# VAPT Report For e-Commerce Platform.

## “Lifestyle Store.” A shopping web application

Report By : Shaswat Manoj Jha

# Introduction :

- This is the vulnerability assessment and penetration testing (VAPT) report for “Lifestyle Store”.
- I have reported all the vulnerabilities found on this store with their screenshots and links.
- I have also provided business risks and expert recommendations to tackle those vulnerabilities.
- This presentation contains a total of 126 slides, which are read only by default, you can opt-in for editing.

Lifestyle Store

# Security Status : Extremely Vulnerable

- **SQLi :** Hackers can steal all records from the databases of the website.
- Hackers can take **complete control** of the website including view, edit, add or delete files and folders via shell upload.
- **IDOR :** Hackers can extract mobile numbers of all customers using user-id.
- Hackers can change the source code and can upload any malicious code, phishing etc. in the website via **shell upload**.
- **XSS :** Hackers can trick users to click on malicious pop up links and steal information  via cross-site-scripting.

# “ Vulnerability Statistics :

| Critical | Severe | Moderate | Low |
|---|---|---|---|
| 12 | 7 | 3 | 3 |

# Vulnerabilities Found :

| NO | SEVERITY | VULNERABILITY | COUNT |
|---|---|---|---|
| 1 | CRITICAL | SQL INJECTION | 2 |
| 2 | SEVERE | STORED AND REFLECTED XSS | 3 |
| 3 | CRITICAL | IDOR | 4 |
| 4 | SEVERE | RATE LIMITING FLAW | 2 |
| 5 | CRITICAL | INSECURE FILE UPLOAD | 2 |
| 6 | MODERATE | CLIENT SIDE FILTER BYPASS | 2 |
| 7 | SEVERE | SERVER MISCONFIGURATION | 2 |
| 8 | CRITICAL | PII LEAKAGE | 4 |
| 9 | CRITICAL | OPEN REDIRECTION | 4 |
| 10 | SEVERE | CSRF | 3 |

# 1.
# SQL Injection
(CRITICAL)

# Below mentioned URL is vulnerable to SQL injection attack.

- **Relevant URL :**
  - http://52.66.203.250/products.php?cat=1
- **Affected Parameters** :
  - category (GET)
- **Payload** :
  - cat=1’

# Other Similar URLs Vulnerable To SQLi :

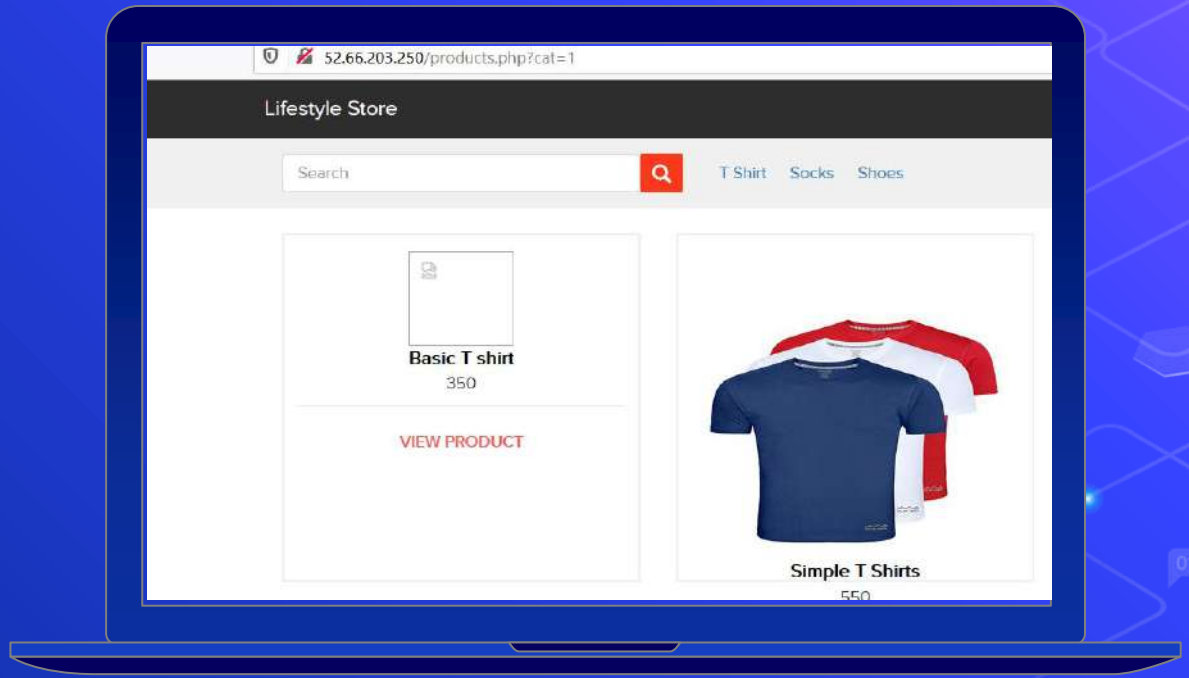
- **URL #1 :**
  - http://52.66.203.250/products.php?cat=2’
- **URL #2 :**
  - http://52.66.203.250/products.php?cat=3’
- **PAYLOAD :**
  - cat=1’

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

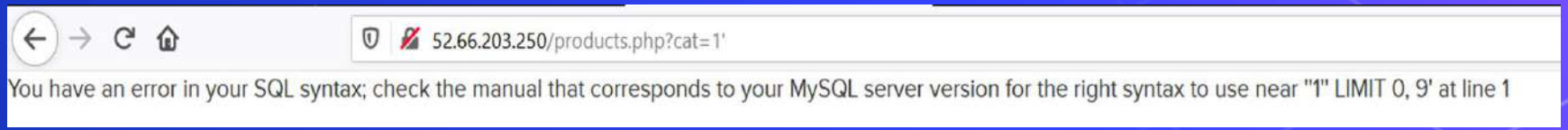
# #1

- Go to the categories option from the home page of the website and click on “T-Shirt”, “Socks” or “Shoes”, to get into the URL, you will see products as per the option you have chosen but notice the get parameter in the URL.



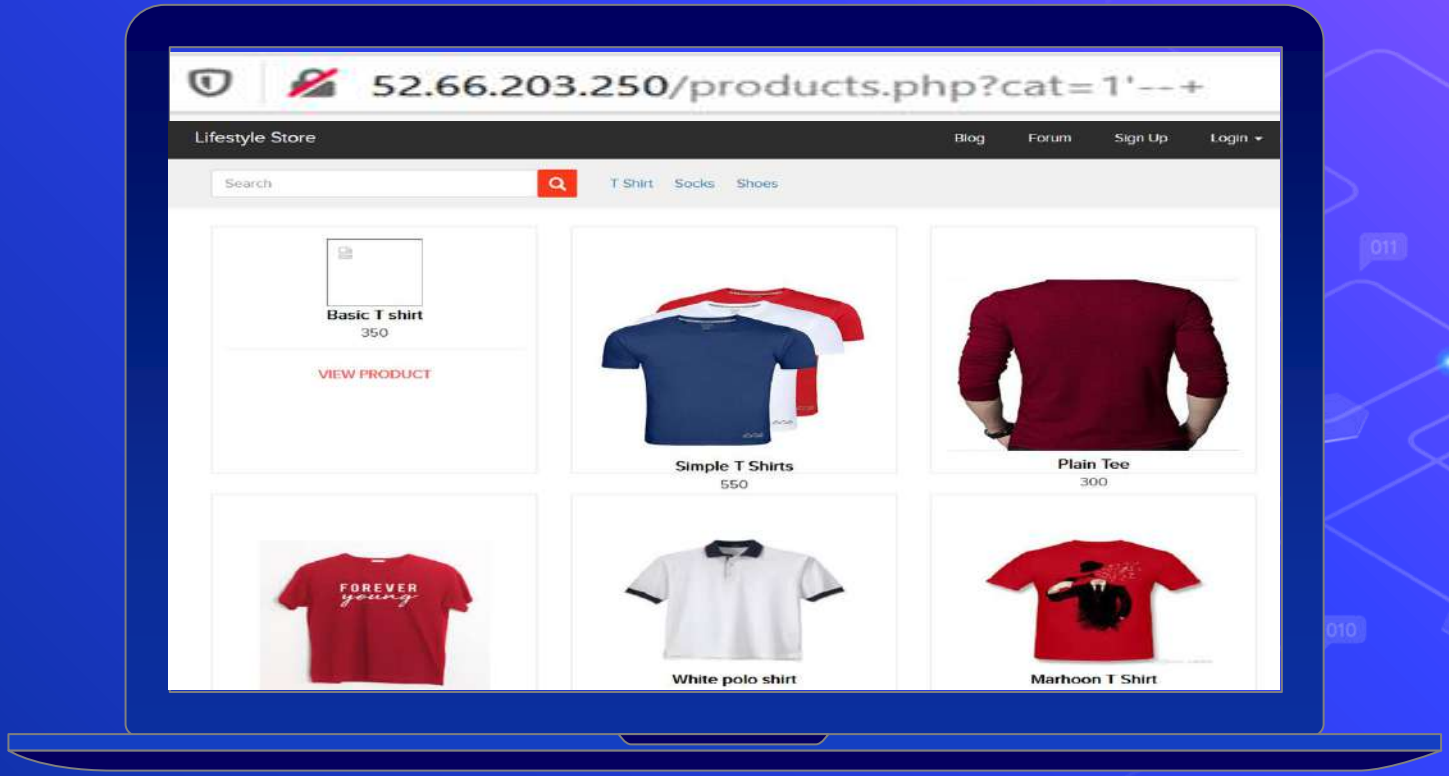
# #2

- When you apply a single quote (‘) in the get parameter, you get a MySQL error.



# #3

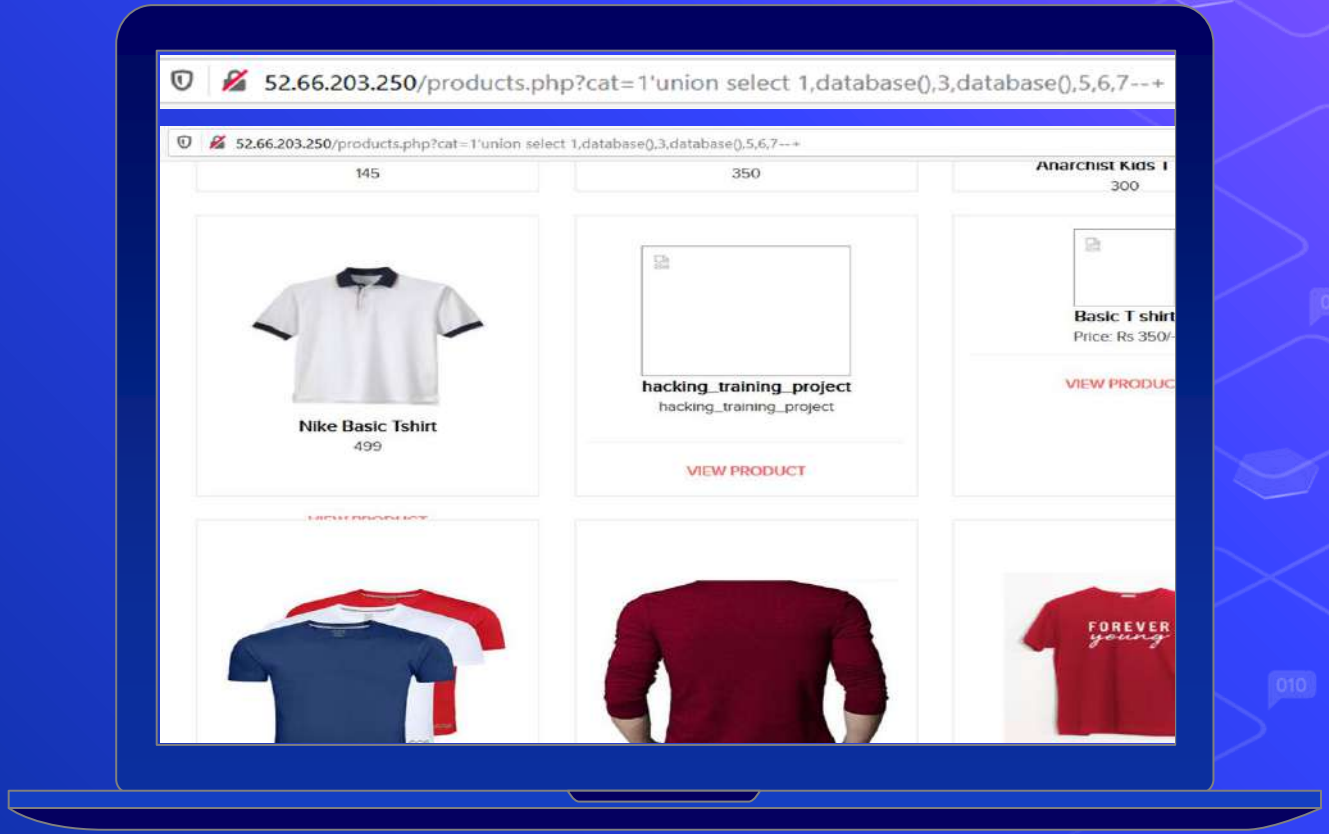
- To confirm the MySQL error, you can put “--+” in the end of the parameter, and the page loads in its original format, this confirms the error.



# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

- Attacker can execute commands as shown below to get critical information, here we have used the payload to find the name of the database of MySQL.



# Proof Of Concept :

- **No. Of Databases : 2**
  - Information_Schema
  - Hacking_training_project

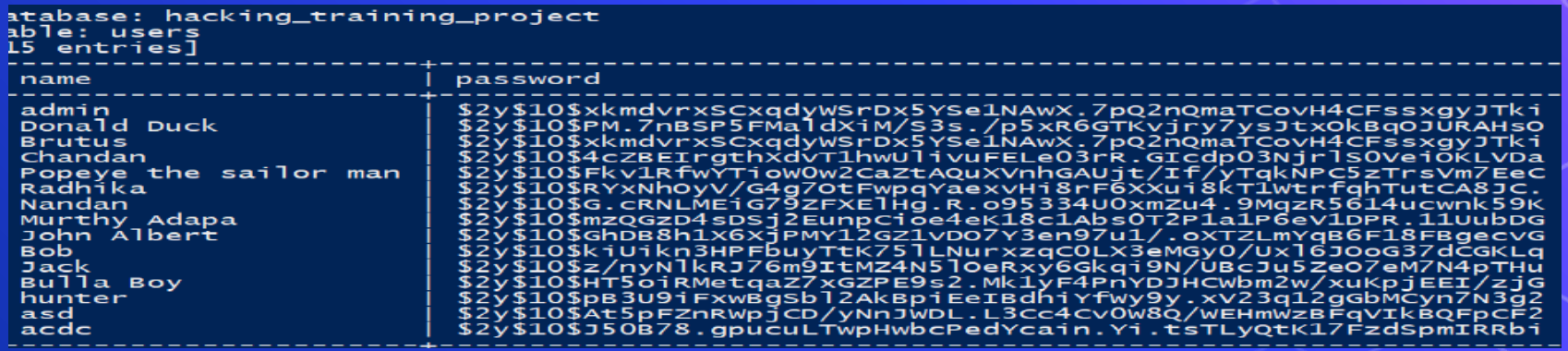
- **No. Of Tables in “Hacking_training_Project” : 10**
  - order_items
  - brands
  - cart_items
  - categories
  - customers
  - orders
  - product_reviews
  - products
  - users
  - sellers

# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

Using this vulnerability, the attacker can execute arbitrary SQL commands on Lifestyle Store's server and gain complete access to the internal database, which also makes all customer data inside it visible.
Below is the screenshot of users table which shows user credentials being leaked.



```
atabase: hacking_training_project
able: users
15 entries]
+----------------------+--------------------------------------------------------------+
| name                 | password                                                     |
+----------------------+--------------------------------------------------------------+
| admin                | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki |
| Donald Duck          | $2y$10$PM.7nBSP5FMa1dXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO |
| Brutus               | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki |
| Chandan              | $2y$10$4cZBEIrgthXdVT1hwUlivuFELeO3rR.GIcdpO3NjrlsOveiOKLVDa |
| Popeye the sailor man | $2y$10$FkvlRfwYTioWOw2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC |
| Radhika              | $2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC. |
| Nandan               | $2y$10$G.cRNLMEiG79ZFXE1Hg.R.o95334UOxmZu4.9MqzR5614ucwnk59K |
| Murthy Adapa         | $2y$10$mzQGzD4sDSj2EunpCioe4eK18c1AbsOT2P1a1P6eV1DPR.11UubDG |
| John Albert          | $2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG |
| Bob                  | $2y$10$kiUikn3HPFbuyTtK75lLNurxzqCOLX3eMGyO/Ux16JOoG37dCGKLq |
| Jack                 | $2y$10$z/nyNlkRJ76m9ItMZ4N5loeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu |
| Bulla Boy            | $2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG |
| hunter               | $2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfwy9y.xV23q12gGbMCyn7N3g2 |
| asd                  | $2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4CvOW8Q/wEHmWzBFqVIkBQFpCF2 |
| acdc                 | $2y$10$J5OB78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi |
+----------------------+--------------------------------------------------------------+
```

The data above can be decoded using Burp Suite, the attacker can decode and use this information to login to admin panel and gain complete admin level access to the website which could lead to a complete compromise of the server and all other servers connected to it.

# Recommendations :

Take the following recommendations to secure and avoid SQL database exploitation.

# Recommendations :

1. Prepared Statements : Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query.

2. Do not run Database Service as admin/root user.

3. Disable/remove default accounts, passwords and databases .

4. Assign each Database user only the required permissions and not all permissions.

5. Character Encoding : Convert the simple codes to different characters like '~'/'^'. It is also suggested to follow html and other encodings.

# References :

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

# 2. Account Takeover Via OTP Bypass

(CRITICAL)

# Below mentioned page has a login form that allows login via OTP that can be brute forced.

- **Relevant URL :**
  - http://52.66.203.250/login/admin.php

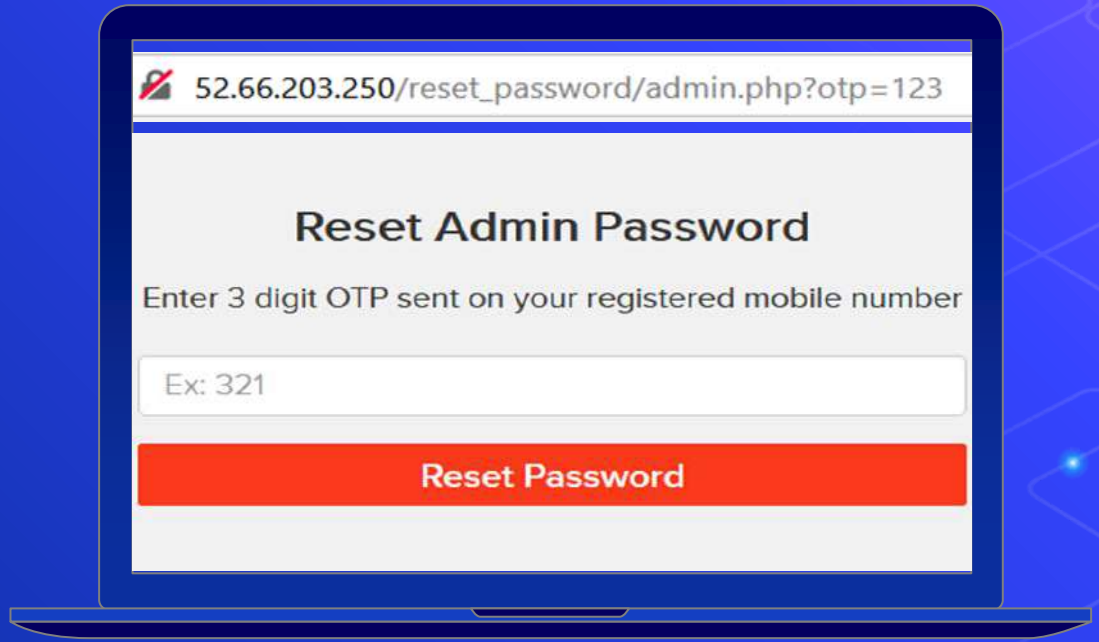
- **Affected Parameters :**
  - OTP (POST)

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

# #1

- Navigate to http://52.66.203.250/login/admin.php you will see a “forgot password” hyperlink which asks for OPT which is sent to victim’s mobile number ,input any anonymous 3 digit OPT and intercept the request with burp suite.

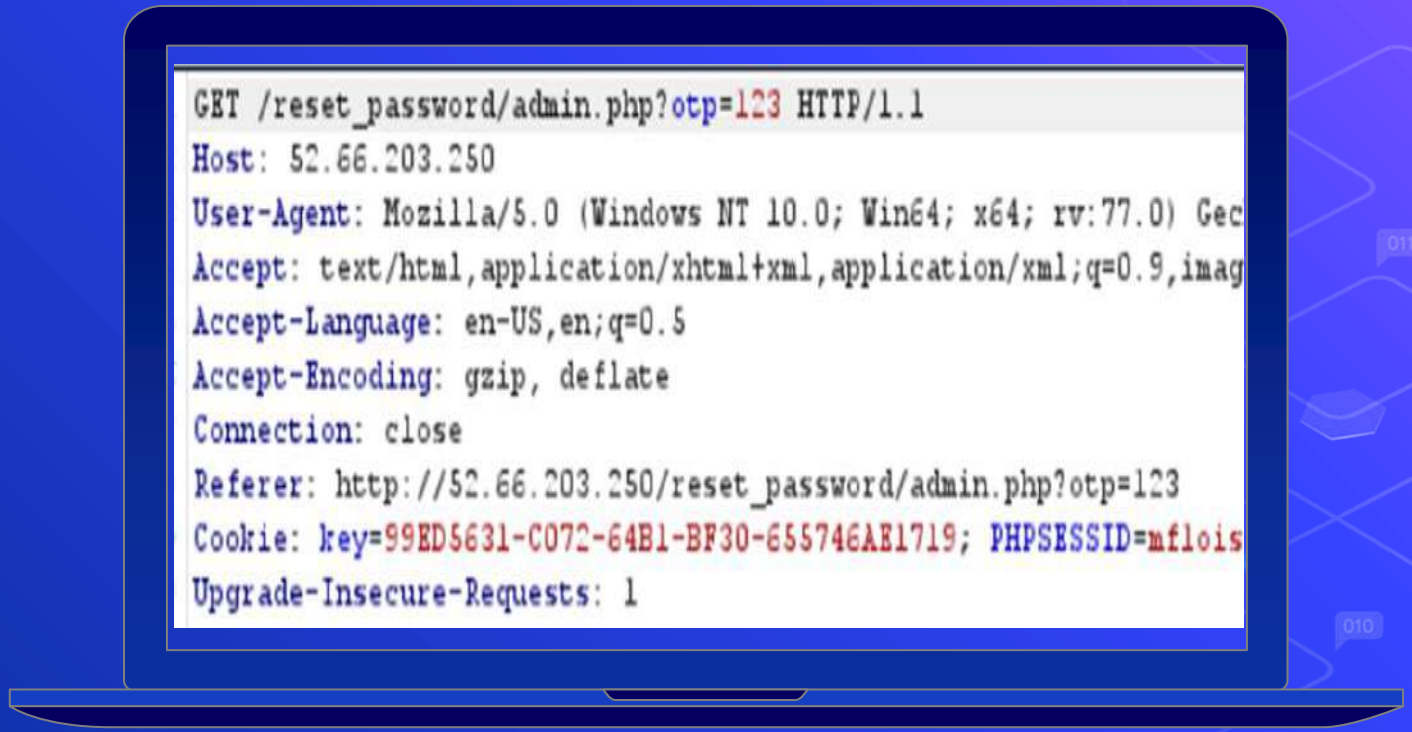


# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

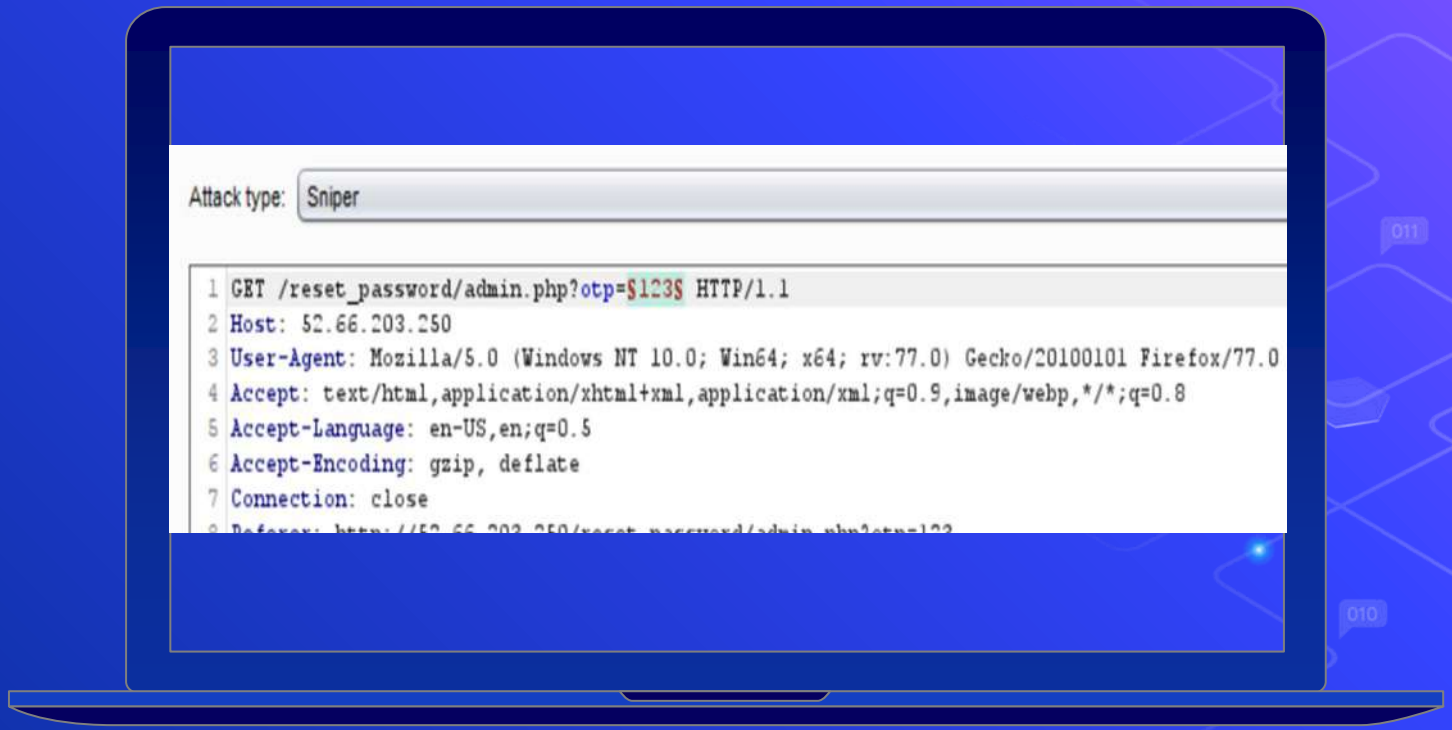
## #2

- The request below will be generated in burp suite using GET parameter.



# #3

- We'll brute force by getting all combination of 3 digit OTP's and input the correct one to bypass.



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- 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 number of trials.

- This leads to complete compromise of personal user data of every customer.

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

Following slide has the recommendations.

# Recommendations :

Take the following recommendations to secure and avoid OTP bypass.

# Recommendations :

Take the following precautions :

1. Use rate limiting checks on the number of times the OTP request is generated.

2. OTP should be at least of 6 digits, a 3 digit OTP can be guessed or brute-forced easily.

3. OTP should expire in a particular time like 2 minutes or 5 minutes to make authorization process more secure.

# References :

https://www.owasp.org/www-project-api-security/

# 3. Unauthorized access to customer details

(CRITICAL)

# The “My Orders” section of the website shows an Insecure Direct Object Reference (IDOR), This allows hackers to get access to any customer’s order details and more.

- **Relevant URL :**
  - http://52.66.203.250/orders/orders.php?customer=2

- **Affected Parameters :**
  - Customer Data (GET)

# Other URL with similar issues.

- **Relevant URL :**
  - http://52.66.203.250/products/details.php?p_id=7.

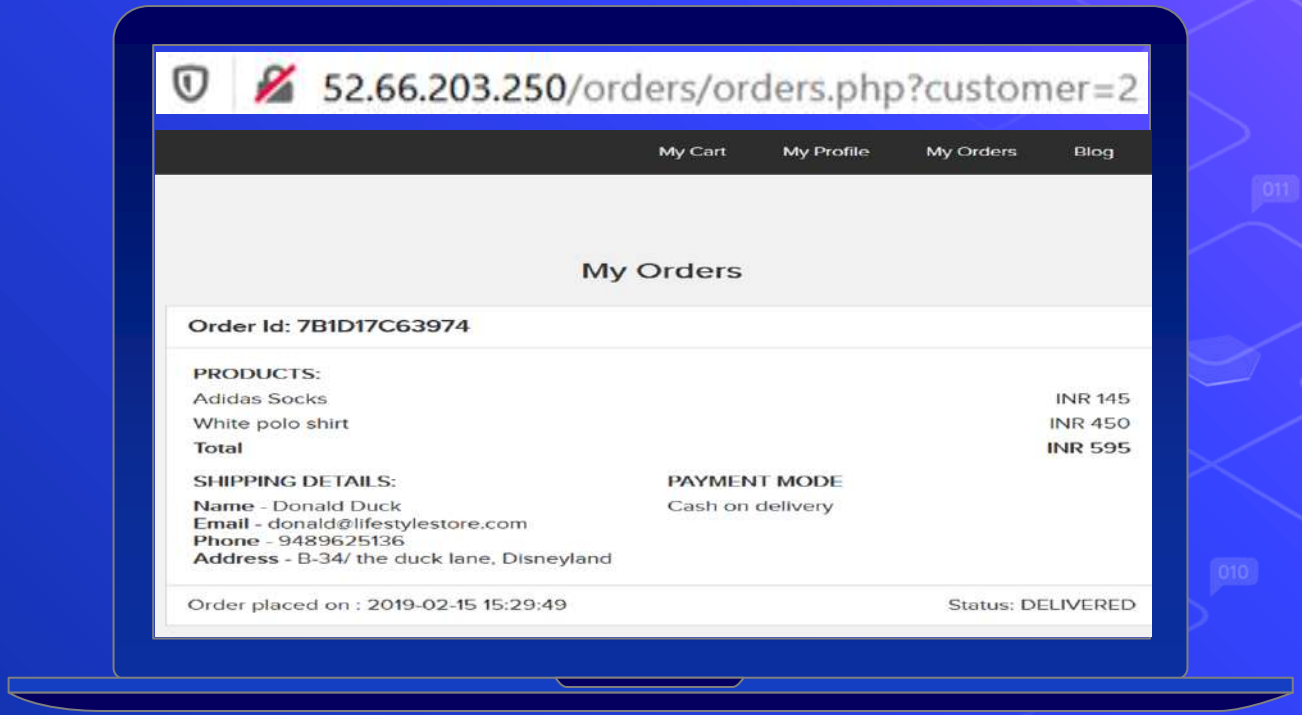
- **Affected Parameters :**
  - P_ID (GET)

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

# #1

- Login to your account and go to “My Orders” section, you’ll see a GET parameter as shown below, “customer=2”.

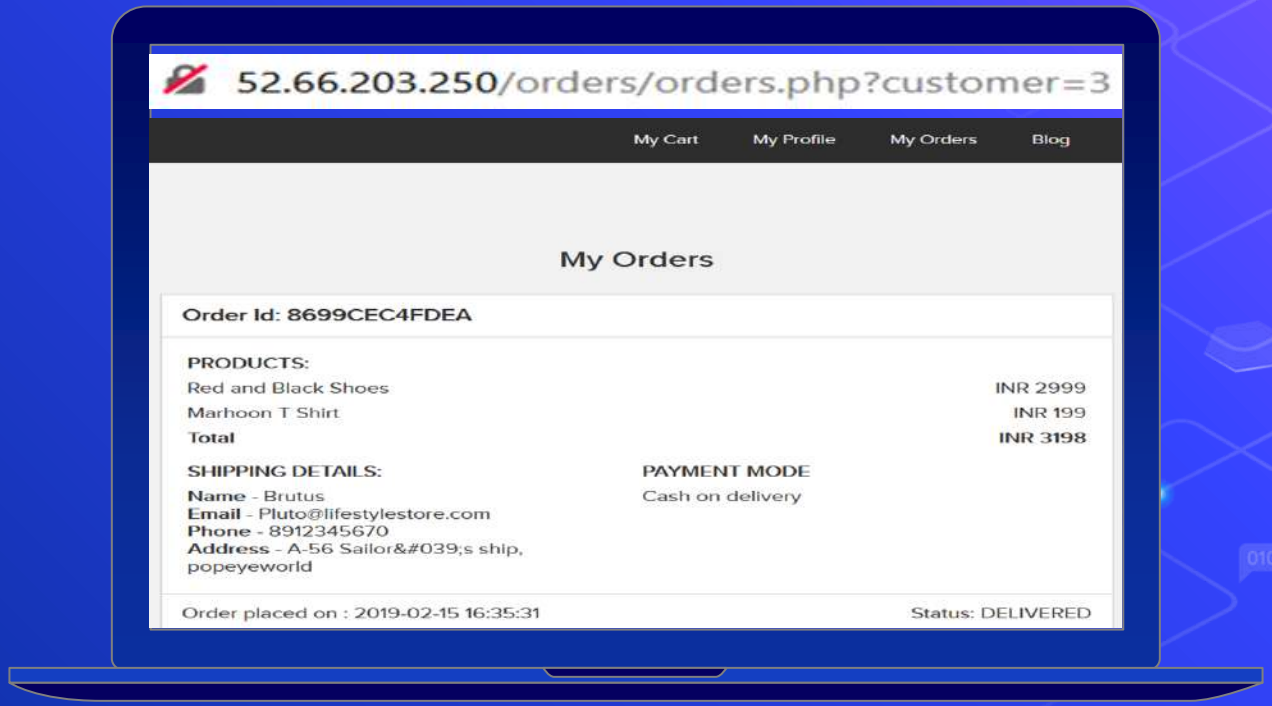


# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

# #2

- Change the customer number to any random number, and we get the details of the new customer ID.



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- This vulnerability can be exploited by malicious hackers to carry out targeted phishing attacks on the users and the information can also be sold to the competitors / black-market.
- More over, as there is no rate limiting checks, attacker can brute-force 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.

Following slide has the recommendations to tackle this extremely high risk.

# Recommendations :

Take the following recommendations to secure and avoid data leakage.

# Recommendations :

Take the following precautions :

1. Make sure user only have the access to self data.

2. Use proper rate limiting checks on the number of request comes from a single user in a small amount of time.

3. Implement proper authentication and authorization checks to make sure that the user has permission to the data he/she is requesting.

# References :

https://owasp.org/www-chapter-ghana/assets/slides/IDOR.pdf
https://portswigger.net/web-security/access-control/idor

# 4. Reflected Cross Site Scripting.

(SEVERE)

# These parameters are vulnerable to XSS.

- **Relevant URL :**
  - 13.234.113.207/products/details.php?p_id=2
- **Affected Parameters :**
  - URL (GET)
- **Payload :**
  - <script>alert(1)</script>

# Other parameter with similar issues.

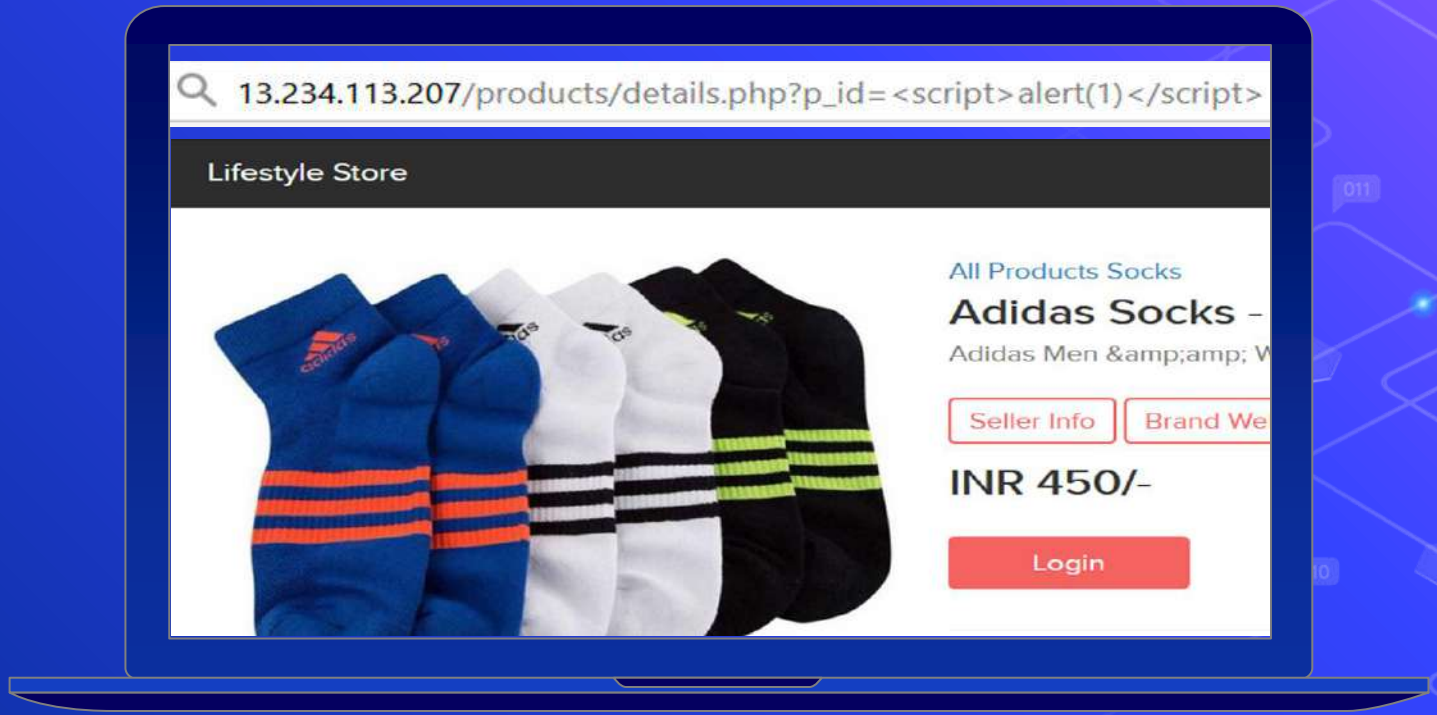
- **Relevant URL :**
  - 13.234.113.207/products/details.php?p_id=2
- **Affected Parameters :**
  - Comment / Review Box
- **Payload :**
  - <script>alert(12)</script>

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

# #1

- Navigate to the site and log in to your account.
- Go to products and select any product. Now the URL shows the id of the product you are on, you can replace that with the payload.

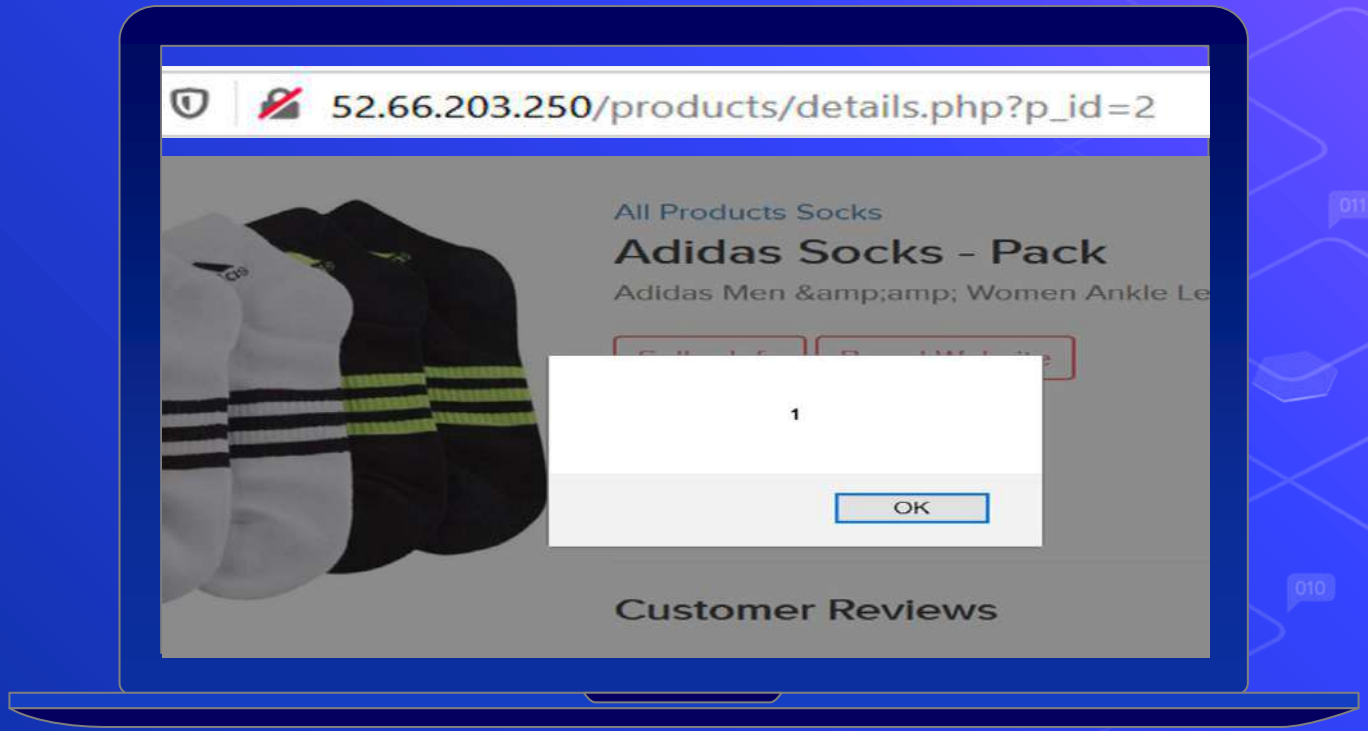


# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

# #2

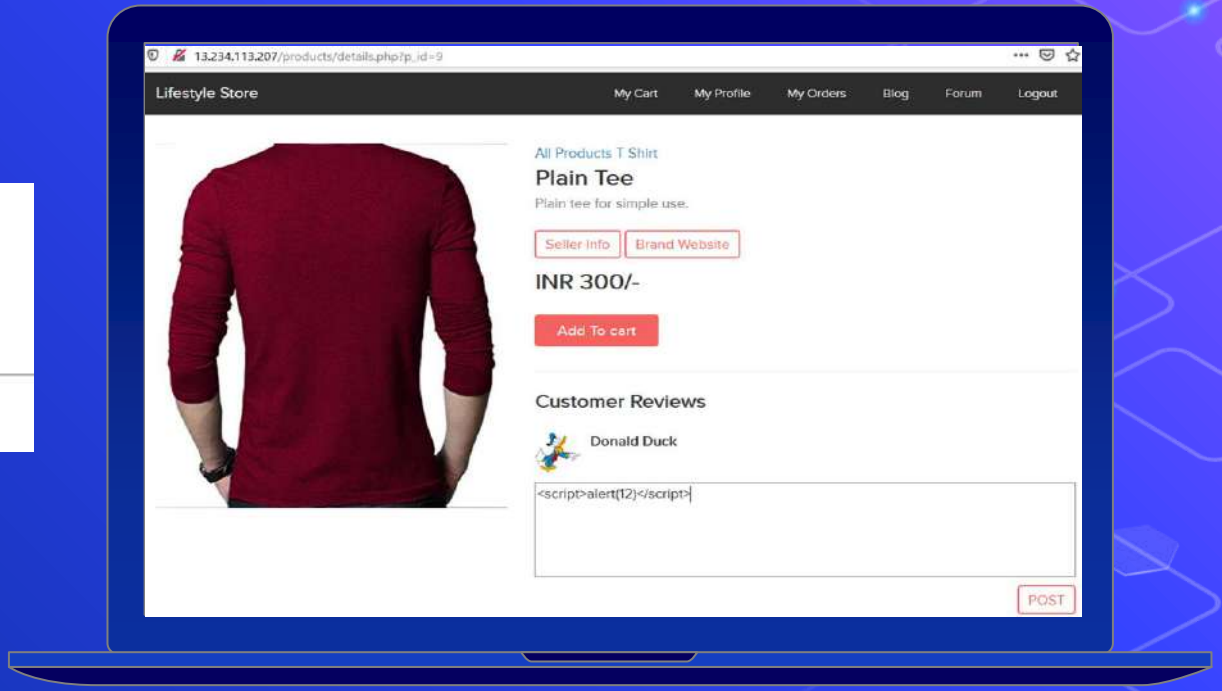
- After entering the payload press enter. The page will redirect you to home page but when you click on the same product again it will show the alert you had just input. Screenshot below.



# #3

- We have a similar issue, in the review / comment section.
- Inject payload <script>alert(12)</script>

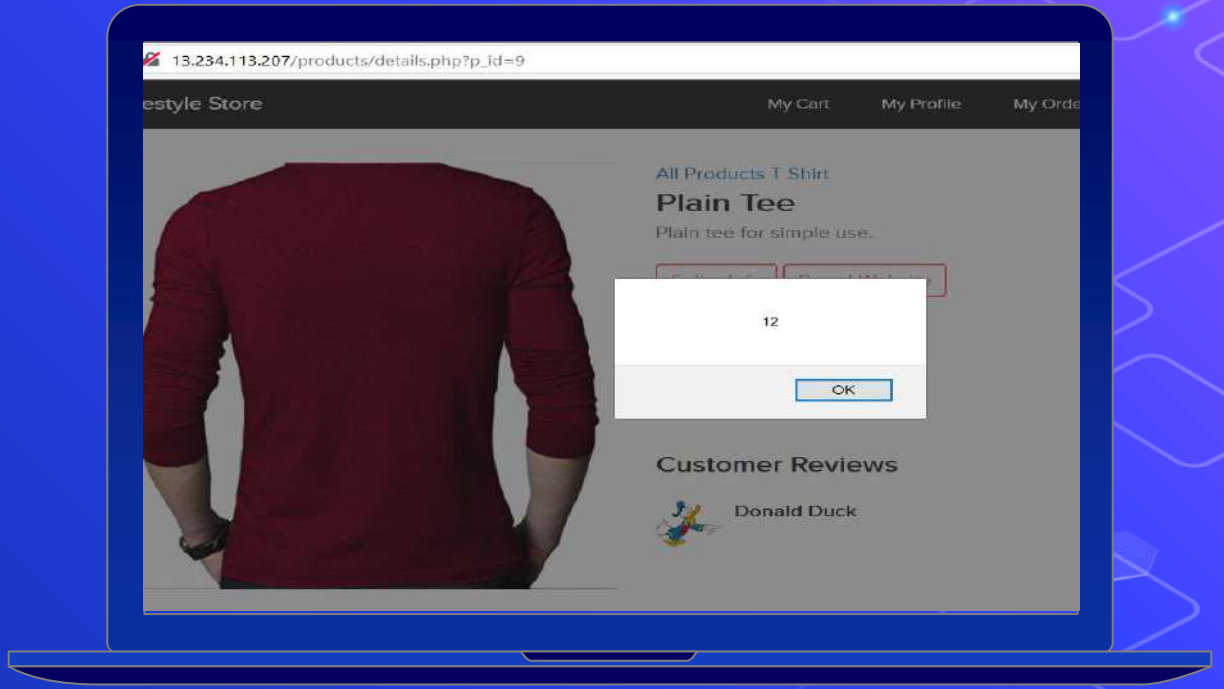
Zoomed

# #4

- We get the alert by the website.

Zoomed

# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- As the attacker can inject arbitrary HTML, CSS and JavaScript via the URL, he/she can put any content on the page like phishing pages, and can even host explicit content that could compromise the reputation of the organization.

- When the attacker sends the link with the payload to the victim, the victim sees a modified content on the website. As the user trusts the website, he/she will trust the content.

Following slide has the recommendations to tackle this risk.

# Recommendations :

Take the following recommendations to secure and avoid defacing / data altering.

# Recommendations :

Take the following precautions :

1. Secure the user input by blocking all the input you don’t want.

2. Encode the HTML / CSS special characters and codes in an encoded form before printing them on the website.

# References :


https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
https://en.wikipedia.org/wiki/Cross-site_scripting
https://www.w3schools.com/html/html_entities.asp

# 5.
# Directory Listings.

(SEVERE)

# This URL is vulnerable.

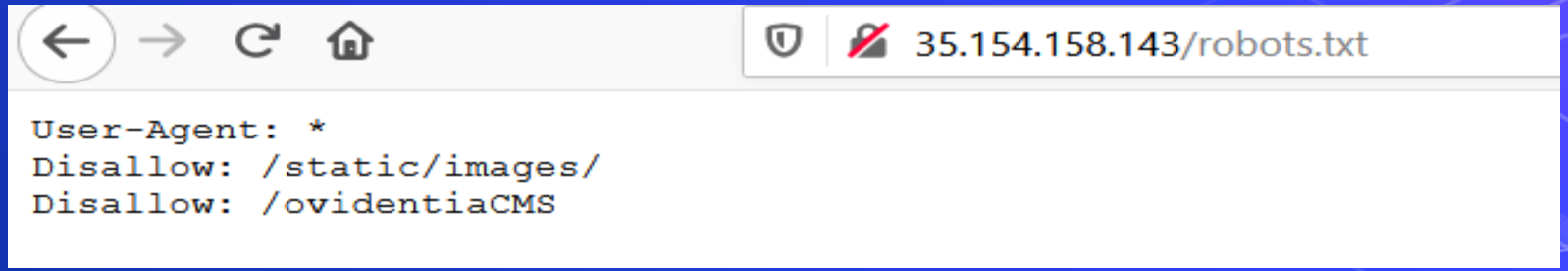
- Relevant URL :
  - http://35.154.158.143/robots.txt

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

## #1

- Go to http://35.154.158.143/robots.txt
- Here, we can see a complete list and directories of the customers, images of products and all the products-page info.
- We can also see the administrator directory.
- This has a weak password flaw in Ovidentia CMS, which lets an attacker access the administrator panel without much trials.

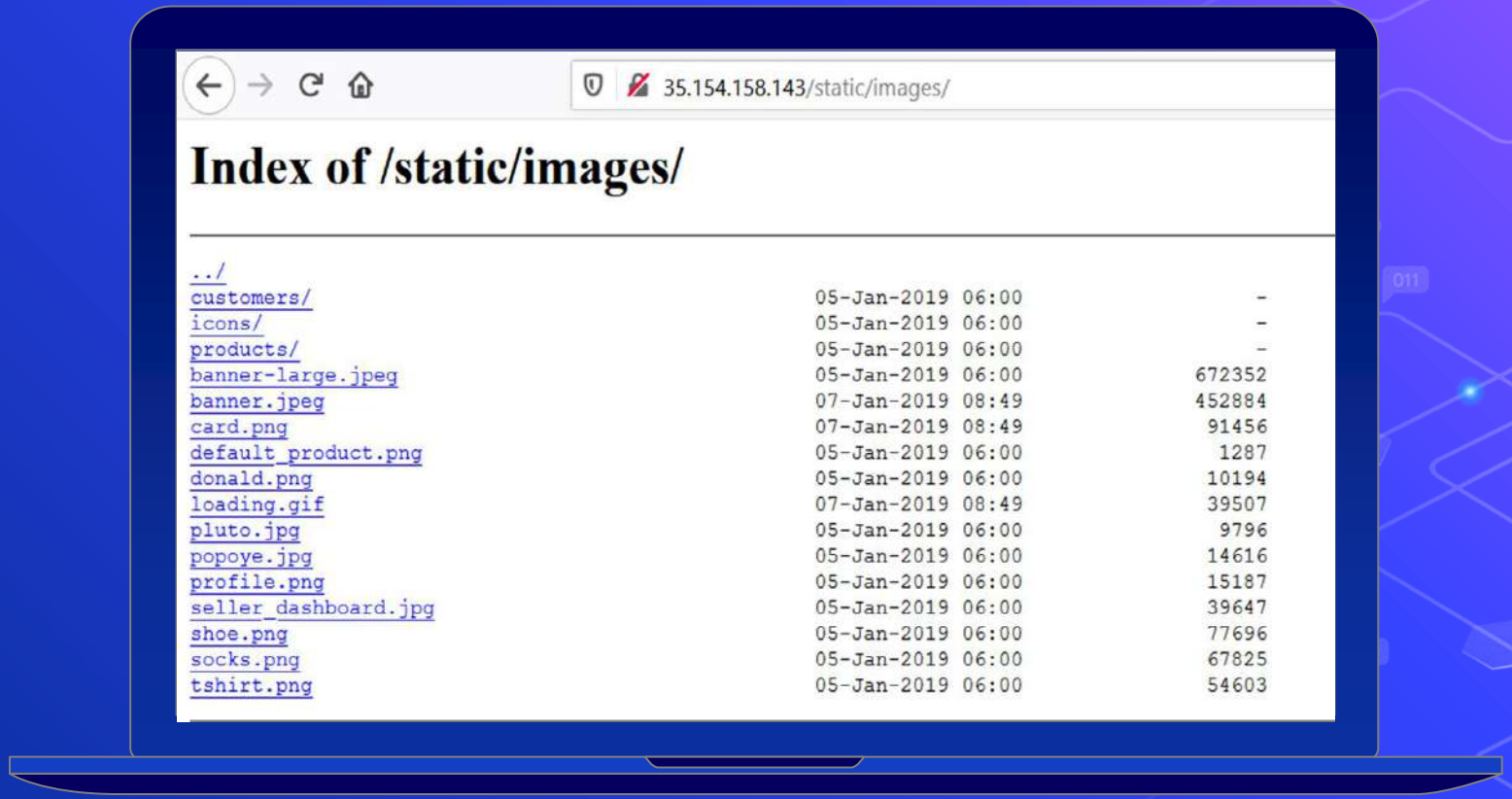


# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

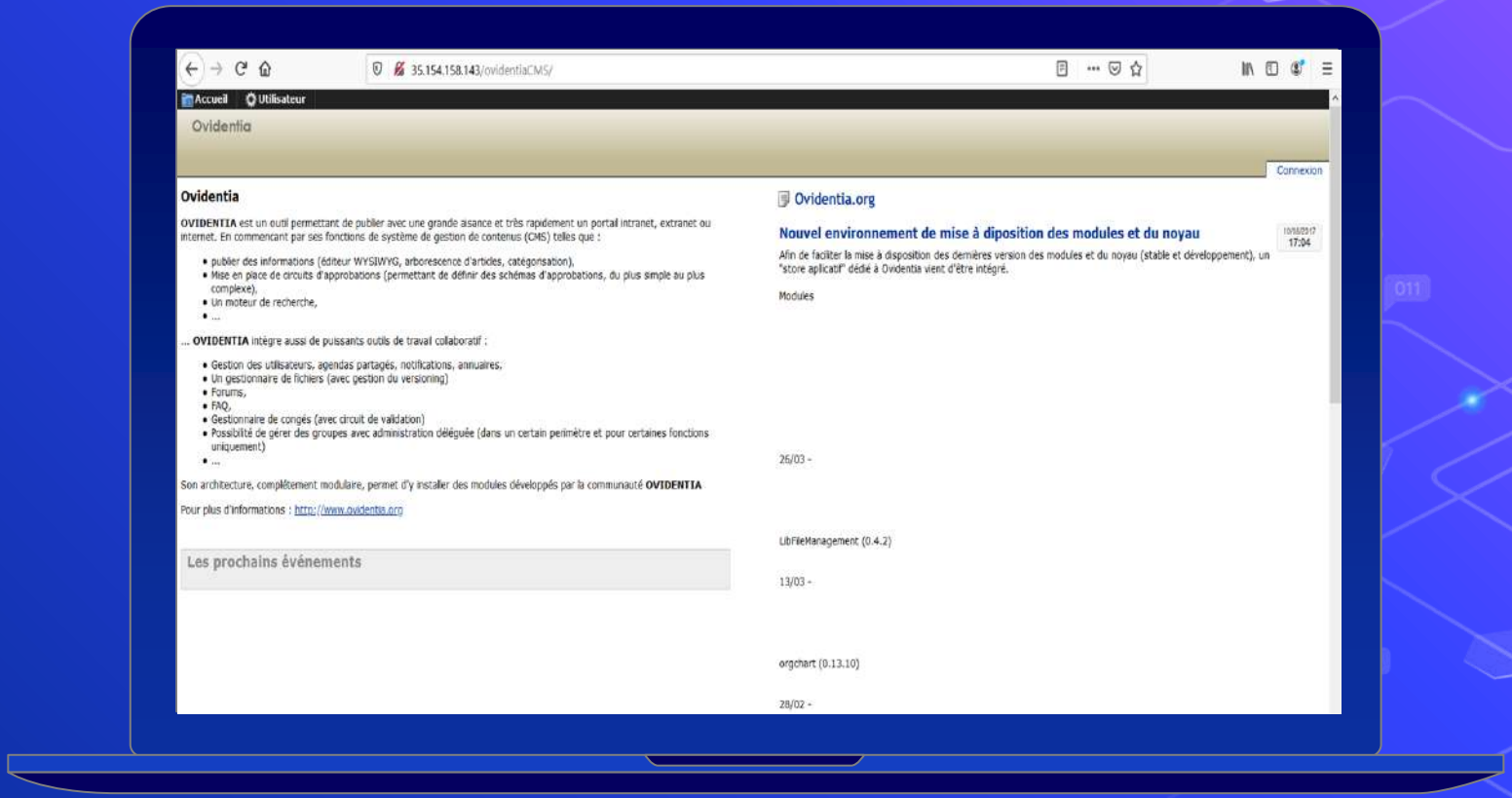
# #2

- /static/images



# #3

- /ovidentiaCMS/



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- The hacker can get all the information on the CMS and also get the privileges to tackle with the administrator login credentials and compromise the complete name of the company.
- A malicious hacker can take important information from seller point of view and what products every seller is selling at what price and also the information of users.

Following slide has the recommendations to tackle this flaw.

# Recommendations :

Take the following recommendations to secure and avoid defacing / data altering.

# Recommendations :

Take the following precautions :

1. Disable all the directory listings.

2. Include a index.html in all folders with default messages.

3. Remove all the default passwords and add your own new strong passwords which must have a special character and a number and must be greater or equal to 8 digits for maximum security.

# References :

https://securitytracker.com/id?1014149
https://www.w3schools.com

# 6. Information Disclosure

(LOW)

# This URL is vulnerable.

- Relevant URL :
  - http://35.154.158.143/Server-status

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

# #1

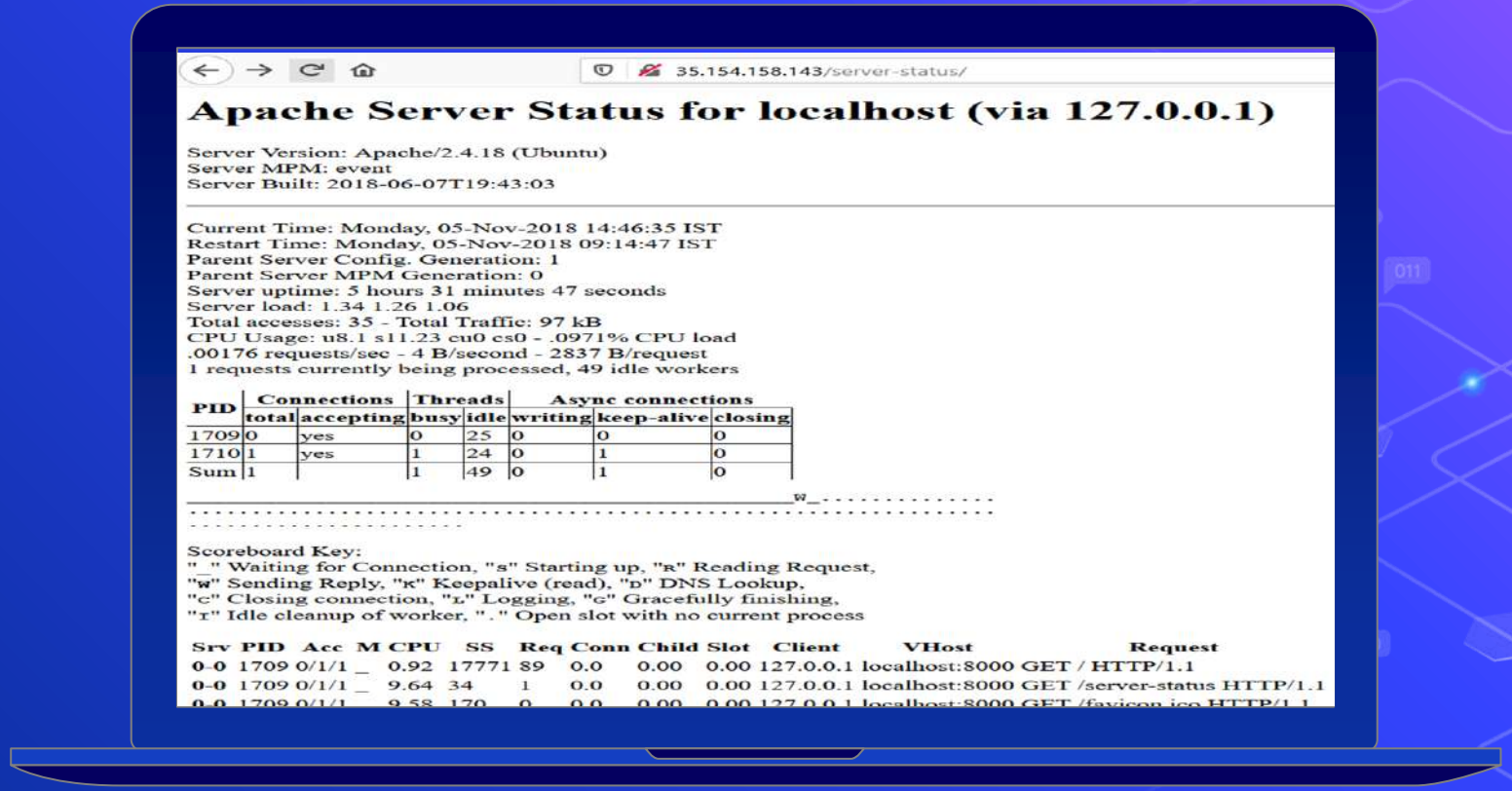
- Go to http://35.154.158.143/robots.txt
- Complete list and directories of the customer and images of products and all the products page info and also the administrator directory is also shown.
- Also this has a weak password flaw in Ovidentia CMS, which lets an attacker access the administrator panel without any tough trials.

# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

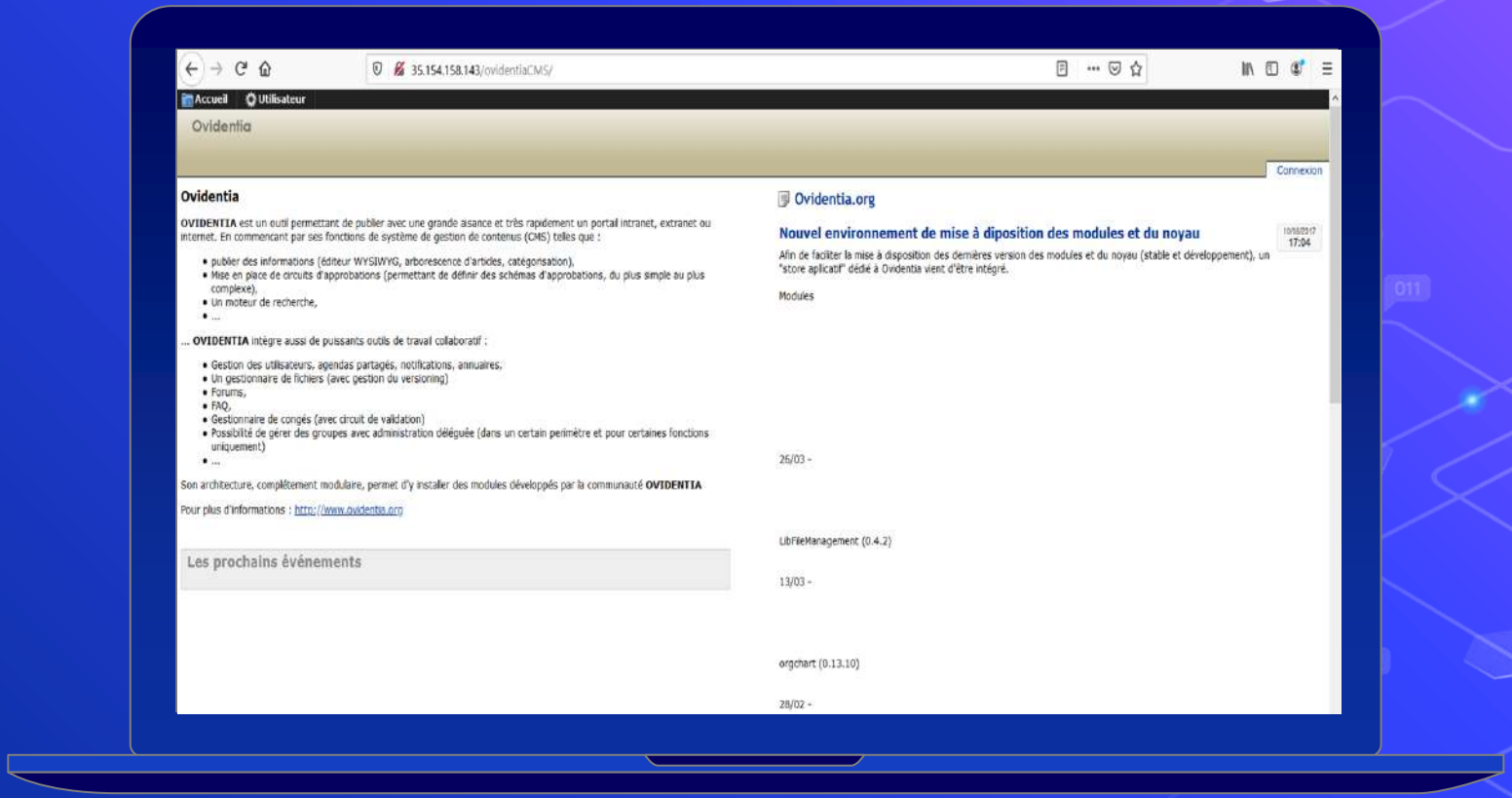
# #2

- Go to http://35.154.158.143/server-status and the critical server information will be shown, as seen below.



# #3

- /ovidentiaCMS/



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- Although this vulnerability does not have a direct impact on the server or the client but a malicious hacker can know things like mapping the server-architecture and map his hacking plan further on what and how to compromise.

Following slide has the recommendations to tackle this low risk vulnerability.

# Recommendations :

Take the following recommendations to secure and avoid defacing / data altering.

# Recommendations :

Take the following precautions :

1. Disable all the directory listings.
2. Disable the server-status.

# References :

https://vuldb.com/?id.88482
https://httpd.apache.org/docs/current/mod/mod_status.html

# 7.
# Personally Identified Information Leakage

(CRITICAL)

# Below mentioned URL LEAKS Critical information via PII leakage.

- **Relevant URL :**
  - http://35.154.158.143/static/images/uploads/products/2socks.jpeg
- **Affected Parameters :**
  - IMAGE (every image after “products/”)

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

# #1

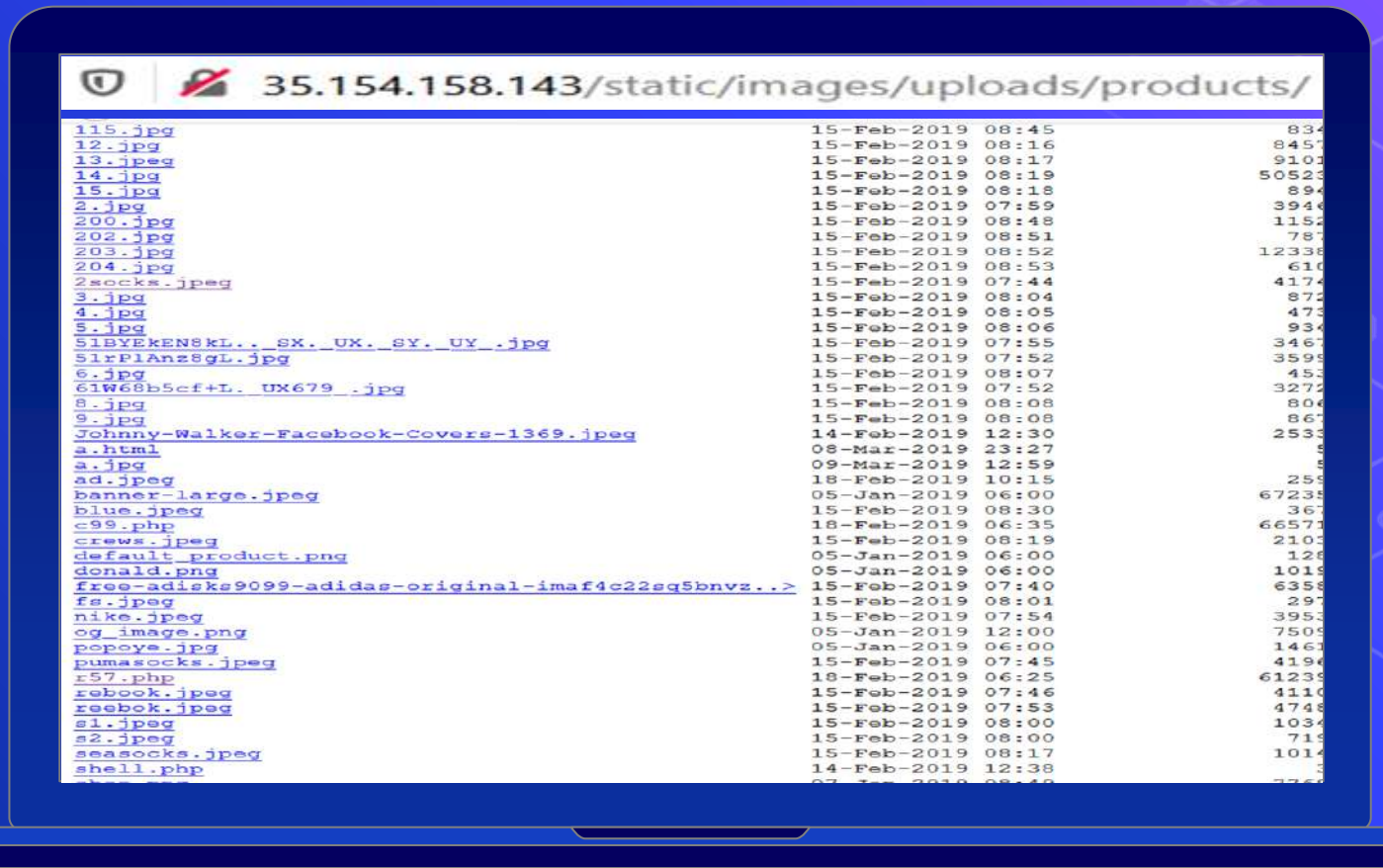
- Go to http://35.154.158.143/products.php and drag any drop product image in a new tab, the following page will look like the one shown below with the image of the product you chose.

# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

# #2

- Now, we remove the image name, i.e.”2socks.jpeg” in our case and hit enter, the following page with details will be shown.



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- A malicious hacker can gain access to the shell application and various other html files which has been shown in the listings.
- It also has a weak-password flaw.
- A hacker can get access to the shell and upload his own malicious codes to make the trusted site a hackers room for phishing and tricking users. This will result in defamation of the website.

Following slide has the recommendations to tackle this extreme risk vulnerability.

# Recommendations :

Take the following recommendations to secure and avoid defacing / data altering.

# Recommendations :

Take the following precautions :

1. Enable 2- factor authentication for sensitive data, and also use stronger passwords of at least 8 characters of different types.

2. Find all PII stored, and encrypt them with various techniques and also disable all the listings.

# References :

https://www.cloudcodes.com/blog/what-is-pii.html
https://hackerone.com/reports/374007

# 8.
# Self Redirection HTML Developer Flaw

(SEVERE)

# Below mentioned URL  has a major development flaw , as it redirects to password reset link without authentication.

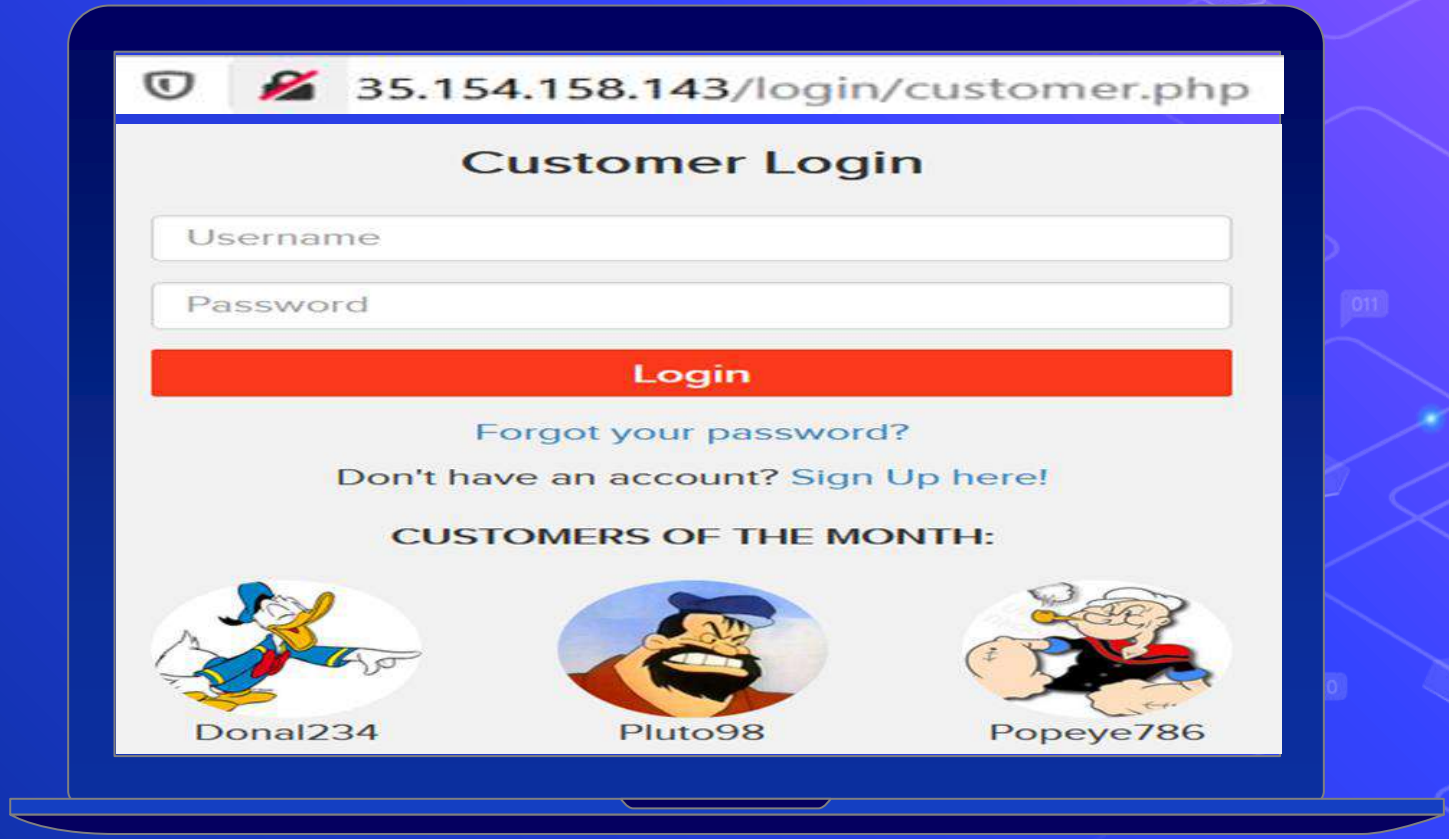
- Relevant URL :
  - http://35.154.158.143/reset_password/customer.php
- Affected Parameters :
  - Password Reset

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

# #1

- Go to http://35.154.158.143/login/customer.php and you will see , names of top customers ,as shown below, take note of any username.

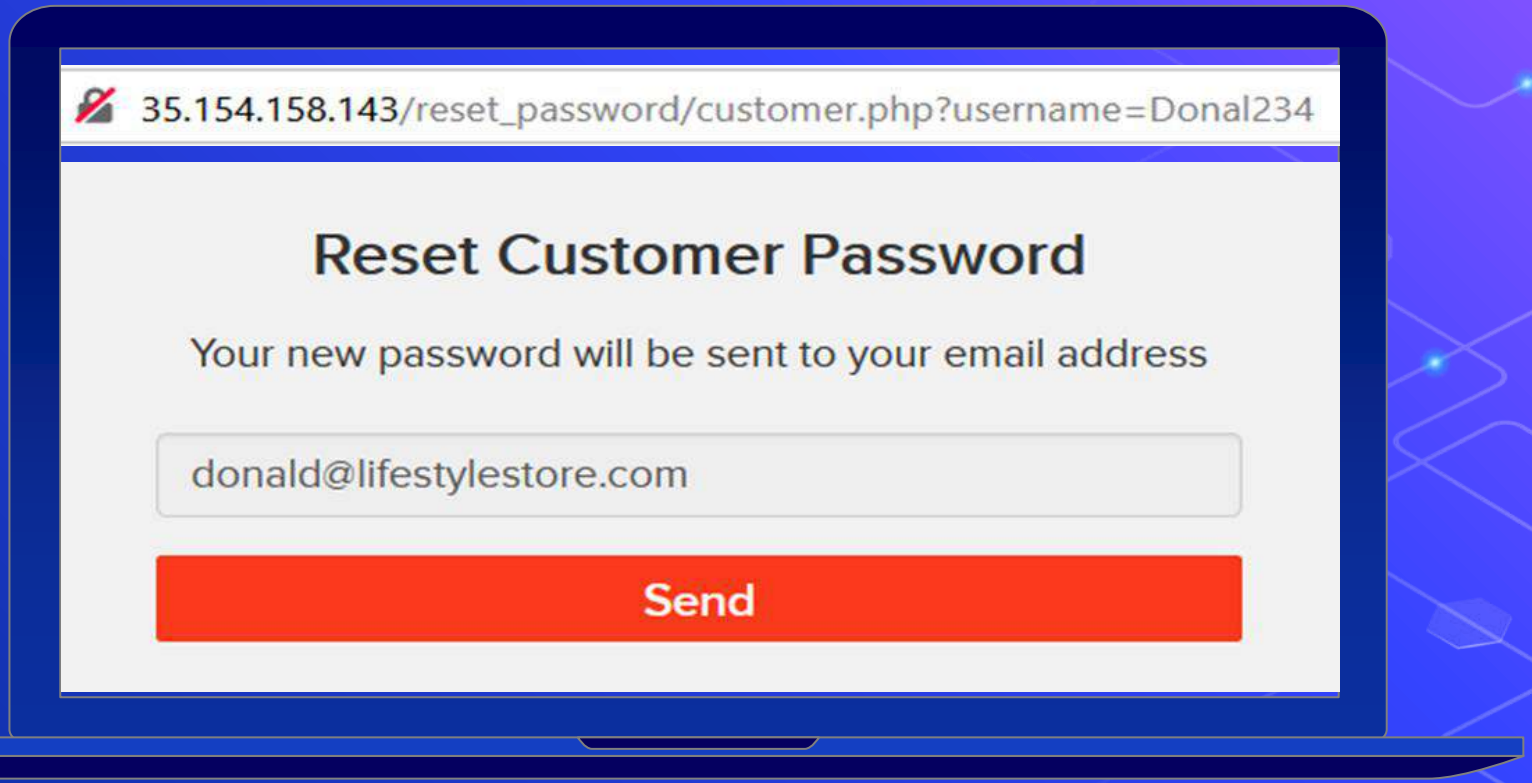


# Proof Of Concept :

Following Slides Have The Proof and Risks Of The Proposed Vulnerability.

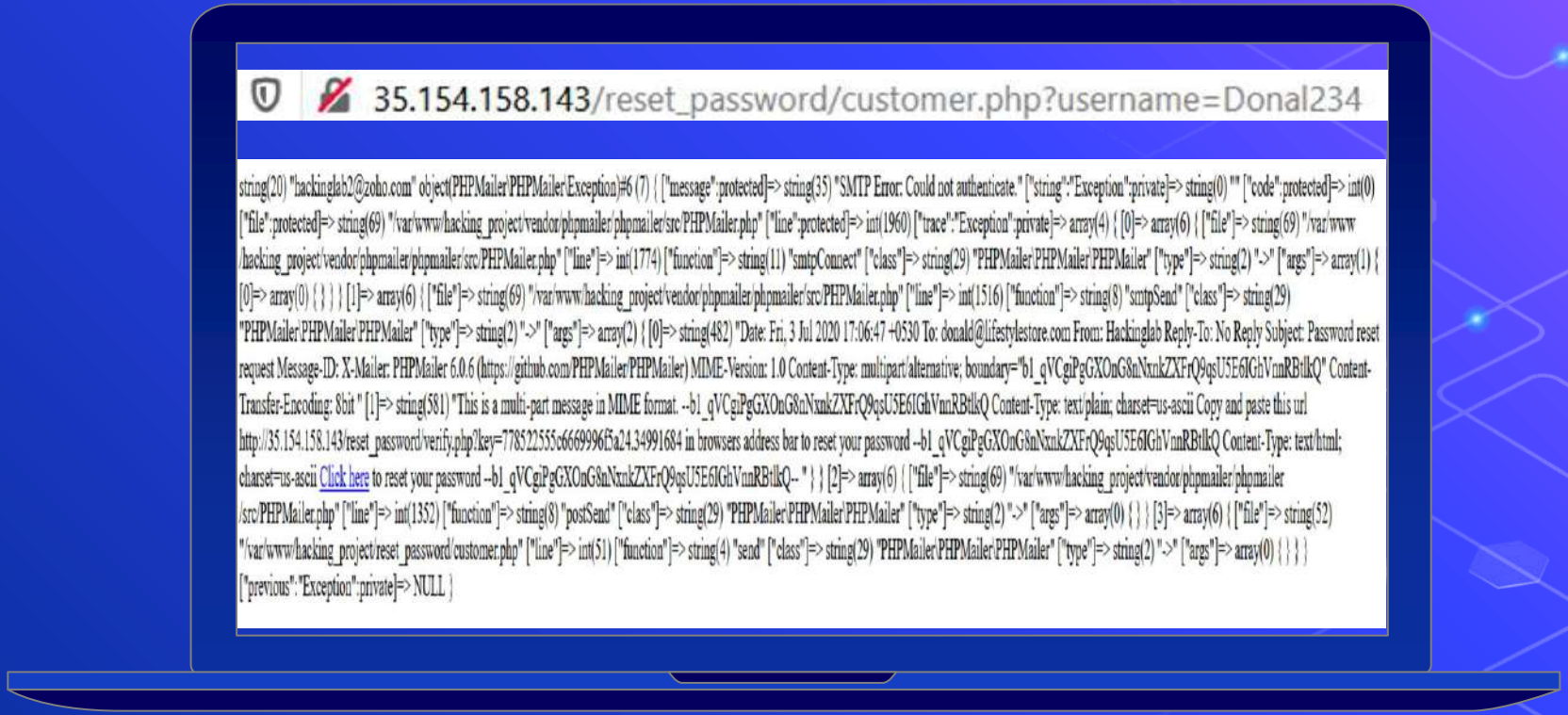
# #2

- After memorizing any username , click “forgot your password” and then input the memorized username and click “reset password”. We’ll see the page like this, click send.



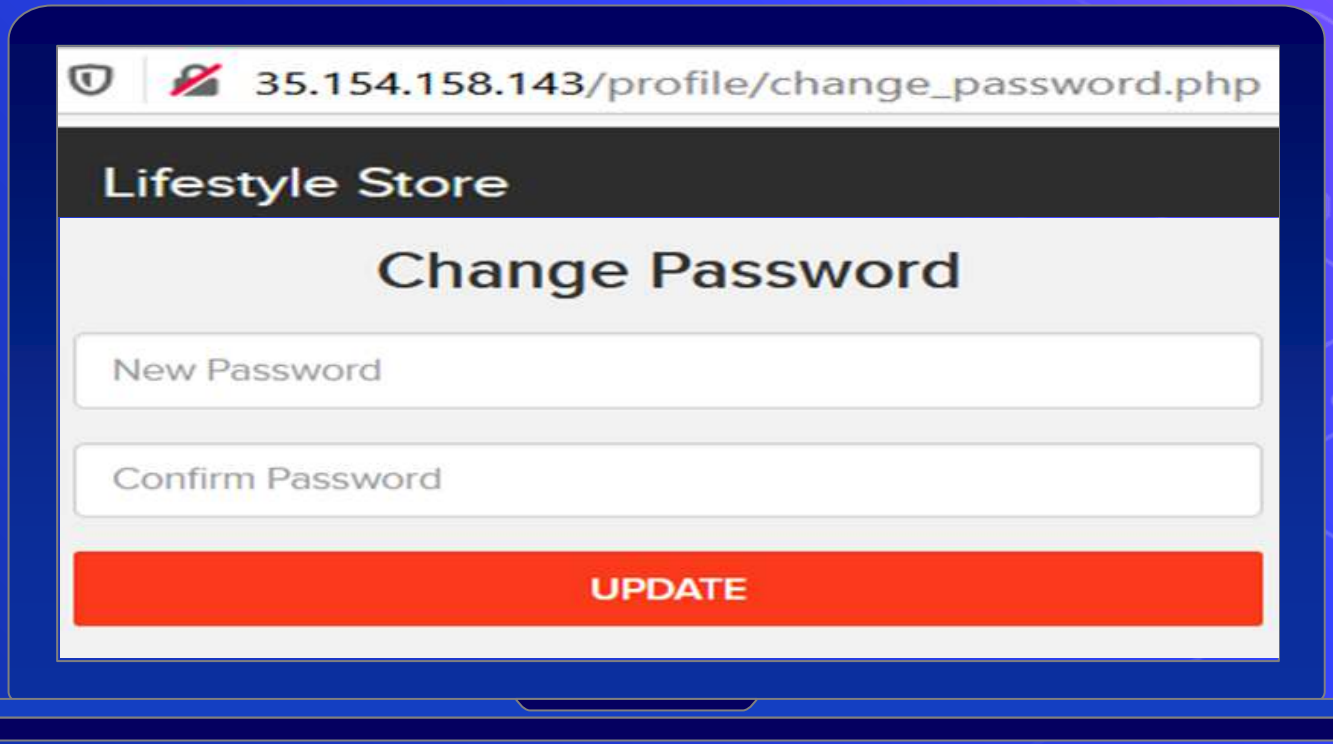
# #3

- When you click send you are directed to a page like this due to development/authentication error, click “click here” button and you’ll be redirected to change the password of the customer.



# #4

- You'll be redirected to change password of the customer id ,and you'll be automatically logged in to that customer's account even if you don't change the password, you can change the password for completely takeover the customer id.



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- The impact on the business is high, as a malicious hacker can get access to every user in the website and can login and perform tasks for his benefits as can also make the site vulnerable for security of customers.
- This will result in the defamation of the website as  the users trust it.

Following slide has the recommendations to tackle this extreme risk vulnerability.

# Recommendations :

Take the following recommendations to secure and avoid defacing / data altering.

# Recommendations :

Take the following precautions :

1. Better authentication should be provided and re-direction site must be re-checked before action.

2. There must be a proper authorization access.

# References :

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

# 9.
# Multiple vulnerabilities found in blog site

**(PII leakage , temporary XSS and more)**

(CRITICAL)

# Below are the affected parameters.

## URL of the Blog Site : http://15.206.75.142/wondercms/

- #1 Relevant URL :
  - http://15.206.75.142/wondercms/files/b374kmini.php
- Affected Parameters :
  - File Upload (POST)
- #2 Relevant URL :
  - http://15.206.75.142/wondercms/home
- Affected Parameters :
  - Files (GET)
- #3 Relevant URL :
  - http://13.235.0.176/wondercms/files/minisell.php
- Affected Parameters :
  - Shell Upload

# Observations :

Following Slides Have The Details About The Vulnerability Observed.

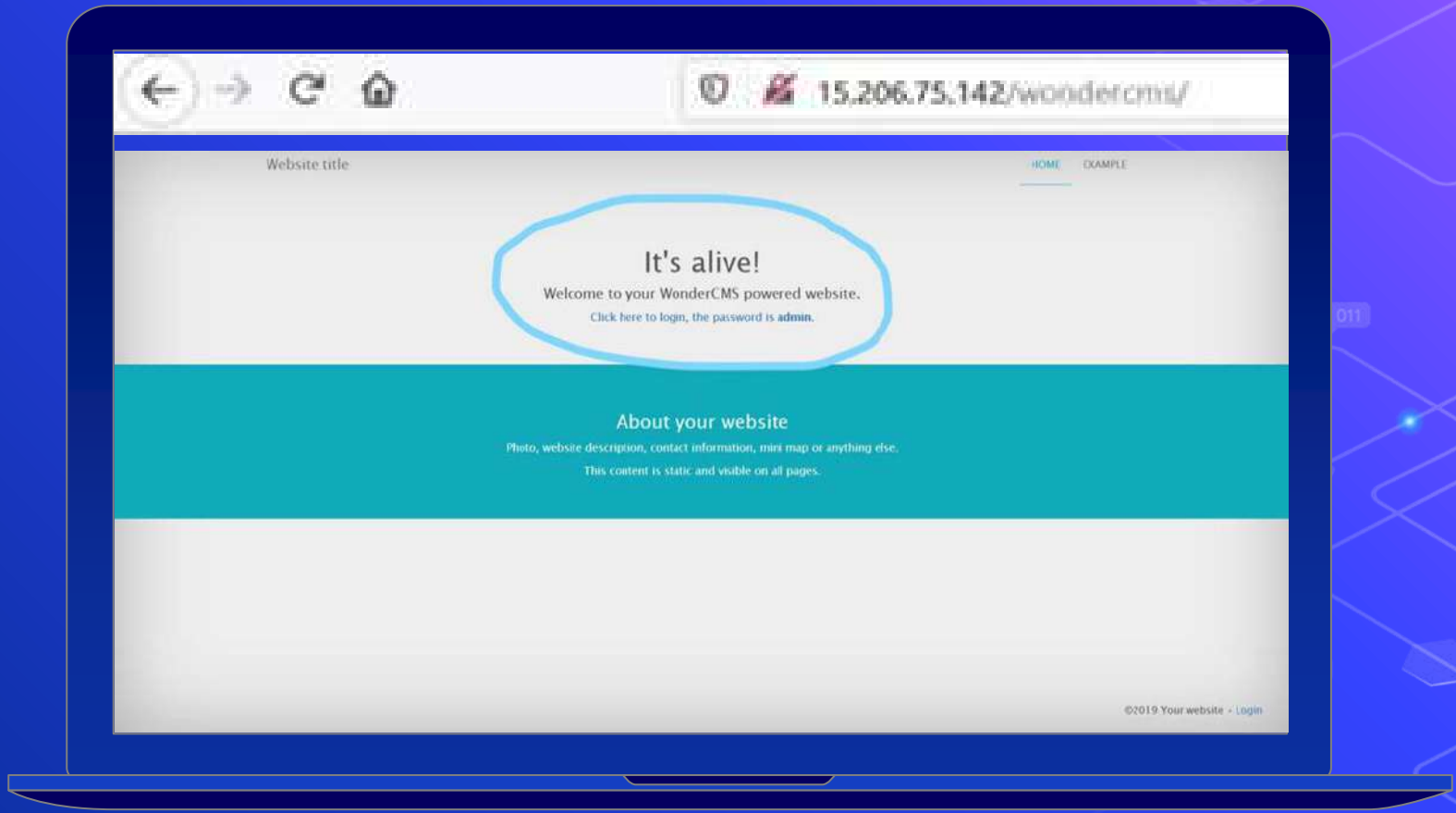
# #1

- At the front page of the blog site we can directly find the login option, that says “click here to login, the password is admin” by clicking on it, it will redirect us to the login page where just by entering ‘admin’ as password one can get access to the admin account of this site.

- This is a proper example of server side misconfiguration.

# #2

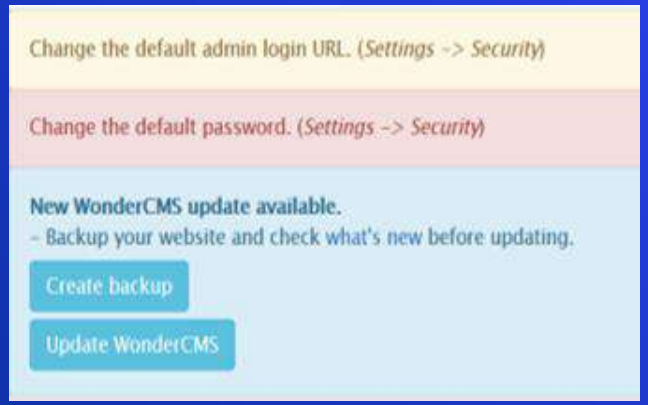
- This is how the logged in page looks like.

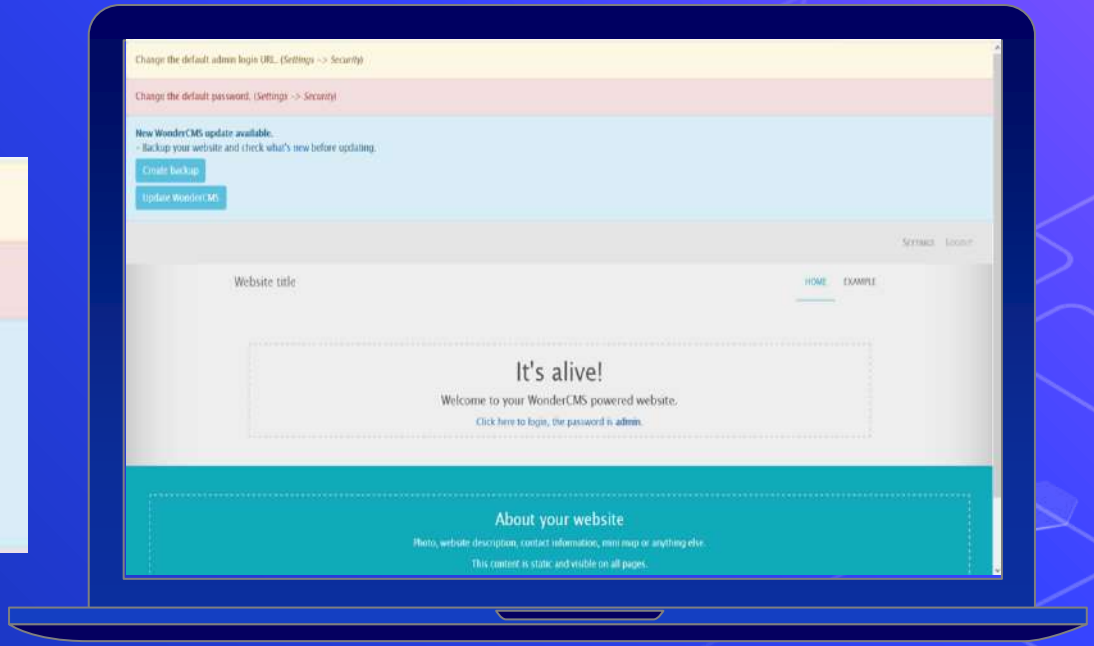


# #3

- When we log-in as the admin ,at the first page of the site we can see some text written as “It’s alive !_ _ _ _” if we click on it , the html codes just starts running and one  Malicious hacker can easily make changes and can use the page as his/her phishing portal . This is a classic example of  temporary XSS.

Zoomed



# #4

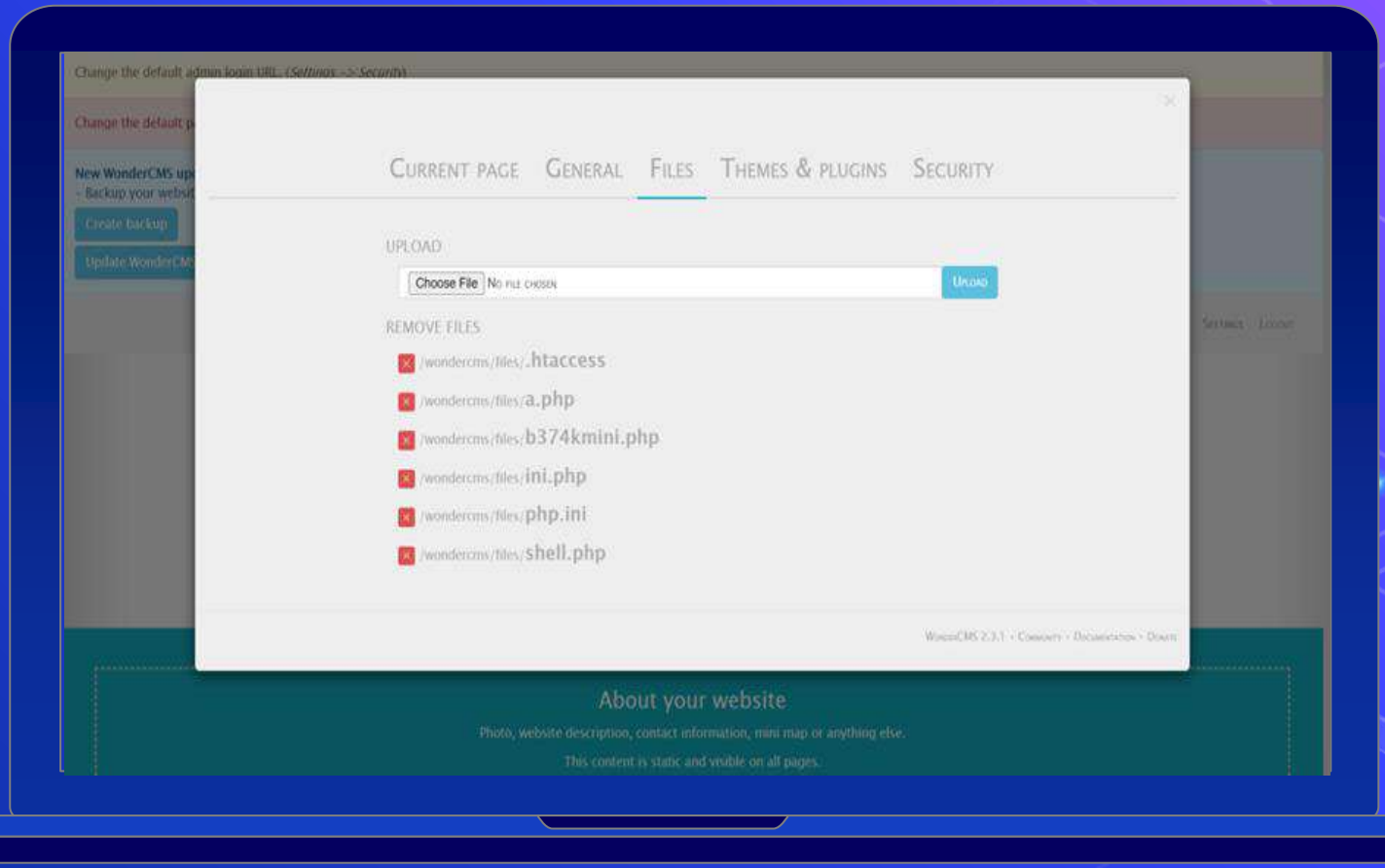
- As we are inside the admin account if we select the settings option and go to ‘files’ menu we can see some files with extension ‘.php’ which can give information about the site and also very much capable of taking over the site.

- Here, if we select the ‘/wondercms/files/b374kmini.php’ option it will redirect us to a page which will give us many crucial information like server IP address, Linux IP address , the server type, the database, shell, the php info, uploads.

- The hacker can also mail someone with the admin compromising many factors one of which is the site reputation.

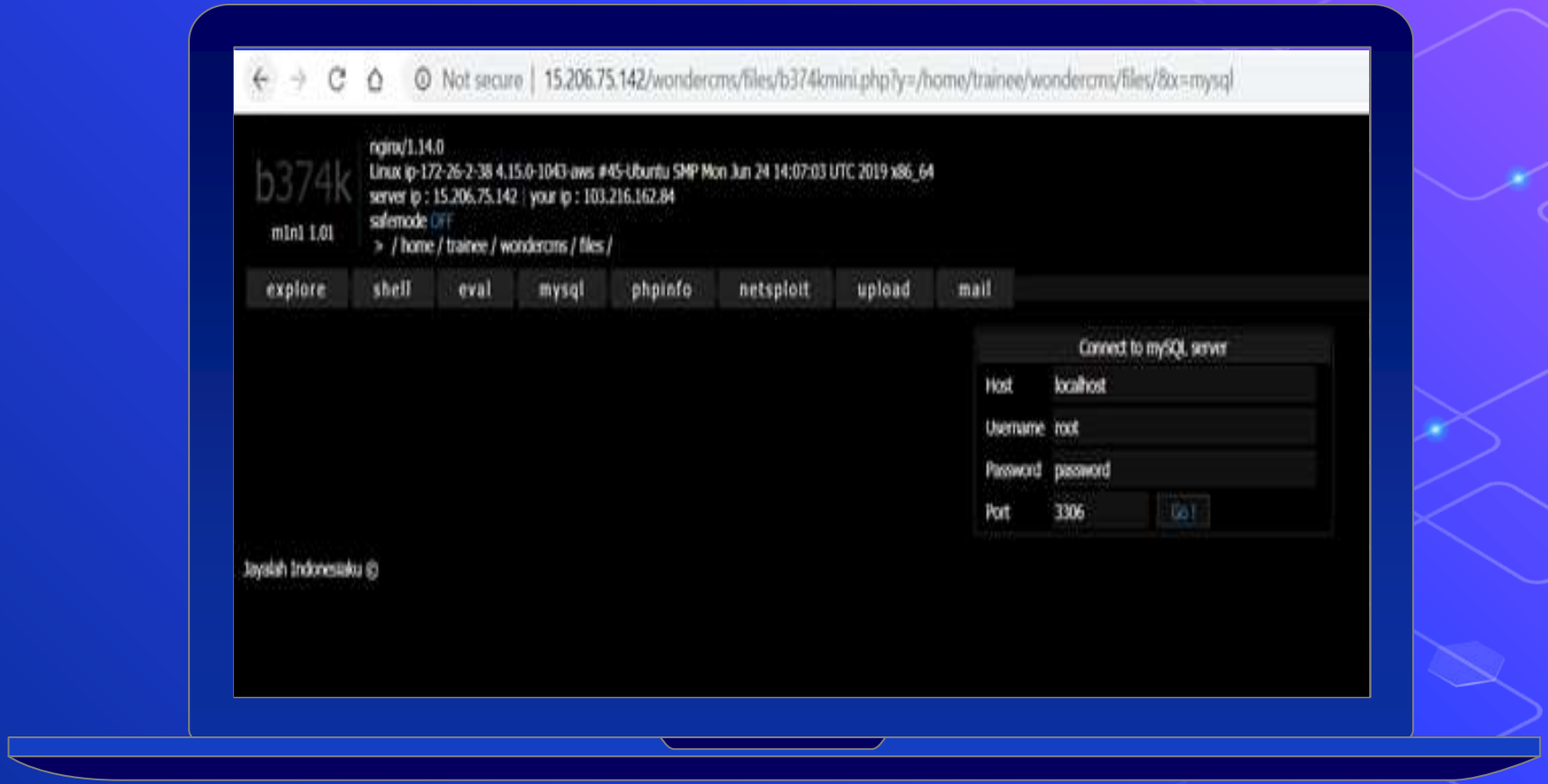
# #5

- Screenshot 1.



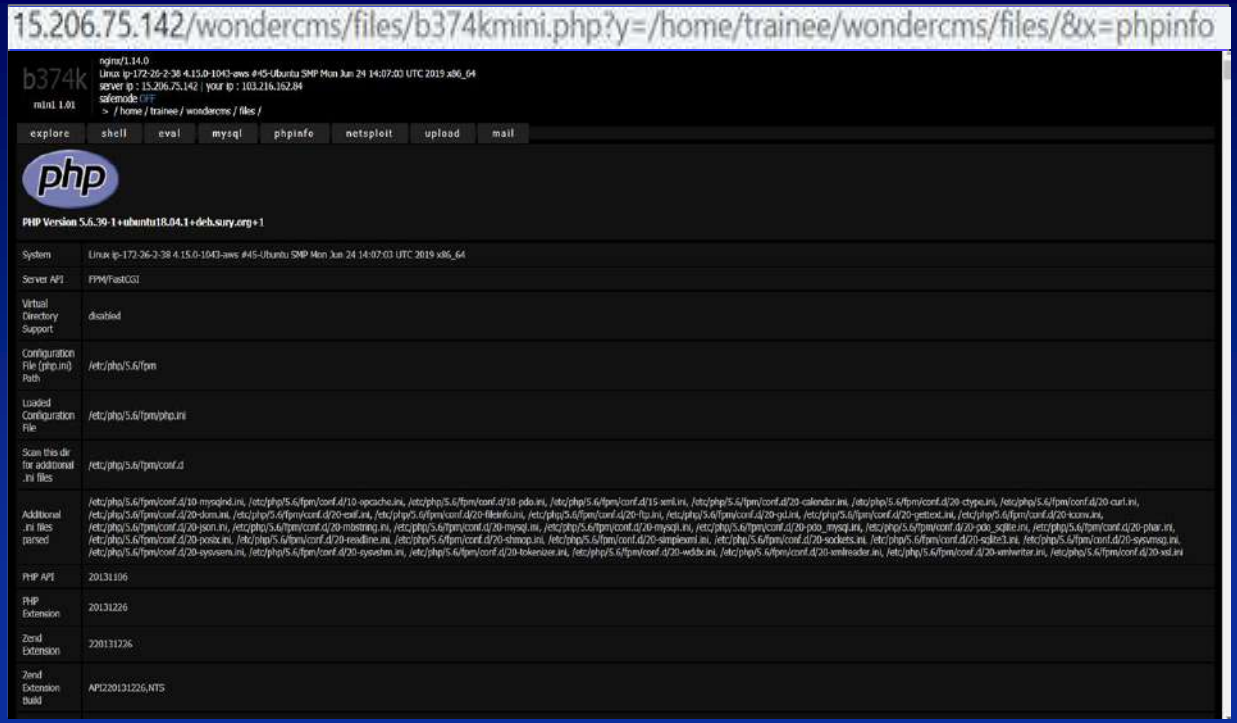
# #6

- Screenshot 2.



# #7

- The page looks like above, from which a attacker can know all system (backend) details like versions, and more.



# #8

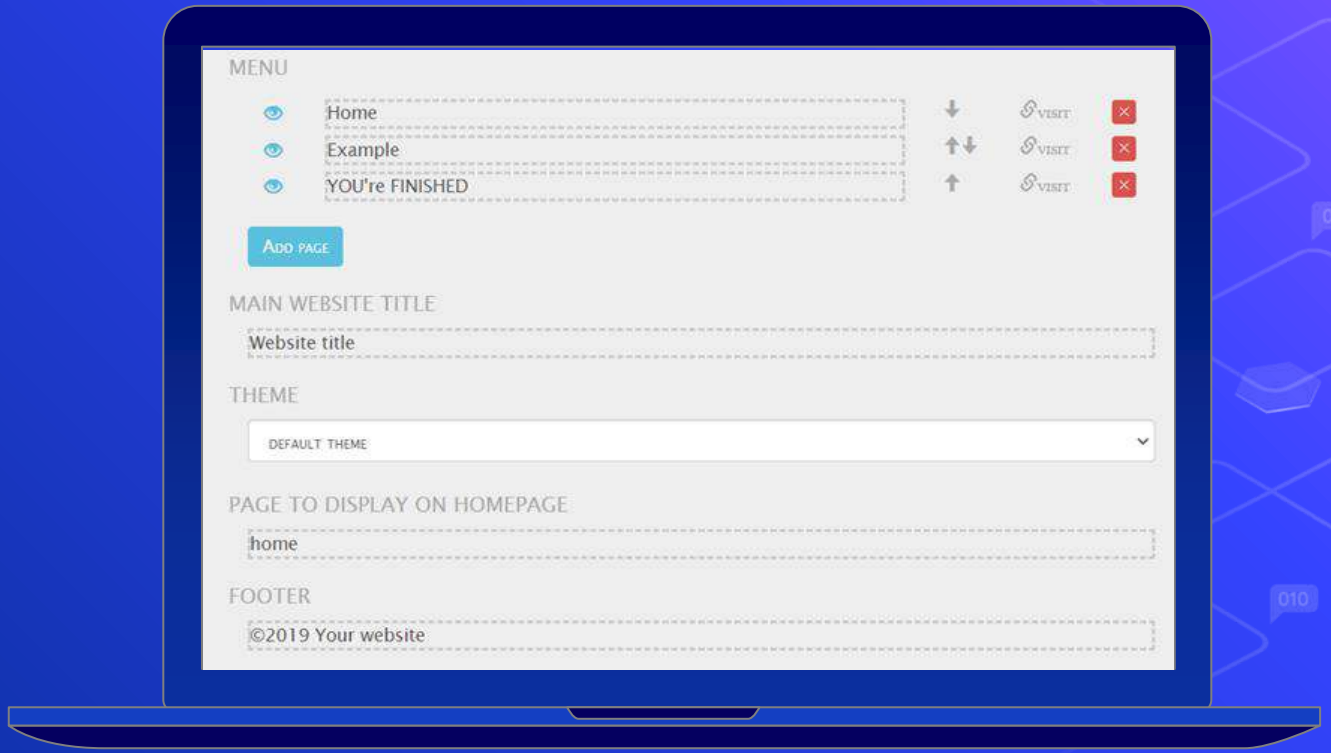
- Zoomed into some critical details from last screenshot.

| | |
|---|---|
| PHP API | 20131106 |
| PHP Extension | 20131226 |
| Zend Extension | 220131226 |
| Zend Extension Build | API220131226,NTS |

| | |
|---|---|
| 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 |

# #9

- When we are in the admin account and if we go to the general options found in the settings tab. We can see input fields, where a malicious hacker can inject and execute malicious client side scripts. Which gets permanently stored in the database.



# #9

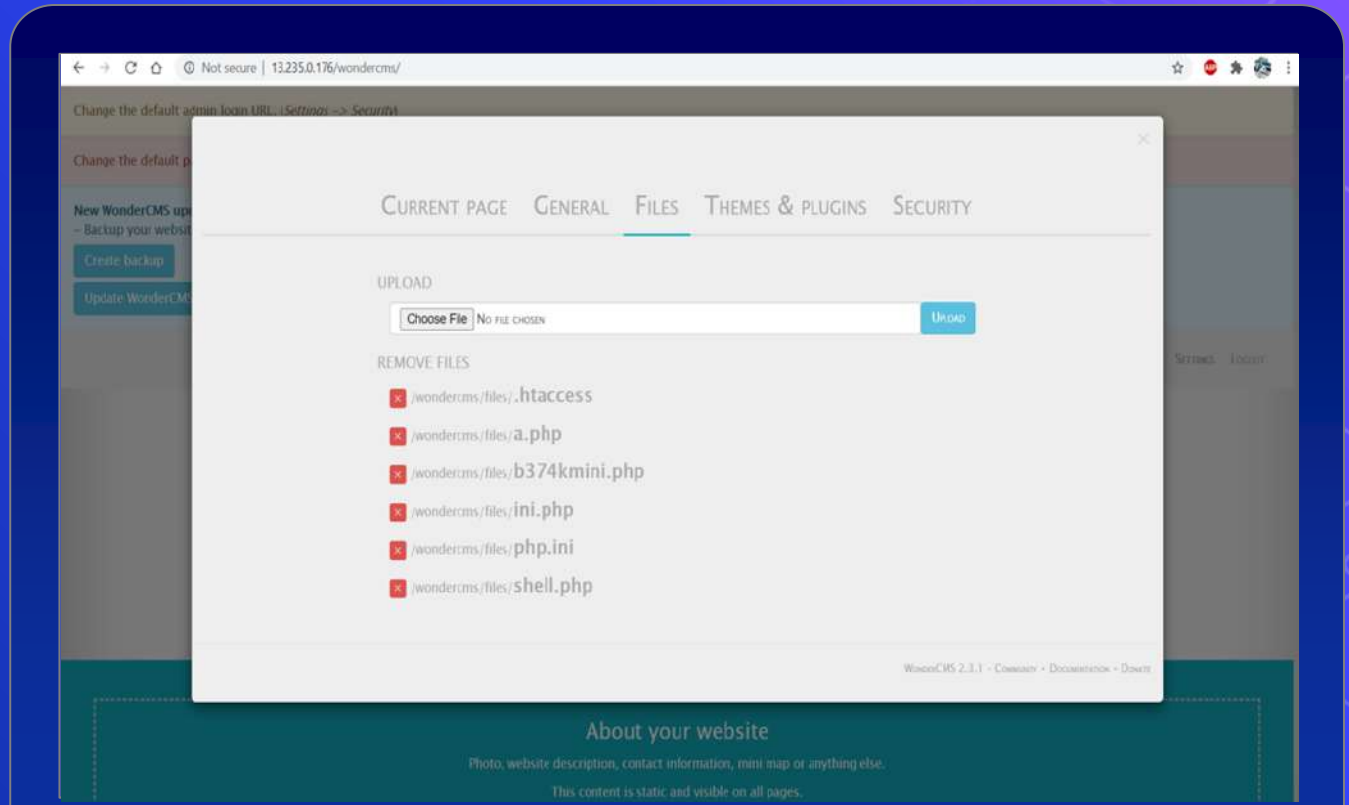
- On the same page under settings, click on files, here we have a file upload option, which has no limitations on the type of file which can be uploaded.

- We can upload files of any extension. Here when we upload a shell into the site, it will give access to take over the whole website.

- This will be a great breakage of the site.

# #10

- Screenshot.



# Business Impact :

The Impact On The Company If The Vulnerability Is Exploited.

# Business Impact :

- As the blog site has multiple vulnerabilities such as temporary XSS, server side errors and development flaws the site is extremely vulnerable as a malicious hacker can indulge in the site and can use all the vulnerable components to do phishing and other attacks on user and perform malicious activities.
- This will result In defamation of the name of the company, money loss and customers never trusting the website again.

Following slide has the recommendations to tackle this extreme risk vulnerability.

# Recommendations :

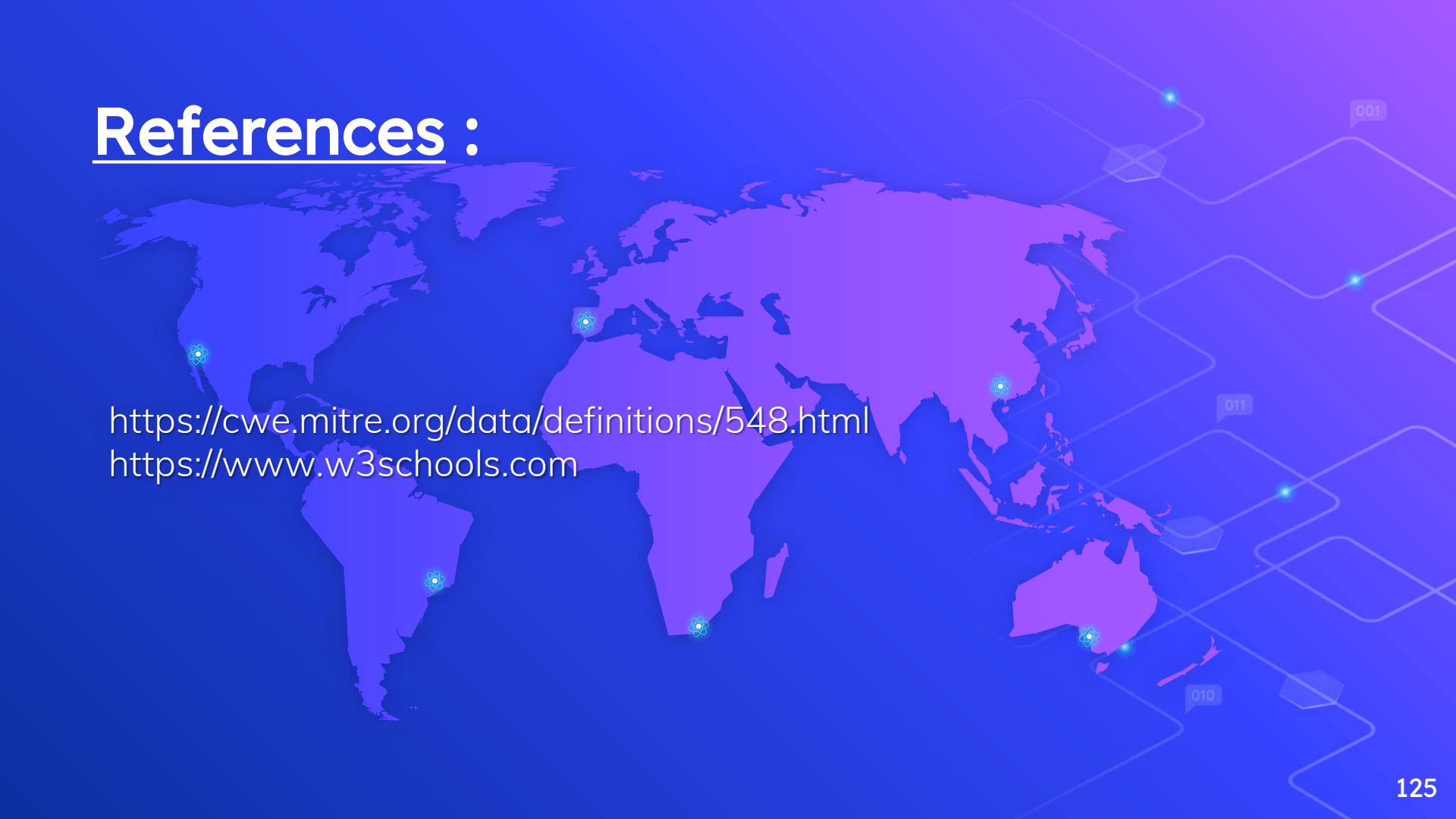
Take the following recommendations to secure and avoid defacing / data altering.

# Recommendations :

Take the following precautions :

1. The website should have proper two factor authentication.
2. Try to develop the back end more stronger and secure.
3. Remove all directory listings and add proper sanitization techniques at checks of the website.

# References :



https://cwe.mitre.org/data/definitions/548.html
https://www.w3schools.com

# Thanks!

**Hi, I am Shaswat Manoj Jha**

I have generated this complete Vulnerability report for the e-commerce platform. : **“Lifestyle Store”**.

You can find me at shaswatmanojjha@gmail.com.

**Feel Free To Contact For Additional Details Or Patch Assistance.**