Forensic Analysis Notes:

Case Information

Case: analyze sunset.jpg file

Analyst: Julius Niyonzima

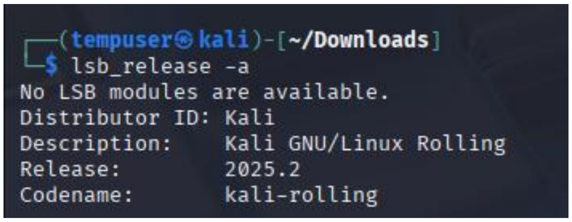
Professor: Dr. Chris Lamb

Email: jnbfg@umkc.edu

Id	File name	Hash(None)	Notes
1	Sunset.jpg	Verified using file command	Suspected steganography

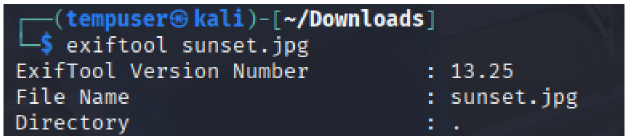
Tools and Environment

Kali Linux v2025.2



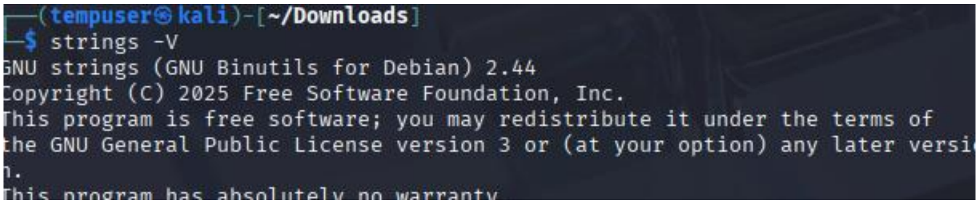
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2025.2
Codename:       kali-rolling
```

Exiftool v13.25



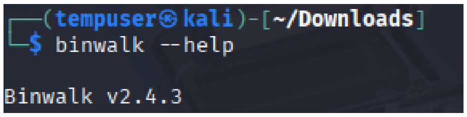
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ exiftool sunset.jpg
ExifTool Version Number         : 13.25
File Name                       : sunset.jpg
Directory                       : .
```

Strings v2.44



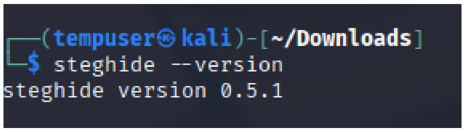
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ strings -V
GNU strings (GNU Binutils for Debian) 2.44
Copyright (C) 2025 Free Software Foundation, Inc.
This program is free software; you may redistribute it under the terms of
the GNU General Public License version 3 or (at your option) any later versi
n.
This program has absolutely no warranty
```

Binwalk v2.4.3



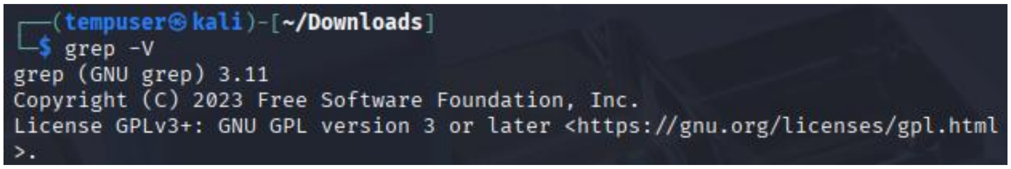
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ binwalk --help

Binwalk v2.4.3
```

Steghid v0.5.1



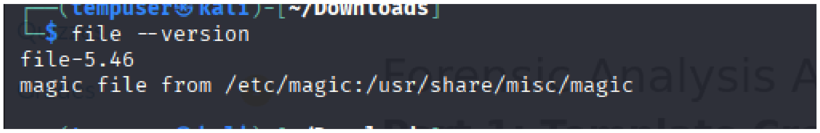
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ steghide --version
steghide version 0.5.1
```

Grep v3.11



```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ grep -V
grep (GNU grep) 3.11
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html
>.
```

File v5.46



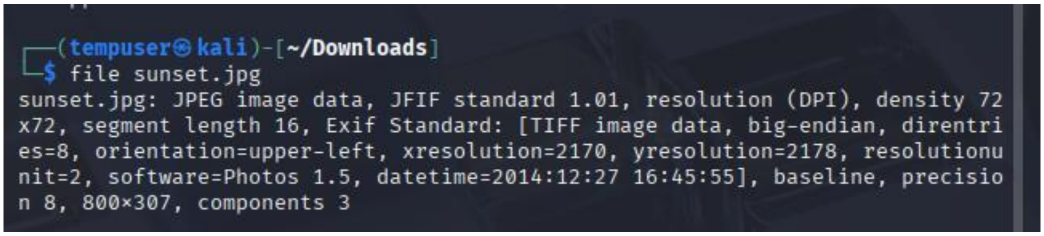
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ file --version
file-5.46
magic file from /etc/magic:/usr/share/misc/magic
```

Analysis Steps :

Step1: Verified file type using the file command

Command: File sunset.jpg

Result: the file was confirmed to be a JPEG image



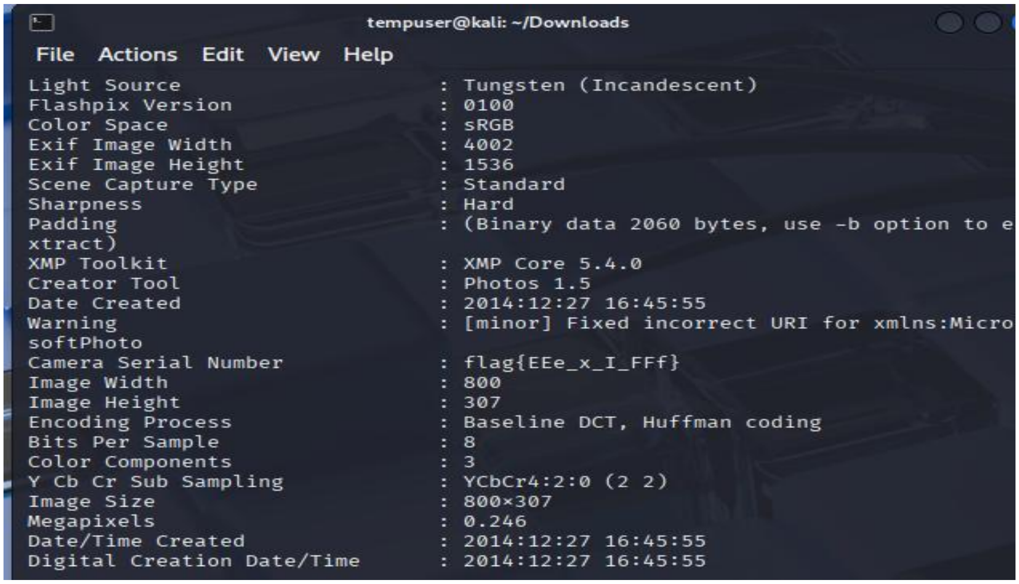
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ file sunset.jpg
sunset.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72
x72, segment length 16, Exif Standard: [TIFF image data, big-endian, direntri
es=8, orientation=upper-left, xresolution=2170, yresolution=2178, resolutionu
nit=2, software=Photos 1.5, datetime=2014:12:27 16:45:55], baseline, precisio
n 8, 800×307, components 3
```

Status: Worked

Step2 : Check the metadata with exfitool

Command : exiftool sunset.jpg

Result: The file was confirmed to be taken with a standard camera, with standard camera info. Flag information was detected



```
Light Source                    : Tungsten (Incandescent)
Flashpix Version                : 0100
Color Space                     : sRGB
Exif Image Width                : 4002
Exif Image Height               : 1536
Scene Capture Type              : Standard
Sharpness                       : Hard
Padding                         : (Binary data 2060 bytes, use -b option to e
xtract)
XMP Toolkit                     : XMP Core 5.4.0
Creator Tool                    : Photos 1.5
Date Created                    : 2014:12:27 16:45:55
Warning                         : [minor] Fixed incorrect URI for xmlns:Micro
softPhoto
Camera Serial Number            : flag{EEe_x_I_FFf}
Image Width                     : 800
Image Height                    : 307
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 800x307
Megapixels                      : 0.246
Date/Time Created               : 2014:12:27 16:45:55
Digital Creation Date/Time      : 2014:12:27 16:45:55
```

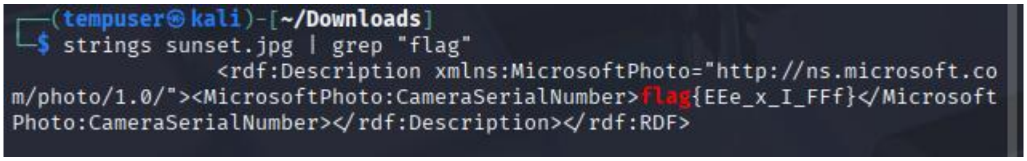
Status: Worked

Step3: Searched for readable string using strings with grep

Command : strings sunset.jpeg | grep “flag”

Result: Extracted all readable ASCII strings from binary-formatted JPEG.

The flag was found to be embedded in the XMP metadata



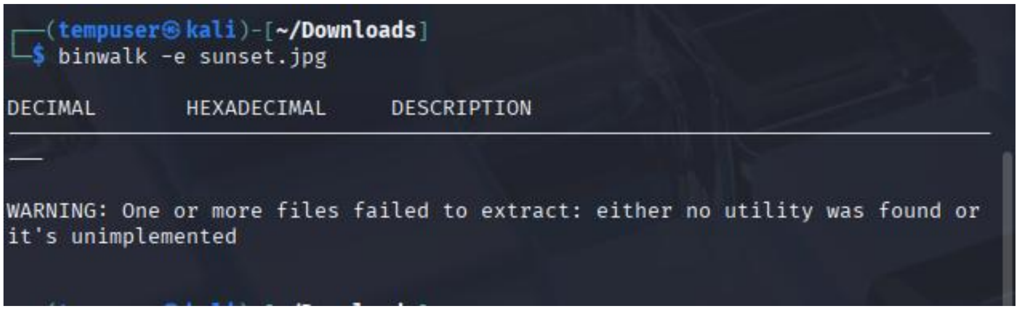
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ strings sunset.jpg | grep "flag"
            <rdf:Description xmlns:MicrosoftPhoto="http://ns.microsoft.co
m/photo/1.0/"><MicrosoftPhoto:CameraSerialNumber>flag{EEe_x_I_FFf}</Microsoft
Photo:CameraSerialNumber></rdf:Description></rdf:RDF>
```

Status: Flag confirmed

Step4: Tested for embedded file using binwalk

Command : binwalk -e sunset.jpg

Result: failed to extract embedded data from the file



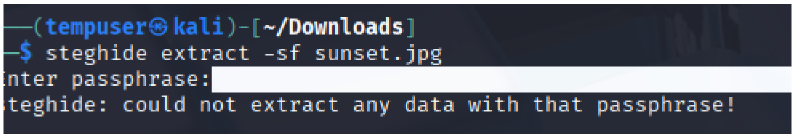
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ binwalk -e sunset.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
--

WARNING: One or more files failed to extract: either no utility was found or
it's unimplemented
```

Status: Did not work

Step 4: Attempted Steganography extraction

Command: steghide extract -sf sunset.jpg

Result: No phrase was provided. I was not able to extract anything from the file



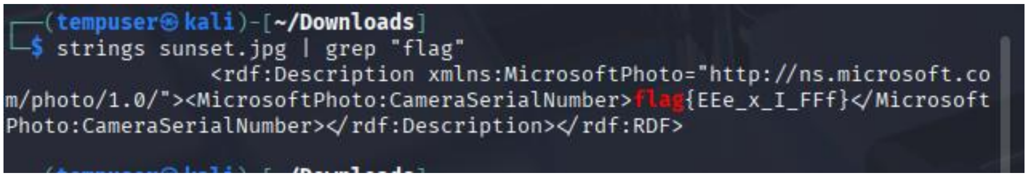
```
──(tempuser㉿kali)-[~/Downloads]
─$ steghide extract -sf sunset.jpg
nter passphrase:
teghide: could not extract any data with that passphrase!
```

Status: Did not work

Findings:

After running various commands, I used the strings command with grep, and the flag was revealed.

Flag {EE_xt_FFr}



```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ strings sunset.jpg | grep "flag"
                <rdf:Description xmlns:MicrosoftPhoto="http://ns.microsoft.co
m/photo/1.0/"><MicrosoftPhoto:CameraSerialNumber>flag{EEe_x_I_FFf}</Microsoft
Photo:CameraSerialNumber></rdf:Description></rdf:RDF>

  ┌──(tempuser㉿kali)-[~/Downloads]
```

No other additional details were uncovered during the analysis

Conclusion.

The flag was successfully located with the help of the strings and grep commands. It is recommended to use hashing and steganography for file integrity and obfuscation.