



# Hacking Environment Web Application

Lifestyle Store  
Detailed Developer Report

# **SECURITY STATUS – EXTREMELY VULNERABLE**

- 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)

# **VULNERABILITY STATISTIC**



## **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 (Critical)

Below mentioned URL in the **T-shirt/socks/shoes** module is vulnerable to SQL injection attack  
Affected URL :

`http://13.233.92.247/products.php?cat=1`

Affected Parameters :  
•cat (GET parameter)

Payload:  
•cat = 1'

Affected URL :

`http://13.233.92.247/products.php?q=socks`

Affected Parameters :  
•q (GET parameter)

Payload:  
•q=socks'

# 1. SQL Injection

SQL Injection (Critical)	<p>Here are other similar SQLi in the application</p> <p><b>Affected URL :</b></p> <p><a href="http://13.233.92.247/products.php?cat=2">http://13.233.92.247/products.php?cat=2</a></p> <p><a href="http://13.233.92.247/products.php?cat=3">http://13.233.92.247/products.php?cat=3</a></p>
-----------------------------	--

# Observation

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

The screenshot shows a web browser window with the URL `13.233.92.247/products.php?cat=1` in the address bar. The page is titled "Lifestyle Store". The navigation menu includes "Blog", "Forum", "Sign Up", and "Login". A search bar with a magnifying glass icon is present. Below the menu, there are three categories: "T Shirt", "Socks", and "Shoes".

The main content displays three products:

- Basic T shirt**: Price 350. An image placeholder is shown.
- Simple T Shirts**: Price 550. An image shows three t-shirts: one dark blue, one white, and one red.
- Plain Tee**: Price 300. An image shows a person wearing a maroon long-sleeved shirt.

Each product has a "VIEW PRODUCT" button below it.

# Observation

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



# Observation

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

# Proof of Concept (PoC):-

## Attacker can dump arbitrary data

- **No of databases:** 2
  - information\_schema
  - hacking\_training\_project
- **No of tables :** 10
  - brands
  - cart\_items
  - categories
  - customers
  - order\_items
  - orders
  - product\_reviews
  - products
  - sellers
  - user

Database: hacking_training_project			
Table: users			
[15 entries]			
user_name	password	phone_number	unique_key
admin	\$2y\$10\$xkmdvrxCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	8521479630	15468927955c66694cba1174.29688447
Donald234	\$2y\$10\$PM.7nBSP5FMaldXiM/S3s./p5xR6GTVjry7ysJtxOkBqOJURAHsO	9489625136	778522555c6669996f5a24.34991684
Pluto98	\$2y\$10\$xkmdvrxCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	8912345670	19486318945c666a037b1432.99985767
chandan	\$2y\$10\$cZBEIrgthxqvT1hwU1ivuFELe03rR.GICdp03Njr1S0VeioKLVDa	7854126395	12404594545c666a3b49e0f8.08173871
Popeye786	\$2y\$10\$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	9745612300	18430379145c666a53af8431.79566371
Radhika	\$2y\$10\$RYxNhOyV/G4g7OtFwpqYaxvHi8rF6xxui8kT1wtrfqhTutCA8JC.	9512300052	15611262655c666b312f73e0.70827297
Nandan	\$2y\$10\$G.cRLNMEiG79ZFXE1Hg.R.o95334U0xmZu4.9MqzR5614ucwnk59K	7845129630	1587354115c666b65bb44a5.36505317
MurthyAdapa	\$2y\$10\$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1dPR.11UubDG	8365738264	16357203785c68f640c699a2.83646347
john	\$2y\$10\$GhDB8h1x6xjPMY12Gz1vD07Y3en97u1/.oXTZLmYqB6F18FBgecvG	6598325015	9946437385c6a435f76bef0.14675944
bob	\$2y\$10\$kiuikn3HPFbuyTtK751LNurxzqc0LX3eMgy0/Ux16J0oG37dCGKLq	8576308560	4305822125c6a43ec507df0.68309267
jack	\$2y\$10\$z/nyNlkRJ76m9ItMZ4N510eRxy6Gkqi9N/UBcJu5Ze07eM7N4pTHu	9848478231	15257114565c6a444692b707.17903432
bul1a	\$2y\$10\$HT5oiRMetqa7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	7645835473	18292501185c6a4493a5ddb0.87138000
hunter	\$2y\$10\$pB3U9iFxwBgSb12AkBpiEeIBdhijYfwy9.y.xv23q12gGbMCyn7N3g2	9788777777	13824560345c80704e821145.26019698
asd	\$2y\$10\$At5pFZnRwpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmwzBFqVIkBQFpCF2	9876543210	8057400125c862a7f5916c9.06111587
acdc	\$2y\$10\$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTlyQtK17FzdSpmIRRbi	9999999999	13104802695c86f43f0c3705.77019309

# **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://www.owasp.org/index.php/SQL_Injection)
- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)

## 2. Access to admin panel

Access to admin  
panel  
(Critical)

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

Affected URL :

<http://13.233.92.247/wondercms/loginURL>

# Observation

- 1) When we navigate to <http://13.233.92.247/wondercms/> URL
- 2) we get the password on the page and login as : admin in the URL  
<http://13.233.92.247/wondercms/loginURL>

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

- Header:** The address bar shows the URL [13.233.92.247/wondercms/](http://13.233.92.247/wondercms/). The page title is "Change the default admin login URL. (Settings -> Security)".
- Middle Content:** A message states "New WonderCMS update available." followed by a bullet point "- Backup your website and check what's new before updating." Below this are two buttons: "Create backup" (in a grey box) and "Update WonderCMS" (in a blue box).
- Bottom Content:** A placeholder website section titled "Website title". It features a large dashed box for a logo, the text "It's alive!", "Welcome to your WonderCMS powered website.", and a link "Click here to login, the password is **admin**".
- Footer:** Navigation links for "HOME" and "E...".

# Proof of Concept (PoC)

Hacker can change the admin password .  
Hacker can also add and delete pages.  
Hacker can upload any malicious file.

The screenshot shows a web browser window with the URL `13.233.92.247/wondercms/`. The page is titled "fault admin login URL. (*Settings -> Security*)". It features a navigation bar with tabs: CURRENT PAGE, GENERAL, FILES, THEMES & PLUGINS, and SECURITY (which is underlined, indicating it is active). Below the tabs, there are three main sections: "ADMIN LOGIN URL", "PASSWORD", and "BACKUP".

- ADMIN LOGIN URL:** A text input field contains the value `loginURL`. A note below the field says: "IMPORTANT: SAVE/REMEMBER YOUR URL AFTER CHANGING /wondercms/loginURL".
- PASSWORD:** This section contains fields for "OLD PASSWORD" and "NEW PASSWORD", and a blue "CHANGE PASSWORD" button.
- BACKUP:** A large blue button labeled "BACKUP WEBSITE" is present. To its right, a link reads "HOW TO RESTORE BACKUPS?".

# 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.owasp.org/index.php/Default_Passwords)
- <https://www.us-cert.gov/ncas/alerts/TA13-175A>

### 3.Arbitrary file uplaod

Arbitrary file upload (Critical)	<p>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 .</p> <p>Affected URL: <a href="http://13.233.92.247/wondercms/">http://13.233.92.247/wondercms/</a></p> <p>Affected parameter: → File upload (POST parameter)</p> <p>The attacker can upload files with extension other than .jpeg .</p> <p>Affected URL : <a href="http://13.233.92.247/profile/16/edit/">http://13.233.92.247/profile/16/edit/</a></p> <p>Affected Parameters : •Upload Profile Photo (POST parameter)</p>
-------------------------------------	--

# Observation

The screenshot shows a web-based file manager interface for WonderCMS. The URL in the browser bar is 13.233.92.247/wondercms/home. The top navigation bar includes tabs for CURRENT PAGE, GENERAL, FILES (which is the active tab), THEMES & PLUGINS, and SECURITY. Below the tabs, there are two main sections: 'UPLOAD' and 'REMOVE FILES'. The 'UPLOAD' section contains a 'Browse...' button, a message 'NO FILE SELECTED.', and a blue 'UPLOAD' button. The 'REMOVE FILES' section lists several files with red 'X' icons next to them, indicating they can be deleted. One file, 'hackhtml.html', is highlighted with a red rectangular box.

MAIN URL: (Settings -> Security)

CURRENT PAGE GENERAL FILES THEMES & PLUGINS SECURITY

UPLOAD

Browse... NO FILE SELECTED. UPLOAD

REMOVE FILES

- /wondercms/files/.htaccess
- /wondercms/files/a.php
- /wondercms/files/b374kmini.php
- /wondercms/files/hackhtml.html
- /wondercms/files/ini.php
- /wondercms/files/php.ini
- /wondercms/files/shell.php

# 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.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/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009))

[\*https://www.owasp.org/index.php/Default\\_Passwords\*](https://www.owasp.org/index.php/Default_Passwords)

[\*https://www.us-cert.gov/ncas/alerts/TA13-175A\*](https://www.us-cert.gov/ncas/alerts/TA13-175A)

# 4. Account Takeover Using OTP Bypass

Account Takeover Using OTP Bypass (Critical)	<p>The below mentioned login page allows login via OTP which can be bruteforced</p> <p><b>Affected URL :</b> <a href="http://13.233.92.247/reset_password/admin.php?otp=123">http://13.233.92.247/reset_password/admin.php?otp=123</a></p> <p><b>Affected Parameters :</b></p> <ul style="list-style-type: none"><li>• OTP (POST parameters)</li></ul>
--	--

# Observation

Navigate to “[http://13.233.92.247/reset\\_password/admin.php?otp](http://13.233.92.247/reset_password/admin.php?otp)” .You will see user login page via “OTP”

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

- Address Bar:** Displays the URL [13.233.92.247/reset\\_password/admin.php?otp=123](http://13.233.92.247/reset_password/admin.php?otp=123).
- Header:** The page title is "Lifestyle Store". To the right, there are navigation links for "Blog", "Forum", "Sign Up", and "Login ▾".
- Content Area:** The main content is titled "Reset Admin Password". It instructs the user to "Enter 3 digit OTP sent on your registered mobile number". Below this is a text input field containing the placeholder "Ex: 321".
- Buttons:** A prominent red button labeled "Reset Password" is centered below the input field.

# Observation

- Following request will be generated containing OTP parameter.
- Now we are bruteforcing it.

Choose an attack type

Attack type: Sniper

Start attack

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://13.233.92.247

Update Host header to match target

Add \$

Clear \$

Auto \$

Refresh

```
1 GET /reset_password/admin.php?otp=$123$ HTTP/1.1
2 Host: 13.233.92.247
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://13.233.92.247/reset_password/admin.php?otp=123
9 Cookie: key=eclzfwf8b8v; PHPSESSID=dc4hto4s7m5dp6o22o98pg9hf4; X-XSRF-TOKEN=1819d654c9c1cald18e8940005ed3d97b03fdbbb7248e44eda28f55830501df3
10 Upgrade-Insecure-Requests: 1
11
12
```

# Observation

- And we easily got the valid otp

Results	Positions	Payloads	Resource pool	Settings	
Filter: Showing all items					
Request	Payload	Status code	Error	Timeout	Length ▾
276	375				
215	314	200			4476
0		200			4380
1	100	200			4380
3	102	200			4380
2	101	200			4380
5	104	200			4380
4	103	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
14	113	200			4380
15	114	200			4380
16	115	200			4380
17	116	200			4380
18	117	200			4380
19	118	200			4380
20	119	200			4380
21	120	200			4380

# POC

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

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

- Address Bar:** Displays the URL `13.233.92.247/reset_password/admin.php?otp=314`.
- Header:** The page title is "Lifestyle Store". The header also includes navigation links: My Cart, My Profile, My Orders, Blog, Forum, and Logout.
- Content Area:** The main content is titled "Enter New Admin Password". It contains two input fields: "New password" and "Confirm password".
- Buttons:** A large orange button labeled "Reset Password" is positioned below the input fields.

# 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/Testing_Multiple_Factors_Authentication_(OWASP-AT-009))

[https://www.owasp.org/index.php/Blocking\\_Brute\\_Force\\_Attacks](https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks)

# 5. CSRF

Unauthorised Access to Customer Details (Critical)	<p>The below mentioned login page allows you to change password without verification and view details of other customers (CSRF).</p> <p>Affected URL :</p> <ul style="list-style-type: none"><li>• <a href="http://13.235.16.36/profile/change_password.php">http://13.235.16.36/profile/change_password.php</a></li></ul> <p>Affected Parameters :</p> <ul style="list-style-type: none"><li>• Update button (POST parameter) We can change the password.</li></ul> <p>Affected URL :</p> <ul style="list-style-type: none"><li>• <a href="http://13.235.16.36/cart/cart.php">http://13.235.16.36/cart/cart.php</a></li></ul> <p>Affected Parameters :</p> <ul style="list-style-type: none"><li>• Remove option (POST parameter)</li></ul> <p>Affected URL :</p> <ul style="list-style-type: none"><li>• <a href="http://13.235.16.36/cart/cart.php">http://13.235.16.36/cart/cart.php</a></li></ul> <p>Affected Parameters :</p> <ul style="list-style-type: none"><li>• Confirm order option (POST parameter)</li></ul>

# Observation

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

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

- Address Bar:** Displays the URL `13.235.16.36/profile/change_password.php`.
- Header Bar:** Contains the text "Lifestyle Store" on the left and navigation links "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout" on the right.
- Content Area:** A "Change Password" form is displayed. It consists of two input fields, both containing seven dots (".....") to represent masked password entries. Below the inputs is a large orange "UPDATE" button.

# Observation

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

The screenshot shows a web browser interface. At the top, there is a navigation bar with icons for back, forward, and search, followed by the URL "file:///C:/Users/91768/OneDrive/Desktop/hacking/hackcsrf.html". Below the navigation bar, there are two input fields labeled "New Password" and "Confirm Password", and a button labeled "Update".

The main content area displays a "Change Password" form. The URL in the address bar is "13.235.16.36/profile/change\_password.php". The page header includes links for "Lifestyle Store", "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout".

The "Change Password" form has two input fields, both containing "....." as placeholder text. Below the inputs is a red "UPDATE" button.

# 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/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="
9              6965db15acabf308e74fa61bde40c623856201cbfe80ff1f28178fa5f13b28f3">
10         <input type="text" name="password" value="12345">
11         <input type="text" name="password_confirm" value="12345">
12         <input type="submit" value="Send">
13     </form>
14 </body>
15 </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 user

# 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)	<p>Below mentioned parameters are vulnerable to reflected XSS</p> <p><b>Affected URL :</b> <a href="http://13.233.92.247/profile/16/edit/">http://13.233.92.247/profile/16/edit/</a></p> <p><b>Affected Parameters :</b> address(POST parameters)</p> <p><b>Payload:</b> &lt;script&gt;alert(1)&lt;/script&gt;</p>
--	--

# Observation

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

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

- Address Bar:** Displays the URL `13.233.92.247/profile/16/edit/`.
- Header:** The page title is "Lifestyle Store". The navigation menu includes links for "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout".
- Content Area:** The main content is titled "My Profile". It contains five input fields:
  - First Name: "lol"
  - Email: "lol@gmail.com"
  - Last Name: "lol"
  - Phone Number: "7680083254"
  - A text input field containing the script: "<script>alert('1')</script>"
- Buttons:** Two prominent buttons are present:
  - A red button labeled "UPLOAD PROFILE PICTURE".
  - A large orange button labeled "UPDATE".

# POC

The screenshot shows a web browser window with the URL `13.233.92.247/profile/profile.php` in the address bar. The page title is "Lifestyle Store". The main content area displays a "My Profile" section with the following information:

lol  
lol@gmail.com

A modal dialog box is overlaid on the page, containing the IP address `13.233.92.247` and a large number `1`. There are also "OK" and "Cancel" buttons at the bottom right of the dialog.

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

[https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)

[https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp)

# 7. Stored Cross Site Scripting (XSS)

Stored Cross Site Scripting (Severe)	<p>Below mentioned parameters are vulnerable to reflected XSS</p> <p><b>Affected URL :</b></p> <ul style="list-style-type: none"><li>• <a href="http://13.233.92.247/profile/16/edit/">http://13.233.92.247/profile/16/edit/</a></li></ul> <p><b>Affected Parameters :</b></p> <ul style="list-style-type: none"><li>• POST button under Customer Review (POST parameters)</li></ul> <p><b>Payloads:</b></p> <ul style="list-style-type: none"><li>• &lt;script&gt;alert('Hacked')&lt;/script&gt;</li><li>• &lt;h1&gt;hey&lt;/h1&gt;</li></ul>

# Observation

Now try entering the payload in review box

13.233.92.247/products/details.php?p\_id=16

Lifestyle Store      My Cart      My Profile      My Orders      Blog      Forum      Logout



All Products T Shirt

## Basic Blue T Shirt

Basic Blue T Shirt for everyday use.

Seller Info

Brand Website

INR 145/-

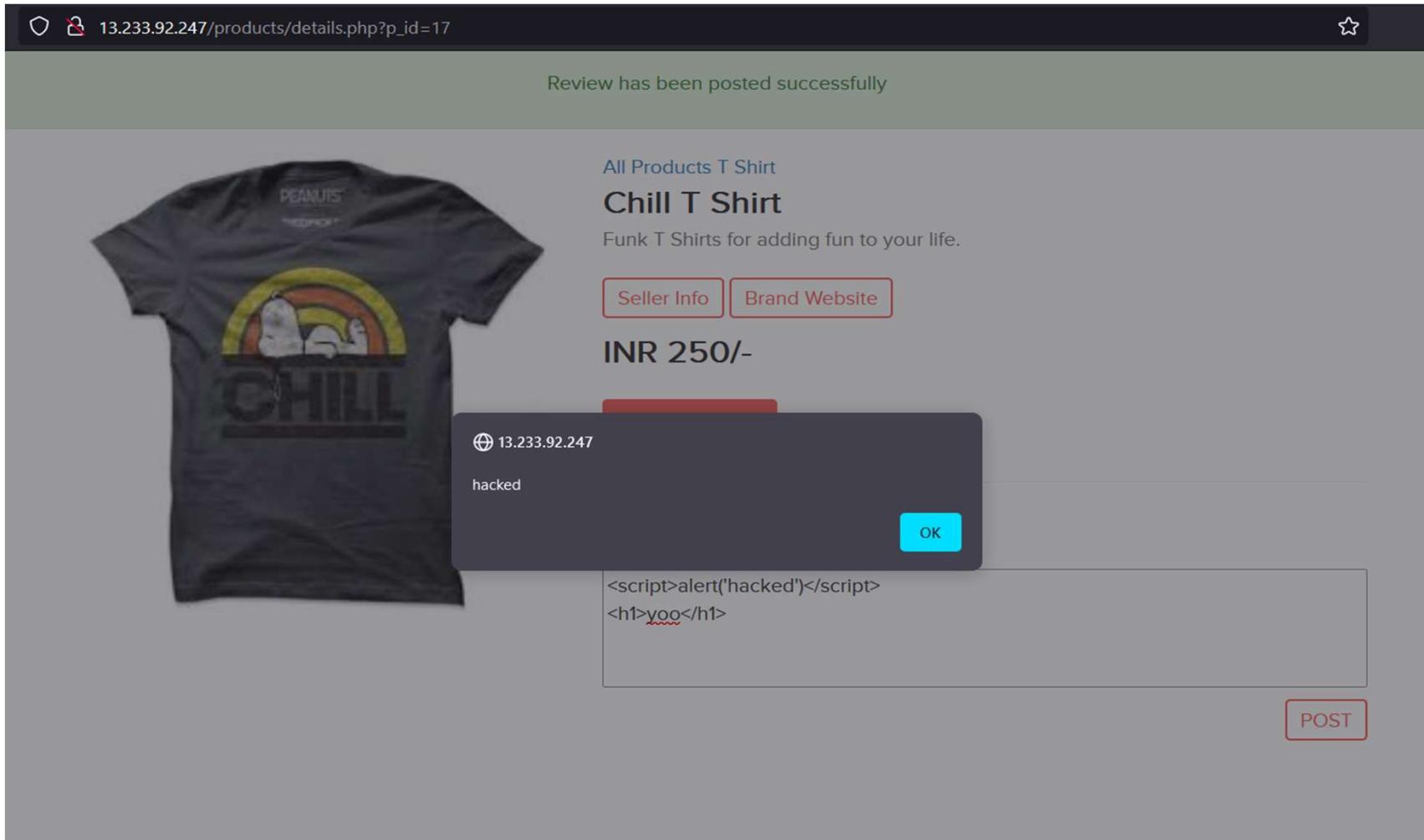
Add To cart

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

[https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)

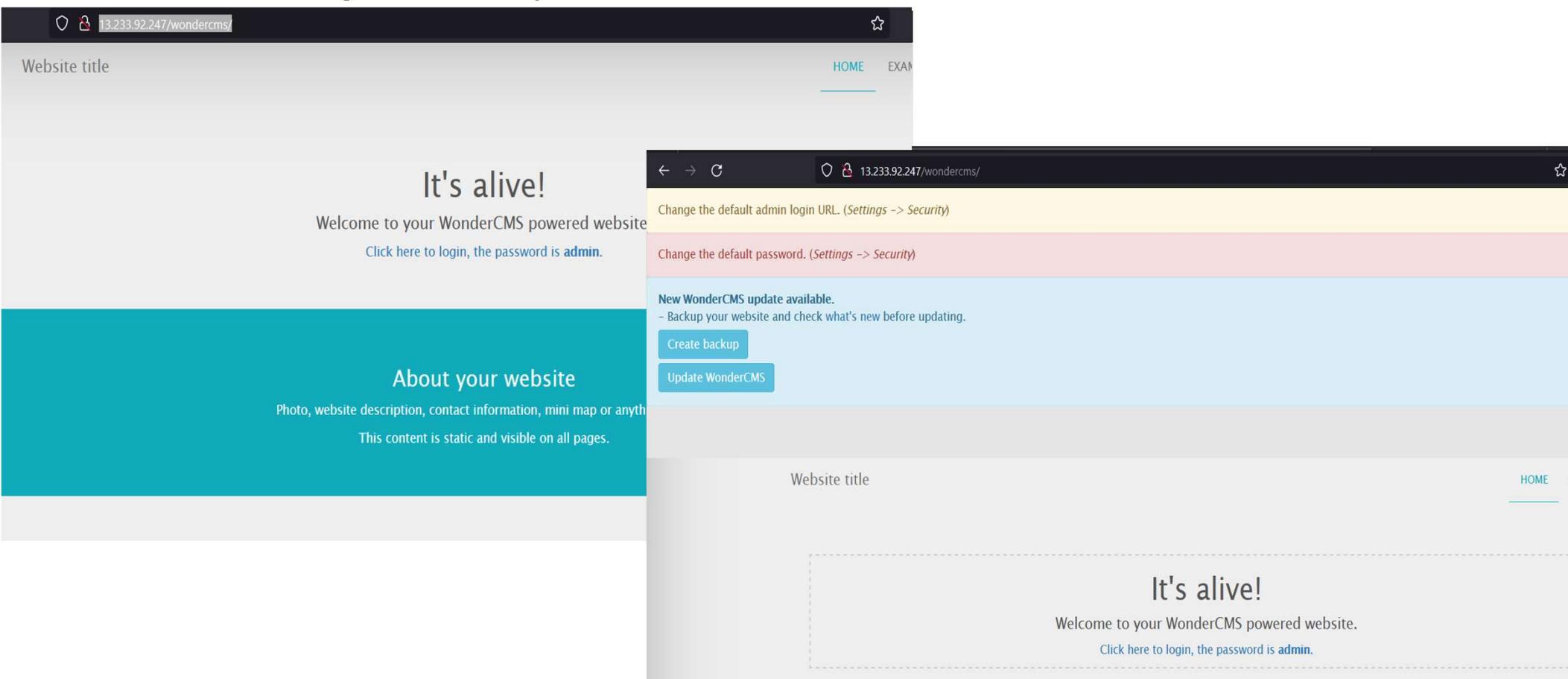
[https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp)

# 8. COMMON PASSWORD

Common password (Severe)	<p>Below mentioned url has weak and very common password</p> <p><b>Affected URL :</b></p> <ul style="list-style-type: none"><li>• <a href="http://13.233.92.247/wondercms/">http://13.233.92.247/wondercms/</a></li></ul>
-----------------------------	---

# 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/>

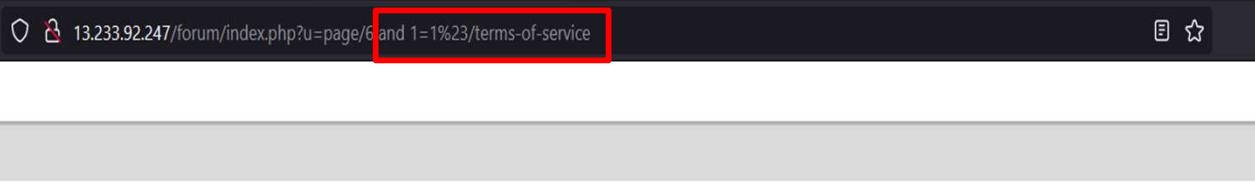
[https://www.owasp.org/index.php/Testing\\_for\\_Weak\\_password\\_policy\\_\(OTG-AUTHN-007\)](https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007))

# 9. Component with known vulnerability

Component with known vulnerability  (Severe)	<ul style="list-style-type: none"><li>• 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.</li><li>• WonderCMS</li><li>• Codoforum (Powered by codologic)</li></ul>

# Observation

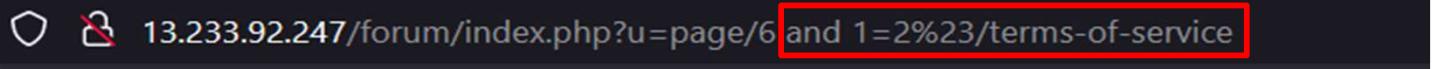
Codologic Vulnerability:- Now you can see that they have blind sql injection vulnerability



## 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



You agree not to modify, copy, distribute, transmit, display, or otherwise use the information obtained from or through the Site.

### Third-party Sites.

The Site may contain links to other websites maintained by third parties. The Site is not associated with, and the party maintaining such websites is not affiliated with, the party by Codologic.

You do not have enough permissions to view this page!

# POC

Codologic Vulnerability,  
It has multiple sql injection vulnerability,  
Check 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)

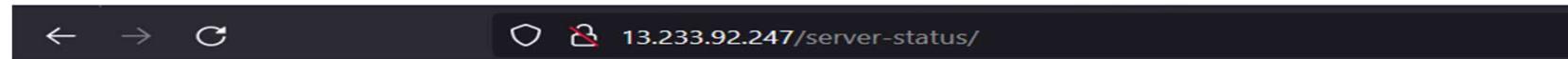
<https://www.exploit-db.com/exploits/37820>

<https://securitywarrior9.blogspot.com/2018/01/vulnerability-in-wonder-cms-leading-to.html>

# 10. Server misconfiguration

Server misconfiguration (Moderate)	<p>Below mentioned url will show you the server related info</p> <p><b>URL</b></p> <p><a href="http://13.233.92.247/server-status/">http://13.233.92.247/server-status/</a></p> <p><a href="http://13.233.92.247/server-info/">http://13.233.92.247/server-info/</a></p>
--	--

# 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

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

### 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

# 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>

# 11. Unauthorized access to user details(IDOR)

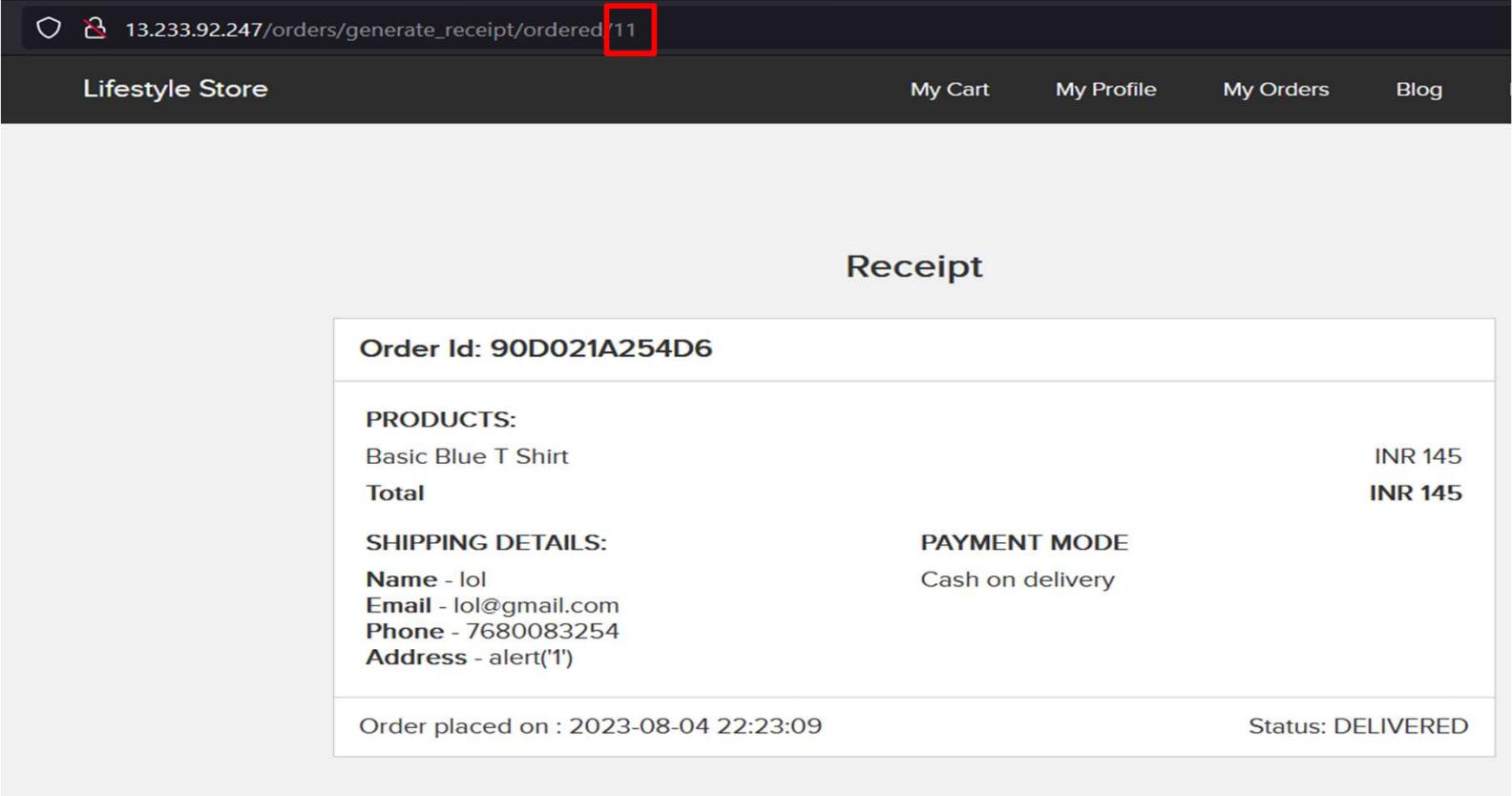
Unauthorized access to user details (Moderate)	<p>Below mentioned url will have vulnerability through which anyone can see the details of another user</p> <p><b>URL</b></p> <p><a href="http://13.233.92.247/orders/generate_receipt/ordered/11">http://13.233.92.247/orders/generate_receipt/ordered/11</a></p> <p>Affected parameter Ordered/11</p> <p><b>Payload</b></p> <p><a href="http://13.233.92.247/orders/generate_receipt/ordered/10">http://13.233.92.247/orders/generate_receipt/ordered/10</a></p>
--	--

# 11. Unauthorized access to user details(IDOR)

Unauthorized access to user details (Moderate)	<p>Below mentioned url will have vulnerability 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.</p> <p><b>URL'S effected:-</b></p> <p><a href="http://13.233.92.247/orders/orders.php?customer=14/">http://13.233.92.247/orders/orders.php?customer=14/</a> <a href="http://13.233.92.247/profile/16/edit/">http://13.233.92.247/profile/16/edit/</a> <a href="http://13.233.92.247/forum/index.php?u=/user/profile/4">http://13.233.92.247/forum/index.php?u=/user/profile/4</a></p>
---	---

# Observation

- When we change the payload we can see the receipts of other users or customers

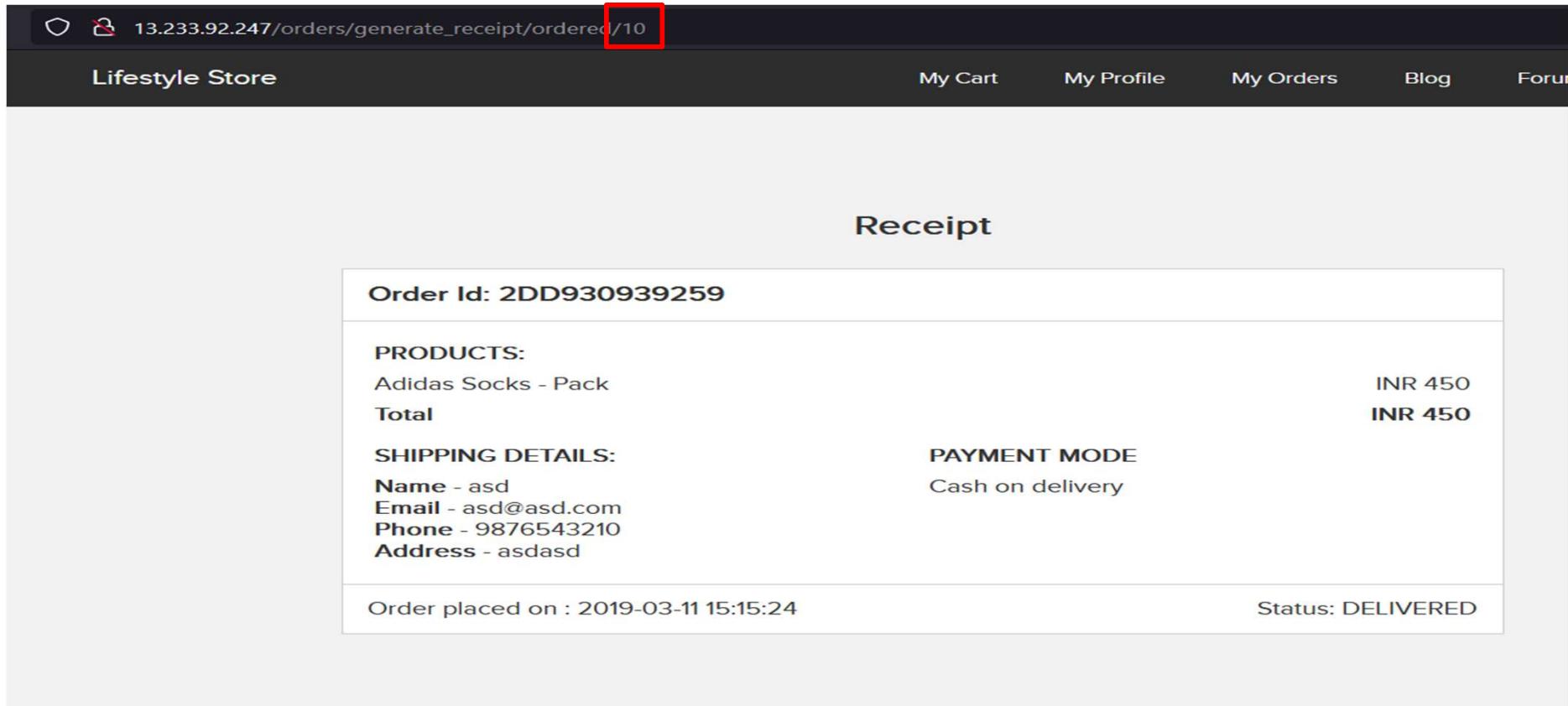


The screenshot shows a web browser window with the following details:

- URL:** 13.233.92.247/orders/generate\_receipt/ordered/11 (The path "/ordered/11" is highlighted with a red box).
- Page Title:** Lifestyle Store
- Navigation:** My Cart, My Profile, My Orders, Blog
- Section:** Receipt
- Order ID:** 90D021A254D6
- Products:**
  - Basic Blue T Shirt INR 145
  - Total INR 145
- Shipping Details:**
  - Name - lol
  - Email - lol@gmail.com
  - Phone - 7680083254
  - Address - alert('1')
- Payment Mode:** Cash on delivery
- Order Status:** Status: DELIVERED
- Order Placed:** Order placed on : 2023-08-04 22:23:09

# POC

- Here you can clearly see the receipt of another user



The screenshot shows a web browser window with a dark theme. The URL bar at the top contains the address `13.233.92.247/orders/generate_receipt/ordered/10`, which is highlighted with a red box. The main content area displays a receipt titled "Receipt". The receipt includes the following details:

Order Id: 2DD930939259	
<b>PRODUCTS:</b> Adidas Socks - Pack <b>Total</b>	INR 450 <b>INR 450</b>
<b>SHIPPING DETAILS:</b> Name - asd Email - asd@asd.com Phone - 9876543210 Address - asdasd	<b>PAYMENT MODE</b> 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 in a massive information leakage.

# Recommendation

Take the following precautions:

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

# References

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

# 12 Directory Listings

Directory listings  
(Moderate)

Below mentioned urls disclose server information. Affected URL :

<http://13.233.92.247/phpinfo.php>

<http://13.233.92.247/robots.txt>

<http://13.233.92.247/composer.lock>

<http://13.233.92.247/composer.json>

<http://13.233.92.247/userlist.txt>

# Observation

User-Agent: \*

Disallow: /static/images/

Disallow: /ovidentiaCMS

13.233.92.247/robots.txt

13.233.92.247/phpinfo.php

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

System: Linux ip-172-26-11-97 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-mysqli.ini, /etc/php/5.6/fpm/conf.d/20-mysqlnd.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: 2013106

PHP Extension: 20131226

Zend Extension: 220131226

Zend Extension Build: API20131226,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

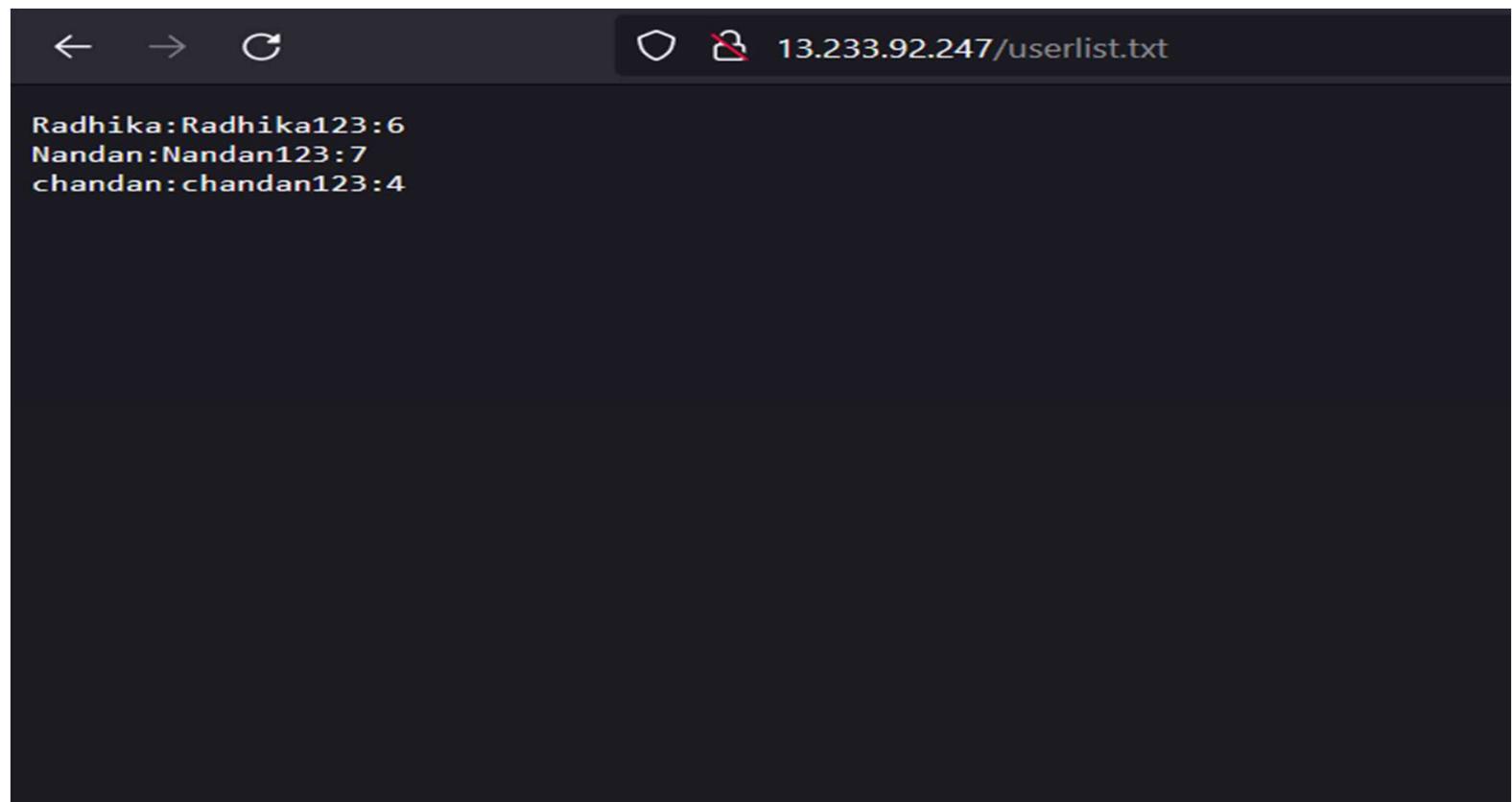
IPv6 Support: enabled

DTrace Support: enabled

Registered PHP Streams: https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

# 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



# 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/>

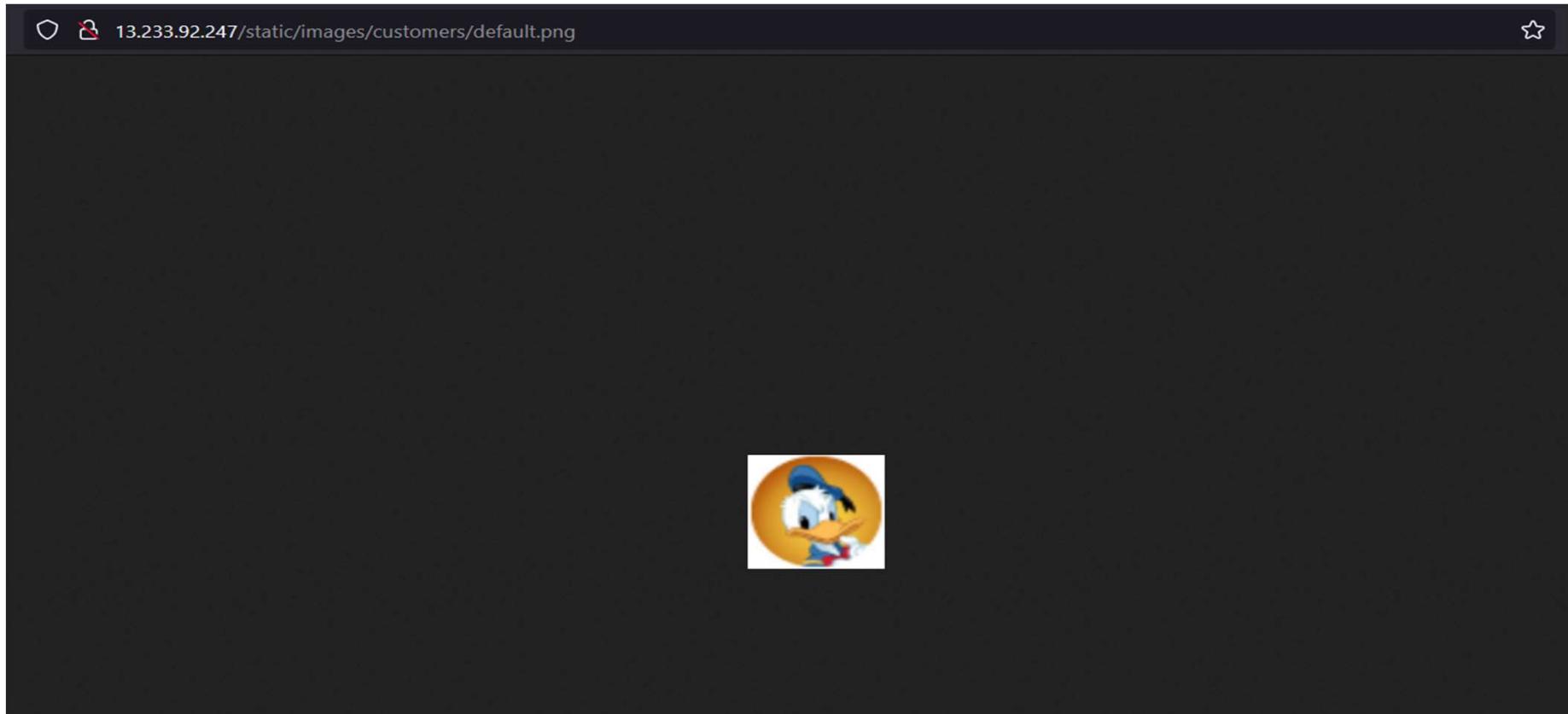
<https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/information-disclosure-phpinfo/>

# 13. Personal Information Leakage

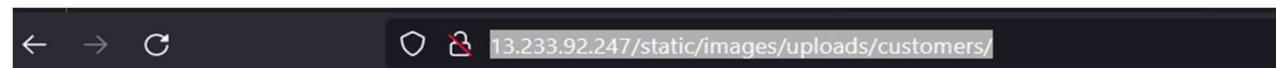
Personal Information Leakage (Low)	<p>Below mentioned urls disclose personal information</p> <p><b>Affected URL :</b></p> <p><a href="http://13.233.92.247/static/images/customers/default.png">http://13.233.92.247/static/images/customers/default.png</a></p> <p><a href="http://13.233.92.247/products/details.php?p%20id=2">http://13.233.92.247/products/details.php?p%20id=2</a></p>
--	--

# Observation

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



# POC



- 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...



## Index of /static/images/uploads/

..		
customers/		
products/		
card.png		

# 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 encryption 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 sid e validation by pass (Low)	<p>In below mentioned urls , we can easily bypass client side and server side validation</p> <p><b>Affected URL :</b></p> <ul style="list-style-type: none"><li>• <a href="http://13.233.92.247/profile/16/edit/">http://13.233.92.247/profile/16/edit/</a></li></ul> <p><b>Affected parameter :</b></p> <ul style="list-style-type: none"><li>• Contact Number (POST Parameter)</li></ul> <p>Payload used:</p> <ul style="list-style-type: none"><li>• 0000000000</li></ul>
--	--

# Observation

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

The screenshot shows a web browser interface for a "Lifestyle Store". The URL in the address bar is 13.233.92.247/profile/16/edit/. The page title is "My Profile". There are five input fields for profile information. The first four fields contain the value "lol". The fifth field contains the value "7680083254", which has been modified from its original value of "1234567890". Below these fields is a red button labeled "UPLOAD PROFILE PICTURE". Below that is a large orange button labeled "UPDATE".

81

# POC

- Mobile number is saved as zero

The screenshot shows a web browser window with the URL `13.233.92.247/profile/profile.php`. The page title is "My Profile". The user information displayed is:

Username: lol  
Contact No.: 0  
Delivery Address: alert('1')

Below the user information are two buttons: "EDIT PROFILE" and "CHANGE PASSWORD", both highlighted with red borders.

# **Business Impact – Moderate**

The data provided by the user ,if incorrect, is not a very big issue but still must be checked for proper validating 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](https://www.owasp.org/index.php/Unvalidated_Input)

# 15. Default Messages

Default messages (Low)	<p>In below mentioned urls ,if add a specific payload it will show default messages</p> <p><b>Affected URL :</b></p> <ul style="list-style-type: none"><li>• <a href="http://13.233.92.247/?includelang=lang/en.php">http://13.233.92.247/?includelang=lang/en.php</a></li></ul> <p><b>Payload</b></p> <ul style="list-style-type: none"><li>• en.php' (GET Parameter)</li></ul>
------------------------	--

# Observation & POC

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

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

- Address Bar:** 13.233.92.247/?includelang=lang/en.php'
- Page Title:** Lifestyle Store
- Navigation:** Back, Forward, Stop, Refresh, Home, Favorites.
- Header:** My Cart, My Profile, My Orders, Blog, Forum, Logout
- Error Message:**

```
Warning: include(lang/en.php'): failed to open stream: No such file or directory in /home/trainee/uploads/code-64cd3a22a1244.php on line 1
Warning: include(): Failed opening 'lang/en.php' for inclusion (include_path='.:/usr/share/php') in /home/trainee/uploads/code-64cd3a22a1244.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](https://www.owasp.org/index.php/Improper_Error_Handling)

# 16. Open redirection

Open Redirection (Low)	<p>In below mentioned urls we can change the path of redirection</p> <p><b>Affected URL :</b></p> <ul style="list-style-type: none"><li>• <a href="http://13.233.92.247/?includelang=lang/fr.php">http://13.233.92.247/?includelang=lang/fr.php</a></li><li>• <a href="http://13.233.92.247/?includelang=lang/en.php">http://13.233.92.247/?includelang=lang/en.php</a></li></ul> <p><b>Payload:-</b></p> <ul style="list-style-type: none"><li>• <a href="http://13.233.92.247/?includelang=www.google.com?lang/en.php">http://13.233.92.247/?includelang=www.google.com?lang/en.php</a></li></ul>
------------------------------	---

# Observation

Here we made changes to the url according to the payload

Screenshot of the Burp Suite Proxy tab showing a captured request. The 'Intercept' button is highlighted in blue, indicating it is active.

Request to `http://13.233.92.247:80`

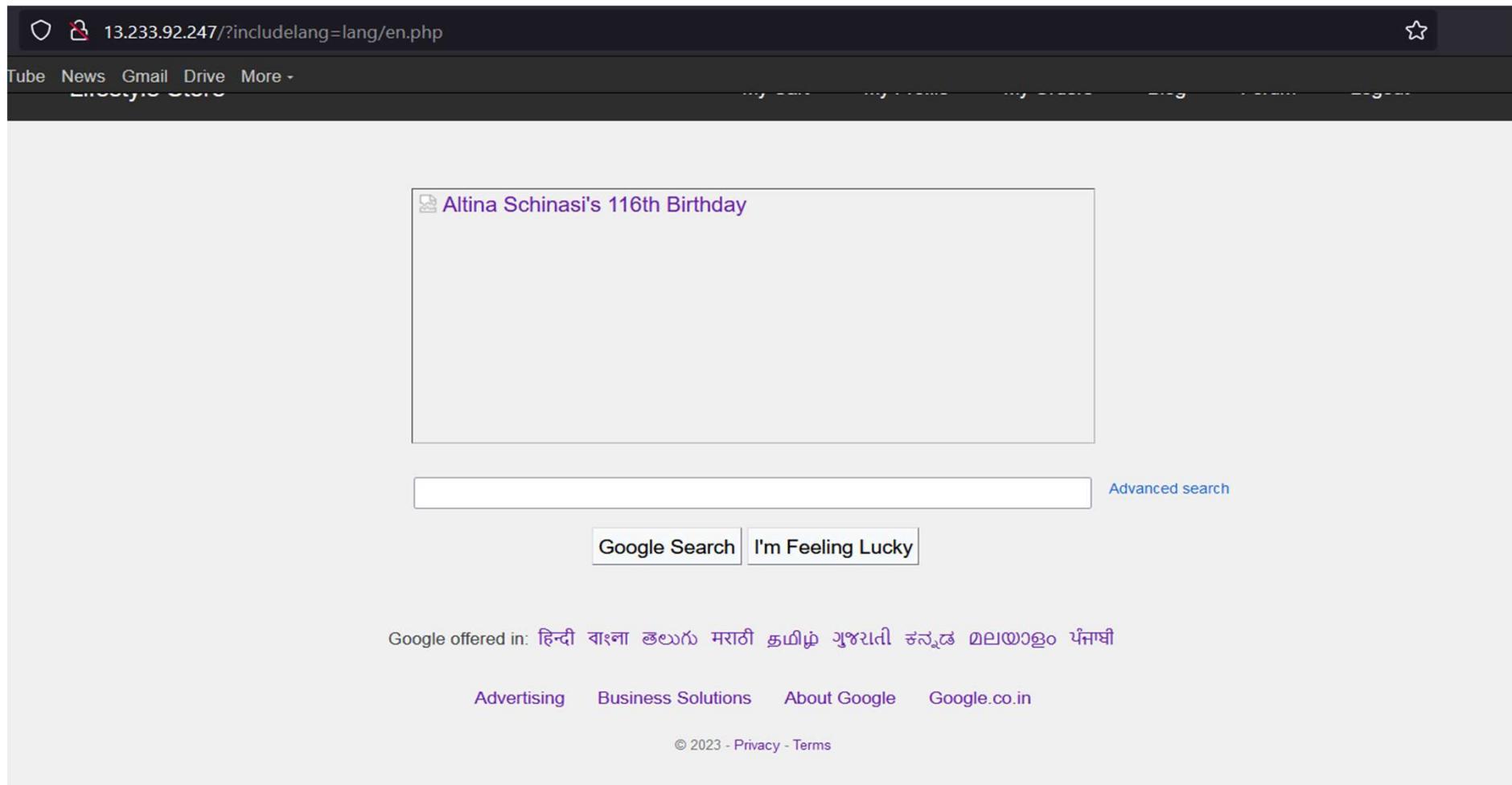
Action buttons: Forward, Drop, Intercept is on (highlighted), Action, Open browser

Pretty Raw Hex

```
1 GET /?includeLang=https://www.google.com?lang=en.php HTTP/1.1
2 Host: 13.233.92.247
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: key=eclzfwf8b8v; PHPSESSID=dc4hto4s7m5dp6o22o98pg9hf4; X-XSRF-TOKEN=
e005784242896ba08ebd89cdd21fbb8e07a3b0e75f5305579195fdb165e5c24a
9 Upgrade-Insecure-Requests: 1
10
11
```

# POC

- We are redirected to google



# **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 URLs such as javascript:

### **REFERENCES:-**

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

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

# THANK YOU

For any further clarifications/patch assistance, please contact: 7680083254,  
Mail:honeyharpreet94@gmail.com