# Objective 2

We completed Objective 1 by entering the gate, so let’s go on to Objective 2. This objective will use a variation on the old Where in the World is Carmen San Diego. We can choose to use Open Source Intellegence (OSINT) to follow Caramel around the world, or use a simple hack with browser web developer tools to cheat. We’ll need to solve a terminal first, though.  
 Graphical user interface, text

Description automatically generated

## Terminal Exif Metadata

The objective tells us to find Piney Sappington in the courtyard to get a hint from him. To get to the courtyard go into Santa’s castle and find your way to the back door.  
A picture containing text, toy

Description automatically generated Text

Description automatically generated

## Assignment: exiftool and searching

If you open the terminal and get a directory listing, you will see that there are 25 image files. Our goal is to examine the file metadata to see which one was altered by Jack.

This terminal shows you some of the metadata that is available in images and documents. To save yourself time, you can learn to use wildcards in a command to execute the command on many files at once. You can also learn to pipe the results of one command into another, like grep, to output just the data you want to see. If you are just starting with the command line, these links may help.

<https://www.tecmint.com/use-wildcards-to-match-filenames-in-linux/>   
<https://www.guru99.com/linux-pipe-grep.html>

### Step 1 question: What is in the metadata?

Use exiftool to examine the metadata available for one file. Is there a field that would help you decide if a file has been modified?

### Step 1 answer.

Use the command exiftool 2021.12.01.docx . I just picked the first file, but any file will do.

Text

Description automatically generated  
<snip>  
Text

Description automatically generated

If wily Jack Frost was not paying attention to the metadata, the Modify Date may have been changed from the original Create Date when he modified the file. Notice Create and Modify Date are the same on this file.

### Step 2 question: Examine the metadata for all the files

Create a command that runs exiftool on all the files in the directory. Then pipe the output to grep and search for “Modify Date”. Here is a link about using wild cards for file names.  
<https://www.tecmint.com/use-wildcards-to-match-filenames-in-linux/>

### Step 2 answer

We can use the wild card, ‘\*’, in our command to run exiftool against all the files in our directory. Then pipe the result into grep "Modify Date".

exiftool \* | grep "Modify Date"  
Text, chat or text message

Description automatically generated

Oops. Our grep command was not specific enough, and we picked up the Zip Modify Date as well. We can fix this by adding the regular expression (regex) **anchor**, ‘^’, to say that the line must start with Modify Date.

exiftool \* | grep "^Modify Date"  
Graphical user interface, text

Description automatically generated

Hmm, there is one that sticks out. You can probably discern that the file name should be 2021:12:21.docx, which is the correct answer. Paste that into the top half of the window and talk to Sappy to get more hints.

Text

Description automatically generated

Note: there are many ways to solve this problem.

### Step 3 Question (optional)

Our grep command in the previous step was helpful, but wouldn’t it have been nice if we had the filename in our output as well? There is a way to make grep match on more than one thing, the way an OR statement does. This link will show you how to do that. Change your grep statement so it catches both File Name and Modify Date.  
<https://www.thegeekstuff.com/2011/10/grep-or-and-not-operators/>

### Step 3 Answer

There are several ways to also search for the filename. The one I chose was

exiftool \* | grep -E '^File Name|^Modify Date'

Text

Description automatically generated

<snip>  
Text

Description automatically generated

Note: regex is case sensitive, but if you don’t want to mess with upper and lower case since just use  
grep -i to make it case insensitive.

## Hints after solving the Exif metadata terminal.

Piney has several hints for us.

|  |  |
| --- | --- |
| Text  Description automatically generated | A picture containing table  Description automatically generated<https://en.wikipedia.org/wiki/Military_Grid_Reference_System> <https://what3words.com/> |
| A picture containing graphical user interface  Description automatically generated  <https://gist.github.com/chriselgee/b9f1861dd9b99a8c1ed30066b25ff80b> |

This talk may also be helpful.

|  |  |
| --- | --- |
| Text  Description automatically generated | <http://www.youtube.com/watch?v=tAot_mcBT9c> |

## Objective 2: Where in the world is Caramel Santiago?

This is a reference to a 1985 computer game called “Where in the World is Carmen San Diego?” To play, you use the three investigations to determine the next city to visit and learn some clues about the elf you are tracking. Take note of the clues and go on to the next city. Once you have traveled to the correct city enough times, the game will ask you for the name of the elf you are chasing. Use your clues in the InterRink search to determine the elf, answer the question, and win the game!

These screenshots show what a city screen (left) and an investigation (right) look like. The investigation is giving a clue concerning the next city to visit.  
A picture containing text

Description automatically generated Qr code

Description automatically generated

The left screenshot shows another investigation with a hint about the next city. The right one gives clues about the elf’s identity. The weather information doesn’t help much, but the Star Trek comment does help. If you play the game, sometimes it gives you the city location in weird coordinates. That’s where Piney’s hint about Coordinate Systems comes in.A picture containing graphical user interface

Description automatically generated Graphical user interface

Description automatically generated with low confidence

InterRink looks like this. The clue helped, but we need more! Note: the elf and locations change each time you restart the game.  
Graphical user interface, text

Description automatically generated Text

Description automatically generated

You can play the game to win and gain the achievement for this objective, but why not cheat, uh hack, our way to success? The site stores enough information in a cookie for you to win easily if you can decode the cookie. Web developers often make mistakes that reveal information or allow evil users (that would be you and I) to send corrupted content to the server. In this case, we can reveal information about the game.

## Assignment: Introduction to Brower Web Developer Tools and Cookies

In this assignment you will get basic exposure to the Web Developer Tools (webdev) in Chrome or Firefox. You will see how a site can use to store information in a cookie and learn what fun it is to exploit cookies. Here are helpful links on how to open Chrome webdev tools and find cookies.  
<https://developer.chrome.com/docs/devtools/open/>   
<https://developer.chrome.com/docs/devtools/storage/cookies/>

Here are links for Firefox.  
<https://developer.mozilla.org/en-US/docs/Tools>   
<https://developer.mozilla.org/en-US/docs/Tools/Storage_Inspector>

### Step 1 question: Give me the cookie

Use the developer tools to locate and extract the contents of the cookie for caramel.kringlecastle.com.

### Step 1 answer

In Chrome, enter webdev (function key F12) and go to the Application tab. On the tree at the right, navigate to the caramel.kringlecastle.com cookie. Select the Cookiepella cookie and copy the long string of text below Cookie Value.

Graphical user interface, text, application, email

Description automatically generated

In Firefox, enter webdev (F12) and go to the Storage tab. On the tree at the right, navigate to the caramel.kringlecastle.com cookie. Select the Cookiepella cookie and copy the long string of text from value. (In the section at the far right, some characters make webdev think the cookie is an array and not a long string, so it is easier to use the value box.)

Graphical user interface, application

Description automatically generated

Either way, you should end up with a string that looks like this. It changes every time you reset the game, but I’ve included my cookie so you can practice with it. The cookie is one long line, so it wraps strangely in Word.

.eJyNUstu2zAQ\_JUFL73Iha2HbfmWJgEaoEWMKEgPcQ4rciUSpkiBohK4Qf69y7SHAIGBXndmdobDfRVkO7ETe-PoBA2Oo3F99E5kCfhuXJzE7lHca0aPNEaQmuSRORA1mQCNw1FqjMDKryy61RncgPPRSFKJcwKNChCuvIw-wC\_t03RgcNTeEUicKAnZAdgRBnLRMKBg8gNFnayw9XOEJqI8wu0zhc76F0Cn4G6e4kdx7yMEQmvZlTDykr\_SeXpPjO0EHYcwTrHLlJQagxJPmbBeYvLlKhp0Eb9McIlTtMQcPyYk9fAofninvMvg2vWWEzB66UdyGnvi6RW5AcORpxcuvlAYM\_hGtjfzwB6P4loZ186h544a6eO\_Bf-3Muk\_M396F8NhXi6pRptxZIcKef5gyDnM4IILCgbf1fuA\_UxM-k1Swx2Nc2uNZPLeoiTtraJwduUTdxTY0w8NkRK7Kq95wN1SOo9Prz1Xy7m8HyOw07NJx3K690dKP\_LKNzTp3UEsu1yWq7boVsui7aiW3aoo122F1Wa9obKs1oryVlWSCpSIUm7LLcm2oKqiut7Ig8j4PiY\_B0k3agcH0W6LKt90y0WNebcoiWhR5ypfFOuuWOJq2267\_CDexNsffMcLgQ.Yczalw.DiqH0OEsZ7rOiKkSBjrZVc-P5Ug

### Step 2 question: What is in the cookie

Use Sappy’s hint about Flask cookies (you won that hint by solving the EXIF meta-data cranpi) to decode the information you need to cheat at the game. You need to know the name of the Elf and the route you need to take through the cities. Once you have that you can win the game very quickly. Here is the link to the site mentioned in the Flask cookies hint. <https://gist.github.com/chriselgee/b9f1861dd9b99a8c1ed30066b25ff80b>

I found the method using the GCHQ Cyber Chef he mentions at the end of the gist to be easiest. Cyber Chef is really cool, hat tip to GCHQ (the UK’s NSA.) <https://gchq.github.io/CyberChef/>

### Step 2 answer.

I will use the cookie I found in step 2, above. Your cookie will undoubtedly be different.

First note that the cookie is separated into parts by periods. We need to remove the period from the beginning and everything after the next period.

eJyNUstu2zAQ\_JUFL73Iha2HbfmWJgEaoEWMKEgPcQ4rciUSpkiBohK4Qf69y7SHAIGBXndmdobDfRVkO7ETe-PoBA2Oo3F99E5kCfhuXJzE7lHca0aPNEaQmuSRORA1mQCNw1FqjMDKryy61RncgPPRSFKJcwKNChCuvIw-wC\_t03RgcNTeEUicKAnZAdgRBnLRMKBg8gNFnayw9XOEJqI8wu0zhc76F0Cn4G6e4kdx7yMEQmvZlTDykr\_SeXpPjO0EHYcwTrHLlJQagxJPmbBeYvLlKhp0Eb9McIlTtMQcPyYk9fAofninvMvg2vWWEzB66UdyGnvi6RW5AcORpxcuvlAYM\_hGtjfzwB6P4loZ186h544a6eO\_Bf-3Muk\_M396F8NhXi6pRptxZIcKef5gyDnM4IILCgbf1fuA\_UxM-k1Swx2Nc2uNZPLeoiTtraJwduUTdxTY0w8NkRK7Kq95wN1SOo9Prz1Xy7m8HyOw07NJx3K690dKP\_LKNzTp3UEsu1yWq7boVsui7aiW3aoo122F1Wa9obKs1oryVlWSCpSIUm7LLcm2oKqiut7Ig8j4PiY\_B0k3agcH0W6LKt90y0WNebcoiWhR5ypfFOuuWOJq2267\_CDexNsffMcLgQ

Now we paste that into Cyber Chef and set up the decoding as in Chris Elgee’s Gist. Be sure to use the urlsafe version of base64.

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

So, for this iteration of the game, the Elf is Piney Sappington. The route we need to follow so that the game will ask us for the Elf’s name is:  
route":["Antwerp, Belgium","Copenhagen, Denmark","Montr\u00e9al, Canada","Placeholder”]

Again, this gets jumbled every time a game starts so your values will probably be different. This is the answer for another game.

Graphical user interface, text

Description automatically generated Text

Description automatically generated