

## Project 2. Web Vulnerability

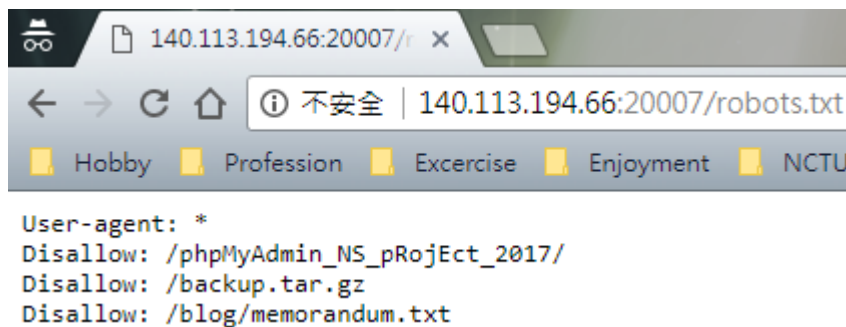
Name: 吳冠霆 Student ID: 0556503

### ● Hacking Steps

0. Bob's blog at <http://140.113.194.66:20007/blog/>

1. robots.txt

From <http://140.113.194.66:20007/robots.txt>, we can get location of Bob's phpMyAdmin, php code of Bob's blog and memo.



2. temporary files

Since <http://140.113.194.66:20007/blog/memorandum.txt> is not available, we try the format of **vim editor temporary file** <http://140.113.194.66:20007/blog/.memorandum.txt.swp> instead and finally get information leakage downloaded.

3. base64 encoding

```
cat memorandum.txt.swp | base64 -d >
memorandum_base64_decode_result
```

Eliminate the first line of memorandum.txt.swp, use Linux command base64 to decode it and pipe output to memorandum\_base64\_decode\_result.

4. XOR encoding & frequency analysis

Online tool: XOR Cracker

URL: <https://wiremask.eu/tools/xor-cracker/>

we use this to decrypt memorandum\_base64\_decode\_result.  
This online tool uses frequency analysis to guess the key used  
for encrypting the data.

The most probable key lengths

Key Length	Probability	Guess Keys
2	9.3%	<input type="button" value="Start"/>
4	14.3%	<input type="button" value="Start"/>
6	8.0%	<input type="button" value="Start"/>
8	18.0%	<input type="button" value="Start"/>
10	6.8%	<input type="button" value="Start"/>
12	9.7%	<input type="button" value="Start"/>
16	12.2%	<input type="button" value="Start"/>
20	6.9%	<input type="button" value="Start"/>
24	8.5%	<input type="button" value="Start"/>
32	6.3%	<input type="button" value="Start"/>

Possible keys

Keys	Decrypted File
fairways	66 61 69 72 77 61 79 73 <input type="button" value="Download"/>
FAIRWAYS	46 41 49 52 57 41 59 53 <input type="button" value="Download"/>

From downloading the decrypted file(XOR result of ciphertext),  
we get the account(**BobIsGod**) and  
password(**delineatedAelfricpresupposes**) to Bob's phpMyAdmin.

## 5. hash collision

Tool: MySQL323 Collider

URL: <https://tobtu.com/mysql323.php>

Then, we can get the encrypted content and hashed password  
of the post "My Lovely Girlfriend!!".

The screenshot shows the phpMyAdmin interface with a table named 'posts' in the 'ns2017fall\_Bob' database. The table has columns: id, title, content, and password. The last row is highlighted, showing the title 'My Lovely Girlfriend!!', the content 'YHV3r0Usu7P/14xbWR9O4hh+cNWutr9LczEYp194fS2Zl', and the password '335e9b3a48bc395a'.

id	title	content	password
1	Sugar - Maroon 5	I'm hurting, baby, I'm broken down I need your lo...	NULL
2	You're beautiful	My life is brilliant. My life is brilliant. My...	NULL
3	おだ のぶなが	Oda Nobunaga (織田 信長 About this sound Oda Nobunaga ...	NULL
4	山本五十六	山本 五十六 (やまもと いそろく、1884年 (明治17年) 4月4日 - 1943年 (昭和18年) 4月...	NULL
5	This is not what you're looking for...	Not this post... Please try another post...	NULL
6	Fake stay night saying	People die if they are killed!!	NULL
7	My Lovely Girlfriend!!	YHV3r0Usu7P/14xbWR9O4hh+cNWutr9LczEYp194fS2Zl	335e9b3a48bc395a

We find out **my\_own\_hash** in functions.php implements mysql323 hash algorithm, so we end up using MySQL323 Collider.

"fHfAg:f4.^8." is the password we want.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

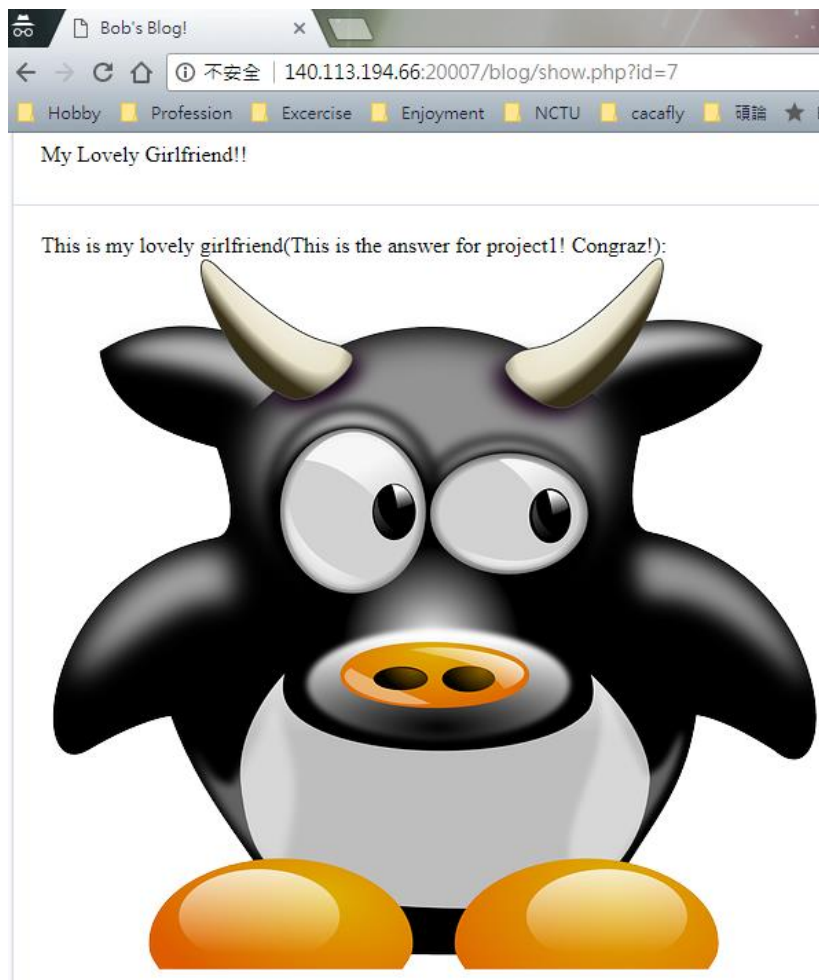
C:\Users\DannyWu>cd Desktop\Project2

C:\Users\DannyWu\Desktop\Project2>cd "MySQL323 Collider"

C:\Users\DannyWu\Desktop\Project2\MySQL323 Collider>"mysql323 collider 32.exe" -
h 335e8b3a48bc395a -m 1024 -t 2
Initializing...
Took 16.38 sec
2.010 Pp/s [50.0% 50.0%]
335e8b3a48bc395a:2266486641673a66342e5e382e "fHfAg:f4.^8."

Crack time: 76.969 seconds
Average speed: 2.039 Pp/s
```

The result of "My Lovely Girlfriend!!"



## ● What I Have Learned

1. Meaning of robots.txt
2. How to access temporary files
3. How frequency analysis works and how to use it for decrypting data
4. Mysql323 hash algorithm and how to break it

## ● **How to Prevent these Vulnerabilities**

1. Don't put important information in robots.txt since it can only prevent web crawlers from accessing these important files. Edit .htaccess to make sure important files can't be accessed.
2. Make sure to clean all of your temporary files.
3. Keep the hash and encryption algorithm you used in secure, or it would be easy to break it since there are plenty of tools can be used for cracking it.