

Matt Long, Walton Lee, Skyler Tom, Max Cohen
CTF Write-up for Team 7

Challenge 1: You are staring right at it

```
matt@matt-Inspiron-N5110: ~  
~: nmap 67.23.79.113  
  
Starting Nmap 6.40 ( http://nmap.org ) at 2015-11-09 12:40 EST  
Nmap scan report for www.pupcast.com (67.23.79.113)  
Host is up (0.056s latency).  
Not shown: 991 closed ports  
PORT      STATE      SERVICE  
21/tcp    open       ftp  
22/tcp    open       ssh  
25/tcp    filtered   smtp  
80/tcp    open       http  
135/tcp   filtered   msrpc  
139/tcp   filtered   netbios-ssn  
445/tcp   filtered   microsoft-ds  
2222/tcp  open       Ethernet/IP-1  
3306/tcp  open       mysql  
  
Nmap done: 1 IP address (1 host up) scanned in 14.41 seconds  
~:  
~:  
~: ftp 67.23.79.113  
Connected to 67.23.79.113.  
220 key{3ade9451b891078b05616e2a3a9754ce33ff3a6e}  
Name (67.23.79.113:matt):
```

Path: nmap 67.23.79.113

Methodology: Obviously we should try attacking the server to see if there are any blatant vulnerabilities staring us in the face. Using the nmap command above and launching an nmap attack on the ip of the capture the flag game, we located multiple open ports on the CTF server. An attempt to connect via the ftp port, the first open one, yields a message containing one of the flags.

If we download the binary and open it up, we can see how to find the key:

Downloads — runme.exe (~/.Downloads) - Vim — vim runtime.exe — 81x24

```
5 <9f>^D\Li^G1ă<9e>ăw5VĖ^A^K@6^@^@6^@^@^@&b ^_6π^^İS<89>^H@E^@^@(<93>π@  
@^F'<8f>Ă^A^E^Qù«ôÄÊ^A>â2m^@^@^@P^D^@^@W<9f>^@^@öw5V<96>·^H@*^@^@*^@  
^@^@ÿÿÿÿÿÿÿ^^İS<89>^H^F@^A^H@^F^D@^Aπ^^İS<89>Ă^A^E^@^@^@^@Ä^A^Höw5V  
Ě¼^H@<^@^@<^@^@<π^^İS<89>,'ë<9f>M<9d>^H^F@^A^H@^F^D@^B,'ë<9f>M<9d>Ă  
A^Hπ^^İS<89>Ă^A^E^@^@^@^@^@^@^@^@^@^@^@^@öw5VĚ¼^H@N^@^@N^@^@  
^@,'ë<9f>M<9d>π^^İS<89>^H@E^@^@<9f>o@^@^@F^WêĂ^A^EĂ^A^HÂÊ@P^AT^M^H@^@  
^@^°Bÿÿ`?^@^@B^D^E^A^C^C^E^A^A^H  
6 ^G2^]·^@^@^@^@D^B@^@öw5V^_Â^H@J^@^@J^@^@^@π^^İS<89>,'ë<9f>M<9d>^H@E^@  
@<^@^@^@^@F·Ă^A^HÂ^A^E@PÂÊâGJ9^T^M ^Rq Ē<9e>^@^@B^D^E^_D^B^H  
7 ^GÊ@i^G2^]·^A^C^C^Göw5VLĂ^H@B^@^@B^@^@^@,'ë<9f>M<9d>π^^İS<89>^H@E^@^@4đ  
<8a>@^@^@FÔUĂ^A^EĂ^A^HÂÊ@P^AT^M âGJ:<80>^P^P^U[t^@^@A^A^H  
8 ^G2^]¹^GÊ@iöw5V<9b>Ă^H@<9f>^@^@<9f>^@^@^@,'ë<9f>M<9d>π^^İS<89>^H@E^@^@  
91>f-@^@^@FP\Ă^A^EĂ^A^HÂÊ@P^AT^M âGJ:<80>^X^P^Uri^@^@A^A^H  
9 ^G2^]¹^GÊ@iWatch the video. The key is the SHA1 sum of the number, as a word  
d in all caps, in the video.  
10 öw5V<9b>Ă^H@B^@^@B^@^@^@,'ë<9f>M<9d>π^^İS<89>^H@E^@^@4"+@^@^@F^O;Ă^A^E  
^A^HÂÊ@P^AT^MfâGJ:<80>^Q^P^U[^V^@^@A^A^H  
11 ^G2^]¹^GÊ@iöw5Vě^H@B^@^@B^@^@^@π^^İS<89>,'ë<9f>M<9d>^H@E^@^@4JP@^@^@Fr  
VĂ^A^HÂ^A^E@PÂÊâGJ:^T^Mf<80>^P^@âjH^@^@A^A^H  
12 █GÊ@j^G2^]¹öw5V\Li^H@B^@^@B^@^@^@,'ë<9f>M<9d>π^^İS<89>^H@E^@^@4-Ă@^@F  
ĚĂ^A^EĂ^A^HÂÊ@P^AT^MfâGJ:<80>^Q^P^U[^K^@^@A^A^H  
13 ^G2^]¹^GÊ@jöw5VYi^H@N^@^@N^@^@^@π^^İS<89>,'ë<9f>M<9d>^H@E^@^@JQ@^@^@Fr  
m Ă^A^HÂ^A^E@PÂÊâGJ:^T^Mg°^P^@ă]0^@^@A^A^H
```

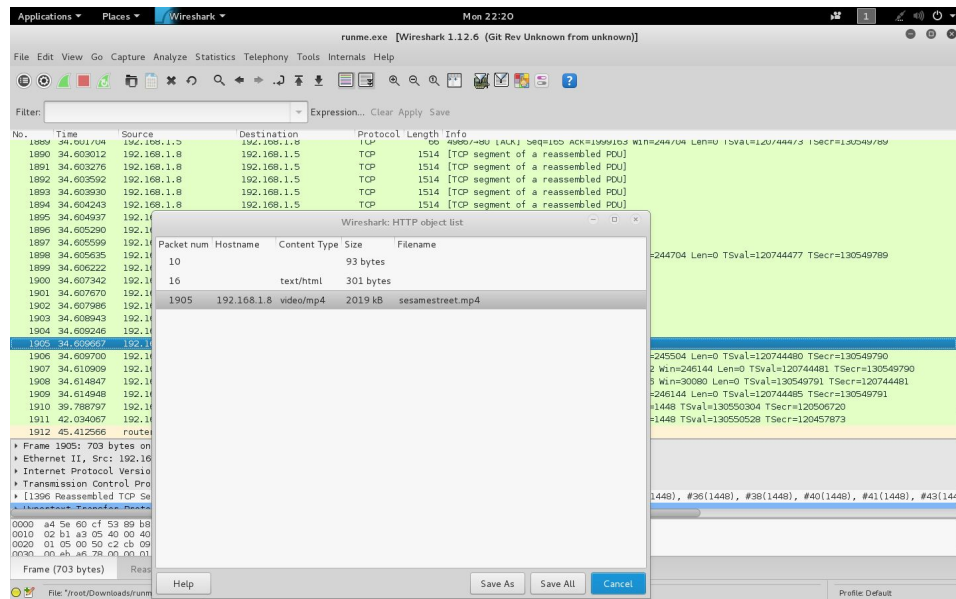
And if we look at a little more of the binary, we can see that it looks like a pcap file we can open in wireshark. We can also see that we want to watch the sesamestreet.mp4 file (the video)

```

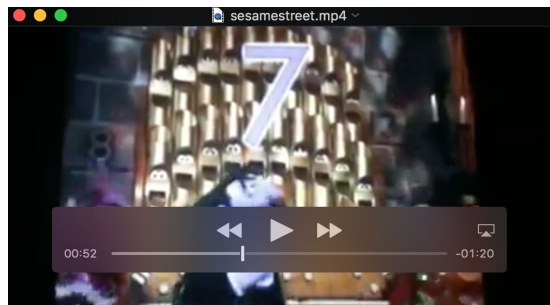
30 ^G2f^@^@^@D^B^@^@Ex5V<93>Á^C^@J^@^@^@ÏS<89>, 'ë<9f>M<9d>^H^E^@^@^@<^@^@^@F.^À^^A^HÄ^^A^E^@PÄË y<9a>$HBÑv ^Rq U^-^@^@B^D^E'^D^B^H
40 ^GÈ^G'^G2f^@^A^C^C^G^Ex5VÄÁ^C^@B^@^@^@B^@^@^@, 'ë<9f>M<9d>ÿ^^ÏS<89>^H^E^@^@4^L^@^@^@F^aÄÄ^^A^EÄ^^A^HÄË^@PHBÑv y<9a>%<80>^P^PUâ<85>^@^@A^A^H
41 ^G2ff^GÈ^G'^Ex5VöÁ^C^@æ^@^@^@æ^@^@^@, 'ë<9f>M<9d>ÿ^^ÏS<89>^H^E^@^@øy^D^@^@F^=¿Ä^^A^EÄ^^A^HÄË^@PHBÑv y<9a>%<80>^X^PU*~c>^@^@A^A^H
42 ^G2ff^GÈ^G'^GET /Movies/sesamestreet.mp4 HTTP/1.1^M
43 User-Agent: Wget/1.16.3 (darwin14.1.0)^M
44 Accept: */*^M
45 Accept-Encoding: identity^M
46 Host: 192.168.1.8^M
47 Connection: Keep-Alive^M
48 ^M
49 ^Ex5V^MÆ^C^@B^@^@^@B^@^@^@ÿ^^ÏS<89>, 'ë<9f>M<9d>^H^E^@^@4<9d>>91>^@^@F^YÖÄ^^A^HÄ^^A^E^@PÄË y<9a>%HBÖ^Z<80>^P^@ëô^K^K^@^@A^A^H
50 ^GÈ^G'^G2ff^Ex5VxÓ^C^@ê^E^@^@ê^E^@^@ÿ^^ÏS<89>, 'ë<9f>M<9d>^H^E^@^@EÜ<9d>>92>^@^@F^T,À^^A^HÄ^^A^E^@PÄË y<9a>%HBÖ^Z<80>^P^@ë2^1^@^@A^A^H
51 ^GÈ^G'^G2ffHTTP/1.1 200 OK^M
52 Date: Sun, 01 Nov 2015 02:25:09 GMT^M
53 Server: Apache/2.2.22 (Debian)^M
54 Last-Modified: Sun, 01 Nov 2015 01:58:41 GMT^M
55 [Tag: "40d4c-1ed1bd-52370fff86a56"^M
56 Accept-Ranges: bytes^M

```

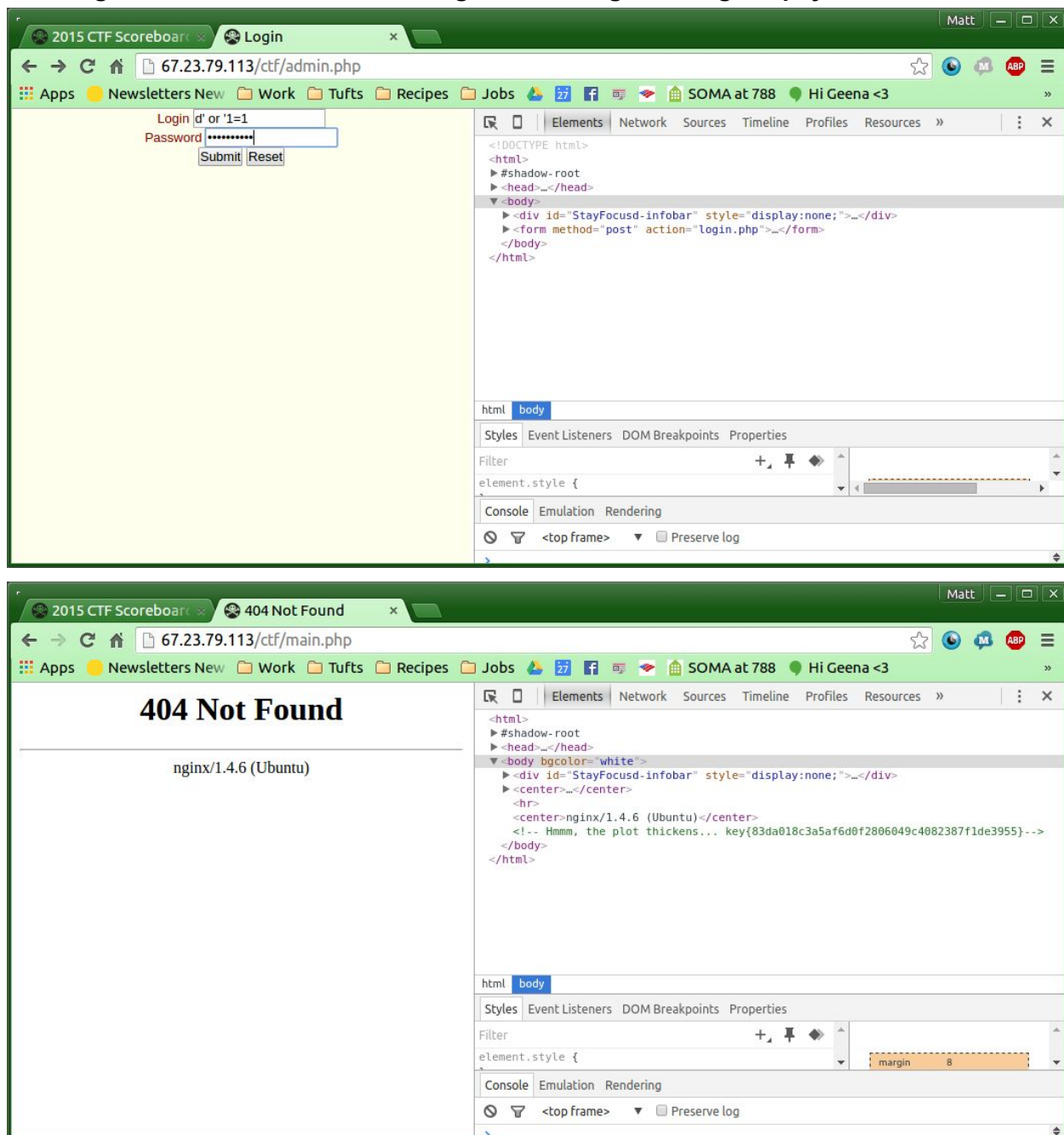
Then when we open it up in Wireshark, we can go to File->Export Objects->HTTP and we can see the sesamestreet.mp4 file and download it



We watch the video, see the number 7, put it in SHA1 as SEVEN and the key is **'cabd534c35ee6a39365f4ed3bce4eafdcc3d4b8d'**



Challenge 6: Don't ask me if something looks wrong. Look again, pay careful attention:



Challenge 9: Buried in the dump, part 1: wide open:



Path: 67.23.79.113/ctf/wp-content/uploads/2015/10/README.txt

Methodology: In the beginning of the class, we learned about how many WordPress sites have files/pages/content out in the open under the /wp-content folder. Browsing through the easily accessible wp-content/uploads path, which is pretty standard on wordpress pages, on the capture-the-flag server gives a series of image files, along with a basic README.txt file containing one of the codes.

Challenge 10: Buried in the dump, part 1A: metadata p0rn:

[illegible]

Along with the README.txt, there were a bunch of pictures of Uncle Herbert from Family Guy, named uncleherbert1 - uncleherbert8. We wanted to check if any keys were hidden in those files, so we ran `cat uncleherbert* > herbert_dump` to concatenate all the uncleherbert files into one, then ran `grep --binary-files=text "key" herbert_dump` to search through the files for the string "key". The `--binary-files=text` argument ended up being unnecessary, because the key was hidden in plain text in the metadata of uncleherbert2.jpg and not within the binary data.