Project 5: Forensics

This project is split into two parts, with the first checkpoint due on **Wednesday, December 5** at **11:59pm** and the second checkpoint due on **Wednesday, December 12** at **11:59pm**. We strongly recommend that you get started early. **Start early**. It may be impossible to complete this project before the deadline unless you begin several days beforehand. Please plan accordingly.

This is a group project; you SHOULD work in **teams of two** and if you are in teams of two, you MUST submit one project per team. Please find a partner as soon as possible. If have trouble forming a team, post to Piazza's partner search forum.

Not all groups will finish all the tasks in all the MPs. The tasks in each MP are designed to be progressively harder with the final tasks in each MP having been designed as *significant* challenges. Some difficult tasks are placed earlier for better understanding of the scenario flow.

Strict NO-leaks policy. In this project you play the role of a computer forensic analyst working to solve a murder case. Since you don't want to be fired for jeopardizing an ongoing criminal investigation, you need to follow a strict policy on collaboration. *You are bound by the Student Code not to communicate with anyone regarding any aspect of the case or your investigation (other than within your group or with course staff).* The number of pieces of evidence you find, the techniques you try, how successful said techniques are, the general process you follow, etc. are all considered part of your solution and must not be discussed with members of other groups.

Solutions MUST be submitted electronically in either one of the group member's GitHub directories, following the submission checklist given at the end of each checkpoint. You MUST use the files distributed to the repository to which you submit, as different partners may receive different files. You should decide on on one partner's repository to use at the beginning of the project, and ignore files distributed to the other partner's repository entirely. Details on the filename and submission guidelines are listed at the end of the document.

"In general, computer forensics is rather ad hoc. Traditional rules of evidence are broken all the time. But this seems like a pretty egregious example."

– Bruce Schneier

Introduction

In this project, you will play the role of a digital forensic analyst and investigate a murder mystery.

A few days after Halloween 2015, a terrible crime occurred on University of Illinois at Urbana Champaign campus. A Hapless Victim was murdered in the dorm. The Victim was last seen alive on November 4, 2015 early afternoon in class, and was discovered dead at approximately 11pm the same day.

Officers were not able to immediately detect the actual sign of the death and asked forensics team to investigate. While waiting for forensics team to share the result, they have performed digital forensics analysis on victim's hard disk. A suicide note was left on the hard disk. Forensics team confirmed that the victim was dead in the way the suicide note mentioned. Hence, the investigation department's initial conclusion to the case was suicide.

However, upon further investigation, they opened the possibility of murder and filtered out few suspects. They obtained a search warrant, and were able to obtain the hard disks of all the suspects. Investigators successfully decrypted the hard disks and created an image for further investigation.

Your job is to conduct a forensic examination of the disk image and document any evidence related to the murder. If you find sufficient evidence, the suspect will be arrested and face trial.

Objectives

- Understand how computer use can leave persistent traces and why such evidence is often difficult to remove or conceal.
- Gain experience applying the security mindset to investigate computer misuse and intrusion.
- Learn how to retrieve information from a disk image without booting the operating system, and understand why this is necessary to preserve forensic integrity.

Guidelines

- You SHOULD work in a group of 2.
- Your answers may or may not be the same as your classmates'.
- All the necessary files to start the project will be given under the folder called "`Forensics`" in your Git repository: `https://github-dev.cs.illinois.edu/cs461-fa18/NETID/Forensics` on the `Forensics` branch. You should merge this branch into your `master` branch, as the `master` branch will be the one graded
- We generated files for you to submit your answers in. You MUST submit your answers in the provided files! You MUST also create and submit an `evidence/` directory containing

relevant files you recover. You SHOULD create and submit an `explanation/` directory explaining how you got each answer. This directory will not be graded initially, but may be used to justify regrades or partial credit.

- Each submission file contains an example of expected format. Failure to follow this format may result in autograder failure. Note: The # included in the example line causes the line to be ignored. Your submission should not have a # in front of it.
- Solution format is case-insensitive.

Read this First

Collaboration: Strictly prohibited outside your group. As stated above, you are bound by the Student Code not to communicate with anyone regarding any aspect of the case or your investigation (other than within your group or with course staff). The number of pieces of evidence you find, the techniques you try, how successful said techniques are, the general process you follow, etc. are all considered part of your solution and must not be discussed with members of other groups. If anyone brings up the project, put your fingers in your ear, start yelling "LALALALA", run around and refer them to your supervisor for an official spokesperson.

Getting Started

The tools and techniques you use for your investigation are up to you, but here are some suggestions to help you get started. This MP requires multiple tools to be downloaded and installed and you MUST use your own system. The decompressed disk image and virtual hard disk image sum up to >10GB. Password-cracking software should be run on private computer. Finally, we **strongly recommend you run Autopsy on Windows** as it provides extra functions which pre-filters interesting evidence and will tremendously save your investigation time.

General Knowledge A general working knowledge of Linux is undoubtedly helpful for this project. If you don't have this yet, you may need to spend time Googling and/or experimenting to get up to speed. The TA will also answer general Linux questions as a last resort. For an excellent reference book, try *UNIX and Linux System Administration Handbook* by Nemeth, Snyder, Hein, and Whaley. Ebook can be found easily by Googling. Also, see `http://en.wikipedia.org/wiki/Disk_partitioning` for some additional background.

Live Analysis Overview Live analysis is a forensic technique in which the investigator examines a running copy of the target system. We suggest using VirtualBox for this purpose.

1. Download the compressed raw disk image (1.5 GB):

   `https://goo.gl/ggnR3U`

   (You need to be logged in to your Illinois account)

2. Decompress the disk image.
   `$ gunzip Victim_Fall_18.raw.gz`

3. Find and open the `sha256sums.txt` file included in your Git repo. This file contains two different checksums for each the victim and suspect VM: compressed (raw.gz) and decompressed (raw) files

4. Verify the checksum. You can repeat this process for checkpoint 2 VM.
   In the same directory as the raw (or compressed) image:
   `On Linux:  $sha256sum -c sha256sums.txt`
   `On OS X: $shasum -a 256 -c sha256sums.txt`

5. Convert the raw disk image to a VirtualBox disk image:
   `$ VBoxManage convertdd Victim_Fall_18.raw forensics.vdi -format VDI`

6. Use the VirtualBox GUI to create a new VM. Select Linux / Ubuntu (32-bit) as the machine type. Select "Use an existing virtual hard disk file" and select the VDI you just created.

7. Start the VM and explore the system.

Dead Analysis Overview In dead analysis, a forensic investigator examines data artifacts from a target system without running the system. We suggest trying dead analysis with the Autopsy open-source forensics tool. We suggest you run Autopsy on Windows if possible, as it is easier to install and provides some additional functionality. However, in case you must use Mac or Linux, we provide some installation instructions. On Mac, you should be able to simply type "`brew install sleuthkit`" and then "`brew install autopsy`". The procedure below details the installation procedure for Ubuntu Linux.

1. Compile and install the Autopsy digital forensics suite: visit below link and follow the instruction.
   `http://www.sleuthkit.org/autopsy/download.php`

   Note: Do not install using "apt-get" command. The version available from apt-get is out-of-date, and does not support ext4. Our VM is based on ext4 which is unable to show all file system on version under 3.0.6. Make sure to download version 4.2.0 (or later) and compile.

2. Launch Autopsy in the background and open the browser-based GUI:
   `$ sudo autopsy &`
   In a browser on the local machine, go to the URL `http://localhost:9999/autopsy`.

3. Create a new case and add the disk image:

   (a) Click New Case. Enter a case name and click New Case.

   (b) Go back to `http://localhost:9999/autopsy` and open the case you created.

   (c) Click Add Host. Enter a host name and input the time zone of the system to analyze. If time zone is not specified, it will use your computer system time zone by default. Click Add Host.

   (d) Click Add Image. Click Add Image File. Enter the path to the decompressed raw disk image. Make sure you select Type=Disk and Import Method=Symlink. Click Next.

   (e) Leave the Image File Details and File System Details as the defaults. (Note that the disk image contains 3 partitions, which Autopsy will allow you to examine separately.) Click Add. Click OK.

   (f) Select a partition to examine and click Analyze. The buttons at the top give you several analysis tools. Try File Analysis and Keyword Search to get started.

4. In addition to hints dropped elsewhere, here is an incomplete list of things to try:

   - Examine the system logs.
   - Check for deleted or encrypted files.
   - Search the drive image for strings that may indicate relevance to your investigation.

Note: If you are having issues with image mounting procedure (or mounts successfully but fails to provide the correcting mounting point), you may not see all file systems and directories with above instructions, so try below alternative.

This procedure will not completely replace Autopsy as you will need to do all the analysis/searches manually. You should find another operating system to run Autopsy properly.

1. Create a directory where you will mount the image.
   `$ sudo mkdir /mnt/victim`

2. Mount the image to the mount directory you created.

   ```
   $ sudo mount -t ext4 -o loop,ro,noexec,offset=1048576
   [raw_image_file_path] /mnt/victim
   ```

   This command mounts the specified image to the specified location as a loop, read-only, no-execute ext4 (the filesystem that linux commonly uses) partition. The offset needs to be specified because your host system will not know exactly at where the linux partition resides within that disk image. The offset is 1048576 because the partition starts at sector 2048, and each sector is 512 bytes. 2048 * 512 = 1048576.

3. Once the image is mounted, you can pretty much browse that partition as if it's part of your own filesystem.

4. Change directory to the mount directory.
   `$ cd /mnt/victim[or full_path_to_the_directory_you_created]`

5. List the files in the directory. You will see files and folders that are similar to your root directory.
   `$ ls`

Password Cracking Password crackers may be helpful in trying to brute-force decrypt password-protected files. Not all the tools will be used for this MP exploration, but here are some useful tools list. John the Ripper (`http://www.openwall.com/john/`) is the canonical Unix password cracker. Hydra (`http://www.thc.org/thc-hydra/`) is a tool used to brute force remote login passwords, fcrackzip (`http://home.schmorp.de/marc/fcrackzip.html`) is a ZIP password cracker, and pdfcrack (`http://sourceforge.net/projects/pdfcrack/`) is a PDF password cracker. John, fcrackzip, and pdfcrack are conveniently available in the Debian package repositories and can be installed with `apt-get`.

When using a password cracker, it is wise to make sure that the password is not susceptible to a dictionary attack and does not use a restricted character set (e.g., lowercase letters, letters only, letters and numbers only) before spending time on a full brute-force crack. It is also a good idea to crack a very vulnerable password first to make sure you are using the tool correctly.

Metadata Viewer There exist multiple metadata viewers, but for our MP, we suggest using ExifTool by Phil Harvey. It can be downloaded at (`http://owl.phy.queensu.ca/~phil/exiftool/`). Follow the instructions on the website to view the metadata of the evidence. You may also be able to use Autopsy to extract useful file metadata.

5.1 Checkpoint 1 (20 points)

The deliverables for this project are your answers to the questions below. Your answers should be *complete* but *concise*.

For each prompt, you may optionally explain the investigatory methods you used and the evidence that supports your conclusion. The files can be included in a separate directory named `explanation/`. These explanations will not be graded but can be considered during manual regrading.

Exploring Victim's Traces

In this part of the project, you will be exploring the traces left on victim's hard disk.

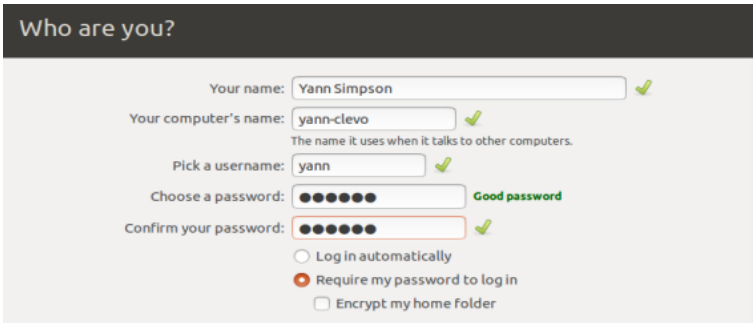
5.1.1 Username (2 points)

(*Difficulty: Easy*)

What is the username of the victim used on OS?

What to submit: Submit a text file named `5.1.1.txt` that contains the username of the victim.

Note: Username is the id used to log on to the system. For example, there is a student, `Yann Simpson`, whose netid is `yann`. To access the University system, the netid, `yann` is used as username to log on. If logged on, they will display either the netid or the full name to indicate the user of the account. Illustrative example is shown below.



example content of 5.1.1.txt

```
# this line is ignored
yann
```

5.1.2 Timezone (2 points)

(*Difficulty: Easy*)

What is the timezone setting of the victim's system?

What to submit: Submit a text file named `5.1.2.txt` that contains the timezone of the victim's system.

Hint: The system is based on one of US time zones. Four time zones in US are EST, CST, MST, and PST. Do NOT submit in any other format, e.g. UTC, America/New York, Eastern Time Zone etc.

example content of 5.1.2.txt

```
EST
```

5.1.3 Conversation (4 points)

(*Difficulty: Easy*)

Whom did the victim have conversation(s) with? List each username in separate line, in the order of occurrences.
If the victim had multiple conversations with the same person in different times, list the username as many times as the conversation history dictates. Note that this does not equal to the message count transferred back and forth for single conversation.

What to submit: Submit a text file named `5.1.3.txt` that contains the list of **usernames** (not Display names nor Computer names) the victim have conversation with in chronological order.

Note: For example, let's assume that Alice is a victim and you are exploring the chat history. The list below is the summary of the chat histories you have explored.

```
<Alice's chat history>
(with bob@chat.org) @ Apr 2 7:00pm
(with eve@chat.org) @ Apr 3 10:00am
(with eve@chat.org) @ Apr 3 2:00pm
```

The usernames are *bob, eve, eve* respectively, without the domain *chat.org*. Then, the solution to this example will be the following.

example content of 5.1.3.txt

```
bob
eve
eve
```

5.1.4 Evidence file (4 points)

(*Difficulty: Easy*)

The police initially thought that the victim committed suicide after finding a certain file on the victim's computer.

1. What is the name of the file? Include the file extension.
2. When was this file last modified? Submit the time in MMddhhmm 24-hr format. (MM=Month, dd=day, hh=hour, mm=minute). For example, 11:30pm on April 3rd is 04032330.

What to submit

1. Submit a text file named `5.1.4_name.txt` that contains the full name of the file.
2. Submit a text file named `5.1.4_time.txt` that contains the last modified time of the file.

example content of 5.1.4_name.txt

```
evidencefile.txt
```

example content of 5.1.4_time.txt

```
04032330
```

5.1.5 Attack (8 points)

(*Difficulty: Medium*)

After initial conclusion, the police started to be suspicious about possibility of the murder. Do you find any trace of the attack on victim's machine?

1. If so, when did the attacker make the first contact to the victim's computer? Any contact can either be successful login or failed attempt. Submit the time in MMddhhmm 24-hr format. (MM=Month, dd=day, hh=hour, mm=minute).
2. List all the IP addresses the attack originated from. Submit IPv4 addresses. If there are multiple attacker IP addresses, include one IP address per each line.
3. What was the victim's IP address during the attack? Submit IPv4 address.

What to submit

1. Submit a text file named `5.1.5_time.txt` that contains the first attack time.
2. Submit a text file named `5.1.5_attackerip.txt` that contains the list of IP addresses the attack came from. Each line in the file contains one IP address.
3. Submit a text file named `5.1.5_victimip.txt` that contains the victim's IP address.

example content of 5.1.5_attackerip.txt

```
1.2.3.4
5.6.7.8
```

Checkpoint 1: Submission Checklist

Inside your `Forensics` directory in GitHub, you will have the auto-generated files named as below. Make sure that your answers for all tasks up to this point are submitted in the following files before **Wednesday, December 5** at **11:59pm**:

GitHub Directory

`https://github-dev.cs.illinois.edu/cs461-fa18cs461-fa18//NETID/Forensics` (`master` branch)

- `partners.txt` [One netid on each line]
- `5.1.1.txt`
- `5.1.2.txt`
- `5.1.3.txt`
- `5.1.4_name.txt`
- `5.1.4_time.txt`
- `5.1.5_time.txt`
- `5.1.5_attackerip.txt`
- `5.1.5_victimip.txt`

5.2 Checkpoint 2 (100 points)

The deliverables for this project are your answers to the questions below. Your answers should be *complete* but *concise* following the submission template.

The suspect VM that you will explore is given individually in your Git directory. Follow and download the disk image provided at the link in `suspect_vm.txt`. We have also created `code.txt` in your Git directory, which you may find useful as you make progress in the MP. The instructions for this file will be given at some point in the project, so be patient.

If you recover files that are relevant to your responses, include them with your submission in a directory named `evidence/`. Do not change the original file name on your submission. For each prompt, you may optionally explain the investigatory methods you used and the evidence that supports your conclusion. The files can be included in a separate directory named `explanation/`.

Investigating Suspect Traces

In this part of the project, you will be exploring the traces left on suspect's hard disk.

5.2.1 Live Analysis (10 points)

(*Difficulty: Hard*)

Now, the police department has obtained multiple suspects' hard disks, distributed them across teams to be investigated in parallel. You are given with one disk image to analyze. Answer the following set of questions to fill out the report.

You were given with one disk image to analyze. Now, you want to try the live analysis of the evidence. You must perform live analysis to answer subset of questions, but you can also perform dead analysis to obtain the evidence.

Let's first take a look at the machine environment. Be careful and specific; e.g., say "Windows 2000" instead of just "Windows."

1. Try booting the suspect's machine and using it normally. What operating system does it boot by default? Include OS name and the distribution release number in separate lines.
2. What specific or potentially dangerous behavior(s) of the default boot OS have on the machine? Find a script that executes the specific behavior. Submit the name of the script file.
3. What operating system did the suspect primarily use? Submit the OS's name and distribution release number in separate lines.

What to submit

1. Submit a text file named `5.2.1_default.txt` that contains the default booting OS.
2. Submit a text file named `5.2.1_behavior.txt` that contains the script file name that executes the specific behavior of the default booting OS.

3. Submit a text file named `5.2.1_primary.txt` that contains the primary OS the suspect used.

Note: Note that OS distribution number is not equivalent to the kernel number. Refer the details at `http://whatsmyos.com/` or `https://www.linux.com/learn/tutorials/824791-how-to-find-your-linux-version-or-distro-release-and-why-it-matters`.

example content of 5.2.1_{default,primary}.txt

```
OS X
10.10.4
```

example content of 5.2.1_behavior.txt

```
scriptfile.ext
```

5.2.2 Dead Analysis (90 points)

We have explored some system details through live analysis in section 5.2.1. There can be risks during live analysis and the act may actually contaminate the evidence. For this section, you will be strictly performing dead analysis to complete the investigation.

Now, with the same disk image you have performed live analysis in the section 5.2.1, you will mount on Autopsy and explore without turning on the system. Answer the following set of questions to fill out the report.

5.2.2.1 Username (5 points) (*Difficulty: Easy*)

What is the OS username of the suspect?

What to submit: Submit a text file named `5.2.2.1.txt` that contains the OS username of the suspect.

5.2.2.2 Conversation (5 points) (*Difficulty: Easy*)

You have investigated conversation history of the victim in section 5.1. Now, you will do the same for the suspect.

1. Whom did the suspect have conversation(s) with? List each username in separate line, in the order of occurrences. If the suspect had multiple conversations with the same person at different times, list the username as many as the conversation history. Note that this does not equal to the message count transferred back and forth for single conversation. *Note: Consider only the chat history similar to checkpoint 1.*

2. What is the suspect's relationship with the victim? Choose the most appropriate one among following choices. Submit just the alphabet of the choice.

(a) victim's best friend who liked victim's girlfriend

(b) victim's girlfriend

(c) victim's boyfriend

(d) course instructor who received bad feedback from victim

(e) boyfriend of the girl whom victim cheated with

(f) victim's course project partner who had to do all the work by himself/herself

(g) none of the above

What to submit

1. Submit a text file named `5.2.2.2_usernames.txt` that contains the list of usernames the suspect have conversation with.
2. Submit a text file named `5.2.2.2_relationship.txt` that contains the relationship of the suspect and the victim.

example content of 5.2.2.2_relationship.txt

```
# alphabet of the choice
k
```

5.2.2.3 Search History (10 points) (*Difficulty: Medium*)

There must be a reason that the user is considered as suspect.

1. Are there any indications that the suspect was trying to conduct an attack? How about any indication that the suspect owned or was researching weapons of the kind involved in the murder? What are the websites that the suspect visited potentially related with murder? List 5 website full links, one link per line in time order. **You must give 5 links from unique domains.** E.g. Two different searches on Google count as 1. Don't forget to check the typos and whether all letters are included.
2. What did the suspect plan to use as a weapon to murder the victim? Note that the toy evidence included in this case is considered lethal and dangerous.

What to submit

1. Submit a text file named `5.2.2.3_link.txt` that contains the list of websites visited related with the murder.
2. Submit a text file named `5.2.2.3_weapon.txt` that contains the weapon that suspect planned to use.

example content of 5.2.2.3_links.txt

```
https://www.google.com/maps
https://www.cnet.com/news/
```

example content of 5.2.2.3_weapon.txt

```
lightsaber
```

5.2.2.4 Encrypted File (10 points) (*Difficulty: Medium*)

Were there any suspicious-looking encrypted files on the machine? If so, what was the password that was used to encrypt this file? Also, attach the decrypted contents as evidence. *Note:* Submit the actual individual files, not the compressed format.

What to submit: Submit a text file named `5.2.2.4.txt` that contains the password of the encrypted file and decrypted contents in `evidence/` directory.

example content of 5.2.2.4.txt

```
p4ssw0rd
```

5.2.2.5 Attack (20 points) (*Difficulty: Hard*)

1. Which account did the suspect try to attack and use to access/log onto the victim's computer? Submit the username of the account. **If they tried to access more than one account, you should submit the last account they tried to access.**

2. What tools did the suspect use to gain access to the victim's computer? As you investigate, be on the lookout for evidence of any other machines or network services that the suspect may have used. Be careful. The suspect might have decided not to use some tools. List the terminal command name of each tool in separate lines in the order of occurrence.

3. List the suspect's IP address(es) used during the attack.

4. Did the suspect successfully connect to the victim's computer? If so, list the filename of the private and public key in separate lines that are generated/saved to be used for authenticating the connection. *Hint:* Try to find what kind of algorithm was used to generate the key pair/signature.

   posssiblely RSA pairs

5. What was the password of the account obtained/used by the suspect for the victim's computer?

What to submit

1. Submit a text file named `5.2.2.5_account.txt` that contains the username of the suspect used to access victim's machine.
2. Submit a text file named `5.2.2.5_tools.txt` that contains the list of tools victim used during the attack.
3. Submit a text file named `5.2.2.5_ip.txt` that contains the ip address(es) of the suspect used in the attack.
4. Submit a text file named `5.2.2.5_connection.txt` that contains whether the connection was successful as well as private and public key filenames.
5. Submit a text file named `5.2.2.5_password.txt` that contains the password of the username the suspect obtained/used during the attack.

example content of 5.2.2.5_tools.txt

```
zip
john
```

example content of 5.2.2.5_connection.txt

```
# first line would be either [yes/no]
no
private_key_file_name
public_key_file_name
```

example content of 5.2.2.5_password.txt

```
p4ssw0rd
```

not same as the cp1?

5.2.2.6 File Export/Recovery (10 points)

(*Difficulty: Hard*)

Did the suspect try to delete any files that may be related with the murder? List one file name that is the most suspicious looking and export or recover the deleted file. *Hint:* If you have difficulty recovering the original deleted file, try to think of an alternative to obtain the file with same content. **Note: It is possible that you will be able to view metadata for a deleted file relatively easily, but recovering the content for that file will be very difficult, or even impossible. This is simply the nature of forensic investigations. For this reason, we recommend completing the rest of the MP before you spend large amounts of time trying to extract deleted files.**

What to submit: Submit a text file named `5.2.2.6.txt` that contains the name of the deleted files with file extension and, if you are able to recover the contents, the obtained files in `evidence/` directory with the original file name.

example content of 5.2.2.6.txt

```
filename.ext
```

5.2.2.7 Escape Plan (20 points)

(*Difficulty: Medium*)

The suspect may have planned for an after-murder escape scenario.

1. Are there any indications that the suspect had an accomplice who was physically present on the night of the crime, or any source that provided the escape method/transportation after the crime? If so, submit the contact information of the accomplice or source company.

2. What is the location of the escape plan? Submit the GPS coordinates (latitude and longitude) in signed 3rd decimal format in separate lines. If you view more then 3 decimal places, do not round and drop from 4th decimal (e.g. -1.2345 -> -1.234). Use ExifTool (`http://owl.phy.queensu.ca/~phil/exiftool/`) to view the metadata if necessary.

   Notes:

   - ExifTool has options to convert geotag coordinates into preferred format.
   - However, if you obtained a geotag from elsewhere in <deg ' "> or <° ' "> format AND cannot use exiftool to convert, then you should use converter from this link: `http://www.gps-coordinates.net/gps-coordinates-converter`. **IMPORTANT:** Different converters can result in slightly different values and may fail our grader if specified tool is not used.
   - Signs for four directions in GPS coordinate: North (+), South (-), East (+), West (-).

   Hints: If you have trouble viewing the file, check the metadata (e.g. file type, file extension, file format etc.) and think of the possibilities to resolve. For example, change the extension of the file.

3. What was the original time of the escape? Submit in 24-hr hhmm format (hh=hour, mm=minute)

4. Is there good evidence that the suspect actually escaped? If so, submit the actual escape time in 24-hr hhmm (hh=hour, mm=minute). If not, simply say, "unknown" without the quotation.

What to submit

1. Submit a text file named `5.2.2.7_accomplice.txt` that contains the contact information of the accomplice.

2. Submit a text file named `5.2.2.7_location.txt` that contains the escape location coordinate.

3. Submit a text file named `5.2.2.7_originaltime.txt` that contains the original planned escape time.

4. Submit a text file named `5.2.2.7_actualtime.txt` that contains the actual escape time.

example content of 5.2.2.7_location.txt

```
# latitude on first line
# longitude on second line
1.234
5.678
```

example content of 5.2.2.7_{originaltime,actualtime}.txt

```
2330
```

5.2.2.8 File Metadata (5 points) (*Difficulty: Easy*)

Now, go back to the victim's VM. What is the creator/writer/author (Full Name) of the evidence file you found on 5.1.4? Do NOT submit the username or the userid.

What to submit: Submit a text file named `5.2.2.8.txt` that contains the creator of the evidence file.

Note: From the same example as 5.1.1, the full name is `Yann Simpson`.

example content of 5.2.2.8.txt

```
Yann Simpson
```

5.2.2.9 Final Decision (5 points) (*Difficulty: Medium*)

Using the evidence you have gathered so far, try to complete the scenario of the suspect you investigated. Do you think the suspect actually murdered the victim? If so, say yes. Otherwise, no.

What to submit: Submit a text file named `5.2.2.9.txt` that contains whether you think the suspect is the real criminal or not. This should be a one word answer ("yes" or "no"). You can optionally include an explanation of your thinking in a file in the `explanation/` directory.

Checkpoint 2: Submission Checklist

Inside your `Forensics/` directory, you will have the auto-generated files named as below. Make sure that your answers for all tasks up to this point are submitted in the following files before **Wednesday, December 12** at **11:59pm**:

Git Directory

`https://github-dev.cs.illinois.edu/cs461-fa18/NETID/Forensics/`

- `partners.txt` [One netid on each line]
- `5.2.1_default.txt`
- `5.2.1_behavior.txt`
- `5.2.1_primary.txt`
- `5.2.2.1.txt`
- `5.2.2.2_usernames.txt`
- `5.2.2.2_relationship.txt`
- `5.2.2.3_link.txt`
- `5.2.2.3_weapon.txt`
- `5.2.2.4.txt`
- `evidence/"decrypted_file"`
- `5.2.2.5_account.txt`
- `5.2.2.5_tools.txt`
- `5.2.2.5_ip.txt`
- `5.2.2.5_connection.txt`
- `5.2.2.5_password.txt`
- `5.2.2.6.txt`
- `evidence/"recovered_file"`
- `5.2.2.7_accomplice.txt`
- `5.2.2.7_location.txt`

- 5.2.2.7_originaltime.txt
- 5.2.2.7_actualtime.txt
- 5.2.2.8.txt
- 5.2.2.9.txt