

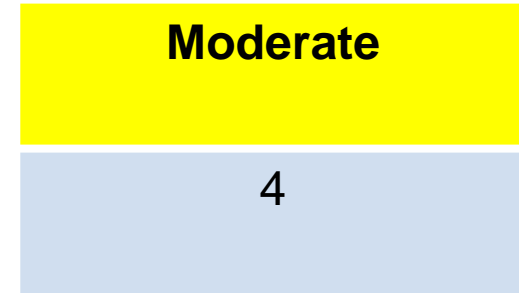
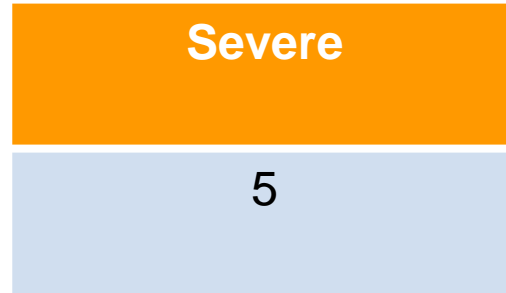
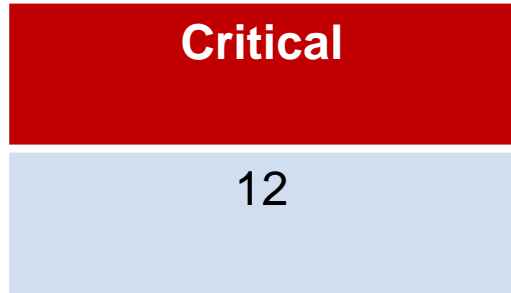
Hacking Environment Web Application

Detailed Developer Report

Security Status – Extremely Vulnerable

- Hacker can steal all records in website databases (SQLi)
- Hacker can take control of complete server including View, Add, Edit, Delete files and folders (Shell Upload)
- Hacker can change source code of application to host malware, phishing pages or even explicit content (Shell Upload)
- Hacker can inject client side code into applications and trick users by changing how page looks to steal information or spoil the name of website (XSS)
- Hacker can extract personal information of all customers using userid (IDOR)

Vulnerability Statistics



Vulnerabilities:

| No | Severity | Vulnerability |
|----|----------|---|
| 1 | Critical | SQL Injection |
| 2 | Moderate | Reflected XSS |
| 3 | Critical | Stored XSS |
| 4 | Severe | Insecure Direct Object Reference (IDOR) |
| 5 | Critical | Rate Limiting Issues |
| 6 | Critical | Insecure File Uploads |
| 7 | Low | Client Side Filter Bypass |
| 8 | Low | Server Misconfigurations |
| 9 | Severe | Components with known vulnerabilities |

Vulnerabilities:

| No | Severity | Vulnerability |
|----|----------|-----------------------------------|
| 10 | Critical | Weak passwords |
| 11 | Low | Default files and pages |
| 12 | Critical | File Inclusion |
| 13 | Severe | PII Leakage |
| 14 | Severe | Open Redirection |
| 15 | Severe | Bruteforce Exploitation |
| 16 | Severe | Forced Browsing |
| 17 | Moderate | Command Execution |
| 18 | Low | Descriptive Error Messages |
| 19 | Critical | Cross Site Request Forgery (CSRF) |

1. SQL Injection

SQL Injection (Critical)

Below mentioned URL in the categorized products page is vulnerable to SQL injection attack

Affected URL :

`http://3.6.39.180/products.php?cat=1'`

Affected Parameters :

- house (GET parameter)

Payload:

- `cat=1'`

1. SQL Injection

SQL Injection
(Critical)

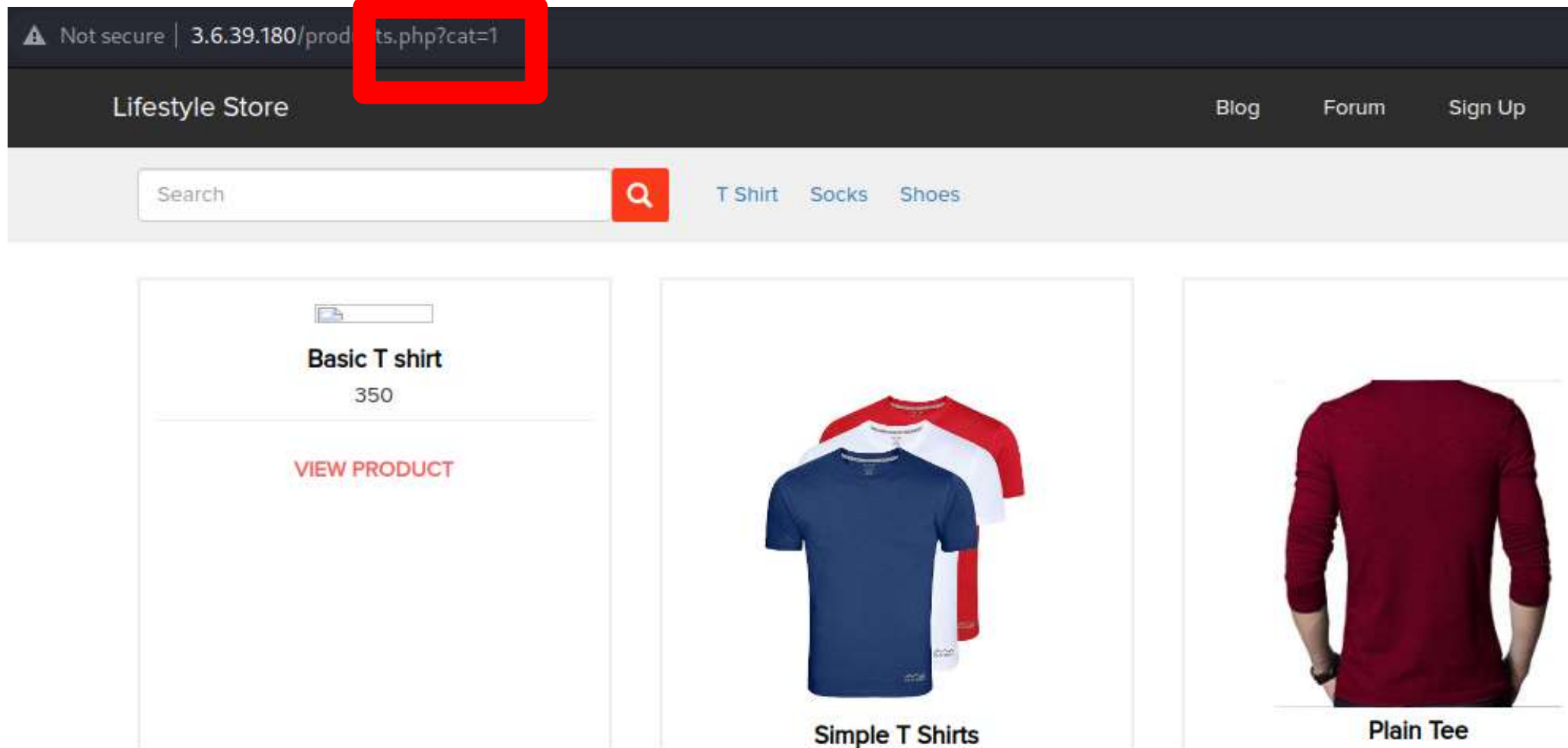
Here are other similar SQLi in the application

Affected URL :

- http://3.6.39.180/products.php?cat=2'
- http://3.6.39.180/products.php?cat=3'

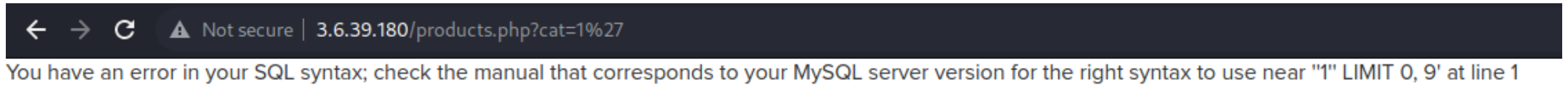
Observation:

- Navigate to Products page where you will see a list of all the products. Click on a category to show only those items. Notice the GET parameter “cat=1” in the URL:



Observation:

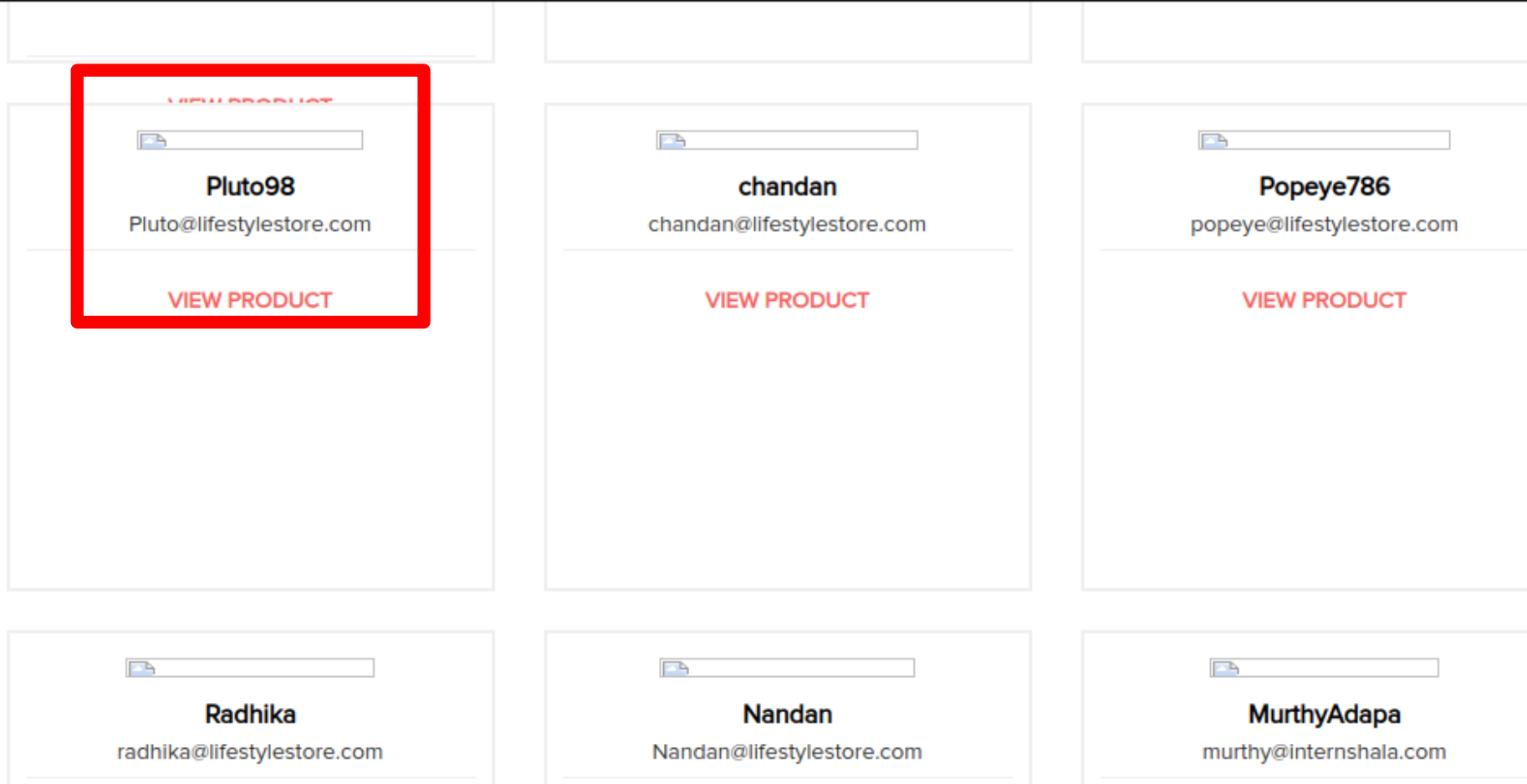
- We apply single quote in cat parameter: `products.php?cat=1'` and we get complete MySQL error:



Proof of Concept (PoC):

- Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the username and email of users but the attacker can extract more info.
- `cat=1' union select 1,user_name,3,password,5,6,7 from users --+`

3.6.39.180/products.php?cat=1' union select 1,user_name,3,email,5,6,7 from users--+



Business Impact – Extremely High

Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.

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.

Recommendation

Take the following precautions to avoid exploitation of SQL injections:

- Prepared Statements: Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query
- Use POST method instead of GET to prevent SQL injection via URLs
- Do not allow input of special characters like quotes,hyphens,plus sign,Brackets,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

References

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

2. Reflected Cross Site Scripting (XSS)

Reflected Cross
Site Scripting
(moderate)

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- `http://3.6.39.180/search/search.php?q=`

Affected Parameters :

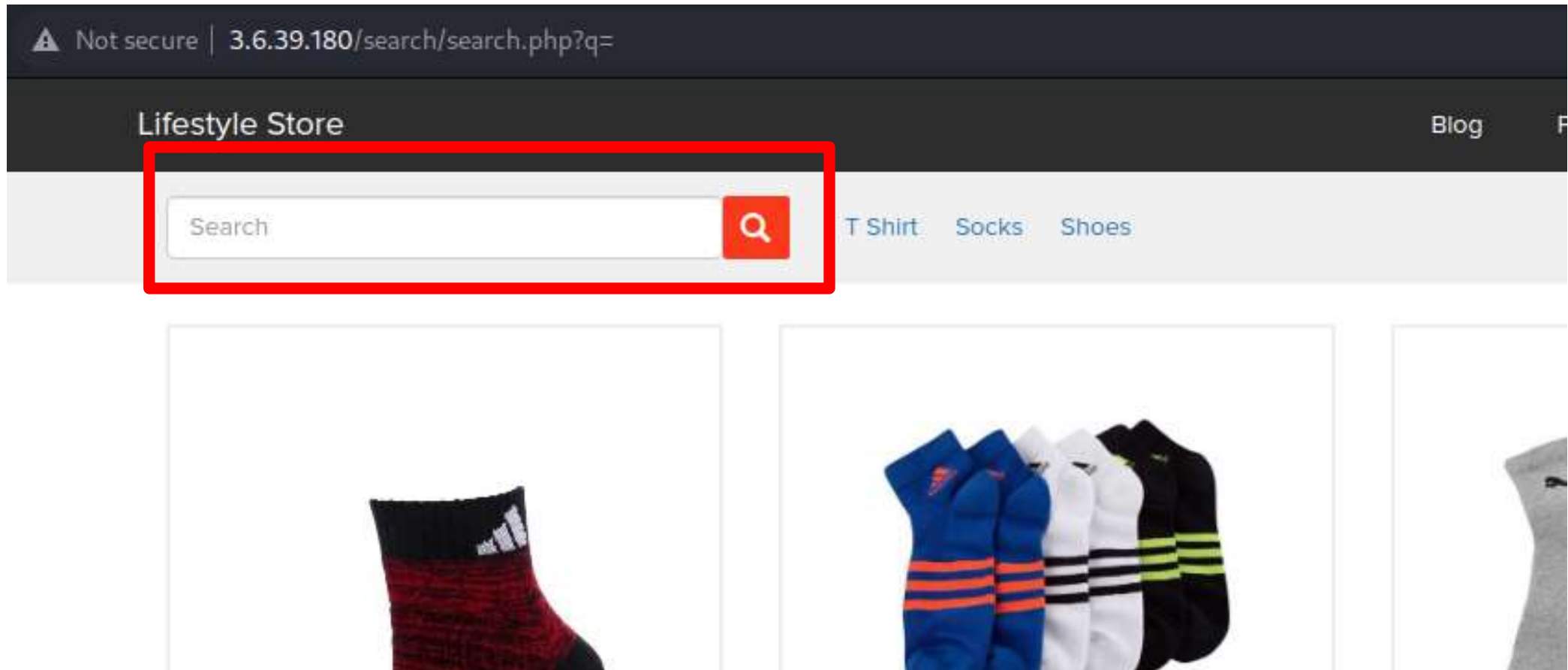
- `q` (GET parameter)

Payload:

- `"><script>alert(1)</script>`

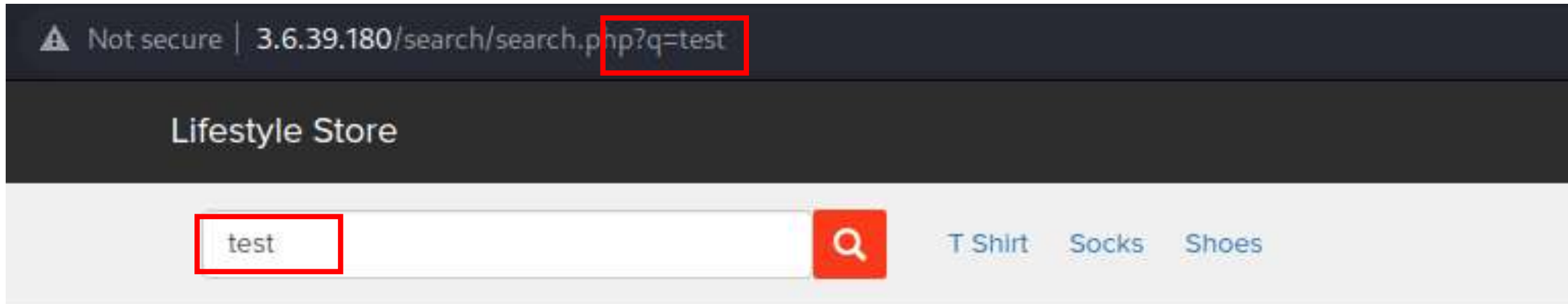
Observation:

Navigate to the products page. You will see a search field to enter some text. This field is Vulnerable to XSS. The URL has a GET parameter 'q'. You will only see this when you search for something. In this example, I have left the field blank and pressed the search button.



Observation:

Enter any text and click the button, you will see it reflected in the next page and value will be in GET parameter **q**



Not secure | 3.6.39.180/search/search.php?q=test

Lifestyle Store

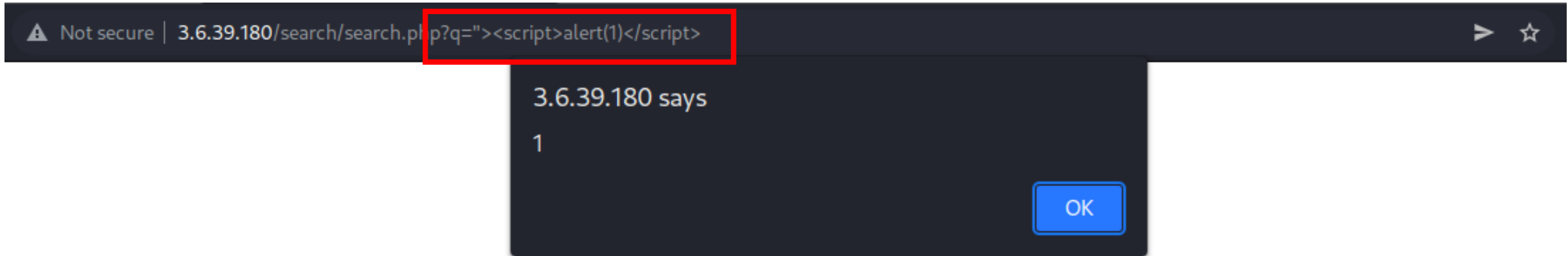
[T Shirt](#)[Socks](#)[Shoes](#)

Proof Of Concept (PoC):

Put the payload instead of 'test': `"><script>alert(1)</script>`

You can either enter it in the search field or directly in the URL.

As you can see we executed custom JS causing popup



Business Impact – Moderate

As attacker can inject arbitrary HTML CSS and JavaScript 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.

The attacker does not gain access to the server directly and can only use this attack on individuals so it does not effect each and every user like the stored XSS would.

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
- Block reserved keywords like 'script' that are used in HTML or JS
- At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.
- To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.

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

3. Stored Cross Site Scripting (XSS)

Stored Cross Site Scripting (Critical)

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- `http://3.6.39.180/products/details.php?p_id=6`

Affected Parameters :

- Reviews section.

Payload:

- `<script>alert(1)</script>`


I've provided 2 videos named "Stored XSS 1.mp4" & "Stored XSS 2.mp4" for better understanding.

Observation:

Navigate to the products page and open a product. You will see field to enter reviews. This field is Vulnerable to XSS. You can enter HTML or JavaScript code directly into the reviews section and post it.

⚠ Not secure | 3.6.39.180/products/details.php?p_id=6

Lifestyle StoreMy CartMy ProfileMy OrdersBlogForumLogout



[All Products T Shirt](#)

Simple T Shirts

Use these t shirts for light summers.

Seller InfoBrand Website

INR 550/-

Add To cart

No reviews yet

POST

Observation:

Enter any HTML or Javascript code in the review section directly like shown in the picture below.



Add To cart

No reviews yet

`Click Me`

POST

Observation:

You can see that a hyperlink named 'Click Me' has been created. Clicking on it will redirect you to the assigned webpage.

Customer Reviews



user

[Click Me](#)

POST

Proof Of Concept (PoC):

Our injection works on every user who visits the URL. Here as an example, you can see that the hyperlink is still visible even though I've logged out.

Not secure | 3.6.39.180/products/details.php?p_id=6

Lifestyle Store Blog Forum Sign Up **Login ▾**

All Products T Shirt

Simple T Shirts


Use these t shirts for light summers.


[Seller Info](#) [Brand Website](#)

INR 550/-

[Login](#)

Customer Reviews

 **user**
[Click Me](#)

 **user**
k

Business Impact – Extremely High:

As attacker can inject arbitrary HTML, CSS and JavaScript via the Review box, any malicious scripts can be executed on the devices of every user that visits that URL. The data of the users can be compromised or the attacker can install malicious software like viruses, malware, spyware, ransomware or the attacker might as well render a device unusable by corrupting important files that are used by the operating system. This way, the malicious actors can cause huge loss to the company or might deface the company's public image.

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.
- Block reserved keywords like 'script' that are used in HTML or JS.
- Encode special characters before submitting it to the server or database.
- Never leave the server or database in admin privilege when not in use.
- Only use admin rights when necessary.
- Provide proper authorization to users & don't give any user permissions that they don't need.
- Encase the input provided by the user between special characters that are not easily guessable.

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

4. Insecure Direct Object Reference (IDOR)

Unauthorised
Access to
Customer
Details
(Critical)

The Edit Profile page suffers from an Insecure Direct Object Reference (IDOR) vulnerability that allows attacker get access to anyone's profile details.

Affected URL :

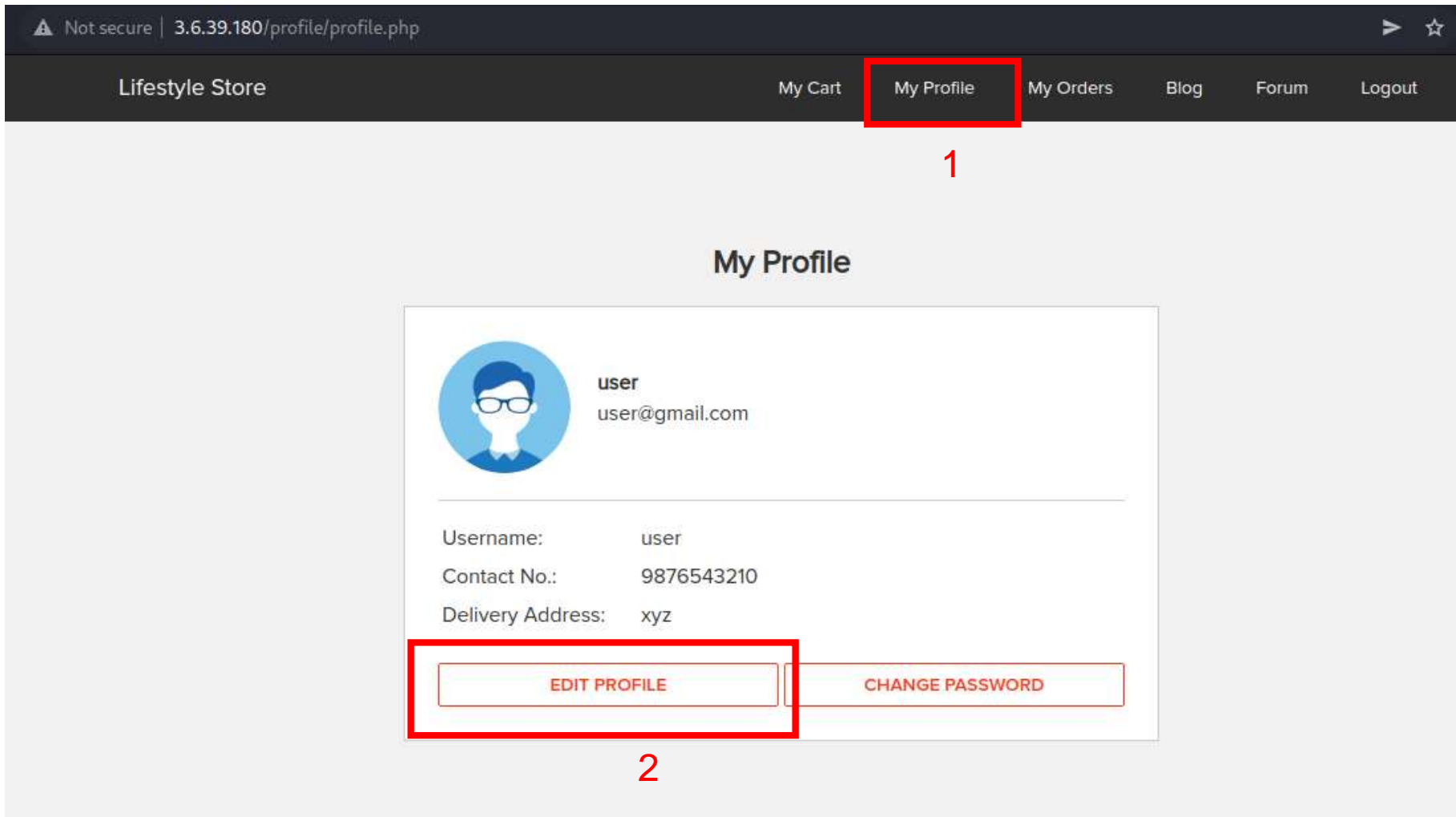
- <http://3.6.39.180/profile/16/edit/>

Affected Parameters :

- </profile/16/edit/>

Observation:

- Navigate to My profile and select Edit Profile from the bottom.



Observation:

- You can see 16 written in the URL. That is the customer ID for which the profile information is displayed. Let's try changing it.

Not secure | 3.6.39.180/profile/16/edit/

Lifestyle Store My Cart My Profile My Orders Blog Forum Logout

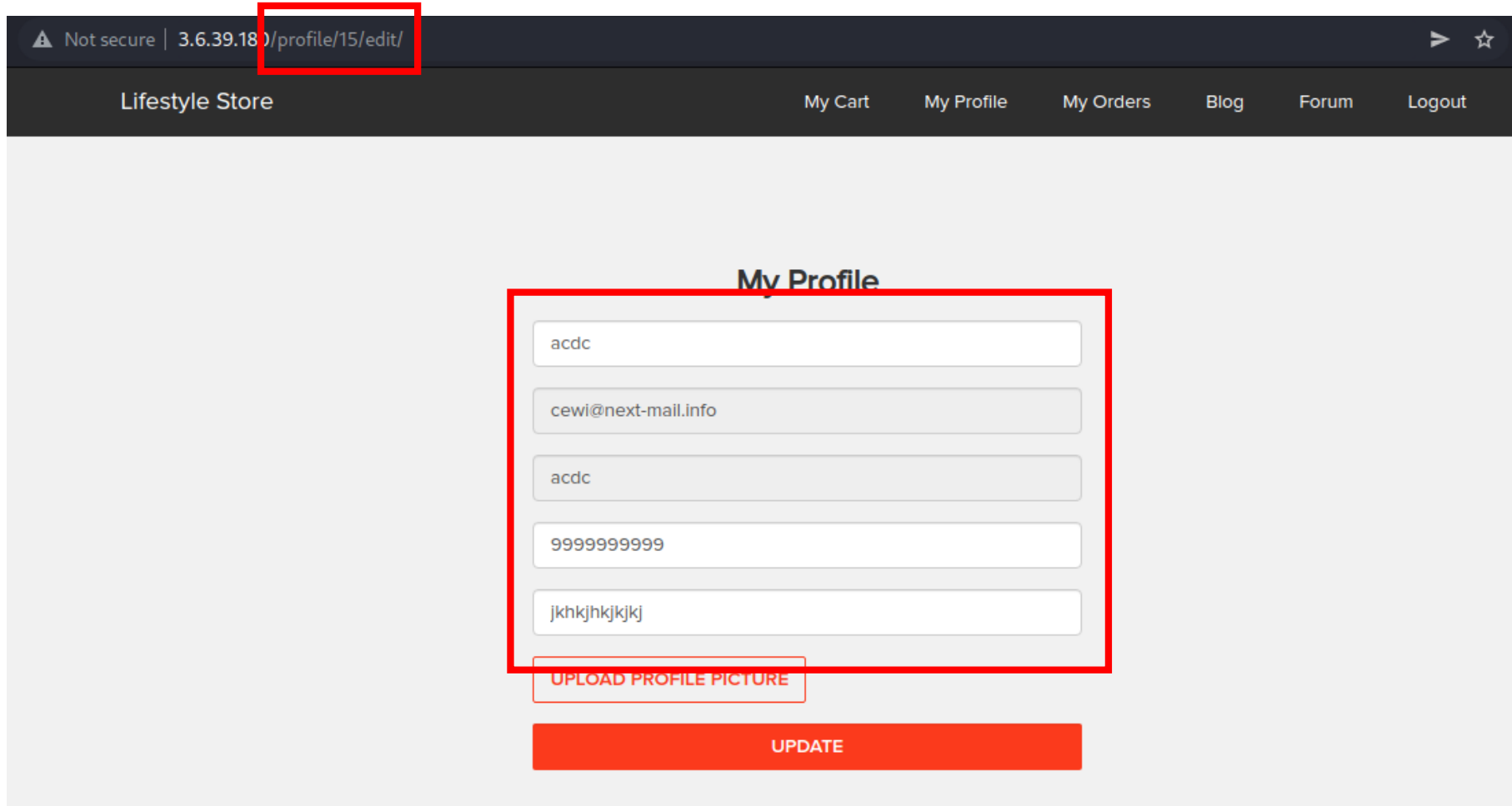
My Profile

UPLOAD PROFILE PICTURE

UPDATE

Proof of Concept (PoC):

- The information displayed also changes when we change the ID number in the URL. We are now viewing the information of another user in an unauthorized method.



The screenshot shows a web browser window with the address bar displaying '3.6.39.180/profile/15/edit/'. The page title is 'Lifestyle Store'. The navigation bar includes links for 'My Cart', 'My Profile', 'My Orders', 'Blog', 'Forum', and 'Logout'. The main content area is titled 'My Profile' and contains a form with five input fields, each containing a different string of characters. A red box highlights the entire form area, and another red box highlights the URL in the address bar. Below the form is a button labeled 'UPDATE PROFILE PICTURE' and a large red button labeled 'UPDATE'.

Not secure | 3.6.39.180/profile/15/edit/

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

My Profile

acdc

cewi@next-mail.info

acdc

9999999999

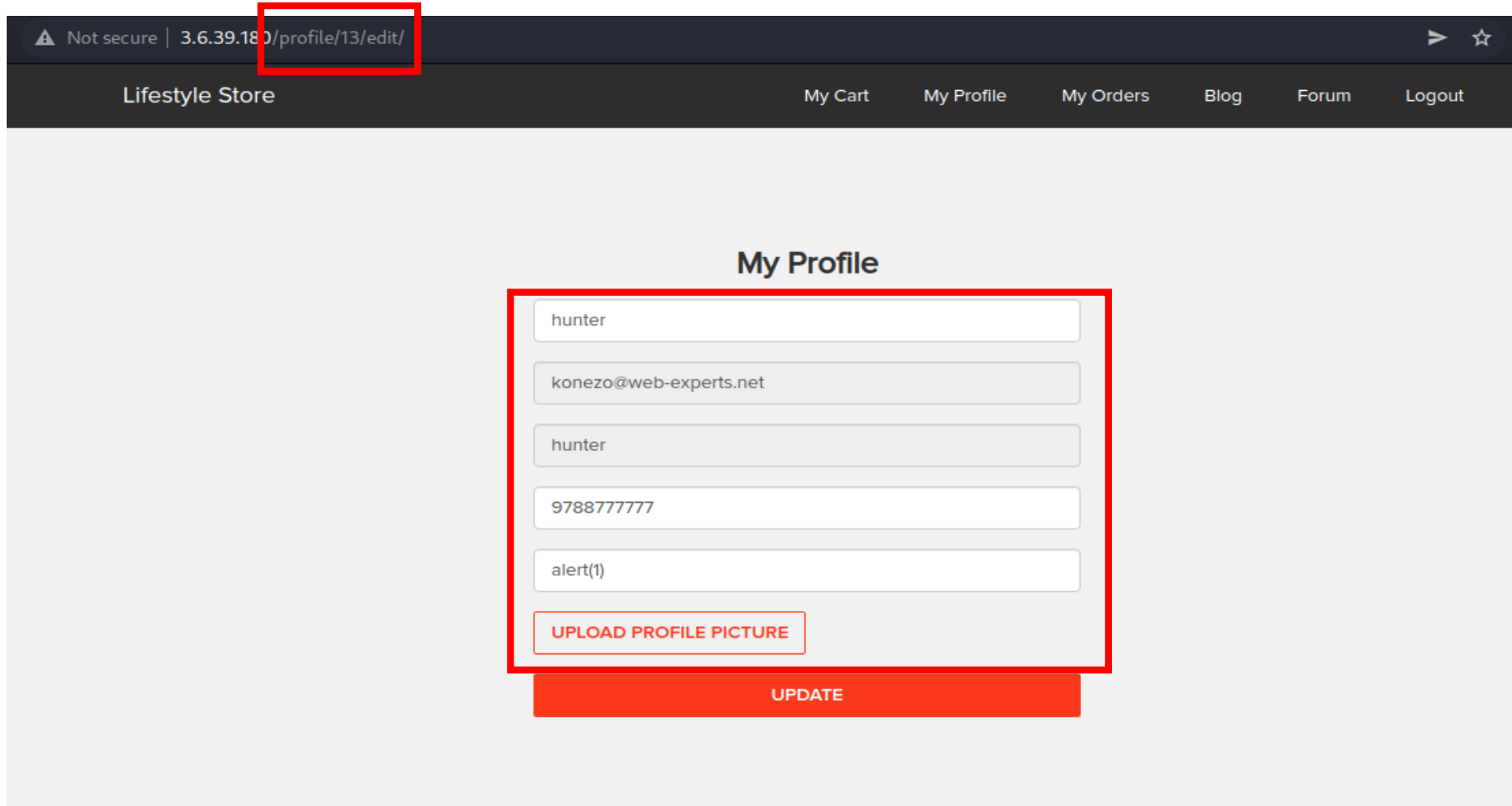
jkhkjkhkjkhkj

UPDATE PROFILE PICTURE

UPDATE

Proof of Concept (PoC):

- Look at another example. This information cannot be updated but it can still be used by the hacker to socially hack the user or to try bruteforcing by using the information provided.



The screenshot shows a web browser window with the address bar displaying '3.6.39.180/profile/13/edit/'. The browser's security indicator shows 'Not secure'. The website's header includes the text 'Lifestyle Store' and navigation links: 'My Cart', 'My Profile', 'My Orders', 'Blog', 'Forum', and 'Logout'. The main content area is titled 'My Profile' and contains a form with the following fields:

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

Below the form fields is a button labeled 'UPLOAD PROFILE PICTURE'. At the bottom of the form is a large red button labeled 'UPDATE'.

4. Insecure Direct Object Reference (IDOR)

Unauthorised
Access to
Order
Details
(Critical)

A similar issue is also found in the following URL

Affected URL :

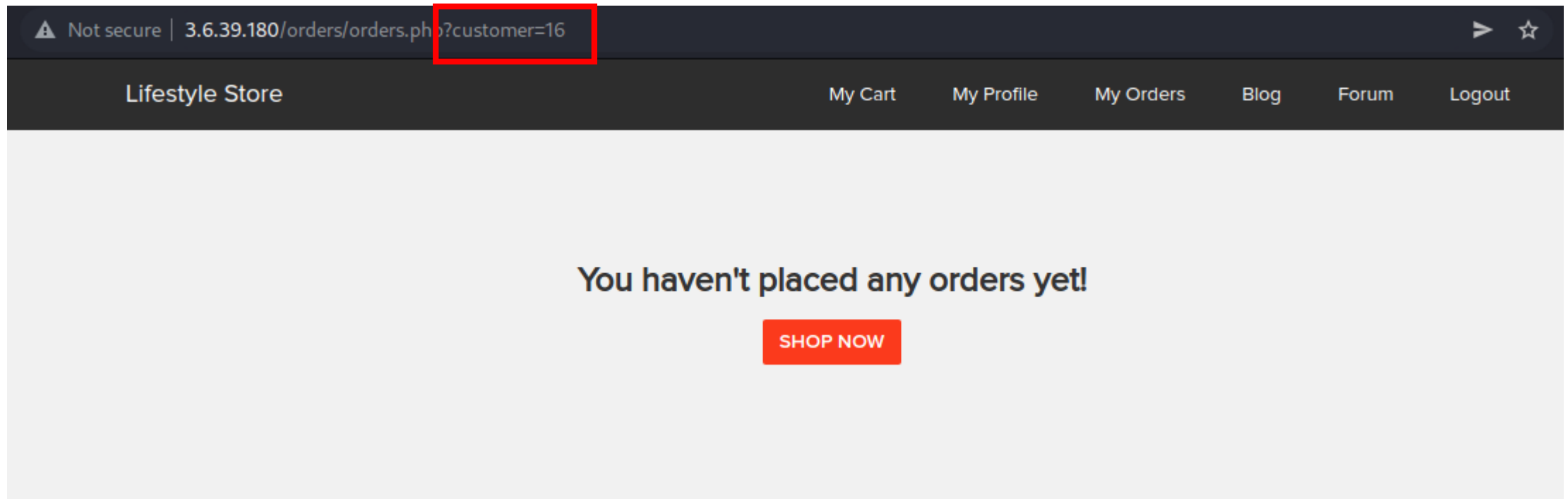
- <http://3.6.39.180/orders/orders.php?customer=16>

Affected Parameters :

- customer

Observation:

- Navigate to My Orders. You can see that I have not placed any orders yet. You can also see my Customer ID is the same as previous instance. Let's try changing the value of customer parameter.



Proof of Concept (PoC):

- On changing the value of the parameter 'customer' from 16 to 14, the website displays the order details of customer 14 without any authorization checks. This can be used by a hacker to find a user's personal information and execute social hacking.

Not secure | 3.6.39.180/orders/orders.php?customer=14

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

My Orders

| | |
|---------------------------------------|---------------------|
| Order Id: 2DD930939259 | |
| PRODUCTS: | |
| Adidas Socks - Pack | INR 450 |
| Total | INR 450 |
| SHIPPING DETAILS: | PAYMENT MODE |
| Name - asd | Cash on delivery |
| Email - asd@asd.com | |
| Phone - 9876543210 | |
| Address - asdasd | |
| Order placed on : 2019-03-11 15:15:24 | Status: DELIVERED |

Business Impact – Extremely High

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

- Mobile Number
- Bill Number
- Billing Period
- Bill Amount and Breakdown

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

5. Rate Limiting Flaw

Account
Takeover by
exploiting
Rate Limiting
Flaw.
(Critical)

Admin account's password can be recovered by forcing the 3 digit OTP as there is no rate limiting function.

Affected URL :

- http://3.6.39.180/reset_password/admin.php

Observation:

- Navigate to the Admin Login page and press Forgot your password button to recover it.

The screenshot displays the Admin Login page for a website titled "Lifestyle Store". The browser's address bar indicates the URL is `3.6.39.180/login/admin.php`. The page features a dark navigation bar with links for "Blog", "Forum", "Sign Up", and "Login". A dropdown menu is open on the right, showing options for "Customer", "Seller", and "Admin", with "Admin" highlighted. The main content area is titled "Admin Login" and contains a login form with two input fields: "Username" and "Password". Below these fields is a prominent red "Login" button. Directly beneath the "Login" button is a link that reads "Forgot your password?". Both the "Login" button and the "Forgot your password?" link are highlighted with red rectangular boxes.

Observation:

- We can see that the OTP is just 3 digits long. We can find the right OTP by providing it with all possible values (000 – 999).

Not secure | 3.6.39.180/reset_password/admin.php

Lifestyle Store Blog Forum Sign Up Login ▾

Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

Reset Password

Observation:

- We have successfully cracked the OTP using Bruteforcing. The OTP is 798.

| Request ^ | Payload | Status | Error | Timeout | Length | Comment |
|-----------|---------|--------|--------------------------|--------------------------|--------|---------|
| 789 | 788 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 790 | 789 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 791 | 790 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 792 | 791 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 793 | 792 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 794 | 793 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 795 | 794 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 796 | 795 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 797 | 796 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 798 | 797 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 799 | 798 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4476 | |
| 800 | 799 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 801 | 800 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 802 | 801 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 803 | 802 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 804 | 803 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 805 | 804 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 806 | 805 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 807 | 806 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 808 | 807 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 809 | 808 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 810 | 809 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 811 | 810 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 812 | 811 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 813 | 812 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 814 | 813 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 815 | 814 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 816 | 815 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 817 | 816 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 818 | 817 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 819 | 818 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |
| 820 | 819 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 4380 | |

Proof of Concept (PoC):

- We have access to the admin dashboard. From here, we can run commands through the console and add products.

Lifestyle Store

My Cart

My Profile

My Orders

Blog

Forum

Logout

Admin Dashboard

CONSOLE

Add Product:

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

Business Impact – Extremely High

- A Malicious hacker can gain complete access to admin account just by Brute-Forcing due to rate limiting flaw as a hacker can attempt as many times as he wants , as there is no bounds in no of tries. This leads to complete compromise of personal user data of every customer.
- Once the attacker logs in as admin, then he can carry out actions on behalf of the victim(admin) which could lead to serious financial loss to him/her, like he can change the name, picture and even price of the products.

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 2-5 minutes.
- OTP should be at least 6 digit and alphanumeric for more security.

References:

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

6. Insecure File Uploads

Server
takeover by
exploiting
Insecure File
Uploads.
(Critical)

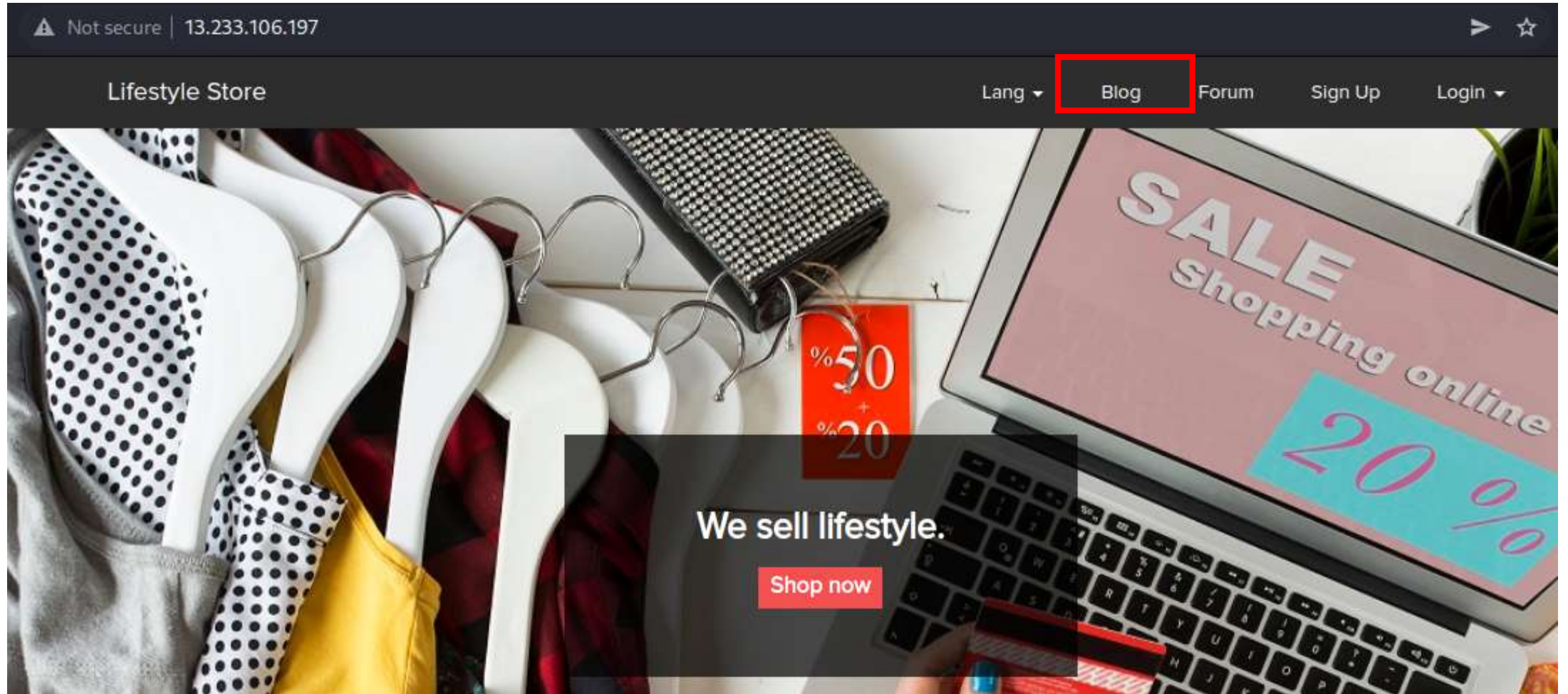
Shell upload is possible due to lack of functionality to check file extensions and access to admin account.

Affected URL :

- <http://13.233.106.197/wondercms/>

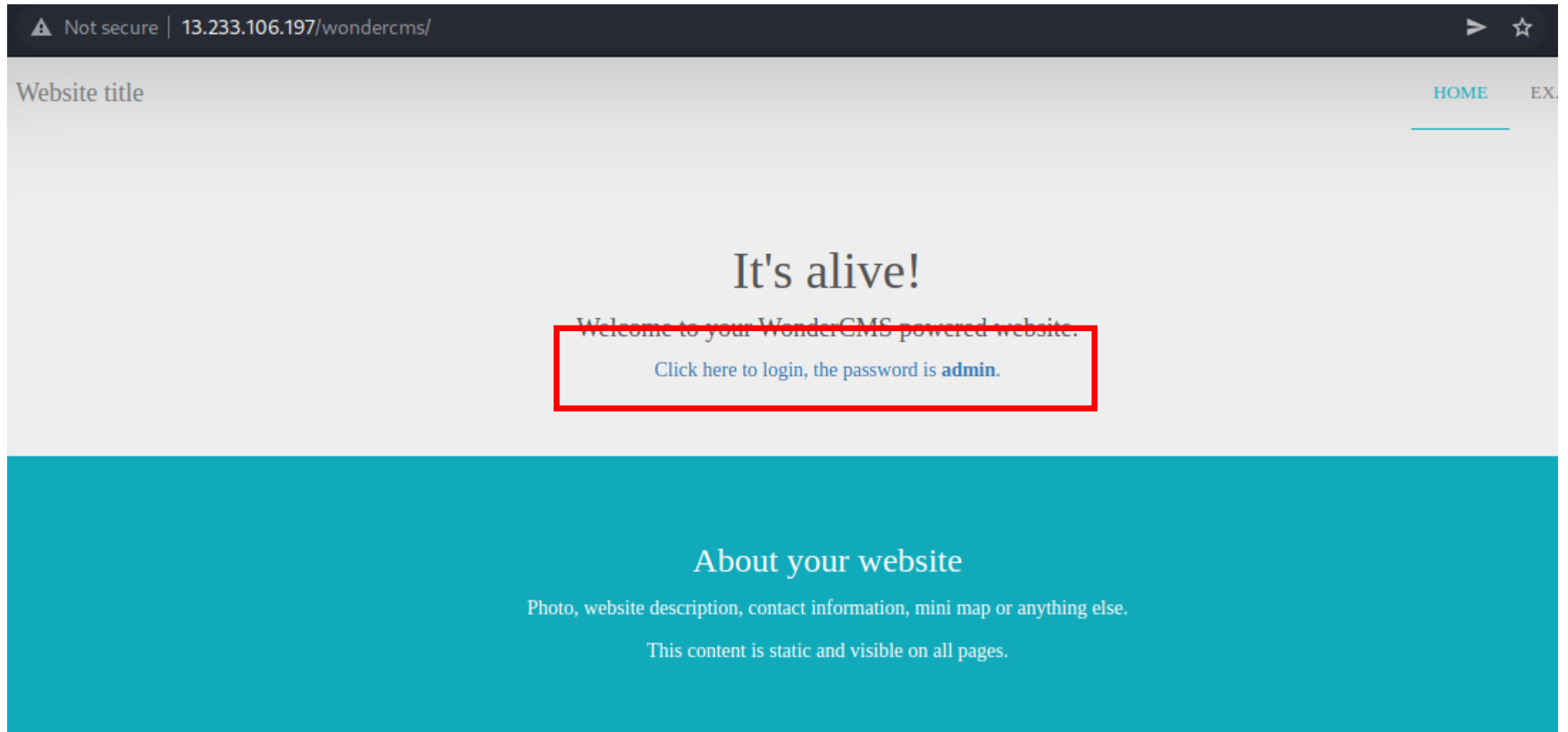
Observation:

- Navigate to the home page and click on Blog.



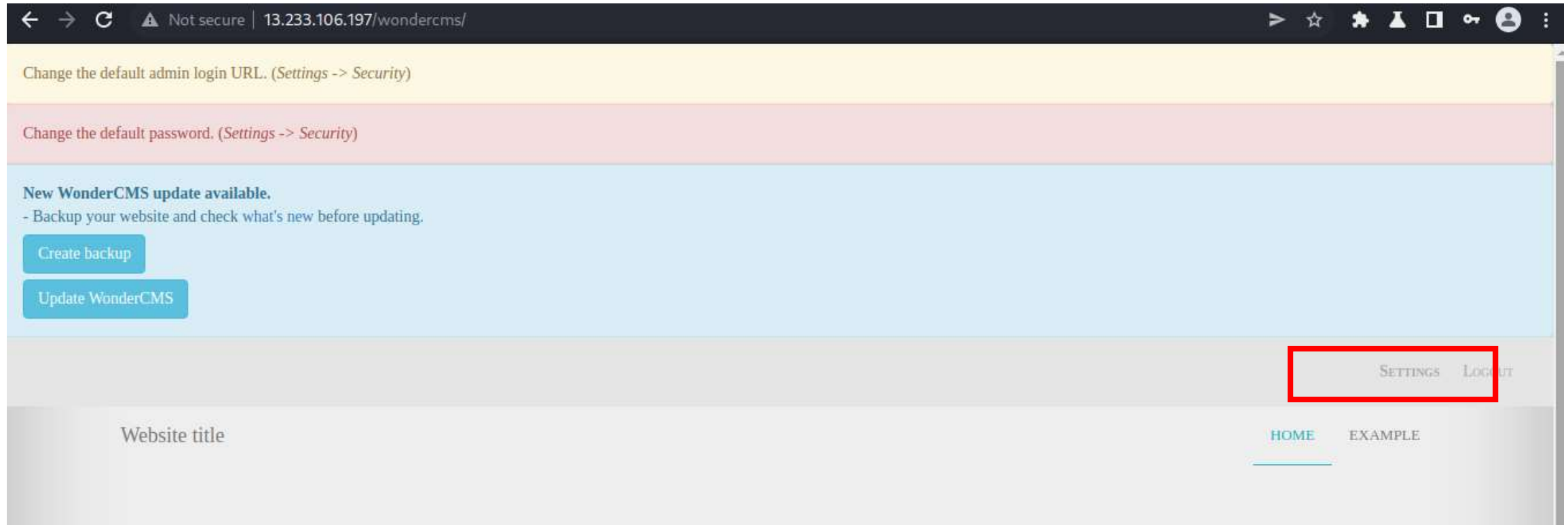
Observation:

- Press the blue hyperlink and use 'admin' as password to get access to admin account.



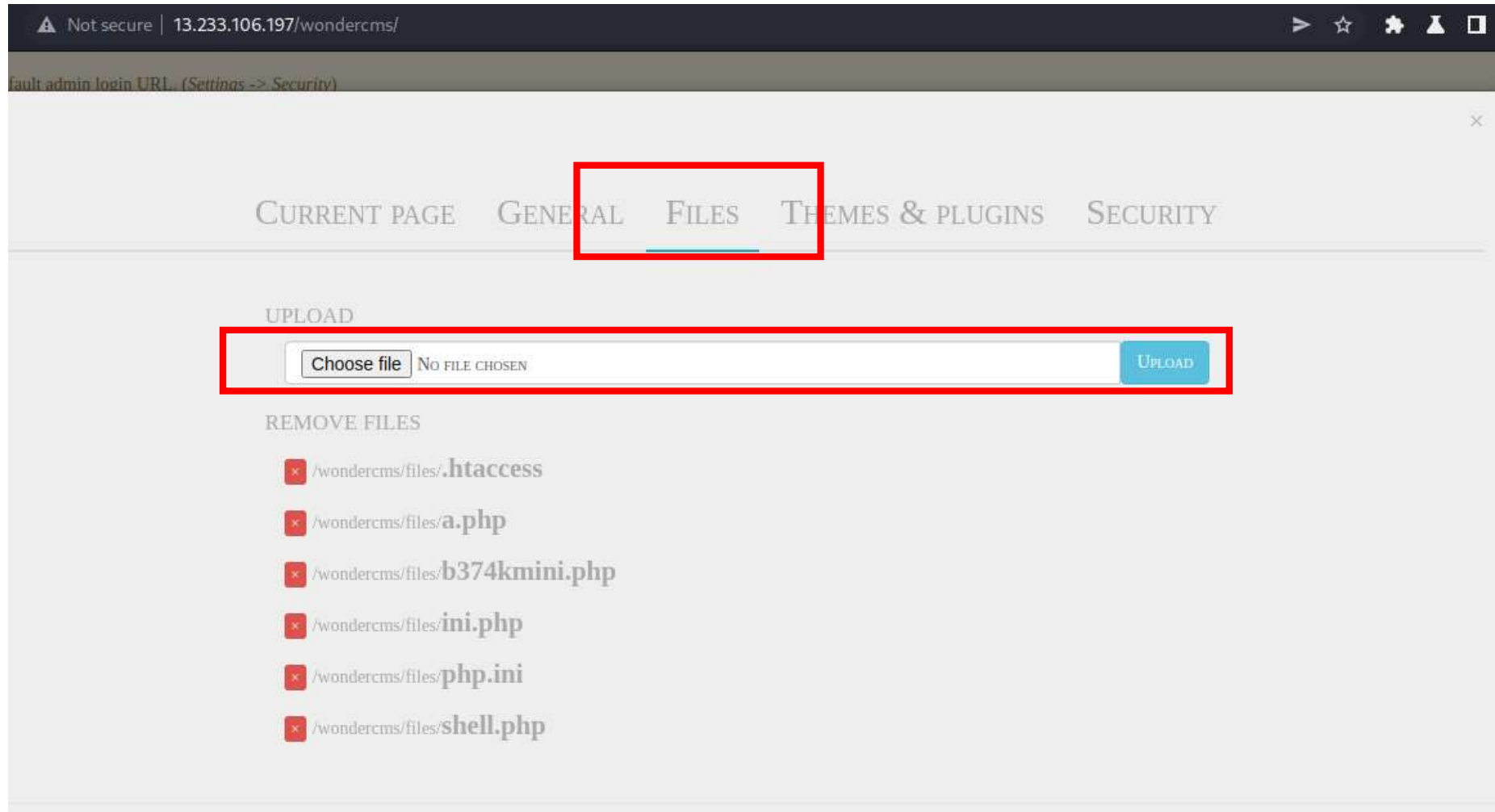
Observation:

- We are now logged in as admin. Press on the settings button.



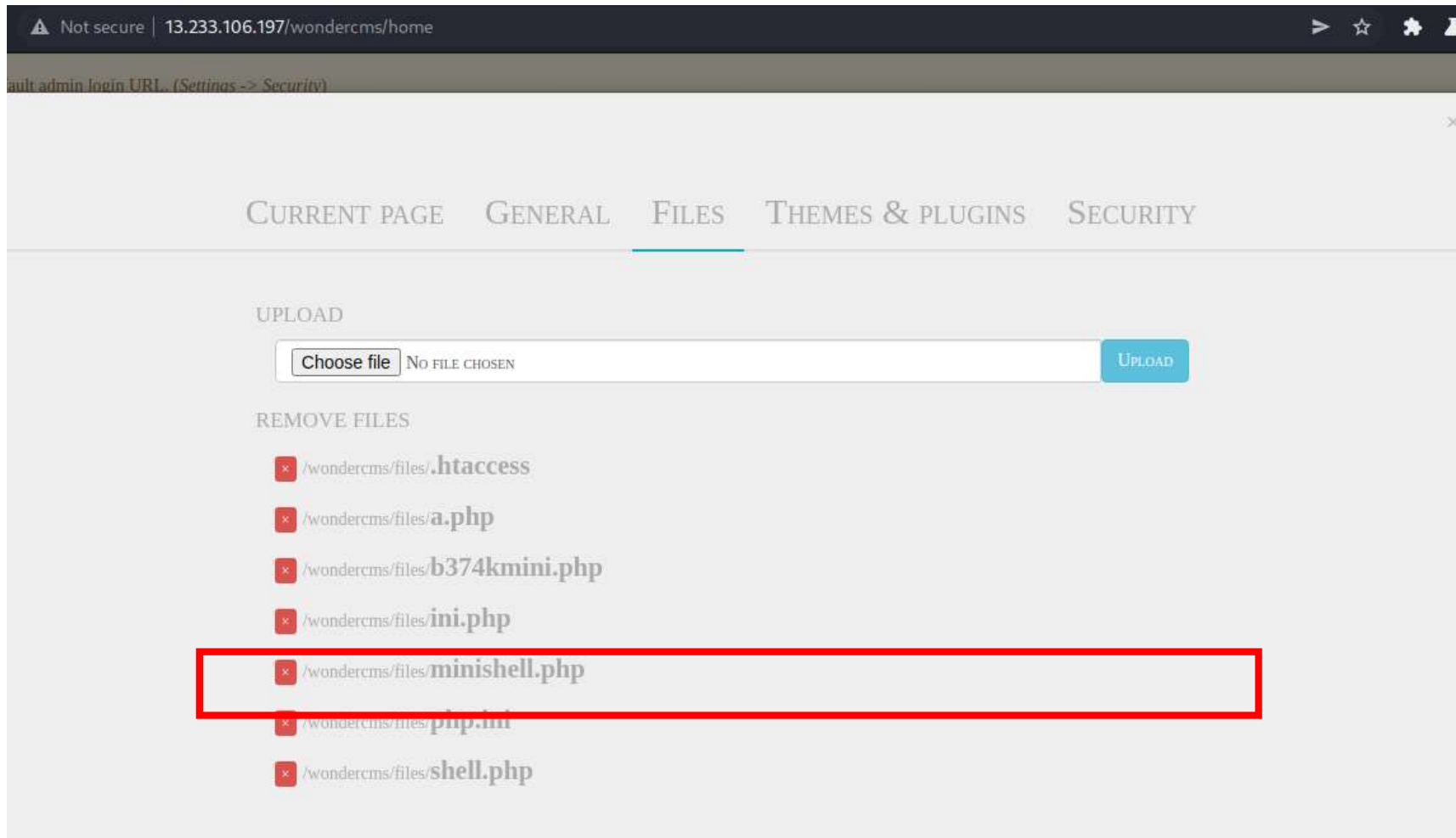
Observation:

- Navigate to the Files section and you will see an option to upload user files. You can upload any file in this panel and it will not block any file types. Let's upload a shell and see what happens.



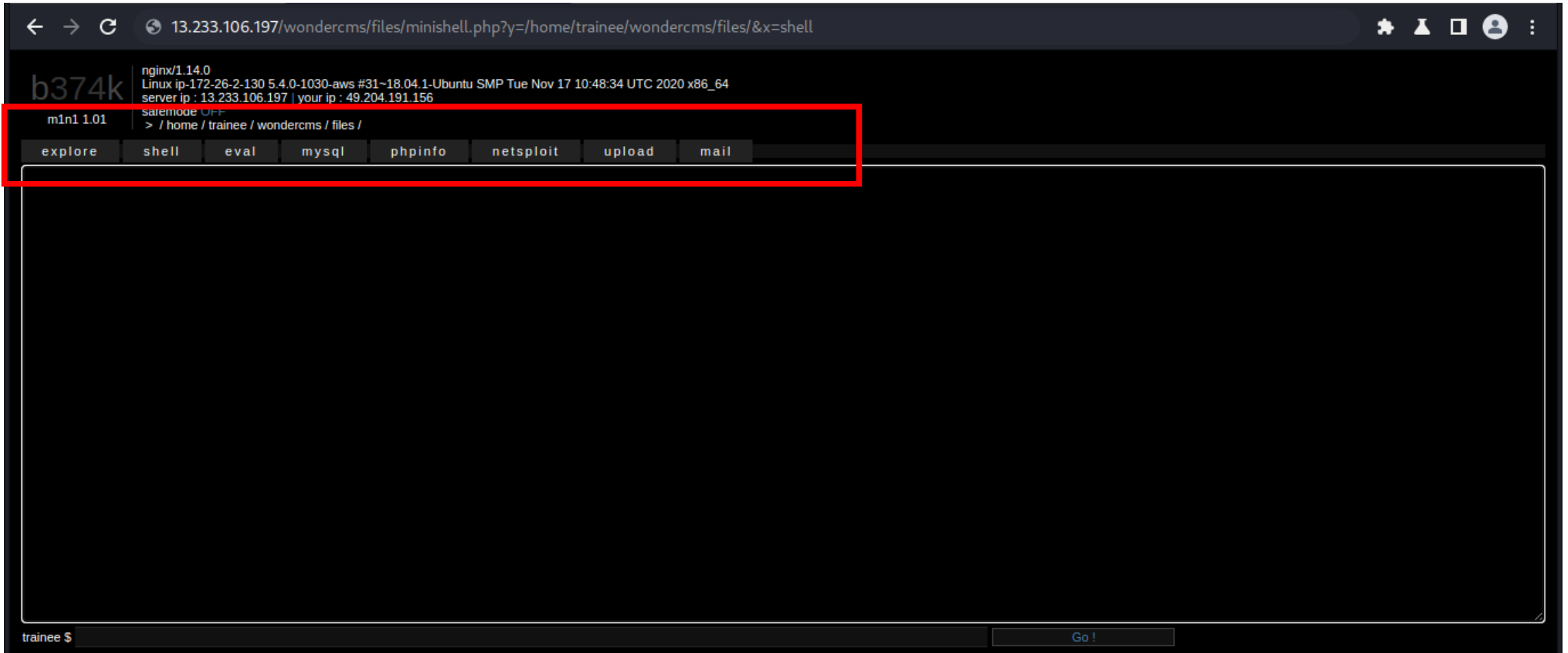
Observation:

- As you can see, our file got uploaded successfully. Now let's try to execute it.



Proof of Concept (PoC):

- Our shell got executed and we now have access to the entire server. We can also run commands from the command line.



Business Impact – Extremely High

There are various consequences for the Insecure file upload:

- including complete system takeover, an overloaded file system or database.
- forwarding attacks to back-end systems.
- client-side attacks, or simple defacement.
- It depends on what the application does with the uploaded file and especially where it is stored.
- The attacker has complete access to the server and can run any terminal commands provided he has the appropriate privileges.
- This is the most impactful vulnerability as the hacker has access to the database and the entire server.
- This shell is essentially a backdoor for the hacker.

Recommendation

- The file types allowed to be uploaded should be restricted to only those that are necessary for business functionality.
- Never accept a filename and its extension directly without having a whitelist filter.
- All the control characters and Unicode and the special characters should be discarded.
- Always check the file before storing it in the database.
- Don't run database as root or admin unless necessary.

References:

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.hackingarticles.in/comprehensive-guide-on-unrestricted-file-upload/>

7. Client Side Filter Bypass

Input of
invalid data
due to Client
Side Filter
Bypass.
(Moderate)

Can enter any value instead of a just numbers at the phone number field in signup page(Burp Suite).

Affected URL :

- <http://13.233.106.197/signup/customer.php>

7. Client Side Filter Bypass

Input of
invalid data
due to Client
Side Filter
Bypass.
(Moderate)

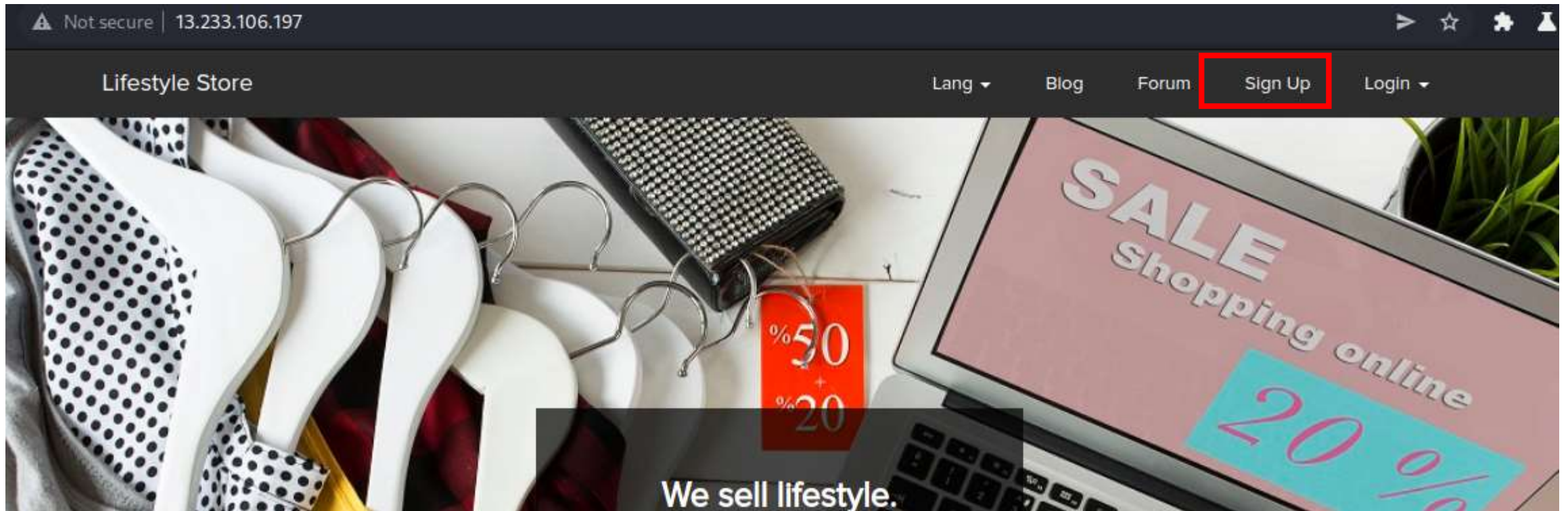
A similar issue is found on the below mentioned url.

Affected URL :

- <http://13.233.106.197/profile/16/edit>

Observation:

- Navigate to the home page and click on Sign Up.



Observation:

- This is the signup page. Here the Contact No. Field is vulnerable.

Not secure | 13.233.106.197/signup/customer.php

Lifestyle Store Blog Forum Sign Up Login ▾

Customer Sign Up

Name

Email

Password

Username

Contact No.

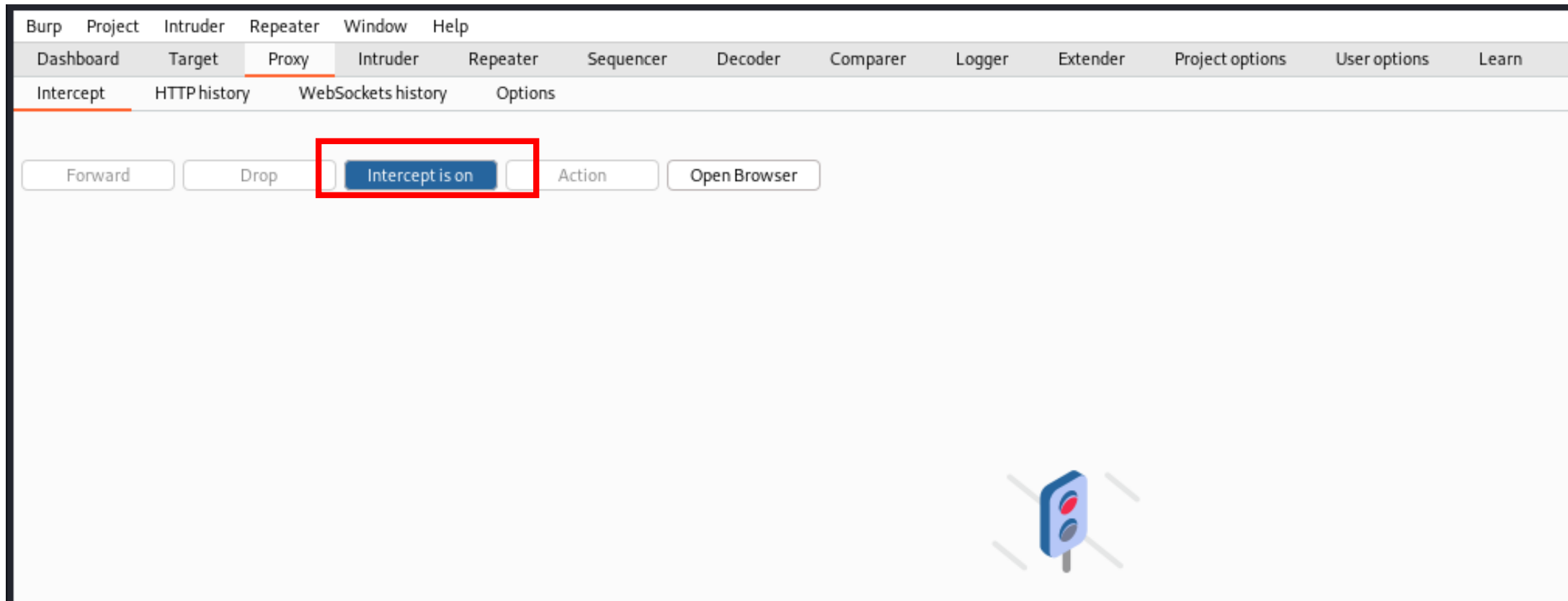
Address

Sign Up

Already have an account? [Login here!](#)

Observation:

- Open Burp Suite and turn Intercept on.



Observation:

- Enter some valid input and click on signup.

Not secure | 13.233.106.197/signup/customer.php

Lifestyle Store Blog Forum Sign Up Login ▾

Customer Sign Up

xyz

xyz@xyz.com

...

xyz

9876543210

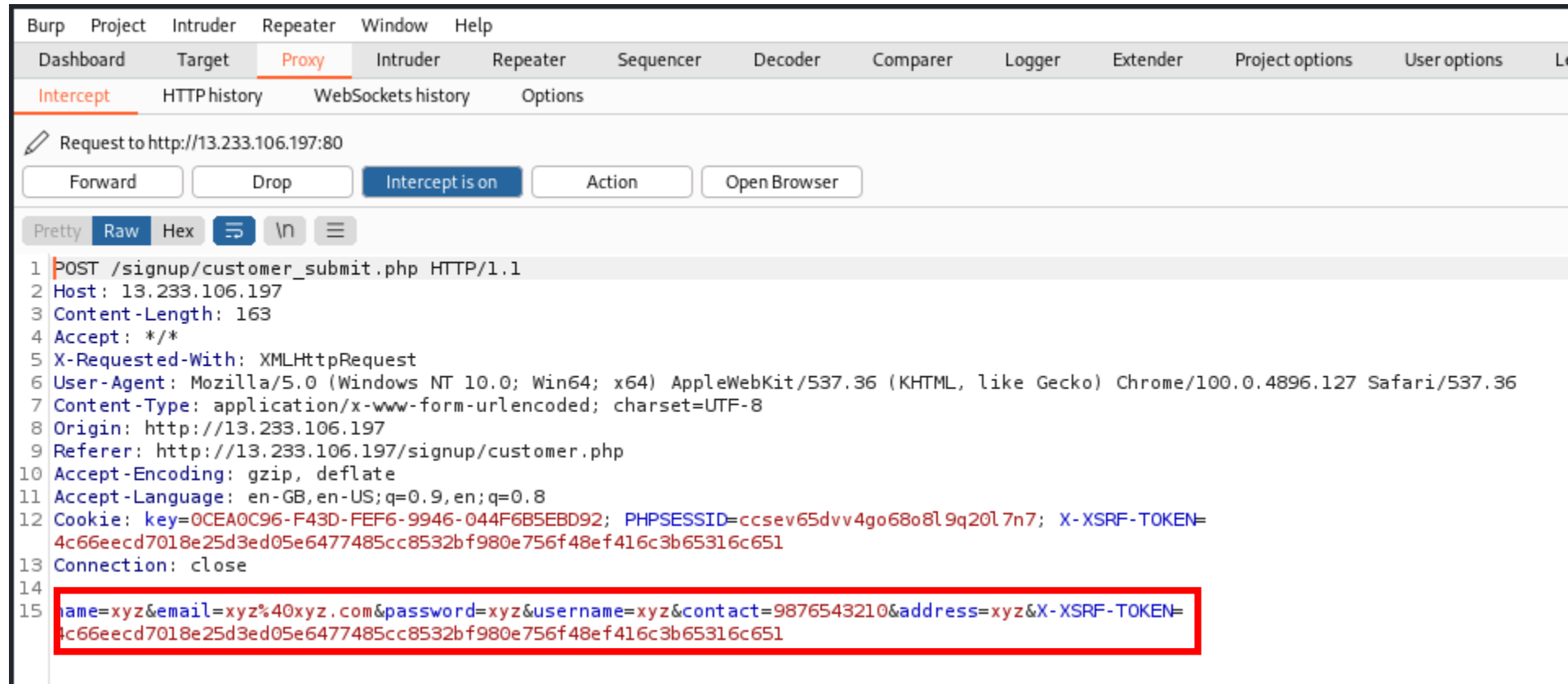
xyz

Sign Up

Already have an account? [Login here!](#)

Observation:

- This is the input we provided.



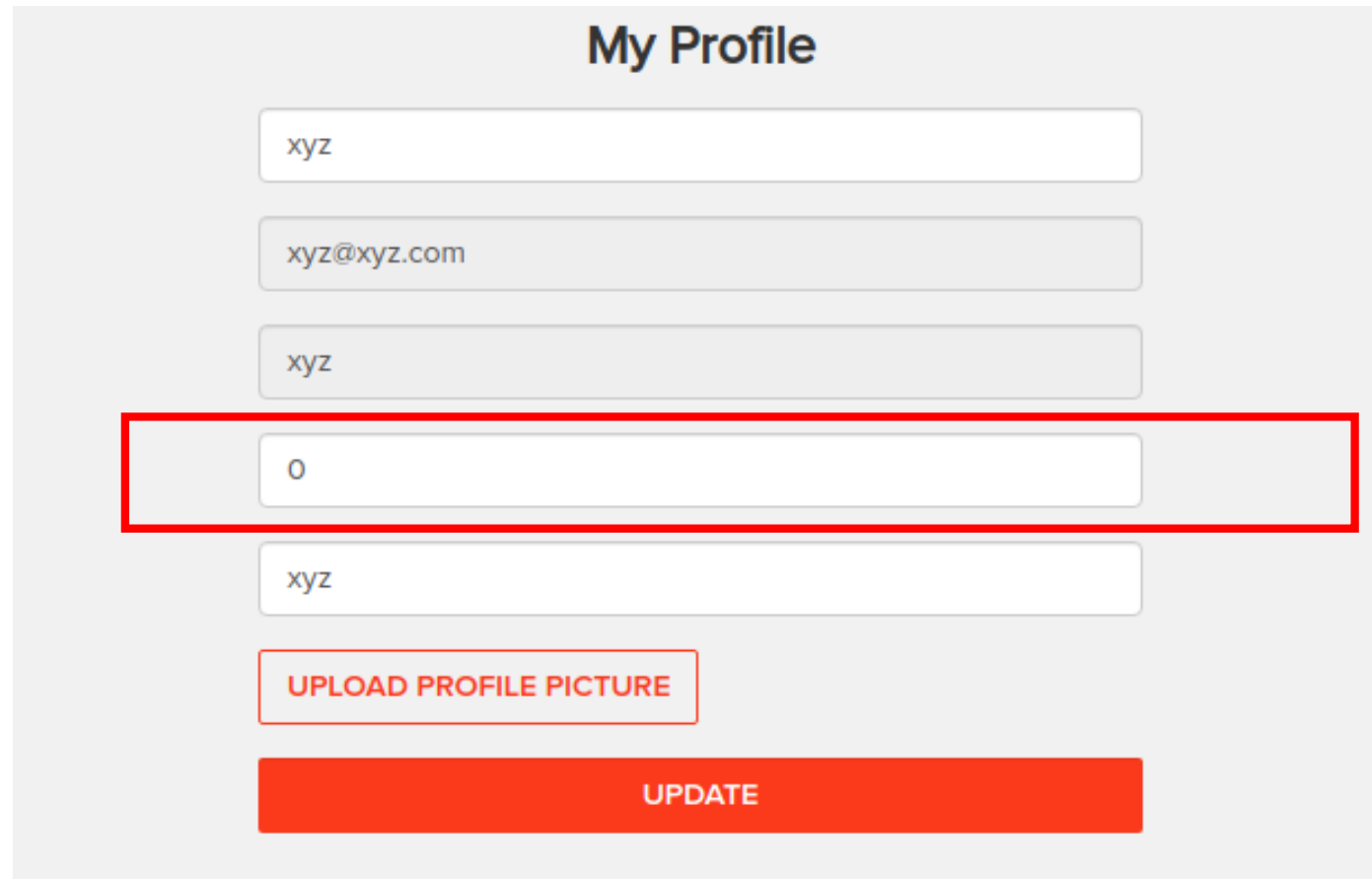
Observation:

- Let's tamper with the data in the contact parameter.

```
10 Accept-Encoding: gzip, deflate
11 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
12 Cookie: key=0CEA0C96-F43D-FEF6-9946-044F6B5EBD92; PHPSESSID=ccsev65dvv4go68o8l9q20l7n7; X-XSRF-TOKEN=
4c66eecd7018e25d3ed05e6477485cc8532bf980e756f48ef416c3b65316c651
13 Connection: close
14
15 name=xyz&email=xyz%40xyz.com&password=xyz&username=xyz&contact=somejunkvalue&address=xyz&X-XSRF-TOKEN=
4c66eecd7018e25d3ed05e6477485cc8532bf980e756f48ef416c3b65316c651
```

Proof of Concept (PoC):

- The Contact field will be '0' on successful signup.



The image shows a web form titled "My Profile" with a light gray background. It contains several input fields and two buttons. The fields are arranged vertically: a text field with "xyz", a text field with "xyz@xyz.com", a text field with "xyz", a text field with "0" (highlighted by a red rectangular box), and a text field with "xyz". Below the fields are two buttons: "UPLOAD PROFILE PICTURE" and "UPDATE".

My Profile

xyz

xyz@xyz.com

xyz

0

xyz

UPLOAD PROFILE PICTURE

UPDATE

Business Impact – Moderate

A malicious hacker can signup and use a valid account without providing his number and possibly hide his identity in case he does something notorious.

Recommendation

- Implement all critical checks on server side side.
- Client-side checks must be treated as decorative .
- 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.
- This will ensure that the hacker cannot bypass the client side filters and exploit them.

References:

- <https://portswigger.net/support/using-burp-to-bypass-client-side-javascript-validation>
- <https://www.slideshare.net/SamBowne/cnit-129s-ch-5-bypassing-clientside-controls>

8. Directory Listing

Directory
Listing
(Moderate)

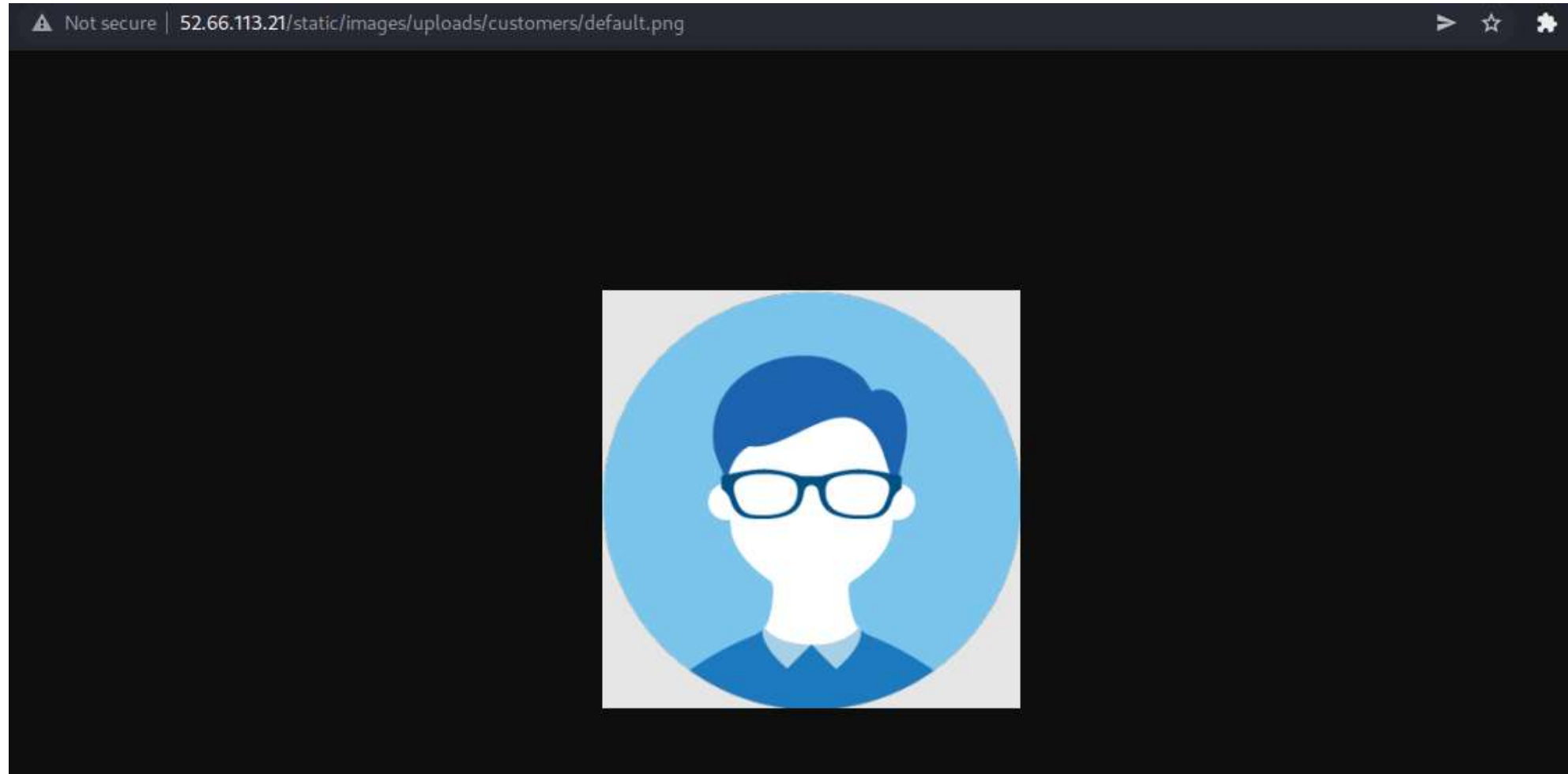
Opening any image gives the full path to the image.

Affected URL :

- <http://52.66.113.21/static/images/uploads/customers/default.png>

Observation

- Open any image in new tab and notice the URL. It shows the full path of the image. This can be used by the hacker to plan further attacks.



Business Impact – Moderate

Although this vulnerability has no direct impact on users or the server, it could provide information about the server and users to the hacker.

Furthermore, an attacker can simply download and see the pictures.

Recommendation

Take the following precautions:

- Disable Directory Listing
- Put an index.html in all folders with default message

References:

<https://cwe.mitre.org/data/definitions/548.html>

<https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/>

9. Information Disclosure Due to Default Pages

| | |
|--|--|
| Information Disclosure due to Apache default Pages (Low) | <p>Below mentioned URL discloses server information</p> <p>Affected URL :</p> <ul style="list-style-type: none">• http://URL/server-status |
|--|--|

9. Information Disclosure Due to Default Pages

Information
Disclosure
due to
Apache
default
Pages
(Severe)

Below mentioned URLs disclose important information.

Affected URL :

- <http://URL/phpinfo.php>
- <http://URL/robots.txt>
- <https://URL/userlist.txt>

Proof of Concept (PoC):

- Navigate to mentioned URL
- Default server-status page opens which discloses server information

← → ↻ ⚠ Not secure | 52.66.113.21/server-status/

Apache Server Status for localhost (via 127.0.0.1)

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

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

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

W_.....
.....
.....

Proof of Concept (PoC):

- Navigate to mentioned URL
- Default phpinfo.php page opens which discloses server information



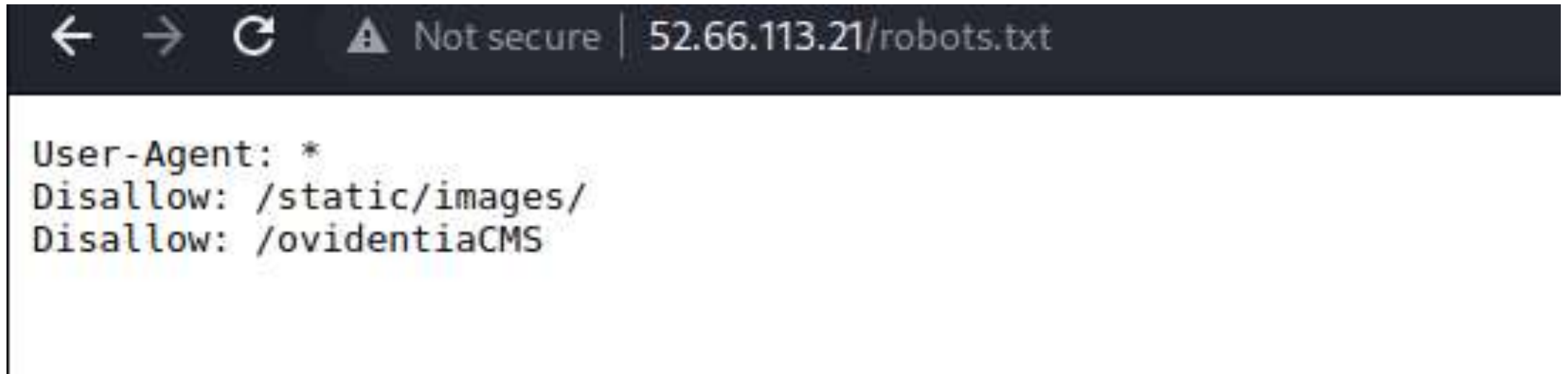
PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1



| | |
|---|--|
| System | Linux ip-172-26-2-248 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 |

Proof of Concept (PoC):

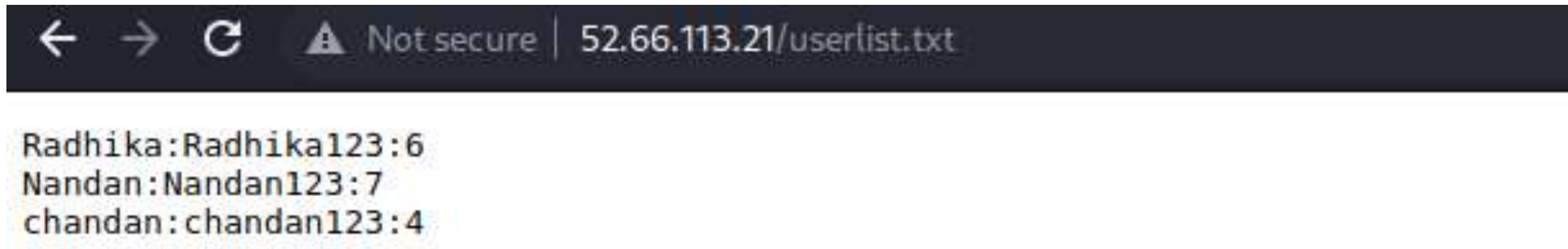
- Navigate to mentioned URL
- Default robots.txt file opens which displays URLs that have been blocked by search engines.

A screenshot of a web browser window. The address bar at the top shows navigation icons (back, forward, refresh), a warning icon, the text "Not secure", and the URL "52.66.113.21/robots.txt". The main content area displays the text of a robots.txt file in a monospaced font.

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

Proof of Concept (PoC):

- Navigate to mentioned URL
- Default userlist.txt file opens which displays the user id and passwords of seller accounts, that too in plain text without any kind of encryption.



Business Impact – High

The attacker can exploit this vulnerability to get login credentials of sellers.
He can get critical information about the server and the hidden pages of the website.
The attacker can mess up a sellers account or may simply delete all data.

Recommendation

Take the following precautions:

- Disable all default pages and folders including server-status, phpinfo.php, robots.txt, userlist.txt.
- Make sure that only authorized users can gain access to such pages.

References:

- <https://portswigger.net/web-security/information-disclosure#:~:text=Information%20disclosure%2C%20also%20known%20as,as%20usernames%20or%20financial%20informati>
- <https://infosecwriteups.com/all-about-information-disclosure-5edb5459a514>
- https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

10. Components with known vulnerabilities

Components
with known
vulnerabilities
(Severe)

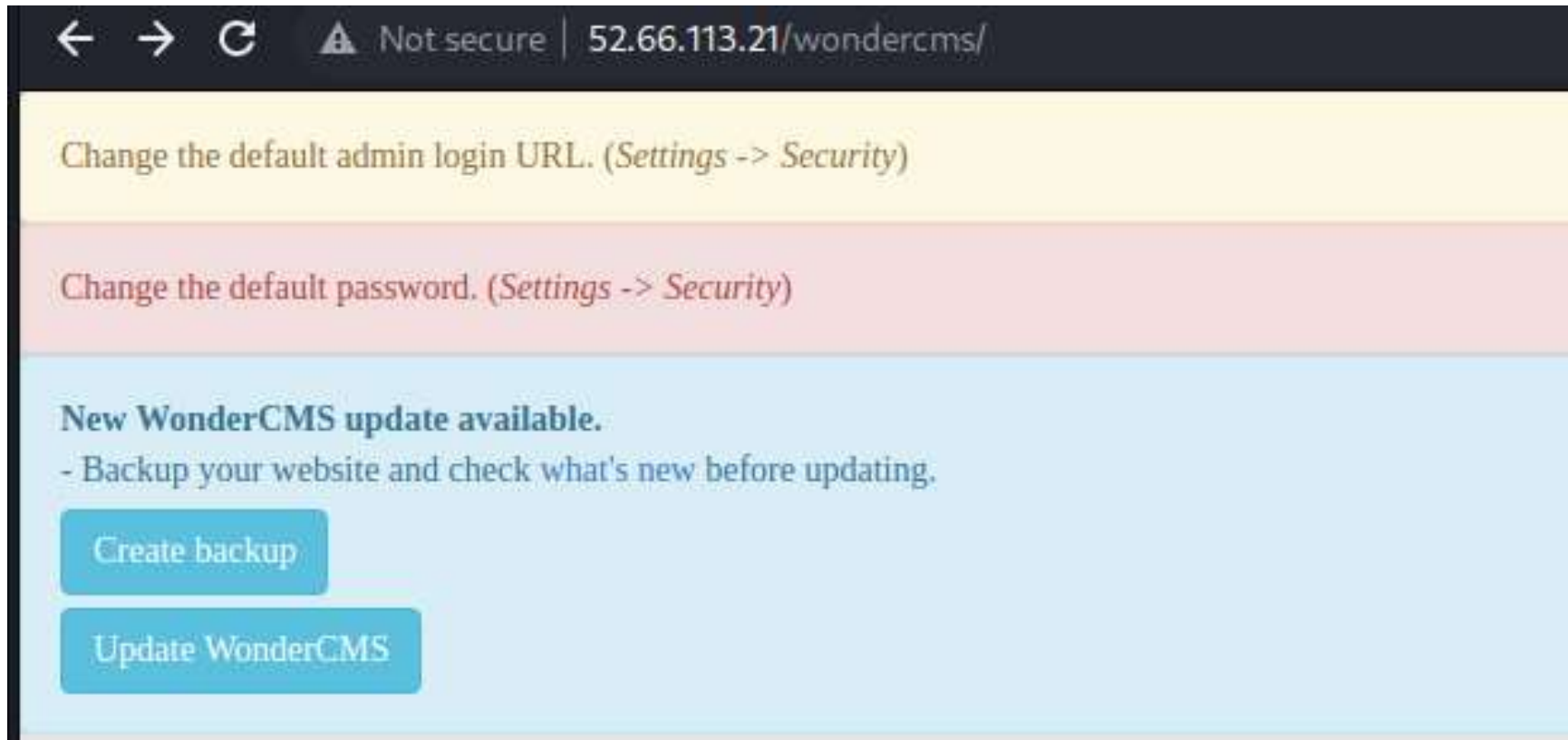
Below mentioned URLs consist of components with known vulnerabilities.

Affected URL :

- <http://URL/forum/>
- <http://URL/wondercms/>

Observation:

- WonderCMS is outdated and has public exploits.



Proof of Concept (PoC):

<https://cve.report/CVE-2019-5956>

Home

CVE.report Search

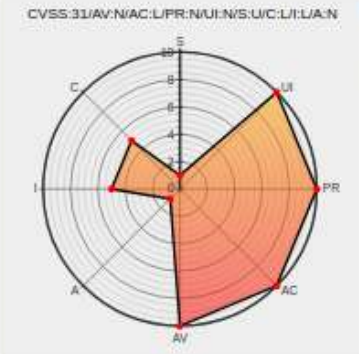
CVE-2019-5956

Published on: 09/12/2019 12:00:00 AM UTC

Last Modified on: 03/23/2021 11:27:59 PM UTC

CVE-2019-5956


[Source: Mitre](#) [Source: Nist](#) [Print: PDF](#)



CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

Certain versions of [Wondercms](#) from [Wondercms](#) contain the following vulnerability:

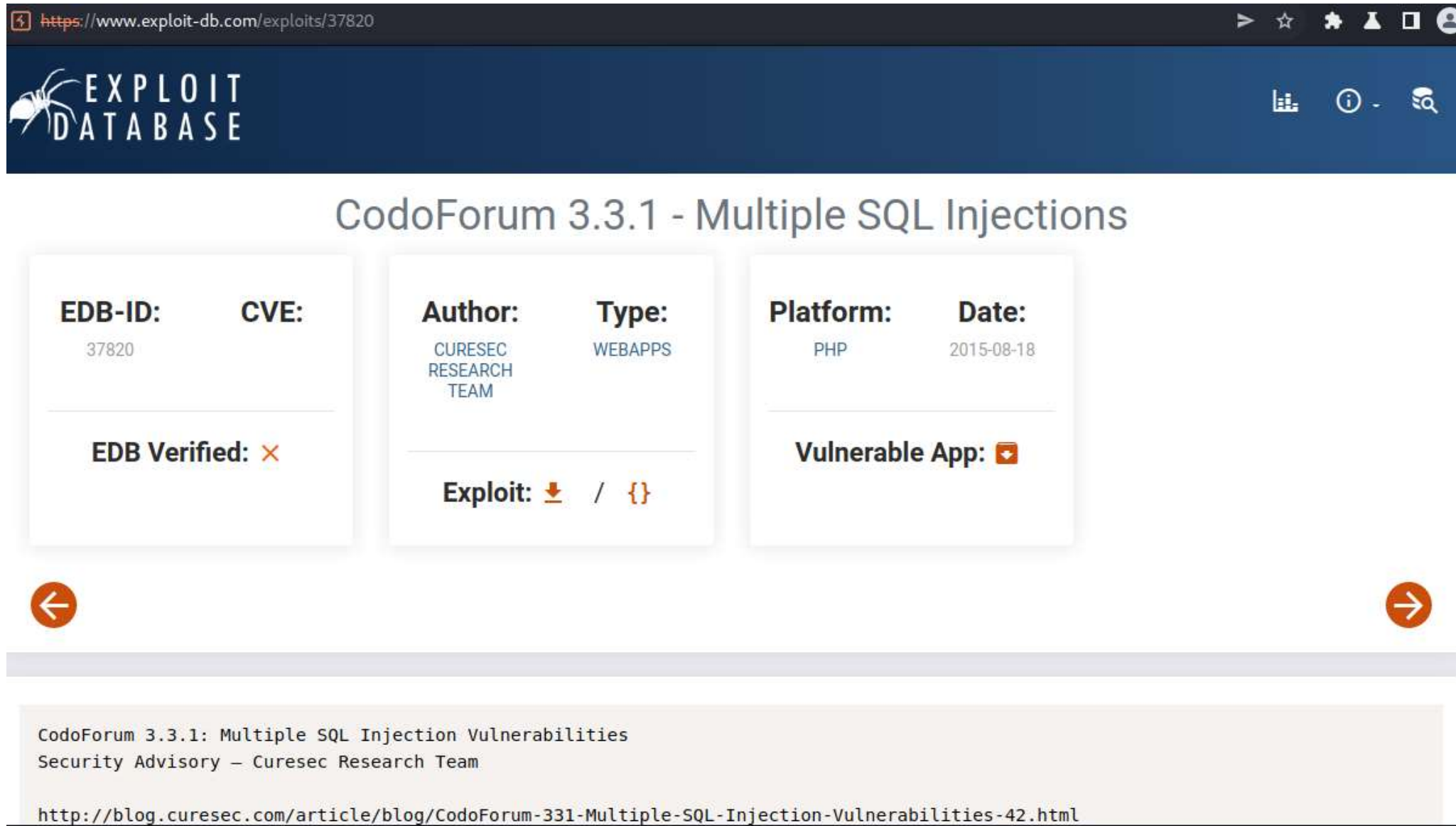
Directory traversal vulnerability in WonderCMS 2.6.0 and earlier allows remote attackers to delete arbitrary files via unspecified vectors.

CVE-2019-5956 has been assigned by  URL Logo vultures@jpcert.or.jp to track the vulnerability - currently rated as **MEDIUM** severity.

Observation:

- CODOLOGIC by Codeforum is also Outdated and has public exploits.

Proof of Concept (PoC):



The screenshot shows a web browser displaying an entry on the Exploit Database. The URL in the address bar is <https://www.exploit-db.com/exploits/37820>. The page title is "CodoForum 3.3.1 - Multiple SQL Injections". The entry details are as follows:

| EDB-ID: | CVE: | Author: | Type: | Platform: | Date: |
|---------|------|-----------------------------|---------|-----------|------------|
| 37820 | | CURESEC RESEARCH TEAM | WEBAPPS | PHP | 2015-08-18 |

Below the details, there are three sections:

- EDB Verified:** ✗
- Exploit:** ⬇ / {}
- Vulnerable App:** 📄

At the bottom, there is a description of the vulnerability and a link to the security advisory:

CodoForum 3.3.1: Multiple SQL Injection Vulnerabilities
Security Advisory – Curesec Research Team
<http://blog.curesec.com/article/blog/CodoForum-331-Multiple-SQL-Injection-Vulnerabilities-42.html>

Business Impact – Extremely High

- Attackers can perform any attacks available publicly.
- The attacker can cause severe damage to the website.
- He may be able to upload backdoor shells as shown previously.

Recommendation

- Keep all components Updated.
- Don't let users see the version of third party components.
- Keep an eye out for the security of third party tools.
- If the integrated third party things are vulnerable, so is your website.
- Shut down the third party components as soon as a vulnerability is found and don't use it unless a patch has been issued to fix it.

References:

- https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities
- https://www.cvedetails.com/vulnerability-list/vendor_id-15088/product_id-30715/version_id-235577/Wondercms-Wondercms-2.3.1.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-15315/Codoforum.html

11. Weak Password

Weak
password/Default password
(Critical)

OvidentiaCMS is using default login credentials that are publicly available online.

Affected URL :

- <http://13.233.106.197/ovidentiaCMS//index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1>

Observation:

- Navigate to ovidentiaCMS and click on 'connexion' to access the login page.

Not secure | 13.235.243.245/ovidentiaCMS/

Utilisateur

Connexion

outil permettant de publier avec une grande aisance et très rapidement un portail intranet, extranet ou internet.
ir ses fonctions de système de gestion de contenus (CMS) telles que :

- informations (éditeur WYSIWYG, arborescence d'articles, catégorisation),
- ice de circuits d'approbations (permettant de définir des schémas d'approbations, du plus simple au plus
- de recherche,

gre aussi de puissants outils de travail collaboratif :

- s utilisateurs, agendas partagés, notifications, annuaires,
- naire de fichiers (avec gestion du versioning)

re de congés (avec circuit de validation)

de gérer des groupes avec administration déléguée (dans un certain périmètre et pour certaines fonctions
it)

omplètement modulaire, permet d'y installer des modules développés par la communauté **OVIDENTIA**

Ovidentia.org

Ce flux d'information n'a pas été mis à jour depuis le 09/03/2019 19:07. Probablement à cause d'une interruption de service, la mise à jour du flux a été désactivée

Mettre à jour

Nouvel environnement de mise à disposition des modules et du noyau

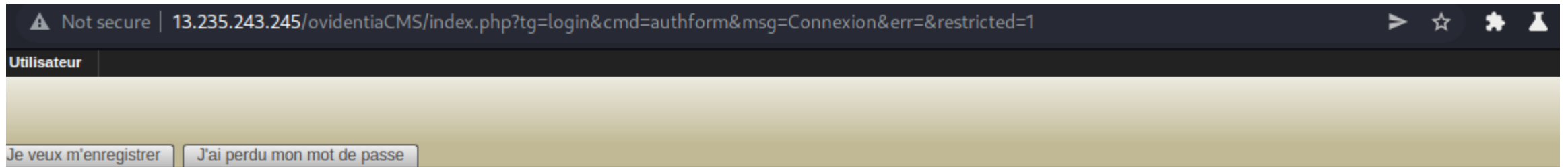
10/08/2017 17:04

Afin de faciliter la mise à disposition des dernières version des modules et du noyau (stable et développement), un "store appllicatif" dédié à Ovidentia vient d'être intégré.

Modules

Observation:

- This page is using default login credentials.



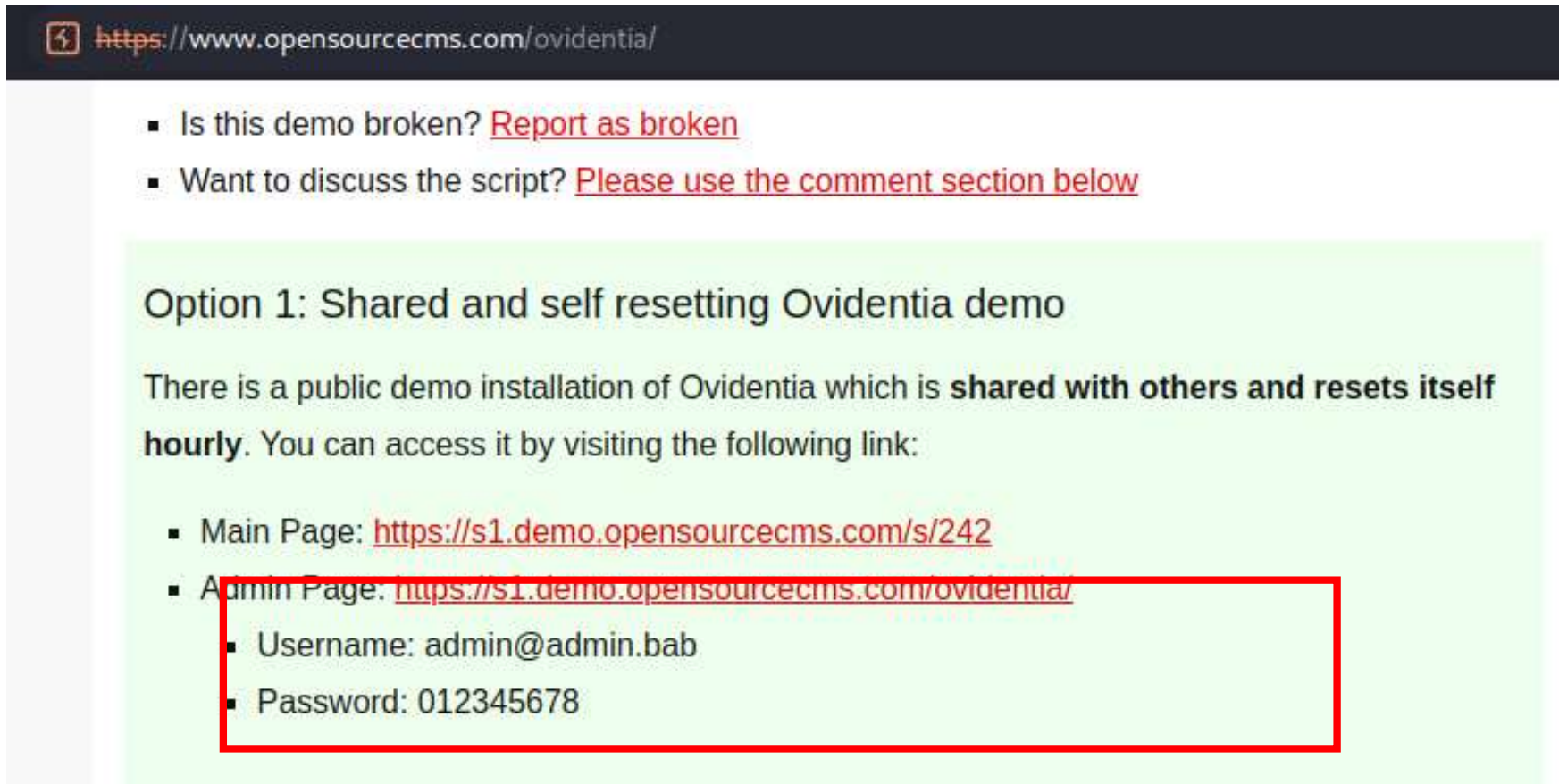
n

| | |
|--|--------------------------|
| Identifiant : | <input type="text"/> |
| Mot de passe : | <input type="password"/> |
| <input type="button" value="Connexion"/> | |

[Portail collaboratif](#) Réalisé par Ovidentia, Ovidentia est une marque déposée par [Cantico](#).

Observation:

- The default login credentials are publicly available online.



The screenshot shows a web browser address bar with the URL <https://www.opensourcecms.com/ovidentia/>. Below the address bar, there are two links: "Is this demo broken? [Report as broken](#)" and "Want to discuss the script? [Please use the comment section below](#)". A green box highlights the "Option 1: Shared and self resetting Ovidentia demo" section. This section contains the text: "There is a public demo installation of Ovidentia which is **shared with others and resets itself hourly**. You can access it by visiting the following link:". Below this text, there are three items listed: "Main Page: <https://s1.demo.opensourcecms.com/s/242>", "Admin Page: <https://s1.demo.opensourcecms.com/ovidentia/>", and "Username: admin@admin.bab". A red box highlights the "Admin Page" link and the default login credentials: "Password: 012345678".

▪ Is this demo broken? [Report as broken](#)

▪ Want to discuss the script? [Please use the comment section below](#)

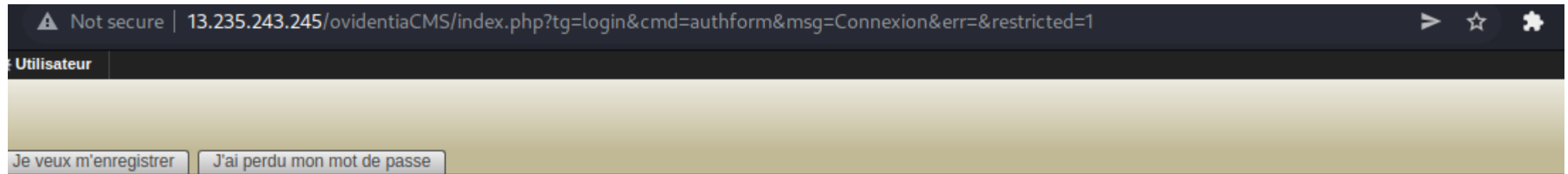
Option 1: Shared and self resetting Ovidentia demo

There is a public demo installation of Ovidentia which is **shared with others and resets itself hourly**. You can access it by visiting the following link:

- Main Page: <https://s1.demo.opensourcecms.com/s/242>
- Admin Page: <https://s1.demo.opensourcecms.com/ovidentia/>
 - Username: admin@admin.bab
 - Password: 012345678

Observation:

- Put the login credentials in the login field and proceed.



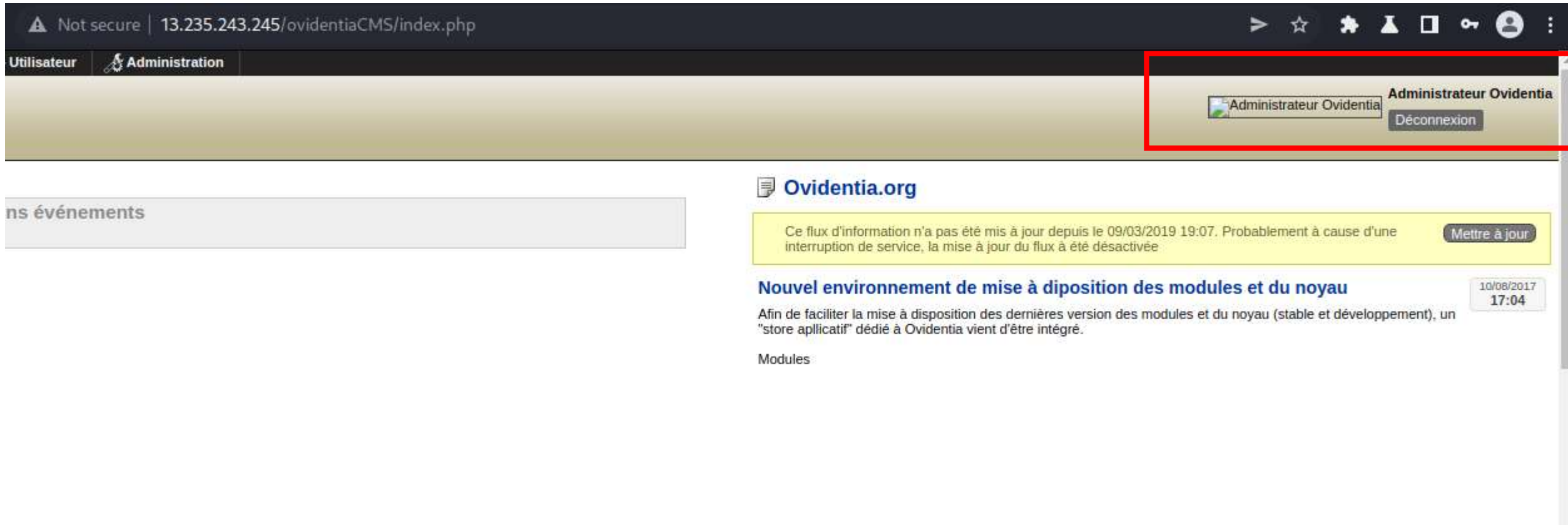
on

| | |
|--|--|
| Identifiant : | <input type="text" value="admin@admin.bab"/> |
| Mot de passe : | <input type="password" value="....."/> |
| <input type="button" value="Connexion"/> | |

[Portail collaboratif](#) Réalisé par Ovidentia, Ovidentia est une marque déposée par [Cantico](#).

Proof of Concept (PoC):

- As you can see, we have successfully logged in into the admin account.



Business Impact – Extremely High

- The attacker can gain admin access over the entire page without having to use any tools or techniques.
- He can simply search for the default login credentials and will be able to login as admin.
- He can run exploits with admin privileges to enhance his attack.

Recommendation

- Don't use default login credentials.
- Make the password hard to guess so that the attacker cannot easily brute force your password.
- Enable 2 Factor authentication for admin accounts.
- Don't let normal users access the admin login page.

References:

- <https://www.opensourcecms.com/ovidential/>
- https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authenticationhttps://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

12. File Inclusion

Remote File
Inclusion
(Critical)

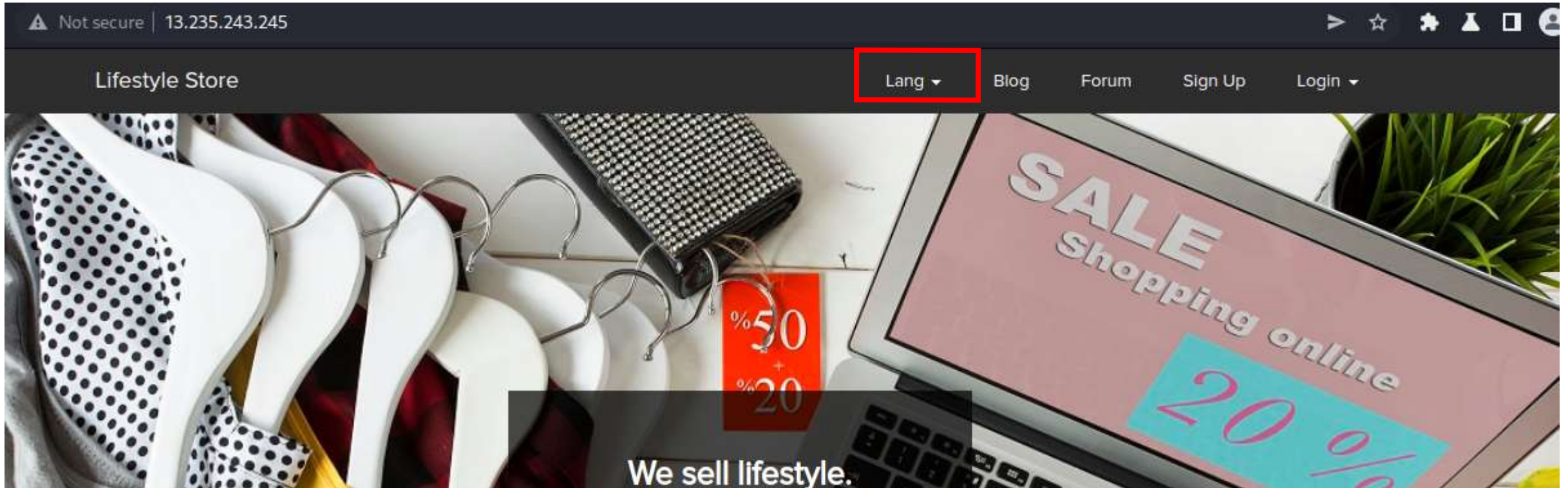
Below mentioned URL is vulnerable to remote file inclusion.

Affected URL :

- <http://13.235.243.245/?includelang=lang/en.php>

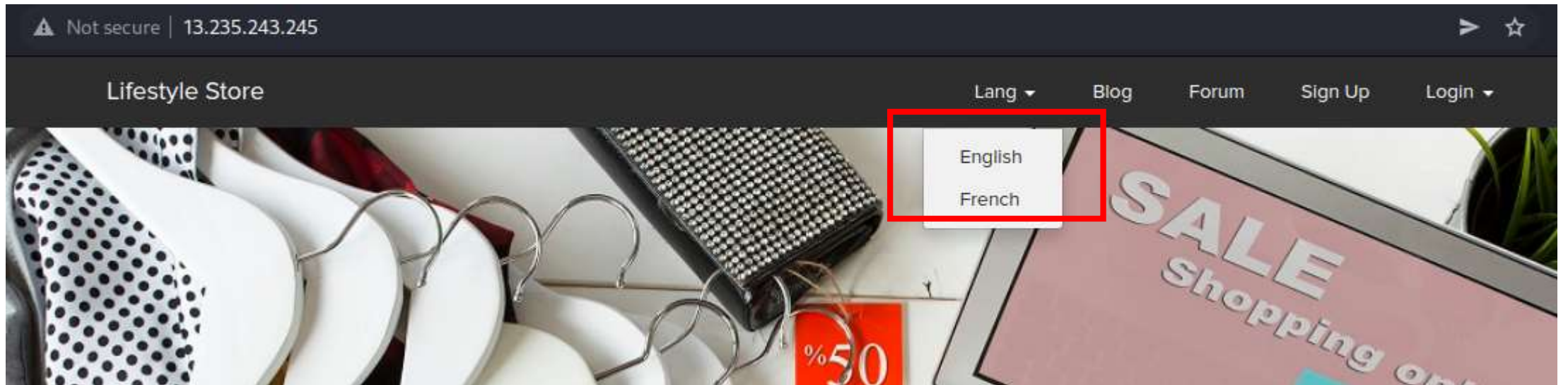
Observation:

- Click on the Lang button.



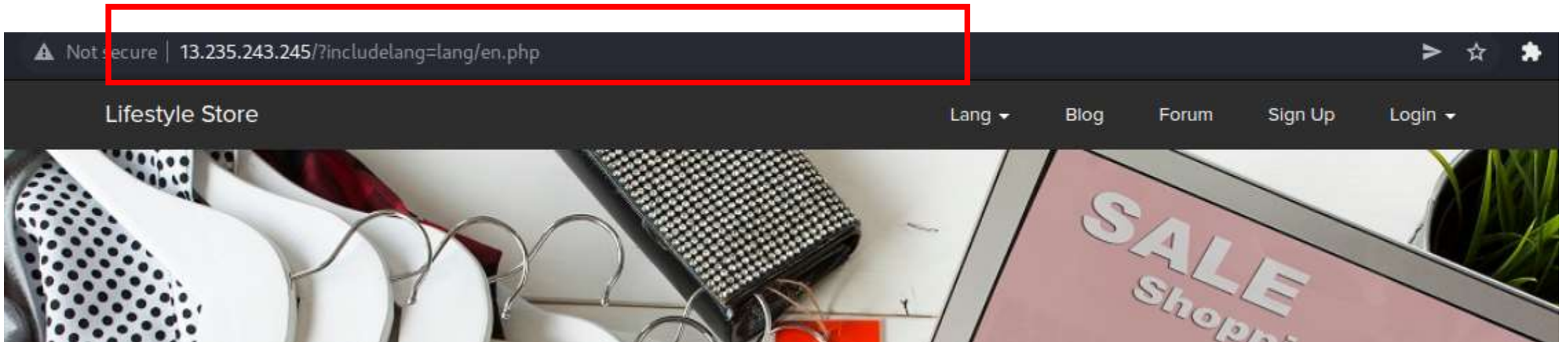
Observation:

- Select a language (Preferably English).



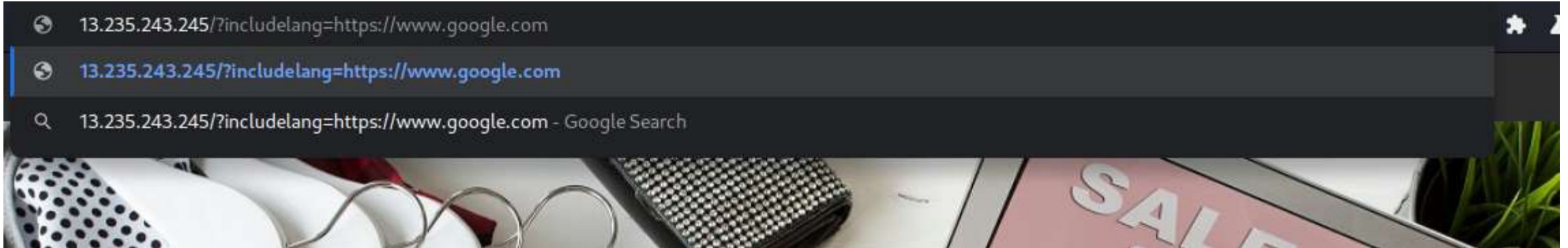
Observation:

- Take a look at the URL.
- The part after 'includelang=' is a URL.
- You can replace it with any URL.



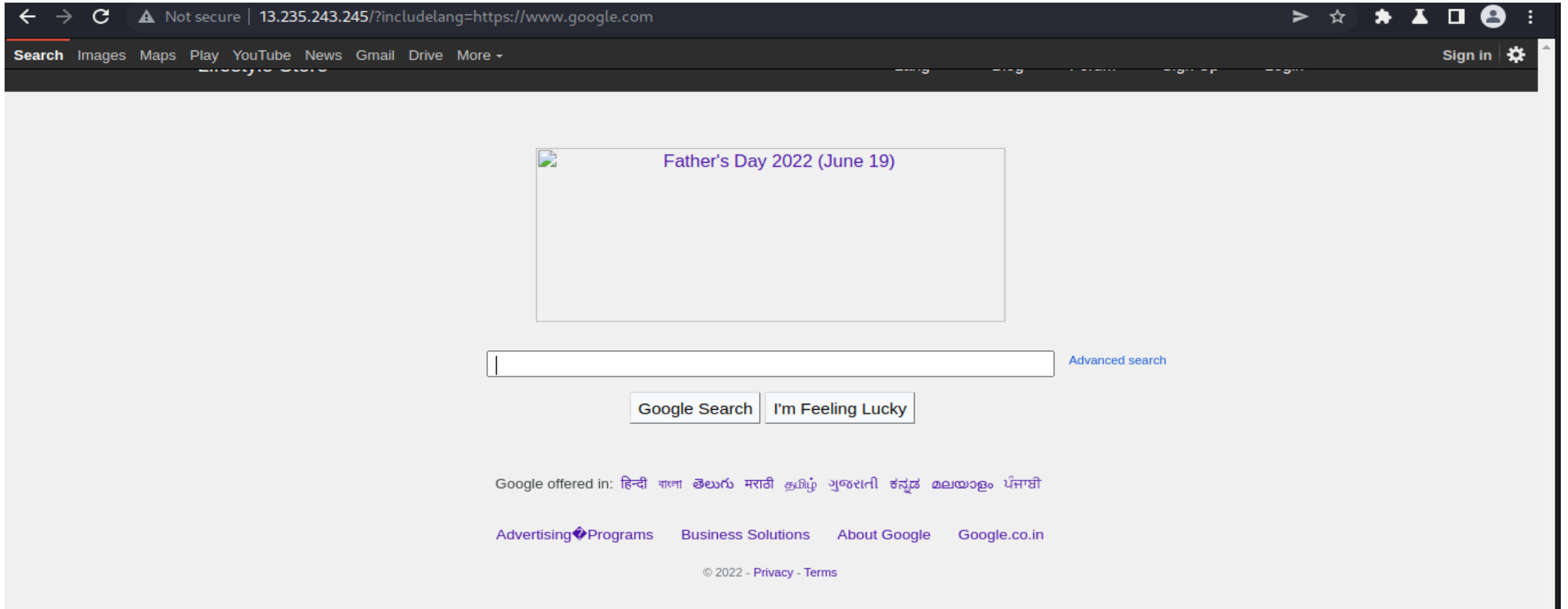
Observation:

- Enter a destination URL to redirect and press enter.
- Here, I have used 'http://www.google.com'.



Proof of Concept (PoC):

- We have been redirected to the specified URL. You can see that it works like an iframe.



Business Impact – Extremely High

- The attacker can put the URL to an online shell and gain access to the victim system.
- The attacker can run remote code to hack the target system.
- The attacker can run commands without the knowledge of the victim.

Recommendation

- Use POST method instead of GET method.
- Don't allow users to change the destination of Redirection.
- Make sure to check for the origin of the request.
- Don't allow any other URL modifications.

References:

- https://en.wikipedia.org/wiki/File_inclusion_vulnerability
- https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.2-Testing_for_Remote_File_Inclusion

13. PII Leakage

PII Leakage
for sellers
(Severe)

Below mentioned URL gives Personally Identifiable Information(PAN number) of sellers.

Affected URL :

- http://13.235.243.245/products/details.php?p_id=16

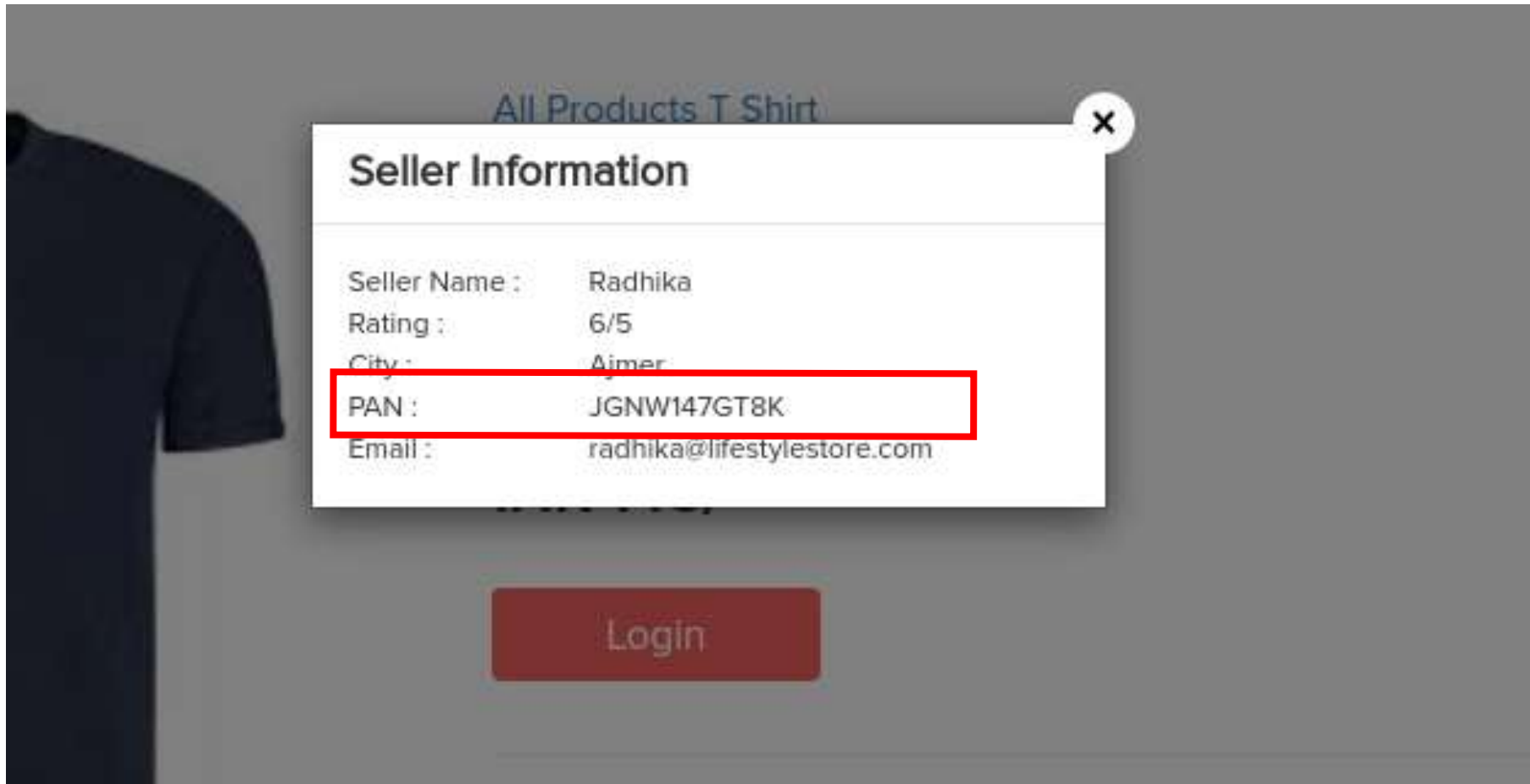
Observation:

- Go to the products page and select a product. Press the 'Seller Info' button.



Proof of Concept (PoC):

- We can see the information of the seller. But the users do not need to know the PAN number of the seller.



Business Impact – Extremely High

- The attacker can use the PAN number of the seller to plan social attacks on the seller.
- He may also use the seller's PAN number to link to other services without the consent of the seller.

Recommendation

- Keep the critical information of users or sellers private to them.
- Others should not be able to see that information.
- Information like PAN number, Credit card details, etc should never be stored in plain text.
- Only the concerned user should be able to access his own data.

References:

- <https://www.geeksforgeeks.org/personally-identifiable-information-leakage-vulnerability/?ref=rp>
- https://en.wikipedia.org/wiki/Information_leakage

14. Open Redirection

Open Redirection (Severe)

Below mentioned URL is vulnerable to remote file inclusion.

Affected URL :

- <http://13.235.243.245/redirect.php?url=>

I have also attached a video named 'Open Redirection 1.mp4' Have a look at it for better understanding.

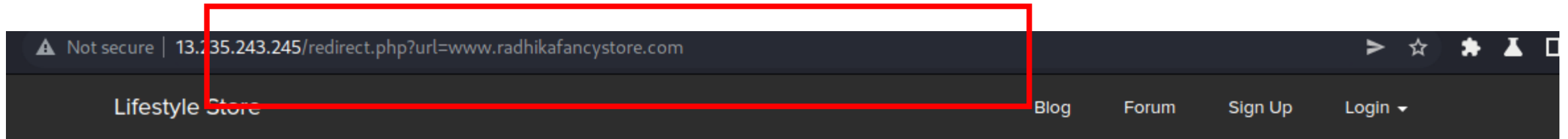
Observation:

- Go to the products page and select a product. Press the 'Brand Website' button.



Observation:

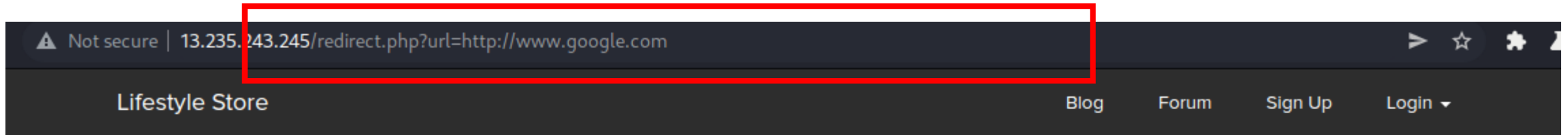
- This URL is vulnerable to open redirection. Let's tamper with the URL.
- You need to tamper with the URL within 10 seconds.
- Alternatively, you can copy the URL and modify it later.



You will be redirected in 5 seconds

Observation:

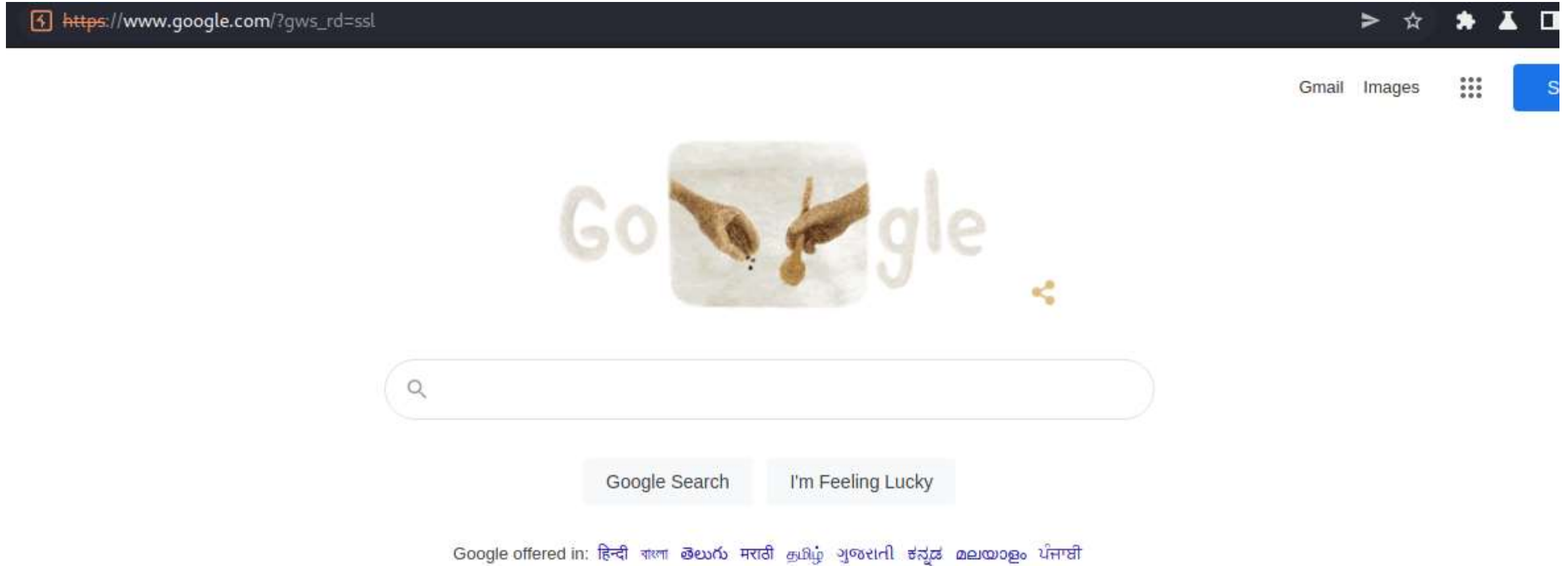
- Let's replace the URL with <http://www.google.com> and see if it redirects to google.com.



You will be redirected in 3 seconds

Proof of Concept (PoC):

- We have been redirected to the specified URL.



Business Impact – Extremely High

- The attacker can redirect the user to download malicious files.
- The attacker can use phishing pages to trick the users into providing critical information.
- The attacker can deface the website.

Recommendation

- Use POST method instead of GET method.
- Don't allow users to change the destination of Redirection.
- Make sure to check for the origin of the request.
- Don't allow any other URL modifications.

References:

- https://en.wikipedia.org/wiki/URL_redirection
- [https://portswigger.net/kb/issues/00500100_open-redirection-reflected#:~:text=Description%3A%20Open%20redirection%20\(reflected\),to%20an%20arbitrary%20external%20domain](https://portswigger.net/kb/issues/00500100_open-redirection-reflected#:~:text=Description%3A%20Open%20redirection%20(reflected),to%20an%20arbitrary%20external%20domain)

16. Command Execution Vulnerability

BruteForce
Command
Execution
Vulnerability
(Critical)

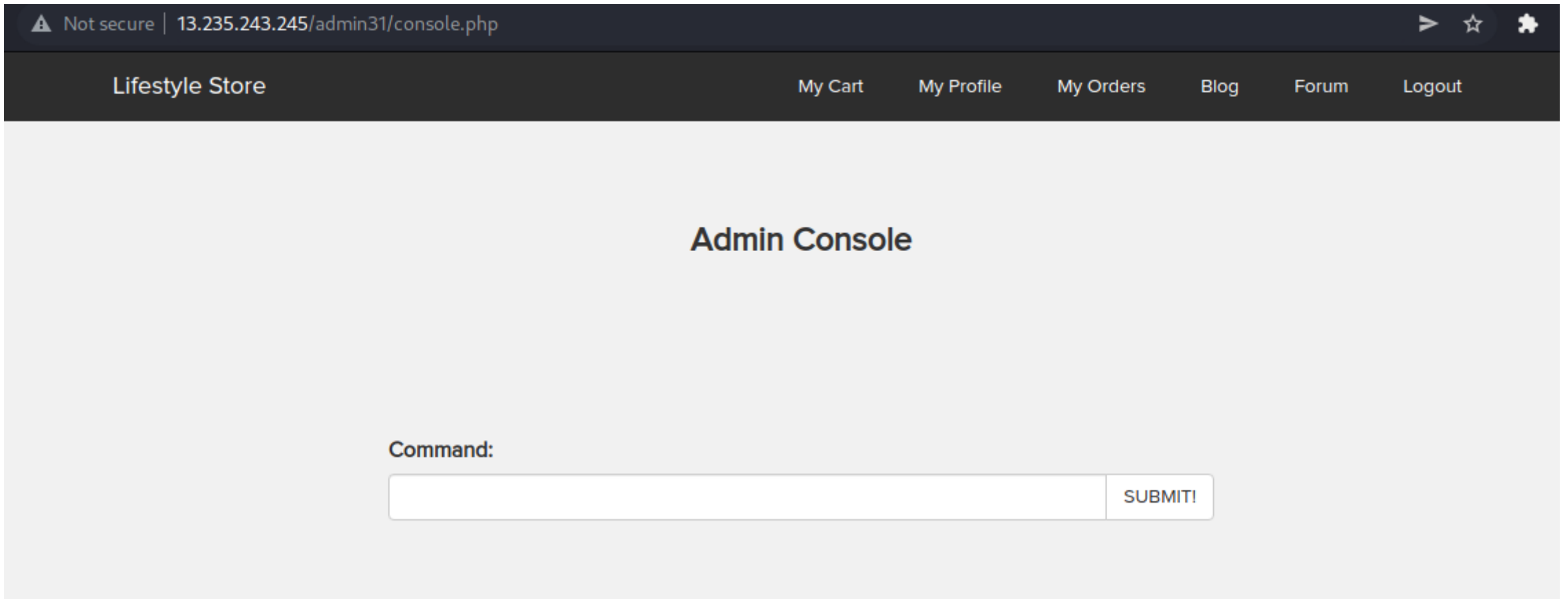
Below mentioned URL allows users to run commands as admin.

Affected URL :

- <http://13.235.243.245/admin31/console>

Observation:

- Visit the specified URL (Must be logged in as a customer).
- You can see a field to enter console commands.



The screenshot shows a web browser window with the address bar displaying "Not secure | 13.235.243.245/admin31/console.php". The browser's navigation bar includes icons for back, star, and settings. Below the browser window is a dark navigation bar for the "Lifestyle Store" website, featuring links for "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout". The main content area has a light gray background and is titled "Admin Console" in bold. At the bottom of this area, there is a "Command:" label, a text input field, and a "SUBMIT!" button.

Not secure | 13.235.243.245/admin31/console.php

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

Admin Console

Command:

SUBMIT!

Observation:

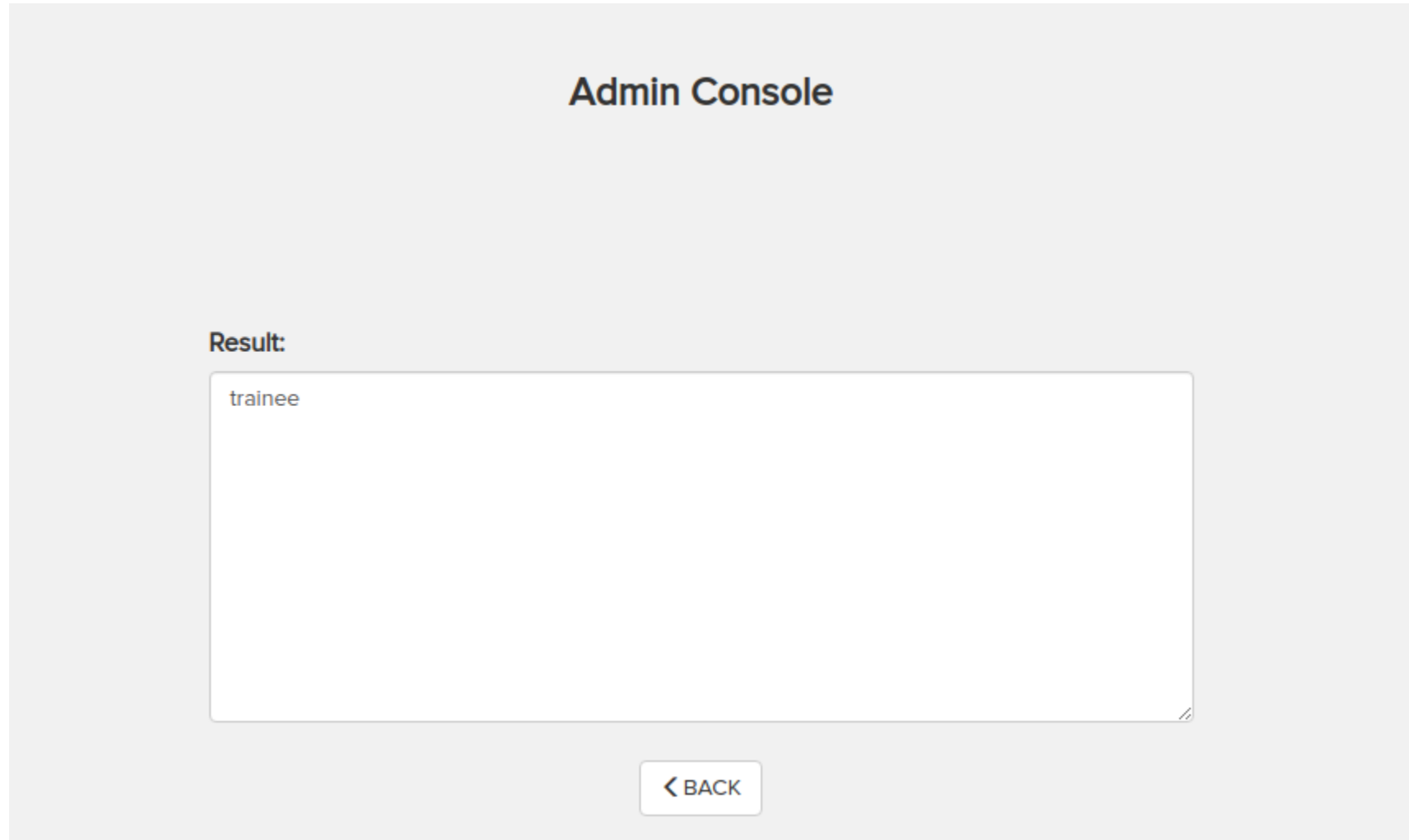
- Let's run the 'whoami' command to check if our commands work.

Admin Console

Command:

Proof Of Concept (PoC):

- The result has been printed. This proves that this page has Command Execution Vulnerability.



Business Impact – Extremely High

- Any user will be able to run commands as the admin.
- The attacker may create a dummy account, access the console and compromise the server.

Recommendation

- Make sure that the console is only accessible by the Admin and not by anyone else.
- Don't let the users access the console.
- Don't leave the console on admin privileges.

References:

- https://en.wikipedia.org/wiki/Arbitrary_code_execution#:~:text=In%20computer%20security%2C%20arbitrary%20code,hardware%20allowing%20arbitrary%20code%20execution.

17. Forced Browsing

Forced
Browsing
Vulnerability
(Critical)

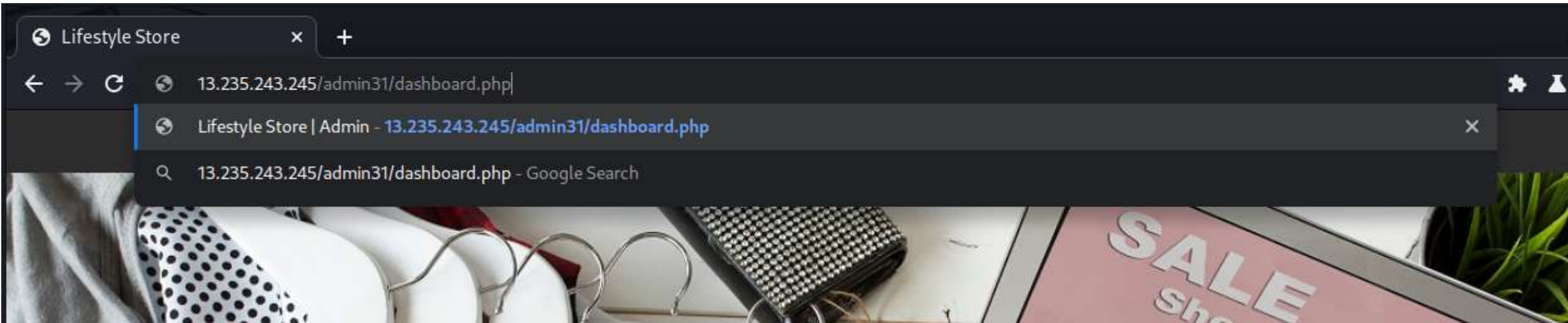
Users can access the admin dashboard by just entering the right URL.

Affected URL :

- <http://13.235.243.245/admin31/dashboard.php>

Observation:

- Enter the provided URL and press enter.
- Note: You must be logged in as a customer before doing this or else it won't work.



Proof Of Concept (PoC):

- We were able to successfully access the admin dashboard. Here, we can add, modify or delete items with admin privileges. We can also access the admin command.

Lifestyle Store

My CartMy ProfileMy OrdersBlogForumLogout

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 |

Business Impact – Extremely High

- Any user can access the admin dashboard and delete all the data.
- The attacker will be able to delete or modify all data in order to deface the website.
- The attacker will be able to run commands as the admin.
- The attacker may create a dummy account, access the console and compromise the server.

Recommendation

- Set up proper authentication and authorization checks on every step.
- Only allow the admin to access the dashboard.
- Ask for confirmation before every critical step like deleting a product, etc.
- Make sure that the console is only accessible by the Admin and not by anyone else.
- Don't leave the console on admin privileges.
- Display a forbidden error when an unauthorized user tries to access the admin dashboard.

References:

- <https://www.geeksforgeeks.org/forced-browsing-ethical-hacking/#:~:text=A%20Forced%20browsing%20attack%20is,level%20for%20the%20same%20user>
- https://owasp.org/www-community/attacks/Forced_browsing
- https://en.wikipedia.org/wiki/Directory_traversal_attack

18. Descriptive Error Messages

Descriptive
Error
messages
(Low)

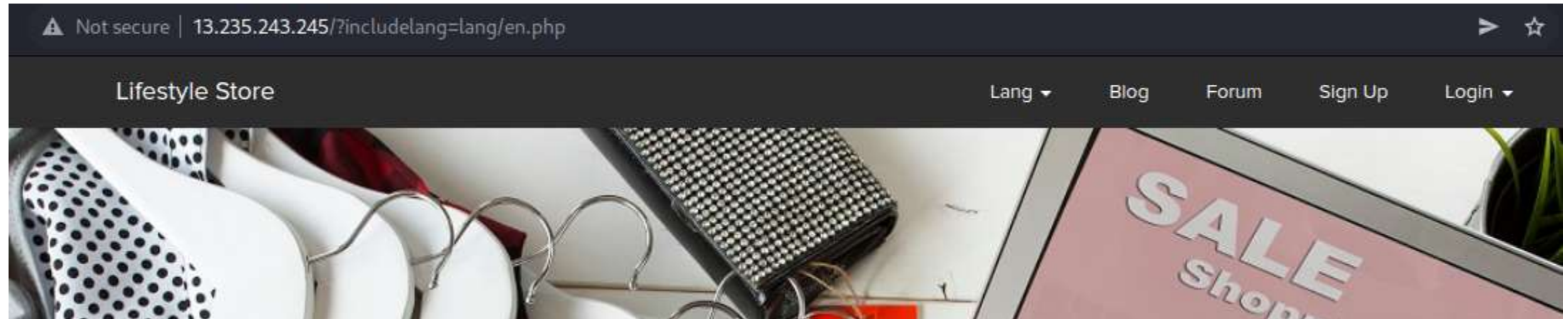
The below mentioned URL gives debug information.

Affected URL :

- <http://13.235.243.245/?includelang=lang/en.php>

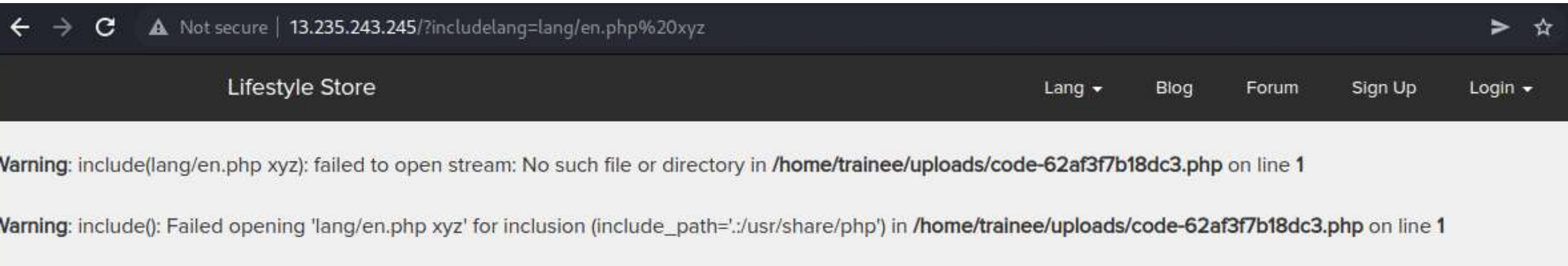
Observation:

- Change the language and notice the URL.



Proof Of Concept (PoC):

- Tampering with the URL gives us debug information.
- We can see the path to a .php file.



Business Impact – low

- Though this is not a big vulnerability and does not disclose critical information, the attacker may use this information to map out the structure of the system.

Recommendation

- Remove the default pages.
- Don't display debug information or warnings to the user.

References:

- https://owasp.org/www-community/Improper_Error_Handling

19. Cross Site Request Forgery (CSRF)

Cross Site
Request
Forgery
(Severe)

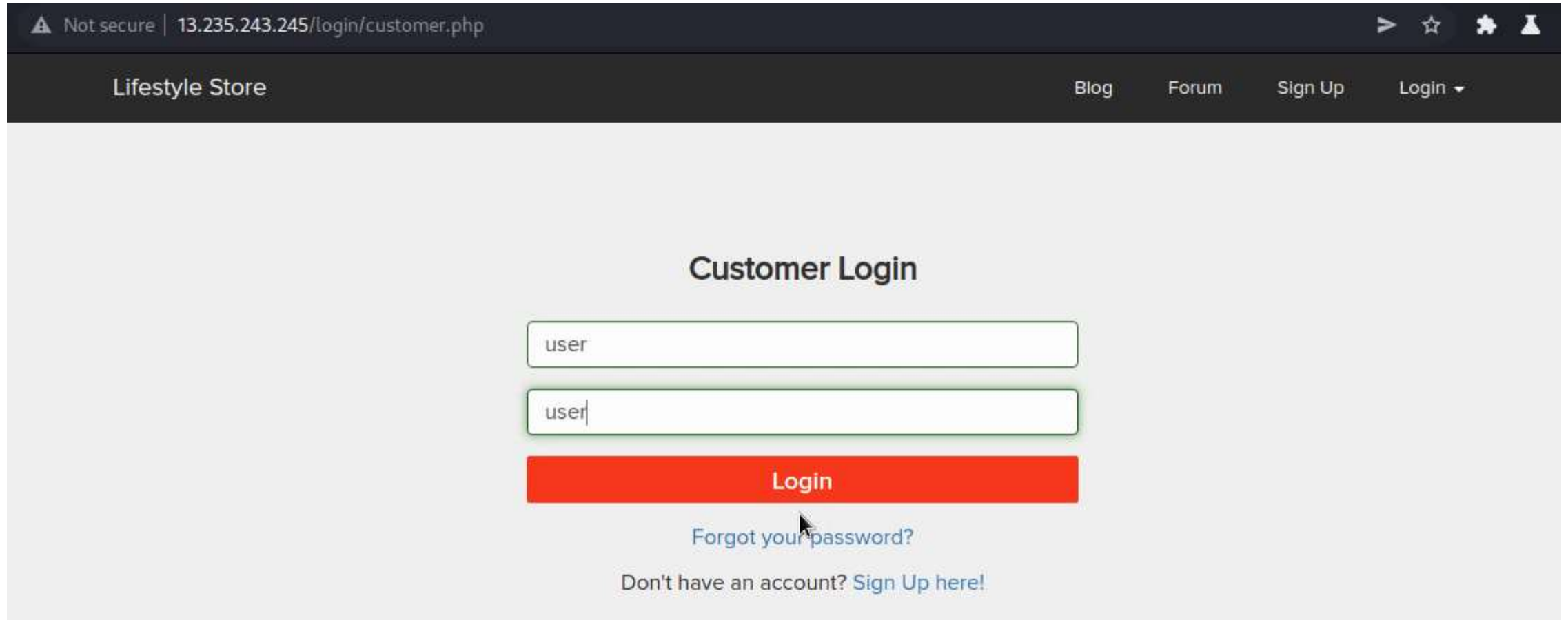
CSRF possible in the below URL.

Affected URL :

- http://13.235.243.245/profile/change_password.php
- I have attached a video named 'CSRF.mp4'
- If you cannot understand the below mentioned process, kindly go through the video.

Observation:

- I have created an account with 'user' as username and password.
- First, I will login using using that account.



The screenshot shows a web browser window with the address bar displaying 'Not secure | 13.235.243.245/login/customer.php'. The page header includes the 'Lifestyle Store' logo and navigation links for 'Blog', 'Forum', 'Sign Up', and 'Login'. The main content area is titled 'Customer Login' and features two input fields, both containing the text 'user'. Below the fields is a red 'Login' button. At the bottom, there are links for 'Forgot your password?' and 'Don't have an account? Sign Up here!'.

Not secure | 13.235.243.245/login/customer.php

Lifestyle Store

Blog Forum Sign Up Login

Customer Login

user

user

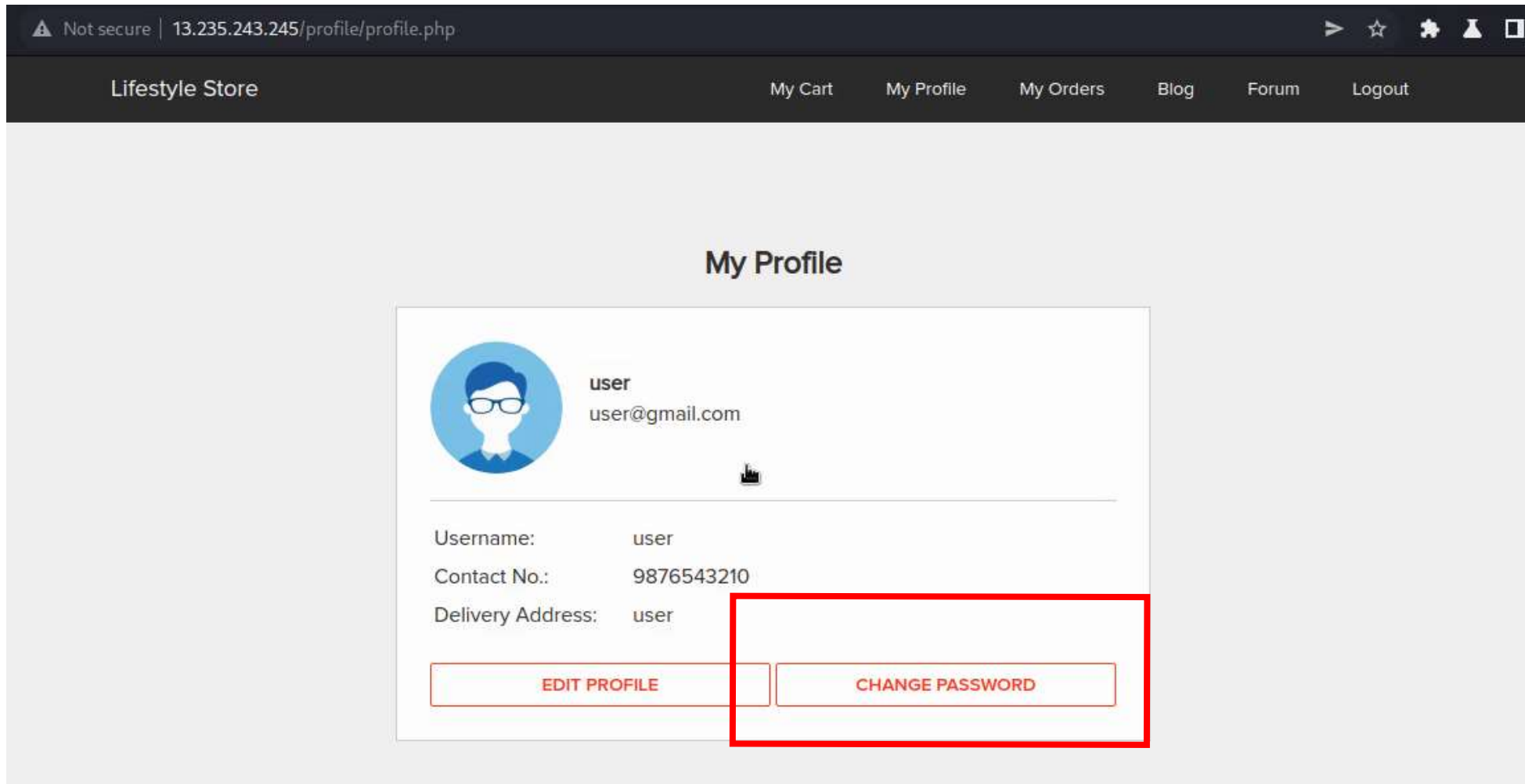
Login

[Forgot your password?](#)

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

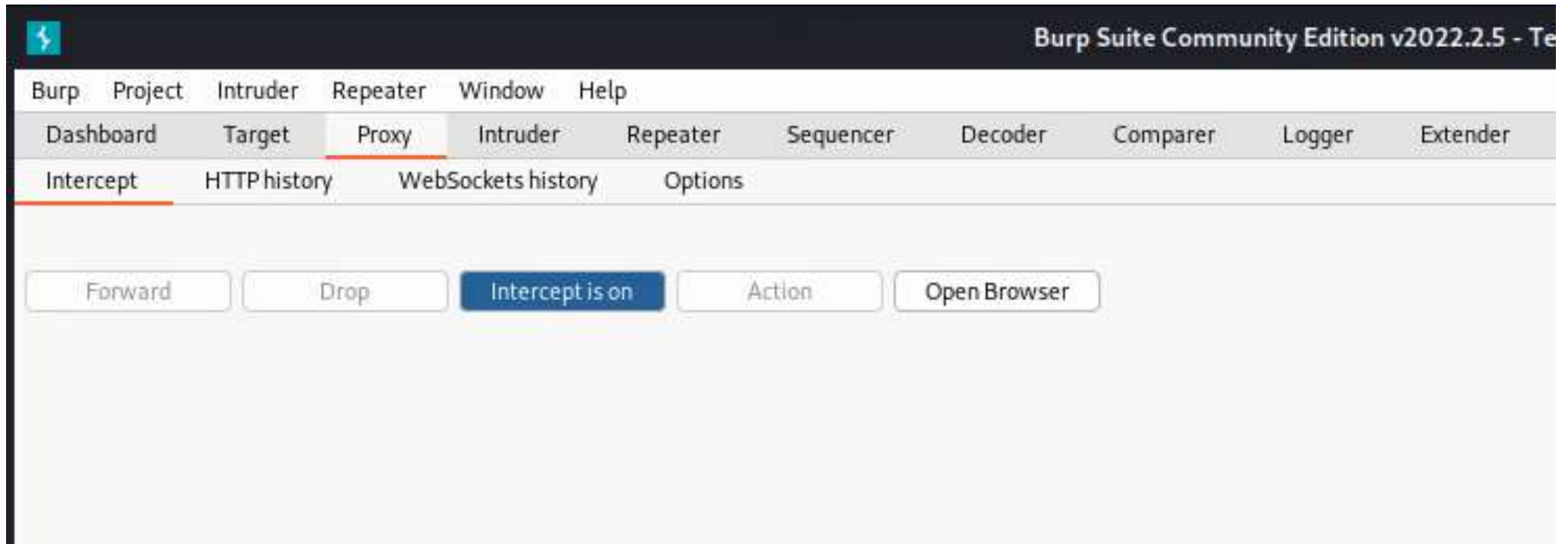
Observation:

- Click on change password Button.



Observation:

- Turn the intercept on.



Observation:

- I have entered 'user' as password in both the fields. Now intercept this request.

45/profile/change_password.php

My Cart My Profile My Orders Blog

Change Password

user

....

UPDATE

Observation:

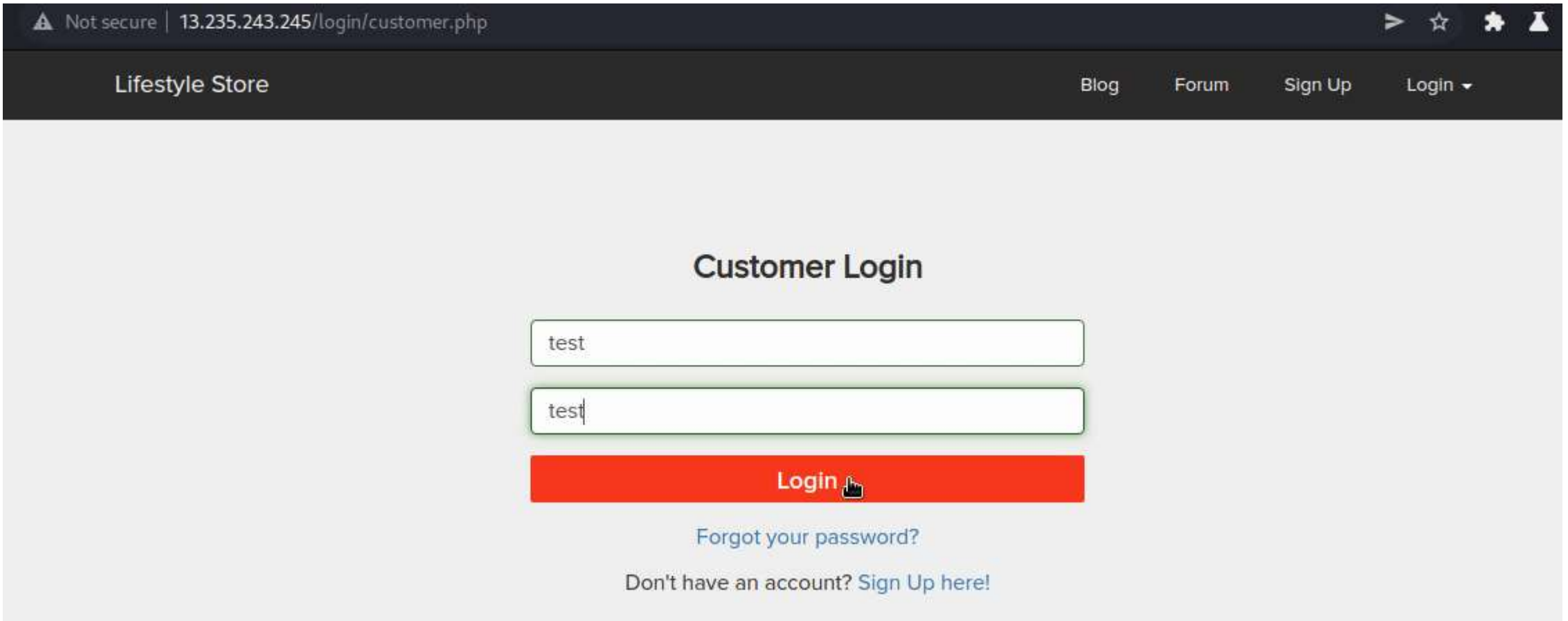
- Send the request to Repeater. Next, delete the Origin and Referer entries from the request.

The screenshot shows the Burp Suite Repeater interface. The top navigation bar includes tabs for Dashboard, Target, Proxy, Intruder, Repeater (selected), Sequencer, Decoder, Comparer, Logger, Extender, Project options, and User options. Below the navigation bar, there is a 'Send' button and navigation controls. The main area is divided into 'Request' and 'Response' sections. The 'Request' section is currently selected and shows a raw HTTP request. The request is a POST to /profile/change_password_submit.php. Headers include Host, Content-Length, Accept, X-Requested-With, User-Agent, and Content-Type. The Origin and Referer headers are highlighted with a red box. The body of the request is a URL-encoded string: password=user&password_confirm=user.

```
1 POST /profile/change_password_submit.php HTTP/1.1
2 Host: 13.235.243.245
3 Content-Length: 35
4 Accept: */*
5 X-Requested-With: XMLHttpRequest
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127
  Safari/537.36
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Origin: http://13.235.243.245
9 Referer: http://13.235.243.245/profile/change_password.php
10 Accept-Encoding: gzip, deflate
11 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
12 Cookie: PHPSESSID=3cf7h3ulud05ah5dc76g1k2r00; key=
  OCEA0C96-F43D-FEF6-9946-044F6B5EBD92; X-XSRF-TOKEN=
  19a7e1552730a92829d121fe46dc4645e849aca14d3f5d333e5217180fd1e617
13 Connection: close
14
15 password=user&password_confirm=user
```

Observation:

- Now logout and login using a different account.
- I am using another account with 'test' as the username and password



The screenshot shows a web browser window with the address bar displaying 'Not secure | 13.235.243.245/login/customer.php'. The page header includes the 'Lifestyle Store' logo and navigation links for 'Blog', 'Forum', 'Sign Up', and 'Login'. The main content area is titled 'Customer Login' and features two input fields, both containing the text 'test'. Below the fields is a red 'Login' button with a hand cursor icon. At the bottom, there are links for 'Forgot your password?' and 'Don't have an account? Sign Up here!'.

Not secure | 13.235.243.245/login/customer.php

Lifestyle Store

Blog Forum Sign Up Login

Customer Login

test

test

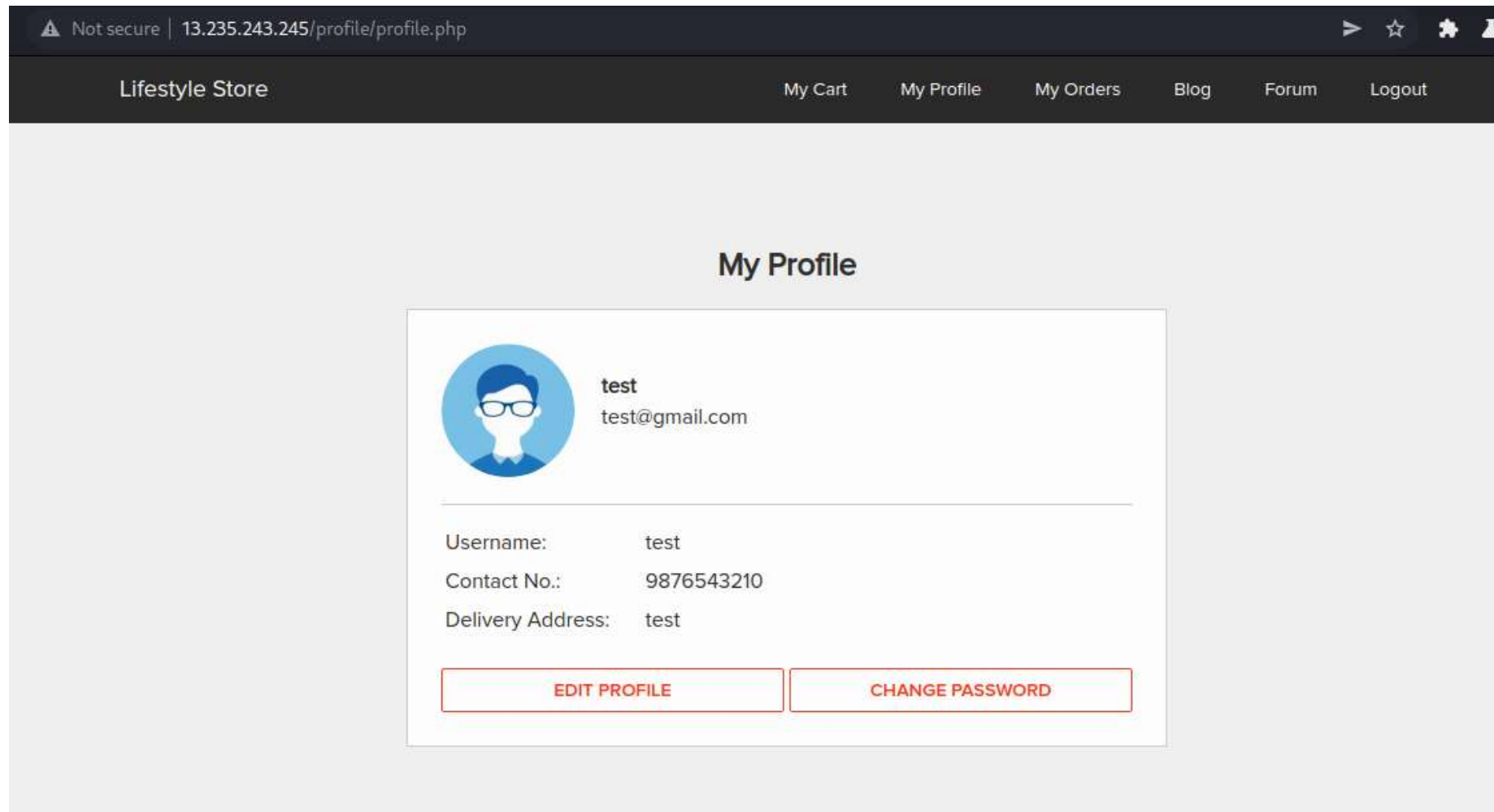
Login

[Forgot your password?](#)

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

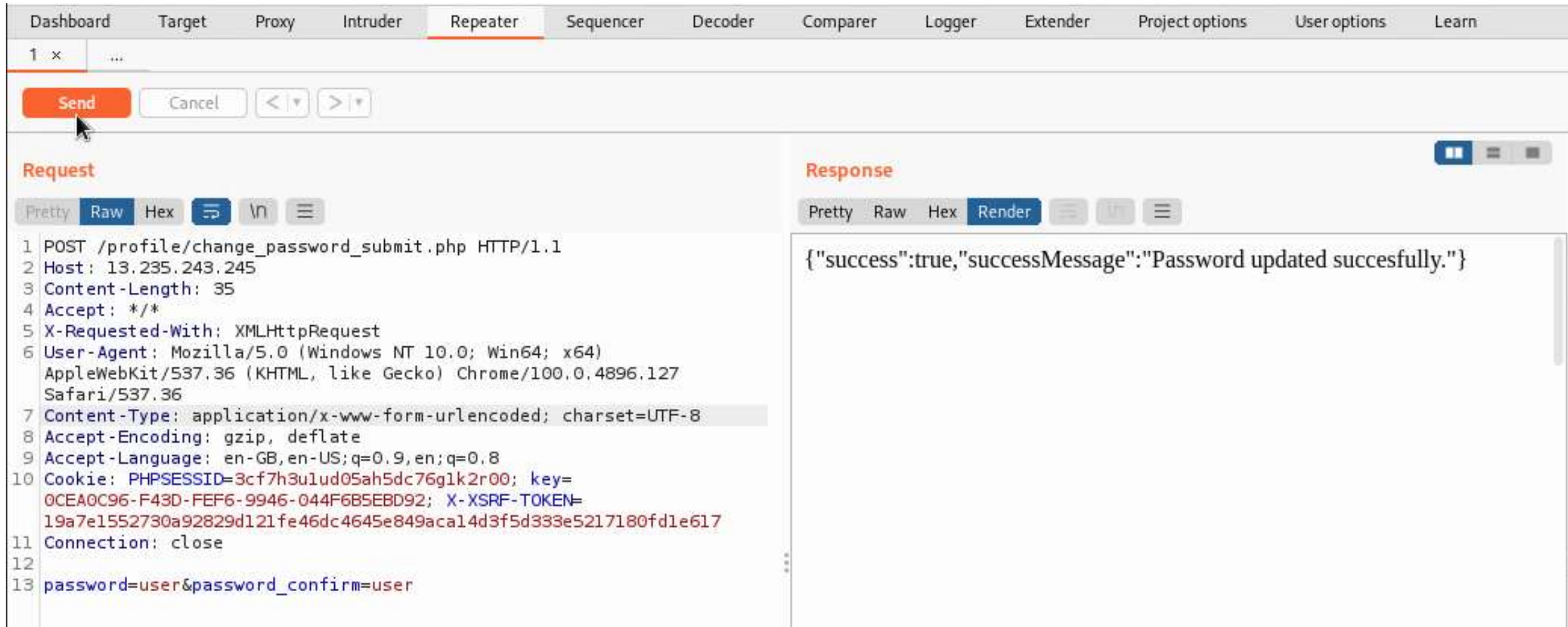
Observation:

- We have successfully logged in.



Observation:

- Go back to the repeater and repeat the request.
- Check the response, It was successful.






The screenshot displays the Burp Suite interface, specifically the Repeater tab. The top navigation bar includes options like Dashboard, Target, Proxy, Intruder, Repeater (selected), Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, and Learn. Below the navigation bar, there's a toolbar with a 'Send' button (highlighted by a mouse cursor), a 'Cancel' button, and navigation arrows. The main area is split into two panels: 'Request' on the left and 'Response' on the right. The 'Request' panel shows an HTTP POST request to '/profile/change_password_submit.php' with various headers and a body containing 'password=user&password_confirm=user'. The 'Response' panel shows a successful JSON response: '{"success":true,"successMessage":"Password updated succesfully."}'.

Dashboard Target Proxy Intruder **Repeater** Sequencer Decoder Comparer Logger Extender Project options User options Learn

1 x ...




Send Cancel < >

Request

Pretty **Raw** Hex   

```
1 POST /profile/change_password_submit.php HTTP/1.1
2 Host: 13.235.243.245
3 Content-Length: 35
4 Accept: */*
5 X-Requested-With: XMLHttpRequest
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127
  Safari/537.36
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
10 Cookie: PHPSESSID=3cf7h3ulud05ah5dc76glk2r00; key=
  OCEA0C96-F43D-FEF6-9946-044F6B5EBD92; X-XSRF-TOKEN=
  19a7e1552730a92829d121fe46dc4645e849aca14d3f5d333e5217180fd1e617
11 Connection: close
12
13 password=user&password_confirm=user
```

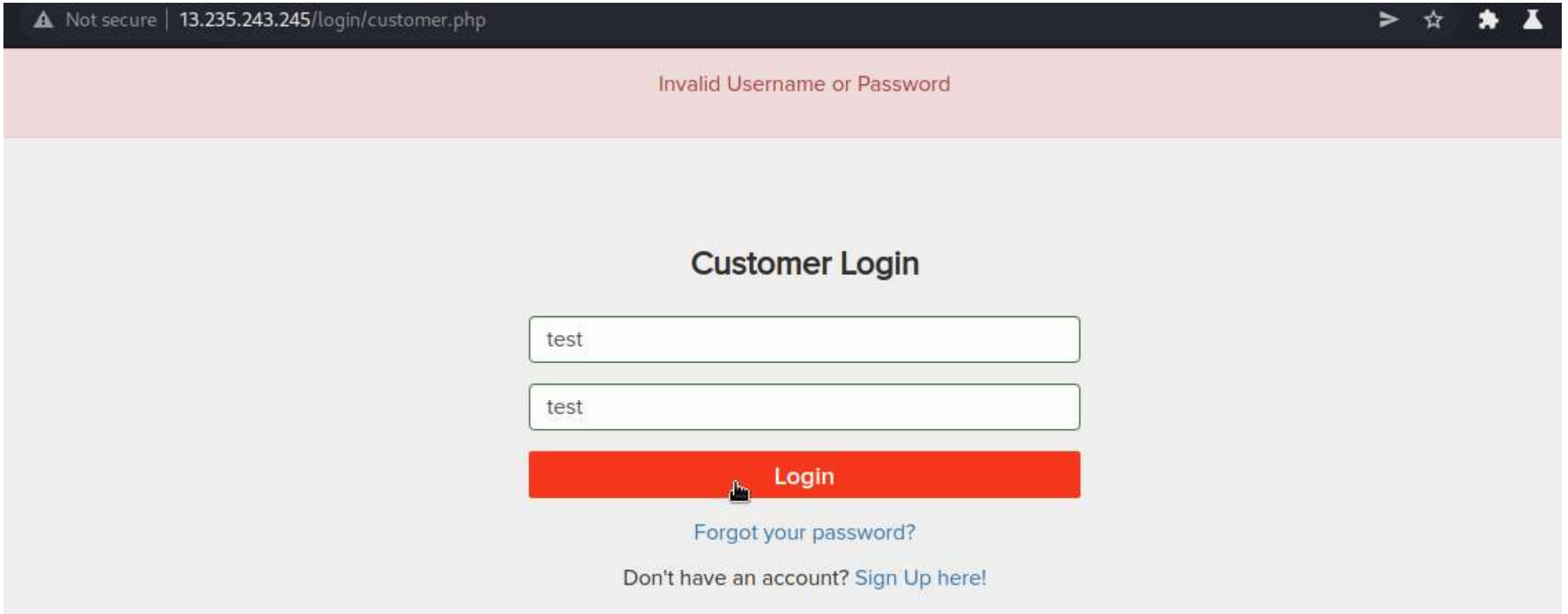
Response

Pretty Raw Hex **Render**   

```
{"success":true,"successMessage":"Password updated succesfully."}
```

Proof Of Concept (PoC):

- Now logout and try to login to the test account using the original credentials.
- It says Invalid username or password.



The screenshot shows a web browser window with the address bar displaying "Not secure | 13.235.243.245/login/customer.php". The page has a light gray background. At the top, a red banner displays the error message "Invalid Username or Password". Below this, the heading "Customer Login" is centered. There are two input fields, both containing the text "test". Below the input fields is a red "Login" button with a mouse cursor icon. Under the button, there is a link "Forgot your password?" and at the bottom, a link "Don't have an account? Sign Up here!".

Not secure | 13.235.243.245/login/customer.php

Invalid Username or Password

Customer Login

test

test

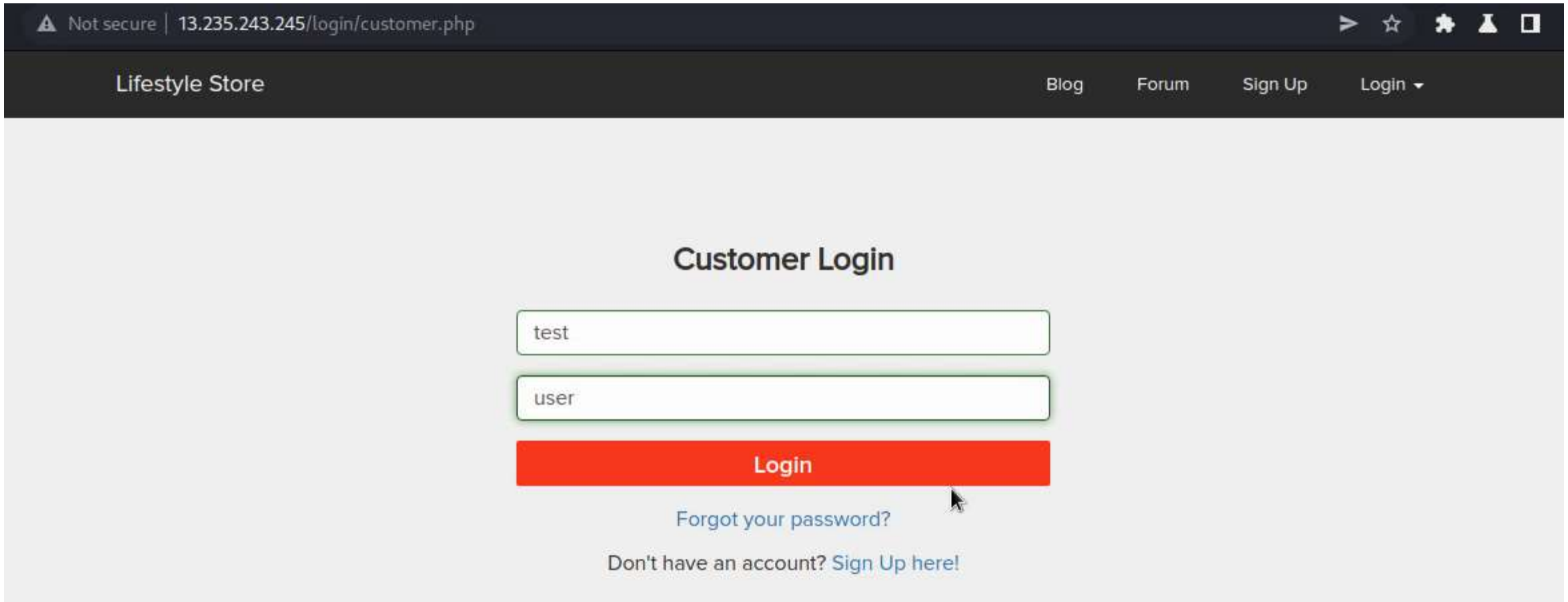
Login

[Forgot your password?](#)

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

Proof Of Concept (PoC):

- Now let's try to login using the password for the attacker's account ('user')



Not secure | 13.235.243.245/login/customer.php

Lifestyle Store Blog Forum Sign Up Login ▾

Customer Login

test

user

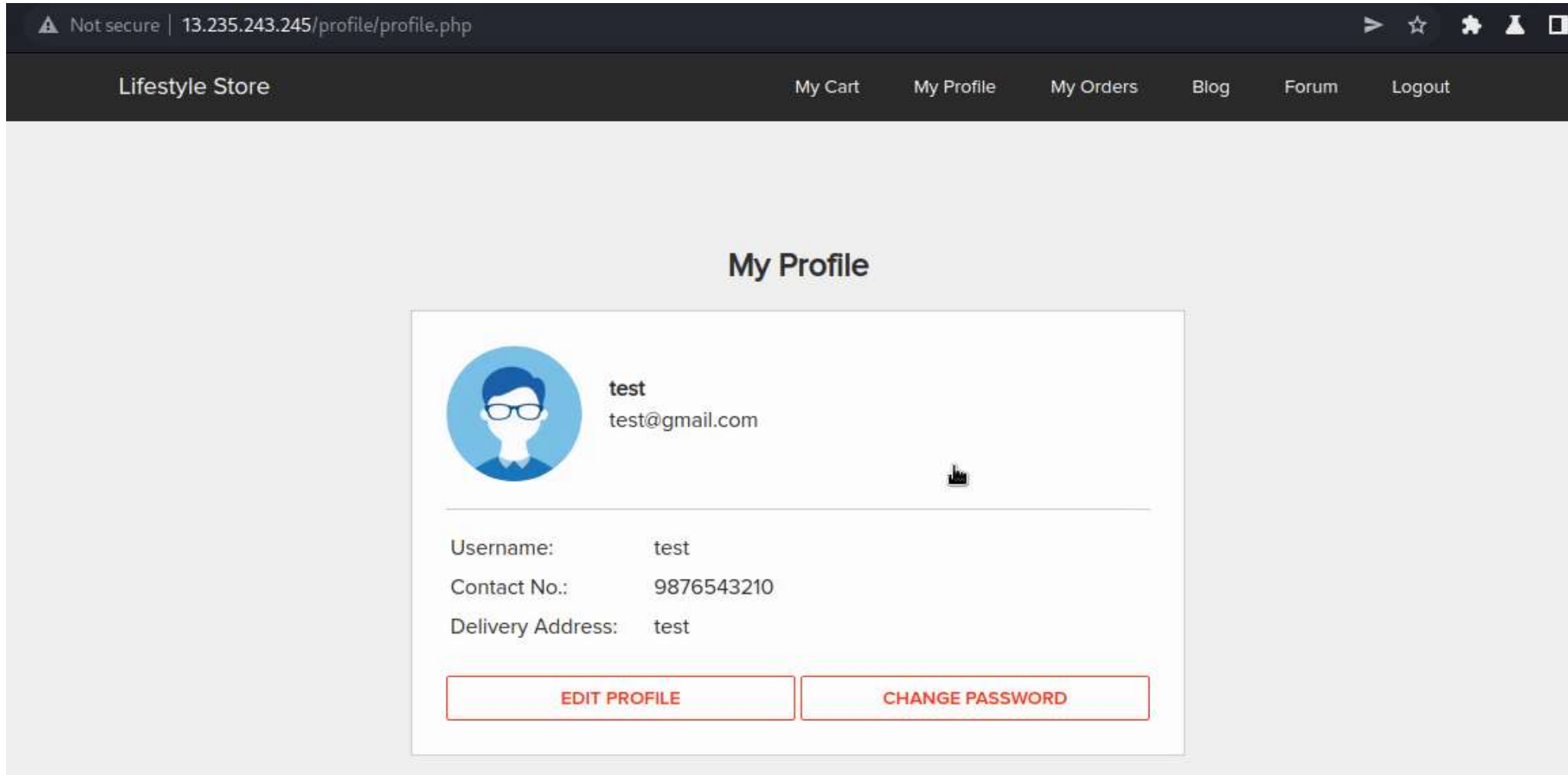
Login

[Forgot your password?](#)

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

Proof Of Concept (PoC):

- We were able to successfully login using the attacker's password.



Business Impact – High

- The attacker can change the password of other users without their knowledge.
- He will be able to use the accounts of the compromised users.
- The victims will lose their personal information to the attacker.

Recommendation

- Set up proper authentication and authorization checks on every step.
- Check for the origin and referer headers.
- Ask for confirmation before every critical step like deleting a product, etc.
- Assign proper authentication checks on important steps like 'password change', etc.
- Confirm with the user before changing password so that the user becomes aware of what is going on.

References:

- <https://owasp.org/www-community/attacks/csrf>
- <https://portswigger.net/web-security/csrf>
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

THANK YOU

For any further clarifications/patch assistance,
please contact: 7075008994 / afnaan2180@gmail.com

~ Mohammed Afnaan Ahmed

