INTERNSHALA

Lifestyle Store
E-Commerce Platform

Detailed Developer Report

Security Status — Extremely Vulnerable

- The Hacker can steal all the records in Lifestyle Store databases with SQLi.
- The Hacker can upload malicious programs by exploiting the file upload vulnerability.
- The Hacker can get account details of some other customer by changing the parameters in the URL link (IDOR).
- The Hacker can get access to seller details and login into the website using customer of the month usernames(PII) .
- The Hacker can send multiple requests (Rate Limiting Flaw).
- The Hacker can add or remove items from the cart (CSRF) .
- The website is very much vulnerable as it uses http instead of https.
- The website is vulnerable as in some modules the website uses GET based instead of POST.

Vulnerability Scale

CRITICAL	SEVERE	MODERATE	LOW
12	20	8	2

Vulnerabilities Found

SNo	Severity	Vulnerability	Count
1	Critical	SQL Injection	2
2	Critical	Insecure Direct Object Reference (IDOR) Vulnerability	6
3	Severe	Cross Site Scripting	2
4	Critical	Arbitrary File Upload	2
5	Critical	Access to Admin page	2
6	Moderate	Forced Browsing	2
7	Moderate	Missing Server Side Validation	2
8	Severe	Open Redirection	2
9	Severe	Brute Force	2
10	Severe	Personally Identifiable Information Leakage	2
11	Moderate	Unauthorized Access to Sellers Details	4
12	Low	Descriptive Error Message	2

SNo	Severity	Vulnerability	Count
13	Severe	Default File Misconfiguration	3
14	Severe	Default / Weak Passwords	4
15	Severe	Components with known Vulnerability	2
16	Severe	Network Protocol Vulnerability	1
17	Severe	Command Execution through Shell Uploading	2
18	Severe	Rate Limiting Flaw	1

Modules in the Website

- **Lang**
    - English
    - French
- **My cart**
- **My Profile**
- **My Orders**
- **Blog**
    - Home
    - Example
- **Sign Up**
- **Log in**
    - Customer
    - Seller
    - Admin

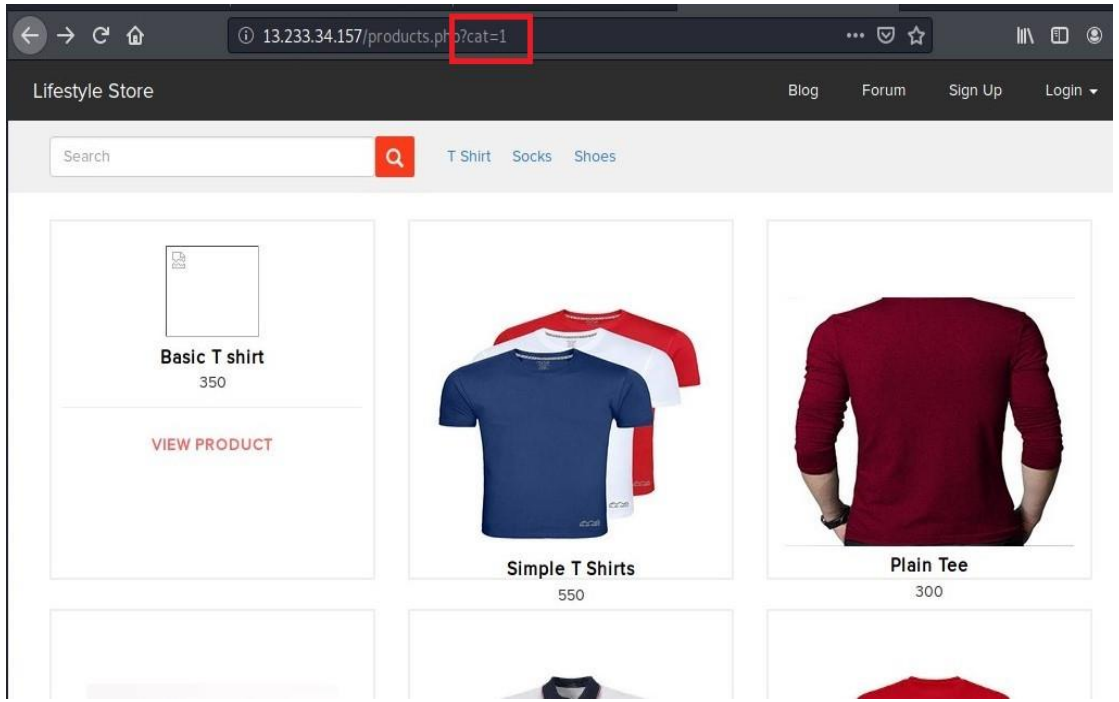
1. SQL Injection

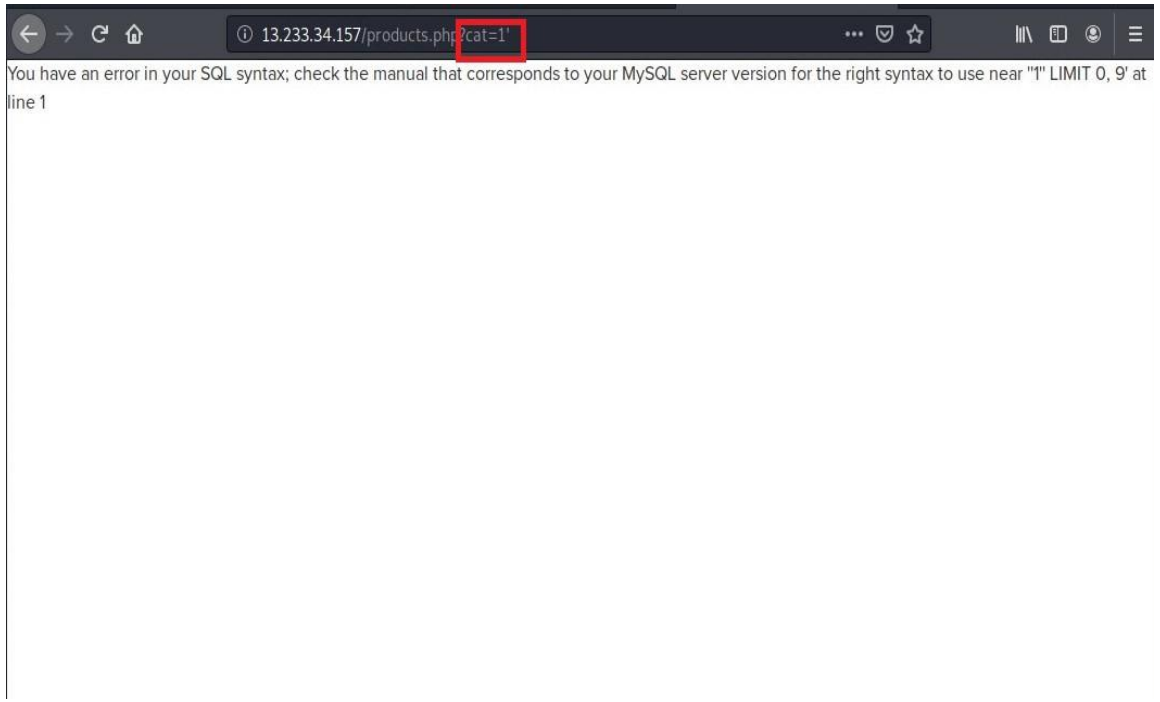
SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

SQL Injection	**In the link http://13.233.34.157/products.php the modules T Shirt/Socks/Shoes is vulnerable to SQL Injection.** **Affected URL :** http://13.233.34.157/products.php?cat=1 **Method used :** GET based **Payload:** cat=1’

Observation

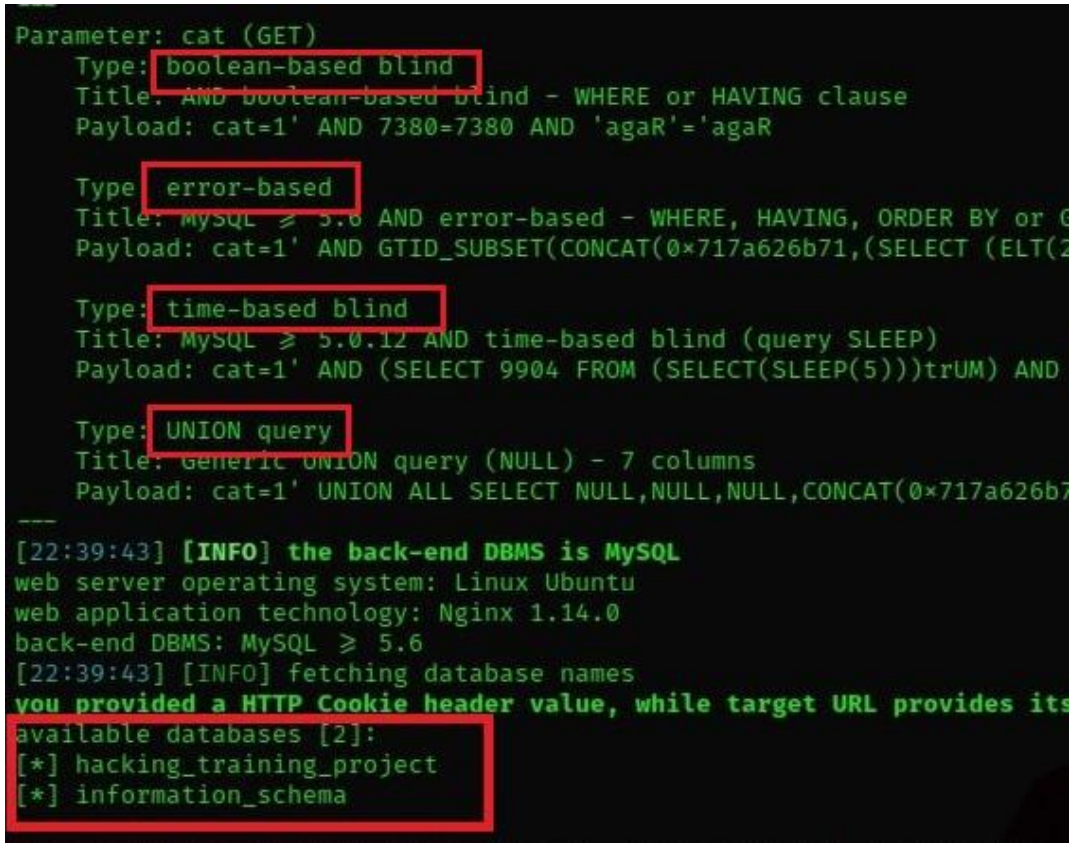
By adding ‘ into the URL we get an descriptive error. This is called Error based SQL Injection.





Proof of Concept (PoC)

- Through SQLi the hacker can run SQL commands on the URL and access the restricted data and harm the site.
- Through Burp Suite by capturing the packet we can get all the details and use them to automate the SQL injection and find what all injections the site is vulnerable to. (Error based , Time based ,BOOLEAN).



- Command used:

python sqlmap.py –r ”pro.txt”(POST)

- Through this we found the database

name and now can access the

database.

Proof of Concept (PoC)

No of databases: 2

- information_schema
- hacking_training_project

No of tables : 10

- brands
- cart_items
- categories
- Customers
- order_items
- Orders
- product_review
- Products
- sellers
- users

name	user_name	address	phone_number
admin	admin	Scholiverse Educare Pvt. Ltd. B-610, Unitech Business Zone, Nirvana Country, South City 2, Gurgaon, India - 122018	8521479630
Donald Duck	Donal234	B-34/ the duck lane, Disneyland	9489625136
Brutus	Pluto98	A-56 Sailor's ship, popeyeworld	8912345670
Chandan	chandan	GF-213, Nehru road, old Delhi market, 120078	7854126395
Popeye the sailor man	Popeye786	B-44 spinach house, Disneyworld	9745612300
Radhika	Radhika	D-60, Ajmer street, Ajmer	9512300052
Nandan	Nandan	WB-45 Wayne house, Batwan world	7845129630
Murthy Adapa	MurthyAdapa	Internshala (Scholiverse Educare Private Limited), B-610, Unitech Business Zone,, Nirvana Country, South City 2	8365738264
John Albert	john	Black street, st.Anna road, 56 Dwell	6598325015
Bob	bob	Bob, 23-Avenue street, construction Arcade, Dallas	8576308560
Jack	jack	234A, 5th Street Mountain view, Washington DC	9848478231
Bulla Boy	bulla	Bulla Boy, 98B, St. Peter road, Ramanin	7645835473
hunter	hunter	alert(1)	9788777777
asd	asd	asdasd	9876543210
acdc	acdc	jkhkjhkjkjkj	9999999999

Business Impact - Critical

- Using this vulnerability the hacker can execute SQL command on site and run malicious commands and gain complete access to databases along with all customer data.

- Using the details the hacker can log-in into the customer’s account and buy the products without the customer knowing and deal damage to the customer as well as the E-commerce website.

Recommendation

- Use of Prepared Statements (with Parameterized Queries)
- Use of Stored Procedures
- Allow-list Input Validation
- Escaping All User Supplied Input
- Do not run Database Service as admin/root user
- Disable/remove default accounts, passwords and databases
- Assign each Database user only the required permissions and not all permissions

References

- https://www.owasp.org/index.php/SQL_Injection

2. IDOR Vulnerability

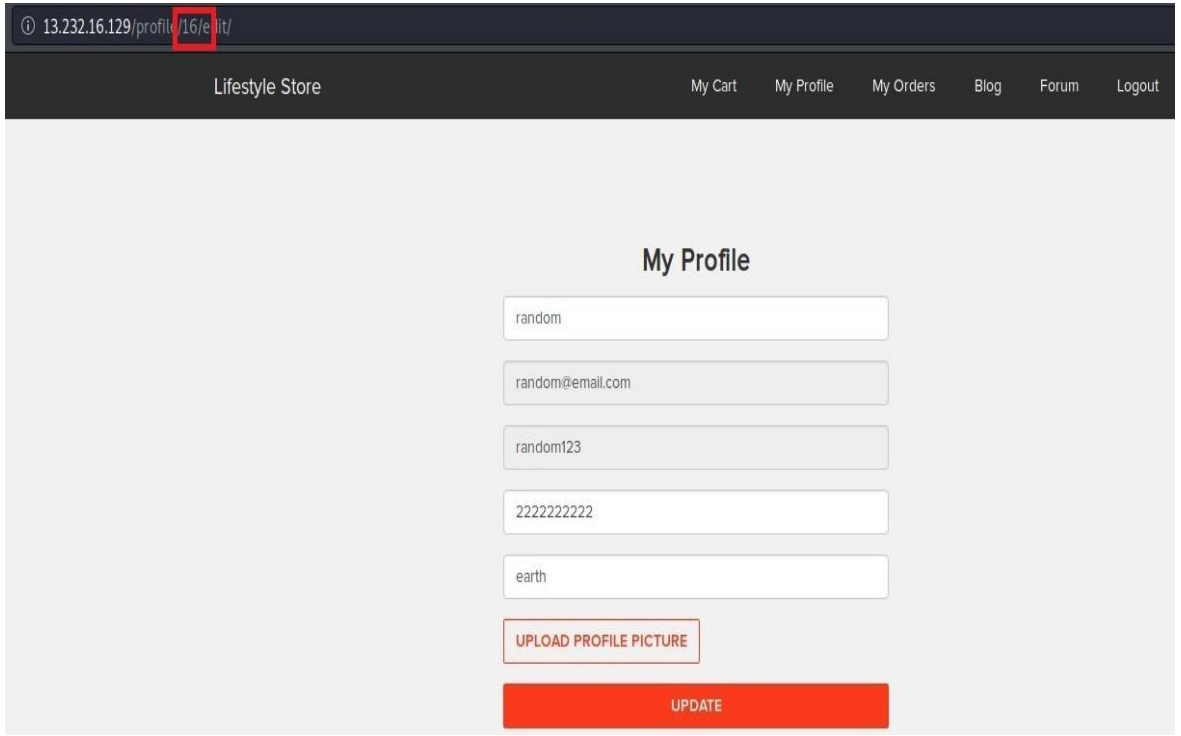
Insecure direct object references (IDOR) are a type of access control vulnerability that arises when an application uses user-supplied input to access objects directly.

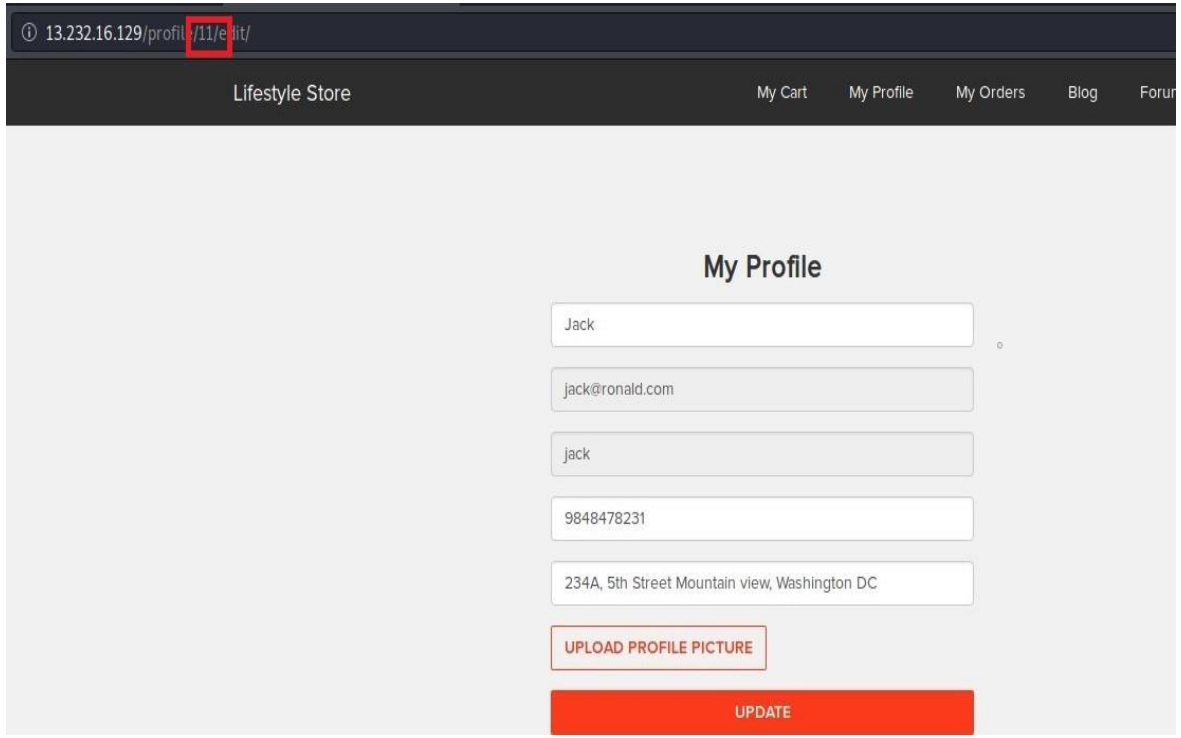
IDOR	**URL:** after clicking on the my profile module we can edit our profile http://13.232.16.129/profile/profile.php . Click on “EDIT PROFILE” **Affected URL :** http://13.232.16.129/profile/16/edit/ **Method used:** GET based **Payload :** profile/16

IDOR	**URL :** http://13.232.16.129/products.php the “My Orders” is vulnerable to IDOR vulnerability. **Affected URL :** http://13.232.16.129/orders/orders.php?customer=16 **Method used :** GET based **Payload :** customer=16

Observation

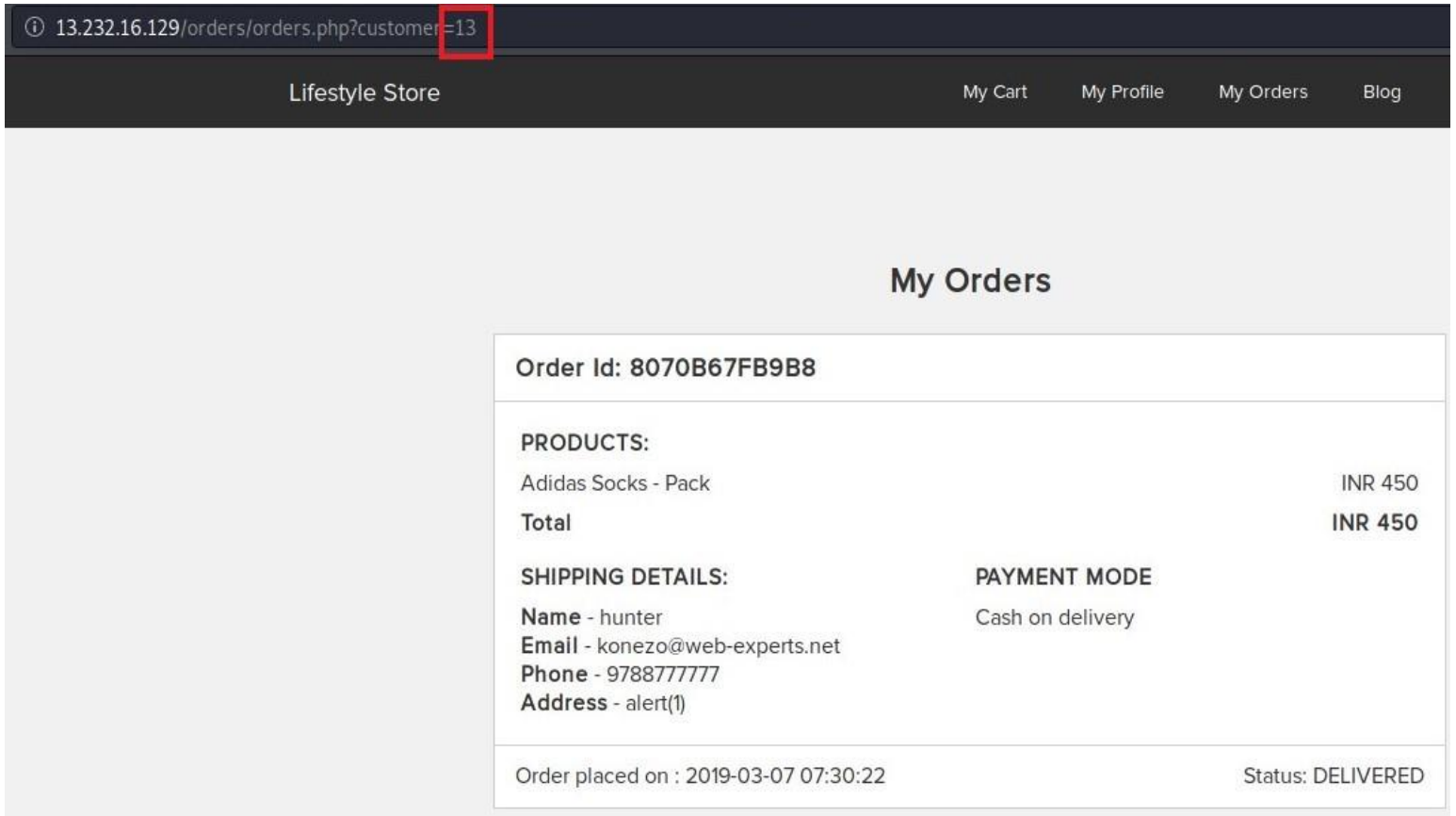
By changing the value form “16” to “11” the hacker can get access to other customer’s profile and change their details.





Proof of Concept(PoC)

By changing the parameter “customer=16” to “customer=13” the hacker can get access to the other customer details and see their orders and other details.



Business Impact – Critical

- With this vulnerability the hacker can get unauthorized access to the customers details and their personal information like address, phone number , email are all disclosed.

- The company may fall into severe trouble as this is a security flaw and the company may be seized for leaking the data.

Recommendation

- Validation of Parameters should be properly implemented.
- Verification of all the Referenced objects should be done.
- Developers should avoid displaying private object references such as keys or file names.

References

- https://www.owasp.org/index.php/Insecure_Configuration_Management
- https://www.geeksforgeeks.org/insecure-direct-object-reference-idor-vulnerability/

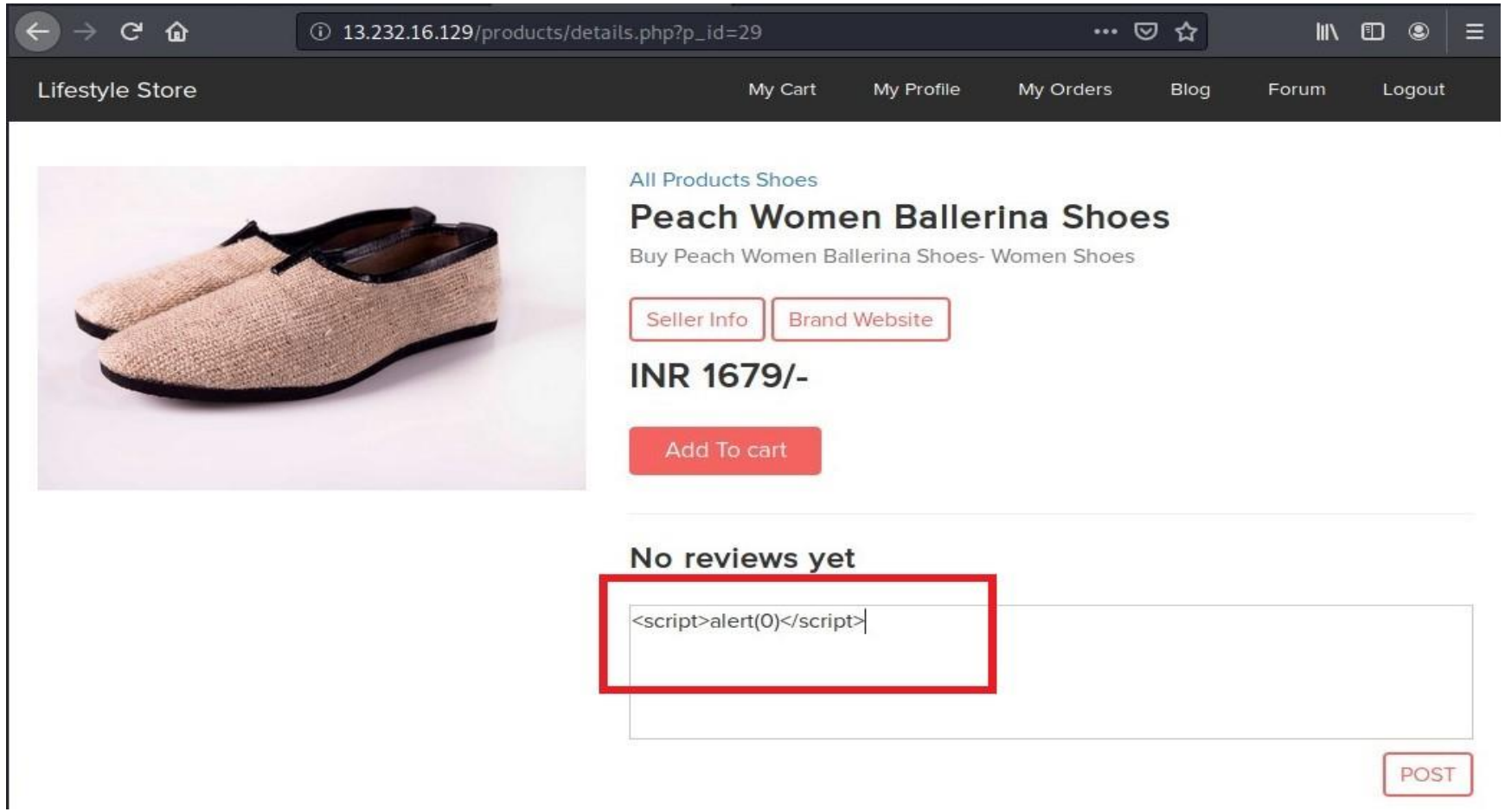
3. Cross Site Scripting

Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application.

XSS	**URL :** http://13.232.16.129/products.php under this URL there are three modules T Shirt/Shoes/Socks **Affected URL :** http://13.232.16.129/products/details.php?p_id=29 **Method Used :** GET based **Payload :** <script>alert(0)</script>

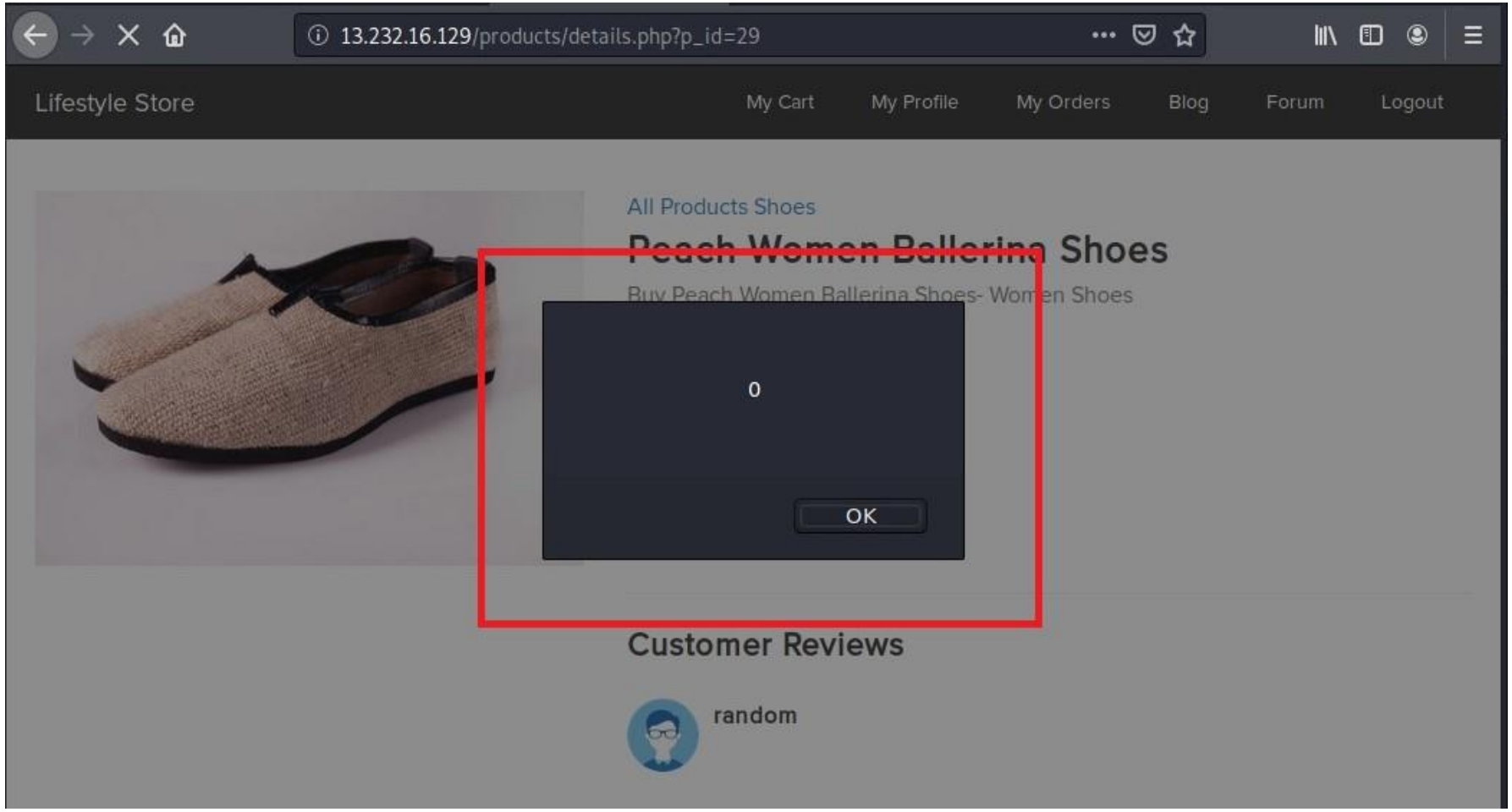
Observation

With this vulnerability the hacker can execute malicious Java Script codes and cause harm to the servers and the database.



Proof of Concept(PoC)

In the revies tab type “**<script>alert(0)</script>**” and post the code and then a **pop up box** appears.



Business Impact – Severe

- As the hacker can inject HTML , CSS and JS via the review box the hacker can hack the website and gain complete control over the server.

- With the help of this vulnerability the hacker can now gain complete access to the victim’s device and steal the information or even post some explicit content on the website.

Recommendations

- **Filter input on arrival** At the point where user input is received, filter as strictly as possible based on what is expected or valid input.

- **Encode data on output** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.

- **Content Security Policy** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.

References

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://portswigger.net/web-security/cross-site-scripting

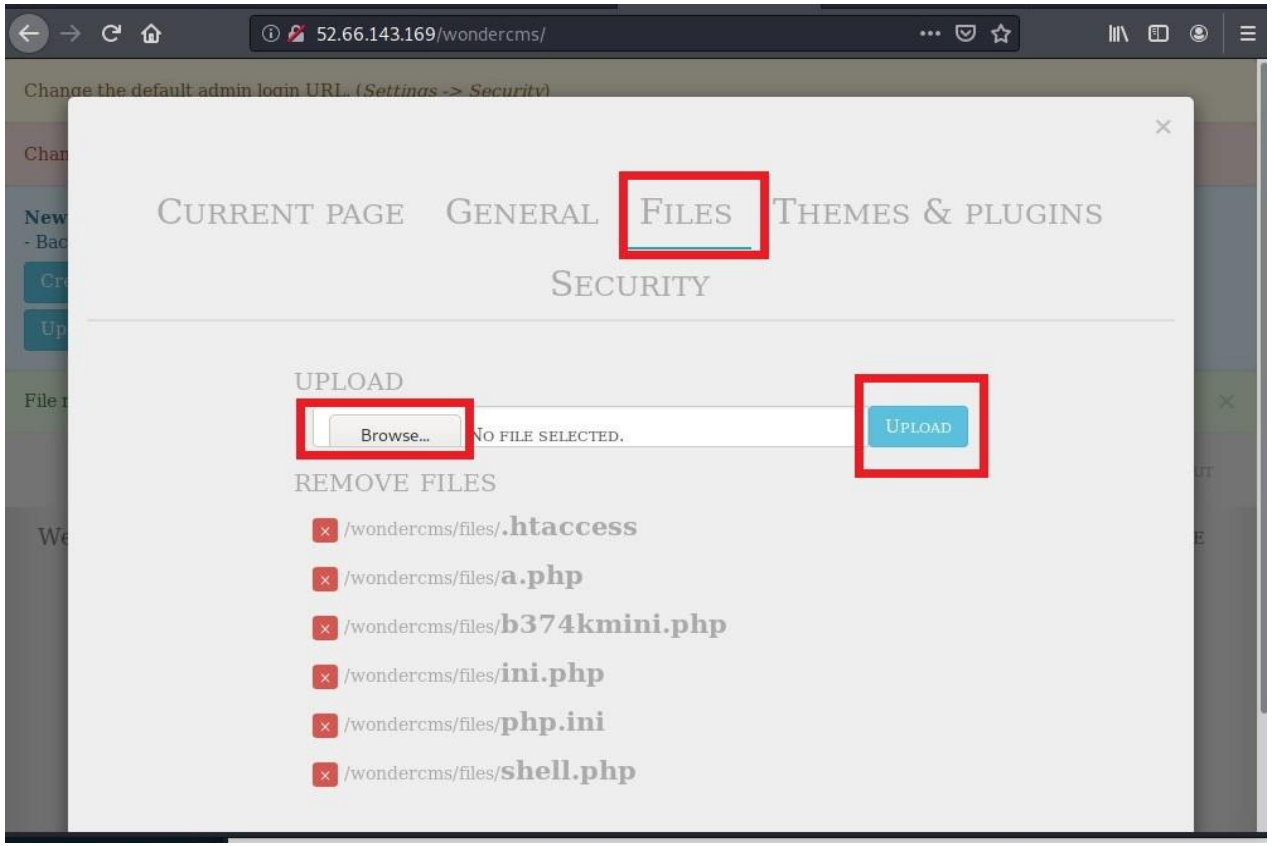
4. Arbitrary File Upload

This type of vulnerability occurs when an application on the website receives user's instructions to download the desired file from somewhere on the Internet and store it, and then the hacker executes this file to cause problems.

Arbitrary File Upload	**URL :** http://52.66.143.169/ in this website the Blog module is vulnerable to arbitrary file upload vulnerability **Method Used :** GET based **Affected URL :** http://52.66.143.169/wondercms/ **Parameter :** Files module under Settings **Payload :** sample.php (basic php program)

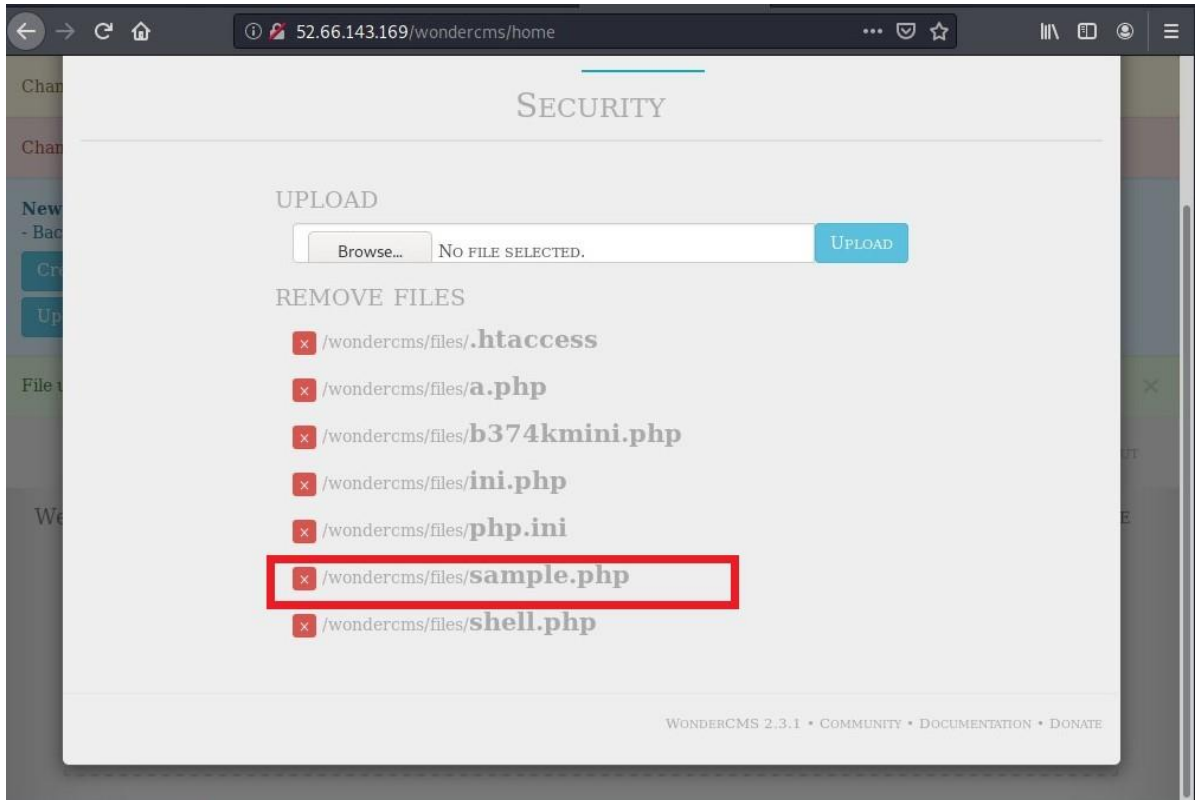
Observation

After logging in into the admin account in the WonderCMS navigate to settings and there under the Files we can upload the files.



Proof of Concept (PoC)

- After uploading the file we can run the program by clicking on it.
- In this way the hacker upload malicious files and run it on the website and can take complete control of the website easily.



Business Impact - Critical

- With this vulnerability the hacker can easily upload files and hack the website and the database.
- Files can be uploaded or deleted.

Recommendation

- The default password should be changed.
- Before uploading the files must be verified properly.

References

- https://www.getastra.com/e/malware/infections/arbitrary-file-upload-vulnerability
- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

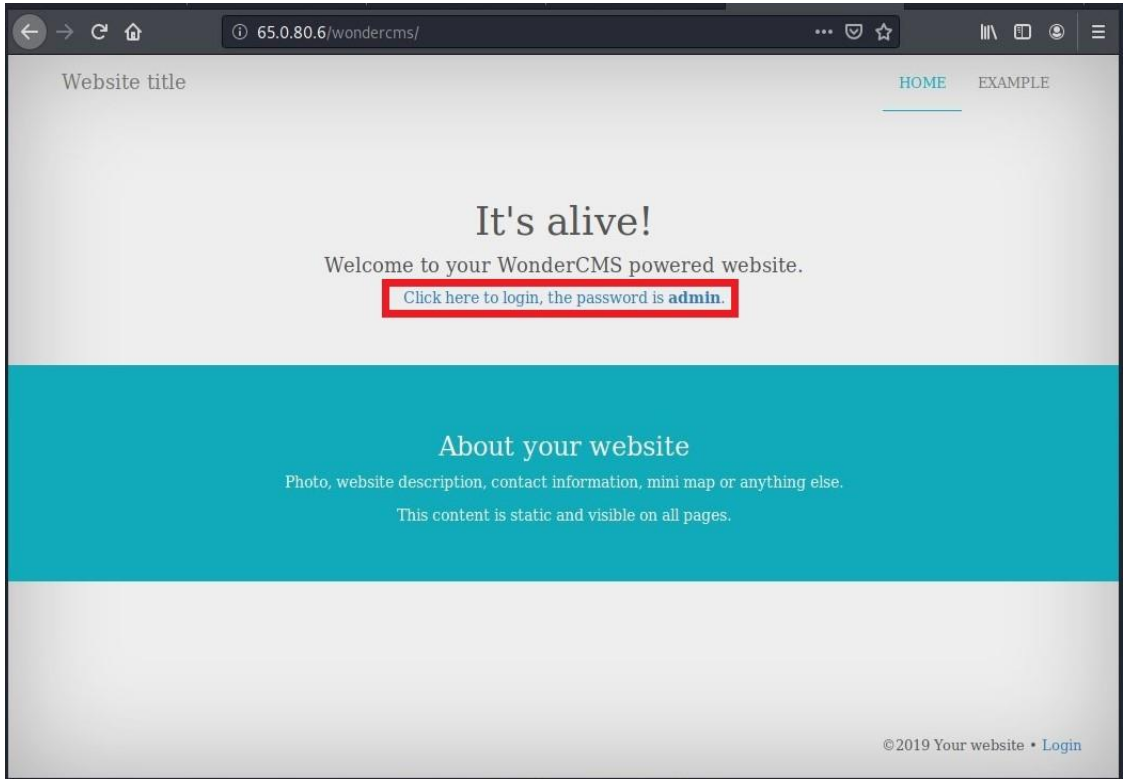
5. Access to Admin page

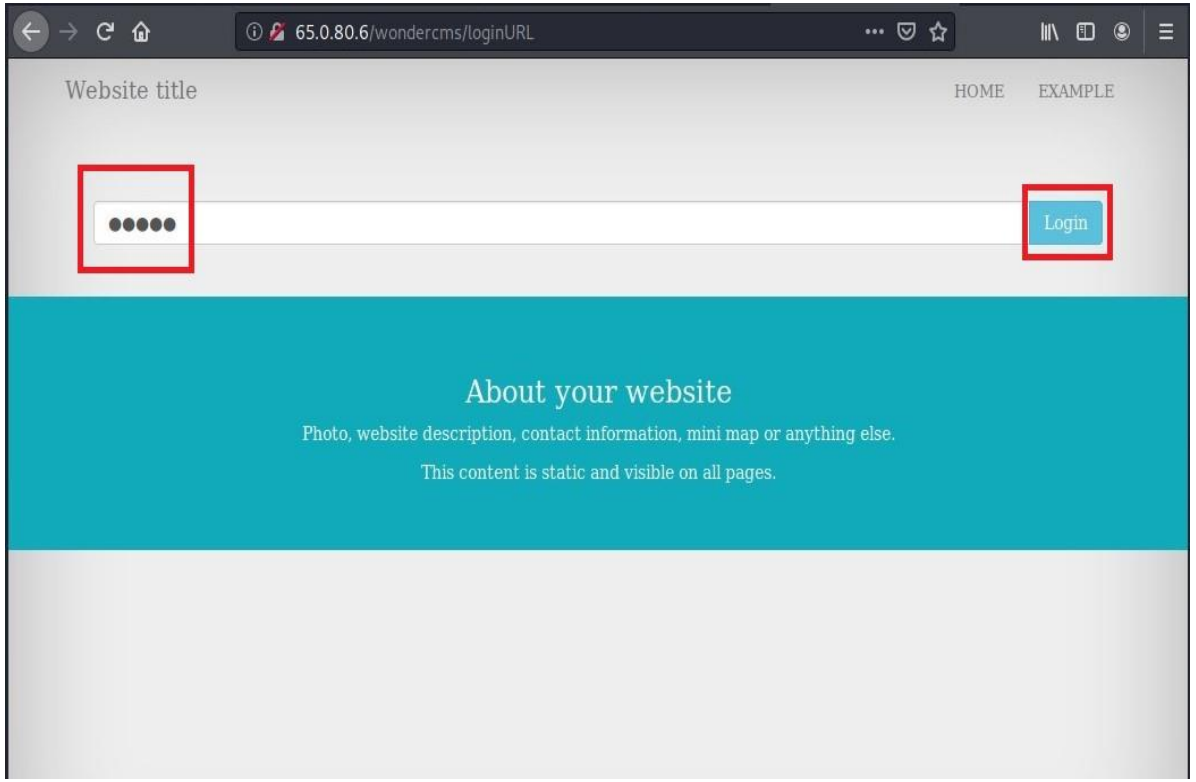
With this vulnerability the hacker can access the admin panel and change the data of the blog and delete the data and cause harm to the website.

Access to Admin page	**URL :** http://65.0.80.6/wondercms/ and click on the login **Affected URL :** http://65.0.80.6/wondercms/loginURL

Observation

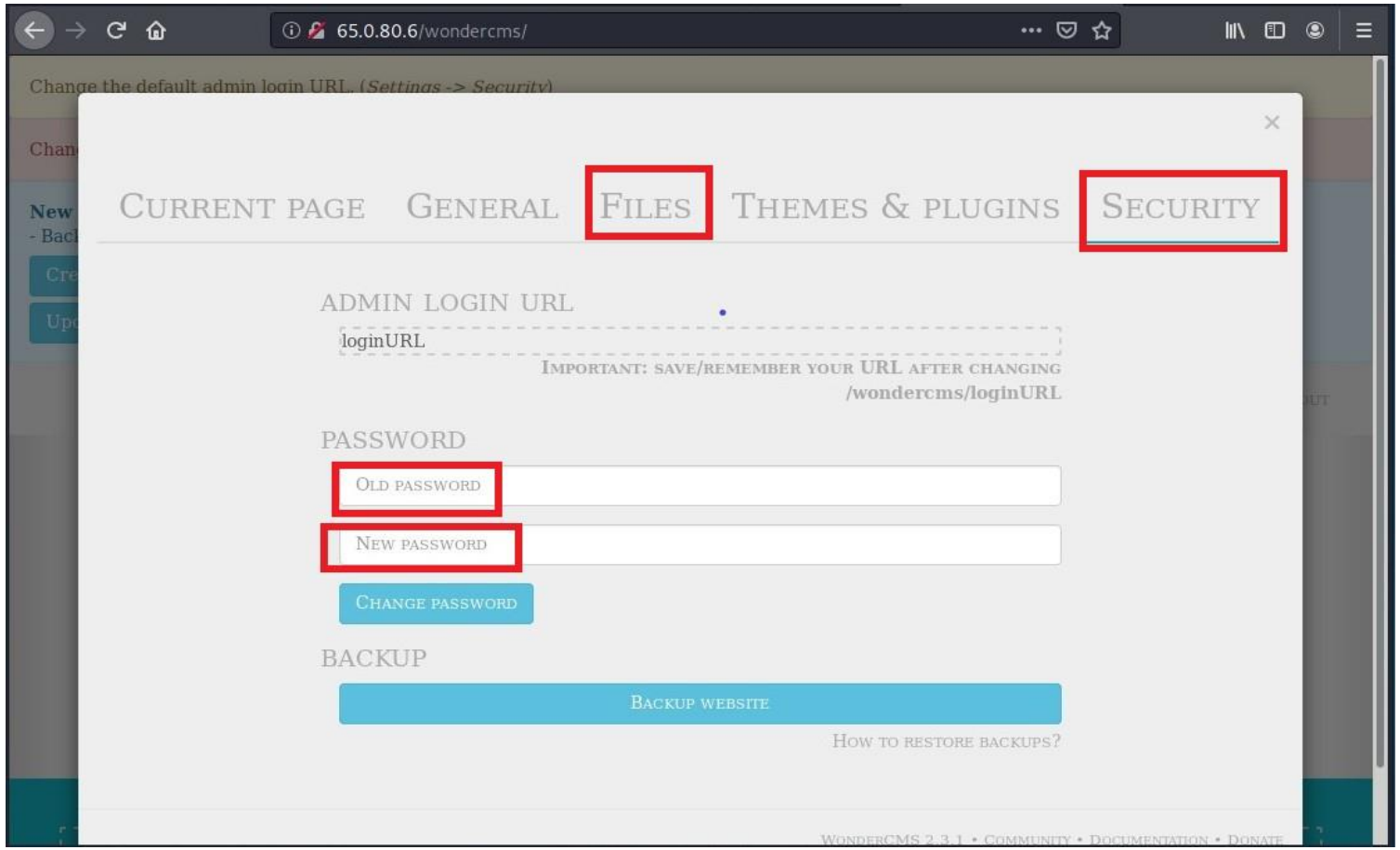
The hacker now logged in as an Admin and can tamper with the data.





Proof of Concept (PoC)

The hacker can now delete Files in the file panel and also change the password of the admin in the security panel.



Business impact – Critical

- Using this vulnerability ,the attacker can get complete access to the blog of the website.
- Files can be deleted and can be very dangerous to the website , as the entire website is in the hands of the hacker.
- The hacker can change the password of the admin log in credentials and not allow the actual admin to access the page.

Recommendation

- The default password should be changed and a strong password must be setup.
- The password must not published on the website and the password should be very strong and minimum of 8 characters long.

References

- https://www.owasp.org/index.php/Default_Passwords

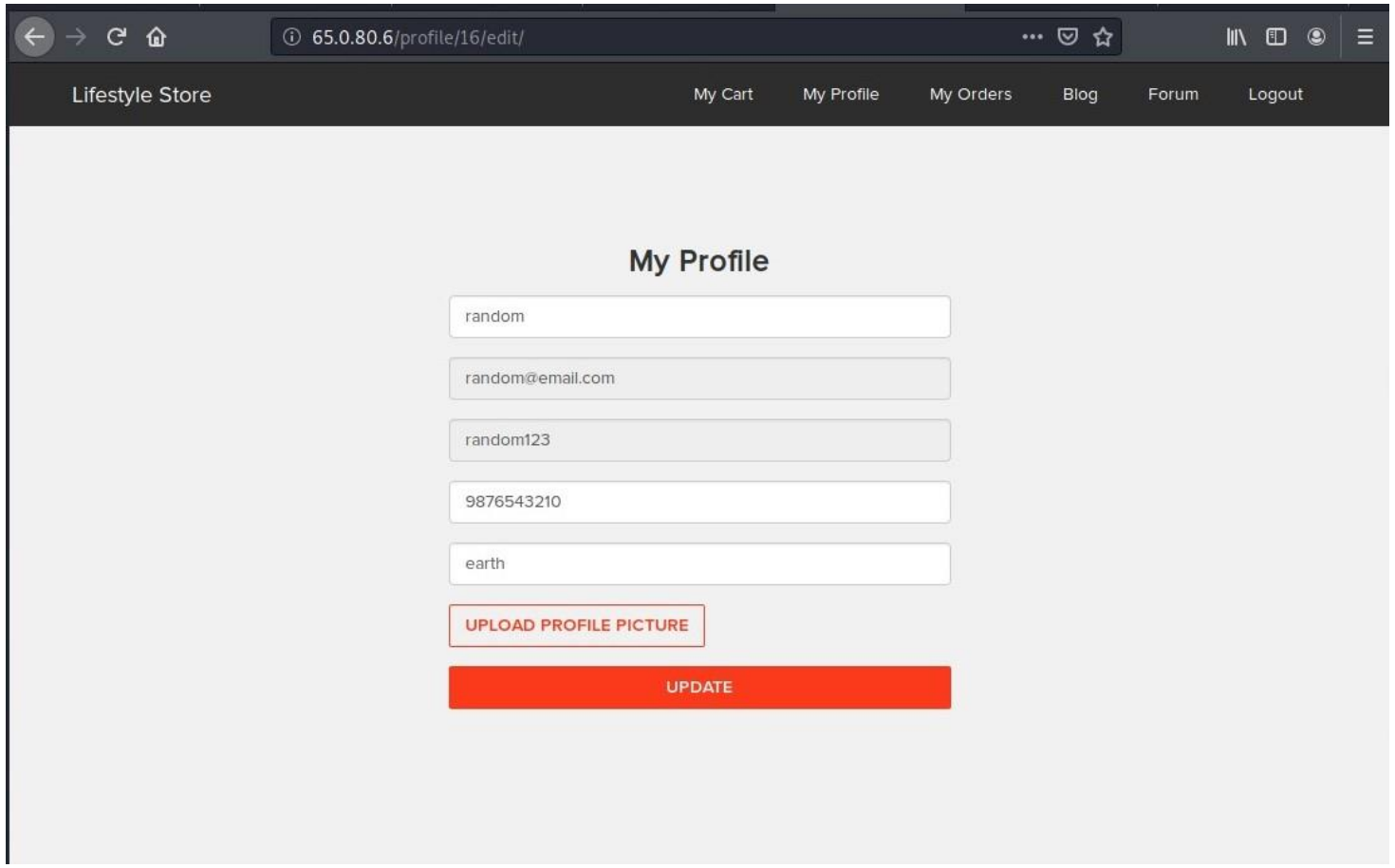
6. Forced Browsing

Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the application, but are still accessible.

Forced Browsing	**URL :** http://65.0.80.6/profile/profile.php in this site the My Profile module is vulnerable to forced browsing. By clicking the edit option we are redirected to a new page. **Affected URL :** http://65.0.80.6/profile/16/edit/ **Method used :** GET based **Payload** : http://65.0.80.6/profile/16/edit/

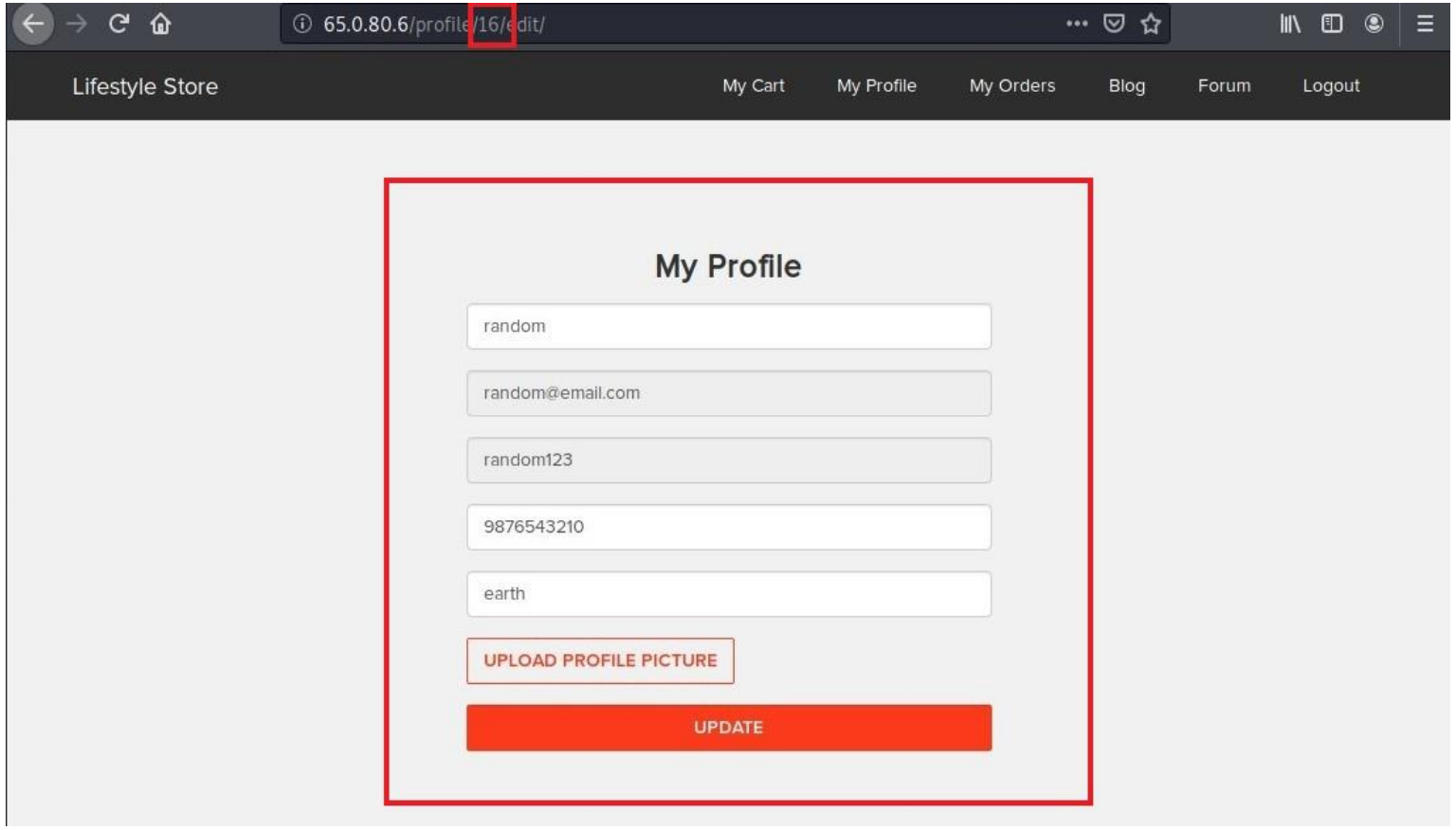
Observation

After going into the edit option the hacker can copy the URL and paste it in the new tab and can still able to access the page.



Proof of Concept (PoC)

The hacker can now access the data in two different tabs and change the parameters in the URL and access other customer's details.



Business Impact - Severe

The potential impact of forced browsing includes unauthorized access to all administration functions and to other user's personal information.

Recommendation

- The developer must never assume that a publicly accessible URL is impossible to find. If it exists, it can be found. Authentication is a must.
- The developer must never assume that once the user is authenticated, they don't need any other access control.

References

- https://www.acunetix.com/blog/web-security-zone/what-is-forced-browsing/
- https://owasp.org/www-community/attacks/Forced_browsing

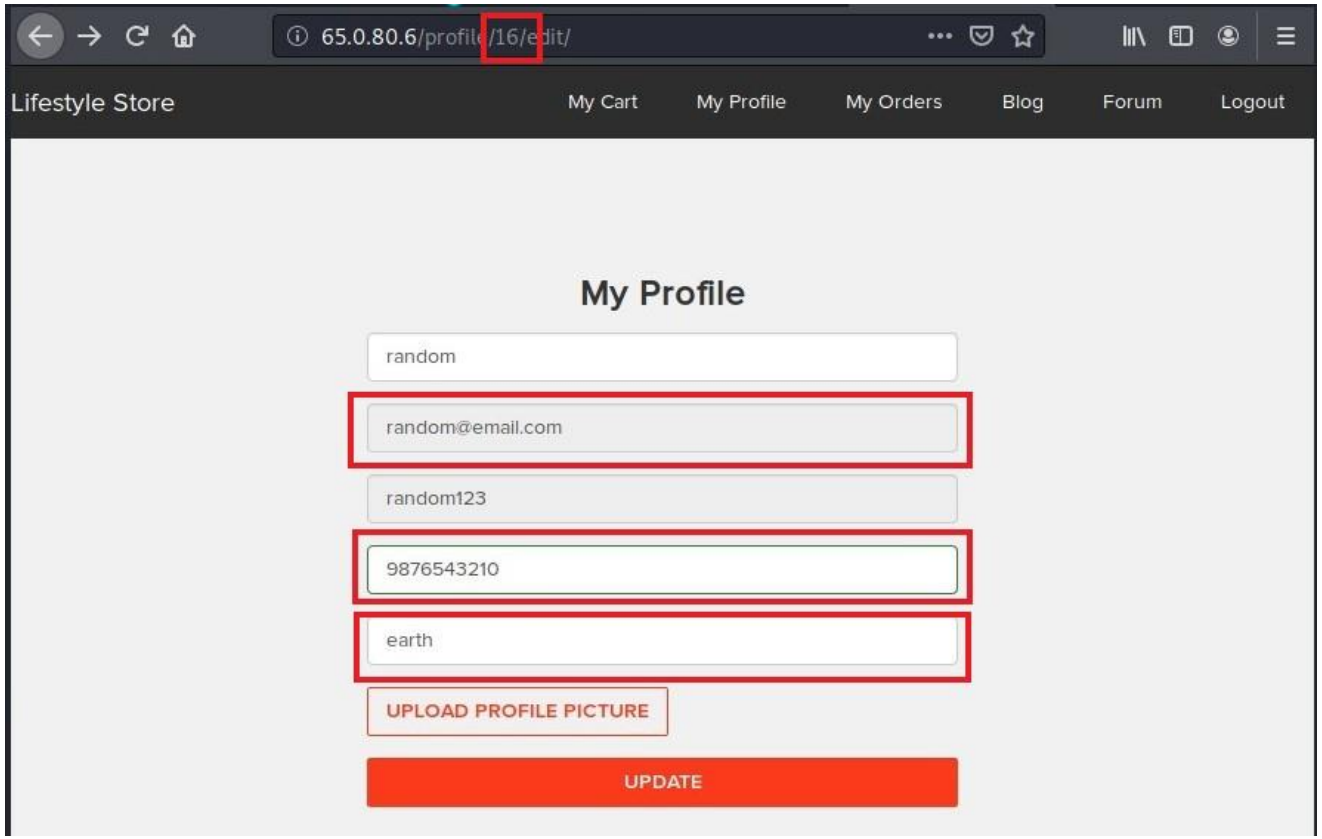
7. Missing Server Side Validation

With this vulnerability the hacker can bypass some client side validation filters.

Missing server side validation	**URL :** http://65.0.80.6/profile/profile.php in this site the My Profile module is vulnerable missing server side validation vulnerability. By clicking the edit option we are redirected to a new page. **Affected URL :** http://65.0.80.6/profile/16/edit/ **Method used :** GET based **Affected Parameter :** contact number **Payload :** 6969696969

Observation

The vulnerability allows the hacker to change the data by intercepting the packet using Burp Suite and tamper the data without validating the data. Ex: contact number , address , name etc..



Proof of Concept(PoC)

The hacker can now create multiple accounts and change the data of the other customers with improper data by bypassing the filters.

Request to http://65.0.80.6:80

Forward | Drop | Intercept is on | Action | Open Browser | Comment this item

Pretty | Raw | \n | Actions

```
1  POST /profile/submit.php HTTP/1.1
2  Host: 65.0.80.6
3  User-Agent: [redacted]
4  Accept: text/plain, */*; q=0.01
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://65.0.80.6/profile/16/edit/
8  X-Requested-With: XMLHttpRequest
9  Content-Type: multipart/form-data; boundary=---------------------------666079074170629916544295895 8
10 Content-Length: 705
11 Connection: close
12 Cookie: key=F958CFD5-7D5D-D82A-1FCE-B9B78E39CB8A; PHPSESSID=j0o14t1t9d737hqvc8p865qmcl; X-XSRF-TOKEN=
   7ca6b449098ab61e7bb13c02299d736455528544e21c56dbaf1e5dba22687322
13
14 -----------------------------666079074170629916544295895 8
15 Content-Disposition: form-data; name="name"
16
17 random
18 -----------------------------666079074170629916544295895 8
19 Content-Disposition: form-data; name="contact"
20
21 6969696969
22 -----------------------------666079074170629916544295895 8
23 Content-Disposition: form-data; name="address"
24
25 earth
26 -----------------------------666079074170629916544295895 8
27 Content-Disposition: form-data; name="user_id"
28
29 16
30 -----------------------------666079074170629916544295895 8
31 Content-Disposition: form-data; name="X-XSRF-TOKEN"
```

65.0.80.6/profile/profile.php

Lifestyle Store | My Cart | My Profile | My Orders | Blog | Forum | Logout

My Profile

random
random@email.com

Username:	random123
Contact No.:	6969696969
Delivery Address:	earth

EDIT PROFILE | CHANGE PASSWORD

Business Impact - Moderate

- By changing the data the database will be inconsistent.

Recommendation

- Implement all critical checks on server side code only.
- Proper filets must be used to validate the information.

References

- https://cwe.mitre.org/data/definitions/20.html
- https://owasp.org/www-community/vulnerabilities/Improper_Data_Validation

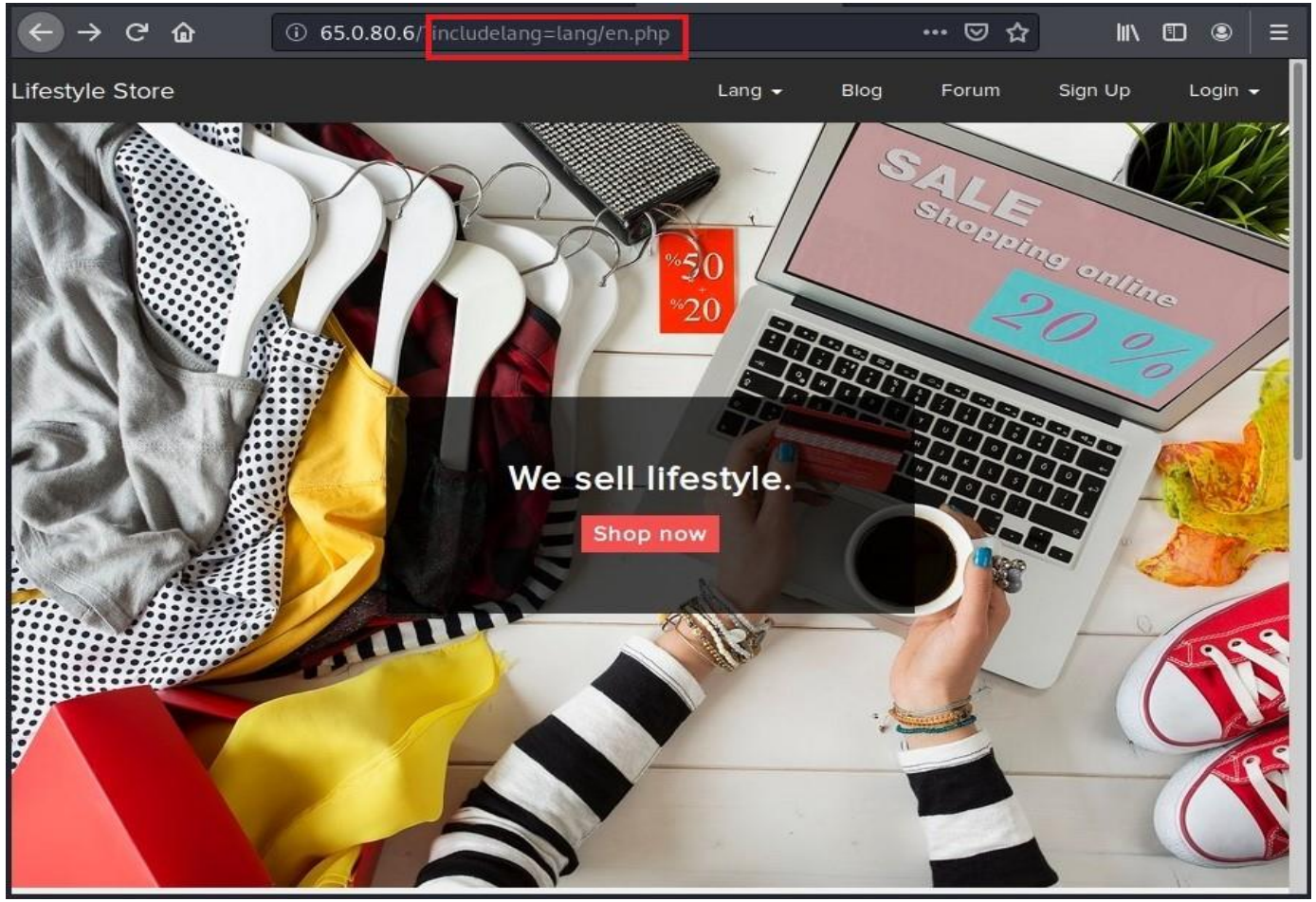
8. Open Redirection

Open redirection vulnerabilities arise when an application incorporates user-controllable data into the target of a redirection in an unsafe way.

Open Redirection	**URL :** http://65.0.80.6/ in this page the Lang module is vulnerable to Open Redirection **Affected URL :** http://65.0.80.6/?includelang=lang/en.php **Method Used :** GET based **Affected Parameter :** includelang=lang/en.php **Payload :** http://65.0.80.6/?includelang=https://github.com/?lang/en.php

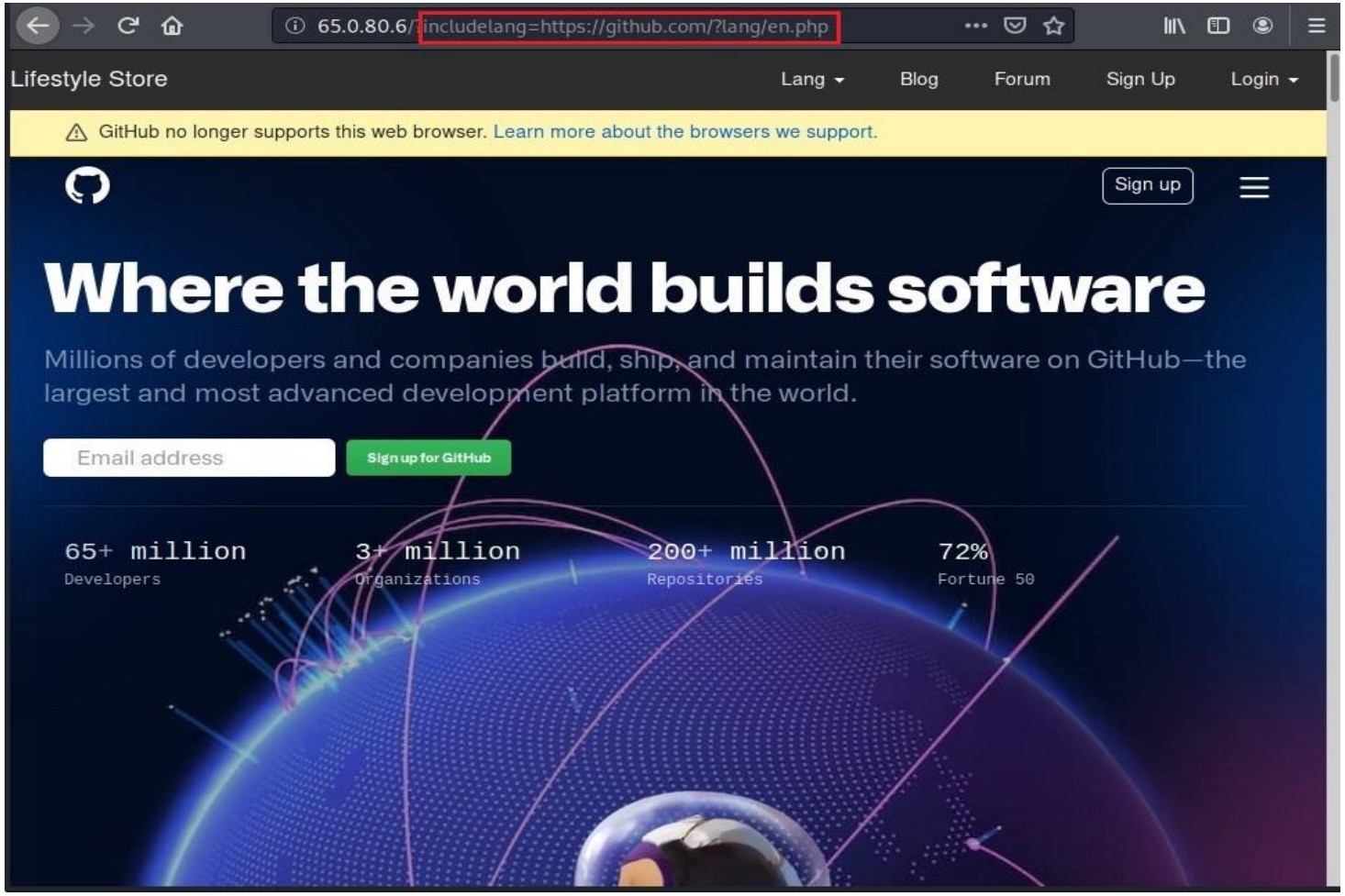
Observation

By changing the value in “**includelang**” parameter the victim will be redirected to another page.



Proof of Concept (PoC)

With this vulnerability the hacker can redirect the victim to some other malicious site and steal the data.



Business Impact – Severe

- An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL.
- Content-Security-Policy bypassing: If you use CSP to protect against XSS and one of the whitelisted domains has an open redirect, this vulnerability may be used to bypass CSP.

Recommendation

- Force all redirects to first go through a page notifying users that they are going off of your site, with the destination clearly displayed, and have them click a link to confirm.
- Check for http protocols.

References

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

- https://www.netsparker.com/blog/web-security/open-redirection-vulnerability-information-prevention/

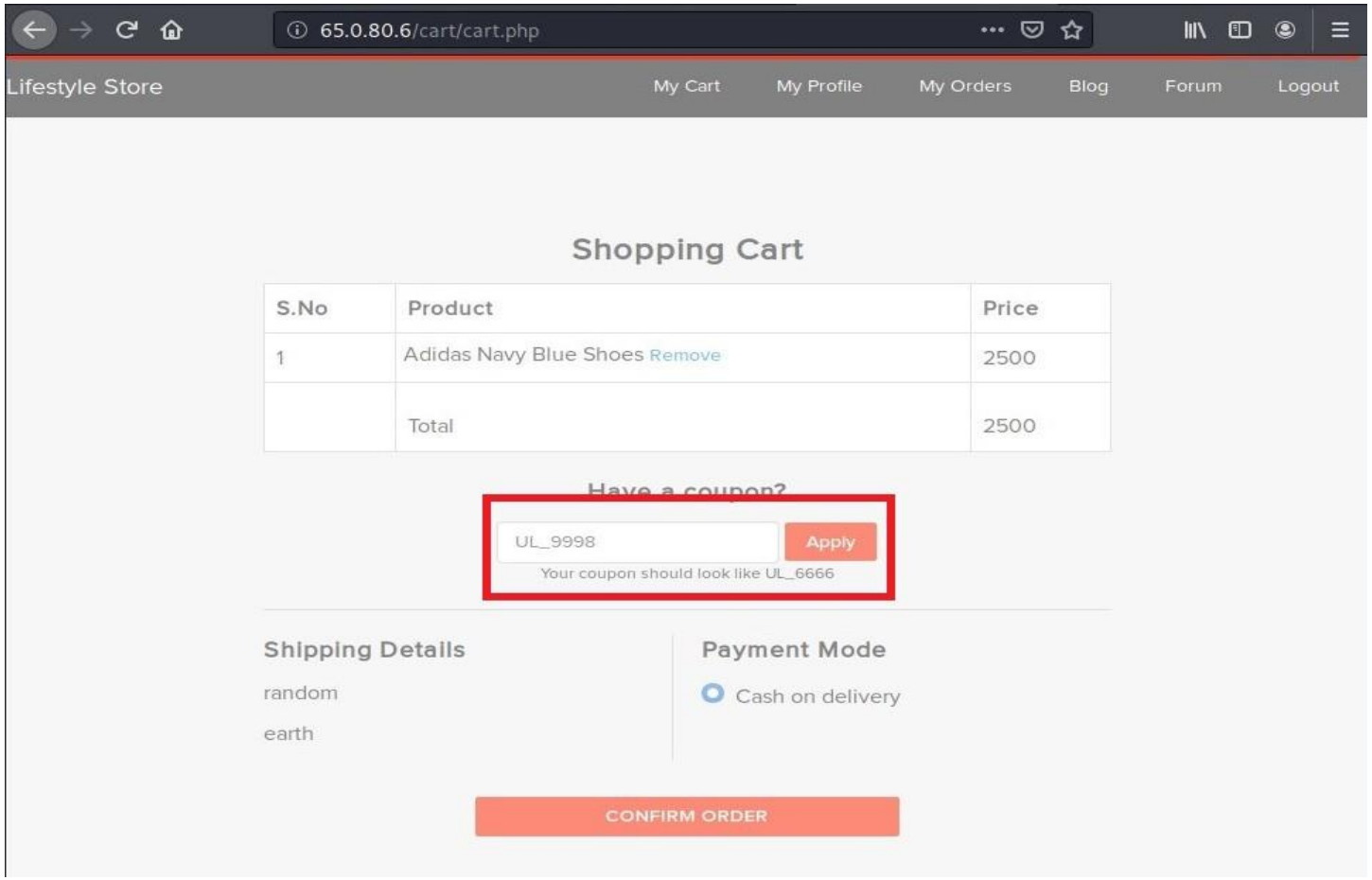
9. Brute Force

Brute force is an attack in which the hacker tries various combination to find the successful results through Burp Suite.

Brute Force	**URL :** http://65.0.80.6/products.php in this URL the My Cart module is vulnerable to Brute Force attacking. **Affected URL :** http://65.0.80.6/cart/cart.php **Method Used :** POST based **Affected Parameter :** Coupon Code **Payload :** UL_1056

Observation

With the help of Burp Suite the hacker can intercept the packet and through brute forcing he can guess the coupon code and can have discount on the products.



Proof of Concept (PoC)

After intercepting the packet the hacker can brute force various combination to find the correct coupon code.

Intercept | HTTP history | WebSockets history | Options

Request to http://65.0.80.6:80

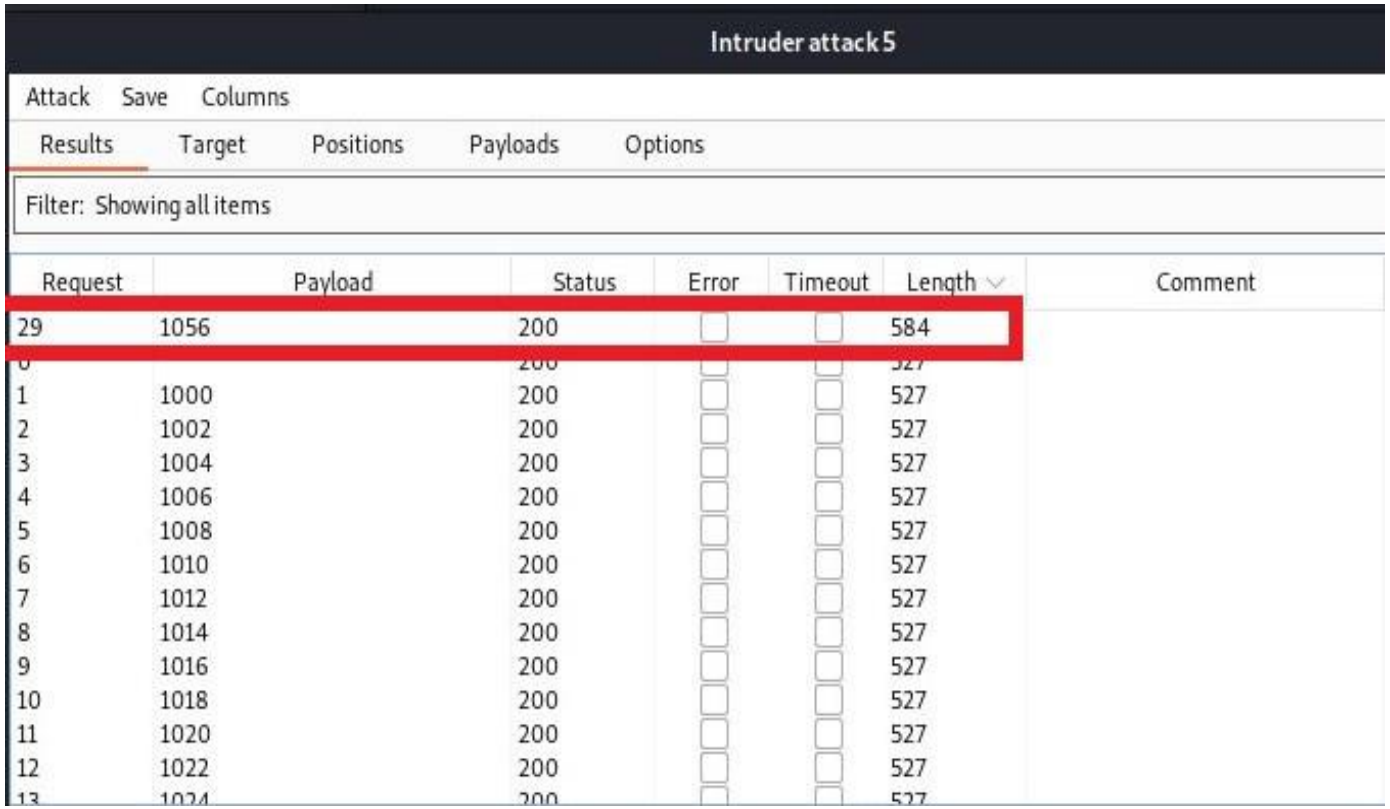
Forward | Drop | Intercept is on | Action | Open Browser | Comment this iter

Pretty | Raw | \n | Actions

```
1  POST /cart/apply_coupon.php HTTP/1.1
2  Host: 65.0.80.6
3  User [redacted]
4  Accept: */*
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://65.0.80.6/cart/cart.php
8  Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9  X-Requested-With: XMLHttpRequest
10 Content-Length: 92
11 Connection: close
12 Cookie: key=F958CFD5-7D5D-D82A-1FCE-B9B78E39CB8A; PHPSESSID=j0o14t1t9d737hqvc8p865qmcl; _octo=GH1.1.300458772.1627800534; tz=Asia%2FDili;
   X-XSRF-TOKEN=02c712acf1138e629ca4d95cbd71e1da0eab9ac089c4588c9c6f4a2fca68d3eb
13
14 coupon=UL_9998&X-XSRF-TOKEN=02c712acf1138e629ca4d95cbd71e1da0eab9ac089c4588c9c6f4a2fca68d3eb
```

Proof of Concept (PoC)

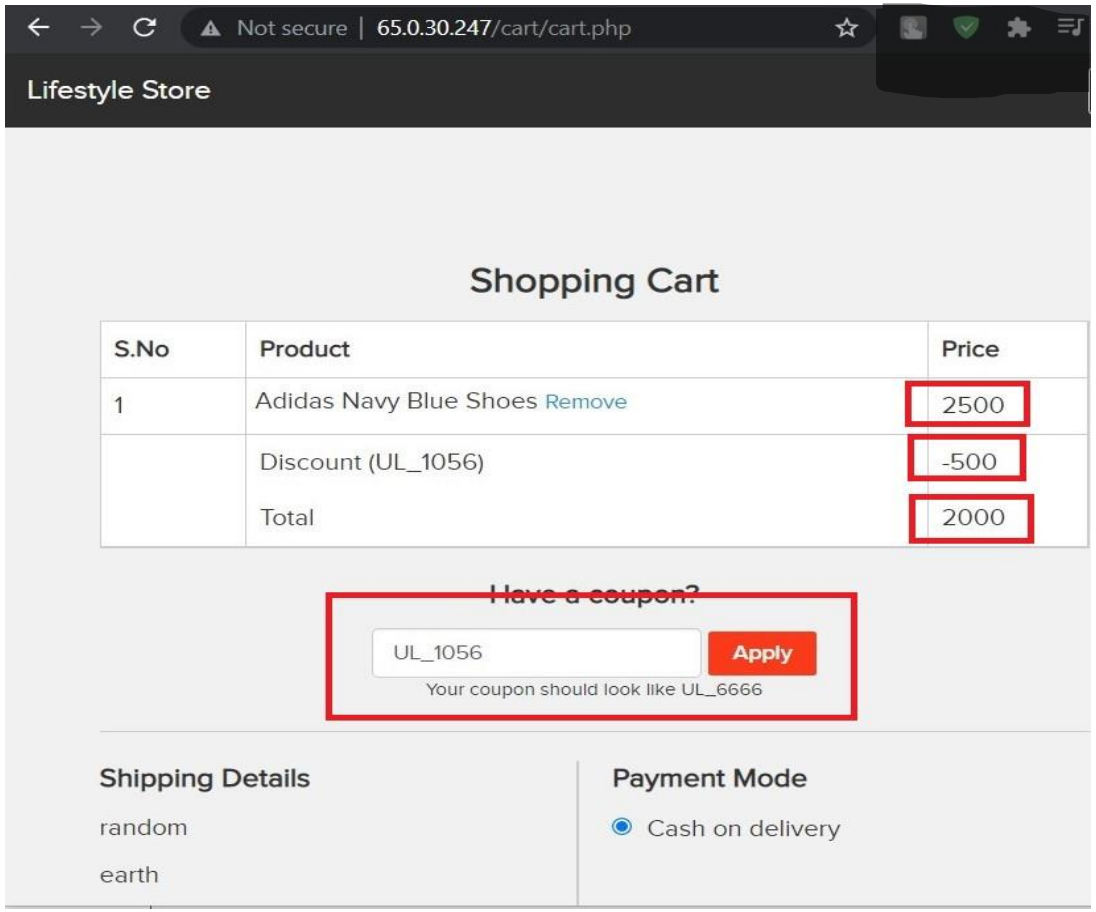
- The brute forcer will try various numbers and returns the valid number.
- After validating various combinations the valid coupon code found is UL_1056.



Request	Payload	Status	Error	Timeout	Length	Comment
29	1056	200			584	
0		200			527	
1	1000	200			527	
2	1002	200			527	
3	1004	200			527	
4	1006	200			527	
5	1008	200			527	
6	1010	200			527	
7	1012	200			527	
8	1014	200			527	
9	1016	200			527	
10	1018	200			527	
11	1020	200			527	
12	1022	200			527	
13	1024	200			527	

Proof of Concept (PoC)

The hacker now found a valid coupon code and can apply on any product he orders and can have the discount.



Business Impact – Severe

- The hacker can use n number of coupon codes and obtain discount on every product he purchases.
- The company can sustain loss due to this vulnerability.

Recommendation

- Use rate-limiting checks on the number of coupon Generation requests and validations.
- The length of the coupon code should be minimum of 8 characters.

References

- https://owasp.org/www-community/attacks/Brute_force_attack
- https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks

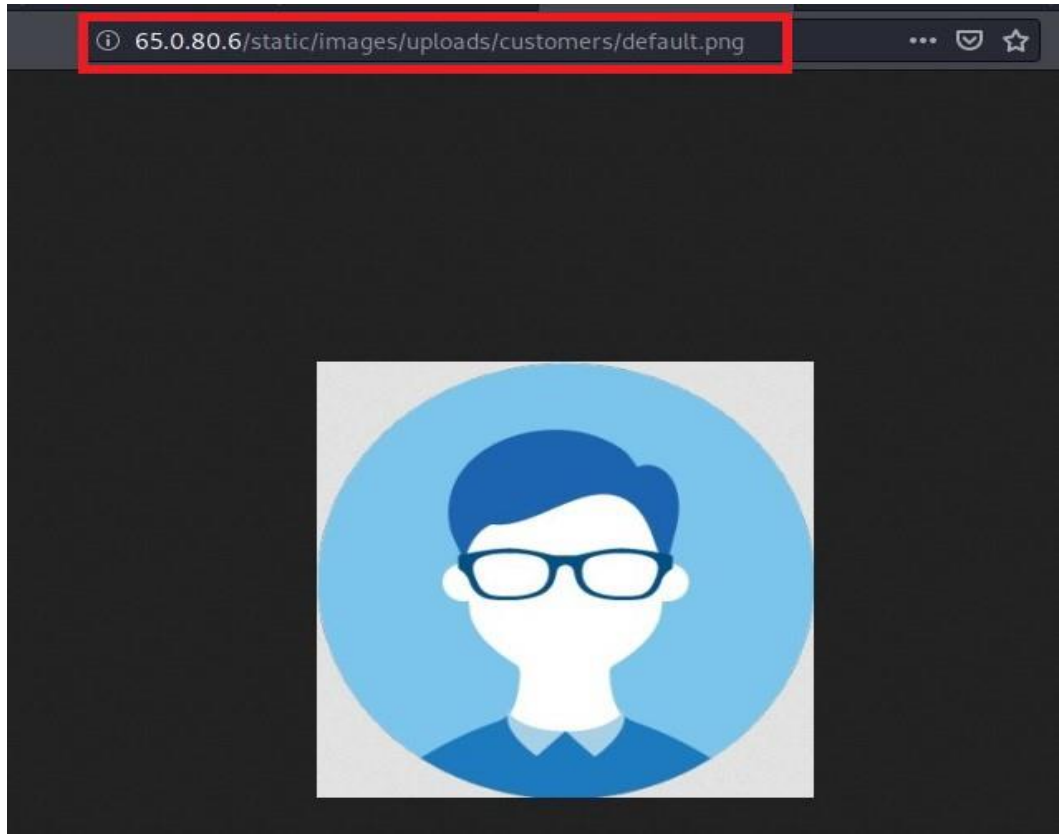
10. Personally Identifiable Information Leakage

This vulnerability can leak some personal information of the customer.

PII Leakage	**URL :** http://65.0.80.6/products.php in this site the module My Profile is vulnerable to PII Leakage. **Affected URL :** http://65.0.80.6/profile/profile.php **Method Used :** GET based **Payload :** image (display picture)

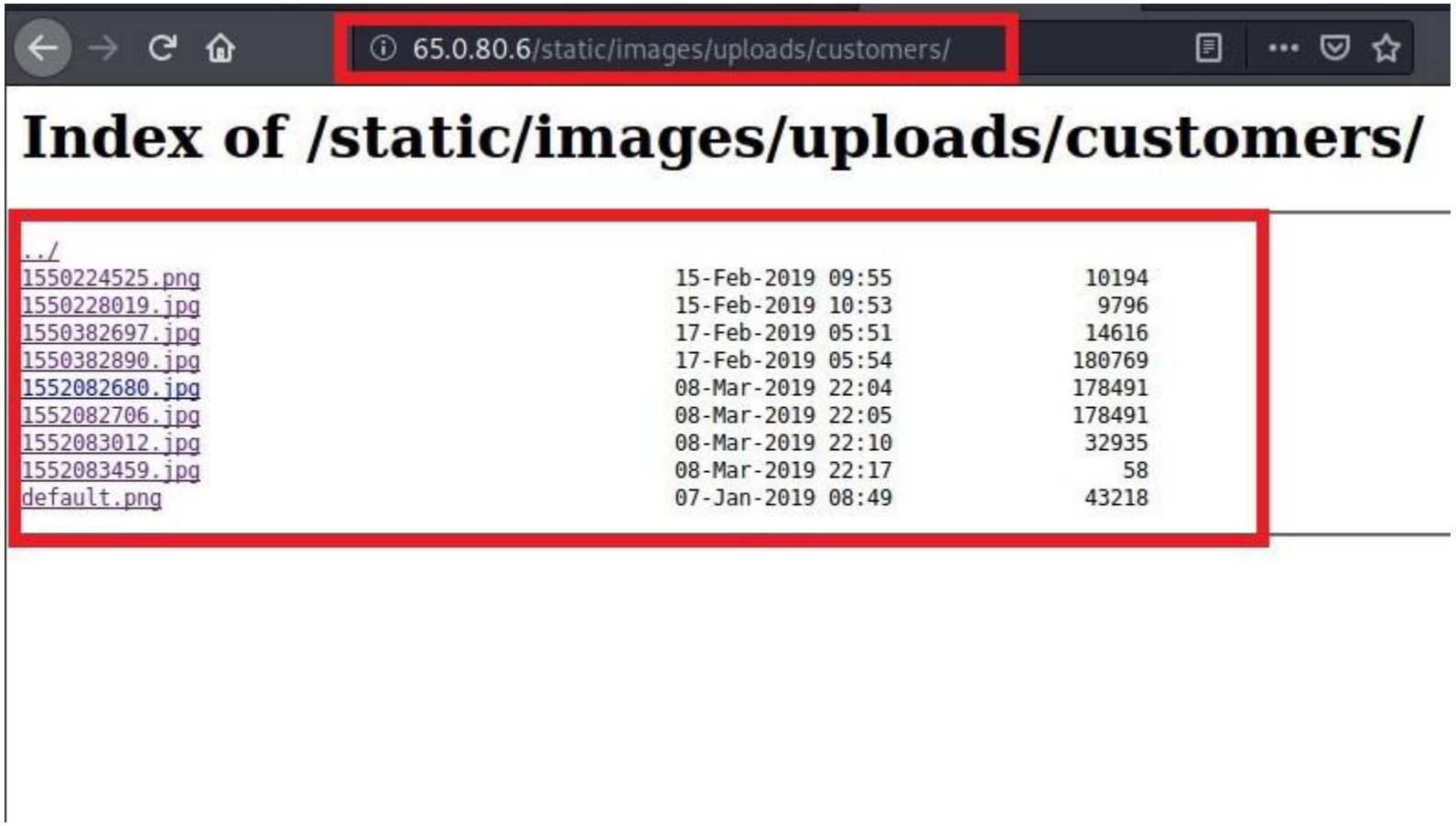
Observation

- After clicking on the “**My Profile**” module right click on the image and select view image and you will be redirected to a new tab in which the image is displayed.



Observation

- Now the hacker will delete the **".png"** parameter and gain access to the images database easily.

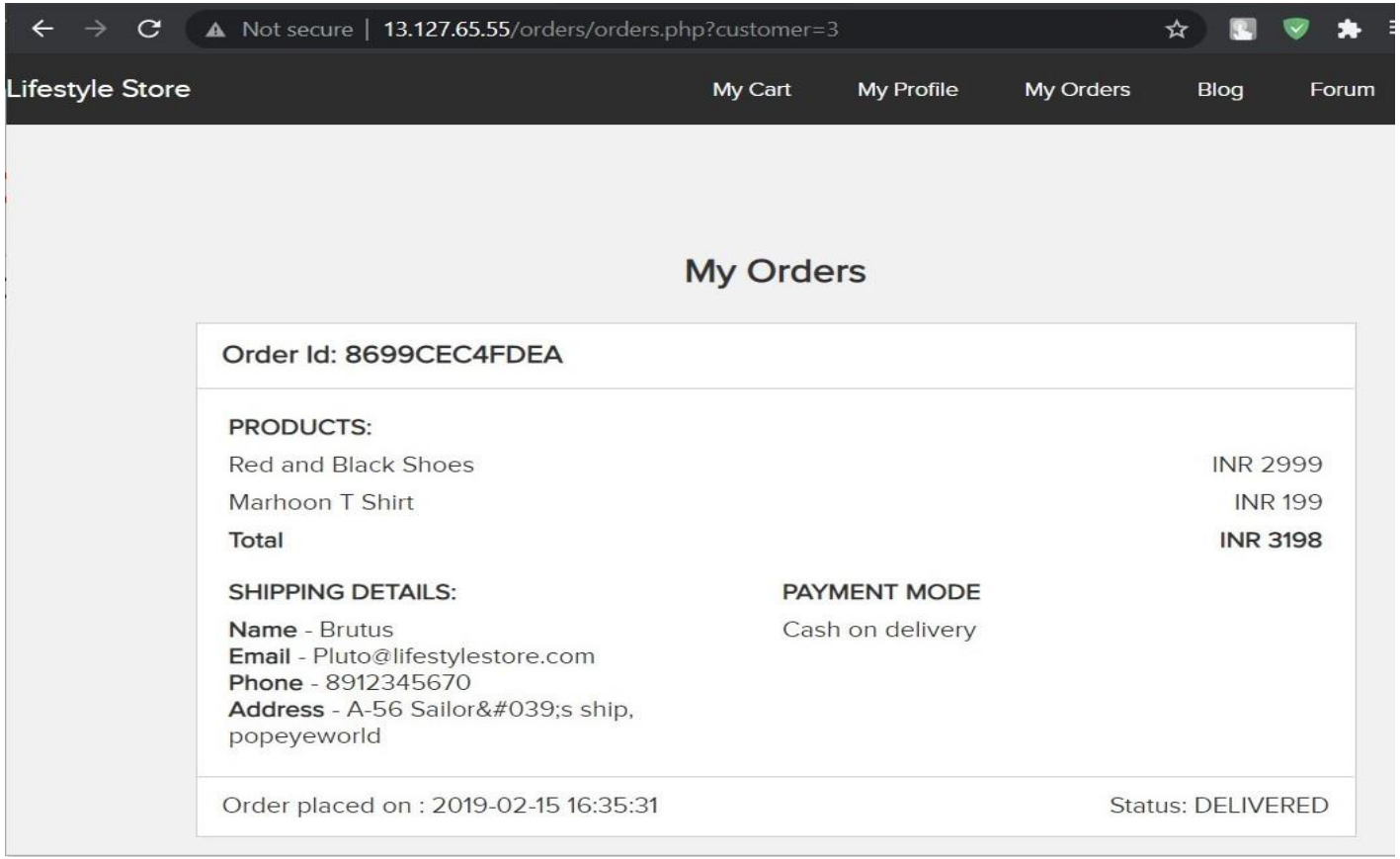


Proof of Concept (PoC)

With this vulnerability the hacker can gain access to other files and also the details of all the customers and their personal information.

Proof of Concept (PoC)

Through IDOR vulnerability the hacker can exactly pin point the victim easily as the victim's personal information are not secured.



Business Impact - Moderate

- The impact on the impact is not likely to happen but if the users get to know that their details are not safe then the buyers will decrease slowly.

Recommendations

- Safely Dispose or Destroy Old Media with Personal Data
- Establish an acceptable usage policy
- Encrypt PII

References

- https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise

- https://cipher.com/blog/25-tips-for-protecting-pii-and-sensitive-data/

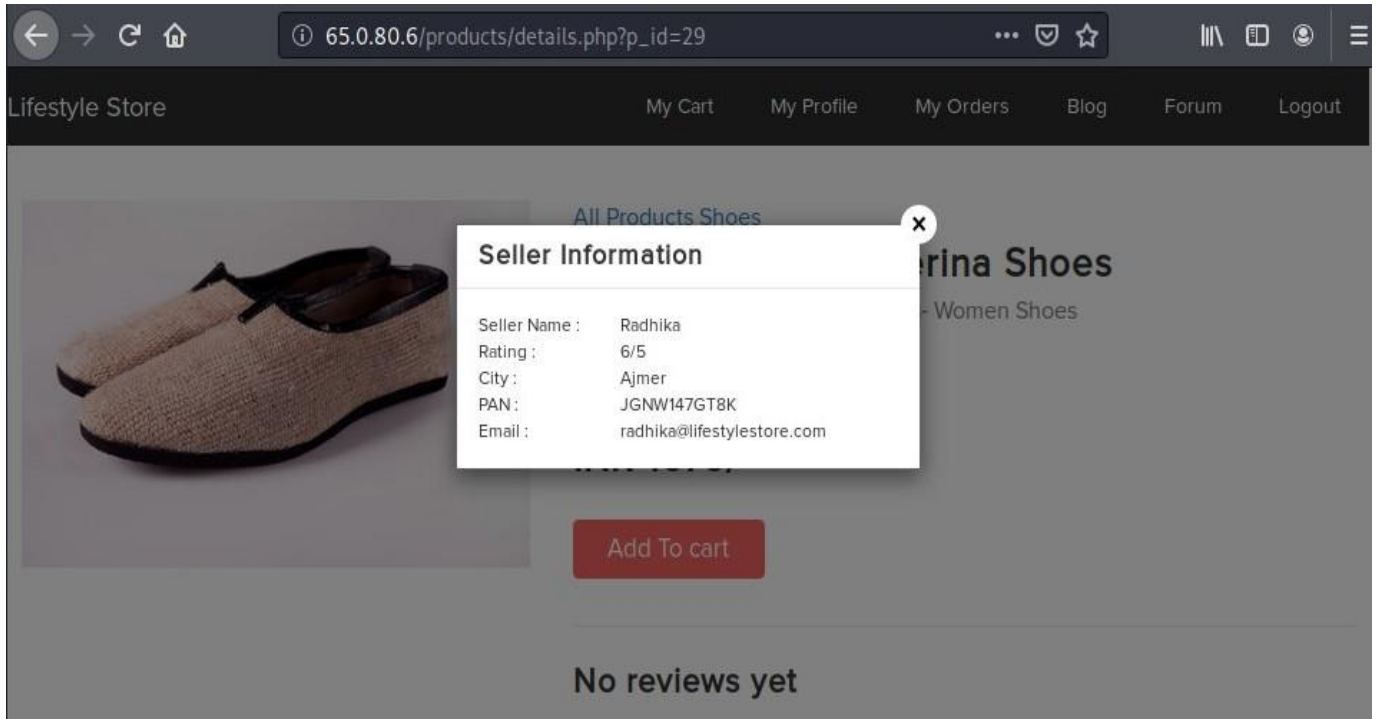
11. Unauthorized access to Seller’s Details

- The sellers details are not be published publicly.
- It should be stored securely.

Unauthorized access to Seller’s Details	**URL :** http://65.0.80.6/products/details.php?p_id=29 **Method Used :** GET based **Affected Module :** Seller info

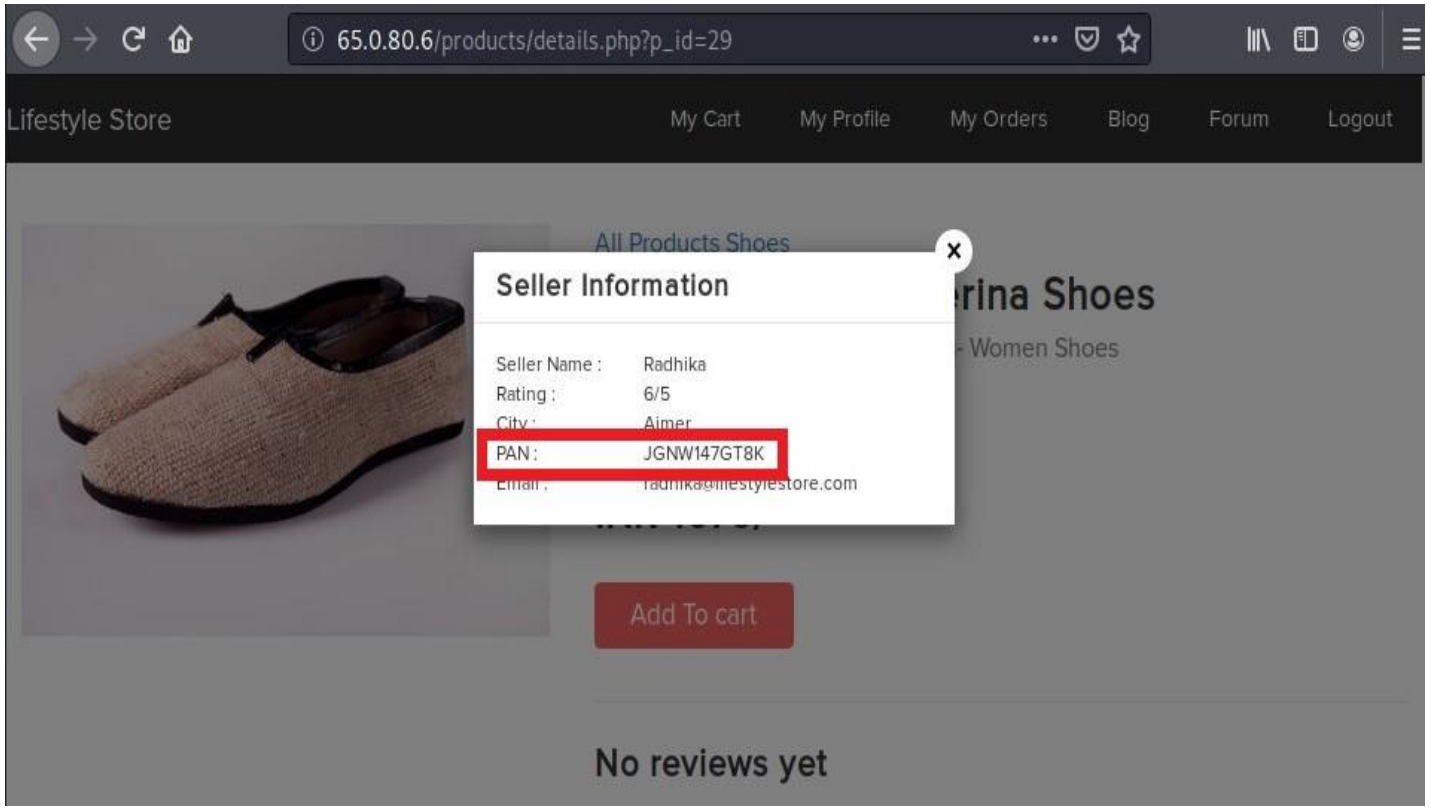
Observation

Some personal details of the sellers are being displayed publicly which not be the case.



Proof of Concept (PoC)

The PAN card number of the seller should not be displayed and instead it should be secured as if an malicious hacker could get hold.



Business Impact – Moderate

- The will be no direct impact on the business.
- The sellers could loose trust on the company and the dealing between them may get cancelled.

Recommendation

- No need to display the personal information of the seller like PAN card number etc..
- Securely store the data in the database.

References

- https://digitalguardian.com/blog/how-secure-personally-identifiable-information-against-loss-or-compromise

- https://cipher.com/blog/25-tips-for-protecting-pii-and-sensitive-data/

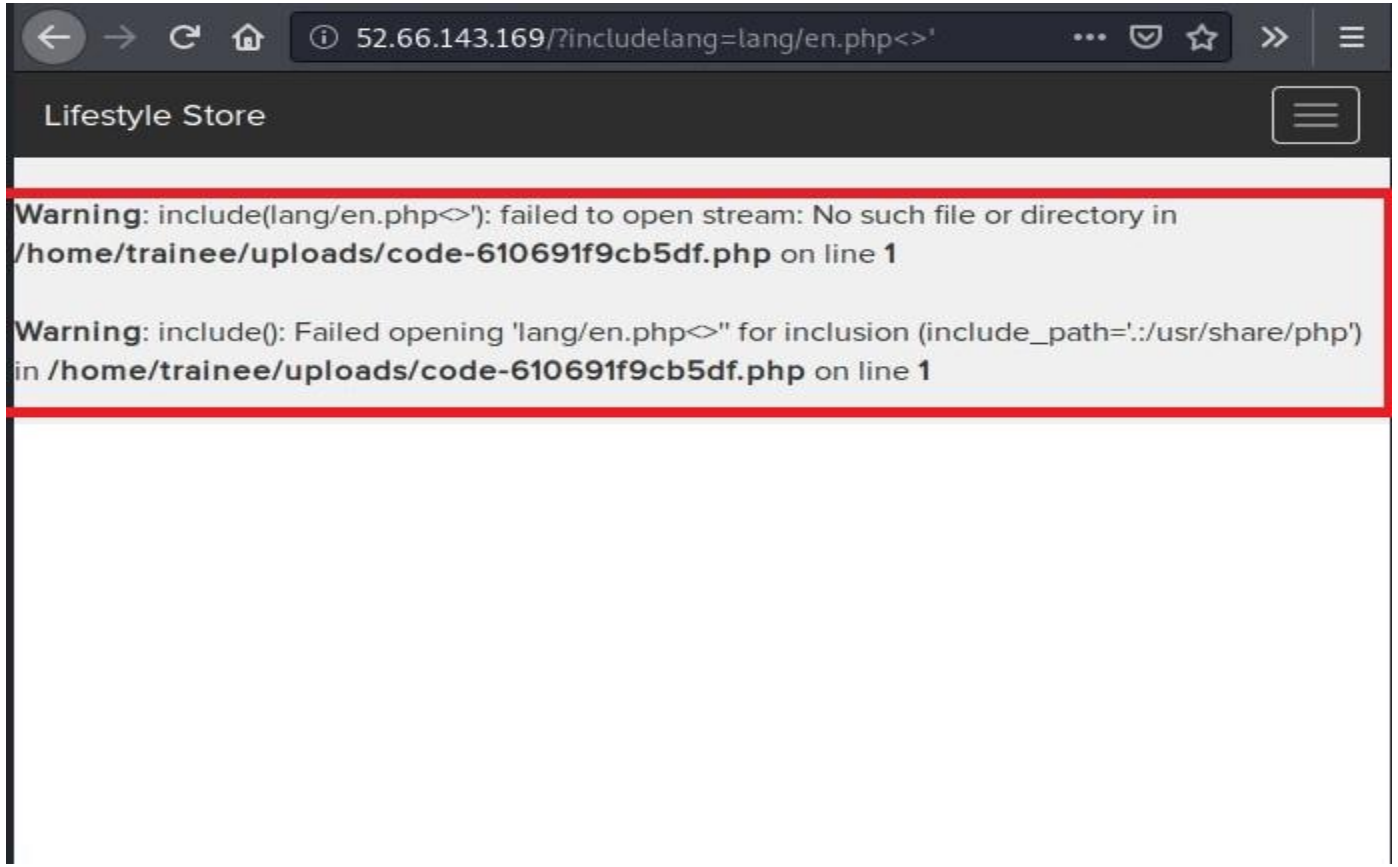
12. Descriptive Error Message

An error message is a message displayed to the user by an operating system or application when an unexpected condition happens.

Descriptive Error Message	**URL :** http://52.66.143.169/ the **Lang** module in the page is vulnerable to Descriptive Error Message. **Affected URL :** http://52.66.143.169/?includelang=lang/en.php **Method Used :** GET based **Payload Used :** <>’

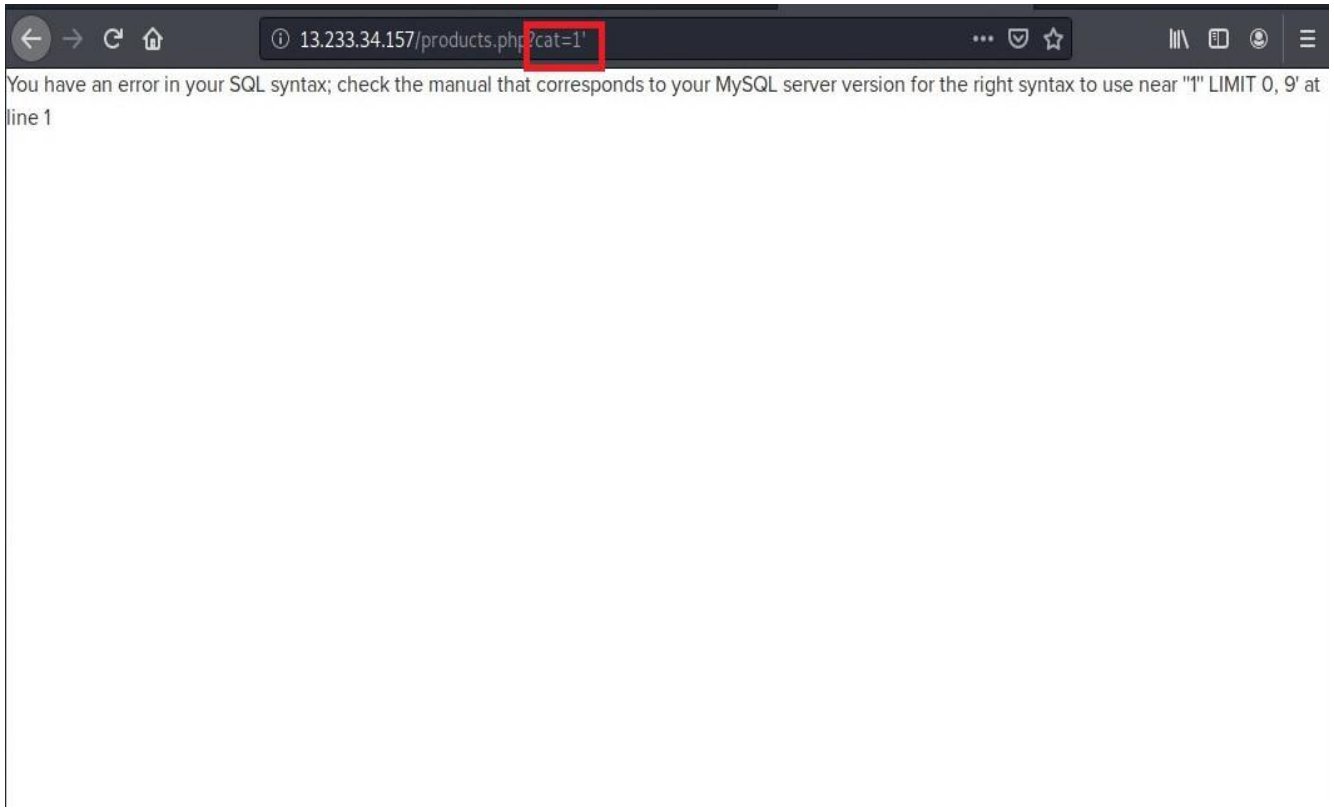
Observation

When the hacker adds some random special characters in the URL an Descriptive Error will be displayed.



Proof of Concept (PoC)

- As the error reveals some valuable information like the directory path the hacker can now access the directories and steal the data.
- These error also reveal if the website is vulnerable to SQL injection.



Business Impact - Moderate

These vulnerabilities does not directly cause an impact on the business, but it reveals some important information about the server and lets the hacker have a clear information of the servers stats.

Recommendation

- Do not display more than what needs to be displayed.
- Turn off Descriptive Error Messages.

Reference

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

13. Default File Misconfiguration

Default File Misconfiguration

URL : http://52.66.143.169/ is vulnerable to Default File Misconfiguration

Affected URL : http://52.66.143.169/server-status/
http://52.66.143.169/robots.txt/
http://52.66.143.169/phpinfo.php/
http://52.66.143.169/userlist.txt/
http://52.66.143.169/server-status/

Observation

By adding **“robots.txt”** in the URL we get the info of the restricted file location.

Observation

By adding **“server-status”** in the URL we get the whole server information.

52.66.143.169/server-status/

Apache Server Status for localhost (via 127.0.0.1)

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

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

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

```
_____________________________________________W_..............
..............................................................
......................
```

Observation

By adding **“phpinfo.php”** we get the PHP related information.

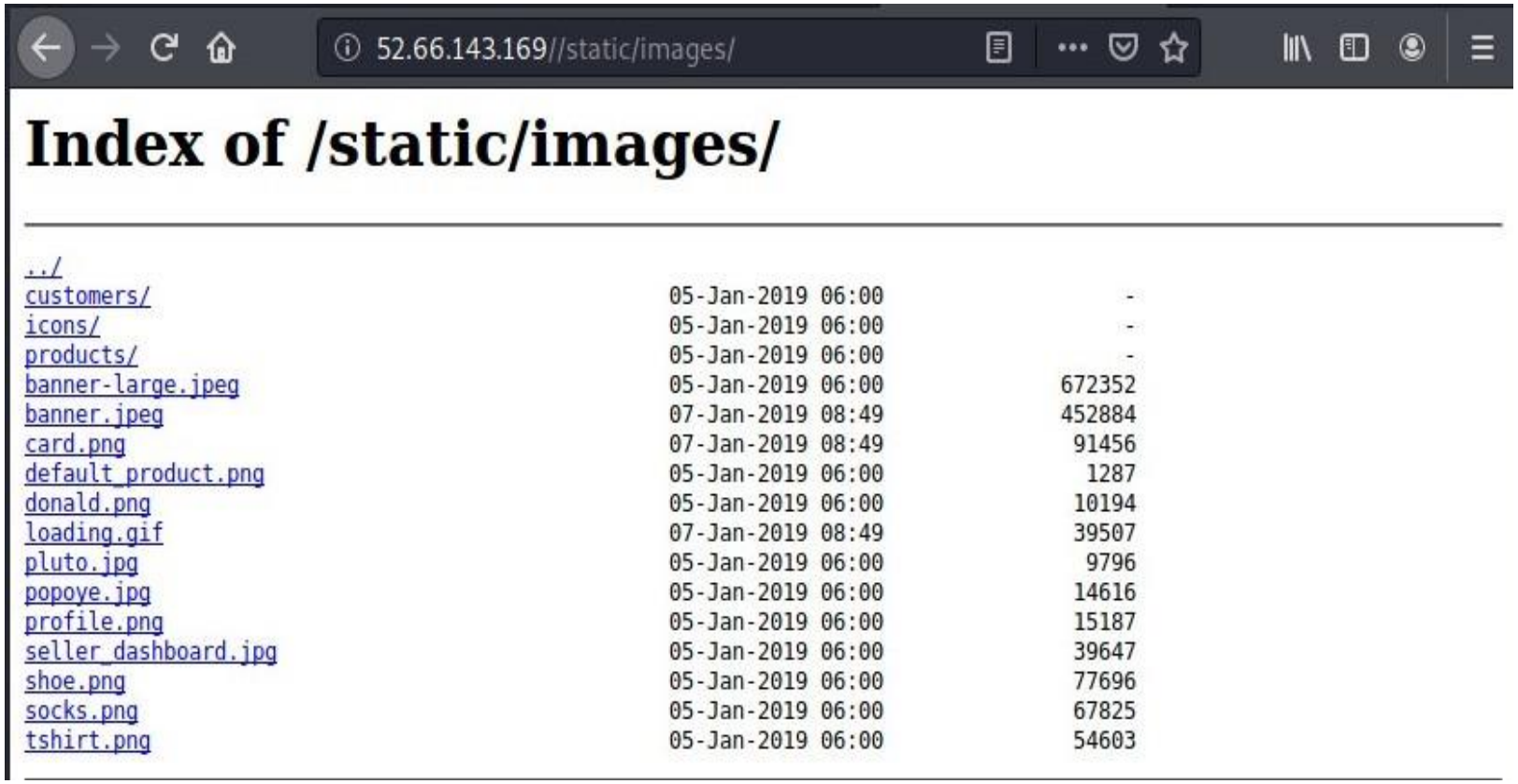
52.66.143.169/phpinfo.php

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

System	Linux ip-172-26-13-178 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/5.6/fpm/conf.d/20-iconv.ini, /etc/php/5.6/fpm/conf.d/20-json.ini, /etc/php/5.6/fpm/conf.d/20-mbstring.ini, /etc/php/5.6/fpm/conf.d/20-mysql.ini, /etc/php/5.6/fpm/conf.d/20-mysqli.ini, /etc/php/5.6/fpm/conf.d/20-pdo_mysql.ini, /etc/php/5.6/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/5.6/fpm/conf.d/20-phar.ini, /etc/php/5.6/fpm/conf.d/20-posix.ini, /etc/php/5.6/fpm/conf.d/20-readline.ini, /etc/php/5.6/fpm/conf.d/20-shmop.ini, /etc/php/5.6/fpm/conf.d/20-simplexml.ini, /etc/php/5.6/fpm/conf.d/20-sockets.ini, /etc/php/5.6/fpm/conf.d/20-sqlite3.ini, /etc/php/5.6/fpm/conf.d/20-sysvmsg.ini, /etc/php/5.6/fpm/conf.d/20-sysvsem.ini, /etc/php/5.6/fpm/conf.d/20-sysvshm.ini, /etc/php/5.6/fpm/conf.d/20-tokenizer.ini, /etc/php/5.6/fpm/conf.d/20-wddx.ini, /etc/php/5.6/fpm/conf.d/20-xmlreader.ini, /etc/php/5.6/fpm/conf.d/20-xmlwriter.ini, /etc/php/5.6/fpm/conf.d/20-xsl.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS

Proof of Concept (PoC)

Because of this vulnerability the hacker can gain access to the file which are restricted to the other users apart from the Admin.



Business Impact – Moderate

- This vulnerability does not have a direct impact to users or the server but it can help the attacker with information about the server and the users.

Recommendations

- Disable access to all the default files and folders including server-status and server-info.

References

https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration

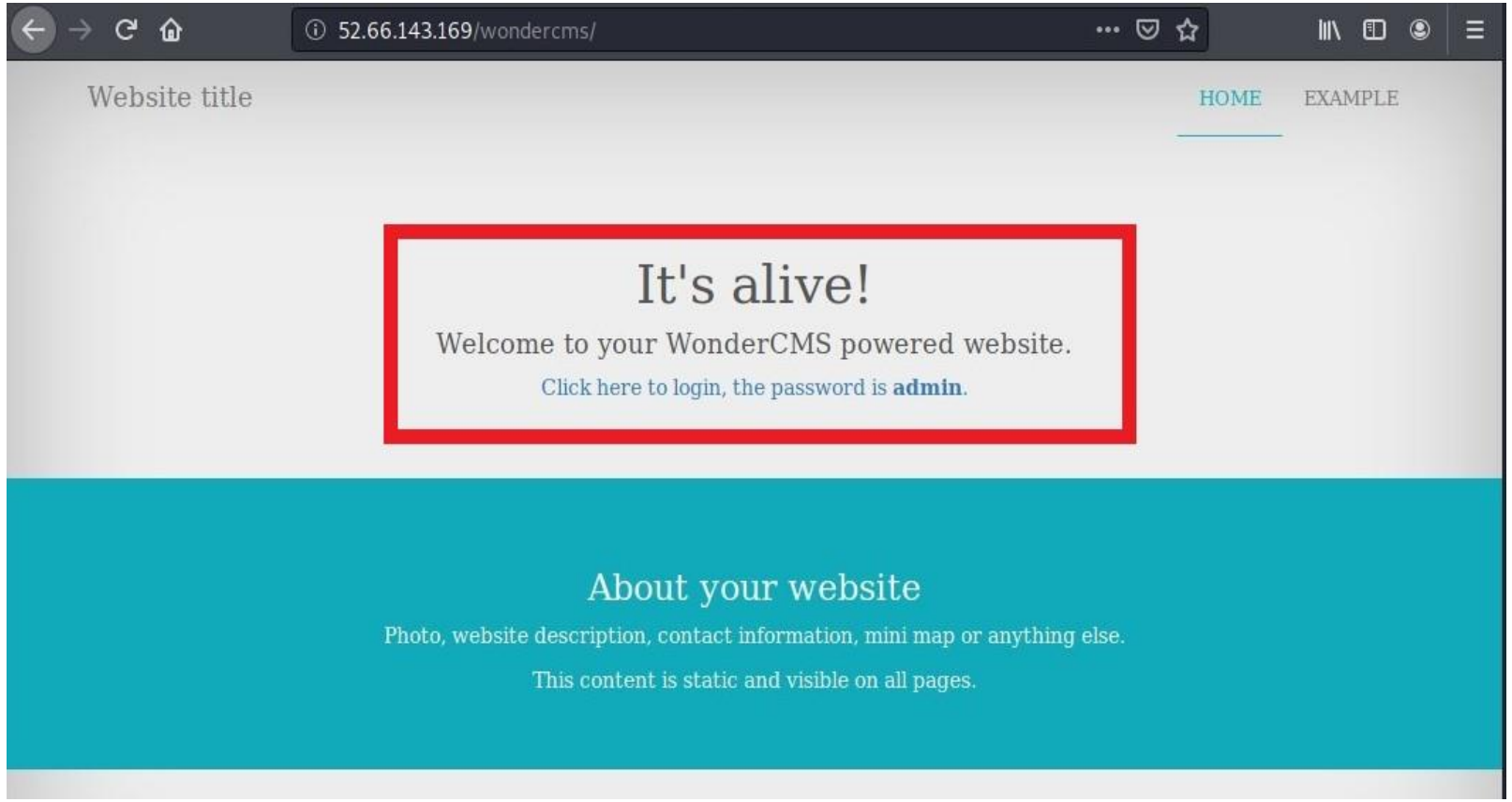
14. Default / Weak Passwords

The default passwords are very much vulnerable and can be guessed easily.

Deafult Password	**URL :** http://52.66.143.169/ in this URL the Blog module has default password **Method Used :** GET based **Affected URL :** http://52.66.143.169/wondercms/

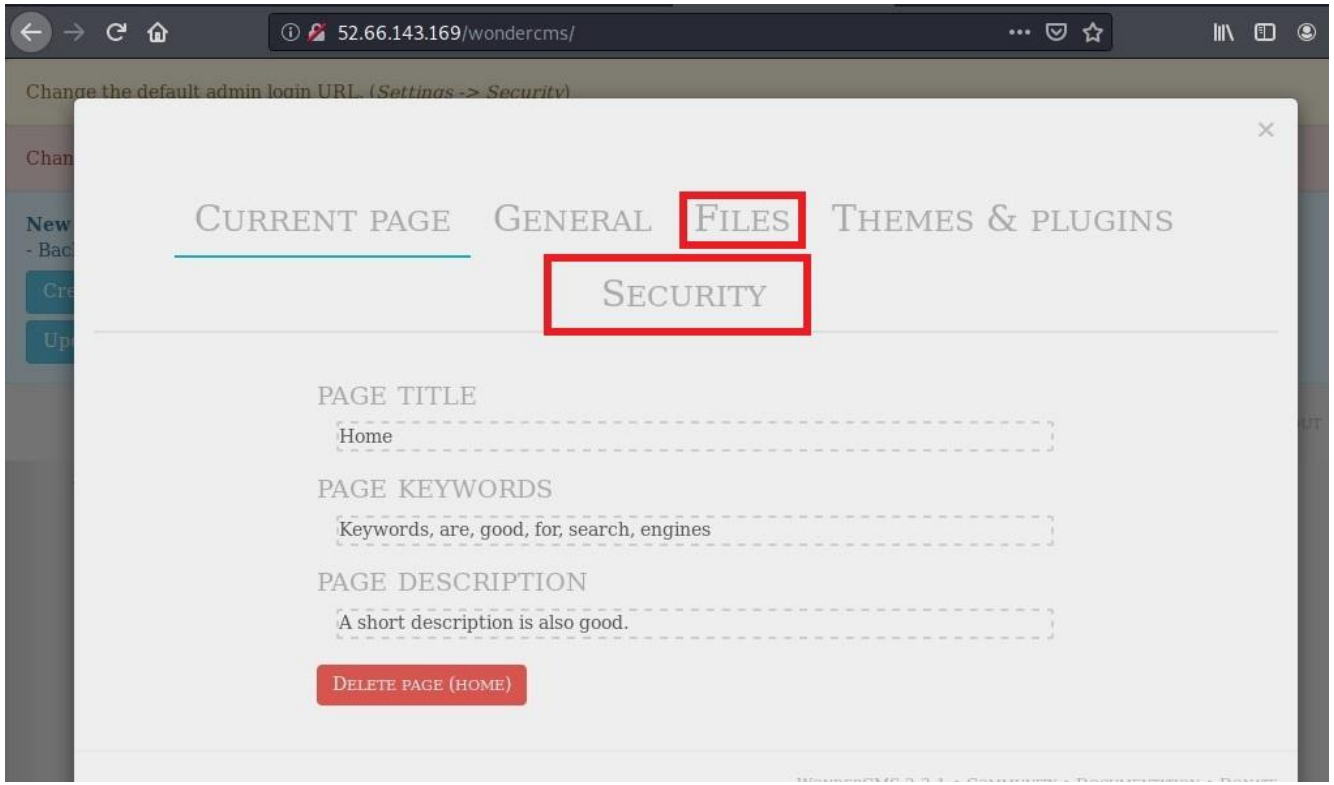
Observation

The hacker can easily guess the password and log in as an admin and take over the website .



Proof of Concept (PoC)

After accessing the admin page the hacker can easily change the security settings and also upload malicious scripts and take over the site.



Business Impact - Severe

Default and common passwords makes it easy for attackers to take control of the admin and make illegal use of them and can harm the website.

Recommendation

- There length of the password must be of minimum 8 characters
- There should be password strength check at every creation of an account.

References

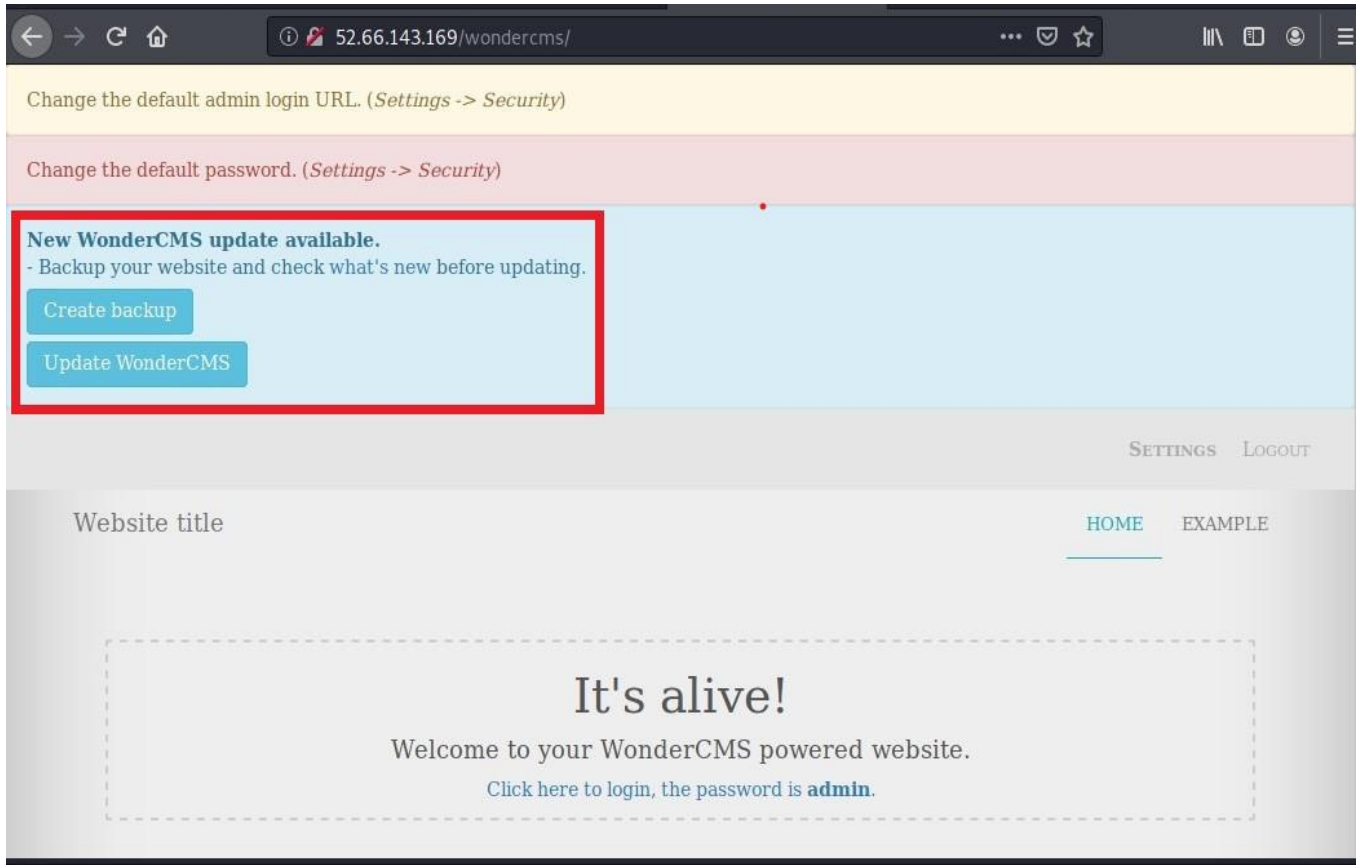
- https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/02-Testing_for_Default_Credentials
- https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

15. Components with known Vulnerabilities

Components with known Vulnerabilities	**URL :** http://52.66.143.169/wondercms/ is not an updated version

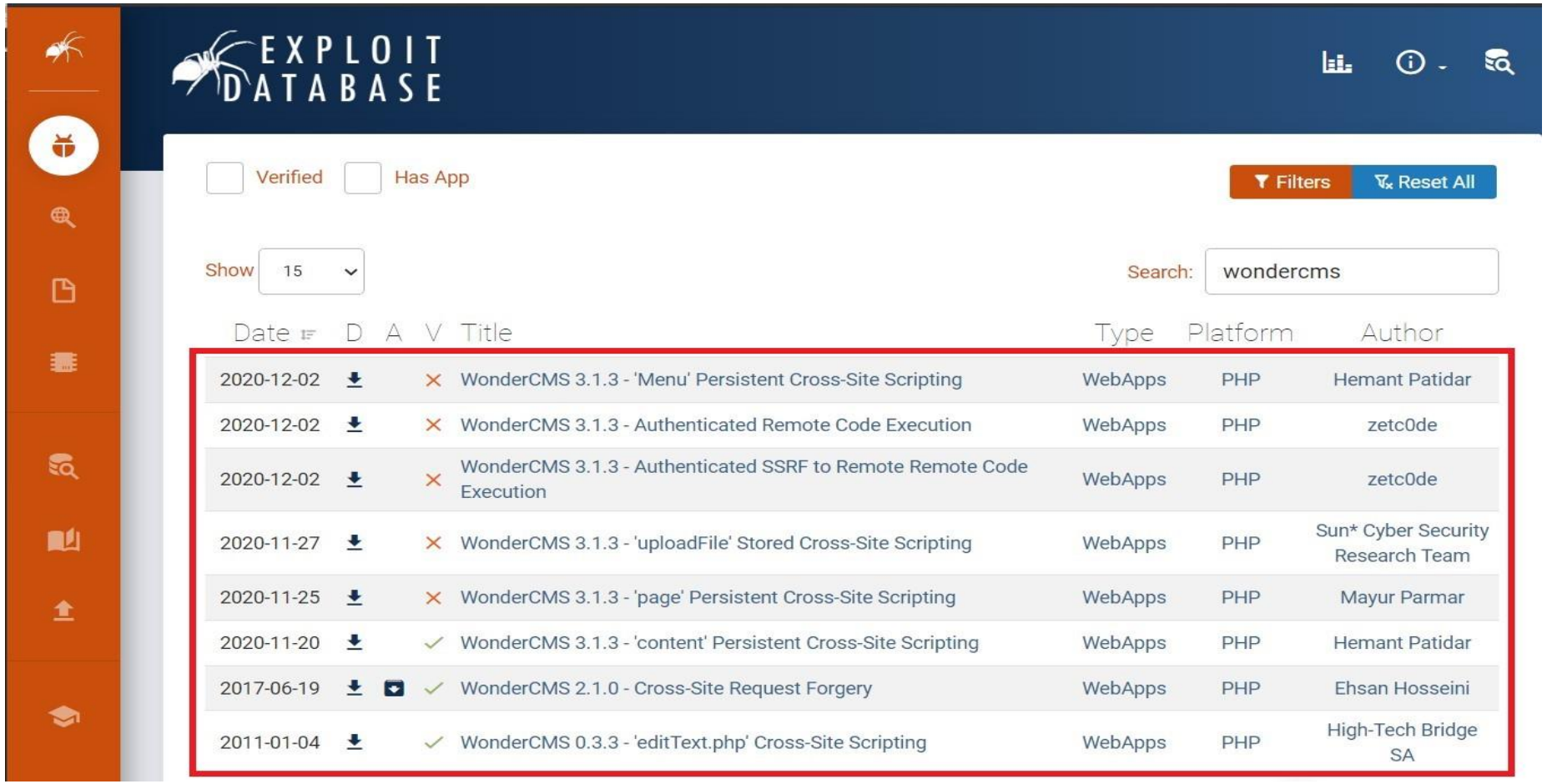
Observation

As the CMS is not up-to-date the hacker can identify the version and the exploits in it easily.



Proof of Concept (PoC)

There are many exploits on the outdated version of WonderCMS through which the hacker can easily get into the website and harm the data.



Date	D	A	V	Title	Type	Platform	Author
2020-12-02				WonderCMS 3.1.3 - 'Menu' Persistent Cross-Site Scripting	WebApps	PHP	Hemant Patidar
2020-12-02				WonderCMS 3.1.3 - Authenticated Remote Code Execution	WebApps	PHP	zetc0de
2020-12-02				WonderCMS 3.1.3 - Authenticated SSRF to Remote Remote Code Execution	WebApps	PHP	zetc0de
2020-11-27				WonderCMS 3.1.3 - 'uploadFile' Stored Cross-Site Scripting	WebApps	PHP	Sun* Cyber Security Research Team
2020-11-25				WonderCMS 3.1.3 - 'page' Persistent Cross-Site Scripting	WebApps	PHP	Mayur Parmar
2020-11-20				WonderCMS 3.1.3 - 'content' Persistent Cross-Site Scripting	WebApps	PHP	Hemant Patidar
2017-06-19				WonderCMS 2.1.0 - Cross-Site Request Forgery	WebApps	PHP	Ehsan Hosseini
2011-01-04				WonderCMS 0.3.3 - 'editText.php' Cross-Site Scripting	WebApps	PHP	High-Tech Bridge SA

Business Impact – Severe

This does not create an direct impact on the business but due to the exploits the server may be hacked easily and the hacker can take control .

Recommendations

- Frequently checks for bugs , exploits and patch them.
- Check for updates regularly.

References

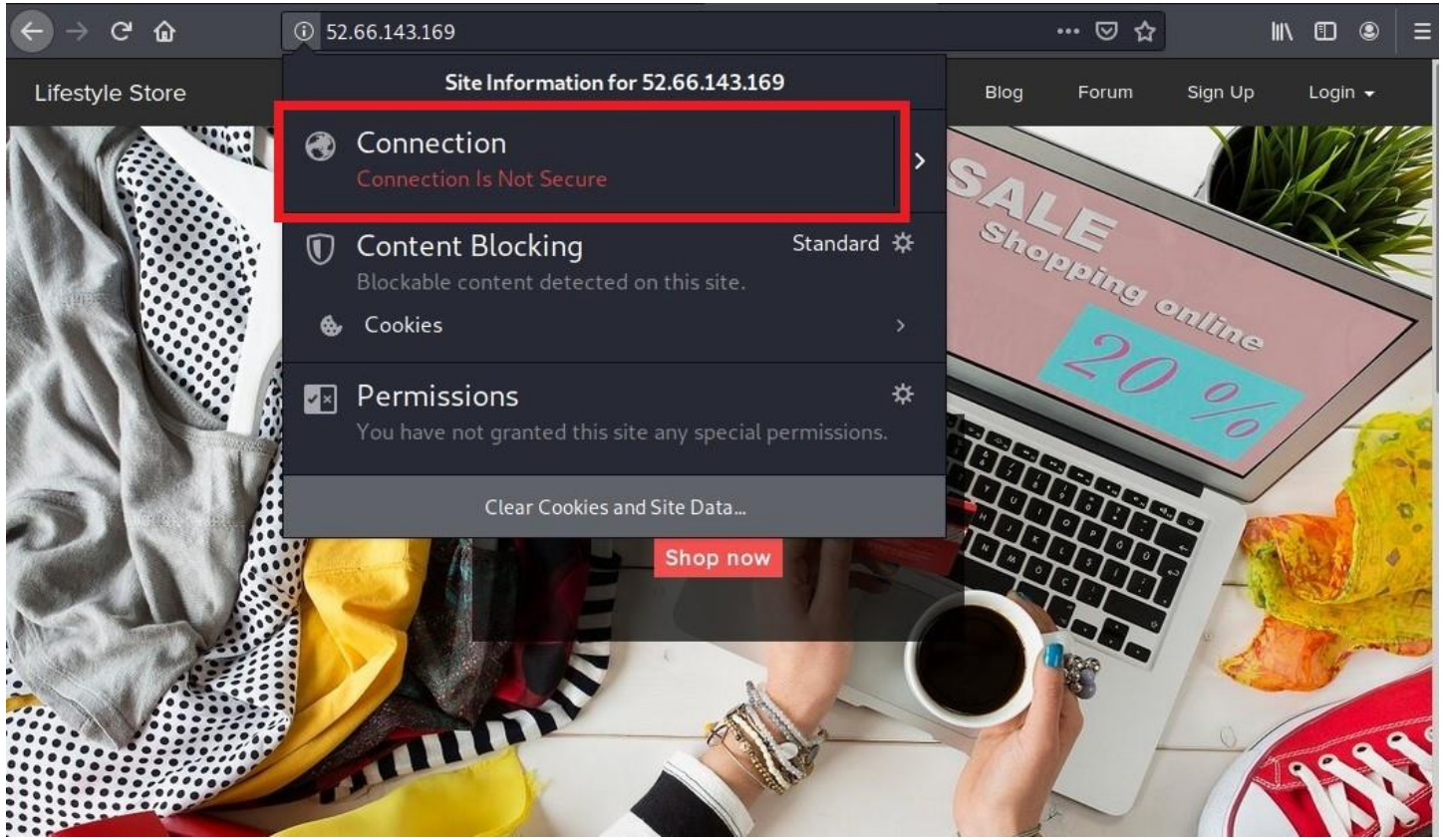
https://sec-consult.com/vulnerability-lab/advisory/multiple-vulnerabilities-in-wondercms/

16. Network Protocols Vulnerability

Network Protocol Vulnerability	Clearly the website is not secure as it uses **HTTP** over **HTTPS** and **GET** based method is adopted in most cases of this website **URL :** http://52.66.143.169/ all the modules included **Affected URL :** http://52.66.143.169/ (the whole website)

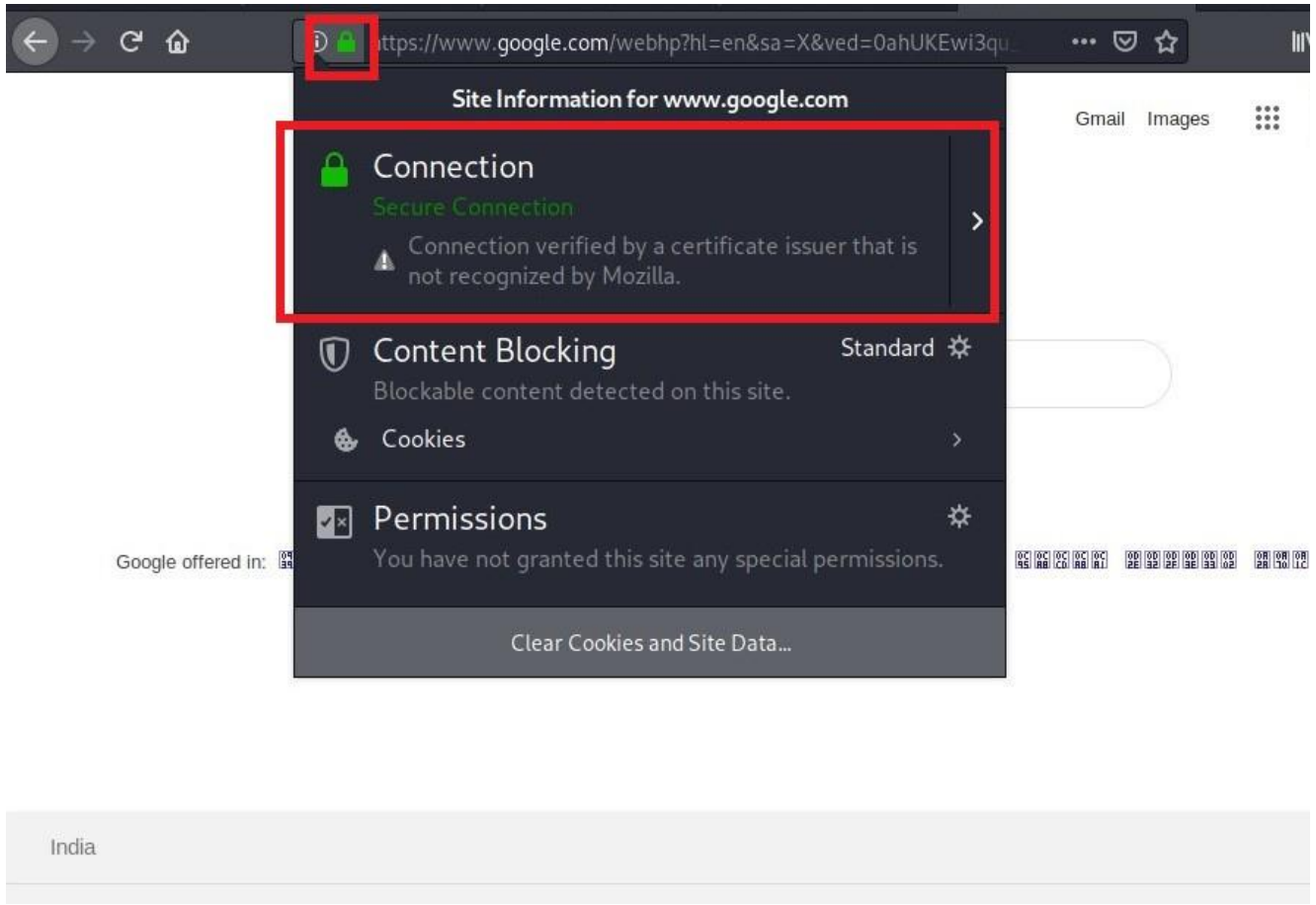
Observation

HTTPS is more secured than HTTP and HTTPS provides encryption of the data.



Proof of Concept (PoC)

Almost every website uses HTTPS these days as it is more secure and provides encryption which HTTP does not provide.



Business Impact - Severe

Although this does not affect the business directly but the website is at great risk when the hacker steals the data it will not be in cipher text but it will be in plain text form.

Recommendation

- Uses HTTPS instead of HTTP

Reference

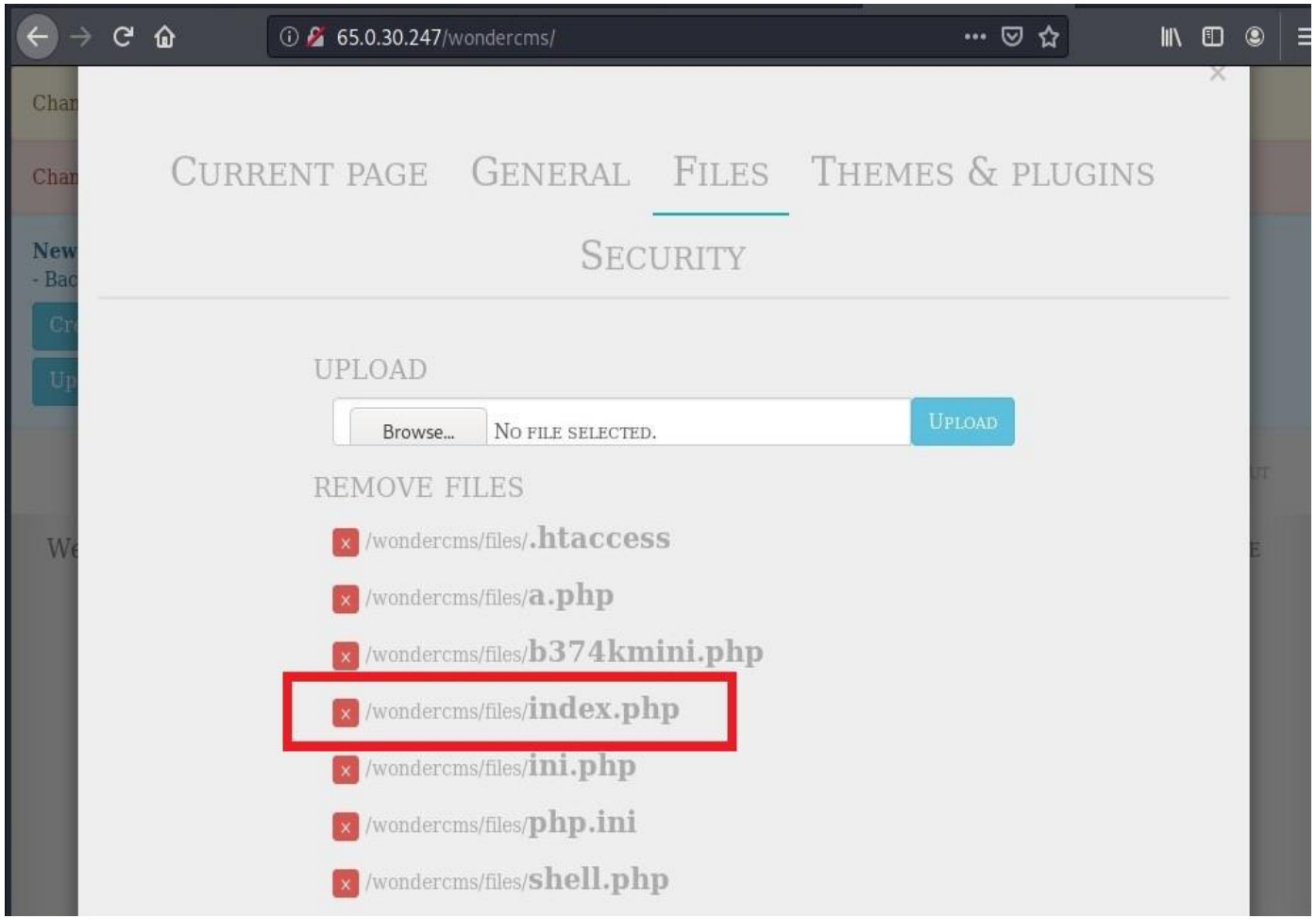
- https://www.cloudflare.com/en-in/learning/ssl/why-use-https/
- https://portswigger.net/web-security/request-smuggling/exploiting

17. Shell Uploading

Shell Uploading	**URL :** http://65.0.30.247/ the Blog module in this page is vulnerable to Shell upload and execution **Affected URL :** http://65.0.30.247/wondercms/ under the **security** module navigate to **Files** and upload the file **Method Used :** GET based **Parameter :** index.php (PHP Shell)

Observation

The hacker can upload the shell and to run it click on the file.

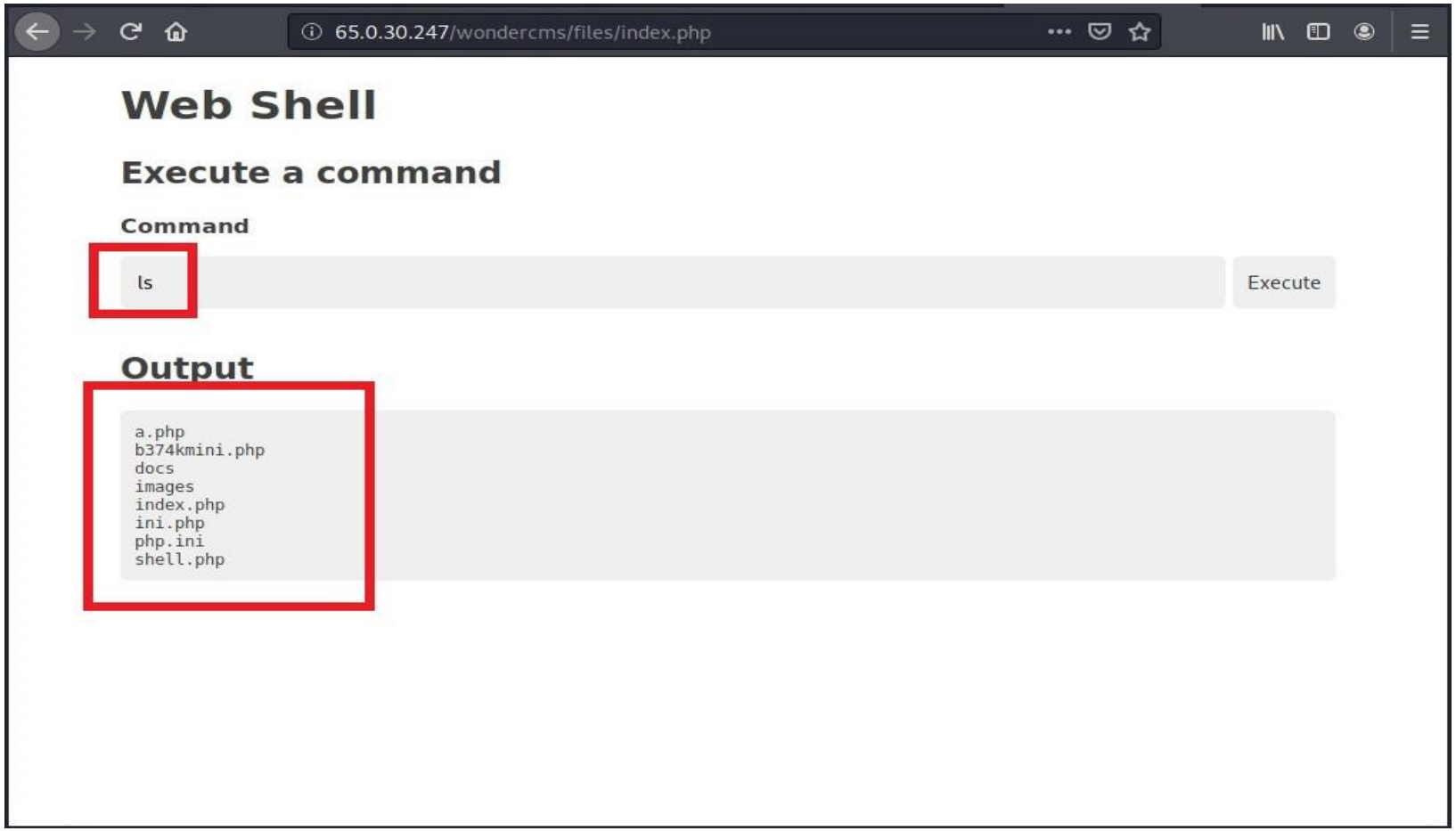


Proof of Concept (PoC)

After uploading the shell the hacker can run the shell and execute the commands and will be able to access the files and directories easily.

Proof of Concept (PoC)

The hacker can run Linux based commands.



Business Impact - Severe

- After successfully logging in to the admin's account the hacker can easily steal sensitive data and cause harm to the website.

- The hacker can run malicious shell scripts and steal the data.

- Other than injecting malicious code , the attacker can even get the details of the websites like its version and he can find the vulnerabilities to that version and easily exploit them and cause damage to the website.

Recommendations

- Only allow specific file extensions.
- Only allow authorized and authenticated users to use the feature.
- Check any file fetched from the Web for content. Make sure it is actually an image or whatever file type you expect.
- Serve fetched files from your application rather than directly via the web server.
- Store files in a non-public accessibly directory if you can.

References

- https://www.wordfence.com/learn/how-to-prevent-file-upload-vulnerabilities/

- https://www.acunetix.com/blog/articles/detection-prevention-introduction-web-shells-part-5/

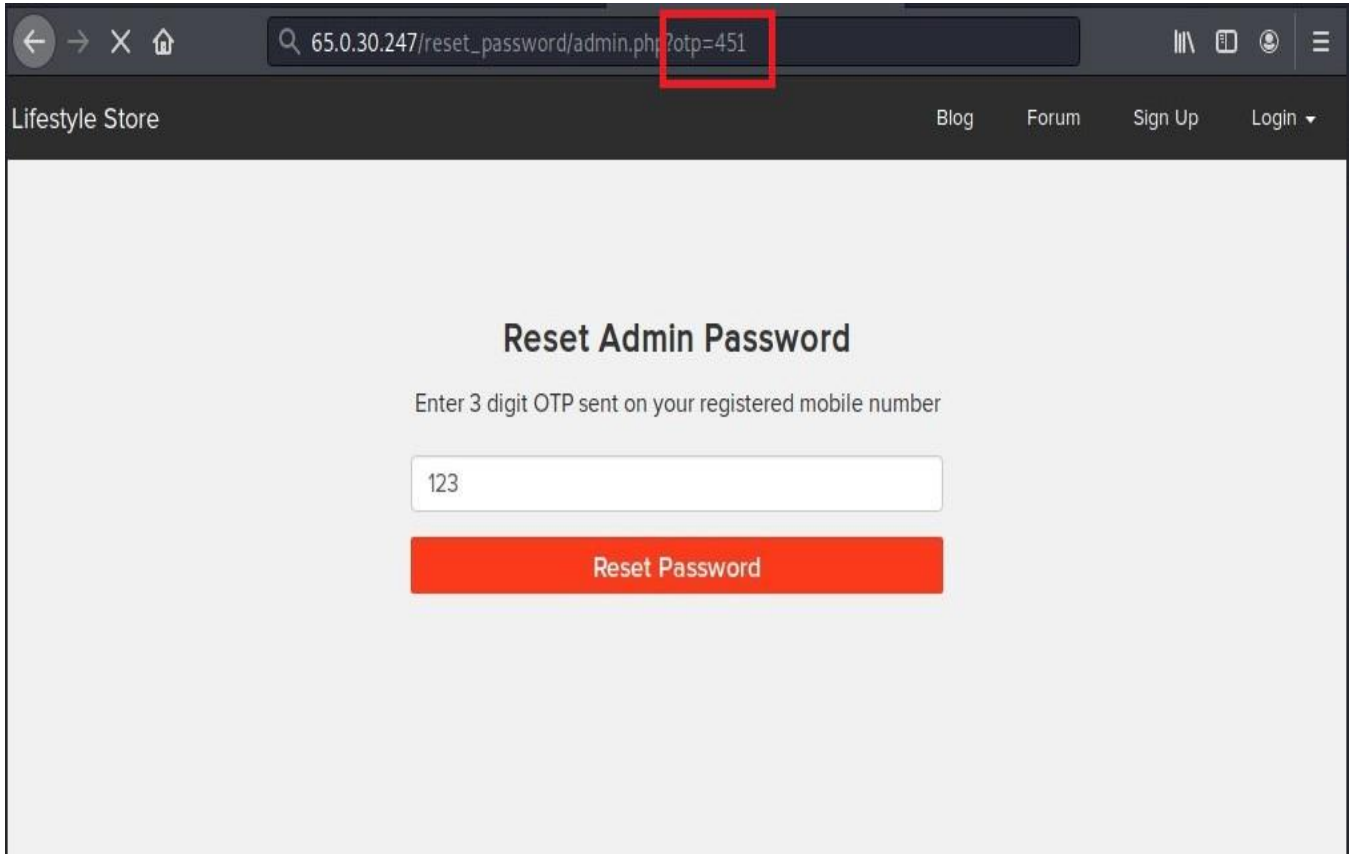
18. Rate Limiting Flaw

Through brute forcing the hacker can guess the OTP and access the accounts.

OTP Bypass	**URL :** http://65.0.30.247/ in this site the **Admin login** module under **Login** module is vulnerable to OTP bypass. **Affected URL :** http://65.0.30.247/reset_password/admin.php?otp=321 **Method Used :** GET based **Affected Parameter :** otp= **Payload Used :** 122

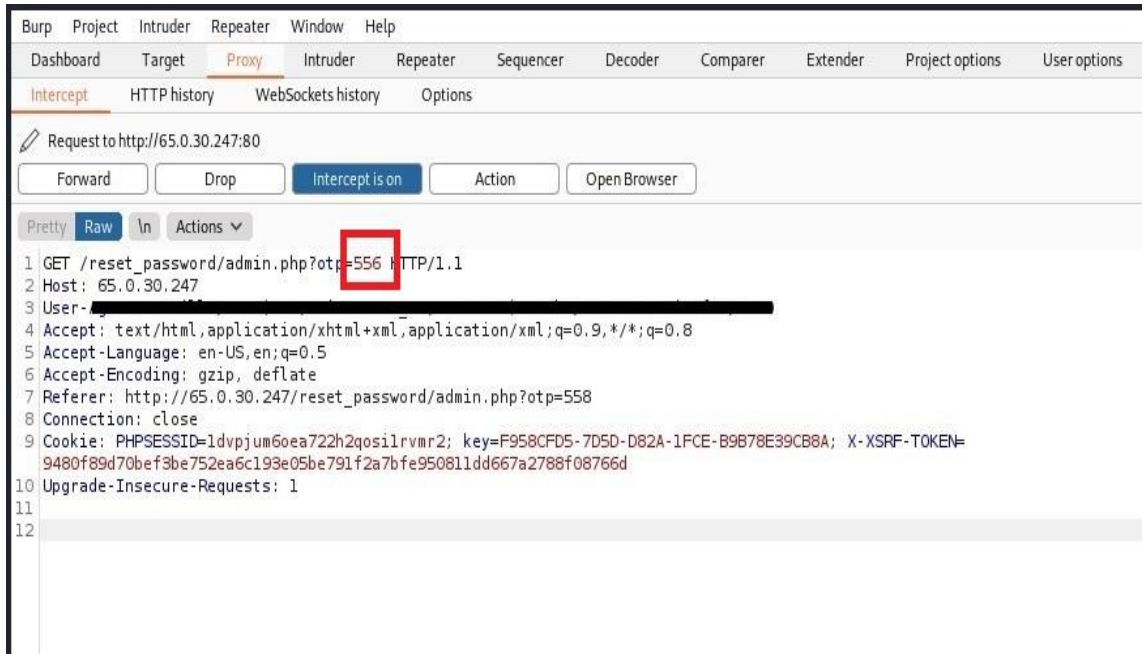
Observation

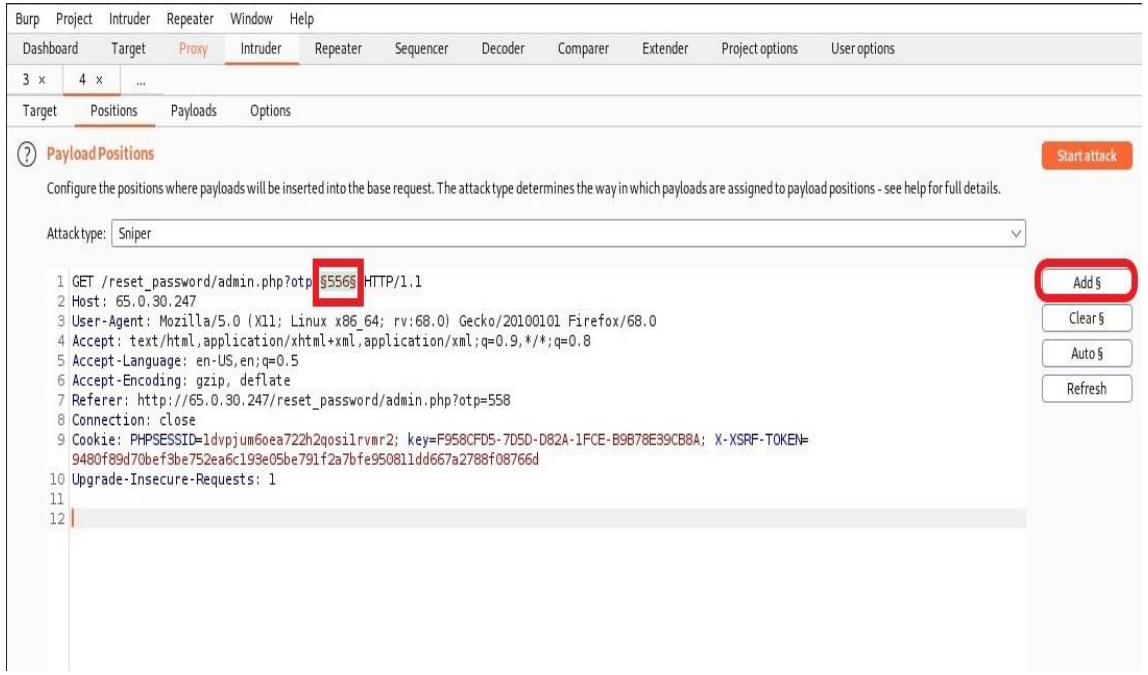
The hacker can intercept the packet and brute force the OTP through Burp Suite and guess the OTP and change the credentials of the admin login.



Observation

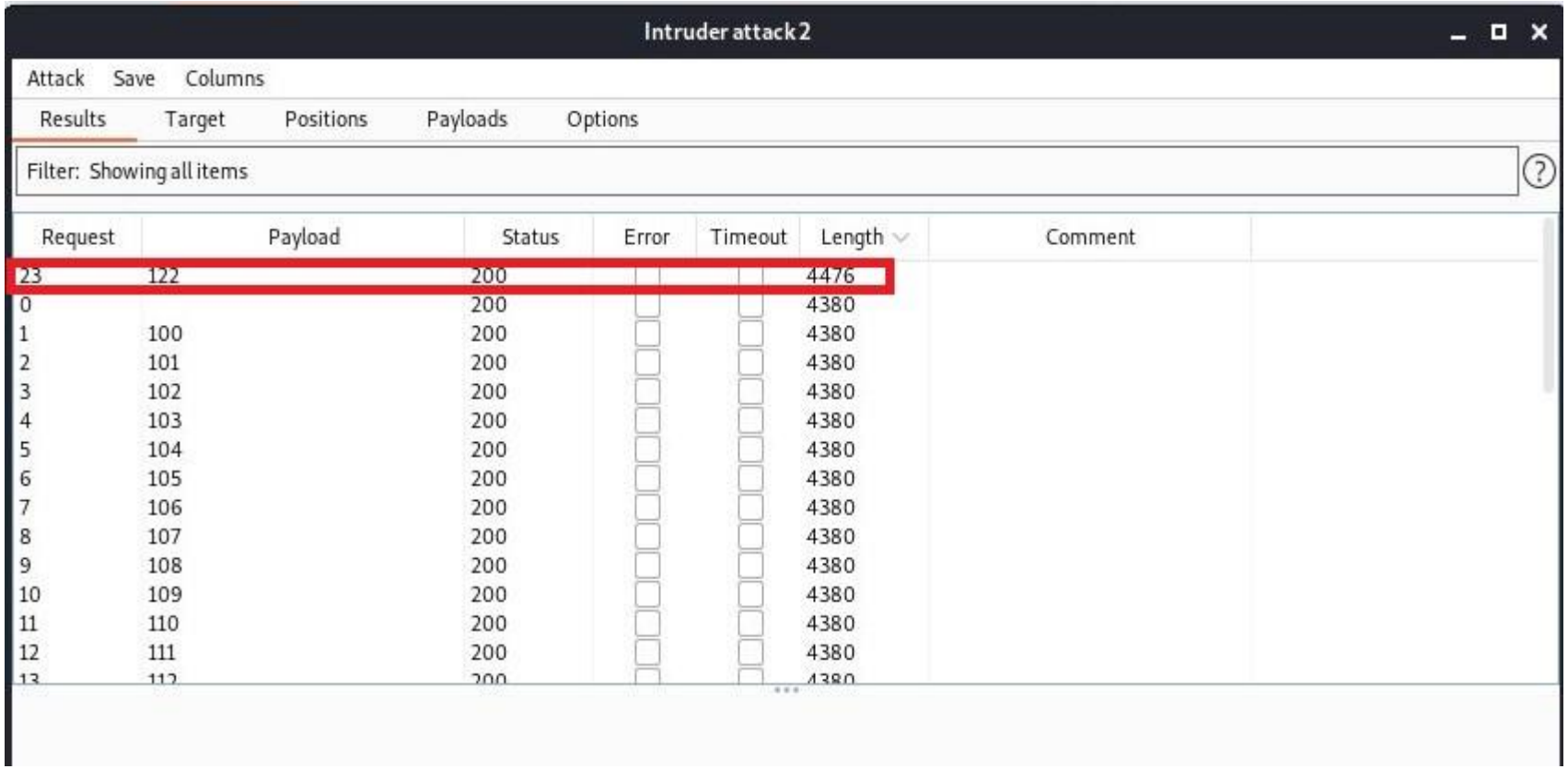
- After intercepting the packet the hacker can add the “otp=123“ parameter and selects the sniper attack mode and enters the range of the OTP.





Observation

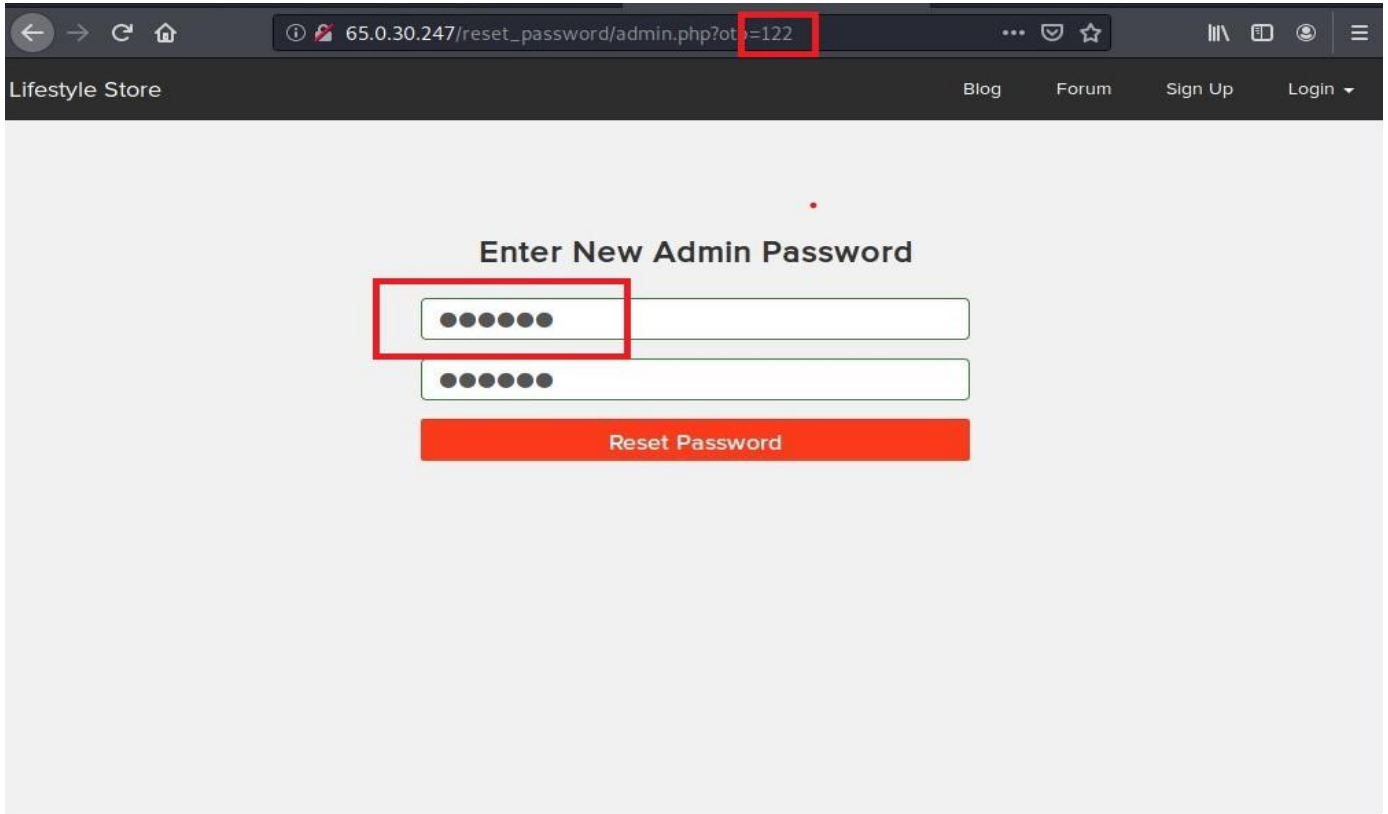
The attack will begin and the valid OTP will be identified after a few minutes and the hacker can enter the otp and change the password of the admin account and not allow the actual admin to log in.



Request	Payload	Status	Error	Timeout	Length	Comment
23	122	200			4476	
0		200			4380	
1	100	200			4380	
2	101	200			4380	
3	102	200			4380	
4	103	200			4380	
5	104	200			4380	
6	105	200			4380	
7	106	200			4380	
8	107	200			4380	
9	108	200			4380	
10	109	200			4380	
11	110	200			4380	
12	111	200			4380	
13	112	200			4380	

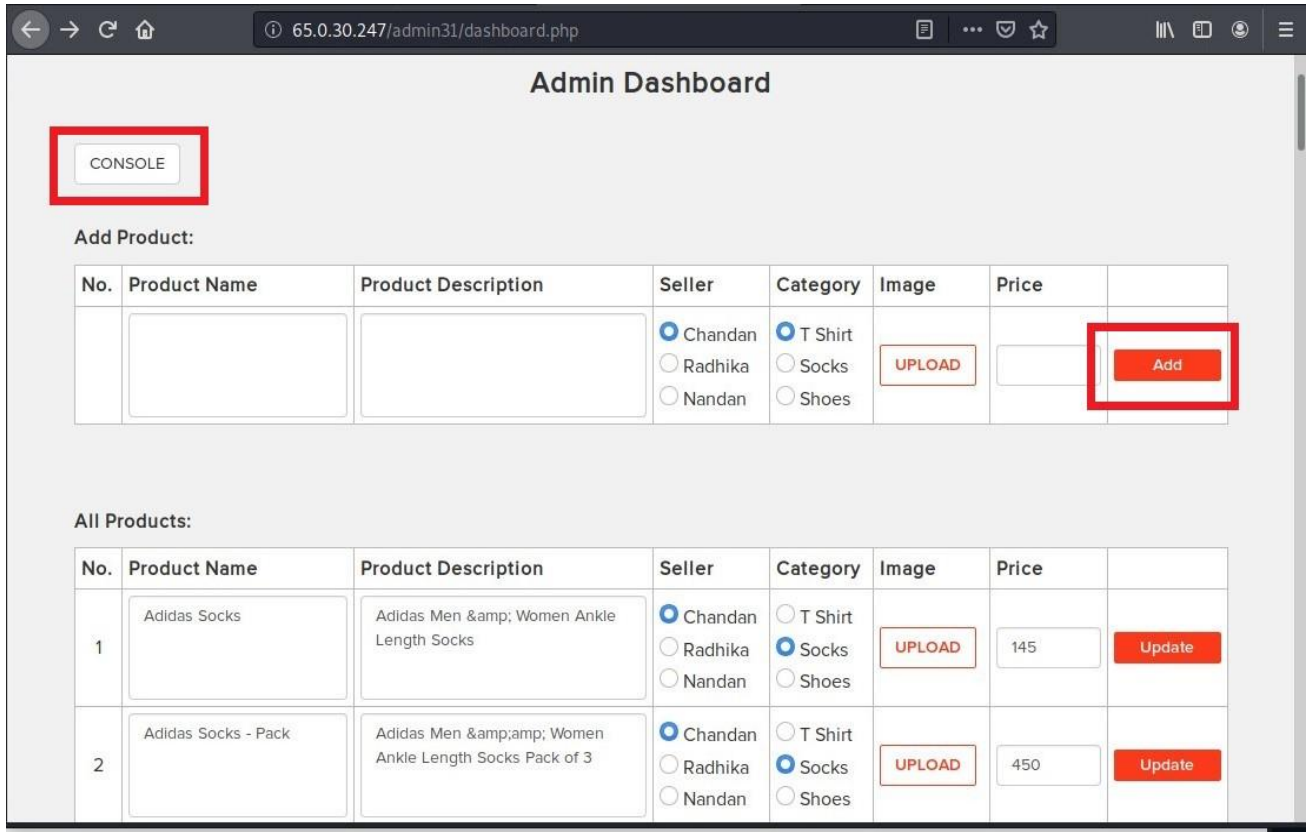
Observation

The OTP has been validated and the hacker can now change the password of the admin account.



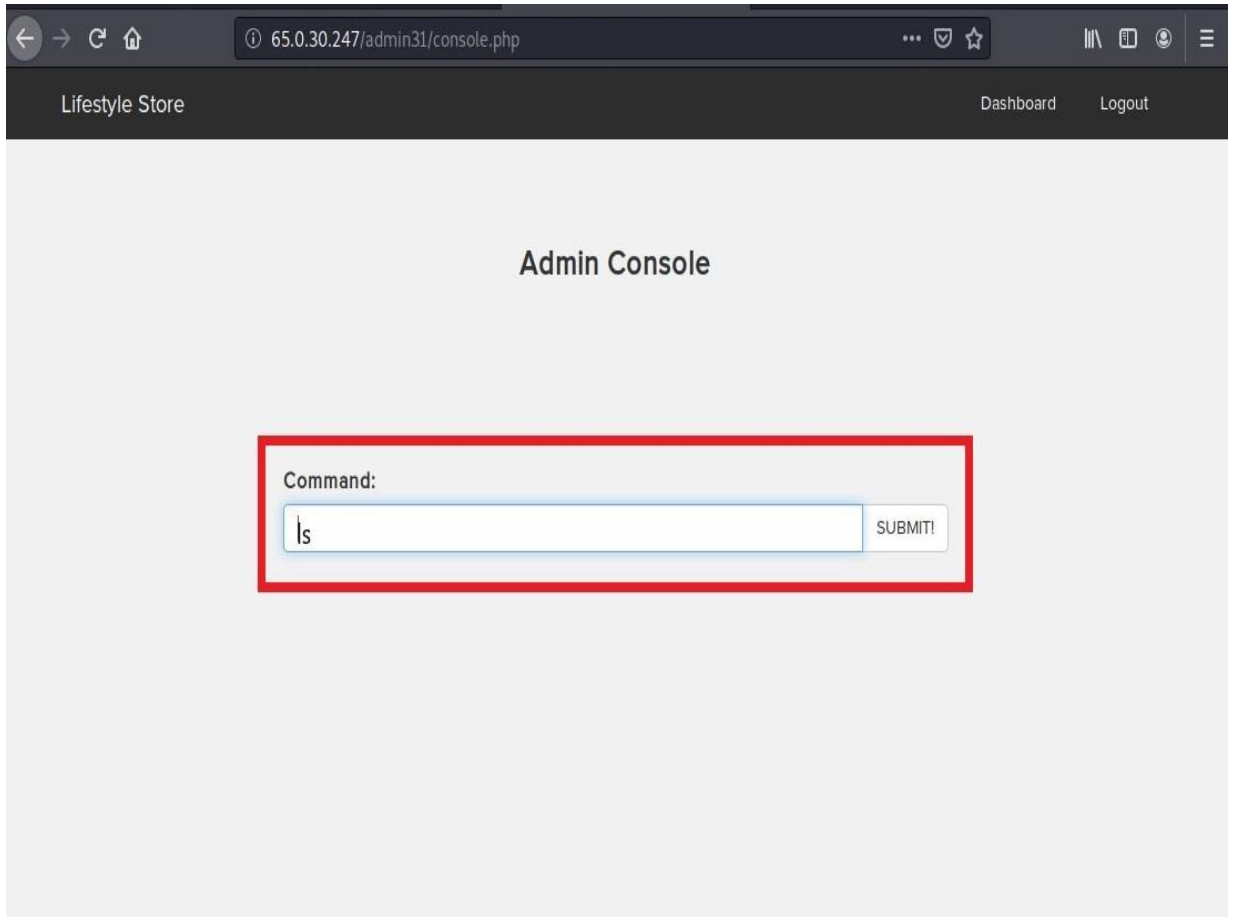
Proof of Concept (PoC)

Through this attack the hacker now has access to the admin account and can harm the website by adding fake data or delete the data and can cause harm to the database and can also steal the sellers details.



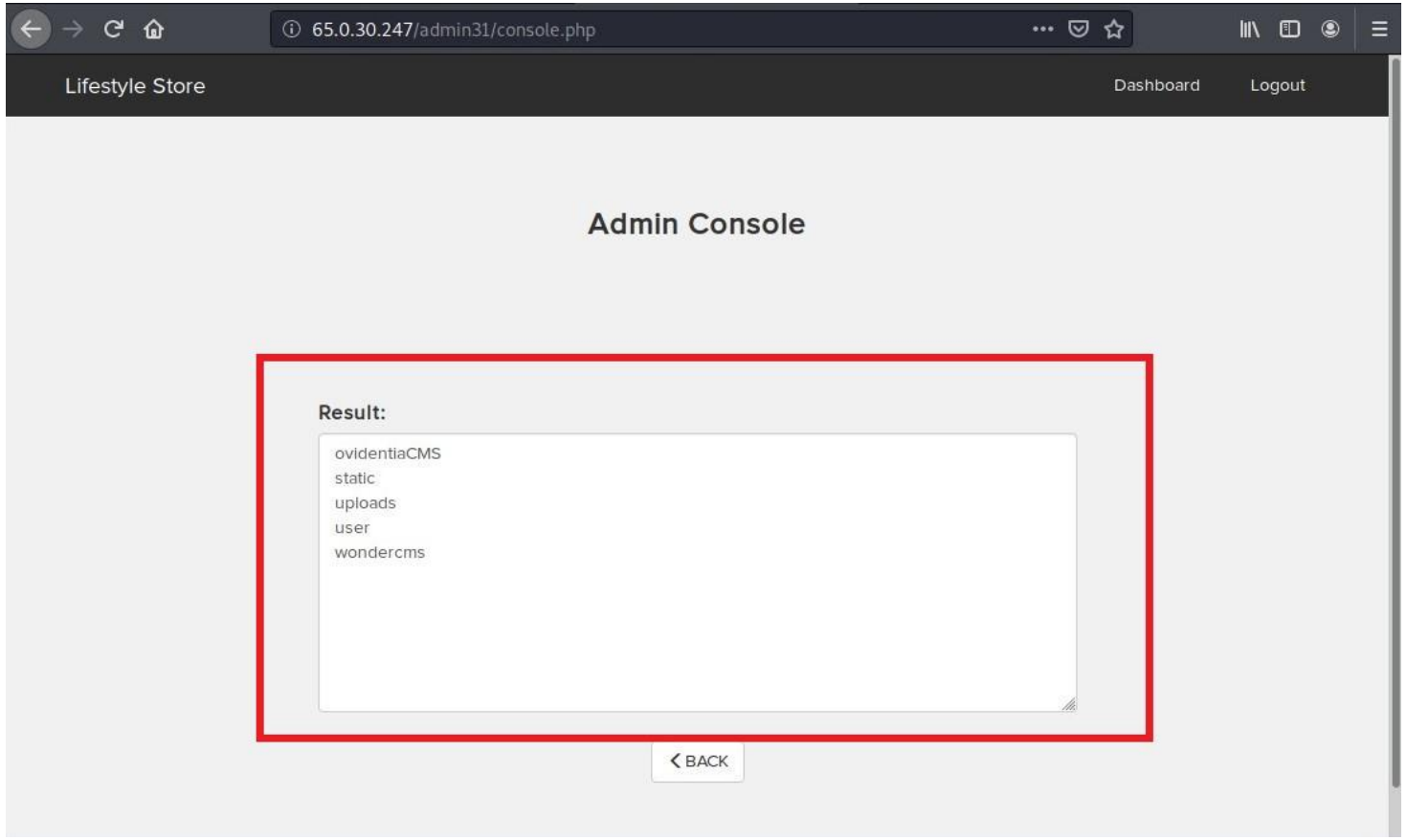
Proof of Concept (PoC)

The hacker now also has access to the console and can go through the directories or files using Linux commands.



Proof of Concept (PoC)

The hacker can now go through the important files related to the website and steal the data.



Business Impact - Critical

- The hacker can easily add or delete data of the sellers and cause financial harm to the company.

- The hacker can also steal personal data of the sellers , customers etc..

Recommendations

- The OTP length should be minimum of 8 characters.

- The OTP generation requests and validation should be checked and blocked if it crosses the limit.

References

- https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/04-Testing_for_Bypassing_Authentication_Schema
- https://phoenixnap.com/kb/prevent-brute-force-attacks
- https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks
- https://cloud.google.com/architecture/rate-limiting-strategies-techniques

THANK YOU

For further clarifications or assistance , please contact:
9494xxxxxx