CS558 Lab 4 Part 1

Konstantino Sparakis

March 16, 2017

NOTE: Should be all correct, I used latex so some solution answers characters might be messed up since they can contain odd characters. So I also included all my answers in a txt, solutions.txt.

Collaborated: No one

Late Days: 0

Total Late Days: 2

Sources:
http://www.unixwiz.net/techtips/sql-injection.html
http://www.comsecglobal.com/FrameWork/Upload/SQL_Smuggling.pdf
http://blog.kotowicz.net/2013/01/abusing-mysql-string-arithmetic-for.html
https://docs.python.org/2/library/md5.html
https://stackoverflow.com/questions/1557571/how-to-get-time-of-a-python-program-execution
https://stackoverflow.com/questions/3437059/does-python-have-a-string-contains-substring-method

1.0 No protection:
Solution:
username=victim&password=hi%27%20OR%20%271%27%3D%271

I used the following input for password

hi’ OR ’1’=’1

The reason this works is because the php code that checks this most likely looks like this

”Select * WHERE username=’+usernam+’ AND password=’+password+’;”

by closing out mom with the ’ it messes the parsing to

”Select * WHERE username=’username’ AND password=’mom’ OR ’1’=’1’;”

so now if password = false becase the 1=1 it will always return true so it returns the user as long as we know the username.

1.1 single quote to two single quote:
Solution:
username=victim&password=hi%5C%27%20OR%201%3D1%20−−%20

Injection used:
”hi\’ OR 1=1 −− ”

the ”−− ” makes anything afterward a comment, also note there is a space at the end of that. We do \’ since it will turn into \’ ’ so sql will escape the first \’ but leave the 2nd quote free to still do this attack.

1.2 with hashing:
Solution:
username=victim&password=6365540
takes about 14 seconds to compute.

I quickly came to the conclusion that I would need to find an md5 hash that would output an sqlinjection in its binary raw format. But to compute this it could take years given that up to this point all my sql injections where of about 14 or characters and given my machine is a laptop.

First Thing I did was look for the shortest possible sql injection, and luckily I was able to find this blog post

http://blog.kotowicz.net/2013/01/abusing-mysql-string-arithmetic-for.html

it outlines how '-' will evaluate to 0 which in turn give our php code will execute to be the equivalent of 1=1, and I even went ahead and tested it on the sqlinjection0 site to see that it in fact works.

username=victim&password=%27-%27

so now from here all I had to do was create a quick and dirty python program that would brute force search for a string that when hashed with md5 it's hash contained '-' and that should work.

My script took about 14 seconds to execute and returns:
Sql injection = 6365540 with hash= '-'De ????J2C Time taken:13.7294299603

so the password 6365540 works.

My script works by simply incrementing a number converting that into a string and hashing it it, checking to see if the hash contains '-'. with enough increments we find a collision. The reason I increment was to avoid having to deal with dictionaries and generating random words.