












## ***Lifestyle Store***









**Detailed Developer Report**

*By Omkar Holkar*

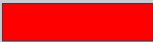


## Vulnerability Statistics

- High - 
- Medium- 
- Low - 

Sr No.	Vulnerability Name	Level
1	Rate Limiting Flaw	
2	Bruteforce Exploitation	
3	SQL Injection	
4	Directory Listing	
5	Insecure Direct Object Reference	
6	Cross-Site Scripting	
7	Weak Passwords	
8	Cross-site request Forgery	

Sr No.	Vulnerability Name	Level
9	Forced Browsing	
10	PII Leakage	
11	Default Files and Pages	
12	Client Side Filter Bypass	
13	Insecure File Upload	
14	Command Execution Vulnerability	
15	Server Misconfiguration	
16	Components with known vulnerability	

## Total no. Of Vulnerabilities - 16

Severity	No. of Vulnerabilites	Color codes
High	7	
Medium	6	
Low	3	

## Security Status – Extremely Vulnerable

- Hacker can steal all records of Web-App databases (SQLi)
- Hacker can take control of complete server including View, Add, Edit, Delete files and folders (File Upload)
- Hacker can inject client side code into applications and trick users by changing how page looks to steal information or spoil the name of Organization. (XSS)
- Hacker can extract details of all customers. (IDOR)

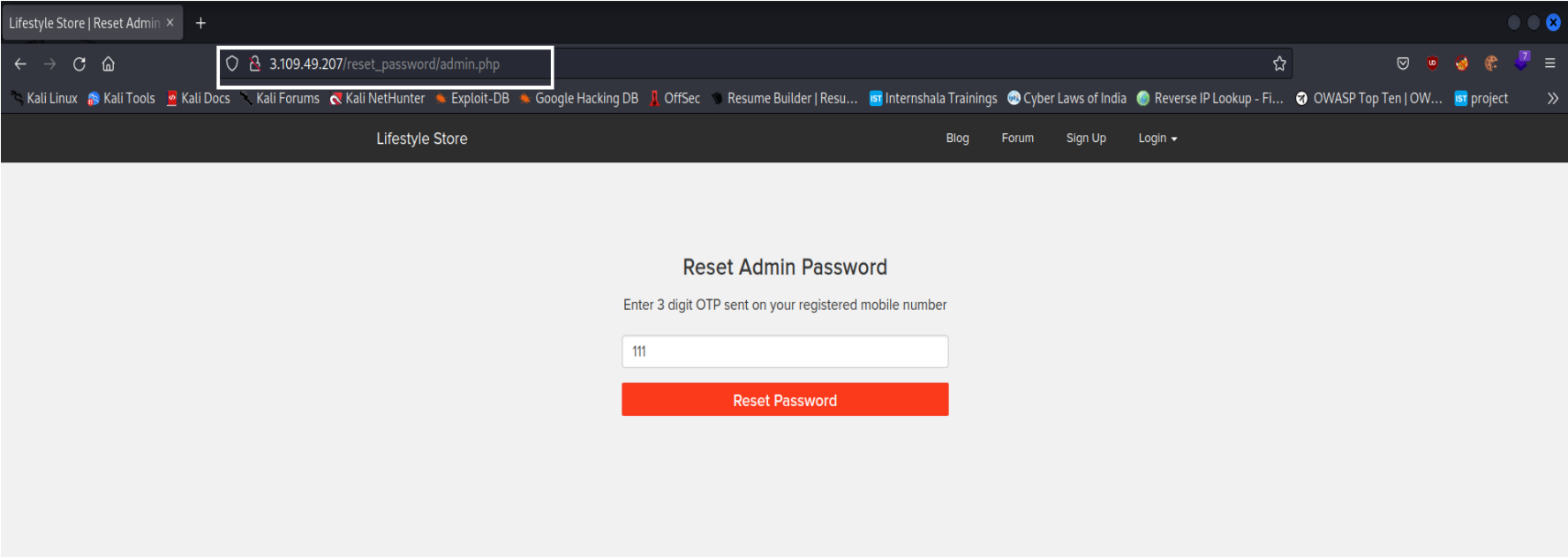
## 1. Rate Limiting Flaw -

- The following URL of *Lifestyle Store* was found vulnerable to Rate Limiting Flaw.

	Rate Limiting Flaw
	<p data-bbox="538 653 868 695">Affected URL :-</p> <p data-bbox="538 762 1730 810"><a href="http://3.109.49.207/reset_password/admin.php?otp=111">http://3.109.49.207/reset_password/admin.php?otp=111</a></p> <p data-bbox="538 870 874 918">Parameter - otp</p>

# Observation

1. visit the affected URL in browser and put any random 3 digit number and click the “Reset Password” button.



2. Repeat the step 1 multiple times and you see there is no limit set for the no. Of times user can enter the otp.

## Exploitation

Steps:-

1. Put any random 3 digit no. And hit the “Reset Button” and capture the request in Burpsuite tool.
2. Send the request to “Intruder” tab, add the value from “otp” field in payloads position.
3. Set the ‘attack type’ “sniper” , ‘payload type’ as “numbers”. In the ‘payload options’ section set the range from 100 to 999 with step count=1 and start the attack.
4. There is one payload with different length than other, put that payload in browser and click on “Reset Password”.
5. Here we see we were able to reset the admin password and can access the admin account.
6. Thus we can change the price of product,add,edit.



9

Burp Suite Professional v2022.8.2 - Temporary Project - licensed to Omkar

Dashboard Target Proxy **Intruder** Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

1 x 2 x **3 x** +

Positions Payloads Resource Pool Options

1 Choose an attack type

Attack type: Sniper

2

3

Start attack

3

Update Host header to match target

Add \$

Clear \$

Auto \$

Refresh

```
1 GET /reset_password/admin.php?otp=$111$ HTTP/1.1
2 Host: 3.109.49.207
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://3.109.49.207/reset_password/admin.php?otp=111
9 Cookie: key=6036056e1g; PHPSESSID=3dgnf0f7ror7im2h701cd85kn6; X-XSRF-TOKEN=06a0f477e412b1246c4aale01c6ae9aaa6f8335e37ce91f84ae2ec2d16c589a8
10 Upgrade-Insecure-Requests: 1
11
12
```

Burp Suite Professional v2022.8.2 - Temporary Project - licensed to Omkar

Dashboard Target Proxy **Intruder** Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

1 x 2 x **3 x** +

Positions **Payloads** Resource Pool Options

4

5

Start attack

7

Start attack

5

1

Numbers

1

900

900

6

6

Sequential

From: 100

To: 999

Step: 1

How many:

4. Intruder attack of http://3.109.49.207 - Temporary attack - Not saved to project file

Attack Save Columns						
Results Positions Payloads Resource Pool Options						
Filter: Showing all items						
Request ^	Payload	Status	Error	Timeout	Length	Comment
816	915	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
817	916	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
818	917	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
819	918	200	<input type="checkbox"/>	<input type="checkbox"/>	4476	
820	919	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
821	920	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
822	921	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
823	922	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	

Proof of Concept (PoC)

Lifestyle Store | Admin

3.109.49.207/admin31/dashboard.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Resume Builder Internshala Trainings Cyber Laws of India Reverse IP Lookup

Admin Dashboard

CONSOLE

Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
			<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD		Add

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men &amp;ump; Women Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	145	Update
2	Adidas Socks - Pack	Adidas Men &amp;ump; Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	450	Update
3	Puma Socks	Men &amp;ump; Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	600	Update
4	Reebok Men Socks	Men Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	1111	Update
5	Basic T shirt	alier(t)	<input type="radio"/> Chandan <input checked="" type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	350	Update
6	Simple T Shirts	Use these t shirts for light summers.	<input type="radio"/> Chandan <input checked="" type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	550	Update

## **Business Impact - High**

1. Hacker can add a new product , change the price of any product , edit the information of specific product.
2. Hacker can spoil the reputation of the company/organization by performing such malicious activities.
3. Altering the price can cause financial loss to the company.

### **Recommendation**

1. Put restriction on no. Of times the users can enter the otp.
2. Use POST method instead of GET while passing sensitive information.
3. Use alphanumeric characters for OTP.

### **References**

1. [https://cheatsheetseries.owasp.org/cheatsheets/Denial\\_of\\_Service\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html)
2. <https://apisecurity.io/encyclopedia/content/owasp/api4-lack-of-resources-and-rate-limiting.htm>

## 2. Bruteforce Exploitation -

- The following URL's of *Lifestyle Store* was found vulnerable to Bruteforce Exploitation-

Bruteforce Exploitation
<p data-bbox="478 632 829 680">Affected URL's:-</p> <p data-bbox="478 740 1308 788"><a href="http://3.109.49.207/login/customer.php">http://3.109.49.207/login/customer.php</a></p> <p data-bbox="478 794 1223 842"><a href="http://3.109.49.207/login/seller.php">http://3.109.49.207/login/seller.php</a></p> <p data-bbox="478 848 1244 896"><a href="http://3.109.49.207/login/admin.php">http://3.109.49.207/login/admin.php</a></p>

## **Obsevation**

1. Visit any of affected url's and fill in the username and password fields.
2. Repeat above step and we see there's no limit for no. Of times users could enter the details in given fields.

## **Exploitation**

Steps:-

1. Visit any of affected URL and fill the inputs with random data.
2. After hitting the 'Login' button capture the login request in Burpsuite tool.
3. Send the captured request to 'intruder' tab.

4. Now select only 'username' and 'password' fields and set the 'attack type' as "Cluster bomb".

5. Now in the payloads tab, set the 'payload type' as "simple list" for both 'payload set'.

6. Using appropriate list for usernames and password in 'Payload options' tab and shooting the "Start attack", we could crack the possible username and passwords of users.

7. look for response length different than others, use it to login to specific user account.



3. Intruder attack of http://3.109.49.207 - Temporary attack - Not saved to project file

Attack Save Columns							
Results Positions Payloads Resource Pool Options							
Filter: Showing all items							
Request	Payload1	Payload 2	Status	Error	Timeout	Length	Comment
132	donal234	Donal234123	200	<input type="checkbox"/>	<input type="checkbox"/>	569	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	542	
1	akash	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	542	

Proof of Concept (PoC)


3.109.49.207/profile/profile.php

3.109.49.207/profile/profile.php

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSecResume Builder | Resu...Internshala TrainingsCyber Laws of IndiaReverse IP Lookup - FI...OWASP

Lifestyle StoreMy CartMy ProfileMy OrdersBlogForumLogout

My Profile



Donald Duck

donald@lifestylestore.com

Username:

Donal234

Contact No.:

9489625136

Delivery Address:

B-34/ the duck lane, Disneyland

EDIT PROFILE

CHANGE PASSWORD



## **Business Impact - High**

1. Hacker can access account information of no. Of users by bruteforcing the credentials.
2. He can edit the user's profile and also make an order.
3. Performing such malicious activities can spoil the company's reputation and would result in business loss.

## **Recommendation**

1. Limit the no. Of times users can enter invalid credentials.
2. Make policy of keeping strong passwords for users.
3. Do not allow users to keep simple and guessable passwords.

## **References**

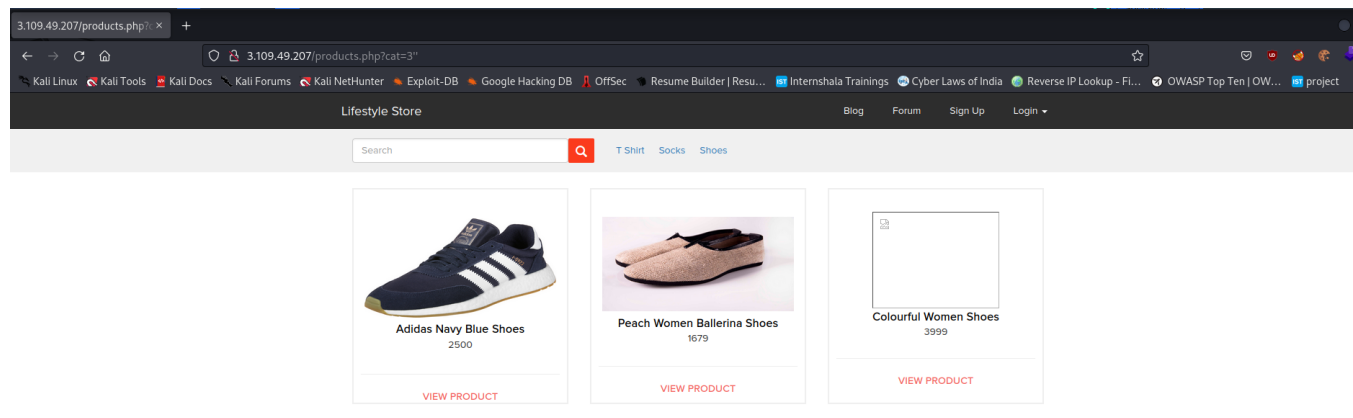
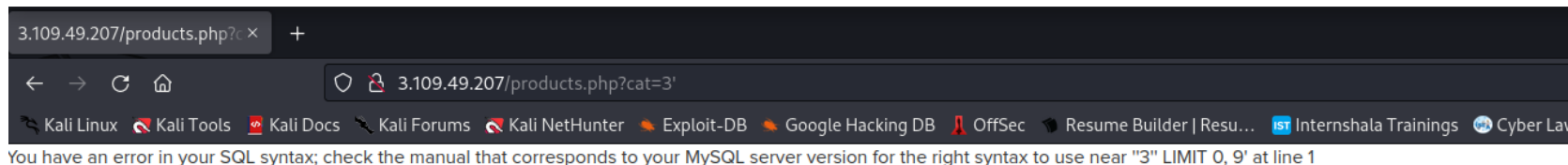
1. [https://owasp.org/www-community/controls/Blocking\\_Brute\\_Force\\_Attacks](https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks)
2. <https://www.itsasap.com/blog/how-to-prevent-brute-force-attacks>
3. <https://www.kaspersky.com/resource-center/definitions/brute-force-attack>

### 3. SQL Injection -

	<b>SQL Injection</b>
	Affected URL:-
	<a href="http://3.109.49.207/products.php?cat=1">http://3.109.49.207/products.php?cat=1</a>
	Parameter- cat

## Observation

1. Visit the affected URL, enter an single qote at the end of url we get an SQL error.
2. After we complete the syntax by putting an another qote the error is gone and web-app loads successfully.



## Exploitation

1. visit the affected URL and enter a single quote.
2. Now after using some SQL statements attacker can successfully gain complete control of the database.
3. All the users, their id's, passwords were extracted.
4. Using John the Ripper tool attacker gets successful in getting passwords from their hashed form.

Here's list of SQL commands used for exploitation-

- [https://drive.google.com/file/d/1Je7VqgCvSWRMy8\\_E5BQ9MPQhEr0\\_AA3s/view?usp=share\\_link](https://drive.google.com/file/d/1Je7VqgCvSWRMy8_E5BQ9MPQhEr0_AA3s/view?usp=share_link)

## Proof of Concept (PoC)

```
(om🐼 beast)-[~/Desktop]  
$ john valid.txt --show  
Donal234:Donal234123  
Radhika:Radhika123  
Popeye786:Popeye786123  
admin:admin
```

## **Business Impact - High**

1. Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.
2. Attacker can login to anyone's account and perform malicious activities like editing account info, making purchases , changing product prices.

## **Recommendations**

1. Whitelist User Input: Whitelist all user input for expected data only. For example if you are expecting a flower name, limit it to alphabets only upto 20 characters in length. If you are expecting some ID, restrict it to numbers only.
2. Assign each Database user only the required permissions and not all permissions.
3. Disable/remove default accounts, passwords and databases

## References

- [https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection)
- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)



## 4. Directory Listing -

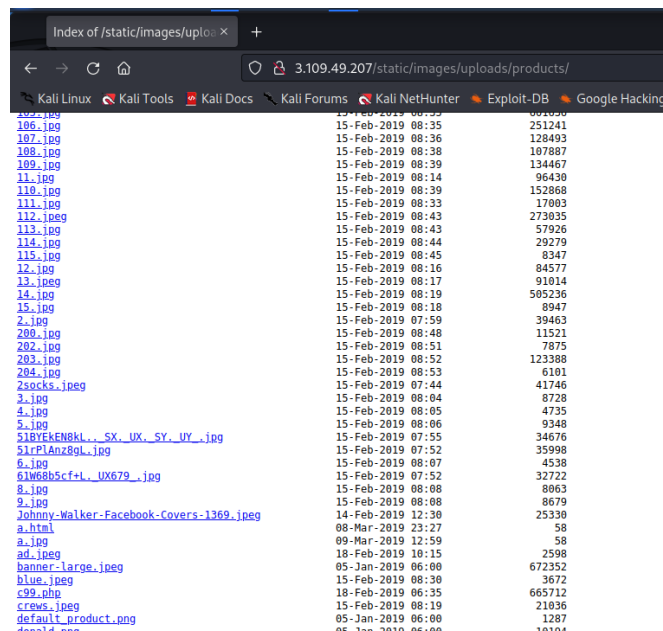
- The following URL's of *Lifestyle Store* was found vulnerable to Directory Listing-

Directory Listing
<p>Affected URL's:-</p> <p><a href="http://3.109.49.207/static/images/uploads/products/">http://3.109.49.207/static/images/uploads/products/</a> <a href="http://3.109.49.207/static/images/uploads/customers/">http://3.109.49.207/static/images/uploads/customers/</a> <a href="http://13.126.144.166/server-status/">http://13.126.144.166/server-status/</a></p>

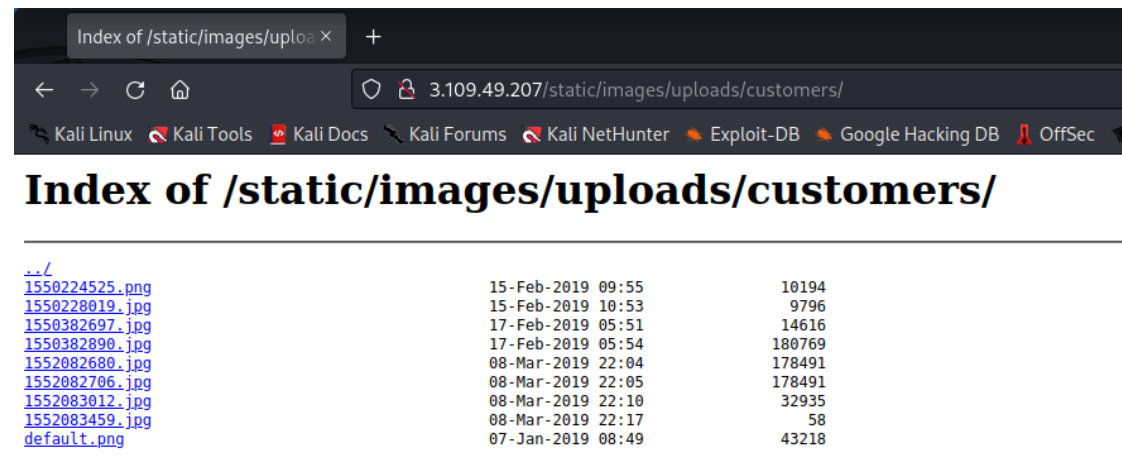
## Observation

- An attacker can easily access the affected URL's and view the content.

## Proof of Concept (PoC)



File Name	Upload Date	Size
100.jpg	15-Feb-2019 08:35	251241
101.jpg	15-Feb-2019 08:36	128493
102.jpg	15-Feb-2019 08:38	107887
103.jpg	15-Feb-2019 08:39	134467
104.jpg	15-Feb-2019 08:14	96430
105.jpg	15-Feb-2019 08:39	152868
106.jpg	15-Feb-2019 08:33	17093
107.jpg	15-Feb-2019 08:43	273035
108.jpg	15-Feb-2019 08:43	57926
109.jpg	15-Feb-2019 08:44	29279
110.jpg	15-Feb-2019 08:45	8347
111.jpg	15-Feb-2019 08:16	84577
112.jpg	15-Feb-2019 08:17	91014
113.jpg	15-Feb-2019 08:19	585236
114.jpg	15-Feb-2019 08:18	8947
115.jpg	15-Feb-2019 07:59	39463
116.jpg	15-Feb-2019 08:48	11521
117.jpg	15-Feb-2019 08:51	7875
118.jpg	15-Feb-2019 08:52	123388
119.jpg	15-Feb-2019 08:53	6101
120.jpg	15-Feb-2019 07:44	41746
121.jpg	15-Feb-2019 08:04	8728
122.jpg	15-Feb-2019 08:05	4735
123.jpg	15-Feb-2019 08:06	9348
124.jpg	15-Feb-2019 07:55	34676
125.jpg	15-Feb-2019 07:52	35998
126.jpg	15-Feb-2019 08:07	4538
127.jpg	15-Feb-2019 07:52	32722
128.jpg	15-Feb-2019 08:08	8063
129.jpg	15-Feb-2019 08:08	8679
130.jpg	14-Feb-2019 12:30	25330
131.jpg	08-Mar-2019 23:27	58
132.jpg	09-Mar-2019 12:59	58
133.jpg	18-Feb-2019 10:15	2598
134.jpg	05-Jan-2019 06:00	672352
135.jpg	15-Feb-2019 08:30	3672
136.jpg	18-Feb-2019 06:35	665712
137.jpg	15-Feb-2019 08:19	21036
138.jpg	05-Jan-2019 06:00	1287
139.jpg	05-Jan-2019 06:00	10000



### Index of /static/images/uploads/customers/

File Name	Upload Date	Size
1550224525.png	15-Feb-2019 09:55	10194
1550228019.jpg	15-Feb-2019 10:53	9796
1550382697.jpg	17-Feb-2019 05:51	14616
1550382890.jpg	17-Feb-2019 05:54	180769
1552082680.jpg	08-Mar-2019 22:04	178491
1552082706.jpg	08-Mar-2019 22:05	178491
1552083012.jpg	08-Mar-2019 22:10	32935
1552083459.jpg	08-Mar-2019 22:17	58
default.png	07-Jan-2019 08:49	43218

## **Business Impact - Low**

- No confidential data is leaked with this vulnerability. It has low impact on business firms.

## **Recommendation**

1. Disable directory listing for entire application.
2. In business needs, create a directory and enable directory listing only for that alone. All web servers have these options to configure.

## **References**

[https://portswigger.net/kb/issues/00600100\\_directory-listing](https://portswigger.net/kb/issues/00600100_directory-listing)

## 5. Insecure Direct Object Reference (IDOR) -

### IDOR

Affected URL's:-

<http://3.109.49.207/profile/15/edit/>

Parameter- profile ID i.e 15

<http://3.109.49.207/orders/orders.php?customer=2>

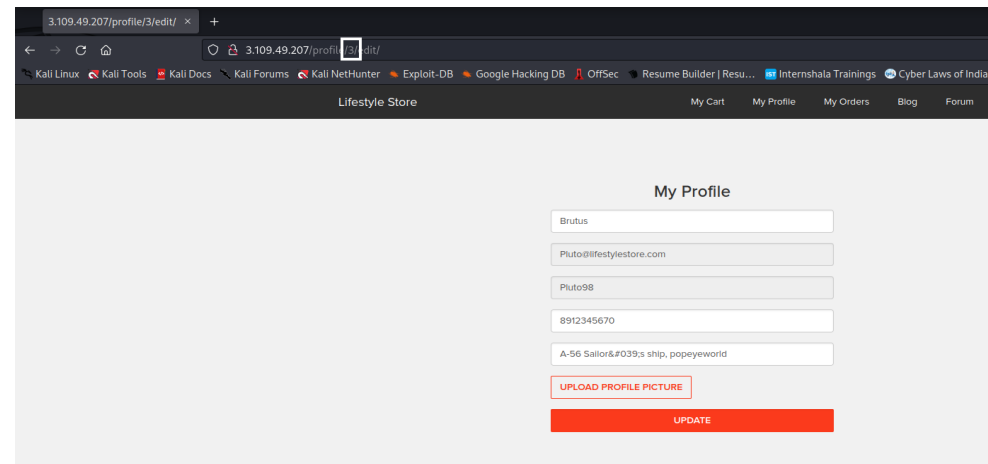
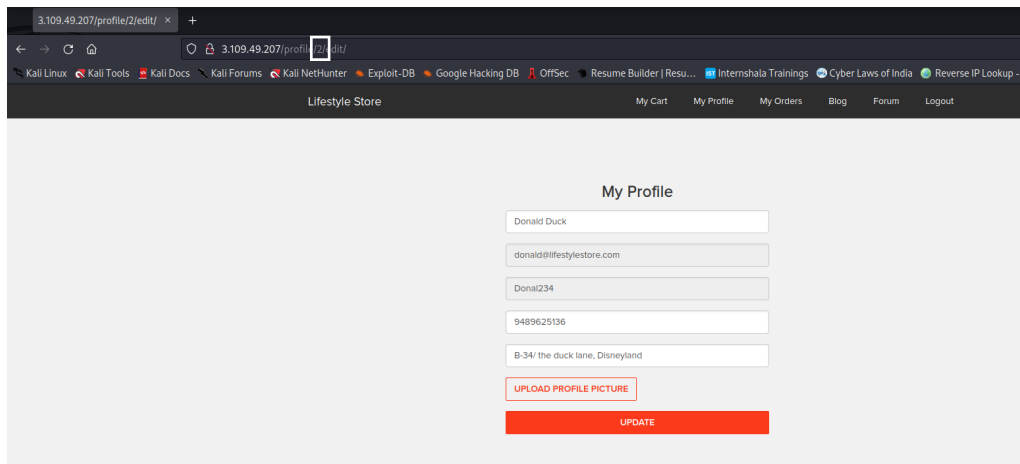
Parameter- customer

## Observations

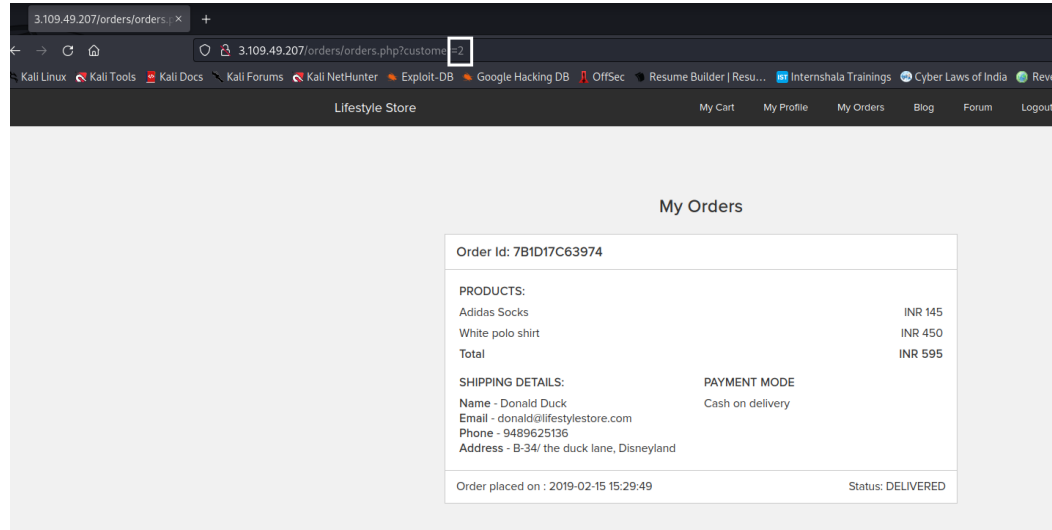
1. Once the attacker is logged-in to the web-app, he can change the profile id value in the URL to see account information of another user.
2. Similarly, when a order receipt is generated, attacker can tamper with the order id in URL to get order details of another account.

## Proof of Concept (PoC)

### *Observation 1-*



## Observation 2-



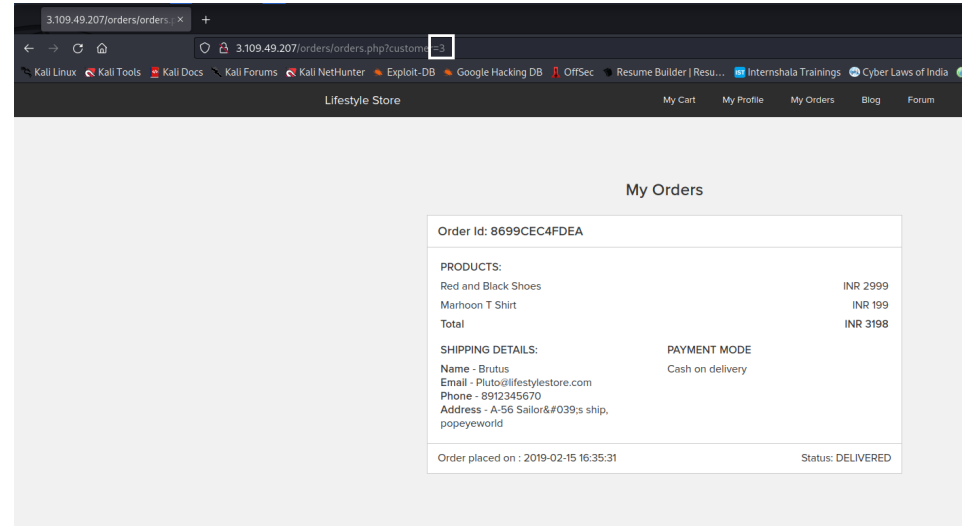
3.109.49.207/orders/orders.php?custom...2

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

### My Orders

Order Id: 7B1D17C63974	
<b>PRODUCTS:</b>	
Adidas Socks	INR 145
White polo shirt	INR 450
Total	INR 595
<b>SHIPPING DETAILS:</b>	<b>PAYMENT MODE</b>
Name - Donald Duck	Cash on delivery
Email - donald@lifestylestore.com	
Phone - 9489625136	
Address - B-34/ the duck lane, Disneyland	
Order placed on : 2019-02-15 15:29:49	Status: DELIVERED



3.109.49.207/orders/orders.php?custom...2

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

### My Orders

Order Id: 8699CEC4FDEA	
<b>PRODUCTS:</b>	
Red and Black Shoes	INR 2999
Marhoon T Shirt	INR 199
Total	INR 3198
<b>SHIPPING DETAILS:</b>	<b>PAYMENT MODE</b>
Name - Brutus	Cash on delivery
Email - Pluto@lifestylestore.com	
Phone - 8912345670	
Address - A-56 Sailor&#039;s ship, popeyeworld	
Order placed on : 2019-02-15 16:35:31	Status: DELIVERED

## **Business Impact - High**

This vulnerability leads to exposure of confidential information which further could be used for malicious activities like social engineering. And further may lead to the account takeover.

## **Recommendations**

1. Validation of Parameters should be properly implemented.
2. Verification of all the Referenced objects should be done.
3. Validation of user input should be properly implemented.

## 6. Cross-site scripting (XSS) -

### XSS

Affected URL's:-

*Reflected Xss :-*

Parameter- at end of URL

*Stored Xss :-*

<http://13.126.144.166/profile/2/edit/>

<http://13.126.144.166/ovidentiaCMS/index.php?tg=groups>

<http://13.126.144.166/ovidentiaCMS/index.php?tg=site&idx=create>

<http://13.126.144.166/ovidentiaCMS/index.php?tg=notes&idx=Create>

<http://13.126.144.166/ovidentiaCMS/index.php?tg=admfaqs&idx=Add>

Parameters – Nom, address, description.



## Exploitation

- When the Url's were visited and specific payload were inserted, they found to be vulnerable to Xss attack.

### Payloads used for exploitation -

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

## **Proof of Concept** (PoC)

*Link to PoC of Xss vulnerability in the URL's -*

- [https://drive.google.com/file/d/1CY\\_E0A-w9H1sNVCwVU7pBPzAUPNGXYlh/view?usp=share\\_link](https://drive.google.com/file/d/1CY_E0A-w9H1sNVCwVU7pBPzAUPNGXYlh/view?usp=share_link)

## **Business Impact - High**

1. Xss attack can severely impact websites and web applications, damage their reputation and relationships with customers.
2. Xss can deface websites, can result in compromised user accounts, and can run malicious code on web pages, which can lead to a compromise of the user's device.
3. Attackers can clone the login page of the web application and then use cross-site scripting vulnerabilities to serve it to the victims.
4. Data leakage - Once the attacker has access to the personal or sensitive information of users, they can demand ransom payments from the organization to delete the data, or leak the information of their customers.

## **Recommendation**

1. Sanitize user input
2. Limit use of user-provided data
3. Validate user input and do not allow special characters to render.

## **References**

[https://cheatsheetseries.owasp.org/cheatsheets/Cross\\_Site\\_Scripting\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html)

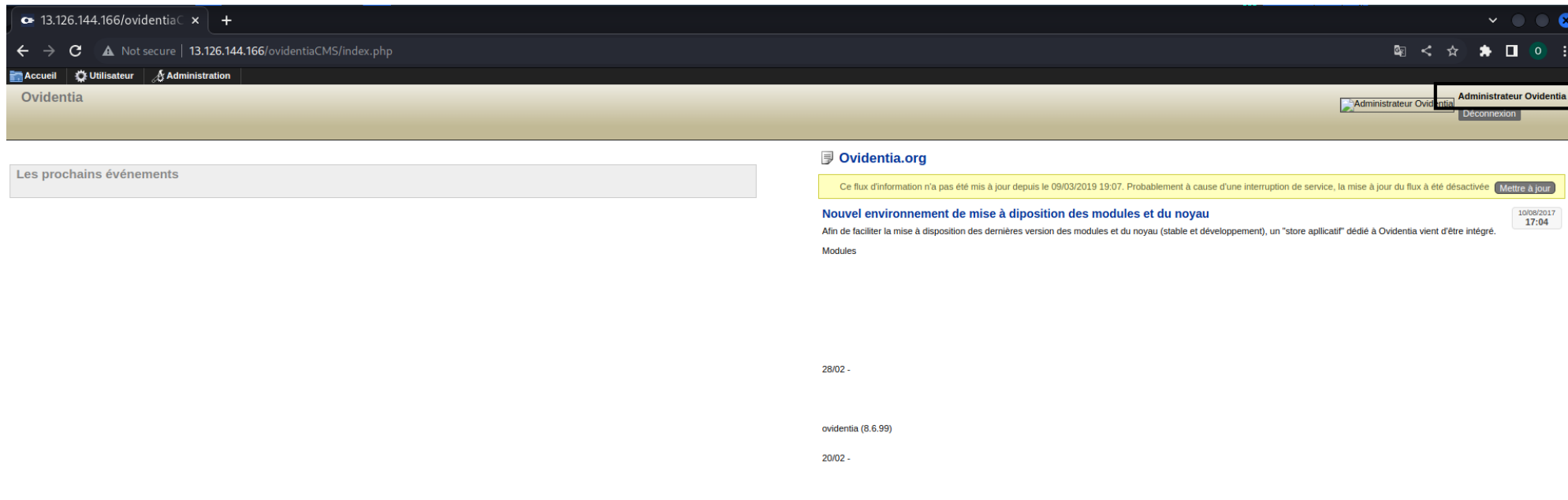
## 7. Weak Passwords -

	<b>Weak Passwords</b>
	<p data-bbox="491 510 804 551">Affected URL:-</p> <p data-bbox="491 618 1832 707"><a href="http://13.126.144.166/ovidientiaCMS/index.php?tg=login&amp;cmd=authform&amp;msg=Connexion&amp;err=&amp;restricted=1">http://13.126.144.166/ovidientiaCMS/index.php?tg=login&amp;cmd=authform&amp;msg=Connexion&amp;err=&amp;restricted=1</a></p> <p data-bbox="491 771 953 812">Parameter- password</p>

### **Observation**

- On visiting the URL, after fuzzing around with password field, The password field was easily cracked and was found vulnerable to weak password attack.
- A hacker could easily access the admin account by entering admin ID and password as '12345678'.

## Proof of Concept (PoC)



## **Business Impact - High**

- Using weak password can cause data breaches , further can lead to account takeover, exposure of sensitive data like financial details, intellectual property etc. Which later on can cause reputation damage to the organization.

## **Recommendations**

1. Use complex password for accounts.
2. Change the passwords every 90 days.
3. Use alpha-numeric and special symbols while creating passwords.
4. Use 8 to 12 digits password length.



## 8. Cross-site Request Forgery (CSRF) -

### CSRF

Affected URL:-

[http://13.126.144.166/profile/change\\_password\\_submit.php](http://13.126.144.166/profile/change_password_submit.php)

Parameter- password,password\_confirm

### **Observation**

- Once a user is logged-in to his/her account. If the user clicks the link provided from a external source then the user's account password could be changed without consent.

### **Exploitation**

1. log-in to the account and click on the 'submit' button of script opened in another tab.
2. After clicking, the user password is changed.

Link to the script -

- [https://drive.google.com/file/d/1GQn9n\\_jSxMjUTQfPipl2Ye\\_uerYgL\\_fi/view?usp=share\\_link](https://drive.google.com/file/d/1GQn9n_jSxMjUTQfPipl2Ye_uerYgL_fi/view?usp=share_link)

## **Proof of Concept (PoC)**

- Here's PoC for CSRF attack. Initially the user is logged in with his account details- username='Donal234' password='12345678' but after clicking on link he was not able to login with same password. The password was changed to "test".

PoC link for CSRF attack:-

[https://drive.google.com/file/d/1pO2qHtKN36bTgRlzxMEbt9Cg\\_DWL7TS/view?usp=share\\_link](https://drive.google.com/file/d/1pO2qHtKN36bTgRlzxMEbt9Cg_DWL7TS/view?usp=share_link)

## **Business Impact - High**

- If an attacker successfully performs a CSRF attack against the victim's account, they can transfer funds, purchase a product, modify account information such as the shipping address, modify the password, or any other action available when the user is signed in.
- The impact can be severe and can damage the client relationship.

## **Remediation**

- <https://www.acunetix.com/websitesecurity/csrf-attacks/>
- <https://www.veracode.com/blog/secure-development/preventing-csrf-attacks>
- [https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site\\_Request\\_Forgery\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html)

## 9. Forced Browsing -

	<b>Forced Browsing</b>
	<p data-bbox="491 510 804 551">Affected URL:-</p> <p data-bbox="491 618 1481 662"><a href="http://13.126.144.166/admin31/dashboard.php">http://13.126.144.166/admin31/dashboard.php</a></p> <p data-bbox="491 671 1417 715"><a href="http://13.126.144.166/admin31/console.php">http://13.126.144.166/admin31/console.php</a></p>

### **Observation**

- Once the user is logged-in to the web-app. He can access the admin dashboard and the admin console without any authorisation which proves the URL's to be vulnerable to Forced-Browsing attack.

### **Exploitation**

1. Log in to account
2. Access the affected URL
3. The url is being accessed without authorisation of user type i.e either he's customer/seller/admin. There are no access rights set.

## **Proof of Concept** (PoC)

*Here is the PoC link for Forced Browsing attack -*

- [https://drive.google.com/file/d/1iyfhvTzkGUncOBFdgnqFDkGCWdWdN1m2/view?usp=share\\_link](https://drive.google.com/file/d/1iyfhvTzkGUncOBFdgnqFDkGCWdWdN1m2/view?usp=share_link)



## **Business Impact - High**

- With this vulnerability, a hacker can access the admin panel without proper authorization, and can fuzz around with the product prices,description,name. This could lead to spoil customer relationship and reputation of the organization.

### **Remediation**

- For every web page that is accessed, the developer must ensure that only the authenticated user is authorized to gain access to the content.
- Authenticated users shouldn't be able to use authorized content. An authorization check is needed at every step to ensure security.
- Creating a white list, allowing explicit access to a set of URLs that are considered, allows part of the application to exercise its functionality as intended. Any request not in this URL space is denied by default.

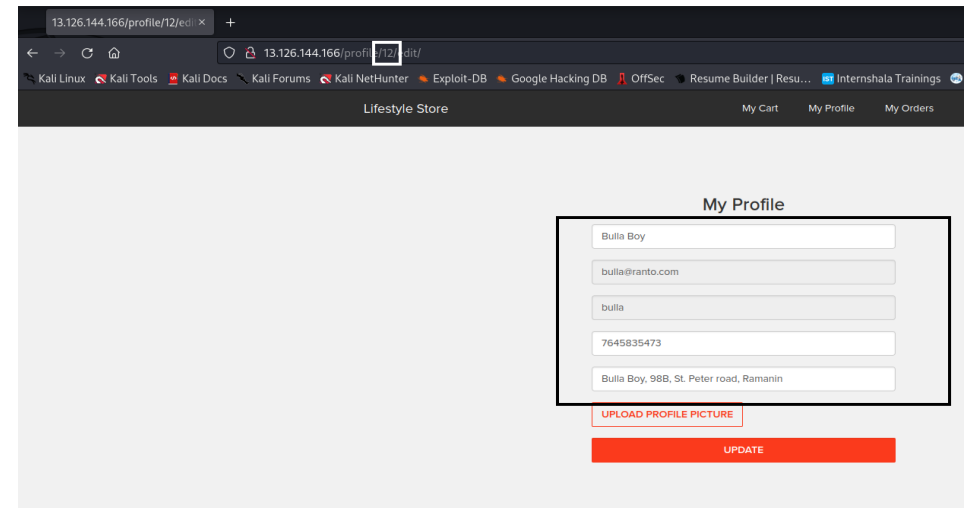
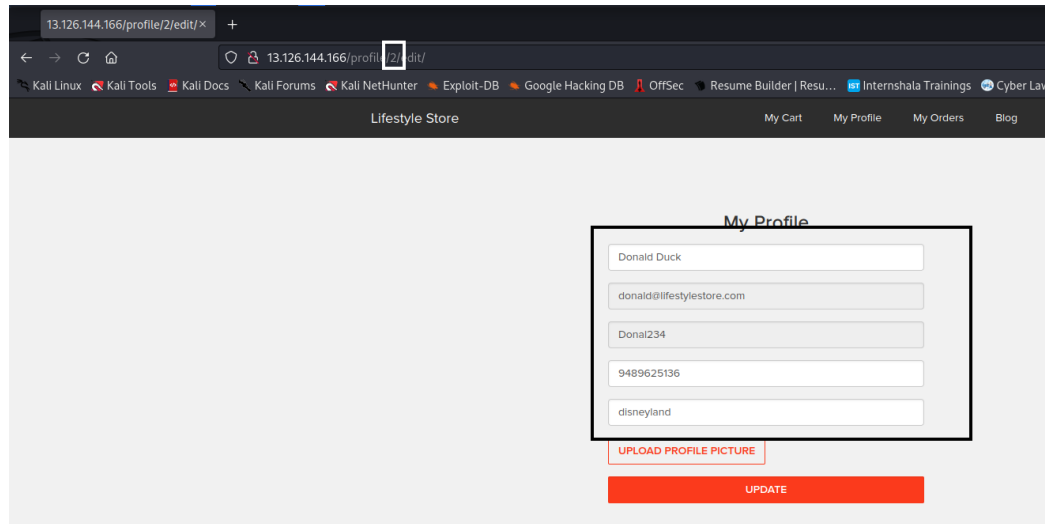
## 10. Personal Identifiable Information Leakage (PII) -

	PII Leakage
	<p data-bbox="491 510 804 551">Affected URL:-</p> <p data-bbox="491 618 1236 665"><a href="http://13.126.144.166/profile/2/edit/">http://13.126.144.166/profile/2/edit/</a></p> <p data-bbox="491 727 915 774">Parameter-profile id</p>

## Observation

- Hackers can change the profile id in the URL of target web-app and can have access to personal information of another user without any proper authentication and authorization.

## Proof of Concept (PoC)



## **Business Impact - High**

- PII leakage allows attacker to steal customers data , target or harm the individual which could lead organization to loose customer relationship, impact on finances, disruption of databases.

## **Remediation**

- <https://www.geeksforgeeks.org/personally-identifiable-information-leakage-vulnerability/>
- <https://www.upguard.com/blog/data-leak-prevention-tips>

## 11. Default Files and Pages -

### Default Files and Pages

Affected URL's:-

<http://13.126.144.166/robots.txt>

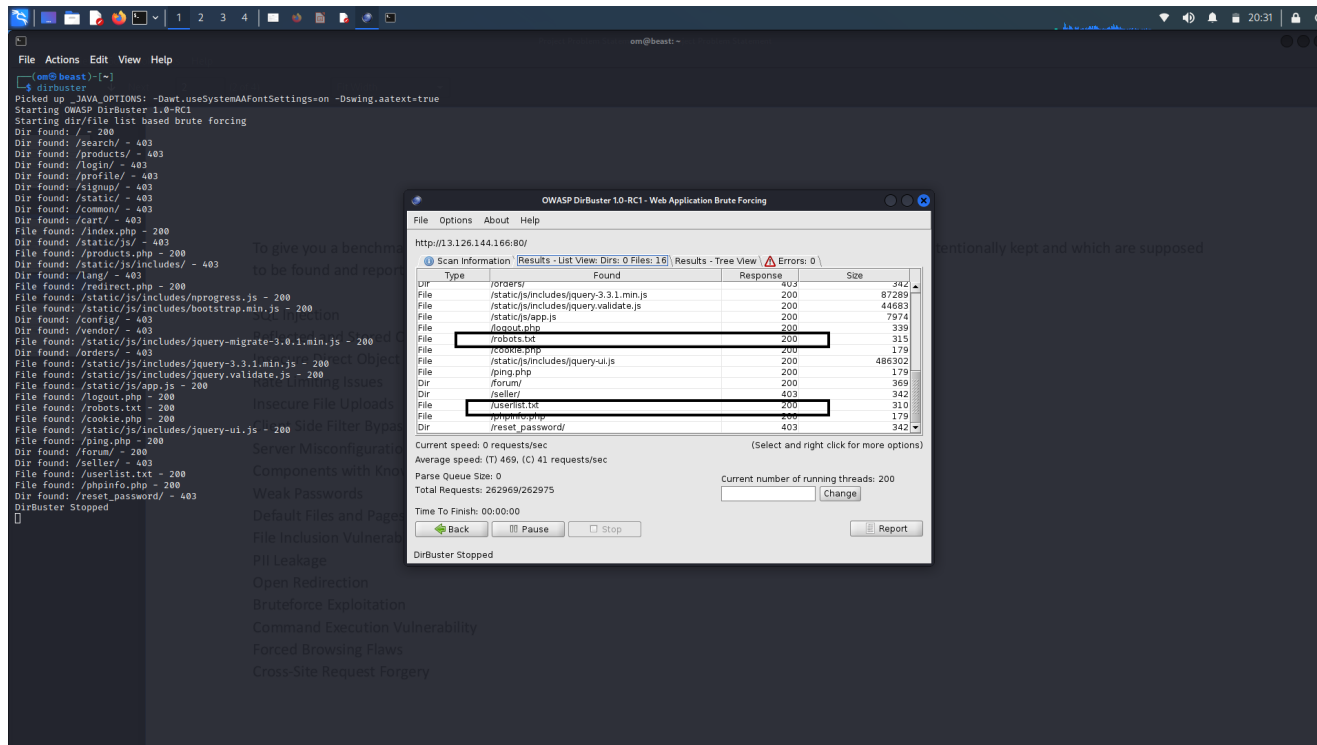
<http://13.126.144.166/ovidientiaCMS/install/install.txt>

<http://13.126.144.166/userlist.txt>

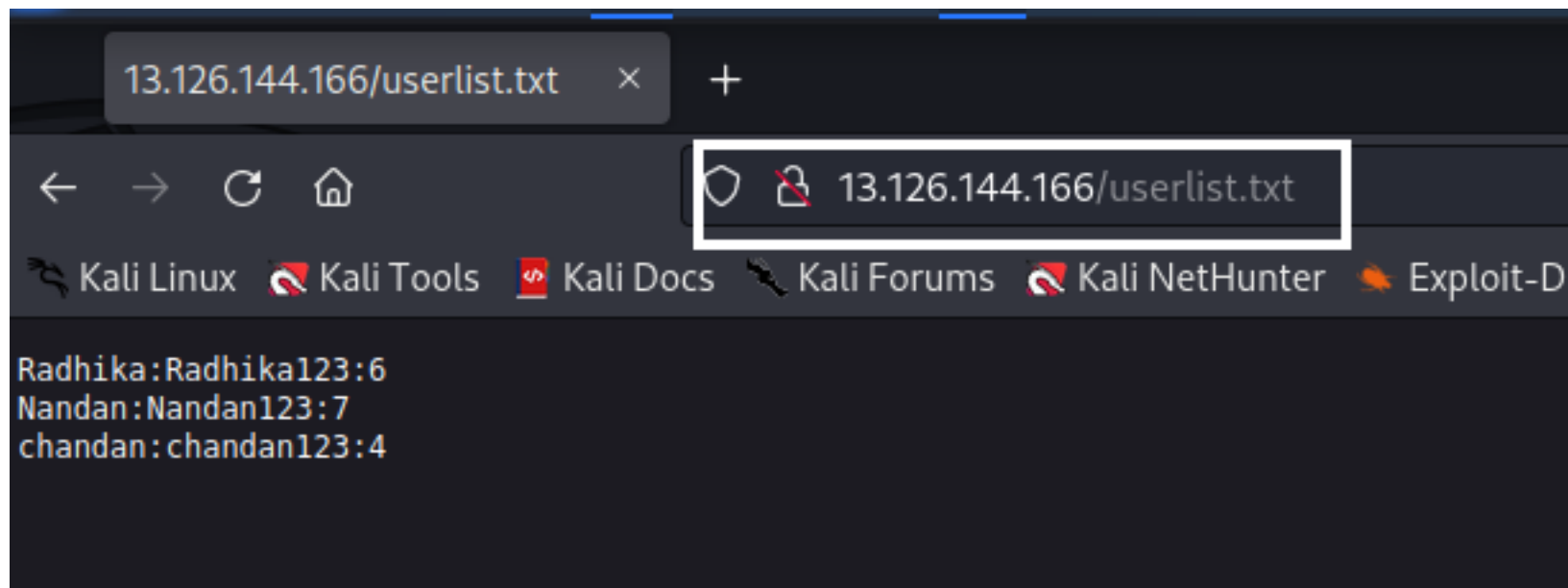
<http://13.126.144.166/phpinfo.php>

# Exploitation

- Using Dirbuster tool a hacker can successfully extract common files and pages from the web-app.



## Proof of Concept (PoC)



*The following URL revealed list of username and their passwords to login to web-app.*

## **Business Impact - High**

- Some of the default files contained juicy information regards web-app and a malicious hacker can use this information to disrupt the working of website, this may cause severe impact for the organization. It may lead to loss of customer relationship.

## **Remediation**

- Do not allow listing of files and folders publicly.



## 12. Client Side Filter Bypass -

	Client Side Filter Bypass
	<p data-bbox="436 521 753 563">Affected URL:-</p> <p data-bbox="436 629 1423 677"><a href="http://13.126.144.166/admin31/dashboard.php">http://13.126.144.166/admin31/dashboard.php</a></p>

## **Observation**

- A hacker can change the file type before uploading to web-app and intercept the request in Burpsuite and revert the file type back to its original form. This way hacker can bypass the client side filters and upload any malicious file to web-app.

## **Proof of Concept (PoC)**

*Link to PoC of Client-Side Filter Bypass -*

- [https://drive.google.com/file/d/16Zu0MLgzzlY7S9Vmvo9sViG8CbFBRYsc/view?usp=share\\_link](https://drive.google.com/file/d/16Zu0MLgzzlY7S9Vmvo9sViG8CbFBRYsc/view?usp=share_link)

## **Business Impact - High**

- A hacker can upload malicious files on the web-app that could reveal critical information from database, also the scripts can affect the users logged-in to the web-app. Sensitive data can be extracted and can cause major damage to the web-app that could spoil the organization's reputation and may impact the customer relation.

## **Remediation**

- Use of special tokens while passing the request to prevent altering with the fields.

## **References**

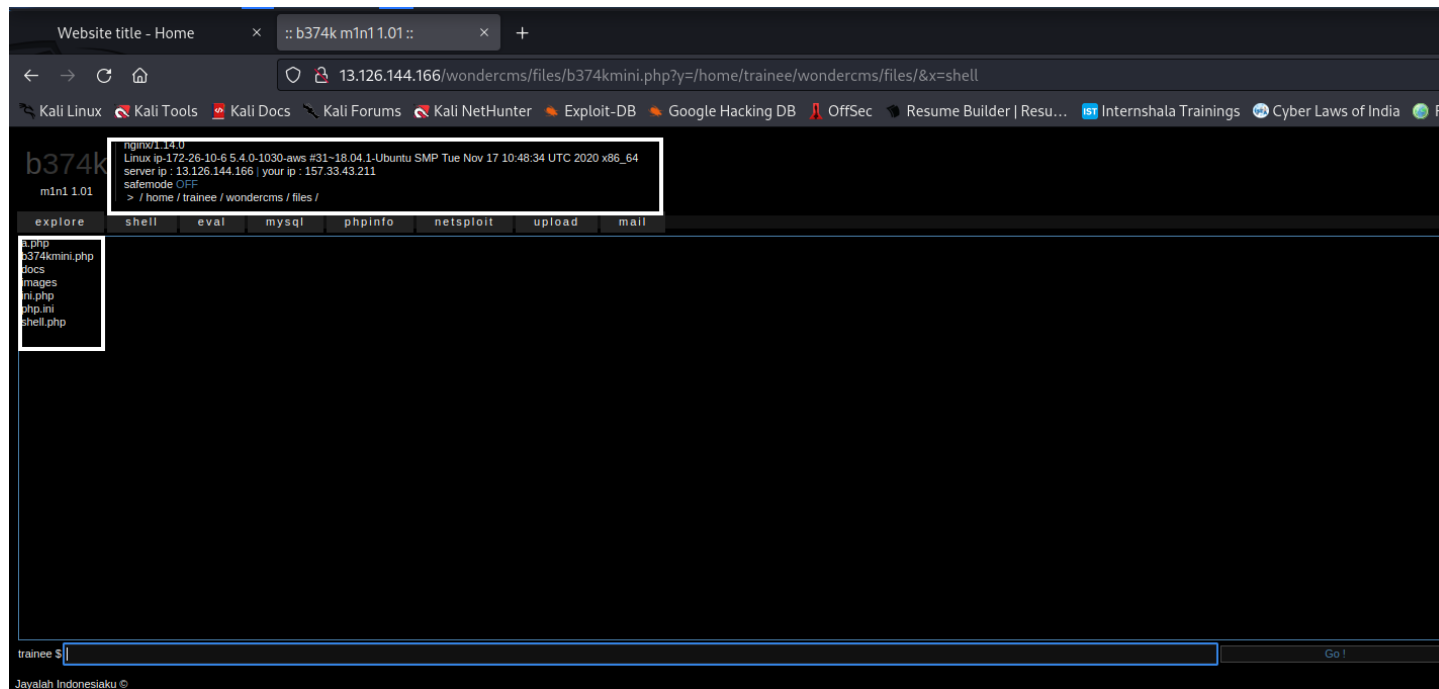
- [https://cheatsheetseries.owasp.org/cheatsheets/Input\\_Validation\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html)

## 13. Insecure File Upload -

	Insecure File Upload
	Affected URL:- <a href="http://13.126.144.166/wondercms/">http://13.126.144.166/wondercms/</a>

## Proof of Concept (PoC)

- Here, we uploaded a web-shell called b374kmini.php that gives a complete
- of web-app. We can execute commands, lists the available files in the directory, along with it we get other critical info displayed about web-app.



## **Business Impact - High**

- Attacker can upload malicious scripts that can deface the website, extract sensitive information leading to severely impact the organization's reputation.

### **Remediation**

- Allow only certain file extension.
- Set maximum file size and name length.
- Allow only authorized users.
- Make sure the fetched file from the web is an expected one.
- Keep your website updated.
- Block uploads from bots and scripts using captcha.

## 14. Command Execution Vulnerability -

	<b>Command Execution Vulnerability</b>
	Affected URL:-  <a href="http://13.126.144.166/admin31/dashboard.php">http://13.126.144.166/admin31/dashboard.php</a>



## **Exploitation**

1. Create a php script and save it in 'png' format to bypass client side filter
2. Intercept the request in burpsuite and revert the file extension back to '.php'
3. When the file is successfully loaded, visit the location where file was uploaded and execute the script.

## **Proof of Concept** (PoC)

*PoC link for File Upload Vulnerability -*

- [https://drive.google.com/file/d/1--hivchj4Yi6RFYWzcVXrTvvSGsV1h42/view?usp=share\\_link](https://drive.google.com/file/d/1--hivchj4Yi6RFYWzcVXrTvvSGsV1h42/view?usp=share_link)

## **Business Impact - High**

- A hacker can upload malicious files on the web-app and execute malicious commands to reveal sensitive information from database, also the scripts can affect the users logged-in to the web-app. Sensitive data can be extracted and can cause major damage to the web-app that could spoil the organization's reputation and may impact the customer relation.

## **Remediation and Reference**

1. Only allow specific file types.
  2. Verify file types and Store uploaded files outside the web root folder.
  3. Randomize uploaded file names.
- [https://owasp.org/www-community/vulnerabilities/Unrestricted\\_File\\_Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)

## 15. Server Misconfiguration -

### Server Misconfiguration

Affected URL:-

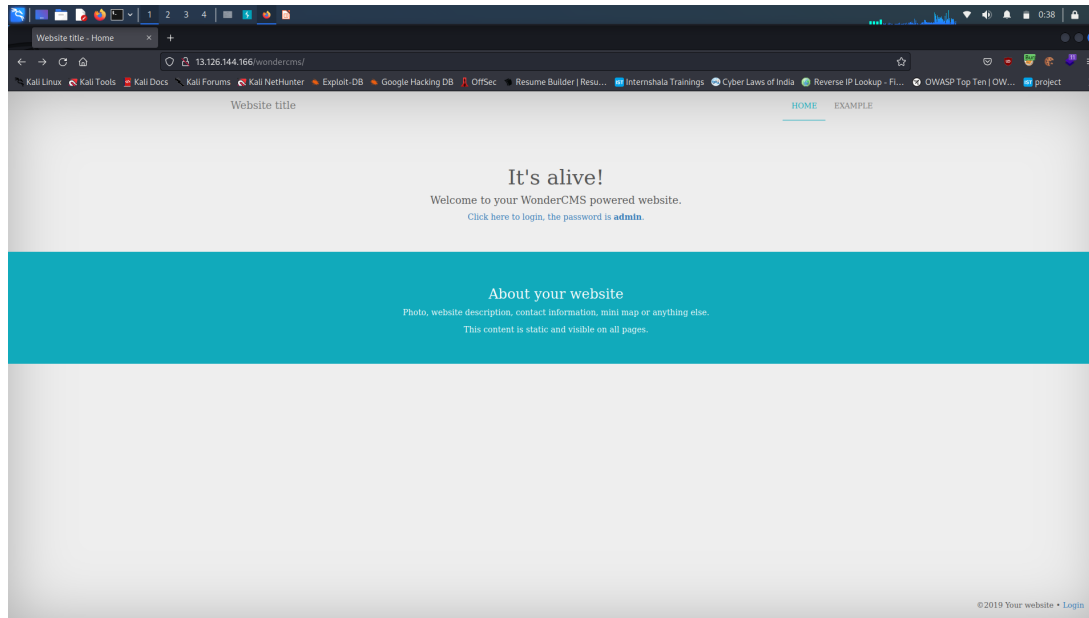
<http://13.126.144.166/wondercms/loginURL>

<http://13.126.144.166/ovidentiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1>

-

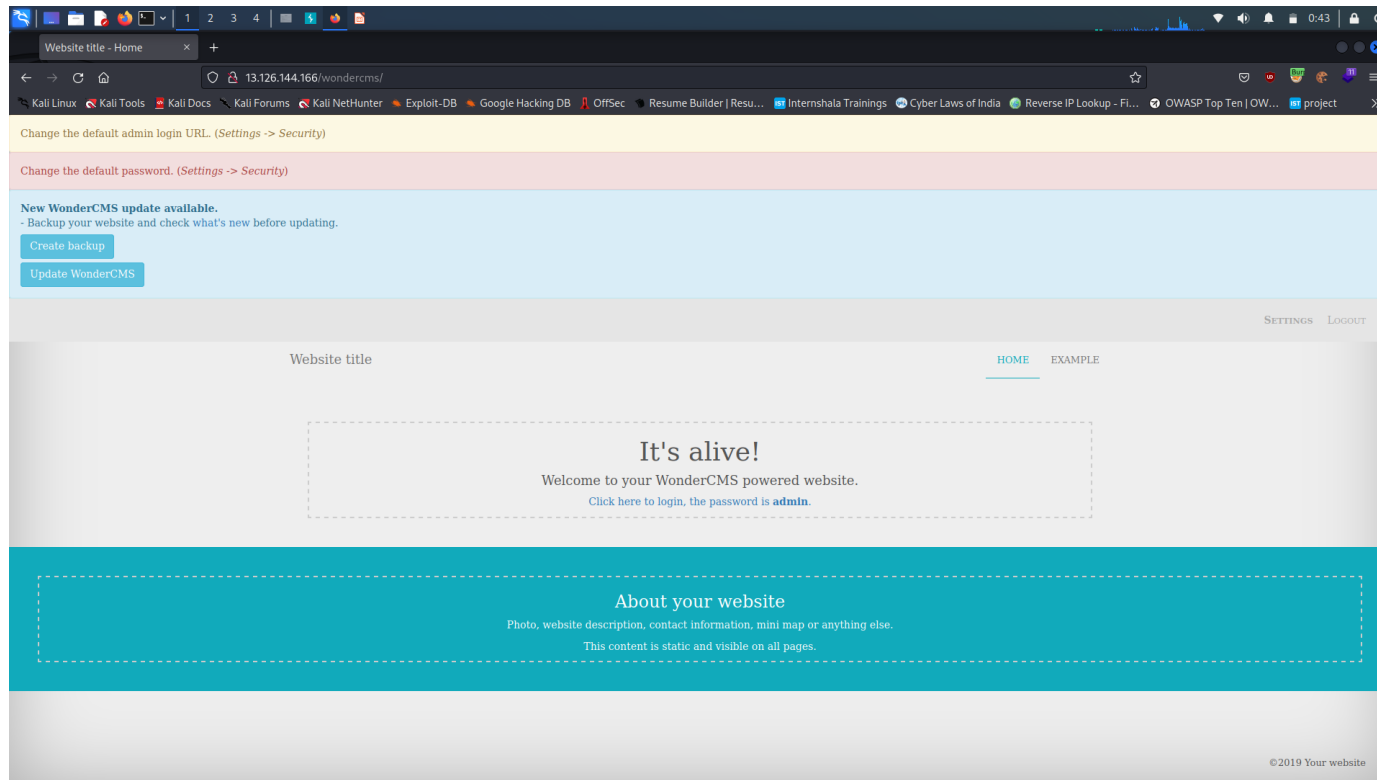
## Observation: 1

Admin login details are clearly mentioned on the page



## Proof of Concept (PoC)

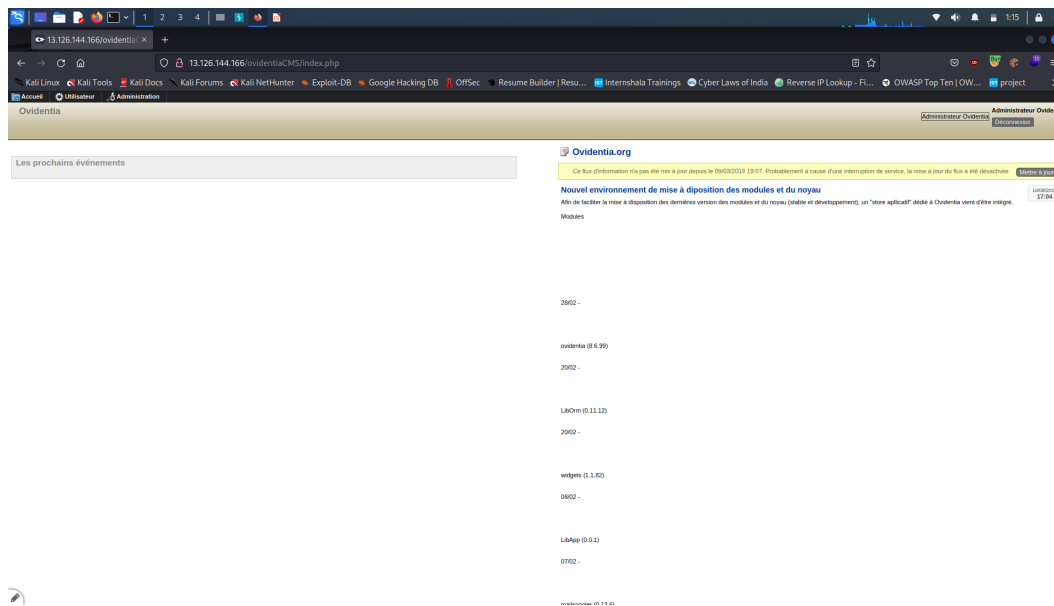
We could successfully login to the admin account



## Observation: 2

- OvidentiaCMS uses default login credentials i.e [username=admin@admin.bab](#) and password=012345678 anyone carrying this credentials can access admin account and alter the website's interface.

## Proof of Concept (PoC)



## **Business Impact - High**

- Anyone visiting the website can login to admin account. Once a malicious user gets admin access he can alter the website contents and can have severe impact on website and its users. The attacker can also deface the website and this all activities could spoil the reputation of organization.

### **References**

- <https://brightsec.com/blog/security-misconfiguration/>
- [https://owasp.org/Top10/A05\\_2021-Security\\_Misconfiguration/](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

## 16. Components with known vulnerability -

	<b>Components with known vulnerability</b>
	<p data-bbox="478 487 783 535">Components:-</p> <p data-bbox="478 595 804 644"><a href="#"><u>Ovidentia 8.6.4</u></a></p> <p data-bbox="478 650 1255 698"><a href="#"><u>OpenSSH 7.6p1 Ubuntu 4ubuntu0.5</u></a></p> <p data-bbox="478 704 744 752"><a href="#"><u>nginx 1.14.0</u></a></p>



## Exploitation

- Using Nmap the web-app was found running an out-dated software versions which further could be used for getting unauthorised access or cause damage to web-app.
- Also, after fuzzing around in OvidentiaCMS it was found to be running on an Out-dated version i.e 8.6.4
- On googling OvidentiaCMS 8.6.4, the website was found vulnerable to XSS attack.

## Proof of Concept (PoC)

```

File Actions Edit View Help
(om@beast)-[~]
$ nmap -A 13.126.144.166 -p 22,80
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-15 01:27 IST
Nmap scan report for ec2-13-126-144-166.ap-south-1.compute.amazonaws.com (13.126.144.166)
Host is up (0.50s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 fad0ea9c8d04842b2aea144dba43cb25 (RSA)
|   256 3741be65a5a51eb730cd7086e4644e9c (ECDSA)
|_  256 0d60efc1576c9f4412bbd9cc36fdbff2 (ED25519)
80/tcp    open  http      nginx 1.14.0 (Ubuntu)
|_ http-robots.txt: 2 disallowed entries
|_ /static/images/ /ovidentiaCMS
|_ http-server-header: nginx/1.14.0 (Ubuntu)
|_ http-cookie-flags:
|   /:
|   PHPSESSID:
|   httponly flag not set
|_ http-title: Lifestyle Store
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.86 seconds

```

Exploit link for OvidentiaCMS -

- <https://github.com/Kitsun3Sec/exploits/blob/master/cms/ovidentia/exploitXSSOvidentia.txt>

## **Business Impact - Low**

- This vulnerability doesn't directly targets the server or cause any potential harm, but a negligible or low impact can occur due to this vulnerability.

### **Recommendation**

1. Remove unused dependencies, unnecessary features, components, files, and documentation.
2. Patch the component as quickly as possible and upgrade to latest versions
3. Monitor for libraries and components that are unmaintained or do not create security patches for older versions.

# THANK YOU

For any further clarification please contact :  
*omkarholkar774@gmail.com*