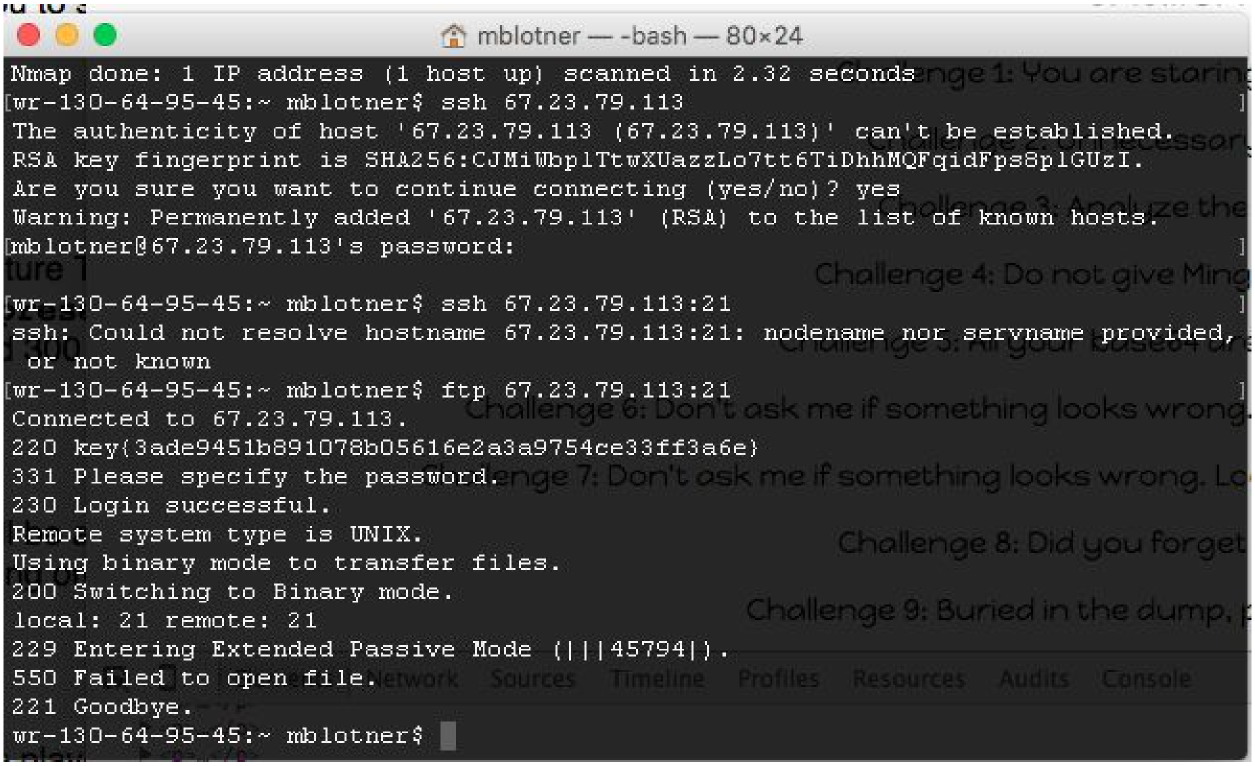
Melissa Blotner, Brian Cefali, Michael Seltzer, Becky Cutler
Comp 116: Security

Capture the Flag 2015

- A screenshot of the flag
- The exact location of the flag (path or file name)
- The exploit or methodology used to find the flag

Flags Found:

1. Unnecessary Service - FTP
    a. Screenshot:

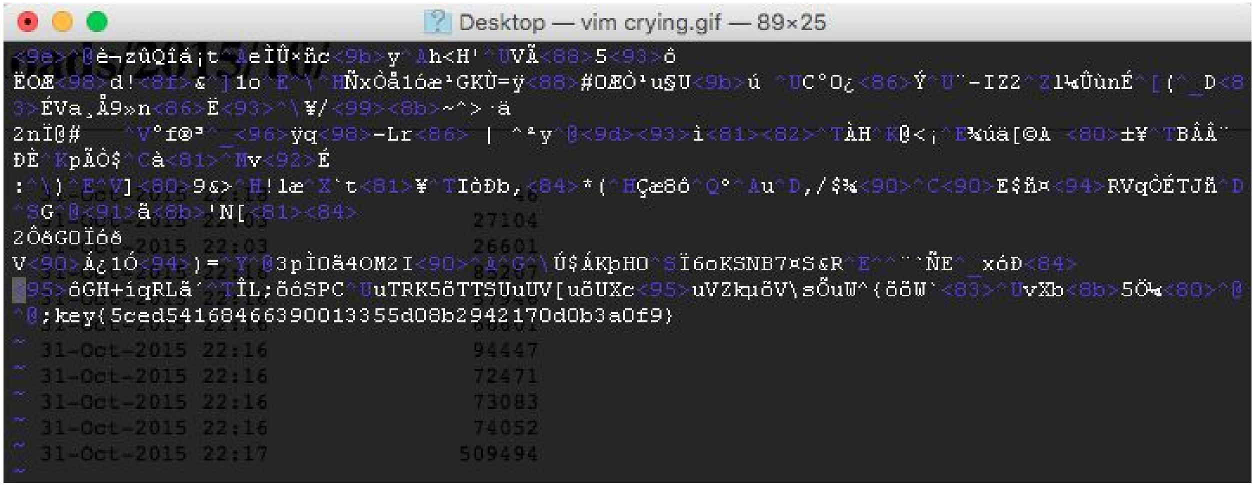


```
Nmap done: 1 IP address (1 host up) scanned in 2.32 seconds
[wr-130-64-95-45:~ mblotner$ ssh 67.23.79.113
The authenticity of host '67.23.79.113 (67.23.79.113)' can't be established.
RSA key fingerprint is SHA256:CJMiWbplTtwXUazzLo7tt6TiDhhMQFqidFps8plGUzI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '67.23.79.113' (RSA) to the list of known hosts.
[mblotner@67.23.79.113's password:

[wr-130-64-95-45:~ mblotner$ ssh 67.23.79.113:21
ssh: Could not resolve hostname 67.23.79.113:21: nodename nor servname provided,
 or not known
[wr-130-64-95-45:~ mblotner$ ftp 67.23.79.113:21
Connected to 67.23.79.113.
220 key{3ade9451b891078b05616e2a3a9754ce33ff3a6e}
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Switching to Binary mode.
local: 21 remote: 21
229 Entering Extended Passive Mode (|||45794|).
550 Failed to open file.
221 Goodbye.
wr-130-64-95-45:~ mblotner$
```

    b. Location of the flag:
        i. ftp 67.23.79.113:21
    c. Methodology
        i. By performing an NMAP scan on 67.23.79.113, there were a number of services found - one in particular, FTP, seemed unnecessary. FTP was on port 21 on this server. FTP was unsecured so we ran “ftp 67.23.79.113:21” and saw the flag after connecting to the server.

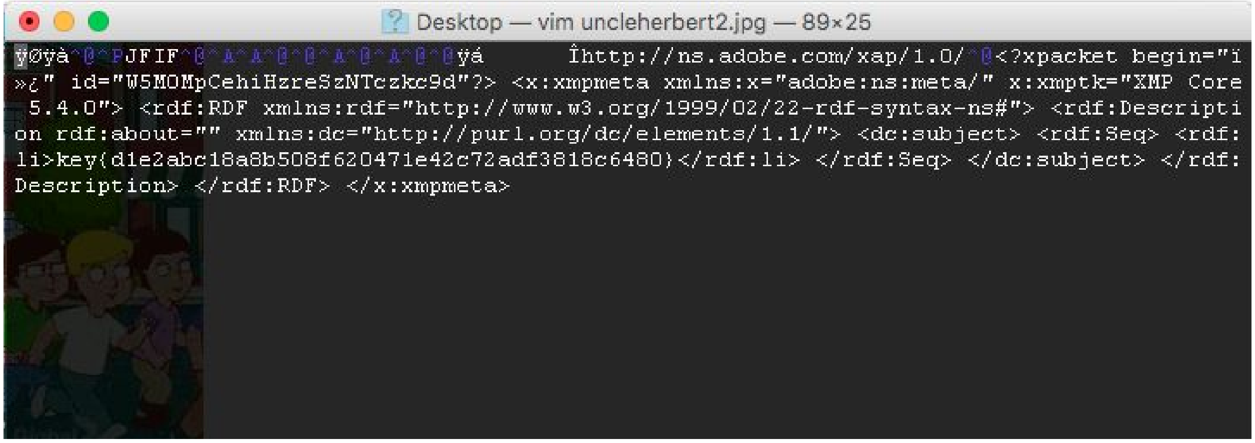
2. crying.gif
    a. Screenshot



```
<9e>^@è¬zûQîá¡t^AeÌÛ×ñc<9b>y^Ah<H'^UVÅ<88>5<93>ô
ËOÆ<98>d!<8f>&^]1o^E^\^HÑxÒå1óæ¹GKÙ=ÿ<88>#OÆÒ¹u§U<9b>ú ^UC°O¿<86>Ý^U¨-IZ2^Z1¼ÛùnÉ^[(^_D<8
3>ÉVa¸Å9»n<86>Ë<93>^\¥/<99><8b>~^>·ä
2nÏ@#    ^V°f®³^_<96>ÿq<98>-Lr<86> | ^ªy^@<9d><93>i<81><82>^TÀH^K@<¡^E¾úä[©A <80>±¥^TBÂÂ¨
ÐÈ^KpÃÒ$^Cà<81>^Mv<92>É
:^\)^E^V]<80>9&>^H!læ^X`t<81>¥^TIòÐb,<84>*(^HÇæ8ô^Q°^Au^D,/$¾<90>^C<90>E$ñ¤<94>RVqÒÉTJñ^D
^SG^@<91>ã<8b>'N[<81><84>
2ÔåGOÏóå
V<90>Á¿1Ó<94>)=^Y^@3pÌOã4OM2I<90>^A^G^\Ú$ÁKþHO^SÏ6oKSNB7¤S&R^E^^¨`ÑE^_xóÐ<84>
<95>ôGH+iqRLã´^TÎL;õõSPC^UuTRK5õTTSUuUV[uõUXc<95>uVZkµõV\sÕuW^{õõW`<83>^UvXb<8b>5Ö¼<80>^@
^@;key{5ced54168466390013355d08b2942170d0b3a0f9}
~
~
~
~
~
~
```

    b. Location of the flag
        i. crying.gif on page at 67.23.79.113/ctf/board.php
    c. Methodology
        i. Downloaded crying.gif to Desktop, and opened the file with VIM. Looking through the binary of the gif file, found the flag at the very bottom of the file.

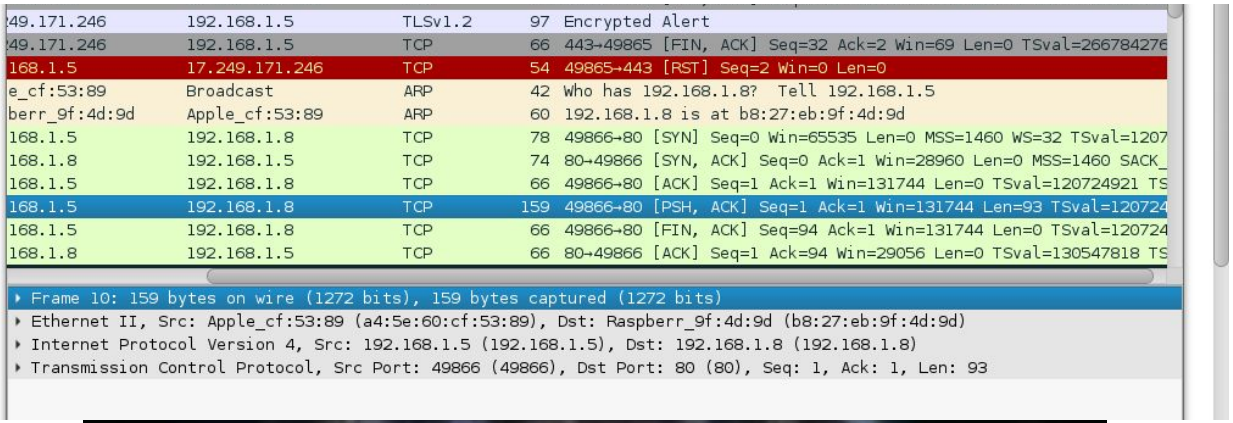
3. uncleherbert2.jpg
    a. Screenshot



```
ÿØÿà^@^PJFIF^@^A^A^@^@^A^@^A^@^@ÿá      Îhttp://ns.adobe.com/xap/1.0/^@<?xpacket begin="ï
»¿" id="W5M0MpCehiHzreSzNTczkc9d"?> <x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="XMP Core
 5.4.0"> <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:Descripti
on rdf:about="" xmlns:dc="http://purl.org/dc/elements/1.1/"> <dc:subject> <rdf:Seq> <rdf:
li>key{d1e2abc18a8b508f620471e42c72adf3818c6480}</rdf:li> </rdf:Seq> </dc:subject> </rdf:
Description> </rdf:RDF> </x:xmpmeta>
```

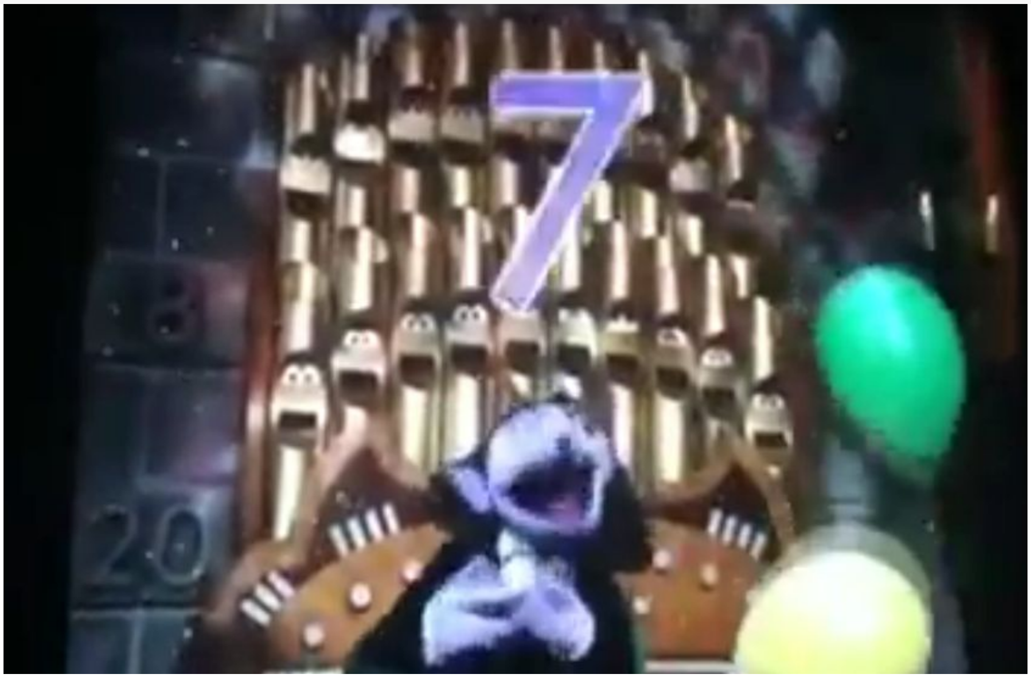
    b. Location of the flag
        i. uncleherbert2.jpg on page at 67.23.79.113/ctf/wp-content/uploads/2015/10/uncleherbert2.jpg
    c. Methodology
        i. Downloaded all of the picture files on page 67.23.79.113/ctf/wp-content/uploads/2015/10 and looked at the binary of all of them. uncleherbert2.jpg had a flag in the header of the binary file.

4. runme.exe



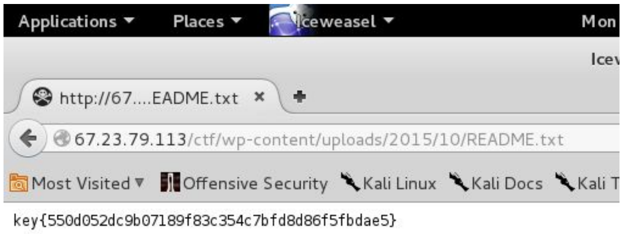
```
49.171.246     192.168.1.5      TLSv1.2    97  Encrypted Alert
49.171.246     192.168.1.5      TCP        66  443→49865 [FIN, ACK] Seq=32 Ack=2 Win=69 Len=0 TSval=266784276
168.1.5        17.249.171.246   TCP        54  49865→443 [RST] Seq=2 Win=0 Len=0
e_cf:53:89     Broadcast        ARP        42  Who has 192.168.1.8?  Tell 192.168.1.5
berr_9f:4d:9d  Apple_cf:53:89   ARP        60  192.168.1.8 is at b8:27:eb:9f:4d:9d
168.1.5        192.168.1.8      TCP        78  49866→80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=1207
168.1.8        192.168.1.5      TCP        74  80→49866 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_
168.1.5        192.168.1.8      TCP        66  49866→80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=120724921 TS
168.1.5        192.168.1.8      TCP       159  49866→80 [PSH, ACK] Seq=1 Ack=1 Win=131744 Len=93 TSval=120724
168.1.5        192.168.1.8      TCP        66  49866→80 [FIN, ACK] Seq=94 Ack=1 Win=131744 Len=0 TSval=120724
168.1.8        192.168.1.5      TCP        66  80→49866 [ACK] Seq=1 Ack=94 Win=29056 Len=0 TSval=130547818 TS

▸ Frame 10: 159 bytes on wire (1272 bits), 159 bytes captured (1272 bits)
▸ Ethernet II, Src: Apple_cf:53:89 (a4:5e:60:cf:53:89), Dst: Raspberr_9f:4d:9d (b8:27:eb:9f:4d:9d)
▸ Internet Protocol Version 4, Src: 192.168.1.5 (192.168.1.5), Dst: 192.168.1.8 (192.168.1.8)
▸ Transmission Control Protocol, Src Port: 49866 (49866), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 93
```



a. Screenshot
b. Location of the Flag
    i. 67.23.79.113/ctf/
c. Methodology
    i. Downloaded binary from site using cURL, then examined with a text editor. File clearly contained packet headers and bodies, so probably a pcap. Opened in Wireshark and one obvious tcp stream yielded a file request stream. The file’s contents were saved to disk by parsing the stream and removing the packet headers, and the mpeg-4 file was played using the default video player.

5. README.txt
   a. Screenshot



```
key{550d052dc9b07189f83c354c7bfd8d86f5fbdae5}
```

   b. Location of the Flag
      i. 67.23.79.113/ctf/wp-content/uploads/2015/10/README.txt
   c. Methodology
      i. Simply knowing the common Wordpress folders to watch out for got me to the /ctf/wp-content/uploads folder. The directory listing was enabled, so I navigated through the folders and found the README.txt file.