CMSC388U

FORENSICS

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Announcements

- Lecture #5 Recording
- Homework #5 due tomorrow
- Homework #6 will be released by tonight (pinky promise)
- Midterm next week!
    - Fine details to come (stay tuned to piazza & ELMS updates)
    - Review session?

What is digital forensics?

- "The recovery, extraction, and analysis of data from a storage media"
- **DFIR:** Digital Forensics and Incident Response
- Who and why?
  - **Law enforcement:** use DF to get information during an investigation
    - (e.g they seize a server and need to figure out what it was doing)
  - **Courts:** Legal discovery, prosecution, defense
  - **Malware analysts:** Figure out what malware did on a system, recover deleted files
  - **Blue Team/SOC Operators:** Responding to security incidents
  - **Pentesters:** Extract information that could be used in an offensive capacity
    - Passwords, hashes, usernames, etc...

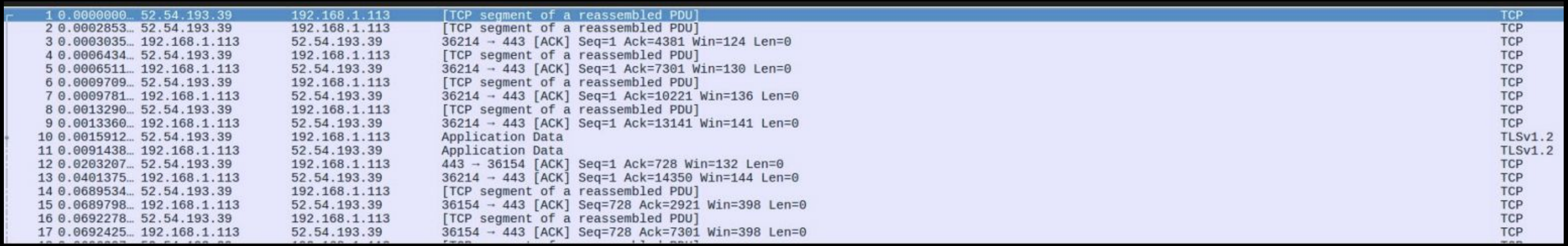
A

Data recovery from media

- We can't just copy files!
    - We want *deleted or damaged* files - can't be accessed typically
- We need to create an image
    - Byte-by-byte copy of the media
    - Look but don't touch - collection now, analysis later
    - **Common extensions:** .iso, .img, .dmg, .bin
        - (when you installed Kali Linux you used one of these!!)

Data recovery/extraction

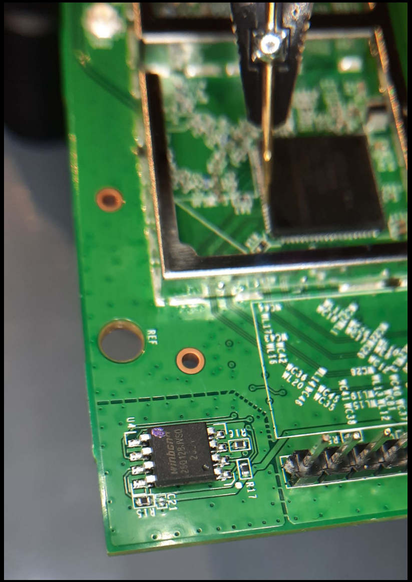
- Mediums that could be analyzed
  - **Physical media:** *HDDs, SSDs, thumbdrives, SIMs*
    - This includes partially destroyed or corrupted information
  - **Storage formats:** *disk images, archives, logs*
  - **Live sources:** *packet captures, memory dumps*

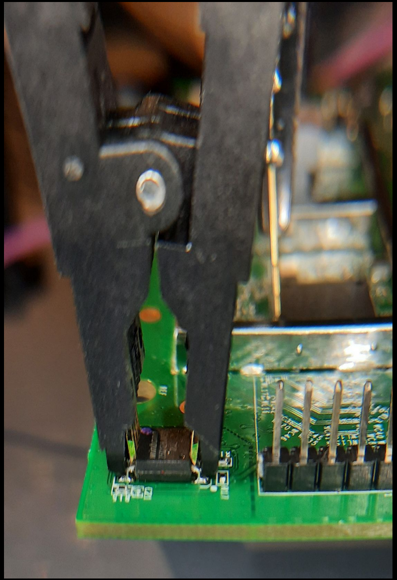


```
 1 0.0000000… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
 2 0.0002853… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
 3 0.0003035… 192.168.1.113    52.54.193.39      36214 → 443 [ACK] Seq=1 Ack=4381 Win=124 Len=0      TCP
 4 0.0006434… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
 5 0.0006511… 192.168.1.113    52.54.193.39      36214 → 443 [ACK] Seq=1 Ack=7301 Win=130 Len=0      TCP
 6 0.0009709… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
 7 0.0009781… 192.168.1.113    52.54.193.39      36214 → 443 [ACK] Seq=1 Ack=10221 Win=136 Len=0     TCP
 8 0.0013290… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
 9 0.0013360… 192.168.1.113    52.54.193.39      36214 → 443 [ACK] Seq=1 Ack=13141 Win=141 Len=0     TCP
10 0.0015912… 52.54.193.39     192.168.1.113     Application Data                                    TLSv1.2
11 0.0091438… 192.168.1.113    52.54.193.39      Application Data                                    TLSv1.2
12 0.0203207… 52.54.193.39     192.168.1.113     443 → 36154 [ACK] Seq=1 Ack=728 Win=132 Len=0       TCP
13 0.0401375… 192.168.1.113    52.54.193.39      36214 → 443 [ACK] Seq=1 Ack=14350 Win=144 Len=0     TCP
14 0.0689534… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
15 0.0689798… 192.168.1.113    52.54.193.39      36154 → 443 [ACK] Seq=728 Ack=2921 Win=398 Len=0    TCP
16 0.0692278… 52.54.193.39     192.168.1.113     [TCP segment of a reassembled PDU]                  TCP
17 0.0692425… 192.168.1.113    52.54.193.39      36154 → 443 [ACK] Seq=728 Ack=7301 Win=398 Len=0    TCP
```

Physical Extraction

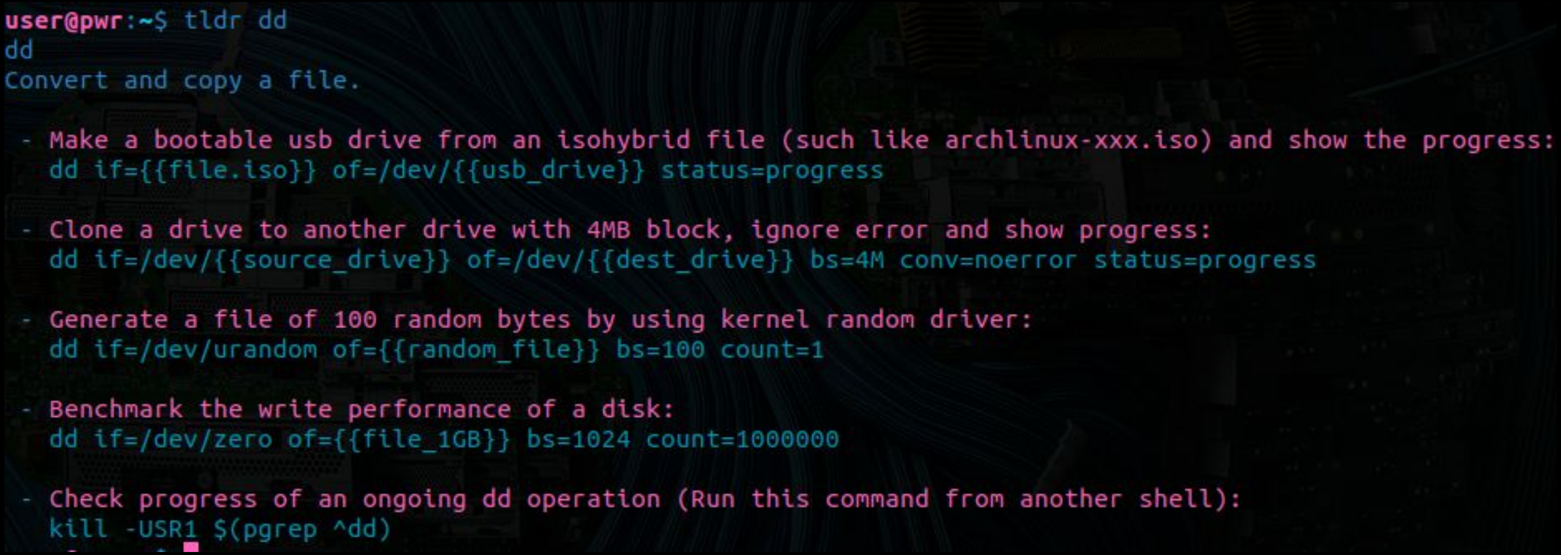
SIFT





Creating Images

- There are several graphical utilities for almost every OS
  - However the CLI based `dd` is the most common



```
user@pwr:~$ tldr dd
dd
Convert and copy a file.

 - Make a bootable usb drive from an isohybrid file (such like archlinux-xxx.iso) and show the progress:
   dd if={{file.iso}} of=/dev/{{usb_drive}} status=progress

 - Clone a drive to another drive with 4MB block, ignore error and show progress:
   dd if=/dev/{{source_drive}} of=/dev/{{dest_drive}} bs=4M conv=noerror status=progress

 - Generate a file of 100 random bytes by using kernel random driver:
   dd if=/dev/urandom of={{random_file}} bs=100 count=1

 - Benchmark the write performance of a disk:
   dd if=/dev/zero of={{file_1GB}} bs=1024 count=1000000

 - Check progress of an ongoing dd operation (Run this command from another shell):
   kill -USR1 $(pgrep ^dd)
```

WARNING

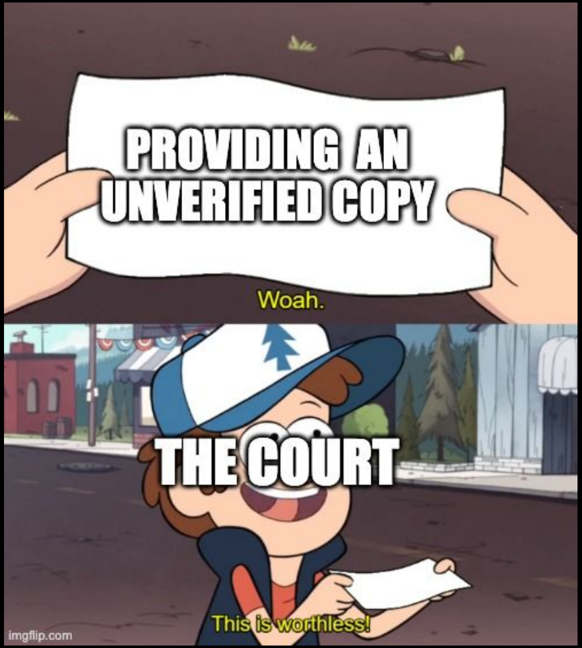
DD is also known lovingly as “disk destroyer”

Double, triple, quadruple check the command you’re going to run or you could SERIOUSLY screw up your computer

This is on the same tier as running sudo rm -rf *.

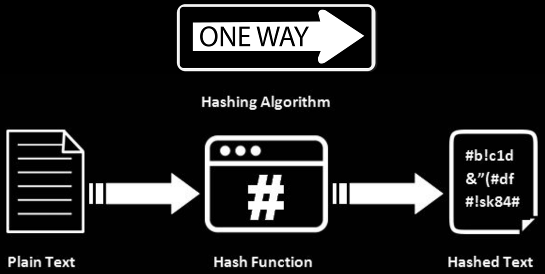
Data analysis

- Great! We have the data… but now what?
- Some DF laws to live by:
  - Start with a verified copy
  - Never work on the source (seriously)
  - If you can't verify the copy, its worthless
- **How do we copy and verify??**



Copy and verify!

- First step is to get a strong cryptographic hash of the image file
  - We'll touch more on what hashes are in crypto but for now don't worry too much about it
  - **Common hashes are:** md5 (weak), sha256, sha512, NTLM (windows)
    - `$ md5sum {filename}`
    - `$ sha256sum {filename}`
    - `$ sha512sum {filename}`
- Make sure to save the hash, without it the verification isn't valid
- Copying: You can use `dd` to make a copy or `cp`
- Make sure that the copy has the same hash as the original



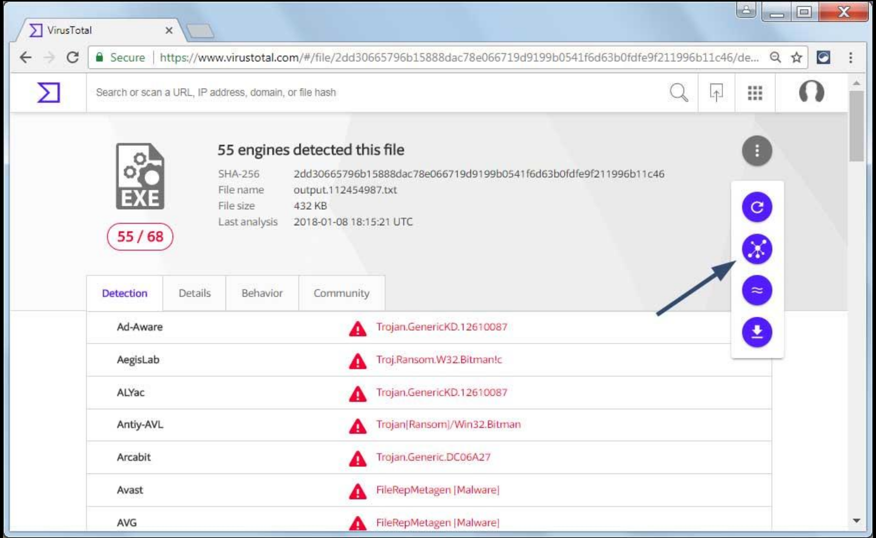
Copy and verify

- Why is this actually important?
  - Makes tampering difficult: everyone knows that the hash of the original file was so any changes will be very apparent
  - Not just about trust: keeps samples organized, prevents mixups
    - Lots of legal cases are lost because of unverified digital evidence (analysis was done on files with different hashes)

A

Hashing uses

- VirusTotal is a site that collections and analyzes urls, programs, etc
    - Malware!
- Users will upload samples and then compare hashes to see if new strains of malware have been created


VirusTotal
Secure
https://www.virustotal.com/#/file/2dd30665796b15888dac78e066719d9199b0541f6d63b0fdfe9f211996b11c46/de...
Search or scan a URL, IP address, domain, or file hash
EXE
55 / 68
Detection
Details
Behavior
Community

55 engines detected this file

SHA-256	2dd30665796b15888dac78e066719d9199b0541f6d63b0fdfe9f211996b11c46
File name	output.112454987.txt
File size	432 KB
Last analysis	2018-01-08 18:15:21 UTC

Ad-Aware	Trojan.GenericKD.12610087
AegisLab	Troj.Ransom.W32.Bitman!c
ALYac	Trojan.GenericKD.12610087
Antiy-AVL	Trojan[Ransom]/Win32.Bitman
Arcabit	Trojan.Generic.DC06A27
Avast	FileRepMetagen [Malware]
AVG	FileRepMetagen [Malware]

Analysis techniques

- How should we analyze our copy?
    - Depends on what we're looking for!
- Content-agnostic analysis
    - Using stuff from last lecture!
    - `file, strings, hexdump, etc.`
    - Generally less accurate, but more verbose
- Content-aware analysis
- Metadata analysis
- Steganographic analysis

Analysis: content-agnostic

- We don't always know what we're actually looking for…
  - Turn to generic tools!
- General gameplan:
  - Maintain a list of interesting signatures
  - Scan input for each signature
  - Give the user a list of offsets corresponding to signature matches

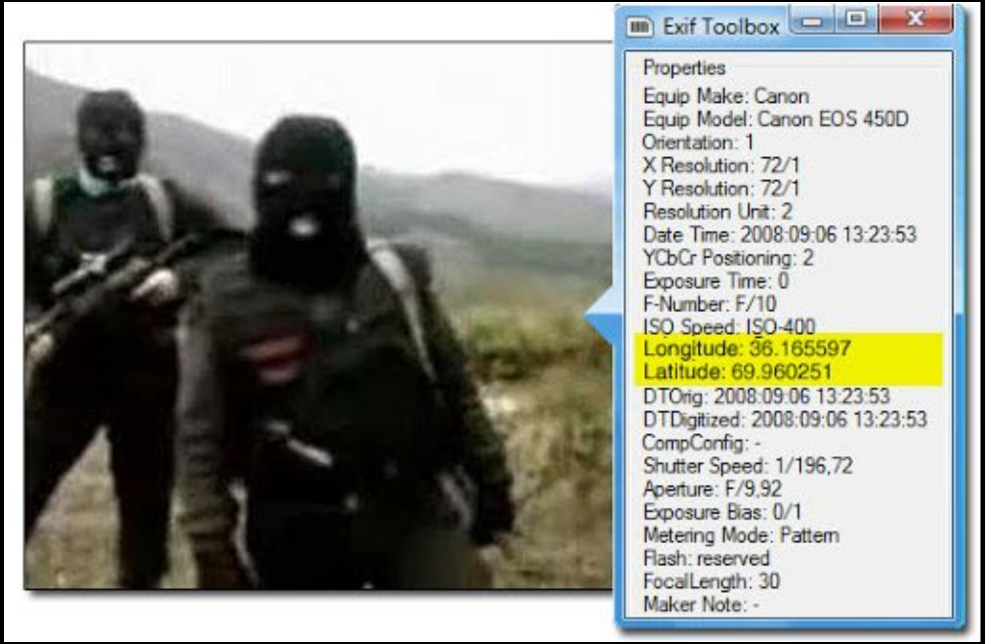
Content-aware analysis

- If we know the format of the file, we can analyse if more precisely!
- Fingerprinting: figure out who generated it based on HOW they generated it
  - Binaries: compilers have their own quirks which can give up info on what made something
- PDFs/rich documents: embedded scripts, embedded change logs, default field values, references to other files
- PCAP files: can give tons of network information
  - Will talk more about in Forensics II

Analysis: metadata

- Files have all kinds of metadata in them
  - Chain of authors, original computer
  - Specifically: PDFs, DOCs, JPG, PNG, MP4, etc...
- Images (JPG/PNG) are especially interesting
  - GPS tags, time taken tags
  - Device/vendor/software tags
    - Why might this be interesting??

oops...



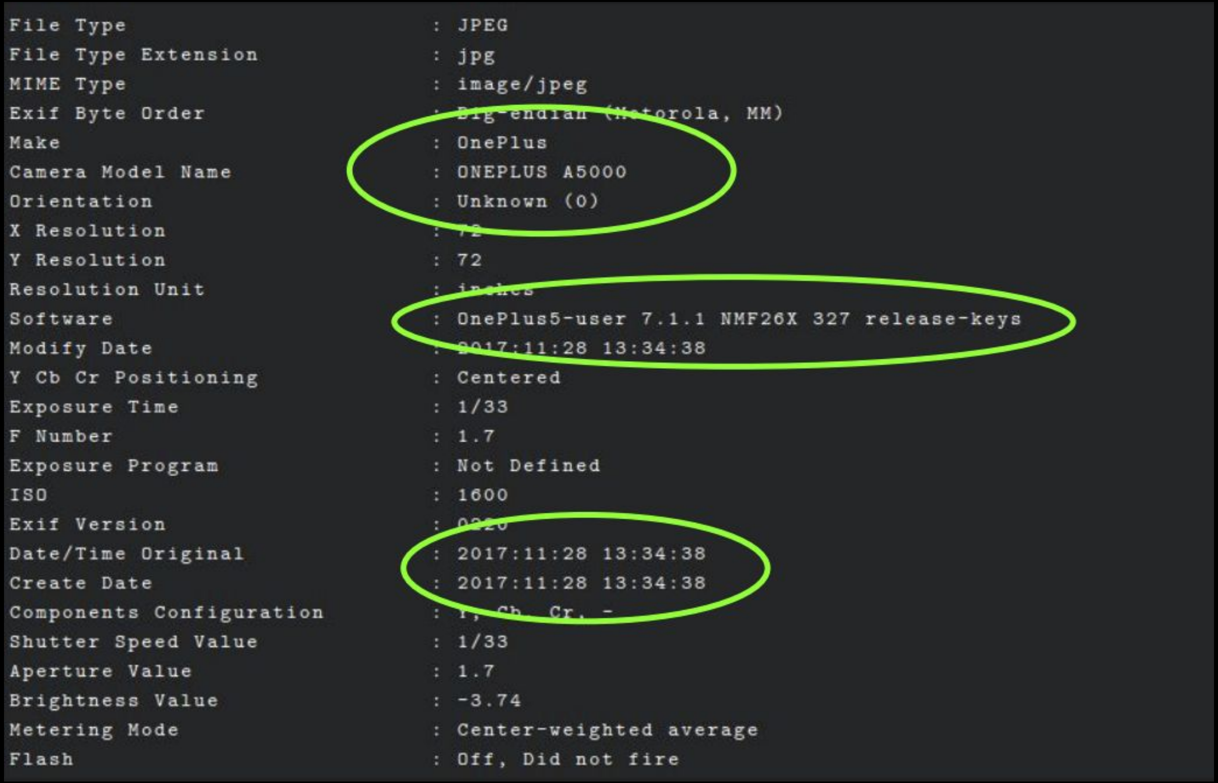
Exif Toolbox

Properties
Equip Make: Canon
Equip Model: Canon EOS 450D
Orientation: 1
X Resolution: 72/1
Y Resolution: 72/1
Resolution Unit: 2
Date Time: 2008:09:06 13:23:53
YCbCr Positioning: 2
Exposure Time: 0
F-Number: F/10
ISO Speed: ISO-400
Longitude: 36.165597
Latitude: 69.960251
DTOrig: 2008:09:06 13:23:53
DTDigitized: 2008:09:06 13:23:53
CompConfig: -
Shutter Speed: 1/196,72
Aperture: F/9,92
Exposure Bias: 0/1
Metering Mode: Pattern
Flash: reserved
FocalLength: 30
Maker Note: -

Exiftool

- `Exiftool` is great for both viewing and modifying image/audio/video metadata:
  - `$ exiftool {filename}` will dump all the metadata from a file
  - `$ exiftool -all= {filename}` clear all tags from an image
  - `$ exiftool -XMP-dc:Creator="Creator" "file name.extension"`
    - Allows you to change copyright
  - List not extensive, look through the man pages for all the other fun stuff

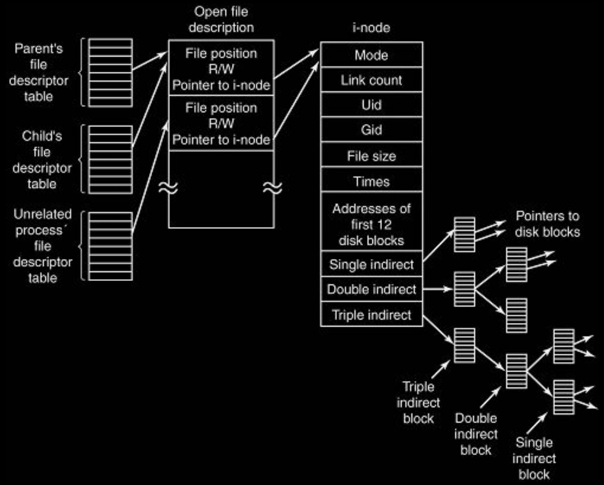
$ exiftool image.jpg

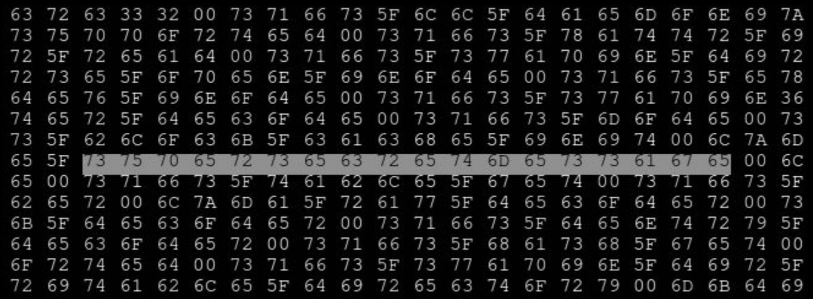


```
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
Exif Byte Order                 : Big-endian (Motorola, MM)
Make                            : OnePlus
Camera Model Name               : ONEPLUS A5000
Orientation                     : Unknown (0)
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : OnePlus5-user 7.1.1 NMF26X 327 release-keys
Modify Date                     : 2017:11:28 13:34:38
Y Cb Cr Positioning             : Centered
Exposure Time                   : 1/33
F Number                        : 1.7
Exposure Program                : Not Defined
ISO                             : 1600
Exif Version                    : 0220
Date/Time Original              : 2017:11:28 13:34:38
Create Date                     : 2017:11:28 13:34:38
Components Configuration        : Y, Cb, Cr, -
Shutter Speed Value             : 1/33
Aperture Value                  : 1.7
Brightness Value                : -3.74
Metering Mode                   : Center-weighted average
Flash                           : Off, Did not fire
```

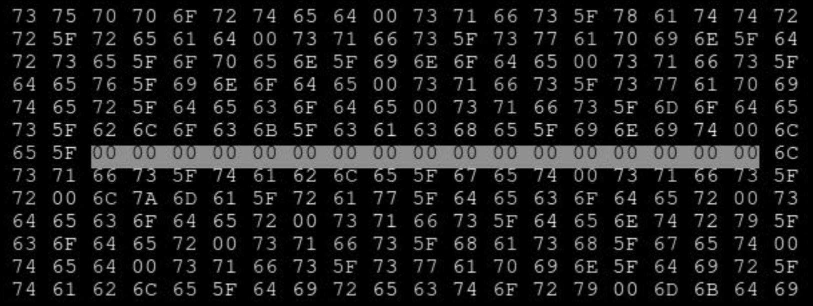
Undelete

- OS no know, no?
- What is deleting a file, actually?
  - Zeroing out disk area?
  - Overwriting the file contents?
- `Extundelete` tool is packaged with kali





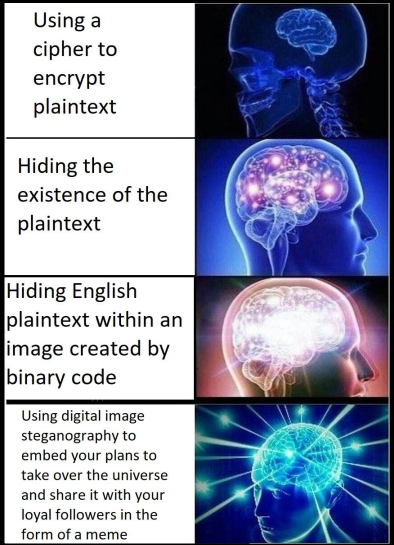
```
63 72 63 33 32 00 73 71 66 73 5F 6C 6C 5F 64 61 65 6D 6F 6E 69 7A
73 75 70 70 6F 72 74 65 64 00 73 71 66 73 5F 78 61 74 74 72 5F 69
72 5F 72 65 61 64 00 73 71 66 73 5F 73 77 61 70 69 6E 5F 64 69 72
72 73 65 5F 6F 70 65 6E 5F 69 6E 6F 64 65 00 73 71 66 73 5F 65 78
64 65 76 5F 69 6E 6F 64 65 00 73 71 66 73 5F 73 77 61 70 69 6E 36
74 65 72 5F 64 65 63 6F 64 65 00 73 71 66 73 5F 6D 6F 64 65 00 73
73 5F 62 6C 6F 63 6B 5F 63 61 63 68 65 5F 69 6E 69 74 00 6C 7A 6D
65 5F 73 75 70 65 72 73 65 63 72 65 74 6D 65 73 73 61 67 65 00 6C
65 00 73 71 66 73 5F 74 61 62 6C 65 5F 67 65 74 00 73 71 66 73 5F
62 65 72 00 6C 7A 6D 61 5F 72 61 77 5F 64 65 63 6F 64 65 72 00 73
6B 5F 64 65 63 6F 64 65 72 00 73 71 66 73 5F 64 65 6E 74 72 79 5F
64 65 63 6F 64 65 72 00 73 71 66 73 5F 68 61 73 68 5F 67 65 74 00
6F 72 74 65 64 00 73 71 66 73 5F 73 77 61 70 69 6E 5F 64 69 72 5F
72 69 74 61 62 6C 65 5F 64 69 72 65 63 74 6F 72 79 00 6D 6B 64 69
```



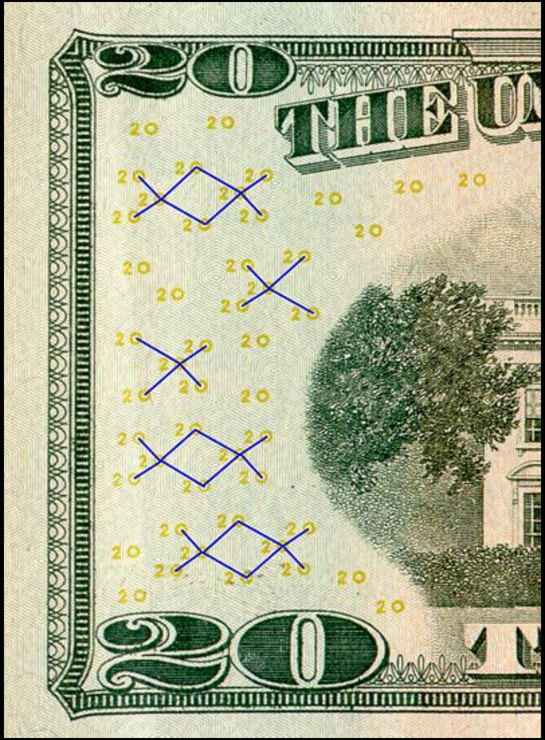
```
73 75 70 70 6F 72 74 65 64 00 73 71 66 73 5F 78 61 74 74 72
72 5F 72 65 61 64 00 73 71 66 73 5F 73 77 61 70 69 6E 5F 64
72 73 65 5F 6F 70 65 6E 5F 69 6E 6F 64 65 00 73 71 66 73 5F
64 65 76 5F 69 6E 6F 64 65 00 73 71 66 73 5F 73 77 61 70 69
74 65 72 5F 64 65 63 6F 64 65 00 73 71 66 73 5F 6D 6F 64 65
73 5F 62 6C 6F 63 6B 5F 63 61 63 68 65 5F 69 6E 69 74 00 6C
65 5F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6C
73 71 66 73 5F 74 61 62 6C 65 5F 67 65 74 00 73 71 66 73 5F
72 00 6C 7A 6D 61 5F 72 61 77 5F 64 65 63 6F 64 65 72 00 73
64 65 63 6F 64 65 72 00 73 71 66 73 5F 64 65 6E 74 72 79 5F
63 6F 64 65 72 00 73 71 66 73 5F 68 61 73 68 5F 67 65 74 00
74 65 64 00 73 71 66 73 5F 73 77 61 70 69 6E 5F 64 69 72 5F
74 61 62 6C 65 5F 64 69 72 65 63 74 6F 72 79 00 6D 6B 64 69
```

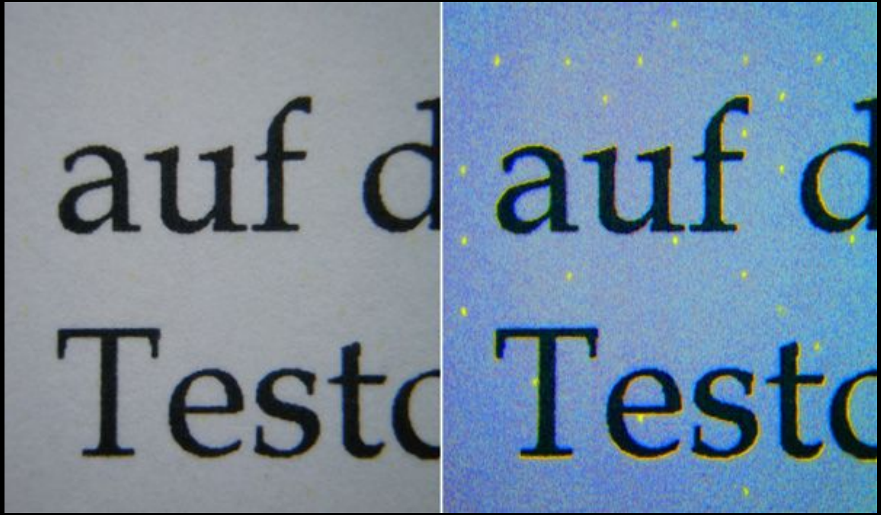
Steganography

- Steganography is the practice of concealing information within other information
  - Encoding text in RGB values of an image, opcodes of executable
  - Often used for physical tracking/linking
    - Printer dots (link paper documents to a printer)
    - Anti-counterfeiting (photocopying money)
    - Anti-piracy (detects attempts to duplicate film)
    - User tracking (screenshot watermarking)
- It is NOT encryption, but can be used to store encrypted information


Using a cipher to encrypt plaintext
Hiding the existence of the plaintext
Hiding English plaintext within an image created by binary code
Using digital image steganography to embed your plans to take over the universe and share it with your loyal followers in the form of a meme

Analysis techniques: steg

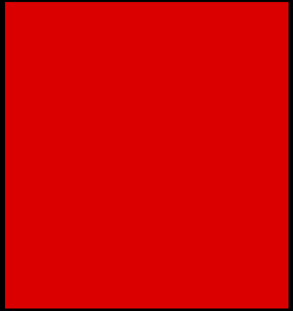


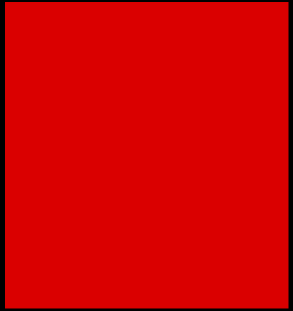


A

Analysis techniques: Steg

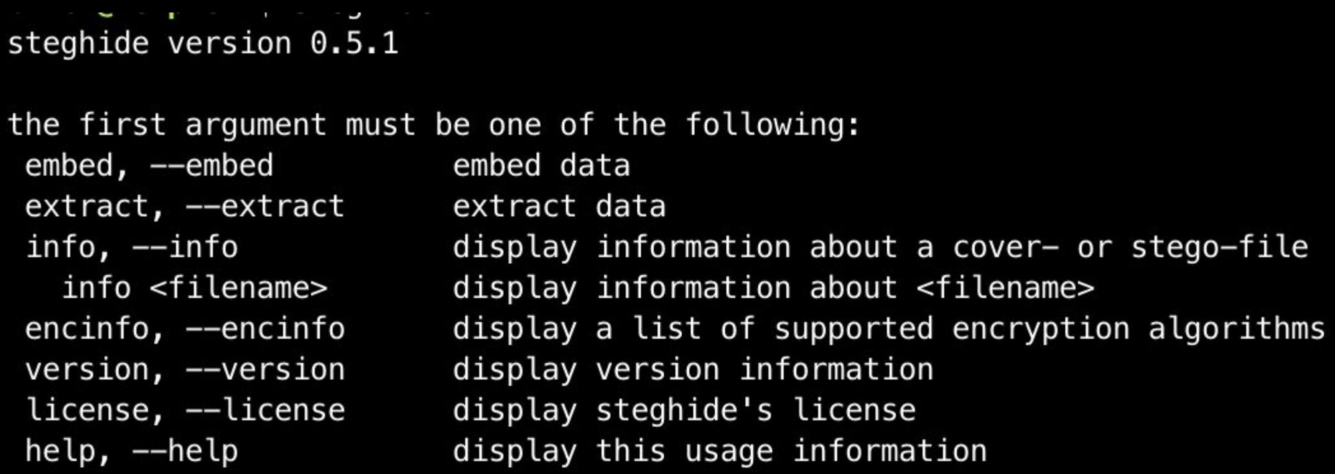
- How does image steg work?
  - **24-bit RGB colorspace:** 8 bits per red/green/blue
    - 1-bit changes to each color -> 3 bits/px ~> 3px/byte
  - **32-bit RGBA colorspace:** 8 bits per red/green/blue/alpha
    - 1-bit changes to each color -> 4bits/px ~> 2px/byte
  - Works because 1-bit chances to color are almost imperceptible





Stego techniques: steghide

- `steghide` can be used to hide data in the RGB/sound values of common image/audio formats
  - JPEG, BMP, WAV, AU
  - Data can be password protected



```
steghide version 0.5.1

the first argument must be one of the following:
 embed, --embed          embed data
 extract, --extract      extract data
 info, --info            display information about a cover- or stego-file
   info <filename>       display information about <filename>
 encinfo, --encinfo      display a list of supported encryption algorithms
 version, --version      display version information
 license, --license      display steghide's license
 help, --help            display this usage information
```