INTERNSHALA
internships that matter

Hacking Environment Web Application

Detailed Developer Report

Security Status – Extremely Vulnerable

- Hacker can steal all records in databases (SQLi)
- Hacker can take control of complete server including View, Add, Edit, Delete files and folders (Shell Upload and and Weak Passwords)
- 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 (XSS)
- Hacker can extract mobile number of all customers using User-id (IDOR)

- Hacker can get access to seller details such as pan card number(PII)
- Hacker can Brute Force and bypass OTP protection in admin panel and coupon code is also vulnerable (Brute force attack)
- Hacker can change the password of victim(CSRF)
- Use off http instead of https.
- Directories are accessible
- Shows Server info
- Hacker can bypass client side filters
- Default errors

Vulnerability Statistics

CRITICAL	SEVERE	MODERATE	LOW
13	10	12	4

Vulnerabilities

NO	Severity	Vulnerability	Count
1	Critical	SQL Injection	3
2	Critical	Insecure File Uploads	1
3	Critical	CSRF	1
4	Critical	Access to admin panel	1
5	Critical	Brute Force Exploitation	2
6	Critical	Command Execution Vulnerability	2
7	Critical	IDOR	2
8	Critical	Stored xss	1
9	Severe	Reflected xss	2
10	Severe	Crypto Configuration Flaw	1
11	Severe	Common Passwords	2
12	Severe	Open Redirection	1
13	Severe	File Inclusion Vulnerability	2
14	Severe	Forced Browsing Flaws	1
15	Severe	Directory Listing	1

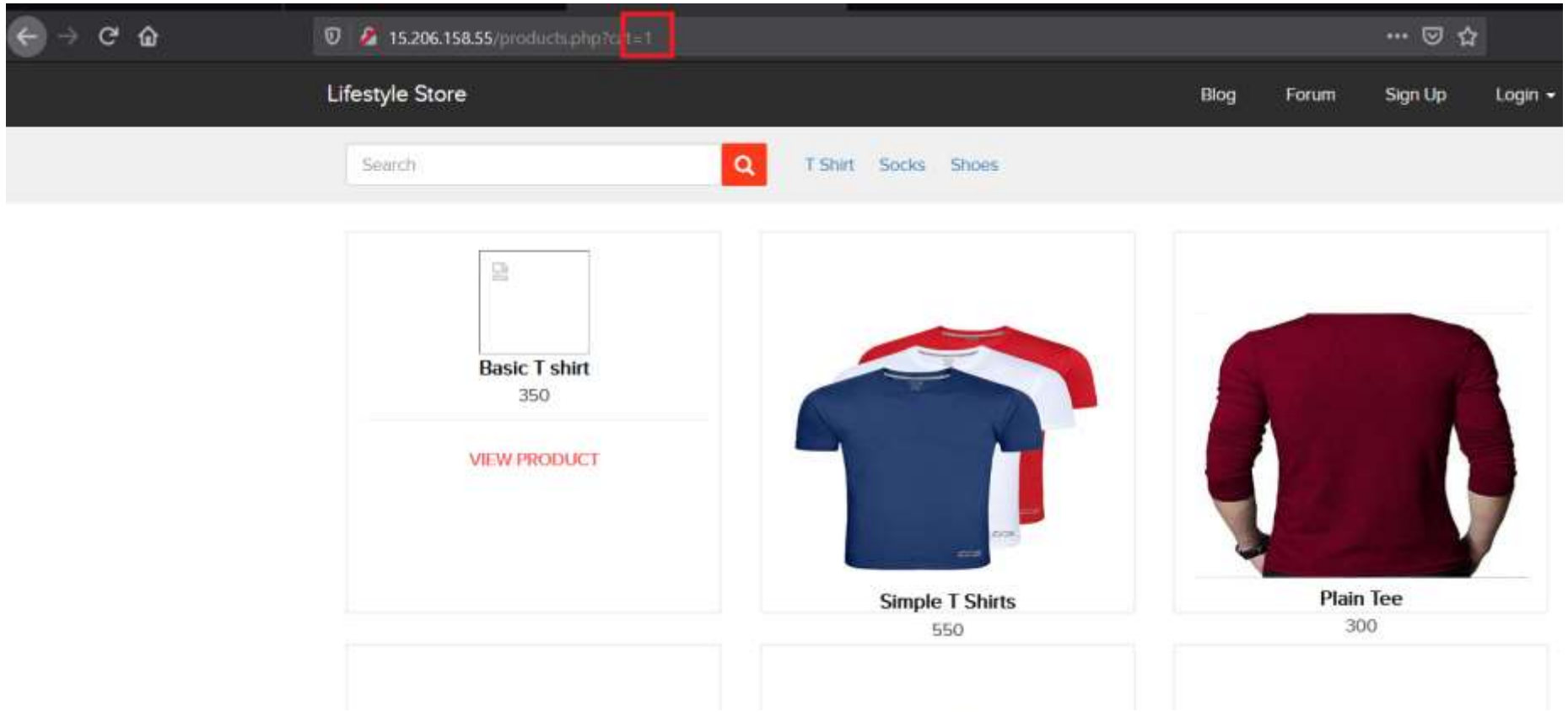
NO	Severity	Vulnerability	Count
16	Moderate	Default Files	6
17	Moderate	PII	3
18	Moderate	Components with known vulnerabilities	3
19	Low	Client-Server Filter Bypass	2
20	Low	Default Error	2

1. SQL Injection

SQL Injection (Critical)	Below mentioned URL in the **T-shirt/socks/shoes module** is vulnerable to SQL injection attack **Affected URL :** • http://15.206.158.55/products.php?cat=1 • http://15.206.158.55/products.php?cat=2 • http://15.206.158.55/products.php?cat=3 **Affected Parameters :** • cat (GET parameter) **Payload:** • cat=1’ • cat=2’ • cat=3’ • cat=1' union select database(),version(),database(),database(),version(),version(),version() --+ • cat=1' union select 1,user_name,3,password,5,6,7 from users --+

Observation

- From the website navigate to T shirts tab . Notice the GET parameter cat=1 in the URL:



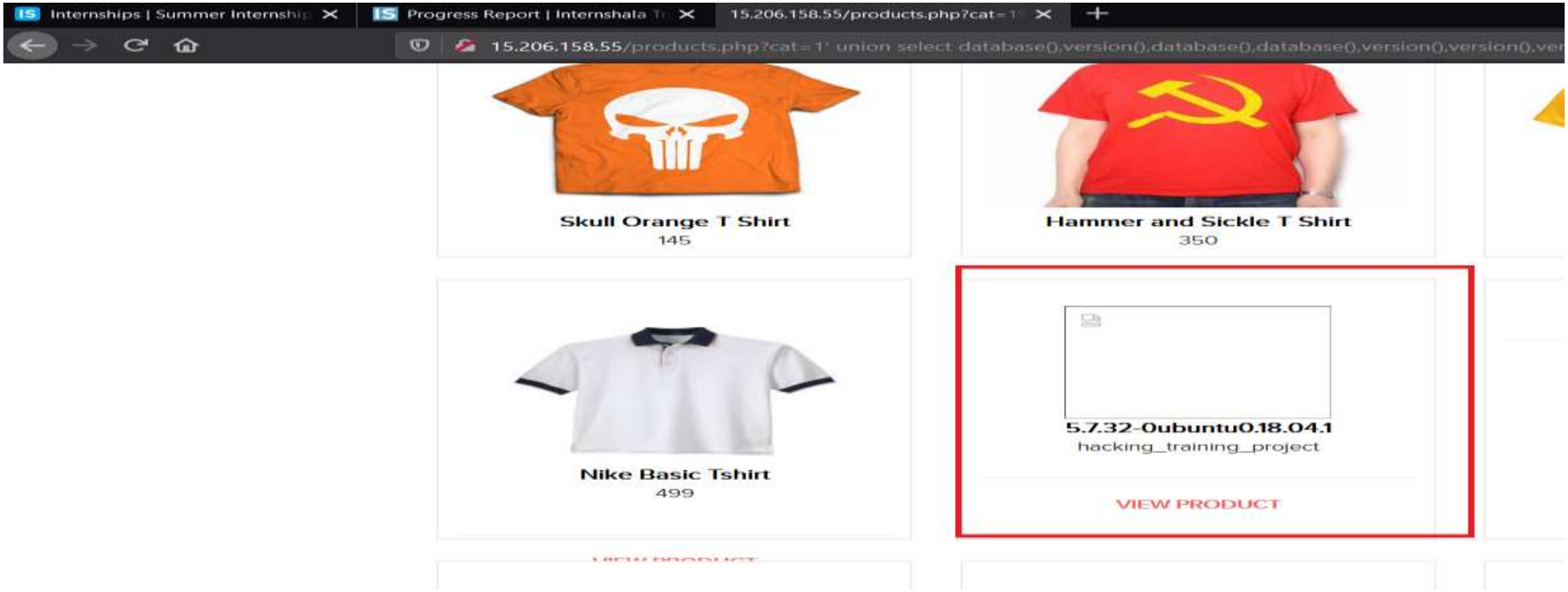
Observation

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

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 "1" LIMIT 0, 9' at line 1

Proof of Concept (PoC)

- Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the database name and MySQL version information
- http://15.206.158.55/products.php?cat= 1' union select database(),version(),database(),database(),version(),version(),version() --+



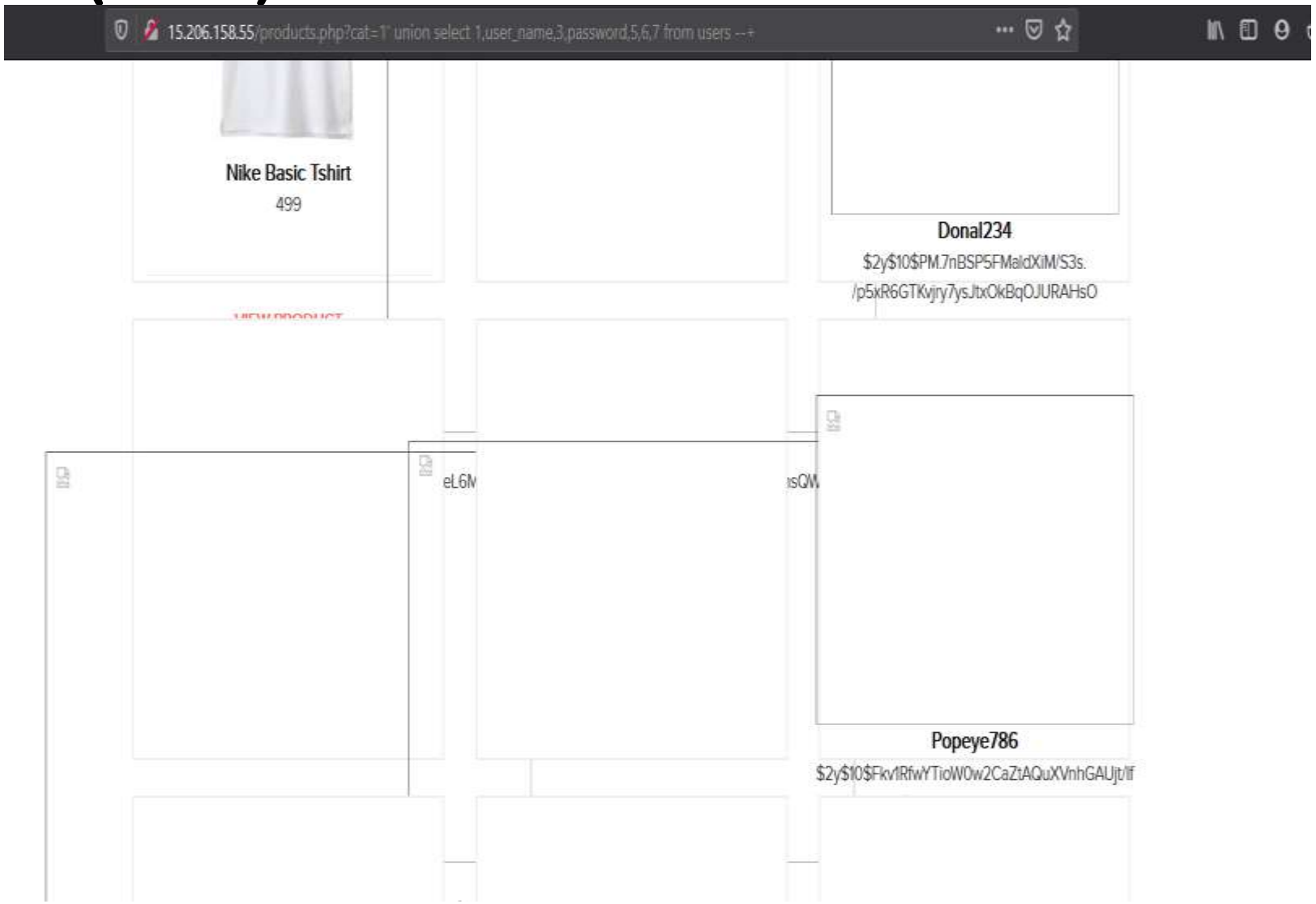
Proof of Concept (PoC)

No of databases: 2

- information_schema
- hacking_training_project

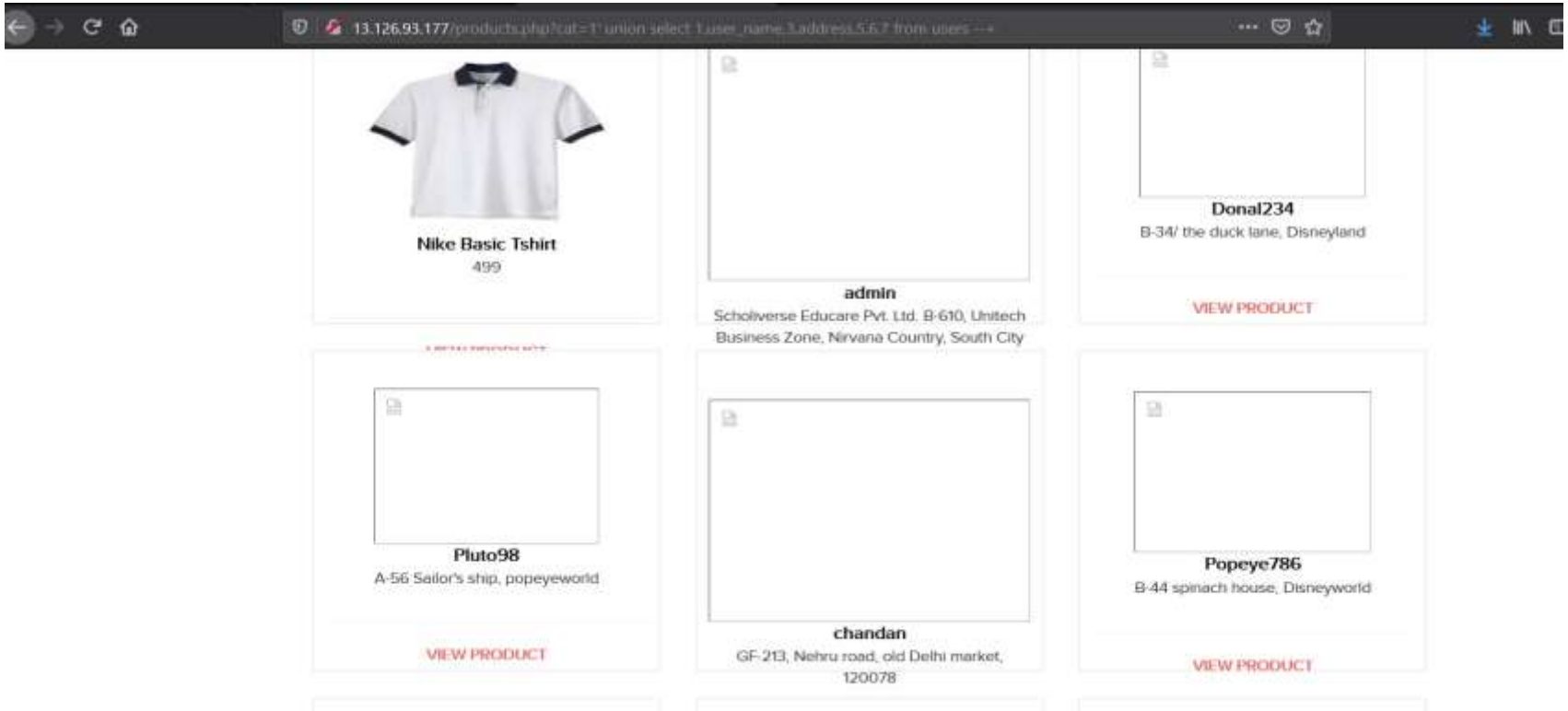
No of tables:

- Id
- Type
- Unique_key
- Email
- User_name
- Name
- Password
- Phone_number
- Address
- Created_at
- Last_updated_at



Business Impact - Extremely High

- Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.
- Below is the screenshot of some information extracted from users table which shows user credentials being leaked .Since the passwords are hashed ,the risk is comparatively low .
- Attacker can use this information to social engineer the customers and admin and gain complete customer access and admin level access to the website which could lead to complete compromise of the server and all other servers connected to it



Recommendation

Take the following precautions to avoid exploitation of SQL injections:

- Whitelist User Input: Whitelist all user input for expected data only. For example if you are expecting a flower name, limit it to alphabets only up to 20 characters in length. If you are expecting some ID, restrict it to numbers only
- Prepared Statements: Use SQL prepared statements available in all web development languages and frameworks to avoid attacker being able to modify SQL query
- Character encoding: If you are taking input that requires you to accept special characters, encode it, Example. Convert all ‘ to \’ , “ to \”, \ to \\. It is also suggested to follow a standard encoding for all special characters such has HTML encoding, URL encoding etc
- Do not run Database Service as admin/root user
- Disable/remove default accounts, passwords and databases
- Assign each Database user only the required permissions and not all permissions

References

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

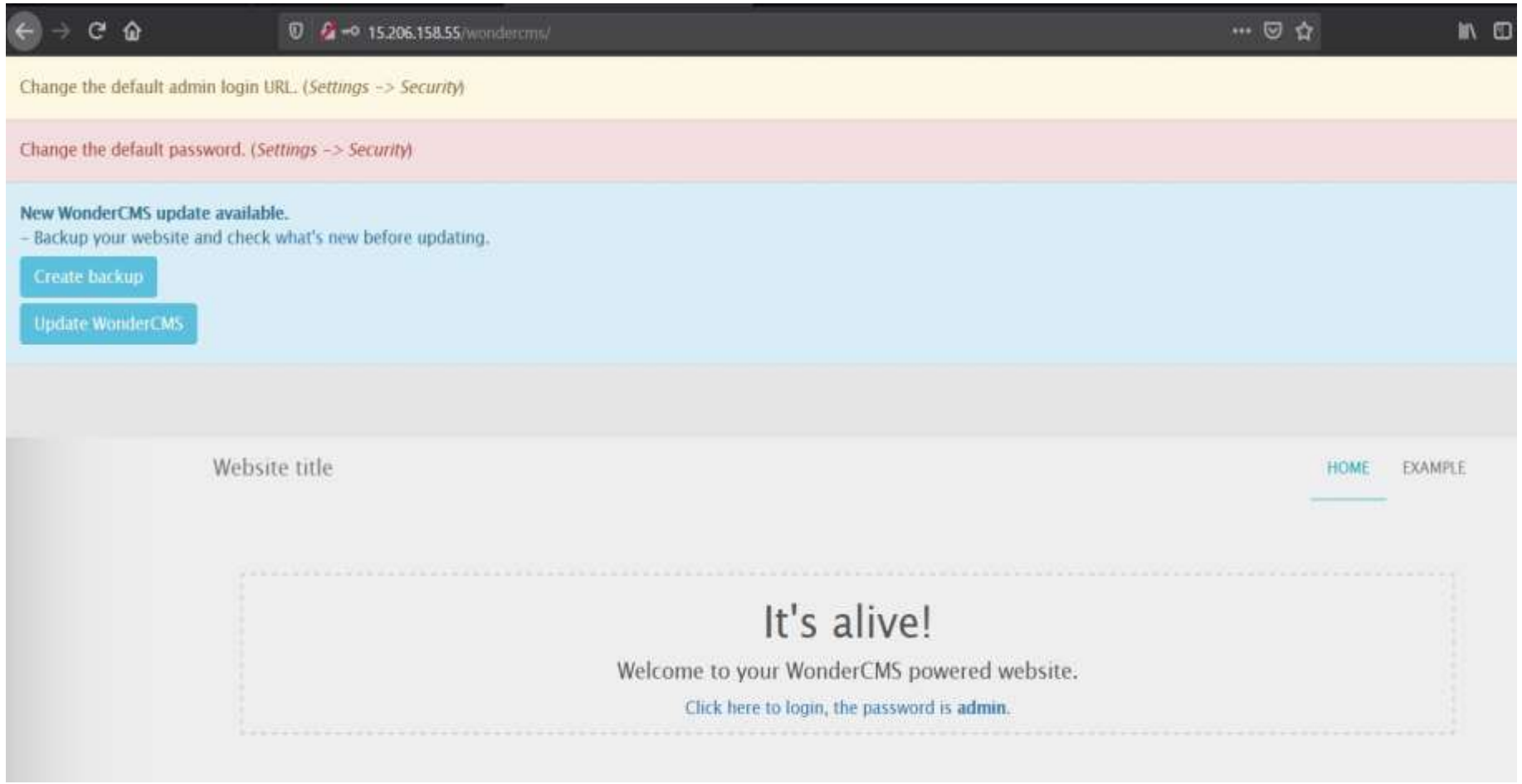
2.Insecure File Uploads

This happens when applications do not implement proper file type checking and allow uploading of files of different file formats. For example, a PHP file instead of a jpeg profile picture.

Insecure/arbitrary File Upload (Critical)	The attacker can upload insecure shells and files and gain access over the entire database and login as the admin and the version is known to have vulnerabilities **Affected URL :** • http://15.206.158.55/wondercms/ **Affected Parameters :** • File Upload (GET parameter) • The attacker can upload files with extension other than .jpeg

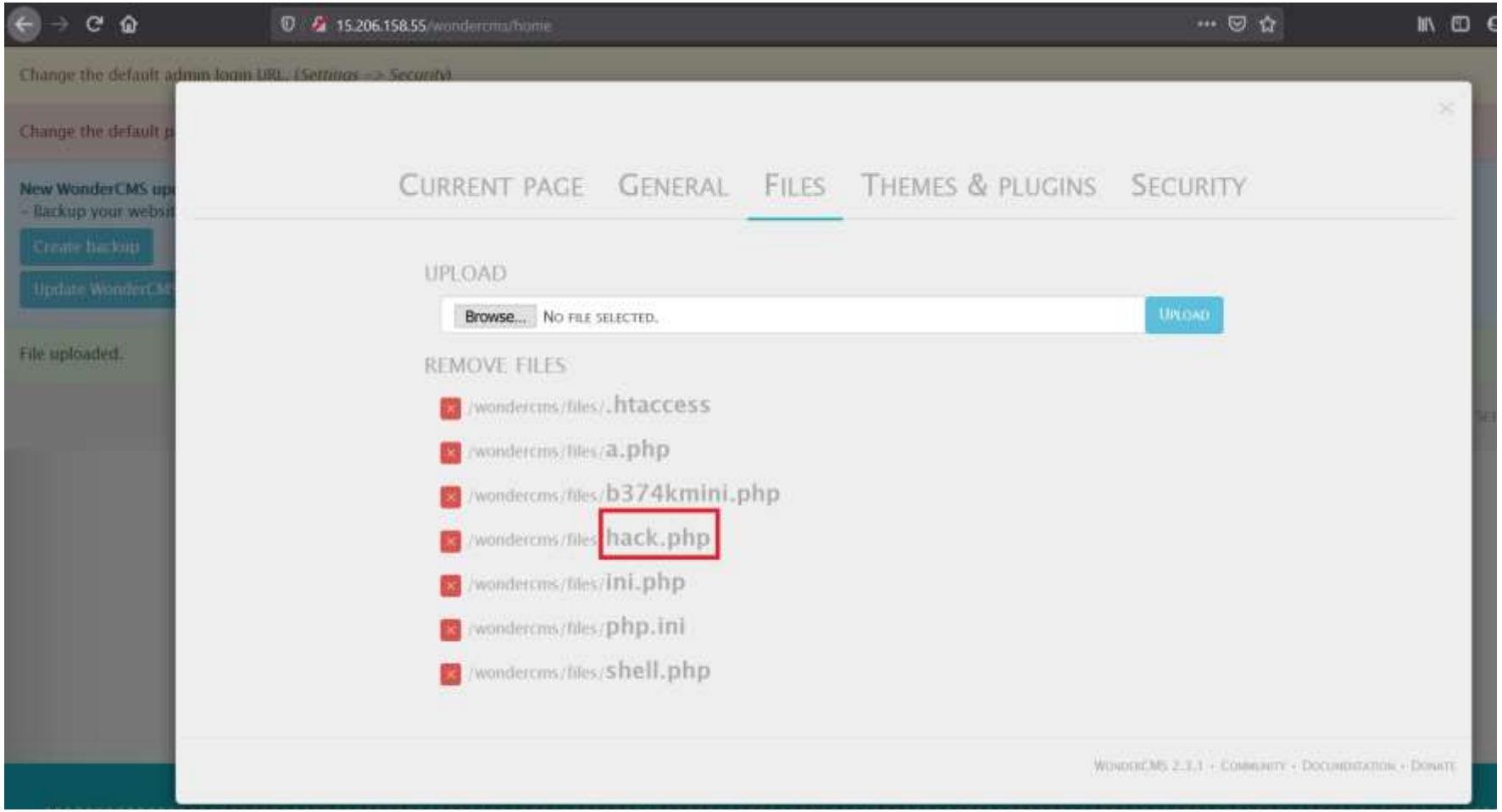
Observation

- Navigate to Blog tab . Now click on Login and put the password - admin. (already mentioned)
- You will see the following page and then click on Settings tab.



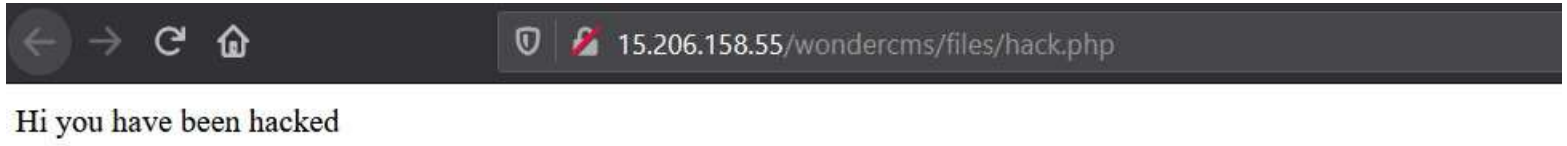
Observation

- Click on Files tab . Here hacker can upload the file like shown .
- Click on the uploaded file hack.php and it will be opened



Proof of Concept (PoC)

- Weak password - admin
- Arbitrary File Inclusion.
- Below is the result of the uploaded file in the previous slide likewise some malicious shell also can be uploaded



Business Impact – Extremely High

- Any backdoor file or shell can be uploaded to get access to the uploaded file on remote server and data can be exfiltrated. The presence of an actual malicious file can compromise the entire system leading to system takeover/ data stealing

Recommendation

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

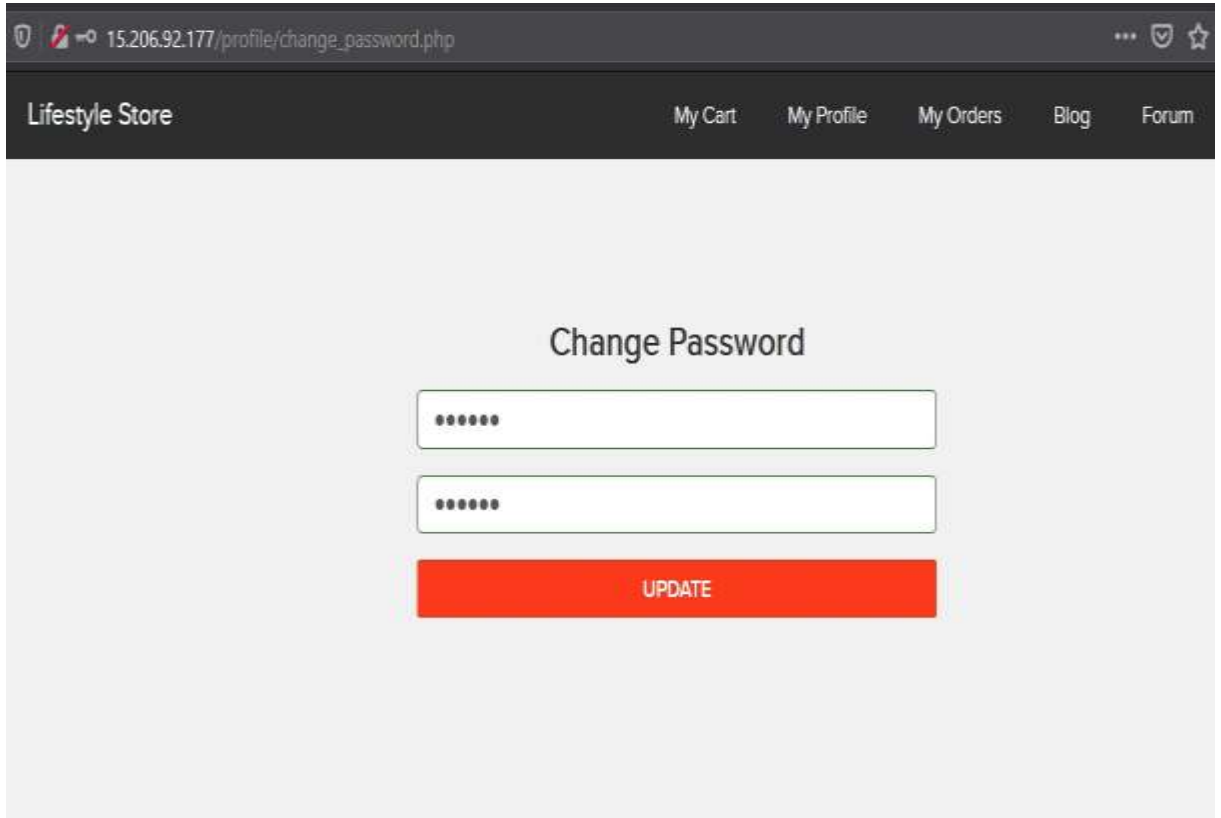
- **https://www.owasp.org/index.php/Unrestricted_File_Upload**
- **https://www.opswat.com/blog/file-upload-protection-best-practices**

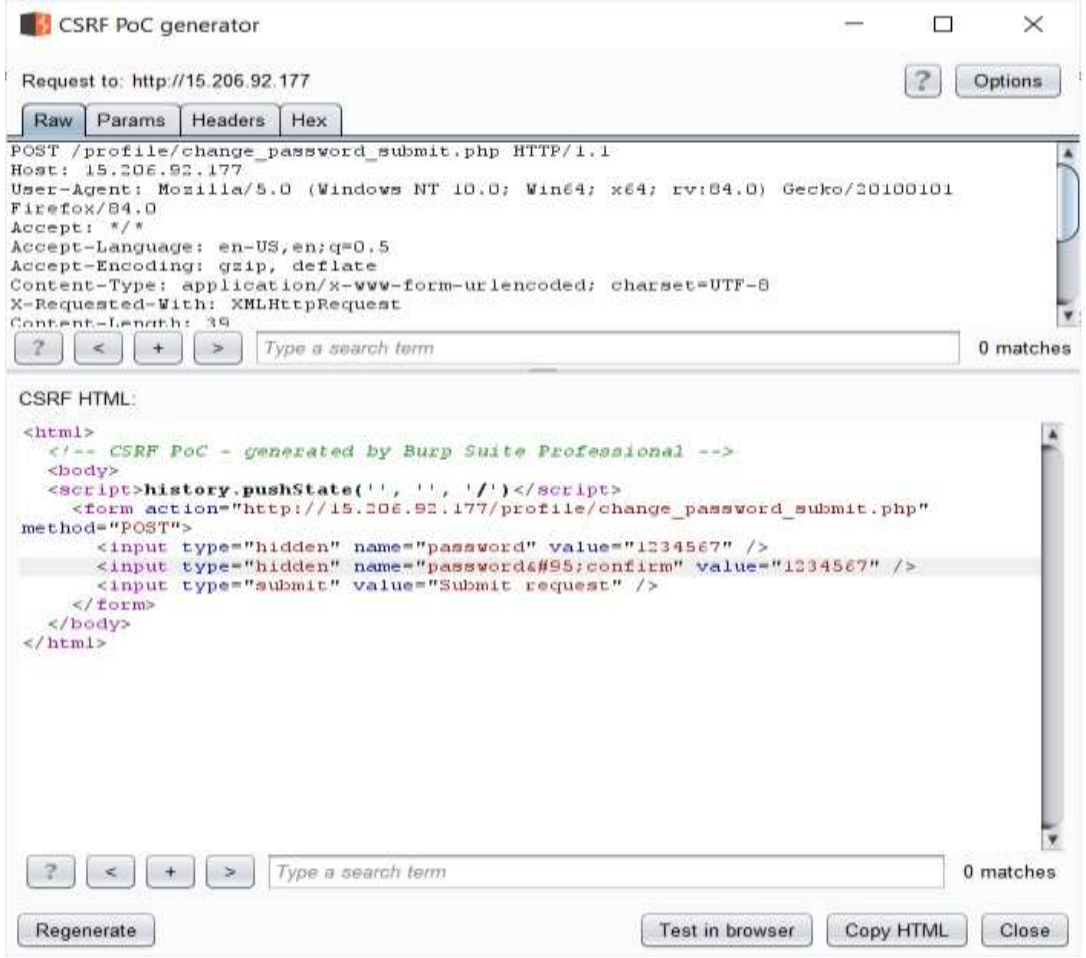
3.CSRF On Password Reset Page

Access to admin panel(Critical)	Below mentioned URL is vulnerable to CSRF on password reset page
	
	Affected URL :
	http://15.206.92.177/profile/change_password.php
	
	Payload
	
	`<html>`
	`  <!-- CSRF PoC - generated by Burp Suite Professional -->`
	`  <body>`
	`  <script>history.pushState('', '', '/')</script>`
	`  <form action="http://15.206.92.177/profile/change_password_submit.php" method="POST">`
	`  <input type="hidden" name="password" value="1234567" />`
	`  <input type="hidden" name="password_confirm" value="1234567" />`
	`  <input type="submit" value="Submit request" />`
	`  </form>`
	`  </body>`
	`</html>`

Observation

- navigate to http://15.206.92.177/profile/change_password.php and capture the request through Local proxy (burp suite)
- Generate a CSRF PoC for the change password request and input a password in the value field and test it in the browser





Proof of Concept (PoC)

- While submitting the request you will see a success message as shown below

Business impact - High

- An attacker can carry out CSRF attack to modify the password of a victim and take over the victim account.

Recommendation

This CSRF protection protects the form against Cross-site Request Forgery attacks because an attacker would also need to guess the token to successfully trick a victim into sending a valid request. The token should also be invalidated after some time and after the user logs out.

References

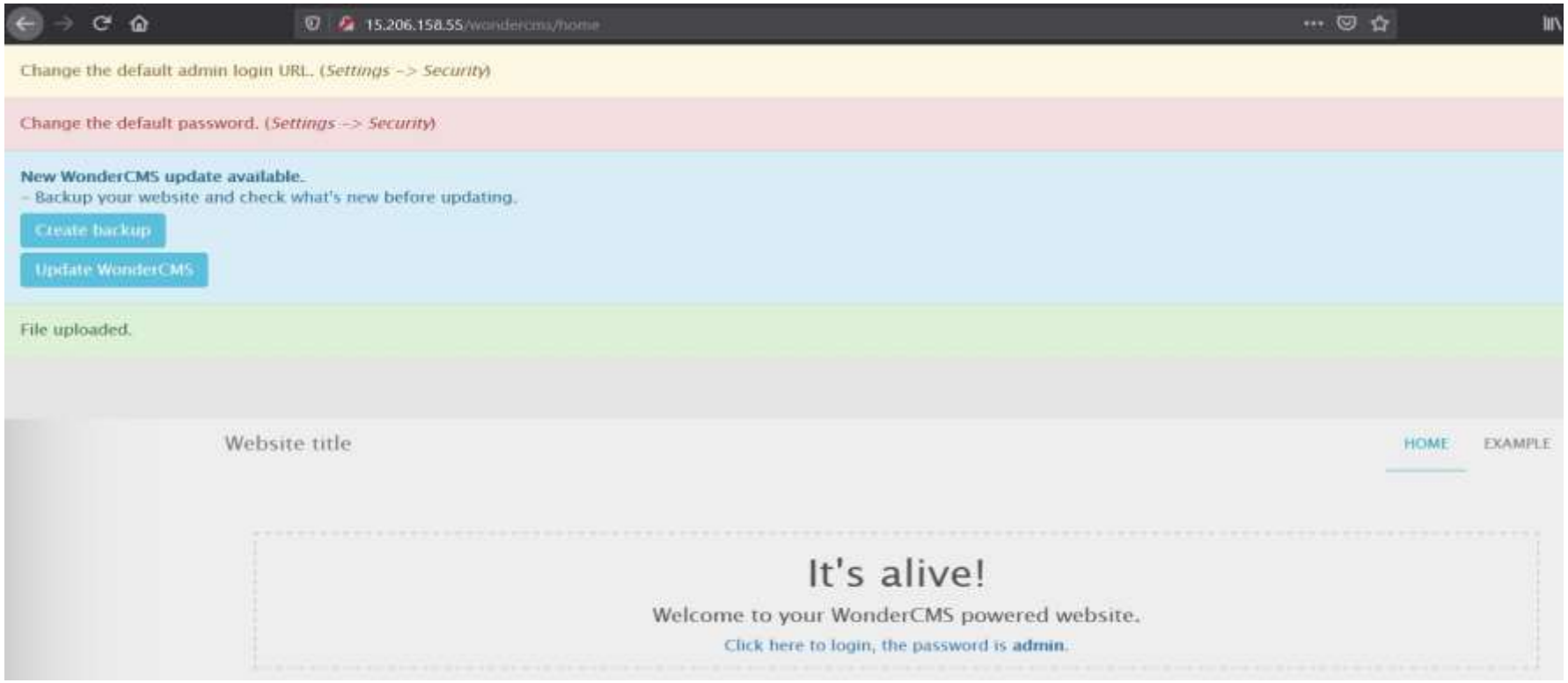
- https://owasp.org/www-community/attacks/csrf
- https://www.acunetix.com/websitesecurity/csrf-attacks/

4.Access to admin panel

Access to admin panel(Critical)	Below mentioned URL is vulnerable to Arbitrary File Upload and making other admin level changes. **Affected URL :** http://15.206.158.55/wondercms/loginURL

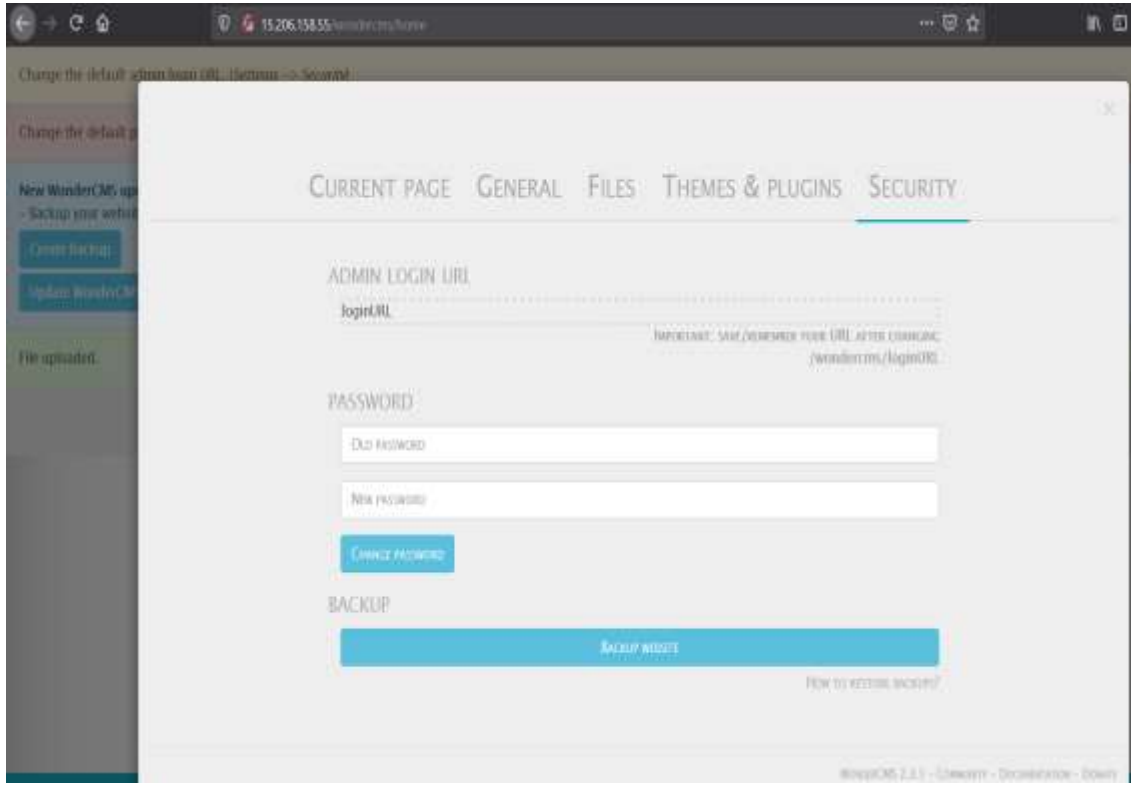
Observation

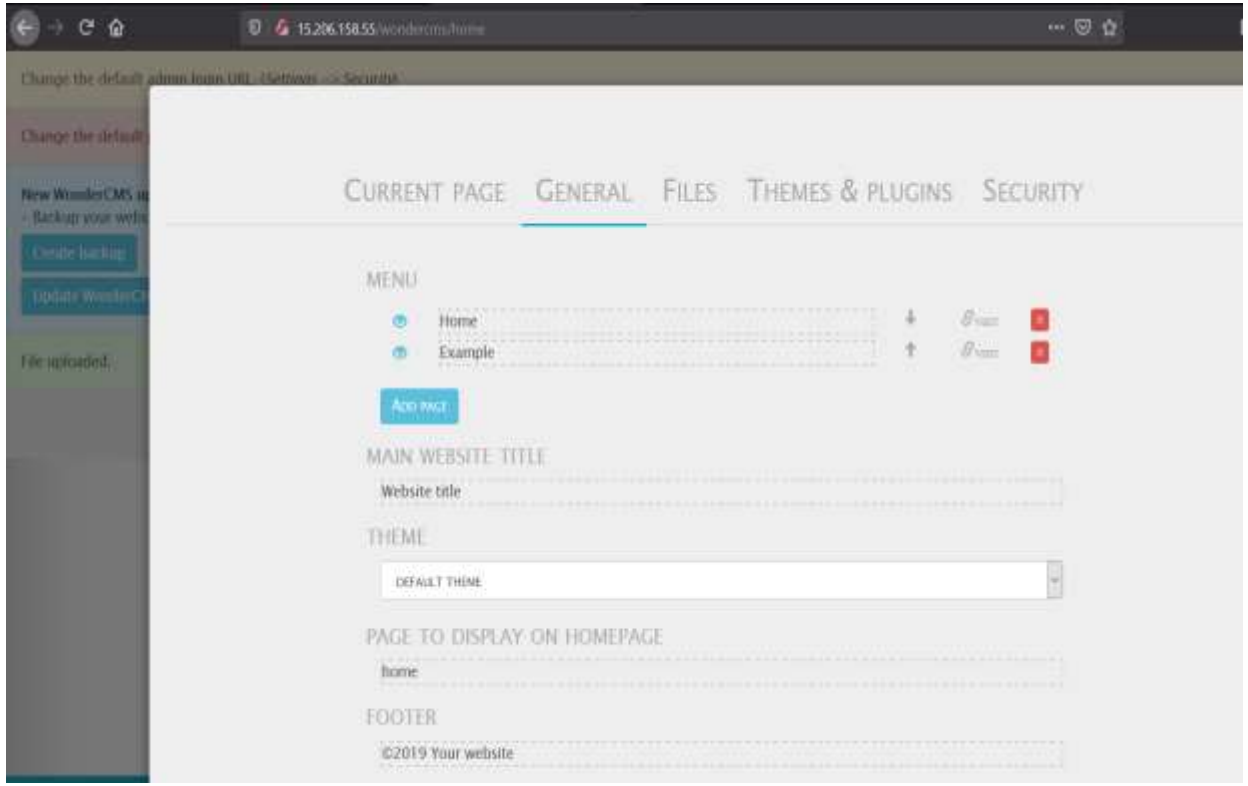
- When we navigate to http://15.206.158.55/wondercms/loginURL
- we get the password on the page: admin and login



Proof of Concept (PoC)

- Hacker can change the admin login password making the actual admin unable to login the next time
- Hacker can also add and delete pages





Business impact - Extremely High

- Using this vulnerability ,the attacker can get complete access to the blog of the website
- The attacker can change the password or even change the url of the admin panel and restrict the admin to access it
- Even pages can be created and deleted along with editing
- Files can be added (without verification) and hence can be dangerous to the entire website, as the control of the entire website can be taken

Recommendation

- The default password should be changed and a strong password must be setup.
- The admin url must also be such that its not accessible to normal users
- Password changing option must be done with 2 to 3 step verification.
- Password must be at least 8 characters long containing numbers, alphanumeric, capital letter, etc
- All the default accounts should be removed
- Password should not be reused

References

- https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)
- https://www.owasp.org/index.php/Default_Passwords
- https://www.us-cert.gov/ncas/alerts/TA13-175A

5.BruteForce Exploitation(OTP Bypass)

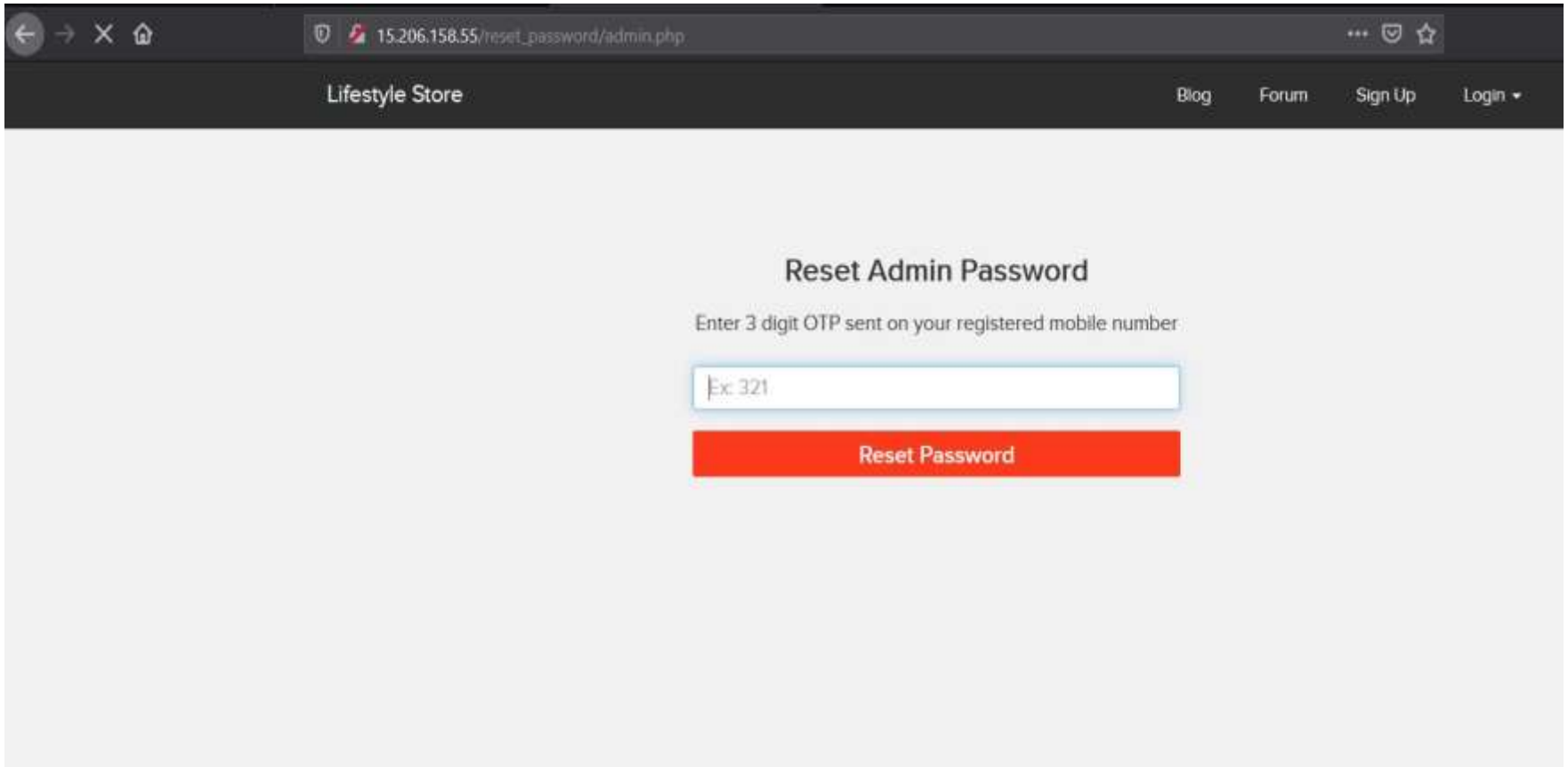
Brute Forcing OTP (Critical)	The admin dashboard at the below mentioned URL has 3-digit otp It is vulnerable to Bruteforcing. So, by brut forcing otp and resetting the password we can gain access **Affected URL :** • http://15.206.158.55/reset_password/admin.php **Affected Parameters :** • OTP (GET parameter) **Payload:** • otp=319

5.BruteForce Exploitation(Coupon Code)

Brute Forcing Coupon Code (Critical)	The coupon code at the below mentioned URL can be brute forced **Affected URL :** • http://15.206.158.55/cart/cart.php **Affected Parameters :** • Apply_coupon(POST parameters) **Payload:** • UL_1247

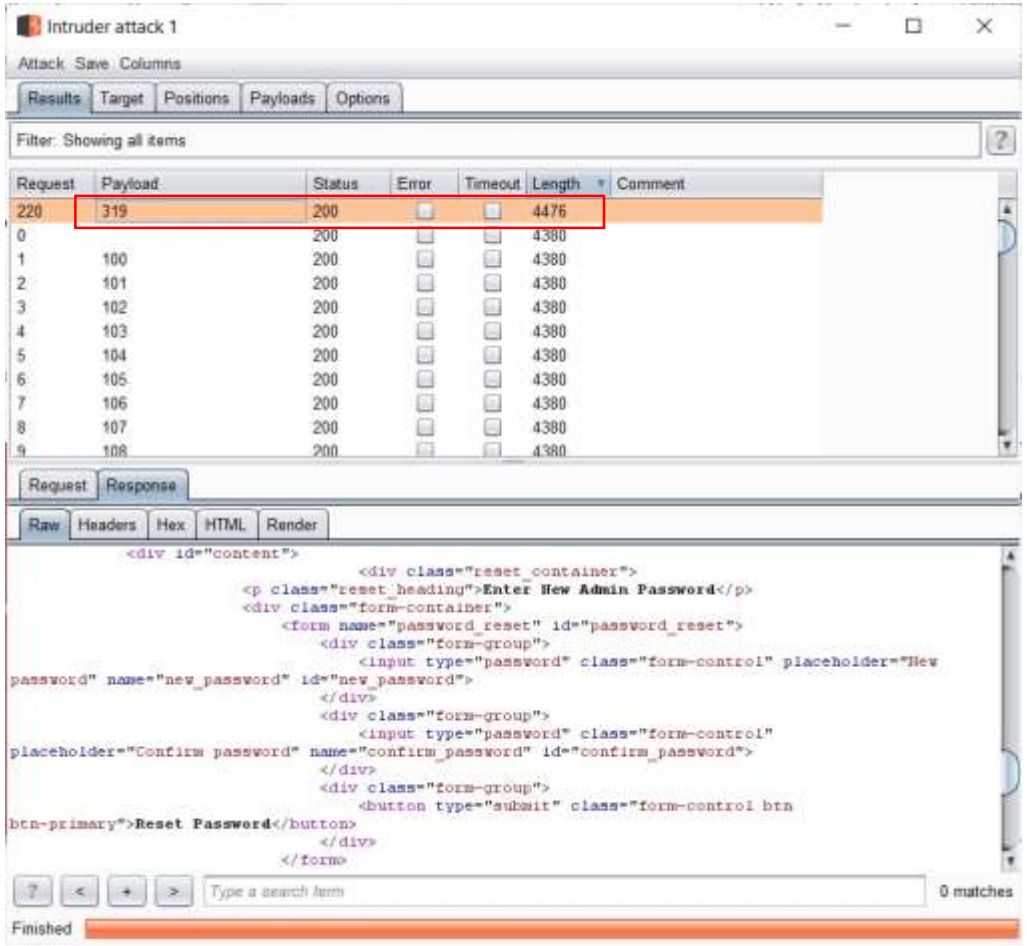
Observation

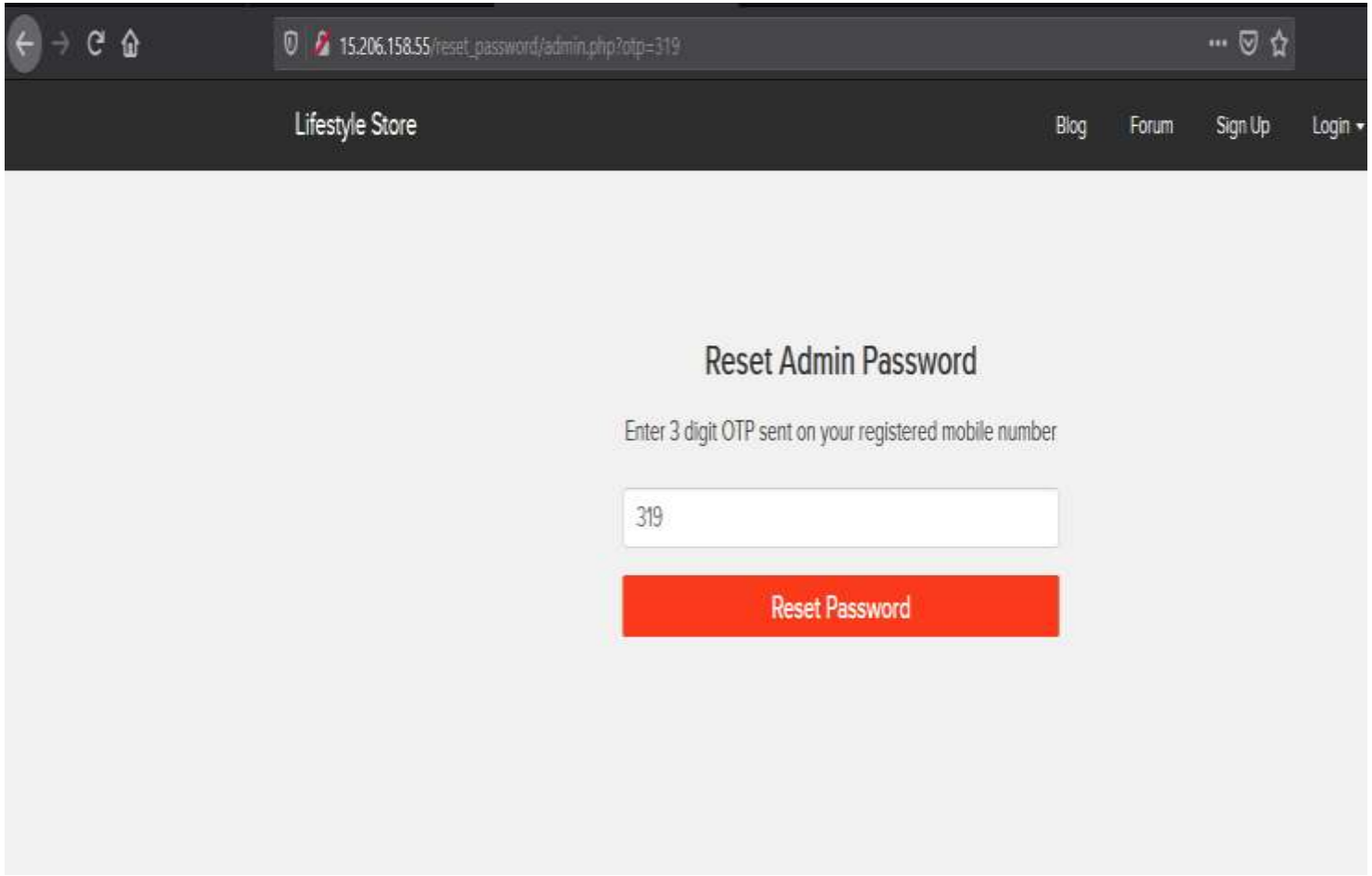
- Navigate to http://15.206.158.55/reset_password/admin.php You will see reset password page via OTP. Enter random otp and capture the requests in a local proxy (Burp Suite)



Observation

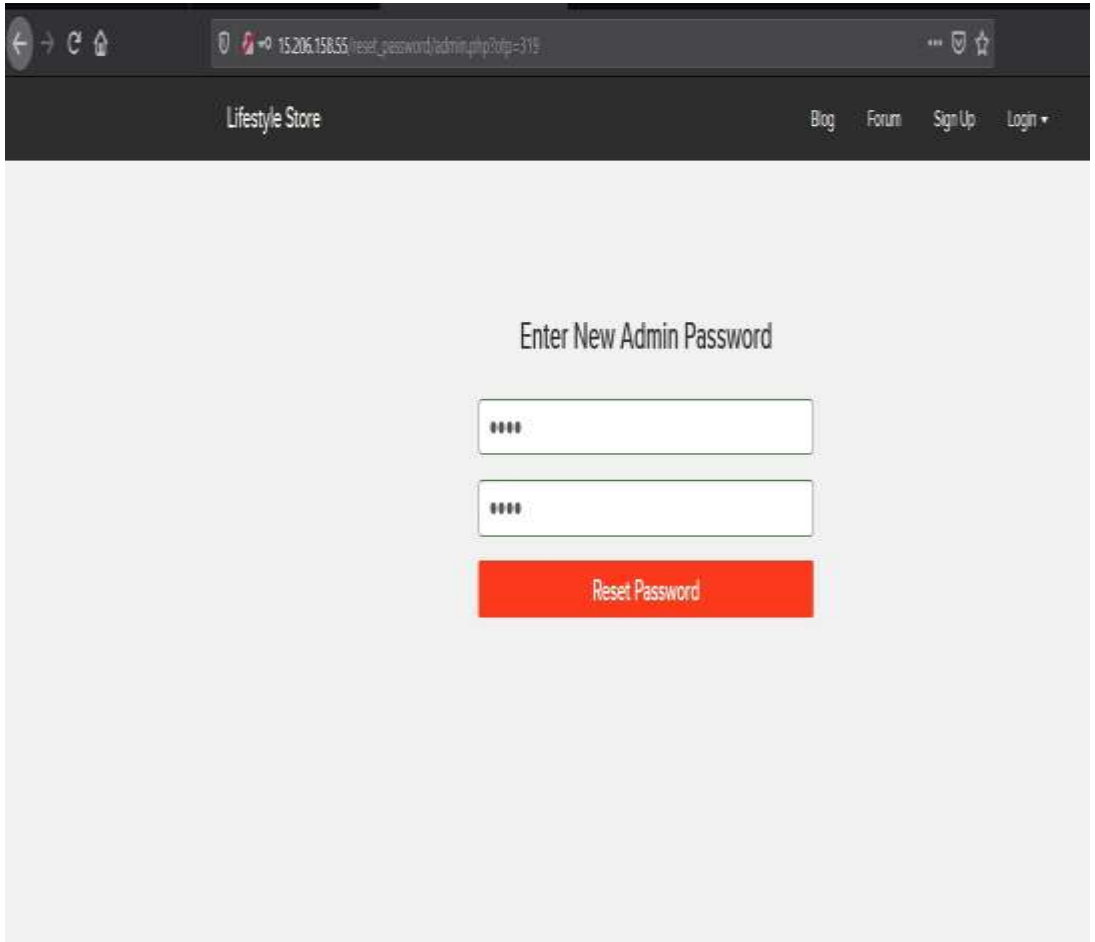
- On brute forcing the 3 digit otp , under the length column the value which is distinct from others yields the correct otp - 319
- Enter this otp in the otp column

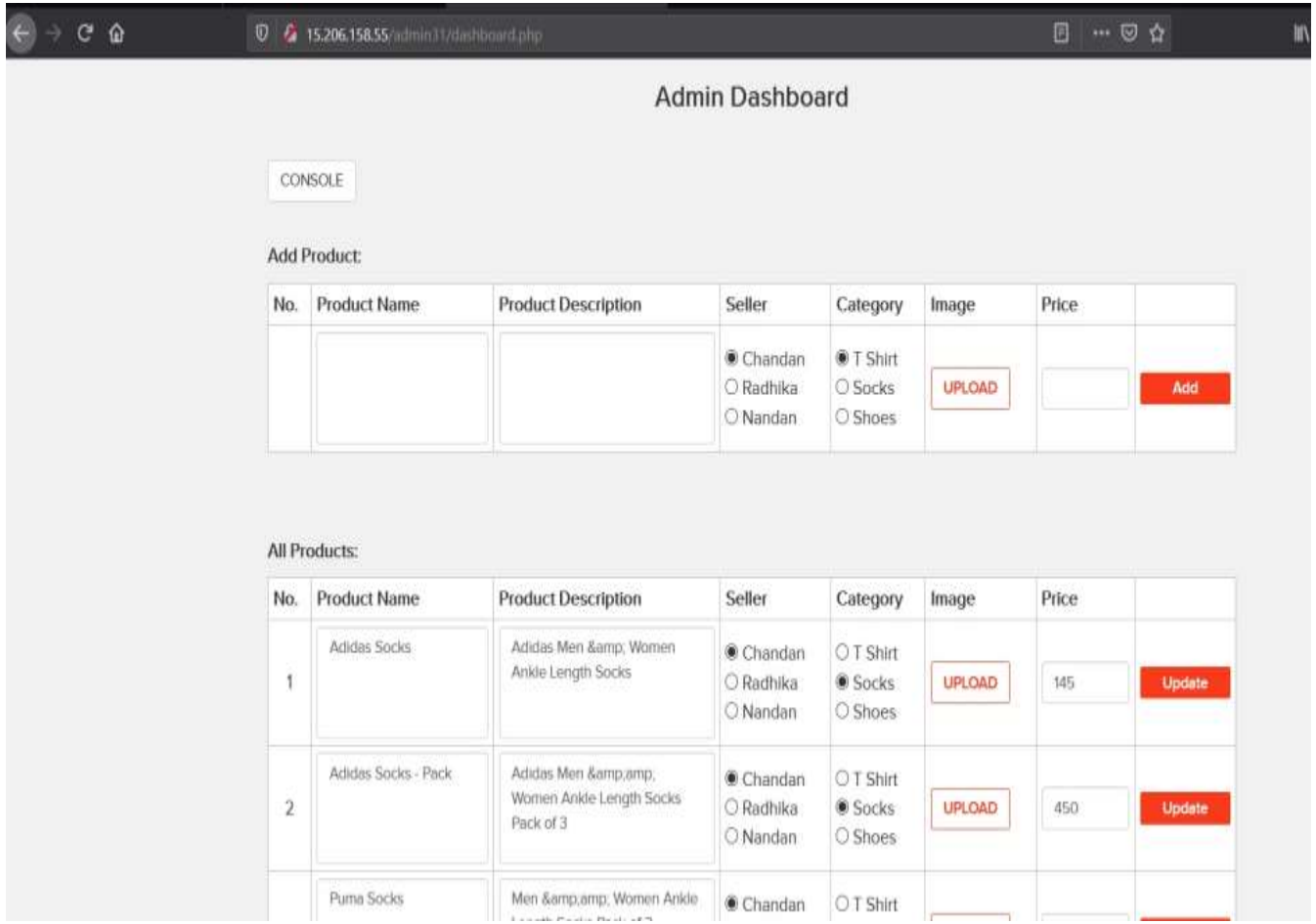




Observation

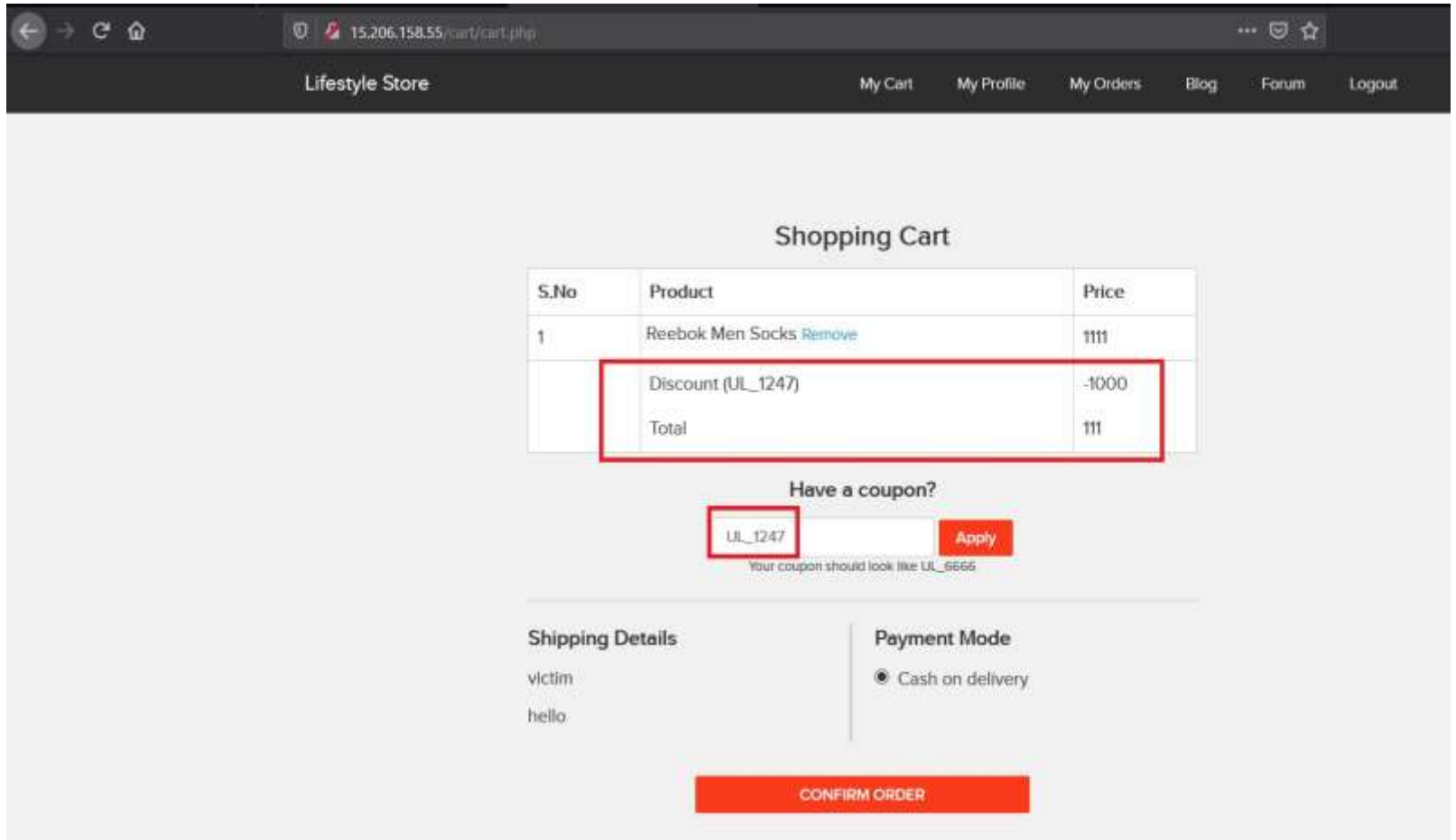
- You will be navigated to the reset password page .Here change the password
- Navigate to http:// 15.206.158.55 /login/admin.php. Enter username-admin and password – 1234 and you will be redirected to admin dashboard





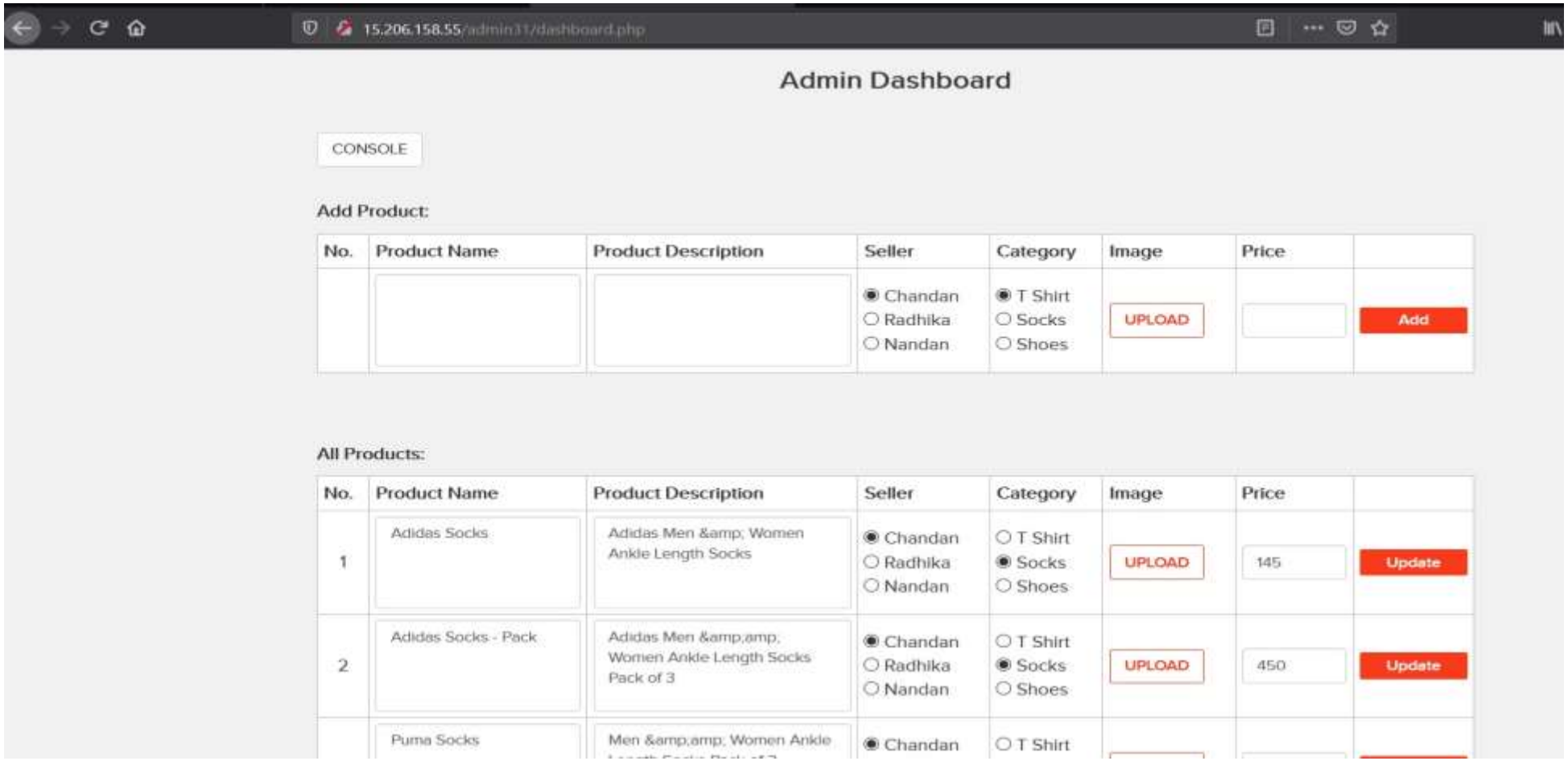
Proof of Concept (PoC)

- At url http://15.206.158.55/cart/cart.php coupon code - UL_1247 is applied.



Business Impact – Extremely High

A malicious hacker can gain access to any account and change the information about the products. This may lead to defamation of the seller and the website which the customer trusts. Attacker once logs in can then carry out actions on behalf of the admin which could lead to serious loss to any user



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

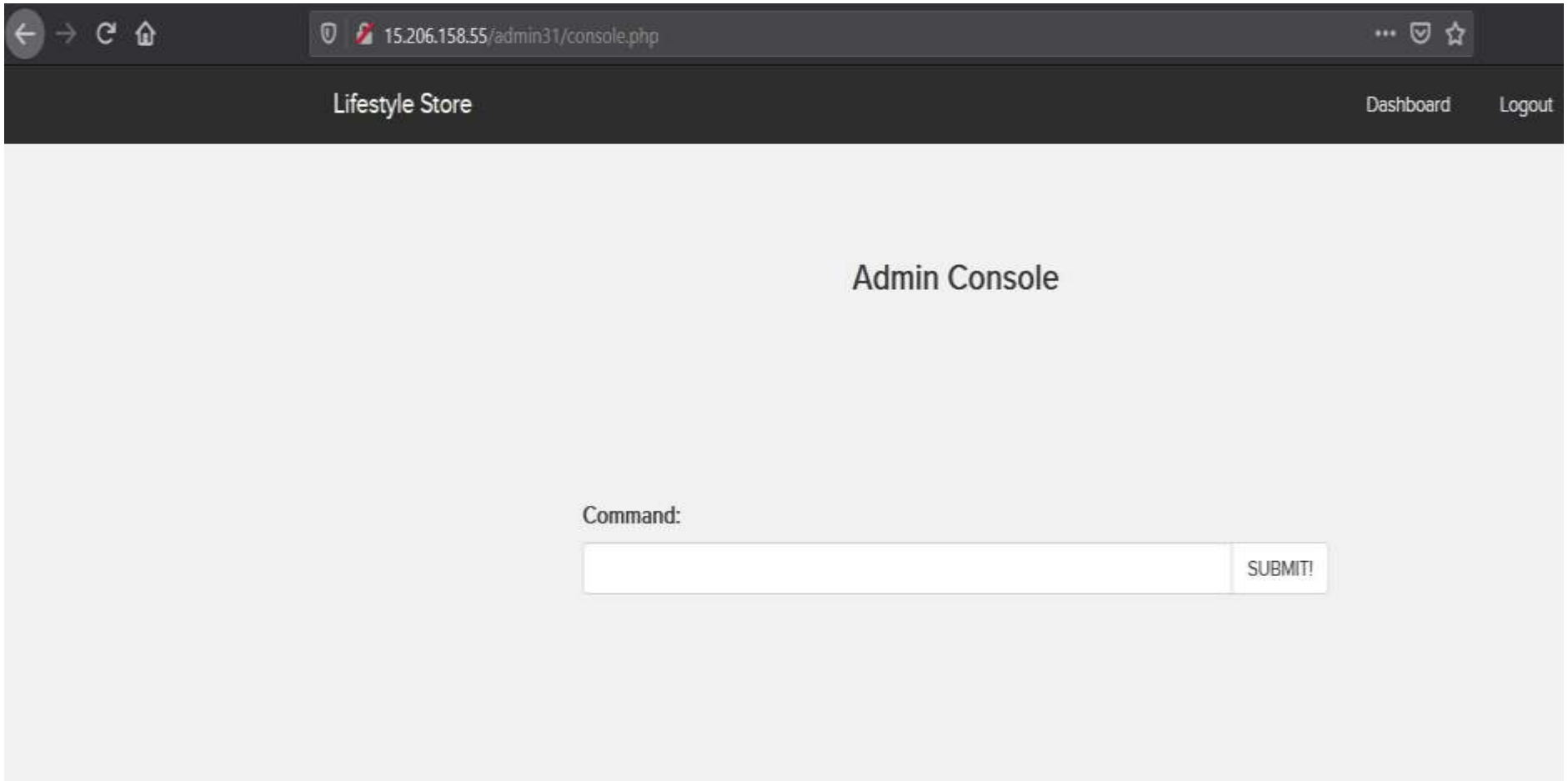
https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

6.Command Execution Vulnerability

Command Execution Vulnerability (Critical)	Below mentioned URLs is vulnerable to Command Execution. **Affected URL :** • http://15.206.158.55/admin31/console.php • http://13.232.156.73/wondercms/files/b374kmini.php?y=/home/trainee/wondercms/files/&x=shell **Affected Parameters :** • Command (POST parameter)

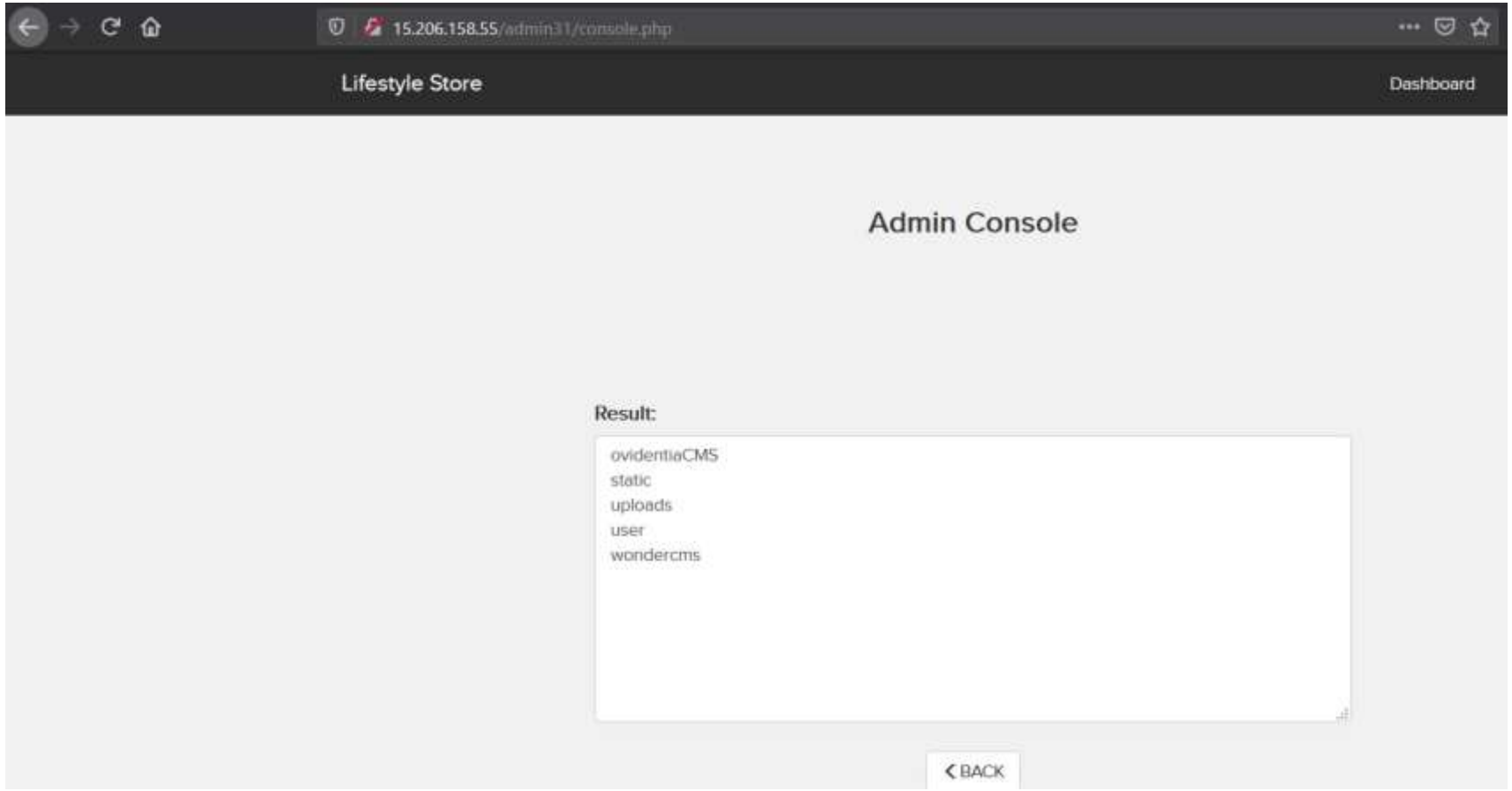
Observation

- Navigate to http://15.206.158.55/admin31/console.php after logging in as the admin and you will see the following page

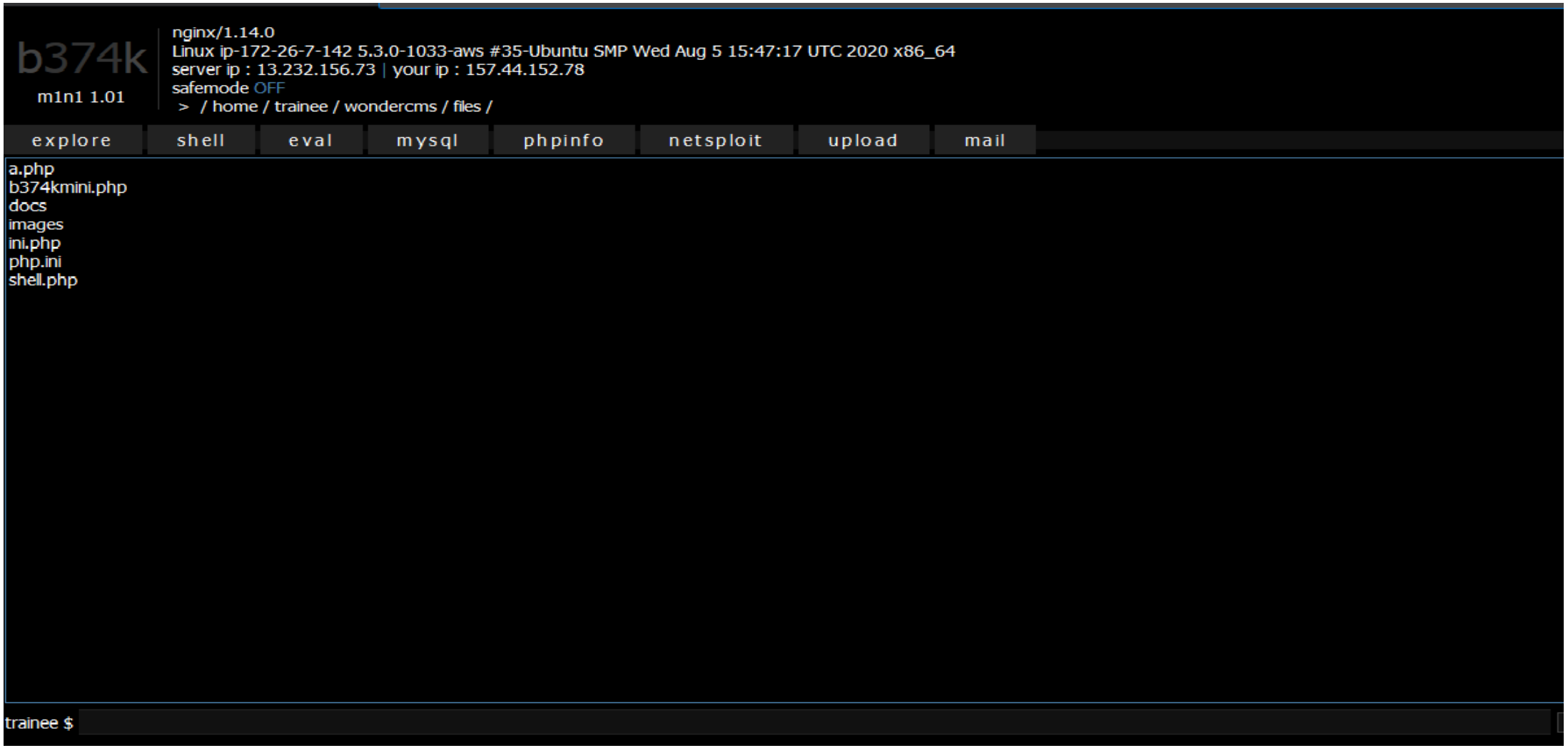


Proof of Concept (PoC)

- When command ls is entered the following output is visible.



Proof of Concept (PoC)



Business Impact – High

- If the attacker enters into the admin account and finally to the console url ,the he can put in any malicious code to extract or even edit data ,as he the has the admin privileges.
- Other than entering malicious code , the attacker can even get the details of the websites and its components like its version and hence find vulnerabilities to exploit them.
- If successfully exploited, impact could cover loss of confidentiality, loss of integrity, loss of availability, and/or loss of accountability

Recommendation

- There should be filters so that malicious code cannot be injected in
- Input validation can be done.
- Output Validation can be done
- Canonicalization can also be done

References

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

7.Unauthorized Access To Customer Details (IDOR)

IDOR (Critical)	The Show My Orders module is vulnerable from an Insecure Direct Object Reference (IDOR) that allows attacker see to anyone's user details. **Affected URL :** • http://15.206.158.55/orders/orders.php?customer=16 **Affected Parameters :** • customer (GET parameters) **Payload:** • http://15.206.158.55/orders/orders.php?customer=5 • http://15.206.158.55/orders/orders.php?customer=2 • http://15.206.158.55/orders/orders.php?customer=3 • http://15.206.158.55/orders/orders.php?customer=5 • http://15.206.158.55/orders/orders.php?customer=8 • http://15.206.158.55/orders/orders.php?customer=13 • http://15.206.158.55/orders/orders.php?customer=14

7.Unauthorized Access To Customer Details (IDOR)

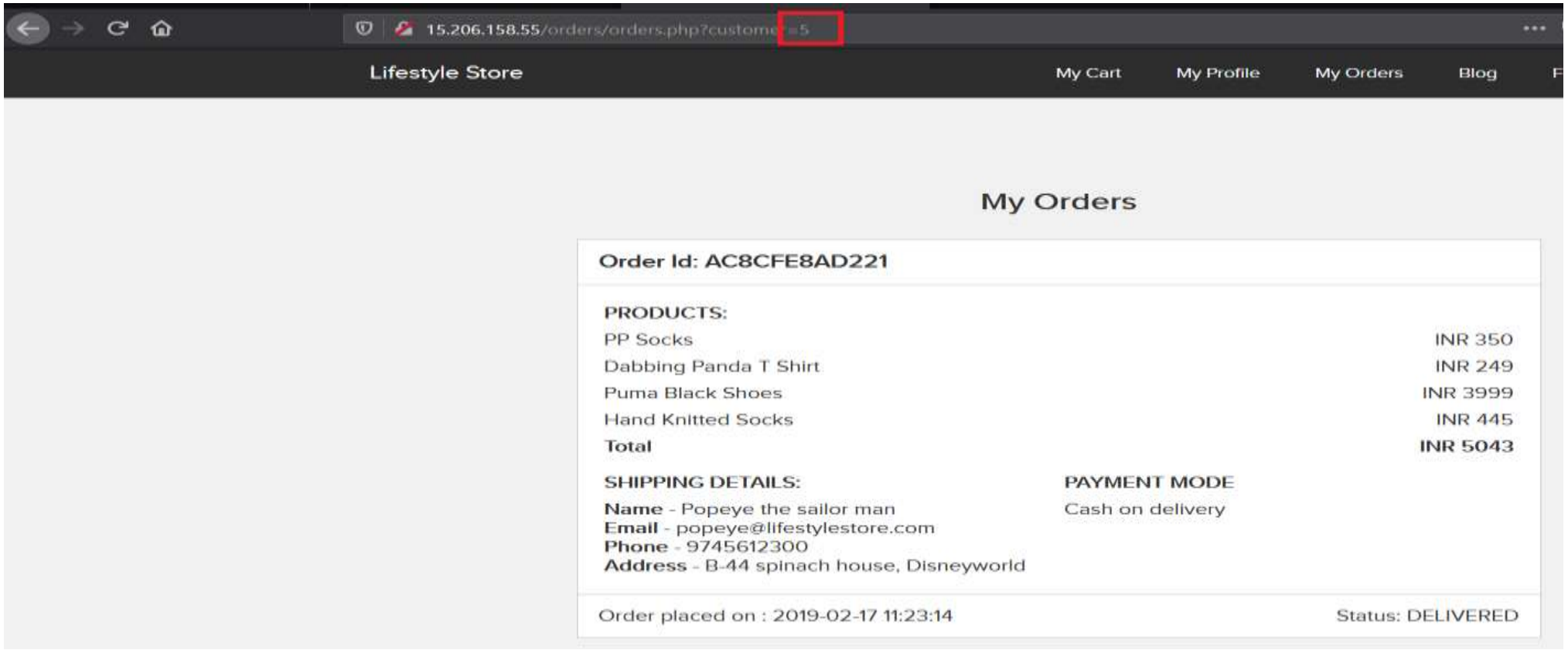
IDOR (Critical)	The Show profile module is vulnerable from an Insecure Direct Object Reference (IDOR) that allows attacker see to anyone's user details. **Affected URL :** • http://15.206.158.55/profile/16/edit/ **Affected Parameters :** • USER_ID (GET parameters) **Payload:** • http://15.206.158.55/orders/orders.php?customer=2 • http://15.206.158.55/orders/orders.php?customer=3 • http://15.206.158.55/orders/orders.php?customer=5 • http://15.206.158.55/orders/orders.php?customer=8 • http://15.206.158.55/orders/orders.php?customer=9 • http://15.206.158.55/orders/orders.php?customer=10 • http://15.206.158.55/orders/orders.php?customer=11 • http://15.206.158.55/orders/orders.php?customer=12 • http://15.206.158.55/orders/orders.php?customer=13 • http://15.206.158.55/orders/orders.php?customer=14 • http://15.206.158.55/orders/orders.php?customer=15

Observation

- Login as customer. Then navigate to the below link

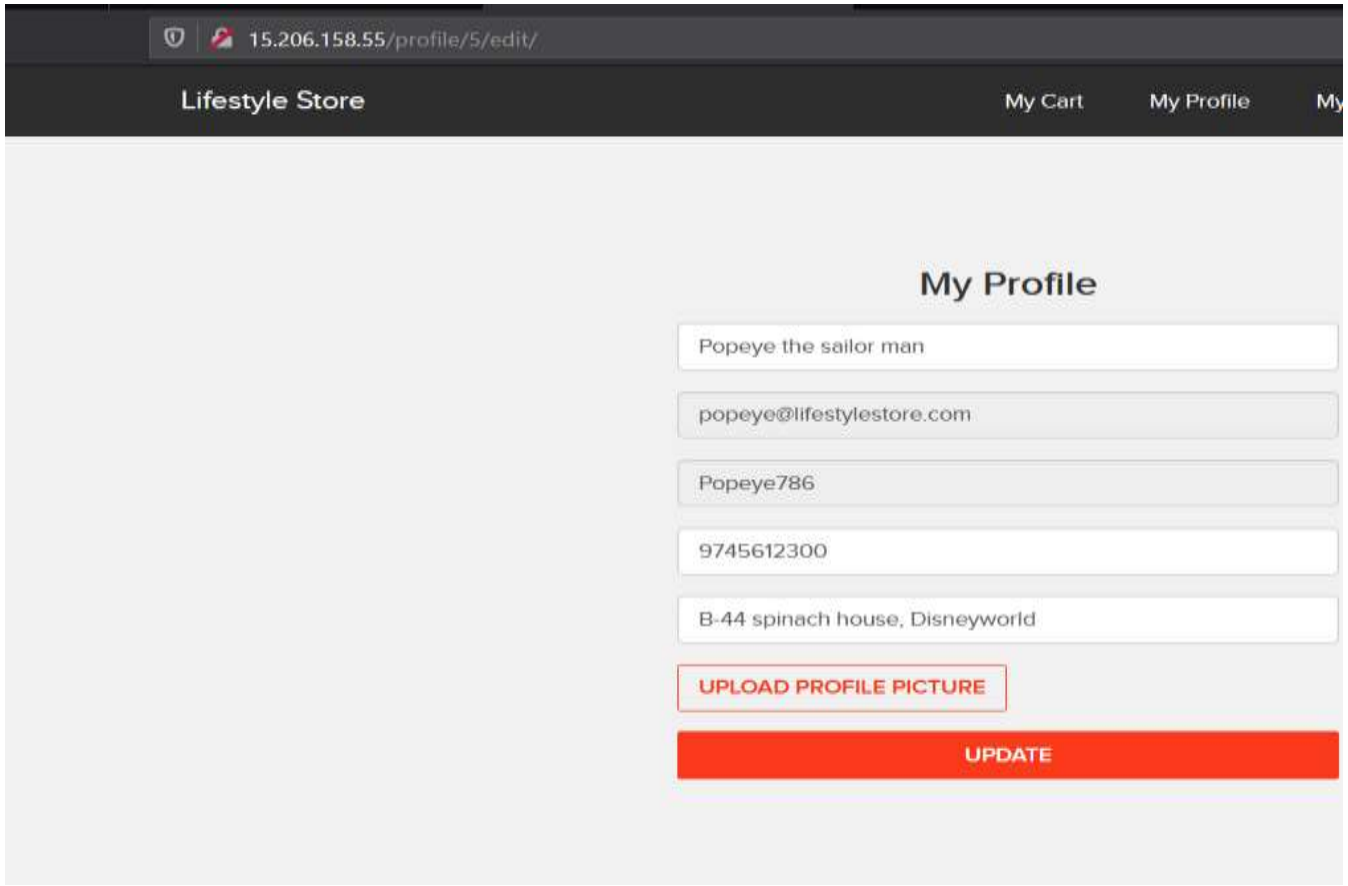
http://15.206.158.55/orders/orders.php?customer=16

- Now remove 16 and insert 5 in the url like shown in the given screenshot and you will see the details of another user



Proof of Concept (PoC)

- Below is the screenshot of the account details of another user accessed from attacked user's account



Business Impact – Extremely High

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

- Mobile Number
- Bill Number
- Billing Period
- Bill Amount and Breakdown
- Phone no. and email address
- Address

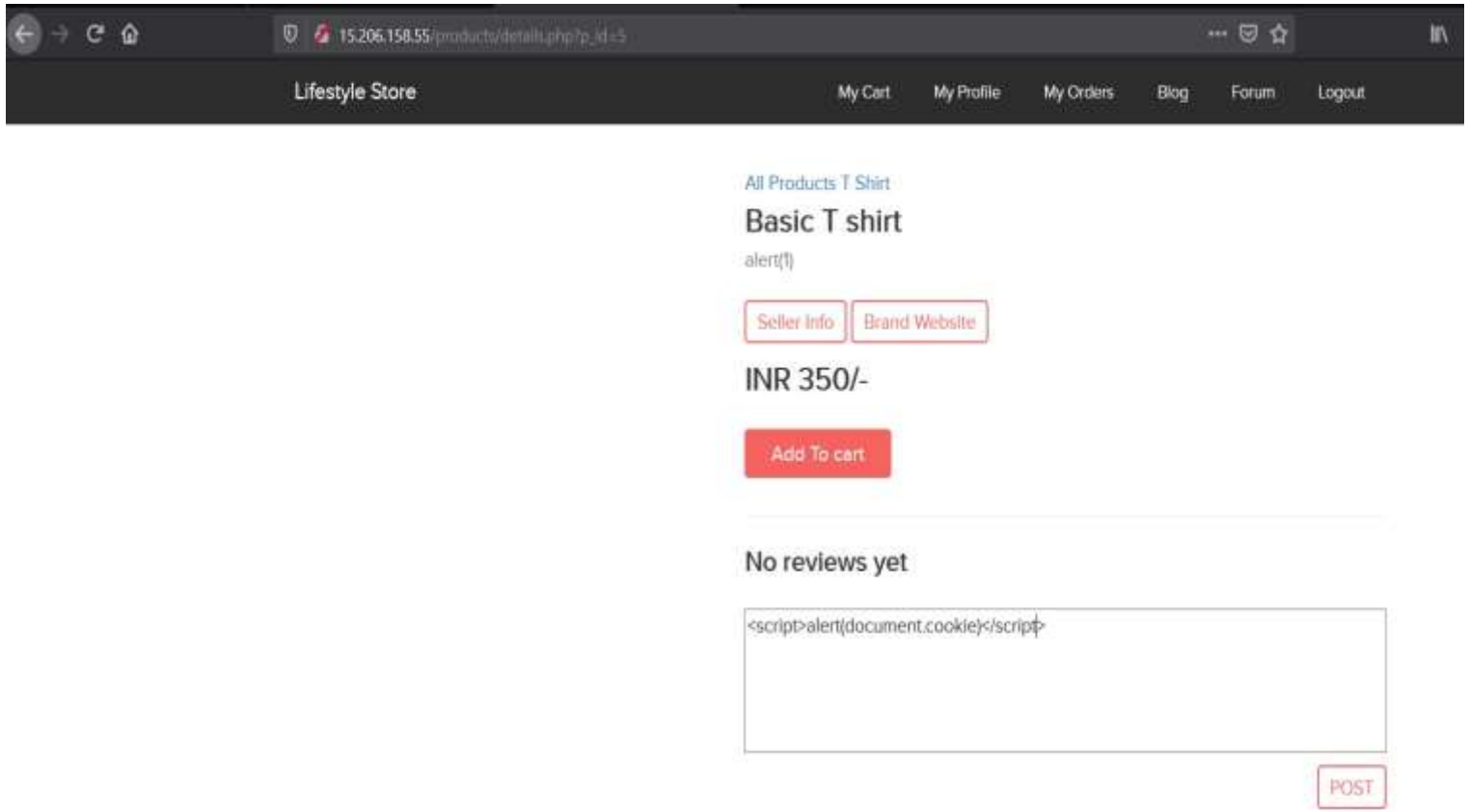
More over, as there is no rate limiting checks, attacker can brute force the user_id for all possible values and get bill information of each and every user of the organization resulting is a massive information leakage.

8.Cross Site Scripting (Stored Xss)

Cross Site Scripting(critical)	Below mentioned parameters are vulnerable to Stored XSS **Affected URL :** 1. http://15.206.158.55/products/details.php?p_id=5 (permanent xss) **Affected Parameters :** 1. POST button under Customer Review (POST parameter) **Payload:** 1. <script>alert(document.cookie)</script>

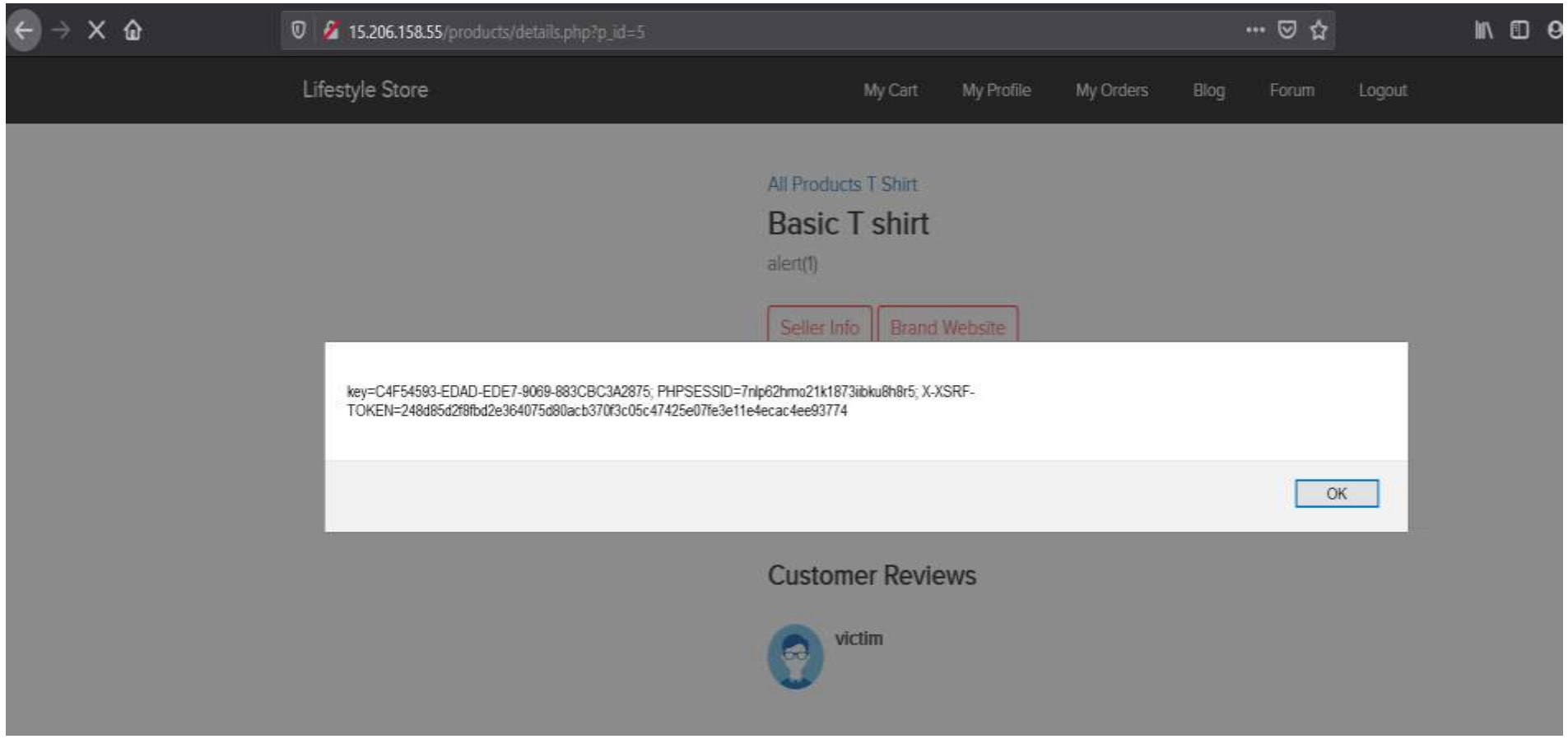
Observation

- Navigate to http://15.206.158.55/products/details.php?p_id=5 .You will see products details. Put in this payload (<script>alert(document.cookie)</script>) on the review box and post it



Proof of Concept (PoC)

- As you can see we executed custom JS causing popup of the cookie

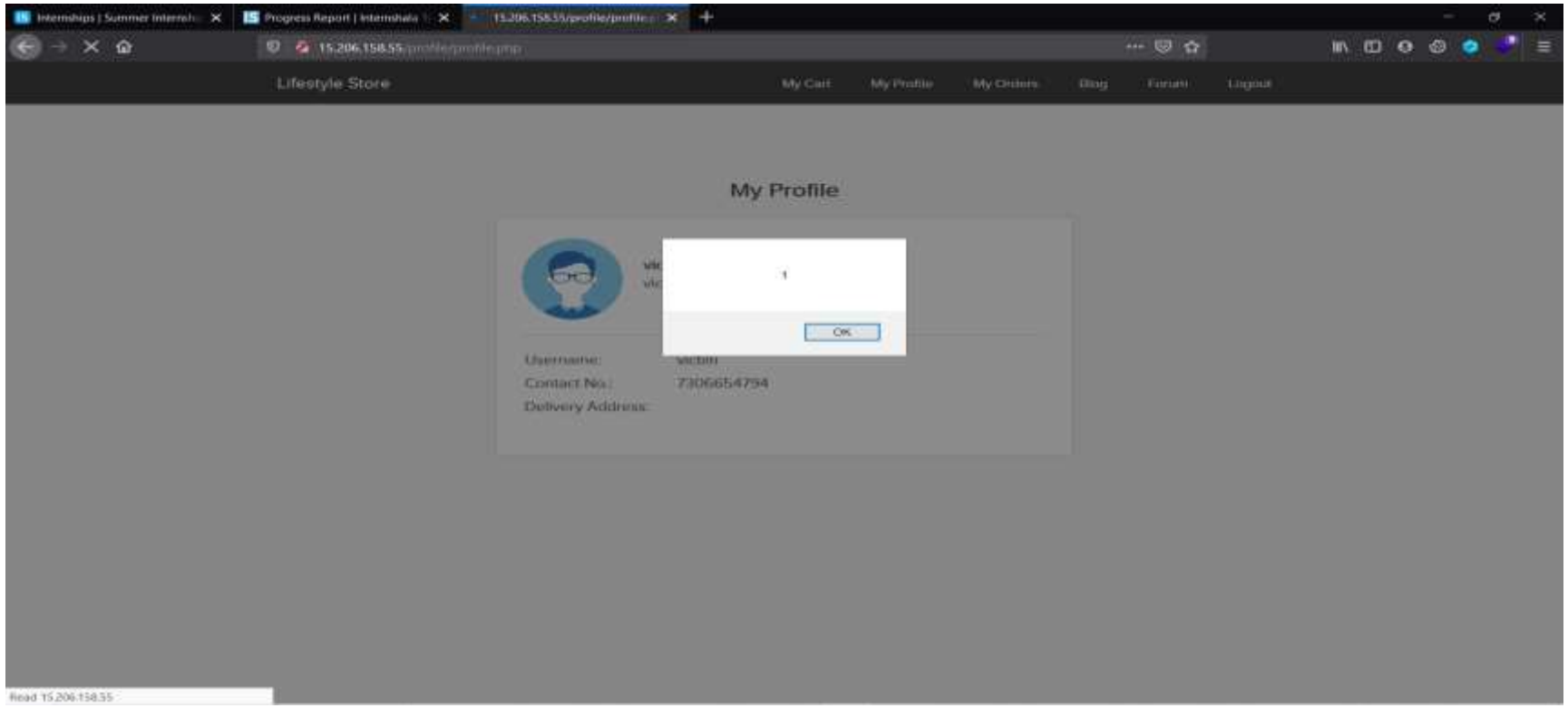


8.Cross Site Scripting (Reflective Xss)

Cross Site Scripting(Severe)	Below mentioned parameters are vulnerable to reflected XSS **Affected URL :** 1. http://15.206.158.55/profile/16/edit 2. http://15.206.158.55/search/search.php?q= **Affected Parameters :** 1. Address (POST parameter) 2. Q= (GET parameter) **Payload:** 1. <script>alert(1)</script> 2. "><script>alert(1)</script>

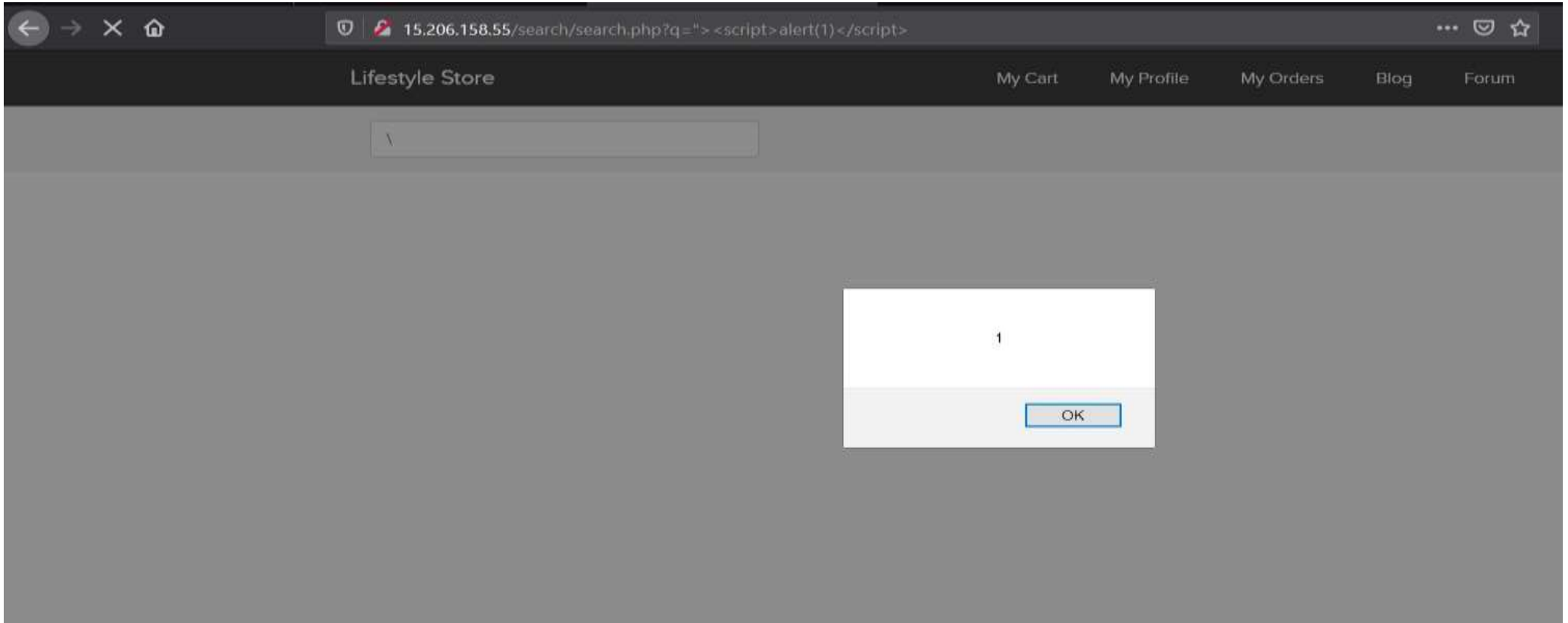
Proof of Concept (PoC)

- When we put <script>alert(1)</script> in to the address box in the profile page “1” is reflected



Proof of Concept (PoC)

- When we put "><script>alert(1)</script>
- In to the url http://15.206.158.55/search/search.php?q= "><script>alert(1)</script>. We get “1” reflected back



Business Impact – High

- As attacker can inject JS payloads 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 organisation
- 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 " %22 < > before printing them on the website.
- Apply Client Side Filters to prevent client side filters bypass.

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

Recommendation

Take the following precautions:

- Implement proper authentication and authorisation checks to make sure that the user has permission to the data he/she is requesting Use proper rate limiting checks on the number of request comes from a single user in a small amount of time
- Make sure each user can only see his/her data only

References

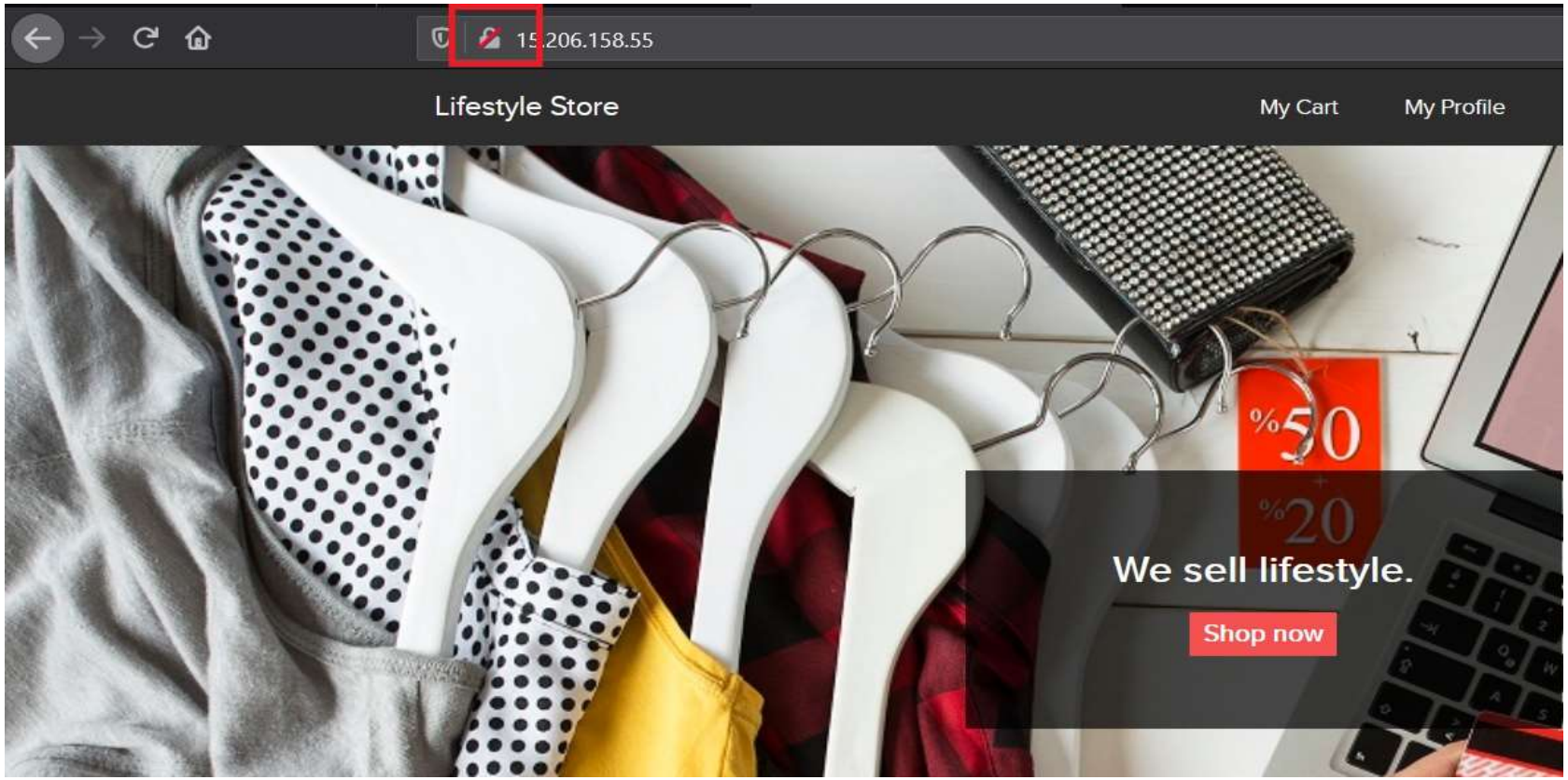
- https://www.owasp.org/index.php/Insecure_Configuration_Management
- https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References

9.Crypto Configuration Flaws

Crypto Configuration Flaws (Severe)	Crypto Configuration Flaws are found in the modules below. **Affected URL :** • http://15.206.158.55/ (All the webpages ,blogs ,forum,etc.)

Observation

- Clearly ,all the webpages use 'http' and not 'https' which is far less secure and not encrypted.



Business Impact - High

- Security is almost halved in http providing easy man-in-the-middle attack and others which makes it easy for attacker to go through the data transmitted over the internet

Recommendation

- Use https instead of http as the protocol

References

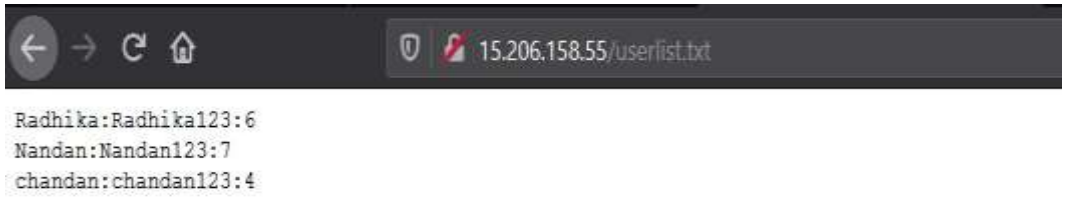
- **https://www.owasp.org/index.php/Category:Cryptographic_Vulnerability**
- **https://www.w3.org/Protocols/rfc2616/rfc2616-sec15.html**

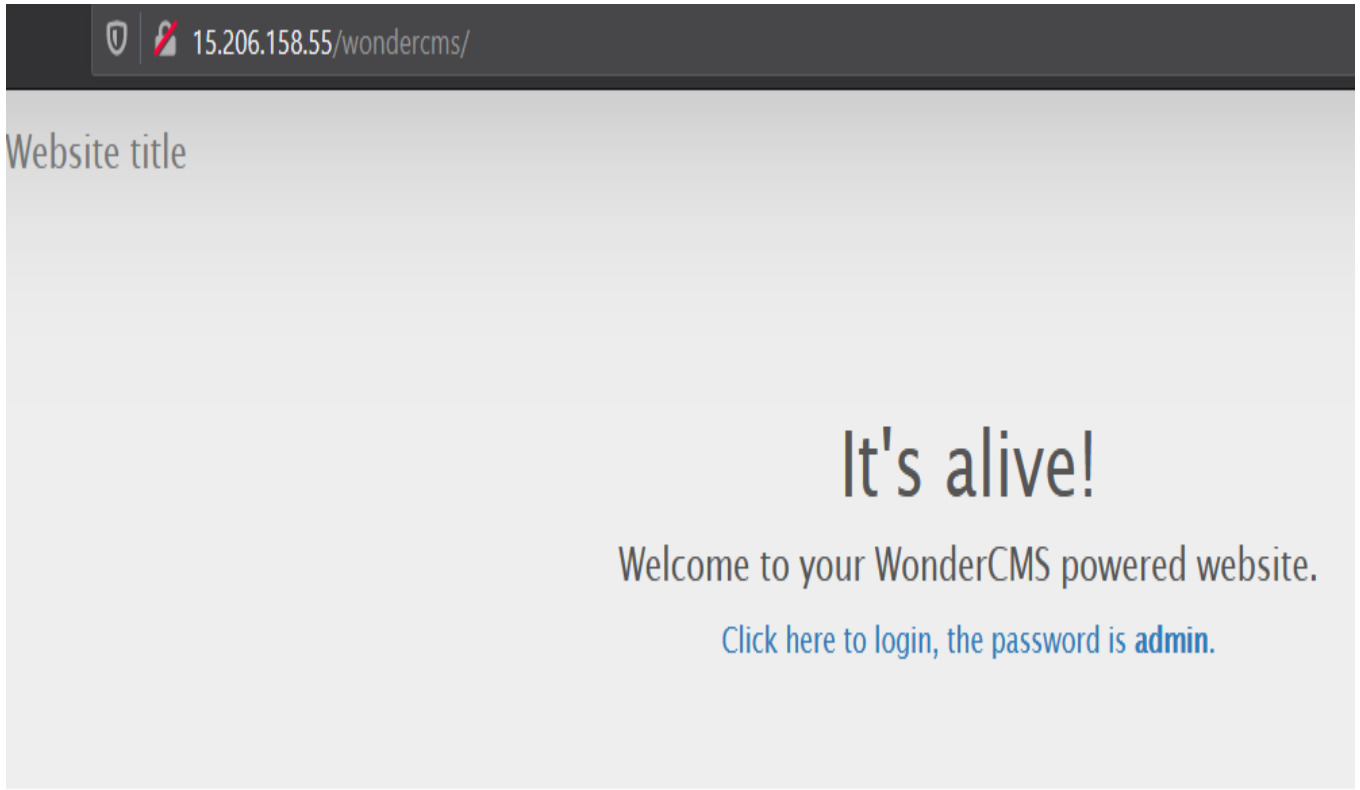
10.Common Passwords

Common Passwords (Severe)	Below given urls have weak passwords **Affected URL :** • http://15.206.158.55/login/seller.php • http://15.206.158.55/wondercms/

Observation

- The passwords of sellers and ,admin of blog ,is very common and easily predictable





Business Impact - High

- Easy, default and common passwords make it easy for attackers to gain access to their accounts illegal use of them and can harm the website to any extent after getting logged into privileged accounts

Recommendation

- There should be password strength check at every creation of an account
- There must be a minimum of 8 characters long password with a mixture of numbers ,alphanumeric ,special characters ,etc
- There should be no repetition of password ,neither on change nor reset
- The password should not be stored on the web, rather should be hashed and stored

References

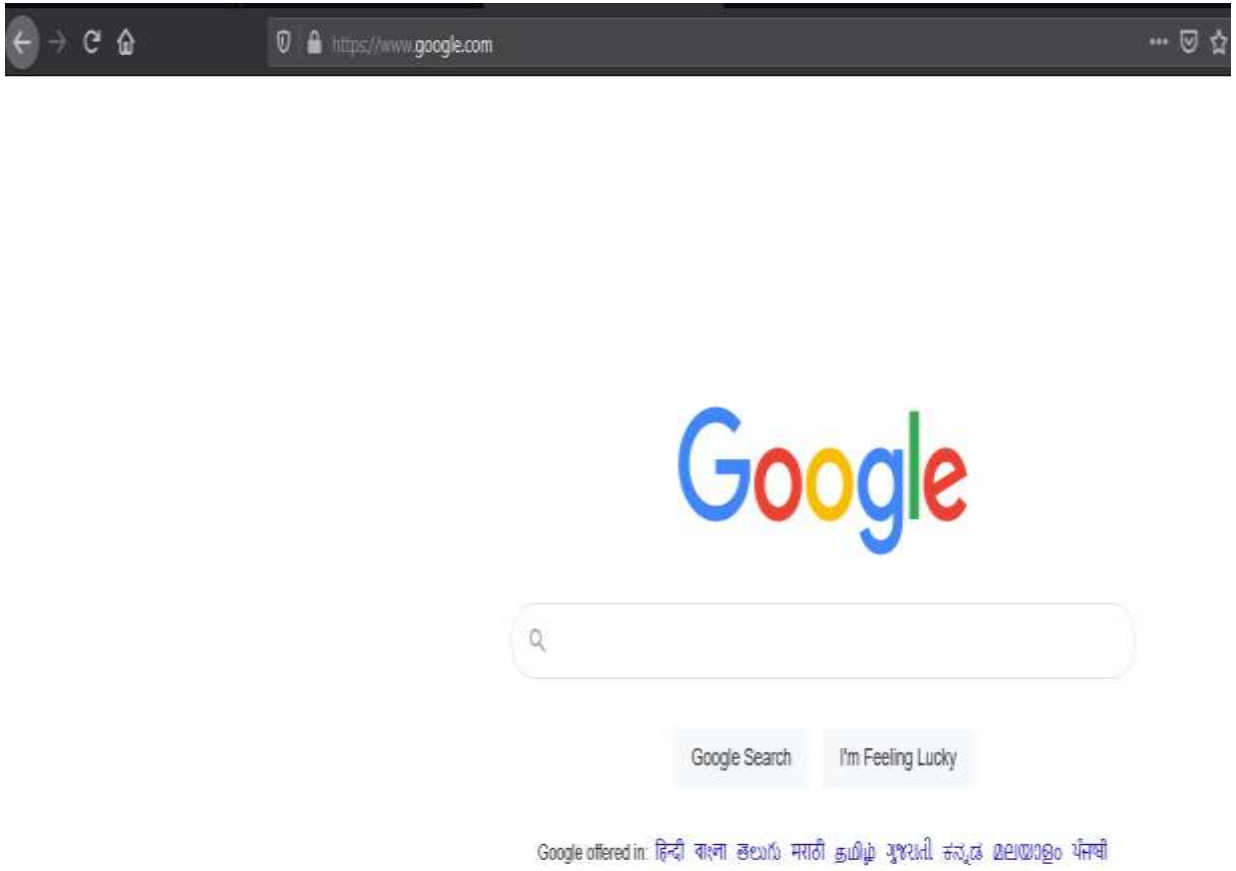
- https://www.acunetix.com/blog/articles/weak-password-vulnerability-common-think/
- https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007)

11.Open Redirection

Open Redirection (Severe)	. **Affected URL :** • http://13.127.47.121/redirect.php?url= • **Affected Parameters :** • url(GET parameters) **Payload:** • https://ww.google.com

Observation

- Navigate to http://13.127.47.121/redirect.php?url=www.radhikafancystore.com
- Now edit the URL like this http://13.127.47.121/redirect.php?url=https://ww.google.com
- You will see the google.com



Business Impact – High

- An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance

Recommendation

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

References

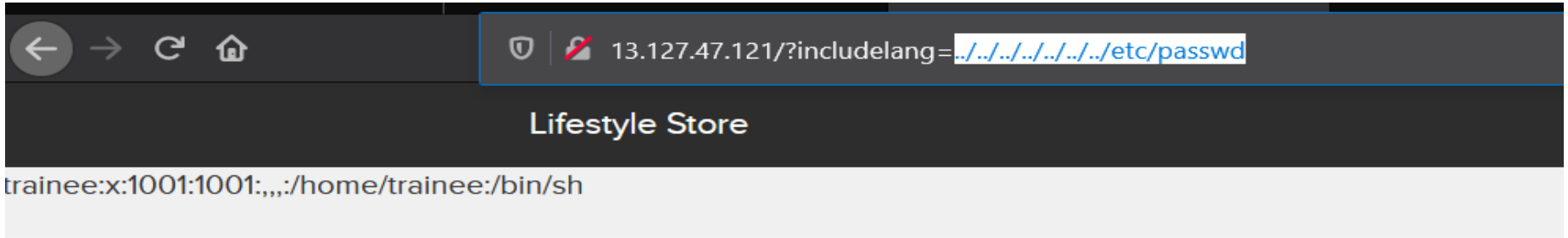
- https://cwe.mitre.org/data/definitions/601.html
- https://www.hacksplaining.com/prevention/open-redirects

12.File inclusion Vulnerabilities

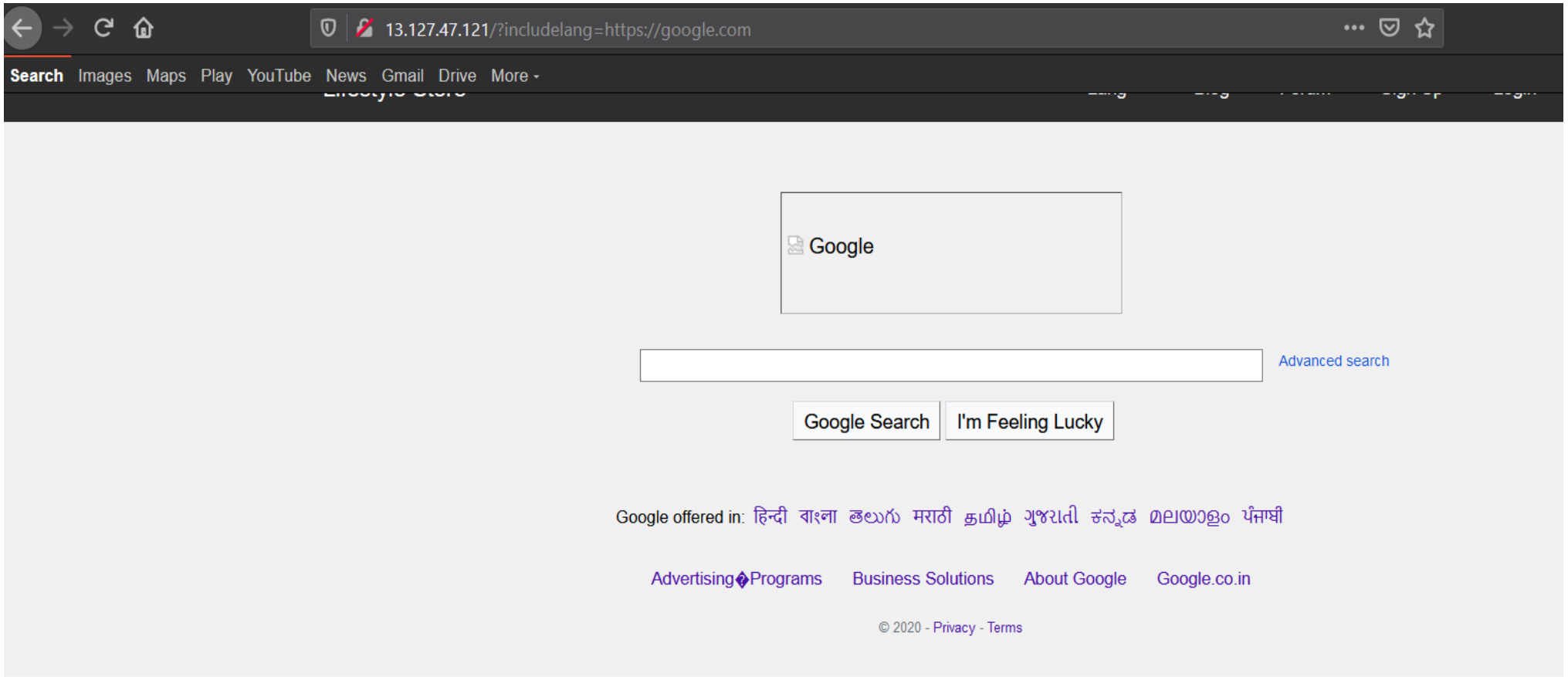
Local File Inclusion (Severe)	Below given URLS is vulnerable to Local file inclusion and Remote file inclusion **Affected URL:** 1. http://13.127.47.121/?includelang= (Local File Inclusion) 2. http://13.127.47.121/?includelang= (Remote File Inclusion) **Affected Parameters:** • lang (GET parameters) • lang (GET parameters) **Payload:** • ../../../../../../../etc/passwd • https://google.com

Observation (LFI)

Due to URL manipulation we are able to read from /etc/passwd which should not be allowed



Proof of Concept (PoC) (RFI)



Business Impact – High

- Attacker will be able to read and execute files on the victim machine if the web server is misconfigured and running with high privileges the attacker may gain access to sensitive information.
- If the attacker is able to place code on the web server through other means, they may be able to execute arbitrary commands

Recommendation

- To safely parse user-supplied filenames it's much better to **maintain a whitelist of acceptable filenames** and use a corresponding identifier (not the actual name) to access the file. Any request containing an invalid identifier can then simply be rejected

References

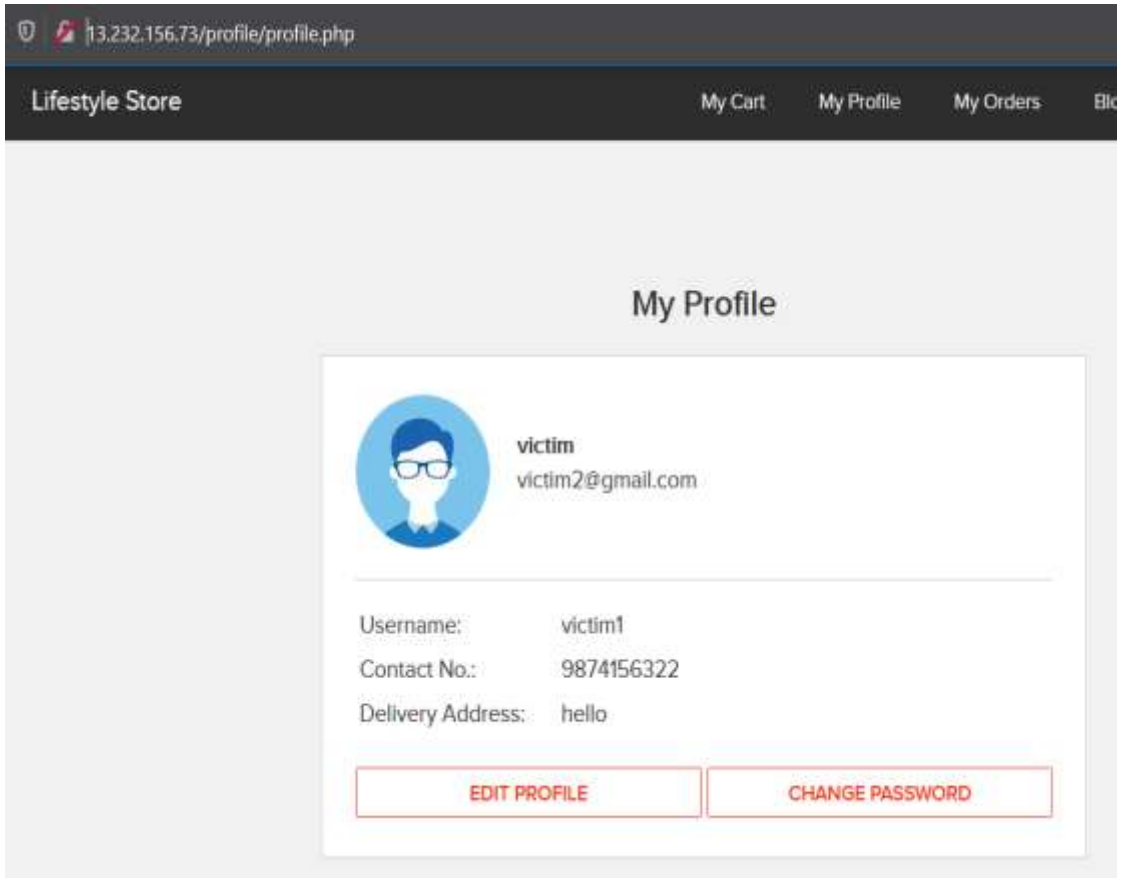
- https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_Local_File_Inclusion
- https://www.pivotpointsecurity.com/blog/file-inclusion-vulnerabilities/

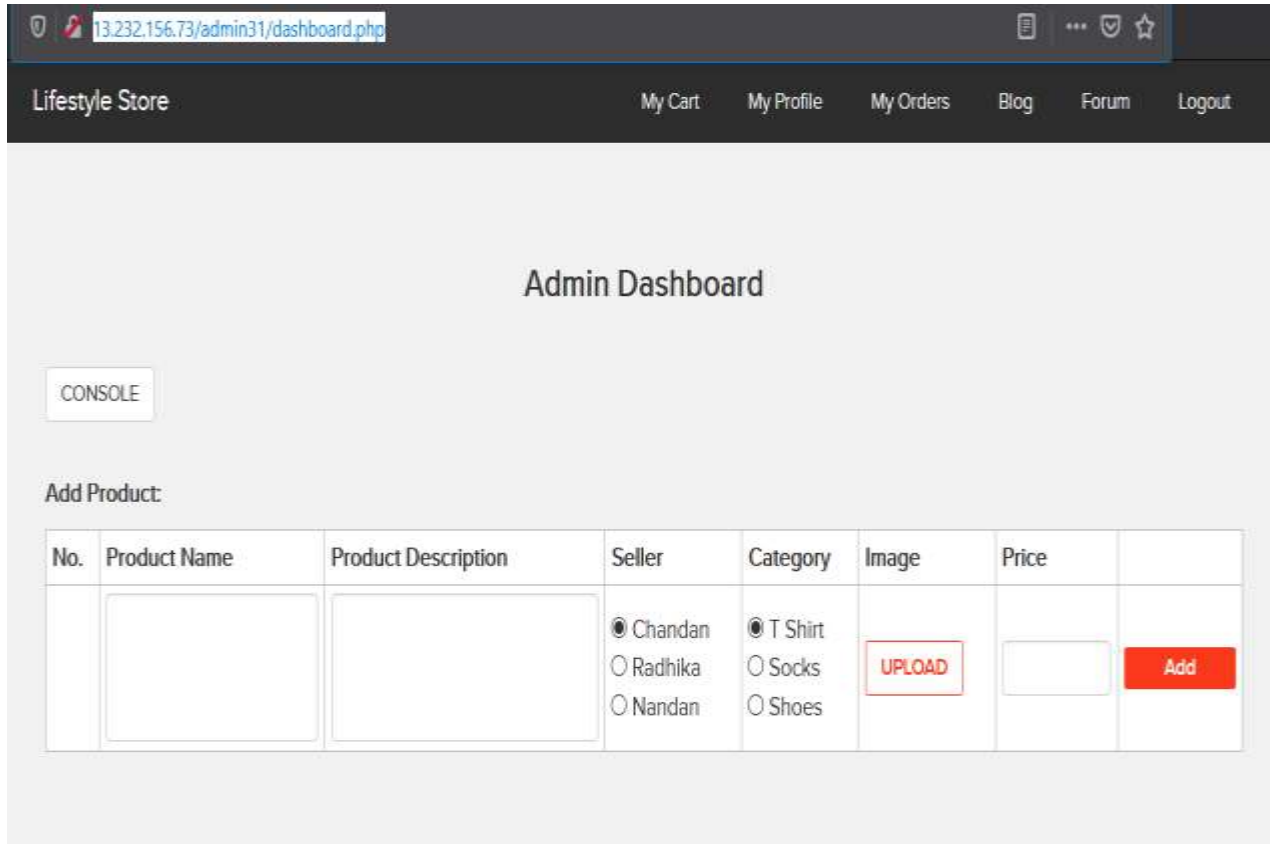
13.Forced Browsing Flaws

Forced Browsing Flaws (Severe)	Below given URL is force Browsing flaws **Affected URL:** • http://13.232.156.73/profile/profile.php **Affected Parameters:** • URL (GET parameter) **Payload:** • admin31/dashboard.php

Observation

- Login to the customer account and change the this URL http://13.232.156.73/profile/profile.php to this
- http://13.232.156.73/admin31/dashboard.php you will be able to access admin panel through customer profile





Business Impact – High

If an attacker is able access the admin panel through forced browsing He/she can gain access to any account and change the information about the products.. Attacker once logs in can then carry out actions on behalf of the admin which could lead to serious loss to any user

Recommendation

- Creating an allow list (or whitelist) involves allowing explicit access to a set of URLs that are considered to be a part of the application to exercise its functionality as intended. Any request not in this URL space is denied by default.
- using proper access control and authorization policies, access is only given to users commensurate with their privileges

References

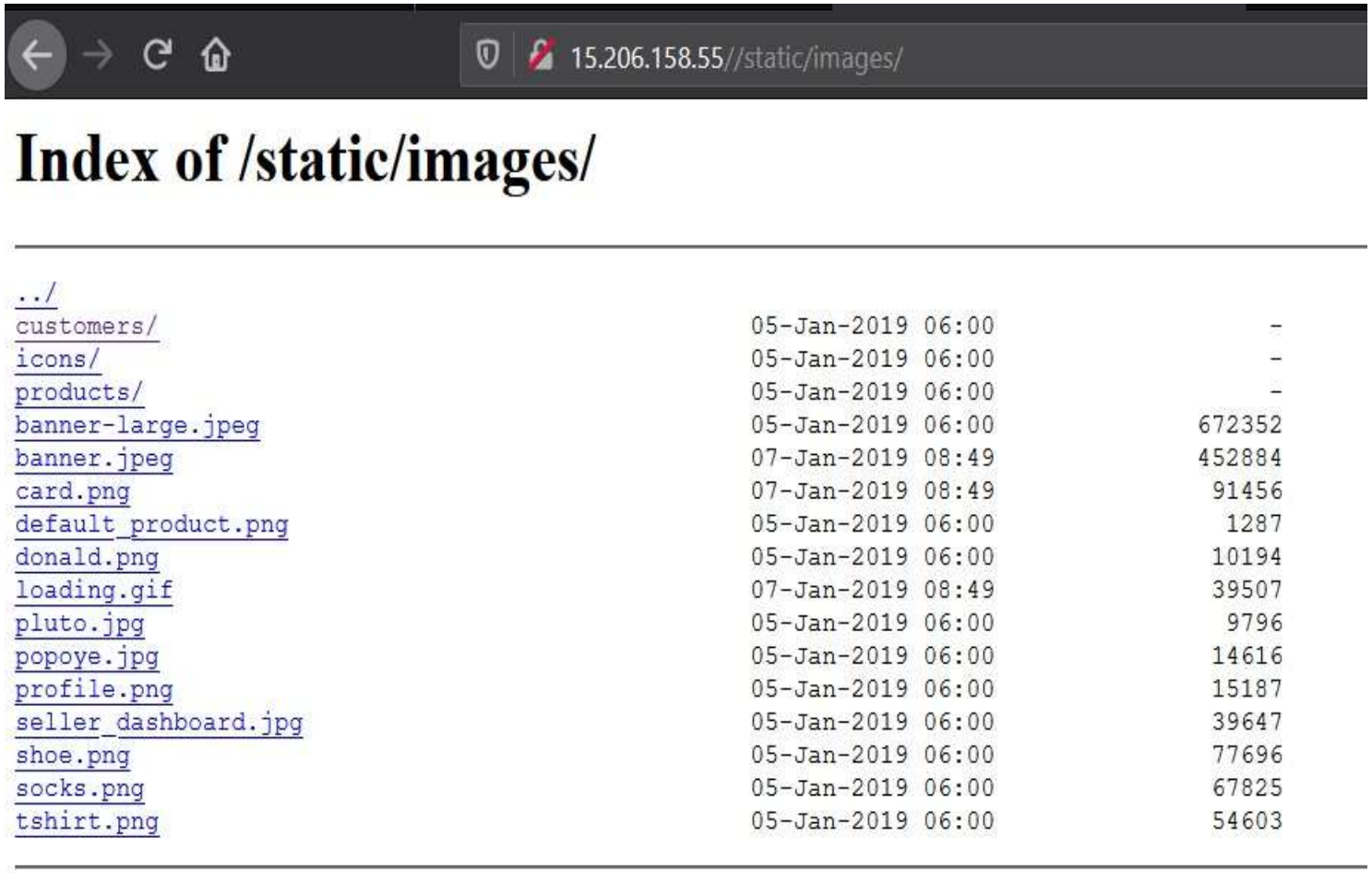
- https://owasp.org/www-community/attacks/Forced_browsing
- https://campus.barracuda.com/product/webapplicationfirewall/doc/42049348/forced-browsing-attack/

14.Directory Listing

Directory Listing (Severe)	**Affected URL :** • http://15.206.158.55/static/images/

Observation

Navigate to http://15.206.158.55/static/images and you will se the following page
You can navigate into any listed directory



Business Impact – Moderate

- Directory listings themselves do not necessarily constitute a security vulnerability Any sensitive resources within the web root should in any case be properly access-controlled, and should not be accessible by an unauthorized party who happens to know or guess the URL

Recommendation

- You can disable directory listing by creating an empty index file (index.php, index.html or any other extension your web server is configured to parse) in the relevant directory.
- Implement a permanent and secure solution by disabling directory listing at web server level

References

- https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/
- https://portswigger.net/kb/issues/00600100_directory-listing

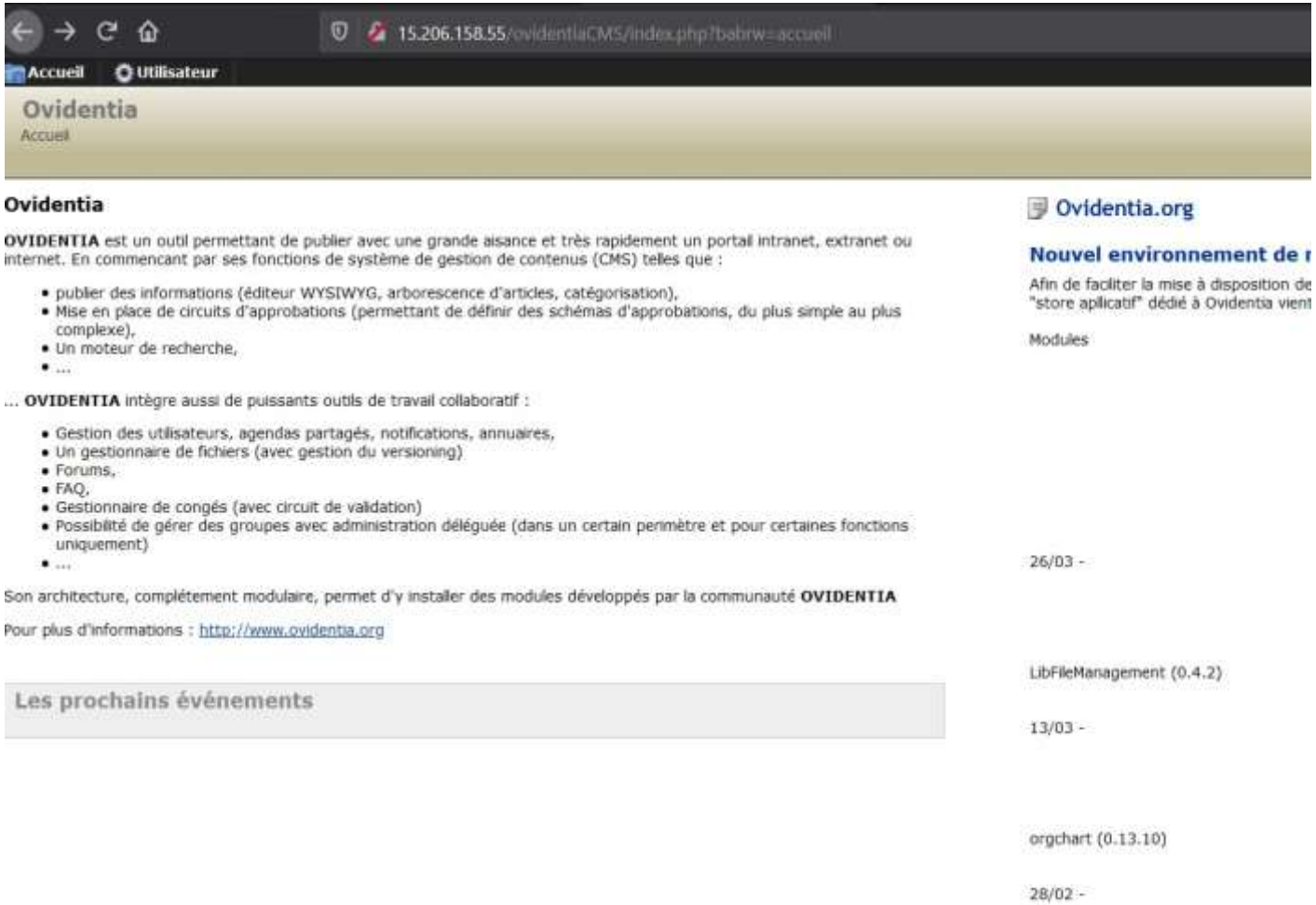
15.Default Files

Default Files (Moderate)	**Affected URL :** • http://15.206.158.55/robots.txt • http://15.206.158.55/users.txt • http://15.206.158.55/composer.json • http://15.206.158.55/composer.lock • http://15.206.158.55/phpinfo.php • http://15.206.158.55/server-status/

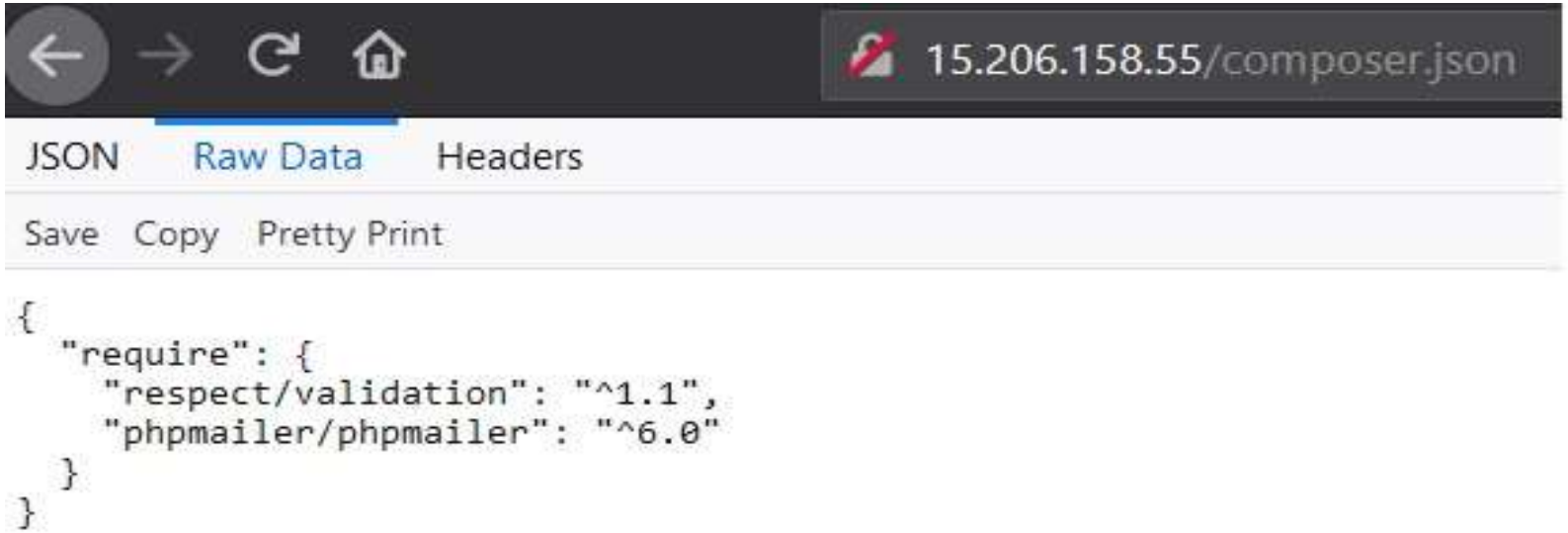
Observation

- Navigate to http://15.206.158.55/robots.txt and you will se the following page
- You can navigate into any listed pages

Proof of Concept (PoC)



Proof of Concept (PoC)



Proof of Concept (PoC)

15.206.158.55/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_.............
.......................................................................
......................
```

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	1709	0/1/1	_	0.92	17771	89	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET / HTTP/1.1
0-0	1709	0/1/1	[illegible]	9.64	34	[illegible]	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1

Proof of Concept (PoC)

15.206.158.55/phpinfo.php

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

System	Linux ip-172-26-12-84 5.3.0-1033-aws #35-Ubuntu SMP Wed Aug 5 15:47:17 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
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring

Business Impact – Moderate

- Although this vulnerability does not have a direct impact to users or the server, though it can aid the attacker with information about the server and the users. Information Disclosure due to default pages are not exploitable in most cases, but are considered as web application security issues because they allows malicious hackers to gather relevant information which can be used later in the attack lifecycle, in order to achieve more than they could if they didn't get access to such information
- Although there is leakage of sellers username and password

Recommendation

- Disable all default pages and folders including server-status and server-info.
- Multiple security checks enabled on important directories

References

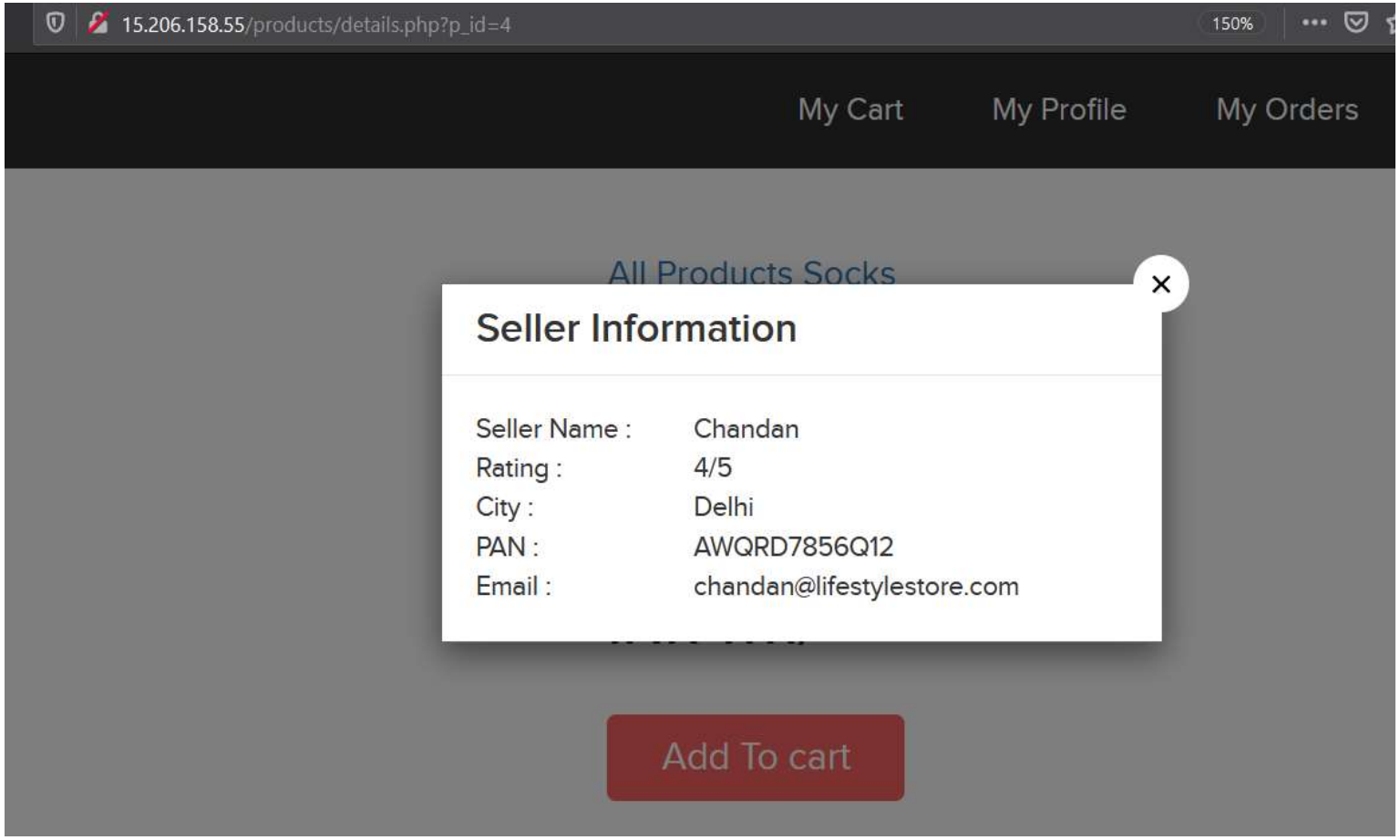
- https://www.netsparker.com/blog/web-security/information-disclosure-issues-attacks/
- https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/informationdisclosure-phpinfo/

16.Unnecessary Details about Sellers (PII)

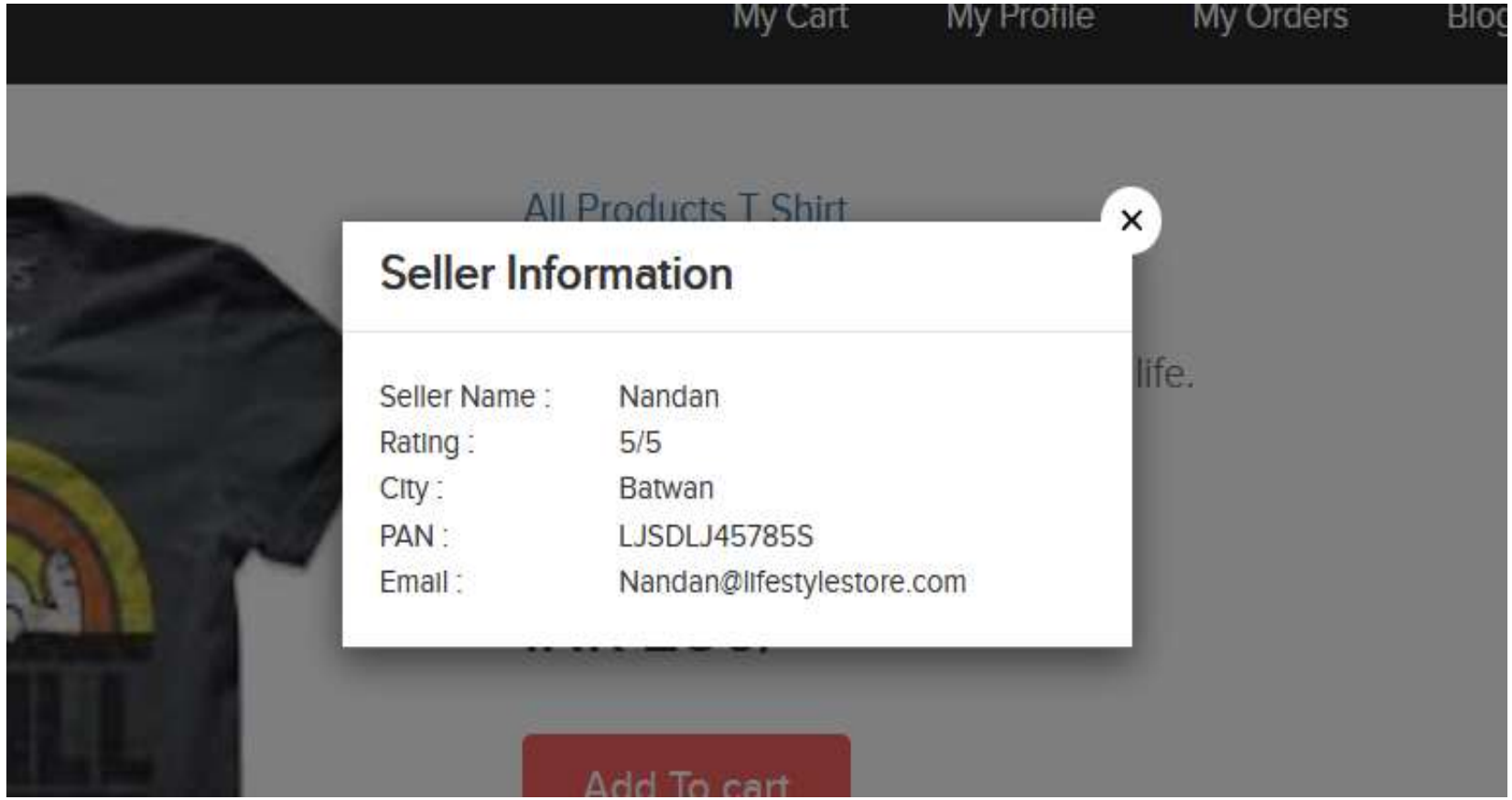
PII (MODERATE)	Below mentioned URL gives the unnecessary details about the seller (PII) **Affected URL :** • http://15.206.158.55/products/details.php?p_id=4 • http://15.206.158.55/products/details.php?p_id=5 • http://15.206.158.55/products/details.php?p_id=17

Observation

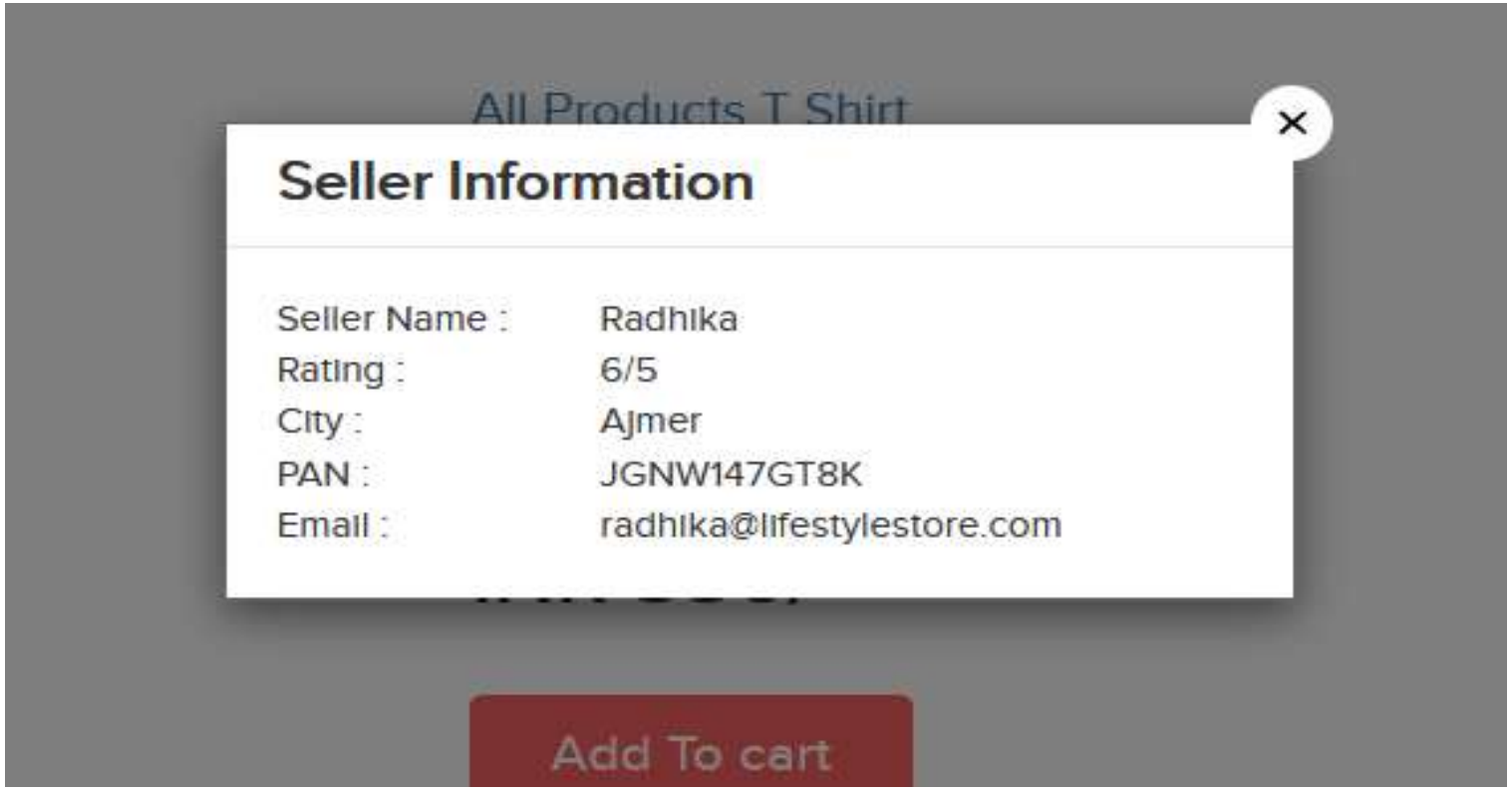
- When we click on the Seller Info option ,we get the details of the seller, even those which are not required like the pan number



Proof of Concept (PoC)



Proof of Concept (PoC)



Business Impact – Moderate

- There is no direct business impact in this case, but this amount of information can definitely lead to social engineering attacks on the seller and can indirectly harm the business
- Exposing PAN card number is harmful

Recommendation

- Only name and email is sufficient as far as the query or help is concerned.

References

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

17.Components with known vulnerabilities

Components with known Vulnerabilities (MODERATE)	• Server used is nginx/1.14.0 appears to be outdated (current is at least 1.19) i.e it is known to have exploitable vulnerabilities. • Wonder CMS 2.3.1 • Codoforum 2015 Codologic

Observation

- wondercms is a outdated and vulnerable, which also leaded to upload shells.
- The nginx/1.14.0 version is also highly vulnerable

Wondercms » Wondercms » 2.3.1 : Security Vulnerabilities

Cpe Name:*cpe:/a:wondercms:wondercms:2.3.1*

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2017-14523	74			2018-01-26	2019-04-30	5.0	None	Remote	Low	Not required	None	Partial	None
** DISPUTED ** WonderCMS 2.3.1 is vulnerable to an HTTP Host header injection attack. It uses user-entered values to redirect pages. NOTE: the vendor reports that exploitation is unlikely because the attack can only come from a local machine or from the administrator as a self attack.														
2	CVE-2017-14522	79		XSS	2018-01-26	2018-02-14	4.3	None	Remote	Medium	Not required	None	Partial	None
** DISPUTED ** In WonderCMS 2.3.1, the application's input fields accept arbitrary user input resulting in execution of malicious JavaScript. NOTE: the vendor disputes this issue stating that this is a feature that enables only a logged in administrator to write execute JavaScript anywhere on their website.														
3	CVE-2017-14521	434			2018-01-26	2019-04-26	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
In WonderCMS 2.3.1, the upload functionality accepts random application extensions and leads to malicious File Upload.														

Total number of vulnerabilities : **3** Page : 1 (This Page)

Proof of Concept (PoC)

15.206.158.55/h

404 Not Found

nginx/1.14.0 (Ubuntu)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2018-16845	835			2018-11-07	2019-10-02	5.8	None	Remote	Medium	Not required	Partial	None	Partial
nginx before versions 1.15.6, 1.14.1 has a vulnerability in the ngx_http_mp4_module, which might allow an attacker to cause infinite loop in a worker process, cause a worker process crash, or might result in worker process memory disclosure by using a specially crafted mp4 file. The issue only affects nginx if it is built with the ngx_http_mp4_module (the module is not built by default) and the .mp4. directive is used in the configuration file. Further, the attack is only possible if an attacker is able to trigger processing of a specially crafted mp4 file with the ngx_http_mp4_module.														
2	CVE-2018-16844	400			2018-11-07	2019-09-10	7.8	None	Remote	Low	Not required	None	None	Complete
nginx before versions 1.15.6 and 1.14.1 has a vulnerability in the implementation of HTTP/2 that can allow for excessive CPU usage. This issue affects nginx compiled with the ngx_http_v2_module (not compiled by default) if the 'http2' option of the 'listen' directive is used in a configuration file.														
3	CVE-2018-16843	400			2018-11-07	2019-09-10	7.8	None	Remote	Low	Not required	None	None	Complete
nginx before versions 1.15.6 and 1.14.1 has a vulnerability in the implementation of HTTP/2 that can allow for excessive memory consumption. This issue affects nginx compiled with the ngx_http_v2_module (not compiled by default) if the 'http2' option of the 'listen' directive is used in a configuration file.														

Proof of Concept (PoC)

© 2015 CODOLOGIC
Powered by Codoforum

Codologic : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2013-5952	79		XSS	2014-03-19	2017-08-28	4.3	None	Remote	Medium	Not required	None	Partial	None

Multiple cross-site scripting (XSS) vulnerabilities in the Freichat (com_freichat) component, possibly 9.4 and earlier, for Joomla! allow remote attackers to inject arbitrary web script or HTML via the (1) id or (2) xhash parameter to client/chat.php or (3) toname parameter to client/plugins/upload/upload.php.

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2014-9261	22	1	Dir. Trav.	2015-03-23	2015-03-24	5.0	None	Remote	Low	Not required	Partial	None	None

The sanitize function in Codoforum 2.5.1 does not properly implement filtering for directory traversal sequences, which allows remote attackers to read arbitrary files via a .. (dot dot) in the path parameter to index.php.

Total number of vulnerabilities : **1** Page : 1 (This Page)

Business Impact – HIGH

- Exploits of every vulnerability detected is regularly made public and hence outdated software can very easily be taken advantage of. If the attacker comes to know about this vulnerability ,he may directly use the exploit to take down the entire system, which is a big risk

Recommendation

- Upgrade to the latest version of Affected Software/theme/plugin/OS which means latest version.
- If upgrade is not possible for the time being, isolate the server from any other critical data and server

References

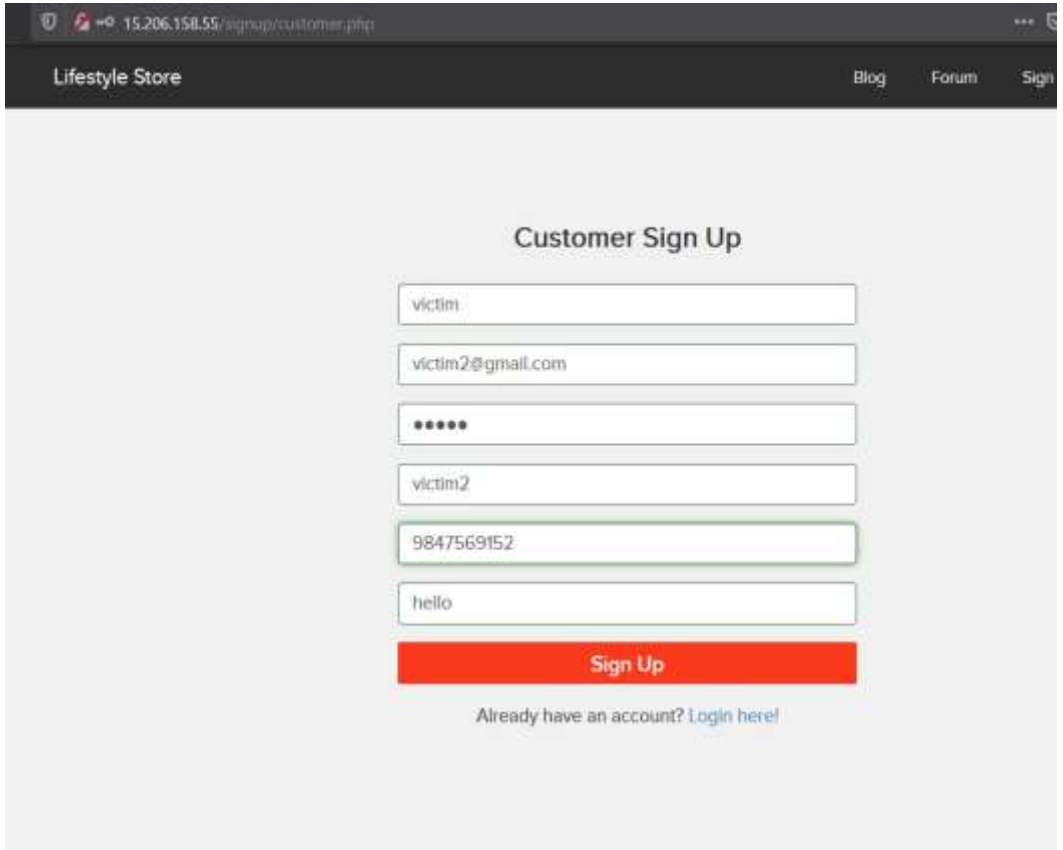
- http://securitywarrior9.blogspot.com/2018/01/vulnerability-in-wonder-cms-leading-to.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-10048/product_id-17956/version_id-267430/Nginx-Nginx-1.14.0.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-15315/product_id-31335/Codoforum-Codoforum.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-13192/Codologic.html

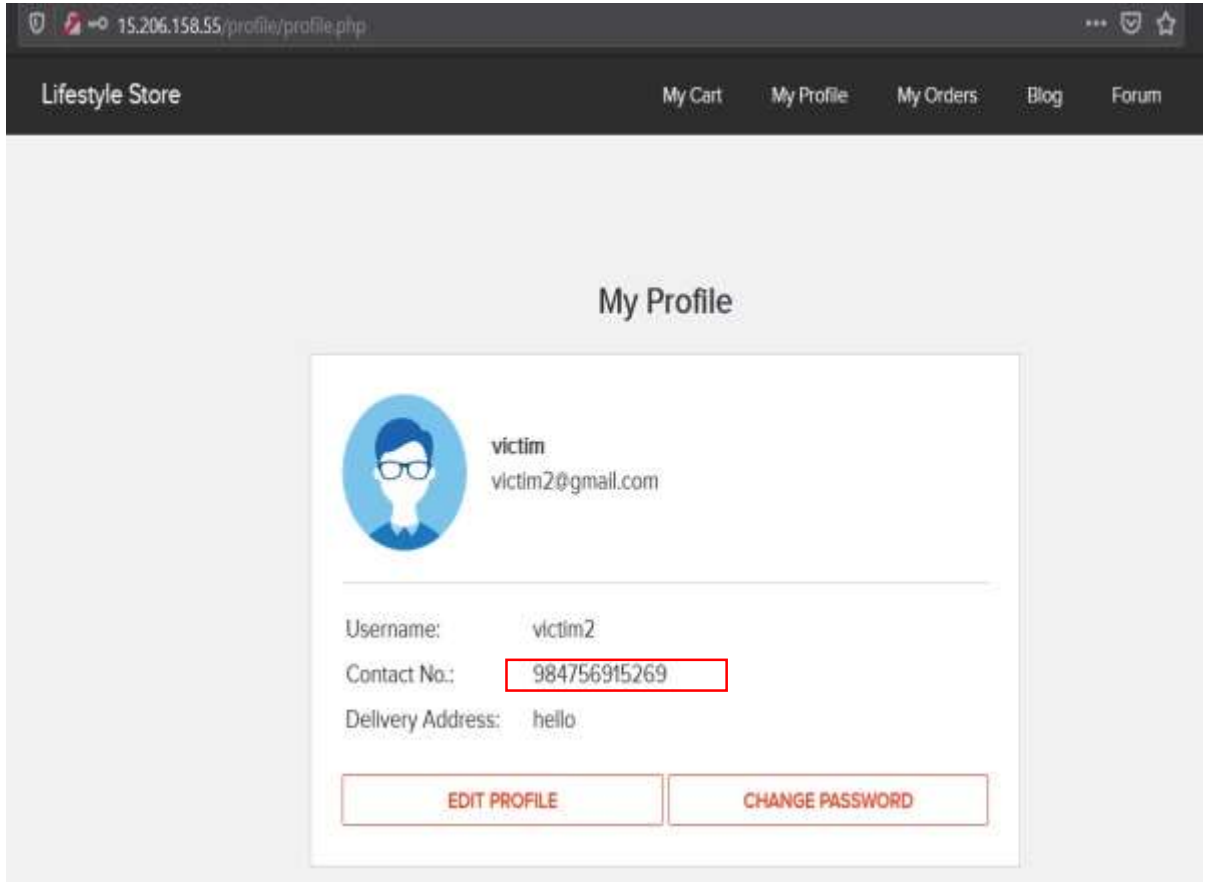
18. Client Side Filter Bypass

Improper Server Side and Client-Side Filters (LOW)	Below mentioned URLs are vulnerable to client-side filter bypass **Affected URL :** • http://15.206.158.55/signup/customer.php • http://15.206.158.55/profile/17/edit/ **Affected parameter:** • Contact Number (POST Parameter) **Payload used:** 984756915269

Observation

- Make an account and input only 10 values in the phone number field and capture it via burp suite
- So it gets validated by client-side and because the server-side doesn't check we can bypass it and input more the 10 values





Business Impact - Low

- The data provided by the user ,if incorrect, is not a very big issue but still must be checked for proper validatory information.

Recommendation

- Implement all critical checks on server side code only.
- All business logic must be implemented and checked on the server code. This includes user input, the flow of applications and even the URL/Modules a user is supposed to access or no

References

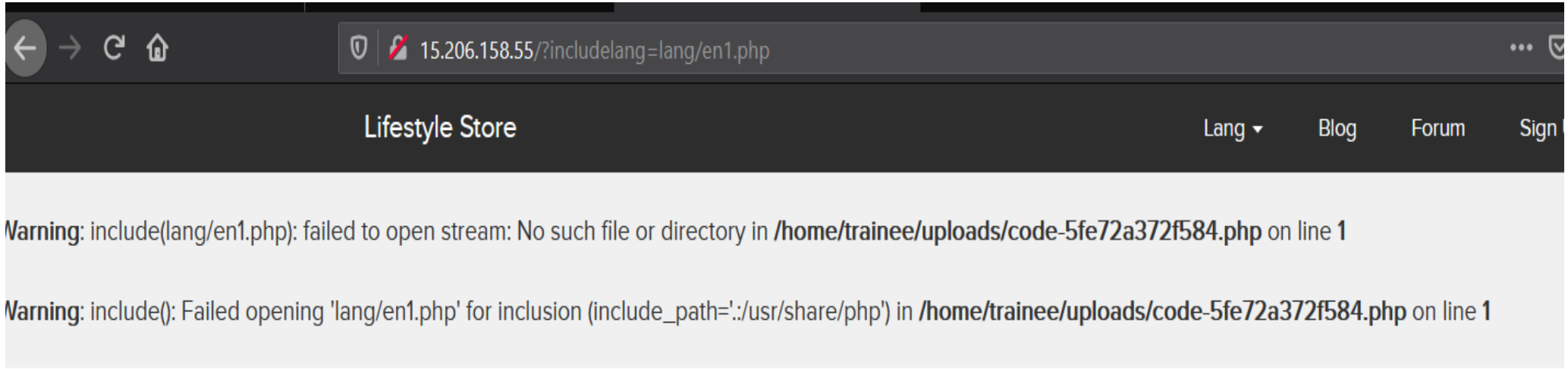
- http://projects.webappsec.org/w/page/13246933/Improper%20Input%20Handling
- https://www.owasp.org/index.php/Unvalidated_Input

19.Default Error Display

Default Error Display (LOW)	Below mentioned url have default error displaying : **Affected URL :** • http://15.206.158.55/?includelang=lang/en.php • http://15.206.158.55/?includelang=lang/en.php **Affected parameter:** • en.php (GET Parameter) • includeLang (GET parameter) **Payload used:** • en1.php • https://www.hacker.com

Observation

- The default error with the path is displayed as



Proof of Concept (PoC)

Lifestyle Store

Lang ▾ Blog Forum Sign Up Login ▾

Warning: include(lang/https://www.hacker.com): failed to open stream: No such file or directory in **/home/trainee/uploads/code-5fe9755b8a5fa.php** on line **1**

Warning: include(): Failed opening 'lang/https://www.hacker.com' for inclusion (include_path='.:/usr/share/php') in **/home/trainee/uploads/code-5fe9755b8a5fa.php** on line **1**

Business Impact - Low

Although this vulnerability does not have a direct impact to users or the server, though it can help the attacker in mapping the server architecture and plan further attacks on the server.

Recommendation

Do not display the default error messages because it not tells about the server but also sometimes about the location. So, whenever there is an error ,send it to the same page or throw some manually written error.

References

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

THANK YOU

For any further clarifications/patch assistance, please contact:
7306654794