Network Security Project2 Report

0416324 An-Fong Hwu 胡安鳳

(Note: this report is written in MD-Like format)

# (懶人包: Lazy people’s pack)Demonstration of the whole cracking process:

[https://youtu.be/UhorgrUoGYo](https://www.youtube.com/edit?o=U&video_id=UhorgrUoGYo)

The key used for encrypting the sensitive photo is:

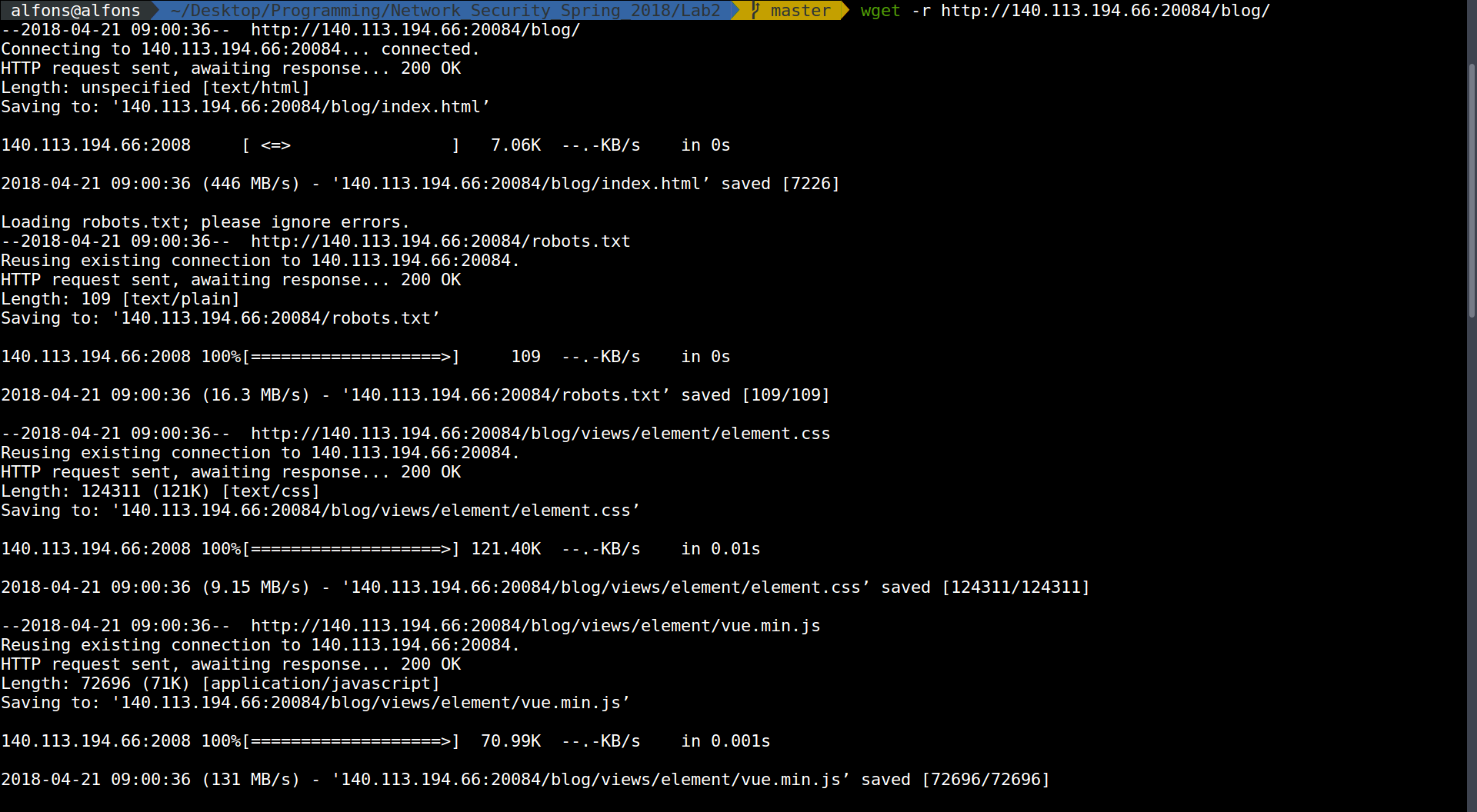
and the final photo is



# How do I implement the hacking of this project

\* Firstly we have only the website and nothing

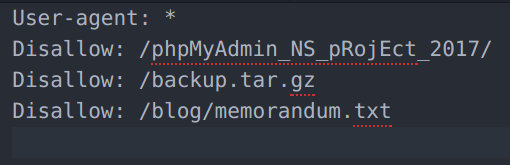
\* Then we can use the wget -r <http://140.113.194.66:20084/blog/> to crawl the content in the website.(r stands for recursively download the folders and subfolders)



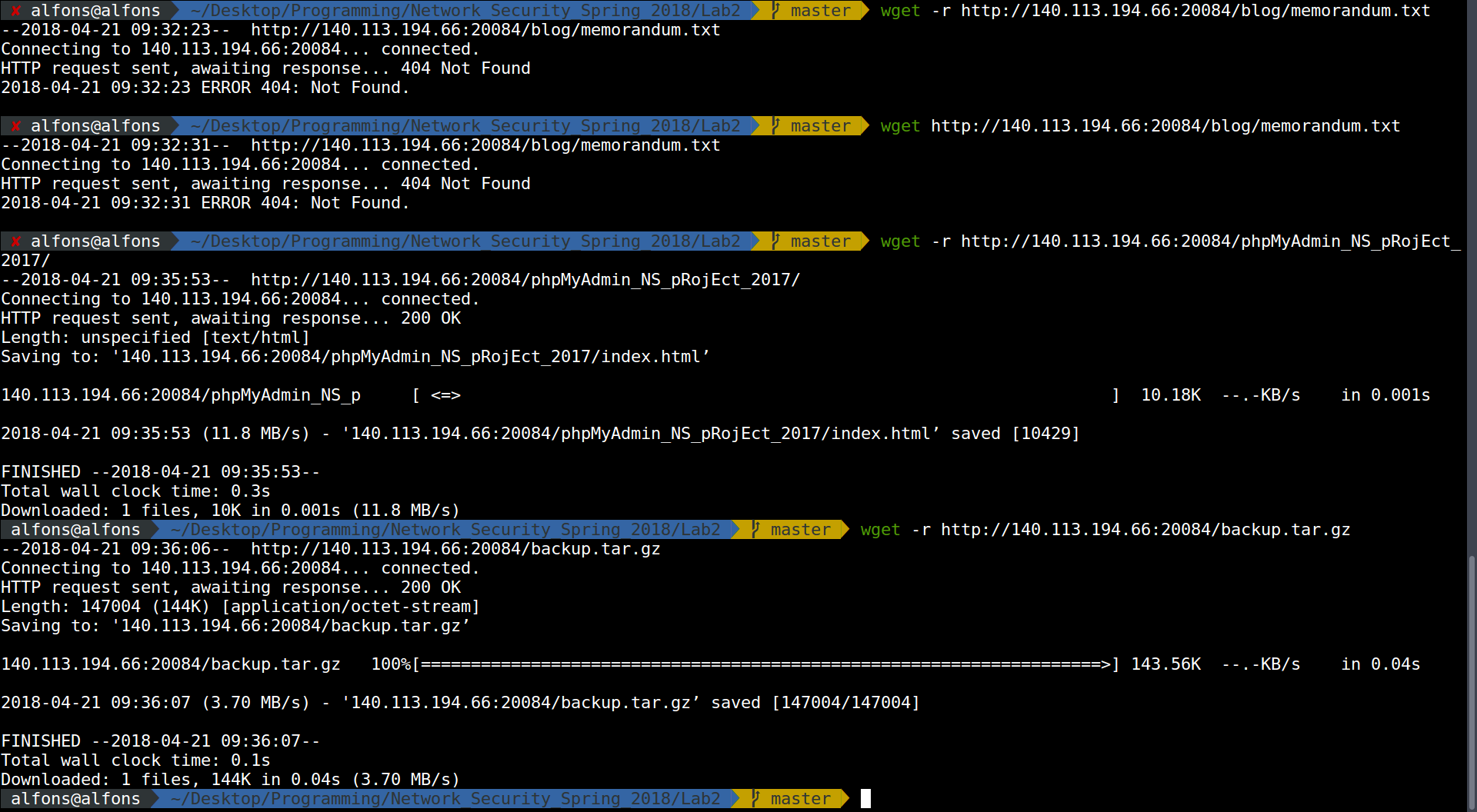
\* The robots.txt tells us what content should not be scanned or displayed in the search engine, thus I think it will show some clue about the web vulnerabilities.

Reference to: [https://blog.keniver.com/2017/03/robots-txt-%E7%9A%84%E4%BD%BF%E7%94%A8%E6%96%B9%E6%B3%95%E8%88%87%E5%AE%89%E5%85%A8%E6%B3%A8%E6%84%8F%E4%BA%8B%E9%A0%85/](https://blog.keniver.com/2017/03/robots-txt-的使用方法與安全注意事項/)

Well goes the saying, “Never try to prove what nobody doubts”,and the robots.txt really does such thing.



\* I tried to download the memorandum.txt but in vain.

TA also tells us that the swap ot temp file may leak some important information.

Then I wget the .memorandum.txt.swp and tried some other temporary files or backup files again and successfully found an encrypted data in .memorandum.txt.swp

\* The encrypted sensitive data is the following (encoded in base64 format like the last project)



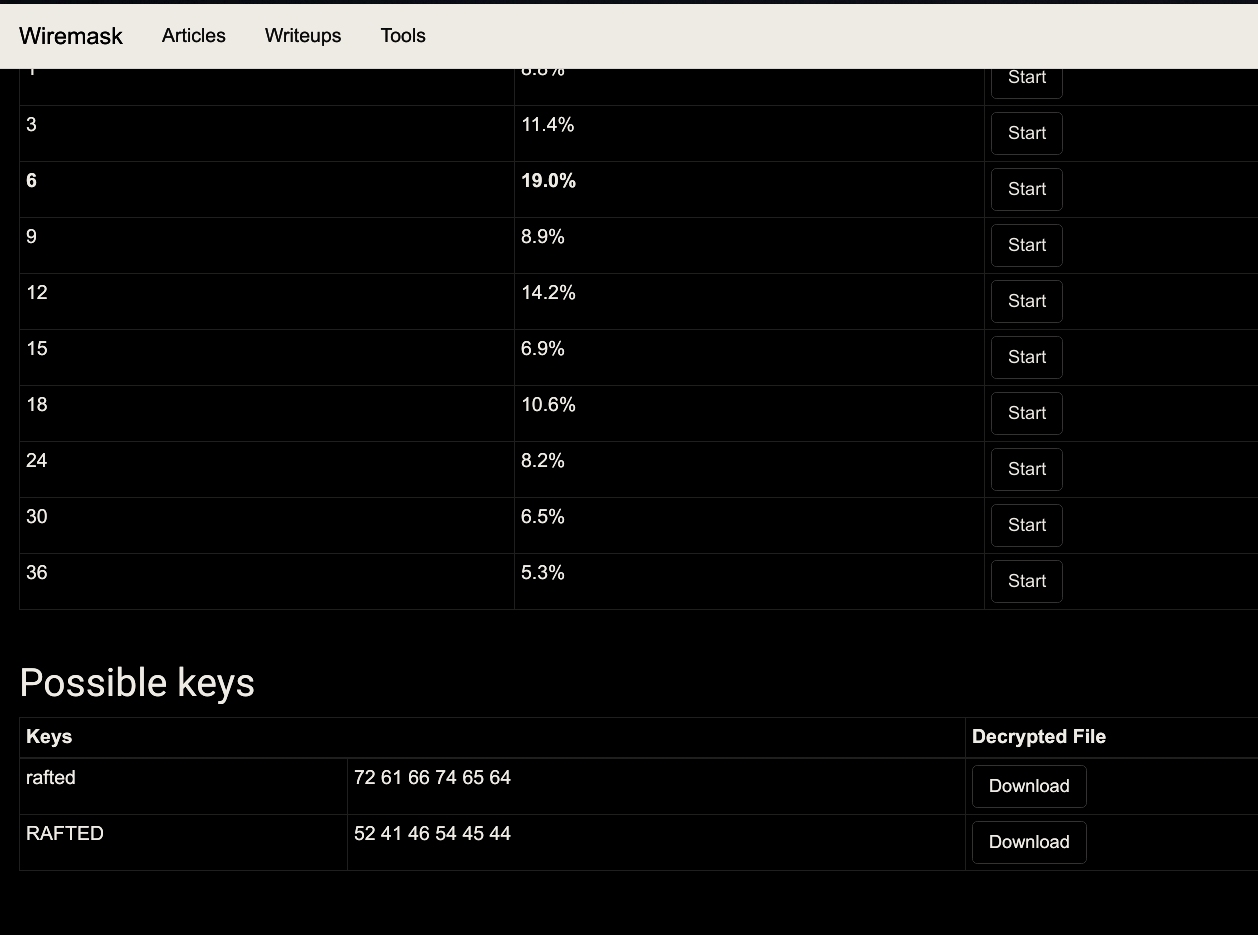
\* I write a python code to process it (embedded the linux script in it)



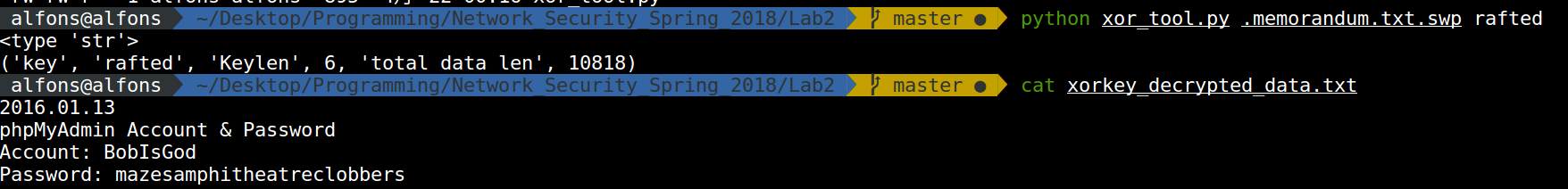
\* The files is using XOR encryption, so I put the decoded\_demo.bin to the website for cracking.

This website: <https://wiremask.eu/tools/xor-cracker/> is used for XOR cracking.

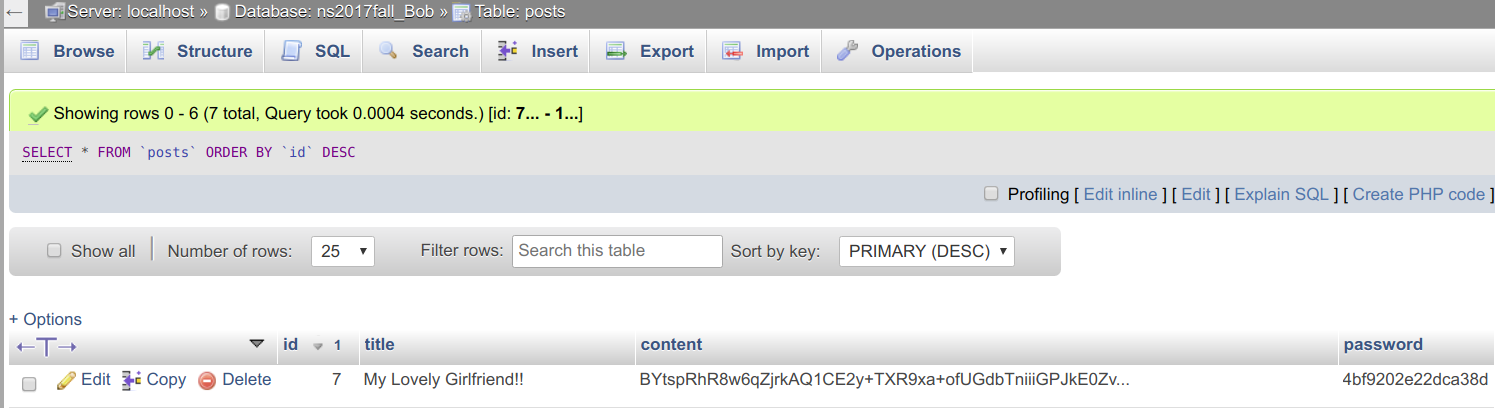
Then there are 2 possible keys



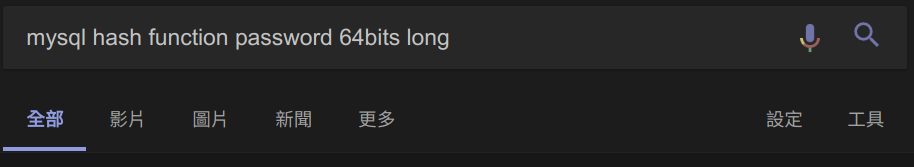
I tried rafted as key in my python script and successfully decrypted the swap file of memorandum.txt



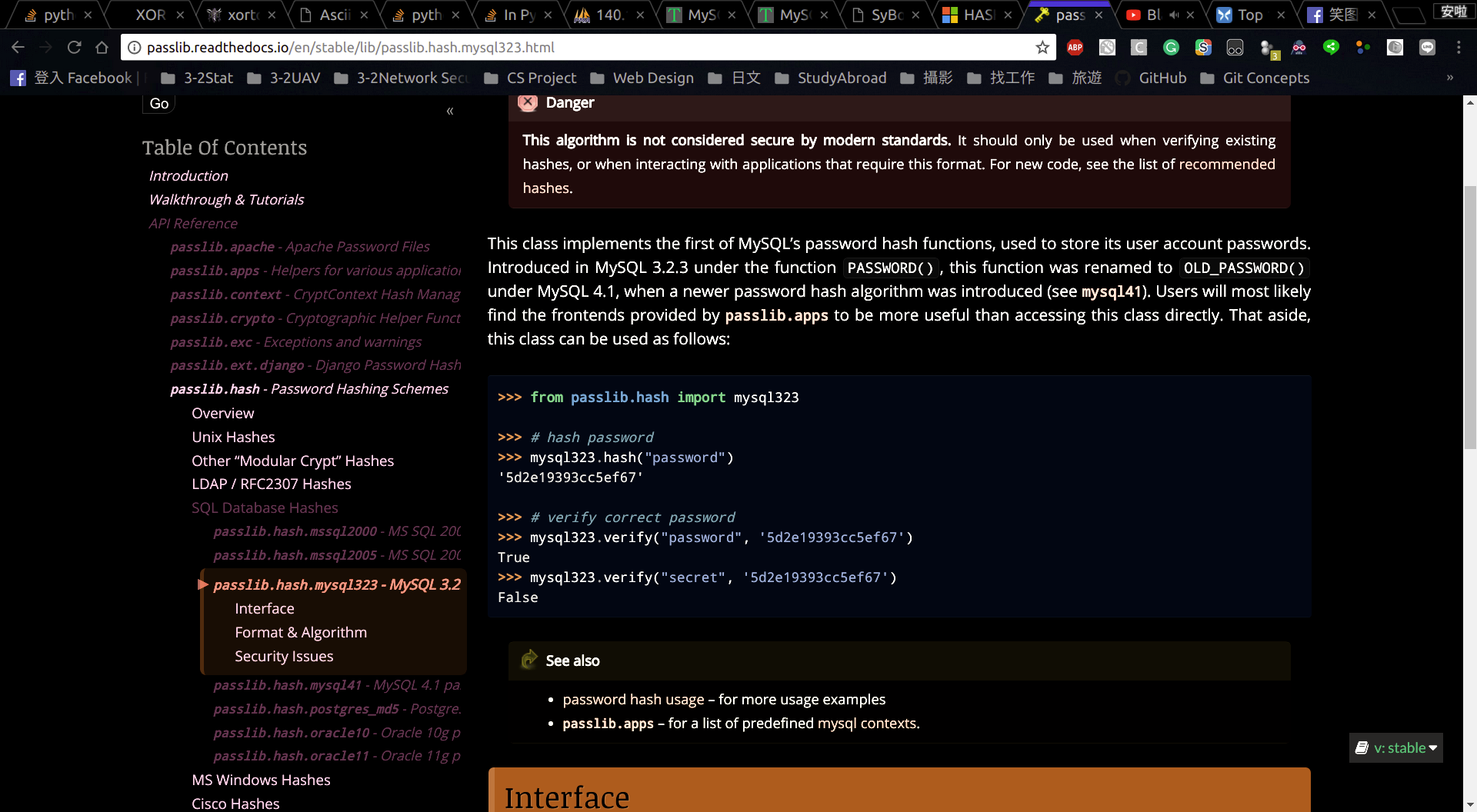
\* After the account name and password of the database is revealed, use them to login to the phpmyadmin in out project and we can see the password after hashing.



\* Googled “What SQL hash will result in an length of 64bits(8 bytes)”

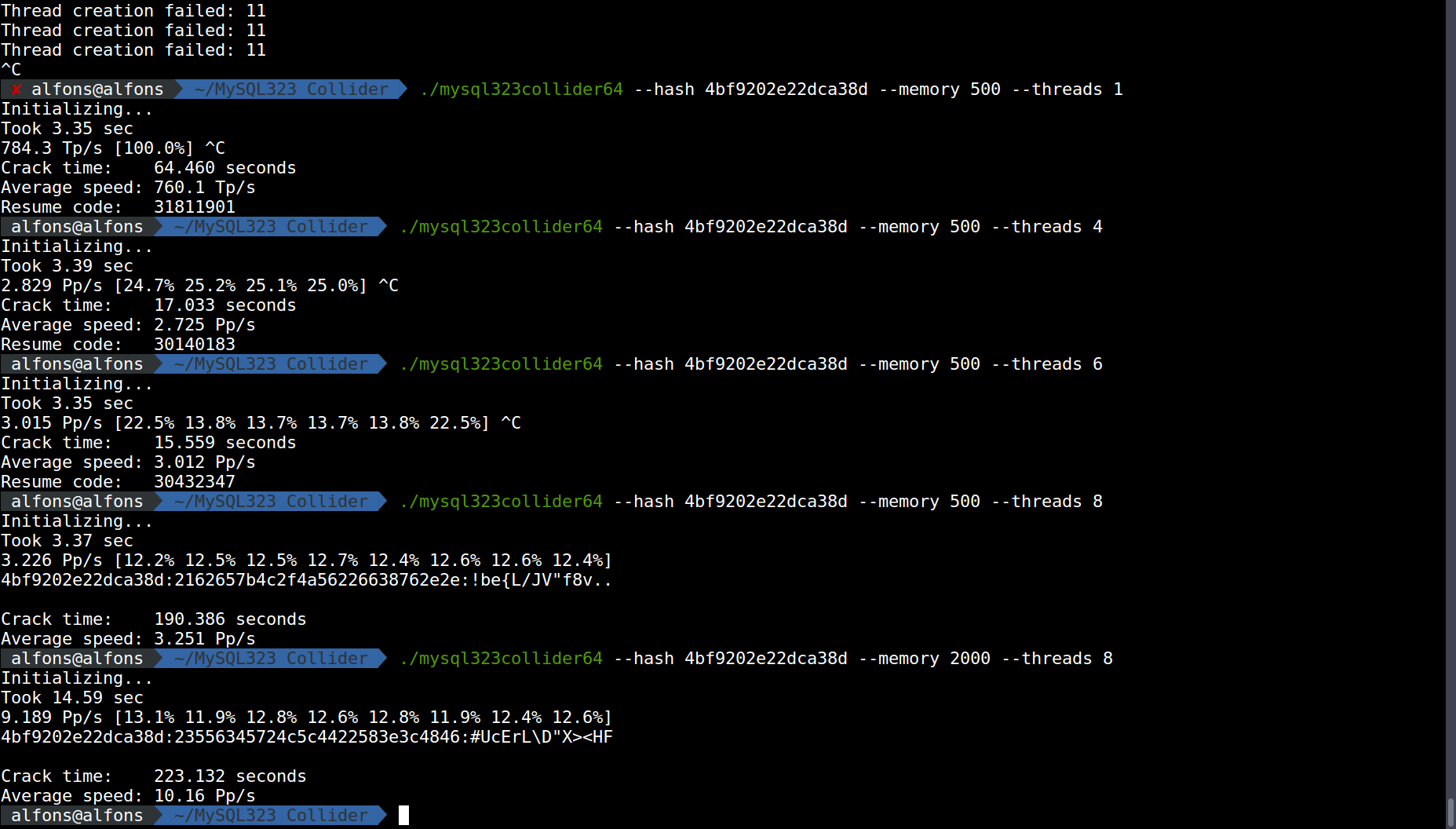


And I found this information

It is 64bits long, hashed with mysql323 hashing.

\* Then search for sql323 hashing cracker and found this website:

<https://www.tobtu.com/mysql323.php> which provides us with the collidor of mysql323 hashing

Usage(in linux)

```shell

chmod + x mysql323collider64

./mysql323collider64 –hash <hashed value we want to collide> --memory <memory limit for rainbow table> --thread <threads for parallel processing>

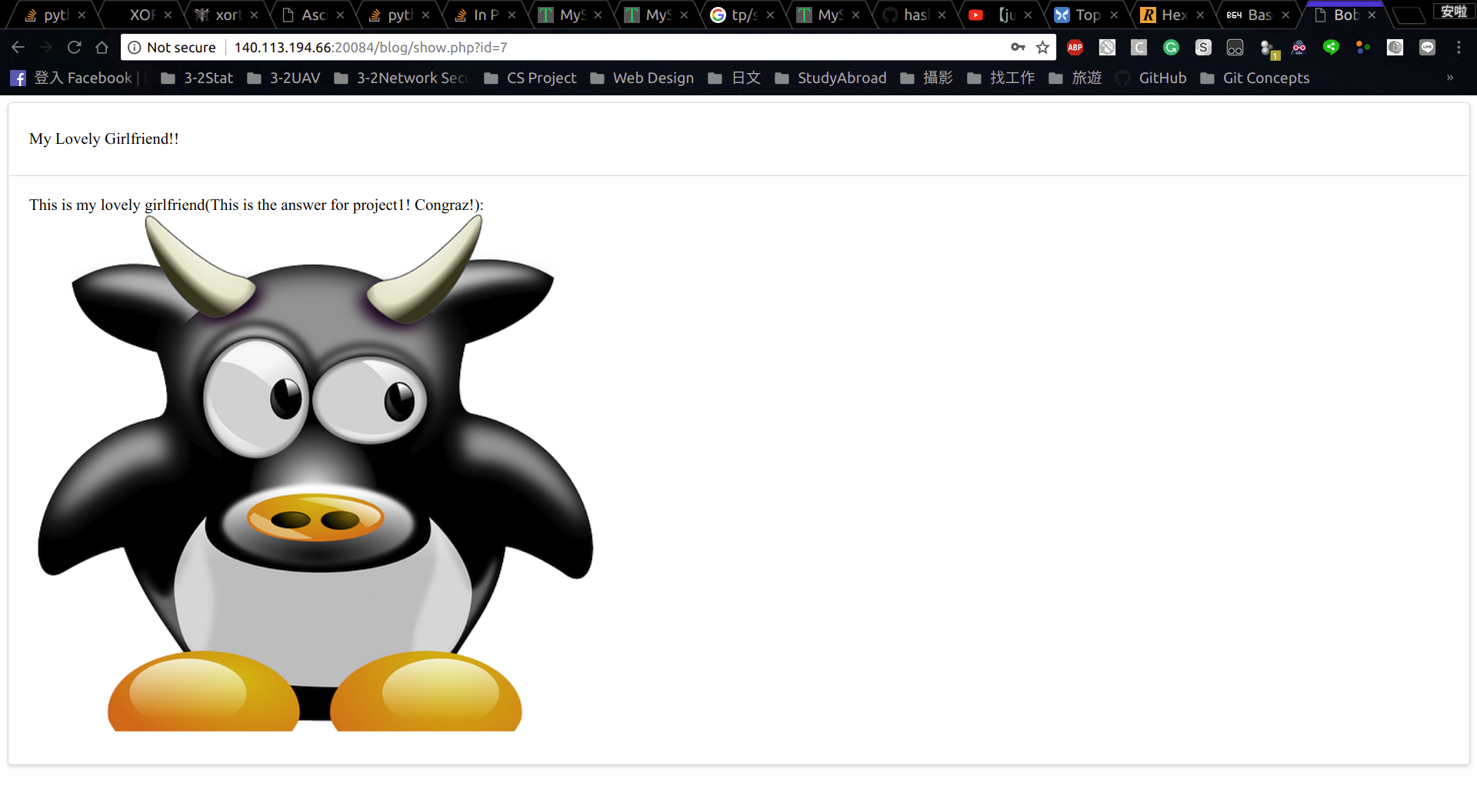
```

The result will be

[hashed value we want to collide]:[cracked key in hex]:[cracked key converted from hex to ascii]

and there are several keys with the same hashed values which provide us an opportunities to log in for picture due to hash collision.

\* I then finally hack into the website.



# What I have learned in this project

\* Some useful scripts in Linux such as cat base64file.txt | base64 --decode > output.txt then it easily decodes the encoded data and output and the wget command to crawl down the website.

\* XOR manipulating with self-written python script for arbitrary key length and corresponding encrypted data.

\* The ability to google tons of problems.

\* Common web vulnerabilities.

\* What does robots.txt do in website.

# How to prevent or patch these vulnerabilities?

\* Never leave the swap file, tmp file and backup files in the website, store in other more secured way, otherwise it may leak some sensitive information.

\* Fix the robots.txt with the following way [https://blog.keniver.com/2017/03/robots-txt-%E7%9A%84%E4%BD%BF%E7%94%A8%E6%96%B9%E6%B3%95%E8%88%87%E5%AE%89%E5%85%A8%E6%B3%A8%E6%84%8F%E4%BA%8B%E9%A0%85/](https://blog.keniver.com/2017/03/robots-txt-的使用方法與安全注意事項/)

robots.txt 應該視為引導爬蟲索引網站的工具, 而不是保護網站的工具, 任何不希望出現在網路上的資料不應該上傳到網路上, 假若逼不得已必須放到網路上, 必須使用 Http Authorization 之類的存取保護機制保護資料.

\* Hash the sensitive data in database with stronger hashing algorithm such as SHA-512, therefore even the database is compromised, the intruder is still not able to crack the confidential data.