SPECTRUM CHALLENGE

SPECTRUM CHALLENGE

As Cyberhunter, I will conduct a thorough forensic analysis of the USB thumb drive found on the suspect to uncover details about the upcoming drug deal. My approach will involve examining the drive's file system for potential evidence, recovering any deleted files, and analyzing the content of documents, images, and communications. I will look for keywords, timestamps, and location data to piece together information about when and where the deal is expected to occur. By extracting and scrutinizing metadata and hidden content, I aim to provide Scotland Yard with critical insights to aid in their investigation and potentially prevent the deal from going down.

Scenario

Scotland yard have intercepted information about one of the biggest drug deals to go down in the city of London. Someone we believe is linked to the deal was arrested. The only item they had in their possession was a USB thumb drive. Unfortunately, one of our junior analysts was unable to find anything of interest. Before we let this suspect go, we would like one of our DF experts to see if they can find anything about the deal before it goes down. Can you find out where and when the deal is expected to go down? Provided with the Retrieved Files and password:btlo.

Challenge submission

a. What time is the meeting happening?
b. What are the supposed coordinates for the deal?
c. Looking into these coordinates, what is the name of this location?

File analysis

After downloading the given file let’s move into a separate directory and see what this file is. After extracting the file, it contain the image file(image.dd) then let's move to see the properties of the image using 'file' command.

```
┌─[jw@parrot]─[~/Desktop/FIELD-IAA/BLUETEAM/SPECTRUM2]
└──╼ $file image.dd
image.dd: DOS/MBR boot sector, code offset 0x3c+2, OEM-ID "mkfs.fat", sectors/cluster 4, reserved sectors 4, root entries 512, Media d
escriptor 0xf8, sectors/FAT 100, sectors/track 62, heads 58, hidden sectors 2048, sectors 102362 (volumes > 32 MB), reserved 0x1, seri
al number 0x2b873cba, unlabeled, FAT (16 bit)
┌─[jw@parrot]─[~/Desktop/FIELD-IAA/BLUETEAM/SPECTRUM2]
```

The image seen as disk image let's import to autopsy tools to see what the image contains.

Del	Type dir / in	Name	Written	Accessed	Created	Size	UID	GID	Meta
Error Parsing File (Invalid Characters?): V/V 1634534: $OrphanFiles 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0 0 0									
	v / v	$FAT1	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	51200	0	0	1634532
	v / v	$FAT2	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	51200	0	0	1634533
	v / v	$MBR	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	512	0	0	1634531
	r / r	noise_samples.zip	2021-08-05 17:41:32 (EAT)	2021-08-05 00:00:00 (EAT)	2021-08-05 17:41:32 (EAT)	24194984	0	0	5
	d / d	photos/	2021-08-05 17:41:32 (EAT)	2021-08-05 00:00:00 (EAT)	2021-08-05 17:41:32 (EAT)	2048	0	0	7

It's seems that the image contain some files and directories such as photos and noise_samples.zip. I use 'mount' command to import the image into /mnt folder ('sudo mount -o loop,ro image.dd /mnt ').

```
┌─[jw@parrot]─[~/Desktop/FIELD-IAA/BLUETEAM/SPECTRUM2]
└──╼ $sudo mount -o loop,ro image.dd /mnt
┌─[jw@parrot]─[~]
└──╼ $cd /mnt
┌─[jw@parrot]─[/mnt]
└──╼ $ls
noise_samples.zip  photos
```

Navigate to photos directory to see what it contains,the directory seems to contain three photos.

```
┌─[jw@parrot]─[/mnt]
└──╼ $cd photos/
┌─[jw@parrot]─[/mnt/photos]
└──╼ $ls
london_bridge.jpeg   london-eye-1447071.jpg  'Millenium Bridge.jpg'
```

Let’s see what we can get from analyzing the exif data from these images.

```
┌─[jw@parrot]─[/mnt/photos]
└──╼ $exiftool london-eye-1447071.jpg
ExifTool Version Number         : 12.57
File Name                       : london-eye-1447071.jpg
Directory                       : .
File Size                       : 387 kB
File Modification Date/Time     : 2021:08:05 20:41:32+03:00
File Access Date/Time           : 2021:08:05 03:00:00+03:00
File Inode Change Date/Time     : 2021:08:05 20:41:32+03:00
File Permissions                : -rwxr-xr-x
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Little-endian (Intel, II)
Image Description               : OLYMPUS DIGITAL CAMERA
Make                            : OLYMPUS CORPORATION
Camera Model Name               : C750UZ
Orientation                     : Horizontal (normal)
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : Adobe Photoshop CS2 Windows
Modify Date                     : 2006:09:13 01:55:53
```

```
┌─[jw@parrot]─[/mnt/photos]
└──╼ $exiftool london_bridge.jpeg
ExifTool Version Number         : 12.57
File Name                       : london_bridge.jpeg
Directory                       : .
File Size                       : 3.8 MB
File Modification Date/Time     : 2021:08:05 20:41:32+03:00
File Access Date/Time           : 2021:08:05 03:00:00+03:00
File Inode Change Date/Time     : 2021:08:05 20:41:32+03:00
File Permissions                : -rwxr-xr-x
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Artist                          : steghide password: cheese on toast
Y Cb Cr Positioning             : Centered
Image Width                     : 5614
Image Height                    : 3743
Encoding Process                : Progressive DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
```

```
┌─[jw@parrot]─[/mnt/photos]
└──╼ $exiftool 'Millenium Bridge.jpg'
ExifTool Version Number         : 12.57
File Name                       : Millenium Bridge.jpg
Directory                       : .
File Size                       : 318 kB
File Modification Date/Time     : 2021:08:05 20:41:32+03:00
File Access Date/Time           : 2021:08:05 03:00:00+03:00
File Inode Change Date/Time     : 2021:08:05 20:41:32+03:00
File Permissions                : -rwxr-xr-x
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 96
Y Resolution                    : 96
Exif Byte Order                 : Little-endian (Intel, II)
Copyright                       : desktopsky.com
Padding                         : (Binary data 4122 bytes, use -b option to extract)
XMP Toolkit                     : Image::ExifTool 11.88
Location                        : name of the challenge
Image Width                     : 1920
Image Height                    : 1080
```

We got some information from above analysis. The first is 'Artist: steghide password: cheese on toast' from london_bridge.jpeg and the second is 'location: name of the challenge' from millenium bridge.jpg means the location is 'spectrum' , but let’s just note it down, it may be helpful later.

Also we have 'noise_samples.zip' let'stry to unzip

```
┌─[jw@parrot]─[/mnt]
└──╼ $unzip noise_samples.zip
Archive:  noise_samples.zip
[noise_samples.zip] brown.wav password:
password incorrect--reenter:
   skipping: brown.wav              incorrect password
   skipping: location.wav           incorrect password
   skipping: wahwah.wav             incorrect password
   skipping: white.wav              incorrect password
```

This too is password protected !! Trying the password from the photos information didn’t yield any result either.

We can try to brute force the password. Let’s do it!!. We will use 'fcrackzip' for this purpose and we will use a dictionary brute forcing.

```
┌─[jw@parrot]─[/mnt]
└──╼ $ls
noise_samples.zip  photos
┌─[jw@parrot]─[/mnt]
└──╼ $fcrackzip -D -p /usr/share/wordlists/rockyou.txt noise_samples.zip
possible pw found: garfield ()
```

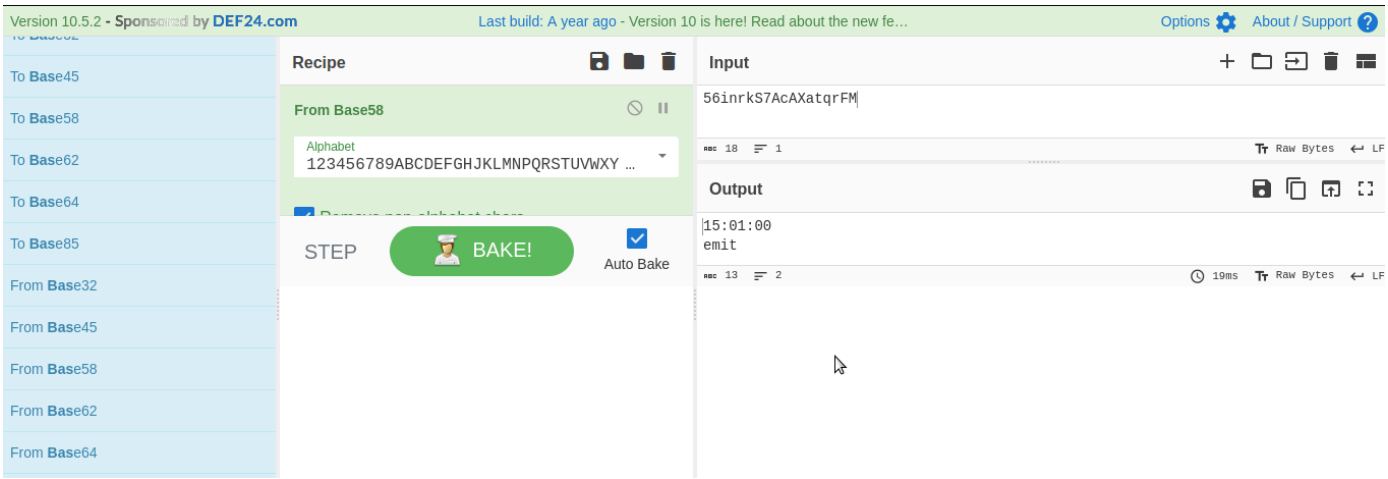
We found the password 'garfield' for zipped file, let's us see inside the file. First create another directory 'noise_samples' copy the file to that directory then unzip the file.

```
┌─[jw@parrot]─[~/noise_samples]
└──╼ $unzip noise_samples.zip
Archive:  noise_samples.zip
[noise_samples.zip] brown.wav password:
  inflating: brown.wav
  inflating: location.wav
  inflating: wahwah.wav
  inflating: white.wav
┌─[jw@parrot]─[~/noise_samples]
└──╼ $ls
brown.wav  location.wav  noise_samples.zip  wahwah.wav  white.wav
```

There are some audio files, remember we saw that steghide information from photos can also be used for hiding data in audio files. Let's see if we can extract something from these files.

```
┌─[jw@parrot]─[~/noise_samples]
└──╼ $steghide extract -sf white.wav -p "cheese on toast"
wrote extracted data to "stardate.txt".
┌─[jw@parrot]─[~/noise_samples]
└──╼ $cat stardate.txt
56inrkS7AcAXatqrFM
```

We got something '56inrkS7AcAXatqrFM' let's us use cyberchef the content of this cyphertext.

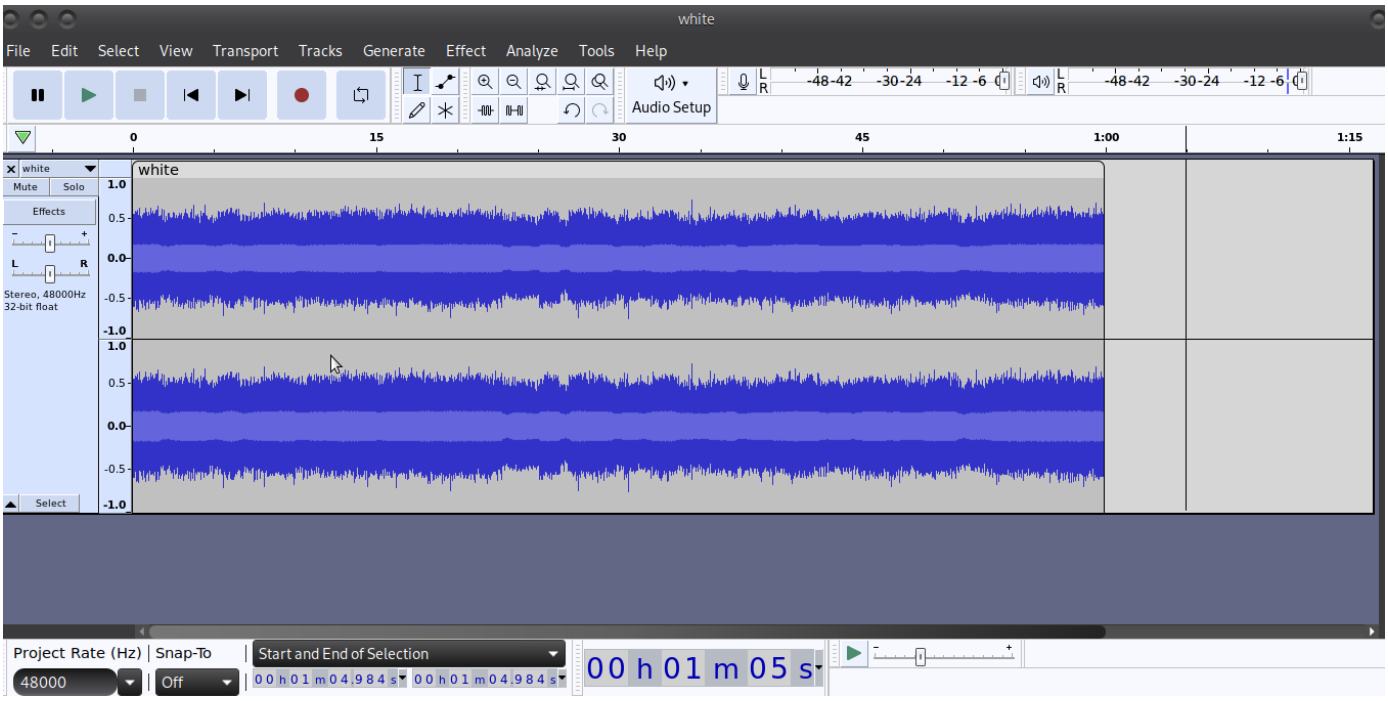


We got a timestamp, but what is this "emit" . Let's observe carefully, what if we reverse "emit", we get "time" so if we reverse "15:01:00" we get "00:10:51".

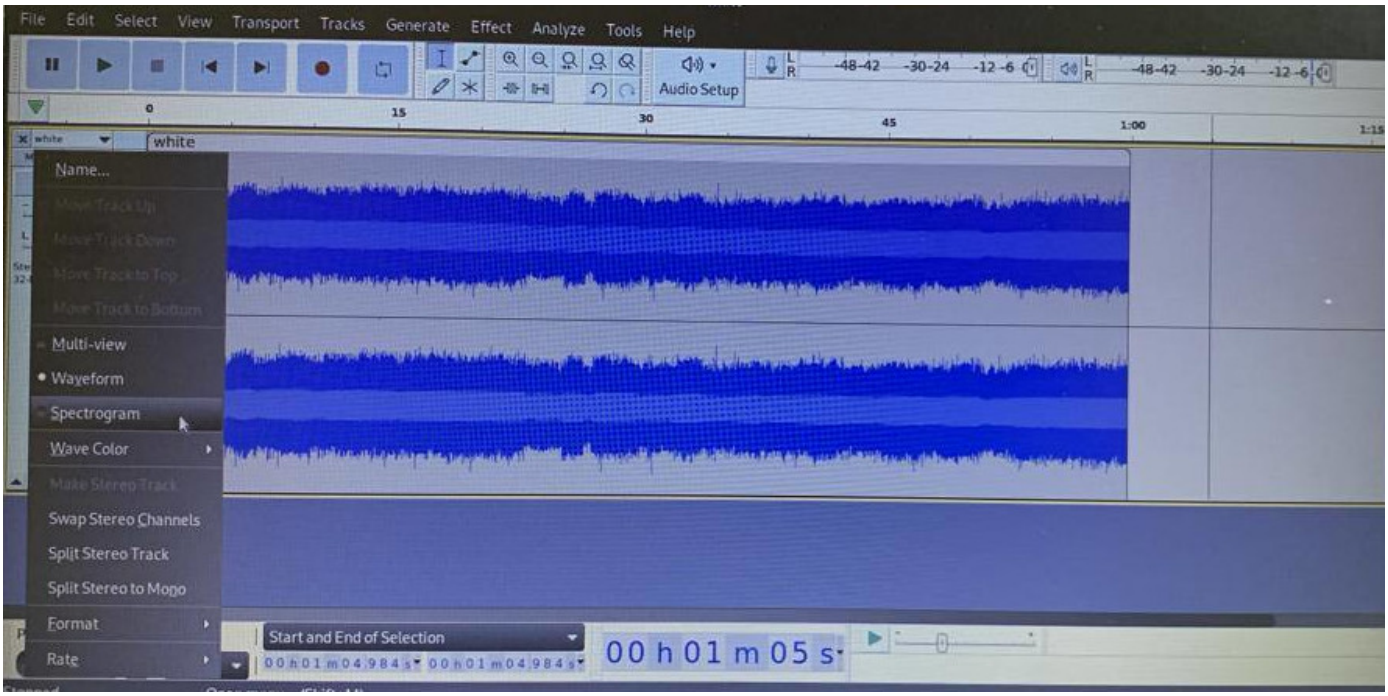
From the submission; What time is the meeting happening?

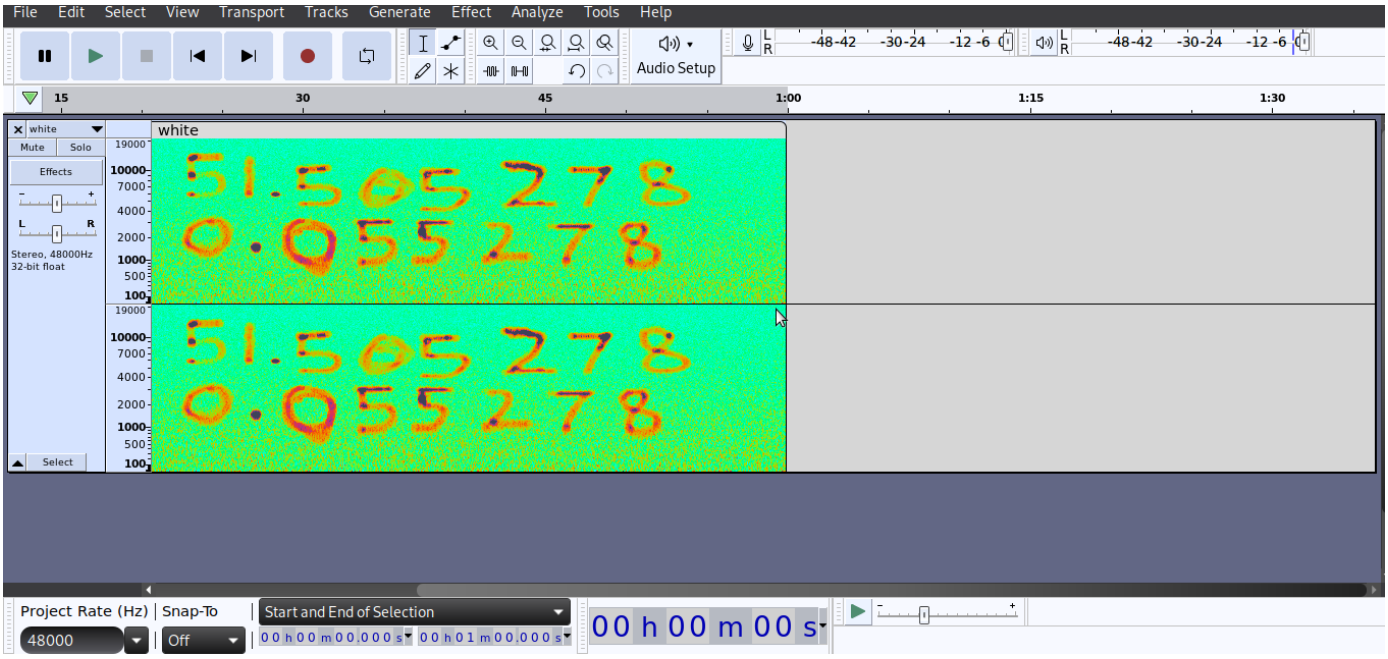
Answer: **00:10:51**

Time to find the location now. Let's get back to those audio files and load them into a audio software, we will use Audacity for this purpose. starting with white.wav.



Nothing we got, but from the photos information we have "location: name of the challenge", so we can navigate to 'spectrogram' to see if we can get anything.



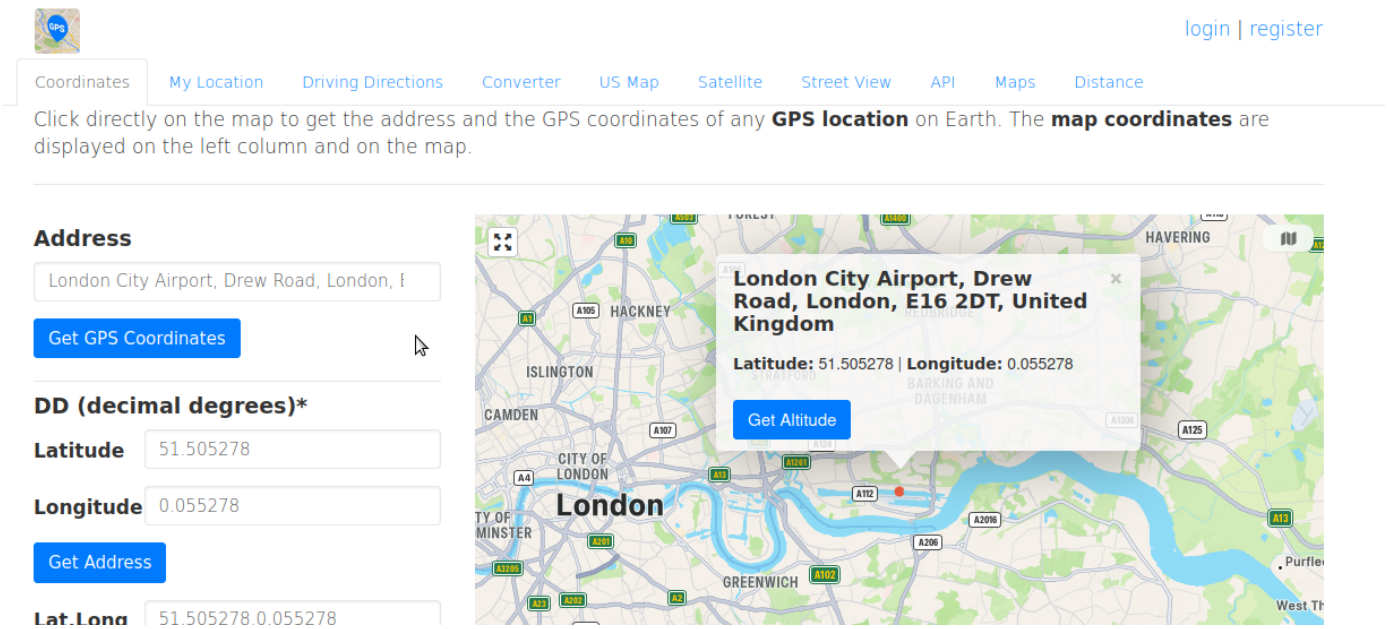


We found GPS coordinates!!!!

From the submission; What are the supposed coordinates for the deal?

Answer: **51.505278,0.055278**

Let's find the corresponding location by using gps-coordnates.net.



I'm done!!!!

BLUE TEAMS LABS ONLINE@cyberhunter
27/08/2024 13:58