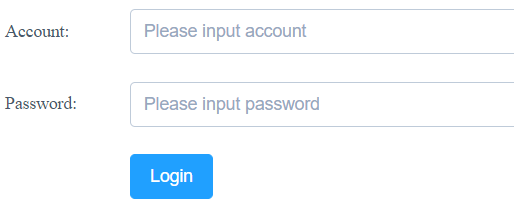
**0416313何元斌 網路安全project1報告**

過程

1. SQL injection:



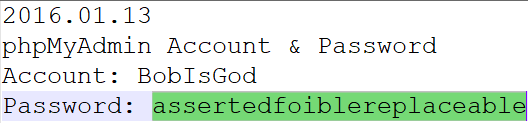
首先是第一關，利用語言的漏洞(沒有過濾特殊字元例如:{ ‘ }、{ = }、{ ; }等等)，我們可以輸入:

*account = { 1’or’1’ = ‘1’;= }*

使原本的SQL變成一個恆等式，並且順帶註解掉下面判斷PASSWORD的CODE，如此便可順利進入。

1. BASE 64 & XOR:

進入後，在第7則post中，有一連串的密文，先利用base 64將其轉成16進位制，再利用xor，將之解密，可以得到一組帳號密碼:



1. Robots.txt:

再將原本的網址:

[*http://140.113.194.81:20150/blog*](http://140.113.194.81:20150/blog)

改成

[*http://140.113.194.81:20150/robots.txt*](http://140.113.194.81:20150/robots.txt)

可以得到該網頁的phpMyAdmin的登入網址，接著利用步驟2中得到的帳號密碼登入，可以看到

**My Lovely Girlfriend!!** 中被加密過後的密碼。

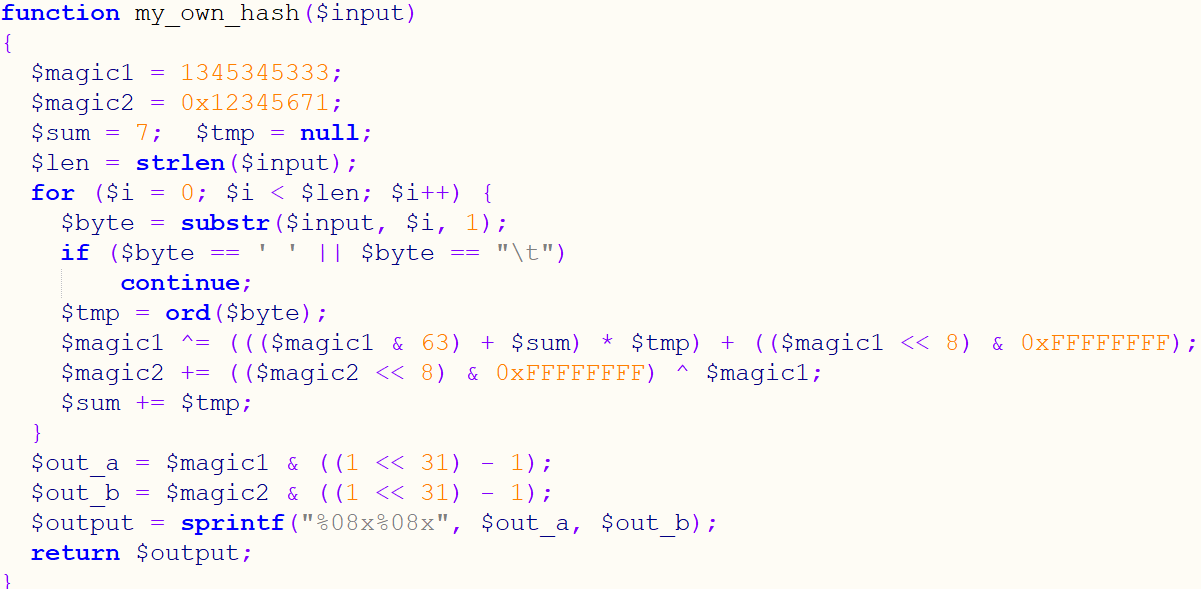


1. GIT & Temporary Files:

用GIT將網頁的修改紀錄(temporary files)下載後，再下指令

**git checksum**

還原到以前的版本，可以得到一個hash function。

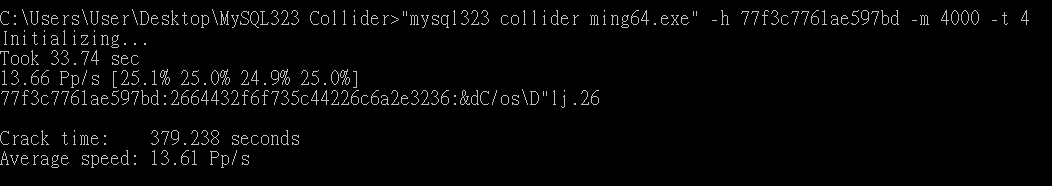


1. One-way hash function & hash collision:

現在有了hash function和hash過後的密文，我們仍然很難得到正確的原始密碼，但，我們無須正確的原始密碼，只需要得到一組經過hash function後，可以和正確密碼得到一樣結果的密碼即可(即為hash collision)，而且，上述的hash function是已經使用很久的加密法(稱為mysql323)；

因此，利用網路上的工具，可以得到一組

**My Lovely Girlfriend!!** 的密碼。



Result



使用工具、參考資料

SQL injection: <https://en.wikipedia.org/wiki/SQL_injection>

BASE 64:

<https://cryptii.com/pipes/base64-to-hex>

XOR:

<https://github.com/hellman/xortool>

GIT & Temporary Files:

<https://git-scm.com/>

hash collision:

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

How to prevent

SQL injection

現在很多資料庫語法都會過濾一些特殊字元，或者是不允許輸入，使用這種較新的系統即可預防。

GIT & Temporary Files

若必須使用GIT來備份系統，記得要設定權限，設定只有管理員能下載特殊的檔案。

hash collision

使用較新的HASH FUNCTION，來避免在網路上可以找到有效的破解法，或是多個plaintext 可以hash到同一個ciphertext。

其他

不要把加密演算法、密碼放到網路上，重要的資料保存在本機等等，皆可有效預防。

What have I learn

GIT

一個廣泛用來修復資料的工具，可以用來輕鬆的還原資料，還可以查看修改紀錄等等，在備份多人共享的文件時相當好用。

robots.txt

可以查看原本不會被放進搜尋列表的頁面。

SQL injection

複習了一下sql語法，也了解為何大部分網站之帳號密碼，為何不允許使用者設定特殊字元為帳號密碼。

hash collision

原本我以為，要破解一個密碼就必須要知道原始的密碼，但其實不用，只需要一組可以讓認證通過的字串即可。

不要被文字誤導

像是在Crack me if you can中，解釋說bob用了**his own encryption method**，因此我認為他是用了substitution method，所以我想利用frequency analysis來解密，結果繞了一大圈。