CORY ALTHEIDE

HOW TO BE A FORENSICS HIPSTER

03 OCT 2012 - #OSDFC

CORY@GOOGLE.COM

DIGITAL ARCHAEOLOGIST

ALLOW MYSELF TO INTRODUCE... MYSELF

- Cory Altheide
- Forensics, IR, Author
- Relatively Tall
- Bacon Number: 2

What skills or expertise do your other connections have?

[redacted] Does [redacted] know about Computer Forensics?

Does Cory Altheide know about Computer Security?

[redacted] Does [redacted] know about Computer Security?

[redacted] Does [redacted] know about Network Security?

OUR AGENDA

- Mobile
- PDFs
- Executables
- Metatools
- Shout Outs

MOBILE

4:00
Google
Google+
Google

Verizon 3G
3:26 PM
Messages
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Notes
Utilities
iTunes
App Store
Game Center
Settings
Phone
Mail
Safari
iPod

MOBILE - iOS

- HFSX file system
  - System volume & "Data" volume
- Collected artifacts similar to OS X:
  - Lots of plist files (binary & text)
- Backups can sometimes be found on host system
  - Can be encrypted
  - Some "protected" files always encrypted

MOBILE - iOS

- Backups:
  - iPhone Backup Analyzer
- Device & Backup:
  - libmobiledevice (logical acquisition)
  - Sogeti iPhone Data Protection Tools (physical)

MOBILE - ANDROID

- YAFFS2/EXT4 mostly
- VFAT for "sdcard"
- Debug mode devices accessible over 'adb' from host system
- Interesting artifacts include *lots* of SQLite databases

MOBILE - ANDROID

- "Open Source Android Forensics (**OSAF**)" project.
  - Mostly geared towards APK/malware analysis
- Viaforensics' tools:
  - **AFLogical** acquisition utility (logical)
  - **Santoku Linux** (including physical acquisition*)
  - Sample device volume images!

PDF ANALYSIS

- Adobe's **P**ortable **D**ocument **F**ormat
- Collection of 8 types of objects
  - "Dictionaries" & "Streams" being the most interesting to us
- Can contain fun stuff like JavaScript & Flash
- Typically used to deliver 0-day & subsequent AV company whitepapers about those same attacks

PDF ANALYSIS - DIDIER STEVENS TOOLS

- Didier Stevens' suite of PDF tools:
    - **pdf-parser.py**: in-depth PDF analysis
    - **make-pdf.py**: Javascript/file embedding
    - **pdfid.py**: quick/naive PDF triage

PDF ANALYSIS - PDFID

```
PDF Header: %PDF-1.3
obj                   10
endobj                10
stream                 2
endstream              2
xref                   1
trailer                1
startxref              1
/Page                  1
/Encrypt               0
/ObjStm                0
/JS                    2
/JavaScript            3
/AA                    0
/OpenAction            1
/AcroForm              0
/JBIG2Decode           0
/RichMedia             0
/Launch                0
/EmbeddedFile          0
/Colors > 2^24         0
```

PDF ANALYSIS - PEEPDF

- Peepdf: a python PDF investigation utility
- Identification of suspicious elements
- Interactive shell to browse & dump elements
- Can extract older versions of document (if present)
- Supports Javascript & Shellcode analysis via Spidermonkey & Libemu (if present)

PDF ANALYSIS - ORIGAMI

- Origami: ruby framework for exploring, dissecting, or generating PDFs.
- Comes with a suite of example tools:
  - **pdfmetadata**: extracts metadata
  - **pdfextract**: dumps all objects (JS, images, etc)
  - **pdfcop**: automated 'badness' detector
  - **pdfwalker**: GTK graphical PDF explorer

PDF ANALYSIS - ORIGAMI

```
PDFcop is running on target `pdf-jsEval.file',
policy = `standard'
File size: 52793 bytes
MD5: a9e2a597df08f99944a06f175d53d003
> Inspecting document structure...
> Inspecting document catalog...
  . OpenAction entry = YES
  >> Inspecting action...
    .. Destination page found.
    >>> Inspecting page...
> Inspecting JavaScript names directory...
Document rejected by policy `standard', caused
by [:allowJSAtOpening].
```

PDF ANALYSIS - MISC

- jsunpack-n: a 'browser emulator'/honeyclient
  - can directly consume & process PDFs, Javascript, SWF, in addition to HTML & pcap.
  - Output is very terse, good for triage
- malpdfobj:
  - generates a JSON representation of a given PDF
  - good for automation

EXECUTABLE ANALYSIS

- PE-COFF (Windows)
- ELF (Linux*)
- Mach-O (OS X, iOS)

PLACEHOLDER FOR FUNNY PICTURE ABOUT EXECUTABLES

EXECUTABLE ANALYSIS - PE

- pefile.py
- Library only, no pre-built tools
- Super easy to use, however:

```
import pefile
badfile = pefile.PE('funnycats.exe')
print badfile.dump_info()
```

- Used in MHL's pescanner.py and many other Python malware analysis utilities

EXECUTABLE ANALYSIS - PE

- pev
- toolkit based on 'libpe': cross-platform PE handler C-library *totally not written by Joachim*.
- **readpe**: parse headers, section info, imports
- **pescan**: PE anomaly detector

EXECUTABLE ANALYSIS - PE

- verify-sigs

- Portable Python Authenticode check for PE binaries.

```
python print_pe_certs.py test_data/SoftwareUpdate.exe
...
Program: Apple Software Update, URL: http://www.apple.com/macosx
Countersignature is present. Timestamp: Fri Jul 25 22:21:53 2008 UTC
Binary is signed with cert issued by:
("{'C': 'US', 'OU': 'Terms of use at https://www.verisign.com/rpa (c)04, VeriSign
Trust Network', 'O': 'VeriSign, Inc.', 'CN': 'VeriSign Class 3 Code Signing 2004
CA'}",
 12451790217711796967571790799059482938L)

Cert chain head issued by:
("{'C': 'US', 'OU': 'Class 3 Public Primary Certification Authority', 'O':
'VeriSign, Inc.'}",
 87155975386774669517273893148021257666L)
  Chain not before: Wed Jun 27 00:00:00 2007 UTC
  Chain not after: Fri Jun 26 23:59:59 2009 UTC
```

EXECUTABLE ANALYSIS - ELF

- file
- readelf
- objdump
- pyelftools: Python library for analyzing ELF files & DWARF debugging info.
  - includes fully-portable readelf clone

EXECUTABLE ANALYSIS - MACH-O

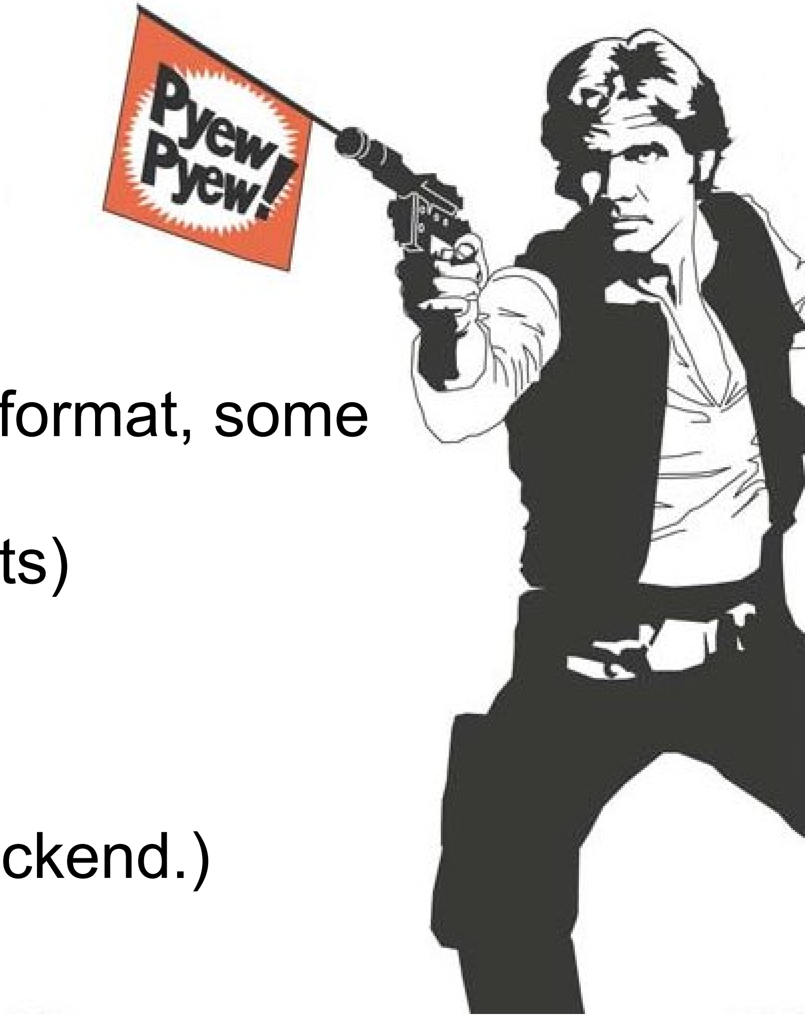
- Macholib: Python portable Mach-O header parsing library
- Included CLI tool of note is **macho_dump**
  - Reads & prints architectures & linked libraries
- Not close to an 'otool' replacement

EXECUTABLE ANALYSIS - PYEW

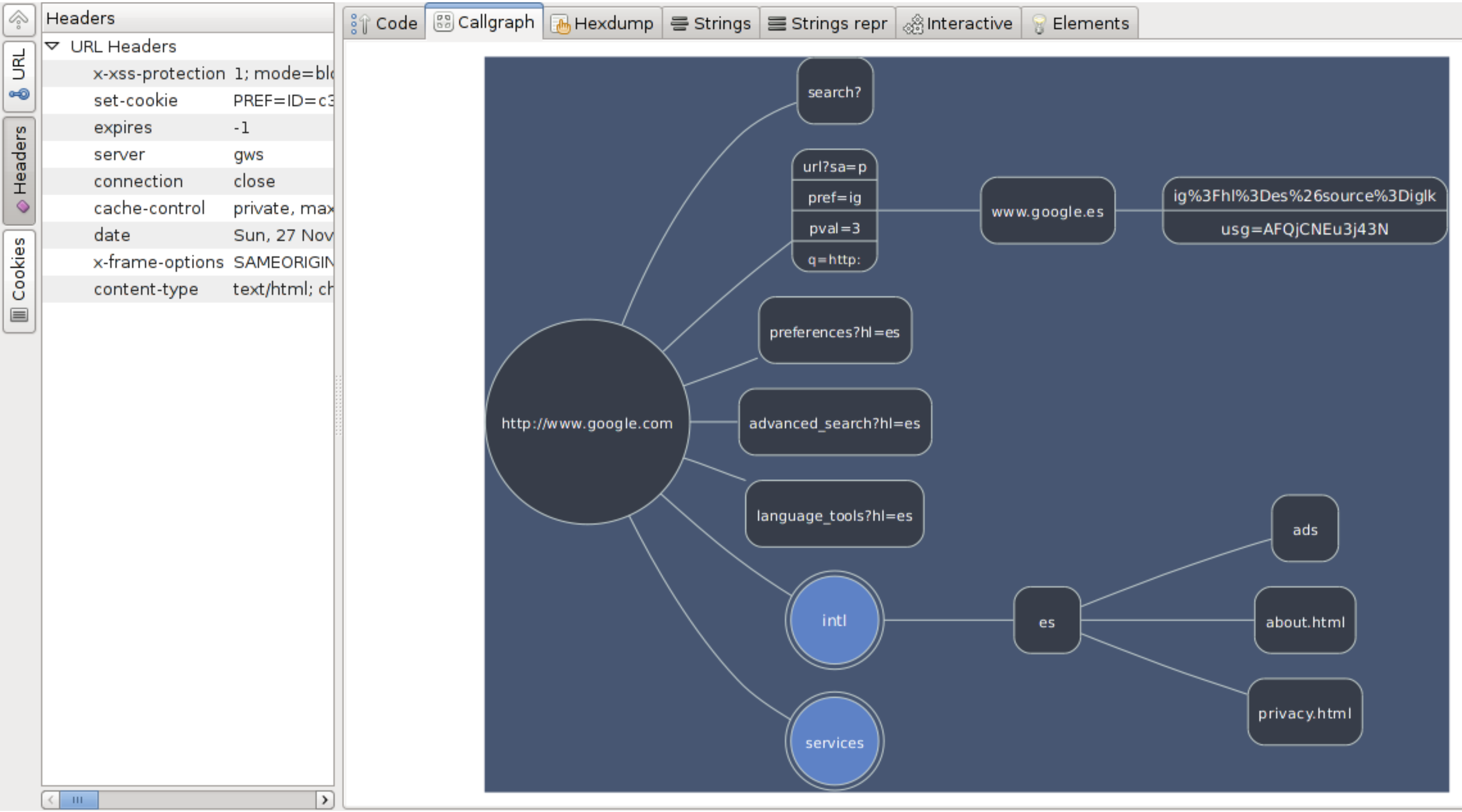
- PYew: Python 'Hiew' work-alike.
- Malware exploration & dissection tool
- PE & ELF binary support.
- Extensible via Python class plugins
- Includes **'gluster'**: a tool used to compare executable similarity based on call graphs

EXECUTABLE ANALYSIS - PYEW

- Not just for executables:
  - PDF
  - OLE2 (Pre 2K3-Office binary format, some additional Windows 7+ artifacts)
- Graphical front-end "**Bokken**"
  - (Can also use **radare** as a backend.)



EXECUTABLE ANALYSIS - BOKKEN



Headers

▽ URL Headers

x-xss-protection	1; mode=bl
set-cookie	PREF=ID=c3
expires	-1
server	gws
connection	close
cache-control	private, max
date	Sun, 27 Nov
x-frame-options	SAMEORIGIN
content-type	text/html; ch

METATOOLS - HACHOIR

- **Hachoir** - Python library for slicing & dicing binary streams.
- Stream browser backed by powerful parser library (70 formats currently supported)
- Lots of useful tools supplied
- *"Hachoir is the French word for a meat grinder (meat mincer), which is used by butchers to divide meat into long tubes; Hachoir is used by computer butchers to divide binary files into fields."*

METATOOLS - HACHOIR

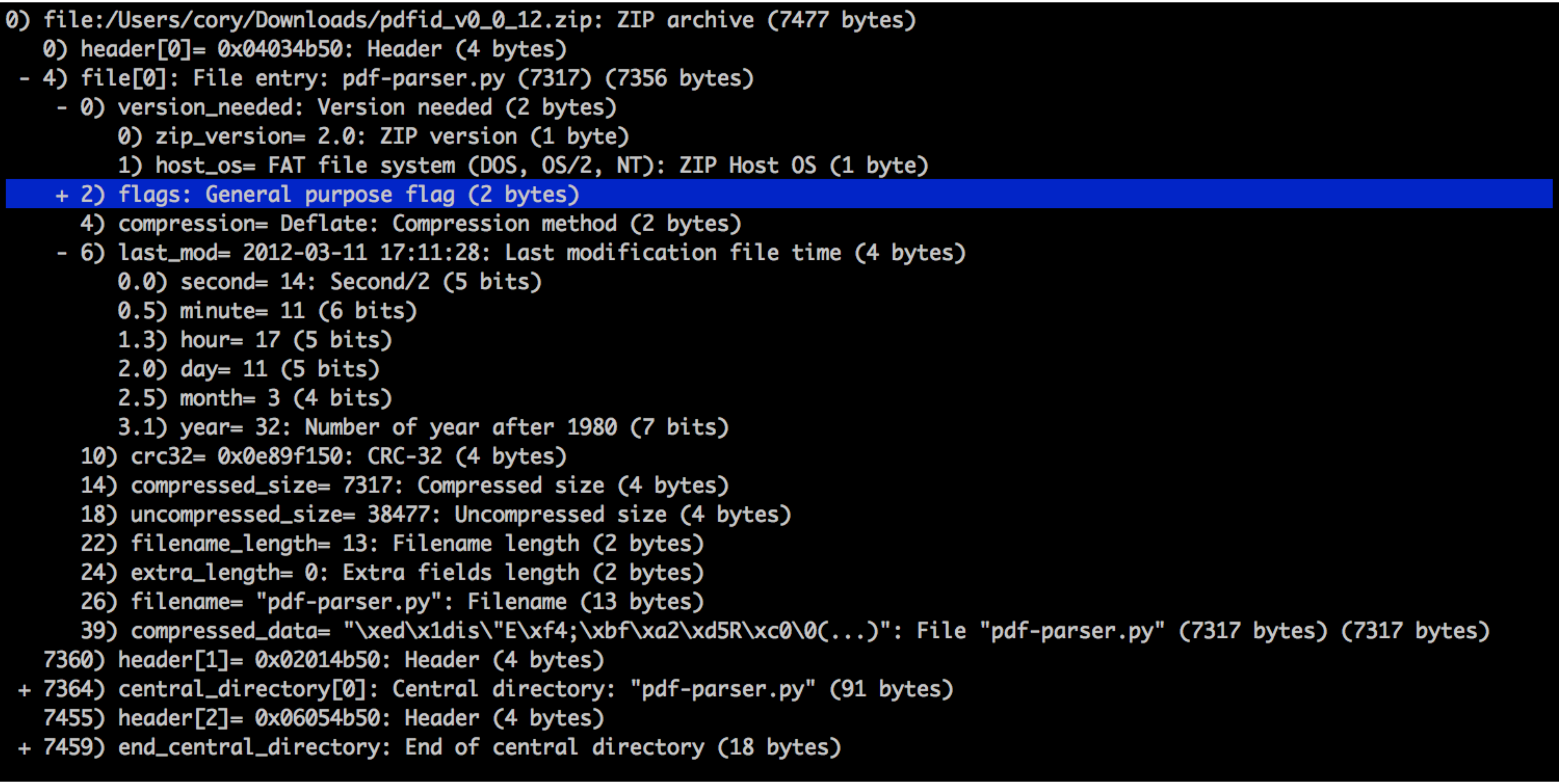
- **Hachoir-metadata**:
  - extract & print metadata from supported file formats
  - Will work on truncated/corrupted files
  - uses only **hachoir-parser** (no other libraries required)

- **Hachoir-subfile**:
  - Locates subordinate files within a binary stream
  - Initiates detection based on magic number, then passes substream to hachoir-parser to validate file.
  - Doesn't (currently) work on compressed streams (patches welcome?)

METATOOLS - HACHOIR

- **Hachoir-urwid/hachoir-wx**:
  - Console (urwid) or GUI (wx) interface for interactively exploring supported files.

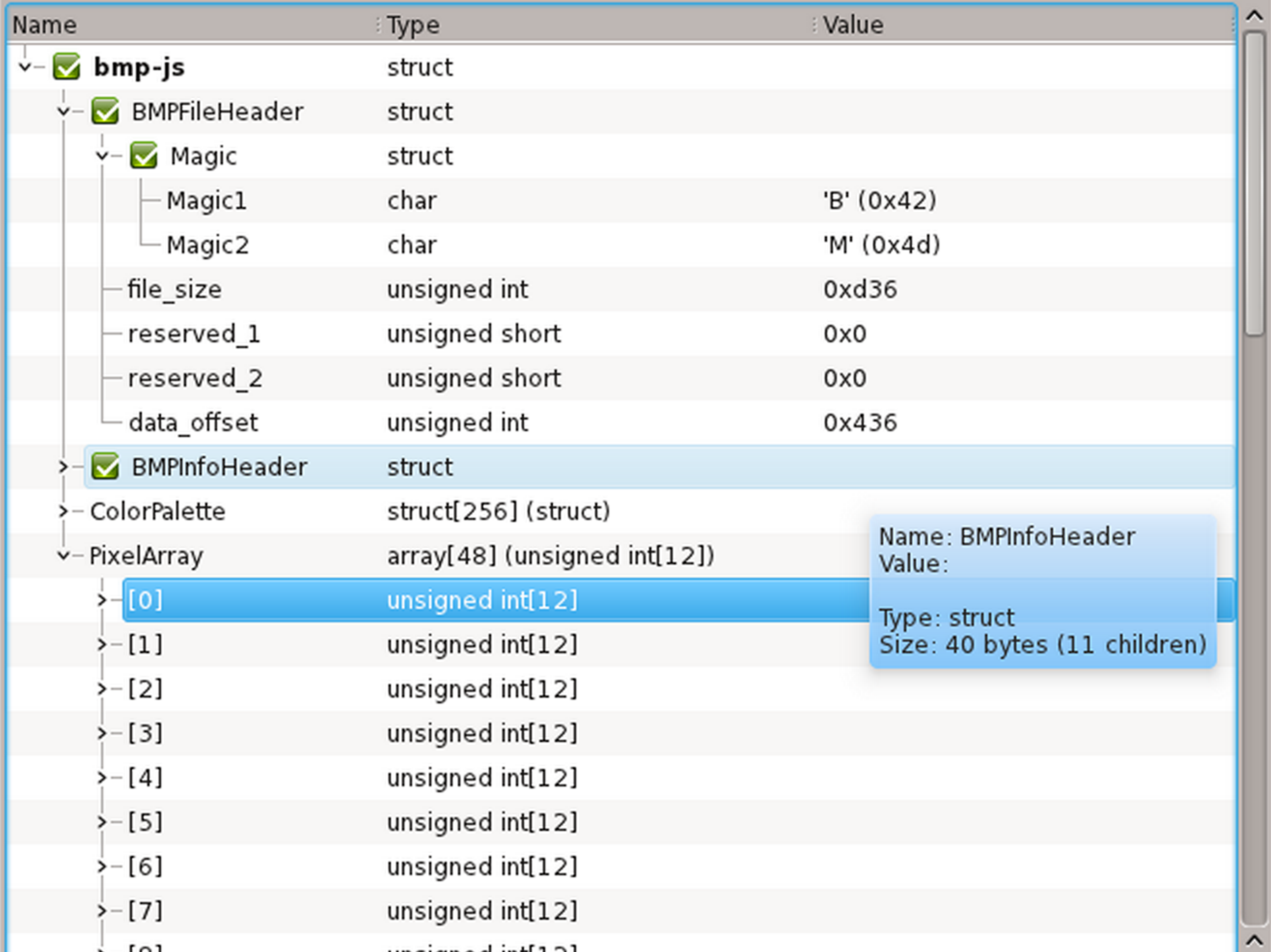
METATOOLS - HACHOIR



```
0) file:/Users/cory/Downloads/pdfid_v0_0_12.zip: ZIP archive (7477 bytes)
   0) header[0]= 0x04034b50: Header (4 bytes)
 - 4) file[0]: File entry: pdf-parser.py (7317) (7356 bytes)
    - 0) version_needed: Version needed (2 bytes)
         0) zip_version= 2.0: ZIP version (1 byte)
         1) host_os= FAT file system (DOS, OS/2, NT): ZIP Host OS (1 byte)
    + 2) flags: General purpose flag (2 bytes)
      4) compression= Deflate: Compression method (2 bytes)
    - 6) last_mod= 2012-03-11 17:11:28: Last modification file time (4 bytes)
         0.0) second= 14: Second/2 (5 bits)
         0.5) minute= 11 (6 bits)
         1.3) hour= 17 (5 bits)
         2.0) day= 11 (5 bits)
         2.5) month= 3 (4 bits)
         3.1) year= 32: Number of year after 1980 (7 bits)
      10) crc32= 0x0e89f150: CRC-32 (4 bytes)
      14) compressed_size= 7317: Compressed size (4 bytes)
      18) uncompressed_size= 38477: Uncompressed size (4 bytes)
      22) filename_length= 13: Filename length (2 bytes)
      24) extra_length= 0: Extra fields length (2 bytes)
      26) filename= "pdf-parser.py": Filename (13 bytes)
      39) compressed_data= "\xed\x1dis\"E\xf4;\xbf\xa2\xd5R\xc0\0(...)": File "pdf-parser.py" (7317 bytes) (7317 bytes)
   7360) header[1]= 0x02014b50: Header (4 bytes)
 + 7364) central_directory[0]: Central directory: "pdf-parser.py" (91 bytes)
   7455) header[2]= 0x06054b50: Header (4 bytes)
 + 7459) end_central_directory: End of central directory (18 bytes)
```

METATOOLS - OKTETA

- Okteta is an open source hex editor for KDE
- Essentially the best free hex editor money can't buy.
- Really well documented in the Okteta Handbook
- Beyond normal hex editor features:
  - Binary Filter w/ bitwise operators
  - Data interpretation w/ structure definitions (ala 010 Editor)



Name	Type	Value
☑ **bmp-js**	struct	
☑ BMPFileHeader	struct	
☑ Magic	struct	
Magic1	char	'B' (0x42)
Magic2	char	'M' (0x4d)
file_size	unsigned int	0xd36
reserved_1	unsigned short	0x0
reserved_2	unsigned short	0x0
data_offset	unsigned int	0x436
☑ BMPInfoHeader	struct	
ColorPalette	struct[256] (struct)	
PixelArray	array[48] (unsigned int[12])	
[0]	unsigned int[12]	
[1]	unsigned int[12]	
[2]	unsigned int[12]	
[3]	unsigned int[12]	
[4]	unsigned int[12]	
[5]	unsigned int[12]	
[6]	unsigned int[12]	
[7]	unsigned int[12]	
[8]	unsigned int[12]	

Name: BMPInfoHeader
Value:

Type: struct
Size: 40 bytes (11 children)

METATOOLS - FORDROP

- Fordrop is an open source forensics collaboration platform
- Upload your artifact-of-interest and collaborate with peers
- Auto-generates graphical timeline for items in your investigation.
- Currently in heavy development.

SHOUTOUTS: TWO KINGS

Will Ballenthin

Joachim Metz

WILLI BALLENTHIN

- MANDIANT NYC-based forensic badass
- **python-registry**
- **INDXParse**
- **ShellBags** parser
- **LFLE** parser
- **Open project to reverse Windows 8's ReFS**

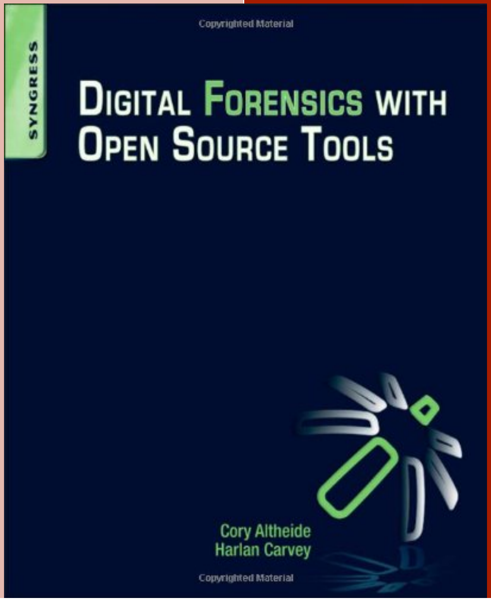
JOACHIM METZ

- Google IR Zurich lib-master
- **libewf**
- **libevt**/**libevtx**
- **libolecf**
- **libregf**
- **libbde**/**libfvde**/**libvshadow**
- Collected works at http://code.google.com/p/libyal

- We're hiring!
- Contact: cory@google.com

THE END

AUTOGRAPHS?
INQUIRIES!
DERISION.



03 Oct 2012 - #OSDFC

CORY@GOOGLE.COM
+CORYALTHEIDE

@CORYALTHEIDE