HackerFrogs Afterschool
Digital Forensics: Steganography

Class:
Digital Forensics

Workshop Number:
AS-FOR-02

Document Version:
1.75

Special Requirements:
Registered account at tryhackme.com

Welcome to HackerFrogs Afterschool!

Hey there HackerFrogs!

This is the second intro to Digital Forensics basics workshop

In the previous workshop we learned about the following Digital Forensic concepts:

EXIF and Image File Metadata

Metadata is data that provides information about other data.
In the context of digital image files, each file has a plethora of metadata information which is generated when the picture is taken

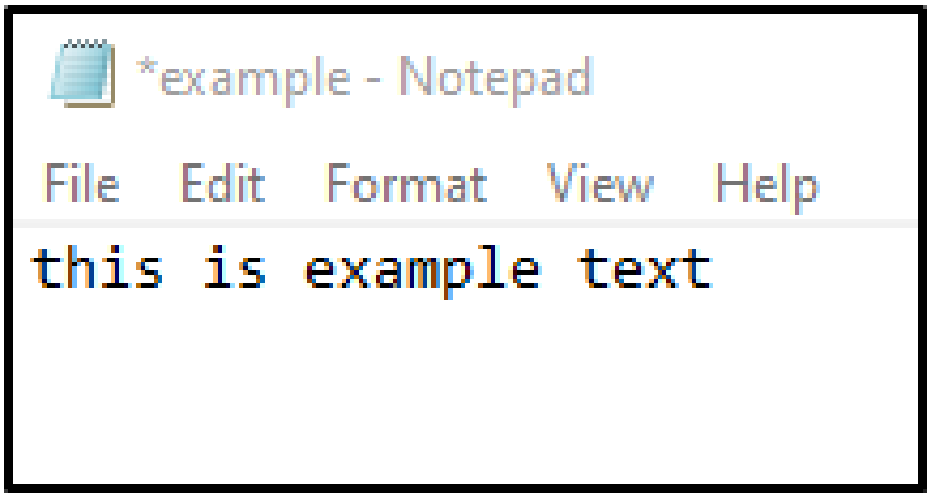
Global Positioning System	
GPS Altitude	31.9 m
GPS Latitude	6deg 14' 7.620"
GPS Longitude	106deg 49' 30.210"
Image Information	
Date and Time	2018:08:24 15:47:27
Manufacturer	Apple
Model	iPhone 6s
Photograph Information	
Aperture	F2.2
Exposure Bias	0 EV
Exposure Mode	Auto
Exposure Program	Auto
Exposure Time	1/874 s
Flash	No, auto
FNumber	F2.2
Focal Length	4.2 mm
ISO Speed Ratings	25
Metering Mode	Multi-segment
Shutter speed	1/874 s
White Balance	Auto

EXIF and Image File Metadata

Photo metadata, called EXIF, is easily accessible, and can reveal information about the device used to take the photo, or even GPS location data about where the photo was taken

Global Positioning System	
GPS Altitude	31.9 m
GPS Latitude	6deg 14' 7.620"
GPS Longitude	106deg 49' 30.210"
Image Information	
Date and Time	2018:08:24 15:47:27
Manufacturer	Apple
Model	iPhone 6s
Photograph Information	
Aperture	F2.2
Exposure Bias	0 EV
Exposure Mode	Auto
Exposure Program	Auto
Exposure Time	1/874 s
Flash	No, auto
FNumber	F2.2
Focal Length	4.2 mm
ISO Speed Ratings	25
Metering Mode	Multi-segment
Shutter speed	1/874 s
White Balance	Auto

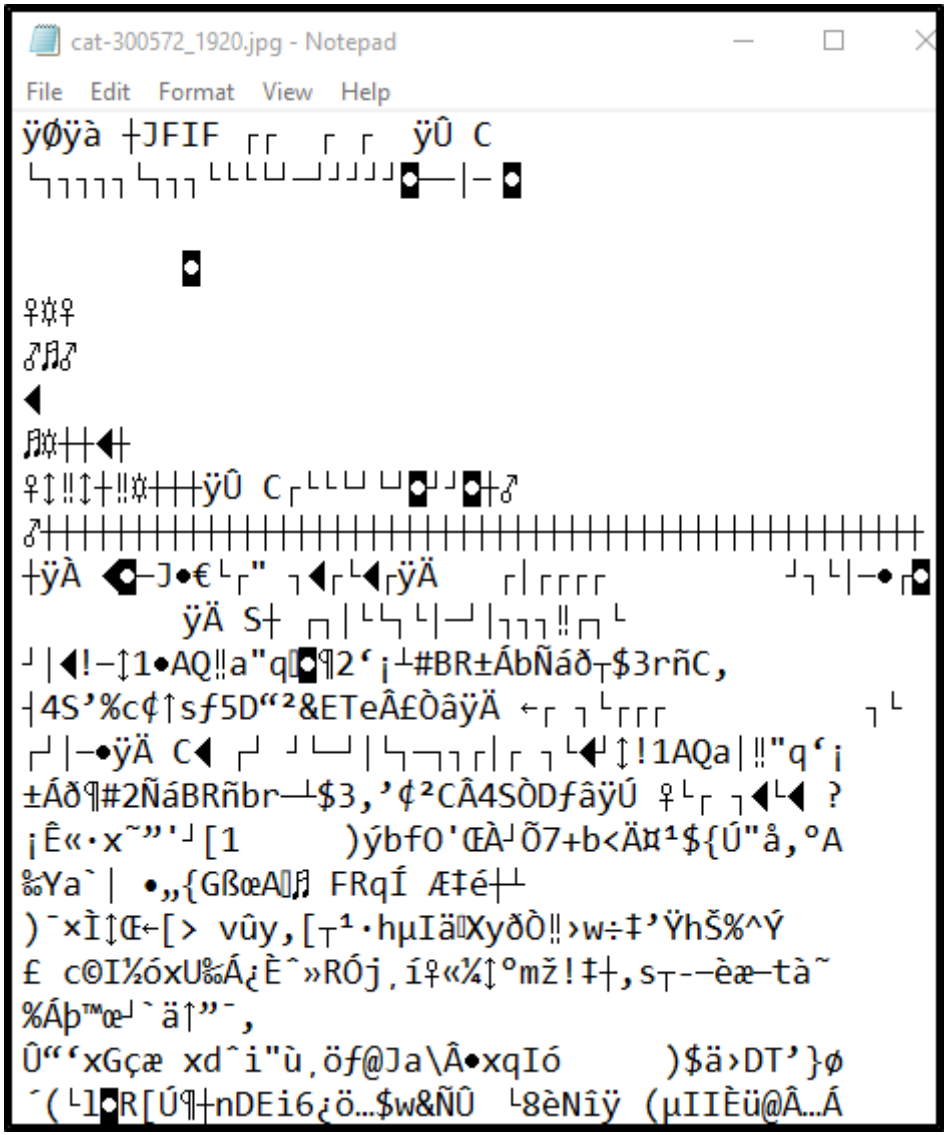
The Contents of Image Files



We learned about the contents of picture files, and the difference between text and data. Text is computer content that is meant to be read by humans

The Contents of Image Files

And data content is meant to be read by computers / software, and is illegible if not accessed through the proper program / app



Embedded Files

Lastly, we learned about embedded files, and how embedded files can be extracted and examined using specialized software

This Workshop's Topics

- steganography

- steghide tool

Steganography

Stegranography is the process of hiding information in another medium. E.g., hiding a hidden message in a picture

Steganography

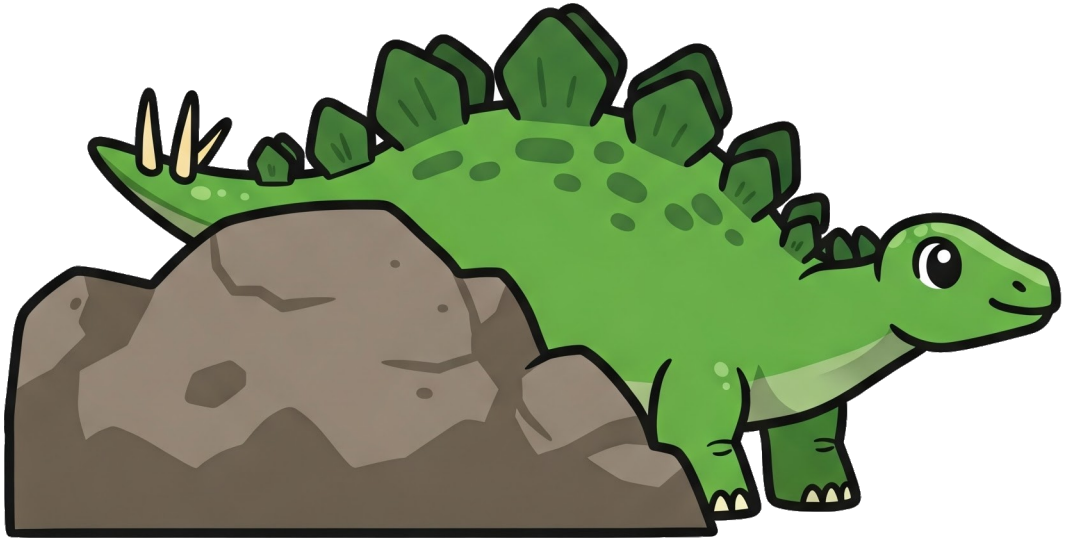
When dealing with digital files, steganography can be used to embed files or messages into different media files, like pictures, videos, and audio files

Let's Learn More at TryHackMe.com

We're going to be using a TryHackMe room to learn more about steganography and the Steghide program. Login to tryhackme.com, then navigate to the link below (task 4):

https://tryhackme.com/r/room/ctfcollectionvol1

Steghide



Steghide is a program which can be used to embed files into certain types of picture and audio files

Steghide “Info”

```
└─$ steghide info Extinction_1577976250757.jpg
"Extinction_1577976250757.jpg":
  format: jpeg
  capacity: 1.3 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "Final_message.txt":
```

If we use the Steghide command with the `info` function, then we can see if there are any embedded files in the media file

Steghide “Extract”

```
┌──(theshyhat㉿hackerfrogs)-[~/Downloads]
└─$ steghide extract -sf Extinction_1577976250757.jpg
Enter passphrase:
wrote extracted data to "Final_message.txt".
```

If we use the Steghide command with the `extract` function, any embedded files will be extracted and copied to the current directory

Let's Try Hiding a File

We'll use the current TryHackMe AttackBox terminal to embed a new file into the dinosaur picture file with Steghide

Steghide

When we use steghide to embed a file into a picture file, we can break down the command like this:

Steghide – Embedding Files

```
1 steghide 2 embed 3 -cf example.jpg 4 -p 'password123' 5 -ef secret.txt
```

1: steghide <= the command itself

2: embed <= the function to be done

3: -cf example.jpg <= the file to be embedded

4: -p 'password123' <= the passphrase to use

5: -ef secret.txt <= the file to embed

Steghide – Getting File Info

```
└─$ steghide info example.jpg
"example.jpg":
  format: jpeg
  capacity: 124.7 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
```

We can use the command illustrated above to list steghide info about a specific file, but if the file is password protected--

Steghide – Getting File Info

```
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

We won't get any info from the program if we don't input the correct password

Steghide – Getting File Info

```
Enter passphrase:
  embedded file "secret.txt":
    size: 25.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

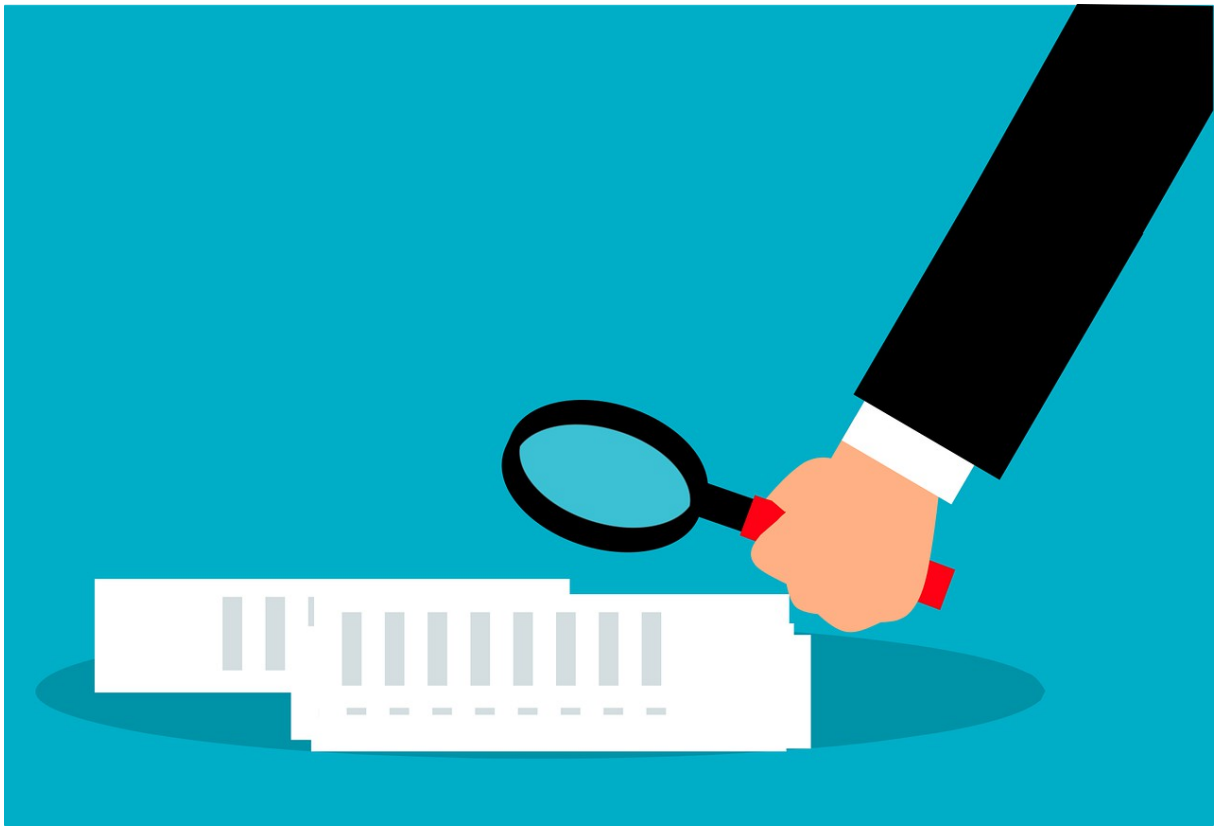
But if we supply the correct password, it'll tell us about any files embedded using Steghide

Steghide – Extracting Files

```
└─$ steghide extract -sf example.jpg
Enter passphrase:
wrote extracted data to "secret.txt".
```

Finally, we can use a command like the above to extract embedded files using Steghide

Summary



Let's review the digital forensics concepts we learned in this workshop:

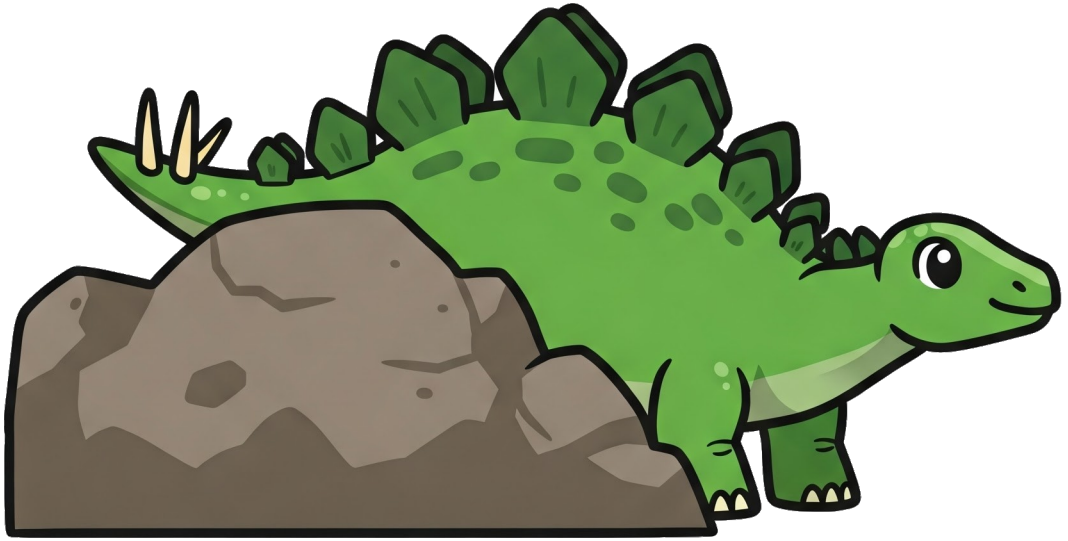
Steganography

Stegranography is the process of hiding information in another medium. E.g., hiding a hidden message in a picture

Steganography

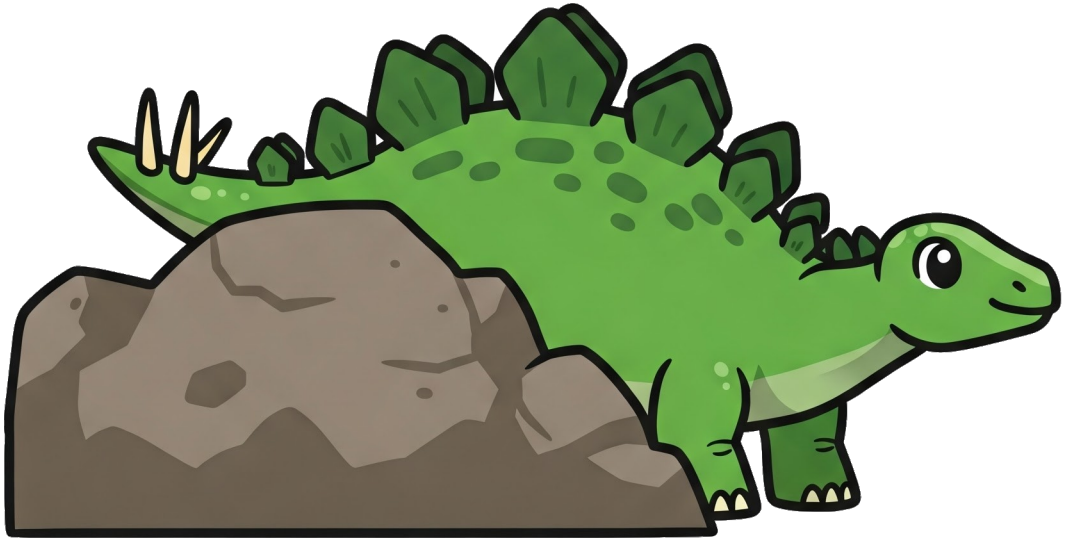
When dealing with digital files, steganography can be used to embed files or messages into different media files, like pictures, videos, and audio files

Steghide



Steghide is a program which can be used to embed files into certain types of picture and audio files

Steghide



Embedding files with Steghide can use password protection, for added security

What's Next?

In the next digital forensics workshop, we'll introduce network traffic analysis with the TryHackMe platform.

Extra Credit

Looking for more study material on this workshop's topics?

See this video's description for links to supplemental documents and exercises!

Until Next Time, HackerFrogs!