Digital Forensics Workshop

BSidesPDX 2023

Who are we?

Portland State University Cybersecurity Club

Our mission is to promote security culture, ethics, research, ongoing education, and development of safer code through playing in Capture the Flag competitions.

PSU Cybersecurity Club

Open to students of **all** majors!

Weekly training meetings!

CTF competitions on weekends!

Upcoming Events:

- National Cyber League
- DoE CyberForce Competition

Find us in the local security community:

- BSidesPDX, Rainsec, OWASP, Def Con, and more...

Travis Noyes

Grad student, Computer Science / Security

Cybersecurity club president

Infosec team intern for PSU

Lifelong learner or something

Lance Miller

PSU Grad Student in the Computer Science Department

Cyber Security Enthusiast

Officer Portland State Cybersecurity Club

David Baker-Robinson

PSU Grad Student in CS

Works for CAT (Computer Action Team)

Enjoys RE (Reverse Engineering) CTF’s

Excited about CyberForce and NCL

Cybersecurity Graduate Certificate

Portland State
UNIVERSITY

CS 591	Introduction to Computer Security
CS 595	Web and Cloud Security
CS 554	Principles of Software Engineering
CS 555	Software Specification and Verification
CS 556	Software Implementation and Testing
CS 576	Computer Security Seminar
CS 585	Cryptography
CS 592	Malware Reverse Engineering
CS 593	**Digital Forensics**
CS 594	Internetworking Protocols
CS 596	Network Security

Coming up

What is digital forensics?

Files and file systems

Tools

File carving demo

Metadata.. and more

CTF

What is digital forensics?

“Forensic science is the use of scientific methods or expertise to investigate crimes or examine evidence that might be presented in a court of law.”

National Institute of Standards and Technology (NIST)

What is *digital* forensics?

Focusing on the collection and examination of data stored electronically

Sources could be from:

- Computers
- Cell phones
- External drives
- Remote storage
- Networks

and much more!

Who needs digital forensics?

Law enforcement,

Companies,

Lawyers,

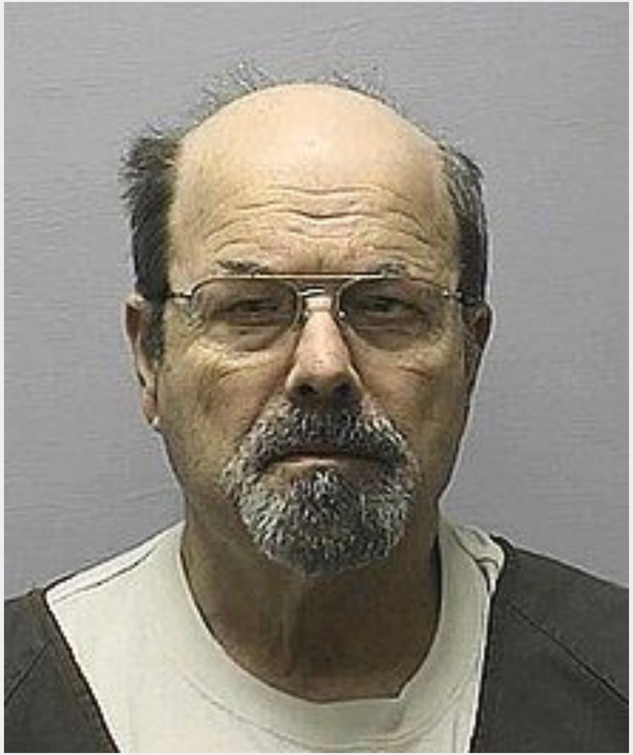
... Everybody!

Digital forensics in real life

The BTK serial killer

- Taunted law enforcement and reporters with pictures and puzzles

- Tracked a floppy disk to his church and found him guilty



Digital forensics in real life

Murdaugh Murders

- Prominent South Carolina attorney accused of killing his wife and son at their home
- He denied being there at the time of the murder
- Secret Service gained access to son's phone and found video evidence proving he lied

Digital forensics in our community

Washington County
Digital Forensics Lab

Northwest Regional
Computer Forensics Lab

Files and File Systems

What is an artifact?

Stuff left behind to discover

Operating System

- Windows event log, registry, dmesg, /var/log

File systems

- Access/modification/creation timestamps, metadata

Digital Artifacts

Disk artifacts

- Physical remnants left by manipulation of the file system
- Different from file system artifacts
- Caches hold data before the magnets actually write things, even though it can report that the write was successful

Digital Artifacts

Higher level

- Browsers, internet history, saved passwords
- Documents / Media (photos, videos, audio)
- Databases, often applications keep these for their internal data
- General application data
    - Very specific, like from Steam, or Microsoft Teams, etc

What are files?

Stored information on a computer

- Basically everything is a file

Size, timestamp, source, type, etc

- Common metadata attached to files
- Metadata + data = file

File types

Media:	Images, videos, audio recordings
Documents:	PDFs, Word / Excel, HTML
Executables:	Applications, scripts, malware

Extension vs file format

File extension

- .jpg, .pdf, etc
- Hints at the type of file

File format

- The actual way data stored is encoded
- Follows a specification for each type of file

Headers and hexadecimal

Each file is a sequence of bytes

Viewable through tools like **`xxd`**

JPEG images

- Start with **`FF D8`**
- End with **`FF D9`**

File systems (Windows)

Master file tables

FAT - File Allocation Table - Windows

NTFS - New Technology File System - Windows

File Systems (Linux / UNIX)

Files are streams of bytes, kept track of as inodes

ZFS - Zettabyte File System - Linux/ MacOsx

EXT - Extended Filesystem - Linux

Allocated vs Unallocated space

- File system tracks what storage is in use or free

- Blocks previously written are not immediately overwritten when released

- Artifacts can be collected if you’re quick enough

Partition maps

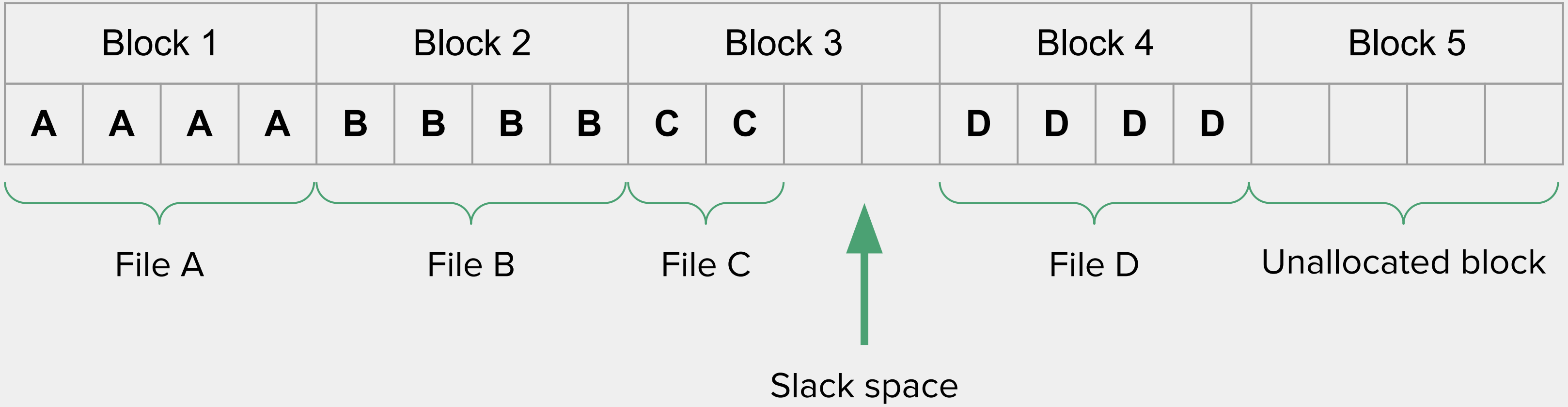
- MBR (Master Boot Record), doesn’t work in UEFI, it’s old school. BIOS only

- GPT (GUID), EFI map

- APFSX (apple computers)

Sparse files

Slack space



Storage hardware types

HDD	ATAPI parallel interface / SCSI or SAS (serial attached SCSI)
SSD	SATA serial interface
NVME	PCIe / M.2
USB Flash	
SD card / CF card	
DVD / BluRay	

Differences in shelf stability, power needs, random data corruption, etc

Disk Images

Types of image files

- Raw Single (*.img, *.dd, *.raw, *.bin)
- Raw Split (*.001, *.002, *.aa, *.ab, etc)
- and many more

Write blockers

- Software vs hardware

Tools

Tools

Ubuntu Linux VM

Linux forensics tools are easy to use

Lots of options for host operating systems

- Windows:	VMWare, VirtualBox, WSL2
- MacOS:	UTM, Parallels

Tools

dd / dc3dd

```
    $ sudo apt install dc3dd
```

AKA Disk Destroyer

- Directly interact with block devices
- Backup partitions or backup entire linux disk
- Wiping a hard drive (overwrite with zeros)

Tools

sha256sum

```
    $ sha256sum <filename>
```

Check the hash of a disk image or file download to make sure it is correct

Especially helpful if you can’t use a package manager for a certain set software and have to manually create it with make utility

Tools

binwalk

```
$ sudo apt install binwalk
```

- Used to search for embedded files and executable code
- Relies on the use of Magic numbers and other file statistics
- Find entropy in files, extract files, search for custom signatures

Tools

xxd

```
    $ xxd <filename>
```

exiftool

```
    $ sudo apt install libimage-exiftool-perl
    $ exiftool <filename>
```

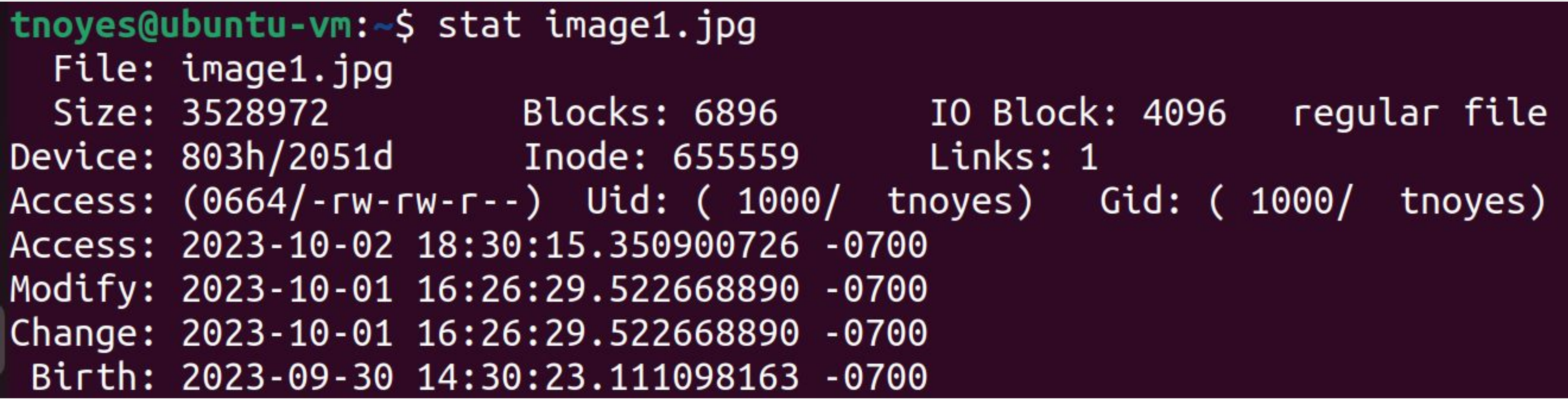
xmllint

```
    $ sudo apt install libxml2-utils
    $ xmllint --format <filename>
```

Tools

stat

```
$ stat <filename>
```

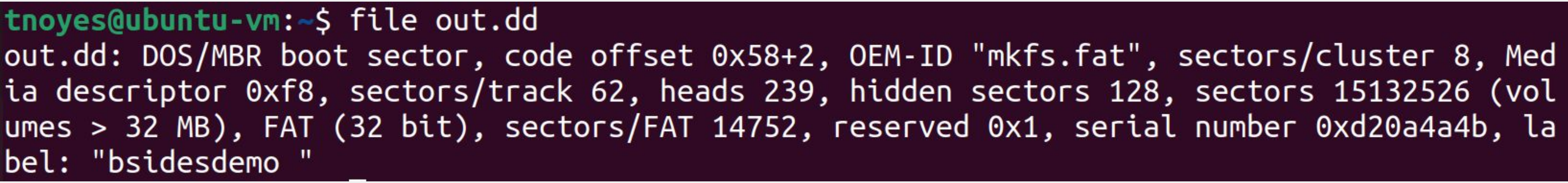


```
tnoyes@ubuntu-vm:~$ stat image1.jpg
  File: image1.jpg
  Size: 3528972         Blocks: 6896       IO Block: 4096   regular file
Device: 803h/2051d      Inode: 655559      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  tnoyes)   Gid: ( 1000/  tnoyes)
Access: 2023-10-02 18:30:15.350900726 -0700
Modify: 2023-10-01 16:26:29.522668890 -0700
Change: 2023-10-01 16:26:29.522668890 -0700
 Birth: 2023-09-30 14:30:23.111098163 -0700
```

Tools

file

```
$ file <filename>
```



```
tnoyes@ubuntu-vm:~$ file out.dd
out.dd: DOS/MBR boot sector, code offset 0x58+2, OEM-ID "mkfs.fat", sectors/cluster 8, Med
ia descriptor 0xf8, sectors/track 62, heads 239, hidden sectors 128, sectors 15132526 (vol
umes > 32 MB), FAT (32 bit), sectors/FAT 14752, reserved 0x1, serial number 0xd20a4a4b, la
bel: "bsidesdemo "
```

Tools

find

grep

Regular expressions

Tools

Sleuthkit

```
$ sudo apt-get install sleuthkit
```

Collection of tools to extract files from a variety of file types (different forms of storage)

fls Lists both unallocated and allocated files of a file system

icat Copies image file with specified inode to standard output

Mmls Lists where partitions begin and end

- Often used in combination with dd

Fancy Tools

Autopsy

- Industry standard
- Open source, very extensible
- Provides a graphical interface for tools like sleuthkit

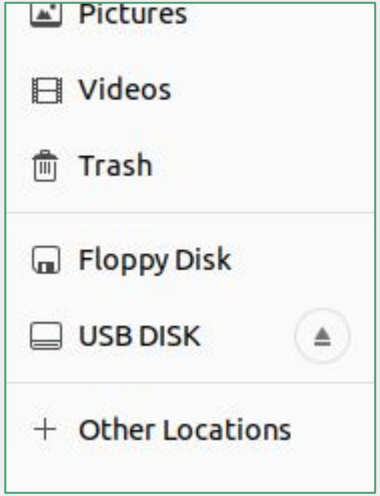
File Carving Demo

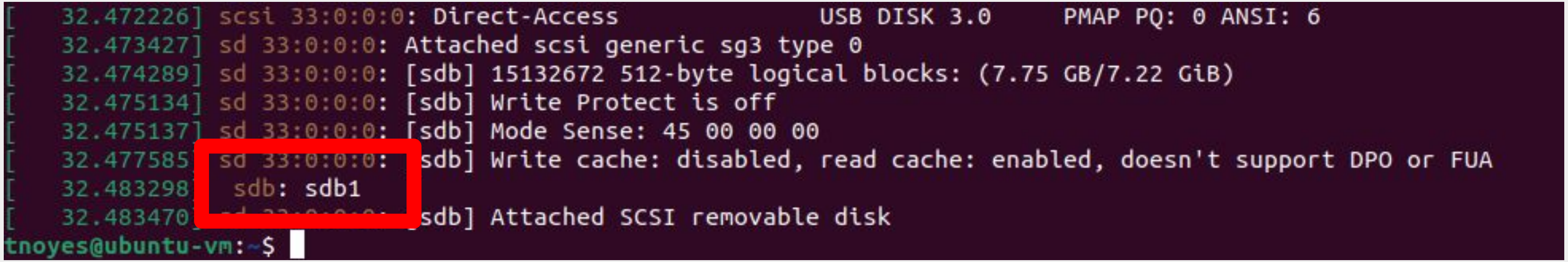
Connect USB device to Ubuntu VM

Plug in the device to an open USB port

Open Terminal and look for the device name

`$ sudo dmesg`   or   `$ sudo fdisk -l`



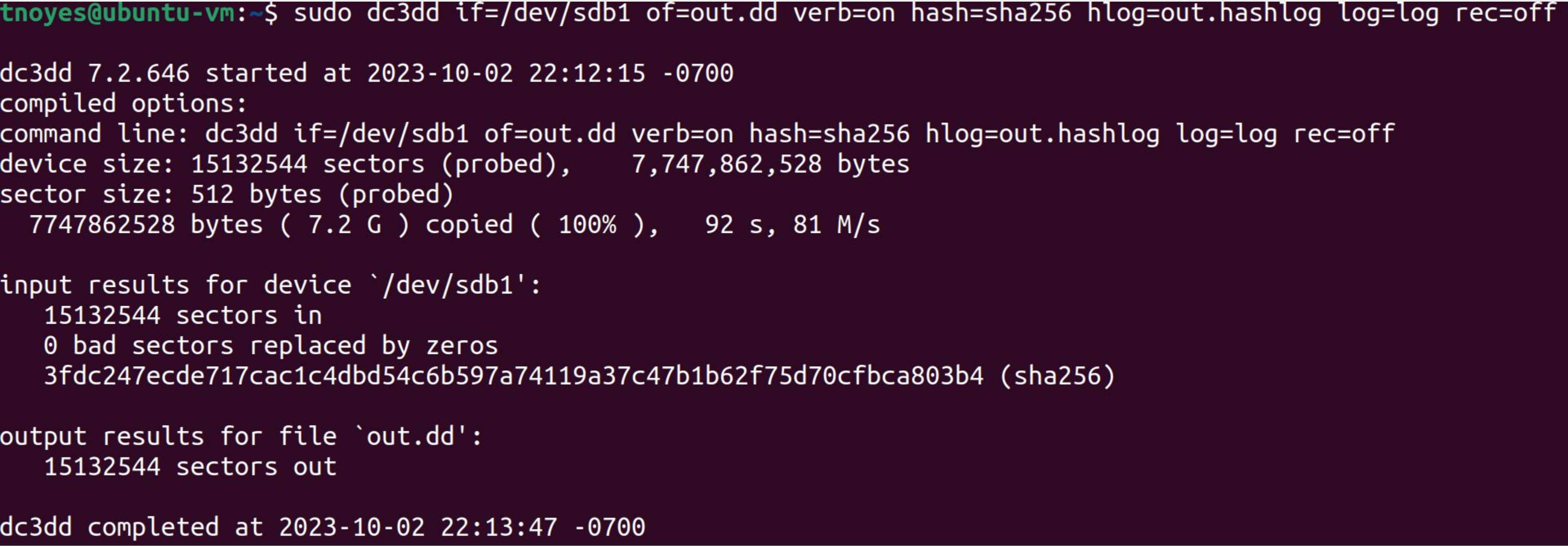


```
[   32.472226] scsi 33:0:0:0: Direct-Access           USB DISK 3.0     PMAP PQ: 0 ANSI: 6
[   32.473427] sd 33:0:0:0: Attached scsi generic sg3 type 0
[   32.474289] sd 33:0:0:0: [sdb] 15132672 512-byte logical blocks: (7.75 GB/7.22 GiB)
[   32.475134] sd 33:0:0:0: [sdb] Write Protect is off
[   32.475137] sd 33:0:0:0: [sdb] Mode Sense: 45 00 00 00
[   32.477585] sd 33:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[   32.483298]  sdb: sdb1
[   32.483470] sd 33:0:0:0: [sdb] Attached SCSI removable disk
tnoyes@ubuntu-vm:~$ 
```

Creating a disk image

Use dc3dd to create a forensic copy of the external storage device

```
$ sudo dc3dd if=/dev/sdb1 of=out.dd verb=on hash=sha256 hlog=out.hashlog log=log rec=off
```

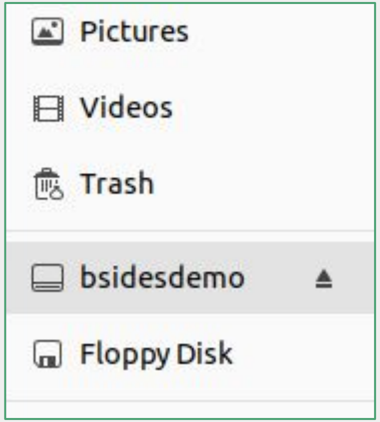


```
tnoyes@ubuntu-vm:~$ sudo dc3dd if=/dev/sdb1 of=out.dd verb=on hash=sha256 hlog=out.hashlog log=log rec=off

dc3dd 7.2.646 started at 2023-10-02 22:12:15 -0700
compiled options:
command line: dc3dd if=/dev/sdb1 of=out.dd verb=on hash=sha256 hlog=out.hashlog log=log rec=off
device size: 15132544 sectors (probed),    7,747,862,528 bytes
sector size: 512 bytes (probed)
  7747862528 bytes ( 7.2 G ) copied ( 100% ),   92 s, 81 M/s

input results for device `/dev/sdb1':
   15132544 sectors in
   0 bad sectors replaced by zeros
   3fdc247ecde717cac1c4dbd54c6b597a74119a37c47b1b62f75d70cfbca803b4 (sha256)

output results for file `out.dd':
   15132544 sectors out

dc3dd completed at 2023-10-02 22:13:47 -0700
```

Mount the Image

```
$ mkdir ~/out
```

```
$ sudo mount -o loop,ro,noexec out.dd ~/out
```



What do the flags mean?

-o (options)

loop mount using an unused unspecified loop device

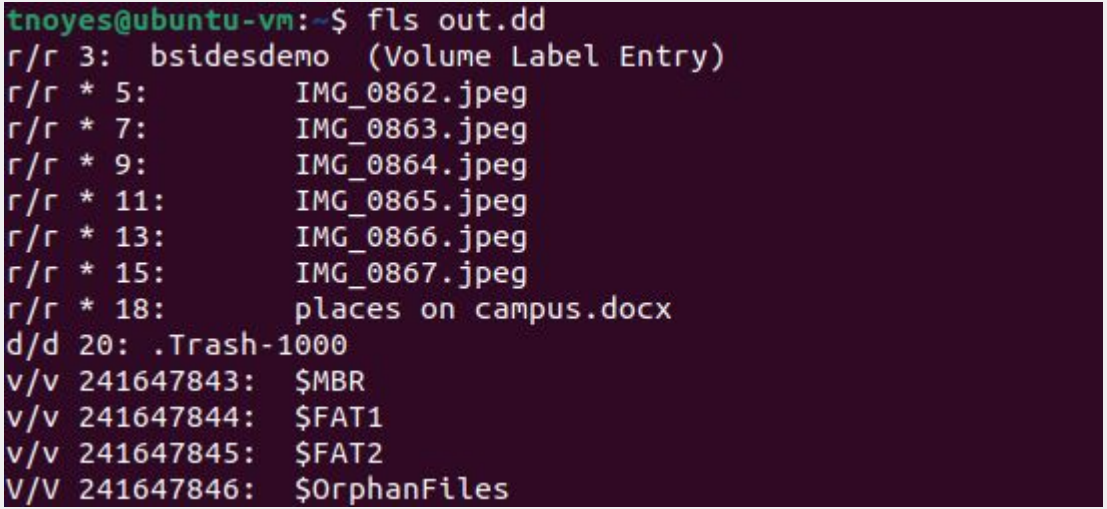
ro mount the filesystem as read only

noexec disables execution of binaries on the mounted filesystem

Start Sleuthing

Use fls (part of sleuthkit) to investigate the image contents, if any exist

```
$ fls out.dd
```



```
tnoyes@ubuntu-vm:~$ fls out.dd
r/r 3:  bsidesdemo  (Volume Label Entry)
r/r * 5:        IMG_0862.jpeg
r/r * 7:        IMG_0863.jpeg
r/r * 9:        IMG_0864.jpeg
r/r * 11:       IMG_0865.jpeg
r/r * 13:       IMG_0866.jpeg
r/r * 15:       IMG_0867.jpeg
r/r * 18:       places on campus.docx
d/d 20: .Trash-1000
v/v 241647843:  $MBR
v/v 241647844:  $FAT1
v/v 241647845:  $FAT2
V/V 241647846:  $OrphanFiles
```

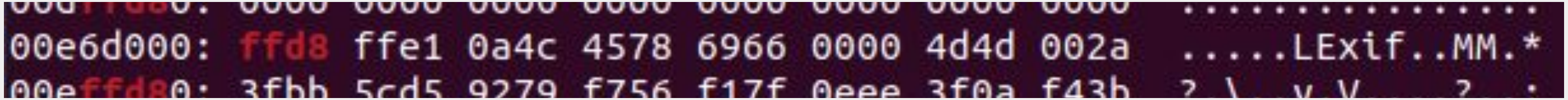
Manual File Carving (jpeg)

JPEG magic numbers

Header	**`0xffd8`**
Footer	**`0xffd9`**

Look for any JPEG file headers in the disk image

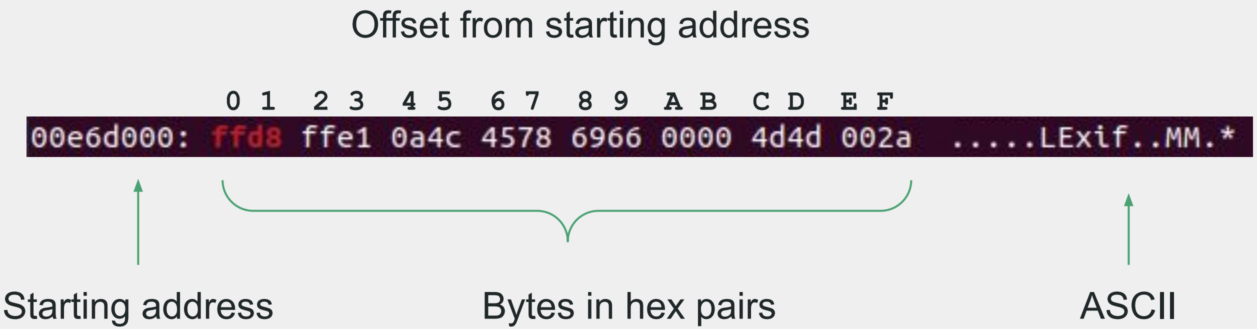
```
$ xxd out.dd | grep -m 30 “ffd8”
```



```
00dffd80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00e6d000: ffd8 ffe1 0a4c 4578 6966 0000 4d4d 002a  .....LExif..MM.*
00effd80: 3fbb 5cd5 9279 f756 f17f 0eee 3f0a f43b  ?.\..y.V....?..:
```

Manual File Carving (jpeg)

How to read xxd + grep output



Offset from starting address

```
           0 1  2 3  4 5  6 7  8 9  A B  C D  E F
00e6d000: ffd8 ffe1 0a4c 4578 6966 0000 4d4d 002a  .....LExif..MM.*
```

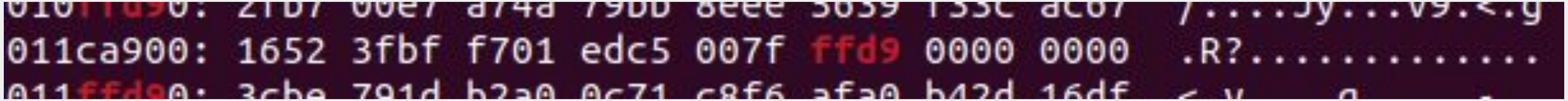
Starting address

Bytes in hex pairs

ASCII

Manual File Carving (jpeg)

Repeat for the footer



```
010ffd90: 2fb7 00e7 a74a 79bb 8eee 5639 f53c ac67  /....Jy...V9.<.g
011ca900: 1652 3fbf f701 edc5 007f ffd9 0000 0000  .R?.............
011ffd90: 3cbe 791d b2a0 0c71 c8f6 afa0 b42d 16df  <.y....q.....-.
```

First byte	**0x00E6d000**
Last byte	**0x011CA900 + 0xC (12 bytes) = 0x011CA90C**

Converting starting and end bytes from hex to decimal

First byte	->	15126528
Last byte	->	18655500

Manual File Carving (jpeg)

Calculate the file size in bytes by subtracting the end from the starting byte

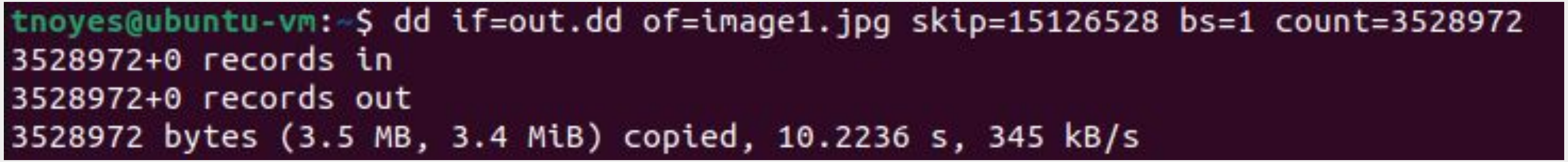
18655500	−	**15126528**	=	**3528972**
Last byte		First byte		File size

3528972 bytes (~3.5 MB)

Manual File Carving (jpeg)

Extract the file with dd

```
$ dd if=out.dd of=image1.jpg skip=15126528 bs=1 count=3528972
```



```
tnoyes@ubuntu-vm:~$ dd if=out.dd of=image1.jpg skip=15126528 bs=1 count=3528972
3528972+0 records in
3528972+0 records out
3528972 bytes (3.5 MB, 3.4 MiB) copied, 10.2236 s, 345 kB/s
```

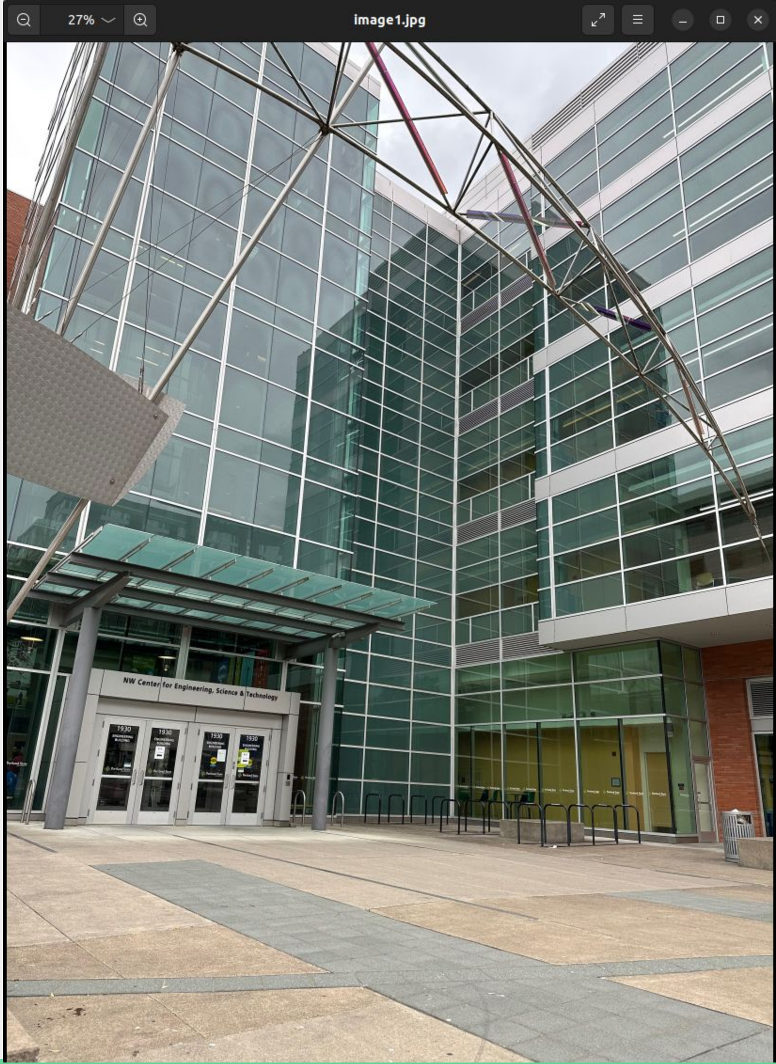
skip This is the starting byte in decimal format we calculated earlier

count The total number of bytes in the file size from earlier

Manual File Carving (jpeg)

View the extracted image

Can you guess where this is?

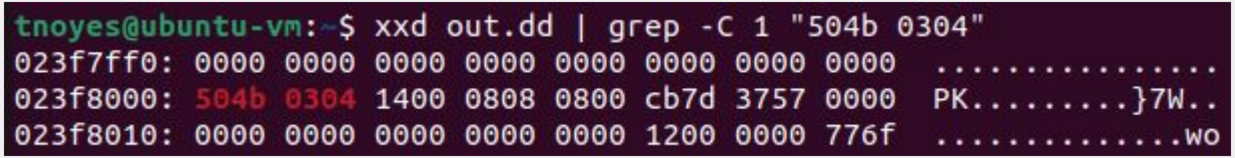


Manual File Carving (.docx)

Header `50 4B 03 04 14 00`

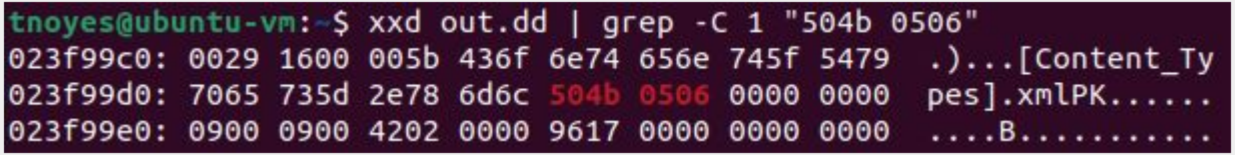
Footer `50 4B 05 06` (plus 18 bytes after)

```
$ xxd out.dd | grep -C 1 “504b 0304”
```



```
tnoyes@ubuntu-vm:~$ xxd out.dd | grep -C 1 "504b 0304"
023f7ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
023f8000: 504b 0304 1400 0808 0800 cb7d 3757 0000  PK.........}7W..
023f8010: 0000 0000 0000 0000 0000 1200 0000 776f  ..............wo
```

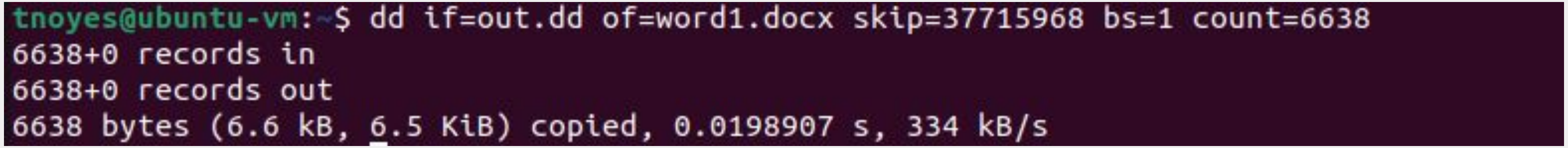
```
$ xxd out.dd | grep -C 1 “504b 0506”
```



```
tnoyes@ubuntu-vm:~$ xxd out.dd | grep -C 1 "504b 0506"
023f99c0: 0029 1600 005b 436f 6e74 656e 745f 5479  .)...[Content_Ty
023f99d0: 7065 735d 2e78 6d6c 504b 0506 0000 0000  pes].xmlPK......
023f99e0: 0900 0900 4202 0000 9617 0000 0000 0000  ....B...........
```

Manual File Carving (.docx)

Start	**0x023f8000**	37715968 (decimal)
End	**0x023F99E0** + **0xE** = **0x23F99EE**	37722606 (decimal)
File size	6638 bytes (~6.6kB)	



```
tnoyes@ubuntu-vm:~$ dd if=out.dd of=word1.docx skip=37715968 bs=1 count=6638
6638+0 records in
6638+0 records out
6638 bytes (6.6 kB, 6.5 KiB) copied, 0.0198907 s, 334 kB/s
```

Wait, are you serious!?
Isn’t there some easier way?

(what I probably said the first time I learned this stuff)

Automated File Carving

foremost

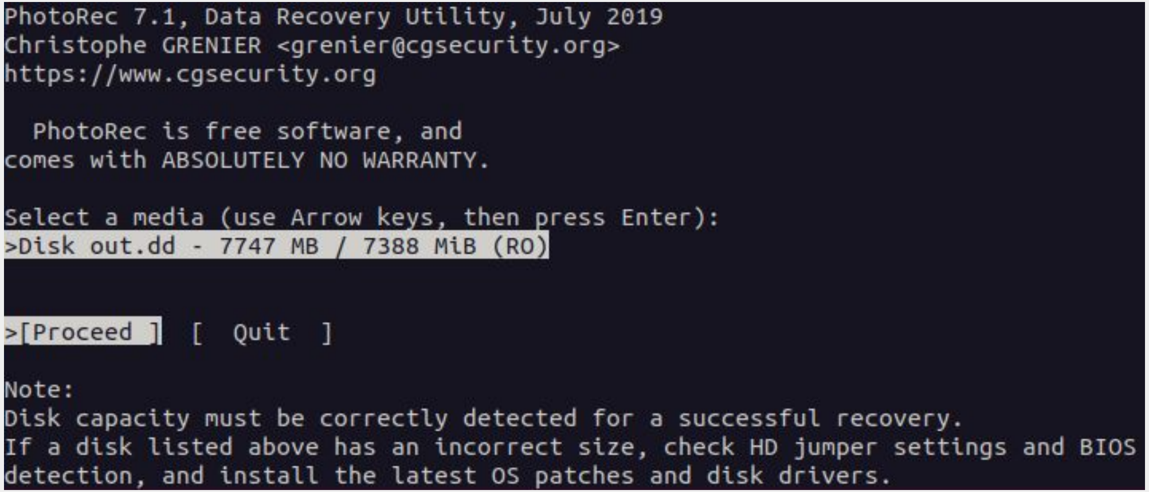
```
    $ sudo apt install foremost

    $ foremost -T -t jpg -i out.dd
```

... and DONE!

Automated File Carving

photorec

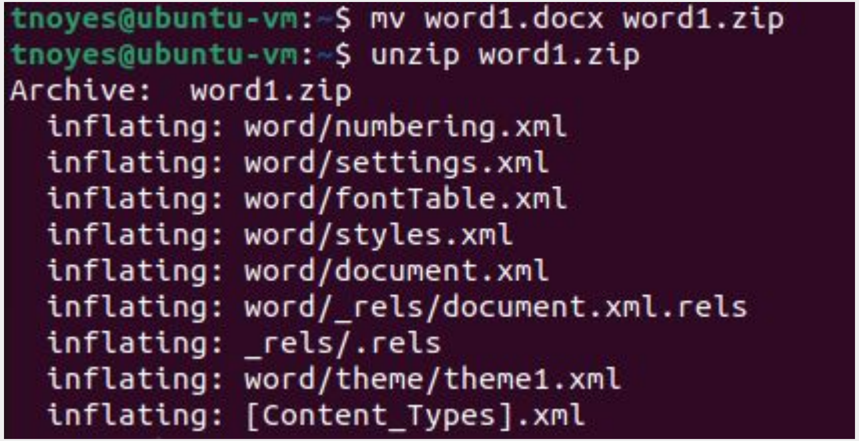
```
$ sudo apt-get install testdisk

$ sudo photorec out.dd
```



```
PhotoRec 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

  PhotoRec is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press Enter):
>Disk out.dd - 7747 MB / 7388 MiB (RO)


>[Proceed ]  [  Quit  ]

Note:
Disk capacity must be correctly detected for a successful recovery.
If a disk listed above has an incorrect size, check HD jumper settings and BIOS
detection, and install the latest OS patches and disk drivers.
```

Word Documents

Try changing a .docx into a .zip file



```
tnoyes@ubuntu-vm:~$ mv word1.docx word1.zip
tnoyes@ubuntu-vm:~$ unzip word1.zip
Archive:  word1.zip
  inflating: word/numbering.xml
  inflating: word/settings.xml
  inflating: word/fontTable.xml
  inflating: word/styles.xml
  inflating: word/document.xml
  inflating: word/_rels/document.xml.rels
  inflating: _rels/.rels
  inflating: word/theme/theme1.xml
  inflating: [Content_Types].xml
```

View with xmllint

```
$ xmllint --format word/document.xml
```

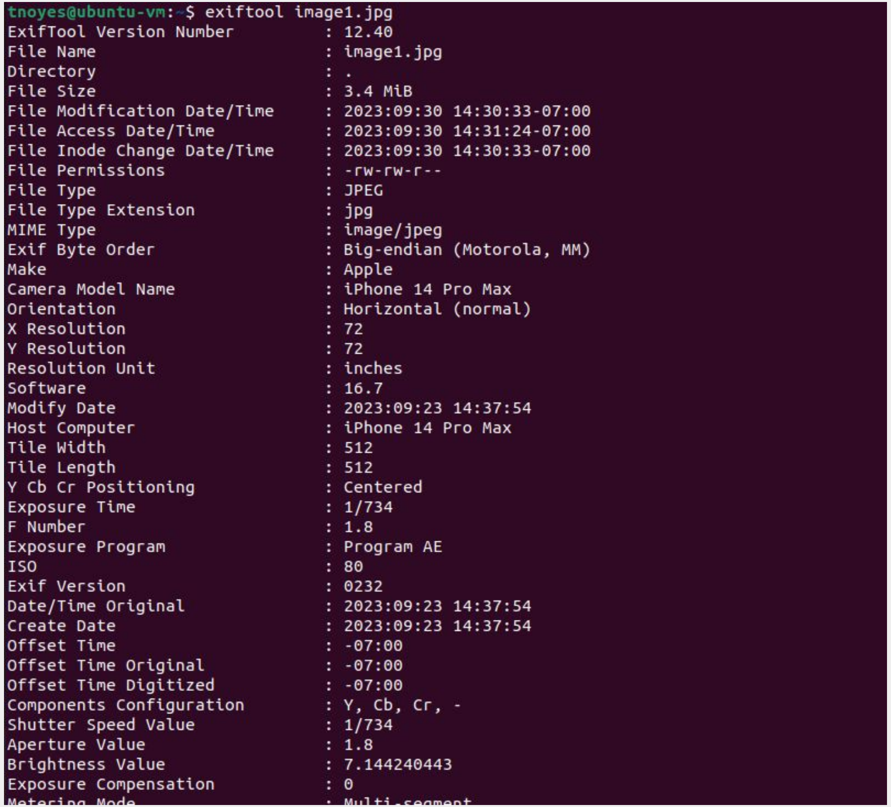
Metadata and More

Examine JPEG metadata

```
$ exiftool image1.jpg
```

- File Type
- Timestamps
- Camera model

… even GPS information!



```
tnoyes@ubuntu-vm:~$ exiftool image1.jpg
ExifTool Version Number         : 12.40
File Name                       : image1.jpg
Directory                       : .
File Size                       : 3.4 MiB
File Modification Date/Time     : 2023:09:30 14:30:33-07:00
File Access Date/Time           : 2023:09:30 14:31:24-07:00
File Inode Change Date/Time     : 2023:09:30 14:30:33-07:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
Exif Byte Order                 : Big-endian (Motorola, MM)
Make                            : Apple
Camera Model Name               : iPhone 14 Pro Max
Orientation                     : Horizontal (normal)
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : 16.7
Modify Date                     : 2023:09:23 14:37:54
Host Computer                   : iPhone 14 Pro Max
Tile Width                      : 512
Tile Length                     : 512
Y Cb Cr Positioning             : Centered
Exposure Time                   : 1/734
F Number                        : 1.8
Exposure Program                : Program AE
ISO                             : 80
Exif Version                    : 0232
Date/Time Original              : 2023:09:23 14:37:54
Create Date                     : 2023:09:23 14:37:54
Offset Time                     : -07:00
Offset Time Original            : -07:00
Offset Time Digitized           : -07:00
Components Configuration        : Y, Cb, Cr, -
Shutter Speed Value             : 1/734
Aperture Value                  : 1.8
Brightness Value                : 7.144240443
Exposure Compensation           : 0
Metering Mode                   : Multi-segment
```

Mapping GPS Coordinates

Create a map using Python!

```
$ sudo apt install python3-pip
$ pip install folium geopy pandas exifread
```

Download this script from Github:

https://github.com/dkmcgrath/sysadmin/blob/main/find_coords.py

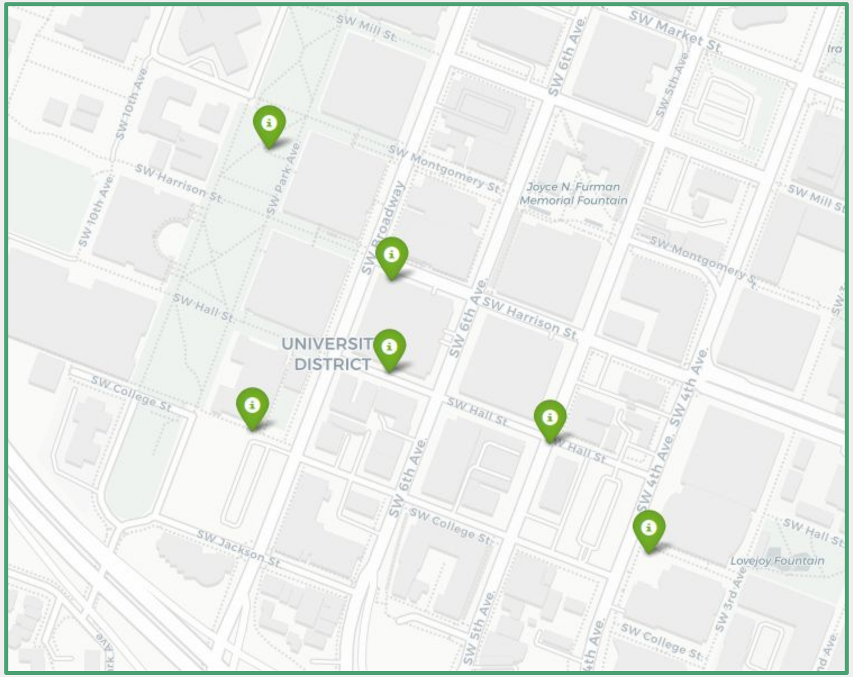
Mapping GPS Coordinates

Execute the script in the same directory as the photos.

```
$ python3 find_coords.py
```

Mapping GPS Coordinates

Open the resulting file called “map.html” in a web browser, and view the map!



CTF

github.com/PSU-Cybersecurity-Club/forensicsdemo

Okay, time to wrap up!

More fun!

Autopsy

Disk Analysis & Autopsy

Linux Server Forensics

Windows Forensics 1

Windows Forensics 2

Wireshark 101

Where to find us...

Website	psusec.org (Coming soon!)
Discord	https://discord.gg/2QKup4BryT
PSU Connect	https://pdx.campuslabs.com/engage/organization/PSUSec

Questions?