



# **SUMMER TRAINING** **REPORT**

on

**Ethical Hacking**  
**Duration: 8 Weeks**

**Submitted by:**

**Name:** Harsh Sonker

**RollNo.:** 200014045005

**Year/Sem:** 3<sup>rd</sup>/5<sup>th</sup>

**Submitted to:**

Er. Rohit,  
Er. Shailendra Kumar Sonkar

**Faculty of Engineering & Technology,**  
**University of Lucknow.**  
**Session: 2022-23**

## **DECLARATION**

I hereby declare that this project based on **ethical hacking** has been carried out by my own efforts and facts arrived at by my own observations. I am hence submitting this project to my college and I also promise that this project has not been submitted in any university before. As this is my original work.

Name: **Harsh Sonker**

Date: **30th Aug, 2022**

# **ACKNOWLEDGEMENT**

First and Foremost, I would like to express my sincere gratitude and thanks towards University of Lucknow for introducing a course like BCA and given all the students a base and a platform to keep abreast with changing technical scenario. I would like to express my profound gratitude and sincere thanks to my Project Guide Er.Rohit/ Er. Shailendra Kumar Sonkar, Faculty for instilling confidence in me to carry out this report. I sincerely acknowledge of both professors for extending their valuable guidance, support and critical reviews towards this report and above all for the moral support they provided me at all stages.

Submitted by  
**Harsh Sonker**  
**(200014045005)**

# CERTIFICATE

INTERNSHALA TRAININGS


## Certificate of Training

**Harsh Sonker**

from University of Lucknow has successfully completed an 8-week online training on **Ethical Hacking**. In the training, Harsh learned Basics of Information Security, Computer Networking and Web Development, Information Gathering and VAPT of some important vulnerabilities in the OWASP top 10, Automating VAPT, and Documenting and Reporting Vulnerabilities.

In the final assessment, Harsh scored 77% marks.

We wish Harsh all the best for future endeavours.



**Sarvesh Agarwal**

FOUNDER & CEO, INTERNSHALA

Date of certification: 2022-08-18

Certificate no. : 806E5B9F-BC14-2C05-BD2B-9FD8554E711A

For certificate authentication, please visit [https://trainings.internshala.com/verify\\_certificate](https://trainings.internshala.com/verify_certificate)

# **TABLE OF CONTENT**

<b>List of figures</b>	.....	7
<b>Abstract</b>	.....	8
<b>CHAPTER: 1</b>	<b>INTRODUCTION TO ETHICAL HACKING.....</b>	9-12
<b>1.1</b>	History of Ethical hacking.....	10
<b>1.2</b>	Top cyber security organizations.....	11
<b>1.3</b>	Cyber security tools.....	12
<b>CHAPTER: 2</b>	<b>CYBERSECURITY AWARENESS AND INNOVATION.....</b>	13
<b>2.1</b>	Digital identity.....	13
<b>2.2</b>	Credentials.....	13
<b>CHAPTER:3</b>	<b>INTRODUCTION TO CYBER SECURITY.....</b>	14-16
<b>3.1</b>	What is Hacking?.....	14
<b>3.2</b>	Types Of Hackers.....	15
<b>3.3</b>	Types of security testing.....	16
<b>CHAPTER: 4</b>	<b>INTRODUCTION TO PROJECT.....</b>	17-20
<b>4.1</b>	Objective.....	17
<b>4.2</b>	Required screenshots.....	18-20
<b>CHAPTER: 5</b>	<b>DEVELOPER'S REPORT E-COMMERCE WEBSITE LIFESTYLE STORE DETAILED PROJECT REPORT.....</b>	21-41
<b>5.1</b>	Vulnerabilities.....	22
<b>5.1.1</b>	SQL injections.....	23-25
<b>5.1.2</b>	Access to admin panel.....	26-27
<b>5.1.3</b>	Arbitrary file upload.....	28-29
<b>5.1.4</b>	Account takeover using OTP bypass.....	30-31
<b>5.1.5</b>	CSRF.....	32-33
<b>5.1.6</b>	Reflected Cross Site Scripting (XSS).....	34-35
<b>5.1.7</b>	Stored Cross Site Scripting (XSS).....	36-37
<b>5.1.8</b>	Common Password.....	38
<b>5.1.9</b>	Component with known vulnerability.....	39-40
<b>5.1.10</b>	Server misconfiguration.....	41

<b>Conclusion</b>	.....	42
<b>Reference</b>	.....	43

# **LIST OF FIGURES**

Fig. 4.1	Screenshot of E-Commerce Website.....	19
Fig. 4.1	Screenshot of E-Commerce Website.....	19
Fig. 4.1	Screenshot of E-Commerce Website.....	20
Fig. 4.1	Screenshot of E-Commerce Website.....	20
Fig. 4.1	Screenshot of E-Commerce Website.....	20
Fig. 4.1	Vulnerability Statistic.....	21
Fig. 4.1	List of Vulnerabilities.....	22
Fig. 4.1	Screenshot of SQL injection.....	23
Fig. 4.1	Screenshot of SQL injection.....	23
Fig. 4.1	Observation.....	23
Fig. 4.1	Screenshot of SQL injection.....	24
Fig. 4.1	POC.....	24
Fig. 4.1	Access to admin panel.....	26
Fig. 4.1	Observation.....	26
Fig. 4.1	POC.....	26
Fig. 4.1	Arbitrary file upload.....	28
Fig. 4.1	Observation.....	28
Fig. 4.1	Arbitrary file upload.....	30
Fig. 4.1	Observation.....	30
Fig. 4.1	Brute force attack.....	30
Fig. 4.1	OTP found.....	30
Fig. 4.1	POC.....	31
Fig. 4.1	Observation.....	32
Fig. 4.1	Observation.....	32
Fig. 4.1	Observation.....	32
Fig. 4.1	POC.....	32
Fig. 4.1	Observation.....	33
Fig. 4.1	Observation.....	33
Fig. 4.1	Reflected Cross Site Scripting (XSS).....	34
Fig. 4.1	Observation.....	34
Fig. 4.1	POC.....	34
Fig. 4.1	Stored Cross Site Scripting (XSS) .....	36
Fig. 4.1	Observation.....	36
Fig. 4.1	Observation.....	36
Fig. 4.1	Common Password.....	38
Fig. 4.1	Observation.....	38
Fig. 4.1	Component with known vulnerability.....	39
Fig. 4.1	Observation.....	39
Fig. 4.1	POC.....	39
Fig. 4.1	Server misconfiguration.....	41
Fig. 4.1	Observation and POC.....	41

## **ABSTRACT**

This paper explores the ethics behind ethical hacking and whether there are problems that lie with this new field of work. Since ethical hacking has been a controversial subject over the past few years, the question remains of the true intentions of ethical hackers. The paper also looks at ways in which future research could be looked into to help keep ethical hacking, ethical.



# **CHAPTER: 1**

## **INTRODUCTION TO ETHICAL HACKING**

Ethical Hacking tutorial provides basic and advanced concepts of Ethical Hacking. Our Ethical Hacking tutorial is developed for beginners and professionals.

Ethical hacking tutorial covers all the aspects associated with hacking. Firstly, we will learn how to install the needed software. After this, we will learn the 4 type of penetration testing section which is network hacking, gaining access, post exploitation, website hacking.

In network hacking section, we will learn how networks work, how to crack Wi-Fi keys and gain access the Wi-Fi networks. In Gaining access section, we will learn how to gain access to the servers and personal computers. In the post-exploitation section, we will learn what can we do with the access that we gained in the previous section. So we learn how to interact with the file system, how to execute a system command, how to open the webcam. In the website hacking section, we will learn how the website works, how to gather comprehensive information about website. In the end, we will learn how to secure our system from the discussed attacks.

## **1.1 History of Ethical hacking**

1. The past history of ethical hacking narrates to us that it wasn't always bad to be a hacker. In fact, as per the history of ethical hacking, the word surfaced in its modern context at the renowned Massachusetts Institute of Technology (MIT). The phrase "ethical hacking" was first used in 1995 by IBM Vice President John Patrick, but the concept has been around for a lot longer.
2. In the 1980s and the 1990s, personal computers gained massive popularity. A major part of the personal information and other confidential records were stored in the form of computer programs. This created a spark in the minds of the hackers to try gaining access to these systems. This information was then sold for a huge profit.
3. Hackers used to be viewed as people who sat locked in a room all day programming nonstop, hours on end. No one seemed to mind hackers back in the 1960s when this was the most widely excepted reputation. In fact, most people had no idea what hacking was.
4. Hacking was a gaining profile in the media – and not a positive one. Hackers were seen as criminals – digital trespassers – who were using their skills to gain access to private computers, steal data and even blackmail businesses into handing over large sums of money. These kinds of hackers are what we describe today as black hat hackers.

## 1.2 Top cyber security organizations

1. **McAfee Secure:** McAfee, LLC is an American global computer security software company headquartered in Santa Clara, California and claims to be the world's largest dedicated technology security company. The company was purchased by Intel in February 2011, and became part of the Intel Security division.
2. **FireEye:** FireEye is a publicly traded cybersecurity company headquartered in Milpitas, California. It has been involved in the detection and prevention of major cyber attacks. It provides hardware, software, and services to investigate cybersecurity attacks, protect against malicious software, and analyze IT security risks.
3. **OWASP:** The Open Web Application Security Project is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.
4. **Kaspersky:** Kaspersky Lab is a multinational cybersecurity and anti-virus provider headquartered in Moscow, Russia and operated by a holding company in the United Kingdom. It was founded in 1997 by Eugene Kaspersky, Natalya Kaspersky, and Alexey De-Monderik; Eugene Kaspersky is currently the CEO.
5. **ISACA:** ISACA is an international professional association focused on IT governance. On its IRS filings, it is known as the Information Systems Audit and Control Association, although ISACA now goes by its acronym only

## 1.3 Cyber security tools

1. **Burp suite pro:** Burp Suite Professional is an advanced set of tools for testing web security - all within a single product. From a basic intercepting proxy to a cutting-edge vulnerability scanner, with Burp Suite Pro, the right tool is never more than a click away. Burp Suite Pro is built by a research-led team.
2. **Acunetix:** Acunetix is an automated web application security testing tool that audits your web applications by checking for vulnerabilities like SQL Injection, Cross site scripting and other exploitable vulnerabilities.
3. **Nikto:** Nikto is a free software command-line vulnerability scanner that scans webservers for dangerous files/CGIs, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received.
4. **OWASP ZAP:** It is an open-source web application security scanner. It is intended to be used by both those new to application security as well as professional penetration testers. It is one of the most active Open Web Application Security Project projects and has been given Flagship status.
5. **Wireshark:** Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

# **CHAPTER: 2**

## **CYBERSECURITY AWARENESS AND INNOVATION**

### **2.1 Digital identity**

A digital identity is information on an entity used by computer systems to represent an external agent. That agent may be a person, organization, application, or device. ISO/IEC 24760-1 defines identity as "set of attributes related to an entity".

#### **➤ Digital identity creation**

A digital identity arises organically from the use of personal information on the web and from the shadow data created by the individual's actions online. Examples of data points that can help form a digital identity include: Username and password.

### **2.2 Credentials**

A credential is a piece of any document that details a qualification, competence, or authority issued to an individual by a third party with a relevant or de facto authority or assumed competence to do so.

Examples of credentials include academic diplomas, academic degrees, certifications, security clearances, identification documents, badges, passwords, user names, keys, powers of attorney, and so on.

## **CHAPTER: 3**

### **Introduction to Cyber Security**

#### **3.1 What is Hacking?**

Hacking is the art or technique of finding and exploiting a security loopholes in an infrastructure like website, software, computer or even a human being and the artist is called hacker.

##### **➤ Loopholes**

A loophole can be referred to a part of a system which is not properly secured and hence can exploited to cause unintended things in the system.

##### **➤ Unethical Hacking**

An unethical hack is one that is done without the target of the hack being aware of it. It is often done to break into a network system to steal information or money, and sometimes to cause damage by inserting a virus or malware program.

##### **➤ Ethical hacking**

Ethical Hacking is an authorized practice of bypassing system security to identify potential data breaches and threats in a network. Ethical hackers aim to investigate the system or network for weak points that malicious hackers can exploit or destroy.

## **3.2 Types Of Hackers**

- 1.White hat hackers
- 2.Black hat hackers
- 3.Grey hat hackers

### **1.White hat hackers**

The term "white hat" in Internet slang refers to an ethical computer hacker, or a computer security expert, who specializes in penetration testing and in other testing methodologies that ensures the security of an organization's information systems.

### **2.Black hat hackers**

Black Hat hackers are criminals who break into computer networks with malicious intent. They may also release malware that destroys files or steals passwords for personal gains.

### **3.Grey hat hackers**

A grey hat is a computer hacker or computer security expert who may sometimes violate laws or typical ethical standards, but does not have the malicious intent typical of a black hat hacker.

### 3.3 Types of security testing

- a) **White box testing:** White-box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. In this type of testing security specialist gets complete assistance from the organization.
  
- b) **Black box testing:** Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. In this type testing expert is given no assistance from organization.
  
- c) **Grey box testing:** Gray-box testing is a combination of white-box testing and black-box testing. The aim of this testing is to search for the defects if any duezz to improper structure or improper usage of applications. In this type of testing expert gets partial assistance from the organization.



## **CHAPTER: 4**

### **INTRODUCTION TO PROJECT**

In this course, we were given videos to learn from. There were a total of 9 chapters in the course and each chapter was divided into modules. Each module had a small test after its completion and after each chapter, we had to attempt a test. Without which we won't be able to move to the next chapter. I made written notes while watching all the videos which later helped me, where needed. Doing this also helped me preserve this knowledge with myself forever.

At the end of the course, we were given a problem statement. Based on which, we had to create the project.

Internshala also provided us with hacking labs for real life experience and for project competition, we were provided a website we had to work on.

## 4.1 Objective

We are glad that you have completed the training and cleared the final test. Now, it's time to test your skills in a practical manner and for that, we have setup a real life-like web application in the form of an online e-commerce portal.

Your task is to test this e-commerce platform and find all possible vulnerabilities and loopholes in it, collect relevant PoCs and then prepare a Detailed Developer Level Report.

For reporting each vulnerability, you must follow the sample report given to you in Module 8 and make sure the following things are mentioned:

- Title of Vulnerability.
- A Short Description.
- Exact URL which has the vulnerability.
- The parameters which are vulnerable (with parameter type like GET, POST, Cookie, Header, etc.).
- Payload that you used to trigger the vulnerability.
- Observation slides containing step by step information to replicate the exploit with PoCs.
- Business Impact of the vulnerability, explaining in detail what can be done by a hacker.
- Recommendations on how to fix the vulnerability.
- Reputed References for the vulnerabilities.

Remember, each and every kind of vulnerability you learnt about, might be somewhere in this application. All you have to do is open the application and start exploring its features. Once you have understood each feature the website has, you can start playing around with it and fuzzing into various places.

A big part of the VA has been already done for you as you have the exact IP and the application which you have to test, but there could be hidden pages and components too, so keep that in mind.

To give you a benchmark and a target to achieve, here is a list of all the vulnerabilities which we have intentionally kept and which are supposed to be found and reported by you:

SQL Injection

Reflected and Stored Cross Site Scripting

Insecure Direct Object Reference

Rate Limiting Issues

Insecure File Uploads

Client Side Filter Bypass

Server Misconfigurations

Components with Known Vulnerabilities

Weak Passwords

Default Files and Pages

File Inclusion Vulnerabilities

PII Leakage

Open Redirection

Bruteforce Exploitation

Command Execution Vulnerability

Forced Browsing Flaws

Cross-Site Request Forgery

So, there are a total of 28 vulnerabilities (some vulnerabilities have more occurrences than 1) intentionally kept but these do not include combinational vulnerabilities like Bruteforce Exploitation and Rate Limiting. If you are able to guess the password, you can either count it in Bruteforcing or count it in rate limiting but yes, while writing recommendations, write recommendations for both. Similarly, if you find a public software that allows PHP file upload, you can either count it in file upload or in Components with known vulnerabilities.

If you do find other general vulnerabilities apart from these you can report them too but do not count them in the 28.

Happy bug hunting!

### Steps to access the Project:

1. Login to [trainings.internshala.com](https://trainings.internshala.com)
2. Go to Ethical Hacking Training
3. Go to Progress Tracker
4. Click on the 'GO TO PROJECT WEB APPLICATION' button.

## 4.2 Required screenshots

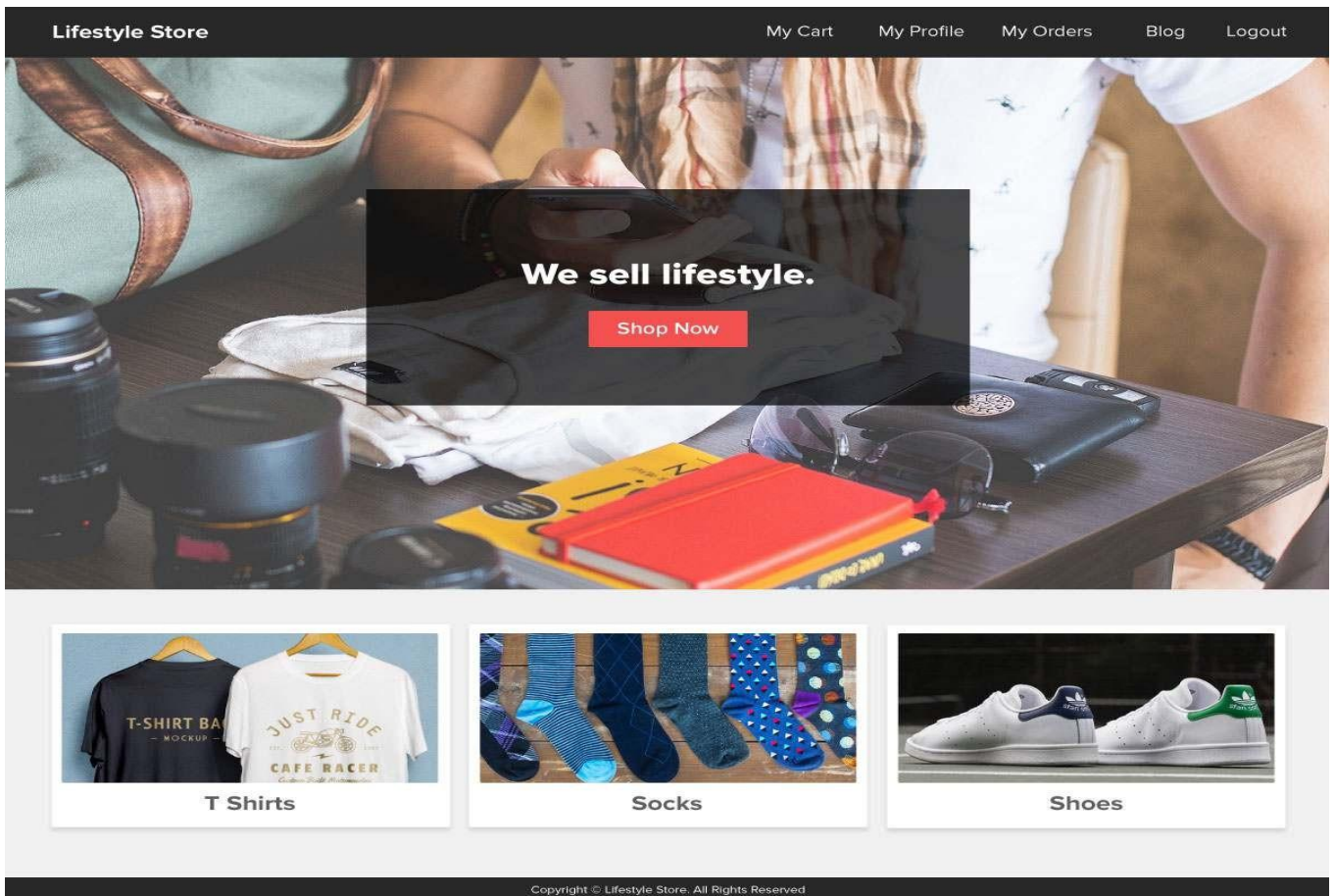


Fig. 4.1 Screenshot of E-Commerce Website

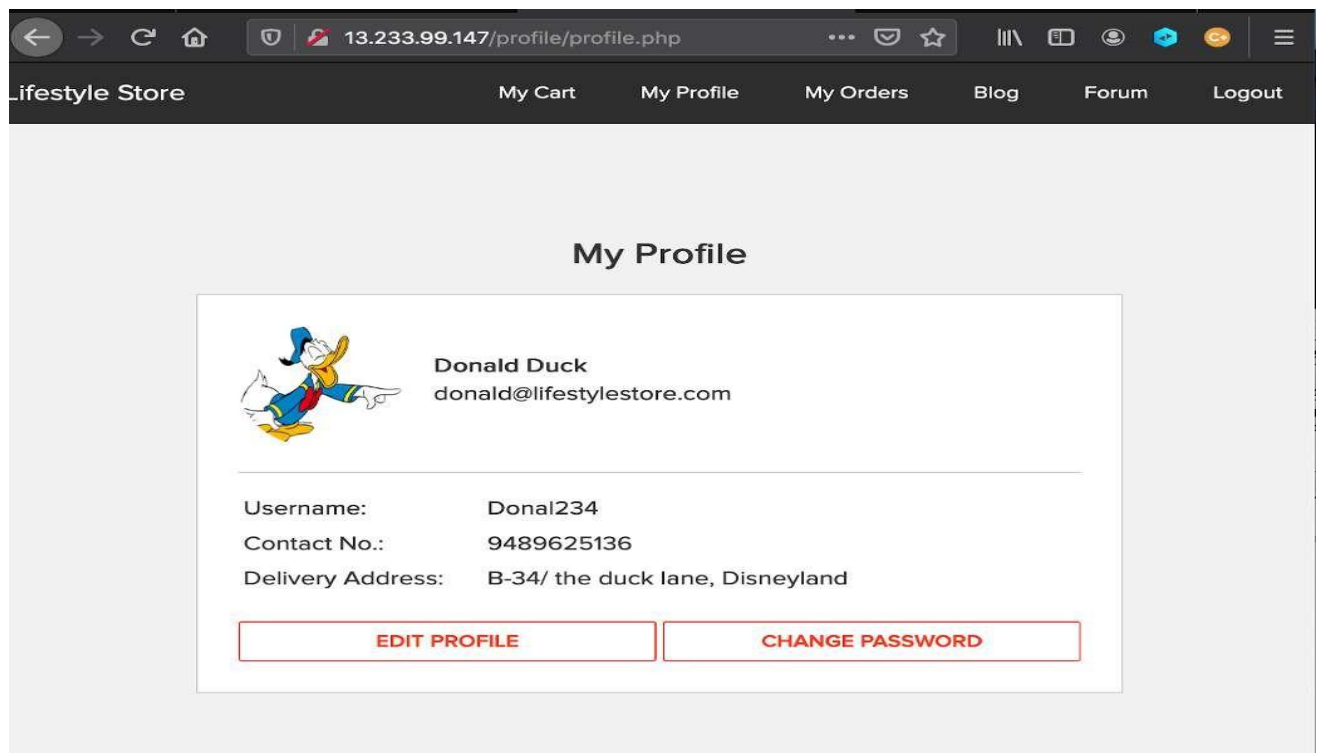


Fig. 4.2 Screenshot of E-Commerce Website

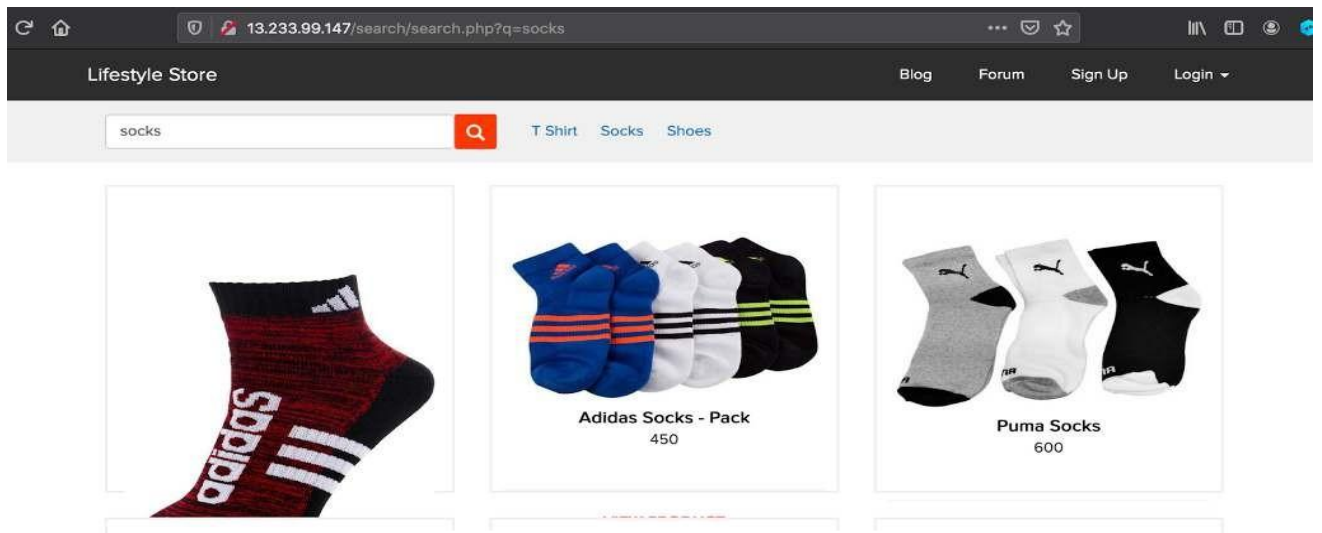


Fig. 4.3 Screenshot of E-Commerce Website

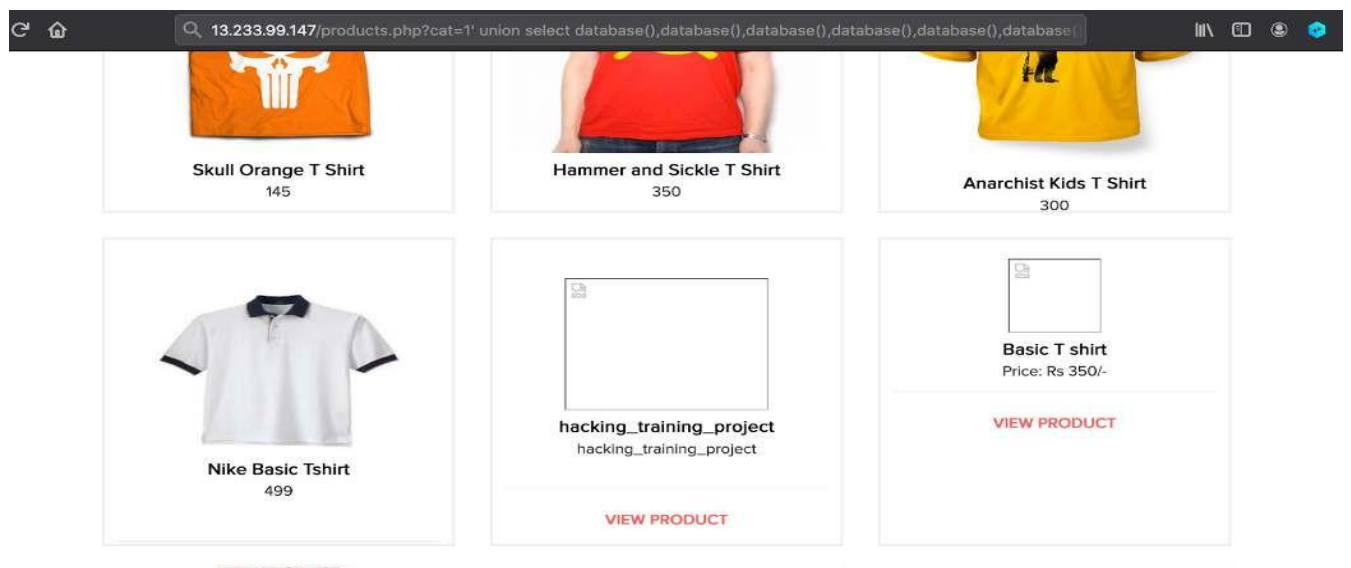


Fig. 4.4 Screenshot of E-Commerce Website



Fig. 4.5 Screenshot of E-Commerce Website

# **CHAPTER: 5**

## **DEVELOPER'S REPORT E-COMMERCE**

### **WEBSITELIFESTYLE STORE DETAILED**

#### **PROJECT REPORT**

#### **Security Status – Extremely Vulnerable**

1. Hacker can steal all records in Internshala databases (SQLi)
2. Hacker can take control of complete server including View, Add, Edit, Delete files and folders (Shell Upload)
3. Hacker can change source code of application to host malware, phishing pages or even explicit content (Shell Upload)
4. Hacker can inject client side code into applications and trick users by changing how page looks to steal information or spoil the name of Internshala (XSS)
5. Hacker can extract mobile number of all customers using Userid (IDOR)

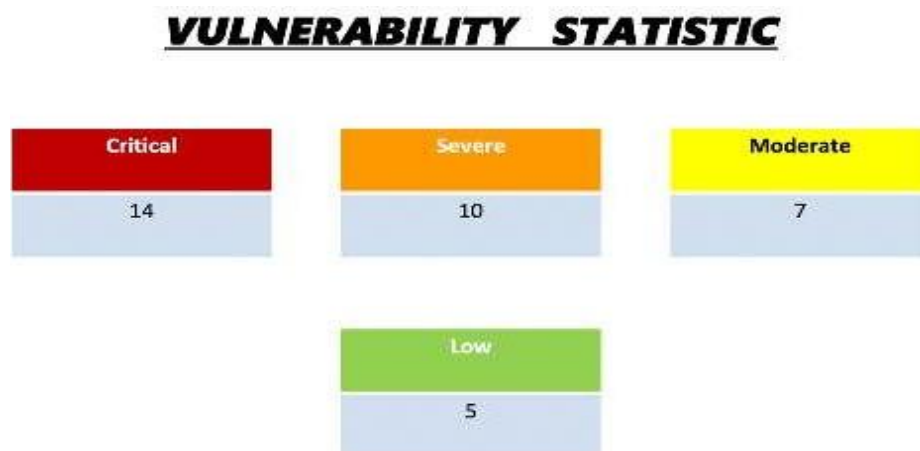


Fig. 5.1 Vulnerability Statistic

**5.1 Vulnerabilities:** A vulnerability is a weakness that can be exploited by cybercriminals to gain unauthorized access to a computer system. After exploiting a vulnerability, a cyberattack can run malicious code, install malware, and even steal sensitive data

S.NO.	SEVERITY	VULNERABILITY	COUNT
1	CRITICAL	SQL injection	3
2	CRITICAL	Access to admin panel	1
3	CRITICAL	Arbitrary file upload	2
4	CRITICAL	Account takeover by OTP bypass	1
5	CRITICAL	CSRF	3
6	SEVERE	Reflected cross site scripting	1
7	SEVERE	Stored cross site scripting	1
8	SEVERE	Common password	1
9	SEVERE	Component with known vulnerability	3
10	MODERATE	Server misconfiguration	1
11	MODERATE	Unauthorized access to user details (IDOR)	4
12	MODERATE	Directory listings	5
13	LOW	Personal Information leakage	2
14	LOW	Client side and server side validation bypass	1
15	LOW	Default error display	1
16	LOW	Open redirection	2

Fig. 5.2 List of Vulnerabilities

### ❖ Abbreviation

1. **SQL:** Structured Query Language
2. **XSS:** Cross-Site Scripting
3. **IDOR:** Insecure Direct Object Reference
4. **CSRF:** Cross-Site Request Forgery
5. **OTP:** One Time Password
6. **POC:** Proof of Concept



## 5.1.1 SQL injections

SQL Injection (Critical)	Below mentioned URL in the <b>T-shirt/socks/shoes</b> module is vulnerable to SQL injection attack Affected URL :
	• <a href="http://15.206.74.73/products.php?cat=1">http://15.206.74.73/products.php?cat=1</a>
	Affected Parameters :
	•cat (GET parameter)
	Payload:
	•cat = 1'
Affected URL :	
• <a href="http://15.206.74.73/products.php?q=socks">http://15.206.74.73/products.php?q=socks</a>	
Affected Parameters :	
•q (GET parameter)	
Payload:	
•q='socks'	

Fig. 5.3 Screenshot of SQL injection

SQL Injection (Critical)	Here are other similar SQLi in the application
	<b>Affected URL :</b>
	<ul style="list-style-type: none"><li>• <a href="http://15.206.74.73/products.php?cat=2">http://15.206.74.73/products.php?cat=2</a></li><li>• <a href="http://15.206.74.73/products.php?cat=3">http://15.206.74.73/products.php?cat=3</a></li></ul>

Fig. 5.4 Screenshot of SQL injection

### ➤ Observation

Navigate to the T-Shirt tab where you will see a number of T-shirts. Notice the GET parameter CAT in the URL:

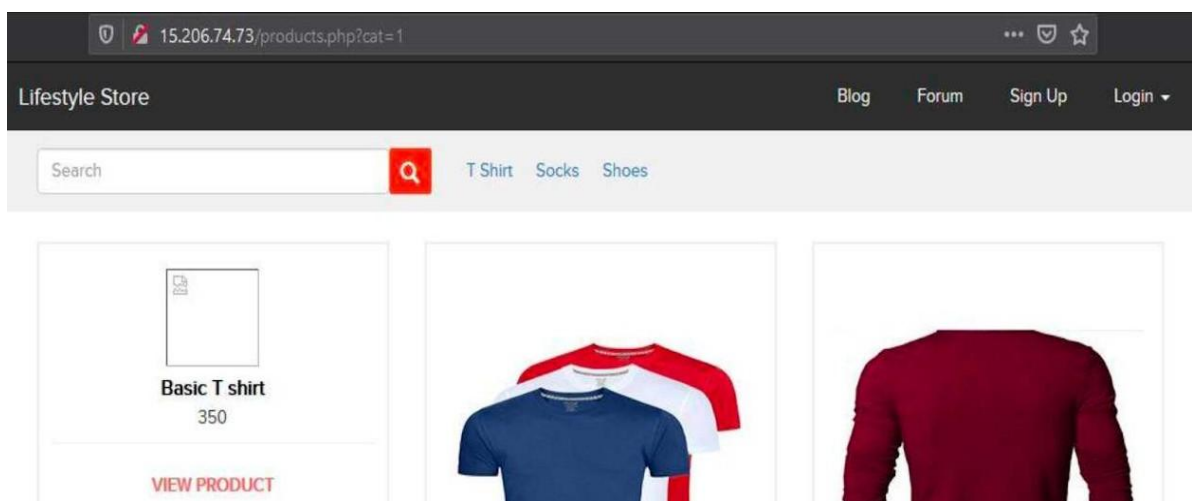


Fig. 5.5 Observation figure

We apply a single quote in cat parameter: **products.php?cat=1'** and we get complete

## ➤ SQL error:

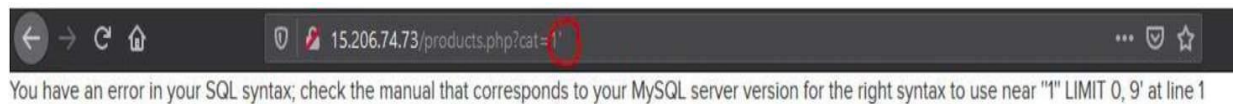


Fig. 5.6 Screenshot of SQL injection

- We then put **--+ : products.php?cat=1--+** and we error is removed confirming SQL injection.
- Now hacker can inject sql or use sqlmap to get access to the database

## ➤ POC: Attacker can attack dump arbitrary data

Database: hacking\_training\_project  
Table: users  
[15 entries]

user_name	password	phone_number	unique_key
admin	\$2y\$10\$xmldvrxSCxqdywSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTKi	8521479630	15468927955c66694cba1174.29688447
Dona1234	\$2y\$10\$PM.7nBSP5Fma1dxIM/S3s./p5xR6GTVjry/ysJtx0kBg0JURAHs0	9489625136	778522555c6669996f5a24.34991684
Pluto98	\$2y\$10\$xmldvrxSCxqdywSrDx5YSe1NAwX.7pQ2nQmaTCovH4CFssxgyJTKi	8912345670	19486318945c666a037b1432.99985767
chandan	\$2y\$10\$4cZBEIrgthXdvT1hwU11vuFELe03rR.GICdp03Njr1S0ve10KLvDa	7854126395	12404594545c666a3b49e0f8.08173871
Popeye786	\$2y\$10\$FkV1RFwTtiow0w2CaZtAQuXVnhGAUjt/IF/yTqkNpC5ZtTsVm7Eec	9745612300	18430379145c666a53af8431.79566371
Radhika	\$2y\$10\$RYXNhoyV/c4g7OtFwpqYaexvH18rF6Xxu18kt1WtrFghTutCA8Jc.	9512300052	15611262655c666b312f73e0.70827297
Nandan	\$2y\$10\$G.CRNLMElG79ZfXE1Hg.R.o95334U0xmZu4.9mqzR5614ucwnk59K	7845129630	1587354115c666b65bb44a5.36505317
MurthyAdapa	\$2y\$10\$zmzQGzD4sDSj2EunpCioe4ek18c1Abs0T2P1a1P6eV1DPR.11UubDG	8365738264	16357203785c68f640c699a2.83646347
john	\$2y\$10\$GhDB8h1x6XjPMY12GZ1VDO7Y3en9/u1/.oxtZLMYgB6F18FBgecvG	6598325015	9946437385c6a435f76bef0.14673944
bob	\$2y\$10\$KtU1kn3HPFbuxTK751LNUrxzq0LX3eMgy0/Ux16J0oc57dcGKLq	8576308560	4305822125c6a43ec307df0.68309267
jack	\$2y\$10\$z2pyN1KrJ76m9ItmZ4N5loerXyG6kqi9N/UBcJu5Ze07eM7N4pTHu	9848478231	15257114565c6a444692b707.17903432
bul1a	\$2y\$10\$HT5oIRMetqaZ7xGZPE9s2.Mk1yF4PnyDJHCWbm2w/xuKpJEEI/zjG	7645835473	18292501185c6a4493a5ddb0.87138000
hunter	\$2y\$10\$pb3U9iFxbBgSb12AkBpiEeIBdh1YfW9y.xv23q12gGbMCyn7N3g2	9788777777	13824560345c80704e821145.26019698
asd	\$2y\$10\$At5pFZnRwpjCD/yNnJWDL.L3Cc4Cv0w8Q/WEHmwZBFqVikBQfPCF2	9876543210	8057400125c862a7f5916c9.06111587
acdc	\$2y\$10\$550B78.gpuculTwpHwbcPedYcain.Yt.tsTLyqtK17FzdsplIRRBt	9999999999	13104802695c86f43f0c3705.77019309

Fig. 5.6 POC

### No of databases: 2

1. information\_schema
2. hacking\_training\_project

### No of tables : 10

3. brands
4. cart\_items
5. categories customers
6. order\_items
7. orders product\_reviews
8. products sellers
9. user

## ➤ Business Impact 3 Extremely High

Using this vulnerability, an attacker can execute arbitrary SQL commands on a Lifestyle store server and gain complete access to internal databases along with all internal user data inside it.

Previous slide has the screenshot of users table which shows user credentials being leaked that too in plain text without any hashing/encryption.

Attackers 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.

## ➤ **RECOMMENDATIONS**

1. Use whitelists, not blacklists
2. Don't trust any user input
3. Adopt the latest technologies
4. Ensure Errors are Not User-Facing
5. Disable/remove default accounts,  
passwords and databases

## 5.1.2 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://13.126.196.134/wondercms/loginURL>

Fig. 5.8 Access to admin panel

### ➤ Observation

1. When we navigate to <http://13.126.196.134/wondercms/url>
2. we get the password on the page and login as: admin in the url <http://13.126.196.134/wondercms/loginURL>

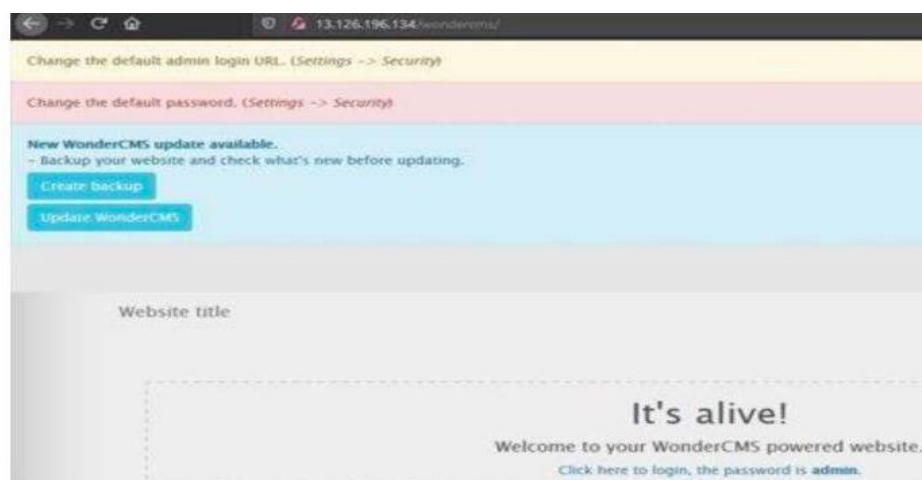


Fig. 5.9 Observation

### ➤ Proof of Concept (PoC)

Hackers can change the admin password . Hackers can also add and delete pages. Hackers can upload any malicious file.

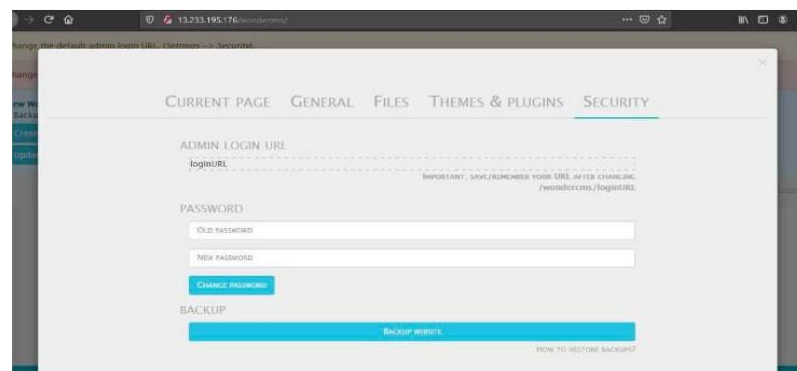


Fig. 5.10 POC

➤ **Business impact - Extremely High**

1. Hacker can do anything with the page, he will have full access to the page and can govern the page according to its will.
2. It is a massive business risk.
3. Loss can be very high.

➤ **RECOMMENDATIONS**

1. The default password should be changed and a strong password must be set up.
2. The admin url must also be such that it's not accessible to normal users.
3. Password changing option must be done with 2 to 3 step verification.

## 5.1.3 Arbitrary file upload

<b>Arbitrary file upload (Critical)</b>	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 : <ul style="list-style-type: none"><li>•http://13.126.196.134/wondercms/Affected Parameters :</li><li>•File Upload (POST parameter)</li></ul>
	The attacker can upload files with extension other than .jpeg .
	Affected URL : <ul style="list-style-type: none"><li>•http://13.126.196.134/profile/2/edit/</li></ul> Affected Parameters : <ul style="list-style-type: none"><li>•Upload Profile Photo (POST parameter)</li></ul>

Fig. 5.11 Arbitrary file upload

### ➤ Observations:

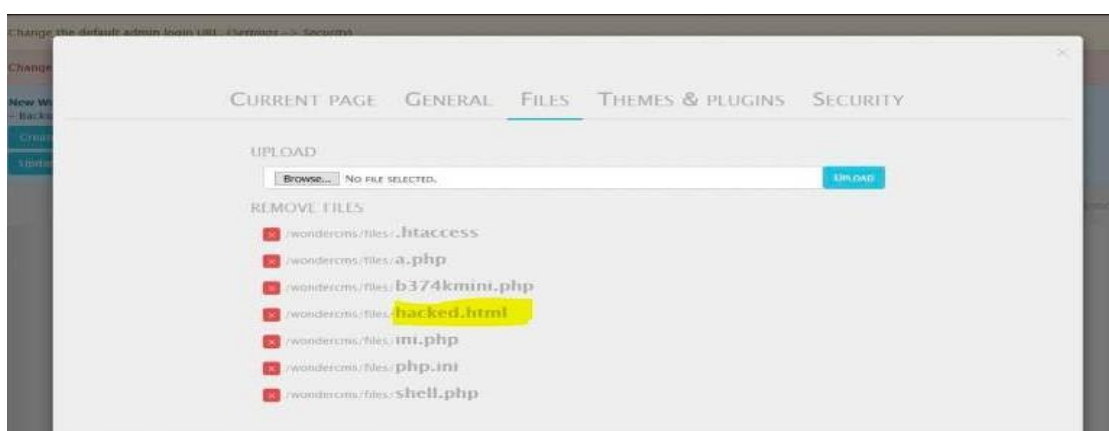


Fig. 5.12 Observation

### ➤ Proof of concept

1. Weak password - admin.
2. Arbitrary File Inclusion.

### ➤ Business Impact 3 Extremely High

A malicious user can access the Dashboard which discloses many critical information of an organisation including: Important files, Passwords, and much more...

Any backdoor file or shell can be uploaded to get access to the uploaded file on a 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**

Take the following precautions:

1. Use a strong password 8 character or more in length with alphanumerics and symbols.
2. It should not contain personal/guessable information.
3. Do not reuse passwords.
4. Disable default accounts and users.
5. Change All Passwords To Strong Unique Passwords.

# 5.1.4 Account takeover using OTP bypass

Account Takeover Using OTP Bypass (Critical)

The below mentioned login page allows login via OTP which can be bruteforced

**Affected URL :**

- [http://13.126.196.134/reset\\_password/admin.php?otp=](http://13.126.196.134/reset_password/admin.php?otp=)

**Affected Parameters :**

- OTP (POST parameters)

Fig. 5.13 Arbitrary file upload

➤ **Observation**

1. Navigate to [http://13.126.196.134/reset\\_password/admin.php?otp=](http://13.126.196.134/reset_password/admin.php?otp=) and You will see the user loginpage via OTP.

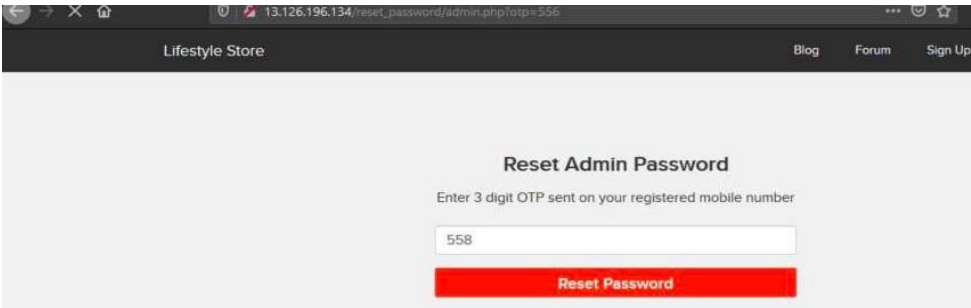


Fig. 5.14 Observation

2. Following request will be generated containing OTP parameters.
3. Now We're Brute Forcing It.



Fig. 5.15 Brute force attack

4. And we easily got the valid otp.

Filter: Showing all items						
Request	Payload	Status	Error	Timeout	Length	Comment
4	103	200	<input type="checkbox"/>	<input type="checkbox"/>	4476	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
1	100	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
2	101	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
3	102	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
5	104	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
6	105	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
7	106	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
8	107	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
9	108	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
10	109	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
11	110	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
12	111	200	<input type="checkbox"/>	<input type="checkbox"/>	4380	
...	...	...	<input type="checkbox"/>	<input type="checkbox"/>	4380	

Fig. 5.16 OTP found

## ➤ POC

- Now a hacker can change the password of the admin dashboard.

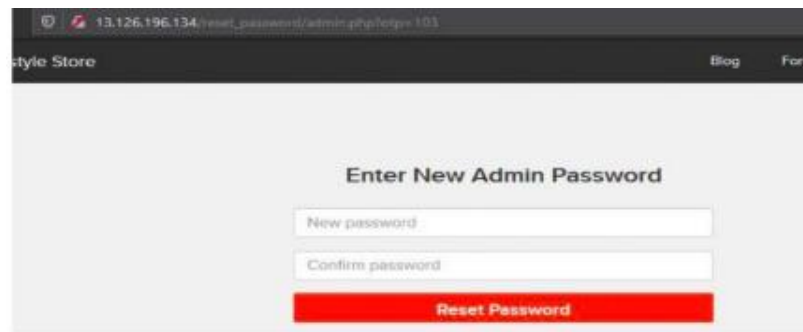


Fig. 5.17 POC

## ➤ Business Impact 3 Extremely High

A malicious hacker can gain complete access to any account just by brute forcing the otp. This leads to complete compromise of personal user data of every customer.

Attackers once logged in can then carry out actions on behalf of the victim which could lead to serious financial loss to him/her.

## ➤ Recommendation

Take the following precautions:

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

## 5.1.5 CSRF

### ➤ Observation:

1. Here you can see a 7 digit password ,but due to CSRF I'll change the password at the moment he wants to update.
2. Here's the file I opened while changing password , when we click on send the password will change to 12345.

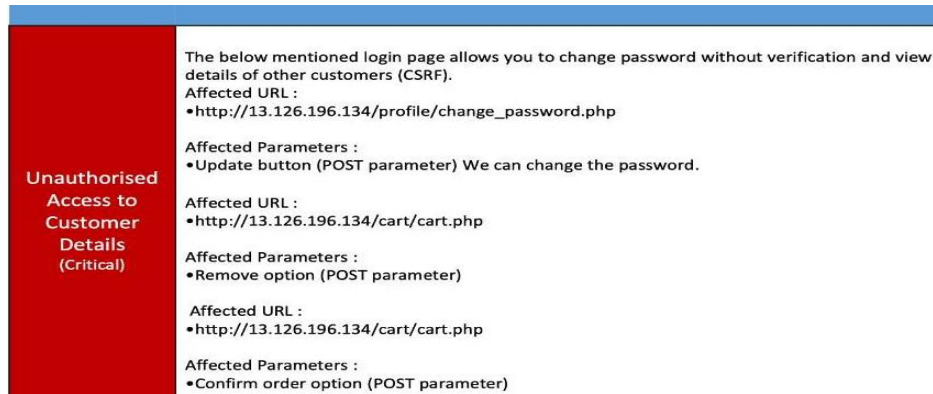


Fig. 5.18 Observation

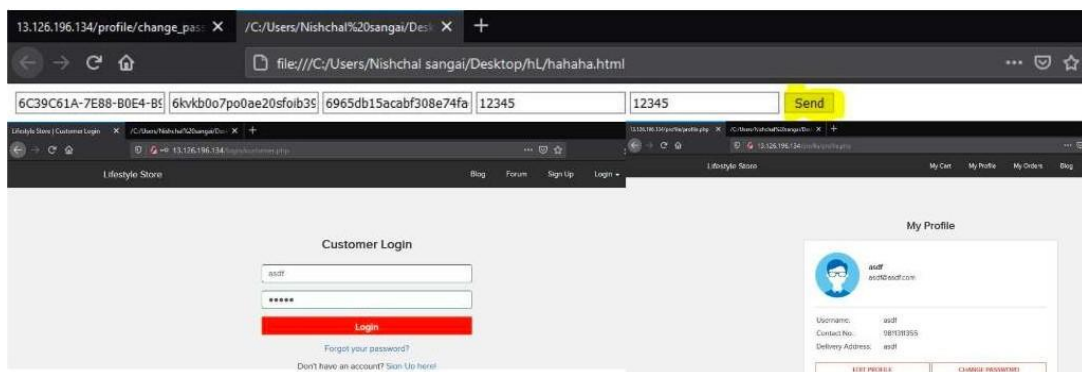


Fig. 5.19 Observation

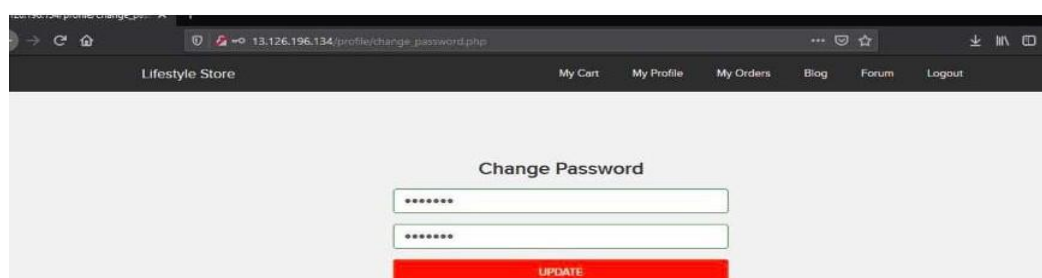


Fig. 5.20 Observation

### ➤ POC: Here's the code generated by burp suite community edition.

```
1 <!DOCTYPE html>
2 <html>
3 <!-- CSRF PoC - generated by Burp Suite i0 SecLab plugin -->
4 <body>
5 <form method="POST" action="http://13.126.196.134:80/profile/change_password_submit.php">
6 <input type="text" name="key" value="6C39C61A-7E88-B0E4-B9D5-FC7E8B773CB1">
7 <input type="text" name="PHPSESSID" value="6kvkb0o7po0ae20sfoib398mn4">
8 <input type="text" name="X-XSRF-TOKEN" value="6965db15acabf308e74fa61bde40c623856201cbfe80ff1f28178fa5f13b28f3">
9 <input type="text" name="password" value="12345">
10 <input type="text" name="password_confirm" value="12345">
11 <input type="submit" value="Send">
12 </form>
13 </body>
14 </html>
```

Fig. 5.21 POC



➤ **Observation:** CSRF in cart

Here you can see, the order is placed unwantedly by the user through CSRF

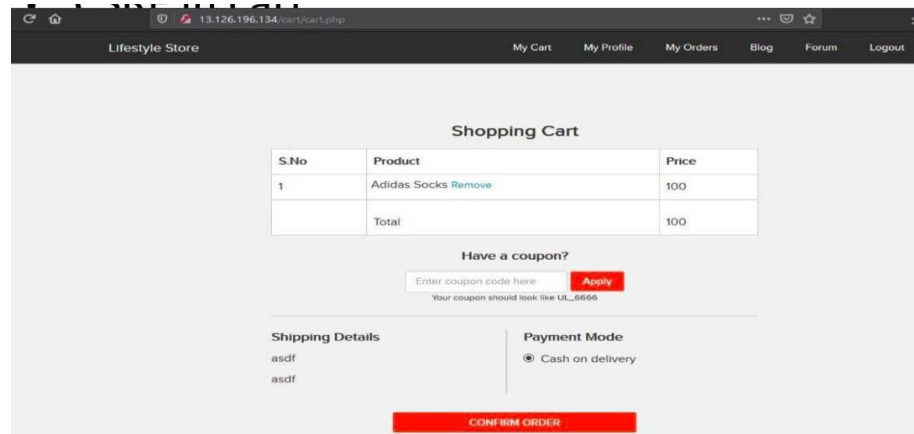


Fig. 5.22 Observation

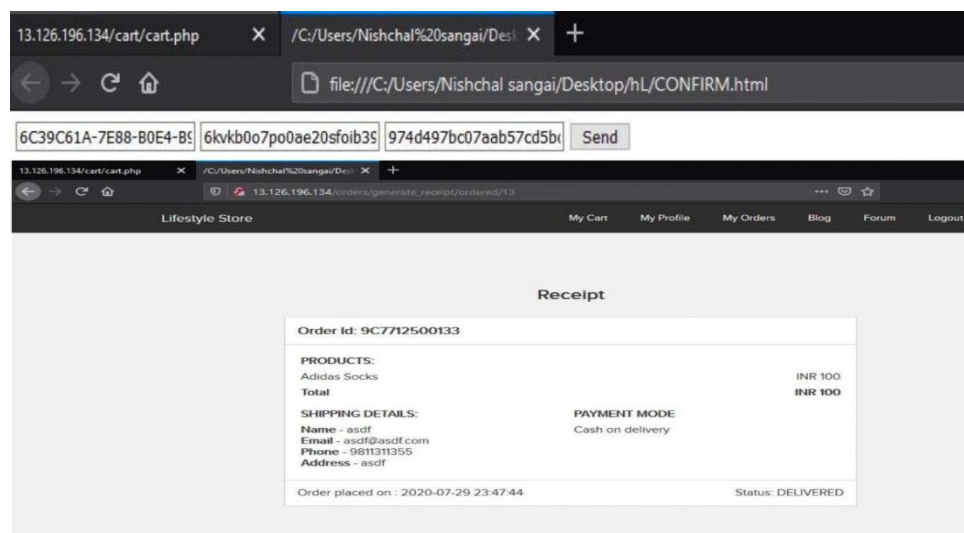


Fig. 5.23 Observation

➤ **Business Impact 3 Very High**

1. Hackers can change the password of any user .
2. Hackers can make users do unwanted things.
3. It makes a very bad impact on the website in front of the user.
4. Hackers can remove and confirm orders in the cart of the user.

➤ **Recommendations:** Take the following precautions:

1. Implement an Anti-CSRF Token.
2. Do not show the customers of the month on the login page.
3. Use the Same Site Flag in Cookies.
4. Check the source of the request made.
5. Take some extra keys or tokens from the user before processing an important request.

## 5.1.6 Reflected Cross Site Scripting (XSS)

Reflected Cross Site Scripting (Severe)	Below mentioned parameters are vulnerable to reflected XSS
	<b>Affected URL :</b> <ul style="list-style-type: none"><li>• <a href="http://13.126.196.134/profile/16/edit/">http://13.126.196.134/profile/16/edit/</a></li></ul>
	<b>Affected Parameters :</b> <ul style="list-style-type: none"><li>• address(POST parameters)</li></ul>
	<b>Payload:</b> <ul style="list-style-type: none"><li>• <code>&lt;script&gt;alert(1)&lt;/script&gt;</code></li></ul>

Fig. 5.24 Reflected Cross Site Scripting (XSS)

### ➤ Observation

Open edit profile through URL and write a script on the address bar.

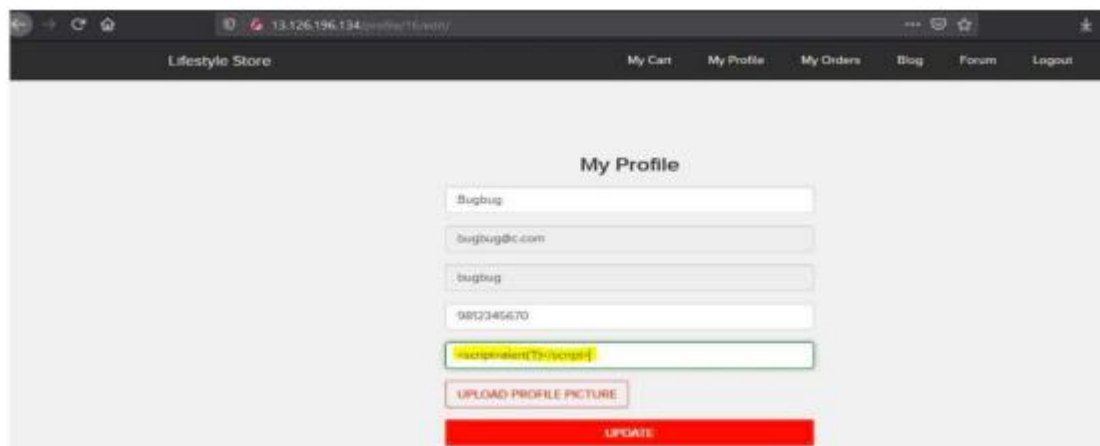


Fig. 5.25 Observation

### ➤ POC

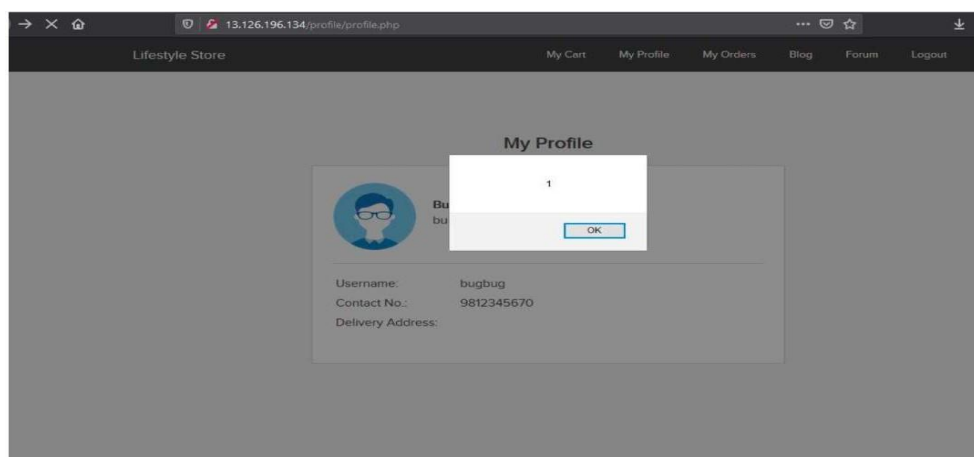


Fig. 5.26 POC

### ➤ Business impact - High

As an attacker can inject arbitrary HTML CSS and JS via the URL, the attacker can put any content on the page like phishing pages, install malware on the victim's device and even host

explicit content that could compromise the reputation of the organisation.

All an attacker needs to do is send the link with the payload to the victim and the 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:

1. Sanitise all user input and block characters you do not want.
2. Convert special HTML characters like < > into HTML entities &lt; &gt; before printing them on the website.

## 5.1.7 Stored Cross Site Scripting (XSS)

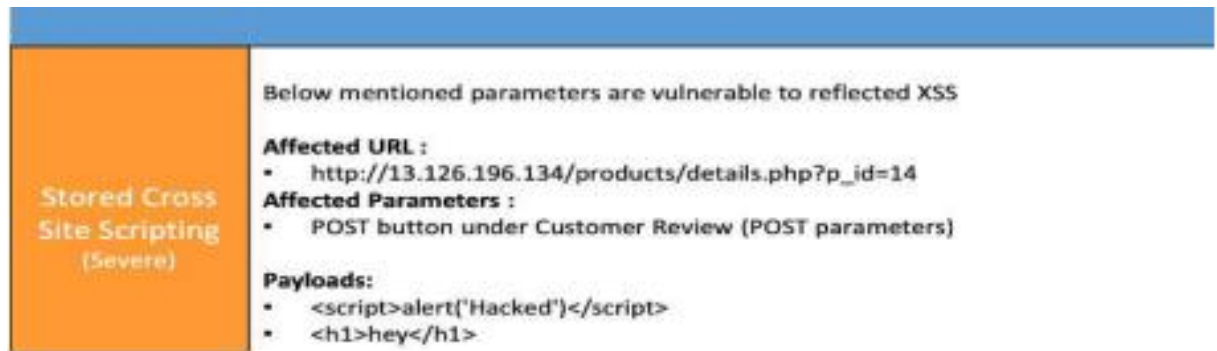


Fig. 5.27 Stored Cross Site Scripting (XSS)

➤ **Observations:** Now try entering the payload in the review box.

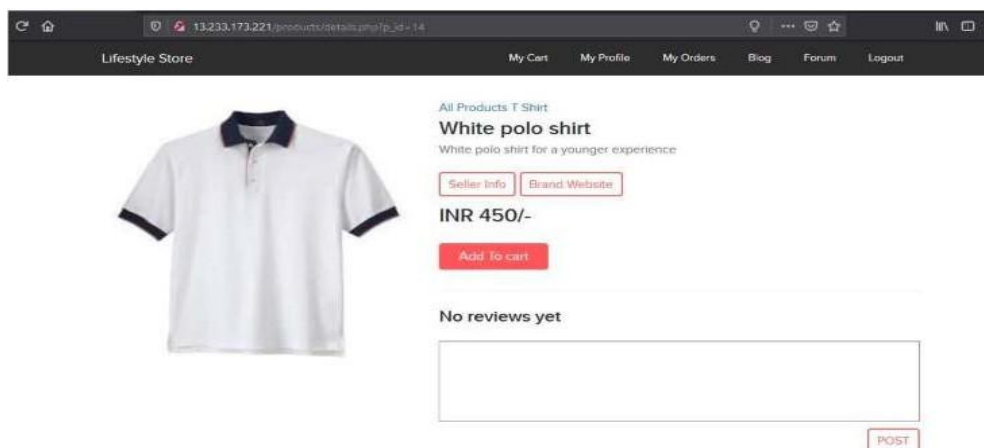


Fig. 5.28 Observation

Hit the post button , you can see stored XSS or permanent XSS.

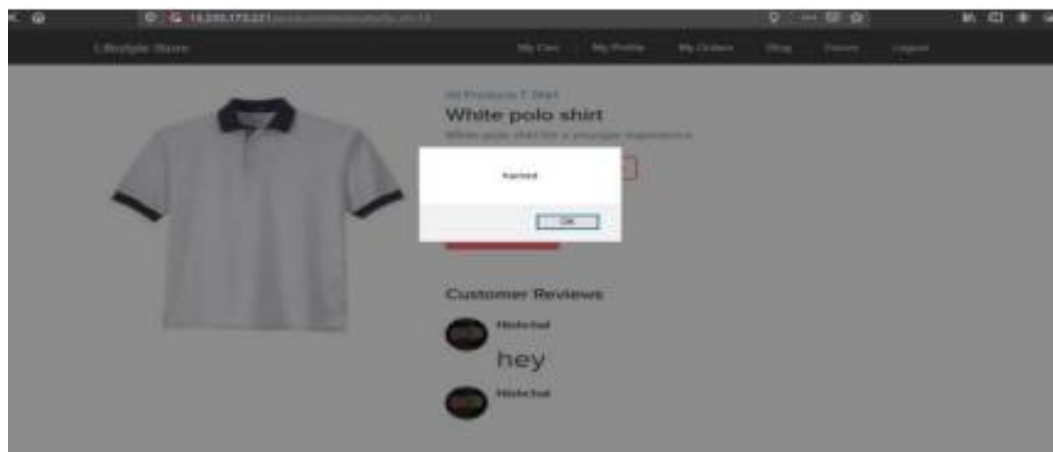


Fig. 5.29 Observation

### ➤ **Business impact - High**

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

All an attacker needs to do is send the link with the payload to the victim

and trusts the website, he/she will trust the content.

➤ **Recommendation:** Take the following precautions:

1. " Sanitize all user input and block characters you do not want.
2. " Convert special HTML characters like < > into HTML entities &lt; &gt; before printing them on the website.

## 5.1.8 COMMON PASSWORD

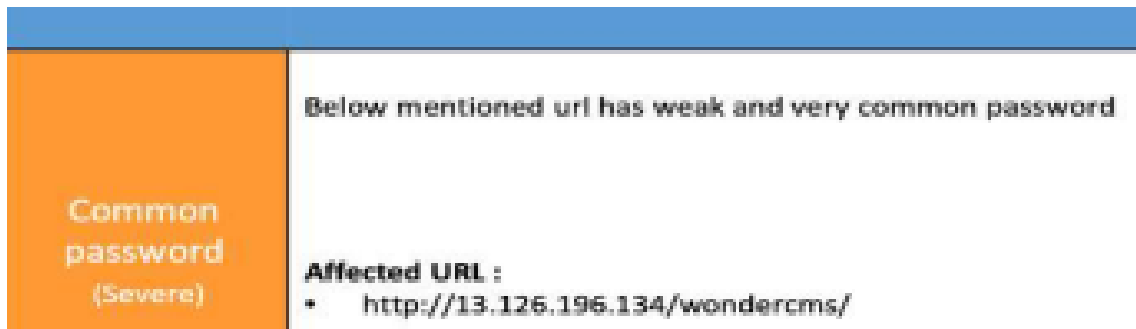


Fig. 5.30 Common Password

### ➤ Observation

- Password is right in front of you

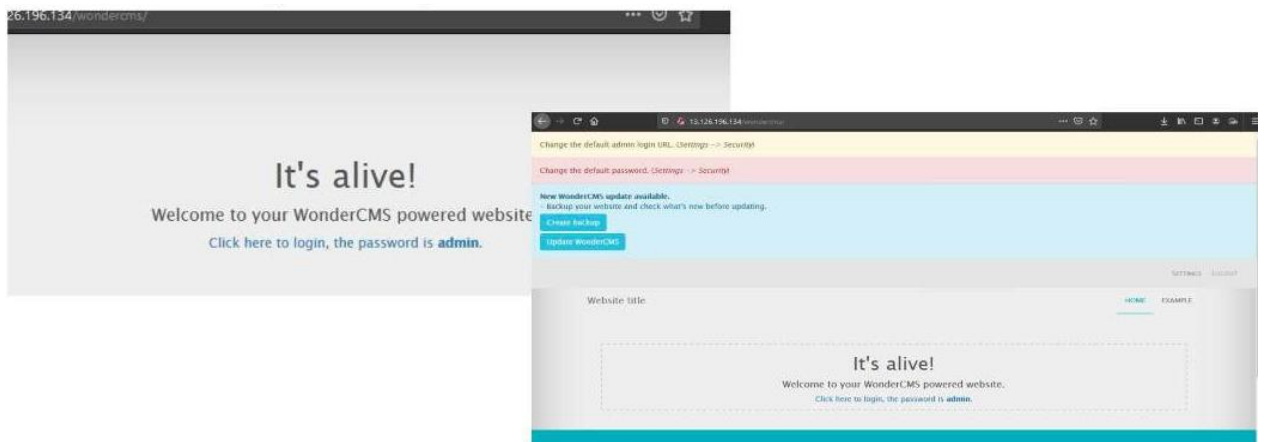


Fig. 5.31 Observation

### ➤ Business Impact 3 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

1. There should be password strength check at every creation of an account.
2. There must be a minimum of 8 characters long password
3. with a mixture of numbers ,alphanumerics, special characters ,etc.
4. There should be no repetition of password ,neither on change nor reset.

## 5.1.9 Component with known vulnerability

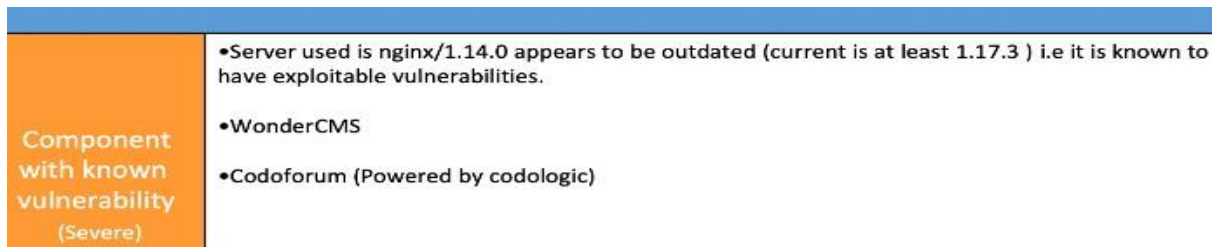


Fig. 5.32 Component with known vulnerability

### ➤ Observation

- Codologic vulnerability: Now you can see that they have blind SQL injection vulnerability.

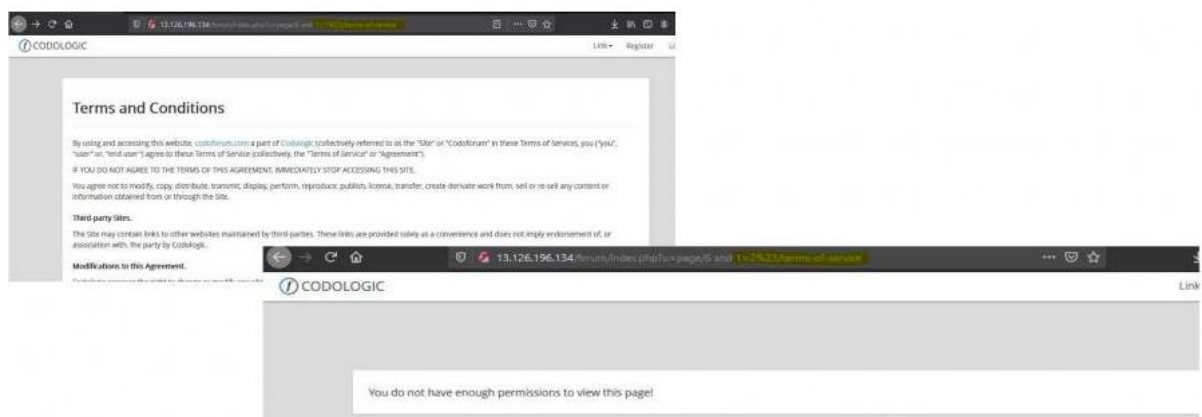


Fig. 5.33 Observation

### ➤ POC

- Codologic vulnerability, it has multiple sql injection vulnerability check the link of exploit-db in reference.

```
Proof of Concept:

http://localhost/codoforum/index.php?u=page/6 and
1=1%23/terms-of-service
-> true (terms and services displayed)
http://localhost/codoforum/index.php?u=page/6 and
1=2%23/terms-of-service
-> false ("You do not have enough permissions to view this page!")

Code:

routes.php:593

$pid = (int) $id;
$user = \CODOF\User\User::get();

$query = 'SELECT title, content FROM ' . PREFIX . 'codo_pages p '
        . ' LEFT JOIN ' . PREFIX . 'codo_page_roles r ON
r.pid=p.id '
        . ' WHERE (r.rid IS NULL OR (r.rid IS NOT NULL AND
r.rid IN (' . implode($user->rids) . ')))'
        . ' AND p.id=' . $id;
```

Fig. 5.34 POC

➤ **Business Impact 3 high**

Exploits of every vulnerability detected are 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.

➤ **Recommendations:**

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



## 5.1.10 Server misconfiguration

Server misconfiguration (Moderate)

Below mentioned url will show you the server related info

**URL**

<http://13.126.196.134/server-status>

<http://13.126.196.134/server-info>

Fig. 5.35 Server misconfiguration

### ➤ Observations and POC:

**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

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
-----	-----	-----	---	-----	----	-----	------	-------	------	--------	-------	---------

Fig. 5.36 Observation and POC

### ➤ Recommendation:

1. Keep the software up to date.
2. Disable all the default accounts and change passwords regularly.
3. Develop strong app architecture and encrypt data which has sensitive information.
4. Make sure that the security settings in the framework and libraries are set to secured values.
5. Perform regular audits and run tools to identify the holes in the system.

## **Conclusion**

We were successfully able to find all the vulnerabilities and discussed their impacts and solutions. Hence completing the project along with the report. As we say above, some vulnerabilities were very harmful for the website, whereas, some were moderately harmful. Learning ethical hacking helped us identify them and solve them. I therefore find this skill very helpful and hope to work with it in the future.

## **Reference**

- [https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection)
- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- [https://www.owasp.org/index.php/Default\\_Passwords](https://www.owasp.org/index.php/Default_Passwords)
- <https://www.us-cert.gov/ncas/alerts/TA13-175A>
- [https://www.owasp.org/index.php/Testing\\_for\\_weak\\_password\\_change\\_or\\_reset\\_functionalities\\_\(OTG-AUTHN-009\)](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.owasp.org/index.php/Default_Passwords)
- <https://www.us-cert.gov/ncas/alerts/TA13-175A>
- [https://www.owasp.org/index.php/Testing\\_Multiple\\_Factors\\_Authentication\\_\(OWASP-AT-009\)](https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009))
- [https://www.owasp.org/index.php/Blocking\\_Brute\\_Force\\_Att](https://www.owasp.org/index.php/Blocking_Brute_Force_Att)
- [https://www.owasp.org/index.php/Cross\\_site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross_site_Scripting_(XSS))
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- [https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp)
- <https://www.acunetix.com/blog/articles/weak-password-vulnerability-common-think/>
- [https://www.owasp.org/index.php/Testing\\_for\\_Weak\\_password\\_policy\\_\(OTG-AUTHN-007\)](https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007))
- <https://usn.ubuntu.com/4099-1/> (for ubuntu) and <https://www.exploit-db.com/exploits/37820>
- <https://securitywarrior9.blogspot.com/2018/01/vulnerability-in-wondercms-leading-to.html>
- <https://www.ifourtechnolab.com/blog/owasp-vulnerability-security-misconfiguration>
- <https://tools.kali.org/web-applications/dirbuster>
- <https://tools.kali.org/information-gathering/masscan>
- <https://www.javatpoint.com/ethical-hacking>
- <https://www.hackingarticles.in/comprehensive-guide-on-crunch-tool/>