the right that seems out of place.



Using Entropy and do_itrim (Cont.)

This sliding entropy graph shows the start of the JEPG image at block 11619. The graph also reveals a drop in entropy at block 11820.



Using Entropy and do_itrim (Cont.)

- *do_itrim* command used to extract the bogus data from the stage 1 carve file
- lower bound = 103936 which is close to block containing extra data
- validator script returns 1 if image is valid
- block size = 512 bytes conform to the raw data file block size
- trim size = 13312 is the amount of extra data from entropy graphs

File extension

Lower bound

Validator return code

Block size

```
do_itrim -e jpeg -l 103936 -r 1 -s 512 -f first-cut.jpeg -t 13312
-- tools/test_jpeg.pl -f %subject
```

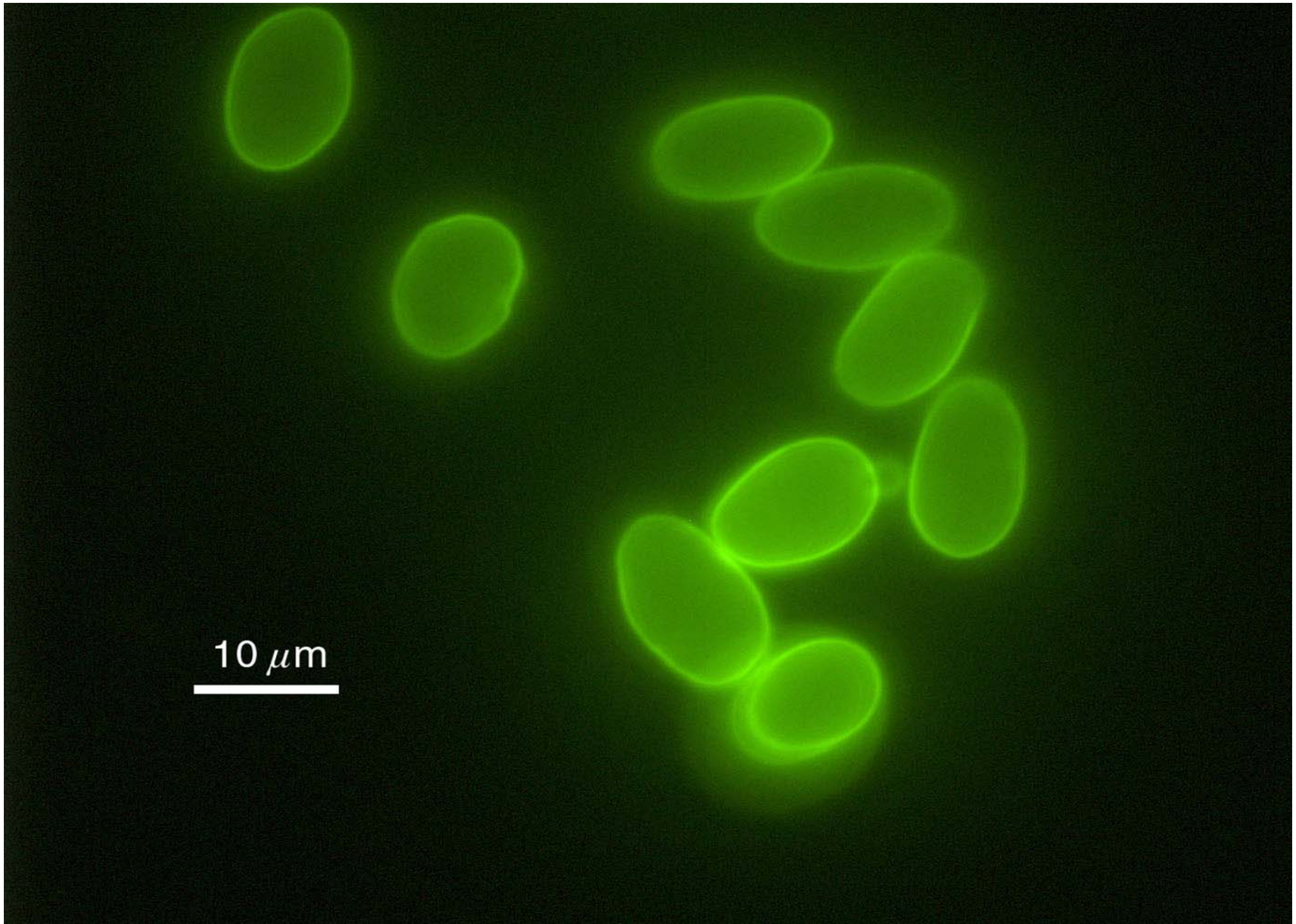
Validator program

Validator Arguments

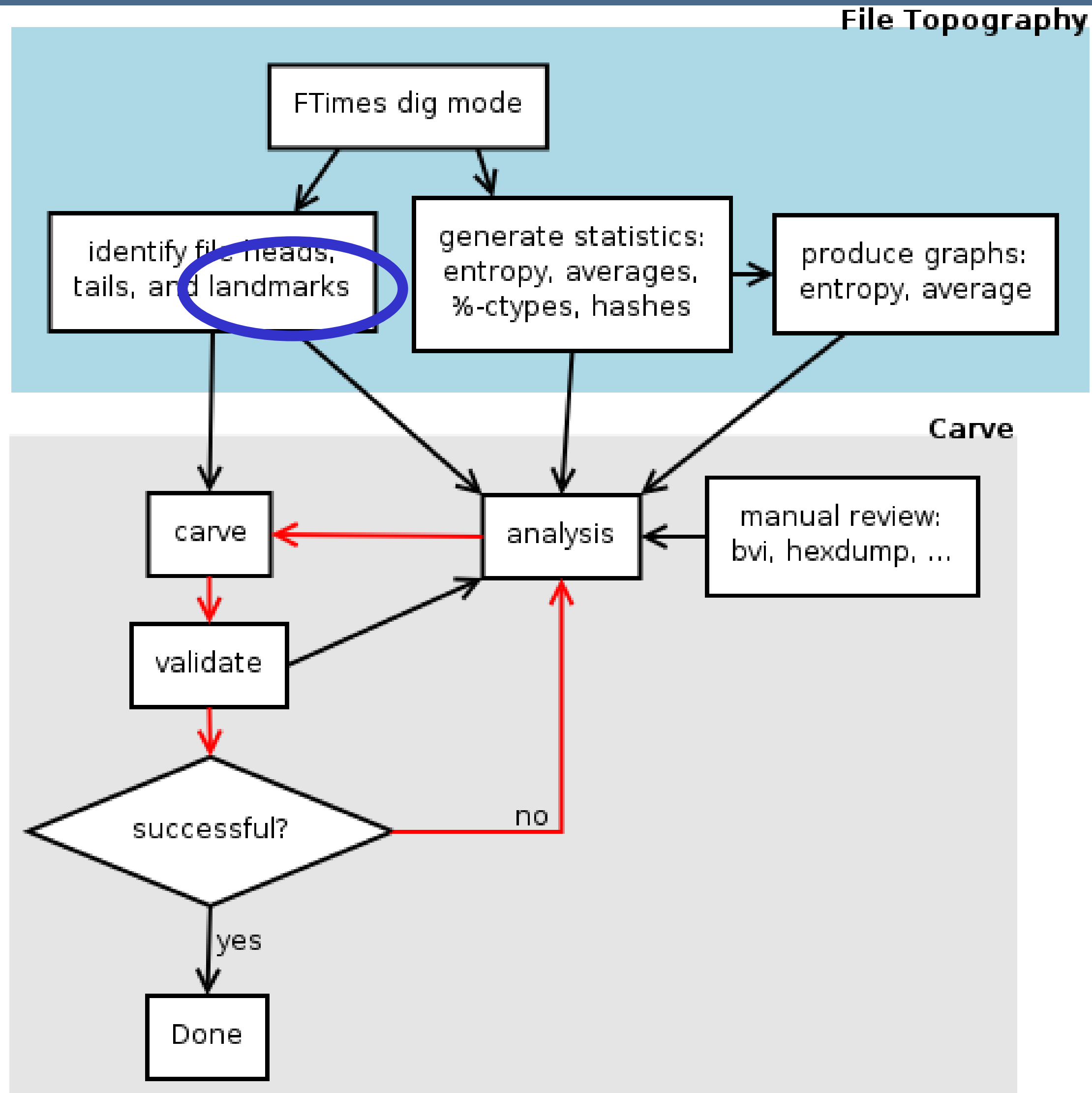
Trim size

Using Entropy and do_itrim (Cont.)

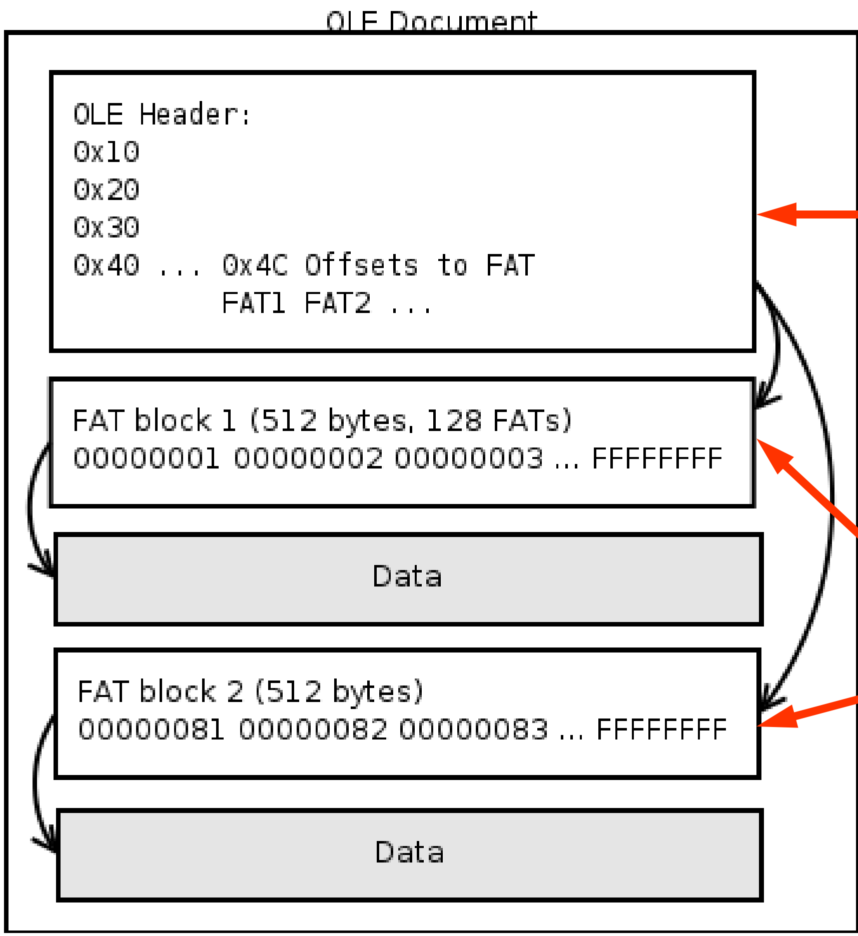
We used do_itrim to carve out the section where the entropy dropped. The result is a verified and complete image (as shown below).



Methodology



XMagic: OLE Documents



xmagic.ole.enumerate-header-fat

```
# XMagic0
regexp:512  =~ ... edited ...
>&64    regexp:512  =~  (?s)(.{12})
>>&12   lelong  != 0xffffffff  \b
>>>&0   lelong  x - \b%X
>>>>&4  lelong  != 0xffffffff  \b,%X
...
```

xmagic.ole.enumerate-fat-blocks

```
# XMagic0
...
```

XMagic: enumerate file struct

combined.cfg

```
DigStringXMagic=xmagic.ole.enumerate-header-fat   sof.ole
DigStringXMagic=xmagic.ole.enumerate-fat-blocks   fat.ole
...
```

```
ftimes -diglean combined.cfg challenge.raw
```

```
"challenge.raw"|xmagic|sof.ole|1050112|689,68A,...
"challenge.raw"|xmagic|fat.ole|1917952|00000001:aaa...
"challenge.raw"|xmagic|fat.ole|1918464|00000081:aaa...
```

```
ole-dig2crv
```

carve.log

```
1050112, blk 2051 -- header FAT block pointers: 689,68A,...
1907200, blk 3725 -- missing FAT block, +10752 byts, +21 blks
1917952, blk 3746 -- valid FAT block #1, 0x689
1918464, blk 3747 -- valid FAT block #2, 0x68A
```

Example of Extracting a Microsoft Document

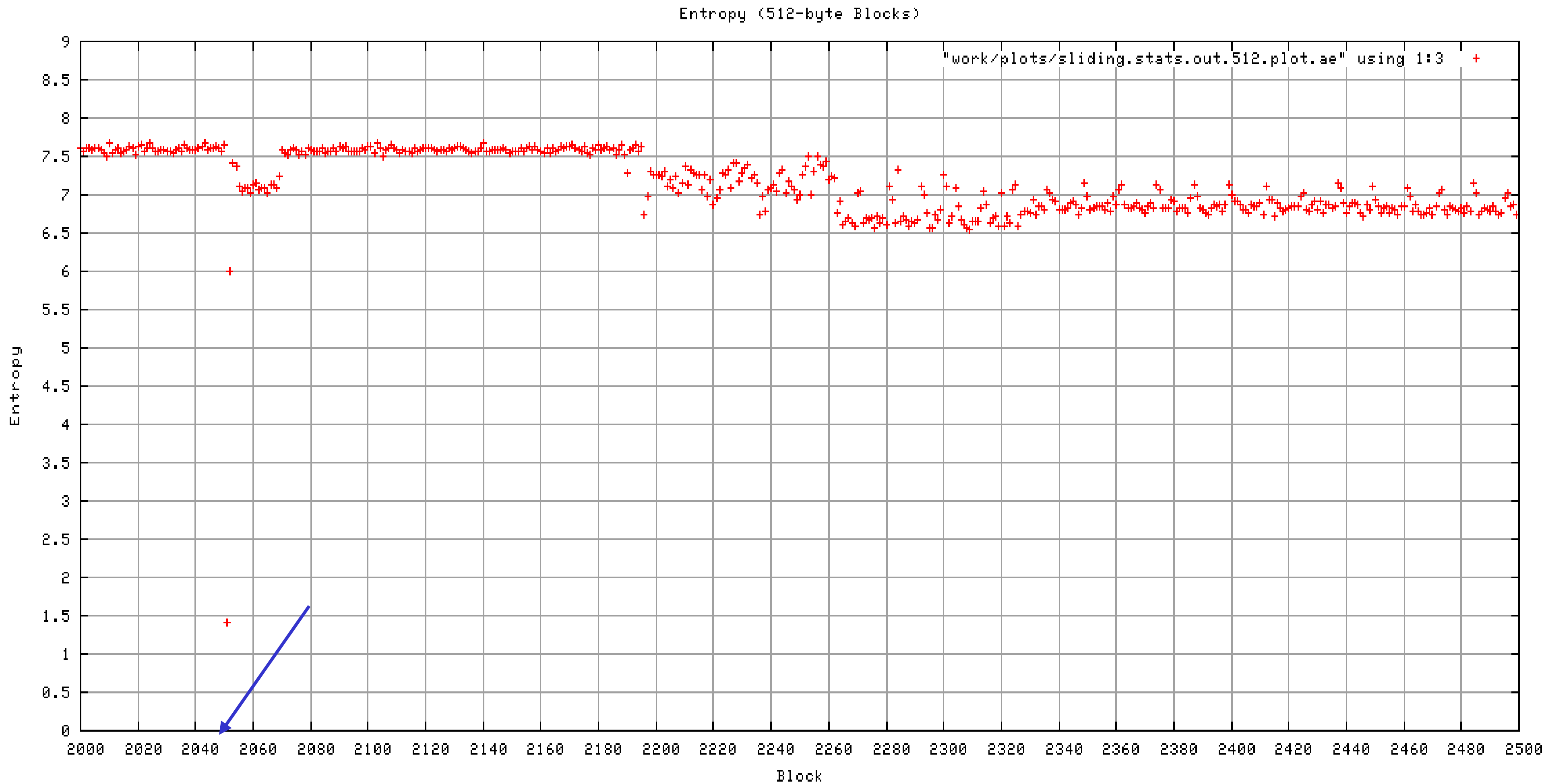
Using Entropy and do_itrim

- This example shows us trying to validate a file carve of a Microsoft Office document.
- The validator program couldn't validate the file and crashed with a segmentation fault.
- We next looked at other data points to figure out why this file did not validate.

```
jay@beast: ole-dump -v dfrws-2006-challenge.raw.1050112.ole
Table of Contents for dfrws-2006-challenge.raw.1050112.ole:
unk1  = 0       unk2  = 0       unk3  = 0       unk4  = 0
unk5  = 3003e   unk6  = 9fffe   unk7  = 6       unk8  = 0
unk9  = 0       unk10 = 0       unk11 = 1000    unk12 = 0
root  = 698
dir flag = fffffffe
# FAT blocks = e
FAT_next_block = fffffffe
# extra FAT blocks = 0
bbd list:
689 68a 68b 68c 68d 68e 68f 690 691 692
693 694 695 699 ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
Segmentation fault
jay@beast:
```

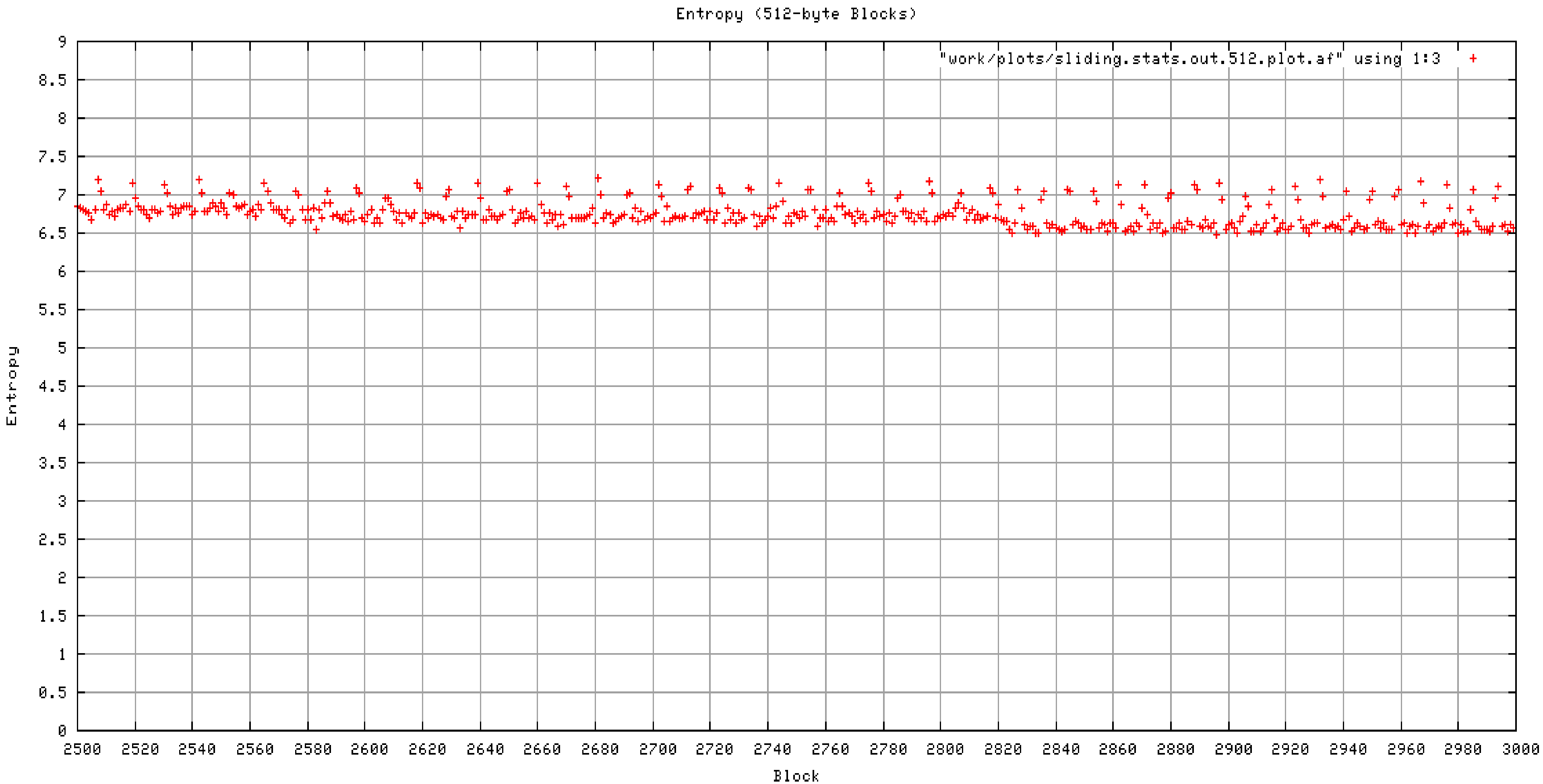
Using Entropy and do_itrim

Here is the first of three entropy graphs for the Microsoft document. We knew from the stage 1 carve that our document began at block 2051.



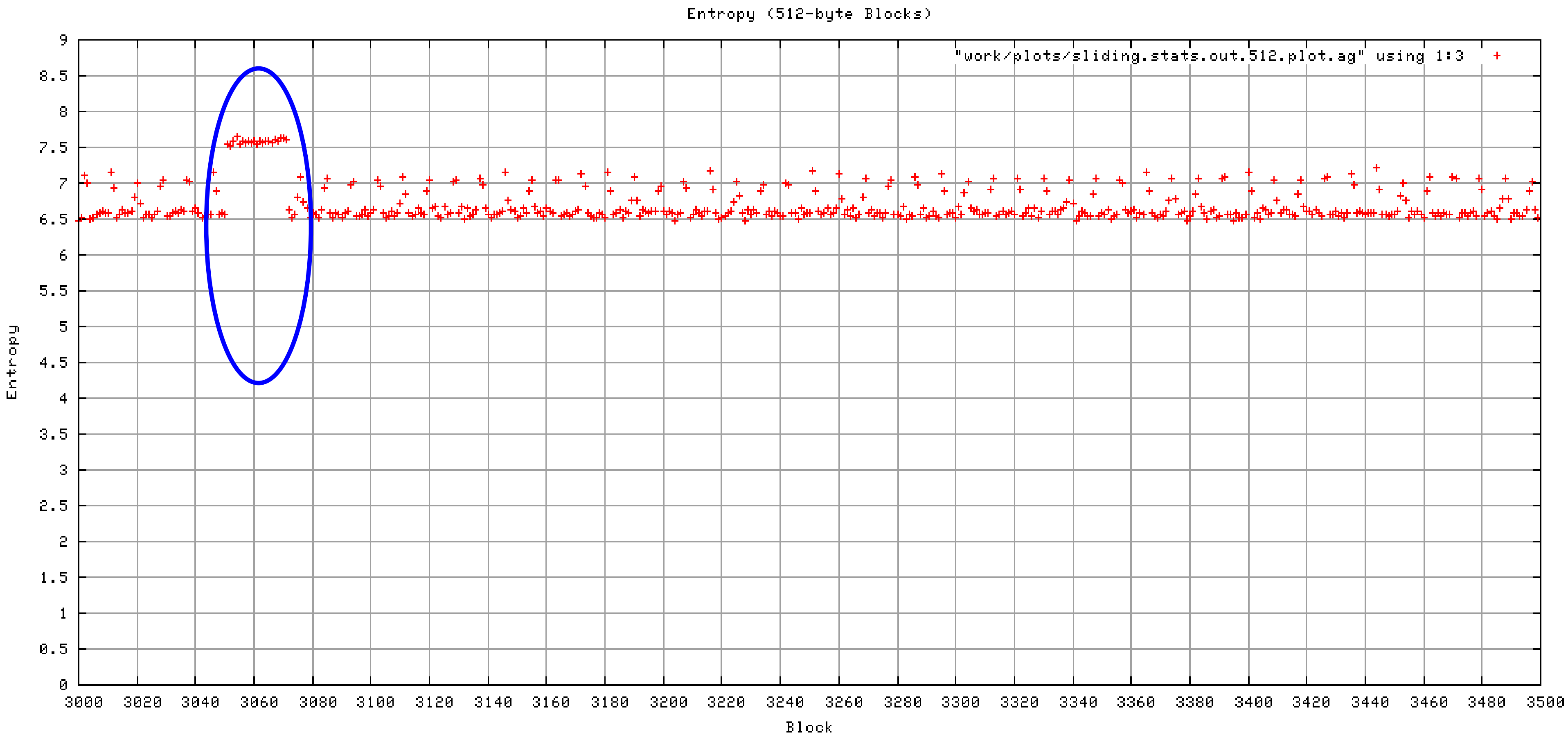
Using Entropy and do_itrim

This entropy graph shows a continuation of the same range of entropy which is a good indication that these blocks are part of the same file.



Using Entropy and do_itrim

In this graph you can see the fluctuation in the entropy starting at block 3051 and ending at block 3072. Our hypotheses was to carve those blocks out to recover the full Microsoft Office document.

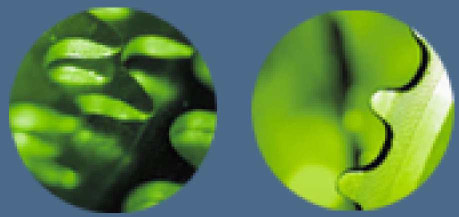


Using Entropy and do_itrim

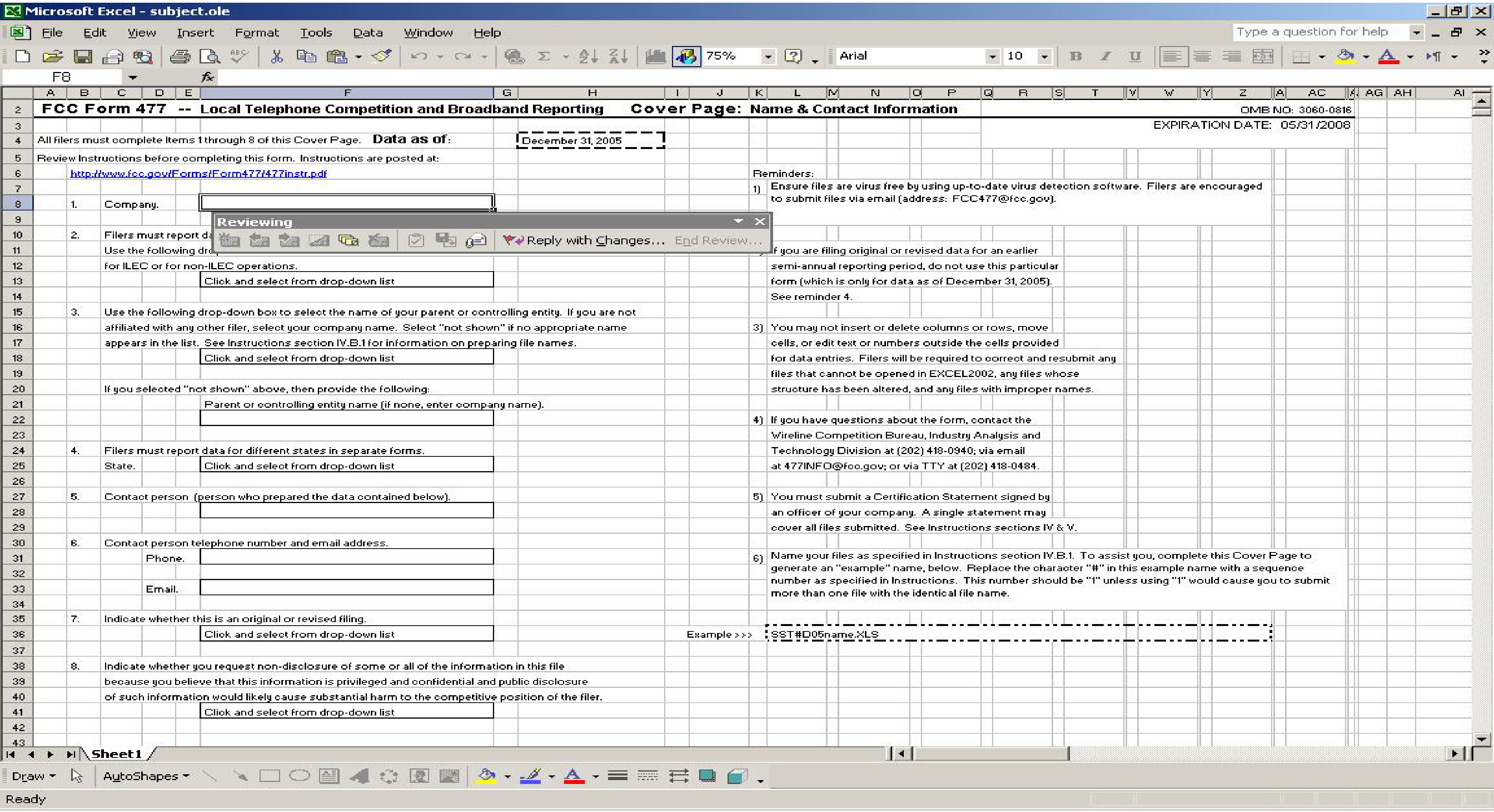
By taking out the section from block 3051 to block 3072 with do_itrim, we carved out the extra data.

```
jay@beast: ./bin/do_itrim -v -l 512000 -e ole -s 512 -f dfrws-2006-challenge.raw.1050112
.ole -t 10752 -- ole-dump %subject
ole-dump ./subject.ole
Table of Contents for ./subject.ole:
Root Entry
Workbook                                                    848333
SummaryInformation                                            4096
DocumentSummaryInformation                                    4096
dfrws-2006-challenge.raw.1050112.ole trimmed 10752 @ 512000 ---> pass
jay@beast: 
```

Using Entropy and do_itrim



Below is a screen shot of the final extracted document.



Microsoft Excel - subject.ole

File Edit View Insert Format Tools Data Window Help

Type a question for help

F8

75% Arial 10

FCC Form 477 -- Local Telephone Competition and Broadband Reporting Cover Page: Name & Contact Information

OMB NO: 3060-0816

EXPIRATION DATE: 05/31/2008

All filers must complete Items 1 through 8 of this Cover Page. **Data as of:** December 31, 2005

Review Instructions before completing this form. Instructions are posted at:

http://www.fcc.gov/Forms/Form477/477instr.pdf

1. Company.

Reviewing — Reply with Changes... End Review...

2. Filers must report d...
Use the following dr...
for ILEC or for non-ILEC operations.
Click and select from drop-down list

3. Use the following drop-down box to select the name of your parent or controlling entity. If you are not affiliated with any other filer, select your company name. Select "not shown" if no appropriate name appears in the list. See Instructions section IV.B.1 for information on preparing file names.
Click and select from drop-down list

If you selected "not shown" above, then provide the following:
Parent or controlling entity name (if none, enter company name).

4. Filers must report data for different states in separate forms.
State. Click and select from drop-down list

5. Contact person (person who prepared the data contained below).

6. Contact person telephone number and email address.
Phone.
Email.

7. Indicate whether this is an original or revised filing.
Click and select from drop-down list

8. Indicate whether you request non-disclosure of some or all of the information in this file because you believe that this information is privileged and confidential and public disclosure of such information would likely cause substantial harm to the competitive position of the filer.
Click and select from drop-down list

Reminders:

1) Ensure files are virus free by using up-to-date virus detection software. Filers are encouraged to submit files via email (address: FCC477@fcc.gov).

...f you are filing original or revised data for an earlier semi-annual reporting period, do not use this particular form (which is only for data as of December 31, 2005). See reminder 4.

3) You may not insert or delete columns or rows, move cells, or edit text or numbers outside the cells provided for data entries. Filers will be required to correct and resubmit any files that cannot be opened in EXCEL2002, any files whose structure has been altered, and any files with improper names.

4) If you have questions about the form, contact the Wireline Competition Bureau, Industry Analysis and Technology Division at (202) 418-0940; via email at 477INFO@fcc.gov; or via TTY at (202) 418-0484.

5) You must submit a Certification Statement signed by an officer of your company. A single statement may cover all files submitted. See Instructions sections IV & V.

6) Name your files as specified in Instructions section IV.B.1. To assist you, complete this Cover Page to generate an "example" name, below. Replace the character "#" in this example name with a sequence number as specified in Instructions. This number should be "1" unless using "1" would cause you to submit more than one file with the identical file name.

Example >>> SST#D05name.XLS

Sheet1

Draw AutoShapes

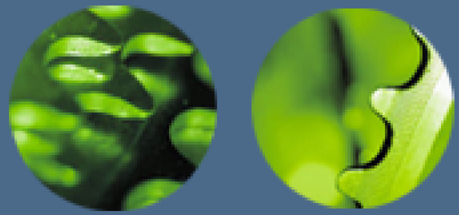
Ready

Example of Extracting a Zip file

Using Entropy and do_itrim

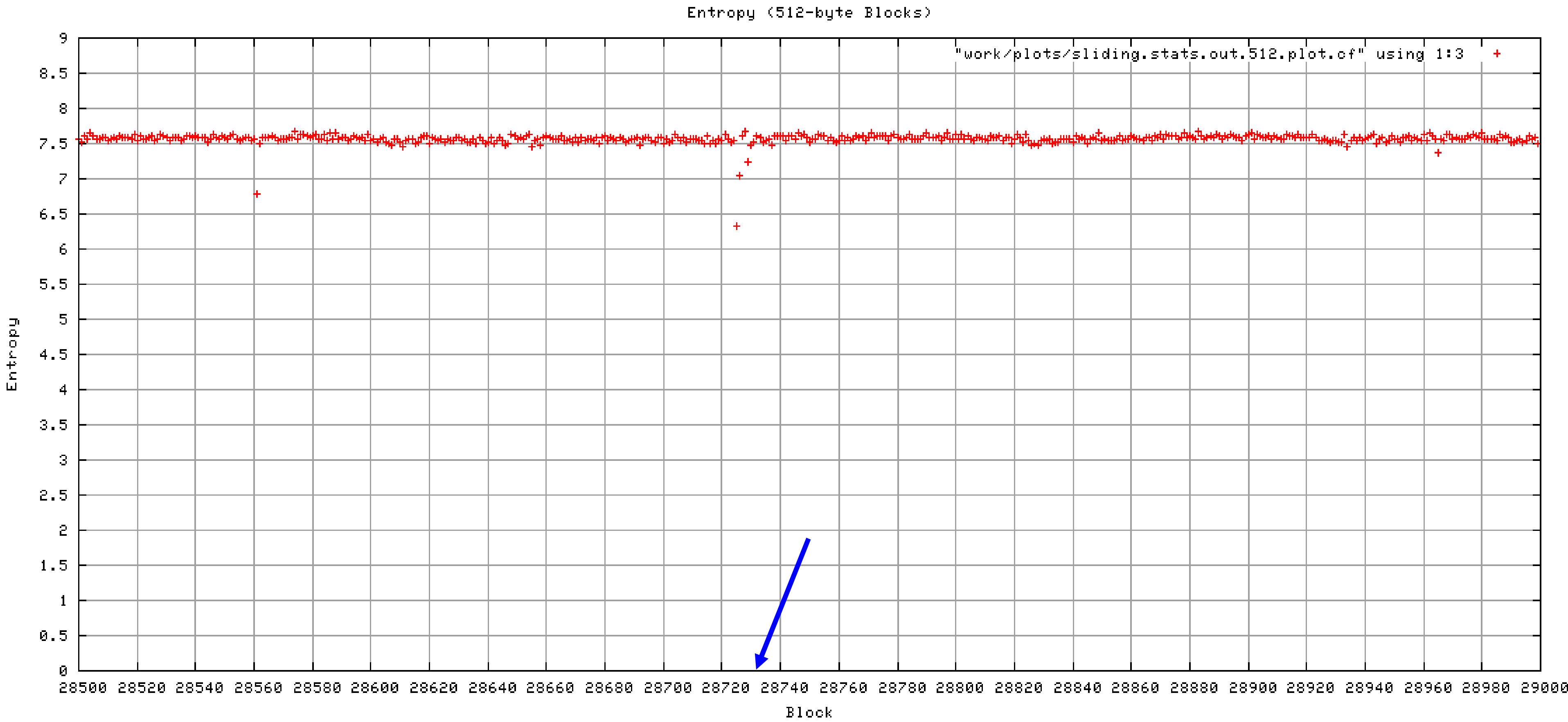
- This example shows us trying to validate a file carve of a ZIP archive file.
- The validator program found errors with the carved file and reported that there are 187904 extra bytes contained within the ZIP file.
- We then looked at other data points to see if we could locate the extra data and carve it out of the ZIP file.

```
jay@beast: unzip -t dfrws-2006-challenge.raw.14709248.zip
Archive:  dfrws-2006-challenge.raw.14709248.zip
warning [dfrws-2006-challenge.raw.14709248.zip]:  187904 extra bytes at beginning or within zipfile
  (attempting to process anyway)
file #1:  bad zipfile offset (local header sig):  187904
  (attempting to re-compensate)
    testing: file1.jpg
  error:  invalid compressed data to inflate
file #2:  bad zipfile offset (local header sig):  488600
  (attempting to re-compensate)
    testing: file2.jpg                OK
At least one error was detected in dfrws-2006-challenge.raw.14709248.zip.
jay@beast: 
```



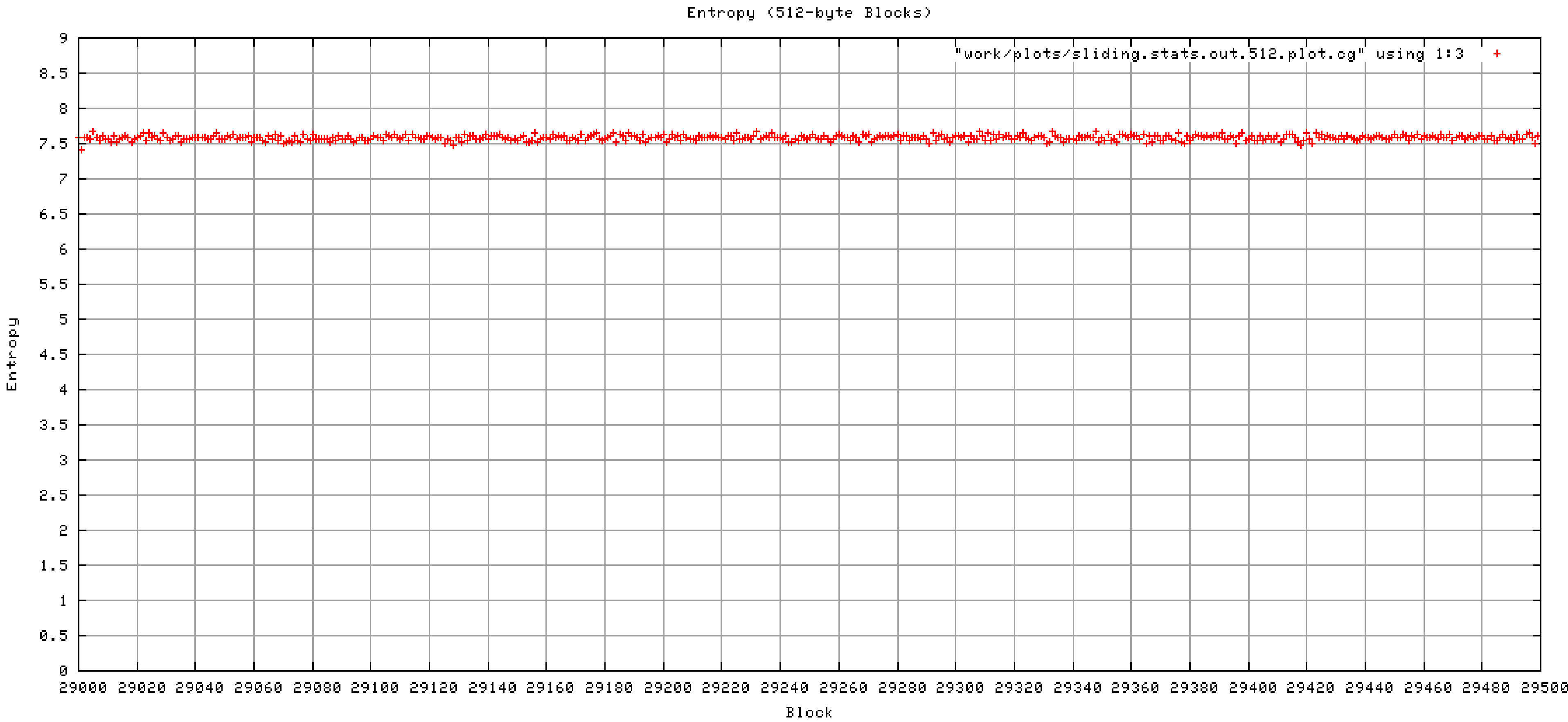
Using Entropy and do_itrim

- Here is the first of three entropy graphs for the ZIP archive.
- We knew from the stage 1 carve that our archive began at block 28729.



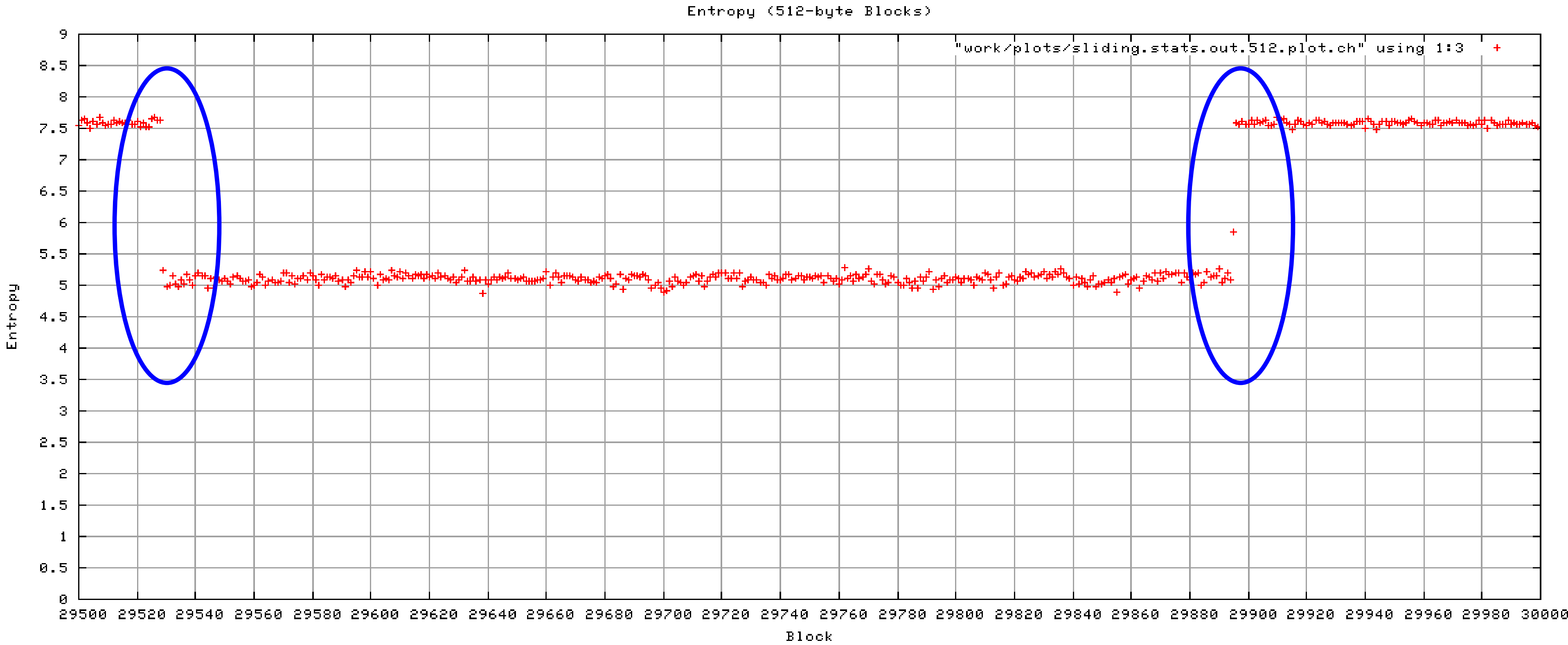
Using Entropy and do_itrim

The entropy values continue along within the ZIP archive.



Using Entropy and do_itrim

- This plot shows a drastic drop in entropy that starts at block 29529 and continues until block 29895 where the entropy returns to the same level as before.
- This is a good indication that this is extra data within the carved ZIP archive.
- The amount of data with the lower entropy matches what unzip reported. (29895-29529)*512+512 = 187904. 512 is the blocksize in bytes.



Using Entropy and do_itrim

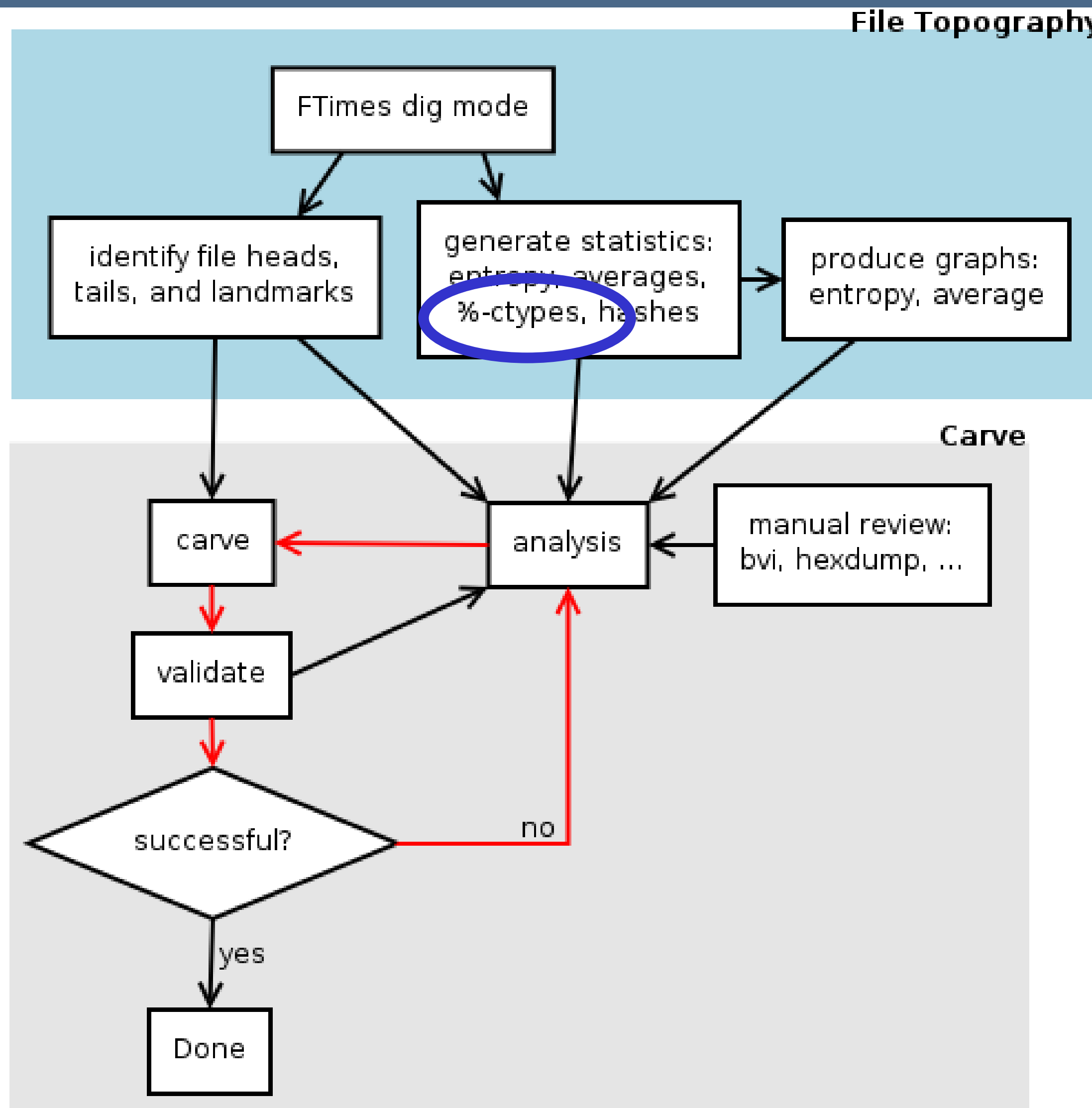
- Using do_itrim, we carved out the extra 187904 bytes.
- The lower bound, 407552 was chosen based on the results of viewing the file's sliding entropy.

```
jay@beast: do_itrim -e zip -l 407552 -s 512 -t 187904 -f dfrws-2006-challenge.raw.14709248.zip
-- unzip -t %subject
dfrws-2006-challenge.raw.14709248.zip trimmed 187904 @ 407552 ---> fail
dfrws-2006-challenge.raw.14709248.zip trimmed 187904 @ 408064 ---> fail
dfrws-2006-challenge.raw.14709248.zip trimmed 187904 @ 408576 ---> fail
dfrws-2006-challenge.raw.14709248.zip trimmed 187904 @ 409088 ---> fail
dfrws-2006-challenge.raw.14709248.zip trimmed 187904 @ 409600 ---> pass
jay@beast:
```

- Here, we manually tested the new ZIP archive to show the contents and validate the archive.
- The file tests OK and we have our final carved ZIP archive.

```
jay@beast: unzip -t subject.zip
Archive:  subject.zip
    testing: file1.jpg               OK
    testing: file2.jpg               OK
No errors detected in compressed data of subject.zip.
jay@beast:
```

Methodology




Sliding Statistics and MySQL

- Sliding percent ctype(3) good for identifying block contents:
  - High % alpha & numeric characters indicates TEXT or HTML
  - ZIP- and JPEG-based blocks contain flat distributions of alpha & numeric characters
- FTimes and XMagic to harvest statistics and topographical info
- Loaded into MySQL so that we could run various analysis queries
- HTML Example:

```
mysql> select * from stats limit 1;
       block: 0                   cntrl: 1.367188
block_offset: 0                   digit: 5.664062
      offset: 0                   lower: 67.773438
   blocksize: 512                 print: 98.632812
       rent1: 4.656387            punct: 8.398438
       rent2: 7.282739            space: 16.601562
       rave1: 86.371094           upper: 1.5625
       rave2: 22197.371094        sha1:95c44d983ef91535ee4a60d90bcb861e9f6f8e11
       alnum: 75                  md5:98d90194d35bae4fcabc0878419deca2
       alpha: 69.335938           html_tags: yes
       ascii: 100
```

Sliding Statistics and MySQL (Cont.)

Here is an example query we used to find contiguous blocks of text that did not contain HTML:

```
SELECT block FROM stats WHERE blocksize = 4096 AND print >=
80 AND html_tags = 'no' AND rent1 > 3 AND rent1 < 6 ORDER BY
block;
```

This query produced the (abbreviated) output shown below. These blocks were then fed to ftimes-group-blocks.pl, which produced output that could be used directly by ftimes-crv2raw.pl to carve text from the raw image.

```
+-------+
| block |
+-------+
|  1478 |
|  1479 |
|  1480 |
|  .... |
|  4964 |
+-------+
382 rows in set
```

Agenda

- Introduction to Digital File Carving
- 2006 File Carving Challenge
- Methodology
- **Conclusion**

Challenge Results

Recovered files	Embedded Files	Challenge Answers
43(1)	10(2)	32

(1) We extracted one additional partial file which was fragmented French text. This file was included in the answer set because we considered it part of the body of evidence which could be relevant to the investigation.

(2) We carved out additional embedded files and included those in our submission due to the fact they were complete files, and we felt that investigators would not want to arbitrarily exclude any file regardless of its location. This technique can be used in other scenarios such as carving out images embedded in Microsoft documents or other types of compound files.

Next Steps

- Inform your forensics team:
  - Free forensics tools they can put to use today.
    - FTimes for system baselining and evidence collection.
    - Download the file carving tools and use them.

      http://www.korelogic.com/Resources/Projects/dfrws_challenge_2006/
- Sliding entropy calculations:
  - Can improve the accuracy of the file carving process
  - Can reduce false positives
  - Show promise for edge detection
- More file carving research is needed
  - Forensic techniques, including file carving, must continue to increase their “granularity” to discern smaller pieces of data.

Forensic Resources

- **Books**
  - Farmer, D., and Venema, W. (2004). Forensic Discovery. Addison-Wesley.
  - Digital Evidence and Computer Crime (E. Casey, Academic Press)
  - Computer Forensics and Privacy (M. Caloyannides, Artech House)
- **Websites**
  - http://www.dfrws.org/ - Digital Forensic Research Workshop
  - http://www.ijde.org/ - International Journal of Digital Evidence
  - http://vip.poly.edu/kulesh/forensics/list.htm - conferences, people, online papers
  - http://www.tucofs.com/tucofs/tucofs.asp?mode=mainmenu – "The Ultimate Collection of Forensic Software"
  - http://www.opensourceforensics.org/
- **Examples of digital forensics software**
  - FTimes
  - Foremost, Scalpel
  - EnCase, FTK, ILook, Sleuthkit
  - WinHex

Questions and Slides

- Questions?

- A version of this briefing is available at:

http://www.korelogic.com/Resources/Projects/dfrws_challenge_2006/