Forensics in CTFs:

Real-life CSI??? (kinda)

Slides by Miguel for **CyberSoc's CTF Cafe**

What is Forensics?

Forensics is a CTF category which revolves around examining hidden data in an otherwise innocuous static file. Forensics in CTFs is pretty dissimilar to real world forensics tasks, but the tools and techniques used are very much applicable!

What makes a good Forensics Challenge solver?

Observation Skills

Spotting patterns,
anomalies etc.

Research Ability

Knowing file formats,
typical protocols etc.

Understanding Toolkits

Tools are your best friend in forensics and help parse unreadable information

Opt: Programming

Scripting languages are useful for harder tasks...

Types of Forensic Challenge

01

File Analysis

You can describe the topic of the section here

Steganography

File [illegible]dden in [illegible]den in fi[illegible]dden i[illegible] hidd-

03

Memory Analysis

Examining disk images, live memory dumps and more!

04

Network Analysis

Looking into network traffic through packet captures.

01

File Analysis

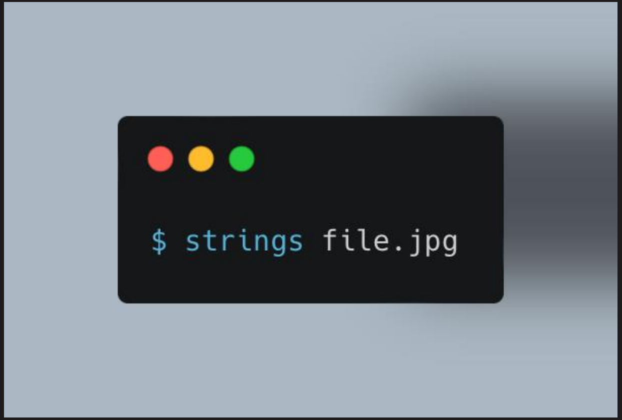
Let’s jump into how to analyse files in Forensics challenges!

Challenge Structure

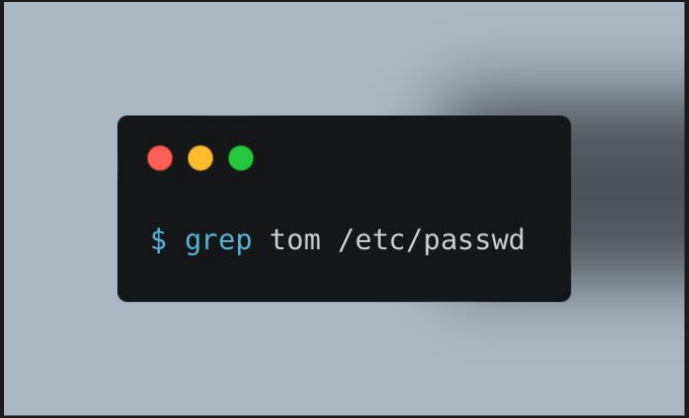
File analysis challenges provide you with a file which contains or leads to the flag in some way.

- Identifying a file format and manipulating accordingly
- Fixing a broken or corrupted file
- Analysing metadata or patterns in file hex
- **In harder cases:** scripting to manipulate binary data

Initial Analysis



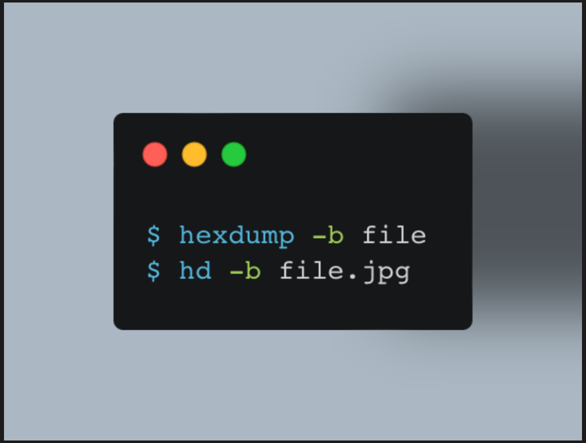
```
$ strings file.jpg
```



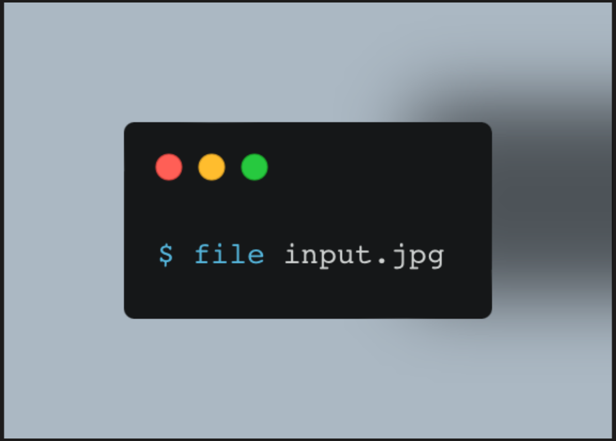
```
$ grep tom /etc/passwd
```

- Command line utilities are a great first step when analysing files.
- **Strings** prints out all plaintext strings in a file.
- **Grep** searches for a specified string in a given file.
- You can also use **bgrep** to find non-text patterns!

Initial Analysis cont.



```
$ hexdump -b file
$ hd -b file.jpg
```



```
$ file input.jpg
```

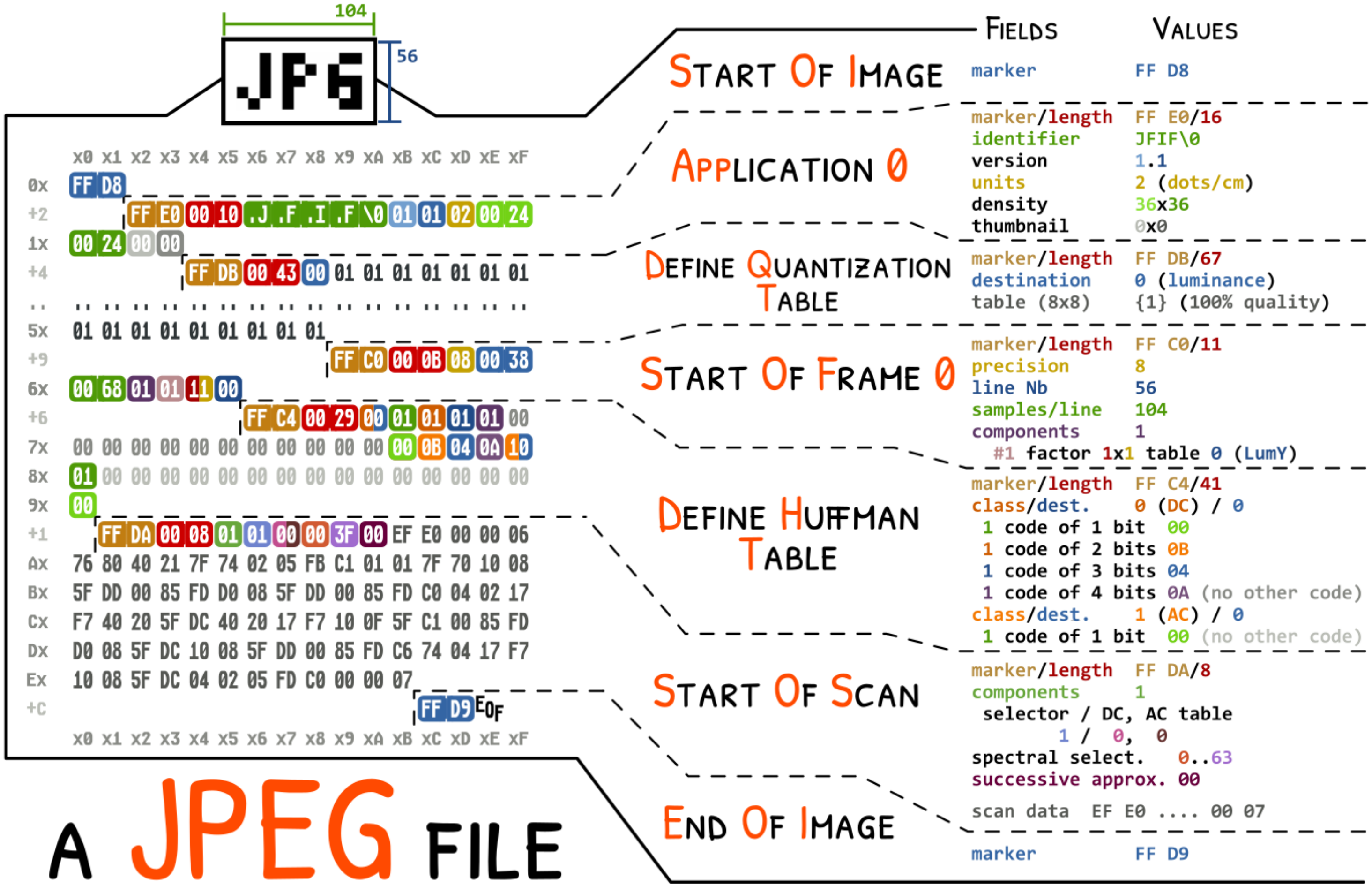
- **Hexdump** will display the content of a file in hex – this is useful for spotting patterns or anomalies.
- Sometimes a file isn't what you think it is... use **file** to check what its file signature indicates.

Common File Formats

- Archive files (ZIP, TGZ)
- Image file formats (JPG, GIF, BMP, PNG)
- Filesystem images (especially EXT4)
- Packet captures (PCAP, PCAPNG)
- PDF
- Video (especially MP4) or Audio (especially WAV, MP3)
- Microsoft’s Office formats (RTF, OLE, OOXML)

Source: https://www.csc.ac.za/?page_id=249

Never seen a file type before? The challenge is probably to do with finding an obscure tool/documentation set to learn how to handle it.



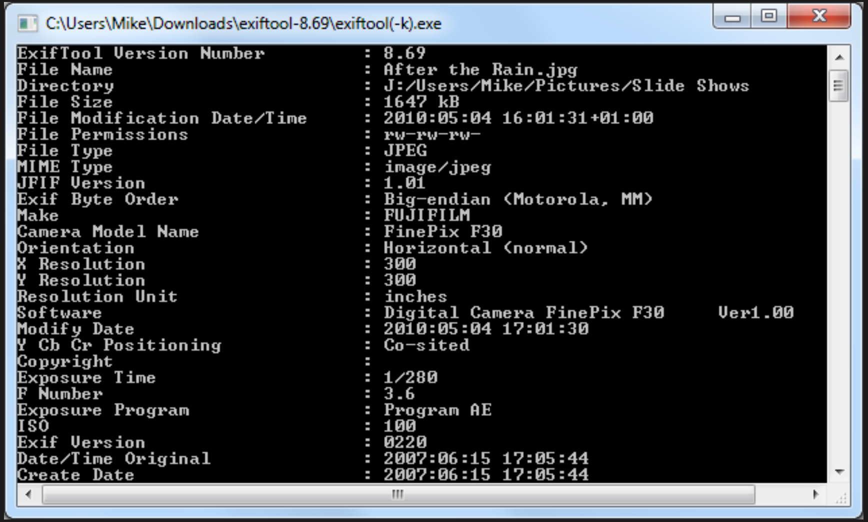
```
     x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF
0x   FF D8
+2         FF E0 00 10 .J .F .I .F \0 01 01 02 00 24
1x   00 24 00 00
+4               FF DB 00 43 00 01 01 01 01 01 01 01
..   .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
5x   01 01 01 01 01 01 01 01 01
+9                              FF C0 00 0B 08 00 38
6x   00 68 01 01 11 00
+6                     FF C4 00 29 00 01 01 01 01 00
7x   00 00 00 00 00 00 00 00 00 00 00 00 0B 04 0A 10
8x   01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9x   00
+1      FF DA 00 08 01 01 00 00 3F 00 EF E0 00 00 06
Ax   76 80 40 21 7F 74 02 05 FB C1 01 01 7F 70 10 08
Bx   5F DD 00 85 FD D0 08 5F DD 00 85 FD C0 04 02 17
Cx   F7 40 20 5F DC 40 20 17 F7 10 0F 5F C1 00 85 FD
Dx   D0 08 5F DC 10 08 5F DD 00 85 FD C6 74 04 17 F7
Ex   10 08 5F DC 04 02 05 FD C0 00 00 07
+C                                       FF D9
     x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF
```

	Fields	Values
Start Of Image	marker	FF D8
Application 0	marker/length	FF E0/16
	identifier	JFIF\0
	version	1.1
	units	2 (dots/cm)
	density	36x36
	thumbnail	0x0
Define Quantization Table	marker/length	FF DB/67
	destination	0 (luminance)
	table (8x8)	{1} (100% quality)
Start Of Frame 0	marker/length	FF C0/11
	precision	8
	line Nb	56
	samples/line	104
	components	1
	#1 factor 1x1 table 0 (LumY)	
Define Huffman Table	marker/length	FF C4/41
	class/dest.	0 (DC) / 0
	1 code of 1 bit	00
	1 code of 2 bits	0B
	1 code of 3 bits	04
	1 code of 4 bits	0A (no other code)
	class/dest.	1 (AC) / 0
	1 code of 1 bit	00 (no other code)
Start Of Scan	marker/length	FF DA/8
	components	1
	selector / DC, AC table	1 / 0, 0
	spectral select.	0..63
	successive approx.	00
	scan data	EF E0 .... 00 07
End Of Image	marker	FF D9

A JPEG FILE

Speaking of files... what's metadata?

- Metadata is data that provides information on data - i.e. where an image was taken, what camera it was taken on etc.
- Most files will have metadata, and whilst it isn't always useful in a CTF, it's worth taking a look for information.
- The best tool for extracting and viewing metadata for a CTF is **exiftool**.



```
ExifTool Version Number         : 8.69
File Name                       : After the Rain.jpg
Directory                       : J:/Users/Mike/Pictures/Slide Shows
File Size                       : 1647 kB
File Modification Date/Time     : 2010:05:04 16:01:31+01:00
File Permissions                : rw-rw-rw-
File Type                       : JPEG
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
Make                            : FUJIFILM
Camera Model Name               : FinePix F30
Orientation                     : Horizontal (normal)
X Resolution                    : 300
Y Resolution                    : 300
Resolution Unit                 : inches
Software                        : Digital Camera FinePix F30     Ver1.00
Modify Date                     : 2010:05:04 17:01:30
Y Cb Cr Positioning             : Co-sited
Copyright                       :
Exposure Time                   : 1/280
F Number                        : 3.6
Exposure Program                : Program AE
ISO                             : 100
Exif Version                    : 0220
Date/Time Original              : 2007:06:15 17:05:44
Create Date                     : 2007:06:15 17:05:44
```

More on Image Files

- Got a broken PNG? Try out **pngcheck** – a command line utility for checking what’s wrong with your images and helpfully identifying what you could fix to get it to open. http://libpng.org/pub/png/apps/pngcheck.html
- Sometimes, image editing is useful – so crack out Photoshop or Gimp and play around with filters and various settings. You can get Gimp here: https://www.gimp.org/
- Need to do more than just the basics? You can script your own image handling functions using PIL, the Python Image Library (now known as Pillow in newer versions). https://pypi.org/project/Pillow/

Analysing Archive Files

- Archive files, such as 7z, rar, tar, gz, and more, are a huge part of CTF challenges!
- Here are some important facts about them:
  - If you have a password-protected .zip, you can still view the file names and original filesizes of its contained files. This does not work on .rar or .7z!
  - Fcrackzip is a great tool for bruteforcing password-protected archive files.
  - There are plenty of command line utilities for checking zip file properties:
    - `zipdetails -v` prints detailed information on the file and its fields.
    - `zipinfo` lists information on the zip's contents.
    - `zip -F input.zip --out output.zip` and `zip -FF input.zip --out output.zip` attempt to repair corrupt zip files.
- Be sure to always check the file specifications if you're stuck on a challenge involving a zip file!

File Analysis Resources

- Handy Graphic Breakdowns of File Formats: https://github.com/corkami/pics/tree/master/binary
- File Signature Repository: https://www.garykessler.net/library/file_sigs.html
- exiftool: https://exiftool.org/
- ImageMagick: http://www.imagemagick.org/script/index.php

02

Memory Analysis

Now THIS is more like CSI

Challenge Structure

You'll be provided with a memory dump, disk image or forensic image of a machine. These challenges often provide you with direct questions you need to answer by mounting, examining and analysing images with various tools.

File Type 1: ISOs, CD-ROMs or .E01s

- If you're given a full disk image as part of a challenge, the goal will usually be to mount this image and then examine it. Mounting can be complicated but is well described here: https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/image-acquisition-and-mount
- Once mounted, you can use the tree command for a more graphical view of files, to see if anything sticks out. If it does, follow steps from file analysis to see what you can find!
- Sometimes, you might be looking for deleted files, hidden volumes, or unallocated space on a disk, and not a typical file. For deleted files, **extundelete** is a great option: http://extundelete.sourceforge.net/
- TestDisk is a great (but slightly complicated) command line tool for fixing corrupted or missing partition tables, or undeleting files: https://www.cgsecurity.org/wiki/TestDisk

Alternatively, Autopsy!

Autopsy is a free tool that provides a more graphical interface for mounting, examining and retrieving files from filesystems. Challenges involving searching for strings, file recovery or checking out Windows-specific forensic artefacts are much easier on Autopsy than in the command line!

It is relatively well-documented and you can find how-tos on it here: http://sleuthkit.org/autopsy/docs/user-docs/4.21.0/

File Type 2: Memdumps/.mem

- Memory dumps, or memdumps, are ”records” of live memory, taken from a running machine.
- This includes plenty of useful, short-term contents – such as clipboard contents, browsing history, and other artefacts that can often only be found in RAM.
- If you see a memdump, it will usually involve Volatility – a free and open-source tool that is available here: https://www.volatilityfoundation.org/releases
- It would take ages to go into the depths of how Volatility works, but if I needed to get clipboard contents for example, I would Google “volatility find clipboard contents” and follow the first page I see. Volatility is extensively used online!
- Note the differences between Volatility version 2 and version 3, and make sure you’re using the correct version for whatever tutorial you’re following!

03

Network Analysis

Last but not least, analysing network captures!

Challenge Structure

Network analysis challenges almost always revolve around PCAPs, or packet captures.

- Hidden messages in PCAPs
- "Packet carving" challenges
- Fixing corrupt PCAPs (sort of file analysis?)
- **In harder cases:** writing scripts to process packet captures

Wireshark Demo

Network Analysis Resources

- Wireshark + Docs: https://www.wireshark.org/ and https://www.wireshark.org/docs/
- Wireshark Cheat Sheet: https://www.comparitech.com/net-admin/wireshark-cheat-sheet/
- PCAPfix: http://f00l.de/hacking/pcapfix.php
- Dpkt for Python: https://dpkt.readthedocs.io/en/latest/

You now have two options!

Recommended Pico Challenges

Easy

- Wireshark doo dooo do doo... + Wireshark twoo twooo two twoo...
- Glory of the Garden

Medium

- Sleuthkit Apprentice
- WhitePages

Hard

- investigation_encoded_1 + investigation_encoded_2
- UnforgottenBits

Forensics Challenge

4 part challenge to test your skills in file/memory analysis.

Suitable for beginners!

Thank you for listening!

Slides will be on the CyberSoc Server after this session.