

# BULK\_EXTRACT LIKE A BOSS

---

I.E., A TALK ABOUT BULK\_EXTRACTOR

Jon Stewart

Lightbox Technologies, Inc

[jon@lightboxtechnologies.com](mailto:jon@lightboxtechnologies.com)







# BULK\_EXTRACTOR

- ❖ Prof. Simson Garfinkel, Naval Postgraduate School
- ❖ ~2006–Present
- ❖ Download: [http://digitalcorpora.org/downloads/bulk\\_extractor/](http://digitalcorpora.org/downloads/bulk_extractor/)
- ❖ Develop: [https://github.com/simsong/bulk\\_extractor](https://github.com/simsong/bulk_extractor)
- ❖ Scans through a disk image and finds a bunch of stuff

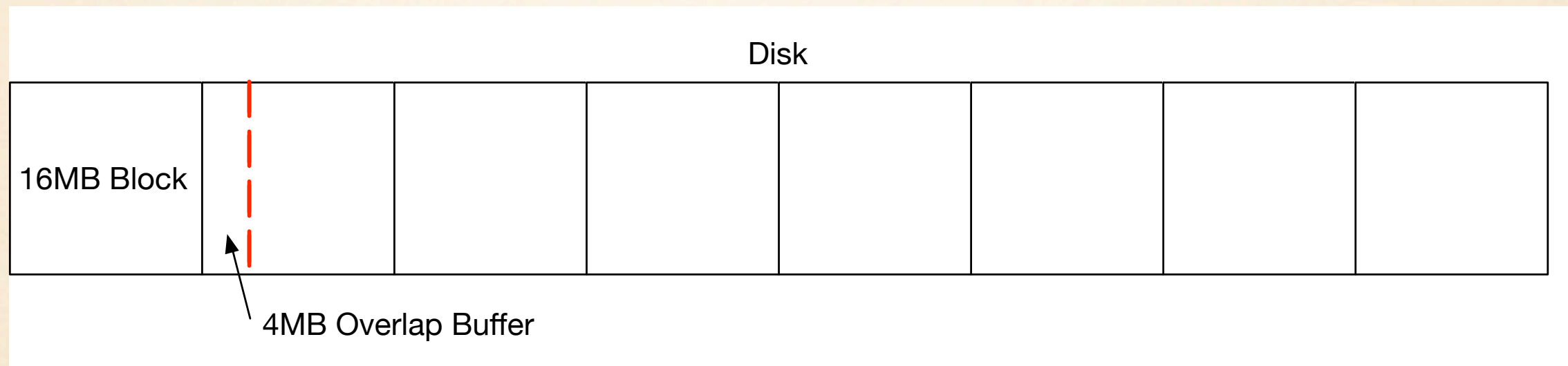
Tuesday, November 5, 13

```
Build bulk_extractor  
untar -xzf bulk_extractor-1.4.1.tgz  
cd bulk_extractor-1.4.1  
./configure && make -j2
```



# PROCESSING MODEL

Scan overlapping “pages” sequentially

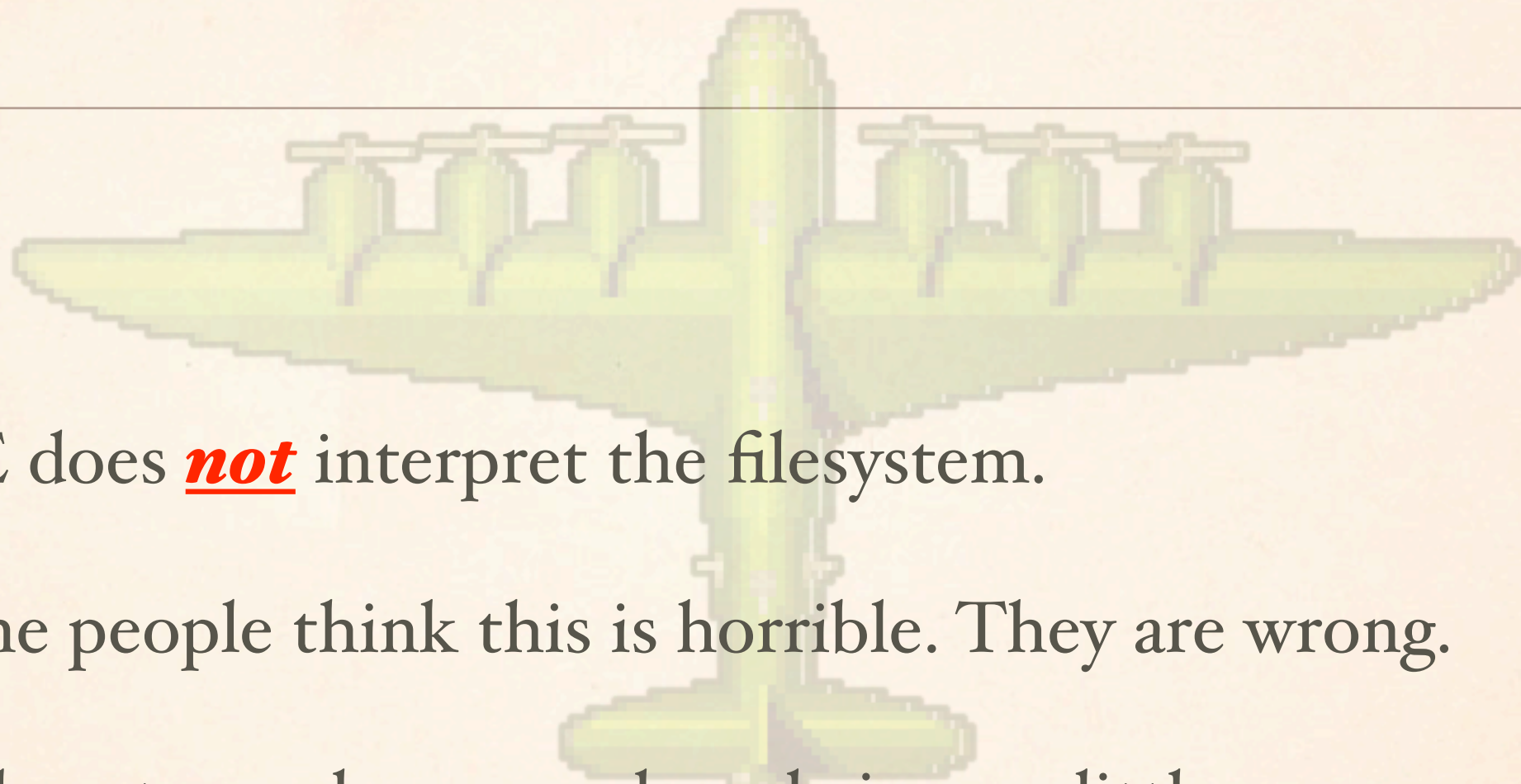


Process them in parallel  
with a threadpool

*Superfast!!*



# B\_E'S WEAKNESS IS ITS VIRTUE



- ❖ B\_E does ***not*** interpret the filesystem.
- ❖ Some people think this is horrible. They are wrong.
- ❖ A filesystem asks so much and gives so little.
- ❖ B\_E finds the stuff you care about, regardless.

# FLEX SCANNERS

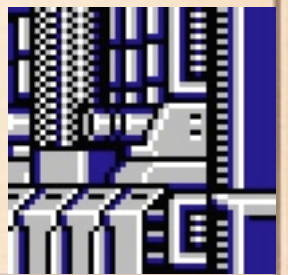


- ❖ accts—find CCNs, phone numbers, dates, FedEx tracking numbers, SSNs
- ❖ email—MsgID, Subject, other headers, Email addresses & domains, IP addresses, MAC addresses, URLs
- ❖ gps—GPS XML records



# MANY SCANNERS

- ❖ kml—Google Earth location data, lotsa' other apps now use
- ❖ net—carve network packets
- ❖ aes—find AES key schedules, kinda slow so maybe disable
- ❖ json—find JSON data
- ❖ elf—find linux executables
- ❖ exif—find & extract EXIF metadata; also, carve the jpegs
- ❖ winpe—find Windows executables and extract metadata
- ❖ winprefetch—find Windows prefetch artifacts
- ❖ windirs—find & parse MFT and FAT records
- ❖ VCard—find & parse VCards
- ❖ wordlist—strings | uniq



# RECURSIVE SCANNERS

- ❖ pdf (decodes text)
  - ❖ zip
  - ❖ rar
  - ❖ gzip
  - ❖ xor (specify value with option)
  - ❖ base16
  - ❖ base64
  - ❖ ascii85 (base85)
  - ❖ hiberfil
- ❖ Fully recursive. Data blocks decoded by these scanners are then passed through all the scanners, including these recursive scanners.
  - ❖ A maximum limit keeps things sane.
  - ❖ These scanners often find snippets that traditional file carvers won't.





# USEFUL OPTIONS

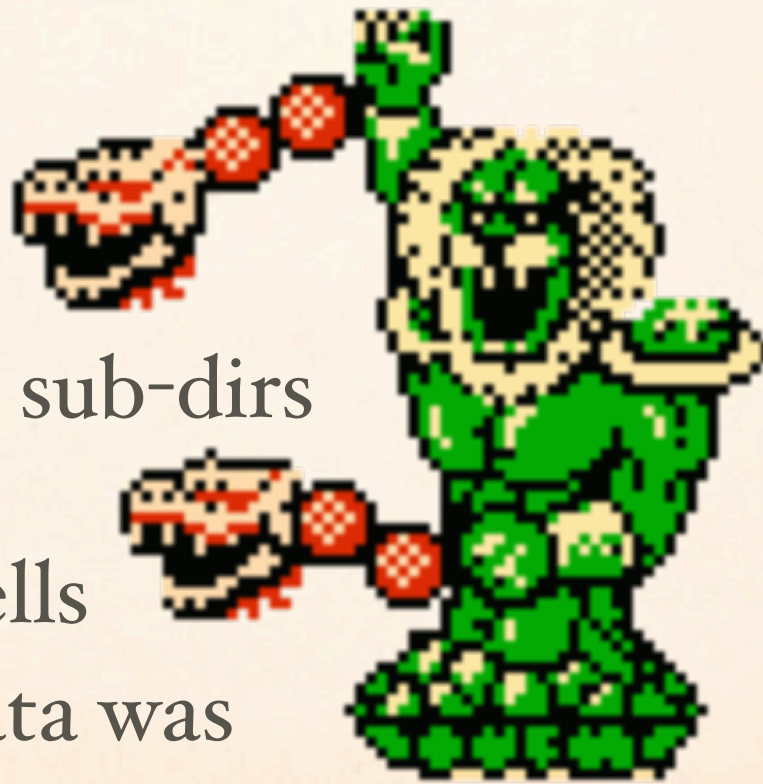
- ❖ Last arg is file to process.  
EWF/AFF/dd/live device
- ❖ or use -R to process a  
directory of files
- ❖ “-C 16” to set context  
window size to 16 bytes
- ❖ “-w ignore.txt” to specify a  
stoplist of strings to ignore
- ❖ -Z to overwrite the output  
directory
- ❖ “-Y 16m” to process  
starting at offset 16,777,216
- ❖ “-Y 16m-32m” to scan a  
range of data
- ❖ “-z 5” to start processing at  
page 5 ( $5 * 16 * 1024 * 1024$ )





# OUTPUT

- ❖ Matches are written to files named *foo.txt*
- ❖ Unique strings are tabulated in *foo\_histogram.txt*
- ❖ Carved files go to sub-dirs
- ❖ “Forensic Path” tells location where data was found.



```
Ovid:bulk_extractor-1.4.1 jon$ ls -la outputDir/
total 455496
drwxr-xr-x 48 jon staff 1632 Nov 4 21:59 .
drwxr-xr-x@ 41 jon staff 1394 Nov 4 21:56 ..
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 aes_keys.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 alerts.txt
-rw-r--r-- 1 jon staff 7899 Nov 4 21:58 ccn.txt
-rw-r--r-- 1 jon staff 1499 Nov 4 21:59 ccn_histogram.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 ccn_track2.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 ccn_track2_histogram.txt
-rw-r--r-- 1 jon staff 1987082 Nov 4 21:59 domain.txt
-rw-r--r-- 1 jon staff 12031 Nov 4 21:59 domain_histogram.txt
-rw-r--r-- 1 jon staff 363424 Nov 4 21:57 elf.txt
-rw-r--r-- 1 jon staff 294315 Nov 4 21:59 email.txt
-rw-r--r-- 1 jon staff 14982 Nov 4 21:59 email_histogram.txt
-rw-r--r-- 1 jon staff 433 Nov 4 21:59 ether.txt
-rw-r--r-- 1 jon staff 220 Nov 4 21:59 ether_histogram.txt
-rw-r--r-- 1 jon staff 18045 Nov 4 21:59 exif.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 find.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 find_histogram.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 gps.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 hex.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 ip.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 ip_histogram.txt
drwxr-xr-x 3 jon staff 102 Nov 4 21:57 jpeg_carved
-rw-r--r-- 1 jon staff 2454 Nov 4 21:58 jpeg_carved.txt
-rw-r--r-- 1 jon staff 221927829 Nov 4 21:59 json.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 kml.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 lightgrep.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 lightgrep_histogram.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 pii.txt
-rw-r--r-- 1 jon staff 5156 Nov 4 21:56 rar.txt
-rw-r--r-- 1 jon staff 25824 Nov 4 21:59 report.xml
-rw-r--r-- 1 jon staff 4721 Nov 4 21:59 rfc822.txt
-rw-r--r-- 1 jon staff 22049 Nov 4 21:58 telephone.txt
-rw-r--r-- 1 jon staff 2427 Nov 4 21:59 telephone_histogram.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 unrar_carved.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 unzip_carved.txt
-rw-r--r-- 1 jon staff 3287102 Nov 4 21:59 url.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 url_facebook-address.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 url_facebook-id.txt
-rw-r--r-- 1 jon staff 351430 Nov 4 21:59 url_histogram.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 url_microsoft-live.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:59 url_searches.txt
-rw-r--r-- 1 jon staff 7414 Nov 4 21:59 url_services.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 vcard.txt
-rw-r--r-- 1 jon staff 3727771 Nov 4 21:59 windirs.txt
-rw-r--r-- 1 jon staff 8469 Nov 4 21:56 winpe.txt
-rw-r--r-- 1 jon staff 0 Nov 4 21:56 winprefetch.txt
-rw-r--r-- 1 jon staff 1090610 Nov 4 21:58 zip.txt
```









# LIGHTGREP



- ❖ v1.4 adds lightgrep scanner for keyword searching
- ❖ On Linux or MacOS X, download, build, and install liblightgrep first: <https://github.com/LightboxTech/liblightgrep>
- ❖ liblightgrep is GPLv3, with a simple C API
- ❖ `$ bulk_extractor -F patterns.txt -x find -o outputDir image.dd`
- ❖ ***Disable the “find” scanner for performance!***

Tuesday, November 5, 13

```
./src/bulk_extractor -F ../be-demo/50.txt -x aes -Y 0-256m -o outputDir ~/ev/lbt.dd  
./src/bulk_extractor -F ../be-demo/1000.txt -x find -x aes -Y 0-256m -o outputDir -Z ~/ev/lbt.dd
```



# SCAN\_LIGHTGREP

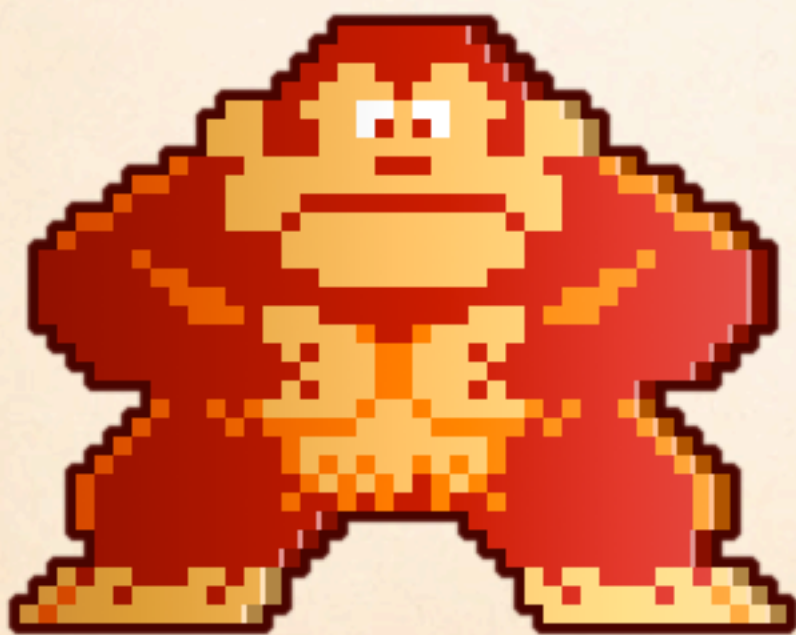
- ◆ Both UTF-8 and UTF-16LE encodings are searched
- ◆ Subset of Perl regular expression syntax
  - ◆ <http://downloads.lightboxtechnologies.com/help/LightgrepCheatSheet.pdf>
- ◆ Full Unicode support
- ◆ “lightgrep.txt” lists individual matches
- ◆ “lightgrep\_histogram.txt” gives frequency statistics









# LIGHTGREP & UNICODER

- ❖ Lightgrep understands character names, scripts, properties, & code points
- ❖ Unicode v6.3 (with latest libicu)
- ❖ Things to look forward to in Unicode 7:



Live Kong, and Prosper

*FDAM2 code chart images of characters 1F594 through 1F596*

1F594		REVERSED VICTORY HAND → 270C  victory hand
1F595		REVERSED HAND WITH MIDDLE FINGER EXTENDED
1F596		RAISED HAND WITH PART BETWEEN MIDDLE AND RING FINGERS

Tuesday, November 5, 13

```
./src/bulk_extractor -E lightgrep -F ../be-demo/unicode_keys -o unicodeOutDir ../SomeUnicodeCharacters.txt
```



# UNICODE EXAMPLE

- ❖ Given text file on the left...
- ❖ These patterns will hit:
  - ❖ `\p{Arabic}+`
  - ❖ `p{Cyrillic}+`
  - ❖ `\N{U+1f3e9}`
  - ❖ `\N{PILE OF POO}`

Arabic Letter Jeem  
U+062C

ج

Cyrillic Capital Letter LJE  
U+0409

Љ

Love Hotel  
U+1f3e9



Pile of Poo  
U+1f4a9





# FUTURE ENCODINGS

- ❖ In next version, can specify encodings per pattern
- ❖ Specify pattern, tab, then comma-separated encodings:
  - ❖ `foo.{1,6}bar` `ASCII,CP-1251,ISO-8559-4`
- ❖ Lightgrep regexps won't conflict with "find" scanner







# DFXML



- ❖ You can use fiwalk to generate DFXML for an image, and then use a Python script to relate b\_e matches back to files
- ❖ Needs Python 3.2+ (you really do want Python 3.2+)
- ❖ `$ fiwalk -z -g -x image.dd > image.xml`
- ❖ `$ python3.3 python/identify_filenames.py --xmlfile image.xml --all b_e_outdir newOutDir`



# THANKS!

---

DOWNLOAD: [http://digitalcorpora.org/downloads/bulk\\_extractor/](http://digitalcorpora.org/downloads/bulk_extractor/)  
DEVELOP: [https://github.com/simsong/bulk\\_extractor](https://github.com/simsong/bulk_extractor)  
CHEATSHEET: <http://downloads.lightboxtechnologies.com/help/LightgrepCheatSheet.pdf>  
LIBLIGHTGREP: <https://github.com/LightboxTech/liblightgrep>

