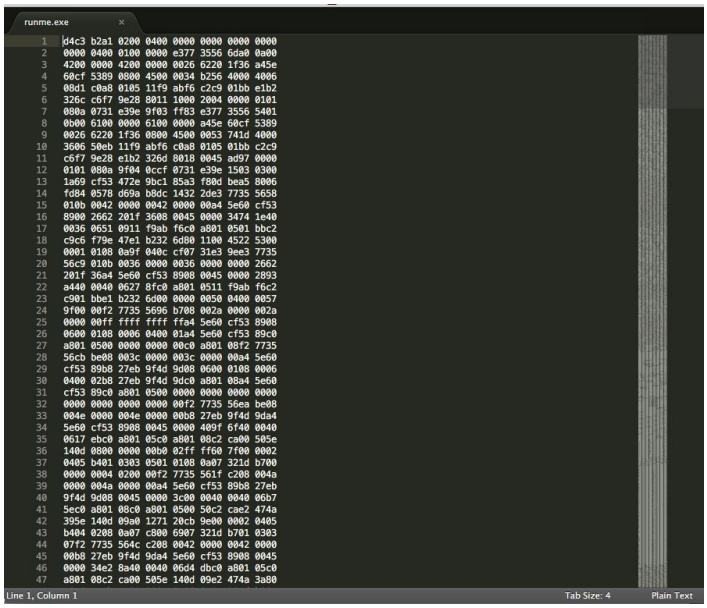
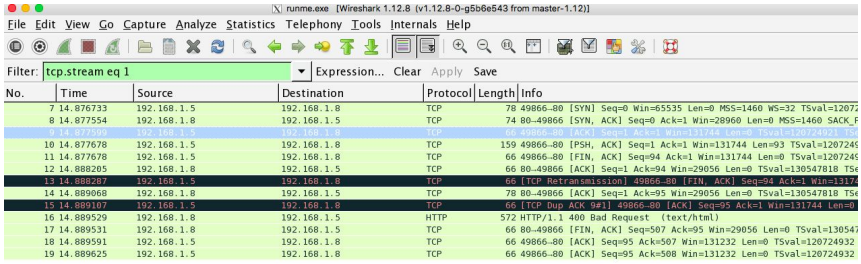
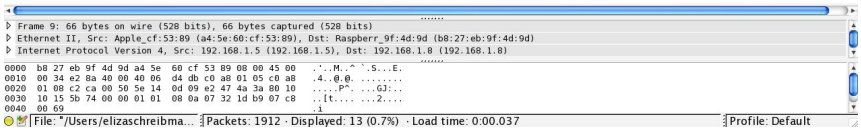
Flags

We found a flag in the runme.exe. After downloading the file and opening it in a text editor, we realized that it looked like a pcap file.

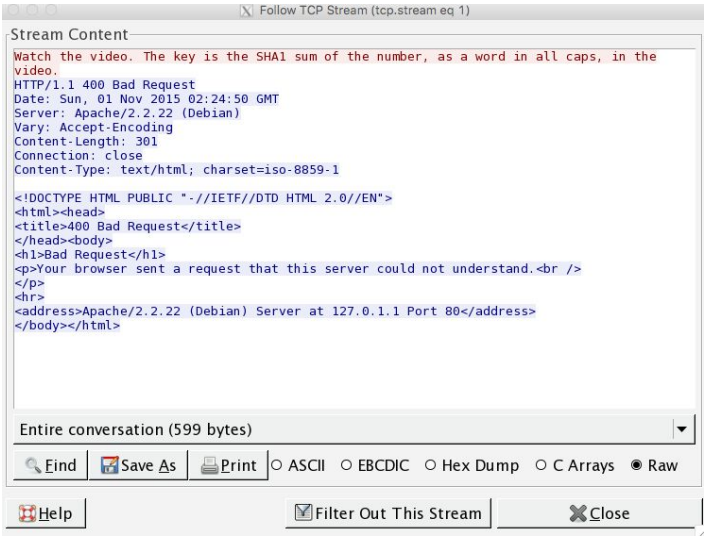


We decided to run wireshark on the file, and follow the tcp stream.





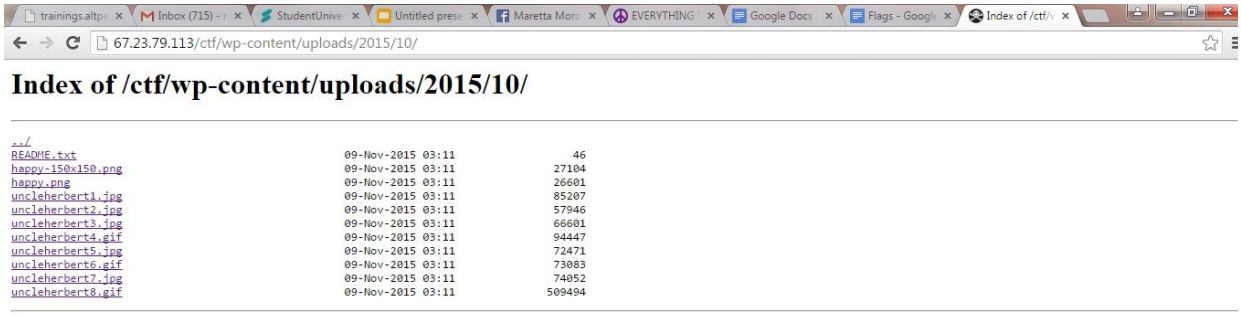
This lead us to the following message:



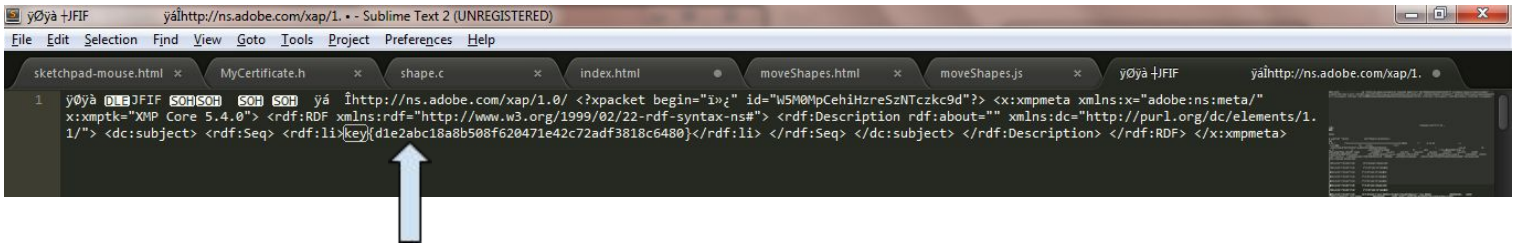
“Watch the video. The key is the SHA1 sum of the number, as a word in all caps, in the video.” To get this video, we had to export the object found in this stream, which produced the video sesamestreet.mp4.

We see a bunch of numbers at the very beginning of the video, so we added them and hashed the number, written out in all caps, to receive the correct flag.

Two flags were found by exploiting the knowledge that the site was a WordPress site. All WordPress sites store uploads in the same place. Thus, even without a link to the uploads we had the ability to go to the correct page. As can be seen in the screenshot below, all of the uploads were accessible once we arrived at this page. A total of two flags were found here. The first in the uncleherbert2.jpg image and the second in README.txt



Flag found in uncleherbert2.jpg

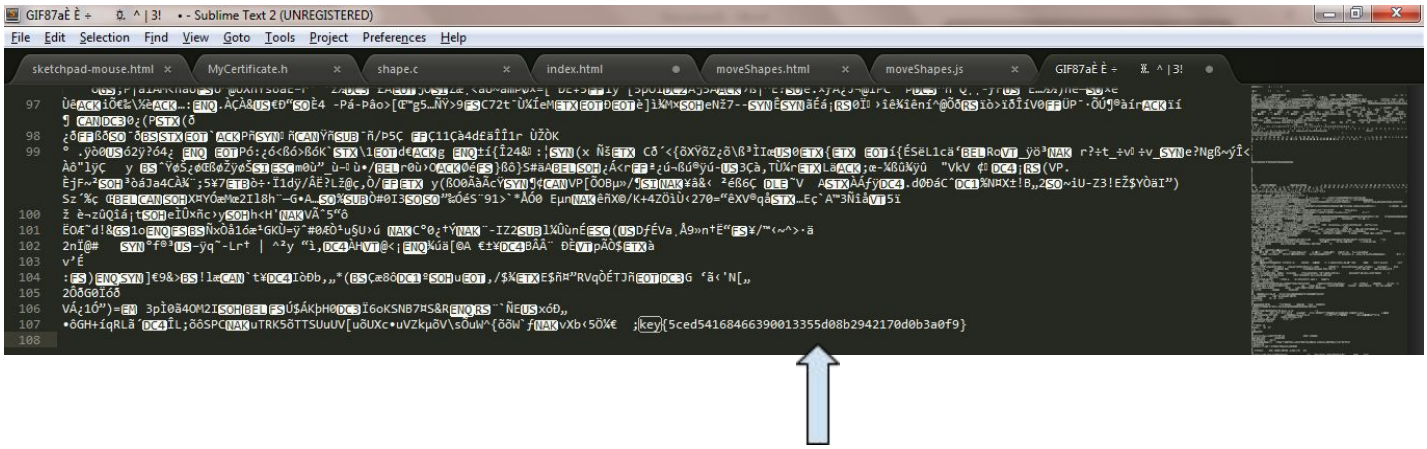


As can be seen in the image data above, the flag was found embedded within the image header for uncleherbert2.jpg. Once we accessed the image data it was easy to pull out the flag from the other data.

Flag found in README.txt

This flag was even simpler to access than in the case of the flag in uncleherbert2.jpg. Once the README.txt was accessed the flag was the only item in the file.

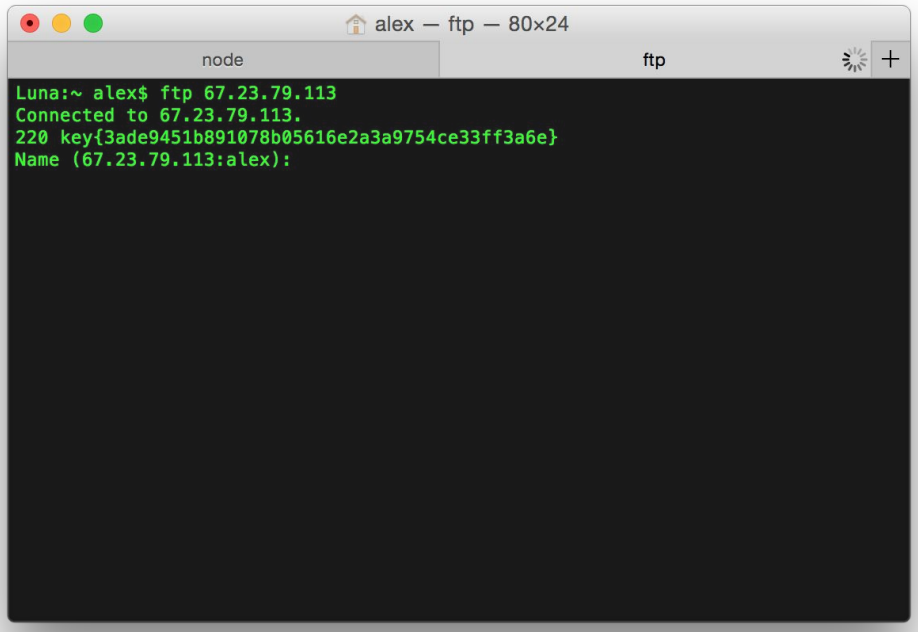
Flag found in crying.gif



A key was found at the end of the data for the crying.gif file found on the scoreboard. By viewing the gif data and searching for the string “key” it was an easy process to find the hidden flag.

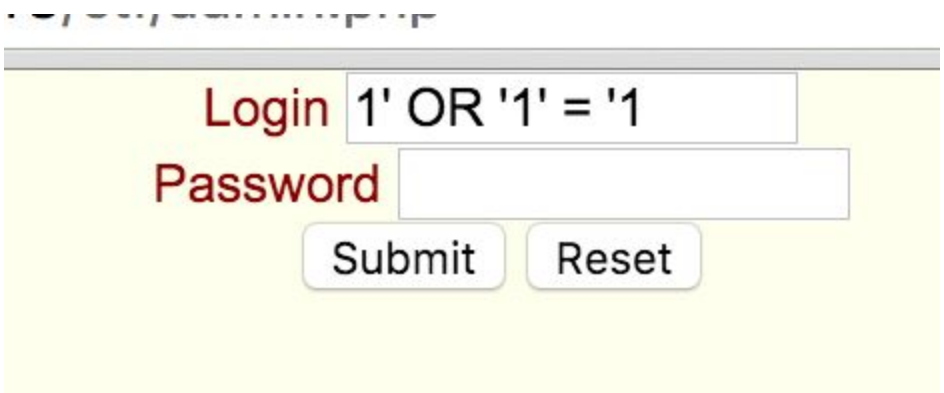
Flag found via FTP

A simple nmap scan found an unsecured FTP process running on Port 21 of the webserver. Attempting to connect resulted in a key being given as the login prompt.

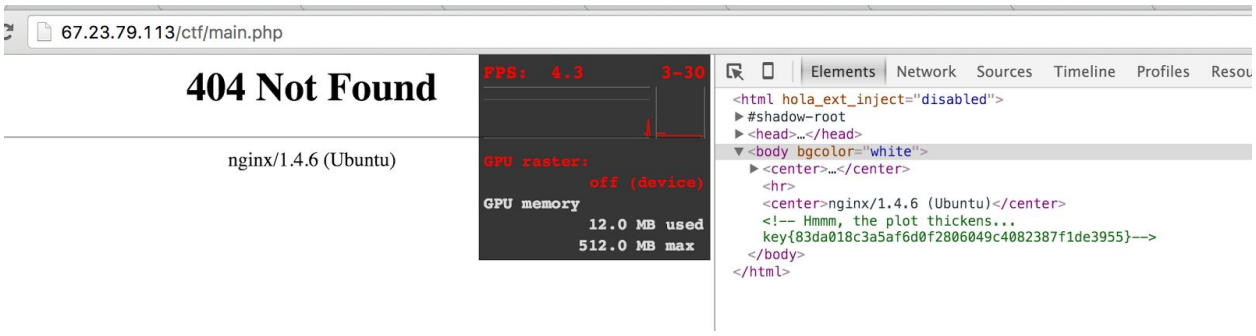


Flag found by SQL Injection

We found a link towards the bottom of the board page that lead to a login page.



Naturally, the first thing that came to mind was to try was SQL injection. Since there is a login, there has to be some form of data storage implemented, most likely using SQL. If the creator did not properly clean out the input from the fields and just used raw SQL statements built using string concatenation, then this SQL injection should work. Luckily for us, that's precisely what the person did. Using 1' OR '1'='1 in the username field allowed us to bypass a password check completely. However, this went to a 404 page. But since this is capture the flag, we decided to inspect the page source anyway just in case it was a fake 404 page.



After inspecting the source, we can see that the 404 page was not an actual 404 page. Inside the center element was a comment containing a key.

Through the database intrusion we discovered user login user names, one being bobo.

```
[+] Interesting header: SERVER: nginx/1.4.6 (Ubuntu)
[+] Interesting header: X-POWERED-BY: PHP/5.5.9-1ubuntu4.14
[!] Registration is enabled: http://67.23.79.113/ctf/wp-login.php?action=register
[+] XML-RPC Interface available under: http://67.23.79.113/ctf/xmlrpc.php
[!] Upload directory has directory listing enabled: http://67.23.79.113/ctf/wp-content/uploads/

[+] WordPress version 4.1-alpha-20141016 identified from meta generator

[+] WordPress theme in use: twentytwelve - v1.5

[+] Name: twentytwelve - v1.5
 |  Location: http://67.23.79.113/ctf/wp-content/themes/twentytwelve/
 |  Style URL: http://67.23.79.113/ctf/wp-content/themes/twentytwelve/style.css
 |  Theme Name: Twenty Twelve
 |  Theme URI: http://wordpress.org/themes/twentytwelve
 |  Description: The 2012 theme for WordPress is a fully responsive theme that looks great on any device. Features...
 |  Author: the WordPress team
 |  Author URI: http://wordpress.org/

[+] Enumerating plugins from passive detection ...
[+] No plugins found
[+] Starting the password brute forcer
  Brute Forcing 'bobo' Time: 00:03:29 <                    > (10326 / 14344393)  0.07%  ETA: 80:55:13
  [+] [SUCCESS] Login : bobo Password : supermodel


  +----+-------+------+------------+
  | Id | Login | Name | Password   |
  +----+-------+------+------------+
  |    | bobo  |      | supermodel |
  +----+-------+------+------------+

[+] Finished: Sat Nov  7 20:30:32 2015
[+] Requests Done: 10391
[+] Memory used: 8.273 MB
[+] Elapsed time: 00:03:32
root@kali:~#
```

Using the wordlist brute force method, we found the password: supermodel. This combination granted us entry to the wordpress site where another key was discovered.